

Netkiller Database 手札

目录

自述

1. 写给读者
2. 作者简介
3. 如何获得文档
4. 打赏 (Donations)
5. 联系方式

I. Database Relational

1. DB-Engines - DB-Engines Ranking
2. SysBench
3. SuperSmack
4. ETL (Extract-Transform-Load)
 1. Kettle
 - 1.1. 安装
Mac 安装
 2. suro
5. 数据迁移
 1. Apache Sqoop
6. GreenSQL
7. Database design & E-R diagram
 1. opensource database design tools
 2. OpenSystemArchitect
 3. SQL Power Architect

II. Search Engine

8. Elasticsearch
 1. 安装 Elasticsearch
 - 1.1. 6.x 安装
 - 1.2. 单机模式 (适用于开发环境) 5.x
 - 1.3. Elasticsearch Cluster 5.x
负载均衡配置

- 1.4. docker-compose 安装
- 1.5. Kubernetes
- 1.6. 安装指定版本的 Elasticsearch
- 1.7. Plugin
 - elasticsearch-analysis-ik
 - elasticsearch-analysis-pinyin
- 1.8. Elastic Cloud on Kubernetes
- 1.9. 发现配置
 - 即将废弃
- 2. 文档API
 - 2.1. 快速上手
 - 2.2. 写入 PUT/POST
 - 2.3. 获取 GET
 - _source
 - 2.4. 检查记录是否存在
 - 2.5. 删除 Delete
 - 2.6. 参数
 - pretty 格式化 json
- 3. 搜索
 - 3.1. URL 搜索
 - 3.2. 分页
- 4. Query DSL
 - 4.1. match 匹配
 - 4.2. multi_match 多字段匹配
 - 4.3. Query bool 布尔条件
 - must
 - should
 - must_not
 - 4.4. filter 过滤
 - 4.5. sort 排序
 - 4.6. _source
 - 4.7. highlight 高亮处理
- 5. 集群管理
 - 5.1. 节点健康状态
 - 5.2. 节点http状态
 - 5.3. 查看master节点

- 5.4. 查看索引的节点分布
- 5.5. 索引的开启与关闭
 - _open
 - _close
- 6. 中文分词插件管理
 - 6.1. 通过 elasticsearch-plugin 命令安装分词插件
 - 6.2. 手工安装插件
 - 6.3. 创建索引
 - 6.4. 删除索引
 - 6.5. 配置索引分词插件
 - 测试分词效果
- 7. 索引管理
 - 7.1. 查看索引
 - 7.2. 删除索引
- 8. 映射
 - 8.1. 查看 _mapping
 - 8.2. 删除 _mapping
 - 8.3. 创建 _mapping
 - 8.4. 更新 mapping
 - 8.5. 修改 _mapping
 - 8.6. 数据类型
 - date
- 9. Alias management 别名管理
 - 9.1. 查看索引别名
 - 9.2. 创建索引别名
 - 9.3. 修改别名
 - 9.4. 删除别名
- 10. Example
 - 10.1. 新闻资讯应用案例
 - 10.2. 文章搜索案例
- 11. Migrating MySQL Data into Elasticsearch using logstash
 - 11.1. 安装 logstash
 - 11.2. 配置 logstash
 - 11.3. 启动 Logstash
 - 11.4. 验证

11.5. 配置模板

- 全量导入

- 多表导入

- 通过 ID 主键字段增量复制数据

- 通过日期字段增量复制数据

- 指定SQL文件

- 参数传递

- 控制返回JDBC数据量

- 输出到不同的 Elasticsearch 中

- 日期格式转换

- example

11.6. 解决数据不对称问题

11.7. 修改 Mapping

12. ElasticHD

13. 安装 Elasticsearch 2.3

- 13.1. RPM 安装

- 13.2. YUM 安装

- 13.3. 测试安装是否正常

- 13.4. Plugin 插件管理

 - 手工安装插件

 - plugin 命令

 - 插件测试

14. FAQ

- 14.1. Plugin [analysis-ik] is incompatible with Elasticsearch [2.3.5]. Was designed for version [2.3.4]

- 14.2. plugin [analysis-ik] is incompatible with version [5.6.1]; was designed for version [5.5.2]

- 14.3. mapper_parsing_exception: failed to parse [ctime]

- 14.4. 配置 JAVA_HOME

- 14.5. memory locking requested for elasticsearch process but memory is not locked

9. Solr

- 1. 安装

- 2. Core Admin

- 2.1. Schema
- 3. 索引
 - 3.1. Indexing a directory of "rich" files
 - 3.2. Indexing Solr XML
 - 3.3. Indexing JSON
 - 3.4. Deleting Data
- 4. 接口
 - 4.1. 查询
- 5. FAQ
 - 5.1. NOTE: Please install Isof as this script needs it to determine if Solr is listening on port 8983.
- 6. Solr 1.3.0
 - 6.1. Embedded Jetty
 - 6.2. Jetty
 - 6.3. Tomcat
 - 6.4. solr-php-client
 - 6.5. multicore
 - 6.6. 中文分词
 - ChineseTokenizerFactory
 - CJK
 - mmseg4j
 - 中文分词“庖丁解牛” Paoding Analysis
- 10. Nutch
- 11. Sphinx
- 12. Lucene
- 13. MG4J
- 14. PhpDig
- 15. Mahout
- 16. Apache Hbase
 - 1. 安装 Apache Hbase
 - 1.1. 单机模式安装
 - 1.2. 伪分布式模式
 - 1.3. 分布式模式部署
 - 1.4. 运维技巧
 - 2. 配置 Apache Hbase
 - 2.1. hbase-env.sh

- 2.2. hbase-site.xml
- 3. Hbase Shell
 - 3.1. 表操作
- 4. Web UI
- 5. Phoenix
 - 5.1. 安装 Phoenix
 - 5.2. sqlline.py 命令行界面
 - 帮助信息
 - 创建表
 - 5.3. SQuirreL SQL Client
- 6. FAQ
 - 6.1. HBaseConfTool : Unsupported major.minor version 51.0
 - 6.2. ignoring option PermSize=128m; support was removed in 8.0
- 17. Apache Hive
 - 1. 安装 Apache Hive
 - 1.1. MySQL
 - 1.2. Hadoop
 - 1.3. Hive
 - 1.4. 启动 Hive
 - 1.5. 访问 Hive
 - 1.6. 配置 hiveserver2
 - 2. beeline
 - 3. 管理 Hive
 - 3.1. 表管理
 - 创建表
 - 显示表
 - 删除表
 - 查看表结构
 - 为表增加字段
 - 修改表名称
 - 使用已有表结构创建新表
 - 3.2. 分区表
 - 创建分区表
 - 显示分区情况

- 增加分区
- 向分区表导入数据

3.3. 视图管理

- 创建视图

- 删除视图

3.4. 数据管理

- 从文本文件导入数据

- 从其他表查询数据并创建新表

- 从其他表查询数据然后插入指定表中

- 从现有表中查询数据然后插入到新的分区表中

3.5. HDFS与本地文件系统管理

- HDFS 目录迁移

- 导出表数据到本地文件

- 导出到HDFS

4. HiveQL - Hive查询语言

- 4.1. JOIN 连接查询

- 4.2. 子查询

5. FAQ

- 5.1. (java.io.IOException: java.security.authorize.AuthorizationException):

- User: hadoop is not allowed to impersonate

- anonymous

18. Apache Sqoop

- 1. 安装 Sqoop

- 2. sqoop2-tool

- 2.1. verify

- 2.2. upgrade

- 3. sqoop2-shell

- 3.1. show version

- 3.2. set

- server

- 要设置可查看具体出错信息

- 3.3. show connector

- 3.4. link

- hdfs-connector

- generic-jdbc-connector

- 3.5. job
 - create job
 - show job
 - start job
 - status job

- 3.6. update
 - link

4. FAQ

- 4.1. Unable to load native-hadoop library for your platform

III. MySQL

19. MySQL Server

1. MySQL Installation

1.1. CentOS

- CentOS 8 Stream + dnf 安装 Mysql
- CentOS 6.2 + MySQL 5.5.25 (RPM)
- MySQL 8.0

1.2. Docker

1.3. Ubuntu/Debian

- mysql-5.5.21-debian6.0-i686.deb

1.4. 源码安装

1.5. 二进制版本

1.6. Installing MySQL on Linux Using the MySQL Yum Repository

- MySQL 5.6

- MySQL 5.7

1.7. Firewall

1.8. Mac OS

1.9. MariaDB

- CentOS 6 YUM 安装 MariaDB

- CentOS 7 安装 MariaDB

1.10. Percona

- Percona yum Repository

- Percona XtraBackup

- 安装 XtraBackup

- innobackupex

备份数据库
恢复数据库

xbstream

xtrabackup

Percona Toolkit - MySQL Management
Software

2. MySQL Plugin

2.1. validate_password

2.2. MySQL Images manager

2.3. MySQL fifo

2.4. 内容输出到文本插件

3. Replication

3.1. Master Slave

Master

Slave

Testing

将现有数据库迁移到主从结构数据库

主从复制安全问题

3.2. Master Master(主主)

Master A

Master B

将Master A 数据库 同步到 Master B 两端数据
库内容保持一致

Master A - B 同步两端数据库

Master A 数据库解除只读权限

查看主主的工作状态

3.3. Semisynchronous Replication

Master

Slave 配置

卸载插件

my.cnf

3.4. multi-master replication

3.5. multi-source replication

3.6. 与复制有关的问题

主从不同步问题

mysql-bin 清理问题

- 跳过 Last_Errno
 - 重置Slave
- 3.7. GTID
 - Master
 - Slave
- 4. MySQL Cluster
 - 4.1. Management node (MGM node)
 - 4.2. Data node
 - 4.3. SQL node
 - 4.4. Starting
 - 4.5. Shutdown
 - 4.6. Testing
- 5. MySQL Proxy
 - 5.1. Ubuntu
 - 5.2. CentOS
 - FAQ
- 6. MySQL Router
 - 6.1. 安装 MySQL Router
 - 6.2. 配置 MySQL Router
 - 主备配置
 - 负载均衡配置
 - 6.3. MySQL Router , Haproxy, LVS 的选择
- 7. my.cnf
 - 7.1. bind-address
 - 7.2. 禁用TCP/IP链接
 - 7.3. 配置字符集
 - 7.4. 最大链接数 max_connections
 - 7.5. 默认引擎 storage-engine
 - 7.6. max_allowed_packet
 - 7.7. skip-name-resolve
 - 7.8. timeout
 - 7.9. 与复制有关的参数
 - 用于主库的选项 Master
 - 用于从库的选项 Slave
 - 逃过错误
 - 7.10. 与 InnoDB 有关的配置项

- 7.11. EVENT 设置
- 7.12. 日志
- 7.13. MySQL 5.7 my.cnf 实例
- 7.14. Example for my.cnf
- 8. variables
 - 8.1. time_zone
 - 8.2. sql_mode
 - 设置 sql_mode
 - 查看 sql_mode
 - 兼容早期 MySQL 版本
 - 5.7.16
 - 8.3. wait_timeout
 - 8.4. table_lock_wait_timeout
 - 8.5. low_priority_updates
 - 8.6. collation_server
 - 8.7. character_set
 - 8.8. datadir
 - 8.9. plugin_dir
 - 8.10. storage_engine
 - 8.11. timeout
 - 8.12. max_connections
 - 8.13. 自动提交 autocommit
- 9. Monitoring
 - 9.1. Analysis and Optimization
 - mytop - top like query monitor for MySQL
 - mtop - MySQL terminal based query monitor
 - innotop
 - mysqlreport - A friendly report of important MySQL status values
 - mysqltuner - MySQL configuration assistant
 - 9.2. Munin
 - 9.3. Cacti
 - 9.4. Monitoring MySQL with SNMP
- 20. Client and Utility Programs
 - 1. mysql - the MySQL command-line tool
 - 1.1. ~/.my.cnf

- 1.2. 屏幕输出到文件
- 1.3. 终端编码
- 1.4. Unix Socket
- 1.5. 重定向巧用
- 1.6. --sigint-ignore 忽略 Ctrl + C
- 1.7. mysql log
2. mysqldump - a database backup program
 - 2.1. 备份数据库并压缩文件
 - 2.2. 备份数据库/表
 - 2.3. 备份到文件
 - 2.4. 备份数据库, 无结构, 只有数据
 - 2.5. 使用完整的insert插入数据
 - 2.6. --extended-insert / --skip-extended-insert
 - 2.7. --skip-lock-tables
 - 2.8. --skip-add-locks
 - 2.9. --where
 - 2.10. 注释信息--comments /--skip-comments
 - 2.11. 不导出注释信息
 - 2.12. 字符集设置
3. mysqladmin - client for administering a MySQL server
 - 3.1. reload
 - 3.2. 更改密码
 - 3.3. status
 - 3.4. process list
4. myisamchk — MyISAM Table-Maintenance Utility
5. mysqlcheck — A Table Maintenance and Repair Program
6. mysqlslap - load emulation client
7. mysqldumpslow - Parse and summarize the MySQL slow query log.
- 8.
9. MySQL慢查询日志 (Slow Query Log)
 - 9.1. MySQL 8.x
 - 9.2. MySQL 5.x

查看设置

开启慢查询日志
配置慢查询日志
删除慢查询日志

10. mysql-admin

21. 数据库管理 (Database Administration)

1. 用户管理 (User Account Management)

1.1. 创建用户

1.2. 删除用户

1.3. 修改用户名

1.4. 修改密码

2. Access Privilege System

2.1. SHOW GRANTS

2.2. show privileges

2.3. Grant privileges

2.4. Revoke privileges

2.5. Show Privileges

2.6.

MAX_QUERIES_PER_HOUR/MAX_UPDATES_P
ER_HOUR

2.7. Table Privileges

2.8. Column Privileges

3. 字符集转换

3.1. 转换 latin1 到 UTF-8

4. 重新整理AUTO_INCREMENT字段

5. 数据库内容替换

6. Kill 脚本

7. SHOW COMMAND

7.1. 查看版本

7.2. status

show status

show master status

show slave status

show plugins

7.3. show processlist

7.4. 线程的使用情况

7.5. DATABASES

- 7.6. 排序统计信息
- 7.7. Key 状态
- 7.8. FUNCTION
- 7.9. PROCEDURE
- 7.10. TRIGGERS
- 7.11. EVENTS
- 7.12. 引擎(ENGINES)
- 7.13. 字符集(Collation)
- 7.14. SHOW GRANTS
- 7.15. validate_password
- 8. Maintenance 数据库维护
 - 8.1. CHECK 检查表
 - 8.2. ANALYZE 分析表
 - 8.3. CHECKSUM
 - 8.4. OPTIMIZE 优化表
 - 8.5. REPAIR 修复
- 9. INFORMATION_SCHEMA
 - 9.1. 查询表字段
 - 9.2. 列出所有触发器
- 10. Backup and Recovery
 - 10.1. Import / Export
 - Export(Backup)
 - Import(Recovery)
 - xml
 - 备份表数据
 - source
 - 使用 mysqlhotcopy 备份 MyISAM 引擎的数据库
 - AutoMySQLBackup
 - xtrabackup - Open source backup tool for InnoDB and XtraDB.
 - Percona yum Repository
 - Creating an Incremental Backup
 - 10.2. Snapshot Backup
 - LVM Snapshot
 - Btrfs Snapshot

22. DDL - Data Definition Language

1. 数据库管理(Database)

1.1. create

1.2. drop

1.3. 修改数据库

1.4. Rename

1.5. CHARACTER

1.6. show create database

2. 表管理(Table)

2.1. 数据类型

SET 集合类型

2.2. create table ... select

2.3. modify table

2.4. TEMPORARY Table

2.5. Collate

2.6. CHARACTER

2.7. DEFAULT

AUTO_INCREMENT

TIMESTAMP NULL DEFAULT NULL ON

UPDATE

表存储位置(DATA DIRECTORY)

2.8. KEY

PRIMARY KEY

2.9. AUTO_INCREMENT 定义初始值

2.10. COMMENT

2.11. Engine 存储引擎

显示当前数据库支持引擎

切换引擎

FEDERATED

BLACKHOLE

ARCHIVE

CSV

3. Partitioning

3.1. RANGE

3.2. LIST

3.3. HASH

- LINEAR HASH
- 3.4. KEY分区
- 3.5. Subpartitioning
- 3.6. 分区管理
 - 新增分区
 - 删除分区
 - 重建分区
 - 分区维护
- 3.7. EXPLAIN PARTITIONS
- 3.8. SHOW CREATE TABLE
- 3.9. INFORMATION_SCHEMA.partitions 表
- 3.10. 分区数据操作
- 4. Index
 - 4.1. SHOW INDEX
 - 4.2. CREATE INDEX
 - 4.3. DROP INDEX
 - 4.4. rebuild
- 5. 外键(Foreign Key)
 - 5.1. FOREIGN KEY (RESTRICT)
- 6. 视图(View)
- 7. 存储过程(PROCEDURE)
 - 7.1. 存储程序
 - 7.2. EXECUTE 执行 SQL
 - 7.3. PREPARE 传递参数
 - 7.4. 存储过程返回数据
 - 7.5. 结果集转JSON
 - 7.6. 例子·过程返回结果
- 8. 函数
 - 8.1. TIMESTAMP TO ISO8601
- 9. 触发器(Trigger)
 - 9.1. create trigger
 - Update 更新出发
 - Delete 删除出发
 - Insert 插入出发
 - 9.2. drop trigger
 - 9.3. show triggers

- SHOW CREATE TRIGGER
- 9.4. EXAMPLE
 - BEFORE/AFTER
 - UUID
 - CALL PROCEDURE
- 10. 事件调度器(EVENT)
 - 10.1. 启用 EVENT
 - 10.2. 创建 EVENT
 - 10.3. 禁用/启用
 - 10.4. show events
 - 10.5. 实例·每月创建一个表
- 23. DML (Data Manipulation Language)
 - 1. INSERT
 - 1.1. INSERT INTO ... SELECT
 - 1.2. INSERT IGNORE
 - 1.3. INSERT...ON DUPLICATE KEY UPDATE
 - 2. REPLACE
 - 3. DELETE
 - 3.1. 删除重复数据
- 24. SQL Statement Syntax
 - 1. DISTINCT
 - 2. group by
 - 3. HAVING
 - 4. REGEXP
 - 5. IN / NOT IN
 - 6. ALL / Any
 - 7. exists, not exists
 - 8. UNION
 - 8.1. UNION ALL
 - 8.2. 两张表字段不对等解决方法
 - 9. OUTFILE/LOAD DATA INFILE
 - 9.1. Export data to CSV from MySQL
 - 9.2. Import data from CSV file.
 - 10. CASE Syntax
 - 11. MySQL 专有命令
 - 11.1. SQL_NO_CACHE

- 11.2. SIGNAL Syntax
- 12. SQL 92
- 25. Functions and Operators
 - 1. COUNT
 - 2. group_concat() 列传行
 - 3. UUID()
 - 4. String
 - 4.1. LEFT/RIGHT
 - 4.2. RPAD/LPAD
 - 4.3. CONCAT
 - 4.4. CONCAT_WS
 - 4.5. 链接所有字段
 - 4.6. GROUP_CONCAT
 - 4.7. replace
 - 4.8. SUBSTRING
 - 4.9. SUBSTRING_INDEX
 - 4.10. AES_ENCRYPT / AES_DECRYPT
 - 5. Date and Time
 - 5.1. year/month/day hour:minite:second
 - 5.2. Unix time
 - 5.3. DATE_FORMAT
 - 5.4. DATE_SUB/DATE_ADD
DATE_ADD
 - 5.5. datediff / timediff
 - 6. 数值函数
 - 6.1. cast 类型转换
 - 6.2. truncate 保留小数位数
 - 6.3. MOD 求余
 - 7. Control Flow Functions
- 26. DCL (Data Control Language)
 - 1. 锁
 - 1.1. 共享锁
 - 1.2. 排他锁
 - 1.3. 锁
 - 表的加锁与解锁
 - 禁止查询

- 1.4. 锁等待与超时
 - 超时设置
 - select for update nowait
 - 2. 事务处理和锁定语句
 - 2.1. 事务隔离级别
 - 2.2. 事务所用到的表
 - 2.3. 解决更新冲突
 - 2.4. 共享锁
 - 2.5. SAVEPOINT
- 27. Optimization
 - 1. Limit 状态
 - 2. 使用 Btrfs 文件系统存储mysql数据
 - 3. 打开表的数量
 - 4. Buffering and Caching
 - 4.1. Query Cache SELECT Options
 - 5. where 优化
 - 6. SHOW PROFILE Syntax SQL性能分析器
 - 7. PROCEDURE ANALYSE()
- 28. MySQL Connectors
 - 1. JDBC
 - 2. ODBC
 - 3. MySQL native driver for PHP - mysqlnd
 - 4. python-mysqldb
- 29. MySQL GUI/Web Manager
 - 1. HeidiSQL
 - 2. Toad for MySQL Freeware
 - 3. phpMyAdmin - MySQL web administration tool
 - 4. Maatkit Essential command-line utilities for MySQL
- 30. Miscellaneous
 - 1. Multi-Master Replication Manager for MySQL
 - 2. MHA
 - 3. HandlerSocket
 - 4. Maatkit
 - 5. Mroonga
 - 6. Amoeba
- 31. FAQ

1. Reset root password 重置MySQL root密码
 - 1.1. MySQL 5.7.x
 - 1.2. MySQL 8.0
2. 查看错误代码
 - 2.1. ERROR 1153 (08S01) at line 3168: Got a packet bigger than 'max_allowed_packet' bytes
 - 2.2. ERROR 1129 (00000): Host 'XXXXXX' is blocked because of many connection errors; unblock with 'mysqladmin flush-hosts'
3. 临时表是否需要建索引
4. [Warning] Changed limits: max_open_files: 5000 (requested 20480)
5. this is incompatible with sql_mode=only_full_group_by
6. ERROR 1071 (42000) at line 25: Specified key was too long; max key length is 767 bytes
7. ERROR 1086 (HY000): File '/var/lib/mysql-files/order.txt' already exists
8. Error Code: 1146. Table 'test.CACHE_UPDATE' doesn't exist
9. ERROR 1273 (HY000) at line 3364: Unknown collation: 'utf8mb4_0900_ai_ci'
10. ERROR 1290 (HY000): The MySQL server is running with the --secure-file-priv option so it cannot execute this statement
11. ERROR 1364: 1364: Field 'id' doesn't have a default value
12. ERROR 1415: Not allowed to return a result set from a trigger
13. ERROR 1503 (HY000): A PRIMARY KEY must include all columns in the table's partitioning function
14. ERROR 1819 (HY000): Your password does not satisfy the current policy requirements
15. ERROR 1820 (HY000): You must reset your password using ALTER USER statement before executing this statement.

16. ERROR 1840 (HY000) at line 24:
@@GLOBAL.GTID_PURGED can only be set when
@@GLOBAL.GTID_EXECUTED is empty.
17. ERROR 3024 (HY000): Query execution was
interrupted, maximum statement execution time
exceeded
18. Authentication plugin 'caching_sha2_password'
cannot be loaded:
/usr/lib64/mysql/plugin/caching_sha2_password.so:
cannot open shared object file: No such file or directory
19.
com.mysql.jdbc.exceptions.jdbc4.MySQLNonTransient
ConnectionException: Public Key Retrieval is not
allowed
20. mysqldump: Couldn't execute 'SELECT
COLUMN_NAME,
21. this is incompatible with
sql_mode=only_full_group_by
22. mysqldump: [Warning] Using a password on the
command line interface can be insecure.
23. mysql: [Warning] Using a password on the
command line interface can be insecure.

IV. Redis

32. Redis 安装
 1. CentOS 8 Stream
 2. CentOS 7
 3. CentOS 6
 - 3.1. 主从同步
 - 3.2. Sentinel
 4. Ubuntu
 5. Mac 安装 Redis
 6. 源码编译安装
 - 7.
 8. Test Redis
33. /etc/redis.conf
 1. 密码认证

- 2. maxmemory-policy TTL 过期策略配置
- 34. redis-cli - Command-line client to redis-server
 - 1. 命令参数
 - 1.1. password
 - 1.2. raw
 - 2. --latency Enter a special mode continuously sampling latency.
 - 3. auth
 - 4. MONITOR
 - 5. info
 - 6. save/bgsave/lastsave
 - 7. config
 - 8. keys
 - 9. 字符串操作
 - 9.1. set/get/del
 - 9.2. setnx
 - 10. expire/ttl
 - 11. 获取 key 类型
 - 12. LIST 数据类型
 - 13. set 无序字符集合
 - 14. zset (有序集合)
 - 15. Pub/Sub 订阅与发布
 - 16. flushdb 清空 Redis 数据
- 35. redis-benchmark 测试工具
- 36. Redis Cluster
- 37. Redis 通信协议
 - 1. 切换DB
 - 2. 监控
- 38. phpRedisAdmin
- 39. Redis 开发
 - 1. 消息订阅与发布
- 40. A fast, light-weight proxy for memcached and redis
- 41. FAQ
 - 1. 清空数据库
 - 2. (error) MISCONF Redis is configured to save RDB snapshots

V. MongoDB

42. Install 安装MongoDB

1. CentOS 8 Stream
2. MacOS 安装 MongoDB
3. 二进制tar包安装
4. Ubuntu MongoDB
5. CentOS 7 MongoDB
6. 从官网安装最新版本的 MongoDB 3.4
Server
Client
工具
7. MongoDB + Hadoop
8. OSCM 一键安装 MongoDB 4.0.2
9. Replication
10. Drivers
Using MongoDB in PHP

43. MongoDB 管理

1. Security and Authentication
超级管理员
数据库访问用户
数据库监控用户
删除用户
更新角色
2. 4.0早期旧版本

44. 命令工具

1. mongo - MongoDB Shell
eval
help
登陆认证
管道操作
2. mongodump - Backup
本地备份
远程备份
3. mongorestore
本地恢复
远程恢复

filter

4. mongostat
5. mongotop
6. mongofiles - Browse and modify a GridFS filesystem.

list 浏览文件
put 上传文件
get 下载
delete 删除

45. MongoDB Shell

1. shutdownServer
2. show 查看命令
 - show dbs
 - show collections
 - show users
 - show profile
3. 切换数据库
4. save
5. insert
6. update
 - multi 更新所有数据
 - upsert 更新, 如果不存在则插入数据
7. remove
 - 删除条件使用 _id
8. 删除 collection
 - 删除字段
9. count()
10. 查询
 - find() MongoDB 2.x
 - find() MongoDB 3.x
 - Query
 - 包含字段
 - 排除字段
 - sort()
 - group()
11. aggregate

- project
 - \$split
 - substr
 - groupby + sum
- 12. Indexes 索引
 - 查看索引
 - 创建索引
 - 删除索引
 - 唯一索引
 - 复合索引
 - 稀疏索引
- 13. Map-Reduce
 - 使用 Map-Reduce 统计Web 服务器 access.log 日志文件
- 14. 内嵌对象
 - Array / List 列表类型
- 15. Javascript 脚本
- 46. Mongo Admin UI
 - 1. RockMongo
 - 2. MongoVUE
- 47. Cassandra
 - 1. Getting Started
 - 1.1. Downloading and Installation
 - 1.2. Running Cassandra
 - 1.3. cli tool
 - 1.4. Testing Cassandra
 - 2. Configure Cassandra
 - 2.1. Environment variables
 - 2.2. log4j.properties
 - 2.3. storage-conf.xml
 - 3. Keyspace
 - 3.1. Schema
 - Keyspace
 - Column family
 - Name
 - Column

Super column Sorting

3.2. Keyspace example

4. Cluster

4.1. Running a cluster

4.2. Running a single node

4.3. nodetool

48. Hypertable

1. Hypertable 安装

1.1. Hypertable standalone 单机安装

1.2. Hypertable on HDFS(hadoop) 安装

1.3. MapR

1.4. Ceph

1.5. 检验安装

2. Code examples

2.1. PHP

3. HQL

3.1. namespace 命名空间管理

3.2. Table 表

4. FAQ

4.1. 切换 DFS Broker

49. CouchBase

1. 安装 CouchBase

1.1. Getting Started with Couchbase on PHP

2. couchbase 命令

2.1. couchbase-cli

50. Memcached

1. 安装 Memcached

1.1. CentOS 下编译

1.2. Ubuntu 下编译安装

1.3. debian/ubuntu

1.4. yum install

CentOS 6.x

CentOS 7.x

2. Memcached 代理

2.1. moxi

- 2.2. memagent
- 51. RethinkDB
- 52. TokyoCabinet/Tyrant
- 53. Flare
- 54. Voldemort
- 55. LevelDB
- 56. HyperDex
- 57. LeoFS
- VI. PostgreSQL
 - 58. PostgreSQL 安装
 - 1. Ubuntu 12.04.1 LTS 安装
 - 2. YUM 默认源安装
 - 3. PostgreSQL 官方 YUM 源安装
 - 3.1. 9.2
 - 3.2. 9.3
 - 3.3. 9.4
 - 3.4. 9.6
 - 3.5. PostgreSQL 10
 - 4. PostgreSQL 配置
 - 4.1. postgresql.conf
 - 4.2. pg_hba.conf
 - 5. 创建dba用户
 - 59. 数据库管理 (Database Administration)
 - 1. 用户管理 (User Account Management)
 - 1.1. 创建用户
 - 1.2. 删除用户
 - 1.3. 修改用户名
 - 1.4. 修改密码
 - 2. Access Privilege System
 - 2.1. SHOW GRANTS
 - 2.2. show privileges
 - 2.3. Grant privileges
 - 2.4. Revoke privileges
 - 2.5. Show Privileges
 - 2.6. MAX_QUERIES_PER_HOUR/MAX_UPDATES_P

- ER_HOUR
- 2.7. Table Privileges
- 2.8. Column Privileges
- 3. 字符集转换
 - 3.1. 转换 latin1 到 UTF-8
- 4. 重新整理AUTO_INCREMENT字段
- 5. 数据库内容替换
- 6. Kill 脚本
- 7. SHOW COMMAND
 - 7.1. 查看版本
 - 7.2. status
 - show status
 - show master status
 - show slave status
 - show plugins
 - 7.3. show processlist
 - 7.4. 线程的使用情况
 - 7.5. DATABASES
 - 7.6. 排序统计信息
 - 7.7. Key 状态
 - 7.8. FUNCTION
 - 7.9. PROCEDURE
 - 7.10. TRIGGERS
 - 7.11. EVENTS
 - 7.12. 引擎(ENGINES)
 - 7.13. 字符集(Collation)
 - 7.14. SHOW GRANTS
 - 7.15. validate_password
- 8. Maintenance 数据库维护
 - 8.1. CHECK 检查表
 - 8.2. ANALYZE 分析表
 - 8.3. CHECKSUM
 - 8.4. OPTIMIZE 优化表
 - 8.5. REPAIR 修复
- 9. INFORMATION_SCHEMA
 - 9.1. 查询表字段

- 9.2. 列出所有触发器
- 10. Backup and Recovery
 - 10.1. Import / Export
 - Export(Backup)
 - Import(Recovery)
 - xml
 - 备份表数据
 - source
 - 使用 mysqlhotcopy 备份 MyISAM 引擎的数据库
 - AutoMySQLBackup
 - xtrabackup - Open source backup tool for InnoDB and XtraDB.
 - Percona yum Repository
 - Creating an Incremental Backup
 - 10.2. Snapshot Backup
 - LVM Snapshot
 - Btrfs Snapshot
- 60. PostgreSQL 命令
 - 1. psql - PostgreSQL interactive terminal
 - 1.1. 执行文件
 - 2. Backup / Restore
 - 2.1. Backup
 - 2.2. Restore
 - 2.3. import
 - 2.4. 导出指定 schema
 - 2.5. PostgreSQL for windows
 - 2.6. 大型数据库备份
 - 2.7. schema之间迁移数据
- 61. DDL - Data Definition Language
 - 1. 数据库管理(Database)
 - 1.1. create
 - 1.2. drop
 - 1.3. 修改数据库
 - 1.4. Rename
 - 1.5. CHARACTER

- 1.6. show create database
- 2. 表管理(Table)
 - 2.1. 数据类型
 - SET 集合类型
 - 2.2. create table ... select
 - 2.3. modify table
 - 2.4. TEMPORARY Table
 - 2.5. Collate
 - 2.6. CHARACTER
 - 2.7. DEFAULT
 - AUTO_INCREMENT
 - TIMESTAMP NULL DEFAULT NULL ON UPDATE
 - 表存储位置(DATA DIRECTORY)
 - 2.8. KEY
 - PRIMARY KEY
 - 2.9. AUTO_INCREMENT 定义初始值
 - 2.10. COMMENT
 - 2.11. Engine 存储引擎
 - 显示当前数据库支持引擎
 - 切换引擎
 - FEDERATED
 - BLACKHOLE
 - ARCHIVE
 - CSV
- 3. Partitioning
 - 3.1. RANGE
 - 3.2. LIST
 - 3.3. HASH
 - LINEAR HASH
 - 3.4. KEY分区
 - 3.5. Subpartitioning
 - 3.6. 分区管理
 - 新增分区
 - 删除分区
 - 重建分区

分区维护

3.7. EXPLAIN PARTITIONS

3.8. SHOW CREATE TABLE

3.9. INFORMATION_SCHEMA.partitions 表

3.10. 分区数据操作

4. Index

4.1. SHOW INDEX

4.2. CREATE INDEX

4.3. DROP INDEX

4.4. rebuild

5. 外键(Foreign Key)

5.1. FOREIGN KEY (RESTRICT)

6. 视图(View)

7. 存储过程(PROCEDURE)

7.1. 存储程序

7.2. EXECUTE 执行 SQL

7.3. PREPARE 传递参数

7.4. 存储过程返回数据

7.5. 结果集转JSON

7.6. 例子·过程返回结果

8. 函数

8.1. TIMESTAMP TO ISO8601

9. 触发器(Trigger)

9.1. create trigger

Update 更新出发

Delete 删除出发

Insert 插入出发

9.2. drop trigger

9.3. show triggers

SHOW CREATE TRIGGER

9.4. EXAMPLE

BEFORE/AFTER

UUID

CALL PROCEDURE

10. 事件调度器(EVENT)

10.1. 启用 EVENT

- 10.2. 创建 EVENT
- 10.3. 禁用/启用
- 10.4. show events
- 10.5. 实例·每月创建一个表
- 62. DML (Data Manipulation Language)
 - 1. INSERT
 - 1.1. INSERT INTO ... SELECT
 - 1.2. INSERT IGNORE
 - 1.3. INSERT...ON DUPLICATE KEY UPDATE
 - 2. REPLACE
 - 3. DELETE
 - 3.1. 删除重复数据
- 63. SQL Statement Syntax
 - 1. DISTINCT
 - 2. group by
 - 3. HAVING
 - 4. REGEXP
 - 5. IN / NOT IN
 - 6. ALL / Any
 - 7. exists, not exists
 - 8. UNION
 - 8.1. UNION ALL
 - 8.2. 两张表字段不对等解决方法
 - 9. OUTFILE/LOAD DATA INFILE
 - 9.1. Export data to CSV from MySQL
 - 9.2. Import data from CSV file.
 - 10. CASE Syntax
 - 11. MySQL 专有命令
 - 11.1. SQL_NO_CACHE
 - 11.2. SIGNAL Syntax
 - 12. SQL 92
- 64. Functions and Operators
 - 1. COUNT
 - 2. group_concat() 列传行
 - 3. UUID()
 - 4. String

- 4.1. LEFT/RIGHT
- 4.2. RPAD/LPAD
- 4.3. CONCAT
- 4.4. CONCAT_WS
- 4.5. 链接所有字段
- 4.6. GROUP_CONCAT
- 4.7. replace
- 4.8. SUBSTRING
- 4.9. SUBSTRING_INDEX
- 4.10. AES_ENCRYPT / AES_DECRYPT
- 5. Date and Time
 - 5.1. year/month/day hour:minute:second
 - 5.2. Unix time
 - 5.3. DATE_FORMAT
 - 5.4. DATE_SUB/DATE_ADD
DATE_ADD
 - 5.5. datediff / timediff
- 6. 数值函数
 - 6.1. cast 类型转换
 - 6.2. truncate 保留小数位数
 - 6.3. MOD 求余
- 7. Control Flow Functions
- 65. PostgreSQL GUI
 - 1. pgAdmin III
 - 2. phpPgAdmin
 - 3. Monitoring
 - 3.1. PgBadger
 - 4. pgModeler - PostgreSQL Database Modeler
- 66. Barman Backup & recovery for PostgreSQL
- 67. pgbouncer - lightweight connection pooler for PostgreSQL
 - 1. 安装 pgbouncer
 - 1.1. Ubuntu
 - 1.2. CentOS
 - 2. 配置 pgbouncer
- 68. Foreign data wrappers

- 69. Barman for PostgreSQL
- 70. Connector
 - 1. Pomm
- 71. Replication
 - 1. Bucardo
- 72. FAQ
 - 1. Reset root password 重置MySQL root密码
 - 1.1. MySQL 5.7.x
 - 1.2. MySQL 8.0
 - 2. 查看错误代码
 - 2.1. ERROR 1153 (08S01) at line 3168: Got a packet bigger than 'max_allowed_packet' bytes
 - 2.2. ERROR 1129 (00000): Host 'XXXXXX' is blocked because of many connection errors; unblock with 'mysqladmin flush-hosts'
 - 3. 临时表是否需要建索引
 - 4. [Warning] Changed limits: max_open_files: 5000 (requested 20480)
 - 5. this is incompatible with sql_mode=only_full_group_by
 - 6. ERROR 1071 (42000) at line 25: Specified key was too long; max key length is 767 bytes
 - 7. ERROR 1086 (HY000): File '/var/lib/mysql-files/order.txt' already exists
 - 8. Error Code: 1146. Table 'test.CACHE_UPDATE' doesn't exist
 - 9. ERROR 1273 (HY000) at line 3364: Unknown collation: 'utf8mb4_0900_ai_ci'
 - 10. ERROR 1290 (HY000): The MySQL server is running with the --secure-file-priv option so it cannot execute this statement
 - 11. ERROR 1364: 1364: Field 'id' doesn't have a default value
 - 12. ERROR 1415: Not allowed to return a result set from a trigger

13. ERROR 1503 (HY000): A PRIMARY KEY must include all columns in the table's partitioning function
14. ERROR 1819 (HY000): Your password does not satisfy the current policy requirements
15. ERROR 1820 (HY000): You must reset your password using ALTER USER statement before executing this statement.
16. ERROR 1840 (HY000) at line 24:
@@GLOBAL.GTID_PURGED can only be set when @@GLOBAL.GTID_EXECUTED is empty.
17. ERROR 3024 (HY000): Query execution was interrupted, maximum statement execution time exceeded
18. Authentication plugin 'caching_sha2_password' cannot be loaded:
/usr/lib64/mysql/plugin/caching_sha2_password.so: cannot open shared object file: No such file or directory
19.
com.mysql.jdbc.exceptions.jdbc4.MySQLNonTransientConnectionException: Public Key Retrieval is not allowed
20. mysqldump: Couldn't execute 'SELECT COLUMN_NAME,
21. this is incompatible with
sql_mode=only_full_group_by
22. mysqldump: [Warning] Using a password on the command line interface can be insecure.
23. mysql: [Warning] Using a password on the command line interface can be insecure.

VII. Oracle

73. Oracle 12c 安装可

1. Server 安装
2. Desktop 安装
3. Oracle Net Services Configuration
 - 3.1. Listener configuration
 - 3.2. Local Net Service Name configuration

4. Oracle 12c Client for Windows 10
5. 卸载 Oracle
6. Linux 安装 sqlplus
74. Privilege
 1. Account
 - 1.1. show user
 - 1.2. DEFAULT_TABLESPACE
 - 1.3. unlock/lock
 2. User
 - 2.1. Create user
 - 2.2. Drop user
 - 2.3. Alter user
 - 2.4. 修改密码
 - 2.5. Reset Password 忘记密码重置密码
 3. grant
 - 3.1. 列出权限
 - 3.2. 查看用户权限
 - 3.3. 吊销权限
 - 3.4. System Privileges 系统权限
EM 权限
Granting and Revoking SYSDBA and
SYSOPER Privileges
 - 3.5. Object Privileges 对象权限
 4. Opening the Encrypted Wallet
75. 数据库管理
 1. lsnrctl
 2. listener.ora
 3. TNS 配置
 - 3.1. HOST
 - 3.2. SID
 4. parameter
 - 4.1. db
 - 4.2. instance_name
 - 4.3. service_name
 - 4.4. global_name
 - 4.5. db_name

- 4.6. db_domain
- 4.7. sga
- 4.8. size
- 4.9. spfile
- 4.10. cache
- 4.11. Character Set
- 5. 进程
- 6. 查看用户
- 7. 显示表
- 8. 显示试图
- 9. PROCEDURE
- 10. FUNCTION
- 11. 视图、存储过程、函数
- 12. 查看存储过程源代码
- 13. 日期时间格式
 - 13.1. Date
 - 13.2. 修改系统日期格式
 - 13.3. TIMESTAMP
 - 13.4.
- 14. 切换字符集
 - 14.1. 切换到 GBK
 - 14.2. 切到 UTF-8
 - 14.3. 切到 AL32UTF8
 - 14.4. 切换过程实例
- 15. Oracle 表空间
 - 15.1. 查询空闲表空间
 - 15.2. 创建表空间
临时表空间
 - 15.3. 更改表空间属性
修改表空间大小
 - 15.4. 删除表空间
- 76. Oracle Gui
 - 1. Create instance
 - 2. Oracle Net Configuration Assistant
 - 3. Oracle Enterprise Manager

3.1. EM Configuration issue.

/opt/oracle/product/11.2.0/dbhome_1/orcl.example.com_orcl not found.

4. Other GUI - phpOraAdmin

77. CLI

1. SQL*Plus

1.1. conn

1.2. startup/shutdown

startup

shutdown

1.3. \$ORACLE_HOME/sqlplus/admin/glogin.sql

1.4. @运行SQL

1.5. 链接数据库

2. exp/imp

2.1. exp

2.2. imp

A用户导出B用户导入

3. expdp/impdp

3.1. expdp

3.2. impdp

3.3. 数据泵演示

3.4. 查看dmp文件的表空间

4. RMAN

4.1. 数据库模式

4.2. 完全备份

4.3. 增量备份

4.4. 恢复数据库

4.5. 是用tar打包rman文件

4.6. 打包 rman 备份文件

78. DDL

1. 字段类型

1.1. Date

1.2. TIMESTAMP

2. 索引

3. 表

3.1.

- 3.2. 约束检查
- 4. SEQUENCE 序列
 - 4.1.
- 79. 程序包
 - 1. 执行包中的过程
- 80. 数据库链接
- 81. Oracle 监控
 - 1. Oracle日志的路径
 - 2. SQL 日志监控
 - 2.1. 查询性能分析
 - 2.2. IO性能分析
 - 3. session
 - 4. SQL Monitoring Report
- 82. Installing Oracle Database
 - 1. Installing Oracle Database 10g Release 2 on Linux x86
 - 2. 11gR2
 - 3. Installing Oracle Client 11g
 - 4. Silence Install - Database
 - 4.1. Response File
 - 4.2. OS 配置脚本
 - 4.3. 运行 runInstaller
 - 5. Silence Install - Client
 - 6. oracle 817 script
 - 7. Script for automatic startup on boot
 - 8. Run level shell script to start Oracle 10g services on RedHat Enterprise Linux (RHAS 4)
- 83. FAQ
 - 1. Reset root password 重置MySQL root密码
 - 1.1. MySQL 5.7.x
 - 1.2. MySQL 8.0
 - 2. 查看错误代码
 - 2.1. ERROR 1153 (08S01) at line 3168: Got a packet bigger than 'max_allowed_packet' bytes
 - 2.2. ERROR 1129 (00000): Host 'XXXXXX' is blocked because of many connection errors;

- unblock with 'mysqladmin flush-hosts'
3. 临时表是否需要建索引
 4. [Warning] Changed limits: max_open_files: 5000 (requested 20480)
 5. this is incompatible with sql_mode=only_full_group_by
 6. ERROR 1071 (42000) at line 25: Specified key was too long; max key length is 767 bytes
 7. ERROR 1086 (HY000): File '/var/lib/mysql-files/order.txt' already exists
 8. Error Code: 1146. Table 'test.CACHE_UPDATE' doesn't exist
 9. ERROR 1273 (HY000) at line 3364: Unknown collation: 'utf8mb4_0900_ai_ci'
 10. ERROR 1290 (HY000): The MySQL server is running with the --secure-file-priv option so it cannot execute this statement
 11. ERROR 1364: 1364: Field 'id' doesn't have a default value
 12. ERROR 1415: Not allowed to return a result set from a trigger
 13. ERROR 1503 (HY000): A PRIMARY KEY must include all columns in the table's partitioning function
 14. ERROR 1819 (HY000): Your password does not satisfy the current policy requirements
 15. ERROR 1820 (HY000): You must reset your password using ALTER USER statement before executing this statement.
 16. ERROR 1840 (HY000) at line 24: @@GLOBAL.GTID_PURGED can only be set when @@GLOBAL.GTID_EXECUTED is empty.
 17. ERROR 3024 (HY000): Query execution was interrupted, maximum statement execution time exceeded
 18. Authentication plugin 'caching_sha2_password' cannot be loaded:

/usr/lib64/mysql/plugin/caching_sha2_password.so:
cannot open shared object file: No such file or directory
19.

com.mysql.jdbc.exceptions.jdbc4.MySQLNonTransient
ConnectionException: Public Key Retrieval is not
allowed

20. mysqldump: Couldn't execute 'SELECT
COLUMN_NAME,

21. this is incompatible with
sql_mode=only_full_group_by

22. mysqldump: [Warning] Using a password on the
command line interface can be insecure.

23. mysql: [Warning] Using a password on the
command line interface can be insecure.

84. 数据库迁移

1. MySQL to PostgreSQL

表格清单

26.1. 更新丢失演示

26.2. 防止更新丢失加锁演示

范例清单

8.1. Elasticsearch master / slave

9.1. Solr - schema.xml

9.2. /etc/profile.d/java.sh

19.1. MySQL 8 创建root账号

19.2. my.cnf

19.3. my.cnf

19.4. my.cnf

22.1. BEFORE/AFTER

22.2. uuid()

24.1. SQL ANY example

27.1. SQL_CACHE 测试

- 27.2. SHOW PROFILE Syntax
- 42.1. MongoDB Test
- 42.2. Using MongoDB in PHP
- 47.1. Twitter
- 47.2. Twissandra
- 50.1. /etc/init.d/memcached
- 58.1. Example for /var/lib/pgsql/10/data/pg_hba.conf
- 61.1. BEFORE/AFTER
- 61.2. uuid()
- 63.1. SQL ANY example
- 67.1. /etc/pgbouncer/pgbouncer.ini
- 75.1. Oracle 字符集切换实例
- 75.2. 创建表空间实例
- 77.1. expdp
- 77.2. impdp
- 77.3. full backup
- 78.1.
- 80.1. DB LINK

Netkiller Database 手札

PostgreSQL, MariaDB, MySQL, Oracle, Redis, MongoDB,
Elasticsearch ...

ISBN#

Mr. Neo Chan, 陈景峯(BG7NYT)

中国广东省深圳市望海路半岛城邦三期
518067
+86 13113668890

<netkiller@msn.com>

文档初创于April 6, 2008

电子书最近一次更新于 2021-08-28 11:29:42

版权 © 2009-2019 Netkiller(Neo Chan). All rights reserved.

版权声明

转载请与作者联系，转载时请务必标明文章原始出处和作者信息及本声明。



<http://www.netkiller.cn>
<http://netkiller.github.io>
<http://netkiller.sourceforge.net>

微信订阅号 netkiller-ebook
微信: 13113668890 请注明
“读者”

QQ: 13721218 请注明“读
者”

QQ群: 128659835 请注明
“读者”



Netkiller Database 手札

陈景峰 著



\$Date: 2013-05-21 19:01:33 +0800 (Tue, 21 May 2013) \$

Netkiller 数据库系列手札

数据库系列手札

[Netkiller Database 手札](#)

| [Netkiller PostgreSQL 手札](#)

| [Netkiller MySQL 手札](#)

| [Netkiller NoSQL 手札](#)

| [Netkiller LDAP 手札](#)

致读者

Netkiller 系列电子书始于 2000 年，风风雨雨走过20年，将在 2020 年终结，之后不在更新。作出这种决定原因很多，例如现在的阅读习惯已经转向短视频，我个人的时间，身体健康情况等等.....

感谢读者粉丝这20年的支持

虽然电子书不再更新，后面我还会活跃在[知乎社区](#)和微信公众号

自述

Netkiller 手札系列电子书 <http://www.netkiller.cn>

Netkiller Database 手札

陈景峰 著



《Netkiller 系列 手札》是一套免费系列电子书，netkiller 是 nickname 从1999 开使用至今，“手札”是札记，手册的含义。

2003年之前我还是以文章形式在BBS上发表各类技术文章，后来发现文章不够系统，便尝试写长篇技术文章加上章节目录等等。随着内容增加，不断修订，开始发布第一版，第二版.....

IT知识变化非常快，而且具有时效性，这样发布非常混乱，经常有读者发现第一版例子已经过时，但他不知道我已经发布第二版。

我便有一种想法，始终维护一个文档，不断更新，使他保持较新的版本不过时。

第一部电子书是《PostgreSQL 实用实例参考》开始我使用 Microsoft Office Word 慢慢随着文档尺寸增加 word 开始表现出力不从心。

我看到PostgreSQL 中文手册使用SGML编写文档，便开始学习 Docbook SGML。使用Docbook写的第一部电子书是《Netkiller Postfix Integrated Solution》这是Netkiller 系列手札的原型。

至于“手札”一词的来历，是因为我爱好摄影，经常去一个台湾摄影网站，名字就叫“摄影家手札”。

由于硬盘损坏数据丢失 《Netkiller Postfix Integrated Solution》的 SGML文件已经不存在；Docbook SGML存在很多缺陷 UTF-8支持不好，转而使用Docbook XML。

目前技术书籍的价格一路飙升，动则¥80，¥100，少则¥50，¥60。技术书籍有时效性，随着技术的革新或淘汰，大批书籍成为废纸垃圾。并且这些书技术内容雷同，相互抄袭，质量越来越差，甚至里面给出的例子错误百出，只能购买影印版，或者翻译的版本。

在这种背景下我便萌生了自己写书的想法，资料主要来源是我的笔记与例子。我并不想出版，只为分享，所以我制作了基于CC License 发行的系列电子书。

本书注重例子，少理论（捞干货），只要你对着例子一步一步操作，就会成功，会让你有成就感并能坚持学下去，因为很多人遇到障碍就会放弃，其实我就是这种人，只要让他看到希望，就能坚持下去。

1. 写给读者

为什么写这篇文章

有很多想法,工作中也用不到所以未能实现,所以想写出来,和大家分享.有一点写一点,写得也不好,只要能看懂就行,就当学习笔记了.

开始零零碎碎写过一些文档,也向维基百科供过稿,但维基经常被ZF封锁,后来发现sf.net可以提供主机存放文档,便做了迁移.并开始了我的写作生涯.

这篇文档是作者20年来对工作的总结,是作者一点一滴的积累起来的,有些笔记已经丢失,所以并不完整.

因为工作太忙整理比较缓慢.目前的工作涉及面比较窄所以新文档比较少.

我现在花在技术上的时间越来越少,兴趣转向摄影,无线电.也想写写摄影方面的心得体会.

写作动力:

曾经在网上看到外国开源界对中国的评价,中国人对开源索取无度,但贡献却微乎其微.这句话一直记在我心中,发誓要为中国开源事业做我仅有的一点微薄贡献

另外写文档也是知识积累,还可以增加在圈内的影响力.

人跟动物的不同,就是人类可以把自己学习的经验教给下一代人.下一代在上一代的基础上再创新,不断积累才有今天.

所以我把自己的经验写出来,可以让经验传承

没有内容的章节:

目前我自己一人维护所有文档,写作时间有限,当我发现一个好主题就会加入到文档中,待我有时间再完善章节,所以你会发现很多章节是空无内容的.

文档目前几乎是流水帐式的写作,维护量很大,先将就着看吧.

我想到哪写到哪,你会发现文章没一个中心,今天这里写点,明天跳过本

章写其它的.

文中例子绝对多,对喜欢复制然后粘贴朋友很有用,不用动手写,也省时间.

理论的东西,网上大把,我这里就不写了,需要可以去网上查.

我爱写错别字,还有一些是打错的,如果发现请指正.

文中大部分试验是在Debian/Ubuntu/Redhat AS上完成.

写给读者

至读者:

我不知道什么时候,我不再更新文档或者退出IT行业去从事其他工作,我必须给这些文档找一个归宿,让他能持续更新下去。

我想捐赠给某些基金会继续运转,或者建立一个团队维护它。

我用了20年时间坚持不停地写作,持续更新,才有今天你看到的《Netkiller 手札》系列文档,在中国能坚持20年,同时没有任何收益的技术类文档,是非常不容易的。

有很多时候想放弃,看到外国读者的支持与国内社区的影响,我坚持了下来。

中国开源事业需要各位参与,不要成为局外人,不要让外国人说:中国对开源索取无度,贡献却微乎其微。

我们参与内核的开发还比较遥远,但是进个人能力,写一些文档还是可能的。

系列文档

下面是我多年积累下来的经验总结,整理成文档供大家参考:

[Netkiller Architect 手札](#)

[Netkiller Developer 手札](#)

[Netkiller PHP 手札](#)

[Netkiller Python 手札](#)

[Netkiller Testing 手札](#)

[Netkiller Cryptography 手札](#)

[Netkiller Linux 手札](#)
[Netkiller FreeBSD 手札](#)
[Netkiller Shell 手札](#)
[Netkiller Security 手札](#)
[Netkiller Web 手札](#)
[Netkiller Monitoring 手札](#)
[Netkiller Storage 手札](#)
[Netkiller Mail 手札](#)
[Netkiller Docbook 手札](#)
[Netkiller Version 手札](#)
[Netkiller Database 手札](#)
[Netkiller PostgreSQL 手札](#)
[Netkiller MySQL 手札](#)
[Netkiller NoSQL 手札](#)
[Netkiller LDAP 手札](#)
[Netkiller Network 手札](#)
[Netkiller Cisco IOS 手札](#)
[Netkiller H3C 手札](#)
[Netkiller Multimedia 手札](#)
[Netkiller Management 手札](#)
[Netkiller Spring 手札](#)
[Netkiller Perl 手札](#)
[Netkiller Amateur Radio 手札](#)

2. 作者简介

陈景峯 ([ネウキ | ㄣㄩㄣ](#))

Nickname: netkiller | English name: Neo chen | Nippon name: ちんけいほう (音訳) | Korean name: 천징봉 | Thailand name: ภูมิภาพภูเข่า | Vietnam: Trần Cảnh Phong

Callsign: [BG7NYT](#) | QTH: ZONE CQ24 ITU44 ShenZhen, China

程序猿，攻城狮，挨踢民工，Full Stack Developer, UNIX like Evangelist, 业余无线电爱好者（呼号：BG7NYT），户外运动，山地骑行以及摄影爱好者。

《Netkiller 系列手札》的作者

成长阶段

1981年1月19日(庚申年腊月十四)出生于黑龙江省青冈县建设乡双富大队第一小队

1989年9岁随父母迁居至黑龙江省伊春市，悲剧的天朝教育，不知道那门子归定，转学必须降一级，我本应该上一年级，但体制让我上学前班，那年多都10岁了

1995年小学毕业，体制规定借读要交3000两银子(我曾想过不升初中)，亲戚单位分楼告别平房，楼里没有地方放东西，把2麻袋书送给我，无意中发现一本电脑书BASIC语言，我竟然看懂了，对于电脑知识追求一发而不可收，后面顶零花钱，压岁钱主要用来买电脑书《MSDOS 6.22》《新编Unix实用大全》《跟我学Foxbase》。。。。。。

1996年第一次接触UNIX操作系统，BSD UNIX, Microsoft Xinux(盖茨亲自写的微软Unix，知道的人不多)

1997年自学Turbo C语言，苦于没有电脑，后来学校建了微机室才第一次使用QBASIC(DOS 6.22 自带命令)，那个年代只能通过软盘拷贝转播，Turbo C编译器始终没有搞到，

1997年第一次上Internet网速只有9600Bps, 当时全国兴起各种信息港域名格式是www.xxxx.info.net, 访问的第一个网站是NASA下载了很多火星探路者拍回的照片，还有“淞沪”sohu的前身

1998~2000年在哈尔滨学习计算机，充足的上机时间，但老师让我们练打字（明伦五笔/WT）打字不超过80个/每分钟还要强化训练，不过这个给我的键盘功夫打了好底。

1999年学校的电脑终于安装了光驱，在一张工具盘上终于找到了Turbo C, Borland C++与Quick Basic编译器，当时对VGA图形编程非常感兴趣，通过INT33中断控制鼠标，使用绘图函数模仿windows界面。还有操作UCDOS中文字库，绘制矢量与点阵字体。

2000年沉迷于Windows NT与Back Office各种技术，神马主域控制器，DHCP，WINS，IIS，域名服务器，Exchange邮件服务器，MS Proxy, NetMeeting...以及ASP+MS SQL开发；用56K猫下载了一张LINUX。ISO镜像，安装后我兴奋的24小时没有睡觉。

职业生涯

2001年来深圳进城打工,成为一名外来务工者. 在一个4人公司做PHP开发，当时PHP的版本是2.0, 开始使用Linux Redhat 6.2.当时很多门户网站都是用FreeBSD,但很难搞到安装盘，在网易社区认识了一个网友,从广州给我寄了一张光盘，FreeBSD 3.2

2002年我发现不能埋头苦干,还要学会"做人".后辗转广州工作了半年，考了一个Cisco CCNA认证。回到深圳重新开始，在车公庙找到一家工作做Java开发

2003年这年最惨,公司拖欠工资16000元,打过两次官司2005才付清.

2004 年开始加入[分布式计算](#)团队,[目前成绩](#)，工作仍然是Java开发并且开始使用PostgreSQL数据库。

2004-10月开始玩户外和摄影

2005-6月成为中国无线电运动协会会员,呼号BG7NYT,进了一部Yaesu FT-60R手台。公司的需要转回PHP与MySQL，相隔几年发现PHP进步很大。在前台展现方面无人能敌，于是便前台使用PHP，后台采用Java开发。

2006 年单身生活了这么多年,终于找到归宿. 工作更多是研究PHP各种框架原理

2007 物价上涨,金融危机，休息了4个月（其实是找不到工作），关外很难上439.460中继，搞了一台Yaesu FT-7800.

2008 终于找到英文学习方法， 《Netkiller Developer 手札》 ，
《Netkiller Document 手札》

2008-8-8 08:08:08 结婚,后全家迁居湖南省常德市

2009 《Netkiller Database 手札》 ,2009-6-13学车，年底拿到C1驾照

2010 对电子打击乐产生兴趣，计划学习爵士鼓。由于我对Linux热爱，我轻松的接管了公司的运维部，然后开发运维两把抓。我印象最深刻的是公司一次上架10个机柜，我们用买服务器纸箱的钱改善伙食。我将40多台服务器安装BOINC做压力测试，获得了中国第二的名次。

2011 平凡的一年，户外运动停止，电台很少开，中继很少上，摄影主要是拍女儿与家人，年末买了一辆山地车

2012 对油笔画产生了兴趣，活动基本是骑行银湖山绿道，

2013 开始学习民谣吉他，同时对电吉他也极有兴趣；最终都放弃了。这一年深圳开始推数字中继2013-7-6日入手Motorola

MOTOTRBO XIR P8668, Netkiller 系列手札从Sourceforge向Github迁移; 年底对MYSQL UDF, Engine与PHP扩展开发产生很浓的兴趣, 拾起遗忘10+年的C, 写了几个mysql扩展(图片处理, fifo管道与ZeroMQ), 10月份入Toyota Rezi 2.5V并写了一篇《攻城狮的苦逼选车经历》

2014-9-8 在淘宝上买了一架电钢琴 Casio Privia PX-5S pro 开始陪女儿学习钢琴, 由于这家钢琴是合成器电钢, 里面有打击乐, 我有对键盘鼓产生了兴趣。

2014-10-2号罗浮山两日游, 对中国道教文化与音乐产生了兴趣, 10月5号用了半天时间学会了简谱。10月8号入Canon 5D Mark III + Canon Speedlite 600EX-RT香港过关被查。

2014-12-20号对乐谱制作产生兴趣
(<https://github.com/SheetMusic/Piano>), 给女儿做了几首钢琴伴奏曲, MuseScore制谱然后生成MIDI与WAV文件。

2015-09-01 晚饭后拿起爵士鼓基础教程尝试在Casio Privia PX-5S pro演练, 经过反复琢磨加上之前学钢琴的乐理知识, 终于在02号晚上, 打出了简单的基本节奏, 迈出了第一步。

2016 对弓箭(复合弓)产生兴趣, 无奈天朝法律法规不让玩。每周游泳轻松1500米无压力, 年底入 xbox one s 和 Yaesu FT-2DR, 同时开始关注功放音响这块

2017 7月9号入 Yamaha RX-V581 功放一台, 连接Xbox打游戏爽翻了, 入Kindle电子书, 计划学习蝶泳, 果断放弃运维和开发知识体系转攻区块链。

2018 从溪山美地搬到半岛城邦, 丢弃了多年攒下的家底。11月开始玩 MMDVM, 使用 Yaesu FT-7800 发射, 连接MMDVM中继板, 树莓派, 覆盖深圳湾, 散步骑车通联两不误。

2019 卖了常德的房子, 住了5次院, 哮喘反复发作, 决定停止电子书更新, 兴趣转到知乎, B站

2020 准备找工作

职业生涯路上继续打怪升级

3. 如何获得文档

下载 Netkiller 手札 (epub,kindle,chm,pdf)

EPUB <https://github.com/netkiller/netkiller.github.io/tree/master/download/epub>

MOBI <https://github.com/netkiller/netkiller.github.io/tree/master/download/mobi>

PDF <https://github.com/netkiller/netkiller.github.io/tree/master/download/pdf>

CHM <https://github.com/netkiller/netkiller.github.io/tree/master/download/chm>

通过 GIT 镜像整个网站

<https://github.com/netkiller/netkiller.github.com.git>

```
$ git clone https://github.com/netkiller/netkiller.github.com.git
```

镜像下载

整站下载

```
wget -m http://www.netkiller.cn/index.html
```

指定下载

```
wget -m wget -m http://www.netkiller.cn/linux/index.html
```

Yum 下载文档

获得光盘介质, RPM包, DEB包, 如有特别需要, 请联系我

YUM 在线安装电子书

<http://netkiller.sourceforge.net/pub/repo/>

```
# cat >> /etc/yum.repos.d/netkiller.repo <<EOF  
[netkiller]
```

```
name=Netkiller Free Books
baseurl=http://netkiller.sourceforge.net/pub/repo/
enabled=1
gpgcheck=0
gpgkey=
EOF
```

查找包

```
# yum search netkiller

netkiller-centos.x86_64 : Netkiller centos Cookbook
netkiller-cryptography.x86_64 : Netkiller cryptography Cookbook
netkiller-docbook.x86_64 : Netkiller docbook Cookbook
netkiller-linux.x86_64 : Netkiller linux Cookbook
netkiller-mysql.x86_64 : Netkiller mysql Cookbook
netkiller-php.x86_64 : Netkiller php Cookbook
netkiller-postgresql.x86_64 : Netkiller postgresql Cookbook
netkiller-python.x86_64 : Netkiller python Cookbook
netkiller-version.x86_64 : Netkiller version Cookbook
```

安装包

```
yum install netkiller-docbook
```


4. 打赏 (Donations)

If you like this documents, please make a donation to support the authors' efforts. Thank you!

您可以通过微信，支付宝，贝宝给作者打赏。

银行(Bank)

招商银行(China Merchants Bank)

开户名：陈景峰

账号：9555500000007459

微信 (Wechat)



支付宝 (Alipay)



PayPal Donations

<https://www.paypal.me/netkiller>

5. 联系方式

主站 <http://www.netkiller.cn/>

备用 <http://netkiller.github.io/>

繁体网站 <http://netkiller.sourceforge.net/>

联系作者

Mobile: +86 13113668890

Email: netkiller@msn.com

QQ群: 128659835 请注明“读者”

QQ: 13721218

ICQ: 101888222

注：请不要问我安装问题！

博客 **Blogger**

知乎专栏 <https://zhuanlan.zhihu.com/netkiller>

LinkedIn: <http://cn.linkedin.com/in/netkiller>

OSChina: <http://my.oschina.net/neochen/>

Facebook: <https://www.facebook.com/bg7nyt>

Flickr: <http://www.flickr.com/photos/bg7nyt/>

Disqus: <http://disqus.com/netkiller/>

solidot: <http://solidot.org/~netkiller/>

SegmentFault: <https://segmentfault.com/u/netkiller>

Reddit: <https://www.reddit.com/user/netkiller/>

Digg: <http://www.digg.com/netkiller>

Twitter: <http://twitter.com/bg7nyt>

weibo: <http://weibo.com/bg7nyt>

Xbox club

我的 xbox 上的ID是 netkiller xbox， 我创建了一个俱乐部 netkiller 欢迎加入。

Radio

CQ CQ CQ DE BG7NYT:

如果这篇文章对你有所帮助,请寄给我一张QSL卡片, qrz.cn or qrz.com or hamcall.net

Personal Amateur Radiostations of P.R.China

ZONE CQ24 ITU44 ShenZhen, China

Best Regards, VY 73! OP. BG7NYT

守听频率 DMR 438.460 -8 Color 12 Slot 2 Group 46001

守听频率 C4FM 439.360 -5 DN/VW

MMDVM Hotspot:

Callsign: BG7NYT QTH: Shenzhen, China

YSF: YSF80337 - CN China 1 - W24166/TG46001

DMR: BM_China_46001 - DMR Radio ID: 4600441

部分 I. Database Relational

What are the difference between DDL, DML and DCL commands?

DDL is Data Definition Language statements. Some examples:

CREATE - to create objects in the database

ALTER - alters the structure of the database

DROP - delete objects from the database

TRUNCATE - remove all records from a table, including all spaces allocated for

the records are removed

COMMENT - add comments to the data dictionary

GRANT - gives user's access privileges to database

REVOKE - withdraw access privileges given with the GRANT command

DML is Data Manipulation Language statements. Some examples:

SELECT - retrieve data from the a database

INSERT - insert data into a table

UPDATE - updates existing data within a table

DELETE - deletes all records from a table, the space for the records remain

CALL - call a PL/SQL or Java subprogram

EXPLAIN PLAN - explain access path to data

LOCK TABLE - control concurrency

DCL is Data Control Language statements. Some examples:

COMMIT - save work done

SAVEPOINT - identify a point in a transaction to which you can later roll back

ROLLBACK - restore database to original since the last COMMIT

SET TRANSACTION - Change transaction options like what rollback segment to use

第 1 章 DB-Engines - DB-Engines Ranking

<http://db-engines.com/en/ranking>

第 2 章 SysBench

第 3 章 SuperSmack

第 4 章 ETL (Extract-Transform-Load)

*ETL*或*ELT* (*extract*)、*转置* (*transform*)、*加载* (*load*)

<http://zh.wikipedia.org/wiki/ETL>

```
Kettle http://www.ketl.org
http://www.cloveretl.org/
http://www.xaware.org/
Apatar http://www.apatar.org/
http://www.enhydra.org/
```

1. Kettle

<http://community.pentaho.com/projects/data-integration/>

1.1. 安装

Mac 安装

```
neo@MacBook-Pro ~ % brew info kettle
kettle: stable 8.2.0.0-342
Pentaho Data Integration software
https://community.hitachivantara.com/docs/DOC-1009931-downloads
Not installed
From: https://github.com/Homebrew/homebrew-core/blob/master/Formula/kettle.rb
==> Caveats
To have launchd start kettle now and restart at login:
  brew services start kettle
Or, if you don't want/need a background service you can just
```

```
run:
  pdicarte /usr/local/etc/kettle/carte-config.xml
==> Analytics
install: 128 (30 days), 400 (90 days), 1,603 (365 days)
install_on_request: 126 (30 days), 386 (90 days), 1,560 (365
days)
build_error: 0 (30 days)
```

```
brew install kettle
```

<https://sourceforge.net/projects/pentaho/files/Data%20Integration/>

2. suro

<https://github.com/Netflix/suro>

第 5 章 数据迁移

data migration

1. Apache Sqoop

<http://sqoop.apache.org/>

```
export SQOOP_HOME=/srv/sqoop/  
export PATH=$SQOOP_HOME/bin:$PATH
```

MYSQL转HDFS-示例

```
./sqoop import --connect jdbc:mysql://192.168.0.1:3306/mydb --  
username root --password mypass --table mytable -m 1 --target-  
dir /mysql/yourtable/$today
```

HDFS转MYSQ-示例

```
./sqoop export --connect jdbc:mysql://192.168.0.1:3306/mydb --  
username root --password mypass --table mytable --fields-  
terminated-by ',' --export-dir /mysql/yourtable/$today
```

第 6 章 GreenSQL

第 7 章 Database design & E-R diagram

1. opensource database design tools

DBDesigner 4 - <http://www.fabforce.net/dbdesigner4/>

2. OpenSystemArchitect

<http://www.codebydesign.com>

3. SQL Power Architect

<http://www.sqlpower.ca/>

部分 II. Search Engine

第 8 章 Elasticsearch

<http://www.elasticsearch.org/>

1. 安装 Elasticsearch

1.1. 6.x 安装

安装 6.x 仓库

```
curl -s
https://raw.githubusercontent.com/oscm/shell/master/search/elastic/elastic-
6.x.sh | bash
```

安装 6.x 包

```
yum install elasticsearch
```

1.2. 单机模式 (适用于开发环境) 5.x

使用 Netkiller OSCM 一键安装 Elasticsearch 5.6.0

```
# Java
curl -s
https://raw.githubusercontent.com/oscm/shell/master/lang/java/openjdk/java-
1.8.0-openjdk.sh | bash

# Install
curl -s
https://raw.githubusercontent.com/oscm/shell/master/search/elasticsearch/elastic
search-5.x.sh | bash

# Bind 0.0.0.0
curl -s
https://raw.githubusercontent.com/oscm/shell/master/search/elasticsearch/network
.bind_host.sh | bash

# Auto create index
curl -s
https://raw.githubusercontent.com/oscm/shell/master/search/elasticsearch/action.
auto_create_index.sh | bash
```

```
# elasticsearch-analysis-ik

curl -s
https://raw.githubusercontent.com/oscm/shell/master/search/elasticsearch/5.5/elasticsearch-analysis-ik-5.6.0.sh | bash
```

通常 elasticsearch-analysis-ik 的版本会比 elasticsearch 慢一个版本，所以请使用下面命令查看版本是否一致，如果不一致可以修改 plugin-descriptor.properties 配置文件，使其一致。

```
root@netkiller /usr/share/elasticsearch/plugins/ik % grep ^version plugin-
descriptor.properties
version=5.5.1
```

启动后使用 jps 命令检查进程是否工作正常

```
root@netkiller /var/log/elasticsearch % jps | grep Elasticsearch
9706 Elasticsearch

root@netkiller /var/log/elasticsearch % ss -lnt | grep 9200
LISTEN      0          128          127.0.0.1:9200                **:
```

1.3. Elasticsearch Cluster 5.x

集群模式需要两个以上的节点，通常是一个 master 节点，多个 data 节点

首先在所有节点上安装 elasticsearch，然后配置各节点的配置文件，对于 5.5.1 不需要配置决定哪些节点属于 master 节点 或者 data 节点。

```
curl -s
https://raw.githubusercontent.com/oscm/shell/master/search/elasticsearch/elasticsearch-5.x.sh | bash
```

配置文件

```
cluster.name: elasticsearch-cluster # 配置集群名称,所有服务器服务器保持一致

node.name: node-1 # 每个节点唯一标识, 每个节点只需改动这里, 一次递增 node-1, node-2,
node-3 ...

network.host: 0.0.0.0

discovery.zen.ping.unicast.hosts: ["172.16.0.20", "172.16.0.21", "172.16.0.22"]
# 所有节点的IP 地址写在这里
```

```
discovery.zen.minimum_master_nodes: 3 # 可以作为master的节点总数, 有多少个节点就写多少
http.cors.enabled: true
http.cors.allow-origin: "*"
```

查看节点状态, 使用curl工具: curl 'http://localhost:9200/_nodes/process?pretty'

```
root@netkiller /var/log/elasticsearch % curl
'http://localhost:9200/_nodes/process?pretty'
{
  "_nodes" : {
    "total" : 2,
    "successful" : 2,
    "failed" : 0
  },
  "cluster_name" : "my-application",
  "nodes" : {
    "-lnKcMbxRpiwExLns0jc9g" : {
      "name" : "node-1",
      "transport_address" : "10.104.3.2:9300",
      "host" : "10.104.3.2",
      "ip" : "10.104.3.2",
      "version" : "5.5.1",
      "build_hash" : "19c13d0",
      "roles" : [
        "master",
        "data",
        "ingest"
      ],
      "process" : {
        "refresh_interval_in_millis" : 1000,
        "id" : 23669,
        "mlockall" : false
      }
    },
    "WVsgYi2HT8GwnZU1kUwFwA" : {
      "name" : "node-2",
      "transport_address" : "10.186.7.221:9300",
      "host" : "10.186.7.221",
      "ip" : "10.186.7.221",
      "version" : "5.5.1",
      "build_hash" : "19c13d0",
      "roles" : [
        "master",
        "data",
        "ingest"
      ],
      "process" : {
        "refresh_interval_in_millis" : 1000,
        "id" : 12641,
        "mlockall" : false
      }
    }
  }
}
```

```
}  
}
```

启动节点后回生成 cluster.name 为文件名的日志文件。

谁先启动谁讲成为master

```
[2017-08-11T17:42:46,018][INFO ][o.e.c.s.ClusterService  ] [node-1] new_master  
{node-1}{-lnKcMBXRpiwExLns0jc9g}{rZcJDIynSzq2Td3yP2kN5A}{10.104.3.2}  
{10.104.3.2:9300}, added {{node-2}{WVsgYi2HT8GwnZU1kUwFwA}  
{X13ShUpAQa2zA1MgcsM3bQ}{10.186.7.221}{10.186.7.221:9300}}, reason: zen-disco-  
elected-as-master ([1] nodes joined)[{node-2}{WVsgYi2HT8GwnZU1kUwFwA}  
{X13ShUpAQa2zA1MgcsM3bQ}{10.186.7.221}{10.186.7.221:9300}]
```

如果master出现故障，其他节点会接管

```
[2017-08-11T17:44:52,797][INFO ][o.e.c.s.ClusterService  ] [node-2] master {new  
{node-2}{WVsgYi2HT8GwnZU1kUwFwA}{v18kQx8sQdGVVohrNqnZ0Q}{10.186.7.221}  
{10.186.7.221:9300}}, removed {{node-1}{-lnKcMBXRpiwExLns0jc9g}  
{rZcJDIynSzq2Td3yP2kN5A}{10.104.3.2}{10.104.3.2:9300}}, added {{node-1}{-  
lnKcMBXRpiwExLns0jc9g}{odnoG9kpQpeX1ltx5KYTSw}{10.104.3.2}{10.104.3.2:9300}},  
reason: zen-disco-elected-as-master ([1] nodes joined)[{node-1}{-  
lnKcMBXRpiwExLns0jc9g}{odnoG9kpQpeX1ltx5KYTSw}{10.104.3.2}{10.104.3.2:9300}]  
[2017-08-11T17:44:53,184][INFO ][o.e.c.r.DelayedAllocationService] [node-2]  
scheduling reroute for delayed shards in [59.5s] (11 delayed shards)  
[2017-08-11T17:44:53,929][INFO ][o.e.c.r.a.AllocationService] [node-2] Cluster  
health status changed from [RED] to [YELLOW] (reason: [shards started  
[[information][0]] ...]).
```

master 节点恢复上线会提示

```
[2017-08-11T17:44:52,855][INFO ][o.e.c.s.ClusterService  ] [node-1]  
detected_master {node-2}{WVsgYi2HT8GwnZU1kUwFwA}{v18kQx8sQdGVVohrNqnZ0Q}  
{10.186.7.221}{10.186.7.221:9300}, added {{node-2}{WVsgYi2HT8GwnZU1kUwFwA}  
{v18kQx8sQdGVVohrNqnZ0Q}{10.186.7.221}{10.186.7.221:9300}}, reason: zen-disco-  
receive(from master [master {node-2}{WVsgYi2HT8GwnZU1kUwFwA}  
{v18kQx8sQdGVVohrNqnZ0Q}{10.186.7.221}{10.186.7.221:9300} committed version  
[44]])
```

负载均衡配置

首先安装 nginx, 这里使用 Netkiller OSCM 一键安装脚本完成。

```
# curl -s  
https://raw.githubusercontent.com/oscm/shell/master/web/nginx/stable/nginx.sh |
```

```
bash
```

因为 elasticsearch 没有用户认证机制我们通常在内网访问他。如果对外提供服务需要增加用户认证。

```
# printf "neo:$(openssl passwd -crypt s3cr3t)n" > /etc/nginx/passwords
```

创建 nginx 配置文件 /etc/nginx/conf.d/elasticsearch.conf

```
upstream elasticsearch {
    server 172.16.0.10:9200;
    server 172.16.0.20:9200;
    server 172.16.0.30:9200;

    keepalive 15;
}

server {
    listen 9200;
    server_name so.netkiller.cn;

    charset utf-8;
    access_log /var/log/nginx/so.netkiller.cn.access.log;
    error_log /var/log/nginx/so.netkiller.cn.error.log;

    auth_basic "Protected Elasticsearch";
    auth_basic_user_file passwords;

    location ~* ^(/_cluster|/_nodes) {
        return 403;
        break;
    }
    location ~* _(open|close) {
        return 403;
        break;
    }
    location / {

        if ($request_filename ~ _shutdown) {
            return 403;
            break;
        }

        if ($request_method !~ ^(GET|HEAD|POST)$) {
            return 403;
        }

        proxy_pass http://elasticsearch;
        proxy_http_version 1.1;
    }
}
```

```
        proxy_set_header Connection "Keep-Alive";
        proxy_set_header Proxy-Connection "Keep-Alive";
    }
}
```

反复使用下面方法请求，最终你会发现 total_opened 会达到你的nginx 配置数量

```
$ curl 'http://test:test@localhost:9200/_nodes/stats/http?pretty' | grep
total_opened
# "total_opened" : 15
```

上面的例子适用于绝大多数场景。

例 8.1. Elasticsearch master / slave

```
upstream elasticsearch {
    server 172.16.0.10:9200;
    server 172.16.0.20:9200 backup;

    keepalive 15;
}

server {
    listen 9200;
    server_name so.netkiller.cn;

    auth_basic "Protected Elasticsearch";
    auth_basic_user_file passwords;

    location ~* ^(/_cluster|/_nodes) {
        return 403;
        break;
    }

    location / {

        if ($request_filename ~ _shutdown) {
            return 403;
            break;
        }
        if ($request_method !~ "HEAD") {
            return 403;
            break;
        }
        if ($request_method ~ "DELETE") {
            return 403;
            break;
        }

        proxy_pass http://elasticsearch;
```

```
        proxy_http_version 1.1;
        proxy_set_header Connection "Keep-Alive";
        proxy_set_header Proxy-Connection "Keep-Alive";
    }
}
```

通过 `limit_except` 可以控制访问权限，例如删除操作。

```
limit_except PUT {
    allow 192.168.1.1;
    deny all;
}
limit_except DELETE {
    allow 192.168.1.1;
    deny all;
}
```

1.4. docker-compose 安装

```
version: '3.8'
services:
  es01:
    image: docker.elastic.co/elasticsearch/elasticsearch:7.9.2
    container_name: es01
    environment:
      - node.name=es01
      - cluster.name=elasticsearch-cluster
      - bootstrap.memory_lock=true
      - discovery.zen.ping.unicast.hosts=es01,es02,es03
      - discovery.zen.minimum_master_nodes=2
      - discovery.zen.ping_timeout=5s
      - node.master=true
      - node.data=true
      - node.ingest=false
      - ES_JAVA_OPTS=-Xms256m -Xmx256m
    ulimits:
      memlock:
        soft: -1
        hard: -1
    volumes:
      - data01:/usr/share/elasticsearch/data
    ports:
      - 9200:9200
      - 9300:9300
    networks:
      - elastic
```



```
es02:
  image: docker.elastic.co/elasticsearch/elasticsearch:7.9.2
  container_name: es02
  environment:
    - node.name=es02
    - cluster.name=elasticsearch-cluster
    - bootstrap.memory_lock=true
    - discovery.zen.ping.unicast.hosts=es01,es02,es03
    - discovery.zen.minimum_master_nodes=2
    - discovery.zen.ping_timeout=5s
    - node.master=true
    - node.data=true
    - node.ingest=false
    - ES_JAVA_OPTS=-Xms256m -Xmx256m
  ulimits:
    memlock:
      soft: -1
      hard: -1
  volumes:
    - data02:/usr/share/elasticsearch/data
  networks:
    - elastic
  depends_on:
    - es01
es03:
  image: docker.elastic.co/elasticsearch/elasticsearch:7.9.2
  container_name: es03
  environment:
    - node.name=es03
    - cluster.name=elasticsearch-cluster
    - bootstrap.memory_lock=true
    - discovery.zen.ping.unicast.hosts=es01,es02,es03
    - discovery.zen.minimum_master_nodes=2
    - discovery.zen.ping_timeout=5s
    - node.master=true
    - node.data=true
    - node.ingest=true
    - ES_JAVA_OPTS=-Xms256m -Xmx256m
  ulimits:
    memlock:
      soft: -1
      hard: -1
  volumes:
    - data03:/usr/share/elasticsearch/data
  networks:
    - elastic
  depends_on:
    - es01
kibana:
  image: docker.elastic.co/kibana/kibana:7.9.2
  container_name: kibana
  environment:
    # SERVER_NAME: kibana.example.org
    ELASTICSEARCH_HOSTS: http://es01:9200
  ports:
    - 5601:5601
  networks:
```

```
    - elastic
  depends_on:
    - es01
volumes:
  data01:
    driver: local
  data02:
    driver: local
  data03:
    driver: local
networks:
  elastic:
    driver: bridge
```

查看节点信息

```
neo@MacBook-Pro-Neo ~/workspace/docker % curl "http://localhost:9200/_cat/nodes?v&pretty"
ip          heap.percent ram.percent cpu load_1m load_5m load_15m node.role
master name
172.19.0.4   48           86 35    4.86    2.90    1.84 dilmrt  *
es03
172.19.0.3   67           86 35    4.86    2.90    1.84 dlmrt  -
es02
172.19.0.2   45           86 35    4.86    2.90    1.84 dlmrt  -
es01
```

1.5. Kubernetes

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: elasticsearch-logging-config
  namespace: kube-public
  labels:
    app: elasticsearch-logging
data:
  limits.conf: |-
    elasticsearch soft memlock unlimited
    elasticsearch hard memlock unlimited
    elasticsearch hard nofile 65536
    elasticsearch soft nofile 65536
---
```

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
```

```
name: elasticsearch-logging
namespace: kube-public
labels:
  app: elasticsearch-logging
spec:
  serviceName: elasticsearch-logging
  replicas: 1
  selector:
    matchLabels:
      app: elasticsearch-logging
  template:
    metadata:
      labels:
        app: elasticsearch-logging
    spec:
      initContainers:
        - name: elasticsearch-logging-init
          image: alpine:latest
          imagePullPolicy: IfNotPresent
          command: ["/sbin/sysctl", "-w", "vm.max_map_count=262144"]
          securityContext:
            privileged: true
      containers:
        - name: elasticsearch-logging
          image: docker.elastic.co/elasticsearch/elasticsearch:7.9.2
          imagePullPolicy: IfNotPresent
          resources:
            limits:
              cpu: 200m
              # memory: "1Gi"
            requests:
              cpu: 200m
              # memory: "1Gi"
          env:
            - name: "NAMESPACE"
              valueFrom:
                fieldRef:
                  fieldPath: metadata.namespace
            - name: node.name
              valueFrom:
                fieldRef:
                  fieldPath: metadata.name
            - name: "cluster.name"
              value: "elasticsearch-cluster"
            - name: "bootstrap.memory_lock"
              # value: "true"
              value: "false"
            - name: "discovery.seed_hosts"
              value: "elasticsearch-logging-0,elasticsearch-logging-1,elasticsearch-logging-2"
            - name: "cluster.initial_master_nodes"
              value: "elasticsearch-logging-0"
            - name: "discovery.find_peers_interval"
              value: "5s"
            - name: "gateway.expected_nodes"
              value: "2"
```

```
- name: "gateway.expected_master_nodes"
  value: "1"

- name: "http.cors.enabled"
  value: "true"
- name: "http.cors.allow-origin"
  value: "*"

- name: "ES_JAVA_OPTS"
  value: "-Xms1g -Xmx1g"
- name: RLIMIT_MEMLOCK
  value: "unlimited"
ports:
- containerPort: 9200
  name: restful
  protocol: TCP
- containerPort: 9300
  name: transport
  protocol: TCP
# readinessProbe:
#   httpGet:
#     scheme: HTTP
#     path: /_cluster/health?local=true
#     port: 9200
#     initialDelaySeconds: 5
# livenessProbe:
#   tcpSocket:
#     port: transport
#     initialDelaySeconds: 20
#     periodSeconds: 10
volumeMounts:
- name: elasticsearch-config
  mountPath: /etc/security/limits.conf
  subPath: limits.conf
- name: elasticsearch-data
  mountPath: /data
volumes:
- name: elasticsearch-data
  emptyDir: {}
# hostPath:
#   path: /data
- name: elasticsearch-config
  configMap:
    name: elasticsearch-logging-config
---
apiVersion: v1
kind: Service
metadata:
  name: elasticsearch-logging
  namespace: kube-public
  labels:
    k8s-app: elasticsearch-logging
    kubernetes.io/cluster-service: "true"
    kubernetes.io/name: "Elasticsearch"
spec:
  selector:
    app: elasticsearch-logging
```

```
type: NodePort
# type: ClusterIP
# clusterIP: None
ports:
- name: restful
  port: 9200
  protocol: TCP
  targetPort: restful
  nodePort: 30092
- name: transport
  port: 9300
  targetPort: transport
```

1.6. 安装指定版本的 Elasticsearch

使用 yum 安装默认为最新版本，我们常常会遇到一个问题 elasticsearch-analysis-ik 的版本晚于 Elasticsearch。如果使用 yum 安装 Elasticsearch 可能 elasticsearch-analysis-ik 插件不支持这个版本，有些版本的 elasticsearch-analysis-ik 可以修改插件配置文件中的版本号，使其与 elasticsearch 版本相同，可以欺骗 elasticsearch 跳过版本不一致异常。

最佳的解决方案是去 [elasticsearch-analysis-ik github](#) 找到兼容的版本，安装我们安装 elasticsearch-analysis-ik 的版本需求来指定安装 elasticsearch

```
Versions
IK version      ES version
master 5.x -> master
5.6.0 5.6.0
5.5.3 5.5.3
5.4.3 5.4.3
5.3.3 5.3.3
5.2.2 5.2.2
5.1.2 5.1.2
1.10.1 2.4.1
1.9.5 2.3.5
1.8.1 2.2.1
1.7.0 2.1.1
1.5.0 2.0.0
1.2.6 1.0.0
1.2.5 0.90.x
1.1.3 0.20.x
1.0.0 0.16.2 -> 0.19.0
```

最新版是 elasticsearch 5.6.1 但分词插件 elasticsearch-analysis-ik 仅能支持到 elasticsearch 版本是 5.6.0

```
root@netkiller /var/log % yum --showduplicates list elasticsearch | expand |
```

```
tail
Repository epel is listed more than once in the configuration
elasticsearch.noarch          5.5.3-1          elasticsearch-5.x
elasticsearch.noarch          5.6.0-1          elasticsearch-5.x
elasticsearch.noarch          5.6.1-1          elasticsearch-5.x
```

安装 5.6.0

```
# yum install elasticsearch-5.6.0-1

Loaded plugins: fastestmirror, langpacks
Repository epel is listed more than once in the configuration
Loading mirror speeds from cached hostfile
Resolving Dependencies
--> Running transaction check
---> Package elasticsearch.noarch 0:5.6.0-1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
=====
=====
Package                               Arch
Version                               Repository
Size
=====
=====
Installing:
  elasticsearch                       noarch
5.6.0-1                               elasticsearch-5.x
32 M

Transaction Summary
=====
=====
=====
Install 1 Package

Total download size: 32 M
Installed size: 36 M
Is this ok [y/d/N]: y
```

1.7. Plugin

Elasticsearch 提供了插件管理命令 `elasticsearch-plugin`

```
root@netkiller ~ % /usr/share/elasticsearch/bin/elasticsearch-plugin -h
A tool for managing installed elasticsearch plugins
```

```
Commands
-----
list - Lists installed elasticsearch plugins
install - Install a plugin
remove - removes a plugin from Elasticsearch

Non-option arguments:
command

Option          Description
-----
-h, --help      show help
-s, --silent    show minimal output
-v, --verbose   show verbose output
```

elasticsearch-analysis-ik

安装插件

```
root@netkiller ~ % /usr/share/elasticsearch/bin/elasticsearch-plugin install
https://github.com/medcl/elasticsearch-analysis-
ik/releases/download/v5.5.1/elasticsearch-analysis-ik-5.5.1.zip
-> Downloading https://github.com/medcl/elasticsearch-analysis-
ik/releases/download/v5.5.1/elasticsearch-analysis-ik-5.5.1.zip
[=====] 100%
-> Installed analysis-ik
```

```
curl -XPOST http://localhost:9200/index/fulltext/_mapping -d'
{
  "properties": {
    "content": {
      "type": "text",
      "analyzer": "ik_max_word",
      "search_analyzer": "ik_max_word"
    }
  }
}'
```

elasticsearch-analysis-pinyin

<https://github.com/medcl/elasticsearch-analysis-pinyin>

1.8. Elastic Cloud on Kubernetes

1.9. 发现配置

<https://www.elastic.co/guide/en/elasticsearch/reference/current/modules-discovery-settings.html>

```
cluster.initial_master_nodes 集群初始化的提供的master候选地址，第一次启动时将从该列表中获得master

discovery.seed_hosts 配置该节点会与哪些候选地址进行通信，hostname,ip ,ip+port
discovery.seed_providers 指定种子地址提供的方式，默认settings，也支持其他Discovery Plugins，包括EC2 Discovery, Azure Classic discovery, GCE discovery ( Google Compute Engine discovery )

discovery.seed_providers: file 当指定的时候，可以在该指定文件中填写ip，方便单独维护。
discovery.find_peers_interval 未发现主节点的重试时间，默认1s

discovery.seed_resolver.timeout 种子提供地址的超时时间
discovery.seed_resolver.max_concurrent_resolvers 并发查找的数量

由以下配置提供未填写端口时的端口范围。
transport.profiles.default.port
transport.port
```

即将废弃

多播配置下，节点向集群发送多播请求，其他节点收到请求后会做出响应。配置参数如下：

```
discovery.zen.ping.multicast.group:224.2.2.4    组地址
discovery.zen.ping.multicast.port: 54328      端口
discovery.zen.ping.multicast.ttl:3            广播消息ttl
discovery.zen.ping.multicast.address:null     绑定的地址，null表示绑定所有
可用的网络接口
discovery.zen.ping.multicast.enabled:true     多播自动发现禁用开关
```

单播配置下，节点向指定的主机发送单播请求，配置如下：

```
# discovery.zen.ping.unicast.hosts:host1:port1,host2:port2
discovery.zen.ping.unicast.hosts=es01,es02,es03
discovery.zen.minimum_master_nodes=2
discovery.zen.ping_timeout=5s
```


2. 文档API

2.1. 快速上手

文档通过 `_index`、`_type`、`_id` 元数据(metadata)，确定 URL 唯一

```
GET /<_index>/<_type>/<_id>
```

```
# curl -XPUT 'http://localhost:9200/website/profile/1' -d '{
  "name" : "neo",
  "nickname" : "netkiller",
  "age" : "35",
  "message" : "Helloworld !!!"
}'

# curl -XGET 'http://localhost:9200/website/profile/1?pretty'
{
  "_index" : "website",
  "_type" : "profile",
  "_id" : "1",
  "_version" : 1,
  "found" : true,
  "_source" : {
    "name" : "neo",
    "nickname" : "netkiller",
    "age" : "35",
    "message" : "Helloworld !!!"
  }
}

# curl -XPUT 'http://localhost:9200/website/blog/1?pretty' -d '{
>   "title": "My first blog entry",
>   "text": "Just trying this out...",
>   "date": "2014/01/01"
> }'
{
  "_index" : "website",
  "_type" : "blog",
  "_id" : "1",
  "_version" : 1,
  "_shards" : {
    "total" : 2,
    "successful" : 1,
    "failed" : 0
  },
  "created" : true
}
```

后面会详细讲解 PUT与GET的使用方法以及相关参数

2.2. 写入 PUT/POST

通过 PUT 写入数据

```
[root@localhost ~]# curl -XPUT 'http://localhost:9200/twitter/tweet/1' -d '{
>   "user" : "kimchy",
>   "post_date" : "2009-11-15T14:12:12",
>   "message" : "trying out Elasticsearch"
> }'
{"_index":"twitter","_type":"tweet","_id":"1","_version":1,"_shards":
{"total":2,"successful":1,"failed":0},"created":true}
```

使用 UUID 替代 _id, 注意使用UUID 必须使用 POST方式提交, 不能使用 PUT。

```
curl -XPOST 'http://localhost:9200/website/news/?pretty' -d '{
  "title": "My first news entry",
  "text": "Just trying this out..."
}'
{
  "_index" : "website",
  "_type" : "news1",
  "_id" : "AVY0RJrvJRTrBLpmYzBH",
  "_version" : 1,
  "_shards" : {
    "total" : 2,
    "successful" : 1,
    "failed" : 0
  },
  "created" : true
}

curl -XGET 'http://localhost:9200/website/news/AVY0RJrvJRTrBLpmYzBH?pretty'
```

提交后会输出 "_id": "AVY0RJrvJRTrBLpmYzBH", 查询时将此放到放到URL中即可。

2.3. 获取 GET

通过 GET 读取数据

```
[root@localhost ~]# curl -XGET 'http://localhost:9200/twitter/tweet/1'
{"_index":"twitter","_type":"tweet","_id":"1","_version":1,"found":true,"_source":{
  "user" : "kimchy",
  "post_date" : "2009-11-15T14:12:12",
  "message" : "trying out Elasticsearch"
}}
```

```
}}
```

_source

只返回 `_source` 数据，去掉元数据

```
# curl -XGET 'http://localhost:9200/website/news1/AVY0Q4SqdtH0Up0t-WB2/_source?pretty'
{
  "title" : "My first news entry",
  "text" : "Just trying this out..."
}
```

选择字段 `_source=title`，超过一个字段使用逗号分隔 `_source=title,text`。

```
# curl -XGET 'http://localhost:9200/website/news1/AVY0Q4SqdtH0Up0t-WB2?_source=title&pretty'
{
  "_index" : "website",
  "_type" : "news1",
  "_id" : "AVY0Q4SqdtH0Up0t-WB2",
  "_version" : 1,
  "found" : true,
  "_source" : {
    "title" : "My first news entry"
  }
}

# curl -XGET 'http://localhost:9200/website/news1/AVY0Q4SqdtH0Up0t-WB2?_source=title,text&pretty'
{
  "_index" : "website",
  "_type" : "news1",
  "_id" : "AVY0Q4SqdtH0Up0t-WB2",
  "_version" : 1,
  "found" : true,
  "_source" : {
    "text" : "Just trying this out...",
    "title" : "My first news entry"
  }
}
```

2.4. 检查记录是否存在

```
[root@localhost elasticsearch]# curl -i -XHEAD
```

```
http://localhost:9200/website/blog/1
HTTP/1.1 200 OK
Content-Type: text/plain; charset=UTF-8
Content-Length: 0

[root@localhost elasticsearch]# curl -i -XHEAD
http://localhost:9200/website/blog/100
HTTP/1.1 404 Not Found
Content-Type: text/plain; charset=UTF-8
Content-Length: 0
```

HTTP/1.1 200 OK 表示已经找到你要的数据

HTTP/1.1 404 Not Found 表示数据不存在

2.5. 删除 Delete

删除 `_index`

```
curl -XDELETE http://localhost:9200/information/?pretty
```

删除 `_mapping`

```
curl -XDELETE http://localhost:9200/information/news/_mapping?pretty
```

删除对象

```
curl -XDELETE http://localhost:9200/information/news/1?pretty
```

2.6. 参数

pretty 格式化 json

```
# curl -XGET 'http://localhost:9200/twitter/tweet/1?pretty'
{
  "_index" : "twitter",
  "_type" : "tweet",
  "_id" : "1",
  "_version" : 1,
  "found" : true,
  "_source" : {
    "user" : "kimchy",
    "post_date" : "2009-11-15T14:12:12",
    "message" : "trying out Elasticsearch"
```

```
}  
}
```

3. 搜索

搜索所有内容

```
# curl -XGET 'http://localhost:9200/_search?pretty'  
# curl -XGET 'http://localhost:9200/_all/_search?pretty'
```

指定 _index 搜索

```
# curl -XGET 'http://localhost:9200/website/_search?pretty'  
# curl -XGET 'http://localhost:9200/website/news/_search?pretty'
```

指定 _type 搜索

```
# curl -XGET 'http://localhost:9200/website,twitter/_search?pretty'  
# curl -XGET 'http://localhost:9200/website/news,blog/_search?pretty'  
# curl -XGET 'http://localhost:9200/website,twitter/news,blog/_search?pretty'
```

所有 _index 包含指定 _type 搜索

```
# curl -XGET 'http://localhost:9200/_all/news,blog/_search?pretty'
```

3.1. URL 搜索

字符串搜索

```
# curl -XGET 'http://localhost:9200/_all/_search?q=neo&pretty'
```

同时满足两个条件

```
+name:neo +age:30
```

查找name为mary 或者 john的数据

```
+name:(mary john)
```

查询姓名是neo或者jam并且年龄小于30岁同时1980-09-10之后出生的

```
+name:(neo jam) +age:<30 +date:>1980-09-10
```

3.2. 分页

该功能与SQL的LIMIT关键字结果一样，Elasticsearch接受size和from两个参数参数：

size: 返回结果集数量，默认10，用法与SQL中的 Limit相同

from: 偏移量，默认0，用法与 SQL中的 Offset相同

如果你想每页显示10个结果，那么请求如下：

```
第一页 GET /_search?size=10  
第二页 GET /_search?size=10&from=10  
第三页 GET /_search?size=10&from=20
```

4. Query DSL

4.1. match 匹配

```
curl -XGET 'http://localhost:9200/information/news/_search?pretty' -d '{
  "query" : {
    "match" : {
      "tag" : "美"
    }
  }
}
```

4.2. multi_match 多字段匹配

multi_match 实现多字段查询

```
curl -XGET 'http://localhost:9200/information/news/_search?pretty' -d '{
  "query": {
    "multi_match": {
      "query": "国际",
      "type": "cross_fields",
      "fields": [ "title", "content" ],
      "operator": "and"
    }
  },
  "from": 0,
  "size": 20,
  "_source": [ "id", "title", "ctime" ],
  "sort": [
    {
      "ctime": { "order": "desc" }
    }
  ]
}
```



```
}  
,
```

4.3. Query bool 布尔条件

Elasticsearch 提供三个布尔条件

must: AND

must_not: NOT

should: OR

must

查询必须满足 tags=天气 and title 包含 台风关键字

```
curl -XPOST  
http://test:123456@so.netkiller.cn/information/article/_search?  
pretty -d'  
{  
  "query": {  
    "bool": {  
      "must": [  
        { "match": { "tags" : "天气" }},  
        { "match": { "title": "台风" } }  
      ]  
    }  
  },  
  "_source":["id","title","ctime"],  
  "highlight" : {  
    "pre_tags" : ["<strong>", "<b>"],  
    "post_tags" : ["</strong >", "</b>"],  
    "fields" : {  
      "content" : {}  
    }  
  }  
}
```

```
}'
```

should

查询必须满足标title or author 两个条件

```
GET /_search
{
  "query": {
    "bool": {
      "should": [
        { "match": { "title": "Linux" }},
        { "match": { "author": "Neo" }}
      ]
    }
  }
}
```

可以嵌套使用

```
GET /_search
{
  "query": {
    "bool": {
      "should": [
        { "match": { "title": "War and Peace" }},
        { "match": { "author": "Leo Tolstoy" }},
        { "bool": {
          "should": [
            { "match": { "translator": "Constance Garnett" }},
            { "match": { "translator": "Louise Maude" }}
          ]
        }}
      ]
    }
  }
}
```

must_not

4.4. filter 过滤

query 相当于 SQL 中的 LIKE 匹配，filter 更像是 where 条件。下面的例子查询 site_id = 23 的数据并且 tags 包含“头条”关键字

```
curl -XGET
'http://test:123456@so.netkiller.cn/information/article/_search?pretty' -d '
{
  "query": {
    "bool": {
      "must": {
        "match": {
          "tags": "头条"
        }
      },
      "filter": {
        "term": {
          "site_id" : "23"
        }
      }
    }
  }
}'
```

4.5. sort 排序

```
curl -XGET 'http://localhost:9200/information/news/_search?pretty' -d '
{
  "query" : {
    "match" : {"tag" : "美"}
  },
  "sort": {
    "ctime": {"order": "desc", "mode": "min"}
  }
}'
```

```
}  
,
```

4.6. _source

```
curl -XGET 'http://localhost:9200/information/news/_search?pretty' -d '  
{  
  "query" : {  
    "match" : {  
      "tag" : "美"  
    }  
  },  
  "_source":["id","title","ctime"]  
}  
,
```

```
curl -XGET 'http://localhost:9200/information/news/_search?pretty' -d '  
{  
  "_source":["id","title","ctime"],  
  "query" : {  
    "match" : {"tag" : "美"}  
  },  
  "sort": {  
    "ctime": {"order": "desc", "mode": "min"}  
  }  
}  
,
```

4.7. highlight 高亮处理

```
curl -XPOST  
http://test:123456@so.netkiller.cn/information/article/_search  
-d'  
{  
  "query" : { "match" : { "content" : "股市" }},  
  "highlight" : {  
    "pre_tags" : ["<strong>", "<b>"],
```

```
    "post_tags" : ["</strong >", "</b>"],
    "fields" : {
      "content" : {}
    }
  }
}
```

5. 集群管理

查看节点信息

```
root@netkiller /var/log/elasticsearch % curl -XGET localhost:9200/
{
  "name" : "node-1",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "I7jirJ2yTr-f2qrBNorQYA",
  "version" : {
    "number" : "5.6.0",
    "build_hash" : "781a835",
    "build_date" : "2017-09-07T03:09:58.087Z",
    "build_snapshot" : false,
    "lucene_version" : "6.6.0"
  },
  "tagline" : "You Know, for Search"
}
```

5.1. 节点健康状态

```
root@netkiller ~ % curl 'http://localhost:9200/_cat/health?v'
epoch      timestamp cluster      status node.total node.data shards pri relo init unassign
pending_tasks max_task_wait_time active_shards_percent
1502445967 18:06:07 my-application yellow      2          2      17  11   0   0      5
0          -                77.3%
```

```
root@netkiller ~ % curl 'http://localhost:9200/_cluster/health'
{"cluster_name":"my-application","status":"yellow","timed_out":false,"number_of_nodes":2,"number_of_data_nodes":2,"active_primary_shards":11,"active_shards":17,"relocating_shards":0,"initializing_shards":0,"unassigned_shards":5,"delayed_unassigned_shards":0,"number_of_pending_tasks":0,"number_of_in_flight_fetch":0,"task_max_waiting_in_queue_millis":0,"active_shards_percent_as_number":77.27272727272727}
```

5.2. 节点http状态

```
root@netkiller ~ % curl 'localhost:9200/_nodes/stats/http?pretty'
{
  "_nodes" : {
    "total" : 2,
    "successful" : 2,
    "failed" : 0
  },
  "cluster_name" : "my-application",
  "nodes" : {
    "-lnKcMBXRpiwExLns0jc9g" : {
      "timestamp" : 1502446878773,
      "name" : "node-1",
      "transport_address" : "10.104.3.2:9300",
      "host" : "10.104.3.2",
      "ip" : "10.104.3.2:9300",
      "roles" : [
```

```

    "master",
    "data",
    "ingest"
  ],
  "http" : {
    "current_open" : 4,
    "total_opened" : 29
  }
},
"WVsgYi2HT8GwnZUlkUwFwA" : {
  "timestamp" : 1502446878782,
  "name" : "node-2",
  "transport_address" : "10.186.7.221:9300",
  "host" : "10.186.7.221",
  "ip" : "10.186.7.221:9300",
  "roles" : [
    "master",
    "data",
    "ingest"
  ],
  "http" : {
    "current_open" : 0,
    "total_opened" : 2
  }
}
}
}
}
}

```

5.3. 查看master节点

```

root@netkiller ~ % curl 'http://localhost:9200/_cat/nodes?v'
ip          heap.percent ram.percent cpu load_1m load_5m load_15m node.role master name
10.104.3.2  31           98  50    0.20  0.55    0.61 mdi      -      node-1
10.186.7.221 25           99  0     0.00  0.01    0.05 mdi      *      node-2

```

5.4. 查看索引的节点分布

```

root@netkiller ~ % curl 'http://localhost:9200/_cat/shards?v'
index      shard prirep state      docs store ip          node
logstash-api 1      p      STARTED 103 342kb 10.186.7.221 node-2
logstash-api 1      r      STARTED 103 342kb 10.104.3.2  node-1
logstash-api 3      p      STARTED 104 404.1kb 10.186.7.221 node-2
logstash-api 3      r      STARTED 104 404.1kb 10.104.3.2  node-1
logstash-api 4      p      STARTED 117 349.2kb 10.186.7.221 node-2
logstash-api 4      r      STARTED 117 349.2kb 10.104.3.2  node-1
logstash-api 2      p      STARTED 113 405.8kb 10.186.7.221 node-2
logstash-api 2      r      STARTED 113 405.8kb 10.104.3.2  node-1
logstash-api 0      p      STARTED 128 488.4kb 10.186.7.221 node-2
logstash-api 0      r      STARTED 128 488.4kb 10.104.3.2  node-1
.kibana      0      p      STARTED 5 31.2kb 10.186.7.221 node-2
.kibana      0      r      STARTED 5 31.2kb 10.104.3.2  node-1
information 1      p      STARTED 10 122.3kb 10.104.3.2  node-1
information 1      r      UNASSIGNED
information 3      p      STARTED 7 159.6kb 10.104.3.2  node-1
information 3      r      UNASSIGNED
information 4      p      STARTED 8 86.1kb 10.104.3.2  node-1
information 4      r      UNASSIGNED
information 2      p      STARTED 11 160kb 10.104.3.2  node-1
information 2      r      UNASSIGNED
information 0      p      STARTED 9 202.8kb 10.104.3.2  node-1

```

```
information 0 r UNASSIGNED
```

5.5. 索引的开启与关闭

```
POST /{index}/_close      关闭索引  
POST /{index}/_open      打开索引
```

_open

```
root@netkiller /var/log/elasticsearch % curl 'http://localhost:9200/_cat/indices?v'  
health status index          uuid                                pri rep docs.count docs.deleted store.size  
pri.store.size  
close information          oygxIi-dR1eB9NoIZtJrxQ  
green open  logstash-spring 9xXtt_L0QvKHibXFH5gjJQ  5  1      4622      0      62.9mb  
31.4mb  
green open  .kibana          9jBBaOomT02EakZlZqnE-g  1  1        10        0      36.6kb  
18.3kb
```

```
root@netkiller /var/log/elasticsearch % curl -XPOST 'http://localhost:9200/information/_open'  
{ "acknowledged":true}#
```

```
root@netkiller /var/log/elasticsearch % curl 'http://localhost:9200/_cat/indices?v'  
health status index          uuid                                pri rep docs.count docs.deleted store.size  
pri.store.size  
green open  information          oygxIi-dR1eB9NoIZtJrxQ  5  1      2417      5      34.5mb  
17.3mb  
green open  logstash-spring 9xXtt_L0QvKHibXFH5gjJQ  5  1      4622      0      62.9mb  
31.4mb  
green open  .kibana          9jBBaOomT02EakZlZqnE-g  1  1        10        0      36.6kb  
18.3kb
```

_close

```
root@netkiller /var/log/elasticsearch % curl -XPOST 'http://localhost:9200/information/_close'  
{ "acknowledged":true}#
```


6. 中文分词插件管理

6.1. 通过 `elasticsearch-plugin` 命令安装分词插件

```
root@netkiller ~ % /usr/share/elasticsearch/bin/elasticsearch-plugin
install https://github.com/medcl/elasticsearch-analysis-
ik/releases/download/v5.5.1/elasticsearch-analysis-ik-5.5.1.zip
-> Downloading https://github.com/medcl/elasticsearch-analysis-
ik/releases/download/v5.5.1/elasticsearch-analysis-ik-5.5.1.zip
[=====] 100%
-> Installed analysis-ik
```

创建 mapping

```
root@netkiller ~ % curl -XPUT http://localhost:9200/information
root@netkiller ~ % curl -XPOST
http://localhost:9200/information/article/_mapping -d'
{
  "properties": {
    "content": {
      "type": "text",
      "analyzer": "ik_max_word",
      "search_analyzer": "ik_max_word"
    },
    "title": {
      "type": "text",
      "analyzer": "ik_max_word",
      "search_analyzer": "ik_max_word"
    }
  }
}'
root@netkiller ~ % curl
"http://localhost:9200/information/article/_mapping?pretty"
{
  "information" : {
    "mappings" : {
      "article" : {
        "properties" : {
          "content" : {
            "type" : "text",
            "analyzer" : "ik_max_word"
          },
          "title" : {
```

```
        "type" : "text",
        "analyzer" : "ik_max_word"
    }
}
}
}
}
```

6.2. 手工安装插件

```
curl -s
https://raw.githubusercontent.com/oscm/shell/master/search/elasticsearch
/elasticsearch-analysis-ik-5.5.0.sh | bash
```

6.3. 创建索引

```
curl -XPUT http://localhost:9200/information
```

6.4. 删除索引

如果索引已经存在请删除后重新创建索引

```
curl -XDELETE http://localhost:9200/information/news/_mapping?pretty
curl -XDELETE http://localhost:9200/information/?pretty
```

6.5. 配置索引分词插件

```
curl -XPOST http://localhost:9200/information/news/_mapping?pretty -d'
{
  "news": {
    "_all": {
      "analyzer": "ik_max_word",
      "search_analyzer": "ik_max_word",
      "term_vector": "no",
      "store": "false"
    }
  }
}
```

```

    },
    "properties": {
      "content": {
        "type": "text",
        "store": "no",
        "term_vector": "with_positions_offsets",
        "analyzer": "ik_max_word",
        "search_analyzer": "ik_max_word",
        "include_in_all": "true",
        "boost": 8
      }
    }
  }
}'

```

测试分词效果

```

curl -XPOST http://localhost:9200/information/news/ -d'
{"title": "越南胡志明游记·教堂·管风琴的天籁之音", "content": "这是我平生第一次去教堂,也是第一次完整的参加宗教仪式。当我驻足教堂外的時候,耳边传来天籁之音,是管风琴,确切的说是电子风琴。真正的管风琴造价昂贵,管风琴通常需要根据教堂尺寸定制,无法量产。我记得中国只有4座管风琴,深圳音乐厅有一座。"}
'

curl -XPOST http://localhost:9200/information/news/ -d'
{"title": "越南胡志明游记·信仰·法事", "content": "佛经的形成过程是与佛教的发展相始终的,按照佛教发展的时间顺序,最早形成的是小乘佛教三藏,之后形成的是大乘佛教三藏,最后形成的是密宗三藏。"}
'

curl -XPOST http://localhost:9200/information/news/_search -d'
{
  "query" : { "term" : { "content" : "佛经" }},
  "highlight" : {
    "pre_tags" : ["<strong>", "<strong>"],
    "post_tags" : ["</strong>", "</strong>"],
    "fields" : {
      "content" : {}
    }
  }
}
'

curl -XPOST http://localhost:9200/information/news/_search -d'
{
  "query" : { "term" : { "content" : "中国" }},
  "highlight" : {
    "pre_tags" : ["<b>", "<i>"],

```

```
"post_tags" : ["</b>", "</i>"],  
"fields" : {  
  "content" : {}  
}  
}'
```

7. 索引管理

7.1. 查看索引

```
root@netkiller ~ % curl 'http://localhost:9200/_cat/indices?v'
health status index          uuid                                pri rep docs.count
docs.deleted store.size pri.store.size
yellow open   information oygxIi-dR1eB9NoIZtJrxQ    5   1   45
42      731kb    731kb
green  open   .kibana     9jBBaOomTO2EakZlZqnE-g    1   1   5
1      62.5kb   31.2kb
green  open   logstash-api WHXZhn3vRWiuVbhrR8rGoEg    5   1   565
0      3.8mb    1.9mb
```

7.2. 删除索引

```
[root@netkiller logstash]# curl -XDELETE
http://localhost:9200/logstash-api-2017.10.03
```

8. 映射

8.1. 查看 _mapping

```
curl -XGET http://localhost:9200/information/news/_mapping?pretty
```

数据结构如下

```
{
  "information" : {
    "mappings" : {
      "news" : {
        "_all" : {
          "analyzer" : "ik_max_word"
        },
        "properties" : {
          "content" : {
            "type" : "string",
            "boost" : 8.0,
            "term_vector" : "with_positions_offsets",
            "analyzer" : "ik_max_word",
            "include_in_all" : true
          },
          "ctime" : {
            "type" : "string"
          },
          "division_category_id" : {
            "type" : "long"
          },
          "tag" : {
            "type" : "string",
            "boost" : 8.0,
            "term_vector" : "with_positions_offsets",
            "analyzer" : "ik_max_word",
            "include_in_all" : true
          },
          "title" : {
            "type" : "string",
```

```
        "boost" : 8.0,  
        "term_vector" : "with_positions_offsets",  
        "analyzer" : "ik_max_word",  
        "include_in_all" : true  
    }  
  }  
}

}
```

8.2. 删除 _mapping

```
curl -XDELETE http://localhost:9200/information/news/_mapping?  
pretty
```

8.3. 创建 _mapping

```
curl -XPOST http://localhost:9200/information/news/_mapping?  
pretty -d'  
{  
  "news": {  
    "_all": {  
      "analyzer": "ik_max_word",  
      "search_analyzer": "ik_max_word",  
      "term_vector": "no",  
      "store": "false"  
    },  
    "properties": {  
      "content": {  
        "type": "string",  
        "store": "no",  
        "term_vector": "with_positions_offsets",  
        "analyzer": "ik_max_word",  
        "search_analyzer": "ik_max_word",  
        "include_in_all": "true",  
        "boost": 8  
      }  
    }  
  }  
}
```

```
}  
}'
```

8.4. 更新 mapping

注意：更新只能用于空的index，如果index中存在数据无法修改 _mapping，必须重建，或者采用别名方案

更新已存在的 mapping，首先我们创建一个 _mapping

```
% curl "http://localhost:9200/information/article/_mapping?  
pretty"  
{  
  "information" : {  
    "mappings" : {  
      "article" : {  
        "properties" : {  
          "content" : {  
            "type" : "text",  
            "analyzer" : "ik_max_word"  
          },  
          "title" : {  
            "type" : "text",  
            "analyzer" : "ik_max_word"  
          }  
        }  
      }  
    }  
  }  
}
```

在这个 _mapping 中增加 ctime 字段，定义时间格式为 yyyy-MM-dd HH:mm:ss


```
% curl -XPOST
http://localhost:9200/information/article/_mapping -d'
{
  "properties": {
    "ctime": {
      "type": "date",
      "format": "yyyy-MM-dd HH:mm:ss"
    }
  }
}'
```

查看预期结果

```
% curl "http://localhost:9200/information/article/_mapping?
pretty"
{
  "information" : {
    "mappings" : {
      "article" : {
        "properties" : {
          "content" : {
            "type" : "text",
            "analyzer" : "ik_max_word"
          },
          "ctime" : {
            "type" : "date",
            "format" : "yyyy-MM-dd HH:mm:ss"
          },
          "title" : {
            "type" : "text",
            "analyzer" : "ik_max_word"
          }
        }
      }
    }
  }
}
```

8.5. 修改 _mapping

修改流程需要经历五步，首先创建新索引，创建新_mapping，导入数据，索引别名，删除旧索引。

当然你也可以删除重建索引，为什么会这么折腾呢？因为这样不用停止业务的情况下进行迁移。

```
# curl -XGET
http://localhost:9200/information_v1/news/_mapping?pretty
{
  "information_v1" : {
    "mappings" : {
      "news" : {
        "_all" : {
          "analyzer" : "ik_max_word"
        },
        "properties" : {
          "content" : {
            "type" : "string",
            "boost" : 8.0,
            "term_vector" : "with_positions_offsets",
            "analyzer" : "ik_max_word",
            "include_in_all" : true
          },
          "ctime" : {
            "type" : "string"
          },
          "division_category_id" : {
            "type" : "long"
          },
          "tag" : {
            "type" : "string",
            "boost" : 8.0,
            "term_vector" : "with_positions_offsets",
            "analyzer" : "ik_max_word",
            "include_in_all" : true
          },
          "title" : {
            "type" : "string",
```

```
        "boost" : 8.0,
        "term_vector" : "with_positions_offsets",
        "analyzer" : "ik_max_word",
        "include_in_all" : true
    }
}
}
}
}
```

注意 ctime 数据类型定义错误，现在需要将它改为date日期类型。

创建 information_v2 索引

```
curl -XPUT http://localhost:9200/information_v2
curl -XPOST http://localhost:9200/information_v2/news/_mapping?
pretty -d'
{
  "news": {
    "_all": {
      "analyzer": "ik_max_word",
      "search_analyzer": "ik_max_word",
      "term_vector": "no",
      "store": "false"
    },
    "properties": {
      "title": {
        "type": "string",
        "store": "no",
        "term_vector": "with_positions_offsets",
        "analyzer": "ik_max_word",
        "search_analyzer": "ik_max_word",
        "include_in_all": "true",
        "boost": 8
      },
      "content": {
        "type": "string",
        "store": "no",
```

```

        "term_vector": "with_positions_offsets",
        "analyzer": "ik_max_word",
        "search_analyzer": "ik_max_word",
        "include_in_all": "true",
        "boost": 8
      },
      "tag": {
        "type": "string",
        "store": "no",
        "term_vector": "with_positions_offsets",
        "analyzer": "ik_max_word",
        "search_analyzer": "ik_max_word",
        "include_in_all": "true",
        "boost": 8
      },
      "ctime": {
        "type": "date"
      }
    }
  }
}'

```

查看全新 _mapping

```

# curl -XGET
http://localhost:9200/information_v2/news/_mapping?pretty
{
  "information_v2" : {
    "mappings" : {
      "news" : {
        "_all" : {
          "analyzer" : "ik_max_word"
        },
        "properties" : {
          "content" : {
            "type" : "string",
            "boost" : 8.0,
            "term_vector" : "with_positions_offsets",
            "analyzer" : "ik_max_word",
            "include_in_all" : true
          },

```

```
        "ctime" : {
          "type" : "date",
          "format" :
"strict_date_optional_time||epoch_millis"
        },
        "tag" : {
          "type" : "string",
          "boost" : 8.0,
          "term_vector" : "with_positions_offsets",
          "analyzer" : "ik_max_word",
          "include_in_all" : true
        },
        "title" : {
          "type" : "string",
          "boost" : 8.0,
          "term_vector" : "with_positions_offsets",
          "analyzer" : "ik_max_word",
          "include_in_all" : true
        }
      }
    }
  }
}
```

现在导入数据，导入完成后修改别名，将information从information_v1 切换到 information_v2

```
curl -XPOST http://localhost:9200/_aliases -d '{
  "actions": [
    { "remove": {
      "alias": "information",
      "index": "information_v1"
    }},
    { "add": {
      "alias": "information",
      "index": "information_v2"
    }
  }
}
```

```
}  
,
```

当所以切换完成information_v1 已经没有什么用处了，这时可以删除information_v1

```
curl -XDELETE http://localhost:9200/information_v1
```

8.6. 数据类型

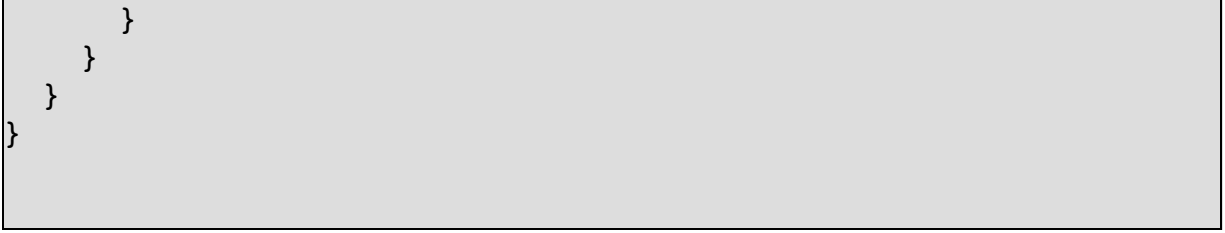
string, date, long, double, boolean or ip.

date

elasticsearch 采用 ISO 8601 标准的 date 格式

```
{"LastUpdate": {  
  "type" : "date",  
  "format" : "yyyy-MM-dd HH:mm:ss"}  
}
```

```
{  
  "mappings": {  
    "my_type": {  
      "properties": {  
        "date": {  
          "type": "date",  
          "format": "yyyy-MM-dd HH:mm:ss || yyyy-MM-dd || epoch_millis"  
        }  
      }  
    }  
  }  
}
```



9. Alias management 别名管理

9.1. 查看索引别名

没有设置任何别名将返回下面的数据结构

```
# curl -XGET http://localhost:9200/_aliases?pretty
{
  "information_v1" : {
    "aliases" : { }
  },
  "information_v2" : {
    "aliases" : { }
  }
}
```

information 是 information_v1 的别名

```
# curl -XGET http://localhost:9200/_aliases?pretty
{
  "information_v1" : {
    "aliases" : {
      "information" : { }
    }
  },
  "information_v2" : {
    "aliases" : { }
  }
}
```

9.2. 创建索引别名

```
curl -XPUT http://localhost:9200/information_v1
curl -XPOST http://localhost:9200/information_v1/news/_mapping?
pretty -d'
```



```
{
  "news": {
    "_all": {
      "analyzer": "ik_max_word",
      "search_analyzer": "ik_max_word",
      "term_vector": "no",
      "store": "false"
    },
    "properties": {
      "title": {
        "type": "string",
        "store": "no",
        "term_vector": "with_positions_offsets",
        "analyzer": "ik_max_word",
        "search_analyzer": "ik_max_word",
        "include_in_all": "true",
        "boost": 8
      },
      "content": {
        "type": "string",
        "store": "no",
        "term_vector": "with_positions_offsets",
        "analyzer": "ik_max_word",
        "search_analyzer": "ik_max_word",
        "include_in_all": "true",
        "boost": 8
      },
      "tag": {
        "type": "string",
        "store": "no",
        "term_vector": "with_positions_offsets",
        "analyzer": "ik_max_word",
        "search_analyzer": "ik_max_word",
        "include_in_all": "true",
        "boost": 8
      }
    },
    "ctime": {
      "type": "date"
    }
  }
}
```

```
curl -XPOST http://localhost:9200/_aliases -d '
{
  "actions": [
    { "add": {
      "alias": "information",
      "index": "information_v1"
    }}
  ]
}
'

{"acknowledged":true}
```

查看结果

```
# curl -XGET http://localhost:9200/_aliases?pretty
{
  "information_v1" : {
    "aliases" : {
      "information" : { }
    }
  },
  "information_v2" : {
    "aliases" : { }
  }
}

# curl -XGET http://localhost:9200/information/?pretty
{
  "information_v1" : {
    "aliases" : {
      "information" : { }
    }
  },
  "mappings" : {
    "news" : {
      "_all" : {
        "analyzer" : "ik_max_word"
      }
    },
    "properties" : {
```

```
    "content" : {
      "type" : "string",
      "boost" : 8.0,
      "term_vector" : "with_positions_offsets",
      "analyzer" : "ik_max_word",
      "include_in_all" : true
    },
    "ctime" : {
      "type" : "date",
      "format" :
"strict_date_optional_time||epoch_millis"
    },
    "tag" : {
      "type" : "string",
      "boost" : 8.0,
      "term_vector" : "with_positions_offsets",
      "analyzer" : "ik_max_word",
      "include_in_all" : true
    },
    "title" : {
      "type" : "string",
      "boost" : 8.0,
      "term_vector" : "with_positions_offsets",
      "analyzer" : "ik_max_word",
      "include_in_all" : true
    }
  }
}
},
"settings" : {
  "index" : {
    "creation_date" : "1471929807430",
    "number_of_shards" : "5",
    "number_of_replicas" : "1",
    "uuid" : "gWl8TTT-QnKbKj2Bg1fG-w",
    "version" : {
      "created" : "2030599"
    }
  }
},
"warmers" : { }
}
```

9.3. 修改别名

```
curl -XPOST http://localhost:9200/_aliases -d '{
  "actions": [
    { "remove": {
      "alias": "information",
      "index": "information_v1"
    }},
    { "add": {
      "alias": "information",
      "index": "information_v2"
    }}
  ]
}'
```

9.4. 删除别名

```
curl -XPOST http://localhost:9200/_aliases -d '{
  "actions": [
    { "remove": {
      "alias": "information", "index": "information_v2"
    }}
  ]
}'
```

10. Example

10.1. 新闻资讯应用案例

```
curl -XDELETE
http://localhost:9200/information_v1/news/_mapping?pretty
curl -XDELETE http://localhost:9200/information_v1/?pretty

curl -XPUT http://localhost:9200/information_v1

curl -XPOST http://localhost:9200/information_v1/news/_mapping?
pretty -d'
{
  "news": {
    "_all": {
      "analyzer": "ik_max_word",
      "search_analyzer": "ik_max_word",
      "term_vector": "no",
      "store": "false"
    },
    "properties": {
      "id": {
        "type": "long"
      },
      "title": {
        "type": "string",
        "store": "no",
        "term_vector": "with_positions_offsets",
        "analyzer": "ik_max_word",
        "search_analyzer": "ik_max_word",
        "include_in_all": "true",
        "boost": 8
      },
      "content": {
        "type": "string",
        "store": "no",
        "term_vector": "with_positions_offsets",
        "analyzer": "ik_max_word",
        "search_analyzer": "ik_max_word",
        "include_in_all": "true",

```

```
        "boost": 8
        },
        "tag": {
        "type": "string",
        "store": "no",
        "term_vector": "with_positions_offsets",
        "analyzer": "ik_max_word",
        "search_analyzer": "ik_max_word",
        "include_in_all": "true",
        "boost": 8
        },
        "ctime": {
        "type": "date"
        }
    }
}
}'

curl -XPOST http://localhost:9200/_aliases -d '
{
  "actions": [
    { "add": {
      "alias": "information",
      "index": "information_v1"
    }}
  ]
}
'
```

```
curl -XGET http://localhost:9200/information/?pretty
```

```
curl -XPOST 'http://localhost:9200/information/news/1?pretty' -
d '{
  "id":1,
  "title":"新闻标题",
  "content":"新闻内容",
  "tag":"新闻标签",
  "ctime":"2011-11-11T11:11:11"
}'
```

```
# curl -XGET 'http://localhost:9200/information/news/1?pretty'
{
  "_index" : "information_v1",
  "_type" : "news",
```

```
"_id" : "1",
"_version" : 1,
"found" : true,
"_source" : {
  "id" : 1,
  "title" : "新闻标题",
  "content" : "新闻内容",
  "tag" : "新闻标签",
  "ctime" : "2011-11-11T11:11:11"
}
}
```

```
curl -XPOST http://localhost:9200/information/news/_search?
pretty -d'
```

```
{
  "query" : { "term" : { "content" : "新闻" }},
  "highlight" : {
    "pre_tags" : ["<b>", "<b>"],
    "post_tags" : ["</b>", "</b>"],
    "fields" : {
      "content" : {}
    }
  }
}'
```

```
curl -XPOST http://localhost:9200/information/news/_search -d'
```

```
{
  "query" : { "term" : { "content" : "王宝强" }},
  "highlight" : {
    "pre_tags" : ["<b>", "<b>"],
    "post_tags" : ["</b>", "</b>"],
    "fields" : {
      "content" : {}
    }
  }
}'
```

```
curl -XPOST http://localhost:9200/information/news/_search -d'
```

```
{
  "query" : { "term" : { "tag" : "娱乐" }},
  "highlight" : {
    "pre_tags" : ["<b>", "<b>"],
    "post_tags" : ["</b>", "</b>"],
    "fields" : {
```

```
        "tag" : {}
      }
    }
  }
}'
```

10.2. 文章搜索案例

```
curl -XDELETE http://localhost:9200/information

curl -XPUT http://localhost:9200/information

curl -H 'Content-Type: application/json' -XPOST
http://localhost:9200/information/article/_mapping -d'
{
  "properties": {
    "title": {
      "type": "text",
      "analyzer": "ik_max_word",
      "search_analyzer": "ik_max_word"
    },
    "description": {
      "type": "text",
      "analyzer": "ik_max_word",
      "search_analyzer": "ik_max_word"
    },
    "content": {
      "type": "text",
      "analyzer": "ik_max_word",
      "search_analyzer": "ik_max_word"
    },
    "ctime": {
      "type": "date",
      "format": "yyyy-MM-dd HH:mm:ss"
    },
    "mtime": {
      "type": "date",
      "format": "yyyy-MM-dd HH:mm:ss"
    }
  }
}
```



```
}  
}'
```

```
curl "http://localhost:9200/information/article/_mapping?  
pretty"
```

11. Migrating MySQL Data into Elasticsearch using logstash

<https://www.elastic.co/guide/en/logstash/current/plugins-inputs-jdbc.html>

11.1. 安装 logstash

安装 JDBC 驱动 和 Logstash

```
curl -s
https://raw.githubusercontent.com/oscm/shell/master/database/mysql/5.7/mysql-connector-java.sh | bash
curl -s
https://raw.githubusercontent.com/oscm/shell/master/search/logstash/logstash-5.x.sh | bash
```

mysql 驱动文件位置在 /usr/share/java/mysql-connector-java.jar

11.2. 配置 logstash

创建配置文件 /etc/logstash/conf.d/jdbc-mysql.conf

```
mysql> desc article;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id             | int(11)       | NO   |     | 0        |       |
| title          | mediumtext    | NO   |     | NULL     |       |
| description     | mediumtext    | YES  |     | NULL     |       |
| author         | varchar(100)  | YES  |     | NULL     |       |
| source         | varchar(100)  | YES  |     | NULL     |       |
| ctime          | datetime      | NO   |     | NULL     |       |
| content        | longtext      | YES  |     | NULL     |       |
+-----+-----+-----+-----+-----+-----+
```

```
7 rows in set (0.00 sec)
```

```
input {
  jdbc {
    jdbc_driver_library => "/usr/share/java/mysql-connector-
java.jar"
    jdbc_driver_class => "com.mysql.jdbc.Driver"
    jdbc_connection_string => "jdbc:mysql://localhost:3306/cms"
    jdbc_user => "cms"
    jdbc_password => "password"
    schedule => "* * * * *"
    statement => "select * from article"
  }
}
output {
  elasticsearch {
    hosts => "localhost:9200"
    index => "information"
    document_type => "article"
    document_id => "%{id}"
  }
}
}
```

11.3. 启动 Logstash

```
root@netkiller /var/log/logstash % systemctl restart logstash

root@netkiller /var/log/logstash % systemctl status logstash
● logstash.service - logstash
   Loaded: loaded (/etc/systemd/system/logstash.service;
   enabled; vendor preset: disabled)
   Active: active (running) since Mon 2017-07-31 09:35:00 CST;
   11s ago
   Main PID: 10434 (java)
   CGroup: /system.slice/logstash.service
```

```
└─10434 /usr/bin/java -XX:+UseParNewGC -
XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=75 -
XX:+UseCMSInitiatingOccupancyOnly -XX:+DisableExplicitGC -
Djava.awt.headless=true -Dfi...

Jul 31 09:35:00 netkiller systemd[1]: Started logstash.
Jul 31 09:35:00 netkiller systemd[1]: Starting logstash...

root@netkiller /var/log/logstash % cat logstash-plain.log
[2017-07-31T09:35:28,169][INFO ]
[logstash.outputs.elasticsearch] Elasticsearch pool URLs
updated {:changes=>{:removed=>[], :added=>
[http://localhost:9200/]}
[2017-07-31T09:35:28,172][INFO ]
[logstash.outputs.elasticsearch] Running health check to see if
an Elasticsearch connection is working
{:healthcheck_url=>http://localhost:9200/, :path=>"/"}
[2017-07-31T09:35:28,298][WARN ]
[logstash.outputs.elasticsearch] Restored connection to ES
instance {:url=>#<Java::JavaNet::URI:0x453a18e9>}
[2017-07-31T09:35:28,299][INFO ]
[logstash.outputs.elasticsearch] Using mapping template from
{:path=>nil}
[2017-07-31T09:35:28,337][INFO ]
[logstash.outputs.elasticsearch] Attempting to install template
{:manage_template=>{"template"=>"logstash-*", "version"=>50001,
"settings"=>{"index.refresh_interval"=>"5s"}, "mappings"=>
{"_default_"=>{"_all"=>{"enabled"=>true, "norms"=>false},
"dynamic_templates"=>[{"message_field"=>
{"path_match"=>"message", "match_mapping_type"=>"string",
"mapping"=>{"type"=>"text", "norms"=>false}}},
{"string_fields"=>{"match"=>"*",
"match_mapping_type"=>"string", "mapping"=>{"type"=>"text",
"norms"=>false, "fields"=>{"keyword"=>{"type"=>"keyword",
"ignore_above"=>256}}}}}], "properties"=>{"@timestamp"=>
{"type"=>"date", "include_in_all"=>false}, "@version"=>
{"type"=>"keyword", "include_in_all"=>false}, "geoip"=>
{"dynamic"=>true, "properties"=>{"ip"=>{"type"=>"ip"},
"location"=>{"type"=>"geo_point"}, "latitude"=>
{"type"=>"half_float"}, "longitude"=>
{"type"=>"half_float"}}}}}}}}
[2017-07-31T09:35:28,344][INFO ]
[logstash.outputs.elasticsearch] Installing elasticsearch
template to _template/logstash
[2017-07-31T09:35:28,465][INFO ]
```

```
[logstash.outputs.elasticsearch] New Elasticsearch output
{:class=>"LogStash::Outputs::ElasticSearch", :hosts=>[#
<Java::JavaNet::URI:0x66df34ae>]}
[2017-07-31T09:35:28,483][INFO ][logstash.pipeline          ]
Starting pipeline {"id"=>"main", "pipeline.workers"=>8,
"pipeline.batch.size"=>125, "pipeline.batch.delay"=>5,
"pipeline.max_inflight"=>1000}
[2017-07-31T09:35:29,562][INFO ][logstash.pipeline          ]
Pipeline main started
[2017-07-31T09:35:29,700][INFO ][logstash.agent           ]
Successfully started Logstash API endpoint {:port=>9600}
[2017-07-31T09:36:01,019][INFO ][logstash.inputs.jdbc    ]
(0.006000s) select * from article
```

11.4. 验证

```
% curl -XGET 'http://localhost:9200/_all/_search?pretty'
```

11.5. 配置模板

全量导入

适合数据没有改变的归档数据或者只能增加没有修改的数据

```
input {
  jdbc {
    jdbc_driver_library => "/usr/share/java/mysql-connector-
java.jar"
    jdbc_driver_class => "com.mysql.jdbc.Driver"
    jdbc_connection_string => "jdbc:mysql://localhost:3306/cms"
    jdbc_user => "cms"
    jdbc_password => "password"
    schedule => "* * * * *"
    statement => "select * from article"
```

```

    }
  }
  output {
    elasticsearch {
      hosts => "localhost:9200"
      index => "information"
      document_type => "article"
      document_id => "%{id}"
    }
  }
}

```

多表导入

多张数据表导入到 Elasticsearch

```

# multiple inputs on logstash jdbc

input {
  jdbc {
    jdbc_driver_library => "/usr/share/java/mysql-connector-
java.jar"
    jdbc_driver_class => "com.mysql.jdbc.Driver"
    jdbc_connection_string => "jdbc:mysql://localhost:3306/cms"
    jdbc_user => "cms"
    jdbc_password => "password"
    schedule => "* * * * *"
    statement => "select * from article"
    type => "article"
  }
  jdbc {
    jdbc_driver_library => "/usr/share/java/mysql-connector-
java.jar"
    jdbc_driver_class => "com.mysql.jdbc.Driver"
    jdbc_connection_string => "jdbc:mysql://localhost:3306/cms"
    jdbc_user => "cms"
    jdbc_password => "password"
    schedule => "* * * * *"
    statement => "select * from comment"
  }
}

```

```

    type => "comment"
  }
}
output {
  elasticsearch {
    hosts => "localhost:9200"
    index => "information"
    document_type => "%{type}"
    document_id => "%{id}"

  }
}

```

需要在每一个jdbc配置项中加入 type 配置，然后 elasticsearch 配置项中加入 document_type => "%{type}"

通过 ID 主键字段增量复制数据

```

input {
  jdbc {
    statement => "SELECT id, mycolumn1, mycolumn2 FROM my_table
WHERE id > :sql_last_value"
    use_column_value => true
    tracking_column => "id"
    tracking_column_type => "numeric"
    # ... other configuration bits
  }
}

```

tracking_column_type => "numeric" 可以声明 id 字段的数据类型，如果不指定将会默认为日期

```

[2017-07-31T11:08:00,193][INFO ][logstash.inputs.jdbc      ]
(0.020000s) select * from article where id > '2017-07-31

```

02:47:00'

如果复制不对称可以加入 `clean_run => true` 配置项，清楚数据

通过日期字段增量复制数据

```
input {
  jdbc {
    statement => "SELECT * FROM my_table WHERE create_date >
:sql_last_value"
    use_column_value => true
    tracking_column => "create_date"
    # ... other configuration bits
  }
}
```

如果复制不对称可以加入 `clean_run => true` 配置项，清楚数据

指定SQL文件

`statement_filepath` 指定 SQL 文件，有时SQL太复杂写入 `statement` 配置项维护部方便，可以将 SQL 写入一个文本文件，然后使用 `statement_filepath` 配置项引用该文件。

```
input {
  jdbc {
    jdbc_driver_library => "/path/to/driver.jar"
    jdbc_driver_class => "org.postgresql.Driver"
    jdbc_url => "jdbc://postgresql"
    jdbc_user => "neo"
    jdbc_password => "password"
    statement_filepath => "query.sql"
  }
}
```


参数传递

将需要复制的条件参数写入 parameters 配置项

```
input {
  jdbc {
    jdbc_driver_library => "mysql-connector-java-5.1.36-
bin.jar"
    jdbc_driver_class => "com.mysql.jdbc.Driver"
    jdbc_connection_string =>
"jdbc:mysql://localhost:3306/mydb"
    jdbc_user => "mysql"
    parameters => { "favorite_artist" => "Beethoven" }
    schedule => "* * * * *"
    statement => "SELECT * from songs where artist =
:favorite_artist"
  }
}
```

控制返回JDBC数据量

```
    jdbc_fetch_size => 1000 #jdbc获取数据的数量大小
    jdbc_page_size => 1000 #jdbc一页的大小,
    jdbc_paging_enabled => true #和jdbc_page_size组合, 将
statement的查询分解成多个查询,相当于: SELECT * FROM table LIMIT
1000 OFFSET 4000
```

输出到不同的 Elasticsearch 中

通过 if [type]=="news" 执行不同的区块, 实现将不同的type输出到指定的 index 中。

```

output {
  if [type]=="news" {
    elasticsearch {
      hosts => "node1.netkiller.cn:9200"
      index => "information"
      document_id => "%{id}"
    }
  }

  if [type]=="comment" {
    elasticsearch {
      hosts => "node2.netkiller.cn:9200"
      index => "information"
      document_id => "%{id}"
    }
  }
}

```

日期格式转换

日期格式化, 将ISO 8601日期格式转换为 %Y-%m-%d %H:%M:%S

```

input {
  jdbc {
    jdbc_driver_library => "/usr/share/java/mysql-connector-java.jar"
    jdbc_driver_class => "com.mysql.jdbc.Driver"
    jdbc_connection_string =>
"jdbc:mysql://127.0.0.1:3306/cms"
    jdbc_user => "cms"
    jdbc_password => "123456"
    schedule => "* * * * *"
    statement => "select * from article limit 5"
  }
}
filter {
  ruby {

```

```

        init => "require 'time'"
        code => "event.set('ctime',
event.get('ctime').time.localtime.strftime('%Y-%m-%d
%H:%M:%S'))"
    }

    ruby {
        init => "require 'time'"
        code => "event.set('mtime',
event.get('mtime').time.localtime.strftime('%Y-%m-%d
%H:%M:%S'))"
    }
}
output {
    stdout {
        codec => rubydebug
    }
}

```

example

下面的例子实现了新数据复制，旧数据更新

```

input {
    jdbc {
        jdbc_driver_library => "/usr/share/java/mysql-connector-
java.jar"
        jdbc_driver_class => "com.mysql.jdbc.Driver"
        jdbc_connection_string => "jdbc:mysql://localhost:3306/cms"
        jdbc_user => "cms"
        jdbc_password => "password"
        schedule => "* * * * *"      #定时cron的表达式,这里是每分钟执行一
次
        statement => "select id, title, description, author,
source, ctime, content from article where id > :sql_last_value"
        use_column_value => true
        tracking_column => "id"
    }
}

```

```

tracking_column_type => "numeric"
record_last_run => true
last_run_metadata_path => "/var/tmp/article.last"
}
jdbc {
  jdbc_driver_library => "/usr/share/java/mysql-connector-
java.jar"
  jdbc_driver_class => "com.mysql.jdbc.Driver"
  jdbc_connection_string => "jdbc:mysql://localhost:3306/cms"
  jdbc_user => "cms"
  jdbc_password => "password"
  schedule => "* * * * *"      #定时cron的表达式,这里是每分钟执行一
次
  statement => "select * from article where ctime >
:sql_last_value"
  use_column_value => true
  tracking_column => "ctime"
  tracking_column_type => "timestamp"
  record_last_run => true
  last_run_metadata_path => "/var/tmp/article-ctime.last"
}
}
output {
  elasticsearch {
    hosts => "localhost:9200"
    index => "information"
    document_type => "article"
    document_id => "%{id}"
    action => "update" # 操作执行的动作,可选值有["index",
"delete", "create", "update"]
    doc_as_upsert => true #支持update模式
  }
}

```

11.6. 解决数据不对称问题

jdbc-input-plugin 只能实现数据库的追加, 对于 elasticsearch 增量写入, 但经常jdbc源一端的数据库可能会做数据库删除或者更新操作。这样一来数据库与搜索引擎的数据库就出现了不对称的情况。

当然你如果有开发团队可以写程序在删除或者更新的时候同步对搜索引擎操作。如果你没有这个能力，可以尝试下面的方法。

这里有一个数据表 `article` , `mtime` 字段定义了 `ON UPDATE CURRENT_TIMESTAMP` 所以每次更新 `mtime` 的时间都会变化

```
mysql> desc article;
+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default          |
| Extra          |               |      |     |                  |
+-----+-----+-----+-----+-----+
| id             | int(11)       | NO   |     | 0                |
| title          | mediumtext    | NO   |     | NULL             |
| description    | mediumtext    | YES  |     | NULL             |
| author         | varchar(100)  | YES  |     | NULL             |
| source         | varchar(100)  | YES  |     | NULL             |
| content        | longtext      | YES  |     | NULL             |
| status         | enum('Y','N') | NO   |     | 'N'              |
| ctime          | timestamp     | NO   |     | CURRENT_TIMESTAMP |
| mtime          | timestamp     | YES  |     | ON UPDATE        |
CURRENT_TIMESTAMP |               |      |     |                  |
+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

logstash 增加 `mtime` 的查询规则



```

jdbc {
  jdbc_driver_library => "/usr/share/java/mysql-connector-
java.jar"
  jdbc_driver_class => "com.mysql.jdbc.Driver"
  jdbc_connection_string => "jdbc:mysql://localhost:3306/cms"
  jdbc_user => "cms"
  jdbc_password => "password"
  schedule => "* * * * *"      #定时cron的表达式,这里是每分钟执行一
次
  statement => "select * from article where mtime >
:sql_last_value"
  use_column_value => true
  tracking_column => "mtime"
  tracking_column_type => "timestamp"
  record_last_run => true
  last_run_metadata_path => "/var/tmp/article-mtime.last"
}

```

创建回收站表，这个事用于解决数据库删除，或者禁用 status = 'N' 这种情况的。

```

CREATE TABLE `elasticsearch_trash` (
  `id` int(11) NOT NULL,
  `ctime` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8

```

为 article 表创建触发器

```

CREATE DEFINER=`dba`@`%` TRIGGER `article_BEFORE_UPDATE` BEFORE
UPDATE ON `article` FOR EACH ROW
BEGIN
  -- 此处的逻辑是解决文章状态变为 N 的时候，需要将搜索引擎中对应的
数据删除。
  IF NEW.status = 'N' THEN

```

```

        insert into elasticsearch_trash(id)
values(OLD.id);
    END IF;
    -- 此处逻辑是修改状态到 Y 的时候，方式elasticsearch_trash仍然存在该文章ID，导致误删除。所以需要删除回收站中得回收记录。
    IF NEW.status = 'Y' THEN
        delete from elasticsearch_trash where id =
OLD.id;
    END IF;
END

CREATE DEFINER=`dba`@`%` TRIGGER `article_BEFORE_DELETE` BEFORE
DELETE ON `article` FOR EACH ROW
BEGIN
    -- 此处逻辑是文章被删除同事将改文章放入搜索引擎回收站。
    insert into elasticsearch_trash(id) values(OLD.id);
END

```

接下来我们需要写一个简单地 Shell 每分钟运行一次，从 elasticsearch_trash 数据表中取出数据，然后使用 curl 命令调用 elasticsearch restful 接口，删除被收回的数据。

11.7. 修改 Mapping

<paraf>需求 Elasticsearch 时间格式从ISO 8601 到 yyyy-MM-dd HH:mm:ss。首先停止 logstash</paraf>

```

systemctl stop logstash

rm -rf /var/tmp/article*

```

修改 /etc/logstash/conf.d/jdbc.conf 配置文件

```

input {

```

```

jdbc {
  jdbc_driver_library => "/usr/share/java/mysql-connector-
java.jar"
  jdbc_driver_class => "com.mysql.jdbc.Driver"
  jdbc_connection_string => "jdbc:mysql://localhost:3306/cms"
  jdbc_user => "cms"
  jdbc_password => "123456"
  schedule => "* * * * *"
  statement => "select * from article where id >
:sql_last_value"
  use_column_value => true
  tracking_column => "id"
  tracking_column_type => "numeric"
  record_last_run => true
  last_run_metadata_path => "/var/tmp/article.last"
}
jjdbc {
  jdbc_driver_library => "/usr/share/java/mysql-connector-
java.jar"
  jdbc_driver_class => "com.mysql.jdbc.Driver"
  jdbc_connection_string => "jdbc:mysql://localhost:3306/cms"
  jdbc_user => "cms"
  jdbc_password => "123456"
  schedule => "* * * * *"      #定时cron的表达式,这里是每分钟执行一
次
  statement => "select * from article where ctime >
:sql_last_value"
  use_column_value => true
  tracking_column => "ctime"
  tracking_column_type => "timestamp"
  record_last_run => true
  last_run_metadata_path => "/var/tmp/article-ctime.last"
}
}
filter {
  ruby {
    code => "event.set('ctime',
event.get('[ctime]').time.localtime.strftime('%Y-%m-%d
%H:%M:%S'))"
  }
  ruby {

```



```
        code => "event.set('mtime',
event.get('[mtime]').time.localtime.strftime('%Y-%m-%d
%H:%M:%S'))"
    }
}

output {
  elasticsearch {
    hosts => "localhost:9200"
    index => "information"
    document_type => "article"
    document_id => "%{id}"
    action => "update"
    doc_as_upsert => true
  }
}
```

删除就的index，重新创建，并配置 mapping。

```
curl -XDELETE http://localhost:9200/information

curl -XPUT http://localhost:9200/information

curl -XPOST
http://localhost:9200/information/article/_mapping -d'
{
  "properties": {
    "title": {
      "type": "text",
      "analyzer": "ik_max_word",
      "search_analyzer": "ik_max_word"
    },
    "description": {
      "type": "text",
      "analyzer": "ik_max_word",
      "search_analyzer": "ik_max_word"
    }
  }
}
```

```
    },
    "content": {
      "type": "text",
      "analyzer": "ik_max_word",
      "search_analyzer": "ik_max_word"
    },
    "ctime": {
      "type": "date",
      "format": "yyyy-MM-dd HH:mm:ss"
    },
    "mtime": {
      "type": "date",
      "format": "yyyy-MM-dd HH:mm:ss"
    }
  }
}'

curl "http://localhost:9200/information/article/_mapping?pretty"
```

启动 logstash 重新复制数据。

```
rm -rf /var/log/logstash/*
systemctl start logstash
```

12. ElasticHD

Elasticsearch 可视化DashBoard, 支持Es监控、实时搜索, Index template快捷替换修改, 索引列表信息查看, SQL converts to DSL等

13. 安装 Elasticsearch 2.3

13.1. RPM 安装

```
yum localinstall  
https://download.elastic.co/elasticsearch/release/org/elasticsearch/dist  
ribution/rpm/elasticsearch/2.3.4/elasticsearch-2.3.4.rpm
```

13.2. YUM 安装

```
rpm --import https://packages.elastic.co/GPG-KEY-elasticsearch  
  
cat >> /etc/yum.repos.d/elasticsearch.repo <<EOF  
[elasticsearch-2.x]  
name=Elasticsearch repository for 2.x packages  
baseurl=https://packages.elastic.co/elasticsearch/2.x/centos  
gpgcheck=1  
gpgkey=https://packages.elastic.co/GPG-KEY-elasticsearch  
enabled=1  
EOF  
  
yum install elasticsearch  
  
sudo /bin/systemctl daemon-reload  
sudo /bin/systemctl enable elasticsearch.service
```

查看 RPM 包中所含文件

```
[root@localhost ~]# rpm -ql elasticsearch-2.3.4-1.noarch  
/etc/elasticsearch  
/etc/elasticsearch/elasticsearch.yml  
/etc/elasticsearch/logging.yml  
/etc/elasticsearch/scripts  
/etc/init.d/elasticsearch  
/etc/sysconfig/elasticsearch  
/usr/lib/sysctl.d  
/usr/lib/sysctl.d/elasticsearch.conf  
/usr/lib/systemd/system/elasticsearch.service  
/usr/lib/tmpfiles.d  
/usr/lib/tmpfiles.d/elasticsearch.conf
```

```
/usr/share/elasticsearch/LICENSE.txt
/usr/share/elasticsearch/NOTICE.txt
/usr/share/elasticsearch/README.textile
/usr/share/elasticsearch/bin
/usr/share/elasticsearch/bin/elasticsearch
/usr/share/elasticsearch/bin/elasticsearch-systemd-pre-exec
/usr/share/elasticsearch/bin/elasticsearch.in.sh
/usr/share/elasticsearch/bin/plugin
/usr/share/elasticsearch/lib
/usr/share/elasticsearch/lib/HdrHistogram-2.1.6.jar
/usr/share/elasticsearch/lib/apache-log4j-extras-1.2.17.jar
/usr/share/elasticsearch/lib/commons-cli-1.3.1.jar
/usr/share/elasticsearch/lib/compiler-0.8.13.jar
/usr/share/elasticsearch/lib/compress-lzf-1.0.2.jar
/usr/share/elasticsearch/lib/elasticsearch-2.3.4.jar
/usr/share/elasticsearch/lib/guava-18.0.jar
/usr/share/elasticsearch/lib/hppc-0.7.1.jar
/usr/share/elasticsearch/lib/jackson-core-2.6.6.jar
/usr/share/elasticsearch/lib/jackson-dataformat-cbor-2.6.6.jar
/usr/share/elasticsearch/lib/jackson-dataformat-smile-2.6.6.jar
/usr/share/elasticsearch/lib/jackson-dataformat-yaml-2.6.6.jar
/usr/share/elasticsearch/lib/jna-4.1.0.jar
/usr/share/elasticsearch/lib/joda-convert-1.2.jar
/usr/share/elasticsearch/lib/joda-time-2.9.4.jar
/usr/share/elasticsearch/lib/jsr166e-1.1.0.jar
/usr/share/elasticsearch/lib/jts-1.13.jar
/usr/share/elasticsearch/lib/log4j-1.2.17.jar
/usr/share/elasticsearch/lib/lucene-analyzers-common-5.5.0.jar
/usr/share/elasticsearch/lib/lucene-backward-codecs-5.5.0.jar
/usr/share/elasticsearch/lib/lucene-core-5.5.0.jar
/usr/share/elasticsearch/lib/lucene-grouping-5.5.0.jar
/usr/share/elasticsearch/lib/lucene-highlighter-5.5.0.jar
/usr/share/elasticsearch/lib/lucene-join-5.5.0.jar
/usr/share/elasticsearch/lib/lucene-memory-5.5.0.jar
/usr/share/elasticsearch/lib/lucene-misc-5.5.0.jar
/usr/share/elasticsearch/lib/lucene-queries-5.5.0.jar
/usr/share/elasticsearch/lib/lucene-queryparser-5.5.0.jar
/usr/share/elasticsearch/lib/lucene-sandbox-5.5.0.jar
/usr/share/elasticsearch/lib/lucene-spatial-5.5.0.jar
/usr/share/elasticsearch/lib/lucene-spatial3d-5.5.0.jar
/usr/share/elasticsearch/lib/lucene-suggest-5.5.0.jar
/usr/share/elasticsearch/lib/netty-3.10.5.Final.jar
/usr/share/elasticsearch/lib/secure-sm-1.0.jar
/usr/share/elasticsearch/lib/snakeyaml-1.15.jar
/usr/share/elasticsearch/lib/spatial4j-0.5.jar
/usr/share/elasticsearch/lib/t-digest-3.0.jar
/usr/share/elasticsearch/modules
/usr/share/elasticsearch/modules/lang-expression
/usr/share/elasticsearch/modules/lang-expression/antlr4-runtime-4.5.1-1.jar
/usr/share/elasticsearch/modules/lang-expression/asm-5.0.4.jar
```

```
/usr/share/elasticsearch/modules/lang-expression/asm-commons-5.0.4.jar
/usr/share/elasticsearch/modules/lang-expression/lang-expression-
2.3.4.jar
/usr/share/elasticsearch/modules/lang-expression/lucene-expressions-
5.5.0.jar
/usr/share/elasticsearch/modules/lang-expression/plugin-
descriptor.properties
/usr/share/elasticsearch/modules/lang-expression/plugin-security.policy
/usr/share/elasticsearch/modules/lang-groovy
/usr/share/elasticsearch/modules/lang-groovy/groovy-2.4.6-indy.jar
/usr/share/elasticsearch/modules/lang-groovy/lang-groovy-2.3.4.jar
/usr/share/elasticsearch/modules/lang-groovy/plugin-
descriptor.properties
/usr/share/elasticsearch/modules/lang-groovy/plugin-security.policy
/usr/share/elasticsearch/modules/reindex
/usr/share/elasticsearch/modules/reindex/plugin-descriptor.properties
/usr/share/elasticsearch/modules/reindex/reindex-2.3.4.jar
/usr/share/elasticsearch/plugins
/var/lib/elasticsearch
/var/log/elasticsearch
/var/run/elasticsearch
```

13.3. 测试安装是否正常

启动

```
/etc/init.d/elasticsearch start
```

链接测试

```
[root@localhost ~]# curl -X GET http://localhost:9200/
{
  "name" : "Jack of Hearts",
  "cluster_name" : "elasticsearch",
  "version" : {
    "number" : "2.3.4",
    "build_hash" : "e455fd0c13dceca8dbbdbb1665d068ae55dabe3f",
    "build_timestamp" : "2016-06-30T11:24:31Z",
    "build_snapshot" : false,
    "lucene_version" : "5.5.0"
  },
  "tagline" : "You Know, for Search"
}
```

13.4. Plugin 插件管理

手工安装插件

```
cd /usr/local/src/  
wget https://github.com/medcl/elasticsearch-analysis-  
ik/releases/download/v1.9.4/elasticsearch-analysis-ik-1.9.4.zip  
cd /usr/share/elasticsearch/plugins  
mkdir ik  
cd ik  
unzip /usr/local/src/elasticsearch-analysis-ik-1.9.4.zip  
  
/etc/init.d/elasticsearch restart
```

plugin 命令

```
plugin -install medcl/elasticsearch-analysis-ik/1.9.0
```

插件测试

```
curl -XDELETE http://localhost:9200/information/news/_mapping?pretty  
curl -XDELETE http://localhost:9200/information/?pretty  
  
curl -XPUT http://localhost:9200/information  
curl -XPOST http://localhost:9200/information/news/_mapping?pretty -d'  
{  
  "news": {  
    "_all": {  
      "analyzer": "ik_max_word",  
      "search_analyzer": "ik_max_word",  
      "term_vector": "no",  
      "store": "false"  
    },  
    "properties": {  
      "content": {  
        "type": "string",  
        "store": "no",  
        "term_vector": "with_positions_offsets",  
        "analyzer": "ik_max_word",  
        "search_analyzer": "ik_max_word",  
        "include_in_all": "true",
```

```
        "boost": 8
      }
    }
  }
}'

curl -XPOST http://localhost:9200/information/news/ -d'
{"title": "越南胡志明游记·教堂·管风琴的天籁之音", "content": "这是我平生第一次去教堂，也是第一次完整的参加宗教仪式。当我驻足教堂外的時候，耳边传来天籁之音，是管风琴，确切的说电子风琴。真正的管风琴造价昂贵，管风琴通常需要根据教堂尺寸定制，无法量产。我记得中国只有4座管风琴，深圳音乐厅有一座。"}
'

curl -XPOST http://localhost:9200/information/news/ -d'
{"title": "越南胡志明游记·信仰·法事", "content": "佛经的形成过程是与佛教的发展相始终的，按照佛教发展的时间顺序，最早形成的是小乘佛教三藏，之后形成的是大乘佛教三藏，最后形成的是密宗三藏。"}
'

curl -XPOST http://localhost:9200/information/news/_search -d'
{
  "query" : { "term" : { "content" : "佛经" }},
  "highlight" : {
    "pre_tags" : ["<strong>", "<strong>"],
    "post_tags" : ["</strong>", "</strong>"],
    "fields" : {
      "content" : {}
    }
  }
}
}'

curl -XPOST http://localhost:9200/information/news/_search -d'
{
  "query" : { "term" : { "content" : "中国" }},
  "highlight" : {
    "pre_tags" : ["<b>", "<i>"],
    "post_tags" : ["</b>", "</i>"],
    "fields" : {
      "content" : {}
    }
  }
}
}'
```


14. FAQ

14.1. Plugin [analysis-ik] is incompatible with Elasticsearch [2.3.5]. Was designed for version [2.3.4]

```
                <![CDATA[
[2016-08-20 19:18:40,930][INFO ][node                ]
[Morg] version[2.3.5], pid[31494], build[90f439f/2016-07-
27T10:36:52Z]
[2016-08-20 19:18:40,930][INFO ][node                ]
[Morg] initializing ...
[2016-08-20 19:18:41,360][ERROR][bootstrap          ]
Exception
java.lang.IllegalArgumentException: Plugin [analysis-ik] is
incompatible with Elasticsearch [2.3.5]. Was designed for
version [2.3.4]
    at
org.elasticsearch.plugins.PluginInfo.readFromProperties(PluginIn
fo.java:118)
    at
org.elasticsearch.plugins.PluginsService.getPluginBundles(Plugin
sService.java:378)
    at org.elasticsearch.plugins.PluginsService.<init>
(PluginsService.java:128)
    at org.elasticsearch.node.Node.<init>(Node.java:158)
    at org.elasticsearch.node.Node.<init>(Node.java:140)
    at
org.elasticsearch.node.NodeBuilder.build(NodeBuilder.java:143)
    at
org.elasticsearch.bootstrap.Bootstrap.setup(Bootstrap.java:178)
    at
org.elasticsearch.bootstrap.Bootstrap.init(Bootstrap.java:270)
    at
org.elasticsearch.bootstrap.Elasticsearch.main(Elasticsearch.jav
a:35)
```

解决方案

```
cd /usr/share/elasticsearch/plugins/ik
vim plugin-descriptor.properties

elasticsearch.version=2.3.4
改为
elasticsearch.version=2.3.5
```

14.2. plugin [analysis-ik] is incompatible with version [5.6.1]; was designed for version [5.5.2]

解决方案

```
root@netkiller /var/log/elasticsearch %
/usr/share/elasticsearch/bin/elasticsearch-plugin list
analysis-ik
WARNING: plugin [analysis-ik] is incompatible with version
[5.6.1]; was designed for version [5.5.2]
```

```
root@netkiller /var/log/elasticsearch %
/usr/share/elasticsearch/bin/elasticsearch-plugin remove
analysis-ik --purge
-> removing [analysis-ik]...
```

手工安装 5.6.0 然后

```
vim /usr/share/elasticsearch/plugins/analysis-ik/plugin-
descriptor.properties

elasticsearch.version=5.5.2
改为
elasticsearch.version=5.6.1
```

14.3. mapper_parsing_exception: failed to parse [ctime]

date 各位为YYYY-MM-ddTHH:mm:ss, 注意中间的字幕T

```
{"type": "date", "format": "YYYY-MM-dd'T'HH:mm:ss.SSSZ"}
curl -XPOST "http://localhost:9200/netkiller/news/" -d'
{
  "content": "Hello World!",
  "CreateDate": "2009-11-15T12:12:12"
}'
```

14.4. 配置 JAVA_HOME

编辑 /etc/sysconfig/elasticsearch 配置文件

```
# Elasticsearch Java path
JAVA_HOME=/srv/java
```

14.5. memory locking requested for elasticsearch process but memory is not locked

```
ERROR: [1] bootstrap checks failed
[1]: memory locking requested for elasticsearch process but
memory is not locked
```

第 9 章 Solr

solr-5.3.0

1. 安装

安装

```
yum install -y unzip java-1.8.0-openjdk

wget http://www.us.apache.org/dist/lucene/solr/5.3.0/solr-5.3.0.tgz
tar zxvf solr-5.3.0.tgz
mv solr-5.3.0 /srv/
ln -s /srv/solr-5.3.0/ /srv/solr

adduser -d /srv/solr -c "Apache Solr" solr
chown solr:solr -R /srv/solr-5.3.0

cp /srv/solr-5.3.0/bin/init.d/solr /etc/init.d/
sed -i 's:/opt/solr:/srv/solr:' /etc/init.d/solr
sed -i 's:/var/solr:/srv/solr/bin:' /etc/init.d/solr

chkconfig --add solr
chkconfig solr on
```

启动与停止

```
# service solr start
Waiting up to 30 seconds to see Solr running on port 8983 [/]
Started Solr server on port 8983 (pid=61909). Happy searching!

# service solr stop
Sending stop command to Solr running on port 8983 ... waiting 5
seconds to allow Jetty process 61909 to stop gracefully.
```

请使用service 或者 /etc/init.d/solr启动，不建议使用root用户如下启动，会造成日志文件无权限等问题。

```
# /srv/solr/bin/solr start
Waiting up to 30 seconds to see Solr running on port 8983 [/]
Started Solr server on port 8983 (pid=56697). Happy searching!
```

我们将solr启动后交给solr用户完成。

```
# ps aux | grep solr
solr      62345 14.5  4.1 4103804 158960 ?        Sl    04:15
0:02 java -server -Xss256k -Xms512m -Xmx512m -XX:NewRatio=3 -
XX:SurvivorRatio=4 -XX:TargetSurvivorRatio=90 -
XX:MaxTenuringThreshold=8 -XX:+UseConcMarkSweepGC -
XX:+UseParNewGC -XX:ConcGCThreads=4 -XX:ParallelGCThreads=4 -
XX:+CMSScavengeBeforeRemark -XX:PretenureSizeThreshold=64m -
XX:+UseCMSInitiatingOccupancyOnly -
XX:CMSInitiatingOccupancyFraction=50 -
XX:CMSMaxAbortablePrecleanTime=6000 -
XX:+CMSParallelRemarkEnabled -XX:+ParallelRefProcEnabled -
verbose:gc -XX:+PrintHeapAtGC -XX:+PrintGCDetails -
XX:+PrintGCDateStamps -XX:+PrintGCTimeStamps -
XX:+PrintTenuringDistribution -
XX:+PrintGCApplicationStoppedTime -
Xloggc:/srv/solr/server/logs/solr_gc.log -Djetty.port=8983 -
DSTOP.PORT=7983 -DSTOP.KEY=solrrocks -Duser.timezone=UTC -
Djetty.home=/srv/solr/server -
Dsolr.solr.home=/srv/solr/server/solr -
Dsolr.install.dir=/srv/solr -jar start.jar -
XX:OnOutOfMemoryError=/srv/solr/bin/oom_solr.sh 8983
/srv/solr/server/logs --module=http
root      62458  0.0  0.0 112640   964 pts/0    S+   04:15
0:00 grep --color=auto solr
```

Solr Admin UI: <http://192.168.4.1:8983/solr/#/>

2. Core Admin

创建 Core

```
$ bin/solr create -c solr_docs

Setup new core instance directory:
/srv/solr/server/solr/solr_docs

Creating new core 'solr_docs' using command:
http://localhost:8983/solr/admin/cores?
action=CREATE&name=solr_docs&instanceDir=solr_docs

{
  "responseHeader":{
    "status":0,
    "QTime":865},
  "core":"solr_docs"}
```

```
$ bin/solr create -c test

Setup new core instance directory:
/srv/solr/server/solr/test

Creating new core 'test' using command:
http://localhost:8983/solr/admin/cores?
action=CREATE&name=test&instanceDir=test

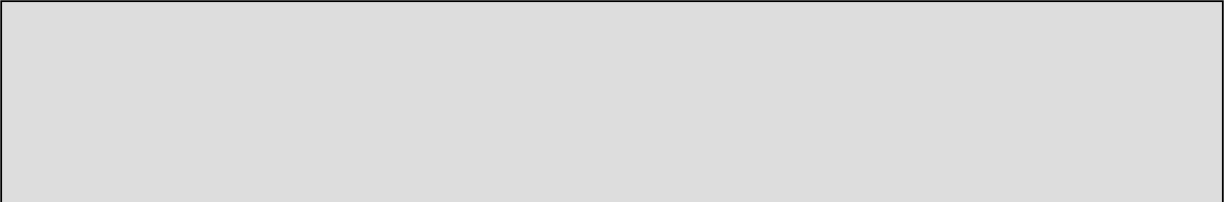
{
  "responseHeader":{
    "status":0,
    "QTime":246},
  "core":"test"}
```

2.1. Schema

```
cd server/solr/
cp -r configsets/sample_techproducts_configs new_core
```



例 9.1. Solr - schema.xml



3. 索引

3.1. Indexing a directory of "rich" files

```
bin/post -c gettingstarted docs/
```

3.2. Indexing Solr XML

```
bin/post -c gettingstarted example/exampledocs/*.xml
```

3.3. Indexing JSON

```
bin/post -c gettingstarted example/exampledocs/books.json
```

3.4. Deleting Data

```
bin/post -c gettingstarted -d "<delete><id>SP2514N</id>  
</delete>"
```


4. 接口

4.1. 查询

```
http://192.168.4.1:8983/solr/solr_docs/browse?q=results
```

5. FAQ

5.1. NOTE: Please install lsof as this script needs it to determine if Solr is listening on port 8983.

```
# /srv/solr/bin/solr start
```

```
NOTE: Please install lsof as this script needs it to determine  
if Solr is listening on port 8983.
```

```
Started Solr server on port 8983 (pid=55873). Happy searching!
```

解决方法

```
yum install -y lsof
```

6. Solr 1.3.0

<http://lucene.apache.org/solr/>

java 采用apt-get安装

例 9.2. /etc/profile.d/java.sh

```
#####  
### Java environment by neo  
#####  
export JAVA_HOME=/usr  
export JRE_HOME=/usr  
export PATH=$PATH:/usr/local/apache-  
tomcat/bin/:/usr/local/jetty-6.1.18/bin  
export CLASSPATH="./:/usr/share/java/:/usr/local/apache-  
solr/example/multicore/lib"  
export JAVA_OPTS="-Xms128m -Xmx1024m"
```

6.1. Embedded Jetty

```
wget http://apache.freelamp.com/lucene/solr/1.3.0/apache-solr-  
1.3.0.tgz  
tar zxvf apache-solr-1.3.0.tgz  
ln -s apache-solr-1.3.0 ../apache-solr  
cd ../apache-solr/example/  
java -jar start.jar
```

multicore: java -Dsolr.solr.home=multicore -jar start.jar

6.2. Jetty

<http://jetty.mortbay.org/jetty/>

过程 9.1. apt-get install

1. install

```
$ sudo apt-get install libxpp3-java  
$ sudo apt-get install solr-jetty
```

2. firewall

```
$ sudo ufw allow 8280
```

3. Testing.

<http://172.16.0.1:8280/>

<http://172.16.0.1:8280/admin/> (user:admin, passwd:admin)

过程 9.2. source codes install

- download

```
wget http://dist.codehaus.org/jetty/jetty-6.1.18/jetty-  
6.1.18.zip
```

6.3. Tomcat

<http://tomcat.apache.org/>

1. download

```
cd /usr/local/src

wget http://apache.etoak.com/tomcat/tomcat-6/v6.0.20/bin/apache-tomcat-6.0.20.tar.gz
wget http://apache.freelamp.com/lucene/solr/1.3.0/apache-solr-1.3.0.tgz

tar zxvf apache-tomcat-6.0.20.tar.gz
ln -s apache-tomcat-6.0.20 ../apache-tomcat

tar zxvf apache-solr-1.3.0.tgz
ln -s apache-solr-1.3.0 ../apache-solr
```

2. solr.xml

```
vim /usr/local/apache-
tomcat/conf/Catalina/localhost/solr.xml

<Context docBase="/usr/local/apache-solr/dist/apache-solr-
1.3.0.war" debug="0" crossContext="true" >
  <Environment name="solr/home" type="java.lang.String"
value="/usr/local/apache-solr/example/solr" override="true"
/>
</Context>
```

6.4. solr-php-client

<http://code.google.com/p/solr-php-client/>

```
wget http://solr-php-
client.googlecode.com/files/SolrPhpClient.2009-03-11.tgz
tar zxvf SolrPhpClient.2009-03-11.tgz
sudo mv SolrPhpClient/Apache /usr/share/php/
```

6.5. multicore

solr.xml

```
vim /usr/local/apache-solr/example/multicore/solr.xml

<?xml version="1.0" encoding="UTF-8" ?>
<solr persistent="false">
  <cores adminPath="/admin/cores">
    <core name="core0" instanceDir="core0" />
    <core name="core1" instanceDir="core1" />

    <core name="article" instanceDir="article" />

  </cores>
</solr>
```

core directory and config file

```
mkdir -p article/conf

vim article/conf/solrconfig.xml

<?xml version="1.0" encoding="UTF-8" ?>
<config>
  <updateHandler class="solr.DirectUpdateHandler2" />
  <requestDispatcher handleSelect="true" >
    <requestParsers enableRemoteStreaming="false"
multipartUploadLimitInKB="2048" />
  </requestDispatcher>
  <requestHandler name="standard"
class="solr.StandardRequestHandler" default="true" />
  <requestHandler name="/update"
class="solr.XmlUpdateRequestHandler" />
  <requestHandler name="/admin/"
class="org.apache.solr.handler.admin.AdminHandlers" />
  <admin>
    <defaultQuery>solr</defaultQuery>
```

```
</admin>
</config>

vim article/conf/schema.xml

<?xml version="1.0" ?>
<schema name="example core zero" version="1.1">
  <types>
    <fieldType name="sint" class="solr.SortableIntField"
sortMissingLast="true" omitNorms="true"/>
    <fieldType name="string" class="solr.StrField"
sortMissingLast="true" omitNorms="true"/>
    <fieldType name="date" class="solr.DateField"
sortMissingLast="true" omitNorms="true"/>
    <fieldType name="text" class="solr.TextField"
positionIncrementGap="100" />
  </types>
  <fields>
    <!-- general -->
    <field name="id" type="sint" indexed="true"
stored="true" multiValued="false" required="true"/>
    <field name="type" type="string" indexed="true"
stored="true" multiValued="false" />
    <field name="name" type="string" indexed="true"
stored="true" multiValued="false" />
    <field name="title" type="string" indexed="true"
stored="true" multiValued="false" />
    <field name="content" type="text" indexed="true"
stored="true" multiValued="false" />
    <field name="timestamp" type="date" indexed="true"
stored="true" default="NOW"/>
  </fields>
  <!-- field to use to determine and enforce document
uniqueness. -->
  <uniqueKey>id</uniqueKey>
  <!-- field for the QueryParser to use when an explicit
fieldname is absent -->
  <defaultSearchField>content</defaultSearchField>
  <!-- SolrQueryParser configuration: defaultOperator="AND|OR" -
-->
  <solrQueryParser defaultOperator="OR"/>
    <copyField source="title" dest="content"/>
    <copyField source="name" dest="content"/>
</schema>
```

commit datas

```
vim test.xml

<add>
  <doc>
    <field name="id">1</field>
    <field name="name">Hello world</field>
  </doc>

  <doc>
    <field name="id">2</field>
    <field name="title">Title Hello world</field>
  </doc>

  <doc>
    <field name="id">3</field>
    <field name="name">Hello world 1</field>
    <field name="content">Content 1</field>
  </doc>

  <doc>
    <field name="id">4</field>
    <field name="name">Name Neo</field>
  </doc>

  <doc>
    <field name="id">5</field>
    <field name="name">Last Chan</field>
  </doc>
</add>

java -Durl=http://localhost:8983/solr/article/update -
Dcommit=yes -jar ../exampledocs/post.jar test.xml
```


6.6. 中文分词

ChineseTokenizerFactory

```
<fieldType name="text" class="solr.TextField" >
  <analyzer>
    <tokenizer
class="org.apache.solr.analysis.ChineseTokenizerFactory"/>
  </analyzer>
</fieldType>
```

CJK

```
<fieldType name="text" class="solr.TextField"
positionIncrementGap="100">
  <analyzer>
    <tokenizer class="solr.CJKTokenizerFactory"/>
  </analyzer>
</fieldType>
```

mmseg4j

<http://code.google.com/p/mmseg4j/>

install

```
$ cd /usr/local/src/
$ wget http://mmseg4j.googlecode.com/files/mmseg4j-1.7.2.zip
$ unzip mmseg4j-1.7.2.zip
$ mkdir /usr/local/apache-solr/example/multicore/lib
```

```
$ cp /usr/local/src/mmseg4j-1.7.2/mmseg4j-all-1.7.2.jar
/usr/local/apache-solr/example/multicore/lib
$ cd mmseg4j-1.7.2/
```

test

```
$ java -Dmmseg.dic.path=/usr/local/apache-solr/example/solr -
jar mmseg4j-all-1.7.2.jar 这里是字符串
$ java -Dmmseg.dic.path=/usr/local/apache-solr/example/solr -cp
.:mmseg4j-all-1.7.2.jar com.chenlb.mmseg4j.example.Simple 这里是
字符串
$ java -Dmmseg.dic.path=/usr/local/apache-solr/example/solr -cp
.:mmseg4j-all-1.7.2.jar com.chenlb.mmseg4j.example.MaxWord 这里
是字符串
```

mmseg4j 在 solr 中主要支持两个参数：mode、dicPath。mode 表示是什么模式分词（有效值：simplex、complex、max-word，如果输入了无效的默认用 max-word。）。dicPath 是词库目录可以是绝对目录，也可以是相对目录（是相对 solr.home 目录下的，dic 就会在 solr.home/dic 目录下找词库文件），如果不指定就是默认在 CWD/data 目录（程序运行当前目录的 data 子目录）下找。

分词例子

```
<fieldtype name="textComplex" class="solr.TextField">
  <analyzer>
    <tokenizer
class="com.chenlb.mmseg4j.solr.MMSegTokenizerFactory"
      mode="complex" dicPath="dic">
    </tokenizer>
  </analyzer>
</fieldtype>

<fieldtype name="textMaxWord" class="solr.TextField">
  <analyzer>
    <tokenizer
```

```

class="com.chenlb.mmseg4j.solr.MMSegTokenizerFactory"
        mode="max-word" dicPath="dic">
        </tokenizer>
    </analyzer>
</fieldtype>

<fieldtype name="textSimple" class="solr.TextField">
    <analyzer>
        <tokenizer
class="com.chenlb.mmseg4j.solr.MMSegTokenizerFactory"
        mode="simple"
dicPath="/usr/local/apache-solr/example/solr/my_dic">
        </tokenizer>
    </analyzer>
</fieldtype>

```

添加到schema.xml

```

<fieldType name="text" class="solr.TextField"
positionIncrementGap="100" >
    <analyzer>
        <tokenizer
class="com.chenlb.mmseg4j.solr.MMSegTokenizerFactory"
mode="complex" dicPath="dic"/>
        <filter class="solr.LowerCaseFilterFactory"/>
    </analyzer>
</fieldType>

```

<http://localhost:8080/solr/admin/analysis.jsp> 在 Field 的下拉菜单选择 name，然后在应用输入 complex。可以看 mmseg4j 的分词的结果。

中文分词“庖丁解牛” Paoding Analysis

```

$ cd /usr/local/src/
$ mkdir paoding-analysis-2.0.4-beta
$ cd paoding-analysis-2.0.4-beta/

```

```
$ wget http://paoding.googlecode.com/files/paoding-analysis-2.0.4-beta.zip
$ unzip paoding-analysis-2.0.4-beta.zip
$ cp paoding-analysis.jar /usr/local/apache-solr/example/multicore/lib/
```

ChineseTokenizerFactory



第 10 章 Nutch

<http://lucene.apache.org/nutch/>

How to Setup Nutch and Hadoop

<http://wiki.apache.org/nutch/NutchHadoopTutorial>

1. 下载

```
$ cd /usr/local/src/  
$ wget http://apache.etoak.com/lucene/nutch/nutch-  
1.0.tar.gz  
$ tar zxvf nutch-1.0.tar.gz  
$ sudo cp -r nutch-1.0 ..  
$ cd ..  
$ sudo ln -s nutch-1.0 apache-nutch
```

2. 创建文件myurl

```
$ cd apache-nutch  
$ mkdir urls  
$ vim urls/myurl  
http://netkiller.8800.org/
```

3. 配置文件 crawl-urlfilter.txt

编辑conf/crawl-urlfilter.txt文件，修改MY.DOMAIN.NAME部分，
把它替换为你想要抓取的域名

```
$ cp conf/crawl-urlfilter.txt conf/crawl-urlfilter.txt.old  
$ vim conf/crawl-urlfilter.txt  
  
# accept hosts in MY.DOMAIN.NAME  
+^http://([a-z0-9]*\.)*MY.DOMAIN.NAME/
```

修改为:

```
# accept hosts in MY.DOMAIN.NAME
+^http://([a-z0-9]*\.)*netkiller.8800.org/
```

4. http.agent.name

```
$ vim conf/nutch-site.xml
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!-- Put site-specific property overrides in this file. -->

<configuration>

<property>
  <name>http.agent.name</name>
  <value>Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US;
rv:1.9.1) Gecko/20090624 Firefox/3.5</value>
  <description>HTTP 'User-Agent' request header. MUST NOT
be empty -
  please set this to a single word uniquely related to your
organization.

  NOTE: You should also check other related properties:

    http.robots.agents
    http.agent.description
    http.agent.url
    http.agent.email
    http.agent.version

  and set their values appropriately.

</description>
</property>

<property>
  <name>http.agent.description</name>
  <value></value>
  <description>Further description of our bot- this text is
used in
  the User-Agent header. It appears in parenthesis after
```

```
the agent name.
  </description>
</property>

<property>
  <name>http.agent.url</name>
  <value>http://netkiller.8800.org/robot.html</value>
  <description>A URL to advertise in the User-Agent header.
This will
  appear in parenthesis after the agent name. Custom
dictates that this
  should be a URL of a page explaining the purpose and
behavior of this
  crawler.
  </description>
</property>

<property>
  <name>http.agent.email</name>
  <value>openunix@163.com</value>
  <description>An email address to advertise in the HTTP
'From' request
  header and User-Agent header. A good practice is to
mangle this
  address (e.g. 'info at example dot com') to avoid
spamming.
  </description>
</property>

</configuration>
```

5. 运行以下命令行开始工作

\$ bin/nutch crawl urls -dir crawl -depth 3 -threads 5

```
bin/nutch crawl <your_url> -dir <your_dir> -depth 2 -
threads 4 >&logs/logs1.log
```

urls 存放需要爬行的url文件的目录，即目录/nutch/urls。

```
-dir  dirnames          设置保存所抓取网页的目录。
-depth depth           表明抓取网页的层次深度
-delay delay           表明访问不同主机的延时，单位为“秒”
-threads threads      表明需要启动的线程数
-topN 50               topN  一个网站保存的最大页面数。

$ nohup bin/nutch crawl /usr/local/apache-nutch/urls -dir
/usr/local/apache-nutch/crawl -depth 5 -threads 50 -topN 50
> /tmp/nutch.log &
```

6. depoly

```
$ cd /usr/local/apache-tomcat/conf/Catalina/localhost
$ vim nutch.xml
<Context docBase="/usr/local/apache-nutch/nutch-1.0.war"
debug="0" crossContext="true" >
</Context>
```

searcher.dir

```
$ vim /usr/local/apache-tomcat/webapps/nutch/WEB-INF/classes/nutch-site.xml

<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!-- Put site-specific property overrides in this file. -->

<configuration>
  <property>
    <name>searcher.dir</name>
    <value>/usr/local/apache-nutch/crawl</value>
  </property>
</configuration>
```




test

<http://172.16.0.1:8080/nutch/>

第 11 章 Sphinx

<http://sphinxsearch.com/>

```
sudo apt-get install sphinxsearch
```

/etc/sphinxsearch/sphinx.conf

```
sudo cp /etc/sphinxsearch/sphinx-min.conf.dist  
/etc/sphinxsearch/sphinx.conf
```

创建测试数据库并导入测试数据

```
$ wget http://sphinxsearch.googlecode.com/svn/trunk/example.sql  
$ mysql -h localhost -uroot -p < example.sql  
$ mysql -h localhost -uroot -p  
CREATE USER 'test'@'localhost' IDENTIFIED BY '';  
GRANT SELECT ON test.* TO 'test'@'localhost';  
FLUSH PRIVILEGES;  
mysql> quit  
  
$ echo "select * from documents" | mysql -utest -p test  
Enter password:  
id      group_id      group_id2      date_added      title  
content  
1       1             5              2011-02-12 15:29:34      test one  
this is my test document number one. also checking search  
within phrases.  
2       1             6              2011-02-12 15:29:34      test two  
this is my test document number two  
3       2             7              2011-02-12 15:29:34      another doc  
this is another group  
4       2             8              2011-02-12 15:29:34      doc number four  
this is to test groups
```

创建索引

sudo indexer <index>

```
$ sudo indexer test1

Sphinx 0.9.8.1-release (r1533)
Copyright (c) 2001-2008, Andrew Aksyonoff

using config file '/etc/sphinxsearch/sphinx.conf'...
indexing index 'test1'...
collected 4 docs, 0.0 MB
sorted 0.0 Mhits, 100.0% done
total 4 docs, 193 bytes
total 0.012 sec, 16531.05 bytes/sec, 342.61 docs/sec
```

```
$ sudo /etc/init.d/sphinxsearch start
Starting sphinx: Sphinx 0.9.8.1-release (r1533)
Copyright (c) 2001-2008, Andrew Aksyonoff

using config file '/etc/sphinxsearch/sphinx.conf'...
creating server socket on 0.0.0.0:3312
sphinx.
```

测试

search "keyword"

```
$ search test
Sphinx 0.9.8.1-release (r1533)
Copyright (c) 2001-2008, Andrew Aksyonoff
```

```
using config file '/etc/sphinxsearch/sphinx.conf'...
index 'test1': query 'test ': returned 3 matches of 3 total in
0.000 sec
```

```
displaying matches:
```

```
1. document=1, weight=2, group_id=1, date_added=Sat Feb 12
15:29:34 2011
```

```
    id=1
```

```
    group_id=1
```

```
    group_id2=5
```

```
    date_added=2011-02-12 15:29:34
```

```
    title=test one
```

```
    content=this is my test document number one. also
```

```
checking search within phrases.
```

```
2. document=2, weight=2, group_id=1, date_added=Sat Feb 12
15:29:34 2011
```

```
    id=2
```

```
    group_id=1
```

```
    group_id2=6
```

```
    date_added=2011-02-12 15:29:34
```

```
    title=test two
```

```
    content=this is my test document number two
```

```
3. document=4, weight=1, group_id=2, date_added=Sat Feb 12
15:29:34 2011
```

```
    id=4
```

```
    group_id=2
```

```
    group_id2=8
```

```
    date_added=2011-02-12 15:29:34
```

```
    title=doc number four
```

```
    content=this is to test groups
```

```
words:
```

```
1. 'test': 3 documents, 5 hits
```

```
wget
```

```
http://sphinxsearch.googlecode.com/svn/trunk/api/sphinxapi.php
```

```
wget http://sphinxsearch.googlecode.com/svn/trunk/api/test.php
```

```
php test.php test
```

第 12 章 Lucene

<http://lucene.apache.org/>

第 13 章 MG4J

<http://mg4j.dsi.unimi.it/>

第 14 章 PhpDig

<http://www.phpdig.net/>

PhpDig is a web spider and search engine written in PHP, using a MySQL database and flat file support. PhpDig builds a glossary with words found in indexed pages. On a search query, it displays a result page containing the search keys, ranked by occurrence.

第 15 章 Mahout

<http://mahout.apache.org/>

第 16 章 Apache Hbase

1. 安装 Apache Hbase

注意：Apache Hbase 不能使用 OpenJDK 启动，需要去 Oracle 网站下载 Server JRE

1.1. 单机模式安装

如果你是第一次安装Hbase，建议你从单机安装开始，这样成功率比较高，不会受挫。Hbase 不比关系型数据库复杂，只是安装比较麻烦，一旦安装号使用起来还是很容易上手的，请直接粘贴复制下面的命令即可完成安装：

```
cd /usr/local/src
wget http://mirrors.hust.edu.cn/apache/hbase/stable/hbase-1.2.6-bin.tar.gz

tar xzf hbase-1.2.6-bin.tar.gz
cp hbase-1.2.6/conf/hbase-site.xml{,.original}
mv hbase-1.2.6 /srv/apache-hbase-1.2.6
ln -s /srv/apache-hbase-1.2.6 /srv/apache-hbase

cp /srv/apache-hbase/conf/hbase-env.sh{,.original}
cat > /srv/apache-hbase/conf/hbase-env.sh <<EOF
export JAVA_HOME=/srv/java
#export HBASE_CLASSPATH=
export HBASE_MANAGES_ZK=true
EOF

cat > /srv/hbase/conf/hbase-site.xml <<EOF
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
    <property>
        <name>hbase.rootdir</name>
```

```
        <value>file:///tmp/hbase-${user.name}</value>
    </property>
</configuration>
EOF
```

启动 Apache Hbase

```
/srv/apache-hbase/bin/start-hbase.sh
```

进入 Hbase shell

```
/srv/apache-hbase/bin/hbase shell
```

关闭 Hbase

```
/srv/apache-hbase/bin/stop-hbase.sh
```

1.2. 伪分布式模式

单机模式基本可能满足我们的学习需要，但无法满足更复杂的需求，例如集成Hive等其他软件，这时我们就需要借助 Hadoop 的HDFS功能实现与其他软件的集成。所谓的伪分布式，就是只有一个Hbase节点，即Master。

这里我假设 Hadoop 已经正确安装，无论你采用什么模式只要能提供 hdfs 服务处即可。Hadoop 安装可以参考作者的相关文档。

首先编辑 conf/hbase-site.xml 配置文件，增加以下配置：

```
<property>
  <name>hbase.cluster.distributed</name>
```

```
<value>true</value>
</property>
```

hbase.cluster.distributed 属性值设置为 true HBase 将运行于分布式模式

然后配置 hbase.rootdir 属性值，指向 HDFS 地址。

```
<property>
  <name>hbase.rootdir</name>
  <value>hdfs://localhost:9000/hbase</value>
</property>
```

现在启动 Hbase ,如果正常使用 jps 可以下面三个线程

```
[hadoop@netkiller conf]$ su - hadoop -c "/srv/apache-hbase/bin/start-hbase.sh"
[hadoop@netkiller conf]$ jps | egrep "(HMaster|HRegionServer|HQuorumPeer)"
```

如果启动正常，你将会看到 HDFS 中的 Hbase 目录。

```
[hadoop@netkiller ~]$ /srv/hadoop/bin/hdfs dfs -ls /hbase
Found 7 items
drwxr-xr-x  - hadoop supergroup          0 2017-06-28 21:55
/hbase/.tmp
drwxr-xr-x  - hadoop supergroup          0 2017-06-28 21:55
/hbase/MasterProcWALs
drwxr-xr-x  - hadoop supergroup          0 2017-06-28 21:55
/hbase/WALs
drwxr-xr-x  - hadoop supergroup          0 2017-06-28 21:55
```

```
/hbase/data
-rw-r--r--    3  hadoop  supergroup          42  2017-06-28  21:55
/hbase/hbase.id
-rw-r--r--    3  hadoop  supergroup          7   2017-06-28  21:55
/hbase/hbase.version
drwxr-xr-x    -  hadoop  supergroup          0   2017-06-28  21:55
/hbase/oldWALS
```

1.3. 分布式模式部署

上一节所讲的伪分不出，就是只有一个Master节点，而真正的分布式摸就是每个节点均独立部署，实现可伸缩，水平扩展，但作为例子这里我们仅仅采用最小化节点配置。

1.4. 运维技巧

检查 Hbase 线程是否启动

```
[neo@netkiller conf]$ jps | grep HMaster
17719 HMaster
```

2. 配置 Apache Hbase

2.1. hbase-env.sh

环境变量配置文件

HBASE_MANAGES_ZK=true 仅用于单机运行，true表示不使用 Zookeeper

2.2. hbase-site.xml

分布式模式的开启与关闭 hbase.cluster.distributed

```
<property>  
  <name>hbase.cluster.distributed</name>  
  <value>true</value>  
</property>
```

3. Hbase Shell

安装完Apache hbase启动后就可以进入 hbase shell了，hbase shell 是与Hbase交互的界面。

```
[neo@netkiller bin]$ hbase shell
2017-06-27 21:07:35,524 WARN [main] util.NativeCodeLoader: Unable to load native-hadoop
library for your platform... using builtin-java classes where applicable
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.2.6, rUnknown, Mon May 29 02:25:32 CDT 2017

hbase(main):001:0>
```

退出 Hbase shell

```
hbase(main):038:0> exit
[neo@netkiller bin]$
```

3.1. 表操作

首先我们做几个基本操作，例如创建表，写入数据，读取数据等等，你会发现Hbase被神话了，操作其实没有那么复杂，远没有关系型数据库复杂。

创建一个名为 t1 的表，使用默认命名空间 namespace=default，这个表有一个列族名(column family) 为 f1。后面会解释什么是 column family，这里你只要对着操作就可以了。

```
hbase(main):002:0> create 't1', 'f1'
0 row(s) in 1.2190 seconds

=> Hbase::Table - t1
```

向t1表插入数据字段名 (key) 是r1,数据值 value 是 'value'

```
hbase(main):014:0> put 't1', 'r1', 'f1', 'value'
0 row(s) in 0.0060 seconds
```

获取表 t1，字段名 (key) 为 r1 的数据

```
hbase(main):032:0> get 't1', 'r1'
COLUMN                                CELL
f1:                                     timestamp=1498613275013, value=value
1 row(s) in 0.0240 seconds
```

列出用户表

```
hbase(main):009:0> list
TABLE
t1
1 row(s) in 0.0450 seconds

=> ["t1"]
hbase(main):010:0>
```

扫描表相当于 select * from t1

```
hbase(main):034:0> scan 't1'
ROW                                COLUMN+CELL
r1                                  column=f1:, timestamp=1498613275013,
value=value
1 row(s) in 0.0140 seconds
```

表的启用与禁用操作

```
hbase(main):010:0> disable 't1'
0 row(s) in 1.3740 seconds

hbase(main):011:0> enable 't1'
0 row(s) in 1.2380 seconds
```

删除表，删除表之前需要先禁用该表，然后使用drop命令删除。

```
hbase(main):036:0> disable 't1'
0 row(s) in 2.2460 seconds

hbase(main):037:0> drop 't1'
0 row(s) in 1.2310 seconds

hbase(main):038:0>
```

4. Web UI

除了 Web Shell Hbase 还提供了 Web UI 地址是:

<http://localhost:16010/master-status>

请确保你的防火墙放行了 16010 端口

```
[neo@netkiller conf]$ iptables-save | grep 16010
-A INPUT -p tcp -m state --state NEW -m tcp --dport 16010 -j
ACCEPT
```


5. Phoenix

[phoenix](#)

Phoenix 是基于 Hbase 的 SQL 层

5.1. 安装 Phoenix

```
cd /usr/local/src/  
wget https://mirrors.tuna.tsinghua.edu.cn/apache/phoenix/apache-  
phoenix-4.12.0-HBase-1.3/bin/apache-phoenix-4.12.0-HBase-1.3-  
bin.tar.gz  
tar zxvf apache-phoenix-4.12.0-HBase-1.3-bin.tar.gz  
cp apache-phoenix-4.12.0-HBase-1.3-bin/phoenix-core-4.12.0-  
HBase-1.3.jar /srv/apache-hbase-1.3.1/lib/  
mv apache-phoenix-4.12.0-HBase-1.3-bin /srv/apache-phoenix-  
4.12.0  
ln -s /srv/apache-phoenix-4.12.0 /srv/apache-phoenix
```

配置环境变量

```
% vim /srv/apache-hbase-1.3.1/conf/hbase-env.sh  
  
export JAVA_HOME=/srv/java  
export HBASE_CLASSPATH=/srv/apache-phoenix  
export HBASE_MANAGES_ZK=true
```

环境配置

```
cat >> ~/.bash_profile << 'EOF'  
export CLASSPATH=$CLASSPATH:/srv/apache-phoenix
```

```
export PATH=$PATH:/srv/apache-phoenix/bin
EOF
```

重启 Hbase

```
su - hadoop -c "/srv/apache-hbase/bin/start-hbase.sh"
```

5.2. sqlline.py 命令行界面

```
[hadoop@VM_7_221_centos ~]$ sqlline.py localhost
```

```
[hadoop@VM_7_221_centos ~]$ sqlline.py
Setting property: [incremental, false]
Setting property: [isolation, TRANSACTION_READ_COMMITTED]
issuing: !connect jdbc:phoenix:localhost:2181:/hbase none none
org.apache.phoenix.jdbc.PhoenixDriver
Connecting to jdbc:phoenix:localhost:2181:/hbase
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/srv/apache-phoenix-4.12.0-
HBase-1.3/phoenix-4.12.0-HBase-1.3-
client.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/srv/apache-hadoop-
2.8.1/share/hadoop/common/lib/slf4j-log4j12-
1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for
an explanation.
17/10/13 10:43:48 WARN util.NativeCodeLoader: Unable to load
native-hadoop library for your platform... using builtin-java
classes where applicable
Connected to: Phoenix (version 4.12)
Driver: PhoenixEmbeddedDriver (version 4.12)
Autocommit status: true
Transaction isolation: TRANSACTION_READ_COMMITTED
Building list of tables and columns for tab-completion (set
fastconnect to true to skip)...
95/95 (100%) Done
```

```
Done
sqlline version 1.2.0
0: jdbc:phoenix:localhost:2181:/hbase>
```

帮助信息

```
0: jdbc:phoenix:localhost> help
!all          Execute the specified SQL against all the
current connections
!autocommit   Set autocommit mode on or off
!batch        Start or execute a batch of statements
!brief        Set verbose mode off
!call         Execute a callable statement
!close        Close the current connection to the database
!closeall     Close all current open connections
!columns      List all the columns for the specified table
!commit       Commit the current transaction (if
autocommit is off)
!connect      Open a new connection to the database.
!dbinfo       Give metadata information about the database
!describe     Describe a table
!dropall      Drop all tables in the current database
!exportedkeys List all the exported keys for the specified
table
!go           Select the current connection
!help         Print a summary of command usage
!history      Display the command history
!importedkeys List all the imported keys for the specified
table
!indexes      List all the indexes for the specified table
!isolation    Set the transaction isolation for this
connection
!list         List the current connections
!manual       Display the SQLLine manual
!metadata     Obtain metadata information
!nativesql    Show the native SQL for the specified
statement
!outputformat Set the output format for displaying results
(table,vertical,csv,tsv,xmlattrs,xmlelements)
!primarykeys  List all the primary keys for the specified
```

```

table
!procedures          List all the procedures
!properties          Connect to the database specified in the
properties file(s)
!quit               Exits the program
!reconnect          Reconnect to the database
!record            Record all output to the specified file
!rehash            Fetch table and column names for command
completion
!rollback          Roll back the current transaction (if
autocommit is off)
!run              Run a script from the specified file
!save             Save the current variabes and aliases
!scan            Scan for installed JDBC drivers
!script          Start saving a script to a file
!set             Set a sqlline variable
!sql             Execute a SQL command
!tables          List all the tables in the database
!typeinfo       Display the type map for the current
connection
!verbose         Set verbose mode on

```

创建表

```

0: jdbc:phoenix:localhost> CREATE TABLE IF NOT EXISTS
us_population (
. . . . . . . . . . . . . . . . .>      state CHAR(2) NOT NULL,
. . . . . . . . . . . . . . . . .>      city VARCHAR NOT NULL,
. . . . . . . . . . . . . . . . .>      population BIGINT
. . . . . . . . . . . . . . . . .>      CONSTRAINT my_pk PRIMARY KEY
(state, city));
No rows affected (2.287 seconds)

0: jdbc:phoenix:localhost> !tables
+-----+-----+-----+-----+
| TABLE_CAT | TABLE_SCHEM | TABLE_NAME | TABLE_TYPE |
REMARKS | TYPE_NAME | SELF_REFERENCING_COL_NAME |

```

REF_GENERATION	INDEX_STATE	IMMUTABLE_ROWS	SALT_BUCKETS
false	SYSTEM	CATALOG	SYSTEM TABLE
false	SYSTEM	FUNCTION	SYSTEM TABLE
false	SYSTEM	SEQUENCE	SYSTEM TABLE
false	SYSTEM	STATS	SYSTEM TABLE
false		US_POPULATION	TABLE

5.3. Squirrel SQL Client

<http://www.squirrelsql.org/>

6. FAQ

6.1. HBaseConfTool : Unsupported major.minor version 51.0

错误提示

```
Exception in thread "main"  
java.lang.UnsupportedClassVersionError:  
org/apache/hadoop/hbase/util/HBaseConfTool : Unsupported  
major.minor version 51.0
```

解决方案，Hbase 不支持 OpenJDK 更换 Oracle Server JRE 后可以解决。

6.2. ignoring option PermSize=128m; support was removed in 8.0

jvm 1.8 之后不再支持PermSize和MaxPermSize选项

```
[neo@netkiller hbase]$ bin/start-hbase.sh  
starting master, logging to /srv/hbase/bin/../../logs/hbase-root-  
master-localhost.localdomain.out  
Java HotSpot(TM) 64-Bit Server VM warning: ignoring option  
PermSize=128m; support was removed in 8.0  
Java HotSpot(TM) 64-Bit Server VM warning: ignoring option  
MaxPermSize=128m; support was removed in 8.0
```

解决方案: "-server -Xms2048m -Xmx4096m"

第 17 章 Apache Hive

Hive是基于Hadoop构建的一套数据仓库分析系统，它提供了丰富的SQL查询方式来分析存储在Hadoop 分布式文件系统中的数据。其在Hadoop的架构体系中承担了一个SQL解析的过程，它提供了对外的入口来获取用户的指令然后对指令进行分析，解析出一个MapReduce程序组成可执行计划，并按照该计划生成对应的MapReduce任务提交给Hadoop集群处理，获取最终的结果。

1. 安装 Apache Hive

安装 Apache Hive 需要 Hadoop和MySQL，这里假设你已经懂得如何安装Hadoop和MySQL，所以一下将采用[Netkiller OSCM](#)一件安装脚本来初始化Hadoop和MySQL,如果需要详细的安装步骤请参考笔者的相关文章。

1.1. MySQL

默认情况下,Hive 使用内嵌的 Derby 数据库保存元数据,通常生产环境会使用 MySQL 来存放 Hive 元数据。

使用下面脚本一键安装MySQL 5.7 安装后会显示mysql的初始密码，是所有初始密码登陆后修改为你的需要密码

```
curl -s
https://raw.githubusercontent.com/oscm/shell/master/database/mysql/5.7/mysql.server.sh | bash

2016-02-16T08:22:58.253030Z 1 [Note] A temporary password is
generated for root@localhost: sd%%my.Ak7Ma
```

安装 MySQL JDBC 连接库。

```
curl -s
https://raw.githubusercontent.com/oscm/shell/master/database/mysql/5.7/mysql-connector-java.sh | bash
```

创建一个 hive 数据库用来存储 Hive 元数据，且数据库访问的用户名和密码都为 hive。

```
mysql> CREATE DATABASE hive;
Query OK, 1 row affected (0.03 sec)
```

创建用户hive并授权访问hive数据库

```
mysql> CREATE USER 'hive'@'localhost' IDENTIFIED BY 'hive';
Query OK, 0 rows affected (0.04 sec)

mysql> GRANT ALL ON hive.* TO 'hive'@'localhost' IDENTIFIED BY
'hive';
Query OK, 0 rows affected (0.01 sec)

mysql> GRANT ALL ON hive.* TO 'hive'@'%' IDENTIFIED BY 'hive';
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> quit;
Bye
```

1.2. Hadoop

安装 Hadoop 采用单机模式


```
curl -s
https://raw.githubusercontent.com/oscm/shell/master/distributed/
hadoop/hadoop-2.8.0.sh | bash
curl -s
https://raw.githubusercontent.com/oscm/shell/master/distributed/
hadoop/single.sh | bash
curl -s
https://raw.githubusercontent.com/oscm/shell/master/distributed/
hadoop/startup.sh | bash
```

1.3. Hive

可以从 Apache 镜像站点中下载最新稳定版的 `apache-hive-2.1.1-bin.tar.gz`

```
cd /usr/local/src
wget http://mirrors.hust.edu.cn/apache/hive/stable-2/apache-
hive-2.1.1-bin.tar.gz

tar xzf apache-hive-2.1.1-bin.tar.gz
mv apache-hive-2.1.1-bin /srv/apache-hive-2.1.1
ln -s /srv/apache-hive-2.1.1/ /srv/apache-hive
chown hadoop:hadoop -R /srv/apache-hive-2.1.1
```

```
cat > /srv/apache-hive/conf/hive-env.sh <<'EOF'
export JAVA_HOME=/srv/java
export HADOOP_HOME=/srv/apache-hadoop
export HBASE_HOME=/srv/apache-hbase
export HIVE_HOME=/srv/apache-hive
export
PATH=$PATH:$JAVA_HOME/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$HI
VE_HOME/bin
EOF
```

```
cat >> ~/.bash_profile <<'EOF'
export JAVA_HOME=/srv/java
export HADOOP_HOME=/srv/apache-hadoop
export HBASE_HOME=/srv/apache-hbase
export HIVE_HOME=/srv/apache-hive
export
PATH=$PATH:$JAVA_HOME/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$HIVE_HOME/bin
EOF

source ~/.bash_profile
```

安装JDBC驱动

```
[root@localhost apache-hive]# ln -s /usr/share/java/mysql-connector-java.jar /srv/apache-hive/lib/
[root@localhost apache-hive]# ll /srv/apache-hive/lib/mysql-connector-java.jar
lrwxrwxrwx 1 root root 40 Jun 29 01:59 /srv/apache-hive/lib/mysql-connector-java.jar -> /usr/share/java/mysql-connector-java.jar
```

修改 hive-site.xml 配置文件，配置工作目录

```
<property>
  <name>system:java.io.tmpdir</name>
  <value>/tmp/hive</value>
</property>
<property>
  <name>system:user.name</name>
  <value>hadoop</value>
</property>

<property>
  <name>hive.querylog.location</name>
  <value>/tmp/live/hadoop</value>
  <description>Location of Hive run time structured log
```

```

file</description>
</property>
<property>
  <name>hive.exec.local.scratchdir</name>
  <value>/tmp/hive</value>
  <description>Local scratch space for Hive jobs</description>
</property>
<property>
  <name>hive.downloaded.resources.dir</name>
  <value>/tmp/hive/${hive.session.id}_resources</value>
  <description>Temporary local directory for added resources
in the remote file system.</description>
</property>

<property>
  <name>hive.querylog.location</name>
  <value>/user/hive/log</value>
  <description>Location of Hive run time structured log
file</description>
</property>

```

把默认的 Derby 修改为 MySQL 需要在该文件中配置 MySQL 数据库连接信息。

```

<property>
  <name>javax.jdo.option.ConnectionURL</name>
  <value>jdbc:derby::;databaseName=metastore_db;create=true</value>
  <description>
    JDBC connect string for a JDBC metastore.
    To use SSL to encrypt/authenticate the connection, provide
database-specific SSL flag in the connection URL.
    For example, jdbc:postgresql://myhost/db?ssl=true for
postgres database.
  </description>
</property>
<property>
  <name>javax.jdo.option.ConnectionDriverName</name>
  <value>org.apache.derby.jdbc.EmbeddedDriver</value>
  <description>Driver class name for a JDBC
metastore</description>

```

```
</property>
<property>
  <name>javax.jdo.option.ConnectionUserName</name>
  <value>APP</value>
  <description>Username to use against metastore
database</description>
</property>
<property>
  <name>javax.jdo.option.ConnectionPassword</name>
  <value>mine</value>
  <description>password to use against metastore
database</description>
</property>
```

将上面配置项 value 改为下面的配置

```
<property>
  <name>javax.jdo.option.ConnectionURL</name>
  <value>jdbc:mysql://localhost:3306/hive?
createDatabaseIfNotExist=true&characterEncoding=UTF-
8&useSSL=false</value>
</property>
<property>
  <name>javax.jdo.option.ConnectionDriverName</name>
  <value>com.mysql.jdbc.Driver</value>
</property>
<property>
  <name>javax.jdo.option.ConnectionUserName</name>
  <value>hive</value>
</property>
<property>
  <name>javax.jdo.option.ConnectionPassword</name>
  <value>hive</value>
</property>
```

1.4. 启动 Hive

启动 Hive 前你必须做两件事，一是创建HDFS目录，二是初始化MySQL 数据库。

为 Hive 创建 HDFS 工作目录并给它们赋相应的权限。

```
[root@localhost ~]$ su - hadoop
[hadoop@localhost ~]$ /srv/apache-hadoop/bin/hdfs dfs -mkdir -p
/user/hive/warehouse
[hadoop@localhost ~]$ /srv/apache-hadoop/bin/hdfs dfs -mkdir -p
/tmp/hive
[hadoop@localhost ~]$ /srv/apache-hadoop/bin/hdfs dfs -chmod g+w
/user/hive/warehouse
[hadoop@localhost ~]$ /srv/apache-hadoop/bin/hdfs dfs -chmod 777
/tmp/hive
```

初始化 MySQL 数据库

```
[hadoop@localhost ~]$ /srv/apache-hive/bin/schematool -dbType
mysql -initSchema
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/srv/apache-hive-
2.1.1/lib/log4j-slf4j-impl-
2.4.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/srv/apache-hadoop-
2.8.0/share/hadoop/common/lib/slf4j-log4j12-
1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for
an explanation.
SLF4J: Actual binding is of type
[org.apache.logging.slf4j.Log4jLoggerFactory]
Metastore connection URL:
jdbc:mysql://localhost:3306/hive?
createDatabaseIfNotExist=true&characterEncoding=UTF-
8&useSSL=false
Metastore Connection Driver :    com.mysql.jdbc.Driver
Metastore connection User:      hive
Starting metastore schema initialization to 2.1.0
Initialization script hive-schema-2.1.0.mysql.sql
```

```
Initialization script completed  
schemaTool completed
```

如果使用内嵌数据库 derby 请使用下面命令初始化

```
schematool -initSchema -dbType derby
```

1.5. 访问 Hive

启动 Hadoop

```
[hadoop@localhost ~]$ /srv/apache-hadoop/sbin/start-all.sh  
This script is Deprecated. Instead use start-dfs.sh and start-  
yarn.sh  
Starting namenodes on [localhost]  
localhost: starting namenode, logging to /srv/apache-hadoop-  
2.8.0/logs/hadoop-hadoop-namenode-localhost.localdomain.out  
localhost: starting datanode, logging to /srv/apache-hadoop-  
2.8.0/logs/hadoop-hadoop-datanode-localhost.localdomain.out  
Starting secondary namenodes [0.0.0.0]  
0.0.0.0: starting secondarynamenode, logging to /srv/apache-  
hadoop-2.8.0/logs/hadoop-hadoop-secondarynamenode-  
localhost.localdomain.out  
starting yarn daemons  
starting resourcemanager, logging to /srv/apache-hadoop-  
2.8.0/logs/yarn-hadoop-resourcemanager-localhost.localdomain.out  
localhost: starting nodemanager, logging to /srv/apache-hadoop-  
2.8.0/logs/yarn-hadoop-nodemanager-localhost.localdomain.out
```

进入 Hive 然后输入 show databases; 测试安装是否正常。

```
[hadoop@localhost conf]$ /srv/apache-hive/bin/hive  
SLF4J: Class path contains multiple SLF4J bindings.
```

```
SLF4J: Found binding in [jar:file:/srv/apache-hive-2.1.1/lib/log4j-slf4j-impl-2.4.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/srv/apache-hadoop-2.8.0/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

Logging initialized using configuration in file:/srv/apache-hive-2.1.1/conf/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
hive> show databases;
OK
default
Time taken: 0.264 seconds, Fetched: 1 row(s)
hive>
```

至此 Apache Hive 已经安装配置完成！

尝试执行下面的SQL命令测试Hive是否正常

```
create database test;
use test;
create table test_table (id int,name string) row format
delimited fields terminated by ',' stored as textfile
insert into test_table values (1, 'Neo');
select * from test_table;
```

1.6. 配置 hiveserver2

hiveserver2 提供远程访问 Hive 服务，用户可以通过IP地址和端口号连接到Hive，类似mysql client

```
[hadoop@localhost ~]$ /srv/apache-hive/bin/hiveserver2
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/srv/apache-hive-2.1.1/lib/log4j-slf4j-impl-2.4.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/srv/apache-hadoop-2.8.0/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple\_bindings for an explanation.
SLF4J: Actual binding is of type
[org.apache.logging.slf4j.Log4jLoggerFactory]
```

检查端口

```
[hadoop@localhost bin]$ ss -lnt | grep 10000
LISTEN      0          50          *:10000     *:*
```

测试 beeline 是否可以正常进入

```
[hadoop@localhost ~]$ /srv/apache-hive/bin/beeline -u
jdbc:hive2://
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/srv/apache-hive-2.1.1/lib/log4j-slf4j-impl-2.4.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/srv/apache-hadoop-2.8.0/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple\_bindings for an explanation.
SLF4J: Actual binding is of type
[org.apache.logging.slf4j.Log4jLoggerFactory]
Connecting to jdbc:hive2://
17/06/29 22:01:16 [main]: WARN session.SessionState:
METASTORE_FILTER_HOOK will be ignored, since
hive.security.authorization.manager is set to instance of
```



```
HiveAuthorizerFactory.
Connected to: Apache Hive (version 2.1.1)
Driver: Hive JDBC (version 2.1.1)
17/06/29 22:01:16 [main]: WARN jdbc.HiveConnection: Request to
set autoCommit to false; Hive does not support autoCommit=false.
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 2.1.1 by Apache Hive
0: jdbc:hive2://> show databases;
OK
+-----+---+
| database_name |
+-----+---+
| default      |
+-----+---+
1 row selected (1.318 seconds)
```

如果是生产环境启动请使用下面的方法

```
[hadoop@localhost ~]$ /srv/apache-hive/bin/hive --service
hiveserver2 &
[1] 20375
[hadoop@localhost ~]$ SLF4J: Class path contains multiple SLF4J
bindings.
SLF4J: Found binding in [jar:file:/srv/apache-hive-
2.1.1/lib/log4j-slf4j-impl-
2.4.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/srv/apache-hadoop-
2.8.0/share/hadoop/common/lib/slf4j-log4j12-
1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple\_bindings for
an explanation.
SLF4J: Actual binding is of type
[org.apache.logging.slf4j.Log4jLoggerFactory]
```

2. beeline

beeline 是 hive 提供的一个新的命令行工具，基于SQLLine CLI的JDBC客户端，beeline 与HiveServer2配合使用，支持嵌入模式和远程模式两种，可以像hive client一样访问本机的hive服务，也可以通过指定ip和端口访问远程hive服务。

hive 官方是推荐使用beeline，因为它还提供了更为友好的交互方式（类似mysql client）

连接到本机的Hive

```
[hadoop@localhost ~]$ /srv/apache-hive/bin/beeline -u
jdbc:hive2://
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/srv/apache-hive-
2.1.1/lib/log4j-slf4j-impl-
2.4.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/srv/apache-hadoop-
2.8.0/share/hadoop/common/lib/slf4j-log4j12-
1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings
for an explanation.
SLF4J: Actual binding is of type
[org.apache.logging.slf4j.Log4jLoggerFactory]
Connecting to jdbc:hive2://
17/06/29 22:01:16 [main]: WARN session.SessionState:
METASTORE_FILTER_HOOK will be ignored, since
hive.security.authorization.manager is set to instance of
HiveAuthorizerFactory.
Connected to: Apache Hive (version 2.1.1)
Driver: Hive JDBC (version 2.1.1)
17/06/29 22:01:16 [main]: WARN jdbc.HiveConnection: Request to
set autoCommit to false; Hive does not support
autoCommit=false.
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 2.1.1 by Apache Hive
0: jdbc:hive2://> show databases;
OK
```

```

+-----+---+
| database_name |
+-----+---+
| default      |
+-----+---+
1 row selected (1.318 seconds)
0: jdbc:hive2://> use default;
OK
No rows affected (0.03 seconds)
0: jdbc:hive2://> show tables;
OK
+-----+---+
| tab_name     |
+-----+---+
| invites     |
| member      |
| passwd      |
| v_test      |
| vipuser     |
+-----+---+
5 rows selected (0.068 seconds)
0: jdbc:hive2://>

```

连接远程主机

```

[hadoop@localhost ~]$ /srv/apache-hive/bin/beeline -u
jdbc:hive2://hadoop@localhost:10000
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/srv/apache-hive-
2.1.1/lib/log4j-slf4j-impl-
2.4.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/srv/apache-hadoop-
2.8.0/share/hadoop/common/lib/slf4j-log4j12-
1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings
for an explanation.
SLF4J: Actual binding is of type
[org.apache.logging.slf4j.Log4jLoggerFactory]
Connecting to jdbc:hive2://hadoop@localhost:10000
Connected to: Apache Hive (version 2.1.1)
Driver: Hive JDBC (version 2.1.1)

```

```
17/06/29 23:05:35 [main]: WARN jdbc.HiveConnection: Request to
set autoCommit to false; Hive does not support
autoCommit=false.
```

```
Transaction isolation: TRANSACTION_REPEATABLE_READ
```

```
Beeline version 2.1.1 by Apache Hive
```

```
0: jdbc:hive2://hadoop@localhost:10000> show databases;
```

```
+-----+---+
| database_name |
+-----+---+
| default      |
+-----+---+
```

```
1 row selected (1.332 seconds)
```

```
0: jdbc:hive2://hadoop@localhost:10000> use default;
```

```
No rows affected (0.038 seconds)
```

```
0: jdbc:hive2://hadoop@localhost:10000> show tables;
```

```
+-----+---+
| tab_name      |
+-----+---+
| invites      |
| member       |
| passwd       |
| t_hive       |
| v_test       |
| vipuser      |
+-----+---+
```

```
6 rows selected (0.049 seconds)
```

```
0: jdbc:hive2://hadoop@localhost:10000> select * from member;
```

```
+-----+-----+-----+-----+---+
| member.name | member.age | member.sex | member.phone |
+-----+-----+-----+-----+---+
| Neo         | 30         | 1         | 13113668890 |
+-----+-----+-----+-----+---+
```

```
No rows selected (1.137 seconds)
```

```
1: jdbc:hive2://hadoop@localhost:10000>
```

3. 管理 Hive

如果你有任何一种关系型数据库的使用经验，那么你将在这里看到非常熟悉的操作。

3.1. 表管理

创建表

```
CREATE TABLE member (name string, age int, sex int);
```

```
hive> CREATE TABLE member (name string, age int, sex int);  
OK  
Time taken: 0.687 seconds  
hive>
```

基于现有的数据建表

```
hive> create table newtable as select * from oldtable;
```

显示表

```
hive> SHOW TABLES;  
OK  
test  
Time taken: 0.041 seconds, Fetched: 1 row(s)  
hive>
```

通配符匹配表名称

```
show tables '*t*';
```

删除表

```
hive> DROP TABLE test;  
OK  
Time taken: 1.337 seconds  
hive>
```

查看表结构

```
hive> CREATE TABLE member (name string, age int, sex int);  
OK  
Time taken: 0.273 seconds  
  
hive> desc member;  
OK  
name                string  
age                  int  
sex                  int  
Time taken: 0.035 seconds, Fetched: 3 row(s)  
hive>
```

为表增加字段

增加一个字段 phone 字符串类型

```
hive> ALTER TABLE member ADD COLUMNS (phone String);  
OK  
Time taken: 0.188 seconds
```

```
hive> desc member;
OK
name                string
age                 int
sex                 int
phone               string
Time taken: 0.033 seconds, Fetched: 4 row(s)
```

修改表名称

将 test 表重命名为 vipuser

```
hive> CREATE TABLE test (name string, age int, sex int);
OK
Time taken: 0.311 seconds
hive> ALTER TABLE test RENAME TO vipuser;
OK
Time taken: 0.115 seconds
hive> desc vipuser;
OK
name                string
age                 int
sex                 int
Time taken: 0.032 seconds, Fetched: 3 row(s)
hive>
```

使用已有表结构创建新表

仅仅创建表结构，不会复制数据过来。

```
hive> CREATE TABLE news_2017 LIKE news;
OK
Time taken: 0.311 seconds
```

3.2. 分区表

创建分区表

```
hive> CREATE TABLE invites (foo INT, bar STRING) PARTITIONED BY (ds STRING);
```

显示分区情况

```
hive> SHOW PARTITIONS passwd;
OK
computer=hadoop
computer=hbase
computer=hive
Time taken: 0.056 seconds, Fetched: 3 row(s)
```

增加分区

```
hive> alter table member add partition (province='shenzhen');
```

向分区表导入数据

```
hive> CREATE TABLE passwd (a string, b string, c string, d string, e string, f string) PARTITIONED BY (computer string) ROW FORMAT DELIMITED FIELDS TERMINATED BY ':';
OK
Time taken: 0.323 seconds
hive> load data local inpath '/etc/passwd' overwrite into table passwd partition(computer="hive");
Loading data to table default.passwd partition (computer=hive)
OK
Time taken: 0.499 seconds
```



```

hive> select * from passwd;
OK
root      x          0          0          root      /root      hive
bin       x          1          1          bin       /bin       hive
daemon   x          2          2          daemon   /sbin      hive
adm       x          3          4          adm       /var/adm   hive
lp        x          4          7          lp        /var/spool/lpd  hive
sync     x          5          0          sync     /sbin      hive
shutdown x          6          0          0        shutdown   /sbin      hive
halt     x          7          0          halt     /sbin      hive
mail     x          8          12         mail     /var/spool/mail  hive
operator x          11         0          operator  /root      hive
games    x          12         100        games    /usr/games  hive
ftp      x          14         50         FTP User  /var/ftp    hive
nobody   x          99         99         Nobody   /           hive
dbus     x          81         81         System message bus  /           hive
polkitd  x          999        998        User for polkitd  /           hive
avahi    x          70         70         Avahi mDNS/DNS-SD Stack
/var/run/avahi-daemon  hive
avahi-autoipd  x          170        170        170      Avahi IPv4LL Stack
/var/lib/avahi-autoipd  hive
postfix  x          89         89         /var/spool/postfix  hive
sshd     x          74         74         Privilege-separated SSH
/var/empty/sshd  hive
ntp      x          38         38         /etc/ntp  hive
rpc      x          32         32         Rpcbind Daemon  /var/lib/rpcbind
hive
qemu     x          107        107        qemu user  /           hive
unbound  x          998        996        Unbound DNS resolver  /etc/unbound
hive
rpcuser  x          29         29         RPC Service User  /var/lib/nfs
hive
nfsnobody  x          65534     65534     65534     Anonymous NFS User
/var/lib/nfs  hive
saslauth x          997        76         "Saslauthd user"
/run/saslauthd  hive
radvd    x          75         75         radvd user  /           hive
nagios   x          1000      1000      /home/nagios  hive
apache   x          48         48         Apache     /usr/share/httpd  hive
exim     x          93         93         /var/spool/exim  hive
tss      x          59         59         Account used by the trousers package
to sandbox the tcspd daemon  /dev/null  hive
git      x          996        994        /var/opt/gitlab  hive
gitlab-www  x          995        993
/var/opt/gitlab/nginx  hive
gitlab-redis  x          994        992
/var/opt/gitlab/redis  hive
gitlab-psql  x          993        991

```

```

/var/opt/gitlab/postgresql      hive
nginx      x          992      990      nginx user      /var/cache/nginx
hive
www        x          80       80       Web Application /www      hive
mysql     x          27       27       MySQL Server   /var/lib/mysql  hive
redis     x          991      989      Redis Database Server
/var/lib/redis      hive
epmd      x          990      988      Erlang Port Mapper Daemon      /tmp
hive
rabbitmq          x          989      987      RabbitMQ messaging server
/var/lib/rabbitmq      hive
solr      x          1001     1001     Apache Solr    /srv/solr      hive
mongodb  x          184      986      MongoDB Database Server
/var/lib/mongodb      hive
test     x          1002     1002     /home/test      hive
sysaccount x          988      985     /home/sysaccount
hive
systemd-bus-proxy      x          987      983     systemd Bus Proxy
/      hive
systemd-network x          986      982     systemd Network Management
/      hive
elasticsearch x          985      980     elasticsearch user
/home/elasticsearch      hive
zabbix x          984      979     Zabbix Monitoring System
/var/lib/zabbix      hive
mysqlrouter x          983      978     MySQL Router
/var/lib/mysqlrouter      hive
hadoop x          1003     1003     /home/hadoop      hive
Time taken: 0.118 seconds, Fetched: 51 row(s)

hive> SHOW PARTITIONS passwd;
OK
computer=hive
Time taken: 0.058 seconds, Fetched: 1 row(s)

```

3.3. 视图管理

创建视图

```

hive> CREATE VIEW v_test AS SELECT name,age FROM member where
age>20;
hive> select * from v_test;

```

删除视图

```
hive> drop view test;  
OK  
Time taken: 0.276 seconds
```

判断视图是否存在

```
hive> DROP VIEW IF EXISTS v_test;  
OK  
Time taken: 0.495 seconds
```

3.4. 数据管理

从文本文件导入数据

首先创建一个文本文件，如下：

```
[root@localhost ~]# cat /tmp/hive.txt  
1      2      3  
2      3      4  
3      4      5  
6      7      8
```

```
hive> CREATE TABLE test (a int, b int, c int) ROW FORMAT DELIMITED  
FIELDS TERMINATED BY '\t';  
OK  
Time taken: 0.294 seconds
```

```
hive> LOAD DATA LOCAL INPATH '/tmp/hive.txt' OVERWRITE INTO TABLE
test;
Loading data to table default.test
OK
Time taken: 0.541 seconds
hive> select * from test;
OK
1      2      3
2      3      4
3      4      5
6      7      8
Time taken: 0.952 seconds, Fetched: 5 row(s)
```

从其他表查询数据并创建新表

```
hive> CREATE TABLE mytable AS SELECT * FROM anytable;
```

从其他表查询数据然后插入指定表中

```
INSERT OVERWRITE TABLE mytable SELECT * FROM other ;
```

从现有表中查询数据然后插入到新的分区表中

```
hive> insert into table table2 partition(created_date) select * from
table1;
hive> insert into table newtable partition(type='1') select * from
oldtable;
```

3.5. HDFS与本地文件系统管理

HDFS 目录迁移

```
[hadoop@localhost ~]$ hdfs dfs -ls /user/hive/warehouse
Found 3 items
drwxrwxr-x   - hadoop supergroup          0 2017-06-29 03:36
/user/hive/warehouse/member
drwxrwxr-x   - hadoop supergroup          0 2017-06-29 03:32
/user/hive/warehouse/test
drwxrwxr-x   - hadoop supergroup          0 2017-06-29 03:41
/user/hive/warehouse/vipuser

[hadoop@localhost ~]$ hdfs dfs -cp /user/hive/warehouse/vipuser
/user/hive/warehouse/vipuser2
```

导出表数据到本地文件

```
hive> INSERT OVERWRITE LOCAL DIRECTORY '/tmp/test' SELECT * FROM
test;

hive> INSERT OVERWRITE LOCAL DIRECTORY '/tmp/test' SELECT * FROM
member;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available
in the future versions. Consider using a different execution engine
(i.e. spark, tez) or using Hive 1.X releases.
Query ID = hadoop_20170629040540_ddeda146-efed-44c4-bb20-
a6453c21cc8e
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1498716998098_0001, Tracking URL =
http://localhost:8088/proxy/application_1498716998098_0001/
Kill Command = /srv/apache-hadoop/bin/hadoop job -kill
job_1498716998098_0001
Hadoop job information for Stage-1: number of mappers: 0; number of
reducers: 0
2017-06-29 04:05:49,221 Stage-1 map = 0%, reduce = 0%
Ended Job = job_1498716998098_0001
Moving data to local directory /tmp/test
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 0 HDFS Write: 0 SUCCESS
```

```
Total MapReduce CPU Time Spent: 0 msec  
OK  
Time taken: 10.54 seconds
```

导出到HDFS

```
hive> insert overwrite directory '/usr/tmp/test' select * from test;
```

4. HiveQL - Hive查询语言

HiveQL 与 SQL 极其相似，SQL语法尽管尝试。

4.1. JOIN 连接查询

```
hive> SELECT t1.a,t1.b,t2.a,t2.b  
      > FROM table1 t1 JOIN table2 t2 on t1.a=t2.a  
      > WHERE t1.a>10;
```

4.2. 子查询

5. FAQ

5.1. adoop.security.authorize.AuthorizationException): User: hadoop is not allowed to impersonate anonymous

编辑 /srv/apache-hadoop/etc/hadoop/core-site.xml 文件，增加如下配置，这里使用 * 匹配，你可以根据实际需要修改。

```
<property>
  <name>hadoop.proxyuser.hadoop.groups</name>
  <value>*</value>
  <description>Allow the superuser oozie to impersonate
any members of the group group1 and group2</description>
</property>
<property>
  <name>hadoop.proxyuser.hadoop.hosts</name>
  <value>*</value>
  <description>The superuser can connect only from host1
and host2 to impersonate a user</description>
</property>
```

下面是我的配置文件仅供参考

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://localhost:9000/</value>
  </property>
  <property>
    <name>hadoop.proxyuser.hadoop.groups</name>
    <value>hadoop</value>
    <description>Allow the superuser oozie to impersonate
any members of the group group1 and group2</description>
```



```
</property>
<property>
  <name>hadoop.proxyuser.hadoop.hosts</name>
  <value>192.168.0.1,127.0.0.1,localhost</value>
  <description>The superuser can connect only from host1
and host2 to impersonate a user</description>
</property>
</configuration>
```

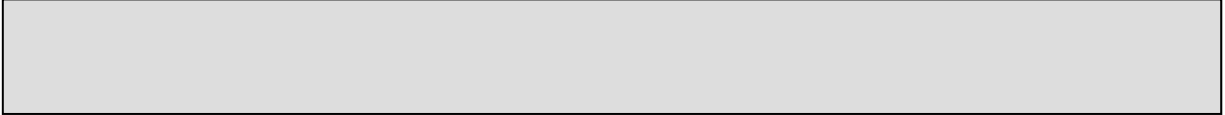
重启 Hadoop

```
[hadoop@localhost hadoop]$ /srv/apache-hadoop/sbin/stop-all.sh
This script is Deprecated. Instead use stop-dfs.sh and stop-
yarn.sh
Stopping namenodes on [localhost]
localhost: stopping namenode
localhost: stopping datanode
Stopping secondary namenodes [0.0.0.0]
0.0.0.0: stopping secondarynamenode
stopping yarn daemons
stopping resourcemanager
localhost: stopping nodemanager
localhost: nodemanager did not stop gracefully after 5 seconds:
killing with kill -9
no proxyserver to stop

[hadoop@localhost hadoop]$ /srv/apache-hadoop/sbin/start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-
yarn.sh
Starting namenodes on [localhost]
localhost: starting namenode, logging to /srv/apache-hadoop-
2.8.0/logs/hadoop-hadoop-namenode-localhost.localdomain.out
localhost: starting datanode, logging to /srv/apache-hadoop-
2.8.0/logs/hadoop-hadoop-datanode-localhost.localdomain.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /srv/apache-
hadoop-2.8.0/logs/hadoop-hadoop-secondarynamenode-
localhost.localdomain.out
starting yarn daemons
starting resourcemanager, logging to /srv/apache-hadoop-
2.8.0/logs/yarn-hadoop-resourcemanager-localhost.localdomain.out
```

```
localhost: starting nodemanager, logging to /srv/apache-hadoop-2.8.0/logs/yarn-hadoop-nodemanager-localhost.localdomain.out
```

刷新配置



第 18 章 Apache Sqoop

1. 安装 Sqoop

OSCM 一键安装

```
curl -s  
https://raw.githubusercontent.com/oscm/shell/master/database/apache-sqoop/sqoop-1.99.7-bin-hadoop200.sh | bash
```

启动 Sqoop

```
/srv/apache-sqoop/bin/sqoop.sh server start
```

检查 Sqoop 线程

```
[hadoop@netkiller ~]$ jps  
2512 SecondaryNameNode  
23729 SqoopJettyServer  
2290 DataNode  
871 ResourceManager  
23885 Jps
```

2. sqoop2-tool

2.1. verify

```
[hadoop@izj6ciilv2rcpgauqg2uuwZ ~]$ sqoop2-tool verify
Setting conf dir: /srv/apache-sqoop/bin/./conf
Sqoop home directory: /srv/apache-sqoop
Sqoop tool executor:
    Version: 1.99.7
    Revision: 435d5e61b922a32d7bce567fe5fb1a9c0d9b1bbb
    Compiled on Tue Jul 19 16:08:27 PDT 2016 by abefine
Running tool: class org.apache.sqoop.tools.tool.VerifyTool
0 [main] INFO org.apache.sqoop.core.SqoopServer -
Initializing Sqoop server.
6 [main] INFO
org.apache.sqoop.core.PropertiesConfigurationProvider -
Starting config file poller thread
Verification was successful.
Tool class org.apache.sqoop.tools.tool.VerifyTool has finished
correctly.
```

2.2. upgrade

```
[hadoop@izj6ciilv2rcpgauqg2uuwZ apache-hadoop]$ sqoop2-tool
upgrade
Setting conf dir: /srv/apache-sqoop/bin/./conf
Sqoop home directory: /srv/apache-sqoop
Sqoop tool executor:
    Version: 1.99.7
    Revision: 435d5e61b922a32d7bce567fe5fb1a9c0d9b1bbb
    Compiled on Tue Jul 19 16:08:27 PDT 2016 by abefine
Running tool: class org.apache.sqoop.tools.tool.UpgradeTool
0 [main] INFO
org.apache.sqoop.core.PropertiesConfigurationProvider -
Starting config file poller thread
```

```
Tool class org.apache.sqoop.tools.tool.UpgradeTool has finished
correctly.
```

3. sqoop2-shell

进入 sqoop2-shell

```
[hadoop@netkiller ~]$ sqoop2-shell
Setting conf dir: /srv/apache-sqoop/bin/./conf
Sqoop home directory: /srv/apache-sqoop
Sqoop Shell: Type 'help' or '\h' for help.

sqoop:000>
```

Sqoop client script:

```
sqoop2-shell /path/to/your/script.sqoop
```

3.1. show version

```
sqoop:000> show version
client version:
  Sqoop 1.99.7 source revision 435d5e61b922a32d7bce567fe5fb1a9c0d9b1bbb
  Compiled by abefine on Tue Jul 19 16:08:27 PDT 2016
```

```
sqoop:000> show version --all
client version:
  Sqoop 1.99.7 source revision 435d5e61b922a32d7bce567fe5fb1a9c0d9b1bbb
  Compiled by abefine on Tue Jul 19 16:08:27 PDT 2016
0 [main] WARN org.apache.hadoop.util.NativeCodeLoader - Unable to
load native-hadoop library for your platform... using builtin-java
classes where applicable
server version:
  Sqoop 1.99.7 source revision 435d5e61b922a32d7bce567fe5fb1a9c0d9b1bbb
  Compiled by abefine on Tue Jul 19 16:08:27 PDT 2016
API versions:
```

```
[v1]
```

3.2. set

server

```
sqoop:000> set server --host master --port 12000 --webapp sqoop
Server is set successfully
```

要设置可查看具体出错信息

```
sqoop:000> set option --name verbose --value true
Verbose option was changed to true
```

3.3. show connector

```
sqoop:000> show connector
0 [main] WARN org.apache.hadoop.util.NativeCodeLoader - Unable to
load native-hadoop library for your platform... using builtin-java
classes where applicable
+-----+-----+-----+
|          Name          | Version |          Class          |
| Supported Directions |         |                         |
+-----+-----+-----+
| generic-jdbc-connector | 1.99.7 | org.apache.sqoop.connector.jdbc.GenericJdbcConnector | FROM/TO
|
| kite-connector         | 1.99.7 | org.apache.sqoop.connector.kite.KiteConnector         | FROM/TO
|
| oracle-jdbc-connector  | 1.99.7 | org.apache.sqoop.connector.jdbc.oracle.OracleJdbcConnector | FROM/TO
|
| ftp-connector          | 1.99.7 |
```

```

org.apache.sqoop.connector.ftp.FtpConnector | TO
|
|  |  | |
|  |  | 1.99.7 |
|  |  |  |
org.apache.sqoop.connector.hdfs.HdfsConnector | FROM/TO
|
|  |  | |
|  |  | 1.99.7 |
|  |  |  |
org.apache.sqoop.connector.kafka.KafkaConnector | TO
|
|  |  | |
|  |  | 1.99.7 |
|  |  |  |
org.apache.sqoop.connector.sftp.SftpConnector | TO
|

```

```

-----+-----+-----+
-----+-----+-----+
sqoop:000>
sqoop list-databases --connect jdbc:mysql://192.168.1.1:3306/ --
username root --password 123456

```

```

sqoop:000> show connector --all
7 connector(s) to show:
Connector with Name: generic-jdbc-connector
Class: 1.99.7
Version: FROM/TO
Supported Directions {4}
link config 1:
Name: linkConfig
Label: Database connection
Help: Contains configuration that is required to establish
connection with your database server.
Input 1:
Name: linkConfig.jdbcDriver
Label: Driver class
Help: Fully qualified class name of the JDBC driver that will be
used for establishing this connection. Check documentation for
instructions how to make the driver's jar files available to Sqoop 2
server.
Type: STRING
Sensitive: false
Editable By: ANY
Overrides:
Size: 128
Input 2:
Name: linkConfig.connectionString
Label: Connection String
Help: JDBC connection string associated with your database
server.
Type: STRING
Sensitive: false
Editable By: ANY

```



```
Overrides:
Size: 2000
Input 3:
Name: linkConfig.username
Label: Username
Help: Username to be used for connection to the database server.
Type: STRING
Sensitive: false
Editable By: ANY
Overrides:
Size: 40
Input 4:
Name: linkConfig.password
Label: Password
Help: Password to be used for connection to the database server.
Type: STRING
Sensitive: true
Editable By: ANY
Overrides:
Size: 40
Input 5:
Name: linkConfig.fetchSize
Label: Fetch Size
Help: Optional hint specifying requested JDBC fetch size.
Type: INTEGER
Sensitive: false
Editable By: ANY
Overrides:
Input 6:
Name: linkConfig.jdbcProperties
Label: Connection Properties
Help: Key-value pairs that should be passed down to JDBC driver
when establishing connection.
Type: MAP
Sensitive: false
Editable By: ANY
Overrides:
link config 2:
Name: dialect
Label: SQL Dialect
Help: Database dialect that should be used for generated queries.
Input 1:
Name: dialect.identifierEnclose
Label: Identifier enclose
Help: Character(s) that should be used to enclose table name,
schema or column names.
Type: STRING
Sensitive: false
Editable By: ANY
Overrides:
Size: 5
```

```
FROM Job config 1:
  Name: fromJobConfig
  Label: Database source
  Help: Specifies source and way how the data should be fetched from
source database.
  Input 1:
    Name: fromJobConfig.schemaName
    Label: Schema name
    Help: Schema name if the table is not stored in default schema.
Note: Not all database systems understands the concept of schema.
    Type: STRING
    Sensitive: false
    Editable By: ANY
    Overrides:
    Size: 50
  Input 2:
    Name: fromJobConfig.tableName
    Label: Table name
    Help: Input table name from from which data will be retrieved.
    Type: STRING
    Sensitive: false
    Editable By: ANY
    Overrides:
    Size: 50
  Input 3:
    Name: fromJobConfig.sql
    Label: SQL statement
    Help: Import data from given query's results set rather then
static table.
    Type: STRING
    Sensitive: false
    Editable By: ANY
    Overrides:
    Size: 2000
  Input 4:
    Name: fromJobConfig.columnList
    Label: Column names
    Help: Subset of columns that should be retrieved from source
table.
    Type: LIST
    Sensitive: false
    Editable By: ANY
    Overrides:
  Input 5:
    Name: fromJobConfig.partitionColumn
    Label: Partition column
    Help: Input column that should be use to split the import into
independent parallel processes. This column will be used in condition of
generated queries.
    Type: STRING
    Sensitive: false
```

```
    Editable By: ANY
    Overrides:
    Size: 50
Input 6:
    Name: fromJobConfig.allowNullValueInPartitionColumn
    Label: Partition column nullable
    Help: Set true if partition column can contain NULL value.
    Type: BOOLEAN
    Sensitive: false
    Editable By: ANY
    Overrides:
Input 7:
    Name: fromJobConfig.boundaryQuery
    Label: Boundary query
    Help: Customize query to retrieve minimal and maximal value of
partition column.
    Type: STRING
    Sensitive: false
    Editable By: ANY
    Overrides:
    Size: 50
FROM Job config 2:
    Name: incrementalRead
    Label: Incremental read
    Help: Configures optional incremental read from the database where
source data are changing over time and only new changes need to be re-
imported.
    Input 1:
        Name: incrementalRead.checkColumn
        Label: Check column
        Help: Column that is checked during incremental read for new
values.
        Type: STRING
        Sensitive: false
        Editable By: ANY
        Overrides:
        Size: 50
    Input 2:
        Name: incrementalRead.lastValue
        Label: Last value
        Help: Last imported value, job will read only newer values.
        Type: STRING
        Sensitive: false
        Editable By: ANY
        Overrides:
        Size: -1
TO Job config 1:
    Name: toJobConfig
    Label: Database target
    Help: Describes target destination and way how data should be
persisted on the RDBMS system.
```

Input 1:

Name: toJobConfig.schemaName

Label: Schema name

Help: Schema name if the table is not stored in default schema.

Note: Not all database systems understands the concept of schema.

Type: STRING

Sensitive: false

Editable By: ANY

Overrides:

Size: 50

Input 2:

Name: toJobConfig.tableName

Label: Table name

Help: Destination table name to store transfer results.

Type: STRING

Sensitive: false

Editable By: ANY

Overrides:

Size: 2000

Input 3:

Name: toJobConfig.columnList

Label: Column names

Help: Subset of columns that will will be written to. Omitted columns have to either allow NULL values or have defined default value.

Type: LIST

Sensitive: false

Editable By: ANY

Overrides:

Input 4:

Name: toJobConfig.stageTableName

Label: Staging table

Help: Name of table with same structure as final table that should be used as a staging destination. Data will be directly written to final table if no staging table is specified.

Type: STRING

Sensitive: false

Editable By: ANY

Overrides:

Size: 2000

Input 5:

Name: toJobConfig.shouldClearStageTable

Label: Clear stage table

Help: If set to true, staging table will be wiped out upon job start.

Type: BOOLEAN

Sensitive: false

Editable By: ANY

Overrides:

Connector with Name: kite-connector

Class: 1.99.7

Version: FROM/TO

Supported Directions {4}

link config 1:

Name: linkConfig

Label: Global configuration

Help: Global configuration options that will be used for both from and to sides.

Input 1:

Name: linkConfig.authority

Label: HDFS host and port

Help: Optional to override HDFS file system location.

Type: STRING

Sensitive: false

Editable By: ANY

Overrides:

Size: 255

Input 2:

Name: linkConfig.confDir

Label: Hadoop conf directory

Help: Directory with Hadoop configuration files. This directory will be added to the classpath.

Type: STRING

Sensitive: false

Editable By: ANY

Overrides:

Size: 255

FROM Job config 1:

Name: fromJobConfig

Label: Source configuration

Help: Configuration options relevant to source dataset.

Input 1:

Name: fromJobConfig.uri

Label: dataset:hdfs://namespace/table

Help: Kite Dataset URI from which data will be read.

Type: STRING

Sensitive: false

Editable By: ANY

Overrides:

Size: 255

TO Job config 1:

Name: toJobConfig

Label: Target configuration

Help: Configuration options relevant to target dataset.

Input 1:

Name: toJobConfig.uri

Label: Dataset URI

Help: Kite Dataset URI where should be data written to.

Type: STRING

Sensitive: false

Editable By: ANY

Overrides:

Size: 255

```
Input 2:
  Name: toJobConfig.fileFormat
  Label: File format
  Help: Storage format that should be used when creating new
dataset.
  Type: ENUM
  Sensitive: false
  Editable By: ANY
  Overrides:
  Possible values: CSV,AVRO,PARQUET
Connector with Name: oracle-jdbc-connector
  Class: 1.99.7
  Version: FROM/TO
  Supported Directions {4}
  link config 1:
    Name: connectionConfig
    Label: Oracle connection configuration
    Help: You must supply the information requested in order to create
an Oracle connection object.
  Input 1:
    Name: connectionConfig.connectionString
    Label: JDBC connection string
    Help: Enter the value of JDBC connection string to be used by
this connector for creating Oracle connections.
    Type: STRING
    Sensitive: false
    Editable By: ANY
    Overrides:
    Size: 128
  Input 2:
    Name: connectionConfig.username
    Label: Username
    Help: Enter the username to be used for connecting to the
database.
    Type: STRING
    Sensitive: false
    Editable By: ANY
    Overrides:
    Size: 40
  Input 3:
    Name: connectionConfig.password
    Label: Password
    Help: Enter the password to be used for connecting to the
database.
    Type: STRING
    Sensitive: true
    Editable By: ANY
    Overrides:
    Size: 40
  Input 4:
    Name: connectionConfig.jdbcProperties
```

Label: JDBC connection properties
Help: Enter any JDBC properties that should be supplied during the creation of connection.

Type: MAP
Sensitive: false
Editable By: ANY
Overrides:

Input 5:

Name: connectionConfig.timeZone
Label: Session time zone
Help: timeZone
Type: STRING
Sensitive: false
Editable By: ANY
Overrides:
Size: -1

Input 6:

Name: connectionConfig.actionName
Label: Session action name
Help: actionName
Type: STRING
Sensitive: false
Editable By: ANY
Overrides:
Size: -1

Input 7:

Name: connectionConfig.fetchSize
Label: JDBC fetch size
Help: fetchSize
Type: INTEGER
Sensitive: false
Editable By: ANY
Overrides:

Input 8:

Name: connectionConfig.initializationStatements
Label: Session initialization statements
Help: initializationStatements
Type: LIST
Sensitive: false
Editable By: ANY
Overrides:

Input 9:

Name: connectionConfig.jdbcUrlVerbatim
Label: Use JDBC connection string verbatim
Help: jdbcUrlVerbatim
Type: BOOLEAN
Sensitive: false
Editable By: ANY
Overrides:

Input 10:

Name: connectionConfig.racServiceName

```
Label: RAC service name
Help: racServiceName
Type: STRING
Sensitive: false
Editable By: ANY
Overrides:
Size: -1
FROM Job config 1:
Name: fromJobConfig
Label: From Oracle configuration
Help: You must supply the information requested in order to create
the FROM part of the job object.
Input 1:
Name: fromJobConfig.tableName
Label: Table name
Help: tableName
Type: STRING
Sensitive: false
Editable By: ANY
Overrides:
Size: 2000
Input 2:
Name: fromJobConfig.columns
Label: Columns
Help: Columns
Type: LIST
Sensitive: false
Editable By: ANY
Overrides:
Input 3:
Name: fromJobConfig.consistentRead
Label: Consistent read
Help: consistentRead
Type: BOOLEAN
Sensitive: false
Editable By: ANY
Overrides:
Input 4:
Name: fromJobConfig.consistentReadScn
Label: Consistent read SCN
Help: consistentReadScn
Type: LONG
Sensitive: false
Editable By: ANY
Overrides:
Input 5:
Name: fromJobConfig.partitionList
Label: Partitions
Help: partitionList
Type: LIST
Sensitive: false
```


Editable By: ANY

Overrides:

Input 6:

Name: fromJobConfig.dataChunkMethod

Label: Data chunk method

Help: dataChunkMethod

Type: ENUM

Sensitive: false

Editable By: ANY

Overrides:

Possible values: ROWID,PARTITION

Input 7:

Name: fromJobConfig.dataChunkAllocationMethod

Label: Data chunk allocation method

Help: dataChunkAllocationMethod

Type: ENUM

Sensitive: false

Editable By: ANY

Overrides:

Possible values: ROUNDROBIN,SEQUENTIAL,RANDOM

Input 8:

Name: fromJobConfig.whereClauseLocation

Label: Where clause location

Help: whereClauseLocation

Type: ENUM

Sensitive: false

Editable By: ANY

Overrides:

Possible values: SUBSPLIT,SPLIT

Input 9:

Name: fromJobConfig.omitLobColumns

Label: Omit LOB columns

Help: omitLobColumns

Type: BOOLEAN

Sensitive: false

Editable By: ANY

Overrides:

Input 10:

Name: fromJobConfig.queryHint

Label: Query hint

Help: queryHint

Type: STRING

Sensitive: false

Editable By: ANY

Overrides:

Size: 2000

Input 11:

Name: fromJobConfig.conditions

Label: Conditions

Help: conditions

Type: STRING

```
Sensitive: false
Editable By: ANY
Overrides:
Size: 2000
TO Job config 1:
  Name: toJobConfig
  Label: To database configuration
  Help: You must supply the information requested in order to create
the TO part of the job object.
  Input 1:
    Name: toJobConfig.tableName
    Label: Table name
    Help: Table name to write data into
    Type: STRING
    Sensitive: false
    Editable By: ANY
    Overrides:
    Size: 2000
  Input 2:
    Name: toJobConfig.columns
    Label: Columns
    Help: Columns
    Type: LIST
    Sensitive: false
    Editable By: ANY
    Overrides:
  Input 3:
    Name: toJobConfig.templateTable
    Label: Template table name
    Help: templateTable
    Type: STRING
    Sensitive: false
    Editable By: ANY
    Overrides:
    Size: 2000
  Input 4:
    Name: toJobConfig.partitioned
    Label: Partitioned
    Help: partitioned
    Type: BOOLEAN
    Sensitive: false
    Editable By: ANY
    Overrides:
  Input 5:
    Name: toJobConfig.nologging
    Label: Nologging
    Help: nologging
    Type: BOOLEAN
    Sensitive: false
    Editable By: ANY
    Overrides:
```

Input 6:

Name: toJobConfig.updateKey
Label: Update key columns
Help: updateKey
Type: LIST
Sensitive: false
Editable By: ANY
Overrides:

Input 7:

Name: toJobConfig.updateMerge
Label: Merge updates
Help: updateMerge
Type: BOOLEAN
Sensitive: false
Editable By: ANY
Overrides:

Input 8:

Name: toJobConfig.dropTableIfExists
Label: Drop table if exists
Help: dropTableIfExists
Type: BOOLEAN
Sensitive: false
Editable By: ANY
Overrides:

Input 9:

Name: toJobConfig.storageClause
Label: Template table storage clause
Help: storageClause
Type: STRING
Sensitive: false
Editable By: ANY
Overrides:
Size: 2000

Input 10:

Name: toJobConfigtemporaryStorageClause
Label: Temporary table storage clause
Help: temporaryStorageClause
Type: STRING
Sensitive: false
Editable By: ANY
Overrides:
Size: 2000

Input 11:

Name: toJobConfig.appendValuesHint
Label: Append values hint usage
Help: appendValuesHint
Type: ENUM
Sensitive: false
Editable By: ANY
Overrides:
Possible values: AUTO,ON,OFF

Input 12:
Name: toJobConfig.parallel
Label: Parallel
Help: parallel
Type: BOOLEAN
Sensitive: false
Editable By: ANY
Overrides:

Connector with Name: ftp-connector

Class: 1.99.7

Version: TO

Supported Directions {4}

link config 1:

Name: linkConfig
Label: FTP Server configuration
Help: Parameters required to connect to an FTP server.

Input 1:

Name: linkConfig.server
Label: Hostname
Help: Hostname for the FTP server.
Type: STRING
Sensitive: false
Editable By: ANY
Overrides:
Size: 256

Input 2:

Name: linkConfig.port
Label: Port
Help: Port for the FTP server. Connector will use 21 if omitted.
Type: INTEGER
Sensitive: false
Editable By: ANY
Overrides:

Input 3:

Name: linkConfig.username
Label: Username
Help: Username that will be used to authenticate connection to
the FTP Server.
Type: STRING
Sensitive: false
Editable By: ANY
Overrides:
Size: 256

Input 4:

Name: linkConfig.password
Label: Password
Help: Password that will be used to authenticate connection to
the FTP Server.
Type: STRING
Sensitive: true
Editable By: ANY

```
    Overrides:
    Size: 256
TO Job config 1:
  Name: toJobConfig
  Label: Output configuration
  Help: Parameters required to store data on the FTP server.
  Input 1:
    Name: toJobConfig.outputDirectory
    Label: Output directory
    Help: Directory on the FTP server to write data to.
    Type: STRING
    Sensitive: false
    Editable By: ANY
    Overrides:
    Size: 260
Connector with Name: hdfs-connector
  Class: 1.99.7
  Version: FROM/TO
  Supported Directions {4}
  link config 1:
    Name: linkConfig
    Label: HDFS cluster
    Help: Contains configuration required to connect to your HDFS
cluster.
  Input 1:
    Name: linkConfig.uri
    Label: URI
    Help: Namenode URI for your cluster.
    Type: STRING
    Sensitive: false
    Editable By: ANY
    Overrides:
    Size: 255
  Input 2:
    Name: linkConfig.confDir
    Label: Conf directory
    Help: Directory on Sqoop server machine with hdfs configuration
files (hdfs-site.xml, ...). This connector will load all files ending
with -site.xml.
    Type: STRING
    Sensitive: false
    Editable By: ANY
    Overrides:
    Size: 255
  Input 3:
    Name: linkConfig.configOverrides
    Label: Additional configs:
    Help: Additional configuration that will be set on HDFS
Configuration object, possibly overriding any keys loaded from
configuration files.
    Type: MAP
```

```
Sensitive: false
Editable By: ANY
Overrides:
FROM Job config 1:
  Name: fromJobConfig
  Label: Input configuration
  Help: Specifies information required to get data from HDFS.
  Input 1:
    Name: fromJobConfig.inputDirectory
    Label: Input directory
    Help: Input directory containing files that should be
transferred.
    Type: STRING
    Sensitive: false
    Editable By: ANY
    Overrides:
    Size: 255
  Input 2:
    Name: fromJobConfig.overrideNullValue
    Label: Override null value
    Help: If set to true, then the null value will be overridden
with the value set in Null value.
    Type: BOOLEAN
    Sensitive: false
    Editable By: ANY
    Overrides:
  Input 3:
    Name: fromJobConfig.nullValue
    Label: Null value
    Help: For file formats that doesn't have native representation
of NULL (as for example text file) use this particular string to decode
NULL value.
    Type: STRING
    Sensitive: false
    Editable By: ANY
    Overrides:
    Size: 255
FROM Job config 2:
  Name: incremental
  Label: Incremental import
  Help: Information relevant for incremental reading from HDFS.
  Input 1:
    Name: incremental.incrementalType
    Label: Incremental type
    Help: Type of incremental import.
    Type: ENUM
    Sensitive: false
    Editable By: ANY
    Overrides:
    Possible values: NONE,NEW_FILES
  Input 2:
```

Name: incremental.lastImportedDate
Label: Last imported date
Help: Datetime stamp of last read file. Next job execution will read only files that have been created after this point in time.
Type: DATETIME
Sensitive: false
Editable By: ANY
Overrides:

TO Job config 1:
Name: toJobConfig
Label: Target configuration
Help: Configuration describing where and how the resulting data should be stored.

Input 1:
Name: toJobConfig.overrideNullValue
Label: Override null value
Help: If set to true, then the null value will be overridden with the value set in Null value.
Type: BOOLEAN
Sensitive: false
Editable By: ANY
Overrides:

Input 2:
Name: toJobConfig.nullValue
Label: Null value
Help: For file formats that doesn't have native representation of NULL (as for example text file) use this particular string to encode NULL value.
Type: STRING
Sensitive: false
Editable By: ANY
Overrides:
Size: 255

Input 3:
Name: toJobConfig.outputFormat
Label: File format
Help: File format that should be used for transferred data.
Type: ENUM
Sensitive: false
Editable By: ANY
Overrides:
Possible values: TEXT_FILE,SEQUENCE_FILE,PARQUET_FILE

Input 4:
Name: toJobConfig.compression
Label: Compression codec
Help: Compression codec that should be use to compress transferred data.
Type: ENUM
Sensitive: false
Editable By: ANY
Overrides:

```
Possible values:
NONE,DEFAULT,DEFLATE,GZIP,BZIP2,LZO,LZ4,SNAPPY,CUSTOM
Input 5:
  Name: toJobConfig.customCompression
  Label: Custom codec
  Help: Fully qualified class name with Hadoop codec
implementation that should be used if none of the build-in options are
suitable.
  Type: STRING
  Sensitive: false
  Editable By: ANY
  Overrides:
  Size: 255
Input 6:
  Name: toJobConfig.outputDirectory
  Label: Output directory
  Help: HDFS directory where transferred data will be written to.
  Type: STRING
  Sensitive: false
  Editable By: ANY
  Overrides:
  Size: 255
Input 7:
  Name: toJobConfig.appendMode
  Label: Append mode
  Help: If set to false, job will fail if output directory already
exists. If set to true then imported data will be stored to already
existing and possibly non empty directory.
  Type: BOOLEAN
  Sensitive: false
  Editable By: ANY
  Overrides:
Connector with Name: kafka-connector
  Class: 1.99.7
  Version: TO
  Supported Directions {4}
    link config 1:
      Name: linkConfig
      Label: Kafka cluster
      Help: Configuration options describing Kafka cluster.
      Input 1:
        Name: linkConfig.brokerList
        Label: Kafka brokers
        Help: Comma-separated list of Kafka brokers in the form of
host:port. It doesn't need to contain all brokers, but at least two are
recommended for high availability
        Type: STRING
        Sensitive: false
        Editable By: ANY
        Overrides:
        Size: 1024
```



```
Input 2:
  Name: linkConfig.zookeeperConnect
  Label: Zookeeper quorum
  Help: Address of Zookeeper used by the Kafka cluster. Usually
host:port. Multiple zookeeper nodes are supported. If Kafka is stored in
its own znode use host:portkafka
  Type: STRING
  Sensitive: false
  Editable By: ANY
  Overrides:
  Size: 255
TO Job config 1:
  Name: toJobConfig
  Label: Output configuration
  Help: Configuration necessary when writing data to Kafka.
Input 1:
  Name: toJobConfig.topic
  Label: Topic
  Help: Name of Kafka topic where data will be written into.
  Type: STRING
  Sensitive: false
  Editable By: ANY
  Overrides:
  Size: 255
Connector with Name: sftp-connector
Class: 1.99.7
Version: TO
Supported Directions {4}
  link config 1:
    Name: linkConfig
    Label: FTP Server configuration
    Help: Parameters required to connect to an SFTP server.
    Input 1:
      Name: linkConfig.server
      Label: Hostname
      Help: Hostname of the SFTP server.
      Type: STRING
      Sensitive: false
      Editable By: ANY
      Overrides:
      Size: 255
    Input 2:
      Name: linkConfig.port
      Label: Port
      Help: Port for the SFTP server. Connector will use 22 if
omitted.
      Type: INTEGER
      Sensitive: false
      Editable By: ANY
      Overrides:
    Input 3:
```

```

    Name: linkConfig.username
    Label: Username
    Help: Username that will be used to authenticate connection to
SFTP serer.
    Type: STRING
    Sensitive: false
    Editable By: ANY
    Overrides:
    Size: 256
    Input 4:
    Name: linkConfig.password
    Label: Password
    Help: Password that will be used to authenticate connection to
the FTP Server.
    Type: STRING
    Sensitive: true
    Editable By: ANY
    Overrides:
    Size: 256
    TO Job config 1:
    Name: toJobConfig
    Label: Output configuration
    Help: Parameters required to store data on the SFTP server.
    Input 1:
    Name: toJobConfig.outputDirectory
    Label: Output directory
    Help: Directory on the SFTP server to write data to.
    Type: STRING
    Sensitive: false
    Editable By: ANY
    Overrides:
    Size: 260
sqoop:000>

```

3.4. link

hdfs-connector

```

sqoop:000> create link -connector hdfs-connector
0 [main] WARN org.apache.hadoop.util.NativeCodeLoader - Unable to
load native-hadoop library for your platform... using builtin-java
classes where applicable
Creating link for connector with name hdfs-connector
Please fill following values to create new link object
Name: hdfs

```

```
HDFS cluster
```

```
URI: hdfs://127.0.0.1:9000
```

```
Conf directory:
```

```
Additional configs::
```

```
There are currently 0 values in the map:
```

```
entry#
```

```
New link was successfully created with validation status OK and name  
hdfs
```

```
sqoop:000>
```

```
sqoop:000> show link
```

```
+-----+-----+-----+  
| Name | Connector Name | Enabled |  
+-----+-----+-----+  
| hdfs | hdfs-connector | true    |  
+-----+-----+-----+
```

generic-jdbc-connector

```
sqoop:000> create link -connector generic-jdbc-connector
```

```
Creating link for connector with name generic-jdbc-connector
```

```
Please fill following values to create new link object
```

```
Name: mysql
```

```
Database connection
```

```
Driver class: com.mysql.jdbc.Driver
```

```
Connection String: jdbc:mysql://127.0.0.1:3306/test
```

```
Username: test
```

```
Password: ****
```

```
Fetch Size:
```

```
Connection Properties:
```

```
There are currently 0 values in the map:
```

```
entry#
```

```
SQL Dialect
```

```
Identifier enclose:
```

```
New link was successfully created with validation status OK and name  
mysql
```

```
sqoop:000> show link
```

Name	Connector Name	Enabled
mysql	generic-jdbc-connector	true
hdfs	hdfs-connector	true

3.5. job

create job

```
sqoop:000> create job -f "mysql" -t "hdfs"
Creating job for links with from name mysql and to name hdfs
Please fill following values to create new job object
Name: from-mysql-to-hdfs

Database source

Schema name: test
Table name: member
SQL statement:
Column names:
There are currently 0 values in the list:
element#
Partition column:
Partition column nullable:
Boundary query:

Incremental read

Check column:
Last value:

Target configuration

Override null value:
Null value:
File format:
 0 : TEXT_FILE
```

```

 1 : SEQUENCE_FILE
 2 : PARQUET_FILE
Choose: 0
Compression codec:
 0 : NONE
 1 : DEFAULT
 2 : DEFLATE
 3 : GZIP
 4 : BZIP2
 5 : LZ0
 6 : LZ4
 7 : SNAPPY
 8 : CUSTOM
Choose: 0
Custom codec:
Output directory: /sqoop/member
Append mode:

Throttling resources

Extractors:
Loaders:

Classpath configuration

Extra mapper jars:
There are currently 0 values in the list:
element#
New job was successfully created with validation status OK and name
from-mysql-to-hdfs

```

show job

```

sqoop:000> show job
+----+-----+-----+-----+-----+
| Id | Name | From Connector | To |
Connector | Enabled |
+----+-----+-----+-----+
| 1 | from-mysql-to-hdfs | mysql (generic-jdbc-connector) | hdfs (hdfs-connector) | true |
+----+-----+-----+-----+

```

start job

```
sqoop:000> start job -n from-mysql-to-hdfs

sqoop:000> start job -n from-mysql-to-hdfs
Submission details
Job Name: from-mysql-to-hdfs
Server URL: http://localhost:12000/sqoop/
Created by: hadoop
Creation date: 2017-07-22 23:18:02 CST
Lastly updated by: hadoop
External ID: job_1499236611045_0001

http://izj6ciilv2rcpgauqg2uuwz:8088/proxy/application_1499236611045_0001/
2017-07-22 23:18:02 CST: BOOTING - Progress is not available
```

启动后进入HDFS查看导入情况

```
[hadoop@netkiller ~]$ hdfs dfs -ls /sqoop

[hadoop@netkiller ~]$ hdfs dfs -ls /member
Found 10 items
-rw-r--r--  3 hadoop supergroup      0 2017-07-22 23:18
/member/310af608-5533-4bc2-bfb8-eea45470b04d.txt
-rw-r--r--  3 hadoop supergroup     48 2017-07-22 23:18
/member/36bc39a5-bc73-4065-a361-ff2d61c4922c.txt
-rw-r--r--  3 hadoop supergroup      0 2017-07-22 23:18
/member/3e855400-84a9-422d-b50c-1baa9666a719.txt
-rw-r--r--  3 hadoop supergroup    140 2017-07-22 23:18
/member/3e8dad92-e0f1-4a74-a337-642cf4e6d634.txt
-rw-r--r--  3 hadoop supergroup     55 2017-07-22 23:18
/member/4a9f47f1-0413-4149-a93a-ed8b51efbc87.txt
-rw-r--r--  3 hadoop supergroup      0 2017-07-22 23:18
/member/4dc5bfe7-1cd9-4d9b-96a8-07e82ed79a71.txt
-rw-r--r--  3 hadoop supergroup      0 2017-07-22 23:18
/member/60dbcc60-61f2-4433-af39-1dfdfc048940.txt
-rw-r--r--  3 hadoop supergroup      0 2017-07-22 23:18
/member/6d02ed89-94d9-4d4b-87ed-d5da9d2bf9fe.txt
-rw-r--r--  3 hadoop supergroup    209 2017-07-22 23:18
/member/cf7b7185-3ab6-4077-943a-26228b769c57.txt
-rw-r--r--  3 hadoop supergroup      0 2017-07-22 23:18
/member/f2e0780d-ad33-4b35-alc7-b3fbc23e303d.txt
```

status job

```
sqoop:000> status job -n from-mysql-to-hdfs
```

3.6. update

link

```
sqoop:000> update link -n mysql
Updating link with name mysql
Please update link:
Name: mysql

Database connection

Driver class: com.mysql.jdbc.Driver
Connection String: jdbc:mysql://127.0.0.1:3306/test
Username: test
Password: ****
Fetch Size:
Connection Properties:
There are currently 0 values in the map:
entry#

SQL Dialect

Identifier enclose:
link was successfully updated with status OK
```

4. FAQ

4.1. Unable to load native-hadoop library for your platform

0 [main] WARN org.apache.hadoop.util.NativeCodeLoader - Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

部分 III. MySQL

第 19 章 MySQL Server

1. MySQL Installation

<http://downloads.mysql.com/archives.php>

1.1. CentOS

CentOS 8 Stream + dnf 安装 Mysql

```
[root@localhost ~]# dnf install -y mysql-server mysql mysql-test
[root@localhost ~]# systemctl enable mysqld
Created symlink /etc/systemd/system/multi-
user.target.wants/mysqld.service →
/usr/lib/systemd/system/mysqld.service.
```

配分配置文件

```
cp /etc/my.cnf{,.original}
cp /etc/my.cnf.d/mysql-server.cnf /etc/mysql-server.cnf.original
cat >> /etc/my.cnf.d/mysql-server.cnf <<EOF

# Add by Neo
character-set-server=utf8mb4
collation-server=utf8mb4_general_ci
explicit_defaults_for_timestamp=true
lower_case_table_names=1
EOF
```

设置文件打开数据

```
cat >> /etc/security/limits.d/20-nofile.conf <<EOF
mysql soft nofile 65535
mysql hard nofile 65535
EOF

mkdir /etc/systemd/system/mysqld.service.d/

cat >> /etc/systemd/system/mysqld.service.d/override.conf <<EOF
[Service]
LimitNOFILE=65000
EOF
```

启动数据库

```
[root@localhost ~]# systemctl start mysqld
```

创建用户

```
mysql> CREATE USER 'root'@'%' IDENTIFIED BY 'test';
Query OK, 0 rows affected (0.05 sec)

mysql> GRANT ALL ON *.* TO 'root'@'%' WITH GRANT OPTION;
Query OK, 0 rows affected (0.03 sec)
```

例 19.1. MySQL 8 创建root账号

```
[root@localhost ~]# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.21 Source distribution
```

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
mysql> CREATE USER 'root'@'%' IDENTIFIED BY 'test';  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> GRANT ALL ON *.* TO 'root'@'%' WITH GRANT OPTION;  
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> FLUSH PRIVILEGES;  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql>
```

CentOS 6.2 + MySQL 5.5.25 (RPM)

准备下面的软件包

```
# ls -l  
MySQL-client-5.5.25-1.el6.x86_64.rpm  
MySQL-devel-5.5.25-1.el6.x86_64.rpm  
MySQL-server-5.5.25-1.el6.x86_64.rpm  
MySQL-shared-5.5.25-1.el6.x86_64.rpm  
MySQL-shared-compat-5.5.25-1.el6.x86_64.rpm
```

使用 yum 本地安装 rpm, yum 可以帮你解决依赖于冲突

```
# yum localinstall MySQL-*
```

```
# /etc/init.d/mysql start
Starting MySQL... SUCCESS!

# /usr/bin/mysqladmin -u root password 'tUG26WSslP30bkbwtMhn'
```

MySQL 8.0

安装

```
curl -s
https://raw.githubusercontent.com/oscm/shell/master/database/mysql/8.0/server.sh | bash
```

启动

```
systemctl enable mysql
systemctl start mysql
```

必须修改密码后才能使用

```
[root@netkiller ~]# grep "A temporary password"
/var/log/mysqld.log
2018-04-03T02:24:16.935070Z 1 [Note] A temporary password is
generated for root@localhost: kMA*d<e#Q3EC
2018-04-20T03:36:31.935143Z 5 [Note] [MY-010454] [Server] A
temporary password is generated for root@localhost: MqatK=hae5F#

[root@netkiller ~]# mysqladmin -u root -p'MqatK=hae5F#' password
mysqladmin: [Warning] Using a password on the command line
interface can be insecure.
```

```
New password:
Confirm new password:
Warning: Since password will be sent to server in plain text,
use ssl connection to ensure password safety.

[root@netkiller ~]# mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.11 MySQL Community Server - GPL

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All
rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or
its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current
input statement.

mysql>
```

创建用户

```
mysql> CREATE USER 'root'@'%' IDENTIFIED BY
'MQiEgelikst7S_6tlXzBomt_4b';
Query OK, 0 rows affected (0.05 sec)

mysql> GRANT ALL ON *.* TO 'root'@'%' ;
Query OK, 0 rows affected (0.03 sec)
```

1.2. Docker

```
docker run --name mysql -d \  
  --restart always \  
  -e MYSQL_ROOT_PASSWORD=123456 \  
  -e MYSQL_DATABASE=test \  
  -e MYSQL_USER=test \  
  -e MYSQL_PASSWORD=test \  
  -p 127.0.0.1:3306:3306 \  
  mysql:latest
```

1.3. Ubuntu/Debian

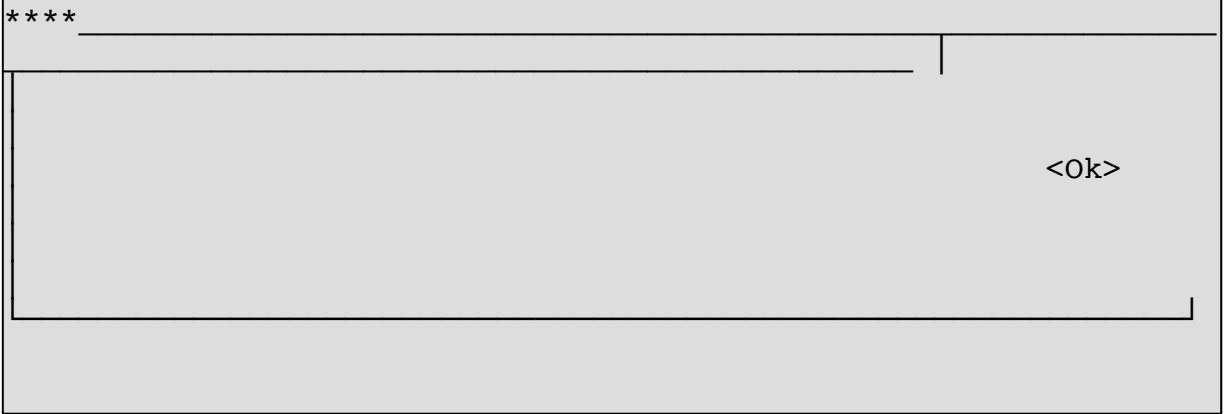
Installation by apt under debian/ubuntu

安装环境 ubuntu 17.10

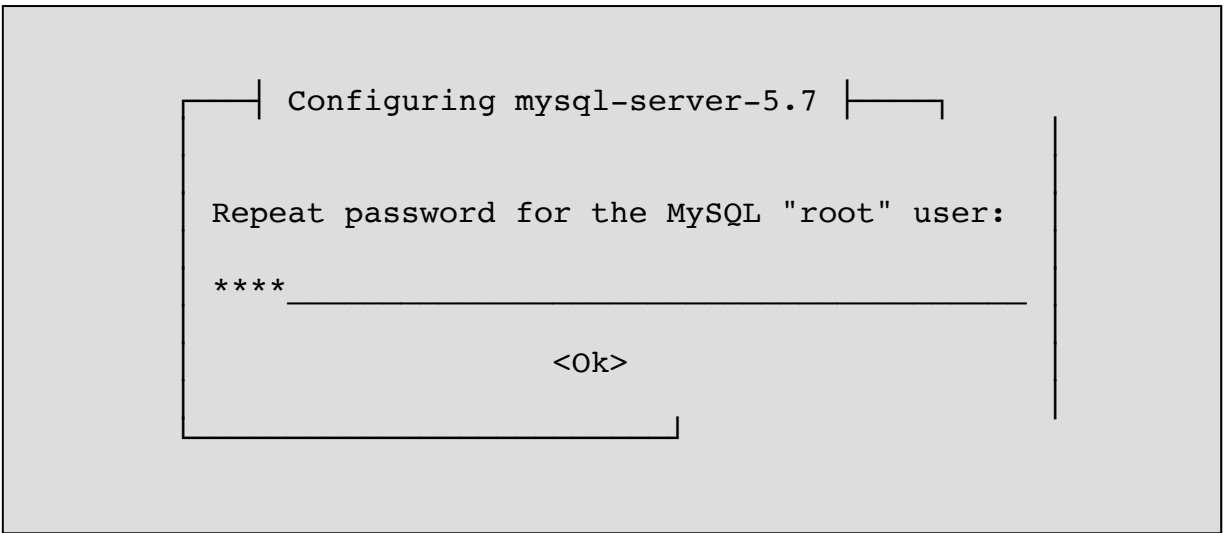
```
sudo apt install mysql-server mysql-  
client
```

New password for the MySQL "root" user

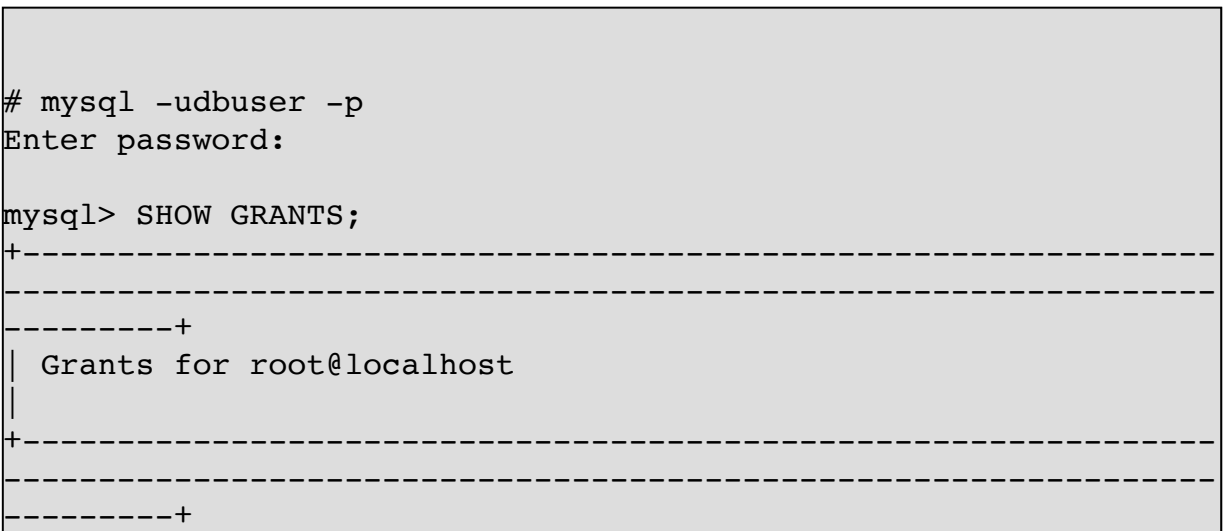
```
Configuring mysql-server-5.7  
While not mandatory, it is highly recommended that you set a  
password for the MySQL administrative "root" user.  
  
If that field is left blank, the password will not be changed.  
  
New password for the MySQL "root" user:
```



Repeat password for the MySQL "root" user



尝试登录，验证是否安装成功




```
| GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' IDENTIFIED  
BY PASSWORD '*C6325DAF39AE6CC34E960D3C65F1398FE467E1D0' WITH  
GRANT OPTION |
```

```
+-----+  
-----+
```

```
1 row in set (0.00 sec)
```

```
GRANT ALL PRIVILEGES ON example.* TO 'dbuser'@'localhost'  
IDENTIFIED BY '*****' WITH GRANT OPTION;  
FLUSH PRIVILEGES;
```

配置监听IP地址，默认数据库只能从 127.0.0.1访问

```
vim                                     neo@netkiller /etc/mysql/mysql.conf.d %  
  
                                         /etc/mysql/mysql.conf.d/mysql.d.cnf  
  
                                         bind-address = 0.0.0.0
```

mysql-5.5.21-debian6.0-i686.deb

```
sudo apt-get install libaiol  
  
sudo groupadd mysql  
sudo useradd -r -g mysql mysql  
  
sudo dpkg -i mysql-5.5.21-debian6.0-i686.deb  
  
cd /opt/mysql/  
sudo chown -R mysql .  
sudo chgrp -R mysql .  
  
cd server-5.5/  
  
sudo support-files/binary-configure
```

```
sudo chown -R mysql data

# Next command is optional
shell> cp support-files/my-medium.cnf /etc/my.cnf

shell> bin/mysqld_safe --user=mysql &

# Next command is optional
sudo cp support-files/mysql.server /etc/init.d/mysql
```

1.4. 源码安装

Installation by source code

```
./configure \
--prefix=/usr/local/$MYSQL_DIR \
--enable-asm \
--enable-local-infile \
--with-charset=utf8 \
--with-collation=utf8_general_ci \
--with-extra-charsets=none \
--with-openssl \
--with-pthread \
--with-unix-socket-path=/var/lib/mysql/mysql.sock \
--with-mysqld-user=mysql \
--with-mysqld-ldflags \
--with-client-ldflags \
--with-comment \
--with-big-tables \
--without-ndb-debug \
--without-docs \
--without-debug \
--without-bench

make && make install
```

/usr/local/\$MYSQL_DIR/bin/mysql_install_db

other option

```
--without-isam
--without-innodb
--without-ndbcluster
--without-blackhole
--without-ibmdb2i
--without-federated
--without-example
--without-comment
--localstatedir=/usr/local/mysql/data
```

1.5. 二进制版本

MySQL binary distribution

```
shell> groupadd mysql
shell> useradd -r -g mysql mysql
shell> cd /usr/local
shell> tar zxvf /path/to/mysql-VERSION-OS.tar.gz
shell> ln -s full-path-to-mysql-VERSION-OS mysql
shell> cd mysql
shell> chown -R mysql .
shell> chgrp -R mysql .
shell> scripts/mysql_install_db --user=mysql
shell> chown -R root .
shell> chown -R mysql data
# Next command is optional
shell> cp support-files/my-medium.cnf /etc/my.cnf
shell> bin/mysqld_safe --user=mysql &
# Next command is optional
shell> cp support-files/mysql.server /etc/init.d/mysql.server
```

install core database

```
[root@test mysql]# ./scripts/mysql_install_db
Installing MySQL system tables...
100428 23:16:20 [Warning] '--skip-locking' is deprecated and
will be removed in a future release. Please use '--skip-
external-locking' instead.
OK
Filling help tables...
100428 23:16:20 [Warning] '--skip-locking' is deprecated and
will be removed in a future release. Please use '--skip-
external-locking' instead.
OK

To start mysqld at boot time you have to copy
support-files/mysql.server to the right place for your system

PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !
To do so, start the server, then issue the following commands:

./bin/mysqladmin -u root password 'new-password'
./bin/mysqladmin -u root -h db.example.com password 'new-
password'

Alternatively you can run:
./bin/mysql_secure_installation

which will also give you the option of removing the test
databases and anonymous user created by default. This is
strongly recommended for production servers.

See the manual for more instructions.

You can start the MySQL daemon with:
cd . ; ./bin/mysqld_safe &

You can test the MySQL daemon with mysql-test-run.pl
cd ./mysql-test ; perl mysql-test-run.pl

Please report any problems with the ./bin/mysqlbug script!
```

set root's password

```
[root@test mysql]# cp support-
```

```
files/mysql.server      /etc/init.d/mysqld
                        [root@test mysql]# /etc/init.d/mysqld
start                  Starting MySQL. [ OK ]
                        [root@test mysql]# ./bin/mysqladmin -u
root                   password 'chen'
                        [root@test mysql]# ./bin/mysqladmin -u
root -h                db.example.com password 'chen'
```

test

```
[root@test mysql]# ./bin/mysql -uroot -pchen
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.1.45 MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the current
input statement.

mysql>
```

1.6. Installing MySQL on Linux Using the MySQL Yum Repository

MySQL 5.6

<http://dev.mysql.com/doc/mysql-repo-excerpt/5.6/en/linux-installation-yum-repo.html>

```
yum localinstall http://dev.mysql.com/get/mysql-community-
```

```
release-el6-5.noarch.rpm
```

安装MySQL Server

```
yum install mysql-server  
chkconfig mysqld on  
service mysqld start
```

修改root密码

```
mysqladmin -u root password 'new-password'
```

安全设置向导

```
/usr/bin/mysql_secure_installation
```

MySQL 5.7

```
yum localinstall -y https://dev.mysql.com/get/mysql57-community-  
release-el7-11.noarch.rpm  
yum install mysql-server -y  
systemctl enable mysqld  
systemctl start mysqld  
  
cp /etc/my.cnf{,.original}  
  
cat >> /etc/security/limits.d/20-nofile.conf <<EOF  
  
mysql soft nofile 40960  
mysql hard nofile 40960  
EOF
```

```
cat >> /etc/my.cnf.d/default.cnf <<EOF
[mysqld]
skip-name-resolve
max_connections=8192
default-storage-engine=INNODB

#wait_timeout=30
#interactive_timeout=30

character-set-server=utf8
collation_server=utf8_general_ci
init_connect='SET NAMES utf8'

explicit_defaults_for_timestamp=true

query_cache_type=1
query_cache_size=512M

[client]
character_set_client=utf8

EOF
```

MySQL 5.7 会随机分配一个密码给用户

```
grep "A temporary password" /var/log/mysqld.log
```

登陆后修改密码

```
ALTER USER root@localhost identified by
'MQiEgelikst7S_6tlXzBomt_4b';
ALTER USER root@localhost PASSWORD EXPIRE NEVER;
```

1.7. Firewall

iptables

```
iptables -A INPUT -i eth0 -p tcp -s xxx.xxx.xxx.xxx --dport 3306  
-j ACCEPT
```

1.8. Mac OS

```
brew install mysql
```

启动

```
brew services start mysql
```

1.9. MariaDB

<http://mariadb.org/>

CentOS 6 YUM 安装 MariaDB

```
cat >> /etc/yum.repos.d/MariaDB.repo <<EOF  
# MariaDB 5.5 CentOS repository list - created 2013-12-04 02:17  
UTC  
# http://mariadb.org/mariadb/repositories/  
[mariadb]  
name = MariaDB  
baseurl = http://yum.mariadb.org/5.5/centos6-amd64  
gpgkey=https://yum.mariadb.org/RPM-GPG-KEY-MariaDB  
gpgcheck=1
```



```
EOF
```

```
# yum search MariaDB | grep MariaDB
===== N/S Matched: MariaDB
=====
MariaDB-Galera-server.x86_64 : MariaDB: a very fast and robust
SQL database
MariaDB-client.x86_64 : MariaDB: a very fast and robust SQL
database server
MariaDB-common.x86_64 : MariaDB: a very fast and robust SQL
database server
MariaDB-compatible.x86_64 : MariaDB: a very fast and robust SQL
database server
MariaDB-devel.x86_64 : MariaDB: a very fast and robust SQL
database server
MariaDB-server.x86_64 : MariaDB: a very fast and robust SQL
database server
MariaDB-shared.x86_64 : MariaDB: a very fast and robust SQL
database server
MariaDB-test.x86_64 : MariaDB: a very fast and robust SQL
database server
```

安装数据库

```
# yum install -y MariaDB-server MariaDB-client
```

指定默认root密码

```
# mysqladmin -u root password 'chen'
```

数据库安全配置

```
# /usr/bin/mysql_secure_installation
/usr/bin/mysql_secure_installation: line 379: find_mysql_client:
command not found
```

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL
MariaDB

SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP
CAREFULLY!

In order to log into MariaDB to secure it, we'll need the
current
password for the root user. If you've just installed MariaDB,
and
you haven't set the root password yet, the password will be
blank,
so you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the
MariaDB
root user without the proper authorisation.

You already have a root password set, so you can safely answer
'n'.

Change the root password? [Y/n]
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!

By default, a MariaDB installation has an anonymous user,
allowing anyone
to log into MariaDB without having to have a user account
created for
them. This is intended only for testing, and to make the
installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n]
... Success!

Normally, root should only be allowed to connect from
'localhost'. This

ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n]
... Success!

By default, MariaDB comes with a database named 'test' that anyone can access. This is also intended only for testing, and should be removed before moving into a production environment.

Remove test database and access to it? [Y/n]
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far will take effect immediately.

Reload privilege tables now? [Y/n]
... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB installation should now be secure.

Thanks for using MariaDB!

进入数据库

```
# mysql -uroot -pchen
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 12
Server version: 5.5.34-MariaDB MariaDB Server

Copyright (c) 2000, 2013, Oracle, Monty Program Ab and others.
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the current
input statement.

MariaDB [(none)]> show databases;
+-----+
| Database                |
+-----+
| information_schema      |
| mysql                   |
| performance_schema     |
+-----+
3 rows in set (0.00 sec)

MariaDB [(none)]> use mysql;
Reading table information for completion of table and column
names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [mysql]>
```

使用上与MySQL并无差异

CentOS 7 安装 MariaDB

```
yum install -y mariadb
```

1.10. Percona

<http://www.percona.com/>

Percona yum Repository

```
# yum install http://www.percona.com/redirect/downloads/percona-
```

```
release/redhat/latest/percona-release-0.1-3.noarch.rpm
```

查看所有percona软件包

```
yum search percona
```

Percona XtraBackup

安装 XtraBackup

通过YUM安装 percona-xtrabackup

```
# yum install percona-xtrabackup
```

通过RPM安装 CentOS 6

http://www.percona.com/downloads/XtraBackup/LATEST/binary/redhat/6/x86_64/

```
# yum install -y  
http://www.percona.com/redir/downloads/XtraBackup/LATEST/binary/  
redhat/6/x86_64/percona-xtrabackup-2.2.6-5042.el6.x86_64.rpm
```

通过RPM安装 CentOS 7

http://www.percona.com/downloads/XtraBackup/LATEST/binary/redhat/7/x86_64/

```
# yum install -y  
http://www.percona.com/redir/downloads/XtraBackup/LATEST/binary/  
redhat/7/x86_64/percona-xtrabackup-2.2.6-5042.el7.x86_64.rpm
```

卸载

```
# yum remove percona-xtrabackup
```

查看文件列表

```
# rpm -ql percona-xtrabackup
/usr/bin/innobackupex
/usr/bin/xbcrypt
/usr/bin/xbstream
/usr/bin/xtrabackup
/usr/share/doc/percona-xtrabackup-2.2.6
/usr/share/doc/percona-xtrabackup-2.2.6/COPYING
```

innobackupex

首先创建备份用户

```
mysql> CREATE USER 'backup'@'localhost' IDENTIFIED BY 's3cret';
mysql> GRANT RELOAD, LOCK TABLES, REPLICATION CLIENT ON *.* TO
'backup'@'localhost';
mysql> FLUSH PRIVILEGES;
```

备份数据库

备份所有数据库

```
# mkdir -p /backup
# innobackupex --user=backup --password=chen /backup/full
```

备份指定数据库

```
# innobackupex --user=backup --password=chen --database=test /backup
```

--defaults-file=/etc/my.cnf 参数

```
# innobackupex --defaults-file=/etc/my.cnf --user=backup --password=chen --database=test /backup
```

备份后打包

```
# innobackupex --user=backup --password=chen --database=test --stream=tar /backup/ > test.tar
```

打包并压缩

```
# innobackupex --user=backup --password=chen --database=test --stream=tar /backup/ | gzip > test.tar.gz
```

备份到远程服务器

```
# innobackupex --user=backup --password=chen --defaults-file=/etc/my.cnf --database=test --stream=tar /backup | gzip | ssh neo@192.168.2.1 cat ">" /backup/backup-2014-11-12.tar.gz
```

增量备份

```
# innobackupex --user=backup --password=chen --database=test /backup/incremental  
# ls /backup/incremental  
2014-11-12_13-45-26  
# innobackupex --user=backup --password=chen --database=test --incremental --incremental-basedir=/backup/incremental/2014-11-
```

```
12_13-45-26/ /backup/incremental
```

恢复数据库

恢复数据首先停止MySQL服务

```
# service mysql stop
```

恢复文件

```
# innobackupex --copy-back /path/to/BACKUP-DIR  
# innobackupex --user=backup --password=chen --apply-log  
/backup/full/2014-11-12_13-45-26/
```

数据恢复完成后修改权限

```
$ chown -R mysql:mysql /var/lib/mysql
```

增量备份恢复方法

```
innobackupex --apply-log --redo-only BASE-DIR  
innobackupex --apply-log --redo-only BASE-DIR --incremental-  
dir=INCREMENTAL-DIR-1  
innobackupex --apply-log BASE-DIR --incremental-dir=INCREMENTAL-  
DIR-2  
innobackupex --apply-log BASE-DIR  
innobackupex --copy-back BASE-DIR
```

xbstream

```
$ innobackupex --stream=tar /tmp
```



```
$ innobackupex --stream=xbstream /root/backup/ >
/root/backup/backup.xbstream
$ innobackupex --stream=xbstream --compress /root/backup/ >
/root/backup/backup.xbstream

$ xbstream -x < backup.xbstream -C /root/backup/
$ innobackupex --compress --stream=xbstream /root/backup/ | ssh
user@otherhost "xbstream -x -C /root/backup/"
```

xtrabackup

```
# xtrabackup --user=backup --password=chen --backup --target-
dir=/backup/backup
```

Percona Toolkit - MySQL Management Software

YUM安装

```
# yum install -y percona-toolkit
```

RPM安装

```
# yum install -y http://www.percona.com/redirect/downloads/percona-
toolkit/LATEST/RPM/percona-toolkit-2.2.11-1.noarch.rpm
```

percona-toolkit 所含的文件

```
# rpm -ql percona-toolkit
/usr/bin/pt-align
/usr/bin/pt-archiver
/usr/bin/pt-config-diff
/usr/bin/pt-deadlock-logger
/usr/bin/pt-diskstats
```

```
/usr/bin/pt-duplicate-key-checker
/usr/bin/pt-fifo-split
/usr/bin/pt-find
/usr/bin/pt-fingerprint
/usr/bin/pt-fk-error-logger
/usr/bin/pt-heartbeat
/usr/bin/pt-index-usage
/usr/bin/pt-ioprofile
/usr/bin/pt-kill
/usr/bin/pt-mext
/usr/bin/pt-mysql-summary
/usr/bin/pt-online-schema-change
/usr/bin/pt-pmp
/usr/bin/pt-query-digest
/usr/bin/pt-show-grants
/usr/bin/pt-sift
/usr/bin/pt-slave-delay
/usr/bin/pt-slave-find
/usr/bin/pt-slave-restart
/usr/bin/pt-stalk
/usr/bin/pt-summary
/usr/bin/pt-table-checksum
/usr/bin/pt-table-sync
/usr/bin/pt-table-usage
/usr/bin/pt-upgrade
/usr/bin/pt-variable-advisor
/usr/bin/pt-visual-explain
/usr/share/doc/percona-toolkit-2.2.11
/usr/share/doc/percona-toolkit-2.2.11/COPYING
/usr/share/doc/percona-toolkit-2.2.11/Changelog
/usr/share/doc/percona-toolkit-2.2.11/INSTALL
/usr/share/doc/percona-toolkit-2.2.11/README
/usr/share/man/man1/percona-toolkit.1p.gz
/usr/share/man/man1/pt-align.1p.gz
/usr/share/man/man1/pt-archiver.1p.gz
/usr/share/man/man1/pt-config-diff.1p.gz
/usr/share/man/man1/pt-deadlock-logger.1p.gz
/usr/share/man/man1/pt-diskstats.1p.gz
/usr/share/man/man1/pt-duplicate-key-checker.1p.gz
/usr/share/man/man1/pt-fifo-split.1p.gz
/usr/share/man/man1/pt-find.1p.gz
/usr/share/man/man1/pt-fingerprint.1p.gz
/usr/share/man/man1/pt-fk-error-logger.1p.gz
/usr/share/man/man1/pt-heartbeat.1p.gz
/usr/share/man/man1/pt-index-usage.1p.gz
/usr/share/man/man1/pt-ioprofile.1p.gz
```

```
/usr/share/man/man1/pt-kill.1p.gz  
/usr/share/man/man1/pt-mext.1p.gz  
/usr/share/man/man1/pt-mysql-summary.1p.gz  
/usr/share/man/man1/pt-online-schema-change.1p.gz  
/usr/share/man/man1/pt-pmp.1p.gz  
/usr/share/man/man1/pt-query-digest.1p.gz  
/usr/share/man/man1/pt-show-grants.1p.gz  
/usr/share/man/man1/pt-sift.1p.gz  
/usr/share/man/man1/pt-slave-delay.1p.gz  
/usr/share/man/man1/pt-slave-find.1p.gz  
/usr/share/man/man1/pt-slave-restart.1p.gz  
/usr/share/man/man1/pt-stalk.1p.gz  
/usr/share/man/man1/pt-summary.1p.gz  
/usr/share/man/man1/pt-table-checksum.1p.gz  
/usr/share/man/man1/pt-table-sync.1p.gz  
/usr/share/man/man1/pt-table-usage.1p.gz  
/usr/share/man/man1/pt-upgrade.1p.gz  
/usr/share/man/man1/pt-variable-advisor.1p.gz  
/usr/share/man/man1/pt-visual-explain.1p.gz
```

2. MySQL Plugin

2.1. validate_password

插件的卸载与安装

```
uninstall plugin validate_password;  
INSTALL PLUGIN validate_password SONAME 'validate_password.so';
```

查看变量设置

```
mysql> SHOW VARIABLES LIKE 'validate_password%';  
+-----+-----+  
| Variable_name | Value |  
+-----+-----+  
| validate_password_check_user_name | OFF |  
| validate_password_dictionary_file | |  
| validate_password_length | 8 |  
| validate_password_mixed_case_count | 1 |  
| validate_password_number_count | 1 |  
| validate_password_policy | MEDIUM |  
| validate_password_special_char_count | 1 |  
+-----+-----+  
7 rows in set (0.00 sec)  
  
mysql>
```

修改策略与密码长度

```
mysql> set global validate_password_policy=0;  
mysql> set global validate_password_length=4;  
mysql> grant all privileges on test.* to 'test'@localhost  
identified by 'chen';
```

2.2. MySQL Images manager

地址: <https://github.com/netkiller/mysql-image-plugin>

```
cd /usr/local/src/  
git clone https://github.com/netkiller/mysql-image-plugin.git  
cd mysql-image-plugin/  
yum install cmake  
  
cmake .  
make && make install  
                                </screen>  
                                <screen>  
Install  
  
create function image_check returns string soname  
'libimage.so';  
create function image_remove returns string soname  
'libimage.so';  
create function image_rename returns string soname  
'libimage.so';  
create function image_crc32 returns string soname  
'libimage.so';  
create function image_move returns string soname 'libimage.so';  
Uninstall  
  
drop function image_check;  
drop function image_remove;  
drop function image_rename;  
drop function image_crc32;  
drop function image_move;  
Example  
  
select image_check('/path/filename.ext');  
select image_remove('/path/filename.ext');  
select image_rename('/path/oldfile.ext', '/path/newfile.ext');  
select image_crc32('/path/filename.ext');  
select image_move('/path/filename.ext', '/path/to/newfile.ext');
```

2.3. MySQL fifo

```
mysql-fifo-plugin
```

```
MySQL Pipes (FIFOs) Plugin
```

```
Build
```

```
cd /usr/local/src/  
git clone https://github.com/netkiller/mysql-fifo-plugin.git  
cd mysql-fifo-plugin/
```

```
cmake .  
make  
make install
```

```
or
```

```
gcc -O3 -g -I/usr/include/mysql -I/usr/include -fPIC -lm -lz  
-shared -o libfifo.so fifo.c  
sudo mv libfifo.so /usr/lib/mysql/plugin/
```

```
Plugin Install and Uninstall
```

```
Install
```

```
create function fifo_create returns string soname 'libfifo.so';  
create function fifo_remove returns string soname 'libfifo.so';  
create function fifo_read returns string soname 'libfifo.so';  
create function fifo_write returns string soname 'libfifo.so';
```

```
Uninstall
```

```
drop function fifo_create;  
drop function fifo_remove;  
drop function fifo_read;  
drop function fifo_write;
```

```
Testing
```

```
创建管道
```

```
mysql> create function fifo_create returns string soname
'libfifo.so';
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> select fifo_create('/tmp/myfifo');
+-----+
| fifo_create('/tmp/myfifo') |
+-----+
| ture                        |
+-----+
1 row in set (0.00 sec)
```

查看管道是否创建

```
$ ls /tmp/myfifo
/tmp/myfifo
```

覆盖测试，正确应该返回false

```
mysql> select fifo_create('/tmp/myfifo');
+-----+
| fifo_create('/tmp/myfifo') |
+-----+
| false                       |
+-----+
1 row in set (0.00 sec)
```

删除管道

```
mysql> select fifo_remove('/tmp/myfifo');
+-----+
| fifo_remove('/tmp/myfifo') |
+-----+
| true                        |
+-----+
1 row in set (0.00 sec)
```

```
mysql> select fifo_remove('/tmp/my');
+-----+
| fifo_remove('/tmp/my')     |
+-----+
| false                       |
+-----+
1 row in set (0.00 sec)
```

删除不存在的管道会提示 false

读管道

在一个终端窗口中运行

```
mysql> select fifo_read('/tmp/myfifo');
```

```
+-----+
| fifo_read('/tmp/myfifo') |
+-----+
| Hello world              |
+-----+
```

```
1 row in set (7.85 sec)
```

在另一个终端窗口中运行

```
mysql> select fifo_write('/tmp/myfifo','Hello world !!!');
```

```
+-----+
| fifo_write('/tmp/myfifo','Hello world !!!') |
+-----+
| true                                         |
+-----+
```

```
1 row in set (0.00 sec)
```

或者

在命令行运行

```
$ echo "Hello world" > /tmp/myfifo
```

在SQL客户端中运行

```
mysql> select fifo_read('/tmp/myfifo');
```

```
+-----+
| fifo_read('/tmp/myfifo') |
+-----+
| Hello world              |
|                           |
+-----+
```

```
1 row in set (0.00 sec)
```

注意上面echo会自动增加换行符, -n参数可以避免

```
$ echo -n "Hello world" > /tmp/myfifo
```

```
mysql> select fifo_read('/tmp/myfifo');
```

```
+-----+
| fifo_read('/tmp/myfifo') |
+-----+
| Hello world              |
+-----+
```

```
1 row in set (0.01 sec)
```

写管道


```
mysql> select fifo_write('/tmp/myfifo','Hello world !!!');
+-----+
| fifo_write('/tmp/myfifo','Hello world !!!') |
+-----+
| true                                         |
+-----+
1 row in set (0.00 sec)

$ cat /tmp/myfifo
Hello world !!!

管道 /tmp/nofifo 不存在会返回false
mysql> select fifo_write('/tmp/nofifo',concat(mobile,'\r\n'))
from demo;
+-----+
| fifo_write('/tmp/nofifo',concat(mobile,'\r\n')) |
+-----+
| false                                           |
| false                                           |
| false                                           |
| false                                           |
| false                                           |
+-----+
5 rows in set (0.01 sec)
```

2.4. 内容输出到文本插件

Plugin Install and Uninstall

```
# Install
create function out2file returns string soname
'liboutfile.so';

# Uninstall
drop function out2file;
```

操作演示

```
# 安装插件

mysql> create function out2file returns string soname
'liboutfile.so';
Query OK, 0 rows affected (0.00 sec)

# 调用插件

mysql> select out2file('/tmp/myoutfile',"Helloworld!!!");
+-----+
| out2file('/tmp/myoutfile',"Helloworld!!!") |
+-----+
| true                                         |
+-----+
1 row in set (0.00 sec)

# 查看文件

root@netkiller ~/mysql-outfile-plugin % cat /tmp/myoutfile
Helloworld!!!
```

触发器应用

```
CREATE TABLE `demo` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(45) DEFAULT NULL,
  `sex` enum('Man','Woman') DEFAULT NULL,
  `address` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8

DROP TRIGGER IF EXISTS `test`.`demo_AFTER_INSERT`;

DELIMITER $$
USE `test`$$
```

```
CREATE DEFINER=`root`@`%` TRIGGER `test`.`demo_AFTER_INSERT`  
AFTER INSERT ON `demo` FOR EACH ROW  
BEGIN  
    set @rev = "";  
    SELECT  
    OUT2FILE('/tmp/demo.log',  
            CONCAT_WS(',',  
                      NEW.id,  
                      NEW.name,  
                      NEW.sex,  
                      NEW.address))  
    INTO @rev;  
END$$  
DELIMITER ;
```

3. Replication

3.1. Master Slave

Master

过程 19.1. Master 设置步骤

1. 配置 my.cnf 文件

确保主服务器主机上my.cnf文件的[mysqld]部分包括一个log-bin选项。该部分还应有一个server-id=Master_id选项

```
# vim /etc/mysql/my.cnf

server-id                = 1
log_bin                  = /var/log/mysql/mysql-bin.log
expire_logs_days        = 10
max_binlog_size          = 100M
binlog_do_db             = test
binlog_ignore_db         = mysql
```

bind-address默认是127.0.0.1你必须更改它，否则Slave将无法链接到 Master

```
#bind-address           = 127.0.0.1
bind-address             = 0.0.0.0
```

重启服务器

```
neo@netkiller:~$ sudo /etc/init.d/mysql reload
* Reloading MySQL database server mysqld      [ OK ]
```

建议使用reload,如果不起作用再用restart

```
mysql> SHOW GLOBAL VARIABLES like 'server_id';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| server_id     | 1     |
+-----+-----+
1 row in set (0.00 sec)
```

2. 登录slave服务器，测试主库3306工作情况，如果看到下面相关信息表示工作正常。

```
# telnet 192.168.1.246 3306
Trying 192.168.1.246...
Connected to 192.168.1.246.
Escape character is '^]'.
I
5.1.61-0ubuntu0.11.10.1-log1W<gs/*'#}p<u[J=5//:
```

3. 创建账户并授予REPLICATION SLAVE权限

```
mysql> GRANT REPLICATION SLAVE,REPLICATION CLIENT ON *.* TO
'replication'@'%.mydomain.com' IDENTIFIED BY 'slavepass';
mysql> FLUSH PRIVILEGES;
```

创建监控账号monitor（可选项），monitor 使用SHOW MASTER STATUS和SHOW SLAVE STATUS命令但没有复制权限

```
GRANT REPLICATION CLIENT ON *.* TO
monitor@'192.168.245.131' IDENTIFIED BY 'monitorpass'
```

4. 锁表禁止写入新数据

```
mysql> FLUSH TABLES WITH READ LOCK;
```

5. 查看Master 工作状态

```
mysql> SHOW MASTER STATUS;
+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----+-----+-----+-----+
| mysql-bin.000002 |      106 | test         | mysql              |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

如果显示下面内容表示，配置不正确

```
mysql> SHOW MASTER STATUS;
Empty set (0.02 sec)
```

取得快照并记录日志名和偏移量后，可以在主服务器上重新启用写活动

```
mysql> UNLOCK TABLES;
```



Slave

过程 19.2. Slave 设置步骤

1. 配置my.cnf

从服务器的ID必须与主服务器的ID不相同,如果设置多个从服务器,每个从服务器必须有一个唯一的server-id值,必须与主服务器的以及其它从服务器的不相同。

```
# vim /etc/mysql/my.cnf

[mysqld]
server-id          = 2
```

2.

```
# service mysql restart
mysql start/running, process 22893
```

3. 指定 master 相关参数

在从服务器上执行下面的语句,用你的系统的实际值替换选项值

```
mysql> CHANGE MASTER TO
->     MASTER_HOST='master_host_name',
->     MASTER_USER='replication_user_name',
->     MASTER_PASSWORD='replication_password',
->     MASTER_LOG_FILE='recorded_log_file_name',
->     MASTER_LOG_POS=recorded_log_position;
```

如果是全新服务器，空数据库可以忽略MASTER_LOG_FILE与MASTER_LOG_POS

```
CHANGE MASTER TO MASTER_HOST='192.168.245.129',
MASTER_USER='replication', MASTER_PASSWORD='slavepass';
```

如果是复制已经存在的数据库需要MASTER_LOG_FILE与MASTER_LOG_POS选项

首先到Master上运行 show master status 找到File与Position

```
mysql> show master status \G
***** 1. row
*****
          File: mysql-bin.000009
         Position: 3988
    Binlog_Do_DB:
 Binlog_Ignore_DB:
1 row in set (0.00 sec)
```

```
CHANGE MASTER TO
  MASTER_HOST='192.168.2.1',
  MASTER_USER='replication',
  MASTER_PASSWORD='kJZBTo3BjMx9AnmD9Ryn',
  MASTER_LOG_FILE='mysql-bin.000009',
  MASTER_LOG_POS=3988;
```

4. 启动从服务器线程

```
mysql> START SLAVE;
Query OK, 0 rows affected (0.00 sec)
```


5. SLAVE STATUS

```
mysql> SHOW SLAVE STATUS\G
***** 1. row
*****
      Slave_IO_State: Connecting to master
        Master_Host: 192.168.245.129
        Master_User: repl
        Master_Port: 3306
        Connect_Retry: 60
        Master_Log_File:
  Read_Master_Log_Pos: 4
        Relay_Log_File: mysqld-relay-bin.000002
        Relay_Log_Pos: 98
  Relay_Master_Log_File:
    Slave_IO_Running: Yes
    Slave_SQL_Running: Yes
      Replicate_Do_DB:
  Replicate_Ignore_DB:
    Replicate_Do_Table:
  Replicate_Ignore_Table:
  Replicate_Wild_Do_Table:
  Replicate_Wild_Ignore_Table:
        Last_Errno: 0
        Last_Error:
        Skip_Counter: 0
  Exec_Master_Log_Pos: 0
        Relay_Log_Space: 98
        Until_Condition: None
        Until_Log_File:
        Until_Log_Pos: 0
    Master_SSL_Allowed: No
    Master_SSL_CA_File:
    Master_SSL_CA_Path:
        Master_SSL_Cert:
        Master_SSL_Cipher:
        Master_SSL_Key:
  Seconds_Behind_Master: NULL
1 row in set (0.00 sec)
```

Testing

1. 登录 master

复制进程的信息

SHOW PROCESSLIST语句可以提供在主服务器上和从服务器上发生的关于复制的信息

```
mysql> SHOW PROCESSLIST\G
***** 1. row
*****
      Id: 62
      User: replication
      Host: ken-hx409.local:36465
      db: NULL
Command: Binlog Dump
      Time: 679
      State: Has sent all binlog to slave; waiting for binlog to
be updated
      Info: NULL
***** 2. row
*****
      Id: 64
      User: root
      Host: localhost
      db: NULL
Command: Query
      Time: 0
      State: NULL
      Info: SHOW PROCESSLIST
2 rows in set (0.00 sec)
```

2. 登录从库，查看复制线程

```
mysql> SHOW PROCESSLIST\G
***** 1. row
*****
```

```

    Id: 273
    User: root
    Host: localhost
    db: NULL
Command: Query
    Time: 0
    State: NULL
    Info: SHOW PROCESSLIST
***** 2. row
*****
    Id: 276
    User: system user
    Host:
    db: NULL
Command: Connect
    Time: 2
    State: Waiting for master to send event
    Info: NULL
***** 3. row
*****
    Id: 277
    User: system user
    Host:
    db: NULL
Command: Connect
    Time: 2
    State: Has read all relay log; waiting for the slave I/O
thread to update it
    Info: NULL
3 rows in set (0.00 sec)

```

如果没有复制进程，请使用start slave;启动

```

mysql> SHOW PROCESSLIST\G
***** 1. row
*****
    Id: 273
    User: root
    Host: localhost
    db: NULL
Command: Query

```

```
Time: 0
State: NULL
Info: SHOW PROCESSLIST
1 row in set (0.02 sec)

mysql> start slave;
Query OK, 0 rows affected (0.00 sec)
```

3. 登录 master

```
mysql> insert into foo(id,data) values(2,'Hello world!!!');
Query OK, 1 row affected (0.00 sec)
```

4. 登录 slave

```
mysql> select * from foo;
```

在master服务器上插入一条记录，你可以立刻在slave服务器上看到变化。

将现有数据库迁移到主从结构数据库

数据库已经存在的情况下怎么迁移

1. Master 锁表禁止写入新数据

```
mysql> FLUSH TABLES WITH READ LOCK;
```

2. Slave 停止复制进程

```
mysql> stop slave;
```

3. 备份Master数据库

```
mysqldump yourdb | gzip > yourdb.sql.gz
```

4. 恢复数据库

如果使用mysqldump备份主服务器的数据，将转储文件装载到从服务器

```
# zcat yourdb.sql.gz | mysql -u root -p yourdb
```

5. 启动 Slave 复制进程

```
mysql> start slave;
```

6. 解除 Master 表锁定

```
mysql> UNLOCK TABLES;
```

MyIASM引擎可以采用下面方法

备份数据库

```
# tar zcvf mysql-snapshot.tar.gz /var/lib/mysql/neo
```

复制给从数据库

```
scp mysql-snapshot.tar.gz neo@192.168.245.131:/tmp
```

snapshot 恢复

```
$ tar zxvf mysql-snapshot.tar.gz  
$ cd /var/lib/mysql  
  
$ mv /tmp/var/lib/mysql/neo .  
$ sudo chown mysql:mysql -R neo
```

重新启动Mysql

```
$ sudo /etc/init.d/mysql restart
```

有兴趣可以看看mysqlhotcopy

主从复制安全问题

复制帐号权限

```
grant replication slave on *.* to 'replication'@'192.168.1.%'  
identified by '000000';
```

主库数据库操作帐号权限

```
grant DELETE, INSERT, SELECT, UPDATE ON your_user.* to  
yourdb@'your_host' identified by 'password' with grant option;
```

从库数据库操作帐号权限

```
grant SELECT ON your_user.* to yourdb@'your_host' identified by  
'password' with grant option;
```

从库必须收回写操作

3.2. Master Master(主主)

Master A

my.cnf 文件加入下面的内容

```
cp /etc/mysql/my.cnf /etc/mysql/my.cnf.old  
vim /etc/mysql/my.cnf  
  
[mysqld]  
server-id = 1  
log-bin=/data/mysql/binlog/binlog  
binlog-do-db = test  
binlog-ignore-db=mysql  
  
log-slave-updates  
sync_binlog=1  
auto_increment_offset=1  
auto_increment_increment=2  
replicate-do-db = test  
replicate-ignore-db = mysql,information_schema
```

创建复制权限

```
grant replication slave on *.* to 'replication'@'192.168.1.%'  
identified by '000000';  
flush privileges;
```

```
mysql>flush tables with read lock;  
  
mysql> show master status\G  
***** 1. row *****  
File: binlog.000007  
Position: 107  
Binlog_Do_DB: test  
Binlog_Ignore_DB: mysql  
1 row in set (0.00 sec)
```

Master B

创建复制权限

```
grant replication slave on *.* to 'replication'@'192.168.1.%'  
identified by '000000';  
flush privileges;
```

my.cnf 文件加入下面的内容

```
[mysqld]  
server-id = 2  
log-bin = /data/mysql/binlog/binlog  
replicate-do-db = test  
replicate-ignore-db = mysql,information_schema  
  
binlog-do-db = test  
binlog-ignore-db=mysql  
log-slave-updates  
sync_binlog=1  
auto_increment_offset=2
```



```
auto_increment_increment=2
```

B 与 A 配置文件不同的两处。

```
server-id = 2  
auto_increment_offset=2
```

```
mysql> show master status\G  
***** 1. row *****  
File: binlog.000005  
Position: 107  
Binlog_Do_DB: test  
Binlog_Ignore_DB: mysql  
1 row in set (0.00 sec)
```

将**Master A** 数据库 同步到 **Master B** 两端数据库内容保持一致

Master A, 首先锁表为只读状态

```
mysqldump --databases test > /tmp/test.sql
```

Master B 创建一个与Master A同名的空数据库,然后将备份文件恢复到数据库中

```
# mysql  
mysql> CREATE DATABASE test;  
mysql>\q  
  
# scp 192.168.1.1:/tmp/test.sql ./  
# mysql -uroot -p test < /tmp/test.sql
```

Master A - B 同步两端数据库

master-A

```
mysql>change master to master_host='192.168.1.2',  
master_user='replication', master_password='000000',  
master_log_file='binlog.000005', master_log_pos=107;
```

master-B

```
mysql>change master to master_host='192.168.1.1',  
master_user='replication', master_password='000000',  
master_log_file='binlog.000007', master_log_pos=107;
```

Master A 数据库解除只读权限

Master A 解锁

```
mysql> UNLOCK TABLES;
```

查看主的工作状态

分别在Master A与B 上运行

```
mysql>show slave status\G;
```

```
Slave_IO_Running: Yes  
Slave_SQL_Running: Yes
```

3.3. Semisynchronous Replication

Master

```
mysql> SHOW VARIABLES LIKE "have_dynamic_loading";
```

```
+-----+-----+  
| Variable_name      | Value |  
+-----+-----+  
| have_dynamic_loading | YES   |  
+-----+-----+  
1 row in set (0.00 sec)
```

```
mysql>
```

Master 配置

```
mysql> install plugin rpl_semi_sync_master SONAME  
'semisync_master.so';  
mysql> set global rpl_semi_sync_master_enabled = 1;  
mysql> set global rpl_semi_sync_master_timeout = 30;  
mysql> select * from mysql.plugin;
```

```
+-----+-----+  
| name              | dl              |  
+-----+-----+  
| rpl_semi_sync_master | semisync_master.so |  
+-----+-----+  
1 row in set (0.00 sec)
```

状态查看

```
mysql> SHOW VARIABLES LIKE "%semi%";
+-----+-----+
| Variable_name | Value |
+-----+-----+
| rpl_semi_sync_master_enabled | ON |
| rpl_semi_sync_master_timeout | 10 |
| rpl_semi_sync_master_trace_level | 32 |
| rpl_semi_sync_master_wait_no_slave | ON |
+-----+-----+
4 rows in set (0.00 sec)
```

Slave 配置

```
install plugin rpl_semi_sync_slave soname 'semisync_slave.so';
set global rpl_semi_sync_slave_enabled = 1;
stop slave io_thread;
start slave io_thread;
```

Slave 状态查看

```
show global status like 'rpl_semi%';
```

卸载插件

卸载插件 UNINSTALL PLUGIN plugin_name

```
UNINSTALL PLUGIN rpl_semi_sync_master;
UNINSTALL PLUGIN rpl_semi_sync_slave;
```

my.cnf

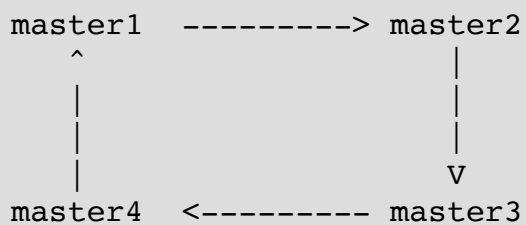
编辑my.cnf加入

```
# On Master
[mysqld]
rpl_semi_sync_master_enabled=1
rpl_semi_sync_master_timeout=1000 # 1 second

# On Slave
[mysqld]
rpl_semi_sync_slave_enabled=1
```

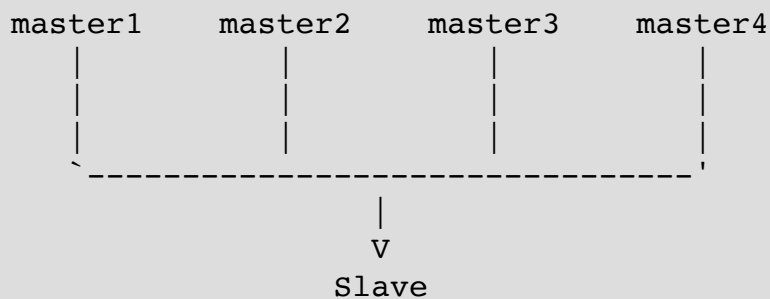
3.4. multi-master replication

MySQL 5.7 以上版本才能使用



3.5. multi-source replication

MySQL 5.7 以上版本才能使用



slave 配置

```
slave> change master to master_host="172.16.0.1",
master_port=3306,
master_user="replication",master_password="password" for
channel="master1";
slave> change master to master_host="172.16.0.2",
master_port=3306,
master_user="replication",master_password="password" for
channel="master2";

slave> start slave for channel="master1";
slave> start slave for channel="master2";
```

检查从服务器状态

```
slave > SHOW SLAVE STATUS FOR CHANNEL="master1"\G
slave > SHOW SLAVE STATUS FOR CHANNEL="master2"\G
```

测试,分别在两个主服务器上创建数据库,然后查看从数据库同步结果.

```
master1 > create database master1;
master2 > create database master2;

slave > show databases like 'master%';
+-----+
| Database (master%) |
+-----+
| master1             |
| master2             |
+-----+
```

3.6. 与复制有关的问题

主从不同步问题

执行下面语句

```
stop slave;
set global sql_slave_skip_counter =1 ;
start slave;

mysql> slave stop;
mysql> set GLOBAL SQL_SLAVE_SKIP_COUNTER=1;
mysql> slave start;
```

Seconds_Behind_Master 值从NULL变为大于等于0是表示已经同步

```
Seconds_Behind_Master: NULL
Seconds_Behind_Master: 8893
```

mysql-bin 清理问题

缺省expire-logs-days为30天。这里设为7天，可根据自己情况调整。

```
[mysqld]
expire-logs-days = 7
```

通过SQL删除

```
删除某个日志:      mysql>PURGE MASTER LOGS TO 'mysql-bin.015';
删除某天前的日志:  mysql>PURGE MASTER LOGS BEFORE '2010-10-25'
```

```
14:00:00';
```

跳过 Last_Errno

修改mysql配置文件 /etc/my.cnf 在 [mysqld]下加一行

```
[mysqld]
slave_skip_errors = 1062
```

跳过所有错误

```
slave-skip-errors=all
```

重置Slave

```
STOP SLAVE;
RESET SLAVE;
START SLAVE;
```

3.7. GTID

5.6 新增功能

```
mysql> show global variables like '%gtid%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| binlog_gtid_simple_recovery | OFF |
| enforce_gtid_consistency | OFF |
| gtid_executed | |
| gtid_mode | OFF |
| gtid_owned | |
| gtid_purged | |
+-----+-----+
```



```
| simplified_binlog_gtid_recovery | OFF |
+-----+-----+
7 rows in set (0.00 sec)
```

Master

```
[root@master mysql]# vim my.cnf

binlog-format=ROW
log-slave-updates=true
gtid-mode=on
enforce-gtid-consistency=true
master-info-repository=TABLE
relay-log-info-repository=TABLE
sync-master-info=1
slave-parallel-workers=2

binlog-checksum=CRC32
master-verify-checksum=1
slave-sql-verify-checksum=1
binlog-rows-query-log_events=1
port=3306
report-host=192.168.1.21
report-port=3306
server_id = 1
```

创建有复制权限的用户

```
GRANT REPLICATION SLAVE,REPLICATION CLIENT ON *.* TO
'replication'@'%mydomain.com' IDENTIFIED BY 'slavepass';
FLUSH PRIVILEGES;
```

Slave

```
[root@slave mysql]# vim my.cnf

relay-log = relay-log
```

```
relay-log-index = relay-log.index
binlog-format=ROW
log-slave-updates=true

gtid-mode=on
enforce-gtid-consistency=true

master-info-repository=TABLE
relay-log-info-repository=TABLE
sync-master-info=1
slave-parallel-workers=2

;binlog-checksum=CRC32
master-verify-checksum=1
slave-sql-verify-checksum=1

binlog-rows-query-log_events=1
report-port=3306
port=3306
report-host=192.168.1.22
server_id = 10
```

登录到Master并进行复制

```
CHANGE MASTER TO
  MASTER_HOST='192.168.2.1',
  MASTER_USER='replication',
  MASTER_PASSWORD='kJZBTo3BjMx9AnmD9Ryn',
  MASTER_AUTO_POSITION=1;
```

就这么简单，你不再需要指定 MASTER_LOG_FILE='mysql-bin.000009', MASTER_LOG_POS=3988 两个参数。

4. MySQL Cluster

The cluster need a lot of server for experiments, if you haven't any server for one, I have a good idea that using Vmware for you.

at first, let's create lots of virtual machine(You MUST have a third server). and then follow me step by step learning how to set up a mysql cluster on your virtual machine.



```
mgm          192.168.0.1          # Management
data         192.168.0.2          # Ndbd Node
data         192.168.0.3          # Ndbd Node
sql          192.168.0.4          # SQL Node
sql          192.168.0.5          # SQL Node
```

4.1. Management node (MGM node)

```
neo@mgm:~$ sudo vim /var/lib/mysql-cluster/config.ini

[NDBD DEFAULT]
NoOfReplicas=2
DataMemory=80M
IndexMemory=18M

[MYSQLD DEFAULT]

[NDB_MGMD DEFAULT]

[TCP DEFAULT]
portnumber=2202

[NDB_MGMD]
hostname=192.168.0.1
datadir=/var/lib/mysql-cluster
```

```
[NDBD]
hostname=192.168.0.2
datadir=/var/lib/mysql-cluster

[NDBD]
hostname=192.168.0.3
datadir=/var/lib/mysql-cluster

[MYSQLD]
hostname=192.168.0.4

[MYSQLD]
hostname=192.168.0.5
```

4.2. Data node

my.cnf

```
neo@data:~$ sudo vim /etc/mysql/my.cnf

[mysqld]
ndbcluster
ndb-connectstring=192.168.0.1 # the IP of the MANAGMENT
SERVER
[mysql_cluster]
ndb-connectstring=192.168.0.1 # the IP of the MANAGMENT
SERVER
```

4.3. SQL node

my.cnf

```
neo@sql:~$ sudo vim /etc/mysql/my.cnf

[mysqld]
ndbcluster
ndb-connectstring=192.168.0.1 # the IP of the MANAGMENT
```

```
SERVER
[mysql_cluster]
ndb-connectstring=192.168.0.1 # the IP of the MANAGMENT
SERVER
```

4.4. Starting

1. starting mgm

```
neo@mgm:~$ sudo ndb_mgmd -f /var/lib/mysql-
cluster/config.ini
```

2. initial ndbd

```
neo@data:~$ sudo ndbd --initial
```

首次运行需要 --initial 参数，以后不需要。

4.5. Shutdown

MGM

```
$ sudo ndb_mgm -e shutdown
```

4.6. Testing

```
neo@mgm:~$ ndb_mgm
-- NDB Cluster -- Management Client --
ndb_mgm> show
Connected to Management Server at: localhost:1186
Cluster Configuration
-----
```

```

[ndbd(NDB)]      2 node(s)
id=2      @192.168.0.2  (Version: 5.0.51, Nodegroup: 0)
id=3      @192.168.0.3  (Version: 5.0.51, Nodegroup: 0, Master)

[ndb_mgmd(MGM)] 1 node(s)
id=1      @192.168.0.1  (Version: 5.0.51)

[mysqld(API)]   2 node(s)
id=4      @192.168.0.4  (Version: 5.0.51)
id=5      @192.168.0.5  (Version: 5.0.51)

ndb_mgm>

```

与没有使用簇的MySQL相比，在MySQL簇内操作数据的方式没有太大的区别。

执行这类操作时应记住三点

1. 表必须用ENGINE=NDB或ENGINE=NDBCLUSTER选项创建，或用ALTER TABLE选项更改，以使用NDB Cluster存储引擎在Cluster内复制它们。如果使用mysqldump的输出从已有数据库导入表，可在文本编辑器中打开SQL脚本，并将该选项添加到任何表创建语句，或用这类选项之一替换任何已有的ENGINE（或TYPE）选项。
2. 另外请记住，每个NDB表必须有一个主键。如果在创建表时用户未定义主键，NDB Cluster存储引擎将自动生成隐含的主键。（注释：该隐含键也将占用空间，就像任何其他的表索引一样。由于没有足够的内存来容纳这些自动创建的键，出现问题并不罕见）。
3. 当你在一个节点上运行create database mydb;你去其他sql node上执行show databases;将不能看到mydb,你需要创建它，然后use mydb; show tables;你将看到同步的表。

SQL Node 1

```

neo@sql:~$ mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 7
Server version: 5.0.51a-3ubuntu5.1 (Ubuntu)

```



```
mysql> create database cluster;
Query OK, 1 row affected (0.00 sec)
```

```
mysql> show databases;
```

Database
information_schema
cluster
example
mydb
mysql
neo

```
6 rows in set (0.13 sec)
```

```
mysql> use cluster;
```

```
Reading table information for completion of table and column
names
```

```
You can turn off this feature to get a quicker startup with -A
```

```
Database changed
```

```
mysql> show tables;
```

Tables_in_cluster
city

```
1 row in set (0.01 sec)
```

```
mysql> select * from city;
```

id	name
1	Shenzhen
2	Guangdong

```
2 rows in set (0.03 sec)
```

```
mysql>
```


5. MySQL Proxy

5.1. Ubuntu

安装环境 Ubuntu 13.04

```
$ sudo apt-get install mysql-proxy
```

ENABLED改为true

```
$ sudo vim /etc/default/mysql-proxy  
ENABLED="true"  
OPTIONS="--defaults-file=/etc/mysql/mysql-proxy.cnf"
```

配置 /etc/mysql/mysql-proxy.cnf

```
$ sudo vim /etc/mysql/mysql-proxy.cnf  
  
[mysql-proxy]  
daemon = true  
user = mysql  
proxy-skip-profiling = true  
keepalive = true  
max-open-files = 2048  
event-threads = 50  
pid-file = /var/run/mysql-proxy.pid  
log-file = /var/log/mysql-proxy.log  
log-level = debug  
admin-address=:4401  
admin-username=admin  
admin-password=passw0rd  
admin-lua-script=/usr/local/lib/mysql-proxy/lua/admin.lua  
proxy-address = 0.0.0.0:3307  
proxy-backend-addresses = 192.168.2.1:3306  
proxy-read-only-backend-addresses=192.168.6.2:3306,  
192.168.6.1:3306
```

```
proxy-lua-script=/usr/lib/mysql-proxy/lua/proxy/balance.lua
```

修改权限，这个步骤不能省略，否则无法启动。

```
$ sudo chmod 0660 /etc/mysql/mysql-proxy.cnf
```

启动mysql-proxy

```
$ sudo /etc/init.d/mysql-proxy start
* Starting MySQL Proxy daemon... [ OK ]
```

测试3307端口

```
$ mysql -hlocalhost -P3307 -uroot -p
```

mysql-proxy 软件包所含文件如下：

```
$ dpkg -L mysql-proxy
/.
/etc
/etc/default
/etc/default/mysql-proxy
/etc/init.d
/etc/init.d/mysql-proxy
/usr
/usr/share
/usr/share/mysql-proxy
/usr/share/mysql-proxy/active-queries.lua
/usr/share/mysql-proxy/active-transactions.lua
/usr/share/mysql-proxy/admin-sql.lua
/usr/share/mysql-proxy/admin.lua
/usr/share/mysql-proxy/analyze-query.lua
/usr/share/mysql-proxy/auditing.lua
/usr/share/mysql-proxy/commit-obfuscator.lua
/usr/share/mysql-proxy/histogram.lua
```

```
/usr/share/mysql-proxy/load-multi.lua
/usr/share/mysql-proxy/ro-balance.lua
/usr/share/mysql-proxy/ro-pooling.lua
/usr/share/mysql-proxy/rw-splitting.lua
/usr/share/mysql-proxy/xtab.lua
/usr/share/doc
/usr/share/doc/mysql-proxy
/usr/share/doc/mysql-proxy/README.TESTS.gz
/usr/share/doc/mysql-proxy/README
/usr/share/doc/mysql-proxy/copyright
/usr/share/doc/mysql-proxy/changelog.Debian.gz
/usr/lib
/usr/lib/libmysql-chassis-glibext.so.0.0.0
/usr/lib/libmysql-chassis-timing.so.0.0.0
/usr/lib/libmysql-chassis.so.0.0.0
/usr/lib/libmysql-proxy.so.0.0.0
/usr/lib/mysql-proxy
/usr/lib/mysql-proxy/lua
/usr/lib/mysql-proxy/lua/proxy
/usr/lib/mysql-proxy/lua/proxy/auto-config.lua
/usr/lib/mysql-proxy/lua/proxy/balance.lua
/usr/lib/mysql-proxy/lua/proxy/commands.lua
/usr/lib/mysql-proxy/lua/proxy/parser.lua
/usr/lib/mysql-proxy/lua/proxy/tokenizer.lua
/usr/lib/mysql-proxy/lua/proxy/test.lua
/usr/lib/mysql-proxy/lua/admin.lua
/usr/lib/mysql-proxy/lua/lfs.so
/usr/lib/mysql-proxy/lua/glib2.so
/usr/lib/mysql-proxy/lua/chassis.so
/usr/lib/mysql-proxy/lua/mysql.so
/usr/lib/mysql-proxy/lua/lpeg.so
/usr/lib/mysql-proxy/lua/posix.so
/usr/lib/mysql-proxy/plugins
/usr/lib/mysql-proxy/plugins/libadmin.so
/usr/lib/mysql-proxy/plugins/libproxy.so
/usr/lib/mysql-proxy/plugins/libreplicant.so
/usr/lib/mysql-proxy/plugins/libdebug.so
/usr/lib/pkgconfig
/usr/lib/pkgconfig/mysql-proxy.pc
/usr/lib/pkgconfig/mysql-chassis.pc
/usr/bin
/usr/bin/mysql-binlog-dump
/usr/bin/mysql-myisam-dump
/usr/bin/mysql-proxy
/usr/include
```

```
/usr/include/network-mysqld.h
/usr/include/network-mysqld-lua.h
/usr/include/network-mysqld-proto.h
/usr/include/network-mysqld-binlog.h
/usr/include/network-mysqld-packet.h
/usr/include/network-mysqld-masterinfo.h
/usr/include/network-conn-pool.h
/usr/include/network-conn-pool-lua.h
/usr/include/network-queue.h
/usr/include/network-socket.h
/usr/include/network-socket-lua.h
/usr/include/network-address.h
/usr/include/network-address-lua.h
/usr/include/sys-pedantic.h
/usr/include/chassis-plugin.h
/usr/include/chassis-log.h
/usr/include/chassis-keyfile.h
/usr/include/chassis-mainloop.h
/usr/include/chassis-path.h
/usr/include/chassis-filemode.h
/usr/include/chassis-limits.h
/usr/include/chassis-event-thread.h
/usr/include/chassis-gtimeval.h
/usr/include/glib-ext.h
/usr/include/glib-ext-ref.h
/usr/include/string-len.h
/usr/include/lua-load-factory.h
/usr/include/lua-scope.h
/usr/include/lua-env.h
/usr/include/network-injection.h
/usr/include/network-injection-lua.h
/usr/include/chassis-shutdown-hooks.h
/usr/include/chassis-exports.h
/usr/include/network-exports.h
/usr/include/network-backend.h
/usr/include/network-backend-lua.h
/usr/include/disable-dtrace.h
/usr/include/lua-registry-keys.h
/usr/include/chassis-stats.h
/usr/include/chassis-timings.h
/usr/include/chassis-frontend.h
/usr/include/chassis-options.h
/usr/include/chassis-win32-service.h
/usr/include/chassis-unix-daemon.h
/usr/include/my_rdtsc.h
```

```
/usr/lib/libmysql-chassis-glibext.so.0
/usr/lib/libmysql-chassis-glibext.so
/usr/lib/libmysql-proxy.so
/usr/lib/libmysql-chassis-timing.so.0
/usr/lib/libmysql-chassis-timing.so
/usr/lib/libmysql-proxy.so.0
/usr/lib/libmysql-chassis.so.0
/usr/lib/libmysql-chassis.so
```

5.2. CentOS

```
# yum install mysql-proxy
```

```
# cat /etc/sysconfig/mysql-proxy
# Options for mysql-proxy
ADMIN_USER="admin"
ADMIN_PASSWORD=""
ADMIN_LUA_SCRIPT="/usr/lib64/mysql-proxy/lua/admin.lua"
PROXY_USER="mysql-proxy"
PROXY_OPTIONS="--daemon --log-level=info --log-use-syslog"
```

修改PROXY_OPTIONS选项

```
#PROXY_OPTIONS="--daemon --log-level=info --log-use-syslog"
PROXY_OPTIONS="--defaults-file=/etc/mysql/mysql-proxy.cnf"
```

```
# mkdir /etc/mysql
# vim /etc/mysql/mysql-proxy.cnf

[mysql-proxy]
daemon = true
user = mysql-proxy
proxy-skip-profiling = true
keepalive = true
;max-open-files = 2048
event-threads = 512
```

```
pid-file = /var/run/mysql-proxy.pid
log-file = /var/log/mysql-proxy.log
log-level = debug
admin-address=:4401
admin-username=admin
admin-password=passwd
admin-lua-script=/usr/lib64/mysql-proxy/lua/admin.lua
proxy-address = 0.0.0.0:3307
proxy-backend-addresses = 192.168.2.1:3306
proxy-read-only-backend-addresses=192.168.6.2:3306,
192.168.6.1:3306
proxy-lua-script=/usr/lib64/mysql-proxy/lua/proxy/balance.lua
```

修复启动脚本BUG

```
# vim /etc/init.d/mysql-proxy

#daemon $prog $PROXY_OPTIONS --pid-file=$PROXY_PID --
user=$PROXY_USER --admin-username="$ADMIN_USER" --admin-lua-
script="$ADMIN_LUA_SCRIPT" --admin-password="$ADMIN_PASSWORD"
注视这行，改为下面的一行代码
daemon $prog $PROXY_OPTIONS --pid-file=$PROXY_PID
```

启动mysql-proxy

```
# chkconfig mysql-proxy on
# service mysql-proxy start
Starting mysql-proxy: [
OK ]
```

FAQ

(critical) (libevent) evsignal_init: socketpair: Too many open files

```
;max-open-files = 2048
```

注释max-open-files = 2048, 使用ulimit -SHn 2048设置

6. MySQL Router

6.1. 安装 MySQL Router

```
# yum install mysql-router -y
```

MySQL Router 软件包中所含的文件。

```
[root@netkiller ~]# rpm -ql mysql-router-2.0.3-1.el7
/etc/mysqlrouter
/etc/mysqlrouter/mysqlrouter.ini
/usr/lib/systemd/system/mysqlrouter.service
/usr/lib/tmpfiles.d/mysqlrouter.conf
/usr/lib64/libmysqlharness.so
/usr/lib64/libmysqlharness.so.0
/usr/lib64/libmysqlrouter.so
/usr/lib64/libmysqlrouter.so.1
/usr/lib64/mysqlrouter
/usr/lib64/mysqlrouter/fabric_cache.so
/usr/lib64/mysqlrouter/keepalive.so
/usr/lib64/mysqlrouter/logger.so
/usr/lib64/mysqlrouter/mysql_protocol.so
/usr/lib64/mysqlrouter/routing.so
/usr/sbin/mysqlrouter
/usr/share/doc/mysql-router-2.0.3
/usr/share/doc/mysql-router-2.0.3/License.txt
/usr/share/doc/mysql-router-2.0.3/README.txt
/var/log/mysqlrouter
/var/run/mysqlrouter
```

6.2. 配置 MySQL Router

默认配置

```
# cat /etc/mysqlrouter/mysqlrouter.ini
```



```
# Copyright (c) 2015, Oracle and/or its affiliates. All rights
reserved.
#
# This program is free software; you can redistribute it and/or
modify
# it under the terms of the GNU General Public License as
published by
# the Free Software Foundation; version 2 of the License.
#
# This program is distributed in the hope that it will be
useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty
of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public
License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA
02110-1301 USA

#
# MySQL Router configuration file
#
# Documentation is available at
#   http://dev.mysql.com/doc/mysql-router/en/

[DEFAULT]
logging_folder = /var/log/mysqlrouter/
plugin_folder = /usr/lib64/mysqlrouter
runtime_folder = /var/run/mysqlrouter
config_folder = /etc/mysqlrouter

[logger]
level = info

# If no plugin is configured which starts a service, keepalive
# will make sure MySQL Router will not immediately exit. It is
# safe to remove once Router is configured.
[keepalive]
interval = 60
```

主备配置

适用于 MySQL Master-Master / Master-Slave 方案，当一台Master出现故障后另一台Master或者Slave接管

```
[routing:failover]
bind_address = 192.168.0.5
bind_port = 3306
max_connections = 1024
mode = read-write
destinations = 192.168.0.10:3306,192.168.0.11:3306
```

负载均衡配置

主要用于输入均衡

```
[routing:balancing]
bind_address = 192.168.0.5
bind_port = 3307
connect_timeout = 3
max_connections = 1024
mode = read-only
destinations = 192.168.0.20:3306,192.168.0.21:3306
```

6.3. MySQL Router , Haproxy, LVS 的选择

MySQL Router目前仍在成长中，如果你只需要负载均衡与主备，那么LVS性能更高，Haproxy也更成熟。

7. my.cnf

7.1. bind-address

MySQL 通过 yum 安装后默认是监听 127.0.0.1 / ::1 如果你希望从其他IP访问 3306端口，需要修改绑定地址为 0.0.0.0

```
bind-address=127.0.0.1
```

0.0.0.0:3306

```
bind-address = 0.0.0.0
```

指定IP地址

```
[root@localhost ~]# ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel
state UP group default qlen 1000
    link/ether 00:e0:70:81:9e:48 brd ff:ff:ff:ff:ff:ff
    inet 192.168.30.10/24 brd 192.168.30.255 scope global noprefixroute
enp2s0
        valid_lft forever preferred_lft forever
    inet 192.168.30.11/24 brd 192.168.30.255 scope global secondary
noprefixroute enp2s0
        valid_lft forever preferred_lft forever
    inet 192.168.30.12/24 brd 192.168.30.255 scope global secondary
noprefixroute enp2s0
        valid_lft forever preferred_lft forever
    inet 192.168.30.13/24 brd 192.168.30.255 scope global secondary
noprefixroute enp2s0
```

```
    valid_lft forever preferred_lft forever
    inet 192.168.30.14/24 brd 192.168.30.255 scope global secondary
noprofixroute enp2s0
    valid_lft forever preferred_lft forever
    inet6 fe80::2e0:70ff:fe81:9e48/64 scope link noprofixroute
    valid_lft forever preferred_lft forever
3: wlp1s0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group
default qlen 1000
    link/ether 40:9f:38:b6:e0:55 brd ff:ff:ff:ff:ff:ff
4: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue
state DOWN group default
    link/ether 02:42:f0:6f:b3:4b brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
    valid_lft forever preferred_lft forever
45: br-a32falca1437: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc
noqueue state DOWN group default
    link/ether 02:42:d7:ae:ea:5d brd ff:ff:ff:ff:ff:ff
    inet 172.18.0.1/16 brd 172.18.255.255 scope global br-a32falca1437
    valid_lft forever preferred_lft forever
```

上面服务器上配置多个IP地址

```
bind-address=192.168.30.10
```

7.2. 禁用TCP/IP链接

与bind-address互斥，skip-networking 开启，只能通过UNIX SOCKET链接，而不能使用IP地址链接

```
[mysqld]
skip-networking
```

7.3. 配置字符集

Configuring Database Character Encoding

```
mysql> SHOW VARIABLES LIKE 'character_set_%';
```

Variable_name	Value
character_set_client	latin1
character_set_connection	latin1
character_set_database	utf8
character_set_filesystem	binary
character_set_results	latin1
character_set_server	latin1
character_set_system	utf8
character_sets_dir	/usr/share/mysqlCharsets/

```
8 rows in set (0.00 sec)
```

Server Character Set and Collation

```
shell> mysqld --character-set-server=latin1
shell> mysqld --character-set-server=latin1 \
  --collation-server=latin1_swedish_ci
```

\$ vim /etc/mysql/my.cnf

```
[mysqld]
character-set-server=utf8
collation_server=utf8_general_ci
init_connect='SET NAMES utf8'

[client]
character_set_client=utf8
```

```
mysql --default-character-set=utf8 -u root -p
```

```
mysql> show variables like 'character%';
```

Variable_name	Value
---------------	-------

```
| character_set_client      | utf8 |
| character_set_connection | utf8 |
| character_set_database   | utf8 |
| character_set_filesystem | binary |
| character_set_results    | utf8 |
| character_set_server     | utf8 |
| character_set_system     | utf8 |
| character_sets_dir       | /usr/share/mysql/charsets/ |
+-----+-----+
8 rows in set (0.00 sec)
```

7.4. 最大链接数 `max_connections`

```
[mysqld]
max_connections=250
```

7.5. 默认引擎 `storage-engine`

```
[mysqld]
default-storage-engine=INNODB
```

7.6. `max_allowed_packet`

```
max_allowed_packet=8M
```

7.7. `skip-name-resolve`

跳过域名解析

```
# vim /etc/mysql/my.cnf

[mysqld]
skip-name-resolve
```

MySQL 登录缓慢，大量用户排队等待

```
mysql> SHOW FULL PROCESSLIST;
```

Id	User	Host	db	Command
718	unauthenticated user	192.168.3.124:42075	NULL	Connect
NULL	login	NULL		
719	unauthenticated user	192.168.3.124:42073	NULL	Connect
NULL	login	NULL		
720	unauthenticated user	192.168.3.124:42074	NULL	Connect
NULL	login	NULL		
721	unauthenticated user	192.168.3.124:42077	NULL	Connect
NULL	login	NULL		
722	unauthenticated user	192.168.3.124:42076	NULL	Connect
NULL	login	NULL		
723	unauthenticated user	192.168.3.124:42079	NULL	Connect
NULL	login	NULL		
724	unauthenticated user	192.168.3.124:42078	NULL	Connect
NULL	login	NULL		
725	unauthenticated user	192.168.3.124:42081	NULL	Connect
NULL	login	NULL		
726	unauthenticated user	192.168.3.124:42080	NULL	Connect
NULL	login	NULL		
727	unauthenticated user	192.168.3.124:42082	NULL	Connect
NULL	login	NULL		
728	unauthenticated user	192.168.3.124:42083	NULL	Connect
NULL	login	NULL		
729	unauthenticated user	192.168.3.124:42085	NULL	Connect
NULL	login	NULL		
730	unauthenticated user	192.168.3.124:42084	NULL	Connect
NULL	login	NULL		
731	unauthenticated user	192.168.3.124:42086	NULL	Connect
NULL	login	NULL		
732	unauthenticated user	192.168.3.124:42087	NULL	Connect
NULL	login	NULL		
733	unauthenticated user	192.168.3.124:42088	NULL	Connect
NULL	login	NULL		
734	unauthenticated user	192.168.3.124:42089	NULL	Connect
NULL	login	NULL		
735	unauthenticated user	192.168.3.124:42090	NULL	Connect
NULL	login	NULL		
736	unauthenticated user	192.168.3.124:42091	NULL	Connect
NULL	login	NULL		
737	unauthenticated user	192.168.3.124:42092	NULL	Connect
NULL	login	NULL		

738	unauthenticated user	192.168.3.124:42093	NULL	Connect
NULL	login	NULL		
739	unauthenticated user	192.168.3.124:42094	NULL	Connect
NULL	login	NULL		
740	unauthenticated user	192.168.3.124:42095	NULL	Connect
NULL	login	NULL		
741	unauthenticated user	192.168.3.124:42096	NULL	Connect
NULL	login	NULL		
742	unauthenticated user	192.168.3.124:42097	NULL	Connect
NULL	login	NULL		
743	unauthenticated user	192.168.3.124:42098	NULL	Connect
NULL	login	NULL		
744	unauthenticated user	192.168.3.124:42099	NULL	Connect
NULL	login	NULL		
745	unauthenticated user	192.168.3.124:42100	NULL	Connect
NULL	login	NULL		
746	unauthenticated user	192.168.3.124:42101	NULL	Connect
NULL	login	NULL		
747	unauthenticated user	192.168.3.124:42102	NULL	Connect
NULL	login	NULL		
748	unauthenticated user	192.168.3.124:42103	NULL	Connect
NULL	login	NULL		
749	unauthenticated user	192.168.3.124:42104	NULL	Connect
NULL	login	NULL		
750	unauthenticated user	192.168.3.124:42068	NULL	Connect
NULL	login	NULL		
751	unauthenticated user	192.168.3.124:42064	NULL	Connect
NULL	login	NULL		
752	unauthenticated user	192.168.3.124:42071	NULL	Connect
NULL	login	NULL		
753	unauthenticated user	192.168.3.124:42072	NULL	Connect
NULL	login	NULL		
754	unauthenticated user	192.168.3.124:42067	NULL	Connect
NULL	login	NULL		
755	unauthenticated user	192.168.3.124:42070	NULL	Connect
NULL	login	NULL		
756	unauthenticated user	192.168.3.124:42069	NULL	Connect
NULL	login	NULL		
757	unauthenticated user	192.168.3.124:42065	NULL	Connect
NULL	login	NULL		
758	unauthenticated user	192.168.3.124:42112	NULL	Connect
NULL	login	NULL		
759	unauthenticated user	192.168.3.50:4872	NULL	Connect
NULL	login	NULL		
761	unauthenticated user	192.168.3.40:36363	NULL	Connect
NULL	login	NULL		
762	neo	www.example.com:56200	NULL	Query
0	NULL	SHOW FULL PROCESSLIST		
+-----+-----+-----+-----+-----+				
+-----+-----+-----+-----+-----+				
44 rows in set (0.00 sec)				


```
mysql> SHOW FULL PROCESSLIST;
```

Id	User	Host	db	Command
718	unauthenticated user	192.168.3.124:42075	NULL	Connect
NULL	login	NULL		
719	unauthenticated user	192.168.3.124:42073	NULL	Connect
NULL	login	NULL		
720	unauthenticated user	192.168.3.124:42074	NULL	Connect
NULL	login	NULL		
721	unauthenticated user	192.168.3.124:42077	NULL	Connect
NULL	login	NULL		
722	unauthenticated user	192.168.3.124:42076	NULL	Connect
NULL	login	NULL		
723	unauthenticated user	192.168.3.124:42079	NULL	Connect
NULL	login	NULL		
724	unauthenticated user	192.168.3.124:42078	NULL	Connect
NULL	login	NULL		
725	unauthenticated user	192.168.3.124:42081	NULL	Connect
NULL	login	NULL		
726	unauthenticated user	192.168.3.124:42080	NULL	Connect
NULL	login	NULL		
727	unauthenticated user	192.168.3.124:42082	NULL	Connect
NULL	login	NULL		
728	unauthenticated user	192.168.3.124:42083	NULL	Connect
NULL	login	NULL		
729	unauthenticated user	192.168.3.124:42085	NULL	Connect
NULL	login	NULL		
730	unauthenticated user	192.168.3.124:42084	NULL	Connect
NULL	login	NULL		
731	unauthenticated user	192.168.3.124:42086	NULL	Connect
NULL	login	NULL		
732	unauthenticated user	192.168.3.124:42087	NULL	Connect
NULL	login	NULL		
733	unauthenticated user	192.168.3.124:42088	NULL	Connect
NULL	login	NULL		
734	unauthenticated user	192.168.3.124:42089	NULL	Connect
NULL	login	NULL		
735	unauthenticated user	192.168.3.124:42090	NULL	Connect
NULL	login	NULL		
736	unauthenticated user	192.168.3.124:42091	NULL	Connect
NULL	login	NULL		
737	unauthenticated user	192.168.3.124:42092	NULL	Connect
NULL	login	NULL		
738	unauthenticated user	192.168.3.124:42093	NULL	Connect
NULL	login	NULL		
739	unauthenticated user	192.168.3.124:42094	NULL	Connect

```

NULL | login | NULL |
| 740 | unauthenticated user | 192.168.3.124:42095 | NULL | Connect |
NULL | login | NULL |
| 741 | unauthenticated user | 192.168.3.124:42096 | NULL | Connect |
NULL | login | NULL |
| 742 | unauthenticated user | 192.168.3.124:42097 | NULL | Connect |
NULL | login | NULL |
| 743 | unauthenticated user | 192.168.3.124:42098 | NULL | Connect |
NULL | login | NULL |
| 744 | unauthenticated user | 192.168.3.124:42099 | NULL | Connect |
NULL | login | NULL |
| 745 | unauthenticated user | 192.168.3.124:42100 | NULL | Connect |
NULL | login | NULL |
| 746 | unauthenticated user | 192.168.3.124:42101 | NULL | Connect |
NULL | login | NULL |
| 747 | unauthenticated user | 192.168.3.124:42102 | NULL | Connect |
NULL | login | NULL |
| 748 | unauthenticated user | 192.168.3.124:42103 | NULL | Connect |
NULL | login | NULL |
| 749 | unauthenticated user | 192.168.3.124:42104 | NULL | Connect |
NULL | login | NULL |
| 750 | unauthenticated user | 192.168.3.124:42068 | NULL | Connect |
NULL | login | NULL |
| 751 | unauthenticated user | 192.168.3.124:42064 | NULL | Connect |
NULL | login | NULL |
| 752 | unauthenticated user | 192.168.3.124:42071 | NULL | Connect |
NULL | login | NULL |
| 753 | unauthenticated user | 192.168.3.124:42072 | NULL | Connect |
NULL | login | NULL |
| 754 | unauthenticated user | 192.168.3.124:42067 | NULL | Connect |
NULL | login | NULL |
| 755 | unauthenticated user | 192.168.3.124:42070 | NULL | Connect |
NULL | login | NULL |
| 756 | unauthenticated user | 192.168.3.124:42069 | NULL | Connect |
NULL | login | NULL |
| 757 | unauthenticated user | 192.168.3.124:42065 | NULL | Connect |
NULL | login | NULL |
| 758 | unauthenticated user | 192.168.3.124:42112 | NULL | Connect |
NULL | login | NULL |
| 759 | unauthenticated user | 192.168.3.50:4872 | NULL | Connect |
NULL | login | NULL |
| 761 | unauthenticated user | 192.168.3.40:36363 | NULL | Connect |
NULL | login | NULL |
| 762 | neo | www.example.com:56200 | NULL | Query |
0 | NULL | SHOW FULL PROCESSLIST |
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+
44 rows in set (0.00 sec)

```

解决方案 my.cnf 配置文件中加入skip-name-resolve

7.8. timeout

```
[mysqld]
wait_timeout=30
interactive_timeout=30
```

如果你没有修改过MySQL的配置，缺省情况下，wait_timeout的初始值是28800。

wait_timeout过大有弊端，其体现就是MySQL里大量的SLEEP进程无法及时释放，拖累系统性能，不过也不能把这个指设置的过小，否则你可能会遭遇到“MySQL has gone away”之类的问题，通常来说，我觉得把wait_timeout设置为10是个不错的选择，但某些情况下可能也会出问题，比如说有一个CRON脚本，其中两次SQL查询的间隔时间大于10秒的话，那么这个设置就有问题了：

(1)interactive_timeout 参数含义：服务器关闭交互式连接前等待活动的秒数。参数默认值：28800秒（8小时）

(2)wait_timeout 参数含义：服务器关闭非交互连接之前等待活动的秒数。

7.9. 与复制有关的参数

用于主库的选项 **Master**

定义 log-bin 文件名

```
log-bin=mysql-bin
```

binlog 保留时间, 过期天数设置

```
expire-logs-days = 30
```

```
binlog-do-db=需要复制的数据库名  
binlog-ignore-db=不需要复制的数据库
```

用于从库的选项 **Slave**

```
replicate-do-db= 指定需要复制的数据库  
replicate-ignore-db= 忽略复制的数据库
```

逃过错误

主从复制经常遇到 Last_Errno: 1062 可以使用下面配置跳过

```
slave_skip_errors=1062
```

7.10. 与 InnoDB 有关的配置项

```
innodb_file_per_table
```

配置后重启mysql运行下面命令将ibdata1拆分到tbl_name.ibd

```
OPTIMIZE TABLE tbl_name;
```

ls /var/lib/mysql/中查看 tbl_name.ibd文件

临时开启

```
SET @@global.innodb_file_per_table = 1;
```

7.11. EVENT 设置

开启EVENT定时任务

```
event_scheduler=on
```

7.12. 日志

操作日志

```
log = mysql.log
```

慢查询日志

```
log-slow-queries = slow.log  
long_query_time = 5
```

错误日志

```
[mysqld_safe]  
log-error=/var/log/mysqld.log
```

7.13. MySQL 5.7 my.cnf 实例

例 19.2. my.cnf

```
[root@netkiller ~]# cat /etc/my.cnf  
# For advice on how to change settings please see  
# http://dev.mysql.com/doc/refman/5.7/en/server-configuration-  
defaults.html  
  
[mysqld]  
#
```

```

# Remove leading # and set to the amount of RAM for the most important
data
# cache in MySQL. Start at 70% of total RAM for dedicated server, else
10%.
# innodb_buffer_pool_size = 128M
#
# Remove leading # to turn on a very important data integrity option:
logging
# changes to the binary log between backups.
# log_bin
#
# Remove leading # to set options mainly useful for reporting servers.
# The server defaults are faster for transactions and fast SELECTs.
# Adjust sizes as needed, experiment to find the optimal values.
# join_buffer_size = 128M
# sort_buffer_size = 2M
# read_rnd_buffer_size = 2M
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock

# Disabling symbolic-links is recommended to prevent assorted security
risks
symbolic-links=0

log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid

!includedir /etc/my.cnf.d

```

例 19.3. my.cnf

```

[root@netkiller ~]# cat /etc/my.cnf.d/default.cnf
[mysqld]
skip-name-resolve
max_connections=4096
default-storage-engine=INNODB

#wait_timeout=300
#interactive_timeout=300

character-set-server=utf8
collation_server=utf8_general_ci
init_connect='SET NAMES utf8'

explicit_defaults_for_timestamp=true

```

```

query_cache_type=1
query_cache_size=512M
table-open-cache=2000

#validate-password=OFF

sql_mode =
STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_Z
ERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION

[client]
default-character-set=utf8
#character_set_client=utf8

```

7.14. Example for my.cnf

例 19.4. my.cnf

```

# Example MySQL config file for very large systems.
#
# This is for a large system with memory of 1G-2G where the system runs
mainly
# MySQL.
#
# In this file, you can use all long options that a program supports.
# If you want to know which options a program supports, run the program
# with the "--help" option.

# The following options will be passed to all MySQL clients
[client]
#password          = your_password
port               = 3306
socket             = /var/lib/mysql/mysql.sock

# Here follows entries for some specific programs
character-set-server=utf8

# The MySQL server
[mysqld]
port               = 3306
socket             = /var/lib/mysql/mysql.sock
skip-external-locking
key_buffer_size   = 384M
max_allowed_packet = 1M
table_open_cache  = 512
sort_buffer_size  = 2M
read_buffer_size  = 2M

```

```
read_rnd_buffer_size = 8M
myisam_sort_buffer_size = 64M
thread_cache_size = 8
query_cache_size = 32M
# Try number of CPU's*2 for thread_concurrency
thread_concurrency = 8

#skip-networking

# Replication Master Server (default)
# binary logging is required for replication
log-bin=mysql-bin

# required unique id between 1 and 2^32 - 1
# defaults to 1 if master-host is not set
# but will not function as a master if omitted
server-id      = 1

# Replication Slave (comment out master section to use this)
#
#
# required unique id between 2 and 2^32 - 1
# (and different from the master)
# defaults to 2 if master-host is set
# but will not function as a slave if omitted
#server-id     = 2
#
# The replication master for this slave - required
#master-host   = <hostname>
#
# The username the slave will use for authentication when connecting
# to the master - required
#master-user   = <username>
#
# The password the slave will authenticate with when connecting to
# the master - required
#master-password = <password>
#
# The port the master is listening on.
# optional - defaults to 3306
#master-port   = <port>
#
# binary logging - not required for slaves, but recommended
#log-bin=mysql-bin
#
# binary logging format - mixed recommended
#binlog_format=mixed

# Uncomment the following if you are using InnoDB tables
#innodb_data_home_dir = /var/lib/mysql
#innodb_data_file_path = ibdata1:2000M;ibdata2:10M:autoextend
```



```
#innodb_log_group_home_dir = /var/lib/mysql
# You can set .._buffer_pool_size up to 50 - 80 %
# of RAM but beware of setting memory usage too high
#innodb_buffer_pool_size = 384M
#innodb_additional_mem_pool_size = 20M
# Set .._log_file_size to 25 % of buffer pool size
#innodb_log_file_size = 100M
#innodb_log_buffer_size = 8M
#innodb_flush_log_at_trx_commit = 1
#innodb_lock_wait_timeout = 50

# Here follows entries for some specific programs
skip-name-resolve
default-storage-engine = INNODB

character-set-server=utf8
collation_server=utf8_general_ci
init_connect='SET NAMES utf8'

max_connections          = 4096
max_connect_errors       = 10

pid-file                 = mysql.pid
log                      = mysql.log
log-error                 = mysql_error.log

log-slow-queries         = slow.log
long_query_time          = 10

[mysqldump]
quick
max_allowed_packet = 16M

[mysql]
no-auto-rehash
# Remove the next comment character if you are not familiar with SQL
#safe-updates

[myisamchk]
key_buffer_size = 256M
sort_buffer_size = 256M
read_buffer = 2M
write_buffer = 2M

[mysqlhotcopy]
interactive-timeout
```

8. variables

```
show variables;  
show global variables;
```

8.1. time_zone

```
SELECT @@global.time_zone, @@session.time_zone;
```

8.2. sql_mode

设置 sql_mode

```
SET GLOBAL sql_mode = 'NO_ENGINE_SUBSTITUTION';  
SET SESSION sql_mode = 'NO_ENGINE_SUBSTITUTION';
```

查看 sql_mode

```
SELECT @@GLOBAL.sql_mode;  
SELECT @@SESSION.sql_mode;
```

兼容早期 MySQL 版本

导入数据库遇到这样的问题

```
[root@netkiller]/tmp# cat cms.sql | mysql -uroot -p cms
```

ERROR 1067 (42000) at line 2194: Invalid default value for 'created_date'

将下面代码加入到 cms.sql 头部可以解决

```
set
@@global.sql_mode='NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION';
```

5.7.16

```
mysql> select version();
+-----+
| version() |
+-----+
| 5.7.16    |
+-----+
1 row in set (0.00 sec)

mysql> SELECT @@global.sql_mode;
+-----+
| @@global.sql_mode
|
+-----+
|
| ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,
| ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION
|
+-----+
1 row in set (0.00 sec)
```

8.3. wait_timeout

```
mysql> show global variables like 'wait_timeout';
```

Variable_name	Value
wait_timeout	10

```
mysql> use mysql;
```

```
Database changed
```

```
mysql> set wait_timeout=10;
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> show variables like '%wait_timeout%';
```

Variable_name	Value
innodb_lock_wait_timeout	50
table_lock_wait_timeout	50
wait_timeout	10

```
3 rows in set (0.00 sec)
```

8.4. table_lock_wait_timeout

```
mysql> set GLOBAL table_lock_wait_timeout=10;
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> show variables like '%table_lock_wait_timeout%';
```

Variable_name	Value
table_lock_wait_timeout	10

```
1 row in set (0.00 sec)
```

8.5. low_priority_updates

```
mysql> use mysql
Database changed

mysql> SET LOW_PRIORITY_UPDATES=1;
Query OK, 0 rows affected (0.00 sec)

mysql> show variables like '%priority%';
+-----+-----+
| Variable_name          | Value |
+-----+-----+
| low_priority_updates   | ON    |
| sql_low_priority_updates | ON    |
+-----+-----+
2 rows in set (0.00 sec)
```

8.6. collation_server

```
mysql> SHOW VARIABLES LIKE 'collation_server';
+-----+-----+
| Variable_name          | Value          |
+-----+-----+
| collation_server       | latin1_swedish_ci |
+-----+-----+
1 row in set (0.01 sec)

mysql>
```

8.7. character_set

```
mysql> show variables like 'character%';
```

Variable_name	Value
character_set_client	utf8
character_set_connection	utf8
character_set_database	utf8
character_set_filesystem	binary
character_set_results	utf8
character_set_server	utf8
character_set_system	utf8
character_sets_dir	/usr/share/mysql/charsets/

8 rows in set (0.00 sec)

链接 MySQL 指定字符集

```
mysql -uroot -h 192.168.0.10 -p --default-character-set=latin1
```

8.8. datadir

```
SHOW VARIABLES LIKE 'datadir';
```

```
mysql> SHOW VARIABLES LIKE 'datadir';
```

Variable_name	Value
datadir	/var/lib/mysql/

1 row in set (0.00 sec)

8.9. plugin_dir

show variables like '%plugin_dir%';

```
mysql> show variables like '%plugin_dir%';
+-----+-----+
| Variable_name | Value                |
+-----+-----+
| plugin_dir    | /usr/lib/mysql/plugin/ |
+-----+-----+
1 row in set (0.00 sec)
```

8.10. storage_engine

```
mysql> show variables like '%storage_engine%';
+-----+-----+
| Variable_name          | Value |
+-----+-----+
| default_storage_engine | InnoDB |
| storage_engine         | InnoDB |
+-----+-----+
2 rows in set (0.00 sec)
```

8.11. timeout

```
show variables like "%timeout%";
```

8.12. max_connections

```
show variables like "max_connections";
```

```
set global max_connections = 200;
```

8.13. 自动提交 autocommit

```
select @@autocommit;  
show variables like "autocommit";  
set autocommit='off'
```


9. Monitoring

<http://netkiller.sourceforge.net/monitoring/index.html>

9.1. Analysis and Optimization

mytop - top like query monitor for MySQL

```
sudo apt-get install mytop
```

```
mytop --host=172.16.0.7 --user=monitor --password=your_passwd
```

mtop - MySQL terminal based query monitor

<http://mtop.sourceforge.net/>

```
sudo apt-get install mtop  
mtop --host=172.16.0.6 --dbuser=monitor --password=your_passwd
```

mkill

```
mkill -sl 180 -fi 'select.*from bad_table' > /var/log/mkill.out  
2> /var/log/mkill.kill
```

innotop

mysqlreport - A friendly report of important MySQL status values

```
# yum install mysqlreport -y
```

```
wget hackmysql.com/scripts/mysqlreport
```

```
[root@database ~]# mysqlreport --user root --password chen
Use of uninitialized value in multiplication (*) at
/usr/bin/mysqlreport line 829.
Use of uninitialized value in formline at /usr/bin/mysqlreport
line 1227.
MySQL 5.0.77-log          uptime 28 23:42:33      Sat Apr 10
18:15:44 2010
```

__ Key

Buffer used	6.54M of	8.00M	%Used:	81.75
Current	1.49M		%Usage:	18.58
Write hit	97.65%			
Read hit	99.81%			

__ Questions

Total	2.22M	0.9/s		
DMS	1.91M	0.8/s	%Total:	86.16
Com_	249.93k	0.1/s		11.25
COM_QUIT	63.68k	0.0/s		2.87
-Unknown	6.26k	0.0/s		0.28
Slow 10 s	52	0.0/s		0.00 %DMS: 0.00
Log: OFF				
DMS	1.91M	0.8/s		86.16
SELECT	1.17M	0.5/s		52.81 61.29
INSERT	276.13k	0.1/s		12.43 14.43
DELETE	264.78k	0.1/s		11.92 13.84
UPDATE	158.14k	0.1/s		7.12 8.26
REPLACE	41.74k	0.0/s		1.88 2.18
Com_	249.93k	0.1/s		11.25
set_option	89.09k	0.0/s		4.01
change_db	59.71k	0.0/s		2.69
show_create	28.57k	0.0/s		1.29

__ SELECT and Sort

Scan	161.33k	0.1/s	%SELECT:	13.76
Range	6.47k	0.0/s		0.55
Full join	1.56k	0.0/s		0.13
Range check	0	0/s		0.00
Full rng join	0	0/s		0.00
Sort scan	34.03k	0.0/s		
Sort range	21.98k	0.0/s		
Sort mrg pass	733	0.0/s		

__ Table Locks

Waited	56	0.0/s	%Total:	0.00
Immediate	2.15M	0.9/s		

__ Tables

Open	64 of 64	%Cache:	100.00
Opened	159.20k	0.1/s	

__ Connections

Max used	36 of 200	%Max:	18.00
Total	63.75k	0.0/s	

__ Created Temp

Disk table	32.80k	0.0/s		
Table	63.69k	0.0/s	Size:	32.0M
File	319	0.0/s		

__ Threads

Running	1 of 1			
Cached	0 of 0	%Hit:		0
Created	63.75k	0.0/s		
Slow	0	0/s		

__ Aborted

Clients	128	0.0/s		
Connects	130	0.0/s		

__ Bytes

Sent	23.89G	9.5k/s
Received	6.36G	2.5k/s

__ InnoDB Buffer Pool

Usage	8.00M of	8.00M	%Used:	100.00
Read hit	99.99%			
Pages				
Free	0		%Total:	0.00
Data	511		99.80	%Drty: 0.00
Misc	1		0.20	
Latched			0.00	
Reads	1.54M	0.6/s		
From file	135	0.0/s	0.01	
Ahead Rnd	4	0.0/s		
Ahead Sql	6	0.0/s		
Writes	868.00k	0.3/s		
Flushes	1.56k	0.0/s		
Wait Free	0	0/s		

__ InnoDB Lock

Waits	0	0/s
Current	0	
Time acquiring		
Total	0 ms	
Average	0 ms	
Max	0 ms	

__ InnoDB Data, Pages, Rows

Data		
Reads	194	0.0/s
Writes	628	0.0/s
fsync	323	0.0/s
Pending		
Reads	0	
Writes	0	
fsync	0	
Pages		
Created	534	0.0/s
Read	201	0.0/s

Written	1.56k	0.0/s
Rows		
Deleted	0	0/s
Inserted	423.82k	0.2/s
Read	1.27M	0.5/s
Updated	0	0/s

mysqltuner - MySQL configuration assistant

```
# mysqltuner

>> MySQLTuner 1.1.1 - Major Hayden <major@mhtx.net>
>> Bug reports, feature requests, and downloads at
http://mysqltuner.com/
>> Run with '--help' for additional options and output
filtering
[!!] Successfully authenticated with no password - SECURITY
RISK!

----- General Statistics -----
-----
[--] Skipped version check for MySQLTuner script
[OK] Currently running supported MySQL version 5.1.69
[OK] Operating on 64-bit architecture

----- Storage Engine Statistics -----
-----
[--] Status: -Archive -BDB -Federated +InnoDB -ISAM -NDBCluster
[!!] InnoDB is enabled but isn't being used
[OK] Total fragmented tables: 0

----- Security Recommendations -----
-----
[!!] User 'root' has no password set.
[!!] User 'root' has no password set.
[!!] User 'root' has no password set.
[!!] User '' has no password set.
[!!] User '' has no password set.
```

```
----- Performance Metrics -----  
-----  
[--] Up for: 18m 55s (42 q [0.037 qps], 7 conn, TX: 27K, RX:  
1K)  
[--] Reads / Writes: 100% / 0%  
[--] Total buffers: 34.0M global + 2.7M per thread (151 max  
threads)  
[OK] Maximum possible memory usage: 449.2M (45% of installed  
RAM)  
[OK] Slow queries: 0% (0/42)  
[OK] Highest usage of available connections: 0% (1/151)  
[OK] Key buffer size / total MyISAM indexes: 8.0M/89.0K  
[!!] Query cache is disabled  
[OK] Temporary tables created on disk: 0% (0 on disk / 4 total)  
[!!] Thread cache is disabled  
[OK] Table cache hit rate: 76% (23 open / 30 opened)  
[OK] Open file limit used: 4% (46/1K)  
[OK] Table locks acquired immediately: 100% (19 immediate / 19  
locks)  
  
----- Recommendations -----  
-----  
General recommendations:  
  Add skip-innodb to MySQL configuration to disable InnoDB  
  MySQL started within last 24 hours - recommendations may be  
inaccurate  
  Enable the slow query log to troubleshoot bad queries  
  Set thread_cache_size to 4 as a starting value  
Variables to adjust:  
  query_cache_size (>= 8M)  
  thread_cache_size (start at 4)
```

9.2. Munin

9.3. Cacti

9.4. Monitoring MySQL with SNMP

mysql-snmp - monitoring MySQL with SNMP



第 20 章 Client and Utility Programs

1. mysql - the MySQL command-line tool

1.1. ~/.my.cnf

```
# mysql_secure_installation config file
[mysql]

[mysqld]

[client]
user=root
password='chen'

[mysqldump]
quick

[mysqladmin]

[mysqlhotcopy]
```

1.2. 屏幕输出到文件

```
mysql>tee /home/neo/screen.txt
mysql>select * from member;
mysql>exit
```

1.3. 终端编码

```
mysql> show variables like 'char%';
+-----+-----+
```


Variable_name	Value
character_set_client	utf8
character_set_connection	utf8
character_set_database	utf8
character_set_filesystem	binary
character_set_results	utf8
character_set_server	utf8
character_set_system	utf8
character_sets_dir	/usr/share/mysql/charsets/

8 rows in set (0.00 sec)

设置终端编码 set names utf8;

```
mysql> select * from category;
+----+-----+-----+-----+-----+-----+
| id | name | description | status | parent_id | path |
+----+-----+-----+-----+-----+-----+
| 1 | ?? | ??????? | Y | NULL | 1/ |
| 4 | ??? | ??? | Y | 1 | 1/4 |
| 5 | ??? | NULL | Y | 4 | 1/4/5 |
| 6 | ??? | NULL | Y | 5 | 1/4/5/6 |
| 7 | ??? | NULL | Y | 6 | 1/4/5/6/7 |
+----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> set names utf8;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from category;
+----+-----+-----+-----+-----+-----+
| id | name | description | status | parent_id | path |
+----+-----+-----+-----+-----+-----+
| 1 | 中国 | 中华人民共和家 | Y | NULL | 1/ |
| 4 | 广东省 | 广东省 | Y | 1 | 1/4 |
| 5 | 深圳市 | NULL | Y | 4 | 1/4/5 |
```

```

| 6 | 宝安区 | NULL | Y | 5 |
1/4/5/6 |
| 7 | 龙华镇 | NULL | Y | 6 |
1/4/5/6/7 |
+-----+-----+-----+-----+-----+
+-----+
5 rows in set (0.00 sec)

```

1.4. Unix Socket

```
mysql -uroot -p -S /tmp/mysql.sock
```

1.5. 重定向巧用

```

echo "show databases;" | mysql -uroot -pneo

cat |mysql -uroot -pneo << EOF
show databases;
EOF

```

1.6. --sigint-ignore 忽略 Ctrl + C

使用该选项方式用户中途通过 Ctrl + C 推出,只能通过 quit 退出

```

$ mysql --sigint-ignore
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 71
Server version: 5.5.32-0ubuntu0.13.04.1 (Ubuntu)

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All
rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or

```

```
its  
affiliates. Other names may be trademarks of their respective  
owners.
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the current  
input statement.
```

```
mysql> quit  
Bye
```

1.7. mysql log

/etc/my.cnf 配置文件

在服务器上的/etc/my.cnf中的[client]加入
tee =/tmp/mysql_history.log即可。

查看log设置

```
show VARIABLES like '%log%';
```

命令行

```
mysql -uroot --tee=/tmp/mysql_history.log
```

```
mysql> tee mysql_history.log  
Logging to file 'mysql_history.log '
```

或者

```
mysql> \T mysql_history.log  
Logging to file 'mysql_history.log '
```

```
mysql> notee  
Outfile disabled.
```

或者

```
mysql> \t  
Outfile disabled.
```

2. mysqldump - a database backup program

2.1. 备份数据库并压缩文件

```
mysqldump -uroot -p dbname | gzip > dbname.backup
```

2.2. 备份数据库/表

```
mysqldump -uroot -p -d database  
mysqldump -uroot -p -d database table
```

2.3. 备份到文件

-r, --result-file=name 输出文件

```
mysqldump --default-character-set=utf8 -r mysql.sql -h localhost -u root  
-p yourdb
```

2.4. 备份数据库，无结构，只有数据

-t, --no-create-info Don't write table creation info.

```
mysqldump -uroot -p -t -d database
```

2.5. 使用完整的insert插入数据

-c, --complete-insert Use complete insert statements.

```
$ mysqldump -hlocalhost -uroot -t neo test

INSERT INTO `test` VALUES (98,'neo','chen'),(112,'jam','zheng'),
(113,'john','meng');

$ mysqldump -hlocalhost -uroot -c -t neo test
INSERT INTO `test` (`userid`,`username`,`password`) VALUES
(98,'neo','chen'),(112,'jam','zheng'),(113,'john','meng');
```

2.6. --extended-insert / --skip-extended-insert

--extended-insert 默认开启

```
INSERT INTO `test` VALUES (98,'neo','chen'),(112,'jam','zheng'),
(113,'john','meng');
```

每条记录使用一次insert

```
$ mysqldump -hlocalhost -uroot --skip-extended-insert -t neo test |more
INSERT INTO `test` VALUES (98,'neo','chen');
INSERT INTO `test` VALUES (111,'neo','chen');
INSERT INTO `test` VALUES (112,'jam','zheng');
INSERT INTO `test` VALUES (113,'john','meng');
```

2.7. --skip-lock-tables

mysqldump 时禁止锁表

使用 --skip-lock-tables 参数，不会影响正在备份的表SELECT操作。

2.8. --skip-add-locks

该参数mysqldump输出中包含下面

默认情况

```
LOCK TABLES `tbl_name` WRITE;
```

如果使用这个参数就不会输出 LOCK TABLE

2.9. --where

```
mysqldump -hlocalhost -umysql -ppasswd database table --where="id>128"
```

2.10. 注释信息--comments /--skip-comments

--comments附加注释信息，默认为打开。可以用--skip-comments取消

```
--  
-- Table structure for table `demo`  
--  
DROP TABLE IF EXISTS `demo`;  
  
...  
...  
  
-- Dump completed on 2014-02-13 13:31:05
```

使用 --skip-comments后

```
DROP TABLE IF EXISTS `demo`;  
...  
...
```

2.11. 不导出注释信息

--compact

```
mysqldump --default-character-set=utf8 -h localhost -u root -p -t  
yourdb yourtable --skip-extended-insert --compact
```

运行后导出纯净的 INSERT 语句

2.12. 字符集设置

```
mysqldump --default-character-set=utf8 -r mysql.sql -h 192.168.0.24 -u  
root -p yourdb
```


3. mysqladmin - client for administering a MySQL server

3.1. reload

```
mysqladmin --user=root --password reload
```

3.2. 更改密码

```
sudo mysqladmin -u root -p password '你的新密码'
```

3.3. status

每个10秒输出一次mysql的状态信息

```
mysqladmin -i 10 extended status
```

```
mysqladmin -h 172.16.0.1 -u monitor -ppasswd status
Uptime: 195824  Threads: 21  Questions: 57744081  Slow queries:
516230  Opens: 13202607  Flush tables: 1  Open tables: 160
Queries per second avg: 294.877
```

```
mysqladmin -h 172.16.0.1 -u monitor -ppasswd extended-status
+-----+-----+
| Variable_name | Value |
+-----+-----+
| Aborted_clients | 60336 |
| Aborted_connects | 4599 |
| Binlog_cache_disk_use | 36 |
| Binlog_cache_use | 100721 |
+-----+-----+
```

Bytes_received	17510873261
Bytes_sent	92890743568
Com_admin_commands	10026660
Com_assign_to_keycache	0
Com_alter_db	0
Com_alter_db_upgrade	0
Com_alter_event	0
Com_alter_function	0
Com_alter_procedure	0
Com_alter_server	0
Com_alter_table	418
Com_alter_tablespace	0
Com_analyze	0
Com_backup_table	0
Com_begin	0
Com_binlog	0
Com_call_procedure	0
Com_change_db	4440400
Com_change_master	1
Com_check	0
Com_checksum	0
Com_commit	30089
Com_create_db	1
Com_create_event	0
Com_create_function	0
Com_create_index	1
Com_create_procedure	0
Com_create_server	0
Com_create_table	211
Com_create_trigger	4
Com_create_udf	0
Com_create_user	0
Com_create_view	2
Com_dealloc_sql	0
Com_delete	36664
Com_delete_multi	0
Com_do	0
Com_drop_db	0
Com_drop_event	0
Com_drop_function	0
Com_drop_index	0
Com_drop_procedure	0
Com_drop_server	0
Com_drop_table	213
Com_drop_trigger	0

Com_drop_user	0
Com_drop_view	4
Com_empty_query	0
Com_execute_sql	0
Com_flush	9
Com_grant	6
Com_ha_close	0
Com_ha_open	0
Com_ha_read	0
Com_help	0
Com_insert	472260
Com_insert_select	0
Com_install_plugin	0
Com_kill	12
Com_load	0
Com_load_master_data	0
Com_load_master_table	0
Com_lock_tables	209
Com_optimize	0
Com_preload_keys	0
Com_prepare_sql	0
Com_purge	0
Com_purge_before_date	0
Com_release_savepoint	0
Com_rename_table	0
Com_rename_user	0
Com_repair	0
Com_replace	4612
Com_replace_select	0
Com_reset	0
Com_restore_table	0
Com_revoke	0
Com_revoke_all	0
Com_rollback	0
Com_rollback_to_savepoint	0
Com_savepoint	0
Com_select	20310686
Com_set_option	9089818
Com_show_authors	0
Com_show_binlog_events	0
Com_show_binlogs	0
Com_show_charsets	24
Com_show_collations	18214
Com_show_column_types	0
Com_show_contributors	0

Com_show_create_db	0
Com_show_create_event	0
Com_show_create_func	0
Com_show_create_proc	0
Com_show_create_table	0
Com_show_create_trigger	0
Com_show_databases	24
Com_show_engine_logs	0
Com_show_engine_mutex	0
Com_show_engine_status	0
Com_show_events	0
Com_show_errors	0
Com_show_fields	147160
Com_show_function_status	3
Com_show_grants	0
Com_show_keys	2
Com_show_master_status	1
Com_show_new_master	0
Com_show_open_tables	0
Com_show_plugins	0
Com_show_privileges	0
Com_show_procedure_status	3
Com_show_processlist	12483
Com_show_profile	0
Com_show_profiles	0
Com_show_slave_hosts	0
Com_show_slave_status	0
Com_show_status	1158
Com_show_storage_engines	0
Com_show_table_status	2
Com_show_tables	29915
Com_show_triggers	0
Com_show_variables	26295
Com_show_warnings	0
Com_slave_start	0
Com_slave_stop	0
Com_stmt_close	0
Com_stmt_execute	0
Com_stmt_fetch	0
Com_stmt_prepare	0
Com_stmt_reprepare	0
Com_stmt_reset	0
Com_stmt_send_long_data	0
Com_truncate	0
Com_uninstall_plugin	0

Com_unlock_tables	209
Com_update	501411
Com_update_multi	23112
Com_xa_commit	0
Com_xa_end	0
Com_xa_prepare	0
Com_xa_recover	0
Com_xa_rollback	0
Com_xa_start	0
Compression	OFF
Connections	4555052
Created_tmp_disk_tables	421231
Created_tmp_files	1172
Created_tmp_tables	2769149
Delayed_errors	0
Delayed_insert_threads	0
Delayed_writes	0
Flush_commands	1
Handler_commit	100721
Handler_delete	133583
Handler_discover	0
Handler_prepare	0
Handler_read_first	404032
Handler_read_key	18292439681
Handler_read_next	33393351305
Handler_read_prev	77792315
Handler_read_rnd	2969739139
Handler_read_rnd_next	41965058450
Handler_rollback	0
Handler_savepoint	0
Handler_savepoint_rollback	0
Handler_update	4595750766
Handler_write	6069006380
Innodb_buffer_pool_pages_data	19
Innodb_buffer_pool_pages_dirty	0
Innodb_buffer_pool_pages_flushed	0
Innodb_buffer_pool_pages_free	493
Innodb_buffer_pool_pages_misc	0
Innodb_buffer_pool_pages_total	512
Innodb_buffer_pool_read_ahead_rnd	1
Innodb_buffer_pool_read_ahead_seq	0
Innodb_buffer_pool_read_requests	77
Innodb_buffer_pool_reads	12
Innodb_buffer_pool_wait_free	0
Innodb_buffer_pool_write_requests	0

Innodb_data_fsyncs	3
Innodb_data_pending_fsyncs	0
Innodb_data_pending_reads	0
Innodb_data_pending_writes	0
Innodb_data_read	2494464
Innodb_data_reads	25
Innodb_data_writes	3
Innodb_data_written	1536
Innodb_dblwr_pages_written	0
Innodb_dblwr_writes	0
Innodb_log_waits	0
Innodb_log_write_requests	0
Innodb_log_writes	1
Innodb_os_log_fsyncs	3
Innodb_os_log_pending_fsyncs	0
Innodb_os_log_pending_writes	0
Innodb_os_log_written	512
Innodb_page_size	16384
Innodb_pages_created	0
Innodb_pages_read	19
Innodb_pages_written	0
Innodb_row_lock_current_waits	0
Innodb_row_lock_time	0
Innodb_row_lock_time_avg	0
Innodb_row_lock_time_max	0
Innodb_row_lock_waits	0
Innodb_rows_deleted	0
Innodb_rows_inserted	0
Innodb_rows_read	0
Innodb_rows_updated	0
Key_blocks_not_flushed	0
Key_blocks_unused	6917
Key_blocks_used	53585
Key_read_requests	35870213140
Key_reads	13788784
Key_write_requests	35265303
Key_writes	2467239
Last_query_cost	0.000000
Max_used_connections	3001
Not_flushed_delayed_rows	0
Open_files	238
Open_streams	0
Open_table_definitions	228
Open_tables	160
Opened_files	20864567

Opened_table_definitions	653
Opened_tables	13202613
Prepared_stmt_count	0
Qcache_free_blocks	10480
Qcache_free_memory	38697120
Qcache_hits	17943956
Qcache_inserts	8251298
Qcache_lowmem_prunes	560647
Qcache_not_cached	11879434
Qcache_queries_in_cache	54611
Qcache_total_blocks	125193
Queries	57755205
Questions	57582352
Rpl_status	NULL
Select_full_join	602236
Select_full_range_join	6851
Select_range	1633467
Select_range_check	0
Select_scan	10981650
Slave_open_temp_tables	0
Slave_retried_transactions	0
Slave_running	OFF
Slow_launch_threads	206
Slow_queries	516237
Sort_merge_passes	548
Sort_range	293328
Sort_rows	2831414035
Sort_scan	2726547
Ssl_accept_renegotiates	0
Ssl_accepts	0
Ssl_callback_cache_hits	0
Ssl_cipher	
Ssl_cipher_list	
Ssl_client_connects	0
Ssl_connect_renegotiates	0
Ssl_ctx_verify_depth	0
Ssl_ctx_verify_mode	0
Ssl_default_timeout	0
Ssl_finished_accepts	0
Ssl_finished_connects	0
Ssl_session_cache_hits	0
Ssl_session_cache_misses	0
Ssl_session_cache_mode	NONE
Ssl_session_cache_overflows	0
Ssl_session_cache_size	0

Ssl_session_cache_timeouts	0	
Ssl_sessions_reused	0	
Ssl_used_session_cache_entries	0	
Ssl_verify_depth	0	
Ssl_verify_mode	0	
Ssl_version		
Table_locks_immediate	46406490	
Table_locks_waited	1428430	
Tc_log_max_pages_used	0	
Tc_log_page_size	0	
Tc_log_page_waits	0	
Threads_cached	33	
Threads_connected	33	
Threads_created	77809	
Threads_running	7	
Uptime	195854	
Uptime_since_flush_status	195854	

3.4. process list

```
[root@development ~]# mysqladmin -u root -p -h 127.0.0.1
processlist
Enter password:
+-----+-----+-----+-----+-----+-----+
| Id      | User      | Host                               | db      | Command |
Time | State | Info                               |         |         |
+-----+-----+-----+-----+-----+-----+
| 23648  | dbuser   | 192.168.3.237:1220                | testdb  | Sleep   |
2733 |      |                                     |         |         |
| 23878  | dbuser   | www.testdb.com:53639              | testdb  | Sleep   | 7
|      |         |                                     |         |         |
| 23881  | root     | localhost:57243                    |         | Query   | 0
|      | show processlist |                                     |         |         |
+-----+-----+-----+-----+-----+-----+
|      |         |                                     |         |         |
+-----+-----+-----+-----+-----+-----+
```



```
mysql -u root -pneo -S /tmp/mysql.sock -e "show full  
processlist;"|grep -v Sleep
```

4. myisamchk — MyISAM Table-Maintenance Utility

先停止mysqld，在--datadir目录运行

```
myisamchk */*.MYI >/dev/null # 检查哪些表需要修复
```

修复用以下命令一个个修复：

```
myisamchk -r table.MYI
```

更省事的做法：

```
myisamchk -r /var/lib/mysql/*.MYI
```

myisamchk可用crontab定时最佳化table

```
0 * * 0 /usr/bin/myisamchk -s /var/lib/mysql/*/*.MYI
```

5. mysqlcheck — A Table Maintenance and Repair Program

即可最佳化所有db

```
mysqlcheck -a -c -o -r --all-databases -uroot -p
-a = Analyse given tables.
-c = Check table for errors
-o = Optimise table
-r = Can fix almost anything except unique keys that aren't
unique
```

```
mysqlcheck -A -o -r -p
```

6. mysqlslap - load emulation client

`--auto-generate-sql, -a`
自动生成测试表和数据

`--auto-generate-sql-load-type=type`
测试语句的类型。取值包括: read, key, write, update和mixed(默认)。

`--number-char-cols=N, -x N`
自动生成的测试表中包含多少个字符类型的列, 默认1

`--number-int-cols=N, -y N`
自动生成的测试表中包含多少个数字类型的列, 默认1

`--number-of-queries=N`
总的测试查询次数(并发客户数×每客户查询次数)

`--query=name, -q`
使用自定义脚本执行测试, 例如可以调用自定义的一个存储过程或者sql语句来执行测试。

`--create-schema`
测试的schema, MySQL中schema也就是database

`--commint=N`
多少条DML后提交一次

`--compress, -C`
如果服务器和客户端支持都压缩, 则压缩信息传递

`--concurrency=N, -c N`
并发量, 也就是模拟多少个客户端同时执行select。可指定多个值, 以逗号或者--delimiter参数指定的值做为分隔符

`--engine=engine_name, -e engine_name`
创建测试表所使用的存储引擎, 可指定多个

`--iterations=N, -i N`
测试执行的迭代次数

`--detach=N`
执行N条语句后断开重连

`--debug-info, -T`
打印内存和CPU的信息

`--only-print`
只打印测试语句而不实际执行

```
mysqlslap -u root -p -h localhost -c 10,50,100,200 -i 1 \  
--engine=myisam --auto-generate-sql-load-type=mixed --number-  
of-queries=50000 \  
--number-char-cols=5 --number-int-cols=5 --auto-generate-sql
```

```
mysqlslap --defaults-file=/etc/my.cnf --concurrency=50,100,200  
--iterations=1 \  
--number-int-cols=4 --number-char-cols=4 --auto-generate-sql --  
auto-generate-sql-add-autoincrement \  
--auto-generate-sql-load-type=mixed --engine=myisam,innodb --  
number-of-queries=200 --debug-info \  
-uroot -p -S/tmp/mysql.sock
```

7. mysqldumpslow - Parse and summarize the MySQL slow query log.

开启记录日志，修改my.cnf加入下面几行

```
--log-slow-queries[=file_name]
```

```
long_query_time = 10
log-slow-queries =
```

long_query_time 是指执行超过10秒的sql会被记录下来。

log-slow-queries设置把日志文件的位置，如果没有给出文件名值，默认未主机名，后缀为-slow.log。如果给出了文件名，但不是绝对路径名，文件则写入数据目录。

```
cat /etc/mysql/my.cnf
```

```
[mysqld]
set-variable=long_query_time=1
log-slow-queries=/var/log/mysql/log-slow-queries.log
```

You must create the file manually and change owners this way:

```
touch /var/log/mysql/log-slow-queries.log
chown mysql:mysql -R /var/log/mysql/log-slow-queries.log
```

```
$ mysqldumpslow /var/log/mysql/log-slow-queries.log
```

mysqldumpslow 参数

1. -s, 是order的顺序, 说明写的不够详细, 俺用下来, 包括看了代码, 主要有c,t,l,r和ac,at,al,ar, t=time, l=lock time, r=rows分别是按照query次数, 时间, lock的时间和返回的记录数来排序, 前面加了a的时倒叙
2. -t, 是top n的意思, 即为返回前面多少条的数据
3. -g, 后边可以写一个正则匹配模式, 大小写不敏感的
4. -g, 后边可以写一个正则匹配模式, 大小写不敏感的

```
mysqldumpslow -s c -t 20 ubuntu-slow.log
```

```
mysqldumpslow -s r -t 20 ubuntu-slow.log
```

8.

<https://www.mycli.net>

```
$ brew update && brew install mycli
```


9. MySQL慢查询日志 (Slow Query Log)

参数说明如下:

slow_query_log:	慢查询开启状态
slow_query_log_file:	慢查询日志存放的位置 (默认设置为 MySQL 的数据存放目录)
long_query_time:	查询超过多少秒才记录

9.1. MySQL 8.x

慢查询日志状态

```
mysql> show variables like '%slow_query_log%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| slow_query_log | ON    |
| slow_query_log_file | /var/lib/mysql/netkiller-slow.log |
+-----+-----+
2 rows in set (0.00 sec)
```

```
set global log_output='TABLE'; -- 开启慢日志,纪录到
mysql.slow_log 表
set global long_query_time=10; -- 设置超过10秒的查询为慢查询
set global slow_query_log='ON'; -- 打开慢日志记录
```

查询慢的 sql 日志

```
select convert(sql_text using utf8) sql_text from mysql.slow_log
```

关闭慢查询日志

```
set global slow_query_log='OFF'; -- 排查完毕后记得关闭日志
```

9.2. MySQL 5.x

查看设置

```
mysql> SHOW VARIABLES LIKE 'slow_query%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| slow_query_log | OFF |
| slow_query_log_file | /var/lib/mysql/netkiller-slow.log |
+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

```
mysql> SHOW VARIABLES LIKE 'long_query_time';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| long_query_time | 10.000000 |
+-----+-----+
1 row in set (0.01 sec)
```

开启慢查询日志

```
mysql> SET GLOBAL slow_query_log=ON;
Query OK, 0 rows affected (0.05 sec)

mysql> SET GLOBAL long_query_time=0.001;
Query OK, 0 rows affected (0.00 sec)
```

配置慢查询日志

```
[mysqld]
log-slow-queries          = /var/lib/mysql/netkiller-
slow.log
long_query_time          = 10
```

删除慢查询日志

执行命令后将清空旧慢查询日志，写入新日志

```
mysqladmin -uroot -p flush-logs
```

10. mysql-admin

```
$ sudo apt-get install mysql-admin
```

运行mysql-admin

```
/usr/bin/mysql-admin
```

运行 mysql-query-browser

```
mysql-query-browser --query="SELECT * FROM users"
```

第 21 章 数据库管理 (Database Administration)

1. 用户管理 (User Account Management)

1.1. 创建用户

```
CREATE USER user [IDENTIFIED BY [PASSWORD] 'password']  
    [, user [IDENTIFIED BY [PASSWORD] 'password']] ...
```

```
CREATE USER 'test'@'xxx.xxx.xxx.xxx' IDENTIFIED BY 'your_password';
```

```
CREATE USER 'root'@'192.168.1.%' IDENTIFIED BY 'password';
```

add a new user by grant

```
GRANT ALL PRIVILEGES ON opencart.* TO 'neo'@'localhost' IDENTIFIED BY  
'chen' WITH GRANT OPTION;  
  
GRANT ALL PRIVILEGES ON *.* TO 'neo'@'localhost' IDENTIFIED BY 'chen'  
WITH GRANT OPTION;  
  
FLUSH PRIVILEGES;
```

MySQL 8.0

```
mysql> CREATE USER 'root'@'%' IDENTIFIED WITH mysql_native_password BY  
'pMQiEgelikst7S_6tlXzB0mt_4b';  
Query OK, 0 rows affected (0.06 sec)  
  
mysql> grant all on *.* to 'root'@'%';
```

```
Query OK, 0 rows affected (0.11 sec)
```

1.2. 删除用户

```
DROP USER user [, user] ...
```

```
mysql> drop user 'root'@'%';  
Query OK, 0 rows affected (0.02 sec)  
  
mysql> drop user admin@'localhost';  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> drop user admin@'127.0.0.1';  
Query OK, 0 rows affected (0.00 sec)
```

1.3. 修改用户名

```
RENAME USER old_user TO new_user [, old_user TO new_user] ...
```

1.4. 修改密码

mysql 5.7 之前的版本

```
SET PASSWORD FOR 'bob'@'%.loc.gov' = PASSWORD('newpass');  
  
SET PASSWORD FOR 'root'@'%' =  
PASSWORD('co2uqAMAholaSOS62146Xoci6ogu4I');
```

MySQL 5.7 之后

```
ALTER USER 'root'@'localhost' IDENTIFIED BY 'your_password';
```

```
mysql> ALTER user 'root'@'%' IDENTIFIED BY 'test';  
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> FLUSH PRIVILEGES;  
Query OK, 0 rows affected (0.00 sec)
```

2. Access Privilege System

```

global privileges
OR (database privileges AND host privileges)
OR table privileges
OR column privileges
OR routine privileges

```

Table 12.1. Permissible Privileges for GRANT and REVOKE

Privilege	Meaning
ALL [PRIVILEGES]	Grant all privileges at specified access level except GRANT OPTION
ALTER	Enable use of ALTER TABLE
ALTER ROUTINE	Enable stored routines to be altered or dropped
CREATE	Enable database and table creation
CREATE ROUTINE	Enable stored routine creation
CREATE TABLESPACE	Enable tablespaces and log file groups to be created, altered, or dropped
CREATE TEMPORARY TABLES	Enable use of CREATE TEMPORARY TABLE
CREATE USER	Enable use of CREATE USER, DROP USER, RENAME USER, and REVOKE ALL PRIVILEGES
CREATE VIEW	Enable views to be created or altered
DELETE	Enable use of DELETE
DROP	Enable databases, tables, and views to be dropped
EVENT	Enable use of events for the Event Scheduler
EXECUTE	Enable the user to execute stored routines
FILE	Enable the user to cause the server to read or write files
GRANT OPTION	Enable privileges to be granted to or removed from other accounts
INDEX	Enable indexes to be created or dropped
INSERT	Enable use of INSERT
LOCK TABLES	Enable use of LOCK TABLES on tables for which you have the SELECT privilege
PROCESS	Enable the user to see all processes with SHOW PROCESSLIST
PROXY	Enable user proxying
REFERENCES	Not implemented
RELOAD	Enable use of FLUSH operations
REPLICATION CLIENT	Enable the user to ask where master or slave servers are
REPLICATION SLAVE	Enable replication slaves to read binary log events from the master
SELECT	Enable use of SELECT
SHOW DATABASES	Enable SHOW DATABASES to show all databases
SHOW VIEW	Enable use of SHOW CREATE VIEW
SHUTDOWN	Enable use of mysqladmin shutdown
SUPER	Enable use of other administrative operations such as CHANGE MASTER TO, KILL, PURGE BINARY LOGS, SET GLOBAL, and mysqladmin debug command
TRIGGER	Enable trigger operations
UPDATE	Enable use of UPDATE


```
USAGE    Synonym for "no privileges"
```

<http://dev.mysql.com/doc/refman/5.5/en/grant.html#grant-table-privileges>

REPLICATION CLIENT 与 REPLICATION SLAVE区别，前者只能使用SHOW MASTER STATUS和SHOW SLAVE STATUS命令监控复制状态，后者才能从主库复制binlog.

2.1. SHOW GRANTS

```
mysql> SHOW GRANTS FOR 'root'@'localhost';
+-----+
| Grants for root@localhost |
+-----+
| GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' WITH GRANT OPTION |
+-----+
1 row in set (0.00 sec)
```

```
mysql> show grants;
+-----+
--+
| Grants for root@localhost |
+-----+
--+
| GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' WITH GRANT OPTION |
+-----+
--+
1 row in set (0.00 sec)
```

2.2. show privileges

```
mysql> show privileges;
+-----+-----+-----+
| Privilege | Context | Comment |
+-----+-----+-----+
| Alter | Tables | To alter the table |
| Alter routine | Functions,Procedures | To alter or drop stored functions/procedures |
| Create | Databases,Tables,Indexes | To create |
```

new databases and tables			
Create routine	Databases		To use
CREATE FUNCTION/PROCEDURE			
Create temporary tables	Databases		To use
CREATE TEMPORARY TABLE			
Create view	Tables		To create
new views			
Create user	Server Admin		To create
new users			
Delete	Tables		To delete
existing rows			
Drop	Databases, Tables		To drop
databases, tables, and views			
Event	Server Admin		To create,
alter, drop and execute events			
Execute	Functions, Procedures		To execute
stored routines			
File	File access on server		To read and
write files on the server			
Grant option	Databases, Tables, Functions, Procedures		To give to
other users those privileges you possess			
Index	Tables		To create or
drop indexes			
Insert	Tables		To insert
data into tables			
Lock tables	Databases		To use LOCK
TABLES (together with SELECT privilege)			
Process	Server Admin		To view the
plain text of currently executing queries			
Proxy	Server Admin		To make
proxy user possible			
References	Databases, Tables		To have
references on tables			
Reload	Server Admin		To reload or
refresh tables, logs and privileges			
Replication client	Server Admin		To ask where
the slave or master servers are			
Replication slave	Server Admin		To read
binary log events from the master			
Select	Tables		To retrieve
rows from table			
Show databases	Server Admin		To see all
databases with SHOW DATABASES			
Show view	Tables		To see views
with SHOW CREATE VIEW			
Shutdown	Server Admin		To shut down
the server			
Super	Server Admin		To use KILL
thread, SET GLOBAL, CHANGE MASTER, etc.			
Trigger	Tables		To use
triggers			
Create tablespace	Server Admin		To
create/alter/drop tablespaces			
Update	Tables		To update
existing rows			
Usage	Server Admin		No
privileges - allow connect only			

```
+-----+-----+-----+
+-----+
31 rows in set (0.00 sec)
```

2.3. Grant privileges

Global privileges

```
GRANT ALL ON *.* TO 'someuser'@'somehost';
GRANT SELECT, INSERT ON *.* TO 'someuser'@'somehost';
```

Database privileges

```
GRANT ALL ON mydb.* TO 'someuser'@'somehost';
GRANT SELECT, INSERT ON mydb.* TO 'someuser'@'somehost';
```

Table privileges

```
GRANT ALL ON mydb.mytbl TO 'someuser'@'somehost';
GRANT SELECT, INSERT ON mydb.mytbl TO 'someuser'@'somehost';
```

Column privileges

```
GRANT SELECT (col1), INSERT (col1,col2) ON mydb.mytbl TO 'someuser'@'somehost';
```

Routine privileges

```
GRANT CREATE ROUTINE ON mydb.* TO 'someuser'@'somehost';
GRANT EXECUTE ON PROCEDURE mydb.myproc TO 'someuser'@'somehost';
```

2.4. Revoke privileges

```
REVOKE
  priv_type [(column_list)]
```

```
[, priv_type [(column_list)]] ...  
ON [object_type] priv_level  
FROM user [, user] ...
```

```
REVOKE ALL PRIVILEGES, GRANT OPTION  
FROM user [, user] ...
```

2.5. Show Privileges

```
mysql> select * from user where user = 'neo'\G  
***** 1. row *****  
      Host: 192.168.0.5  
      User: neo  
      Password: *7564B7B0A062C9523700601CBA1DCE1F861D6270  
      Select_priv: Y  
      Insert_priv: Y  
      Update_priv: Y  
      Delete_priv: Y  
      Create_priv: Y  
      Drop_priv: Y  
      Reload_priv: Y  
      Shutdown_priv: Y  
      Process_priv: Y  
      File_priv: Y  
      Grant_priv: N  
      References_priv: Y  
      Index_priv: Y  
      Alter_priv: Y  
      Show_db_priv: Y  
      Super_priv: Y  
      Create_tmp_table_priv: Y  
      Lock_tables_priv: Y  
      Execute_priv: Y  
      Repl_slave_priv: Y  
      Repl_client_priv: Y  
      Create_view_priv: Y  
      Show_view_priv: Y  
      Create_routine_priv: Y  
      Alter_routine_priv: Y  
      Create_user_priv: Y  
      Event_priv: Y  
      Trigger_priv: Y  
      ssl_type:  
      ssl_cipher:  
      x509_issuer:  
      x509_subject:  
      max_questions: 0  
      max_updates: 0  
      max_connections: 0  
      max_user_connections: 0  
1 row in set (0.00 sec)  
  
mysql>
```

2.6. MAX_QUERIES_PER_HOUR/MAX_UPDATES_PER_HOUR

```
GRANT USAGE ON *.* TO ...  
  WITH MAX_QUERIES_PER_HOUR 500 MAX_UPDATES_PER_HOUR 100;
```

2.7. Table Privileges

授权tmp用户只能访问tablename表

```
GRANT ALL PRIVILEGES ON tmp.tabname TO 'tmp'@'%' IDENTIFIED BY 'chen' WITH GRANT  
OPTION;
```

如果用户已经存在仅仅是分配权限可以使用下面方法

```
GRANT ALL ON mydb.mytbl TO 'someuser'@'somehost';  
GRANT SELECT, INSERT ON mydb.mytbl TO 'someuser'@'somehost';
```

2.8. Column Privileges

mydb.mytbl 表 col1字段允许查询，col1,col2允许插入

```
GRANT SELECT (col1), INSERT (col1,col2) ON mydb.mytbl TO 'someuser'@'somehost';
```

3. 字符集转换

找出指定字符集的表

```
select TABLE_SCHEMA, TABLE_NAME, TABLE_COLLATION from
information_schema.tables where table_collation =
'utf8mb4_0900_ai_ci' and table_schema = 'your_schema';
```

```
SELECT
    CONCAT(
        'ALTER TABLE ',
        TABLE_NAME,
        ' CONVERT TO CHARACTER SET utf8mb4 COLLATE
utf8mb4_general_ci;'
    )
FROM
    information_schema.`TABLES`
WHERE
    TABLE_SCHEMA = 'DATABASE_NAME';
```

3.1. 转换 latin1 到 UTF-8

```
UPDATE category SET
name=convert(cast(convert(name using latin1) as binary) using
utf8),
description=convert(cast(convert(description using latin1) as
binary) using utf8)
```

4. 重新整理AUTO_INCREMENT字段

AUTO_INCREMENT 并非按照我们意愿，顺序排列，经常会跳过一些数字，例如当插入失败的时候，再次插入会使用新的值。有时会造成浪费，我们可以使用下面SQL重新编排AUTO_INCREMENT序列。

```
SET @newid=0;
UPDATE mytable SET id = (SELECT @newid:=@newid+ 1);
```

使用max()查看最大值，然后使用 alter修改起始位置。

```
select max(id) from mytable;
ALTER TABLE mytable AUTO_INCREMENT = 1000;
```

注意外键，需要 ON UPDATE CASCADE 支持，否则无法更新。
CONSTRAINT `FK_group_has_contact_contact` FOREIGN KEY
(`contact_id`) REFERENCES `contact` (`id`) ON UPDATE CASCADE
ON DELETE CASCADE,

```
CREATE TABLE `contact` (  
  `id` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT COMMENT  
'唯一ID',  
  `name` VARCHAR(50) NOT NULL COMMENT '姓名',  
  `mobile` VARBINARY(32) NULL DEFAULT NULL COMMENT '手机号  
码',  
  `email` VARBINARY(50) NULL DEFAULT NULL COMMENT '电子邮  
件',  
  `mobile_digest` VARCHAR(32) NULL DEFAULT NULL COMMENT  
'摘要',
```

```

        `email_digest` VARCHAR(32) NULL DEFAULT NULL COMMENT
'邮件摘要',
        `birthday` DATE NULL DEFAULT NULL COMMENT '生日',
        `description` VARCHAR(255) NULL DEFAULT NULL COMMENT
'备注描述',
        `status`
        ENUM('Subscription','Unsubscribe') NOT NULL DEFAULT
'Subscription' COMMENT '订阅状态',
        `ctime` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
COMMENT '创建时间',
        `mtime` TIMESTAMP NULL DEFAULT NULL ON UPDATE
CURRENT_TIMESTAMP COMMENT '修改时间',
        PRIMARY KEY (`id`),
        UNIQUE INDEX `digest` (`mobile_digest`, `email_digest`)
)
COMMENT='会员手机短信与电子邮件映射表'
COLLATE='utf8_general_ci'
ENGINE=InnoDB
AUTO_INCREMENT=43642;

CREATE TABLE `group` (
        `id` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,
        `name` VARCHAR(50) NOT NULL,
        `description` VARCHAR(512)
        NOT NULL,
        `ctime` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
        PRIMARY KEY (`id`),
        UNIQUE INDEX `name` (`name`)
)
COMMENT='短信分组'
COLLATE='utf8_general_ci'
ENGINE=InnoDB
AUTO_INCREMENT=8;

CREATE TABLE `group_has_contact` (
        `id` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,
        `group_id` INT(10) UNSIGNED NOT NULL,
        `contact_id` INT(10) UNSIGNED NOT NULL,
        `ctime` TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP,
        PRIMARY KEY (`id`),
        UNIQUE INDEX `group_contact` (`group_id`,
`contact_id`),
        INDEX `FK_group_has_contact_contact`
        (`contact_id`),
        CONSTRAINT `FK_group_has_contact_contact` FOREIGN KEY

```



```
(`contact_id`) REFERENCES `contact` (`id`) ON UPDATE CASCADE ON  
DELETE CASCADE,  
    CONSTRAINT `FK_group_has_contact_group` FOREIGN KEY  
(`group_id`) REFERENCES `group` (`id`) ON UPDATE CASCADE ON  
DELETE CASCADE  
)  
COMMENT='N:M'  
COLLATE='utf8_general_ci'  
ENGINE=InnoDB  
AUTO_INCREMENT=55764;
```

5. 数据库内容替换

```
#!/bin/bash
HOST='localhost'
USER='neo'
PASS='chen'

SDB='neo'
TDB='netkiller'
MYSQLDUMP="mysqldump"
MYSQLDUMPOPTS="-h${HOST} -u${USER} -p${PASS}"

MYSQL="mysql"
MYSQLOPTS="-h${HOST} -u${USER} -p${PASS}"
#SED="sed -e 's/netkiller\.8800\.org/netkiller\.sf\.net/g' -e
's/陈景峰/景峰/g' -e 's/Neo/Netkiller/g'"

$MYSQL $MYSQLOPTS <<SQL
DROP DATABASE $TDB;
CREATE DATABASE $TDB DEFAULT CHARACTER SET utf8 COLLATE
utf8_general_ci;
SQL

$MYSQLDUMP $MYSQLDUMPOPTS ${SDB} | sed -e
's/netkiller\.8800\.org/netkiller\.sf\.net/g' -e 's/陈景峰/景
峰/g' -e 's/Neo/Netkiller/g' | $MYSQL $MYSQLOPTS ${TDB}
#echo "$MYSQLDUMP $MYSQLDUMPOPTS ${SDB} | $SED | $MYSQL
$MYSQLOPTS ${TDB}"
```

6. Kill 脚本

查询出锁定的表

```
SELECT concat('KILL ',id,';') FROM information_schema.processlist
WHERE user='root';
```

```
SELECT concat('KILL ',id,';') FROM information_schema.processlist
WHERE command='Locked' and user='root';
```

```
SELECT concat('KILL ',id,';') FROM information_schema.processlist
WHERE command='Locked' and user='root' and db='test';
```

拼装kill命令后输入到kill.sql, source 将从kill.sql读取sql命令并执行。

```
SELECT concat('KILL ',id,';') FROM
information_schema.processlist WHERE user='root' INTO OUTFILE
'/tmp/kill.sql';
source /tmp/kill.sql;
```

```
mysqladmin -uroot -p processlist | grep Sleep |awk '{if
(length($2) > 1) print "Kill "$2}'|xargs mysqladmin -uroot kill
```

7. SHOW COMMAND

7.1. 查看版本

Server

```
mysql> select version();
+-----+
| version() |
+-----+
| 5.0.77 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> status;
-----
mysql Ver 14.12 Distrib 5.0.77, for redhat-linux-gnu (x86_64)
using readline 5.1

Connection id: 1533
Current database:
Current user: root@localhost
SSL: Not in use
Current pager: stdout
Using outfile: ''
Using delimiter: ;
Server version: 5.0.77 Source distribution
Protocol version: 10
Connection: Localhost via UNIX socket
Server characterset: latin1
Db characterset: latin1
Client characterset: latin1
Conn. characterset: latin1
UNIX socket: /var/lib/mysql/mysql.sock
Uptime: 1 day 21 hours 40 min 52 sec

Threads: 1 Questions: 22172 Slow queries: 0 Opens: 3130 Flush
tables: 1 Open tables: 64 Queries per second avg: 0.135
-----
```

Client

```
[root@development ~]# mysql -V
mysql Ver 14.12 Distrib 5.0.77, for redhat-linux-gnu (x86_64)
using readline 5.1
```

7.2. status

```
mysql> show status;
mysql> show global status;
```

show status

数据库性能状态

(1) QPS (每秒Query量)

$QPS = \text{Questions(or Queries)} / \text{seconds}$

```
mysql > show /*50000 global */ status like 'Question';
```

(2) TPS (每秒事务量)

$TPS = (\text{Com_commit} + \text{Com_rollback}) / \text{seconds}$

```
mysql > show status like 'Com_commit';
```

```
mysql > show status like 'Com_rollback';
```

(3) key Buffer 命中率

$\text{key_buffer_read_hits} = (1 - \text{key_reads} / \text{key_read_requests}) * 100\%$

$\text{key_buffer_write_hits} = (1 - \text{key_writes} / \text{key_write_requests}) * 100\%$

```
mysql> show status like 'Key%';
```

(4) InnoDB Buffer命中率

$\text{innodb_buffer_read_hits} = (1 - \text{innodb_buffer_pool_reads} /$

```

innodb_buffer_pool_read_requests) * 100%

mysql> show status like 'innodb_buffer_pool_read%';

(5)Query Cache命中率
Query_cache_hits = (Qcahce_hits / (Qcache_hits + Qcache_inserts
)) * 100%;

mysql> show status like 'Qcache%';
(6)Table Cache状态量
mysql> show status like 'open%';

(7)Thread Cache 命中率
Thread_cache_hits = (1 - Threads_created / connections ) * 100%

mysql> show status like 'Thread%';

mysql> show status like 'Connections';

(8)锁定状态
mysql> show status like '%lock%';

(9)复制延时量
mysql > show slave status

(10) Tmp Table 状况(临时表状况)
mysql > show status like 'Create_tmp%';
(11) Binlog Cache 使用状况
mysql > show status like 'Binlog_cache%';

(12) InnoDB_log_waits 量
mysql > show status like 'innodb_log_waits';

```

show master status

```

mysql> show master status;
+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB |
Binlog_Ignore_DB |
+-----+-----+-----+-----+

```

```
-----+
| DBMaster-bin.000018 | 409468882 | example      |
|
+-----+-----+-----+-----+
-----+
1 row in set (0.00 sec)

mysql>
```

show slave status

```
mysql> show slave status/G

得到的列表会有类似下面的数据：

File: mysql-bin.000001
Position: 1374
Binlog_Do_DB: test
Binlog_Ignore_DB: mysql

Slave_IO_State: Waiting for master to send event
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
```

show plugins

```
mysql> SHOW PLUGINS;
+-----+-----+-----+-----+-----+
| Name          | Status  | Type          | Library | License |
+-----+-----+-----+-----+-----+
| binlog        | ACTIVE  | STORAGE ENGINE | NULL    | GPL     |
| partition     | ACTIVE  | STORAGE ENGINE | NULL    | GPL     |
| ARCHIVE       | ACTIVE  | STORAGE ENGINE | NULL    | GPL     |
| BLACKHOLE     | ACTIVE  | STORAGE ENGINE | NULL    | GPL     |
| CSV           | ACTIVE  | STORAGE ENGINE | NULL    | GPL     |
```

FEDERATED	DISABLED	STORAGE ENGINE	NULL	GPL
MEMORY	ACTIVE	STORAGE ENGINE	NULL	GPL
InnoDB	ACTIVE	STORAGE ENGINE	NULL	GPL
MyISAM	ACTIVE	STORAGE ENGINE	NULL	GPL
MRG_MYISAM	ACTIVE	STORAGE ENGINE	NULL	GPL

10 rows in set (0.00 sec)

7.3. show processlist

```
show full processlist;
```

</screen>

<screen><![CDATA[

命令: show processlist;

如果是root帐号, 你能看到所有用户的当前连接。如果是其它普通帐号, 只能看到自己占用的连接。

show processlist;只列出前100条, 如果想全列出请使用show full processlist;

```
mysql> show processlist;
```

命令: show status;

Aborted_clients 由于客户没有正确关闭连接已经死掉, 已经放弃的连接数量。

Aborted_connects 尝试已经失败的MySQL服务器的连接的次数。

Connections 试图连接MySQL服务器的次数。

Created_tmp_tables 当执行语句时, 已经被创造了的隐含临时表的数量。

Delayed_insert_threads 正在使用的延迟插入处理器线程的数量。

Delayed_writes 用INSERT DELAYED写入的行数。

Delayed_errors 用INSERT DELAYED写入的发生某些错误(可能重复键值)的行数。

Flush_commands 执行FLUSH命令的次数。

Handler_delete 请求从一张表中删除行的次数。

Handler_read_first 请求读入表中第一行的次数。

Handler_read_key 请求数字基于键读行。

Handler_read_next 请求读入基于一个键的一行的次数。

Handler_read_rnd 请求读入基于一个固定位置的一行的次数。

Handler_update 请求更新表中一行的次数。

Handler_write 请求向表中插入一行的次数。

Key_blocks_used 用于关键字缓存的块的数量。

Key_read_requests 请求从缓存读入一个键值的次数。

Key_reads 从磁盘物理读入一个键值的次数。

Key_write_requests 请求将一个关键字块写入缓存次数。
Key_writes 将一个键值块物理写入磁盘的次数。
Max_used_connections 同时使用的连接的最大数目。
Not_flushed_key_blocks 在键缓存中已经改变但是还没被清空到磁盘上的键块。
Not_flushed_delayed_rows 在INSERT DELAY队列中等待写入的行的数量。
Open_tables 打开表的数量。
Open_files 打开文件的数量。
Open_streams 打开流的数量(主要用于日志记载)
Opened_tables 已经打开的表的数量。
Questions 发往服务器的查询的数量。
Slow_queries 要花超过long_query_time时间的查询数量。
Threads_connected 当前打开的连接的数量。
Threads_running 不在睡眠的线程数量。
Uptime 服务器工作了多少秒。

```
</screen>
</section>

<section id="binary">
  <title>binary 日志</title>
  <screen>
    <![CDATA[
```

```
mysql> show binary logs;
```

Log_name	File_size
mysql-bin.000001	19544
mysql-bin.000002	974751
mysql-bin.000003	107
mysql-bin.000004	3976040
mysql-bin.000005	126
mysql-bin.000006	350063
mysql-bin.000007	6826
mysql-bin.000008	3879494
mysql-bin.000009	126
mysql-bin.000010	494
mysql-bin.000011	17286686
mysql-bin.000012	15003942
mysql-bin.000013	1709321

```
13 rows in set (0.00 sec)
```

7.4. 线程的使用情况

```
mysql> SHOW STATUS LIKE 'threads%';
+-----+-----+
| Variable_name      | Value |
+-----+-----+
| Threads_cached     | 0     |
| Threads_connected  | 1     |
| Threads_created    | 1     |
| Threads_running    | 1     |
+-----+-----+
4 rows in set (0.01 sec)
```

7.5. DATABASES

```
SHOW DATABASES;
</screen>
</section>
<section id="table">
  <title>TABLE</title>
  <screen><![CDATA[
SHOW TABLE STATUS FROM `dbname`;
</screen>
</section>
<section id="created_tmp">
  <title>临时表</title>
  <screen>
  <![CDATA[
mysql> SHOW STATUS LIKE 'created_tmp%';
+-----+-----+
| Variable_name      | Value |
+-----+-----+
| Created_tmp_disk_tables | 0     |
| Created_tmp_files    | 5     |
| Created_tmp_tables    | 0     |
+-----+-----+
3 rows in set (0.00 sec)
```

7.6. 排序统计信息

```
mysql> SHOW STATUS LIKE "sort%";
```

Variable_name	Value
Sort_merge_passes	0
Sort_range	0
Sort_rows	0
Sort_scan	0

```
4 rows in set (0.00 sec)
```

7.7. Key 状态

```
mysql> show status like '%key_read%';
```

Variable_name	Value
Key_read_requests	6
Key_reads	3

```
2 rows in set (0.00 sec)
```

7.8. FUNCTION

```
SHOW FUNCTION STATUS WHERE `Db`='dbname';
```

7.9. PROCEDURE

```
SHOW PROCEDURE STATUS WHERE `Db`='dbname';
```

7.10. TRIGGERS

```
SHOW TRIGGERS FROM `dbname`;
```

```
mysql> SHOW TRIGGERS LIKE '%trigger_name%'\G
Empty set (0.00 sec)

mysql> SHOW TRIGGERS LIKE '%demo%'\G
***** 1. row
*****
          Trigger: demo_AFTER_INSERT
             Event: INSERT
             Table: demo
        Statement: BEGIN
        set @rev = "";
        SELECT
        OUT2FILE('/tmp/demo.log',
          CONCAT_WS(',',
            NEW.id,
            NEW.name,
            NEW.sex,
            NEW.address))
        INTO @rev;
        END
          Timing: AFTER
         Created: 2017-11-23 11:47:58.10
        sql_mode:
ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION
          Definer: root@%
character_set_client: utf8
collation_connection: utf8_general_ci
  Database Collation: utf8_general_ci
1 row in set (0.00 sec)
```

7.11. EVENTS

```
SHOW EVENTS FROM `dbname`;
```

7.12. 引擎(ENGINES)

```
mysql> SHOW ENGINES;
```

```
+-----+-----+-----+-----+
| Engine          | Support | Comment                               |
| Transactions   | XA      | Savepoints                           |
+-----+-----+-----+-----+
+-----+
| CSV             | YES     | CSV storage engine                   |
| NO              | NO      |                                       |
| MRG_MYISAM     | YES     | Collection of identical MyISAM      |
tables                                     | NO      | NO  | NO
|
| PERFORMANCE_SCHEMA | YES     | Performance Schema                   |
| NO              | NO      |                                       |
| BLACKHOLE       | YES     | /dev/null storage engine             |
(anything you write to it disappears) | NO      | NO  |
NO
| MEMORY          | YES     | Hash based, stored in memory,        |
useful for temporary tables             | NO      | NO  | NO
|
| FEDERATED       | NO      | Federated MySQL storage engine       |
| NULL            | NULL    |                                       |
| ARCHIVE         | YES     | Archive storage engine               |
| NO              | NO      |                                       |
| MyISAM          | YES     | MyISAM storage engine                |
| NO              | NO      |                                       |
```

```

| InnoDB          | DEFAULT | Supports transactions, row-
level locking, and foreign keys | YES      | YES  | YES
|
+-----+-----+-----+-----+
-----+
9 rows in set (0.00 sec)

```

7.13. 字符集(Collation)

```

mysql> SHOW COLLATION;
+-----+-----+-----+-----+
--+-----+
| Collation          | Charset | Id  | Default |
Compiled | Sortlen |
+-----+-----+-----+-----+
--+-----+
| big5_chinese_ci   | big5    | 1  | Yes     | Yes
| 1 |
| big5_bin          | big5    | 84 |         | Yes
| 1 |
| dec8_swedish_ci   | dec8    | 3  | Yes     | Yes
| 1 |
| dec8_bin          | dec8    | 69 |         | Yes
| 1 |
| cp850_general_ci  | cp850   | 4  | Yes     | Yes
| 1 |
| cp850_bin         | cp850   | 80 |         | Yes
| 1 |
| hp8_english_ci    | hp8     | 6  | Yes     | Yes
| 1 |
| hp8_bin           | hp8     | 72 |         | Yes
| 1 |
| koi8r_general_ci  | koi8r   | 7  | Yes     | Yes
| 1 |
| koi8r_bin         | koi8r   | 74 |         | Yes
| 1 |
| latin1_german1_ci | latin1  | 5  |         | Yes
| 1 |

```

latin1_swedish_ci 1	latin1	8	Yes	Yes
latin1_danish_ci 1	latin1	15		Yes
latin1_german2_ci 2	latin1	31		Yes
latin1_bin 1	latin1	47		Yes
latin1_general_ci 1	latin1	48		Yes
latin1_general_cs 1	latin1	49		Yes
latin1_spanish_ci 1	latin1	94		Yes
latin2_czech_cs 4	latin2	2		Yes
latin2_general_ci 1	latin2	9	Yes	Yes
latin2_hungarian_ci 1	latin2	21		Yes
latin2_croatian_ci 1	latin2	27		Yes
latin2_bin 1	latin2	77		Yes
swe7_swedish_ci 1	swe7	10	Yes	Yes
swe7_bin 1	swe7	82		Yes
ascii_general_ci 1	ascii	11	Yes	Yes
ascii_bin 1	ascii	65		Yes
ujis_japanese_ci 1	ujis	12	Yes	Yes
ujis_bin 1	ujis	91		Yes
sjis_japanese_ci 1	sjis	13	Yes	Yes
sjis_bin 1	sjis	88		Yes
hebrew_general_ci 1	hebrew	16	Yes	Yes
hebrew_bin 1	hebrew	71		Yes
tis620_thai_ci	tis620	18	Yes	Yes

4					
tis620_bin		tis620		89	Yes
1					
euckr_korean_ci		euckr		19	Yes Yes
1					
euckr_bin		euckr		85	Yes
1					
koi8u_general_ci		koi8u		22	Yes Yes
1					
koi8u_bin		koi8u		75	Yes
1					
gb2312_chinese_ci		gb2312		24	Yes Yes
1					
gb2312_bin		gb2312		86	Yes
1					
greek_general_ci		greek		25	Yes Yes
1					
greek_bin		greek		70	Yes
1					
cp1250_general_ci		cp1250		26	Yes Yes
1					
cp1250_czech_cs		cp1250		34	Yes
2					
cp1250_croatian_ci		cp1250		44	Yes
1					
cp1250_bin		cp1250		66	Yes
1					
cp1250_polish_ci		cp1250		99	Yes
1					
gbk_chinese_ci		gbk		28	Yes Yes
1					
gbk_bin		gbk		87	Yes
1					
latin5_turkish_ci		latin5		30	Yes Yes
1					
latin5_bin		latin5		78	Yes
1					
armscii8_general_ci		armscii8		32	Yes Yes
1					
armscii8_bin		armscii8		64	Yes
1					
utf8_general_ci		utf8		33	Yes Yes
1					
utf8_bin		utf8		83	Yes
1					

utf8_unicode_ci	utf8	192		Yes
8				
utf8_icelandic_ci	utf8	193		Yes
8				
utf8_latvian_ci	utf8	194		Yes
8				
utf8_romanian_ci	utf8	195		Yes
8				
utf8_slovenian_ci	utf8	196		Yes
8				
utf8_polish_ci	utf8	197		Yes
8				
utf8_estonian_ci	utf8	198		Yes
8				
utf8_spanish_ci	utf8	199		Yes
8				
utf8_swedish_ci	utf8	200		Yes
8				
utf8_turkish_ci	utf8	201		Yes
8				
utf8_czech_ci	utf8	202		Yes
8				
utf8_danish_ci	utf8	203		Yes
8				
utf8_lithuanian_ci	utf8	204		Yes
8				
utf8_slovak_ci	utf8	205		Yes
8				
utf8_spanish2_ci	utf8	206		Yes
8				
utf8_roman_ci	utf8	207		Yes
8				
utf8_persian_ci	utf8	208		Yes
8				
utf8_esperanto_ci	utf8	209		Yes
8				
utf8_hungarian_ci	utf8	210		Yes
8				
utf8_sinhala_ci	utf8	211		Yes
8				
utf8_general_mysql500_ci	utf8	223		Yes
1				
ucs2_general_ci	ucs2	35	Yes	Yes
1				
ucs2_bin	ucs2	90		Yes

1						
ucs2_unicode_ci		ucs2		128		Yes
8						
ucs2_icelandic_ci		ucs2		129		Yes
8						
ucs2_latvian_ci		ucs2		130		Yes
8						
ucs2_romanian_ci		ucs2		131		Yes
8						
ucs2_slovenian_ci		ucs2		132		Yes
8						
ucs2_polish_ci		ucs2		133		Yes
8						
ucs2_estonian_ci		ucs2		134		Yes
8						
ucs2_spanish_ci		ucs2		135		Yes
8						
ucs2_swedish_ci		ucs2		136		Yes
8						
ucs2_turkish_ci		ucs2		137		Yes
8						
ucs2_czech_ci		ucs2		138		Yes
8						
ucs2_danish_ci		ucs2		139		Yes
8						
ucs2_lithuanian_ci		ucs2		140		Yes
8						
ucs2_slovak_ci		ucs2		141		Yes
8						
ucs2_spanish2_ci		ucs2		142		Yes
8						
ucs2_roman_ci		ucs2		143		Yes
8						
ucs2_persian_ci		ucs2		144		Yes
8						
ucs2_esperanto_ci		ucs2		145		Yes
8						
ucs2_hungarian_ci		ucs2		146		Yes
8						
ucs2_sinhala_ci		ucs2		147		Yes
8						
ucs2_general_mysql500_ci		ucs2		159		Yes
1						
cp866_general_ci		cp866		36		Yes
1						

cp866_bin	cp866	68		Yes
1				
keybcs2_general_ci	keybcs2	37	Yes	Yes
1				
keybcs2_bin	keybcs2	73		Yes
1				
macce_general_ci	macce	38	Yes	Yes
1				
macce_bin	macce	43		Yes
1				
macroman_general_ci	macroman	39	Yes	Yes
1				
macroman_bin	macroman	53		Yes
1				
cp852_general_ci	cp852	40	Yes	Yes
1				
cp852_bin	cp852	81		Yes
1				
latin7_estonian_cs	latin7	20		Yes
1				
latin7_general_ci	latin7	41	Yes	Yes
1				
latin7_general_cs	latin7	42		Yes
1				
latin7_bin	latin7	79		Yes
1				
utf8mb4_general_ci	utf8mb4	45	Yes	Yes
1				
utf8mb4_bin	utf8mb4	46		Yes
1				
utf8mb4_unicode_ci	utf8mb4	224		Yes
8				
utf8mb4_icelandic_ci	utf8mb4	225		Yes
8				
utf8mb4_latvian_ci	utf8mb4	226		Yes
8				
utf8mb4_romanian_ci	utf8mb4	227		Yes
8				
utf8mb4_slovenian_ci	utf8mb4	228		Yes
8				
utf8mb4_polish_ci	utf8mb4	229		Yes
8				
utf8mb4_estonian_ci	utf8mb4	230		Yes
8				
utf8mb4_spanish_ci	utf8mb4	231		Yes

8	utf8mb4_swedish_ci	utf8mb4	232		Yes
8	utf8mb4_turkish_ci	utf8mb4	233		Yes
8	utf8mb4_czech_ci	utf8mb4	234		Yes
8	utf8mb4_danish_ci	utf8mb4	235		Yes
8	utf8mb4_lithuanian_ci	utf8mb4	236		Yes
8	utf8mb4_slovak_ci	utf8mb4	237		Yes
8	utf8mb4_spanish2_ci	utf8mb4	238		Yes
8	utf8mb4_roman_ci	utf8mb4	239		Yes
8	utf8mb4_persian_ci	utf8mb4	240		Yes
8	utf8mb4_esperanto_ci	utf8mb4	241		Yes
8	utf8mb4_hungarian_ci	utf8mb4	242		Yes
8	utf8mb4_sinhala_ci	utf8mb4	243		Yes
8	cp1251_bulgarian_ci	cp1251	14		Yes
1	cp1251_ukrainian_ci	cp1251	23		Yes
1	cp1251_bin	cp1251	50		Yes
1	cp1251_general_ci	cp1251	51	Yes	Yes
1	cp1251_general_cs	cp1251	52		Yes
1	utf16_general_ci	utf16	54	Yes	Yes
1	utf16_bin	utf16	55		Yes
1	utf16_unicode_ci	utf16	101		Yes
8	utf16_icelandic_ci	utf16	102		Yes
8	utf16_latvian_ci	utf16	103		Yes
8					

utf16_romanian_ci 8	utf16	104		Yes
utf16_slovenian_ci 8	utf16	105		Yes
utf16_polish_ci 8	utf16	106		Yes
utf16_estonian_ci 8	utf16	107		Yes
utf16_spanish_ci 8	utf16	108		Yes
utf16_swedish_ci 8	utf16	109		Yes
utf16_turkish_ci 8	utf16	110		Yes
utf16_czech_ci 8	utf16	111		Yes
utf16_danish_ci 8	utf16	112		Yes
utf16_lithuanian_ci 8	utf16	113		Yes
utf16_slovak_ci 8	utf16	114		Yes
utf16_spanish2_ci 8	utf16	115		Yes
utf16_roman_ci 8	utf16	116		Yes
utf16_persian_ci 8	utf16	117		Yes
utf16_esperanto_ci 8	utf16	118		Yes
utf16_hungarian_ci 8	utf16	119		Yes
utf16_sinhala_ci 8	utf16	120		Yes
cp1256_general_ci 1	cp1256	57	Yes	Yes
cp1256_bin 1	cp1256	67		Yes
cp1257_lithuanian_ci 1	cp1257	29		Yes
cp1257_bin 1	cp1257	58		Yes
cp1257_general_ci 1	cp1257	59	Yes	Yes
utf32_general_ci	utf32	60	Yes	Yes

1	utf32_bin	utf32	61		Yes
1	utf32_unicode_ci	utf32	160		Yes
8	utf32_icelandic_ci	utf32	161		Yes
8	utf32_latvian_ci	utf32	162		Yes
8	utf32_romanian_ci	utf32	163		Yes
8	utf32_slovenian_ci	utf32	164		Yes
8	utf32_polish_ci	utf32	165		Yes
8	utf32_estonian_ci	utf32	166		Yes
8	utf32_spanish_ci	utf32	167		Yes
8	utf32_swedish_ci	utf32	168		Yes
8	utf32_turkish_ci	utf32	169		Yes
8	utf32_czech_ci	utf32	170		Yes
8	utf32_danish_ci	utf32	171		Yes
8	utf32_lithuanian_ci	utf32	172		Yes
8	utf32_slovak_ci	utf32	173		Yes
8	utf32_spanish2_ci	utf32	174		Yes
8	utf32_roman_ci	utf32	175		Yes
8	utf32_persian_ci	utf32	176		Yes
8	utf32_esperanto_ci	utf32	177		Yes
8	utf32_hungarian_ci	utf32	178		Yes
8	utf32_sinhala_ci	utf32	179		Yes
1	binary	binary	63	Yes	Yes

geostd8_general_ci	geostd8	92	Yes	Yes
1				
geostd8_bin	geostd8	93		Yes
1				
cp932_japanese_ci	cp932	95	Yes	Yes
1				
cp932_bin	cp932	96		Yes
1				
eucjpms_japanese_ci	eucjpms	97	Yes	Yes
1				
eucjpms_bin	eucjpms	98		Yes
1				

7.14. SHOW GRANTS

```

MariaDB [test]> SHOW GRANTS;
+-----+
| Grants for root@localhost |
+-----+
| GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' IDENTIFIED
BY PASSWORD '*C6325DAF39AE6CC34E960D3C65F1398FE467E1D0' WITH
GRANT OPTION |
| GRANT PROXY ON ''@'' TO 'root'@'localhost' WITH GRANT OPTION
|
+-----+
2 rows in set (0.00 sec)

```

7.15. validate_password

```


```

```
SHOW VARIABLES LIKE 'validate_password.%';
```

```
mysql> SHOW VARIABLES LIKE 'validate_password.%';
```

Variable_name	Value
validate_password.check_user_name	ON
validate_password.dictionary_file	
validate_password.length	8
validate_password.mixed_case_count	1
validate_password.number_count	1
validate_password.policy	MEDIUM
validate_password.special_char_count	1

```
7 rows in set (0.00 sec)
```


8. Maintenance 数据库维护

8.1. CHECK 检查表

```
<![CDATA[  
CHECK TABLE `dbname`.`actions`;  
CHECK TABLE `dbname`.`actions` QUICK FAST MEDIUM EXTENDED  
CHANGED;
```

8.2. ANALYZE 分析表

```
ANALYZE TABLE `dbname`.`actions`;
```

8.3. CHECKSUM

```
CHECKSUM TABLE `dbname`.`actions` QUICK;
```

8.4. OPTIMIZE 优化表

```
OPTIMIZE TABLE `dbname`.`actions`;
```

8.5. REPAIR 修复

```
REPAIR TABLE `dbname`.`members`;
```

```
SHOW TABLE STATUS LIKE 'members';
```

9. INFORMATION_SCHEMA

9.1. 查询表字段

```
SELECT
    GROUP_CONCAT(COLUMN_NAME) AS fields
FROM
    INFORMATION_SCHEMA.Columns
WHERE
    table_name = 'mytable'
    AND table_schema = 'test';
```

9.2. 列出所有触发器

```
select trigger_schema, trigger_name, action_statement from
information_schema.triggers

select * from information_schema.triggers where
information_schema.triggers.trigger_schema like '%test%';

select * from information_schema.triggers where
information_schema.triggers.trigger_name like '%trigger_name%' and
information_schema.triggers.trigger_schema like '%data_base_name%';
```

10. Backup and Recovery

10.1. Import / Export

Export(Backup)

```
mysqldump -hlocalhost -proot -p**** mydb > mydb.sql
```

gzip

```
mysqldump -hlocalhost -proot -p**** mydb | gzip > mydb.sql.gz
```

Import(Recovery)

```
mysql -hlocalhost -proot -p**** mydb < mydb.sql
```

gunzip

```
gunzip mydb.sql.gz -c | mysql -hlocalhost -proot -p**** mydb
```

xml

export xml

```
$ mysqldump -uusername -ppasswd -X -t database table -r  
filename.xml
```

备份表数据

```
SELECT * INTO OUTFILE 'file_name' FROM tbl_name  
LOAD DATA INFILE 'file_name' REPLACE INTO TABLE tbl_name
```

source

```
mysql> use your_db  
mysql> SOURCE database.sql
```

使用 **mysqlhotcopy** 备份 **MyISAM** 引擎的数据库

shell> mysqlhotcopy db_name /path/to/some/dir

```
mysql:~# mysqlhotcopy --user=neo --password=chen shop  
/tmp/backup  
Locked 100 tables in 0 seconds.  
Flushed tables (`shop`.`account_log`, `shop`.`ad`,  
`shop`.`ad_custom`, `shop`.`ad_position`,  
`shop`.`admin_action`,  
`shop`.`admin_log`, `shop`.`admin_message`,  
`shop`.`admin_user`, `shop`.`adsense`, `shop`.`affiliate_log`,  
...  
...  
...  
`shop`.`user_rank`, `shop`.`users`, `shop`.`virtual_card`,  
`shop`.`volume_price`, `shop`.`vote`, `shop`.`vote_log`,
```

```
`shop`.`vote_option`, `shop`.`wholesale`) in 0 seconds.  
Copying 299 files...  
Copying indices for 0 files...  
Unlocked tables.  
mysqlhotcopy copied 100 tables (299 files) in 0 seconds (0  
seconds overall).
```

AutoMySQLBackup

<https://sourceforge.net/projects/automysqlbackup/>

xtrabackup - Open source backup tool for InnoDB and XtraDB.

<https://launchpad.net/percona-xtrabackup>

Percona yum Repository

```
$ rpm -Uvh http://www.percona.com/downloads/percona-  
release/percona-release-0.0-1.x86_64.rpm
```

```
# yum search xtrabackup  
=====
```

= N/S Matched: XtraBackup

```
=====
```

==
holland-xtrabackup.noarch : Xtrabackup plugin for Holland
percona-xtrabackup.x86_64 : XtraBackup online backup for MySQL / InnoDB
percona-xtrabackup-debuginfo.x86_64 : Debug information for package percona-xtrabackup
percona-xtrabackup-test.x86_64 : Test suite for Percona Xtrabackup

```
# yum install percona-xtrabackup
```

Creating an Incremental Backup

```
xtrabackup --backup --target-dir=/data/backups/base --  
datadir=/var/lib/mysql/
```

10.2. Snapshot Backup

LVM Snapshot

```
# mysql -uroot -pmysql  
mysql> flush tables with read lock;  
mysql>flush logs;  
mysql>system lvcreate -L1024M -s -n snap0 /dev/vg00/lvol00  
mysql>show master status;  
mysql>unlock tables;  
mysql>quit
```

Btrfs Snapshot

```
# btrfs subvolume snapshot /data /data/backup_2013-03-20  
Create a snapshot of '/data' in '/data/backup_2013-03-20'  
  
btrfs subvolume list /data  
ID 315 gen 172 top level 5 path backup_2013-03-10  
ID 320 gen 178 top level 5 path backup_2013-03-20
```

第 22 章 DDL - Data Definition Language

1. 数据库管理(Database)

1.1. create

Creating a UTF-8 database

```
CREATE DATABASE db_name DEFAULT CHARACTER SET utf8 COLLATE  
utf8_general_ci;
```

Create a UTF-8 database with binary UTF-8 collation.

```
CREATE DATABASE dbname CHARACTER SET utf8 COLLATE utf8_bin;
```

1.2. drop

```
DROP DATABASE db_name;
```

1.3. 修改数据库

```
ALTER DATABASE dbname DEFAULT CHARACTER SET utf8 COLLATE  
utf8_general_ci;
```


1.4. Rename

```
RENAME {DATABASE | SCHEMA} db_name TO new_db_name;
```

before 5.0 version

```
[neo@development ~]$ mysqldump -uroot -pchen db_old | mysql -uroot -pchen db_new
```

1.5. CHARACTER

```
ALTER DATABASE <database_name> CHARACTER SET utf8;
```

1.6. show create database

```
mysql> show create database dbname;
```

```
+-----+-----+
| Database | Create Database
|
+-----+-----+
| dbname   | CREATE DATABASE `dbname` /*!40100 DEFAULT
CHARACTER SET utf8 */
+-----+-----+
1 row in set (0.00 sec)
```

2. 表管理(Table)

2.1. 数据类型

SET 集合类型

SET 集合类型，此类型适合用于多项选择场景，例如保存表单中的checkbox。

```
CREATE TABLE `QA` (  
  `id` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,  
  `question` VARCHAR(255) NOT NULL COMMENT '问题描述',  
  `answer` SET('A','B','C','D') NOT NULL COMMENT '问题答案',  
  PRIMARY KEY (`id`)  
)  
COMMENT='Multiple Choice'  
COLLATE='utf8_general_ci'  
ENGINE=InnoDB;
```

插入数据

```
INSERT INTO `QA` (`id`, `question`, `answer`) VALUES  
  (1, 'Netkiller 系列手札始于那一年? A.2000年, B.2008年, C.2010年, D.2016年', 'A'),  
  (2, 'Netkiller 系列手札有哪些? A.《Netkiller Scals 手札》, B.《Netkiller Java 手札》, C.《Netkiller Linux 手札》, D.《Netkiller EMC 手札》', 'B,C'),  
  (3, 'XXXXXXXX', 'C,D'),  
  (4, 'XXXXXXXX', 'A,B,C'),  
  ...  
  ...  
  (1000, 'XXXXXXXX', 'B,C,D'),  
  ...  
  ...  
  (5000, 'XXXXXXXX', 'A,B,C,D');
```

查询 SET 结果集，MySQL为SET配备了FIND_IN_SET函数

```
select * from QA where FIND_IN_SET('B',`answer`);
```

下面两种方法也能实现，但不推荐使用。

```
select question, answer from QA where locate('B',answer)>0;  
select question, answer from QA where POSITION('B' in answer)>0;
```

查询多个答案

```
select question, answer from QA where answer = 'B,C';
```

2.2. create table ... select

创建空表

```
create table admin_user_history select * from admin_user where 1 <> 1;
```

创建有数据的表

```
create table admin_user_history select * from admin_user;
```

2.3. modify table

modify table

```
ALTER TABLE ecs_users add user_picture varchar(255);
```

2.4. TEMPORARY Table

临时表将在你连接期间存在。一旦断开时将自动删除表并释放所用的空间。你在连接期间删除该表也同样释放空间。

```
CREATE TEMPORARY TABLE tmp_table (  
    key VARCHAR(10) NOT NULL,  
    value INTEGER NOT NULL  
)
```

声明临时表是一个HEAP表，允许你指定在内存中创建它

```
CREATE TEMPORARY TABLE tmp_mem_table (  
    key VARCHAR(10) NOT NULL,  
    value INTEGER NOT NULL  
) TYPE = HEAP
```

2.5. Collate

```
ALTER TABLE `tmp_cats` COLLATE='utf8_general_ci', CONVERT TO CHARSET utf8;
```

2.6. CHARACTER

```
ALTER TABLE <table_name> CONVERT TO CHARACTER SET utf8;  
alter table <table_name> convert to charset utf8mb4;
```

2.7. DEFAULT

AUTO_INCREMENT

定义 AUTO_INCREMENT 起始值

```
CREATE TABLE `bank_account` (  
    `id` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT COMMENT '自增唯一ID',  
    `name` VARCHAR(50) NOT NULL DEFAULT '0' COMMENT '帐号名称(Name on account)',  
    PRIMARY KEY (`id`)  
)  
COMMENT='银行帐号'  
COLLATE='utf8_general_ci'  
ENGINE=InnoDB  
AUTO_INCREMENT=2;
```

设置 AUTO_INCREMENT

```
ALTER TABLE `accounts`  
    AUTO_INCREMENT=792257;
```

TIMESTAMP NULL DEFAULT NULL ON UPDATE

```
alter table cms.article ADD COLUMN `mtime` TIMESTAMP NULL DEFAULT NULL ON UPDATE  
CURRENT_TIMESTAMP COMMENT '更改时间';
```

更新时间

```
`mtime` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE  
CURRENT_TIMESTAMP COMMENT '更改时间',
```

```
CREATE TABLE `bank_account` (  
  `id` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT COMMENT '自增唯一ID',  
  `bank_name` VARCHAR(255) NOT NULL DEFAULT '0' COMMENT '银行名字(Bank Name)',  
  `name` VARCHAR(50) NOT NULL DEFAULT '0' COMMENT '帐号名称(Name on account)',  
  `account_number` VARCHAR(50) NOT NULL DEFAULT '0' COMMENT '银行帐号(Account  
Number)',  
  `branch_location` VARCHAR(255) NOT NULL DEFAULT '0' COMMENT '支行位置(Branch  
Location)',  
  `description` VARCHAR(255) NOT NULL DEFAULT '0' COMMENT '银行描述',  
  `status` ENUM('Y','N') NOT NULL DEFAULT 'N' COMMENT '银行帐号状态',  
  `ctime` DATETIME NOT NULL COMMENT '创建时间',  
  `mtime` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP  
COMMENT '更改时间',  
  PRIMARY KEY (`id`)  
)  
COMMENT='银行帐号'  
COLLATE='utf8_general_ci'  
ENGINE=InnoDB  
AUTO_INCREMENT=2;
```

表存储位置(DATA DIRECTORY)

```
CREATE TABLE IF NOT EXISTS `tab_name` (  
  `id` int(11) DEFAULT NULL,  
  `purchased` date DEFAULT NULL,  
  KEY `Index 1` (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8  
/*!50100 PARTITION BY LIST (YEAR(purchased))  
(PARTITION p0 VALUES IN (1990) DATA DIRECTORY = '/www/data' ENGINE = InnoDB) */;
```

2.8. KEY

PRIMARY KEY

一般主键定义

```
PRIMARY KEY (`id`),
```

复合主键

```
PRIMARY KEY (`id`, `user_id`),
```

2.9. AUTO_INCREMENT 定义初始值

```
DROP TABLE IF EXISTS users;  
CREATE TABLE user(  
    id          INT     NOT NULL AUTO_INCREMENT  
PRIMARY KEY(id)  
)ENGINE=InnoDB AUTO_INCREMENT=100 DEFAULT CHARSET=utf8;
```

修改 auto_increment 起始值

```
alter table tabl auto_increment=n
```

2.10. COMMENT

```
ALTER TABLE `neo`.`stuff` COMMENT = '用户表' ;  
ALTER TABLE `neo`.`stuff` CHANGE COLUMN `name` `name` VARCHAR(50) NULL DEFAULT NULL  
COMMENT '姓名' ;  
ALTER TABLE `neo`.`stuff` CHANGE COLUMN `password` `password` VARCHAR(50) NULL DEFAULT  
NULL COMMENT '用户密码' ;  
  
CREATE TABLE `stuff` (  
    `id` int(11) NOT NULL AUTO_INCREMENT,  
    `name` varchar(50) DEFAULT NULL COMMENT ''姓名'',  
    `password` varchar(50) DEFAULT NULL COMMENT ''用户密码'',  
    `created` date NOT NULL DEFAULT ''0000-00-00'',  
    PRIMARY KEY (`id`,`created`)  
) ENGINE=MyISAM AUTO_INCREMENT=5 DEFAULT CHARSET=latin1 COMMENT=''用户表''  
/*!50100 PARTITION BY HASH (year(created))  
PARTITIONS 10 */
```

2.11. Engine 存储引擎

显示当前数据库支持引擎

```
mysql> show engines;
+-----+-----+-----+
| Engine          | Support | Comment                                     |
| Transactions | XA      | Savepoints |
+-----+-----+-----+
| FEDERATED       | NO      | Federated MySQL storage engine           |
| NULL            | NULL    | NULL                                       |
| MRG_MYISAM      | YES     | Collection of identical MyISAM tables     |
| NO              | NO      | NO                                         |
| MyISAM          | YES     | MyISAM storage engine                    |
| NO              | NO      | NO                                         |
| BLACKHOLE       | YES     | /dev/null storage engine (anything you write to it disappears) |
| NO              | NO      | NO                                         |
| MEMORY          | YES     | Hash based, stored in memory, useful for temporary tables |
| NO              | NO      | NO                                         |
| CSV             | YES     | CSV storage engine                       |
| NO              | NO      | NO                                         |
| ARCHIVE         | YES     | Archive storage engine                   |
| NO              | NO      | NO                                         |
| PERFORMANCE_SCHEMA | YES     | Performance Schema                       |
| NO              | NO      | NO                                         |
| InnoDB          | DEFAULT | Supports transactions, row-level locking, and foreign keys |
| YES            | YES     | YES                                         |
+-----+-----+-----+
9 rows in set (0.00 sec)
```

切换引擎

修改与切换引擎

```
ALTER TABLE `test` ENGINE=BLACKHOLE;
ALTER TABLE `test` ENGINE=InnoDB;
```

FEDERATED

启用 FEDERATED 引擎, 服务器环境 Ubuntu 13.04

```
$ sudo vim /etc/mysql/conf.d/federated.cnf
[mysqld]
federated

$ sudo service mysql restart
```



```
mysql> show engines;
```

```
+-----+-----+-----+
| Engine      | Support | Comment
| Transactions | XA      | Savepoints |
+-----+-----+-----+
| FEDERATED   | YES     | Federated MySQL storage engine
| NO          | NO      |
| MRG_MYISAM  | YES     | Collection of identical MyISAM tables
| NO          | NO      |
| MyISAM      | YES     | MyISAM storage engine
| NO          | NO      |
| BLACKHOLE   | YES     | /dev/null storage engine (anything you write to it
disappears) | NO      | NO
| MEMORY      | YES     | Hash based, stored in memory, useful for temporary
tables      | NO      | NO
| CSV         | YES     | CSV storage engine
| NO          | NO      |
| ARCHIVE     | YES     | Archive storage engine
| NO          | NO      |
| PERFORMANCE_SCHEMA | YES     | Performance Schema
| NO          | NO      |
| InnoDB      | DEFAULT | Supports transactions, row-level locking, and foreign
keys       | YES     | YES
+-----+-----+-----+
9 rows in set (0.00 sec)
```

A 服务器

```
CREATE TABLE `t1` (
  `id` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(50) NOT NULL,
  `sex` ENUM('Y','N') NULL DEFAULT 'Y',
  `passwd` VARCHAR(50) NULL DEFAULT NULL,
  `ctime` TIMESTAMP NOT NULL DEFAULT '0000-00-00 00:00:00',
  `mtime` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
  PRIMARY KEY (`id`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
AUTO_INCREMENT=4;
```

B 服务器

```
DROP TABLE `users`;

CREATE TABLE `users` (
  `id` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(50) NOT NULL,
  `sex` ENUM('Y','N') NULL DEFAULT 'Y',
```

```

        `ctime` TIMESTAMP NOT NULL DEFAULT '0000-00-00 00:00:00',
        `mtime` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
        PRIMARY KEY (`id`)
) ENGINE=FEDERATED connection = 'mysql://www:qwer123@192.168.2.1:3306/test/t1';

```

上面字段描述是你需要的字段，并非所有字段。这里屏蔽了passwd字段

提示

```
connection = 'mysql://用户名:密码@主机:端口/数据库/表名'
```

```

mysql> DROP TABLE `users`;
Query OK, 0 rows affected (0.00 sec)

mysql> CREATE TABLE `users` (
  -> `id` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,
  -> `name` VARCHAR(50) NOT NULL,
  -> `sex` ENUM('Y','N') NULL DEFAULT 'Y',
  -> `ctime` TIMESTAMP NOT NULL DEFAULT '0000-00-00 00:00:00',
  -> `mtime` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  -> PRIMARY KEY (`id`)
  -> ) ENGINE=FEDERATED connection = 'mysql://www:qwer123@192.168.2.1:3306/test/t1';
Query OK, 0 rows affected (0.06 sec)

mysql>
mysql> show tables;
+-----+
| Tables_in_test |
+-----+
| users          |
+-----+
1 row in set (0.00 sec)

mysql> desc users;
+-----+-----+-----+-----+-----+-----+
| Field | Type                | Null | Key | Default                | Extra          |
+-----+-----+-----+-----+-----+-----+
| id    | int(10) unsigned   | NO   | PRI | NULL                   | auto_increment |
| name  | varchar(50)        | NO   |     | NULL                   |                |
| sex   | enum('Y','N')      | YES  |     | Y                       |                |
| ctime | timestamp          | NO   |     | 0000-00-00 00:00:00   |                |
| mtime | timestamp          | NO   |     | CURRENT_TIMESTAMP     | on update     |
CURRENT_TIMESTAMP |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select * from users;
+-----+-----+-----+-----+-----+-----+
| id | name | sex | ctime                | mtime                |
+-----+-----+-----+-----+-----+-----+

```

```

+----+-----+-----+-----+-----+-----+
| 1 | neo | Y | 0000-00-00 00:00:00 | 2013-05-17 18:05:09 |
| 2 | zen | Y | 0000-00-00 00:00:00 | 2013-05-17 18:05:11 |
| 3 | lily | N | 0000-00-00 00:00:00 | 2013-05-17 18:05:22 |
+----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)

```

FEDERATED 与 mysqldump 问题!

切记, mysqldump 只会dump出使用FEDERATED引擎表的结构,不会包含数据。

BLACKHOLE

```
CREATE TABLE test(id INT, val CHAR(10)) ENGINE = BLACKHOLE;
```

ARCHIVE

归档(是适用于存放大量数据的存储引擎),仅支持select、insert操作;不支持delete、update、索引等操作;使用zlib无损算法压缩数据,节省磁盘空间;

适用场景:适用于大量可查询但不能删除的历史数据保存;

基于 order 表创建 order_audit 归档表

```
create table order_audit engine=archive as select * from `order`;
```

order_audit 表结构如下

```
CREATE TABLE `order_audit` (
  `id` int(10) unsigned NOT NULL DEFAULT '0' COMMENT '订单ID',
  `name` varchar(45) NOT NULL COMMENT '订单名称',
  `price` float NOT NULL COMMENT '价格',
  `ctime` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间'
) ENGINE=ARCHIVE DEFAULT CHARSET=utf8
```

```
mysql> show table status like 'order_audit';
```

```

+-----+-----+-----+-----+-----+-----+-----+
| Name          | Engine | Version | Row_format | Rows | Avg_row_length | Data_length |
+-----+-----+-----+-----+-----+-----+-----+
| order_audit   | ARCHIVE |          |             |      |                 |             |
+-----+-----+-----+-----+-----+-----+-----+
| Max_data_length | Index_length | Data_free | Auto_increment | Create_time |
+-----+-----+-----+-----+-----+-----+-----+
| Update_time    | Check_time | Collation | Checksum | Create_options |
+-----+-----+-----+-----+-----+-----+

```

```

Comment |
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
| order_audit | ARCHIVE | 10 | Compressed | 4 | 2215 | 8861 |
0 | 0 | 0 | NULL | NULL | 2017-11-16 17:30:34 |
NULL | utf8_general_ci | NULL | | | |
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)

```

CSV

创建表

```

CREATE TABLE `csv_table` (
  `id` int(11) NOT NULL,
  `name` varchar(45) NOT NULL,
  `age` int(11) NOT NULL
) ENGINE=CSV DEFAULT CHARSET=utf8

```

查看表状态

```

mysql> show table status like 'csv_table';
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
| Name      | Engine | Version | Row_format | Rows | Avg_row_length | Data_length |
Max_data_length | Index_length | Data_free | Auto_increment | Create_time |
Update_time | Check_time | Collation      | Checksum | Create_options | Comment |
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
| csv_table | CSV    | 10      | Dynamic    | 2    | 0              | 0          |
0 | 0 | 0 | NULL | NULL | NULL | NULL |
| utf8_general_ci | NULL | | | | |
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

```

插入数据

```

insert into csv_table values (1,'Neo',37),(2,'Jam',40);

```

查看数据

```
mysql> SELECT * FROM test.csv_table;
+----+-----+-----+
| id | name | age |
+----+-----+-----+
|  1 | Neo  |  37 |
|  2 | Jam  |  40 |
+----+-----+-----+
2 rows in set (0.00 sec)
```

CSV 引擎是可以直接将csv文件复制出来的，表存储在 /var/lib/mysql/ 目录

```
root@netkiller /etc/nginx/conf.d % ls -l /var/lib/mysql/test/csv*
/var/lib/mysql/test/csv_table.CSM
/var/lib/mysql/test/csv_table.CSV
/var/lib/mysql/test/csv_table.frm
```

.*CSM,*.frm 是表结构文件，*.CSV 是我们需要的文件，纯文本，可以使用Excel打开。

```
root@netkiller /etc/nginx/conf.d % cat /var/lib/mysql/test/csv_table.CSV
1,"Neo",37
2,"Jam",40
```

3. Partitioning

```
mysql> SHOW VARIABLES LIKE '%partition%';
```

```
+-----+-----+
| Variable_name | Value |
+-----+-----+
| have_partitioning | YES |
+-----+-----+
1 row in set (0.00 sec)
```

3.1. RANGE

18.5.1. Partitioning Keys, Primary Keys, and Unique Keys
This section discusses the relationship of partitioning keys with primary keys and unique keys. The rule governing this relationship can be expressed as follows: All columns used in the partitioning expression for a partitioned table must be part of every unique key that the table may have.

In other words, every unique key on the table must use every column in the table's partitioning expression. (This also includes the table's primary key, since it is by definition a unique key. This particular case is discussed later in this section.) For example, each of the following table creation statements is invalid:

SQL code:

```
mysql> create table tx (
->     id int not null ,
->     info_time date,
->     primary key(id,info_time)
-> )
-> PARTITION BY RANGE(info_time div 100)
-> (
->     PARTITION p_2008_11 VALUES LESS THAN (200812),
```

```
-> PARTITION p_2008_12 VALUES LESS THAN (200901),
-> PARTITION p_2009_01 VALUES LESS THAN (200902),
-> PARTITION p_2009_02 VALUES LESS THAN (200903),
-> PARTITION p_2009_03 VALUES LESS THAN (200904),
-> PARTITION p_2009_04 VALUES LESS THAN (200905),
-> PARTITION p_catch_all VALUES LESS THAN MAXVALUE
-> );
Query OK, 0 rows affected (0.17 sec)

mysql>
```

```
CREATE TABLE t1 (
  year_col INT,
  some_data INT
)
PARTITION BY RANGE (year_col) (
  PARTITION p0 VALUES LESS THAN (1991),
  PARTITION p1 VALUES LESS THAN (1995),
  PARTITION p2 VALUES LESS THAN (1999),
  PARTITION p3 VALUES LESS THAN (2002),
  PARTITION p4 VALUES LESS THAN (2006),
  PARTITION p5 VALUES LESS THAN MAXVALUE
);
```

e.g.2

```
CREATE TABLE rc (
  a INT NOT NULL,
  b INT NOT NULL
)
PARTITION BY RANGE COLUMNS(a,b) (
  PARTITION p0 VALUES LESS THAN (10,5),
  PARTITION p1 VALUES LESS THAN (20,10),
  PARTITION p2 VALUES LESS THAN (MAXVALUE,15),
  PARTITION p3 VALUES LESS THAN (MAXVALUE,MAXVALUE)
);
```

```
CREATE TABLE part_tab
(
    c1 int default NULL,
    c2 varchar(30) default NULL,
    c3 date default NULL
) engine=myisam
PARTITION BY RANGE (year(c3)) (
    PARTITION p0 VALUES LESS THAN (2000) ,
    PARTITION p1 VALUES LESS THAN (2001) ,
    PARTITION p2 VALUES LESS THAN (2002) ,
    PARTITION p3 VALUES LESS THAN (2003) ,
    PARTITION p4 VALUES LESS THAN (2004) ,
    PARTITION p12 VALUES LESS THAN (2012),
    PARTITION p13 VALUES LESS THAN MAXVALUE
);
```

3.2. LIST

```
CREATE TABLE client_firms (
    id INT,
    name VARCHAR(35)
)
PARTITION BY LIST (id) (
    PARTITION r0 VALUES IN (1, 5, 9, 13, 17, 21),
    PARTITION r1 VALUES IN (2, 6, 10, 14, 18, 22),
    PARTITION r2 VALUES IN (3, 7, 11, 15, 19, 23),
    PARTITION r3 VALUES IN (4, 8, 12, 16, 20, 24)
);
```



```

CREATE TABLE lc (
  a INT NULL,
  b INT NULL
)
PARTITION BY LIST COLUMNS(a,b) (
  PARTITION p0 VALUES IN( (0,0), (NULL,NULL) ),
  PARTITION p1 VALUES IN( (0,1), (0,2), (0,3), (1,1), (1,2)
),
  PARTITION p2 VALUES IN( (1,0), (2,0), (2,1), (3,0), (3,1)
),
  PARTITION p3 VALUES IN( (1,3), (2,2), (2,3), (3,2), (3,3)
)
);

```

```

CREATE TABLE th (id INT, name VARCHAR(30), adate DATE)
PARTITION BY LIST(YEAR(adate))
(
  PARTITION p1999 VALUES IN (1995, 1999, 2003)
  DATA DIRECTORY = '/var/appdata/95/data'
  INDEX DIRECTORY = '/var/appdata/95/idx',
  PARTITION p2000 VALUES IN (1996, 2000, 2004)
  DATA DIRECTORY = '/var/appdata/96/data'
  INDEX DIRECTORY = '/var/appdata/96/idx',
  PARTITION p2001 VALUES IN (1997, 2001, 2005)
  DATA DIRECTORY = '/var/appdata/97/data'
  INDEX DIRECTORY = '/var/appdata/97/idx',
  PARTITION p2000 VALUES IN (1998, 2002, 2006)
  DATA DIRECTORY = '/var/appdata/98/data'
  INDEX DIRECTORY = '/var/appdata/98/idx'
);

```

3.3. HASH

```

CREATE TABLE `test` (
  `userid` int(10) unsigned NOT NULL auto_increment,
  `username` int(10) unsigned NOT NULL DEFAULT '0',
  `password` int(10) unsigned NOT NULL DEFAULT '0',

  primary key (`userid`),
  KEY `userid` (`username`)
) ENGINE=InnoDB
PARTITION BY HASH(userid)
PARTITIONS 8;

```

使用HASH (year(created)) 替代 RANGE(year(created))

```

CREATE TABLE stuff (
  id INT AUTO_INCREMENT,
  name varchar(50),
  password varchar(50),
  created DATE,
  PRIMARY KEY (id, created)
)
PARTITION BY RANGE(year(created)) (
  PARTITION p0 VALUES LESS THAN (2010),
  PARTITION p1 VALUES LESS THAN (2012),
  PARTITION p2 VALUES LESS THAN MAXVALUE
);

```

更好的方法

```

CREATE TABLE stuff (
  id INT AUTO_INCREMENT,
  name varchar(50),
  password varchar(50),
  created DATE,
  PRIMARY KEY (id, created)
)
PARTITION BY HASH (year(created)) PARTITIONS 10;

```

我们演示一下

```
mysql> CREATE TABLE stuff (  
-> id INT AUTO_INCREMENT,  
-> name varchar(50),  
-> password varchar(50),  
-> created DATE,  
-> PRIMARY KEY (id, created)  
-> )  
-> PARTITION BY HASH (year(created)) PARTITIONS 10;  
Query OK, 0 rows affected (0.08 sec)
```

```
mysql> insert into stuff (name,password,created)  
values('neo','test','2010-10-1');  
Query OK, 1 row affected (0.06 sec)
```

```
mysql> insert into stuff (name,password,created)  
values('neo1','test','2012-2-1');  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> insert into stuff (name,password,created)  
values('neo2','test','2012-3-5');  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> insert into stuff (name,password,created)  
values('neo4','test','2011-1-5');  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> SELECT  
-> partition_name part,  
-> partition_expression expr,  
-> partition_description descr,  
-> table_rows  
-> FROM  
-> INFORMATION_SCHEMA.partitions  
-> WHERE  
-> TABLE_SCHEMA = schema()  
-> AND TABLE_NAME='stuff';
```

part	expr	descr	table_rows
p0	year(created)	NULL	1
p1	year(created)	NULL	1
p2	year(created)	NULL	2

p3	year(created)	NULL	0
p4	year(created)	NULL	0
p5	year(created)	NULL	0
p6	year(created)	NULL	0
p7	year(created)	NULL	0
p8	year(created)	NULL	0
p9	year(created)	NULL	0

10 rows in set (0.02 sec)

```
mysql> EXPLAIN PARTITIONS SELECT * FROM stuff WHERE
created='2011-01-05'\G
```

```
***** 1. row
*****
```

```
      id: 1
  select_type: SIMPLE
        table: stuff
  partitions: p1
         type: system
possible_keys: NULL
          key: NULL
      key_len: NULL
         ref: NULL
        rows: 1
      Extra:
```

1 row in set (0.08 sec)

```
mysql> EXPLAIN PARTITIONS SELECT * FROM stuff WHERE
created='2012-03-05'\G
```

```
***** 1. row
*****
```

```
      id: 1
  select_type: SIMPLE
        table: stuff
  partitions: p2
         type: ALL
possible_keys: NULL
          key: NULL
      key_len: NULL
         ref: NULL
        rows: 2
      Extra: Using where
```

1 row in set (0.00 sec)

LINEAR HASH

```
CREATE TABLE employees (  
    id INT NOT NULL,  
    fname VARCHAR(30),  
    lname VARCHAR(30),  
    hired DATE NOT NULL DEFAULT '1970-01-01',  
    separated DATE NOT NULL DEFAULT '9999-12-31',  
    job_code INT,  
    store_id INT  
)  
PARTITION BY LINEAR HASH( YEAR(hired) )  
PARTITIONS 4;
```

3.4. KEY分区

按照KEY进行分区类似于按照HASH分区，除了HASH分区使用的用户定义的表达式，而KEY分区的哈希函数是由MySQL 服务器提供。MySQL 簇（Cluster）使用函数MD5()来实现KEY分区；

```
CREATE TABLE tk (  
    col1 INT NOT NULL,  
    col2 CHAR(5),  
    col3 DATE  
)  
PARTITION BY LINEAR KEY (col1)  
PARTITIONS 3;
```

3.5. Subpartitioning

```
CREATE TABLE ts (id INT, purchased DATE)
```

```

PARTITION BY RANGE( YEAR(purchased) )
SUBPARTITION BY HASH( TO_DAYS(purchased) )
SUBPARTITIONS 2 (
    PARTITION p0 VALUES LESS THAN (1990),
    PARTITION p1 VALUES LESS THAN (2000),
    PARTITION p2 VALUES LESS THAN MAXVALUE
);

CREATE TABLE ts1 (id INT, purchased DATE)
PARTITION BY RANGE( YEAR(purchased) )
SUBPARTITION BY HASH( MONTH(purchased) )
SUBPARTITIONS 2 (
    PARTITION p0 VALUES LESS THAN (1990),
    PARTITION p1 VALUES LESS THAN (2000),
    PARTITION p2 VALUES LESS THAN MAXVALUE
);

```

3.6. 分区管理

新增分区

mysql 5.5+

为已经存在表添加分区

```
ALTER TABLE tbl_name ADD PARTITION PARTITIONS 6;
```

新增 RANGE分区

```
ALTER TABLE category ADD PARTITION (PARTITION p4 VALUES IN
(100,200,300,400)
    DATA DIRECTORY = '/data/category'
    INDEX DIRECTORY = '/data/category');
```

新增 LIST分区

```
CREATE TABLE expenses (  
    expense_date DATE NOT NULL,  
    category VARCHAR(30),  
    amount DECIMAL (10,3)  
);  
  
ALTER TABLE expenses  
PARTITION BY LIST COLUMNS (category)  
(  
    PARTITION p01 VALUES IN ( 'lodging', 'food'),  
    PARTITION p02 VALUES IN ( 'flights', 'ground  
transportation'),  
    PARTITION p03 VALUES IN ( 'leisure', 'customer  
entertainment'),  
    PARTITION p04 VALUES IN ( 'communications'),  
    PARTITION p05 VALUES IN ( 'fees')  
);
```

新增 HASH分区

```
CREATE TABLE t1 (  
    id INT,  
    year_col INT  
);  
  
ALTER TABLE t1  
    PARTITION BY HASH(id)  
    PARTITIONS 8;
```

```
/* 在MySQL 5.1中*/  
CREATE TABLE t2  
(  
    dt DATE  
)
```

```

PARTITION BY RANGE (TO_DAYS(dt))
(
  PARTITION p01 VALUES LESS THAN (TO_DAYS('2007-01-01')),
  PARTITION p02 VALUES LESS THAN (TO_DAYS('2008-01-01')),
  PARTITION p03 VALUES LESS THAN (TO_DAYS('2009-01-01')),
  PARTITION p04 VALUES LESS THAN (MAXVALUE));

SHOW CREATE TABLE t2 \G
***** 1. row
*****
      Table: t2
Create Table: CREATE TABLE `t2` (
  `dt` date DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1
/*!50100 PARTITION BY RANGE (TO_DAYS(dt))
(PARTITION p01 VALUES LESS THAN (733042) ENGINE = MyISAM,
PARTITION p02 VALUES LESS THAN (733407) ENGINE = MyISAM,
PARTITION p03 VALUES LESS THAN (733773) ENGINE = MyISAM,
PARTITION p04 VALUES LESS THAN MAXVALUE ENGINE = MyISAM) */

/*在MySQL 5.5中*/
CREATE TABLE t2
(
  dt DATE
)
PARTITION BY RANGE COLUMNS (dt)
(
  PARTITION p01 VALUES LESS THAN ('2007-01-01'),
  PARTITION p02 VALUES LESS THAN ('2008-01-01'),
  PARTITION p03 VALUES LESS THAN ('2009-01-01'),
  PARTITION p04 VALUES LESS THAN (MAXVALUE));

SHOW CREATE TABLE t2 \G
***** 1. row
*****
      Table: t2
Create Table: CREATE TABLE `t2` (
  `dt` date DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1
/*!50500 PARTITION BY RANGE COLUMNS(dt)
(PARTITION p01 VALUES LESS THAN ('2007-01-01') ENGINE =
MyISAM,
PARTITION p02 VALUES LESS THAN ('2008-01-01') ENGINE =

```



```
MyISAM,  
PARTITION p03 VALUES LESS THAN ('2009-01-01') ENGINE =  
MyISAM,  
PARTITION p04 VALUES LESS THAN (MAXVALUE) ENGINE = MyISAM)  
*/
```

删除分区

删除分区 p0

```
ALTER TABLE users DROP PARTITION p0;
```

重建分区

使用 REORGANIZE 重建分区。

RANGE 分区重建

```
ALTER TABLE users REORGANIZE PARTITION p0,p1 INTO (PARTITION  
p0 VALUES LESS THAN (6000000));
```

将原来的 p0,p1 分区合并起来，放到新的 p0 分区中。

LIST 分区重建

```
ALTER TABLE users REORGANIZE PARTITION p0,p1 INTO (PARTITION  
p0 VALUES IN(0,1,4,5,8,9,12,13));
```

将原来的 p0,p1 分区合并起来，放到新的 p0 分区中。

HASH/KEY 分区重建

```
ALTER TABLE users REORGANIZE PARTITION COALESCE PARTITION 2;  
分区的数量改为2,
```

注意：在这里数量只能减少不能增加。想要增加可以用 ADD PARTITION 方法

调整HASH/KEY分区数量，将分区总数扩展到8个。

```
ALTER TABLE users ADD PARTITION PARTITIONS 8;
```

分区维护

重建分区：这和先删除保存在分区中的所有记录，然后重新插入它们，具有同样的效果。它可用于整理分区碎片。

示例：

```
ALTER TABLE t1 REBUILD PARTITION (p0, p1);
```

- **优化分区：**如果从分区中删除了大量的行，或者对一个带有可变长度的行（也就是说，有VARCHAR，BLOB，或TEXT类型的列）作了许多修改，可以使用“ALTER TABLE ... OPTIMIZE PARTITION”来收回没有使用的空间，并整理分区数据文件的碎片。

示例：

```
ALTER TABLE t1 OPTIMIZE PARTITION (p0, p1);
```

在一个给定的分区表上使用“OPTIMIZE PARTITION”等同于在那个分区上运行CHECK PARTITION，ANALYZE PARTITION，和REPAIR PARTITION。

- **分析分区：**读取并保存分区的键分布。

示例：

```
ALTER TABLE t1 ANALYZE PARTITION (p3);
```

- **修补分区：**修补被破坏的分区。

示例：

```
ALTER TABLE t1 REPAIR PARTITION (p0,p1);
```

- **检查分区：**可以使用几乎与对非分区表使用CHECK TABLE 相同的方式检查分区。

示例：

```
ALTER TABLE trb3 CHECK PARTITION (p1);
```

3.7. EXPLAIN PARTITIONS

EXPLAIN PARTITIONS

```
mysql> EXPLAIN PARTITIONS SELECT * FROM users\G
***** 1. row *****
      id: 1
  select_type: SIMPLE
        table: users
  partitions: p0,p1,p2,p3,p4,p5,p6
         type: ALL
possible_keys: NULL
          key: NULL
      key_len: NULL
         ref: NULL
         rows: 7
      Extra:
1 row in set (0.03 sec)

mysql> EXPLAIN PARTITIONS SELECT * FROM users WHERE id < 5\G
***** 1. row *****
      id: 1
  select_type: SIMPLE
        table: users
  partitions: p0,p1,p2,p3,p4,p5,p6
         type: range
possible_keys: PRIMARY
          key: PRIMARY
      key_len: 4
         ref: NULL
         rows: 7
      Extra: Using where
1 row in set (0.00 sec)
```

3.8. SHOW CREATE TABLE

SHOW CREATE TABLE

```

mysql> SHOW CREATE TABLE users\G
***** 1. row *****
      Table: users
Create Table: CREATE TABLE `users` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `username` varchar(20) NOT NULL DEFAULT '',
  `birthday` datetime DEFAULT NULL,
  PRIMARY KEY (`id`,`username`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=latin1
/*!50100 PARTITION BY KEY (id,username)
PARTITIONS 7 */
1 row in set (0.00 sec)

```

3.9. INFORMATION_SCHEMA.partitions 表

```

SELECT
  partition_name part,
  partition_expression expr,
  partition_description descr,
  table_rows
FROM
  INFORMATION_SCHEMA.partitions
WHERE
  TABLE_SCHEMA = schema()
  AND TABLE_NAME='employees';

```

```

select
  partition_name part,
  partition_expression expr,
  from_seconds(partition_description) descr,
  table_rows
FROM

```

```
INFORMATION_SCHEMA.partitions
WHERE
    TABLE_SCHEMA = 'test'
    AND TABLE_NAME='t2';
```

3.10. 分区数据操作

指定分区查询

```
SELECT * FROM employees PARTITION (p0, p2);

SELECT count(1) FROM employees PARTITION (p0);
SELECT count(1) FROM employees PARTITION (p0, p2);
SELECT count(1) FROM employees PARTITION (p0, p2, p1);
```

删除分区中的记录

```
DELETE FROM employees PARTITION (p0, p1);
```

更新指定分区

```
UPDATE employees PARTITION (p0) SET store_id = 2 WHERE fname
= 'Jill';
```

指定分区连表查询

```
SELECT e.id, s.city FROM employees AS e JOIN stores PARTITION
(p1) AS s ...;
```

将某个表迁移到分区上

```
ALTER TABLE employees EXCHANGE PARTITION p0 WITH TABLE  
employees2;
```

4. Index

4.1. SHOW INDEX

```
SHOW INDEX FROM tbl_name
```

垂直显示

```
SHOW INDEX FROM tbl_name\G
```

4.2. CREATE INDEX

```
CREATE INDEX index_name  
ON table_name (column_name)
```

CREATE UNIQUE INDEX

```
CREATE UNIQUE INDEX index_name  
ON table_name (column_name)
```

4.3. DROP INDEX

```
DROP INDEX index_name ON tbl_name
```

4.4. rebuild

```
SHOW INDEX FROM tbl_name  
alter index IND_PK rebuild;
```


5. 外键(Foreign Key)

ON DELETE, ON UPDATE 事件触发限制, 可选参数: RESTRICT | CASCADE | SET NULL | NO ACTION

1. RESTRICT (限制外表中的外键改动)
2. CASCADE (跟随外键改动)
3. SET NULL (设空值)
4. SET DEFAULT (设默认值)
5. NO ACTION (无动作, 默认的)

5.1. FOREIGN KEY (RESTRICT)

```
CREATE TABLE `bank_account_group_has_bank_account` (  
    `id` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,  
    `bank_account_group_id` INT(10) UNSIGNED NOT NULL  
DEFAULT '0',  
    `bank_account_id` INT(10) UNSIGNED NOT NULL DEFAULT  
'0',  
    PRIMARY KEY (`id`),  
    INDEX  
`FK_bank_account_group_has_bank_account_bank_account`  
(`bank_account_id`),  
    INDEX  
`FK_bank_account_group_has_bank_account_bank_account_group`  
(`bank_account_group_id`),  
    CONSTRAINT  
`FK_bank_account_group_has_bank_account_bank_account` FOREIGN  
KEY (`bank_account_id`) REFERENCES `bank_account` (`id`),  
    CONSTRAINT  
`FK_bank_account_group_has_bank_account_bank_account_group`  
FOREIGN KEY (`bank_account_group_id`) REFERENCES  
`bank_account_group` (`id`)  
)
```

```
COMMENT='bank_account_group 与 bank_account 的 N:M 关系'  
COLLATE='utf8_general_ci'  
ENGINE=InnoDB  
AUTO_INCREMENT=35;
```

6. 视图(View)

```
CREATE VIEW view_name AS  
SELECT column_name(s)  
FROM table_name  
WHERE condition
```

update view

```
SQL CREATE OR REPLACE VIEW Syntax  
CREATE OR REPLACE VIEW view_name AS  
SELECT column_name(s)  
FROM table_name  
WHERE condition
```

7. 存储过程(PROCEDURE)

7.1. 存储程序

存储过程没有返回数据，需使用call proc()调用

```
CREATE DEFINER=`neo`@`%` PROCEDURE `angelfund`(IN `puid`
VARCHAR(50), IN `ptime` DATETIME)
LANGUAGE SQL
NOT DETERMINISTIC
CONTAINS SQL
SQL SECURITY DEFINER
COMMENT ''
BEGIN

    DECLARE fusername VARCHAR(16) DEFAULT NULL;
    DECLARE fname VARCHAR(16) DEFAULT NULL;
    DECLARE fmembers_date VARCHAR(20) DEFAULT NULL;

    SELECT username,name,FROM_UNIXTIME(createtime) INTO
    fusername,fname,fmembers_date FROM members WHERE username =
    puid;

    IF fusername IS NOT NULL THEN
        INSERT IGNORE INTO
    angelfund(username,name,members_date,accounts_date,endtime,`st
    atus`,op,operator,`description`)
    value(fusername,fname,fmembers_date,ptime,DATE_ADD(ptime,
    INTERVAL +1 MONTH),'N','N','computer','');
    END IF;

END
```

调用过程

```
call angelfund('100','2013-10-10 10:10:10');
```

7.2. EXECUTE 执行 SQL

在过程中运行SQL，下面的例子是文件导出的例子。

```
DROP procedure IF EXISTS `export_file`;  
  
DELIMITER $$  
CREATE DEFINER=`dba`@`%` PROCEDURE `export_file`(IN file_name  
char(64), IN tabname char(64))  
BEGIN  
    set @sql = concat('SELECT * INTO OUTFILE  
' , "/var/lib/mysql-files/" , file_name , "' FROM ' , tabname) ;  
    -- select @sql;  
    PREPARE stmt FROM @sql;  
    EXECUTE stmt;  
    Deallocate prepare stmt;  
END$$  
  
DELIMITER ;
```

call 存储过程

```
call test.export_file('test', 'mytable');
```

7.3. PREPARE 传递参数

```
mysql> PREPARE stmt1 FROM 'SELECT SQRT(POW(?,2) + POW(?,2)) AS  
hypotenuse';  
Query OK, 0 rows affected (0.00 sec)  
Statement prepared
```

```

mysql> SET @a = 3;
Query OK, 0 rows affected (0.00 sec)

mysql> SET @b = 4;
Query OK, 0 rows affected (0.00 sec)

mysql> EXECUTE stmt1 USING @a, @b;
+-----+
| hypotenuse |
+-----+
|           5 |
+-----+
1 row in set (0.00 sec)

mysql> DEALLOCATE PREPARE stmt1;
Query OK, 0 rows affected (0.00 sec)

mysql>

```

7.4. 存储过程返回数据

```

USE `test`;
DROP procedure IF EXISTS `test`;

DELIMITER $$
USE `test`$$
CREATE DEFINER=`dba`@`%` PROCEDURE `test`(in a int, in b int
,out num int)
BEGIN

    set num = a + b;

END$$

DELIMITER ;

```

运行后返回结果 10

```
set @num = 0;
call test(3,7,@num);
select @num;
```

7.5. 结果集转JSON

```
USE `netkiller`;
DROP procedure IF EXISTS `table2json`;

DELIMITER $$
USE `netkiller`$$
CREATE DEFINER=`neo`@`%` PROCEDURE `table2json`(
IN `schema` VARCHAR(32),
IN `table` VARCHAR(32),
IN `id` VARCHAR(10),
OUT rev VARCHAR(1024)
)
BEGIN
    SET @column = NULL;
    SET @str = NULL;

    SELECT
    GROUP_CONCAT(fields) AS col INTO @column FROM (
        SELECT
        COLUMN_NAME) AS fields
    FROM
    INFORMATION_SCHEMA.Columns
    WHERE
    table_name = `table`
    AND table_schema = `schema`)
```

```

AS tmpatable;

    -- SELECT @column;

    SET @sql = CONCAT('SELECT json_object(',@column, ' )
as json INTO @str FROM ', `table`,` where id = ', `id`);

    -- SELECT @sql;

    PREPARE stmt FROM @sql;
    EXECUTE stmt;
    Deallocate prepare stmt;

    set rev = @str;

END$$

DELIMITER ;

```

使用实例

```

set @rev = '0';
call netkiller.table2json('test', 'test', '1', @rev);
select @rev;

```

7.6. 例子·过程返回结果

```

USE `netkiller`;
DROP procedure IF EXISTS `trigger2json`;

DELIMITER $$
USE `netkiller`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `trigger2json`(

```



```

IN `schema` VARCHAR(32),
IN `table` VARCHAR(32),
OUT rev VARCHAR(1024)
)
BEGIN
    SET @column = NULL;
    SET @str = NULL;

    SELECT
    GROUP_CONCAT(fields) AS col
INTO @column FROM
    (SELECT
        CONCAT('"' , COLUMN_NAME, '"', NEW.' , COLUMN_NAME) AS
fields
    FROM
        INFORMATION_SCHEMA.Columns
    WHERE
        table_name = `table`
        AND table_schema = `schema`) AS tmpTable;

-- SELECT @column;

    SET @sql = CONCAT('SELECT json_object(' , @column, ' )
as json INTO @str ');

    -- SELECT @sql;

    PREPARE stmt FROM @sql;
    EXECUTE stmt;
    Deallocate prepare stmt;

    set rev = @str;

END$$

DELIMITER ;

```

```
set @rev = '0';  
call neo.trigger2json('gw', 'member', @rev);  
select @rev;
```

8. 函数

函数会返回数据，调用函数使用 `select fun()`，不能使用 `call` 调用，否则提示

```
mysql> call myfun();
ERROR 1305 (42000): PROCEDURE test.myfun does not exist
```

下面做一个实验

```
CREATE TABLE `t` (
  `id` INT(11) UNSIGNED NOT NULL AUTO_INCREMENT,
  `n` INT(11) UNSIGNED NULL DEFAULT '0',
  PRIMARY KEY (`id`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
AUTO_INCREMENT=5;

CREATE DEFINER=`neo`@`%` FUNCTION `myfun`()
  RETURNS int(11)
  LANGUAGE SQL
  NOT DETERMINISTIC
  READS SQL DATA
  SQL SECURITY DEFINER
  COMMENT ''
BEGIN
  INSERT INTO t (n) VALUES(rand()*100);
  RETURN LAST_INSERT_ID();
END
```

```
mysql> select myfun();
+-----+
| myfun() |
+-----+
|      9 |
+-----+
1 row in set, 2 warnings (0.07 sec)
```

8.1. TIMESTAMP TO ISO8601

```
USE `netkiller`;
DROP function IF EXISTS `timestamp_to_iso8601`;

DELIMITER $$
USE `netkiller`$$
CREATE DEFINER=`neo`@`db.netkiller.cn` FUNCTION
`timestamp_to_iso8601`(dt timestamp) RETURNS varchar(24) CHARSET utf8
BEGIN

    RETURN DATE_FORMAT( CONVERT_TZ(dt, @@session.time_zone,
'+00:00') , '%Y-%m-%dT%T.000Z' );

END$$

DELIMITER ;
```

调用函数

```
mysql> select timestamp_to_iso8601(current_timestamp()) as iso8601;
+-----+
| iso8601 |
+-----+
| 2017-12-07T07:21:22.000Z |
+-----+
1 row in set (0.00 sec)
```

9. 触发器(Trigger)

9.1. create trigger

Update 更新出发

实现 history 历史表功能，BEFORE update 做到数据库更新自动备份

```
CREATE TABLE user_history SELECT * FROM user WHERE 1 <> 1

DELIMITER //
CREATE TRIGGER user_history BEFORE update ON user FOR EACH ROW
BEGIN
insert into user_history SELECT * FROM user WHERE id = OLD.id;
END; //
DELIMITER ;
```

判断某字段数据修改满足条件后出发。

```
CREATE DEFINER=`dba`@`%` TRIGGER
`cms`.`jc_content_BEFORE_UPDATE` BEFORE UPDATE ON `jc_content`
FOR EACH ROW
BEGIN
    IF NEW.status = '1' THEN
        insert into `neo`.elasticsearch_trash(id)
values(OLD.content_id);
    END IF;
    IF NEW.status = '2' THEN
        delete from `neo`.elasticsearch_trash where id
= OLD.content_id;
    END IF;
END
```

Delete 删除出发

```
CREATE DEFINER=`dba`@`%` TRIGGER
`cms`.`jc_content_BEFORE_DELETE` BEFORE DELETE ON `jc_content`
FOR EACH ROW
BEGIN
    insert into `neo`.elasticsearch_trash(id)
values(OLD.content_id);
END
```

Insert 插入出发

9.2. drop trigger

```
DROP TRIGGER admin_user_history;

DELIMITER //
CREATE TRIGGER admin_user_history BEFORE update ON admin_user
FOR EACH ROW
BEGIN
insert into admin_user_history SELECT * FROM admin_user WHERE
user_id = OLD.user_id;
END; //
DELIMITER;
```

9.3. show triggers

```
show triggers;
```

SHOW CREATE TRIGGER

```
mysql> SHOW CREATE TRIGGER ins_sum\G
***** 1. row *****
      Trigger: ins_sum
      sql_mode:
STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION
SQL Original Statement: CREATE DEFINER=`me`@`localhost` TRIGGER
ins_sum
                        BEFORE INSERT ON account
                        FOR EACH ROW SET @sum = @sum +
NEW.amount
      character_set_client: utf8
      collation_connection: utf8_general_ci
      Database Collation: latin1_swedish_ci
      Created: 2013-07-09 10:39:34.96
```

9.4. EXAMPLE

BEFORE/AFTER

例 22.1. BEFORE/AFTER

```
DROP TRIGGER MY_TEST_MONITOR;
DELIMITER //
CREATE TRIGGER MY_TEST_MONITOR BEFORE insert ON MY_TEST FOR
EACH ROW
BEGIN
    INSERT INTO MY_TEST_MONITOR SELECT * FROM MY_TEST WHERE
TICKET = NEW.TICKET;
END; //
DELIMITER;
```

```

DROP TRIGGER MY_TEST_MONITOR;
DELIMITER //
CREATE TRIGGER MY_TEST_MONITOR AFTER insert ON MY_TEST FOR EACH
ROW
BEGIN
    INSERT INTO MY_TEST_MONITOR SELECT * FROM MY_TEST WHERE
TICKET = NEW.TICKET;
END; //
DELIMITER;

```

通过触发器保护数据，防止重复插入数据

```

CREATE DEFINER=`neo`@`%` TRIGGER `members_before_insert` BEFORE
INSERT ON `members` FOR EACH ROW BEGIN
    IF new.username IS NOT NULL THEN
        IF exists(select m.username from members m
where m.username = new.username) THEN
            set new.username = '';
        END IF;
    END IF;
END

```

UUID

例 22.2. uuid()

```

delimiter $$
CREATE TABLE `member` (
  `uuid` char(36) NOT NULL,
  `username` varchar(20) DEFAULT NULL,
  `password` varchar(32) DEFAULT NULL,
  PRIMARY KEY (`uuid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8$$

CREATE
DEFINER=`root`@`%`

```



```

TRIGGER `test`.`member_before_insert`
BEFORE INSERT ON `test`.`member`
FOR EACH ROW
SET new.uuid = uuid()
$$

```

CALL PROCEDURE

```

CREATE DEFINER=`neo`@`%` TRIGGER `accounts_angelfund` AFTER
INSERT ON `accounts` FOR EACH ROW BEGIN

    IF new.paymode = 'angelfund' THEN
        call angelfund(new.name,new.ctime);
    END IF;

END

CREATE DEFINER=`neo`@`%` PROCEDURE `angelfund`(IN `puid`
VARCHAR(50), IN `ptime` DATETIME)
LANGUAGE SQL
NOT DETERMINISTIC
CONTAINS SQL
SQL SECURITY DEFINER
COMMENT ''
BEGIN

    DECLARE fusername VARCHAR(16) DEFAULT NULL;
    DECLARE fchinese_name VARCHAR(16) DEFAULT NULL;
    DECLARE fmembers_date VARCHAR(20) DEFAULT NULL;

    SELECT username,chinese_name,FROM_UNIXTIME(createtime)
INTO fusername,fchinese_name,fmembers_date FROM members WHERE
username = puid;

    IF fusername IS NOT NULL THEN
        INSERT IGNORE INTO
angelfund(username,chinese_name,members_date,accounts_date,endt
ime,`status`,op,operator,`description`)
value(fusername,fchinese_name,fmembers_date,ptime,DATE_ADD(ptim
e, INTERVAL +1 MONTH),'N','N','computer','');
    END IF;

```

END

10. 事件调度器(EVENT)

10.1. 启用 EVENT

```
set GLOBAL event_scheduler=ON;
```

my.cnf 配置

```
event_scheduler=on
```

查看状态

```
mysql> select @@GLOBAL.event_scheduler;
+-----+
| @@GLOBAL.event_scheduler |
+-----+
| ON                        |
+-----+
1 row in set (0.00 sec)

mysql> SHOW VARIABLES LIKE 'event_scheduler';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| event_scheduler | ON   |
+-----+-----+
1 row in set (0.01 sec)
```



```

-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| netkiller | captcha      | neo@%   | SYSTEM   | RECURRING |
NULL       | 5            | MINUTE  |          | 2013-07-08
16:27:03  | NULL | ENABLED |          | 1 | utf8
| utf8_general_ci | utf8_general_ci |
| netkiller | sms_ips_log | neo@%   | SYSTEM   | RECURRING |
NULL       | '0 5'       | DAY_HOUR |          | 2013-07-09
14:39:51  | NULL | ENABLED |          | 1 | utf8
| utf8_general_ci | utf8_general_ci |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

```

```

mysql> show events \G;
***** 1. row
*****
          Db: netkiller
          Name: captcha
          Definer: neo@%
          Time zone: SYSTEM
          Type: RECURRING
          Execute at: NULL
          Interval value: 5
          Interval field: MINUTE
          Starts: 2013-07-08 16:27:03
          Ends: NULL
          Status: ENABLED
          Originator: 1
character_set_client: utf8
collation_connection: utf8_general_ci
  Database Collation: utf8_general_ci
***** 2. row
*****
          Db: netkiller
          Name: sms_ips_log
          Definer: neo@%
          Time zone: SYSTEM
          Type: RECURRING
          Execute at: NULL
          Interval value: '0 5'
          Interval field: DAY_HOUR

```

```
        Starts: 2013-07-09 14:39:51
          Ends: NULL
        Status: ENABLED
      Originator: 1
character_set_client: utf8
collation_connection: utf8_general_ci
  Database Collation: utf8_general_ci
2 rows in set (0.00 sec)

ERROR:
No query specified
```

10.5. 实例·每月创建一个表

每月创建一张新表，适用于分表的场景

```
CREATE DEFINER=`neo`@`netkiller` EVENT `logging`
  ON SCHEDULE
    EVERY 1 MONTH STARTS '2017-12-11 15:51:00'
  ON COMPLETION PRESERVE
  ENABLE
  COMMENT '每月自动创建表'
DO BEGIN
  declare _table_date varchar(10);
  select date_format(date_add(curdate(),interval 1
month),'%Y%m') into _table_date;
  call logging(_table_date);
END
```

```
CREATE DEFINER=`neo`@`netkiller` PROCEDURE `logging`(
  IN `table_date` VARCHAR(10)
)
LANGUAGE SQL
NOT DETERMINISTIC
```

```

CONTAINS SQL
SQL SECURITY DEFINER
COMMENT ''
BEGIN
    set @_table_name = CONCAT('log_',table_date);
    set @_create = "CREATE TABLE If Not Exists ";
    set @_param = "(
        `id` INT(11) NOT NULL AUTO_INCREMENT,
        `type` VARCHAR(255) NULL DEFAULT NULL
COMMENT '日志类型 1: 网站 2: IOS 3:Android',
        `url` VARCHAR(640) NULL DEFAULT NULL
COMMENT '用户访问url',
        `serverIp` VARCHAR(255) NULL DEFAULT
NULL COMMENT '服务器ip',
        `bodyBytesSent` VARCHAR(255) NULL
DEFAULT NULL,
        `bytesSent` VARCHAR(255) NULL DEFAULT
NULL COMMENT '参数字节数',
        `browser` VARCHAR(255) NULL DEFAULT
NULL COMMENT '浏览器信息',
        `ctime` TIMESTAMP NULL DEFAULT
CURRENT_TIMESTAMP,
        `mtime` TIMESTAMP NULL DEFAULT NULL
ON UPDATE CURRENT_TIMESTAMP,
        PRIMARY KEY (`id`),
        INDEX `ctime` (`ctime`,
`deviceType`,`isFirst`),
        INDEX `userIp` (`userIp`),
        INDEX `deviceId` (`deviceId`),
        INDEX `account` (`account`)
    )
    COMMENT='APP 访问记录'
    COLLATE='utf8_general_ci'
    ENGINE=InnoDB
    ;";

    SET @sql = CONCAT(@_create,@_table_name,@_param);
    PREPARE stmt FROM @sql;
    EXECUTE stmt;
    Deallocate prepare stmt;
END

```

第 23 章 DML (Data Manipulation Language)

```
SELECT - retrieve data from the a database
INSERT - insert data into a table
UPDATE - updates existing data within a table
DELETE - deletes all records from a table, the space for the
records remain
CALL - call a PL/SQL or Java subprogram
EXPLAIN PLAN - explain access path to data
LOCK TABLE - control concurrency
```

1. INSERT

1.1. INSERT INTO ... SELECT

```
SET @OLDTMP_SQL_MODE=@SQL_MODE, SQL_MODE='';
DELIMITER //
CREATE TRIGGER `members_mobile_insert` BEFORE INSERT ON
`members_mobile` FOR EACH ROW BEGIN
    insert into members_location(id,province,city) select
NEW.id,mobile_location.province,mobile_location.city from
mobile_location where mobile_location.id = md5(LEFT(NEW.number,
7));
END//
DELIMITER ;
SET SQL_MODE=@OLDTMP_SQL_MODE;
```

1.2. INSERT IGNORE

INSERT IGNORE 与INSERT INTO的区别就是INSERT IGNORE会忽略数据库中已经存在的数据，如果数据库没有数据，就插入新的数据，如果有数据的话就跳过这条数据。

```
insert ignore into table(name) select name from table2
```


1.3. INSERT...ON DUPLICATE KEY UPDATE

```
create table foo (id serial primary key, u int, unique key
(u));

insert into foo (u) values (10);
insert into foo (u) values (10) on duplicate key update u = 20;

mysql> select * from foo;
+----+-----+
| id | u     |
+----+-----+
|  1 |  20  |
+----+-----+
```

```
DROP TRIGGER IF EXISTS `cms`.`jc_content_BEFORE_DELETE`;

DELIMITER $$
USE `cms`$$
CREATE DEFINER=`5kwords`@`%` TRIGGER `jc_content_BEFORE_DELETE`
BEFORE DELETE ON `jc_content`
FOR EACH ROW BEGIN

    insert into `cms`.elasticsearch_trash(id)
values(OLD.content_id) on duplicate key update ctime = now();
    insert into `cms`.trash(id,`type`, site_id)
values(OLD.content_id, "delete", OLD.site_id) on duplicate key
update `type`="delete", ctime = now();

END$$
DELIMITER ;
```

2. REPLACE

replace 类似 ON DUPLICATE KEY UPDATE，插入过程遇到已经存在的字段，会更新处理。

```
replace into (id) value('1')
```

3. DELETE

3.1. 删除重复数据

```
delete from member group by username having count(username) > 1
```

第 24 章 SQL Statement Syntax

Structured Query Language

1. DISTINCT

```
SELECT DISTINCT user.name FROM user
```

```
SELECT DISTINCT user.name FROM user
```

2. group by

统计重复的手机号吗

```
select * from (select count(mobile) as c, mobile from member
where length(mobile) >= 11 group by mobile) as m where m.c > 1;
```

3. HAVING

```
select * from accounts where paymode='alipay' group by name  
having count(name) >1;
```

4. REGEXP

正则匹配

判断非数字字符

```
select '看89700' regexp '^[0-9]+$'  
select '89看700' regexp '^[0-9]+$'  
select '89700看' regexp '^[0-9]+$'
```

应用到实际工作中

```
select count(*) from accounts a where a.name != '' and not  
a.name regexp '^[0-9]+$';  
select count(*) from accounts a, members m where a.member =  
m.id and a.name != '' and not a.name regexp '^[0-9]+$'  
group by member;  
SELECT * FROM tablename WHERE SUBSTRING(fieldname, 1, 1)  
REGEXP '[[:digit:]]';
```

5. IN / NOT IN

```
select * from members where id in ('1','100','1000');  
select * from members where group_id in (select id from  
members_group);
```


6. ALL / Any

NOT IN 与 \neq ALL 两个语句是相同的:

```
SELECT s1 FROM t1 WHERE s1 <> ALL (SELECT s1 FROM t2);  
SELECT s1 FROM t1 WHERE s1 NOT IN (SELECT s1 FROM t2);
```

IN 与 "= ANY" 两个语句是一样的:

```
SELECT s1 FROM t1 WHERE s1 = ANY (SELECT s1 FROM t2);  
SELECT s1 FROM t1 WHERE s1 IN (SELECT s1 FROM t2);
```

例 24.1. SQL ANY example

```
select * from members where id = any(select members_id from  
accounts where id < 100);
```

7. exists, not exists

```
SELECT c.id, companyname
FROM customers c
WHERE EXISTS(
    SELECT orderid FROM orders o WHERE o.customer_id = cu.id)
```

8. UNION

union 分页问题

```
(SELECT a FROM tbl_name_a WHERE a=10 AND B=1)
UNION
(SELECT a FROM tbl_name_b WHERE a=11 AND B=2)
ORDER BY a LIMIT 10;
```

```
select * from (
    select a from tbl_name_a WHERE a=10 AND B=1
    union all
    select a from tbl_name_b WHERE a=10 AND B=1
) tbl_name
order by a limit 0,1;
```

8.1. UNION ALL

UNION ALL 不会合并重复的记录

```
select a,b from tbl_name_a WHERE a=10 AND B=1
union all
select a,b from tbl_name_b WHERE a=10 AND B=1
```

8.2. 两张表字段不对等解决方法

```
SELECT * FROM
(
    SELECT contract_address, decimals, name, symbol, seq,
```

```
logo FROM token
  UNION
  SELECT contract_address, decimals, name, symbol, 100,
'https://www.netkiller.cn/images/eth.jpg' FROM user_token
WHERE address = '0xB94054c174995AE2A9E7fcf6c7924635FBa8ECF7'
AND contract_address NOT IN (SELECT contract_address FROM
token)
  ) AS tmp
ORDER BY seq
```

9. OUTFILE/LOAD DATA INFILE

查询结果输出到文件

```
SELECT * FROM tablename INTO OUTFILE '/tmp/tablename.txt';
```

使用tee将屏幕输出到文件

```
mysql>tee /home/neo/screen.txt  
mysql>select * from user;  
mysql>exit
```

```
SELECT * INTO OUTFILE '/home/mark/Orders.txt'  
  FIELDS  
  TERMINATED BY = ','  
  FROM Orders  
  WHERE Order_Date >= '2000-01-01'
```

```
LOAD DATA INFILE 'data.txt' INTO TABLE db2.my_table;
```

9.1. Export data to CSV from MySQL

```
SELECT *  
INTO OUTFILE '/tmp/products.csv'  
FIELDS TERMINATED BY ','  
ENCLOSED BY '"'
```

```
ESCAPED BY '\\\  
LINES TERMINATED BY '\\n'  
FROM products
```

9.2. Import data from CSV file.

```
LOAD DATA LOW_PRIORITY LOCAL INFILE 'C:\\hx.csv' IGNORE INTO  
TABLE `tmp`.`creditlog`  
CHARACTER SET gbk FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED  
BY '"' ESCAPED BY '"' LINES TERMINATED BY '\\r\\n'  
(`ctime`, `login`, `mode`, `type`, `prevavailcredit`, `change`,  
`newavailcredit`, `comment`);
```

10. CASE Syntax

```
CASE case_value
  WHEN when_value THEN statement_list
  [WHEN when_value THEN statement_list] ...
  [ELSE statement_list]
END CASE
```

Or:

```
CASE
  WHEN search_condition THEN statement_list
  [WHEN search_condition THEN statement_list] ...
  [ELSE statement_list]
END CASE
```

11. MySQL 专有命令

11.1. SQL_NO_CACHE

```
SELECT /*!40001 SQL_NO_CACHE */ * FROM table
```

11.2. SIGNAL Syntax

```
DROP TRIGGER `members_before_insert`;  
CREATE DEFINER=`neo`@`%` TRIGGER `members_before_insert`  
BEFORE INSERT ON `members` FOR EACH ROW BEGIN  
    IF new.username IS NOT NULL THEN  
        IF not exists(select username from  
members_available where username = new.username) THEN  
            /*set new.username = NULL;*/  
            SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT =  
'An error occurred', MYSQL_ERRNO = 1001;  
        END IF;  
    END IF;  
END;
```


12. SQL 92

insert + select

```
insert into product_type_commission select id,5,1,1,0,0,0,0,0,0
from product_type where title='notebook' and is_physical=0;
```

update table1,table2

```
begin;
ALTER TABLE `customer` ADD COLUMN `cutoff_time` TIMESTAMP NOT
NULL default '0000-00-00 00:00:00';
update customer,agent set customer.cutoff_time =
agent.cutoff_time where customer.id = agent.id;
ALTER TABLE `agent` DROP COLUMN `cutoff_time`;
commit;
```

update table1 set field1 = (select value from table2)

```
UPDATE
    transaction
SET
    transaction.total_sold_price = (
        SELECT
            SUM(transaction_item.price)
        FROM
            transaction_item
            WHERE transaction_item.transaction_id = 100
    )
WHERE
    transaction.id = 100
```

update table1, (select * from other) as table2 set table1.field1 =
table2.field1

```

UPDATE
    transaction,(
        SELECT
SUM(product_item.bought_price) AS total_bought_price,
transaction_item.transaction_id
FROM
transaction_item
WHERE
transaction_item.transaction_id IN ( '123','456' )
) as total
SET
    transaction.total_bought_price =
total.total_bought_price
WHERE
    transaction.id = total.transaction_id

```

join + subquery

```

select u.*,t.category,t.items,t.[property] from
tb_sysregchkusers as u left join (select a.items as category,
b.* from (select id, items from tb_sysregchktask where
categoryid=0) as a left join tb_sysregchktask as b on
b.categoryid=a.id ) as t on u.taskID=t.id

select * from tb_sysregchklog where
CONVERT(datetime,CONVERT(varchar(10),checkTime,120)) between
convert(datetime,'2007-12-12') and convert(datetime,'2007-12-
12')

```

```

select DISTINCT user_point_history.user_id,user.username,
(select count(id) from transaction where id =
user_point_history.transaction_id) as transactions,
(SELECT SUM(u_p_h.points) FROM user_point_history as u_p_h
WHERE u_p_h.type != 'RDMP' AND u_p_h.status IN('pr','ac') AND
u_p_h.user_id = user_point_history.user_id) as
total_points_earned,
(SELECT SUM(u_p_h.points) FROM user_point_history as u_p_h

```

```
WHERE u_p_h.type = 'RDMP' AND u_p_h.status IN('pr','ac') AND
u_p_h.user_id = user_point_history.user_id) as
total_points_redeemed
from user_point_history,user where user_point_history.user_id =
user.id;
```

(total_points_earned - total_points_redeemed) as
current_balance_points

```
select user_id, username, transactions, total_points_earned,
total_points_redeemed, (total_points_earned -
total_points_redeemed) as current_balance_points
from (select DISTINCT user_point_history.user_id,user.username,
(select count(id) from transaction where id =
user_point_history.transaction_id) as transactions,
(SELECT SUM(u_p_h.points) FROM user_point_history as u_p_h
WHERE u_p_h.type != 'RDMP' AND u_p_h.status IN('pr','ac') AND
u_p_h.user_id = user_point_history.user_id) as
total_points_earned,
(SELECT SUM(u_p_h.points) FROM user_point_history as u_p_h
WHERE u_p_h.type = 'RDMP' AND u_p_h.status IN('pr','ac') AND
u_p_h.user_id = user_point_history.user_id) as
total_points_redeemed
from user_point_history,user where user_point_history.user_id =
user.id) as user_performance;
```

subquery作为一个字段使用

```
select product_type_attribute.*, (select 'selected' from
product_type_attribute_set where
product_type_attribute_set.product_type_attribute_id =
product_type_attribute.id and
product_type_attribute_set.product_type_id = 26) as selected
from product_type_attribute;
```

第 25 章 Functions and Operators

1. COUNT

count()

```
SELECT (SELECT count(1) FROM ecs_category) as 'Export category  
count',  
       (SELECT count(1) FROM ecs_goods) as 'Goods count',  
       (SELECT count(1) FROM ecs_goods_attr) as 'Attr count';
```

2. group_concat() 列传行

```
SELECT tags FROM neo.article;  
  
linux  
redis  
mysql  
java  
php
```

tags字段专为一行显示

```
SELECT group_concat(tags) FROM neo.article;  
  
linux,redis,mysql,java,php
```

distinct 去除重复数据

```
select group_concat(distinct author) from neo.article;
```

以id分组，把name字段的值打印在一行，分号分隔

```
select id,group_concat(tags separator ';') from neo.article  
group by tags;
```

排序结果

```
select group_concat(distinct author order by author desc) from  
neo.article;
```

3. UUID()

```
SELECT UUID(),LENGTH(UUID()),UUID_SHORT(),  
LENGTH(UUID_SHORT());
```

4. String

4.1. LEFT/RIGHT

LEFT(str,len)

```
mysql> select left(concat('1','0000000'),5) as number;
+-----+
| number |
+-----+
| 10000  |
+-----+
1 row in set (0.00 sec)
```

RIGHT(str,len)

```
mysql> select right(concat('0000000','1'),5) as number;
+-----+
| number |
+-----+
| 00001  |
+-----+
1 row in set (0.00 sec)
```

4.2. RPAD/LPAD

补齐长度用'0'填充

RPAD(str,len,padstr)

```
mysql> select rpad('10',5,'0') as txt;
+-----+
| txt   |
+-----+
| 10000 |
+-----+
1 row in set (0.01 sec)
```

LPAD(str,len,padstr)

```
mysql> select lpad('10',5,'0') as txt;
+-----+
| txt   |
+-----+
| 00010 |
```

```
+-----+
1 row in set (0.00 sec)
```

4.3. CONCAT

CONCAT(str1,str2,...)

```
mysql> select concat('Neo',' ','Chen') as Name;
+-----+
| Name      |
+-----+
| Neo Chen  |
+-----+
1 row in set (0.00 sec)
```

4.4. CONCAT_WS

```
SELECT CONCAT_WS(',','Neo','Chen');
Neo,Chen

SELECT CONCAT_WS('-', 'Neo', 'Chen');
Neo-Chen
```

使用逗号链接字符串

```
SELECT
  CONCAT_WS(',','id,name,age)
FROM
  mytable
```

4.5. 链接所有字段

当我使用 `select CONCAT_WS(",",*) as string from tab` 时发现不支持 * 操作。

解决方案如下

```
SET @column = NULL;
```



```

SELECT
    GROUP_CONCAT(COLUMN_NAME) AS fields INTO @column
FROM
    INFORMATION_SCHEMA.Columns
WHERE
    table_name = 'mytable'
    AND table_schema = 'test';

-- select @column;

SET @sql = CONCAT('SELECT CONCAT_WS(",",',@column, ' ) FROM mytable');

select @sql;

PREPARE stmt FROM @sql;
EXECUTE stmt;
DEALLOCATE PREPARE stmt;

```

4.6. GROUP_CONCAT

```

mysql> select GROUP_CONCAT(CONVERT( username , CHAR (16)) order by username
desc) as username from test;
+-----+
| username |
+-----+
| jam,jam2,john,john2,john3,neo,neo1,neo2 |
+-----+
6 rows in set, 1 warning (0.01 sec)

```

4.7. replace

```

select replace(goods_desc,':8000','') from ecs_goods;

update ecs_goods set goods_desc=replace(goods_desc,':8000','');

```

4.8. SUBSTRING

```

mysql> SELECT SUBSTRING('netkiller',4,4);
+-----+
| SUBSTRING('netkiller',4,4) |
+-----+
| kill |
+-----+
1 row in set (0.00 sec)

```

与left,right 相同的用法

```
select right('M2014030615410572307:DEPOSIT', 7);
SELECT SUBSTRING('M2014030615410572307:DEPOSIT', -7);
```

4.9. SUBSTRING_INDEX

```
SELECT SUBSTRING_INDEX('M2014030615410572307:DEPOSIT', ':', -1);
SELECT SUBSTRING_INDEX('M2014030615410572307:DEPOSIT', ':', 1);
```

4.10. AES_ENCRYPT / AES_DECRYPT

简单用法

```
mysql> select AES_ENCRYPT('helloworld','key');
+-----+
| AES_ENCRYPT('helloworld','key') |
+-----+
|                                  |
+-----+
1 row in set (0.00 sec)

mysql> select AES_DECRYPT(AES_ENCRYPT('helloworld','key'),'key');
+-----+
| AES_DECRYPT(AES_ENCRYPT('helloworld','key'),'key') |
+-----+
| helloworld                                         |
+-----+
1 row in set (0.00 sec)

mysql>
```

加密数据入库

```
CREATE TABLE `encryption` (
  `mobile` VARBINARY(16) NOT NULL,
  `key` VARCHAR(32) NOT NULL
)
ENGINE=InnoDB;

INSERT INTO encryption(`mobile`,`key`)VALUES(
```

```
AES_ENCRYPT('13691851789',md5('13691851789')), md5('13691851789'))  
select AES_DECRYPT(mobile,`key`), length(mobile) from encryption;
```

5. Date and Time

```
SELECT NOW(),CURRENT_TIMESTAMP(),SYSDATE();
```

5.1. year/month/day hour:minute:second

```
mysql> select year('2012-03-20');
```

```
+-----+
| year('2012-03-20') |
+-----+
|                2012 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> select month('2012-03-20');
```

```
+-----+
| month('2012-03-20') |
+-----+
|                    3 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> select day('2012-03-20');
```

```
+-----+
| day('2012-03-20') |
+-----+
|                   20 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> select hour('12:30:55');
```

```
+-----+
| hour('12:30:55') |
+-----+
|                  12 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> select minute('12:30:55');
+-----+
| minute('12:30:55') |
+-----+
|                    30 |
+-----+
1 row in set (0.00 sec)

mysql> select second('12:30:55');
+-----+
| second('12:30:55') |
+-----+
|                    55 |
+-----+
1 row in set (0.00 sec)
```

5.2. Unix time

语法: FROM_UNIXTIME(unix_timestamp,format)

返回表示 Unix 时间标记的一个字符串，根据format字符串格式化。format可以包含与DATE_FORMAT()函数列出的条目同样的修饰符。根据format字符串格式化date值。

下列修饰符可以被用在format字符串中：

%M 月名字(January.....December)

%W 星期名字(Sunday.....Saturday)

%D 有英语前缀的月份的日期(1st, 2nd, 3rd, 等等。)

%Y 年, 数字, 4 位

%y 年, 数字, 2 位

%a 缩写的星期名字(Sun.....Sat)

%d 月份中的天数, 数字(00.....31)

%e 月份中的天数, 数字(0.....31)

%m 月, 数字(01.....12)

%c 月, 数字(1.....12)

%b 缩写的月份名字(Jan.....Dec)

%j 一年中的天数(001.....366)
 %H 小时(00.....23)
 %k 小时(0.....23)
 %h 小时(01.....12)
 %I 小时(01.....12)
 %l 小时(1.....12)
 %i 分钟, 数字(00.....59)
 %r 时间,12 小时(hh:mm:ss [AP]M)
 %T 时间,24 小时(hh:mm:ss)
 %S 秒(00.....59)
 %s 秒(00.....59)
 %p AM或PM
 %w 一个星期中的天数(0=Sunday6=Saturday)
 %U 星期(0.....52), 这里星期天是星期的第一天
 %u 星期(0.....52), 这里星期一是星期的第一天
 %% 一个文字“%”。

```

mysql> SELECT UNIX_TIMESTAMP('2005-03-27 02:00:00');
+-----+
| UNIX_TIMESTAMP('2005-03-27 02:00:00') |
+-----+
|                               1111885200 |
+-----+
mysql> SELECT FROM_UNIXTIME(1111885200);
+-----+
| FROM_UNIXTIME(1111885200) |
+-----+
| 2005-03-27 03:00:00      |
+-----+
  
```

```

SELECT UNIX_TIMESTAMP('2012-01-01 00:00:00');
SELECT UNIX_TIMESTAMP('2012-07-30 00:00:00');
SELECT UNIX_TIMESTAMP();
  
```

```

SELECT UNIX_TIMESTAMP('2009-08-06') ;
SELECT UNIX_TIMESTAMP( curdate( ) );

select FROM_UNIXTIME(UNIX_TIMESTAMP('2012-07-30 00:00:00'),
'%Y-%m-%d');
SELECT FROM_UNIXTIME( 1249488000, '%Y年%m月%d日' );

SELECT FROM_UNIXTIME(time_stamp, '%Y-%m-%d %H:%i:%S') FROM
test.transaction_history;

select FROM_UNIXTIME(createtime, '%m') as month, count(1) as
count from members where createtime BETWEEN
UNIX_TIMESTAMP('2012-01-01 00:00:00') and
UNIX_TIMESTAMP('2012-12-31 00:00:00') group by
FROM_UNIXTIME(createtime, '%m');
select FROM_UNIXTIME(createtime, '%m') as month, count(1) as
count from members where createtime BETWEEN
UNIX_TIMESTAMP('2011-01-01 00:00:00') and
UNIX_TIMESTAMP('2011-12-31 00:00:00') group by
FROM_UNIXTIME(createtime, '%m');

select FROM_UNIXTIME(createtime, '%m-%d') as month, count(1)
as count from members where createtime BETWEEN
UNIX_TIMESTAMP('2011-01-01 00:00:00') and
UNIX_TIMESTAMP('2011-12-31 00:00:00') group by
FROM_UNIXTIME(createtime, '%m-%d');
select FROM_UNIXTIME(createtime, '%m-%d') as month, count(1)
as count from members where createtime BETWEEN
UNIX_TIMESTAMP('2012-01-01 00:00:00') and
UNIX_TIMESTAMP('2012-12-31 00:00:00') group by
FROM_UNIXTIME(createtime, '%m-%d');

```

5.3. DATE_FORMAT

DATE_FORMAT() 函数用于以不同的格式显示日期/时间数据。

语法

DATE_FORMAT(date, format)

date 参数是合法的日期。format 规定日期/时间的输出格式。

可以使用的格式有：

格式	描述
%a	缩写星期名
%b	缩写月名
%c	月, 数值
%D	带有英文前缀的月中的天
%d	月的天, 数值(00-31)
%e	月的天, 数值(0-31)
%f	微秒
%H	小时 (00-23)
%h	小时 (01-12)
%I	小时 (01-12)
%i	分钟, 数值(00-59)
%j	年的天 (001-366)
%k	小时 (0-23)
%l	小时 (1-12)
%M	月名
%m	月, 数值(00-12)
%p	AM 或 PM
%r	时间, 12-小时 (hh:mm:ss AM 或 PM)
%S	秒(00-59)
%s	秒(00-59)
%T	时间, 24-小时 (hh:mm:ss)
%U	周 (00-53) 星期日是一周的第一天
%u	周 (00-53) 星期一是一周的第一天
%V	周 (01-53) 星期日是一周的第一天, 与 %X 使用
%v	周 (01-53) 星期一是一周的第一天, 与 %x 使用
%W	星期名
%w	周的天 (0=星期日, 6=星期六)
%X	年, 其中的星期日是周的第一天, 4 位, 与 %V 使用
%x	年, 其中的星期一是周的第一天, 4 位, 与 %v 使用
%Y	年, 4 位
%y	年, 2 位

实例

下面的脚本使用 `DATE_FORMAT()` 函数来显示不同的格式。我们使用 `NOW()` 来获得当前的日期/时间：

```
DATE_FORMAT(NOW(), '%b %d %Y %h:%i %p')
DATE_FORMAT(NOW(), '%m-%d-%Y')
DATE_FORMAT(NOW(), '%d %b %y')
DATE_FORMAT(NOW(), '%d %b %Y %T:%f')
```



```
SELECT DATE_FORMAT(NOW(), '%Y-%m-%d');
```

```
select DATE_FORMAT(asctime, '%Y-%m-%d') as Date, count(1) as  
Count from logging where tag='www' and facility='login' group  
by DATE_FORMAT(asctime, '%Y-%m-%d') order by asctime desc;
```

5.4. DATE_SUB/DATE_ADD

当前时间向后推10天

```
mysql> select DATE_SUB(now(), INTERVAL 240 HOUR);
```

```
+-----+  
| DATE_SUB(now(), INTERVAL 240 HOUR) |  
+-----+  
| 2012-03-09 10:26:03 |  
+-----+
```

```
1 row in set (0.00 sec)
```

```
mysql> select DATE_SUB(now(), INTERVAL 24 HOUR);
```

```
+-----+  
| DATE_SUB(now(), INTERVAL 24 HOUR) |  
+-----+  
| 2012-03-18 10:28:43 |  
+-----+
```

```
1 row in set (0.00 sec)
```

```
DELETE from Message where created < DATE_sub(now(), INTERVAL  
240 HOUR);
```

```
select * from PRICES_HISTORY where DATE_FORMAT(TIME  
, GET_FORMAT(DATE, 'ISO')) = (  
select if ( WEEKDAY(CURRENT_DATE())=6 ,  
DATE_SUB(CURRENT_DATE(), INTERVAL 1 DAY) , CURRENT_DATE()  
)
```

DATE_ADD

```
SELECT DATE_ADD('1998-01-02', INTERVAL 31 DAY);
```

5.5. datediff / timediff

计算时间差，两个时间相减结果

```
mysql> select timediff('22:20:00','17:30:00');
```

```
+-----+
| timediff('22:20:00','17:30:00') |
+-----+
| 04:50:00 |
+-----+
```

```
1 row in set (0.00 sec)
```

```
mysql> select datediff('2008-08-08 12:00:00', '2008-08-01
00:00:00');
```

```
+-----+
| datediff('2008-08-08 12:00:00', '2008-08-01 00:00:00') |
+-----+
| 7 |
+-----+
```

```
1 row in set (0.00 sec)
```

6. 数值函数

6.1. cast 类型转换

```
mysql> SELECT cast(SUBSTRING('123456789',1,4) as UNSIGNED) *
100;
+-----+
| cast(SUBSTRING('123456789',1,4) as UNSIGNED) * 100 |
+-----+
|                                                    123400 |
+-----+
1 row in set (0.00 sec)
```

6.2. truncate 保留小数位数

```
select profit, deficit, concat(truncate((profit /
deficit)*100,2),'%') as percentage from ((select count(*) as
profit from angelfund where profit > 0) as profit, (select
count(*) as deficit from angelfund where profit < 0) as
deficit);
```

6.3. MOD 求余

```
mysql> select 9 mod 5;
+-----+
| 9 mod 5 |
+-----+
|         4 |
+-----+
```

```
1 row in set (0.00 sec)
```

```
mysql> select mod(5,2);
```

```
+-----+
```

```
| mod(5,2) |
```

```
+-----+
```

```
|          1 |
```

```
+-----+
```

```
1 row in set (0.00 sec)
```

```
mysql> select mod(5,2);
```

7. Control Flow Functions

CASE

```
mysql> SELECT CASE 1 WHEN 1 THEN 'one'
->      WHEN 2 THEN 'two' ELSE 'more' END;
-> 'one'
mysql> SELECT CASE WHEN 1>0 THEN 'true' ELSE 'false' END;
-> 'true'
mysql> SELECT CASE BINARY 'B'
->      WHEN 'a' THEN 1 WHEN 'b' THEN 2 END;
-> NULL
```

IFNULL

```
mysql> SELECT IFNULL("TEST", 'OK');
+-----+
| IFNULL("TEST", 'OK') |
+-----+
| TEST                  |
+-----+
1 row in set (0.00 sec)

mysql> SELECT IFNULL(NULL, 'OK');
+-----+
| IFNULL(NULL, 'OK') |
+-----+
| OK                  |
+-----+
1 row in set (0.00 sec)
```

NULLIF()

IF

```
mysql> SELECT IFNULL("TEST", 'OK');
```

```
+-----+  
| IFNULL("TEST", 'OK') |  
+-----+  
| TEST                  |  
+-----+
```

```
1 row in set (0.00 sec)
```

```
mysql> SELECT IFNULL(NULL, 'OK');
```

```
+-----+  
| IFNULL(NULL, 'OK') |  
+-----+  
| OK                  |  
+-----+
```

```
1 row in set (0.00 sec)
```

第 26 章 DCL (Data Control Language)

```
COMMIT - save work done
SAVEPOINT - identify a point in a transaction to which you can
later roll back
ROLLBACK - restore database to original since the last COMMIT
SET TRANSACTION - Change transaction options like what rollback
segment to use
```

1. 锁

锁机制

1) 共享锁：由读表操作加上的锁，加锁后其他用户只能获取该表或行的共享锁，不能获取排它锁，也就是说只能读不能写
2) 排它锁：由写表操作加上的锁，加锁后其他用户不能获取该表或行的任何锁，典型是mysql事务中的

锁的范围：

行锁：对某行记录加上锁

表锁：对整个表加上锁

共享锁(share mode), 排他锁(for update)

1.1. 共享锁

1.2. 排他锁

下面做作一个实验，验证锁的效果

终端一,首先进入事务状态然后运行下面语句

```
mysql> begin;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from t1 where id='3' for update;
+----+-----+-----+-----+
| id | name  | ctime                | mtime                |
+----+-----+-----+-----+
|  3 | test  | 0000-00-00 00:00:00 | 2013-01-14 13:05:41 |
+----+-----+-----+-----+
1 row in set (0.00 sec)
```

终端二, 查询表中数据

```
mysql> select * from t1;
+----+-----+-----+-----+
| id | name  | ctime                | mtime                |
+----+-----+-----+-----+
|  1 | neo   | 0000-00-00 00:00:00 | 2013-01-14 13:00:00 |
|  2 | zen   | 0000-00-00 00:00:00 | 2013-01-14 13:00:43 |
|  3 | test  | 0000-00-00 00:00:00 | 2013-01-14 13:05:41 |
+----+-----+-----+-----+
3 rows in set (0.00 sec)
```

增加“for update”查询非锁定记录

```
mysql> select * from t1 where id=2 for update;
+----+-----+-----+-----+
| id | name  | ctime                | mtime                |
+----+-----+-----+-----+
|  2 | zen   | 0000-00-00 00:00:00 | 2013-01-14 13:00:43 |
+----+-----+-----+-----+
1 row in set (0.00 sec)
```

查询被锁定记录


```
mysql> select * from t1 where id=3 for update;
ERROR 1205 (HY000): Lock wait timeout exceeded; try restarting
transaction
```

查询所有记录，因为记录中包含了id=3那条，所以也不允许查询。

```
mysql> select * from t1 for update;
ERROR 1205 (HY000): Lock wait timeout exceeded; try restarting
transaction
```

测试修改记录

```
mysql> UPDATE `t1` SET `name`='testaa' WHERE `id`=3;
ERROR 1205 (HY000): Lock wait timeout exceeded; try restarting
transaction
```

提示

在没有出现ERROR 1205 (HY000)的这段时间，只要终端一中执行commit,rollback锁就释放了.终端二中的语句就会运行。

select trx_query from information_schema.innodb_trx; 可以查看被锁的SQL语句

1.3. 锁

表的加锁与解锁

```
LOCK TABLES tablename WRITE;
LOCK TABLES tablename READ;

...
...

UNLOCK TABLES;
```

```
CREATE TABLE `locking` (
  `name` VARCHAR(50) NULL DEFAULT NULL
)
ENGINE=InnoDB
;

mysql> insert into locking values('test');
Query OK, 1 row affected (0.02 sec)

mysql> select * from locking;
+-----+
| name |
+-----+
| test |
+-----+
1 row in set (0.00 sec)

mysql> UNLOCK TABLES;
```

```
mysql> LOCK TABLES locking READ;
Query OK, 0 rows affected (0.00 sec)

mysql> insert into locking values('test');
ERROR 1099 (HY000): Table 'locking' was locked with a READ lock
and can't be updated

mysql> LOCK TABLE locking WRITE;
```

```
Query OK, 0 rows affected (0.00 sec)

mysql> select * from locking;
+-----+
| name |
+-----+
| test |
| test |
+-----+
2 rows in set (0.00 sec)

mysql> insert into locking values('test');
Query OK, 1 row affected (0.05 sec)

mysql> UNLOCK TABLES;
```

禁止查询

```
mysql> LOCK TABLE locking AS myalias READ;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from locking;
ERROR 1100 (HY000): Table 'locking' was not locked with LOCK
TABLES

mysql> select * from locking as myalias;
+-----+
| name |
+-----+
| test |
| test |
| test |
+-----+
3 rows in set (0.00 sec)
```

1.4. 锁等待与超时

当你开启了事务 begin 忘记，或者各种原因没有commit也没有rollback。悲剧了！

超时设置

```
begin;
SET SESSION wait_timeout = 60;
select * from locking for update;
```

60秒内如果没有commit/rollback将自动释放本次事务。

select for update nowait

使用 for update 是会遇到一个问题，就是其他用户会漫长的等待，而我们需要程序非阻塞运行，当遇到 for update 的时候应该立即返回此表已被加锁。

mysql 并没有实现 nowait 关键字（类似Oracle的功能），但又一个方法能够达到同样目的。

```
mysql> select @@innodb_version;
+-----+
| @@innodb_version |
+-----+
| 5.6.24           |
+-----+
1 row in set (0.05 sec)

mysql> select * from locking;
ERROR 1100 (HY000): Table 'locking' was not locked with LOCK TABLES
```

此时需要等待很长时间才能提示 “Table 'locking' was not locked with LOCK TABLES”

```
mysql> set session innodb_lock_wait_timeout=1;
Query OK, 0 rows affected (0.02 sec)

mysql> select * from locking for update;
ERROR 1205 (HY000): Lock wait timeout exceeded; try restarting
transaction
```

设置 innodb_lock_wait_timeout 参数后，很快就返回

```
mysql> show variables like 'innodb_lock_wait_timeout';
+-----+-----+
| Variable_name          | Value |
+-----+-----+
| innodb_lock_wait_timeout | 1     |
+-----+-----+
1 row in set (0.00 sec)

mysql> show global variables like 'innodb_lock_wait_timeout';
+-----+-----+
| Variable_name          | Value |
+-----+-----+
| innodb_lock_wait_timeout | 50    |
+-----+-----+
1 row in set (0.00 sec)
```

innodb_lock_wait_timeout 默认值是 50

2. 事务处理和锁定语句

Transactional and Locking Statements

开始事务 begin、start transaction 或者 set autocommit=0

事务的特征：原子性 (Atomicity)、一致性 (Consistency)、隔离性 (Isolation) 和持久性 (Durability)，这四个特性简称ACID特性。

原子性：事务是数据库的逻辑工作单位，事务中包括的所有操作要么都做，要么都不做。

一致性：事务执行的结果必须是使数据库从一个一致性的状态变到另外一个一致性状态。

隔离性：一个事务的执行不能被其他事务干扰。即一个事务内部的操作及使用的数据对其他事务是隔离的，并发执行的各个事务之间互相不干扰。

持久性：一个事务一旦成功提交，对数据库中数据的修改就是持久性的。接下来其他的其他操作或故障不应该对其执行结果有任何影响。

2.1. 事务隔离级别

事务隔离模式

1) READ UNCOMMITTED

SELECT的时候允许脏读，即SELECT会读取其他事务修改而还没有提交的数据。

2) READ COMMITTED

SELECT的时候无法重复读，即同一个事务中两次执行同样的查询语句，若在第一次与第二次查询之间时间段，其他事务又刚好修改了其查询的数据且提交了，则两次读到的数据不一致。

3) REPEATABLE READ

SELECT的时候可以重复读，即同一个事务中两次执行同样的查询语句，得到的数据始终都是一致的。实现的原理是，在一个事务对数据行执行读取或写入操作时锁定了这些数据行。

但是这种方式又引发了幻想读的问题。因为只能锁定读取或写入的行，不能阻止另一个事务插入数据，后期执行同样的查询会产生更多的结果。

4) SERIALIZABLE

与可重复读的唯一区别是，默认把普通的SELECT语句改成SELECT ... LOCK IN

SHARE MODE。即为查询语句涉及到的数据加上共享锁，阻塞其他事务修改真实数据。serializable模式中，事务被强制为依次执行。这是SQL标准建议的默认行为。

可以通过下列语句查询全局和当前会话的事务隔离级别：

```
SELECT @@global.tx_isolation;
SELECT @@tx_isolation;
```

查看InnoDB系统级别的事务隔离级别：

```
mysql> SELECT @@global.tx_isolation;
```

查看InnoDB会话级别的事务隔离级别：

```
mysql> SELECT @@tx_isolation;
```

修改InnoDB系统级别的事务隔离级别：

```
mysql> set global transaction isolation level read committed;
```

修改InnoDB会话级别的事务隔离级别：

```
mysql> set session transaction isolation level read committed;
```

2.2. 事务所用到的表

information_schema

```
select * from innodb_trx;
select * from innodb_lock_waits;
select * from innodb_locks;
```

2.3. 解决更新冲突

```
CREATE TABLE `account` (
  `id` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,
```

```

`user` VARCHAR(50) NOT NULL DEFAULT '0',
`cash` FLOAT NOT NULL DEFAULT '0',
`point` INT(10) UNSIGNED NOT NULL DEFAULT '0',
PRIMARY KEY (`id`),
UNIQUE INDEX `user` (`user`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB;

```

```

INSERT INTO `test`.`account` (`user`, `cash`, `point`) VALUES
('neo', 10, 10);

```

下面通过account表，我来模拟一个返点场景，例如电商网站经常会用到“返点”，购买一定数量的商品赠送一定的点数，可以通过点数买东西，这样涉及到点的加于减操作。

表 26.1. 更新丢失演示

Session A	Session B
<pre>select point into @point from account where user='neo';</pre>	
	<pre>select point into @point from account where user='neo';</pre>
<pre>update account set point=@point+20 where user='neo';</pre>	


```
update account set
point=@point+50 where
user='neo';
```

看看最后用户有多少点?

```
mysql> select point from account where user='neo';
+-----+
| point |
+-----+
|     30 |
+-----+
1 row in set (0.00 sec)
```

傻了吧，老板发火，测试不能重现，运维说这是程序计算错误，程序员说程序没有错误，这样的场景国内很多公司都出现过吧？

问题出在哪里呢？出在并发上，很多web程序员很少考虑并发是产生的问题，怎么解决？很多方案，在我的职业生涯过程就见过很多奇葩方案，都能解决问题但不太完美。

如果更新语句改为 `update account set point=@point+50 where user='neo' and point=@point;` 会更保险,但仍然不能解决同意时间所产生的更新操作

下面是通过事务与锁彻底解决上面的问题。

```
mysql> SELECT @@tx_isolation;
+-----+
| @@tx_isolation |
+-----+
| REPEATABLE-READ |
```

```
+-----+
1 row in set (0.00 sec)
```

检查事务隔离级别为: REPEATABLE-READ

表 26.2. 防止更新丢失加锁演示

Session A	Session B
<pre>begin; select point into @point from account where user='neo' for update;</pre>	
	<pre>begin; select point into @point from account where user='neo' for update;</pre>
	执行到此处会挂起
<pre>update account set point=@point+20 where user='neo'; commit;</pre>	
	<pre>update account set point=@point+50 where user='neo';</pre>

```
commit;
```

上面解决更新覆盖问题，但从数据库设计角度是不应该这样设计表的。仅供参考

```
CREATE TABLE `account` (  
  `id` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,  
  `user` VARCHAR(50) NOT NULL DEFAULT '0',  
  `cash` FLOAT NOT NULL DEFAULT '0',  
  `point` INT(10) NOT NULL DEFAULT '0',  
  PRIMARY KEY (`id`)  
)  
COLLATE='utf8_general_ci'  
ENGINE=InnoDB;
```

每一次数据变化新增一条数据

```
INSERT INTO `test`.`account` (`user`, `point`) VALUES ('neo',  
-10);  
INSERT INTO `test`.`account` (`user`, `point`) VALUES ('neo',  
-5);  
INSERT INTO `test`.`account` (`user`, `point`) VALUES ('neo',  
30);  
INSERT INTO `test`.`account` (`user`, `point`) VALUES ('neo',  
-20);
```

计算剩余点数

```
select sum(point) as point from account where user='neo';
```

2.4. 共享锁

先加上共享锁，此时也会对mytable表加上IS锁

```
begin;  
select * from mytable where id=1 for share;
```

观察IS锁

```
select * from performance_schema.data_locks;
```

2.5. SAVEPOINT

```
DROP PROCEDURE IF EXISTS doOrder;  
  
DELIMITER $$  
  
CREATE PROCEDURE doOrder(IN orderUUID VARCHAR(40))  
  BEGIN  
    DECLARE EXIT HANDLER FOR SQLEXCEPTION ROLLBACK TO  
    sp_order;  
  
    START TRANSACTION;  
    SAVEPOINT sp_order;  
  
    -- doing my updates and selects here...
```

```
COMMIT;
```

```
END $$
```

```
DELIMITER ;
```

第 27 章 Optimization

1. Limit 状态

```
$ sudo cat /proc/`pidof mysqld`/limits
Limit                Soft Limit           Hard Limit
Units
Max cpu time         unlimited            unlimited
seconds
Max file size        unlimited            unlimited
bytes
Max data size        unlimited            unlimited
bytes
Max stack size       10485760             unlimited
bytes
Max core file size   0                    unlimited
bytes
Max resident set     unlimited            unlimited
bytes
Max processes        62662                62662
processes
Max open files       20480                 20480
files
Max locked memory    65536                 65536
bytes
Max address space    unlimited             unlimited
bytes
Max file locks       unlimited            unlimited
locks
Max pending signals  62662                62662
signals
Max msgqueue size   819200                819200
bytes
Max nice priority    0                     0
Max realtime priority 0                     0
Max realtime timeout unlimited              unlimited
us
```

2. 使用 Btrfs 文件系统存储mysql数据

```
#!/bin/sh
systemctl stop mysqld

btrfs subvolume create /srv/@mysql
btrfs subvolume list /srv/

UUID=$(blkid | grep btrfs | sed -e 's/.*UUID="\
([\^"]*\)"\.*\/\1/' )
# UUID=786f570d-fe5c-4d5f-832a-c1b0963dd4e6 /srv btrfs defaults
1 1
cat << EOF >> /etc/fstab
UUID=${UUID} /var/lib/mysql btrfs
noatime,nodiratime,subvol=@mysql 0 2
EOF

mkdir /tmp/mysql
mv /var/lib/mysql/* /tmp/mysql/

mount /var/lib/mysql/
chown mysql:mysql /var/lib/mysql

mv /tmp/mysql/* /var/lib/mysql/

systemctl start mysqld
```

3. 打开表的数量

```
mysql> SHOW STATUS LIKE 'open%tables';
```

Variable_name	Value
Open_tables	100
Opened_tables	1

2 rows in set (0.04 sec)

4. Buffering and Caching

查看缓存是否开启

```
MySQL> select @@query_cache_type;  
MySQL> show variables like 'query_cache_type';
```

开启与关闭缓存

```
MySQL> set query_cache_type=on;  
MySQL> set query_cache_type=off;
```

查看缓存状态

```
show variables like 'have_query_cache';
```

查询缓存的大小

```
MySQL> select @@global.query_cache_size;  
MySQL> select @@query_cache_size;
```

查看最大缓存限制，如果集大于该数则不缓存。

```
MySQL> select @@global.query_cache_limit;
```

清除缓存/重置缓存

```
MySQL> flush tables;  
MySQL> flush query cache;  
MySQL> reset query cache;
```

查询缓存性能

```
MySQL> show status like 'qcache%';  
  
MySQL> show status like 'qcache_q%';  
+-----+-----+  
| Variable_name | Value |  
+-----+-----+  
| Qcache_queries_in_cache | 1 |  
+-----+-----+  
1 row in set (0.00 sec)  
  
MySQL> show status like 'qcache_f%';  
+-----+-----+  
| Variable_name | Value |  
+-----+-----+  
| Qcache_free_blocks | 1 |  
| Qcache_free_memory | 16766728 |  
+-----+-----+  
2 rows in set (0.00 sec)
```

4.1. Query Cache SELECT Options

Two query cache-related options may be specified in SELECT statements:

SQL_CACHE

The query result is cached if it is cacheable and the value of the `query_cache_type` system variable is ON or DEMAND.

SQL_NO_CACHE

The query result is not cached.

Examples:

```
SELECT SQL_CACHE id, name FROM customer;
SELECT SQL_NO_CACHE id, name FROM customer;

SELECT /*! SQL_NO_CACHE */ stuff FROM table
```

例 27.1. SQL_CACHE 测试

下面的例子中你将看到缓存变化

```
flush tables;
show status like 'qcache_q%';
select sql_cache * from members limit 5;
show status like 'qcache_q%';
select sql_cache * from members limit 10;
show status like 'qcache_q%';
```

显示当前缓存中的信息数量：

```
MySQL> show status like 'qcache_q%';
```

其中各个参数的意义如下：

`Qcache_free_blocks`：缓存中相邻内存块的个数。数目大说明可能有碎片。`FLUSH QUERY CACHE`会对缓存中的碎片进行整理，从而得到一个空闲块。

`Qcache_free_memory`：缓存中的空闲内存。

`Qcache_hits`：每次查询在缓存中命中时就增大

`Qcache_inserts`：每次插入一个查询时就增大。命中次数除以插入次数就是不中比率。

`Qcache_lowmem_prunes`：缓存出现内存不足并且必须要进行清理以便为更多查询提供空间的次数。这个数字最好长时间来看；如果这个数字在不断增长，就表示可能碎片非常严重，或者内存很少。（上面的 `free_blocks`和`free_memory`可以告诉您属于哪种情况）

`Qcache_not_cached`：不适合进行缓存的查询的数量，通常是由于这些查询不是 `SELECT` 语句或者用了 `now()` 之类的函数。

`Qcache_queries_in_cache`：当前缓存的查询（和响应）的数量。

`Qcache_total_blocks`：缓存中块的数量。

5. where 优化

where 条件的顺序影响查询速度

```
EXPLAIN select *,from_unixtime(sendtime) from sms where  
id='461539' and content like '13%';  
/* 0 rows affected, 1 rows found. Duration for 1 query: 0.218  
sec. */
```

```
EXPLAIN select *,from_unixtime(sendtime) from sms where content  
like '13%' and id='461539';  
/* 0 rows affected, 1 rows found. Duration for 1 query: 0.717  
sec. */
```

6. SHOW PROFILE Syntax SQL性能分析器

例 27.2. SHOW PROFILE Syntax

```
set profiling = 1; select * from mytab; show profile for query  
1;
```

7. PROCEDURE ANALYSE()

数据列优化

```
SELECT ... FROM ... WHERE ... PROCEDURE ANALYSE([max_elements,  
[max_memory]])
```

第 28 章 MySQL Connectors

1. JDBC

JDBC connection settings

```
jdbc:mysql://hostname:port/database?  
autoReconnect=true&useUnicode=true&characterEncoding=utf8
```

confluence.cfg.xml

```
<property  
name="hibernate.connection.url">jdbc:mysql://hostname:port/da  
tabase?  
autoReconnect=true&useUnicode=true&characterEncoding=  
utf8</property>
```


2. ODBC



3. MySQL native driver for PHP - mysqlnd



4. python-mysqldb

```
$ apt-cache search python | grep mysql
python-mysqldb - A Python interface to MySQL
python-mysqldb-dbg - A Python interface to MySQL (debug
extension)

$ sudo apt-get install python-mysqldb
```

```
# -*- coding: utf-8 -*-
#mysqldb
import time, MySQLdb

#连接
conn=MySQLdb.connect(host="localhost",user="root",passwd="",db=
"test",charset="utf8")
cursor = conn.cursor()

#写入
sql = "insert into user(name,created) values(%s,%s)"
param = ("neo",int(time.time()))
n = cursor.execute(sql,param)
print n

#更新
sql = "update user set name=%s where id=3"
param = ("jam")
n = cursor.execute(sql,param)
print n

#查询
n = cursor.execute("select * from user")
for row in cursor.fetchall():
    for r in row:
        print r

#删除
sql = "delete from user where name=%s"
param =("neo")
n = cursor.execute(sql,param)
```

```
print n
cursor.close()

#关闭
conn.close()
```

第 29 章 MySQL GUI/Web Manager

1. HeidiSQL

<http://www.heidisql.com/>

2. Toad for MySQL Freeware

<http://toadsoft.veriomigrations.com/>

3. phpMyAdmin - MySQL web administration tool

homepage: <http://www.phpmyadmin.net/>

```
$ wget
http://nchc.dl.sourceforge.net/sourceforge/phpmyadmin/phpMyAdmin-3.1.3.1-all-languages.tar.bz2
$ tar jxvf phpMyAdmin-3.1.3.1-all-languages.tar.bz2
$ ln -s phpMyAdmin-3.1.3.1-all-languages phpMyAdmin
```

4. Maatkit Essential command-line utilities for MySQL

<http://www.maatkit.org/>

第 30 章 Miscellaneous

1. Multi-Master Replication Manager for MySQL

2. MHA

<https://code.google.com/p/mysql-master-ha/>

3. HandlerSocket

4. Maatkit

<http://www.maatkit.org/>

5. Mroonga

Mroonga 是一个 MySQL 存储引擎，基于 **Groonga**，提供完整的全文搜索引擎。**Groonga** 是一款可嵌入式的全文搜寻引擎，具有储存功能和全文搜寻的检索功能。

<http://mroonga.org/>

<http://groonga.org/>

6. Amoeba

<http://sourceforge.net/projects/amoeba/>

Amoeba(变形虫)项目,该开源框架于2008年 开始发布一款 Amoeba for Mysql软件。这个软件致力于MySQL的分布式数据库前端代理层,它主要在应用层访问MySQL的 时候充当SQL路由功能,专注于分布式数据库代理层(Database Proxy) 开发。座落与 Client、DB Server(s)之间,对客户端透明。具有负载均衡、高可用性、SQL 过滤、读写分离、可路由相关的到目标数据库、可并发请求多台数据库合并结果。通过 Amoeba你能够完成多数据源的高可用、负载均衡、数据切片的功能,目前Amoeba已在很多 企业的生产线上使用。

第 31 章 FAQ

1. Reset root password 重置MySQL root密码

忘记root密码是使用 --skip-grant-tables 启动项

CentOS 6.x

```
# vim /etc/init.d/mysqld

$exec --skip-grant-tables --datadir="$datadir" --socket="$socketfile" \
  --pid-file="$mypidfile" \
  --basedir=/usr --user=mysql >/dev/null 2>&1 &
```

```
# /etc/init.d/mysqld restart
Stopping mysqld:           [ OK ]
Starting mysqld:          [ OK ]

# mysqladmin -u root flush-privileges password "newpassword"
```

1.1. MySQL 5.7.x

CentOS 7.x

添加 skip-grant-tables=1 选项，然后重启mysql

```
# cat /etc/my.cnf
# For advice on how to change settings please see
# http://dev.mysql.com/doc/refman/5.6/en/server-configuration-
defaults.html

[mysqld]
#
# Remove leading # and set to the amount of RAM for the most important
data
```

```
# cache in MySQL. Start at 70% of total RAM for dedicated server, else
10%.
# innodb_buffer_pool_size = 128M
#
# Remove leading # to turn on a very important data integrity option:
logging
# changes to the binary log between backups.
# log_bin
#
# Remove leading # to set options mainly useful for reporting servers.
# The server defaults are faster for transactions and fast SELECTs.
# Adjust sizes as needed, experiment to find the optimal values.
# join_buffer_size = 128M
# sort_buffer_size = 2M
# read_rnd_buffer_size = 2M
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
skip-grant-tables=1
# Disabling symbolic-links is recommended to prevent assorted security
risks
symbolic-links=0

# Recommended in standard MySQL setup
sql_mode=NO_ENGINE_SUBSTITUTION,STRICT_TRANS_TABLES

[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

```
# systemctl restart mysqld
```

```
update mysql.user set authentication_string=password('netkiller') where
user='root' and Host = 'localhost';
flush privileges;
quit;
```

```
# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
```



```
Server version: 5.7.14 MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> update mysql.user set authentication_string=password('netkiller')
where user='root' and Host = 'localhost';
Query OK, 1 row affected, 1 warning (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 1

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

mysql> quit;
Bye
```

删除 skip-grant-tables=1 重启MySQL

1.2. MySQL 8.0

```
[root@localhost log]# vim /etc/my.cnf

[mysqld]
skip-grant-table
```

```
ALTER USER root@localhost identified by 'MQiEgelikst7S_6tlXzB0mt';
ALTER USER root@localhost PASSWORD EXPIRE NEVER;
```

2. 查看错误代码

```
mysql> \! perror 6
OS error code 6: No such device or address
```

2.1. ERROR 1153 (08S01) at line 3168: Got a packet bigger than 'max_allowed_packet' bytes

```
max_allowed_packet=500M
```

2.2. ERROR 1129 (00000): Host 'XXXXXXX' is blocked because of many connection errors; unblock with 'mysqladmin flush-hosts'

连接在中途被中断了的连接请求。在 max_connect_errors 次失败请求后，mysql 阻止该主机进一步的连接，直到管理员执行命令 mysqladmin flush-hosts。

```
mysql> flush hosts;
```

```
set global
max_connect_errors=1000;
```

```
max_connect_errors=10000
```

3. 临时表是否需要建索引

答案：要

4. [Warning] Changed limits: max_open_files: 5000 (requested 20480)

```
2018-01-08T01:34:44.515973Z 0 [Warning] Changed limits:
max_open_files: 5000 (requested 10240)
2018-01-08T01:34:44.516402Z 0 [Warning] Changed limits:
table_open_cache: 1471 (requested 2000)
```

提出出现在 CentOS 7 ulimit 配置没有问题的情况下mysql日志提示 Warning

```
# ulimit -Sa | grep "open files"
open files (-n) 40960
```

```
[root@netkiller ~]# cat /proc/`pidof
mysqld`/limits
Limit Soft Limit Hard Limit Units
Max cpu time unlimited unlimited
seconds
Max file size unlimited unlimited bytes
Max data size unlimited unlimited bytes
Max stack size 8388608 unlimited bytes
Max core file size 0 unlimited bytes
Max resident set unlimited unlimited
bytes
Max processes 63494 63494 processes
Max open files 5000 5000 files
Max locked memory 65536 65536 bytes
Max address space unlimited unlimited
bytes
Max file locks unlimited unlimited
locks
Max pending signals 63494 63494 signals
Max msgqueue size 819200 819200 bytes
```

```
Max nice priority 0 0
Max realtime priority 0 0
Max realtime timeout unlimited
unlimited us
```

动态改变

```
[root@netkiller ~]# egrep '^(Limit|Max
open files)' /proc/`pidof mysqld`/limits
Limit Soft Limit Hard Limit Units
Max open files 5000 5000 files
```

问题的出现原因是systemctl启动脚本覆盖了ulimit配置

```
# cat
/usr/lib/systemd/system/mysqld.service | grep -A2
open_files_limit
# Sets open_files_limit
LimitNOFILE = 5000
```

解决方法，直接修改上面的数值，不建议修改mysqld.service，这样会影响你下次升级。请采用下面的方案完美解决：

```
mkdir /usr/lib/systemd/system/mysqld.service.d
cat >> /usr/lib/systemd/system/mysqld.service.d/override.conf
<<EOF
[Service]
LimitNOFILE=40960
EOF
```

重启 MySQL

```
systemctl daemon-reload
systemctl restart mysqld
```

检查是否生效

```
mysql> show variables like 'open_files_limit';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| open_files_limit | 65535 |
+-----+-----+
1 row in set (0.01 sec)
```

5. this is incompatible with sql_mode=only_full_group_by

ERROR 1055 (42000): Expression #1 of SELECT list is not in GROUP BY clause and contains nonaggregated column 'mydb.contact.id' which is not functionally dependent on columns in GROUP BY clause; this is incompatible with sql_mode=only_full_group_by

```
mysql> select @@version;
+-----+
| @@version |
+-----+
| 5.7.10    |
+-----+
1 row in set (0.00 sec)

mysql> select @@GLOBAL.sql_mode;
+-----+
| @@GLOBAL.sql_mode
|
+-----+
|
| ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_
| DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_S
| UBSSTITUTION |
+-----+
1 row in set (0.00 sec)

mysql> SET sql_mode = '';
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> select id,name from contact group by name limit 10;
+-----+-----+
```

id	name
84046	张伟
80259	张磊
784	王岩
87685	杨钊

10 rows in set (0.07 sec)

不建议设置 SET sql_mode = "", 正确方式如下:

```
mysql> set global
sql_mode='STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION';
mysql> set session
sql_mode='STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION';
```

或者采用

Adding only one mode to sql_mode without removing existing ones:

```
SET sql_mode=(SELECT CONCAT(@@sql_mode,',<mode_to_add>'));
```

Removing only a specific mode from sql_mode without removing others:

```
SET sql_mode=(SELECT
REPLACE(@@sql_mode,'<mode_to_remove>',''));
```

In your case, if you want to remove only ONLY_FULL_GROUP_BY mode, then use below command:

```
SET sql_mode=(SELECT REPLACE(@@sql_mode, 'ONLY_FULL_GROUP_BY',
```



```
''));
```

6. ERROR 1071 (42000) at line 25: Specified key was too long; max key length is 767 bytes

这个保存通常出现在高版本数据库向低版本数据导入数据，活着云主机例如阿里云。

```
mysql> show variables like '%innodb_large_prefix%';
+-----+-----+
| Variable_name      | Value |
+-----+-----+
| innodb_large_prefix | OFF   |
+-----+-----+
1 row in set (0.00 sec)
```

解决方案

```
innodb_large_prefix=ON
```

7. ERROR 1086 (HY000): File '/var/lib/mysql-files/order.txt' already exists

SELECT * FROM tablename INTO OUTFILE 不支持覆盖操作，这是MySQL从安全角度考虑的。

```
mysql> SELECT * FROM `order` INTO OUTFILE '/var/lib/mysql-  
files/order.txt';  
ERROR 1086 (HY000): File '/var/lib/mysql-files/order.txt'  
already exists
```

8. Error Code: 1146. Table 'test.CACHE_UPDATE' doesn't exist

问题分析，首先确认表是存在的，但是无法读取。可以判定是 lower_case_table_names=1 选项的问题，开启后表以小写方式打开。

```
Error Code: 1146. Table 'test.CACHE_UPDATE' doesn't exist
```

如果是 MySQL 8.0 之前没有开启 lower_case_table_names=1，现在需要开启，加入配置后将无法启动，解决办法是，你需要重做 mysql data 目录

```
[root@localhost ~]# rm -rf /var/lib/mysql/*
[root@localhost ~]# systemctl restart mysqld
[root@localhost ~]# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.21 Source distribution

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show variables like '%lower_case_table_names%';
+-----+-----+
| Variable_name          | Value |
+-----+-----+
| lower_case_table_names | 1     |
+-----+-----+
1 row in set (0.01 sec)

mysql> SELECT CURRENT_SERIAL FROM CACHE_UPDATE WHERE ID=1;
+-----+
| CURRENT_SERIAL |
+-----+
|                7 |
+-----+
```

```
+-----+  
1 row in set (0.00 sec)
```

9. ERROR 1273 (HY000) at line 3364: Unknown collation: 'utf8mb4_0900_ai_ci'

找出指定字符集的表

```
select TABLE_SCHEMA, TABLE_NAME, TABLE_COLLATION from
information_schema.tables where table_collation =
'utf8mb4_0900_ai_ci' and table_schema = 'your_schema';
```

```
SELECT
    CONCAT(
        'ALTER TABLE ',
        TABLE_NAME,
        ' CONVERT TO CHARACTER SET utf8mb4 COLLATE
utf8mb4_general_ci;'
    )
FROM
    information_schema.`TABLES`
WHERE
    TABLE_SCHEMA = 'DATABASE_NAME';
```

10. ERROR 1290 (HY000): The MySQL server is running with the --secure-file-priv option so it cannot execute this statement

MySQL 不允许向 `secure_file_priv` 意外的目录导出文件。

```
mysql> SELECT * FROM `order` INTO OUTFILE '/tmp/order.txt';
ERROR 1290 (HY000): The MySQL server is running with the --
secure-file-priv option so it cannot execute this statement
```

```
mysql> show variables like '%secure%';
```

Variable_name	Value
require_secure_transport	OFF
secure_auth	ON
secure_file_priv	/var/lib/mysql-files/

```
3 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM `order` INTO OUTFILE '/var/lib/mysql-
files/order.txt';
```

```
Query OK, 3 rows affected (0.00 sec)
```

```
root@netkiller ~ % cat /var/lib/mysql-files/order.txt
```

```
1      Tom      22      2017-11-16 17:23:15
2      Neo      34.65   2017-11-16 17:29:28
3      Cici     34.98   2017-11-16 17:30:29
```

在 `my.cnf` 中加入 `secure-file-priv=/tmp` 可以修改为你需要的目录。

11. ERROR 1364: 1364: Field 'id' doesn't have a default value

```
set
@@SESSION.sql_mode='NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION'
;
SELECT @@GLOBAL.sql_mode;
UPDATE `cms`.`content` SET `source`='test' WHERE
`content_id`='1099';
```


12. ERROR 1415: Not allowed to return a result set from a trigger

触发器中不允许返回结果集，解决方法是顶一个变量，然后将返回值返回给变量。

```
DROP TRIGGER IF EXISTS `test`.`demo_AFTER_INSERT`;  
  
DELIMITER $$  
USE `test`$$  
CREATE DEFINER=`root`@`%` TRIGGER `test`.`demo_AFTER_INSERT`  
AFTER INSERT ON `demo` FOR EACH ROW  
BEGIN  
    set @rev = "";  
    SELECT  
    OUT2FILE('/tmp/demo.log',  
            CONCAT_WS(',',  
                      NEW.id,  
                      NEW.name,  
                      NEW.sex,  
                      NEW.address))  
    INTO @rev;  
END$$  
DELIMITER ;
```

13. ERROR 1503 (HY000): A PRIMARY KEY must include all columns in the table's partitioning function

<http://dev.mysql.com/doc/refman/5.1/en/partitioning-limitations-partitioning-keys-unique-keys.html>

14. ERROR 1819 (HY000): Your password does not satisfy the current policy requirements

MySQL 5.7 密码强度，必须含有0-9，a-z,A-Z以及“-”或“_”

<https://dev.mysql.com/doc/refman/5.7/en/validate-password-options-variables.html>

禁用密码安全策略（早起5.7版本可用，新版已经废弃这个选项）

```
password                                # cat /etc/my.cnf | grep validate-  
                                         validate-password=OFF
```

新的方式修改策略与密码长度

```
mysql> set global validate_password_policy=0;  
mysql> set global validate_password_length=4;  
mysql> grant all privileges on test.* to 'test'@localhost  
identified by 'chen';
```

15. ERROR 1820 (HY000): You must reset your password using ALTER USER statement before executing this statement.

这个错误来自 MySQL 5.7，首次登陆MySQL 5.7 必须修改密码

```
ALTER USER 'root'@'localhost'  
IDENTIFIED BY 'your_password';
```

16. ERROR 1840 (HY000) at line 24: @@GLOBAL.GTID_PURGED can only be set when @@GLOBAL.GTID_EXECUTED is empty.

问题出现在 MySQL 5.7 导入数据库时候

```
[www@testing ~]$ zcat netkiller.2021-08-19.sql.gz | mysql
netkiller
ERROR 1840 (HY000) at line 24: @@GLOBAL.GTID_PURGED can only be
set when @@GLOBAL.GTID_EXECUTED is empty.
```

解决方案

执行 reset master;

```
[www@testing ~]$ mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 25669
Server version: 5.7.35 MySQL Community Server (GPL)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All
rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or
its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current
input statement.

mysql> reset master;
```

```
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> exit
```

```
Bye
```

```
[www@testing ~]$ zcat netkiller.2021-08-19.sql.gz | mysql  
netkiller
```

17. ERROR 3024 (HY000): Query execution was interrupted, maximum statement execution time exceeded

```
mysql> select * from cert;
ERROR 3024 (HY000): Query execution was interrupted, maximum
statement execution time exceeded
mysql> SET GLOBAL max_execution_time=10;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from cert;
Empty set (0.08 sec)

mysql> show variables like 'max_execution_time';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| max_execution_time | 10 |
+-----+-----+
1 row in set (0.01 sec)

mysql> select /*+ max_execution_time(3000)*/ count(*) from
cert;
+-----+
| count(*) |
+-----+
|          0 |
+-----+
1 row in set (0.29 sec)
```

18. Authentication plugin

**'caching_sha2_password' cannot be loaded:
/usr/lib64/mysql/plugin/caching_sha2_password.s
o: cannot open shared object file: No such file or
directory**

这个故障出现在 MySQL 8.0 上，用户使用 mysql client 5.7 链接 MySQL 8.0 提示如下

```
[root@netkiller ~]# mysql -h 193.112.95.53 -uroot -p
Enter password:
ERROR 2059 (HY000): Authentication plugin
'caching_sha2_password' cannot be loaded:
/usr/lib64/mysql/plugin/caching_sha2_password.so: cannot open
shared object file: No such file or directory
```

解决方案，创建用户使用 mysql_native_password 密码

```
mysql> CREATE USER 'root'@'%' IDENTIFIED WITH
mysql_native_password BY 'pMQiEgelikst7S_6tlXzB0mt_4b';
Query OK, 0 rows affected (0.08 sec)

mysql> grant all on *.* to 'root'@'%';
Query OK, 0 rows affected (0.08 sec)
```

重新链接

```
[root@netkiller ~]# mysql -h 193.112.95.53 -uneo -p
```


Enter password:

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 24

Server version: 8.0.11 MySQL Community Server - GPL

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

19. com.mysql.jdbc.exceptions.jdbc4.MySQLNonTransientConnectionException: Public Key Retrieval is not allowed

问题出在现在 MySQL 8.0 版本

解决方法：在连接后面添加 allowPublicKeyRetrieval=true

```
spring.datasource.url=jdbc:mysql://192.168.0.1:3306/test?  
useUnicode=true&characterEncoding=UTF-  
8&serverTimezone=UTC&useSSL=false&allowPublicKeyRetrieval=true
```

20. mysqldump: Couldn't execute 'SELECT COLUMN_NAME,

问题出现在 MySQL 8.0 备份 MySQL 5.7 数据库时。

```
mysqldump: Couldn't execute 'SELECT COLUMN_NAME,  
JSON_EXTRACT(HISTOGRAM, '$."number-of-buckets-specified"')  
FROM information_schema.COLUMN_STATISTICS          WHERE  
SCHEMA_NAME = 'testra' AND TABLE_NAME = 'branch';': Unknown  
table 'column_statistics' in information_schema (1109)
```

解决办法，使用 --column-statistics=0 选项

```
mysqldump -hdb.netkiller.cn -uroot -ptest neo --column-  
statistics=0
```

21. this is incompatible with sql_mode=only_full_group_by

```
Expression #2 of SELECT list is not in GROUP BY clause and contains nonaggregated column 'test.table.username' which is not functionally dependent on columns in GROUP BY clause; this is incompatible with sql_mode=only_full_group_by
```

问题出在 MySQL 5.7 向 MySQL 8.0 迁移。

查询 sql_mode 设置

```
select @@global.sql_mode;
'ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION'
```

临时解决方案，去掉 ONLY_FULL_GROUP_BY 即可

```
set
@@GLOBAL.sql_mode='STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,
NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION';
set
@@SESSION.sql_mode='STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,
NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION';
```

彻底解决，有三种解决方案：

- 第一种，将 MySQL 的版本降回到 5.7
- 第二种，关闭 `only_full_group_by` 检查
- 第三种，修改sql使其遵守`only_full_group_by`语法规则

采用哪个方案？打工人的方案，就是解决眼前问题，如果你对项目负责，最好采用第三种方案。

我的职业生涯遇到无数次，因版本太低同时发现重大漏洞，厂商已经不在维护该版本，此时一边是黑客攻击，一边你又无法短时间内升级到新版本，公司又不能接受停服，你会怎么应对？我用了很多非常规手段，给开发团队争取了一周的时间，升级系统。

22. mysqldump: [Warning] Using a password on the command line interface can be insecure.

```
mysqldump: [Warning] Using a password on the command line interface can be insecure.
```

```
vim ~/.my.cnf  
  
[mysqldump]  
user=root  
password=123456
```

```
[root@netkiller neo]# mysqldump -h 192.168.30.40 neo  
[root@netkiller neo]# mysqldump --defaults-extra-file=~/.my.cnf  
netkiller > netkiller.sql
```

23. mysql: [Warning] Using a password on the command line interface can be insecure.

当使用 `-pnetkiller` 在命令行出现密码的时候，会提示下面信息。

```
[www@testing ~]$ mysql -h127.0.0.1 -uneo -pnetkiller test
mysql: [Warning] Using a password on the command line interface
can be insecure.
Reading table information for completion of table and column
names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 25623
Server version: 5.7.35 MySQL Community Server (GPL)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All
rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or
its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current
input statement.

mysql>
```

创建 `~/.my.cnf` 配置文件，将密码写入配置

```
[www@testing ~]$ cat ~/.my.cnf
[mysql]
host=127.0.0.1
```

```
user=neo  
password=netkiller
```

这时直接使用 mysql 命令即可进入。

```
[www@testing ~]$ mysql  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 25622  
Server version: 5.7.35 MySQL Community Server (GPL)  
  
Copyright (c) 2000, 2020, Oracle and/or its affiliates. All  
rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or  
its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current  
input statement.  
  
mysql>
```


部分 IV. Redis

<http://redis.io/>

第 32 章 Redis 安装

1. CentOS 8 Stream

```
[root@netkiller ~]# dnf install -y redis
[root@netkiller ~]# systemctl enable redis
```

```
cp /etc/redis.conf{,.original}
sed -i 's/bind 127.0.0.1/bind 0.0.0.0/g' /etc/redis.conf
sed -i 's/timeout 0/timeout 30/' /etc/redis.conf
sed -i 's/# maxclients 10000/maxclients 1000/' /etc/redis.conf
sed -i 's/# maxmemory-policy noeviction/maxmemory-policy
volatile-lru/g' /usr/local/etc/redis.conf

random=$(cat /dev/urandom | tr -cd [:alnum:] | fold -w32 | head
-n 1)
sed -i "s/# requirepass foobared/requirepass ${random}/g"
/etc/redis.conf
echo "Redis random password is: ${random}"

cat >> /etc/security/limits.d/20-nofile.conf <<EOF

redis soft nofile 65500
redis hard nofile 65500
EOF

cat >> /etc/sysctl.conf <<EOF
# Set up for Redis
vm.overcommit_memory = 1
net.core.somaxconn = 1024
EOF

sysctl -w net.core.somaxconn=1024
sysctl -w vm.overcommit_memory=1
```

```
cat >> /etc/rc.local <<EOF

# Set up for Redis
echo never > /sys/kernel/mm/transparent_hugepage/enabled
EOF

echo never > /sys/kernel/mm/transparent_hugepage/enabled

systemctl enable redis
systemctl start redis
systemctl status redis
```

2. CentOS 7

[Netkiller OSCM](#) 一键安装

```
curl -s  
https://raw.githubusercontent.com/oscm/shell/master/database/redis/redis.sh | bash
```

3. CentOS 6

安装fedora的YUM源,

```
rpm -Uvh http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-7.noarch.rpm
```

安装redis

```
# yum install redis  
# chkconfig redis on  
# service redis start
```

备份配置文件,

```
# cp /etc/redis.conf /etc/redis.conf.original
```

3.1. 主从同步

主从同步配置非常简单, 只需在从服务器 `/etc/redis.conf` 文件中开启 `slaveof` 即可

```
slaveof 192.168.2.1 6379
```

查看 `/var/log/redis/redis.log` 日志, 可以看到同步情况

```
[20274] 09 Jul 13:13:53 * Server started, Redis version 2.4.10
```

```
[20274] 09 Jul 13:13:53 * DB loaded from disk: 0 seconds
[20274] 09 Jul 13:13:53 * The server is now ready to accept
connections on port 6379
[20274] 09 Jul 13:13:54 * Connecting to MASTER...
[20274] 09 Jul 13:13:54 * MASTER <-> SLAVE sync started
[20274] 09 Jul 13:13:54 * Non blocking connect for SYNC fired
the event.
[20274] 09 Jul 13:13:54 * MASTER <-> SLAVE sync: receiving 672
bytes from master
[20274] 09 Jul 13:13:54 * MASTER <-> SLAVE sync: Loading DB in
memory
[20274] 09 Jul 13:13:54 * MASTER <-> SLAVE sync: Finished with
success
```

3.2. Sentinel

4. Ubuntu

```
$ sudo apt-get install redis-server
```

```
$ dpkg -s redis-server
Package: redis-server
Status: install ok installed
Priority: optional
Section: database
Installed-Size: 208
Maintainer: Chris Lamb <lamby@debian.org>
Architecture: amd64
Source: redis
Version: 2:1.2.6-1
Depends: libc6 (>= 2.7), adduser
Conffiles:
 /etc/redis/redis.conf a19bad63017ec19def2c3a8a07bdc362
 /etc/logrotate.d/redis-server 06755b99ef70d62a56cff94cbfc36de7
 /etc/init.d/redis-server 3742555c10ab16fdd67fcbaf92faf694
 /etc/bash_completion.d/redis-cli
848565df7f222dc03c8d5cb34b9e0188
Description: Persistent key-value database with network
interface
 Redis is a key-value database in a similar vein to memcache
but the dataset
 is non-volatile. Redis additionally provides native support
for atomically
 manipulating and querying data structures such as lists and
sets.
.
 The dataset is stored entirely in memory and periodically
flushed to disk.
Homepage: http://code.google.com/p/redis/
```

```
$ cat /etc/redis/redis.conf
```

```
# Redis configuration file example

# By default Redis does not run as a daemon. Use 'yes' if you
need it.
# Note that Redis will write a pid file in /var/run/redis.pid
when daemonized.
daemonize yes

# When run as a daemon, Redis write a pid file in
/var/run/redis.pid by default.
# You can specify a custom pid file location here.
pidfile /var/run/redis.pid

# Accept connections on the specified port, default is 6379
port 6379

# If you want you can bind a single interface, if the bind
option is not
# specified all the interfaces will listen for connections.
#
bind 127.0.0.1

# Close the connection after a client is idle for N seconds (0
to disable)
timeout 300

# Set server verbosity to 'debug'
# it can be one of:
# debug (a lot of information, useful for development/testing)
# notice (moderately verbose, what you want in production
probably)
# warning (only very important / critical messages are logged)
loglevel notice

# Specify the log file name. Also 'stdout' can be used to force
# the demon to log on the standard output. Note that if you use
standard
# output for logging but daemonize, logs will be sent to
/dev/null
logfile /var/log/redis/redis-server.log

# Set the number of databases. The default database is DB 0,
you can select
# a different one on a per-connection basis using SELECT <dbid>
```



```
where
# dbid is a number between 0 and 'databases'-1
databases 16

##### SNAPSHOTTING
#####
#
# Save the DB on disk:
#
#   save <seconds> <changes>
#
# Will save the DB if both the given number of seconds and
the given
# number of write operations against the DB occurred.
#
# In the example below the behaviour will be to save:
# after 900 sec (15 min) if at least 1 key changed
# after 300 sec (5 min) if at least 10 keys changed
# after 60 sec if at least 10000 keys changed
save 900 1
save 300 10
save 60 10000

# Compress string objects using LZF when dump .rdb databases?
# For default that's set to 'yes' as it's almost always a win.
# If you want to save some CPU in the saving child set it to
'no' but
# the dataset will likely be bigger if you have compressible
values or keys.
rdbcompression yes

# The filename where to dump the DB
dbfilename dump.rdb

# For default save/load DB in/from the working directory
# Note that you must specify a directory not a file name.
dir /var/lib/redis

##### REPLICATION
#####

# Master-Slave replication. Use slaveof to make a Redis
instance a copy of
# another Redis server. Note that the configuration is local to
the slave
```

```
# so for example it is possible to configure the slave to save
the DB with a
# different interval, or to listen to another port, and so on.
#
# slaveof <masterip> <masterport>

# If the master is password protected (using the "requirepass"
configuration
# directive below) it is possible to tell the slave to
authenticate before
# starting the replication synchronization process, otherwise
the master will
# refuse the slave request.
#
# masterauth <master-password>

##### SECURITY
#####

# Require clients to issue AUTH <PASSWORD> before processing
any other
# commands. This might be useful in environments in which you
do not trust
# others with access to the host running redis-server.
#
# This should stay commented out for backward compatibility and
because most
# people do not need auth (e.g. they run their own servers).
#
# requirepass foobared

##### LIMITS
#####

# Set the max number of connected clients at the same time. By
default there
# is no limit, and it's up to the number of file descriptors
the Redis process
# is able to open. The special value '0' means no limits.
# Once the limit is reached Redis will close all the new
connections sending
# an error 'max number of clients reached'.
#
# maxclients 128
```

```
# Don't use more memory than the specified amount of bytes.
# When the memory limit is reached Redis will try to remove
keys with an
# EXPIRE set. It will try to start freeing keys that are going
to expire
# in little time and preserve keys with a longer time to live.
# Redis will also try to remove objects from free lists if
possible.
#
# If all this fails, Redis will start to reply with errors to
commands
# that will use more memory, like SET, LPUSH, and so on, and
will continue
# to reply to most read-only commands like GET.
#
# WARNING: maxmemory can be a good idea mainly if you want to
use Redis as a
# 'state' server or cache, not as a real DB. When Redis is used
as a real
# database the memory usage will grow over the weeks, it will
be obvious if
# it is going to use too much memory in the long run, and
you'll have the time
# to upgrade. With maxmemory after the limit is reached you'll
start to get
# errors for write operations, and this may even lead to DB
inconsistency.
#
# maxmemory <bytes>

##### APPEND ONLY MODE
#####

# By default Redis asynchronously dumps the dataset on disk. If
you can live
# with the idea that the latest records will be lost if
something like a crash
# happens this is the preferred way to run Redis. If instead
you care a lot
# about your data and don't want to that a single record can
get lost you should
# enable the append only mode: when this mode is enabled Redis
will append
# every write operation received in the file appendonly.log.
This file will
```

```
# be read on startup in order to rebuild the full dataset in
memory.
#
# Note that you can have both the async dumps and the append
only file if you
# like (you have to comment the "save" statements above to
disable the dumps).
# Still if append only mode is enabled Redis will load the data
from the
# log file at startup ignoring the dump.rdb file.
#
# The name of the append only file is "appendonly.log"
#
# IMPORTANT: Check the BGREWRITEAOF to check how to rewrite the
append
# log file in background when it gets too big.

appendonly no

# The fsync() call tells the Operating System to actually write
data on disk
# instead to wait for more data in the output buffer. Some OS
will really flush
# data on disk, some other OS will just try to do it ASAP.
#
# Redis supports three different modes:
#
# no: don't fsync, just let the OS flush the data when it
wants. Faster.
# always: fsync after every write to the append only log .
Slow, Safest.
# everysec: fsync only if one second passed since the last
fsync. Compromise.
#
# The default is "always" that's the safer of the options. It's
up to you to
# understand if you can relax this to "everysec" that will
fsync every second
# or to "no" that will let the operating system flush the
output buffer when
# it want, for better performances (but if you can live with
the idea of
# some data loss consider the default persistence mode that's
snapshotting).
```

```
appendfsync always
# appendfsync everysec
# appendfsync no

##### ADVANCED CONFIG
#####

# Glue small output buffers together in order to send small
replies in a
# single TCP packet. Uses a bit more CPU but most of the times
it is a win
# in terms of number of queries per second. Use 'yes' if
unsure.
glueoutputbuf yes

# Use object sharing. Can save a lot of memory if you have many
common
# string in your dataset, but performs lookups against the
shared objects
# pool so it uses more CPU and can be a bit slower. Usually
it's a good
# idea.
#
# When object sharing is enabled (shareobjects yes) you can use
# shareobjectspoolsize to control the size of the pool used in
order to try
# object sharing. A bigger pool size will lead to better
sharing capabilities.
# In general you want this value to be at least the double of
the number of
# very common strings you have in your dataset.
#
# WARNING: object sharing is experimental, don't enable this
feature
# in production before of Redis 1.0-stable. Still please try
this feature in
# your development environment so that we can test it better.
shareobjects no
shareobjectspoolsize 1024
```

```
$ sudo /etc/init.d/redis-server start
```

5. Mac 安装 Redis

```
neo@MacBook-Pro ~ % brew install redis
```

Redis 被安装在 `/usr/local/Cellar/redis/5.0.4` 目录下

启动 Redis

```
brew services start redis
```

前台运行，这种方式用于调试

```
redis-server /usr/local/etc/redis.conf
```

6. 源码编译安装

这里仍然使用 [Netkiller OSCM](#) 安装，源码安装

```
curl -s  
https://raw.githubusercontent.com/oscm/shell/master/database/redis/source/redis-5.0.4.sh | bash
```

7.

8. Test Redis

<http://redis.io/commands>

```
$ redis-cli info
redis_version:1.2.6
arch_bits:64
multiplexing_api:epoll
uptime_in_seconds:859
uptime_in_days:0
connected_clients:1
connected_slaves:0
used_memory:619490
used_memory_human:604.97K
changes_since_last_save:0
bgsave_in_progress:0
last_save_time:1311100746
bgrewriteaof_in_progress:0
total_connections_received:4
total_commands_processed:0
role:master

$ redis-cli set name neo
OK
$ redis-cli get name
neo

$ telnet localhost 6379
Trying ::1...
telnet: connect to address ::1: Connection refused
Trying 127.0.0.1...
Connected to localhost (127.0.0.1).
Escape character is '^]'.
get name
$3
neo
quit
Connection closed by foreign host.
```

第 33 章 /etc/redis.conf

参数说明, redis.conf 配置项说明如下:

1. Redis默认不是以守护进程的方式运行, 可以通过该配置项修改, 使用yes启用守护进程

```
daemonize no
```

2. 当Redis以守护进程方式运行时, Redis默认会把pid写入/var/run/redis.pid文件, 可以通过pidfile指定

```
pidfile /var/run/redis.pid
```

3. 指定Redis监听端口, 默认端口为6379, 作者在自己的一篇博文中解释了为什么选用6379作为默认端口, 因为6379在手机按键上MERZ对应的号码, 而MERZ取自意大利歌女Alessia Merz的名字

```
port 6379
```

4. 绑定的主机地址

```
bind 127.0.0.1
```

5. 当 客户端闲置多长时间后关闭连接, 如果指定为0, 表示关闭该功能

```
timeout 300
```

6. 指定日志记录级别, Redis总共支持四个级别: debug、verbose、notice、warning, 默认为verbose

```
loglevel verbose
```

7. 日志记录方式, 默认为标准输出, 如果配置Redis为守护进程方式运行, 而这里又配置为日志记录方式为标准输出, 则日志将会发送给/dev/null

```
logfile stdout
```

8. 设置数据库的数量, 默认数据库为0, 可以使用SELECT <dbid>命令在连接上指定数据库id

```
databases 16
```

9. 指定在多长时间之内, 有多少次更新操作, 就将数据同步到数据文件, 可以多个条件配合

```
save <seconds> <changes>
```

Redis默认配置文件中提供了三个条件:

```
save 900 1
```

```
save 300 10
```

```
save 60 10000
```

分别表示900秒 (15分钟) 内有1个更改, 300秒 (5分钟) 内有10个更改以及60秒内有10000个更改。

10. 指定存储至本地数据库时是否压缩数据, 默认为yes, Redis采用LZF压缩, 如果为了节省CPU时间, 可以关闭该选项, 但会导致数据库文件变的巨大

```
rdbcompression yes
```

11. 指定本地数据库文件名, 默认值为dump.rdb
dbfilename dump.rdb
12. 指定本地数据库存放目录
dir ./
13. 设置当本机为slav服务时, 设置master服务的IP地址及端口, 在Redis启动时, 它会自动从master进行数据同步
slaveof <masterip> <masterport>
14. 当master服务设置了密码保护时, slav服务连接master的密码
masterauth <master-password>
15. 设置Redis连接密码, 如果配置了连接密码, 客户端在连接Redis时需要通过AUTH <password>命令提供密码, 默认关闭
requirepass foobared
16. 设置同一时间最大客户端连接数, 默认无限制, Redis可以同时打开的客户端连接数为Redis进程可以打开的最大文件描述符数, 如果设置 maxclients 0, 表示不作限制。当客户端连接数到达限制时, Redis会关闭新的连接并向客户端返回max number of clients reached错误信息
maxclients 128
17. 指定Redis最大内存限制, Redis在启动时会把数据加载到内存中, 达到最大内存后, Redis会先尝试清除已到期或即将到期的Key, 当此方法处理后, 仍然到达最大内存设置, 将无法再进行写入操作, 但仍然可以进行读取操作。Redis新的vm机制, 会把Key存放内存, Value会存放在swap区
maxmemory <bytes>
18. 指定是否在每次更新操作后进行日志记录, Redis在默认情况下是异步的把数据写入磁盘, 如果不开启, 可能会在断电时导致一段时间内的数据丢失。因为 redis本身同步数据文件是按上面save条件来同步的, 所以有的数据会在一段时间内只存在于内存中。默认为no
appendonly no
19. 指定更新日志文件名, 默认为appendonly.aof
appendfilename appendonly.aof
20. 指定更新日志条件, 共有3个可选值:
no: 表示等待操作系统进行数据缓存同步到磁盘 (快)
always: 表示每次更新操作后手动调用fsync()将数据写到磁盘 (慢, 安全)
everysec: 表示每秒同步一次 (折衷, 默认值)
appendfsync everysec
21. 指定是否启用虚拟内存机制, 默认值为no, 简单的介绍一下, vm机制将数据分页存放, 由Redis将访问量较少的页即冷数据swap到磁盘上, 访问多的页面由磁盘自动换出到内存中 (在后面的文章我会仔细分析Redis的VM机制)
vm-enabled no
22. 虚拟内存文件路径, 默认值为/tmp/redis.swap, 不可多个Redis实例共享
vm-swap-file /tmp/redis.swap
23. 将所有大于vm-max-memory的数据存入虚拟内存, 无论vm-max-memory设置多小, 所有索引数据都是内存存储的(Redis的索引数据 就是keys), 也就是说, 当vm-max-memory设置为0的时候, 其实是所有value都存在于磁盘。默认值为0
vm-max-memory 0

24. Redis swap文件分成了很多的page, 一个对象可以保存在多个page上面, 但一个page上不能被多个对象共享, vm-page-size是要根据存储的数据大小来设定的, 作者建议如果存储很多小对象, page大小最好设置为32或者64bytes; 如果存储很大对象, 则可以使用更大的page, 如果不 确定, 就使用默认值

```
vm-page-size 32
```

25. 设置swap文件中的page数量, 由于页表(一种表示页面空闲或使用的bitmap)是在放在内存中的, 在磁盘上每8个pages将消耗1byte的内存。

```
vm-pages 134217728
```

26. 设置访问swap文件的线程数, 最好不要超过机器的核数, 如果设置为0, 那么所有对swap文件的操作都是串行的, 可能会造成比较长时间的延迟。默认值为4

```
vm-max-threads 4
```

27. 设置在向客户端应答时, 是否把较小的包合并为一个包发送, 默认为开启

```
glueoutputbuf yes
```

28. 指定在超过一定的数量或者最大的元素超过某一临界值时, 采用一种特殊的哈希算法

```
hash-max-zipmap-entries 64
```

```
hash-max-zipmap-value 512
```

29. 指定是否激活重置哈希, 默认为开启(后面在介绍Redis的哈希算法时具体介绍)

```
activeresharding yes
```

30. 指定包含其它的配置文件, 可以在同一主机上多个Redis实例之间使用同一份配置文件, 而同时各个实例又拥有自己的特定配置文件

```
include /path/to/local.conf
```

1. 密码认证

打开 /etc/redis.conf 修改 requirepass 配置项

```
# vim /etc/redis.conf  
  
requirepass test123
```

测试

```
# service redis restart  
Stopping redis-server: [ OK ]
```

```
Starting redis-server: [
OK ]

# redis-cli
redis 127.0.0.1:6379> set h helloworld
(error) ERR operation not permitted
```

auth test123

```
redis 127.0.0.1:6379> auth test123
OK
redis 127.0.0.1:6379> set h helloworld
OK
redis 127.0.0.1:6379> get h
"helloworld"
redis 127.0.0.1:6379> exit
```

redis-cli -a 参数指定密码

```
# redis-cli -a test123
redis 127.0.0.1:6379> set h helloworld
OK
redis 127.0.0.1:6379> get h
"helloworld"
redis 127.0.0.1:6379> exit
```

通过 config 动态改变密码，无需重新启动 redis 进程

```
# redis-cli -a test123
redis 127.0.0.1:6379> config get requirepass
1) "requirepass"
2) "test123"
```

```
redis 127.0.0.1:6379> config set requirepass passabc
OK
redis 127.0.0.1:6379> auth passabc
OK
redis 127.0.0.1:6379> set h helloworld
OK
redis 127.0.0.1:6379> get h
"helloworld"
redis 127.0.0.1:6379> exit
```

注意: config 不能保存到配置文件

```
# grep requirepass /etc/redis.conf

# If the master is password protected (using the "requirepass"
configuration
# requirepass foobared
requirepass test123
```

master/slave 模式, master 有密码, slave 怎样配置?

```
masterauth password
```

2. maxmemory-policy TTL 过期策略配置

- 1、volatile-lru: 只对设置了过期时间的key进行LRU (默认值)
- 2、allkeys-lru : 删除lru算法的key
- 3、volatile-random: 随机删除即将过期key
- 4、allkeys-random: 随机删除
- 5、volatile-ttl : 删除即将过期的
- 6、noeviction : 永不过期, 返回错误

第 34 章 redis-cli - Command-line client to redis-server

1. 命令参数

1.1. password

```
-a <password>      Password to use when connecting to the
server.

[root@netkiller conf.d]# redis-cli -a
hsM8NK8b71vFQKFOS55jbWJrA1TYgI4e
```

1.2. raw

--raw Use raw formatting for replies (default when STDOUT is not a tty).

有时显示这样的数据

```
[hadoop@netkiller ~]$ redis-cli
127.0.0.1:6379> ZREVRANGE FASTNEWS_DATA_STORE_KEY 0 1
1) "
{"title\":"\xe4\xb8\x8a\xe8\xaf\x81\xe7\xbb\xbc\xe6\x8c\x87\xe5\x91\xa8\xe7\xba\xbf\xe4\xba\x94\xe8\xbf\x9e\xe9\x98\xb3\xef\xbc\x8c\xe5\x88\x9b\xe4\xb8\x9a\xe6\x9d\xbf\xe8\xbf\x9e\xe7\xbb\xad\xe4\xb8\xa4\xe5\x91\xa8\xe5\xa4\xa7\xe8\xb7\x8c\xef\xbc\x9b
\xe4\xb8\x8a\xe8\xaf\x81\xe7\xbb\xbc\xe6\x8c\x87\xe5\x91\xa8\xe4\xba\x94\xe6\x94\xb6\xe7\x9b\x98\xe8\xb7\x8c0.23%\xef\xbc\x8c\xe4\xb8\x8b\xe8\xb7\x8c7.60\xe7\x82\xb9\xef\xbc\x8c\xe6\x8a\xa53237.98\xe7\x82\xb9\xef\xbc\x9b
\xe6\xb7\xb1\xe8\xaf\x81\xe6\x88\x90\xe6\x8c\x87\xe5\x91\xa8\xe
```



```

4\xba\x94\xe6\x94\xb6\xe7\x9b\x98\xe8\xb7\x8c0.03%\xef\xbc\x8c\
xe4\xb8\x8b\xe8\xb7\x8c3.50\xe7\x82\xb9\xef\xbc\x8c\xe6\x8a\xa5
10363.48\xe7\x82\xb9\xef\xbc\x9b
\xe6\xb2\xaa\xe6\xb7\xb1300\xe8\x82\xa1\xe6\x8c\x87\xe5\x91\xa8
\xe4\xba\x94\xe6\x94\xb6\xe7\x9b\x98\xe8\xb7\x8c0.52%\xef\xbc\x
8c\xe4\xb8\x8b\xe8\xb7\x8c19.49\xe7\x82\xb9\xef\xbc\x8c\xe6\x8a
\xa53728.39\xe7\x82\xb9\xef\xbc\x9b
\xe5\x88\x9b\xe4\xb8\x9a\xe6\x9d\xbf\xe6\x8c\x87\xe6\x95\xb0\xe
5\x91\xa8\xe4\xba\x94\xe6\x94\xb6\xe7\x9b\x98\xe6\xb6\xa80.08%\
xef\xbc\x8c\xe4\xb8\x8a\xe6\xb6\xa81.31\xe7\x82\xb9\xef\xbc\x8c
\xe6\x8a\xa51689.92\xe7\x82\xb9\\r\\n\"}"
2) "
{"title": "\xe4\xb8\x8a\xe8\xaf\x81\xe7\xbb\xbc\xe6\x8c\x87\x
e5\x91\xa8\xe4\xba\x94\xe6\x94\xb6\xe7\x9b\x98\xe4\xb8\x8b\xe8\
\xb7\x8c0.21%\xef\xbc\x8c\xe6\x8a\xa53237.98\xe7\x82\xb9\xef\xbc
\x9b\xe6\xb7\xb1\xe8\xaf\x81\xe6\x88\x90\xe6\x8c\x87\xe5\x91\xa
8\xe4\xba\x94\xe6\x94\xb6\xe7\x9b\x98\xe5\xbe\xae\xe8\xb7\x8c0.
02%\xef\xbc\x8c\xe6\x8a\xa510364.82\xe7\x82\xb9\xef\xbc\x9b\xe5
\x88\x9b\xe4\xb8\x9a\xe6\x9d\xbf\xe6\x8c\x87\xe6\x95\xb0\xe5\x9
1\xa8\xe4\xba\x94\xe6\x94\xb6\xe7\x9b\x98\xe5\xbe\xae\xe6\xb6\x
a80.09%\xef\xbc\x8c\xe6\x8a\xa51690.15\xe7\x82\xb9\"}"
127.0.0.1:6379>

```

加入 --raw 参数后可以显示可以阅读的数据

```

[hadoop@VM_3_2_centos ~]$ redis-cli --raw
127.0.0.1:6379> ZREVRANGE FASTNEWS_DATA_STORE_KEY 0 1
{"title": "上证综指周线五连阳，创业板连续两周大跌"}
{"title": "上证综指周五收盘下跌0.21%，报3237.98点"}
127.0.0.1:6379>

```

2. --latency Enter a special mode continuously sampling latency.

参数的功能是从客户端发出一条命令到客户端接受到该命令的反馈所用的最长响应时间

```
# redis-cli --latency -h 192.168.2.1  
min: 1, max: 210, avg: 3.64 (13453 samples)
```

3. auth

认证密码

```
[root@netkiller ~]# redis-cli
127.0.0.1:6379> keys *
(error) NOAUTH Authentication required.
127.0.0.1:6379> auth hsM8NKb71vjbWJrAlTYgI4
OK
127.0.0.1:6379> keys *
1) "HK50(1605)"
2) "GBPUSD"
3) "USDCHF"
4) "SP500(1609)"
5) "NZDJPY"
6) "AUDNZD"
7) "EURGBP"
8) "CLN6"
9) "BU6"
```

4. MONITOR

```
$ redis-cli monitor
```

5. info

```
redis 127.0.0.1:6379> info
redis_version:2.4.10
redis_git_sha1:00000000
redis_git_dirty:0
arch_bits:64
multiplexing_api:epoll
gcc_version:4.4.6
process_id:29663
uptime_in_seconds:1189
uptime_in_days:0
lru_clock:1018411
used_cpu_sys:0.10
used_cpu_user:0.09
used_cpu_sys_children:0.01
used_cpu_user_children:0.00
connected_clients:1
connected_slaves:0
client_longest_output_list:0
client_biggest_input_buf:0
blocked_clients:0
used_memory:730664
used_memory_human:713.54K
used_memory_rss:7225344
used_memory_peak:730720
used_memory_peak_human:713.59K
mem_fragmentation_ratio:9.89
mem_allocator:jemalloc-2.2.5
loading:0
aof_enabled:0
changes_since_last_save:0
bgsave_in_progress:0
last_save_time:1373332622
bgrewriteaof_in_progress:0
total_connections_received:4
total_commands_processed:14
expired_keys:0
evicted_keys:0
keyspace_hits:3
keyspace_misses:0
```

```
pubsub_channels:0  
pubsub_patterns:0  
latest_fork_usec:744  
vm_enabled:0  
role:master  
db0:keys=4,expires=0  
redis 127.0.0.1:6379>
```

6. save/bgsave/lastsave

save/bgsave 保存持久化将数据，lastsave 查看相关信息

```
redis 127.0.0.1:6379> save
OK
redis 127.0.0.1:6379> bgsave
Background saving started
redis 127.0.0.1:6379> lastsave
(integer) 1373335757
```

7. config

```
CONFIG GET dir
CONFIG GET dbfilename

config set stop-writes-on-bgsave-error yes
CONFIG SET dir /tmp
CONFIG SET dbfilename temp.rdb
```


8. keys

查询所有key

```
172.18.52.15:6379> keys *
1) "www.example.com:743f10d0f1dc569ed5893856e14c1fb7captcha"
2) "www.example.com:d88e0b6c54a235763dd731bcc0914439captcha"
3)
"www.example.com:17f9091cb44f3cc5bb411eb801f07be8member_login"
4)
"www.example.com:10ff594fd42f4c81212020555cfb586amember_login_i
nput"
5) "www.example.com:a759ba5232ce324d0e6ae8da9290beaecaptcha"
6) "www.example.com:37c78410af02d66a542d15b9707f215bcaptcha"
7)
"www.example.com:9f5070e217f4eac9ald15f9b8dbe7148deposit_1_temp
_var"
8) "www.example.com:6c1a13c9396df2c35613043923bfe338captcha"
9) "www.example.com:b611080c0627154871ea0e1498793238captcha"
10)
"www.example.com:2792241f8d0f075528db2b50e0c9c684member_login"
```

查询指定key

```
172.18.50.15:6379> set name neo
OK
172.18.50.15:6379> keys name
1) "name"
```

9. 字符串操作

9.1. set/get/del

```
172.18.52.15:6379> set name neo
OK
172.18.52.15:6379> get name
"neo"
172.18.52.15:6379> keys name
1) "name"
172.18.52.15:6379> del name
(integer) 1
172.18.52.15:6379> get name
(nil)
```

9.2. setnx

SETNX key value

当 key 不存在时将 key 的值设为 value，若给定的 key 已经存在，则 SETNX 不做任何动作。SETNX 是(SET if Not eXists) (如果不存在，则 SET)的简写。

返回值:

设置成功, 返回 1
设置失败, 返回 0

```
redis> EXISTS neo # neo 不存在
(integer) 0
```

```
redis> SETNX neo "chen" # neo 设置成功
(integer) 1
```

```
redis> SETNX neo "netkiller" # 尝试覆盖 neo , 失败
```

```
(integer) 0
```

```
redis> GET neo  
"chen"
```

```
# 没有被覆盖
```

10. expire/ttl

EXPIRE 设置过期时间, TTL 可以查询过期时间倒计时。

```
172.18.52.165:6379> set name neo
OK
172.18.52.165:6379> ttl name
(integer) -1
172.18.52.165:6379> expire name 30
(integer) 1
172.18.52.165:6379> ttl name
(integer) 22
172.18.52.165:6379> ttl name
(integer) 9
172.18.52.165:6379> ttl name
(integer) -1
172.18.52.165:6379> get name
(nil)
```

注意ttl返回-1有两种情况，一是没有设置过期时间，另一种是该key已经过期不存在。

11. 获取 key 类型

```
root@netkiller ~ % redis-cli  
127.0.0.1:6379> TYPE "logstash:redis"  
list
```

12. LIST 数据类型

获取 list 列表长度

```
127.0.0.1:6379> TYPE "logstash:redis"
list
127.0.0.1:6379> LLEN "logstash:redis"
(integer) 69
127.0.0.1:6379> lpop "logstash:redis"
127.0.0.1:6379>
127.0.0.1:6379> LLEN "logstash:redis"
(integer) 68
```

1. LPUSH/LPUSHX/LRANGE:

/> redis-cli #在Shell提示符下启动redis客户端工具。

```
redis 127.0.0.1:6379> del queue:test
```

```
(integer) 1
```

#queue:test键并不存在，该命令会创建该键及与其关联的List，之后在将参数中的values从左到右依次插入。

```
redis 127.0.0.1:6379> lpush queue:test a b c d
```

```
(integer) 4
```

#取从位置0开始到位置2结束的3个元素。

```
redis 127.0.0.1:6379> lrange queue:test 0 2
```

```
1) "d"
```

```
2) "c"
```

```
3) "b"
```

#取链表中的所有元素，其中0表示第一个元素，-1表示最后一个元素。

```
redis 127.0.0.1:6379> lrange queue:test 0 -1
```

```
1) "d"
```

```
2) "c"
```

```
3) "b"
```

```
4) "a"
```

#queue:test2键此时并不存在，因此该命令将不会进行任何操作，其返回值为0。

```
redis 127.0.0.1:6379> lpushx queue:test2 e
```

```
(integer) 0
```

#可以看到queue:test2没有关联任何List Value。

```
redis 127.0.0.1:6379> lrange queue:test2 0 -1
(empty list or set)
#queue:test键此时已经存在，所以该命令插入成功，并返回链表中当前元素的数量。
redis 127.0.0.1:6379> lpushx queue:test e
(integer) 5
#获取该键的List Value的头部元素。
redis 127.0.0.1:6379> lrange queue:test 0 0
1) "e"

2. LPOP/LLEN:
redis 127.0.0.1:6379> lpush queue:test a b c d
(integer) 4
redis 127.0.0.1:6379> lpop queue:test
"d"
redis 127.0.0.1:6379> lpop queue:test
"c"
#在执行lpop命令两次后，链表头部的两个元素已经被弹出，此时链表中元素的数量是2
redis 127.0.0.1:6379> llen queue:test
(integer) 2

3. LREM/LSET/LINDEX/LTRIM:
#为后面的示例准备测试数据。
redis 127.0.0.1:6379> lpush queue:test a b c d a c
(integer) 6
#从头部(left)向尾部(right)变量链表，删除2个值等于a的元素，返回值为实际删除的数量。
redis 127.0.0.1:6379> lrem queue:test 2 a
(integer) 2
#看出删除后链表中的所有元素。
redis 127.0.0.1:6379> lrange queue:test 0 -1
1) "c"
2) "d"
3) "c"
4) "b"
#获取索引值为1(头部的第二个元素)的元素值。
redis 127.0.0.1:6379> lindex queue:test 1
"d"
#将索引值为1(头部的第二个元素)的元素值设置为新值e。
redis 127.0.0.1:6379> lset queue:test 1 e
OK
#查看是否设置成功。
redis 127.0.0.1:6379> lindex queue:test 1
"e"
```

```
#索引值6超过了链表中元素的数量, 该命令返回nil。
redis 127.0.0.1:6379> lindex queue:test 6
(nil)
#设置的索引值6超过了链表中元素的数量, 设置失败, 该命令返回错误信息。
redis 127.0.0.1:6379> lset queue:test 6 hh
(error) ERR index out of range
#仅保留索引值0到2之间的3个元素, 注意第0个和第2个元素均被保留。
redis 127.0.0.1:6379> ltrim queue:test 0 2
OK
#查看trim后的结果。
redis 127.0.0.1:6379> lrange queue:test 0 -1
1) "c"
2) "e"
3) "c"

4. LINSERT:
#删除该键便于后面的测试。
redis 127.0.0.1:6379> del queue:test
(integer) 1
#为后面的示例准备测试数据。
redis 127.0.0.1:6379> lpush queue:test a b c d e
(integer) 5
#在a的前面插入新元素a1。
redis 127.0.0.1:6379> linsert queue:test before a a1
(integer) 6
#查看是否插入成功, 从结果看已经插入。注意lindex的index值是0-based。
redis 127.0.0.1:6379> lindex queue:test 0
"e"
#在e的后面插入新元素e2, 从返回结果看已经插入成功。
redis 127.0.0.1:6379> linsert queue:test after e e2
(integer) 7
#再次查看是否插入成功。
redis 127.0.0.1:6379> lindex queue:test 1
"e2"
#在不存在的元素之前或之后插入新元素, 该命令操作失败, 并返回-1。
redis 127.0.0.1:6379> linsert queue:test after k a
(integer) -1
#为不存在的key插入新元素, 该命令操作失败, 返回0。
redis 127.0.0.1:6379> linsert queue:test1 after a a2
(integer) 0

5. RPOP/RPUSH/RPUSHX/RPOP/RPOPLPUSH:
#删除该键, 以便于后面的测试。
redis 127.0.0.1:6379> del queue:test
(integer) 1
```



```
#从链表的尾部插入参数中给出的values, 插入顺序是从左到右依次插入。
redis 127.0.0.1:6379> rpush queue:test a b c d
(integer) 4
#通过lrange的可以获悉rpush在插入多值时的插入顺序。
redis 127.0.0.1:6379> lrange queue:test 0 -1
1) "a"
2) "b"
3) "c"
4) "d"
#该键已经存在并且包含4个元素, rpushx命令将执行成功, 并将元素e插入到链表的尾部。
redis 127.0.0.1:6379> rpushx queue:test e
(integer) 5
#通过lindex命令可以看出之前的rpushx命令确实执行成功, 因为索引值为4的元素已经是新元素了。
redis 127.0.0.1:6379> lindex queue:test 4
"e"
#由于queue:test2键并不存在, 因此该命令不会插入数据, 其返回值为0。
redis 127.0.0.1:6379> rpushx queue:test2 e
(integer) 0
#在执行rpoplpush命令前, 先看一下queue:test中链表的元素有哪些, 注意他们的位置关系。
redis 127.0.0.1:6379> lrange queue:test 0 -1
1) "a"
2) "b"
3) "c"
4) "d"
5) "e"
#将queue:test的尾部元素e弹出, 同时再插入到queue:test2的头部(原子性的完成这两步操作)。
redis 127.0.0.1:6379> rpoplpush queue:test queue:test2
"e"
#通过lrange命令查看queue:test在弹出尾部元素后的结果。
redis 127.0.0.1:6379> lrange queue:test 0 -1
1) "a"
2) "b"
3) "c"
4) "d"
#通过lrange命令查看queue:test2在插入元素后的结果。
redis 127.0.0.1:6379> lrange queue:test2 0 -1
1) "e"
#将source和destination设为同一键, 将queue:test中的尾部元素移到其头部。
redis 127.0.0.1:6379> rpoplpush queue:test queue:test
"d"
#查看移动结果。
```

```
redis 127.0.0.1:6379> lrange queue:test 0 -1
```

- 1) "d"
- 2) "a"
- 3) "b"
- 4) "c"

13. set 无序字符集合

Set和List类型不同的是，Set集合中不允许出现重复的元素，和List类型相比，Set类型在功能上还存在着一个非常重要的特性，即在服务器端完成多个Sets之间的聚合计算操作，如unions、intersections和differences。由于这些操作均在服务端完成，因此效率极高，而且也节省了大量的网络IO开销。

1. SADD/SMEMBERS/SCARD/SISMEMBER:

#插入测试数据，由于该键set:test之前并不存在，因此参数中的三个成员都被正常插入。

```
redis 127.0.0.1:6379> sadd set:test a b c
(integer) 3
```

#由于参数中的a在set:test中已经存在，因此本次操作仅仅插入了d和e两个新成员。

```
redis 127.0.0.1:6379> sadd set:test a d e
(integer) 2
```

#判断a是否已经存在，返回值为1表示存在。

```
redis 127.0.0.1:6379> sismember set:test a
(integer) 1
```

#判断f是否已经存在，返回值为0表示不存在。

```
redis 127.0.0.1:6379> sismember set:test f
(integer) 0
```

#通过smembers命令查看插入的结果，从结果可以，输出的顺序和插入顺序无关。

```
redis 127.0.0.1:6379> smembers set:test
```

```
1) "c"
2) "d"
3) "a"
4) "b"
5) "e"
```

#获取Set集合中元素的数量。

```
redis 127.0.0.1:6379> scard set:test
(integer) 5
```

2. SPOP/SREM/SRANDMEMBER/SMOVE:

```
#删除该键，便于后面的测试。
redis 127.0.0.1:6379> del set:test
(integer) 1
#为后面的示例准备测试数据。
redis 127.0.0.1:6379> sadd set:test a b c d
(integer) 4
#查看Set中成员的位置。
redis 127.0.0.1:6379> smembers set:test
1) "c"
2) "d"
3) "a"
4) "b"
#从结果可以看出，该命令确实是随机的返回了某一成员。
redis 127.0.0.1:6379> srandmember set:test
"c"
#Set中尾部的成员b被移出并返回，事实上b并不是之前插入的第一个或最后一个成员。
redis 127.0.0.1:6379> spop set:test
"b"
#查看移出后Set的成员信息。
redis 127.0.0.1:6379> smembers set:test
1) "c"
2) "d"
3) "a"
#从Set中移出a、d和f三个成员，其中f并不存在，因此只有a和d两个成员被移出，返回为2。
redis 127.0.0.1:6379> srem set:test a d f
(integer) 2
#查看移出后的输出结果。
redis 127.0.0.1:6379> smembers set:test
1) "c"
#为后面的smove命令准备数据。
redis 127.0.0.1:6379> sadd set:test a b
(integer) 2
redis 127.0.0.1:6379> sadd set:test2 c d
(integer) 2
#将a从set:test移到set:test2，从结果可以看出移动成功。
redis 127.0.0.1:6379> smove set:test set:test2 a
(integer) 1
#再次将a从set:test移到set:test2，由于此时a已经不是set:test的成员了，因此移动失败并返回0。
redis 127.0.0.1:6379> smove set:test set:test2 a
(integer) 0
```

#分别查看set:test和set:test2的成员，确认移动是否真的成功。

```
redis 127.0.0.1:6379> smembers set:test
1) "b"
redis 127.0.0.1:6379> smembers set:test2
1) "c"
2) "d"
3) "a"
```

3. SDIFF/SDIFFSTORE/SINTER/SINTERSTORE:

#为后面的命令准备测试数据。

```
redis 127.0.0.1:6379> sadd set:test a b c d
(integer) 4
redis 127.0.0.1:6379> sadd set:test2 c
(integer) 1
redis 127.0.0.1:6379> sadd set:test3 a c e
(integer) 3
```

#set:test和set:test2相比，a、b和d三个成员是两者之间的差异成员。再用这个结果继续和set:test3进行差异比较，b和d是set:test3不存在的成员。

```
redis 127.0.0.1:6379> sdiff set:test set:test2 set:test3
1) "d"
2) "b"
```

#将3个集合的差异成员存在在diffkey关联的Set中，并返回插入的成员数量。

```
redis 127.0.0.1:6379> sdiffstore diffkey set:test set:test2
set:test3
(integer) 2
```

#查看一下sdiffstore的操作结果。

```
redis 127.0.0.1:6379> smembers diffkey
1) "d"
2) "b"
```

#从之前准备的数据就可以看出，这三个Set的成员交集只有c。

```
redis 127.0.0.1:6379> sinter set:test set:test2 set:test3
1) "c"
```

#将3个集合中的交集成员存储到与interkey关联的Set中，并返回交集成员的数量。

```
redis 127.0.0.1:6379> sinterstore interkey set:test
set:test2 set:test3
(integer) 1
```

#查看一下sinterstore的操作结果。

```
redis 127.0.0.1:6379> smembers interkey
1) "c"
```

#获取3个集合中的成员的并集。

```
redis 127.0.0.1:6379> sunion set:test set:test2 set:test3
```

```
1) "b"
```

```
2) "c"
```

```
3) "d"
```

```
4) "e"
```

```
5) "a"
```

#将3个集合中成员的并集存储到unionkey关联的set中，并返回并集成员的数量。

```
redis 127.0.0.1:6379> sunionstore unionkey set:test  
set:test2 set:test3
```

```
(integer) 5
```

#查看一下suiionstore的操作结果。

```
redis 127.0.0.1:6379> smembers unionkey
```

```
1) "b"
```

```
2) "c"
```

```
3) "d"
```

```
4) "e"
```

```
5) "a"
```

14. zset (有序集合)

添加到集合

```
root@netkiller ~ % redis-cli -n 16
127.0.0.1:6379[16]> zadd book 1 "Linux"
1
127.0.0.1:6379[16]> zadd book 2 "Java"
1
127.0.0.1:6379[16]> zadd book 3 "Python"
1
127.0.0.1:6379[16]> zadd book 4 "PHP"
1
127.0.0.1:6379[16]>
```

zrange 查看集合内容

```
127.0.0.1:6379[16]> zrange book 0 -1 withscores
1) "Linux"
2) "1"
3) "Java"
4) "2"
5) "Perl"
6) "5"
```

指定开始和结束范围

```
127.0.0.1:6379[16]> zrange book 0 4
Linux
Java
Python
```

```
PHP
127.0.0.1:6379[16]> zrange book 1 4
Java
Python
PHP
127.0.0.1:6379[16]> zrange book 2 3
Python
PHP
```

zrem 删除集合成员

```
127.0.0.1:6379[16]> zadd book 5 "Rabby"
1
127.0.0.1:6379[16]> zrange book 4 5
Perl
Rabby
127.0.0.1:6379[16]> zrem book Rabby
1
127.0.0.1:6379[16]> zrange book 4 5
Perl
127.0.0.1:6379[16]> zrem book PHP Python
2
```

zcard 返回成员数量

```
127.0.0.1:6379[16]> zcard book
3
```




15. Pub/Sub 订阅与发布

redis 提供基本的MQ 功能，下面我们做一个演示

开启第一个终端窗口，订阅first second

```
$ redis-cli
redis 127.0.0.1:6379> SUBSCRIBE first second
Reading messages... (press Ctrl-C to quit)
1) "subscribe"
2) "first"
3) (integer) 1
1) "subscribe"
2) "second"
3) (integer) 2
```

开启第二个终端窗口，分别发送first second

```
$ redis-cli
redis 127.0.0.1:6379> PUBLISH second Hello
(integer) 1
redis 127.0.0.1:6379> PUBLISH first Helloworld!!!
(integer) 1
redis 127.0.0.1:6379> quit
```

现在切换到第一个终端窗口，应该能够看到发送过来的字符串

```
$ redis-cli
redis 127.0.0.1:6379> SUBSCRIBE first second
Reading messages... (press Ctrl-C to quit)
1) "subscribe"
2) "first"
3) (integer) 1
1) "subscribe"
2) "second"
3) (integer) 2
```

1) "message"

2) "second"

3) "Hello"

1) "message"

2) "first"

3) "Helloworld!!!"

16. flushdb 清空 Redis 数据

```
root@netkiller ~ % redis-cli
127.0.0.1:6379> flushdb
OK
127.0.0.1:6379>
```

第 35 章 redis-benchmark 测试工具

redis-benchmark 基准性能测试

```
用法 redis-benchmark [-h <host>][-p ] [-c <clients>][-n ]> [-k <boolean>]
```

选项:

选项	说明
-h <hostname>	主机名 (默认 127.0.0.1)
-p <port>	主机端口 (默认 6379)
-s <socket>	UNIX socket (会覆盖 -h -p 设置的内容)
-a <password>	密码 (密码错误之类不会直接报错, 而是在操作时才会报错, 这时可以使用 Redis 的 AUTH 命令再次认证)
-c <clients>	客户端的并发数量 (默认是50)
-n <requests>	客户端请求总量 (默认是100000)
-d <size>	使用 SET/GET 添加的数据的字节大小 (默认 2)
-dbnum <db>	选择一个数据库进行测试 (默认 0)
-k <boolean>	客户端是否使用keepalive, 1为使用, 0为不使用, (默认为 1)
-r <keyspacelen>	使用 SET/GET/INCR 命令添加数据 key, SADD 添加随机数据, keyspacelen 指定的是添加 键的数量
-P <numreq>	每个请求 pipeline 的数据量 (默认为1, 没有 pipeline)
-q	仅仅显示redis-benchmark的requests per second信息
--csv	将结果按照csv格式输出, 便于后续处理
-l	循环测试
-t <tests>	可以对指定命令进行基准测试
-I	空闲模式 只打开N个空闲连接并等待。

代表256各个客户端同时请求 Redis, 一共执行 20000 次。redis-benchmark会对各类数据结构的命令进行测试, 并给出性能指标:

```
redis-benchmark -c 256 -n 20000
```

第 36 章 Redis Cluster

`CLUSTER INFO` 打印集群的信息

`CLUSTER NODES` 列出集群当前已知的所有节点 (node) , 以及这些节点的相关信息。

`CLUSTER RESET reset` 重置

`CLUSTER SAVECONFIG` 强制节点保存集群当前状态到磁盘上。

`CLUSTER SLOTS` 获得slot在节点上的映射关系

`CLUSTER MEET` 将 ip 和 port 所指定的节点添加到集群当中, 让它成为集群的一份子。

`CLUSTER FORGET` 从集群中移除 node_id 指定的节点。

`CLUSTER REPLICATE` 将当前节点设置为 node_id 指定的节点的从节点。

`CLUSTER SLAVES` 列出该slave节点的master节点

`CLUSTER ADDSLOTS [slot ...]` 将一个或多个槽 (slot) 指派 (assign) 给当前节点。

`CLUSTER DELSLOTS [slot ...]` 移除一个或多个槽对当前节点的指派。

`CLUSTER FLUSHSLOTS` 移除指派给当前节点的所有槽, 让当前节点变成一个没有指派任何槽的节点。

`CLUSTER SETSLOT NODE` 将槽 slot 指派给 node_id 指定的节点, 如果槽已经指派给另一个节点, 那么先让另一个节点删除该槽, 然后再进行指派。

`CLUSTER SETSLOT MIGRATING` 将本节点的槽 slot 迁移到 node_id 指定的节点中。

`CLUSTER SETSLOT IMPORTING` 从 node_id 指定的节点中导入槽 slot 到本节点。

`CLUSTER SETSLOT STABLE` 取消对槽 slot 的导入 (import) 或者迁移 (migrate) 。

`CLUSTER KEYSLOT` 计算键 key 应该被放置在哪个槽上。

`CLUSTER COUNTKEYSINSLOT` 返回槽 slot 目前包含的键值对数量。

`CLUSTER GETKEYSINSLOT` 返回 count 个 slot 槽中的键

`READONLY` 在集群中的salve节点开启只读模式

`READWRITE` 禁止读取请求跳转到集群中的salve节点上clust

第 37 章 Redis 通信协议

1. 切换DB

select n 切换DB， n表示数据库ID

```
# telnet 192.168.41.160 6379
Trying 192.168.41.160...
Connected to 192.168.41.160.
Escape character is '^]'.
select 1
+OK
```

2. 监控

telnet方式

```
# telnet 172.18.52.13 6379
Trying 172.18.52.163...
Connected to 172.18.52.13.
Escape character is '^]'.
MONITOR
+OK
+1425454378.190210 "MONITOR"
+1425454381.165317 "GET" "admin:633"
+1425454381.165725 "SET" "admin:633" "
{"id\":\"633\",\"username\":\"7209\",\"password\":\"eea5981a4f
d021b8d78f8431084ba760\",\"status\":\"N\",\"belong_user_id\":\"
133\",\"level_id\":\"67\",\"create_time\":1425454381,\"session_
id\":\"11s609t9gq8nj7vc94hb1i3s25\"}"
+1425454381.166088 "EXPIRE" "admin:633" "3600"
+1425454387.956387 "GET" "admin:633"
```

使用 nc 监控状态

```
# (echo -en "MONITOR\r\n"; sleep 10) | nc 172.18.52.13 6379
```


第 38 章 phpRedisAdmin

<https://github.com/ErikDubbelboer/phpRedisAdmin>

Example

You can find an example database at
<http://dubbelboer.com/phpRedisAdmin/>

第一种方法

```
git clone https://github.com/ErikDubbelboer/phpRedisAdmin
cd phpRedisAdmin
git clone https://github.com/nrk/premis
```

第二种方法

```
git clone https://github.com/ErikDubbelboer/phpRedisAdmin.git
cd phpRedisAdmin
git submodule init
git submodule update
```

第 39 章 Redis 开发

1. 消息订阅与发布

订阅

```
<?php
$redis = new Redis();
$redis->connect('127.0.0.1',6379);
$channel = $argv[1]; // channel
$redis->subscribe(array('channel'.$channel), 'callback');
function callback($instance, $channelName, $message) {
    echo $channelName, "==>", $message,PHP_EOL;
}
```

发布

```
<?php
$redis = new Redis();
$redis->connect('127.0.0.1',6379);
$channel = $argv[1]; // channel
$msg = $argv[2]; // msg
$redis->publish('channel'.$channel, $msg);
```

第 40 章 **A fast, light-weight proxy for memcached and redis**

<https://github.com/twitter/twemproxy>

第 41 章 FAQ

1. 清空数据库

```
FLUSHDB - Removes data from your connection's CURRENT database.  
FLUSHALL - Removes data from ALL databases.
```

2. (error) MISCONEF Redis is configured to save RDB snapshots

(error) MISCONEF Redis is configured to save RDB snapshots, but it is currently not able to persist on disk. Commands that may modify the data set are disabled, because this instance is configured to report errors during writes if RDB snapshotting fails (stop-writes-on-bgsave-error option). Please check the Redis logs for details about the RDB error.

临时解决方案

```
# redis-cli
127.0.0.1:6379> config set stop-writes-on-bgsave-error no
OK
127.0.0.1:6379> set name neo
OK
127.0.0.1:6379> get name
"neo"
```

原因是数据库持久化写入硬盘出现问题，可能是数据库目录权限不足。

排查方法

```
CONFIG GET dir
CONFIG GET dbfilename

config set stop-writes-on-bgsave-error yes
CONFIG SET dir /tmp
CONFIG SET dbfilename temp.rdb
```

尝试解决

```
# redis-cli
127.0.0.1:6379> CONFIG GET dir
1) "dir"
2) "/"
127.0.0.1:6379> CONFIG GET dbfilename
1) "dbfilename"
2) "dump.rdb"
127.0.0.1:6379> config set stop-writes-on-bgsave-error yes
OK
127.0.0.1:6379> CONFIG SET dir /tmp
OK
127.0.0.1:6379> set test aaaa
OK
127.0.0.1:6379> get test
"aaaa"
```

如果确认是 dir 权限问题，我们就通过修改redis.conf 中得dir配置解决。

部分 V. MongoDB

<http://www.mongodb.org/>

1. FAQ

1.1. MongoDB 3.x 启用认证后恢复数据库需指定 collection

```
# mongorestore -u yourdb dump/  
Enter password:  
  
2017-06-09T11:55:58.566+0800 Failed:  
error connecting to db server: server returned error on SASL  
authentication step: Authentication failed.
```

```
# mongorestore -u yourdb -d yourdb  
dump/yourdb
```

1.2. MongoDB 2.x 早期版本用户管理

```
> use admin  
switched to db admin  
> db.addUser('neo','chen')  
{  
  "user" : "neo",  
  "readOnly" : false,  
  "pwd" : "68ace374737253d87e0ec91d4fcb673d"  
}  
  
> db.system.users.find()  
{ "_id" : ObjectId("4c481404b9db6474d2fcb76f"), "user" : "neo",  
  "readOnly" : false, "pwd" : "68ace374737253d87e0ec91d4fcb673d"  
}
```

```
> db.auth('neo','chen')
1
```

1.3. Failed: netkiller.assets: error reading database: command listCollections requires authentication

```
[root@ecs-3705 ~]# mongorestore dump/
2018-11-05T11:48:08.981+0800    preparing collections to
restore from
2018-11-05T11:48:08.982+0800    Failed: netkiller.assets: error
reading database: command listCollections requires
authentication
```

需要认证，请使用 -u 用户名 -p 密码 -d 数据库 来恢复

```
[root@netkiller ~]# mongorestore -h 127.0.0.1 -u netkiller -p
netkiller -d netkiller dump/netkiller/
```


第 42 章 Install 安装MongoDB

1. CentOS 8 Stream

```
#!/bin/sh

cat << 'EOF' >> /etc/yum.repos.d/mongodb-org-5.0.repo
[mongodb-org-5.0]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-
-org/5.0/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://www.mongodb.org/static/pgp/server-5.0.asc
EOF

dnf install -y mongodb-org-server
dnf install -y mongodb-org-shell
dnf install -y mongodb-org-tools

cp /etc/mongod.conf{,.original}

cat << 'EOF' >> /etc/security/limits.d/20-nofile.conf
mongod soft nofile 65000
mongod hard nofile 65000
EOF

systemctl is-enabled mongod
systemctl start mongod
```

2. MacOS 安装 MongoDB

```
brew install mongodb
```

启动

```
brew services start mongodb
```

3. 二进制tar包安装

Install MongoDB

```
wget http://fastdl.mongodb.org/linux/mongodb-linux-x86_64-1.5.5.tgz  
  
debian:/srv# tar xzf mongodb-linux-x86_64-1.5.5.tgz  
debian:/srv# ln -s mongodb-linux-x86_64-1.5.5 mongodb
```

Create a data directory

By default MongoDB will store data in /data/db, but it won't automatically create that directory. To create it, do:

```
$ sudo mkdir -p /data/db/  
$ sudo chown `id -u` /data/db
```

You can also tell MongoDB to use a different data directory, with the --dbpath option.

Run and connect to the server

First, start the MongoDB server in one terminal:

```
$ ./mongodb/bin/mongod
```

In a separate terminal, start the shell, which will connect to localhost by default:

```
$ ./mongodb/bin/mongo  
> db.foo.save( { a : 1 } )
```

```
> db.foo.find()
```

Congratulations, you've just saved and retrieved your first document with MongoDB!

例 42.1. MongoDB Test

```
debian:/srv/mongodb/bin# ./mongo
MongoDB shell version: 1.5.5
connecting to: test
[initandlisten] Thu Jul 22 16:42:07 connection accepted from
127.0.0.1:42876 #1
> db.foo.save({a:1})
Thu Jul 22 16:42:23 allocating new datafile /data/db/test.ns,
filling with zeroes...
Thu Jul 22 16:42:23 done allocating datafile /data/db/test.ns,
size: 16MB, took 0.025 secs
Thu Jul 22 16:42:23 allocating new datafile /data/db/test.0,
filling with zeroes...
Thu Jul 22 16:42:23 done allocating datafile /data/db/test.0,
size: 64MB, took 0.105 secs
[conn1] Thu Jul 22 16:42:23 building new index on { _id: 1 }
for test.foo
[conn1] Thu Jul 22 16:42:23 Buildindex test.foo idxNo:0 { name:
"_id_", ns: "test.foo", key: { _id: 1 } }
[conn1] Thu Jul 22 16:42:23 done for 0 records 0secs
[conn1] Thu Jul 22 16:42:23 insert test.foo 136ms
> Thu Jul 22 16:42:23 allocating new datafile /data/db/test.1,
filling with zeroes...
Thu Jul 22 16:42:24 done allocating datafile /data/db/test.1,
size: 128MB, took 0.228 secs
> db.foo.find()
{ "_id" : ObjectId("4c48046f74050cbf6c9a0ef6"), "a" : 1 }

> use neo
switched to db neo
> db.foo.save({a:1})
Thu Jul 22 16:54:50 allocating new datafile /data/db/neo.ns,
filling with zeroes...
Thu Jul 22 16:54:50 done allocating datafile /data/db/neo.ns,
size: 16MB, took 0.026 secs
```

```
Thu Jul 22 16:54:50 allocating new datafile /data/db/neo.0,
filling with zeroes...
Thu Jul 22 16:54:50 done allocating datafile /data/db/neo.0,
size: 64MB, took 0.122 secs
[conn1] Thu Jul 22 16:54:50 building new index on { _id: 1 }
for neo.foo
[conn1] Thu Jul 22 16:54:50 Buildindex neo.foo idxNo:0 { name:
"_id_", ns: "neo.foo", key: { _id: 1 } }
Thu Jul 22 16:54:50 allocating new datafile /data/db/neo.1,
filling with zeroes...
[conn1] Thu Jul 22 16:54:50 done for 0 records 0.01secs
[conn1] Thu Jul 22 16:54:50 insert neo.foo 164ms
> Thu Jul 22 16:54:50 done allocating datafile /data/db/neo.1,
size: 128MB, took 0.217 secs

> db.foo.find()
{ "_id" : ObjectId("4c48075a74050cbf6c9a0ef7"), "a" : 1 }
>

> db.neo.save({a:1})
[conn1] Thu Jul 22 16:58:32 building new index on { _id: 1 }
for neo.neo
[conn1] Thu Jul 22 16:58:32 Buildindex neo.neo idxNo:0 { name:
"_id_", ns: "neo.neo", key: { _id: 1 } }
[conn1] Thu Jul 22 16:58:32 done for 0 records 0.024secs
> db.neo.find()
{ "_id" : ObjectId("4c48083874050cbf6c9a0ef8"), "a" : 1 }
```

Starting Mongo

Running as a Daemon

```
$ ./mongod --fork --logpath /var/log/mongodb.log --logappend
```

4. Ubuntu MongoDB

```
$ sudo apt-get install mongodb-server mongodb-clients
```

```
$ /etc/init.d/mongodb  
Usage: /etc/init.d/mongodb {start|stop|force-  
stop|restart|force-reload|status}
```

5. CentOS 7 MongoDB

CentOS 默认 MongoDB 是 2.6.12

```
[root@iz623h9icu8Z ~]# yum info mongodb
Loaded plugins: langpacks
Repodata is over 2 weeks old. Install yum-cron? Or run: yum
makecache fast
Available Packages
Name           : mongodb
Arch           : x86_64
Version        : 2.6.12
Release        : 3.el7
Size           : 43 M
Repo           : epel/x86_64
Summary        : High-performance, schema-free document-oriented
database
URL            : http://www.mongodb.org
License        : AGPLv3 and zlib and ASL 2.0
Description    : Mongo (from "humongous") is a high-performance,
open source, schema-free
                 : document-oriented database. MongoDB is written in
C++ and offers the following
                 : features:
                 :   * Collection oriented storage: easy storage
of object/JSON-style data
                 :   * Dynamic queries
                 :   * Full index support, including on inner
objects and embedded arrays
                 :   * Query profiling
                 :   * Replication and fail-over support
                 :   * Efficient storage of binary data including
large objects (e.g. photos
                 :   and videos)
                 :   * Auto-sharding for cloud-level scalability
(currently in early alpha)
                 :   * Commercial Support Available
                 :
                 : A key goal of MongoDB is to bridge the gap
between key/value stores (which are
                 : fast and highly scalable) and traditional RDBMS
systems (which are deep in
```

```
: functionality).
```

```
# yum install mongodb-server  
# chkconfig mongod on  
# service mongod start
```

单独安装客户端

```
# yum install mongod
```


6. 从官网安装最新版本的 MongoDB 3.4

官网的rpm包是如下

```
[root@netkiller ~]# yum search mongodb | grep "\-org"
mongodb-org.x86_64 : MongoDB open source document-oriented
database system
mongodb-org-mongos.x86_64 : MongoDB sharded cluster query router
mongodb-org-server.x86_64 : MongoDB database server
mongodb-org-shell.x86_64 : MongoDB shell client
mongodb-org-tools.x86_64 : MongoDB tools
```

```
#!/bin/sh
cat << 'EOF' >> /etc/yum.repos.d/mongodb-org-3.4.repo
[mongodb-org-3.4]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-
org/3.4/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://www.mongodb.org/static/pgp/server-3.4.asc
EOF
```

Server

```
yum install -y mongodb-org-server

cp /etc/mongod.conf{,.original}

systemctl is-enabled mongod
systemctl start mongod
```

Client

```
yum install -y mongodb-org-shell
```

```
[root@netkiller ~]# mongo

MongoDB shell version v3.4.0
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.4.0
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
  http://docs.mongodb.org/
Questions? Try the support group
  http://groups.google.com/group/mongodb-user
Server has startup warnings:
2016-11-30T11:34:36.493+0800 I STORAGE [initandlisten]
2016-11-30T11:34:36.560+0800 I CONTROL [initandlisten]
2016-11-30T11:34:36.560+0800 I CONTROL [initandlisten] **
WARNING: Access control is not enabled for the database.
2016-11-30T11:34:36.560+0800 I CONTROL [initandlisten] **
Read and write access to data and configuration is unrestricted.
2016-11-30T11:34:36.560+0800 I CONTROL [initandlisten]
2016-11-30T11:34:36.560+0800 I CONTROL [initandlisten]
2016-11-30T11:34:36.560+0800 I CONTROL [initandlisten] **
WARNING: /sys/kernel/mm/transparent_hugepage/enabled is
'always'.
2016-11-30T11:34:36.560+0800 I CONTROL [initandlisten] **
We suggest setting it to 'never'
2016-11-30T11:34:36.560+0800 I CONTROL [initandlisten]
2016-11-30T11:34:36.560+0800 I CONTROL [initandlisten] **
WARNING: /sys/kernel/mm/transparent_hugepage/defrag is 'always'.
2016-11-30T11:34:36.560+0800 I CONTROL [initandlisten] **
We suggest setting it to 'never'
2016-11-30T11:34:36.560+0800 I CONTROL [initandlisten]
> show dbs
admin 0.000GB
local 0.000GB
> exit
bye
```

工具

```
# yum install mongodb-org-tools

# rpm -ql mongodb-org-tools.x86_64 0:3.4.1-1.el7
/usr/bin/bsondump
/usr/bin/mongodump
/usr/bin/mongoexport
/usr/bin/mongofiles
/usr/bin/mongoimport
/usr/bin/mongooplog
/usr/bin/mongoperf
/usr/bin/mongorestore
/usr/bin/mongostat
/usr/bin/mongotop
/usr/share/man/man1/bsondump.1
/usr/share/man/man1/mongodump.1
/usr/share/man/man1/mongoexport.1
/usr/share/man/man1/mongofiles.1
/usr/share/man/man1/mongoimport.1
/usr/share/man/man1/mongooplog.1
/usr/share/man/man1/mongoperf.1
/usr/share/man/man1/mongorestore.1
/usr/share/man/man1/mongostat.1
/usr/share/man/man1/mongotop.1
```

7. MongoDB + Hadoop

Hadoop Connector

<http://docs.mongodb.org/ecosystem/tutorial/getting-started-with-hadoop/>

```
git clone https://github.com/mongodb/mongo-hadoop.git
git checkout release-1.0
```

```
# vim build.sbt
hadoopRelease in ThisBuild := "cdh4"
```

```
./sbt package
```

```
wget --no-check-certificate
https://github.com/downloads/mongodb/mongo-java-driver/mongo-
2.7.3.jar
cp mongo-2.7.3.jar /usr/lib/hadoop/lib/
cp core/target/mongo-hadoop-core_cdh3u3-1.0.0.jar
/usr/lib/hadoop/lib/
```

待续.....

8. OSCM 一键安装 MongoDB 4.0.2

安装 MongoDB

```
curl -s  
https://raw.githubusercontent.com/oscm/shell/master/database/mongodb/mongo  
db.org/mongodb-4.0.2.sh | bash
```

创建管理和数据库用户

```
# mongo  
  
use admin;  
db.createUser(  
  {  
    user: "admin",  
    pwd: "chen",  
    roles: [ "dbAdmin", "dbOwner", "userAdmin" ]  
  }  
);  
  
use products  
db.createUser(  
  {  
    user: "accountUser",  
    pwd: "password",  
    roles: [ "readWrite", "dbAdmin" ]  
  }  
)  
)
```

开启认证

```
curl -s  
https://raw.githubusercontent.com/oscm/shell/master/database/mongodb/mongo  
db.org/security.authorization.enabled.sh | bash
```

9. Replication

很多教程上面采用手工配置主从复制，我不建议你这样启动，请采用修改/etc/mongod.conf配置文件的方案。

```
创建主：
mongod -port 27017 -dbpath /var/lib/mongoddb -master

创建从：
mongod -port 27017 -dbpath /var/lib/mongoddb -slave -source
master_ip_address:27017
    </screen>

    <section>
        <title>Master</title>
        <screen><![CDATA[
sed -i "s/#master = true/master = true/" /etc/mongod.conf

systemctl restart mongod
    </screen>
    </section>
    <section>
        <title>Slave</title>
        <screen><![CDATA[
sed -i "s/#slave = true/slave = true/" /etc/mongod.conf
sed -i "s/#source = arg/source = mongodb.master.example.com/"
/etc/mongod.conf

systemctl restart mongod
    </screen>
    </section>
    <section>
        <title>测试</title>
        <para>进入 Master</para>
        <screen>
        <![CDATA[
[root@localhost ~]# mongo
MongoDB shell version: 2.6.11
connecting to: test
Welcome to the MongoDB shell.
```

```

For interactive help, type "help".
For more comprehensive documentation, see
    http://docs.mongodb.org/
Questions? Try the support group
    http://groups.google.com/group/mongodb-user
Server has startup warnings:
2015-11-14T15:51:21.215+0800 [initandlisten]
2015-11-14T15:51:21.215+0800 [initandlisten] ** WARNING:
Readahead for /var/lib/mongodb is set to 4096KB
2015-11-14T15:51:21.215+0800 [initandlisten] **           We
suggest setting it to 256KB (512 sectors) or less
2015-11-14T15:51:21.215+0800 [initandlisten] **
http://dochub.mongodb.org/core/readahead
>
>
> db.foo.save({'name':'neo','address':
{'city':'shenzhen','post':518000},'phone':
[13113668890,13322993040]})
WriteResult({ "nInserted" : 1 })
> db.foo.find();
{ "_id" : ObjectId("5646e881a11081d5998bf70c"), "name" : "neo",
"address" : { "city" : "shenzhen", "post" : 518000 }, "phone" :
[ 13113668890, 13322993040 ] }
>

```

进入 Slave

```

[root@localhost ~]# mongo
MongoDB shell version: 2.6.11
connecting to: test
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
    http://docs.mongodb.org/
Questions? Try the support group
    http://groups.google.com/group/mongodb-user
Server has startup warnings:
2015-11-14T15:51:23.668+0800 [initandlisten]
2015-11-14T15:51:23.668+0800 [initandlisten] ** WARNING:
Readahead for /var/lib/mongodb is set to 4096KB
2015-11-14T15:51:23.668+0800 [initandlisten] **           We

```

```
suggest setting it to 256KB (512 sectors) or less
2015-11-14T15:51:23.668+0800 [initandlisten] **
http://dochub.mongodb.org/core/readahead
> db.foo.find();
{ "_id" : ObjectId("5646e881a11081d5998bf70c"), "name" : "neo",
"address" : { "city" : "shenzhen", "post" : 518000 }, "phone" :
[ 13113668890, 13322993040 ] }
>
```


10. Drivers

Using MongoDB in PHP

Installing the PHP Driver

```
sudo pecl install mongo
```

Open your php.ini file and add to it:

```
extension=mongo.so
```

例 42.2. Using MongoDB in PHP

```
[root@subversion html]# cat mongo.php
<?php

// connect
$m = new Mongo('192.168.3.9');

// select a database
$db = $m->comedy;
$collection = $db->cartoons;

// add an element
$obj = array( "title" => "Calvin and Hobbes", "author" => "Bill
Watterson" );
$collection->insert($obj);

// add another element, with a different "shape"
$obj = array( "title" => "XKCD", "online" => true );
$collection->insert($obj);
```

```
// find everything in the collection
$cursor = $collection->find();

// iterate through the results
foreach ($cursor as $obj) {
    echo $obj["title"] . "\n";
}

// disconnect
$m->close();

?>
```

```
[root@subversion html]# php mongo.php
Calvin and Hobbes
XKCD
[root@subversion html]# php mongo.php
Calvin and Hobbes
XKCD
Calvin and Hobbes
XKCD
</screen>
<para></para>
<screen>
<![CDATA[
> use comedy
switched to db comedy
> db.foo.find()
> db.cartoons.find()
{ "_id" : ObjectId("4c481d2b9503c17611000000"), "title" :
"Calvin and Hobbes", "author" : "Bill Watterson" }
{ "_id" : ObjectId("4c481d2b9503c17611010000"), "title" :
"XKCD", "online" : true }
{ "_id" : ObjectId("4c481d2f9503c17711000000"), "title" :
"Calvin and Hobbes", "author" : "Bill Watterson" }
{ "_id" : ObjectId("4c481d2f9503c17711010000"), "title" :
"XKCD", "online" : true }
>
```

第 43 章 MongoDB 管理

1. Security and Authentication

开启用户认证

```
sed -i "32s/#security:/security:/" /etc/mongod.conf  
sed -i "33 i \\ \ authorization: enabled" /etc/mongod.conf
```

权限

数据库用户角色

read: 只读数据权限

readWrite: 读写数据权限

数据库管理角色

dbAdmin: 在当前db中执行管理操作的权限

dbOwner: 在当前db中执行任意操作

userAdmin: 在当前db中管理user的权限

备份和还原角色

backup

restore

跨库角色

readAnyDatabase: 在所有数据库上都有读取数据的权限

readWriteAnyDatabase: 在所有数据库上都有读写数据的权限

userAdminAnyDatabase: 在所有数据库上都有管理user的权限

dbAdminAnyDatabase: 管理所有数据库的权限

集群管理

clusterAdmin: 管理机器的最高权限

clusterManager: 管理和监控集群的权限

clusterMonitor: 监控集群的权限

hostManager: 管理Server

超级权限

root: 超级用户

超级管理员

Database Administration Roles

```
use admin
db.createUser(
  {
    user: "admin",
    pwd: passwordPrompt(), // or cleartext password
    roles: [ { role: "userAdminAnyDatabase", db: "admin" },
"readWriteAnyDatabase" ]
  }
)
```

```
db.createUser(
  {
    user: "eos",
    pwd: "eos", // or cleartext password
    roles: [ { role: "userAdmin", db: "eos" }, "readWrite" ]
  }
)
```

数据库访问用户

注意，只有创建了超级管理后，下面的操作才会生效

MongoDB

```
use products
db.createUser(
  {
    user: "accountUser",
    pwd: "password",
    roles: [ "readWrite", "dbAdmin" ]
  }
)
```

早期版本

```
> use neo
switched to db neo
> db.addUser('neo','chen')
{
  "user" : "neo",
  "readOnly" : false,
  "pwd" : "68ace374737253d87e0ec91d4fcb673d"
}

> db.system.users.find()
{ "_id" : ObjectId("4c481404b9db6474d2fcb76f"), "user" : "neo",
"readOnly" : false, "pwd" : "68ace374737253d87e0ec91d4fcb673d" }

> db.auth('neo','chen')
1
```

数据库监控用户

```
db.createUser(
  {
    user: "monitor",
    pwd: "netkiller",
    roles: [ "clusterMonitor" ]
  }
)
```

删除用户

Deleting Users 删除用户

To delete a user:

```
> db.getUsers();
[
  {
    "_id" : "test.monitor",
    "user" : "monitor",
    "db" : "test",
```

```
        "roles" : [
          {
            "role" : "dbOwner",
            "db" : "test"
          }
        ]
      }
    ]
]
> db.dropUser('monitor')
ture
> db.getUsers();
[ ]
```

更新角色

```
db.updateUser( "monitor",
  {
    roles: [ "read", "clusterMonitor" ]
  }
)
```

2.4.0早期旧版本

开启认证

```
# vim /etc/mongodb.conf  
auth = true
```

重载配置文件

```
# /etc/init.d/mongod reload  
Stopping mongod: [ OK ]  
Starting mongod: [ OK ]
```

```
use admin;  
db.createUser(  
  {  
    user: "admin",  
    pwd: "WkAFdmfVQpP1oAEkz4YV1MCDxkG36TAi",  
    roles: [ "dbAdmin", "dbOwner", "userAdmin" ]  
  }  
);
```

早期版本删除用户

```
db.system.users.remove( { user: username } )
```



第 44 章 命令工具

1. mongo - MongoDB Shell

eval

```
# mongo
MongoDB shell version: 2.2.3
connecting to: test
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
    http://docs.mongodb.org/
Questions? Try the support group
    http://groups.google.com/group/mongodb-user
>
```

3.4

```
[root@netkiller ~]# mongo
MongoDB shell version v3.4.1
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.4.1
Server has startup warnings:
2017-01-03T11:26:57.516+0800 I CONTROL [initandlisten]
2017-01-03T11:26:57.516+0800 I CONTROL [initandlisten] **
WARNING: Access control is not enabled for the database.
2017-01-03T11:26:57.516+0800 I CONTROL [initandlisten] **
Read and write access to data and configuration is
unrestricted.
2017-01-03T11:26:57.516+0800 I CONTROL [initandlisten]
2017-01-03T11:26:57.516+0800 I CONTROL [initandlisten]
2017-01-03T11:26:57.516+0800 I CONTROL [initandlisten] **
WARNING: /sys/kernel/mm/transparent_hugepage/enabled is
'always'.
```

```
2017-01-03T11:26:57.516+0800 I CONTROL [initandlisten] **
We suggest setting it to 'never'
2017-01-03T11:26:57.516+0800 I CONTROL [initandlisten]
2017-01-03T11:26:57.516+0800 I CONTROL [initandlisten] **
WARNING: /sys/kernel/mm/transparent_hugepage/defrag is
'always'.
2017-01-03T11:26:57.516+0800 I CONTROL [initandlisten] **
We suggest setting it to 'never'
2017-01-03T11:26:57.516+0800 I CONTROL [initandlisten]
>
```

```
# mongo 127.0.0.1:27017/admin --eval "db.stats()"
```

help

help

```
db.help() help on DB methods
db.foo.help() help on collection methods
```

登陆认证

```
# mongo -u<user> -p<password> --authenticationDatabase <db>
<host>/<db>
```

管道操作

```
cat data.bson | mongo test
```



2. mongodump - Backup

本地备份

如果没有开启用户认证

```
root@ubuntu:~/neo# mongodump -d eos --gzip
```

开启用户认证

```
# mongodump -uneo -p -d test -o /tmp/  
connected to: 127.0.0.1  
Enter password:  
Tue Sep 8 10:12:33.011 DATABASE: test to /tmp/test  
Tue Sep 8 10:12:33.012 test.system.indexes to  
/tmp/test/system.indexes.bson  
Tue Sep 8 10:12:33.043 12 objects  
Tue Sep 8 10:12:33.043 test.bios to /tmp/test/bios.bson  
Tue Sep 8 10:12:33.043 1 objects  
Tue Sep 8 10:12:33.043 Metadata for test.bios to  
/tmp/test/bios.metadata.json  
Tue Sep 8 10:12:33.043 test.system.users to  
/tmp/test/system.users.bson  
Tue Sep 8 10:12:33.044 2 objects  
Tue Sep 8 10:12:33.044 Metadata for test.system.users to  
/tmp/test/system.users.metadata.json  
Tue Sep 8 10:12:33.044 test.fs.chunks to  
/tmp/test/fs.chunks.bson  
Tue Sep 8 10:12:33.045 2 objects  
Tue Sep 8 10:12:33.045 Metadata for test.fs.chunks to  
/tmp/test/fs.chunks.metadata.json  
Tue Sep 8 10:12:33.045 test.fs.files to /tmp/test/fs.files.bson  
Tue Sep 8 10:12:33.046 2 objects  
Tue Sep 8 10:12:33.046 Metadata for test.fs.files to  
/tmp/test/fs.files.metadata.json
```

```
Tue Sep 8 10:12:33.046 test.images.chunks to
/tmp/test/images.chunks.bson
Tue Sep 8 10:12:33.167 12 objects
Tue Sep 8 10:12:33.167 Metadata for test.images.chunks to
/tmp/test/images.chunks.metadata.json
Tue Sep 8 10:12:33.167 test.images.files to
/tmp/test/images.files.bson
Tue Sep 8 10:12:33.168 3 objects
Tue Sep 8 10:12:33.168 Metadata for test.images.files to
/tmp/test/images.files.metadata.json
Tue Sep 8 10:12:33.168 test.img.chunks to
/tmp/test/img.chunks.bson
Tue Sep 8 10:12:33.175 4 objects
Tue Sep 8 10:12:33.175 Metadata for test.img.chunks to
/tmp/test/img.chunks.metadata.json
Tue Sep 8 10:12:33.175 test.img.files to
/tmp/test/img.files.bson
Tue Sep 8 10:12:33.176 1 objects
Tue Sep 8 10:12:33.176 Metadata for test.img.files to
/tmp/test/img.files.metadata.json
```

通过指定dbpath在本地导出bson文件

```
mongodump --dbpath /var/lib/mongodb --out /opt/backup --db test
--username backup --password passwd
```

远程备份

短参数

```
mongodump -h mongodb.example.net -p 27017 -u neo -p password -d
netkiller -c yourcollection
```

长参数

```
mongodump --host mongodb.example.net --port 27017 --username  
backup --password passwd --db mdb --collection some
```

3. mongorestore

本地恢复

直接从dump恢复备份

```
[root@netkiller www]# ls
backup dump

[root@netkiller www]# mongorestore dump/
```

```
mongorestore --dbpath /var/lib/mongodb --journal /opt/backu
```

用户认账

```
[root@netkiller ~]# mongorestore -ueos -peos --
authenticationDatabase=eos --gzip dump/
```

远程恢复

```
[root@netkiller www]# mongorestore -h 127.0.0.1 -u neo -p chen
/tmp/test/
connected to: 127.0.0.1
Tue Sep 8 10:18:31.360 /tmp/test/system.users.bson
Tue Sep 8 10:18:31.360 going into namespace [test.system.users]
Tue Sep 8 10:18:31.361 warning: Restoring to test.system.users
```

```
without dropping. Restored data will be inserted without
raising errors; check your server log
2 objects found
Tue Sep 8 10:18:31.361 Creating index: { key: { _id: 1 }, ns:
"test.system.users", name: "_id_" }
Tue Sep 8 10:18:31.406 Creating index: { key: { user: 1,
userSource: 1 }, unique: true, ns: "test.system.users", name:
"user_1_userSource_1" }
Tue Sep 8 10:18:31.406 /tmp/test/img.chunks.bson
Tue Sep 8 10:18:31.406 going into namespace [test.img.chunks]
Tue Sep 8 10:18:31.407 warning: Restoring to test.img.chunks
without dropping. Restored data will be inserted without
raising errors; check your server log
4 objects found
Tue Sep 8 10:18:31.409 Creating index: { name: "_id_", key: {
_id: 1 }, ns: "test.img.chunks" }
Tue Sep 8 10:18:31.409 Creating index: { name:
"files_id_1_n_1", key: { files_id: 1, n: 1 }, unique: true, ns:
"test.img.chunks" }
Tue Sep 8 10:18:31.409 /tmp/test/fs.files.bson
Tue Sep 8 10:18:31.409 going into namespace [test.fs.files]
Tue Sep 8 10:18:31.410 warning: Restoring to test.fs.files
without dropping. Restored data will be inserted without
raising errors; check your server log
2 objects found
Tue Sep 8 10:18:31.410 Creating index: { name: "_id_", key: {
_id: 1 }, ns: "test.fs.files" }
Tue Sep 8 10:18:31.410 /tmp/test/images.chunks.bson
Tue Sep 8 10:18:31.410 going into namespace
[test.images.chunks]
Tue Sep 8 10:18:31.411 warning: Restoring to test.images.chunks
without dropping. Restored data will be inserted without
raising errors; check your server log
12 objects found
Tue Sep 8 10:18:31.414 Creating index: { name: "_id_", key: {
_id: 1 }, ns: "test.images.chunks" }
Tue Sep 8 10:18:31.414 Creating index: { name:
"files_id_1_n_1", key: { files_id: 1, n: 1 }, unique: true, ns:
"test.images.chunks" }
Tue Sep 8 10:18:31.414 /tmp/test/images.files.bson
Tue Sep 8 10:18:31.414 going into namespace [test.images.files]
Tue Sep 8 10:18:31.414 warning: Restoring to test.images.files
without dropping. Restored data will be inserted without
raising errors; check your server log
3 objects found
```



```
Tue Sep 8 10:18:31.415 Creating index: { name: "_id_", key: {
_id: 1 }, ns: "test.images.files" }
Tue Sep 8 10:18:31.415 /tmp/test/fs.chunks.bson
Tue Sep 8 10:18:31.415 going into namespace [test.fs.chunks]
Tue Sep 8 10:18:31.415 warning: Restoring to test.fs.chunks
without dropping. Restored data will be inserted without
raising errors; check your server log
2 objects found
Tue Sep 8 10:18:31.416 Creating index: { name: "_id_", key: {
_id: 1 }, ns: "test.fs.chunks" }
Tue Sep 8 10:18:31.416 Creating index: { name:
"files_id_1_n_1", key: { files_id: 1, n: 1 }, unique: true, ns:
"test.fs.chunks" }
Tue Sep 8 10:18:31.416 /tmp/test/img.files.bson
Tue Sep 8 10:18:31.416 going into namespace [test.img.files]
Tue Sep 8 10:18:31.417 warning: Restoring to test.img.files
without dropping. Restored data will be inserted without
raising errors; check your server log
1 objects found
Tue Sep 8 10:18:31.417 Creating index: { name: "_id_", key: {
_id: 1 }, ns: "test.img.files" }
Tue Sep 8 10:18:31.417 /tmp/test/bios.bson
Tue Sep 8 10:18:31.417 going into namespace [test.bios]
Tue Sep 8 10:18:31.417 warning: Restoring to test.bios without
dropping. Restored data will be inserted without raising
errors; check your server log
1 objects found
Tue Sep 8 10:18:31.417 Creating index: { key: { _id: 1 }, ns:
"test.bios", name: "_id_" }
```

恢复到指定数据库

```
# mongorestore -h 127.0.0.1 -d test123 /tmp/test
```

```
mongorestore --host mongodb.example.net --port 27017 --username
backup --password password --db test --collection some
/data/backup
```

filter

如果只想恢复部分数据，可以使用--filter

```
$ mongorestore -h 127.0.0.1 -d test123 /tmp/test --filter  
'{"field": 1}'
```

4. mongostat

```
# mongostat
connected to: 127.0.0.1
insert query update delete getmore command flushes mapped vsize
res faults locked db idx miss % qr|qw ar|aw netIn netOut conn
time
*0 *0 *0 *0 0 1|0 0 848m 1.92g 162m 0 wechat:0.0% 0 0|0 0|0 62b
4k 1 10:38:53
*0 *0 *0 *0 0 1|0 0 848m 1.92g 162m 0 wechat:0.0% 0 0|0 0|0 62b
4k 1 10:38:54
*0 *0 *0 *0 0 1|0 0 848m 1.92g 162m 0 wechat:0.0% 0 0|0 0|0 62b
4k 1 10:38:55
*0 *0 *0 *0 0 1|0 0 848m 1.92g 162m 0 wechat:0.0% 0 0|0 0|0 62b
4k 1 10:38:56
*0 *0 *0 *0 0 1|0 0 848m 1.92g 162m 0 wechat:0.0% 0 0|0 0|0 62b
4k 1 10:38:57
```

5. mongotop

```
# mongotop
connected to: 127.0.0.1

ns total read write 2015-09-08T02:23:46
passport.system.users 0ms 0ms 0ms
passport.system.namespaces 0ms 0ms 0ms
passport.system.indexes 0ms 0ms 0ms
member.system.users 0ms 0ms 0ms
member.system.namespaces 0ms 0ms 0ms
member.system.indexes 0ms 0ms 0ms
```

6. mongofiles - Browse and modify a GridFS filesystem.

list 浏览文件

```
# mongofiles list
connected to: 127.0.0.1
/etc/passwd 2176
/tmp/test1.php 192
```

put 上传文件

```
# mongofiles put /bin/ls
connected to: 127.0.0.1
added file: { _id: ObjectId('55ee4c68bd053b7418404c53'),
filename: "/bin/ls", chunkSize: 261120, uploadDate: new
Date(1441680488106), md5: "ca226dd605e91b72e0d2060a6357c28f",
length: 109208 }
done!

# mongofiles list
connected to: 127.0.0.1
/etc/passwd 2176
/tmp/test1.php 192
/bin/ls 109208
```

上传指定数据库

```
# mongofiles put -d images -c img /etc/fstab
connected to: 127.0.0.1
added file: { _id: ObjectId('55ee4d5416377f58d0a9e714'),
filename: "/etc/fstab", chunkSize: 261120, uploadDate: new
Date(1441680724579), md5: "381185dc0c4807b88406b452b4acc2e8",
length: 1067 }
done!

# mongofiles list -d images -c img
```

```
connected to: 127.0.0.1
/etc/fstab 1067
```

collection 参数有 bug 需要注意。

-c img 似乎无效, 可能是mongofiles的bug. 使用PHP测试上传是可以指定collection, 并且没有任何问题。

```
# mongofiles put -d images --collection abc /etc/nfsmount.conf
connected to: 127.0.0.1
added file: { _id: ObjectId('55ee4f5ef4b26bc3189dc8a5'),
filename: "/etc/nfsmount.conf", chunkSize: 261120, uploadDate:
new Date(1441681246083), md5:
"ce3b9fee8612087cbb69d46db34ce8ec", length: 3605 }
done!
```

```
# mongofiles -d images --collection abc list
connected to: 127.0.0.1
/etc/fstab      1067
/etc/passwd     2555
/etc/goaccess.conf 6956
/etc/krb5.conf  449
/etc/nfsmount.conf 3605
```

```
# mongo images
> show collections;
abc.fs.chunks
abc.fs.files
fs.chunks
fs.files
system.indexes
>
> db.abc.fs.files.find();
>
```

使用 --collection 参数可以看到abc已经创建, 但我们去 db.abc.fs.files.find();发现里面没有任何数据, 文件仍然被上传到 abc.fs.files

get 下载

如果 /tmp/test123 存在则会覆盖

```
# mongofiles get /tmp/test123
connected to: 127.0.0.1
done write to: /tmp/test123
```

-l 指定路径，相当于另存。

```
# mongofiles get /tmp/test123 -l /tmp/aabbcc
connected to: 127.0.0.1
done write to: /tmp/aabbcc
```

delete 删除

```
# mongofiles list
connected to: 127.0.0.1
/etc/passwd 2176
/tmp/test1.php 192
/bin/ls 109208
/tmp/test123 6

# mongofiles delete /tmp/test123
connected to: 127.0.0.1
done!

# mongofiles list
connected to: 127.0.0.1
/etc/passwd 2176
/tmp/test1.php 192
/bin/ls 109208
```

第 45 章 MongoDB Shell

1. shutdownServer

关闭MongoDB数据库

```
db.shutdownServer()
```


2. show 查看命令

show dbs

show dbs show database names

```
> show dbs
local    (empty)
logging  0.203125GB
test     0.203125GB
```

show collections

show collections show collections in current database

```
> show collections
bios
system.indexes
```

另一种用法是show tables

```
> show tables
bios
system.indexes
```

show users

show users show users in current database



show profile

show profile show most recent system.profile entries with time >= 1ms

```
> show profile
db.system.profile is empty
Use db.setProfilingLevel(2) will enable profiling
Use db.system.profile.find() to show raw profile entries
```

3. 切换数据库

```
use <db name>                set curent database to <db name>
> use logging
switched to db logging
```

4. save

存储嵌套的对象

```
db.foo.save({'name': 'neo', 'address':  
{ 'city': 'shenzhen', 'post': 518000 }, 'phone':  
[ 13113668890, 13322993040 ]})
```

存储数组对象

```
db.foo.save({'Uid': 'netkiller@msn.com', 'phone':  
[ '13322993040', '13113668890' ]})
```

5. insert

```
db.bios.insert(
  {
    _id: 1,
    name: { first: 'John', last: 'Backus' },
    birth: new Date('Dec 03, 1924'),
    death: new Date('Mar 17, 2007'),
    contribs: [ 'Fortran', 'ALGOL', 'Backus-Naur Form', 'FP'
  ],
  awards: [
    {
      award: 'W.W. McDowell Award',
      year: 1967,
      by: 'IEEE Computer Society'
    },
    {
      award: 'National Medal of Science',
      year: 1975,
      by: 'National Science Foundation'
    },
    {
      award: 'Turing Award',
      year: 1977,
      by: 'ACM'
    },
    {
      award: 'Draper Prize',
      year: 1993,
      by: 'National Academy of Engineering'
    }
  ]
}
)
```

6. update

根据query条件修改，如果不存在则插入，允许修改多条记录

```
db.foo.update({'yy':5},{'$set':
{'xx':2}},upsert=true,multi=true)
```

multi 更新所有数据

update 第一个参数是条件，当不写条件时将匹配所有数据。

```
db.getCollection('certificate').update({},{'$set':
{'icon':'52bfbb7d92b3f41da2e4103f1990c054990be863.png'}},upsert
=false,multi=true)
```

upsert 更新，如果不存在则插入数据

```
db.getCollection('shippingAddress').update({'memberId':'0000000
0'},{'$set':{'defaults': false}},upsert=true,multi=true)
```

7. remove

删除uid=10的记录

```
db.foo.remove({'uid':10})
```

删除所有的记录

```
db.foo.remove()
```

删除条件使用 **`_id`**

```
db.foo.remove({ "_id" :  
ObjectId("56e10b66a22ef1b1408b4567")})  
  
db.getCollection('goods').remove({ "_id":  
ObjectId("5bbdbd197099aa06abf6fb1a")})
```

8. 删除 collection

```
db.collection.drop()
```

删除字段

```
db.getCollection('table').update({}, {$unset: {field: 1}})
```


9. count()

```
> db.access.count()  
51528  
> db.access.count()  
104401
```

10. 查询

find() MongoDB 2.x

查找所有 所有记录

```
db.foo.find() list objects in collection foo
db.foo.find( { a : 1 } ) list objects in foo
where a == 1
```

查找一条记录

```
db.foo.findOne()
```

根据条件检索10条记录

```
db.foo.find({'name':'neo'}).limit(10)
```

sort排序

```
db.foo.find({'name':'neo'}).sort({'Dt',-1})
db.foo.find().sort({'Ct':-1}).limit(1)
```

count记录统计操作

```
db.foo.count()
```

distinct操作,去重复查询指定列,

```
db.foo.distinct('name')
```

”>=”操作

```
db.foo.find({"timestamp": {"$gte" : 2}})
```

子对象的查找

```
db.foo.find({'address.city': 'shenzhen'})
```

find() MongoDB 3.x

```
db.getCollection('tracker').find({name:"81004892"})
```

Query

包含字段

```
db.getCollection('pyramidSelling').find({}, {'phone':1})
```

排除字段

```
db.getCollection('pyramidSelling').find({}, {'phone':0})
```

sort()

```
db.getCollection('tracker').find({name:"81004892"}).sort({ctime: -1})
```

group()

group()类似SQL中的Group by

```
> db.test.group({key: {remote_addr: true}, initial: {count: 0}, reduce:
function(obj, prev) {prev.count++}});
[
  {
    "remote_addr" : "192.168.2.76",
    "count" : 3
  },
  {
    "remote_addr" : "192.168.2.70",
    "count" : 1
  }
]
```

11. aggregate

project

\$split

```
ObjectId("591a710320156761bdf68a06"),
"mis.domain.PyramidSelling",
...
...
"status" : true,
"createdDate" :
ISODate("2017-05-16T03:24:51.511Z")
}
```

```
db.getCollection('pyramidSelling').aggregate([
  { $project : { _class : { $split: ["$_class", "."] } } }
]);
```

substr

```
db.getCollection('pyramidSelling').aggregate(
  [
    {
      $project: {
        userName: 1,
        phone: {
          prefix: { $substr: [
```

```
"$phone", 0, 3 ] },
                                mobile: { $substr: [
"$phone", 3, 11 ] }
                                },
                                }
                                ]
                                )
```

groupby + sum

select username, sum(balance) as total from users group by member.

```
db.member.aggregate([ {
  $group: {
    _id: "$username",
    total: { $sum: "$balance" }
  }
}])
```

12. Indexes 索引

增加索引: 1(ascending),-1(descending)

查看索引

```
db.getCollection('product').getIndexes()
```

```
[
  {
    "v" : 2,
    "key" : {
      "_id" : 1
    },
    "name" : "_id_",
    "ns" : "netkiller.product"
  },
  {
    "v" : 2,
    "unique" : true,
    "key" : {
      "uuid" : 1
    },
    "name" : "uuid",
    "ns" : "netkiller.product",
    "sparse" : true
  },
  {
    "v" : 2,
    "key" : {
      "nfc" : 1
    },
    "name" : "nfc",
    "ns" : "netkiller.product"
  },
  {
    "v" : 2,
    "unique" : true,
    "key" : {
      "qrcode" : 1
    },
    "name" : "qrcode",
```

```
    "ns" : "netkiller.product",
    "sparse" : true
  },
  {
    "v" : 2,
    "key" : {
      "memberId" : 1
    },
    "name" : "memberId",
    "ns" : "netkiller.product"
  },
  {
    "v" : 2,
    "unique" : true,
    "key" : {
      "transactionId" : 1
    },
    "name" : "transactionId",
    "ns" : "netkiller.product",
    "sparse" : true
  }
]
```

查看索引信息

```
db.logging.getIndexes()
[
  {
    "v" : 1,
    "key" : {
      "_id" : 1
    },
    "ns" : "logging.logging",
    "name" : "_id_"
  }
]
```

查看索引名与排序方式

```
db.getCollection('member').getIndexKeys();
[
```



```
{
  "_id" : 1
},
{
  "mobile" : 1
},
{
  "username" : 1
},
{
  "wechat" : 1
}
]
```

创建索引

增加索引

```
db.foo.ensureIndex({firstname: 1, lastname: 1}, {unique: true});
```

索引子对象

```
db.logging.users.ensureIndex({address.city:1})
```

删除索引

```
db.getCollection('product').dropIndex("memberId")
```

根据索引名删除索引

```
> db.logging.users.dropIndex('name_1')
```

```
{ "nIndexesWas" : 2, "ok" : 1 }
> db.logging.users.getIndexKeys()
[ { "_id" : 1 } ]
```

唯一索引

```
db.members.createIndex( { "user_id": 1 }, { unique: true } )
```

```
> db.apple.createIndex({"devicetoken":1},{unique: true})
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
```

复合索引

```
db.getCollection('foo').ensureIndex({"address":1,"phone":1})
```

稀疏索引

```
db.getCollection('article').ensureIndex({"uuid": 1}, {"unique":
true,"sparse":true});
```

作用,唯一索引只允许一条索引字段为空的记录存在,之后就不允许插入了。再次插入为 null 的记录时会报错:

```
E11000 duplicate key error index: dup key: { : null };
```

“sparse”的作用就是当 uuid 在文档中不存在，或为空值，则不进入索引，从而避免上述问题。

13. Map-Reduce

使用 Map-Reduce 统计Web 服务器 access.log 日志文件

首先将web服务器access.log导入到mongodb,参考<http://netkiller.github.io/article/log.html> 格式如下:

```
{
  "_id" : ObjectId("51553efcd8616be7e5395c0d"),
  "remote_addr" : "192.168.2.76",
  "remote_user" : "-",
  "time_local" : "29/Mar/2013:09:20:31 +0800",
  "request" : "GET /tw/ad.jpg HTTP/1.1",
  "status" : "200",
  "body_bytes_sent" : "5557",
  "http_referer" : "http://www.example.com/tw/",
  "http_user_agent" : "Mozilla/5.0 (Windows NT 6.1;
WOW64) AppleWebKit/537.17 (KHTML, like Gecko)
Chrome/24.0.1312.57 Safari/537.17",
  "http_x_forwarded_for" : "-"
}
```

创建map方法

```
var mapFunction1 = function() {
  emit(this.remote_addr, {count:1});
};
```

创建reduce方法

```
var reduceFunction1 = function(key, values) {
  var total = 0;
  values.forEach(function (value) {total +=
value.count;});
  return {ipaddr: key, count:total};
};
```

```
};
```

分析数据

```
db.access.mapReduce(mapFunction1, reduceFunction1, {out :  
"resultCollection"});
```

输出结果

```
db.resultCollection.find();
```

Demo 数据库

```
> db.resultCollection.find();  
{ "_id" : "192.168.2.109", "value" : { "ipaddr" :  
"192.168.2.109", "count" : 554 } }  
{ "_id" : "192.168.2.38", "value" : { "ipaddr" :  
"192.168.2.38", "count" : 26 } }  
{ "_id" : "192.168.2.39", "value" : { "ipaddr" :  
"192.168.2.39", "count" : 72 } }  
{ "_id" : "192.168.2.40", "value" : { "ipaddr" :  
"192.168.2.40", "count" : 3564 } }  
{ "_id" : "192.168.2.49", "value" : { "ipaddr" :  
"192.168.2.49", "count" : 955 } }  
{ "_id" : "192.168.2.5", "value" : { "ipaddr" : "192.168.2.5",  
"count" : 2 } }  
{ "_id" : "192.168.2.76", "value" : { "ipaddr" :  
"192.168.2.76", "count" : 60537 } }  
{ "_id" : "192.168.3.12", "value" : { "ipaddr" :  
"192.168.3.12", "count" : 9577 } }  
{ "_id" : "192.168.3.14", "value" : { "ipaddr" :  
"192.168.3.14", "count" : 343 } }  
{ "_id" : "192.168.3.18", "value" : { "ipaddr" :  
"192.168.3.18", "count" : 1006 } }  
{ "_id" : "192.168.3.26", "value" : { "ipaddr" :  
"192.168.3.26", "count" : 2714 } }
```

```
{ "_id" : "192.168.6.19", "value" : { "ipaddr" :  
"192.168.6.19", "count" : 668 } }  
{ "_id" : "192.168.6.2", "value" : { "ipaddr" : "192.168.6.2",  
"count" : 123760 } }  
{ "_id" : "192.168.6.30", "value" : { "ipaddr" :  
"192.168.6.30", "count" : 1196 } }  
{ "_id" : "192.168.6.35", "value" : { "ipaddr" :  
"192.168.6.35", "count" : 1050 } }  
>
```

14. 内嵌对象

Array / List 列表类型

```
db.foo.save(  
{  
  'name': 'neo',  
  'address': [{ 'city': 'shenzhen', 'post': "518000" },  
{ 'city': 'heilongjiang', 'post': "135000" }],  
  'phone': ["13113668800", "13322993040", "13266884444"]  
}  
)
```

```
{  
  "_id" : ObjectId("5bbefed1b04a8c0d7395d1b5"),  
  "name" : "neo",  
  "address" : [  
    {  
      "city" : "shenzhen",  
      "post" : "518000"  
    },  
    {  
      "city" : "heilongjiang",  
      "post" : "135000"  
    }  
  ],  
  "phone" : [  
    "13113668800",  
    "13322993040",  
    "13266884444"  
  ]  
}
```

删除数组元素

```
db.getCollection('foo').update({"_id":ObjectId("5bbeff40b04a8c0d7395d1b6")}, {"$pull":{"phone":"13266884444"}})
```

删除数组中的对象

```
db.getCollection('foo').update({"_id":ObjectId("5bbeff40b04a8c0d7395d1b6")}, {"$pull":{"address":{"city":"heilongjiang"}}})
```

查找替换

```
db.getCollection('foo').update({"address.city":"shenzhen"}, {"$set":{"address.$.post":"000000"}})
```


15. Javascript 脚本

```
db.numbers.drop()

var counter = 0
while (counter<=1000){
  db.numbers.save({"value":counter})
  counter = counter + 1;
}
```

第 46 章 Mongo Admin UI

<http://docs.mongodb.org/ecosystem/tools/administration-interfaces/>

1. RockMongo

<http://code.google.com/p/rock-php/>

2. MongoVUE

<http://blog.mongovue.com/>

MongoVUE 是收费软件，提供的功能比较完善，可以免费试用

第 47 章 Cassandra

<http://incubator.apache.org/cassandra/>

1. Getting Started

1.1. Downloading and Installation

```
$ cd /srv/
```

```
$ cd /usr/local/src/  
  
$ sudo wget -c  
http://apache.freelamp.com/cassandra/0.5.1/apache-cassandra-  
0.5.1-bin.tar.gz  
$ sudo tar zxvf apache-cassandra-0.5.1-bin.tar.gz  
$ cp -r /usr/local/src/apache-cassandra-0.5.1 /srv/  
$ cd /srv/  
$ sudo ln -s apache-cassandra-0.5.1 apache-cassandra  
$ cd apache-cassandra
```

1.2. Running Cassandra

Running Cassandra

```
$ bin/cassandra  
$ Listening for transport dt_socket at address: 8888  
INFO - Saved Token not found. Using  
70882909557229809272696372631016976044  
INFO - Starting up server gossip
```

1.3. cli tool

cli

\$ bin/cassandra-cli

```
neo@db:/srv/apache-cassandra$ bin/cassandra-cli
Welcome to cassandra CLI.

Type 'help' or '?' for help. Type 'quit' or 'exit' to quit.
cassandra>
```

```
cassandra> connect localhost/9160
Connected to localhost/9160
```

1.4. Testing Cassandra

test

```
cassandra> show keyspaces
Keyspace1
system
```

insert value

```
cassandra> set Keyspace1.Standard1['member']['name']='neo'
Value inserted.
cassandra> set Keyspace1.Standard1['member']['age']='27'
```

```
Value inserted.
cassandra> set Keyspace1.Standard1['member']
['email']='openunix@163.com'
Value inserted.
cassandra>
cassandra> get Keyspace1.Standard1['member']
=> (column=name, value=neo, timestamp=1271070497471)
=> (column=email, value=openunix@163.com,
timestamp=1271070498334)
=> (column=age, value=27, timestamp=1271070497519)
Returned 3 results.
cassandra>
```

2. Configure Cassandra

2.1. Environment variables

```
CASSANDRA_HOME=/srv/apache-cassandra
```

2.2. log4j.properties

```
[root@db apache-cassandra]# vim conf/log4j.properties
log4j.appender.R=org.apache.log4j.RollingFileAppender
log4j.appender.file.maxFileSize=20MB
log4j.appender.file.maxBackupIndex=50
log4j.appender.R.layout=org.apache.log4j.PatternLayout
log4j.appender.R.layout.ConversionPattern=%5p [%t] %d{ISO8601}
%F (line %L) %m%n
# Edit the next line to point to your logs directory
log4j.appender.R.File=/var/log/cassandra/system.log

# Application logging options
#log4j.logger.com.facebook=DEBUG
#log4j.logger.com.facebook.infrastructure.gms=DEBUG
#log4j.logger.com.facebook.infrastructure.db=DEBUG
```

2.3. storage-conf.xml

```
[root@db apache-cassandra]# vim conf/storage-conf.xml
```

3. Keyspace

3.1. Schema

Keyspace

Column family

Name

Column

Super column

Sorting

3.2. Keyspace example

例 47.1. Twitter

```
<Keyspace Name="Twitter">
<ColumnFamily CompareWith="UTF8Type" Name="Statuses" />
<ColumnFamily CompareWith="UTF8Type" Name="StatusAudits" />
<ColumnFamily CompareWith="UTF8Type" Name="StatusRelationships"
CompareSubcolumnsWith="TimeUUIDType" ColumnType="Super" />
<ColumnFamily CompareWith="UTF8Type" Name="Users" />
<ColumnFamily CompareWith="UTF8Type" Name="UserRelationships"
CompareSubcolumnsWith="TimeUUIDType" ColumnType="Super" />
</Keyspace>
```

例 47.2. Twissandra

```
<Keyspaces>
```



```

<Keyspace Name="Twissandra">
  <ColumnFamily CompareWith="UTF8Type" Name="User"/>
  <ColumnFamily CompareWith="BytesType" Name="Username"/>
  <ColumnFamily CompareWith="BytesType" Name="Friends"/>
  <ColumnFamily CompareWith="BytesType" Name="Followers"/>
  <ColumnFamily CompareWith="UTF8Type" Name="Tweet"/>
  <ColumnFamily CompareWith="LongType" Name="Timeline"/>
  <ColumnFamily CompareWith="LongType" Name="Userline"/>

<ReplicaPlacementStrategy>org.apache.cassandra.locator.RackUnawareStrategy</ReplicaPlacementStrategy>

  <!-- Number of replicas of the data -->
  <ReplicationFactor>1</ReplicationFactor>

<EndPointSnitch>org.apache.cassandra.locator.EndPointSnitch</EndPointSnitch>

</Keyspace>
</Keyspaces>

```

Schema Layout

In Cassandra, the way that your data is structured is very closely tied to how it will be retrieved. Let's start with the user ColumnFamily. The key is a user id, and the columns are the properties on the user:

```

User = {
  'a4a70900-24e1-11df-8924-001ff3591711': {
    'id': 'a4a70900-24e1-11df-8924-001ff3591711',
    'username': 'ericflo',
    'password': '****',
  },
}

```

Since some of the URLs on the site actually have the username, we need to be able to map from the username to the user id:

```
Username = {
  'ericflo': {
    'id': 'a4a70900-24e1-11df-8924-001ff3591711',
  },
}
```

Friends and followers are keyed by the user id, and then the columns are the friend user id and follower user ids, and we store a timestamp as the value because it's interesting information to have:

```
Friends = {
  'a4a70900-24e1-11df-8924-001ff3591711': {
    # friend id: timestamp of when the friendship was added
    '10cf667c-24e2-11df-8924-001ff3591711':
'1267413962580791',
    '343d5db2-24e2-11df-8924-001ff3591711':
'1267413990076949',
    '3f22b5f6-24e2-11df-8924-001ff3591711':
'1267414008133277',
  },
}

Followers = {
  'a4a70900-24e1-11df-8924-001ff3591711': {
    # friend id: timestamp of when the followership was
added
    '10cf667c-24e2-11df-8924-001ff3591711':
'1267413962580791',
    '343d5db2-24e2-11df-8924-001ff3591711':
'1267413990076949',
    '3f22b5f6-24e2-11df-8924-001ff3591711':
'1267414008133277',
  },
}
```

Tweets are stored in a way similar to users:

```
Tweet = {
```

```

    '7561a442-24e2-11df-8924-001ff3591711': {
      'id': '89da3178-24e2-11df-8924-001ff3591711',
      'user_id': 'a4a70900-24e1-11df-8924-001ff3591711',
      'body': 'Trying out Twissandra. This is awesome!',
      '_ts': '1267414173047880',
    },
  }
}

```

The Timeline and Userline column families keep track of which tweets should appear, and in what order. To that effect, the key is the user id, the column name is a timestamp, and the column value is the tweet id:

```

Timeline = {
  'a4a70900-24e1-11df-8924-001ff3591711': {
    # timestamp of tweet: tweet id
    1267414247561777: '7561a442-24e2-11df-8924-001ff3591711',
    1267414277402340: 'f0c8d718-24e2-11df-8924-001ff3591711',
    1267414305866969: 'f9e6d804-24e2-11df-8924-001ff3591711',
    1267414319522925: '02ccb5ec-24e3-11df-8924-001ff3591711',
  },
}

Userline = {
  'a4a70900-24e1-11df-8924-001ff3591711': {
    # timestamp of tweet: tweet id
    1267414247561777: '7561a442-24e2-11df-8924-001ff3591711',
    1267414277402340: 'f0c8d718-24e2-11df-8924-001ff3591711',
    1267414305866969: 'f9e6d804-24e2-11df-8924-001ff3591711',
    1267414319522925: '02ccb5ec-24e3-11df-8924-001ff3591711',
  },
}

```

4. Cluster

4.1. Running a cluster

```
<Seed>127.0.0.1</Seed>
```

改为

```
<Seed>172.16.0.1</Seed>
```

```
<ListenAddress>localhost</ListenAddress>  
改为:  
<ListenAddress>172.16.0.1</ListenAddress>
```

```
<ThriftAddress>localhost</ThriftAddress>  
改为:  
<ThriftAddress>0.0.0.0</ThriftAddress>
```

\$ bin/cassandra

4.2. Running a single node

```
<Seed>127.0.0.1</Seed>
```

改为

```
<Seed>172.16.0.2</Seed>
```

```
<Seeds>
  <Seed>172.16.0.1</Seed>
  <Seed>172.16.0.2</Seed>
  <Seed>172.16.0.3</Seed>
  <Seed>172.16.0.4</Seed>
  <Seed>172.16.0.5</Seed>
</Seeds>
```

```
<ListenAddress>localhost</ListenAddress>
改为:
<ListenAddress>172.16.0.2</ListenAddress>
```

```
<ThriftAddress>localhost</ThriftAddress>
改为:
<ThriftAddress>0.0.0.0</ThriftAddress>
```

\$ bin/cassandra

4.3. nodetool

```
nodeprobe -host 172.16.0.1 ring
```

第 48 章 Hypertable

<http://hypertable.org/>

1. Hypertable 安装

Hypertable 的几种安装方式

单机：安装于单机，采用本地文件系统

Hadoop：分布式安装，使用Hadoop(HDFS)作为存储

MapR：分布式安装，在MapR之上

Ceph：分布式安装，在Ceph之上

1.1. Hypertable standalone 单机安装

过程 48.1. Hypertable standalone 安装过程

1. 安装 Hypertable 软件包

```
# cd /usr/local/src/  
# wget  
http://cdn.hypertable.com/packages/0.9.7.0/hypertable-  
0.9.7.0-linux-x86_64.rpm
```

2. 安装 Hypertable，我个人比较喜欢用yum localinstall他会解决软件之间的依赖关系

```
# yum localinstall hypertable-0.9.7.0-linux-x86_64.rpm
```



```
=====
Install          13 Package(s)
```

3. Hypertable 默认安装在 /opt/hypertable/0.9.7.0

备份配置文件,

```
# cd /opt/hypertable/0.9.7.0/conf
# cp hypertable.cfg hypertable.cfg.original
```

4. FHS-IZE 安装

```
# bin/fhsize.sh
Setting up /var/opt/hypertable
Setting up /etc/opt/hypertable
fhsize /opt/hypertable/0.9.7.0: success
```

5. 设计 "CURRENT" 连接

```
# cd /opt/hypertable
# ln -s 0.9.7.0 current
```

6. 安装 notification-hook.sh 脚本.

```
# cp conf/notification-hook.tmpl conf/notification-hook.sh
# chmod o+x conf/notification-hook.sh
```

测试 notification-hook.sh脚本 .

```
/opt/hypertable/current/conf/notification-hook.sh "Test
Message" "This is a test."
```

7. 启动 hypertable

```
# /opt/hypertable/current/bin/start-all-servers.sh local
DFS broker: available file descriptors: 1024
Started DFS Broker (local)
Started Hyperspace
Started Hypertable.Master
/proc/sys/vm/swappiness = 60
Started Hypertable.RangeServer
Started ThriftBroker
```

```
# /opt/hypertable/current/bin/ht shell

Welcome to the hypertable command interpreter.
For information about Hypertable, visit
http://hypertable.com

Type 'help' for a list of commands, or 'help shell' for a
list of shell meta commands.

hypertable>
```

8. 测试安装是否有效

```
# /opt/hypertable/current/bin/ht shell

Welcome to the hypertable command interpreter.
For information about Hypertable, visit
http://hypertable.com

Type 'help' for a list of commands, or 'help shell' for a
list of shell meta commands.

hypertable> help

USE ..... Sets the current namespace
```

```
CREATE NAMESPACE ... Creates a new namespace
DROP NAMESPACE ..... Removes a namespace
EXISTS TABLE ..... Check if table exists
CREATE TABLE ..... Creates a table
DELETE ..... Deletes all or part of a row from a
table
DESCRIBE TABLE ..... Displays a table's schema
DROP TABLE ..... Removes a table
RENAME TABLE ..... Renames a table
DUMP TABLE ..... Create efficient backup file
ALTER TABLE ..... Add/remove column family from existing
table
INSERT ..... Inserts data into a table
LOAD DATA INFILE ... Loads data from a TSV input file into
a table
SELECT ..... Selects (and display) cells from a
table
SHOW CREATE TABLE .. Displays CREATE TABLE command used to
create table
SHOW TABLES ..... Displays only the list of tables in
the current namespace
GET LISTING ..... Displays the list of tables and
namespace in the current namespace

Statements must be terminated with ';'. For more
information on
a specific statement, type 'help <statement>', where
<statement> is from
the preceding list.

hypertable>quit
```

9. 停止 hypertable

运行下列命令停止 Hypertable

```
$ /opt/hypertable/current/bin/stop-servers.sh
```

1.2. Hypertable on HDFS(hadoop) 安装

Hadoop - HDFS 安装指南

过程 48.2. Hypertable on HDFS

1. 创建工作目录

```
$ hadoop fs -mkdir /hypertable
$ hadoop fs -chmod 777 /hypertable
```

2. 安装 Java 运行环境

```
yum install java-1.7.0-openjdk
yum localinstall
http://ftp.cuhk.edu.hk/pub/packages/apache.org/hadoop/commo
n/hadoop-1.1.2/hadoop-1.1.2-1.x86_64.rpm
```

3. 修改 jrun bug

```
cp /opt/hypertable/current/bin/jrun
/opt/hypertable/current/bin/jrun.old

vim /opt/hypertable/current/bin/jrun
#HT_JAR=`ls -l /opt/hypertable/doug/current/lib/java/*.jar
| grep "hypertable-[^-]*.jar" | awk 'BEGIN {FS="/"} {print
$NF}'`
HT_JAR=`ls -l /opt/hypertable/current/lib/java/*.jar | grep
"hypertable-[^-]*.jar" | awk 'BEGIN {FS="/"} {print $NF}'`
```

```
export JAVA_HOME=/usr
export HADOOP_HOME=/usr
export HYPERTABLE_HOME=/opt/hypertable/current
```

4. hypertable.cfg

```
# cat conf/hypertable.cfg
#
# hypertable.cfg
#
# HDFS Broker
#HdfsBroker.Hadoop.ConfDir=/etc/hadoop/conf
HdfsBroker.Hadoop.ConfDir=/etc/hadoop
# Ceph Broker
CephBroker.MonAddr=192.168.6.2:6789
# Local Broker
DfsBroker.Local.Root=fs/local
# DFS Broker - for clients
DfsBroker.Port=38030
# Hyperspace
Hyperspace.Replica.Host=localhost
Hyperspace.Replica.Port=38040
Hyperspace.Replica.Dir=hyperspace
# Hypertable.Master
#Hypertable.Master.Host=localhost
Hypertable.Master.Port=38050
# Hypertable.RangeServer
Hypertable.RangeServer.Port=38060
Hyperspace.KeepAlive.Interval=30000
Hyperspace.Lease.Interval=1000000
Hyperspace.GracePeriod=200000
# ThriftBroker
ThriftBroker.Port=38080
```

Hadoop 配置文件 /etc/hadoop/core-site.xml

```
# cat /etc/hadoop/core-site.xml
```

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!-- Put site-specific property overrides in this file. -->

<configuration>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://namenode.example.com:9000</value>
  </property>
  <property>
    <name>hadoop.tmp.dir</name>
    <value>/var/tmp/hadoop</value>
  </property>
</configuration>
```

Hadoop 配置文件 /etc/hadoop/hdfs-site.xml

```
# cat /etc/hadoop/hdfs-site.xml
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!-- Put site-specific property overrides in this file. -->

<configuration>
  <property>
    <name>dfs.name.dir</name>
    <value>/var/hadoop/name1</value>
    <description> </description>
  </property>
  <property>
    <name>dfs.data.dir</name>
    <value>/var/hadoop/hdfs/data1</value>
    <description> </description>
  </property>
  <property>
    <name>dfs.replication</name>
    <value>2</value>
  </property>
</configuration>
```

5. 启动 dfsbroker

```
# /opt/hypertable/current/bin/set-hadoop-distro.sh cdh4
Hypertable successfully configured for Hadoop cdh4
```

```
# /opt/hypertable/current/bin/start-dfsbroker.sh hadoop
DFS broker: available file descriptors: 1024
Started DFS Broker (hadoop)
```

查看启动日志

```
# tail -f /opt/hypertable/current/log/DfsBroker.hadoop.log
log4j:WARN No appenders could be found for logger
(org.apache.hadoop.conf.Configuration).
log4j:WARN Please initialize the log4j system properly.
HdfsBroker.dfs.client.read.shortcircuit=false
HdfsBroker.dfs.replication=2
HdfsBroker.Server.fs.default.name=hdfs://namenode.example.c
om:9000
Apr 23, 2013 6:43:18 PM org.hypertable.AsyncComm.IOHandler
DeliverEvent
INFO: [/192.168.6.25:53556 ; Tue Apr 23 18:43:18 HKT 2013]
Connection Established
Apr 23, 2013 6:43:18 PM
org.hypertable.DfsBroker.hadoop.ConnectionHandler handle
INFO: [/192.168.6.25:53556 ; Tue Apr 23 18:43:18 HKT 2013]
Disconnect - COMM broken connection : Closing all open
handles from /192.168.6.25:53556
Closed 0 input streams and 0 output streams for client
connection /192.168.6.25:53556
```

1.3. MapR

1.4. Ceph

修改 CephBroker.MonAddr 对应的服务器与端口号即可

```
# cat hypertable.cfg
#
# hypertable.cfg
#
# HDFS Broker
HdfsBroker.Hadoop.ConfDir=/etc/hadoop/conf
# Ceph Broker
CephBroker.MonAddr=192.168.6.2:6789
# Local Broker
DfsBroker.Local.Root=fs/local
# DFS Broker - for clients
DfsBroker.Port=38030
# Hyperspace
Hyperspace.Replica.Host=localhost
Hyperspace.Replica.Port=38040
Hyperspace.Replica.Dir=hyperspace
# Hypertable.Master
Hypertable.Master.Port=38050
# Hypertable.RangeServer
Hypertable.RangeServer.Port=38060
Hyperspace.KeepAlive.Interval=30000
Hyperspace.Lease.Interval=1000000
Hyperspace.GracePeriod=200000
# ThriftBroker
ThriftBroker.Port=38080
```

启动 dfsbroker

```
# /opt/hypertable/current/bin/start-dfsbroker.sh ceph
```


1.5. 检验安装

创建一个表

```
# echo "USE '/'; CREATE TABLE foo ( c1, c2 ); GET LISTING;" \  
> | /opt/hypertable/current/bin/ht shell --batch  
foo  
sys      (namespace)  
tmp      (namespace)
```

插入一些数据

```
# echo "USE '/'; INSERT INTO foo VALUES('001', 'c1', 'very'), \  
> ('000', 'c1', 'Hypertable'), ('001', 'c2', 'easy'),  
( '000', 'c2', 'is');" \  
> | /opt/hypertable/current/bin/ht shell --batch
```

查询数据

```
# echo "USE '/'; SELECT * FROM foo;" \  
> | /opt/hypertable/current/bin/ht shell --batch  
000      c1      Hypertable  
000      c2      is  
001      c1      very  
001      c2      easy
```

如果你想清楚所有表运行下面命令

```
$ /opt/hypertable/current/bin/stop-servers.sh
```

```
$ /opt/hypertable/current/bin/clean-database.sh
```

2. Code examples

http://hypertable.com/documentation/code_examples/

2.1. PHP

设置环境变量

```
export PHPTHRIFT_ROOT=/opt/hypertable/current/lib/php
```

安装PHP环境

```
# yum install php-cli
```

建立测试文件

```
# vim lib/php/test.php
<?php
if (!isset($GLOBALS['THRIFT_ROOT']))
    $GLOBALS['THRIFT_ROOT'] = getenv('PHPTHRIFT_ROOT');

require_once dirname(__FILE__).'/ThriftClient.php';

$client = new Hypertable_ThriftClient("localhost", 38080);
$namespace = $client->namespace_open("");

echo "HQL examples\n";
print_r($client->hql_query($namespace, "show tables"));
print_r($client->hql_query($namespace, "select * from foo"));
```

运行测试程序


```
    )
    [value] => is
  )
[2] => Hypertable_ThriftGen_Cell Object
(
  [key] => Hypertable_ThriftGen_Key Object
  (
    [row] => 001
    [column_family] => c1
    [column_qualifier] =>
    [timestamp] => 1361518099519878003
    [revision] => 1361518099519878003
    [flag] => 255
  )
  [value] => very
)
[3] => Hypertable_ThriftGen_Cell Object
(
  [key] => Hypertable_ThriftGen_Key Object
  (
    [row] => 001
    [column_family] => c2
    [column_qualifier] =>
    [timestamp] => 1361518099519878004
    [revision] => 1361518099519878004
    [flag] => 255
  )
  [value] => easy
)
)
[scanner] =>
[mutator] =>
)
```

3. HQL

3.1. namespace 命名空间管理

```
hypertable> create namespace test;

    Elapsed time:  0.22 s

hypertable> use test;

    Elapsed time:  0.00 s

hypertable> drop namespace test;

    Elapsed time:  1.55 s
```

3.2. Table 表

创建表

```
CREATE TABLE logging (
    uuid,
    tag,
    asctime,
    facility,
    priority,
    message,
    operator
)
```

```
hypertable> CREATE TABLE logging (
    -> uuid,
    -> tag,
```

```
-> asctime,  
-> facility,  
-> priority,  
-> message,  
-> operator  
-> );
```

```
Elapsed time: 2.14 s
```

列出所有表

```
hypertable> show tables;  
logging
```

```
Elapsed time: 0.00 s  
hypertable>
```

显示创建该表的HQL

```
hypertable> show create table logging;
```

```
CREATE TABLE logging (  
  uuid,  
  tag,  
  asctime,  
  facility,  
  priority,  
  message,  
  operator,  
  ACCESS GROUP default (uuid, tag, asctime, facility, priority,  
message, operator)  
)
```

```
Elapsed time: 0.00 s  
hypertable>
```

删除表

```
hypertable> drop table logging;  
  
Elapsed time: 1.33 s
```


4. FAQ

4.1. 切换 DFS Broker

从local切换到ceph

```
/opt/hypertable/current/bin/stop-servers.sh ;  
/opt/hypertable/current/bin/start-dfsbroker.sh local ;  
/opt/hypertable/current/bin/clean-database.sh  
rm -rf /opt/hypertable/current/hyperspace/*  
/opt/hypertable/current/fs/*  
/opt/hypertable/current/run/rsml_backup/*  
/opt/hypertable/current/run/last-dfs
```

启用ceph

```
# /opt/hypertable/current/bin/start-dfsbroker.sh ceph  
DFS broker: available file descriptors: 1024  
Waiting for DFS Broker (ceph) (localhost:38030) to come up...
```

第 49 章 CouchBase

Membase + CouchDB = CouchBase, CouchBase是Membase + CouchDB两个项目合并而来。

1. 安装 CouchBase

进入 <http://www.couchbase.com/download> 找到适合你的版本，然后使用yum install 安装，我个人习惯使用yum而不是rpm，因为yum可以解决包之间的依赖问题。

```
# yum install
http://packages.couchbase.com/releases/2.2.0/couchbase-server-
community_2.2.0_x86_64.rpm
```

CouchBase 安装后会启动起来，同时启动脚本也做了设置

```
# chkconfig couchbase-server --list
couchbase-server      0:off   1:off   2:on    3:on    4:on
5:on   6:off
```

Web 管理界面<http://localhost:8091/index.html>

1.1. Getting Started with Couchbase on PHP

安装C开发包

```
# wget -O/etc/yum.repos.d/couchbase.repo
http://packages.couchbase.com/rpm/couchbase-centos62-
x86_64.repo
# yum install -y libcouchbase-devel
```

安装PHP扩展

```
# pecl search couchbase
Retrieving data...0%
Matched packages, channel pecl.php.net:
=====
Package    Stable/(Latest) Local
couchbase  1.2.2 (stable)      Couchbase Server PHP extension

# pecl install couchbase
```

配置扩展

```
cat > /srv/php/etc/conf.d/couchbase.ini <<EOF
extension=couchbase.so
EOF
```

测试代码

```
<?php
// adjust these parameters to match your installation
$cb = new Couchbase("127.0.0.1:8091", "", "", "default");
$cb->set("a", 101);
var_dump($cb->get("a"));
?>
```

```
# php test.php
int(101)
```

2. couchbase 命令

2.1. couchbase-cli

```
couchbase-cli server-list -c 192.168.0.1:8091 -u Administrator  
-p password --output=json
```

```
# couchbase-cli server-list -c 192.168.2.16:8091 -u  
Administrator -p password  
ns_1@127.0.0.1 192.168.2.16:8091 healthy active
```

第 50 章 Memcached

1. 安装 Memcached

1.1. CentOS 下编译

libevent

```
# yum install libevent libevent-devel -y
```

memcache

```
# wget http://memcached.googlecode.com/files/memcached-1.4.5.tar.gz
# tar xzf memcached-1.4.5.tar.gz
# cd memcached-1.4.5
# ./configure --prefix=/usr/local/memcached-1.4.5
# make && make install
```

start

```
# ln -s /usr/local/memcached-1.4.5 /usr/local/memcached
# /usr/local/memcached/bin/memcached -d -m 128 -p 11211 -u nobody -l 172.16.0.1
```

1.2. Ubuntu 下编译安装

<http://www.monkey.org/~provos/libevent/>

```
cd /usr/local/src/
wget http://www.monkey.org/~provos/libevent-1.4.13-
stable.tar.gz
tar zxf libevent-1.4.13-stable.tar.gz
cd libevent-1.4.13-stable
./configure --prefix=/usr/local/libevent-1.4.13-stable
make
make install
make verify

ln -s /usr/local/libevent-1.4.13-stable /usr/local/libevent
ln -s /usr/local/libevent/lib/* /usr/lib/
ln -s /usr/local/libevent/include/* /usr/include/
ln -s /usr/local/libevent/lib/* /usr/local/lib/
ln -s /usr/local/libevent/include/* /usr/local/include/
```

<http://www.danga.com/memcached/>

```
cd /usr/local/src/
wget http://memcached.googlecode.com/files/memcached-
1.4.5.tar.gz
tar zxf memcached-1.4.5.tar.gz
cd memcached-1.4.5
./configure --prefix=/usr/local/memcached-1.4.5 --with-
libevent=/usr/local/libevent
make
make install

ln -s /usr/local/memcached-1.4.5/ /usr/local/memcached
ln -s /usr/local/memcached/bin/memcached /usr/sbin/memcached
```

**/usr/local/memcached/bin/memcached -d -m 2048 -l 127.0.0.1 -p 11211 -
u root -c 15000 -P /tmp/memcached.pid**

例 50.1. /etc/init.d/memcached

```
#!/bin/bash
# memcached init file for memcached
```

```

#
# chkconfig: - 100 100
# description: a distributed memory object caching system
# author: Neo Chen<openunix@163.com>
#
# processname: /usr/sbin/memcached
# config:
# pidfile: /var/run/memcached

# source function library
. /etc/init.d/functions

OPTIONS="-d -m 2048 -l 127.0.0.1 -p 11211 -u root -c 4096 -P
/var/run/memcached"
USER=daemon
RETVAL=0
prog="memcached"

start() {
    echo -n "Starting $prog: "
    if [ $UID -ne 0 ]; then
        RETVAL=1
        failure
    else
        daemon --user=$USER /usr/sbin/memcached
$OPTIONS
        RETVAL=$?
        [ $RETVAL -eq 0 ] && touch
/var/lock/subsys/memcached
    fi;
    echo
    return $RETVAL
}

stop() {
    echo -n "Stopping $prog: "
    if [ $UID -ne 0 ]; then
        RETVAL=1
        failure
    else
        killproc /usr/sbin/memcached
        RETVAL=$?
        [ $RETVAL -eq 0 ] && rm -f
/var/lock/subsys/memcached
    fi;
}

```

```
        echo
        return $RETVAL
    }

reload(){
    echo -n $"Reloading $prog: "
    killproc /usr/sbin/memcached -HUP
    RETVAL=$?
    echo
    return $RETVAL
}

restart(){
    stop
    start
}

condrestart(){
    [ -e /var/lock/subsys/memcached ] && restart
    return 0
}

case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    restart)
        restart
        ;;
# reload)
#     reload
#     ;;
    condrestart)
        condrestart
        ;;
    status)
        status memcached
        RETVAL=$?
        ;;
    *)
        echo $"Usage: $0
{start|stop|status|restart|condrestart}"
```



```
RETVAL=1
esac
exit $RETVAL
```

`/etc/init.d/memcached`

```
chmod +x /etc/init.d/memcached
```

`flush_all`指令清空memcache中的数据

```
$ telnet 172.16.3.51 11511
Trying 172.16.3.51...
Connected to 172.16.3.51.
Escape character is '^]'.
flush_all
OK
quit
Connection closed by foreign host.
```

1.3. debian/ubuntu

```
$ sudo apt-get install memcache
```

`/etc/memcached.conf`

```
$ cat /etc/memcached.conf
# memcached default config file
# 2003 - Jay Bonci <jaybonci@debian.org>
# This configuration file is read by the start-memcached script
provided as
# part of the Debian GNU/Linux distribution.
```

```
# Run memcached as a daemon. This command is implied, and is
not needed for the
# daemon to run. See the README.Debian that comes with this
package for more
# information.
-d

# Log memcached's output to /var/log/memcached
logfile /var/log/memcached.log

# Be verbose
# -v

# Be even more verbose (print client commands as well)
# -vv

# Start with a cap of 64 megs of memory. It's reasonable, and
the daemon default
# Note that the daemon will grow to this size, but does not
start out holding this much
# memory
-m 64

# Default connection port is 11211
-p 11211

# Run the daemon as root. The start-memcached will default to
running as root if no
# -u command is present in this config file
-u nobody

# Specify which IP address to listen on. The default is to
listen on all IP addresses
# This parameter is one of the only security measures that
memcached has, so make sure
# it's listening on a firewalled interface.
-l 127.0.0.1

# Limit the number of simultaneous incoming connections. The
daemon default is 1024
# -c 1024

# Lock down all paged memory. Consult with the README and
homepage before you do this
# -k
```

```
# Return error when memory is exhausted (rather than removing
items)
# -M

# Maximize core file limit
# -r
```

restart

```
$ sudo /etc/init.d/memcached restart
```

1.4. yum install

CentOS 6.x

```
# yum install memcached
# chkconfig memcached on
# chkconfig --list memcached

# cat /etc/sysconfig/memcached
PORT="11211"
USER="memcached"
MAXCONN="1024"
CACHESIZE="64"
OPTIONS=""

# /etc/init.d/memcached start
Starting memcached:
OK ]
```

CentOS 7.x

```
[root@netkiller ~]# yum install -y memcached
[root@netkiller ~]# rpm -ql memcached.x86_64
/etc/sysconfig/memcached
/usr/bin/memcached
/usr/bin/memcached-tool
/usr/lib/systemd/system/memcached.service
/usr/share/doc/memcached-1.4.15
/usr/share/doc/memcached-1.4.15/AUTHORS
/usr/share/doc/memcached-1.4.15/CONTRIBUTORS
/usr/share/doc/memcached-1.4.15/COPYING
/usr/share/doc/memcached-1.4.15/ChangeLog
/usr/share/doc/memcached-1.4.15/NEWS
/usr/share/doc/memcached-1.4.15/README.md
/usr/share/doc/memcached-1.4.15/protocol.txt
/usr/share/doc/memcached-1.4.15/readme.txt
/usr/share/doc/memcached-1.4.15/threads.txt
/usr/share/man/man1/memcached-tool.1.gz
/usr/share/man/man1/memcached.1.gz

[root@netkiller ~]# cat /etc/sysconfig/memcached
PORT="11211"
USER="memcached"
MAXCONN="1024"
CACHE_SIZE="64"
OPTIONS=""

[root@netkiller ~]# systemctl enable memcached
Created symlink from /etc/systemd/system/multi-
user.target.wants/memcached.service to
/usr/lib/systemd/system/memcached.service.

[root@netkiller ~]# systemctl start memcached

[root@netkiller ~]# systemctl status memcached
● memcached.service - Memcached
   Loaded: loaded (/usr/lib/systemd/system/memcached.service;
   enabled; vendor preset: disabled)
   Active: active (running) since Wed 2018-07-04 11:28:42 CST;
   5s ago
   Main PID: 4186 (memcached)
   CGroup: /system.slice/memcached.service
           └─4186 /usr/bin/memcached -u memcached -p 11211 -m
   64 -c 1024
```

```
Jul 04 11:28:42 netkiller systemd[1]: Started Memcached.  
Jul 04 11:28:42 netkiller systemd[1]: Starting Memcached...
```

2. Memcached 代理

2.1. moxi

couchbase 使用 moxi 为用户提供 memcached 负载均衡功能

2.2. memagent

第 51 章 RethinkDB

<http://www.rethinkdb.com/>

第 52 章 TokyoCabinet/Tyrant

<http://www.162cm.com/p/tokyotyrant.html>

```
# yum install tokyocabinet tokyocabinet-devel -y
```


第 53 章 Flare

第 54 章 Voldemort

第 55 章 LevelDB

LevelDB is a fast key-value storage library written at Google that provides an ordered mapping from string keys to string values.

<http://code.google.com/p/leveldb/>

第 56 章 HyperDex

<http://hyperdex.org/>

第 57 章 LeoFS

LeoFS is an Unstructured Object Storage for the Web and a highly available, distributed, eventually consistent storage system.

部分 VI. PostgreSQL

[PostgreSQL 实用实例参考](#)

第 58 章 PostgreSQL 安装

1. Ubuntu 12.04.1 LTS 安装

安装环境

PostgreSQL 9.1

\$ sudo apt-get install postgresql

```
$ sudo apt-get install postgresql
```

更改postgres管理员用户密码

```
$ sudo passwd postgres
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

然后切换到postgres用户环境

```
$ su - postgres
Password:
Added user postgres.
```

进入psql客户端, PostgreSQL的psql命令相当于sqlplus,mysql命令

```
$ psql
psql (9.1.6)
Type "help" for help.

postgres=#
```

退出\q

```
postgres=# \q
```


2. YUM 默认源安装

```
# yum search postgresql | grep ^postgresql
postgresql-odbc.x86_64 : PostgreSQL ODBC driver
postgresql.i686 : PostgreSQL client programs
postgresql.x86_64 : PostgreSQL client programs
postgresql-contrib.x86_64 : Contributed modules distributed
with PostgreSQL
postgresql-devel.i686 : PostgreSQL development header files and
libraries
postgresql-devel.x86_64 : PostgreSQL development header files
and libraries
postgresql-docs.x86_64 : Extra documentation for PostgreSQL
postgresql-ip4r.x86_64 : IPv4 and IPv4 range index types for
PostgreSQL
postgresql-jdbc.noarch : JDBC driver for PostgreSQL
postgresql-libs.i686 : The shared libraries required for any
PostgreSQL clients
postgresql-libs.x86_64 : The shared libraries required for any
PostgreSQL
postgresql-pgpool-II.i686 : Pgpool is a connection
pooling/replication server
postgresql-pgpool-II.x86_64 : Pgpool is a connection
pooling/replication server
postgresql-pgpool-II-recovery.x86_64 : PGPool recovery add-on
for PostgreSQL
postgresql-plparrot.x86_64 : A PostgreSQL procedural language
for the Parrot
postgresql-plperl.x86_64 : The Perl procedural language for
PostgreSQL
postgresql-plpython.x86_64 : The Python procedural language for
PostgreSQL
postgresql-plruby.x86_64 : PostgreSQL Ruby Procedural Language
postgresql-pltcl.x86_64 : The Tcl procedural language for
PostgreSQL
postgresql-server.x86_64 : The programs needed to create and
run a PostgreSQL
postgresql-test.x86_64 : The test suite distributed with
PostgreSQL
postgresql_autodoc.noarch : PostgreSQL AutoDoc Utility
postgresql-pgpool-II-devel.i686 : The development files for
pgpool-II
```

```
postgresql-pgpool-II-devel.x86_64 : The development files for
pgpool-II
postgresql-plruby-doc.x86_64 : Documentation for plruby
```

默认YUM源上面只有 8.4.13

```
# yum info postgresql-server
Loaded plugins: fastestmirror, presto, refresh-packagekit
Loading mirror speeds from cached hostfile
 * base: centos.ust.hk
 * epel: mirror01.idc.hinet.net
 * extras: centos.ust.hk
 * updates: centos.ust.hk
Installed Packages
Name           : postgresql-server
Arch           : x86_64
Version        : 8.4.13
Release        : 1.el6_3
Size           : 14 M
Repo           : installed
From repo      : base
Summary        : The programs needed to create and run a
PostgreSQL server
URL            : http://www.postgresql.org/
License        : PostgreSQL
Description    : The postgresql-server package includes the
programs needed to create
                  : and run a PostgreSQL server, which will in turn
allow you to create
                  : and maintain PostgreSQL databases. PostgreSQL is
an advanced
                  : Object-Relational database management system
(DBMS) that supports
                  : almost all SQL constructs (including
transactions, subselects and
                  : user-defined types and functions). You should
install
                  : postgresql-server if you want to create and
maintain your own
                  : PostgreSQL databases and/or your own PostgreSQL
server. You also need
                  : to install the postgresql package.
```

```
# yum install postgresql-server
```

```
# chkconfig postgresql on
```

```
# service postgresql initdb
```

```
# service postgresql start
```

3. PostgreSQL 官方 YUM 源安装

CentOS 6.4 环境 YUM 地址 <http://yum.postgresql.org/>

3.1.9.2

```
# yum install http://yum.postgresql.org/9.2/redhat/rhel-6-  
x86_64/pgdg-centos92-9.2-6.noarch.rpm  
# yum list postgres*  
  
# yum install postgresql92-server postgresql92  
  
chkconfig postgresql-9.2 on  
service postgresql-9.2 initdb  
service postgresql-9.2 start
```

配置文件的位置

```
# ls /var/lib/pgsql/9.2/data/*.conf  
/var/lib/pgsql/9.2/data/pg_hba.conf  
/var/lib/pgsql/9.2/data/pg_ident.conf  
/var/lib/pgsql/9.2/data/postgresql.conf
```

3.2.9.3

```
# yum install http://yum.postgresql.org/9.3/redhat/rhel-6-  
x86_64/pgdg-centos93-9.3-1.noarch.rpm  
# yum install postgresql93-server postgresql93
```

```
# chkconfig postgresql-9.3 on  
# service postgresql-9.3 initdb  
# service postgresql-9.3 start
```

9.2 升级到 9.3

备份数据库

```
# su - postgres
$ pg_dump -f wechat.sql wechat
```

升级数据库

关闭9.2，启动9.3

```
# chkconfig postgresql-9.2 off
# service postgresql-9.2 stop

# service postgresql-9.3 start
```

回复数据库

```
# su - postgres
$ createuser -r -s -P dba
$ createuser -P wechat
$ createdb -E UTF8 -O wechat wechat
$ cat wechat.sql | psql wechat
```

合并配置文件

```
$ cp 9.3/data/postgresql.conf{,.original}
$ cp 9.3/data/pg_hba.conf{,.original}
$ vimdiff 9.2/data/postgresql.conf 9.3/data/postgresql.conf
$ vimdiff 9.2/data/pg_hba.conf 9.3/data/pg_hba.conf
```

注意 vimdiff 命令是文件对比于合并工具，建议你选择你比较熟悉的工具。

```
# service postgresql-9.3 restart
```

3.3.9.4

CentOS 6

<https://github.com/oscm/shell/blob/master/database/postgresql/postgresql93-centos6.sh>

CentOS 7

<https://github.com/oscm/shell/blob/master/database/postgresql/postgresql93-centos7.sh>

```
#!/bin/bash

# CentOS 7
yum install -y
https://download.postgresql.org/pub/repos/yum/9.6/redhat/rhel-7-x86_64/pgdg-centos96-9.6-3.noarch.rpm
yum install -y postgresql96-server postgresql96-contrib

systemctl initdb postgresql-9.6
chkconfig postgresql-9.6 on

cp /var/lib/pgsql/9.6/data/postgresql.conf{,.original}
cp /var/lib/pgsql/9.6/data/pg_hba.conf{,.original}
cp /var/lib/pgsql/9.6/data/pg_ident.conf{,.original}

sed -i "s/#listen_addresses = 'localhost'/listen_addresses = '*'/" /var/lib/pgsql/9.6/data/postgresql.conf

systemctl start postgresql-9.6

iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 5432 -j ACCEPT
systemctl save iptables
```

一键安装

```
curl -s  
https://raw.githubusercontent.com/oscm/shell/master/database/postgresql/postgresql93-centos7.sh | bash
```

3.4.9.6

<https://github.com/oscm/shell/blob/master/database/postgresql/postgresql96-centos7.sh>

一键安装

```
curl -s  
https://raw.githubusercontent.com/oscm/shell/master/database/postgresql/postgresql96-centos7.sh | bash
```

3.5. PostgreSQL 10

一键安装 PostgreSQL 10

```
curl -s  
https://raw.githubusercontent.com/oscm/shell/master/database/postgresql/10/repository.sh | bash  
curl -s  
https://raw.githubusercontent.com/oscm/shell/master/database/postgresql/10/postgresql-server.sh | bash
```

监听所有适配器地址

```
curl -s
https://raw.githubusercontent.com/oscm/shell/master/database/postgresql/10/listen_addresses.all.sh | bash
```

创建数据库

```
CREATE ROLE test LOGIN PASSWORD 'test' NOSUPERUSER NOINHERIT
NOCREATEDB NOCREATEROLE;
CREATE DATABASE test WITH OWNER = test ENCODING = 'UTF8'
TABLESPACE = pg_default;
```

配置访问控制列表 /var/lib/pgsql/10/data/pg_hba.conf 加入

```
# vim /var/lib/pgsql/10/data/pg_hba.conf
host * dba 0.0.0.0/0 md5
host test test 0.0.0.0/0 md5
```

确认防火墙已经放行5432端口

```
# cat /etc/sysconfig/iptables | grep 5432 -A INPUT -s 172.16.0.0/24 -p
tcp -m state --state NEW -m tcp --dport 5432 -j ACCEPT
```

例 58.1. Example for /var/lib/pgsql/10/data/pg_hba.conf

```
# cat /var/lib/pgsql/10/data/pg_hba.conf
# PostgreSQL Client Authentication Configuration File
# =====
#
# Refer to the "Client Authentication" section in the
PostgreSQL
# documentation for a complete description of this file. A
short
```



```
# synopsis follows.
#
# This file controls: which hosts are allowed to connect, how
clients
# are authenticated, which PostgreSQL user names they can use,
which
# databases they can access. Records take one of these forms:
#
# local      DATABASE  USER  METHOD  [OPTIONS]
# host      DATABASE  USER  ADDRESS METHOD  [OPTIONS]
# hostssl   DATABASE  USER  ADDRESS METHOD  [OPTIONS]
# hostnossl DATABASE  USER  ADDRESS METHOD  [OPTIONS]
#
# (The uppercase items must be replaced by actual values.)
#
# The first field is the connection type: "local" is a Unix-
domain
# socket, "host" is either a plain or SSL-encrypted TCP/IP
socket,
# "hostssl" is an SSL-encrypted TCP/IP socket, and "hostnossl"
is a
# plain TCP/IP socket.
#
# DATABASE can be "all", "sameuser", "samerole", "replication",
a
# database name, or a comma-separated list thereof. The "all"
# keyword does not match "replication". Access to replication
# must be enabled in a separate record (see example below).
#
# USER can be "all", a user name, a group name prefixed with
"+", or a
# comma-separated list thereof. In both the DATABASE and USER
fields
# you can also write a file name prefixed with "@" to include
names
# from a separate file.
#
# ADDRESS specifies the set of hosts the record matches. It
can be a
# host name, or it is made up of an IP address and a CIDR mask
that is
# an integer (between 0 and 32 (IPv4) or 128 (IPv6) inclusive)
that
# specifies the number of significant bits in the mask. A host
name
```

```
# that starts with a dot (.) matches a suffix of the actual
host name.
# Alternatively, you can write an IP address and netmask in
separate
# columns to specify the set of hosts.  Instead of a CIDR-
address, you
# can write "samehost" to match any of the server's own IP
addresses,
# or "samenet" to match any address in any subnet that the
server is
# directly connected to.
#
# METHOD can be "trust", "reject", "md5", "password", "scram-
sha-256",
# "gss", "sspi", "ident", "peer", "pam", "ldap", "radius" or
"cert".
# Note that "password" sends passwords in clear text; "md5" or
# "scram-sha-256" are preferred since they send encrypted
passwords.
#
# OPTIONS are a set of options for the authentication in the
format
# NAME=VALUE.  The available options depend on the different
# authentication methods -- refer to the "Client
Authentication"
# section in the documentation for a list of which options are
# available for which authentication methods.
#
# Database and user names containing spaces, commas, quotes and
other
# special characters must be quoted.  Quoting one of the
keywords
# "all", "sameuser", "samerole" or "replication" makes the name
lose
# its special character, and just match a database or username
with
# that name.
#
# This file is read on server startup and when the server
receives a
# SIGHUP signal.  If you edit the file on a running system, you
have to
# SIGHUP the server for the changes to take effect, run "pg_ctl
reload",
# or execute "SELECT pg_reload_conf()".
```

```

#
# Put your actual configuration here
# -----
#
# If you want to allow non-local connections, you need to add
more
# "host" records.  In that case you will also need to make
PostgreSQL
# listen on a non-local interface via the listen_addresses
# configuration parameter, or via the -i or -h command line
switches.

# TYPE  DATABASE          USER            ADDRESS
METHOD
host    *                  dba             0.0.0.0/0      md5
host    test              test           0.0.0.0/0      md5

# "local" is for Unix domain socket connections only
local  all              all
peer
# IPv4 local connections:
host   all              all             127.0.0.1/32
ident
# IPv6 local connections:
host   all              all             ::1/128
ident
# Allow replication connections from localhost, by a user with
the
# replication privilege.
local  replication     all
peer
host   replication     all             127.0.0.1/32
ident
host   replication     all             ::1/128
ident

```

4. PostgreSQL 配置

su 到 postgres 用户

```
$ su - postgres
Password:
$ pwd
/var/lib/postgresql
$
```

备份配置文件，防止修改过程中损毁

```
cp /etc/postgresql/9.1/main/postgresql.conf
/etc/postgresql/9.1/main/postgresql.conf.original
cp /etc/postgresql/9.1/main/pg_hba.conf
/etc/postgresql/9.1/main/pg_hba.conf.original
```

4.1. postgresql.conf

启用tcp/ip连接，去掉下面注释,修改为你需要的IP地址，默认为localhost

```
listen_addresses = 'localhost'
```

如果有多个网络适配器可以指定 'ip' 或 '*' 任何接口上的IP地址都可能listen.

```
$ sudo vim /etc/postgresql/9.1/main/postgresql.conf

listen_addresses = '*'
```

4.2. pg_hba.conf

pg_hba.conf配置文件的权限需要注意以下， -rw-r----- 1 postgres postgres 4649 Dec 5 18:00 pg_hba.conf

```
$ ll /etc/postgresql/9.1/main/
total 52
drwxr-xr-x 2 postgres postgres 4096 Dec 6 09:40 ./
drwxr-xr-x 3 postgres postgres 4096 Dec 5 18:00 ../
-rw-r--r-- 1 postgres postgres 316 Dec 5 18:00 environment
-rw-r--r-- 1 postgres postgres 143 Dec 5 18:00 pg_ctl.conf
-rw-r----- 1 postgres postgres 4649 Dec 5 18:00 pg_hba.conf
-rw-r----- 1 postgres postgres 1636 Dec 5 18:00 pg_ident.conf
-rw-r--r-- 1 postgres postgres 19259 Dec 5 18:00
postgresql.conf
-rw-r--r-- 1 postgres postgres 378 Dec 5 18:00 start.conf
```

pg_hba.conf配置文件负责访问权限控制

```
# TYPE DATABASE USER ADDRESS
METHOD

# "local" is for Unix domain socket connections only
local all all
peer
# IPv4 local connections:
host all all 127.0.0.1/32
md5
# IPv6 local connections:
host all all ::1/128
md5
```

TYPE

local 本地使用unix/socket 方式连接, host 使用tcp/ip socket 方式连接

DATABASE

数据库名.

USER

用户名.

ADDRESS

允许连接的IP地址，可以使用子网掩码.

METHOD

认证加密方式.

下面我们做一个简单测试，首先配置pg_hba.conf文件

```
$ sudo vi /etc/postgresql/9.1/main/pg_hba.conf
host * dba 0.0.0.0/0 md5
host test test 0.0.0.0/0 md5
```

运行创建数据,用户 的SQL语句

```
CREATE ROLE test LOGIN PASSWORD 'test' NOSUPERUSER NOINHERIT
NOCREATEDB NOCREATEROLE;

CREATE DATABASE test WITH OWNER = test ENCODING = 'UTF8'
TABLESPACE = pg_default;
```

进入psql

```
$ psql
psql (9.1.6)
Type "help" for help.

postgres=# CREATE ROLE test LOGIN PASSWORD 'test' NOSUPERUSER
NOINHERIT NOCREATEDB NOCREATEROLE;
```

```

CREATE ROLE
postgres=# CREATE DATABASE test WITH OWNER = test ENCODING =
'UTF8' TABLESPACE = pg_default;
CREATE DATABASE
postgres=# \q

```

使用psql登录

```

$ psql -hlocalhost -Utest test
Password for user test:
psql (9.1.6)
SSL connection (cipher: DHE-RSA-AES256-SHA, bits: 256)
Type "help" for help.

test=> \l

                                List of databases
  Name          | Owner          | Encoding | Collate         | Ctype          |
-----+-----+-----+-----+-----+-----+
postgres       | postgres      | UTF8     | en_US.UTF-8    | en_US.UTF-8    |
template0      | postgres      | UTF8     | en_US.UTF-8    | en_US.UTF-8    |
=c/postgres    +
|              |              |          |                |                |
postgres=CTc/postgres
templatel1     | postgres      | UTF8     | en_US.UTF-8    | en_US.UTF-8    |
=c/postgres    +
|              |              |          |                |                |
postgres=CTc/postgres
test           | test          | UTF8     | en_US.UTF-8    | en_US.UTF-8    |
(4 rows)

test=>

```

5. 创建dba用户

创建一个远程维护数据库dba用户，具有创建数据库与创建用户的权限

```
CREATE USER dba PASSWORD 'dba' CREATEDB CREATEUSER;
```

进入psql

```
$ psql
psql (9.1.6)
Type "help" for help.

postgres=# CREATE USER dba PASSWORD 'dba' CREATEDB CREATEUSER;
CREATE ROLE
postgres=# \q
```

使用psql登录

```
$ psql -hlocalhost -Udba postgres
Password for user dba:
psql (9.1.6)
SSL connection (cipher: DHE-RSA-AES256-SHA, bits: 256)
Type "help" for help.

postgres=#
```


第 59 章 数据库管理 (Database Administration)

1. 用户管理 (User Account Management)

1.1. 创建用户

```
CREATE USER user [IDENTIFIED BY [PASSWORD] 'password']  
    [, user [IDENTIFIED BY [PASSWORD] 'password']] ...
```

```
CREATE USER 'test'@'xxx.xxx.xxx.xxx' IDENTIFIED BY 'your_password';
```

```
CREATE USER 'root'@'192.168.1.%' IDENTIFIED BY 'password';
```

add a new user by grant

```
GRANT ALL PRIVILEGES ON opencart.* TO 'neo'@'localhost' IDENTIFIED BY  
'chen' WITH GRANT OPTION;  
  
GRANT ALL PRIVILEGES ON *.* TO 'neo'@'localhost' IDENTIFIED BY 'chen'  
WITH GRANT OPTION;  
  
FLUSH PRIVILEGES;
```

MySQL 8.0

```
mysql> CREATE USER 'root'@'%' IDENTIFIED WITH mysql_native_password BY  
'pMQiEgelikst7S_6tlXzB0mt_4b';  
Query OK, 0 rows affected (0.06 sec)  
  
mysql> grant all on *.* to 'root'@'%';
```

```
Query OK, 0 rows affected (0.11 sec)
```

1.2. 删除用户

```
DROP USER user [, user] ...
```

```
mysql> drop user 'root'@'%';  
Query OK, 0 rows affected (0.02 sec)  
  
mysql> drop user admin@'localhost';  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> drop user admin@'127.0.0.1';  
Query OK, 0 rows affected (0.00 sec)
```

1.3. 修改用户名

```
RENAME USER old_user TO new_user [, old_user TO new_user] ...
```

1.4. 修改密码

mysql 5.7 之前的版本

```
SET PASSWORD FOR 'bob'@'%.loc.gov' = PASSWORD('newpass');  
  
SET PASSWORD FOR 'root'@'%' =  
PASSWORD('co2uqAMAholaSOS62146Xoci6ogu4I');
```

MySQL 5.7 之后

```
ALTER USER 'root'@'localhost' IDENTIFIED BY 'your_password';
```

```
mysql> ALTER user 'root'@'%' IDENTIFIED BY 'test';  
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> FLUSH PRIVILEGES;  
Query OK, 0 rows affected (0.00 sec)
```

2. Access Privilege System

```
global privileges
OR (database privileges AND host privileges)
OR table privileges
OR column privileges
OR routine privileges
```

Table 12.1. Permissible Privileges for GRANT and REVOKE

Privilege	Meaning
ALL [PRIVILEGES]	Grant all privileges at specified access level except GRANT OPTION
ALTER	Enable use of ALTER TABLE
ALTER ROUTINE	Enable stored routines to be altered or dropped
CREATE	Enable database and table creation
CREATE ROUTINE	Enable stored routine creation
CREATE TABLESPACE	Enable tablespaces and log file groups to be created, altered, or dropped
CREATE TEMPORARY TABLES	Enable use of CREATE TEMPORARY TABLE
CREATE USER	Enable use of CREATE USER, DROP USER, RENAME USER, and REVOKE ALL PRIVILEGES
CREATE VIEW	Enable views to be created or altered
DELETE	Enable use of DELETE
DROP	Enable databases, tables, and views to be dropped
EVENT	Enable use of events for the Event Scheduler
EXECUTE	Enable the user to execute stored routines
FILE	Enable the user to cause the server to read or write files
GRANT OPTION	Enable privileges to be granted to or removed from other accounts
INDEX	Enable indexes to be created or dropped
INSERT	Enable use of INSERT
LOCK TABLES	Enable use of LOCK TABLES on tables for which you have the SELECT privilege
PROCESS	Enable the user to see all processes with SHOW PROCESSLIST
PROXY	Enable user proxying
REFERENCES	Not implemented
RELOAD	Enable use of FLUSH operations
REPLICATION CLIENT	Enable the user to ask where master or slave servers are
REPLICATION SLAVE	Enable replication slaves to read binary log events from the master
SELECT	Enable use of SELECT
SHOW DATABASES	Enable SHOW DATABASES to show all databases
SHOW VIEW	Enable use of SHOW CREATE VIEW
SHUTDOWN	Enable use of mysqladmin shutdown
SUPER	Enable use of other administrative operations such as CHANGE MASTER TO, KILL, PURGE BINARY LOGS, SET GLOBAL, and mysqladmin debug command
TRIGGER	Enable trigger operations
UPDATE	Enable use of UPDATE

```
USAGE    Synonym for "no privileges"
```

<http://dev.mysql.com/doc/refman/5.5/en/grant.html#grant-table-privileges>

REPLICATION CLIENT 与 REPLICATION SLAVE区别，前者只能使用SHOW MASTER STATUS和SHOW SLAVE STATUS命令监控复制状态，后者才能从主库复制binlog.

2.1. SHOW GRANTS

```
mysql> SHOW GRANTS FOR 'root'@'localhost';
+-----+
| Grants for root@localhost |
+-----+
| GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' WITH GRANT OPTION |
+-----+
1 row in set (0.00 sec)
```

```
mysql> show grants;
+-----+
--+
| Grants for root@localhost |
+-----+
--+
| GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' WITH GRANT OPTION |
+-----+
--+
1 row in set (0.00 sec)
```

2.2. show privileges

```
mysql> show privileges;
+-----+-----+-----+
| Privilege | Context | Comment |
+-----+-----+-----+
| Alter | Tables | To alter the table |
| Alter routine | Functions,Procedures | To alter or drop stored functions/procedures |
| Create | Databases,Tables,Indexes | To create
```

new databases and tables			
Create routine	Databases		To use
CREATE FUNCTION/PROCEDURE			
Create temporary tables	Databases		To use
CREATE TEMPORARY TABLE			
Create view	Tables		To create
new views			
Create user	Server Admin		To create
new users			
Delete	Tables		To delete
existing rows			
Drop	Databases, Tables		To drop
databases, tables, and views			
Event	Server Admin		To create,
alter, drop and execute events			
Execute	Functions, Procedures		To execute
stored routines			
File	File access on server		To read and
write files on the server			
Grant option	Databases, Tables, Functions, Procedures		To give to
other users those privileges you possess			
Index	Tables		To create or
drop indexes			
Insert	Tables		To insert
data into tables			
Lock tables	Databases		To use LOCK
TABLES (together with SELECT privilege)			
Process	Server Admin		To view the
plain text of currently executing queries			
Proxy	Server Admin		To make
proxy user possible			
References	Databases, Tables		To have
references on tables			
Reload	Server Admin		To reload or
refresh tables, logs and privileges			
Replication client	Server Admin		To ask where
the slave or master servers are			
Replication slave	Server Admin		To read
binary log events from the master			
Select	Tables		To retrieve
rows from table			
Show databases	Server Admin		To see all
databases with SHOW DATABASES			
Show view	Tables		To see views
with SHOW CREATE VIEW			
Shutdown	Server Admin		To shut down
the server			
Super	Server Admin		To use KILL
thread, SET GLOBAL, CHANGE MASTER, etc.			
Trigger	Tables		To use
triggers			
Create tablespace	Server Admin		To
create/alter/drop tablespaces			
Update	Tables		To update
existing rows			
Usage	Server Admin		No
privileges - allow connect only			

```
+-----+-----+-----+
+-----+
31 rows in set (0.00 sec)
```

2.3. Grant privileges

Global privileges

```
GRANT ALL ON *.* TO 'someuser'@'somehost';
GRANT SELECT, INSERT ON *.* TO 'someuser'@'somehost';
```

Database privileges

```
GRANT ALL ON mydb.* TO 'someuser'@'somehost';
GRANT SELECT, INSERT ON mydb.* TO 'someuser'@'somehost';
```

Table privileges

```
GRANT ALL ON mydb.mytbl TO 'someuser'@'somehost';
GRANT SELECT, INSERT ON mydb.mytbl TO 'someuser'@'somehost';
```

Column privileges

```
GRANT SELECT (col1), INSERT (col1,col2) ON mydb.mytbl TO 'someuser'@'somehost';
```

Routine privileges

```
GRANT CREATE ROUTINE ON mydb.* TO 'someuser'@'somehost';
GRANT EXECUTE ON PROCEDURE mydb.myproc TO 'someuser'@'somehost';
```

2.4. Revoke privileges

```
REVOKE
  priv_type [(column_list)]
```

```
[, priv_type [(column_list)]] ...  
ON [object_type] priv_level  
FROM user [, user] ...
```

```
REVOKE ALL PRIVILEGES, GRANT OPTION  
FROM user [, user] ...
```

2.5. Show Privileges

```
mysql> select * from user where user = 'neo'\G  
***** 1. row *****  
      Host: 192.168.0.5  
      User: neo  
      Password: *7564B7B0A062C9523700601CBA1DCE1F861D6270  
      Select_priv: Y  
      Insert_priv: Y  
      Update_priv: Y  
      Delete_priv: Y  
      Create_priv: Y  
      Drop_priv: Y  
      Reload_priv: Y  
      Shutdown_priv: Y  
      Process_priv: Y  
      File_priv: Y  
      Grant_priv: N  
      References_priv: Y  
      Index_priv: Y  
      Alter_priv: Y  
      Show_db_priv: Y  
      Super_priv: Y  
      Create_tmp_table_priv: Y  
      Lock_tables_priv: Y  
      Execute_priv: Y  
      Repl_slave_priv: Y  
      Repl_client_priv: Y  
      Create_view_priv: Y  
      Show_view_priv: Y  
      Create_routine_priv: Y  
      Alter_routine_priv: Y  
      Create_user_priv: Y  
      Event_priv: Y  
      Trigger_priv: Y  
      ssl_type:  
      ssl_cipher:  
      x509_issuer:  
      x509_subject:  
      max_questions: 0  
      max_updates: 0  
      max_connections: 0  
      max_user_connections: 0  
1 row in set (0.00 sec)  
  
mysql>
```


2.6. MAX_QUERIES_PER_HOUR/MAX_UPDATES_PER_HOUR

```
GRANT USAGE ON *.* TO ...  
  WITH MAX_QUERIES_PER_HOUR 500 MAX_UPDATES_PER_HOUR 100;
```

2.7. Table Privileges

授权tmp用户只能访问tablename表

```
GRANT ALL PRIVILEGES ON tmp.tabname TO 'tmp'@'%' IDENTIFIED BY 'chen' WITH GRANT  
OPTION;
```

如果用户已经存在仅仅是分配权限可以使用下面方法

```
GRANT ALL ON mydb.mytbl TO 'someuser'@'somehost';  
GRANT SELECT, INSERT ON mydb.mytbl TO 'someuser'@'somehost';
```

2.8. Column Privileges

mydb.mytbl 表 col1字段允许查询，col1,col2允许插入

```
GRANT SELECT (col1), INSERT (col1,col2) ON mydb.mytbl TO 'someuser'@'somehost';
```

3. 字符集转换

找出指定字符集的表

```
select TABLE_SCHEMA, TABLE_NAME, TABLE_COLLATION from
information_schema.tables where table_collation =
'utf8mb4_0900_ai_ci' and table_schema = 'your_schema';
```

```
SELECT
    CONCAT(
        'ALTER TABLE ',
        TABLE_NAME,
        ' CONVERT TO CHARACTER SET utf8mb4 COLLATE
utf8mb4_general_ci;'
    )
FROM
    information_schema.`TABLES`
WHERE
    TABLE_SCHEMA = 'DATABASE_NAME';
```

3.1. 转换 latin1 到 UTF-8

```
UPDATE category SET
name=convert(cast(convert(name using latin1) as binary) using
utf8),
description=convert(cast(convert(description using latin1) as
binary) using utf8)
```

4. 重新整理AUTO_INCREMENT字段

AUTO_INCREMENT 并非按照我们意愿，顺序排列，经常会跳过一些数字，例如当插入失败的时候，再次插入会使用新的值。有时会造成浪费，我们可以使用下面SQL重新编排AUTO_INCREMENT序列。

```
SET @newid=0;
UPDATE mytable SET id = (SELECT @newid:=@newid+ 1);
```

使用max()查看最大值，然后使用 alter修改起始位置。

```
select max(id) from mytable;
ALTER TABLE mytable AUTO_INCREMENT = 1000;
```

注意外键，需要 ON UPDATE CASCADE 支持，否则无法更新。
CONSTRAINT `FK_group_has_contact_contact` FOREIGN KEY
(`contact_id`) REFERENCES `contact` (`id`) ON UPDATE CASCADE
ON DELETE CASCADE,

```
CREATE TABLE `contact` (  
  `id` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT COMMENT  
'唯一ID',  
  `name` VARCHAR(50) NOT NULL COMMENT '姓名',  
  `mobile` VARBINARY(32) NULL DEFAULT NULL COMMENT '手机号  
码',  
  `email` VARBINARY(50) NULL DEFAULT NULL COMMENT '电子邮  
件',  
  `mobile_digest` VARCHAR(32) NULL DEFAULT NULL COMMENT  
'摘要',
```

```

        `email_digest` VARCHAR(32) NULL DEFAULT NULL COMMENT
'邮件摘要',
        `birthday` DATE NULL DEFAULT NULL COMMENT '生日',
        `description` VARCHAR(255) NULL DEFAULT NULL COMMENT
'备注描述',
        `status`
        ENUM('Subscription','Unsubscribe') NOT NULL DEFAULT
'Subscription' COMMENT '订阅状态',
        `ctime` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
COMMENT '创建时间',
        `mtime` TIMESTAMP NULL DEFAULT NULL ON UPDATE
CURRENT_TIMESTAMP COMMENT '修改时间',
        PRIMARY KEY (`id`),
        UNIQUE INDEX `digest` (`mobile_digest`, `email_digest`)
)
COMMENT='会员手机短信与电子邮件映射表'
COLLATE='utf8_general_ci'
ENGINE=InnoDB
AUTO_INCREMENT=43642;

CREATE TABLE `group` (
    `id` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,
    `name` VARCHAR(50) NOT NULL,
    `description` VARCHAR(512)
    NOT NULL,
    `ctime` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (`id`),
    UNIQUE INDEX `name` (`name`)
)
COMMENT='短信分组'
COLLATE='utf8_general_ci'
ENGINE=InnoDB
AUTO_INCREMENT=8;

CREATE TABLE `group_has_contact` (
    `id` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,
    `group_id` INT(10) UNSIGNED NOT NULL,
    `contact_id` INT(10) UNSIGNED NOT NULL,
    `ctime` TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (`id`),
    UNIQUE INDEX `group_contact` (`group_id`,
`contact_id`),
    INDEX `FK_group_has_contact_contact`
    (`contact_id`),
    CONSTRAINT `FK_group_has_contact_contact` FOREIGN KEY

```

```
(`contact_id`) REFERENCES `contact` (`id`) ON UPDATE CASCADE ON  
DELETE CASCADE,  
    CONSTRAINT `FK_group_has_contact_group` FOREIGN KEY  
(`group_id`) REFERENCES `group` (`id`) ON UPDATE CASCADE ON  
DELETE CASCADE  
)  
COMMENT='N:M'  
COLLATE='utf8_general_ci'  
ENGINE=InnoDB  
AUTO_INCREMENT=55764;
```

5. 数据库内容替换

```
#!/bin/bash
HOST='localhost'
USER='neo'
PASS='chen'

SDB='neo'
TDB='netkiller'
MYSQLDUMP="mysqldump"
MYSQLDUMPOPTS="-h${HOST} -u${USER} -p${PASS}"

MYSQL="mysql"
MYSQLOPTS="-h${HOST} -u${USER} -p${PASS}"
#SED="sed -e 's/netkiller\.8800\.org/netkiller\.sf\.net/g' -e
's/陈景峰/景峰/g' -e 's/Neo/Netkiller/g'"

$MYSQL $MYSQLOPTS <<SQL
DROP DATABASE $TDB;
CREATE DATABASE $TDB DEFAULT CHARACTER SET utf8 COLLATE
utf8_general_ci;
SQL

$MYSQLDUMP $MYSQLDUMPOPTS ${SDB} | sed -e
's/netkiller\.8800\.org/netkiller\.sf\.net/g' -e 's/陈景峰/景
峰/g' -e 's/Neo/Netkiller/g' | $MYSQL $MYSQLOPTS ${TDB}
#echo "$MYSQLDUMP $MYSQLDUMPOPTS ${SDB} | $SED | $MYSQL
$MYSQLOPTS ${TDB}"
```

6. Kill 脚本

查询出锁定的表

```
SELECT concat('KILL ',id,';') FROM information_schema.processlist
WHERE user='root';
```

```
SELECT concat('KILL ',id,';') FROM information_schema.processlist
WHERE command='Locked' and user='root';
```

```
SELECT concat('KILL ',id,';') FROM information_schema.processlist
WHERE command='Locked' and user='root' and db='test';
```

拼装kill命令后输入到kill.sql, source 将从kill.sql读取sql命令并执行。

```
SELECT concat('KILL ',id,';') FROM
information_schema.processlist WHERE user='root' INTO OUTFILE
'/tmp/kill.sql';
source /tmp/kill.sql;
```

```
mysqladmin -uroot -p processlist | grep Sleep |awk '{if
(length($2) > 1) print "Kill "$2}'|xargs mysqladmin -uroot kill
```

7. SHOW COMMAND

7.1. 查看版本

Server

```
mysql> select version();
+-----+
| version() |
+-----+
| 5.0.77 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> status;
-----
mysql Ver 14.12 Distrib 5.0.77, for redhat-linux-gnu (x86_64)
using readline 5.1

Connection id: 1533
Current database:
Current user: root@localhost
SSL: Not in use
Current pager: stdout
Using outfile: ''
Using delimiter: ;
Server version: 5.0.77 Source distribution
Protocol version: 10
Connection: Localhost via UNIX socket
Server characterset: latin1
Db characterset: latin1
Client characterset: latin1
Conn. characterset: latin1
UNIX socket: /var/lib/mysql/mysql.sock
Uptime: 1 day 21 hours 40 min 52 sec

Threads: 1 Questions: 22172 Slow queries: 0 Opens: 3130 Flush
tables: 1 Open tables: 64 Queries per second avg: 0.135
-----
```


Client

```
[root@development ~]# mysql -V
mysql Ver 14.12 Distrib 5.0.77, for redhat-linux-gnu (x86_64)
using readline 5.1
```

7.2. status

```
mysql> show status;
mysql> show global status;
```

show status

数据库性能状态

(1) QPS (每秒Query量)

$QPS = \text{Questions(or Queries)} / \text{seconds}$

```
mysql > show /*50000 global */ status like 'Question';
```

(2) TPS (每秒事务量)

$TPS = (\text{Com_commit} + \text{Com_rollback}) / \text{seconds}$

```
mysql > show status like 'Com_commit';
```

```
mysql > show status like 'Com_rollback';
```

(3) key Buffer 命中率

$\text{key_buffer_read_hits} = (1 - \text{key_reads} / \text{key_read_requests}) * 100\%$

$\text{key_buffer_write_hits} = (1 - \text{key_writes} / \text{key_write_requests}) * 100\%$

```
mysql> show status like 'Key%';
```

(4) InnoDB Buffer命中率

$\text{innodb_buffer_read_hits} = (1 - \text{innodb_buffer_pool_reads} /$

```

innodb_buffer_pool_read_requests) * 100%

mysql> show status like 'innodb_buffer_pool_read%';

(5)Query Cache命中率
Query_cache_hits = (Qcahce_hits / (Qcache_hits + Qcache_inserts
)) * 100%;

mysql> show status like 'Qcache%';
(6)Table Cache状态量
mysql> show status like 'open%';

(7)Thread Cache 命中率
Thread_cache_hits = (1 - Threads_created / connections ) * 100%

mysql> show status like 'Thread%';

mysql> show status like 'Connections';

(8)锁定状态
mysql> show status like '%lock%';

(9)复制延时量
mysql > show slave status

(10) Tmp Table 状况(临时表状况)
mysql > show status like 'Create_tmp%';
(11) Binlog Cache 使用状况
mysql > show status like 'Binlog_cache%';

(12) InnoDB_log_waits 量
mysql > show status like 'innodb_log_waits';

```

show master status

```

mysql> show master status;
+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB |
Binlog_Ignore_DB |
+-----+-----+-----+-----+

```

```
-----+
| DBMaster-bin.000018 | 409468882 | example      |
|
+-----+-----+-----+-----+
-----+
1 row in set (0.00 sec)

mysql>
```

show slave status

```
mysql> show slave status/G

得到的列表会有类似下面的数据：

File: mysql-bin.000001
Position: 1374
Binlog_Do_DB: test
Binlog_Ignore_DB: mysql

Slave_IO_State: Waiting for master to send event
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
```

show plugins

```
mysql> SHOW PLUGINS;
+-----+-----+-----+-----+-----+
| Name          | Status  | Type          | Library | License |
+-----+-----+-----+-----+-----+
| binlog        | ACTIVE  | STORAGE ENGINE | NULL    | GPL     |
| partition     | ACTIVE  | STORAGE ENGINE | NULL    | GPL     |
| ARCHIVE       | ACTIVE  | STORAGE ENGINE | NULL    | GPL     |
| BLACKHOLE     | ACTIVE  | STORAGE ENGINE | NULL    | GPL     |
| CSV           | ACTIVE  | STORAGE ENGINE | NULL    | GPL     |
```

FEDERATED	DISABLED	STORAGE ENGINE	NULL	GPL
MEMORY	ACTIVE	STORAGE ENGINE	NULL	GPL
InnoDB	ACTIVE	STORAGE ENGINE	NULL	GPL
MyISAM	ACTIVE	STORAGE ENGINE	NULL	GPL
MRG_MYISAM	ACTIVE	STORAGE ENGINE	NULL	GPL

10 rows in set (0.00 sec)

7.3. show processlist

```
show full processlist;
```

</screen>

<screen><![CDATA[

命令: show processlist;

如果是root帐号, 你能看到所有用户的当前连接。如果是其它普通帐号, 只能看到自己占用的连接。

show processlist;只列出前100条, 如果想全列出请使用show full processlist;

```
mysql> show processlist;
```

命令: show status;

Aborted_clients 由于客户没有正确关闭连接已经死掉, 已经放弃的连接数量。

Aborted_connects 尝试已经失败的MySQL服务器的连接的次数。

Connections 试图连接MySQL服务器的次数。

Created_tmp_tables 当执行语句时, 已经被创造了的隐含临时表的数量。

Delayed_insert_threads 正在使用的延迟插入处理器线程的数量。

Delayed_writes 用INSERT DELAYED写入的行数。

Delayed_errors 用INSERT DELAYED写入的发生某些错误(可能重复键值)的行数。

Flush_commands 执行FLUSH命令的次数。

Handler_delete 请求从一张表中删除行的次数。

Handler_read_first 请求读入表中第一行的次数。

Handler_read_key 请求数字基于键读行。

Handler_read_next 请求读入基于一个键的一行的次数。

Handler_read_rnd 请求读入基于一个固定位置的一行的次数。

Handler_update 请求更新表中一行的次数。

Handler_write 请求向表中插入一行的次数。

Key_blocks_used 用于关键字缓存的块的数量。

Key_read_requests 请求从缓存读入一个键值的次数。

Key_reads 从磁盘物理读入一个键值的次数。

Key_write_requests 请求将一个关键字块写入缓存次数。
Key_writes 将一个键值块物理写入磁盘的次数。
Max_used_connections 同时使用的连接的最大数目。
Not_flushed_key_blocks 在键缓存中已经改变但是还没被清空到磁盘上的键块。
Not_flushed_delayed_rows 在INSERT DELAY队列中等待写入的行的数量。
Open_tables 打开表的数量。
Open_files 打开文件的数量。
Open_streams 打开流的数量(主要用于日志记载)
Opened_tables 已经打开的表的数量。
Questions 发往服务器的查询的数量。
Slow_queries 要花超过long_query_time时间的查询数量。
Threads_connected 当前打开的连接的数量。
Threads_running 不在睡眠的线程数量。
Uptime 服务器工作了多少秒。

```
</screen>
</section>

<section id="binary">
  <title>binary 日志</title>
  <screen>
    <![CDATA[
```

```
mysql> show binary logs;
```

Log_name	File_size
mysql-bin.000001	19544
mysql-bin.000002	974751
mysql-bin.000003	107
mysql-bin.000004	3976040
mysql-bin.000005	126
mysql-bin.000006	350063
mysql-bin.000007	6826
mysql-bin.000008	3879494
mysql-bin.000009	126
mysql-bin.000010	494
mysql-bin.000011	17286686
mysql-bin.000012	15003942
mysql-bin.000013	1709321

```
13 rows in set (0.00 sec)
```

7.4. 线程的使用情况

```
mysql> SHOW STATUS LIKE 'threads%';
+-----+-----+
| Variable_name      | Value |
+-----+-----+
| Threads_cached     | 0     |
| Threads_connected  | 1     |
| Threads_created    | 1     |
| Threads_running    | 1     |
+-----+-----+
4 rows in set (0.01 sec)
```

7.5. DATABASES

```
SHOW DATABASES;
    </screen>
  </section>
  <section id="table">
    <title>TABLE</title>
    <screen><![CDATA[
SHOW TABLE STATUS FROM `dbname`;
    </screen>
  </section>
  <section id="created_tmp">
    <title>临时表</title>
    <screen>
    <![CDATA[
mysql> SHOW STATUS LIKE 'created_tmp%';
+-----+-----+
| Variable_name      | Value |
+-----+-----+
| Created_tmp_disk_tables | 0     |
| Created_tmp_files    | 5     |
| Created_tmp_tables    | 0     |
+-----+-----+
3 rows in set (0.00 sec)
```

7.6. 排序统计信息

```
mysql> SHOW STATUS LIKE "sort%";
```

Variable_name	Value
Sort_merge_passes	0
Sort_range	0
Sort_rows	0
Sort_scan	0

```
4 rows in set (0.00 sec)
```

7.7. Key 状态

```
mysql> show status like '%key_read%';
```

Variable_name	Value
Key_read_requests	6
Key_reads	3

```
2 rows in set (0.00 sec)
```

7.8. FUNCTION

```
SHOW FUNCTION STATUS WHERE `Db`='dbname';
```

7.9. PROCEDURE

```
SHOW PROCEDURE STATUS WHERE `Db`='dbname';
```

7.10. TRIGGERS

```
SHOW TRIGGERS FROM `dbname`;
```

```
mysql> SHOW TRIGGERS LIKE '%trigger_name%'\G
Empty set (0.00 sec)

mysql> SHOW TRIGGERS LIKE '%demo%'\G
***** 1. row
*****
          Trigger: demo_AFTER_INSERT
             Event: INSERT
             Table: demo
        Statement: BEGIN
        set @rev = "";
        SELECT
        OUT2FILE('/tmp/demo.log',
          CONCAT_WS(',',
            NEW.id,
            NEW.name,
            NEW.sex,
            NEW.address))
        INTO @rev;
        END
          Timing: AFTER
         Created: 2017-11-23 11:47:58.10
        sql_mode:
ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION
          Definer: root@%
character_set_client: utf8
collation_connection: utf8_general_ci
  Database Collation: utf8_general_ci
1 row in set (0.00 sec)
```


7.11. EVENTS

```
SHOW EVENTS FROM `dbname`;
```

7.12. 引擎(ENGINES)

```
mysql> SHOW ENGINES;
```

```
+-----+-----+-----+-----+
| Engine          | Support | Comment          |
| Transactions   | XA      | Savepoints      |
+-----+-----+-----+-----+
+-----+
| CSV            | YES     | CSV storage engine
| NO             | NO      |
| MRG_MYISAM    | YES     | Collection of identical MyISAM
tables          | NO      | NO      | NO
|
| PERFORMANCE_SCHEMA | YES     | Performance Schema
| NO             | NO      |
| BLACKHOLE      | YES     | /dev/null storage engine
(anything you write to it disappears) | NO      | NO      |
NO
| MEMORY         | YES     | Hash based, stored in memory,
useful for temporary tables | NO      | NO      | NO
|
| FEDERATED      | NO      | Federated MySQL storage engine
| NULL          | NULL    | NULL
| ARCHIVE        | YES     | Archive storage engine
| NO             | NO      | NO
| MyISAM         | YES     | MyISAM storage engine
| NO             | NO      | NO
```

```

| InnoDB          | DEFAULT | Supports transactions, row-
level locking, and foreign keys | YES      | YES  | YES
|
+-----+-----+-----+-----+
-----+
9 rows in set (0.00 sec)

```

7.13. 字符集(Collation)

```

mysql> SHOW COLLATION;
+-----+-----+-----+-----+
--+-----+
| Collation          | Charset | Id  | Default |
Compiled | Sortlen |
+-----+-----+-----+-----+
--+-----+
| big5_chinese_ci   | big5    | 1  | Yes     | Yes
| 1 |
| big5_bin          | big5    | 84 |         | Yes
| 1 |
| dec8_swedish_ci   | dec8    | 3  | Yes     | Yes
| 1 |
| dec8_bin          | dec8    | 69 |         | Yes
| 1 |
| cp850_general_ci  | cp850   | 4  | Yes     | Yes
| 1 |
| cp850_bin         | cp850   | 80 |         | Yes
| 1 |
| hp8_english_ci    | hp8     | 6  | Yes     | Yes
| 1 |
| hp8_bin           | hp8     | 72 |         | Yes
| 1 |
| koi8r_general_ci  | koi8r   | 7  | Yes     | Yes
| 1 |
| koi8r_bin         | koi8r   | 74 |         | Yes
| 1 |
| latin1_german1_ci | latin1  | 5  |         | Yes
| 1 |

```

latin1_swedish_ci 1	latin1	8	Yes	Yes
latin1_danish_ci 1	latin1	15		Yes
latin1_german2_ci 2	latin1	31		Yes
latin1_bin 1	latin1	47		Yes
latin1_general_ci 1	latin1	48		Yes
latin1_general_cs 1	latin1	49		Yes
latin1_spanish_ci 1	latin1	94		Yes
latin2_czech_cs 4	latin2	2		Yes
latin2_general_ci 1	latin2	9	Yes	Yes
latin2_hungarian_ci 1	latin2	21		Yes
latin2_croatian_ci 1	latin2	27		Yes
latin2_bin 1	latin2	77		Yes
swe7_swedish_ci 1	swe7	10	Yes	Yes
swe7_bin 1	swe7	82		Yes
ascii_general_ci 1	ascii	11	Yes	Yes
ascii_bin 1	ascii	65		Yes
ujis_japanese_ci 1	ujis	12	Yes	Yes
ujis_bin 1	ujis	91		Yes
sjis_japanese_ci 1	sjis	13	Yes	Yes
sjis_bin 1	sjis	88		Yes
hebrew_general_ci 1	hebrew	16	Yes	Yes
hebrew_bin 1	hebrew	71		Yes
tis620_thai_ci	tis620	18	Yes	Yes

4					
tis620_bin		tis620		89	Yes
1					
euckr_korean_ci		euckr		19	Yes Yes
1					
euckr_bin		euckr		85	Yes
1					
koi8u_general_ci		koi8u		22	Yes Yes
1					
koi8u_bin		koi8u		75	Yes
1					
gb2312_chinese_ci		gb2312		24	Yes Yes
1					
gb2312_bin		gb2312		86	Yes
1					
greek_general_ci		greek		25	Yes Yes
1					
greek_bin		greek		70	Yes
1					
cp1250_general_ci		cp1250		26	Yes Yes
1					
cp1250_czech_cs		cp1250		34	Yes
2					
cp1250_croatian_ci		cp1250		44	Yes
1					
cp1250_bin		cp1250		66	Yes
1					
cp1250_polish_ci		cp1250		99	Yes
1					
gbk_chinese_ci		gbk		28	Yes Yes
1					
gbk_bin		gbk		87	Yes
1					
latin5_turkish_ci		latin5		30	Yes Yes
1					
latin5_bin		latin5		78	Yes
1					
armscii8_general_ci		armscii8		32	Yes Yes
1					
armscii8_bin		armscii8		64	Yes
1					
utf8_general_ci		utf8		33	Yes Yes
1					
utf8_bin		utf8		83	Yes
1					

utf8_unicode_ci	utf8	192		Yes
8				
utf8_icelandic_ci	utf8	193		Yes
8				
utf8_latvian_ci	utf8	194		Yes
8				
utf8_romanian_ci	utf8	195		Yes
8				
utf8_slovenian_ci	utf8	196		Yes
8				
utf8_polish_ci	utf8	197		Yes
8				
utf8_estonian_ci	utf8	198		Yes
8				
utf8_spanish_ci	utf8	199		Yes
8				
utf8_swedish_ci	utf8	200		Yes
8				
utf8_turkish_ci	utf8	201		Yes
8				
utf8_czech_ci	utf8	202		Yes
8				
utf8_danish_ci	utf8	203		Yes
8				
utf8_lithuanian_ci	utf8	204		Yes
8				
utf8_slovak_ci	utf8	205		Yes
8				
utf8_spanish2_ci	utf8	206		Yes
8				
utf8_roman_ci	utf8	207		Yes
8				
utf8_persian_ci	utf8	208		Yes
8				
utf8_esperanto_ci	utf8	209		Yes
8				
utf8_hungarian_ci	utf8	210		Yes
8				
utf8_sinhala_ci	utf8	211		Yes
8				
utf8_general_mysql500_ci	utf8	223		Yes
1				
ucs2_general_ci	ucs2	35	Yes	Yes
1				
ucs2_bin	ucs2	90		Yes

1						
ucs2_unicode_ci		ucs2		128		Yes
8						
ucs2_icelandic_ci		ucs2		129		Yes
8						
ucs2_latvian_ci		ucs2		130		Yes
8						
ucs2_romanian_ci		ucs2		131		Yes
8						
ucs2_slovenian_ci		ucs2		132		Yes
8						
ucs2_polish_ci		ucs2		133		Yes
8						
ucs2_estonian_ci		ucs2		134		Yes
8						
ucs2_spanish_ci		ucs2		135		Yes
8						
ucs2_swedish_ci		ucs2		136		Yes
8						
ucs2_turkish_ci		ucs2		137		Yes
8						
ucs2_czech_ci		ucs2		138		Yes
8						
ucs2_danish_ci		ucs2		139		Yes
8						
ucs2_lithuanian_ci		ucs2		140		Yes
8						
ucs2_slovak_ci		ucs2		141		Yes
8						
ucs2_spanish2_ci		ucs2		142		Yes
8						
ucs2_roman_ci		ucs2		143		Yes
8						
ucs2_persian_ci		ucs2		144		Yes
8						
ucs2_esperanto_ci		ucs2		145		Yes
8						
ucs2_hungarian_ci		ucs2		146		Yes
8						
ucs2_sinhala_ci		ucs2		147		Yes
8						
ucs2_general_mysql500_ci		ucs2		159		Yes
1						
cp866_general_ci		cp866		36		Yes
1						

cp866_bin	cp866	68		Yes
1				
keybcs2_general_ci	keybcs2	37	Yes	Yes
1				
keybcs2_bin	keybcs2	73		Yes
1				
macce_general_ci	macce	38	Yes	Yes
1				
macce_bin	macce	43		Yes
1				
macroman_general_ci	macroman	39	Yes	Yes
1				
macroman_bin	macroman	53		Yes
1				
cp852_general_ci	cp852	40	Yes	Yes
1				
cp852_bin	cp852	81		Yes
1				
latin7_estonian_cs	latin7	20		Yes
1				
latin7_general_ci	latin7	41	Yes	Yes
1				
latin7_general_cs	latin7	42		Yes
1				
latin7_bin	latin7	79		Yes
1				
utf8mb4_general_ci	utf8mb4	45	Yes	Yes
1				
utf8mb4_bin	utf8mb4	46		Yes
1				
utf8mb4_unicode_ci	utf8mb4	224		Yes
8				
utf8mb4_icelandic_ci	utf8mb4	225		Yes
8				
utf8mb4_latvian_ci	utf8mb4	226		Yes
8				
utf8mb4_romanian_ci	utf8mb4	227		Yes
8				
utf8mb4_slovenian_ci	utf8mb4	228		Yes
8				
utf8mb4_polish_ci	utf8mb4	229		Yes
8				
utf8mb4_estonian_ci	utf8mb4	230		Yes
8				
utf8mb4_spanish_ci	utf8mb4	231		Yes

8	utf8mb4_swedish_ci	utf8mb4	232		Yes
8	utf8mb4_turkish_ci	utf8mb4	233		Yes
8	utf8mb4_czech_ci	utf8mb4	234		Yes
8	utf8mb4_danish_ci	utf8mb4	235		Yes
8	utf8mb4_lithuanian_ci	utf8mb4	236		Yes
8	utf8mb4_slovak_ci	utf8mb4	237		Yes
8	utf8mb4_spanish2_ci	utf8mb4	238		Yes
8	utf8mb4_roman_ci	utf8mb4	239		Yes
8	utf8mb4_persian_ci	utf8mb4	240		Yes
8	utf8mb4_esperanto_ci	utf8mb4	241		Yes
8	utf8mb4_hungarian_ci	utf8mb4	242		Yes
8	utf8mb4_sinhala_ci	utf8mb4	243		Yes
8	cp1251_bulgarian_ci	cp1251	14		Yes
1	cp1251_ukrainian_ci	cp1251	23		Yes
1	cp1251_bin	cp1251	50		Yes
1	cp1251_general_ci	cp1251	51	Yes	Yes
1	cp1251_general_cs	cp1251	52		Yes
1	utf16_general_ci	utf16	54	Yes	Yes
1	utf16_bin	utf16	55		Yes
1	utf16_unicode_ci	utf16	101		Yes
8	utf16_icelandic_ci	utf16	102		Yes
8	utf16_latvian_ci	utf16	103		Yes
8					

utf16_romanian_ci 8	utf16	104		Yes
utf16_slovenian_ci 8	utf16	105		Yes
utf16_polish_ci 8	utf16	106		Yes
utf16_estonian_ci 8	utf16	107		Yes
utf16_spanish_ci 8	utf16	108		Yes
utf16_swedish_ci 8	utf16	109		Yes
utf16_turkish_ci 8	utf16	110		Yes
utf16_czech_ci 8	utf16	111		Yes
utf16_danish_ci 8	utf16	112		Yes
utf16_lithuanian_ci 8	utf16	113		Yes
utf16_slovak_ci 8	utf16	114		Yes
utf16_spanish2_ci 8	utf16	115		Yes
utf16_roman_ci 8	utf16	116		Yes
utf16_persian_ci 8	utf16	117		Yes
utf16_esperanto_ci 8	utf16	118		Yes
utf16_hungarian_ci 8	utf16	119		Yes
utf16_sinhala_ci 8	utf16	120		Yes
cp1256_general_ci 1	cp1256	57	Yes	Yes
cp1256_bin 1	cp1256	67		Yes
cp1257_lithuanian_ci 1	cp1257	29		Yes
cp1257_bin 1	cp1257	58		Yes
cp1257_general_ci 1	cp1257	59	Yes	Yes
utf32_general_ci	utf32	60	Yes	Yes

1	utf32_bin	utf32	61		Yes
1	utf32_unicode_ci	utf32	160		Yes
8	utf32_icelandic_ci	utf32	161		Yes
8	utf32_latvian_ci	utf32	162		Yes
8	utf32_romanian_ci	utf32	163		Yes
8	utf32_slovenian_ci	utf32	164		Yes
8	utf32_polish_ci	utf32	165		Yes
8	utf32_estonian_ci	utf32	166		Yes
8	utf32_spanish_ci	utf32	167		Yes
8	utf32_swedish_ci	utf32	168		Yes
8	utf32_turkish_ci	utf32	169		Yes
8	utf32_czech_ci	utf32	170		Yes
8	utf32_danish_ci	utf32	171		Yes
8	utf32_lithuanian_ci	utf32	172		Yes
8	utf32_slovak_ci	utf32	173		Yes
8	utf32_spanish2_ci	utf32	174		Yes
8	utf32_roman_ci	utf32	175		Yes
8	utf32_persian_ci	utf32	176		Yes
8	utf32_esperanto_ci	utf32	177		Yes
8	utf32_hungarian_ci	utf32	178		Yes
8	utf32_sinhala_ci	utf32	179		Yes
1	binary	binary	63	Yes	Yes

geostd8_general_ci	geostd8	92	Yes	Yes
1				
geostd8_bin	geostd8	93		Yes
1				
cp932_japanese_ci	cp932	95	Yes	Yes
1				
cp932_bin	cp932	96		Yes
1				
eucjpms_japanese_ci	eucjpms	97	Yes	Yes
1				
eucjpms_bin	eucjpms	98		Yes
1				

7.14. SHOW GRANTS

```

MariaDB [test]> SHOW GRANTS;
+-----+
| Grants for root@localhost |
+-----+
| GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' IDENTIFIED
BY PASSWORD '*C6325DAF39AE6CC34E960D3C65F1398FE467E1D0' WITH
GRANT OPTION |
| GRANT PROXY ON ''@'' TO 'root'@'localhost' WITH GRANT OPTION
|
+-----+
2 rows in set (0.00 sec)

```

7.15. validate_password

```


```

```
SHOW VARIABLES LIKE 'validate_password.%';
```

```
mysql> SHOW VARIABLES LIKE 'validate_password.%';
```

Variable_name	Value
validate_password.check_user_name	ON
validate_password.dictionary_file	
validate_password.length	8
validate_password.mixed_case_count	1
validate_password.number_count	1
validate_password.policy	MEDIUM
validate_password.special_char_count	1

```
7 rows in set (0.00 sec)
```

8. Maintenance 数据库维护

8.1. CHECK 检查表

```
<![CDATA[  
CHECK TABLE `dbname`.`actions`;  
CHECK TABLE `dbname`.`actions` QUICK FAST MEDIUM EXTENDED  
CHANGED;
```

8.2. ANALYZE 分析表

```
ANALYZE TABLE `dbname`.`actions`;
```

8.3. CHECKSUM

```
CHECKSUM TABLE `dbname`.`actions` QUICK;
```

8.4. OPTIMIZE 优化表

```
OPTIMIZE TABLE `dbname`.`actions`;
```

8.5. REPAIR 修复

```
REPAIR TABLE `dbname`.`members`;
```

```
SHOW TABLE STATUS LIKE 'members';
```

9. INFORMATION_SCHEMA

9.1. 查询表字段

```
SELECT
    GROUP_CONCAT(COLUMN_NAME) AS fields
FROM
    INFORMATION_SCHEMA.Columns
WHERE
    table_name = 'mytable'
    AND table_schema = 'test';
```

9.2. 列出所有触发器

```
select trigger_schema, trigger_name, action_statement from
information_schema.triggers

select * from information_schema.triggers where
information_schema.triggers.trigger_schema like '%test%';

select * from information_schema.triggers where
information_schema.triggers.trigger_name like '%trigger_name%' and
information_schema.triggers.trigger_schema like '%data_base_name%';
```

10. Backup and Recovery

10.1. Import / Export

Export(Backup)

```
mysqldump -hlocalhost -proot -p**** mydb > mydb.sql
```

gzip

```
mysqldump -hlocalhost -proot -p**** mydb | gzip > mydb.sql.gz
```

Import(Recovery)

```
mysql -hlocalhost -proot -p**** mydb < mydb.sql
```

gunzip

```
gunzip mydb.sql.gz -c | mysql -hlocalhost -proot -p**** mydb
```

xml

export xml

```
$ mysqldump -uusername -ppasswd -X -t database table -r  
filename.xml
```

备份表数据

```
SELECT * INTO OUTFILE 'file_name' FROM tbl_name  
LOAD DATA INFILE 'file_name' REPLACE INTO TABLE tbl_name
```

source

```
mysql> use your_db  
mysql> SOURCE database.sql
```

使用 **mysqlhotcopy** 备份 **MyISAM** 引擎的数据库

shell> mysqlhotcopy db_name /path/to/some/dir

```
mysql:~# mysqlhotcopy --user=neo --password=chen shop  
/tmp/backup  
Locked 100 tables in 0 seconds.  
Flushed tables (`shop`.`account_log`, `shop`.`ad`,  
`shop`.`ad_custom`, `shop`.`ad_position`,  
`shop`.`admin_action`,  
`shop`.`admin_log`, `shop`.`admin_message`,  
`shop`.`admin_user`, `shop`.`adsense`, `shop`.`affiliate_log`,  
...  
...  
...  
`shop`.`user_rank`, `shop`.`users`, `shop`.`virtual_card`,  
`shop`.`volume_price`, `shop`.`vote`, `shop`.`vote_log`,
```

```
`shop`.`vote_option`, `shop`.`wholesale`) in 0 seconds.  
Copying 299 files...  
Copying indices for 0 files...  
Unlocked tables.  
mysqlhotcopy copied 100 tables (299 files) in 0 seconds (0  
seconds overall).
```

AutoMySQLBackup

<https://sourceforge.net/projects/automysqlbackup/>

xtrabackup - Open source backup tool for InnoDB and XtraDB.

<https://launchpad.net/percona-xtrabackup>

Percona yum Repository

```
$ rpm -Uvh http://www.percona.com/downloads/percona-  
release/percona-release-0.0-1.x86_64.rpm
```

```
# yum search xtrabackup  
=====  
= N/S Matched: XtraBackup  
=====  
==  
holland-xtrabackup.noarch : Xtrabackup plugin for Holland  
percona-xtrabackup.x86_64 : XtraBackup online backup for MySQL  
/ InnoDB  
percona-xtrabackup-debuginfo.x86_64 : Debug information for  
package percona-xtrabackup  
percona-xtrabackup-test.x86_64 : Test suite for Percona  
Xtrabackup
```

```
# yum install percona-xtrabackup
```

Creating an Incremental Backup

```
xtrabackup --backup --target-dir=/data/backups/base --  
datadir=/var/lib/mysql/
```

10.2. Snapshot Backup

LVM Snapshot

```
# mysql -uroot -pmysql  
mysql> flush tables with read lock;  
mysql>flush logs;  
mysql>system lvcreate -L1024M -s -n snap0 /dev/vg00/lvol00  
mysql>show master status;  
mysql>unlock tables;  
mysql>quit
```

Btrfs Snapshot

```
# btrfs subvolume snapshot /data /data/backup_2013-03-20  
Create a snapshot of '/data' in '/data/backup_2013-03-20'  
  
btrfs subvolume list /data  
ID 315 gen 172 top level 5 path backup_2013-03-10  
ID 320 gen 178 top level 5 path backup_2013-03-20
```

第 60 章 PostgreSQL 命令

1. psql - PostgreSQL interactive terminal

```
postgres=# \?
一般性
\copyright          显示PostgreSQL的使用和发行许可条款
\errverbose         以最冗长的形式显示最近的错误消息
\g [文件] or;      执行查询 (并把结果写入文件或 |管道)
\gexec             执行策略, 然后执行其结果中的每个值
\gset [PREFIX]     执行查询并把结果存到psql变量中
\q                退出 psql
\crosstabview [COLUMNS] 执行查询并且以交叉表显示结果
\watch [SEC]       每隔SEC秒执行一次查询

帮助
\? [commands]     显示反斜线命令的帮助
\? options         显示 psql 命令行选项的帮助
\? variables       显示特殊变量的帮助
\h [名称]          SQL命令语法上的说明, 用*显示全部命令的语法说明

查询缓存区
\e [FILE] [LINE]   使用外部编辑器编辑查询缓存区 (或文件)
\ef [FUNCNAME [LINE]] 使用外部编辑器编辑函数定义
\ev [VIEWNAME [LINE]] 用外部编辑器编辑视图定义
\p                显示查询缓存区的内容
\r                重置 (清除) 查询缓存区
\w 文件            将查询缓存区的内容写入文件

输入/输出
\copy ...          执行 SQL COPY, 将数据流发送到客户端主机
\echo [字符串]     将字符串写到标准输出
\i 文件            从文件中执行命令
\ir FILE           与 \i类似, 但是相对于当前脚本的位置
\o [文件]          将全部查询结果写入文件或 |管道
\qecho [字符串]   将字符串写到查询输出串流(参考 \o)

资讯性
(选项: s = 显示系统对象, + = 其余的详细信息)
\d[S+]            列出表, 视图和序列
```

	名称	描述表, 视图, 序
\d[S+]		列, 或索引
\da[S] [模式]		列出聚合函数
\dA[+] [PATTERN]		list access methods
\db[+] [模式]		列出表空间
\dc[S+] [PATTERN]		列表转换
\dC[+] [PATTERN]		列出类型强制转换
\dd[S] [PATTERN]		显示没有在别处显示的对象描述
\ddp [模式]		列出默认权限
\dD[S+] [PATTERN]		列出共同值域
\det[+] [PATTERN]		列出引用表
\des[+] [模式]		列出外部服务器
\deu[+] [模式]		列出用户映射
\dew[+] [模式]		列出外部数据封装器
\df[antw][S+] [模式]		列出[只包括 聚合/常规/触发器/窗口]函数
\dF[+] [模式]		列出文本搜索配置
\dFd[+] [模式]		列出文本搜索字典
\dFp[+] [模式]		列出文本搜索解析器
\dFt[+] [模式]		列出文本搜索模版
\dg[S+] [PATTERN]		列出角色
\di[S+] [模式]		列出索引
\dl		列出大对象, 功能与\lo_list相同
\dL[S+] [PATTERN]		列出所有过程语言
\dm[S+] [PATTERN]		列出所有物化视图
\dn[S+] [PATTERN]		列出所有模式
\do[S] [模式]		列出运算符
\dO[S+] [PATTERN]		列出所有校对规则
\dp [模式]		列出表, 视图和序列的访问权限
\drds [模式1 [模式2]]		列出每个数据库的角色设置
\ds[S+] [模式]		列出序列
\dt[S+] [模式]		列出表
\dT[S+] [模式]		列出数据类型
\du[S+] [PATTERN]		列出角色
\dv[S+] [模式]		列出视图
\dE[S+] [PATTERN]		列出引用表
\dx[+] [PATTERN]		列出扩展
\dy [PATTERN]		列出所有事件触发器
\l[+] [PATTERN]		列出所有数据库
\sf[+] FUNCNAME		显示一个函数的定义
\sv[+] VIEWNAME		显示一个视图的定义
\z [模式]		和\dp的功能相同
格式化		
\a		在非对齐模式和对齐模式之间切换
\C [字符串]		设置表的标题, 或如果没有的标题就取消
\f [字符串]		显示或设定非对齐模式查询输出的字段分隔符

```

\H          切换HTML输出模式 (目前是 关闭)
\pset [NAME [VALUE]] 设置表输出选项
                (NAME :=
{format|border|expanded|fieldsep|fieldsep_zero|footer|null|
numericlocale|recordsep|recordsep_zero|tuples_only|title|tableat
tr|pager|
unicode_border_linestyle|unicode_column_linestyle|unicode_header
_linestyle})
\t [开|关]          只显示记录 (目前是 关闭)
\T [字符串]        设置HTML <表格>标签属性, 或者如果没有的话取消
设置
\x [on|off|auto]    切换扩展输出模式(目前是 关闭)

连接
\c[onnect] {[DBNAME|- USER|- HOST|- PORT|-] | conninfo}
                连接到新数据库 (当
前是"postgres")
\encoding [编码名称] 显示或设定客户端编码
\password [USERNAME] 安全地为用户更改口令
\conninfo          显示当前连接的相关信息

操作系统
\cd [目录]          更改目前的工作目录
\setenv NAME [VALUE] 设置或清空环境变量
\timing [开|关]    切换命令计时开关 (目前是 关闭)
\! [命令]          在 shell中执行命令或启动一个交互式
shell

变量
\prompt [文本] 名称 提示用户设定内部变量
\set [名称 [值数]] 设定内部变量, 若无参数则列出全部变量
\unset 名称        清空(删除)内部变量

大对象
\lo_export LOBID 文件
\lo_import 文件 [注释]
\lo_list
\lo_unlink LOBID   大对象运算

```

1.1. 执行文件

```
psql -U user -d your-db -f your-table.sql
```

2. Backup / Restore

2.1. Backup

本地备份

```
pg_dump --compress=9 --file=/backup/mydb.sql.gz mydb
```

Backup

```
pg_dump -i -h localhost -p 5432 -U user -Fc -c -f  
"your_db.backup" your_db
```

2.2. Restore

Restore

```
pg_restore -i -h localhost -p 5432 -U user -d your_db -v -c -Fc  
"your_db.backup"
```

2.3. import

导入

\i file.backup


```
postgres=# \i /home/neo/woodart.backup
```

2.4. 导出指定 schema

```
pg_dump -h [host] -U [dbuser] -s [dbname] -n [schema] > [file path]
```

2.5. PostgreSQL for windows

local -> remote

```
set PG_HOME="C:\Program Files\PostgreSQL\8.1\bin"  
%PG_HOME%\pg_dump.exe -i -h localhost -p 5432 -U woodart -Fc -c  
-f "woodart.backup" woodart  
%PG_HOME%\pg_restore.exe -i -h woodart.8800.org -p 5432 -U  
woodart -d woodart -v -c -Fc "woodart.backup"
```

2.6. 大型数据库备份

split 命令可以将大型文件切成小块以适应文件系统限制的单个文件大小。

分割,每650M为一个文件, 试用于光盘备份

```
$ pg_dump dbname | split -b 650m - filename
```

合并，并且恢复到数据库中。

```
$ createdb dbname  
$ cat filename* | psql dbname
```

2.7. schema之间迁移数据

从 schema 中导出数据

```
pg_dump -h 192.168.1.242 -U user -s database -n schema >  
schema.sql
```

导入目标数据库

```
psql -U user -d your-db -f schema.sql
```

第 61 章 DDL - Data Definition Language

1. 数据库管理(Database)

1.1. create

Creating a UTF-8 database

```
CREATE DATABASE db_name DEFAULT CHARACTER SET utf8 COLLATE  
utf8_general_ci;
```

Create a UTF-8 database with binary UTF-8 collation.

```
CREATE DATABASE dbname CHARACTER SET utf8 COLLATE utf8_bin;
```

1.2. drop

```
DROP DATABASE db_name;
```

1.3. 修改数据库

```
ALTER DATABASE dbname DEFAULT CHARACTER SET utf8 COLLATE  
utf8_general_ci;
```

1.4. Rename

```
RENAME {DATABASE | SCHEMA} db_name TO new_db_name;
```

before 5.0 version

```
[neo@development ~]$ mysqldump -uroot -pchen db_old | mysql -uroot -pchen db_new
```

1.5. CHARACTER

```
ALTER DATABASE <database_name> CHARACTER SET utf8;
```

1.6. show create database

```
mysql> show create database dbname;
```

```
+-----+-----+
| Database | Create Database
|
+-----+-----+
| dbname   | CREATE DATABASE `dbname` /*!40100 DEFAULT
CHARACTER SET utf8 */
+-----+-----+
1 row in set (0.00 sec)
```

2. 表管理(Table)

2.1. 数据类型

SET 集合类型

SET 集合类型，此类型适合用于多项选择场景，例如保存表单中的checkbox。

```
CREATE TABLE `QA` (  
  `id` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,  
  `question` VARCHAR(255) NOT NULL COMMENT '问题描述',  
  `answer` SET('A','B','C','D') NOT NULL COMMENT '问题答案',  
  PRIMARY KEY (`id`)  
)  
COMMENT='Multiple Choice'  
COLLATE='utf8_general_ci'  
ENGINE=InnoDB;
```

插入数据

```
INSERT INTO `QA` (`id`, `question`, `answer`) VALUES  
  (1, 'Netkiller 系列手札始于那一年? A.2000年, B.2008年, C.2010年, D.2016年', 'A'),  
  (2, 'Netkiller 系列手札有哪些? A.《Netkiller Scals 手札》, B.《Netkiller Java 手札》,  
C.《Netkiller Linux 手札》, D.《Netkiller EMC 手札》', 'B,C'),  
  (3, 'XXXXXXXXX', 'C,D'),  
  (4, 'XXXXXXXXX', 'A,B,C'),  
  ...  
  ...  
  (1000, 'XXXXXXXXXX', 'B,C,D'),  
  ...  
  ...  
  (5000, 'XXXXXXXXXX', 'A,B,C,D');
```

查询 SET 结果集，MySQL为SET配备了FIND_IN_SET函数

```
select * from QA where FIND_IN_SET('B',`answer`);
```

下面两种方法也能实现，但不推荐使用。

```
select question, answer from QA where locate('B',answer)>0;  
select question, answer from QA where POSITION('B' in answer)>0;
```

查询多个答案

```
select question, answer from QA where answer = 'B,C';
```

2.2. create table ... select

创建空表

```
create table admin_user_history select * from admin_user where 1 <> 1;
```

创建有数据的表

```
create table admin_user_history select * from admin_user;
```

2.3. modify table

modify table

```
ALTER TABLE ecs_users add user_picture varchar(255);
```

2.4. TEMPORARY Table

临时表将在你连接期间存在。一旦断开时将自动删除表并释放所用的空间。你在连接期间删除该表也同样释放空间。

```
CREATE TEMPORARY TABLE tmp_table (  
    key VARCHAR(10) NOT NULL,  
    value INTEGER NOT NULL  
)
```

声明临时表是一个HEAP表，允许你指定在内存中创建它

```
CREATE TEMPORARY TABLE tmp_mem_table (  
    key VARCHAR(10) NOT NULL,  
    value INTEGER NOT NULL  
) TYPE = HEAP
```

2.5. Collate

```
ALTER TABLE `tmp_cats` COLLATE='utf8_general_ci', CONVERT TO CHARSET utf8;
```

2.6. CHARACTER

```
ALTER TABLE <table_name> CONVERT TO CHARACTER SET utf8;  
alter table <table_name> convert to charset utf8mb4;
```

2.7. DEFAULT

AUTO_INCREMENT

定义 AUTO_INCREMENT 起始值

```
CREATE TABLE `bank_account` (  
    `id` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT COMMENT '自增唯一ID',  
    `name` VARCHAR(50) NOT NULL DEFAULT '0' COMMENT '帐号名称(Name on account)',  
    PRIMARY KEY (`id`)  
)  
COMMENT='银行帐号'  
COLLATE='utf8_general_ci'  
ENGINE=InnoDB  
AUTO_INCREMENT=2;
```

设置 AUTO_INCREMENT

```
ALTER TABLE `accounts`  
    AUTO_INCREMENT=792257;
```


TIMESTAMP NULL DEFAULT NULL ON UPDATE

```
alter table cms.article ADD COLUMN `mtime` TIMESTAMP NULL DEFAULT NULL ON UPDATE  
CURRENT_TIMESTAMP COMMENT '更改时间';
```

更新时间

```
`mtime` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE  
CURRENT_TIMESTAMP COMMENT '更改时间',
```

```
CREATE TABLE `bank_account` (  
  `id` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT COMMENT '自增唯一ID',  
  `bank_name` VARCHAR(255) NOT NULL DEFAULT '0' COMMENT '银行名字(Bank Name)',  
  `name` VARCHAR(50) NOT NULL DEFAULT '0' COMMENT '帐号名称(Name on account)',  
  `account_number` VARCHAR(50) NOT NULL DEFAULT '0' COMMENT '银行帐号(Account  
Number)',  
  `branch_location` VARCHAR(255) NOT NULL DEFAULT '0' COMMENT '支行位置(Branch  
Location)',  
  `description` VARCHAR(255) NOT NULL DEFAULT '0' COMMENT '银行描述',  
  `status` ENUM('Y','N') NOT NULL DEFAULT 'N' COMMENT '银行帐号状态',  
  `ctime` DATETIME NOT NULL COMMENT '创建时间',  
  `mtime` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP  
COMMENT '更改时间',  
  PRIMARY KEY (`id`)  
)  
COMMENT='银行帐号'  
COLLATE='utf8_general_ci'  
ENGINE=InnoDB  
AUTO_INCREMENT=2;
```

表存储位置(DATA DIRECTORY)

```
CREATE TABLE IF NOT EXISTS `tab_name` (  
  `id` int(11) DEFAULT NULL,  
  `purchased` date DEFAULT NULL,  
  KEY `Index 1` (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8  
/*!50100 PARTITION BY LIST (YEAR(purchased))  
(PARTITION p0 VALUES IN (1990) DATA DIRECTORY = '/www/data' ENGINE = InnoDB) */;
```

2.8. KEY

PRIMARY KEY

一般主键定义

```
PRIMARY KEY (`id`),
```

复合主键

```
PRIMARY KEY (`id`, `user_id`),
```

2.9. AUTO_INCREMENT 定义初始值

```
DROP TABLE IF EXISTS users;  
CREATE TABLE user(  
    id          INT     NOT NULL AUTO_INCREMENT  
PRIMARY KEY(id)  
)ENGINE=InnoDB AUTO_INCREMENT=100 DEFAULT CHARSET=utf8;
```

修改 auto_increment 起始值

```
alter table tabl auto_increment=n
```

2.10. COMMENT

```
ALTER TABLE `neo`.`stuff` COMMENT = '用户表' ;  
ALTER TABLE `neo`.`stuff` CHANGE COLUMN `name` `name` VARCHAR(50) NULL DEFAULT NULL  
COMMENT '姓名' ;  
ALTER TABLE `neo`.`stuff` CHANGE COLUMN `password` `password` VARCHAR(50) NULL DEFAULT  
NULL COMMENT '用户密码' ;  
  
CREATE TABLE `stuff` (  
    `id` int(11) NOT NULL AUTO_INCREMENT,  
    `name` varchar(50) DEFAULT NULL COMMENT ''姓名'',  
    `password` varchar(50) DEFAULT NULL COMMENT ''用户密码'',  
    `created` date NOT NULL DEFAULT ''0000-00-00'',  
    PRIMARY KEY (`id`,`created`)  
) ENGINE=MyISAM AUTO_INCREMENT=5 DEFAULT CHARSET=latin1 COMMENT=''用户表''  
/*!50100 PARTITION BY HASH (year(created))  
PARTITIONS 10 */
```

2.11. Engine 存储引擎

显示当前数据库支持引擎

```
mysql> show engines;
+-----+-----+-----+
| Engine          | Support | Comment                                     |
| Transactions | XA      | Savepoints |
+-----+-----+-----+
| FEDERATED       | NO      | Federated MySQL storage engine           |
| NULL            | NULL    | NULL                                       |
| MRG_MYISAM      | YES     | Collection of identical MyISAM tables     |
| NO              | NO      | NO                                         |
| MyISAM          | YES     | MyISAM storage engine                    |
| NO              | NO      | NO                                         |
| BLACKHOLE       | YES     | /dev/null storage engine (anything you write to it disappears) |
| NO              | NO      | NO                                         |
| MEMORY          | YES     | Hash based, stored in memory, useful for temporary tables |
| NO              | NO      | NO                                         |
| CSV             | YES     | CSV storage engine                       |
| NO              | NO      | NO                                         |
| ARCHIVE         | YES     | Archive storage engine                   |
| NO              | NO      | NO                                         |
| PERFORMANCE_SCHEMA | YES     | Performance Schema                       |
| NO              | NO      | NO                                         |
| InnoDB          | DEFAULT | Supports transactions, row-level locking, and foreign keys |
| YES            | YES     | YES                                         |
+-----+-----+-----+
9 rows in set (0.00 sec)
```

切换引擎

修改与切换引擎

```
ALTER TABLE `test` ENGINE=BLACKHOLE;
ALTER TABLE `test` ENGINE=InnoDB;
```

FEDERATED

启用 FEDERATED 引擎, 服务器环境 Ubuntu 13.04

```
$ sudo vim /etc/mysql/conf.d/federated.cnf
[mysqld]
federated

$ sudo service mysql restart
```

```
mysql> show engines;
+-----+-----+-----+
| Engine | Support | Comment |
| Transactions | XA | Savepoints |
+-----+-----+-----+
| FEDERATED | YES | Federated MySQL storage engine |
| NO | NO | |
| MRG_MYISAM | YES | Collection of identical MyISAM tables |
| NO | NO | |
| MyISAM | YES | MyISAM storage engine |
| NO | NO | |
| BLACKHOLE | YES | /dev/null storage engine (anything you write to it |
disappears) | NO | NO |
| MEMORY | YES | Hash based, stored in memory, useful for temporary |
tables | NO | NO |
| CSV | YES | CSV storage engine |
| NO | NO | |
| ARCHIVE | YES | Archive storage engine |
| NO | NO | |
| PERFORMANCE_SCHEMA | YES | Performance Schema |
| NO | NO | |
| InnoDB | DEFAULT | Supports transactions, row-level locking, and foreign |
keys | YES | YES |
+-----+-----+-----+
9 rows in set (0.00 sec)
```

A 服务器

```
CREATE TABLE `t1` (
  `id` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(50) NOT NULL,
  `sex` ENUM('Y','N') NULL DEFAULT 'Y',
  `passwd` VARCHAR(50) NULL DEFAULT NULL,
  `ctime` TIMESTAMP NOT NULL DEFAULT '0000-00-00 00:00:00',
  `mtime` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
  PRIMARY KEY (`id`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
AUTO_INCREMENT=4;
```

B 服务器

```
DROP TABLE `users`;

CREATE TABLE `users` (
  `id` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(50) NOT NULL,
  `sex` ENUM('Y','N') NULL DEFAULT 'Y',
```

```

        `ctime` TIMESTAMP NOT NULL DEFAULT '0000-00-00 00:00:00',
        `mtime` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
        PRIMARY KEY (`id`)
) ENGINE=FEDERATED connection = 'mysql://www:qwer123@192.168.2.1:3306/test/t1';

```

上面字段描述是你需要的字段，并非所有字段。这里屏蔽了passwd字段

提示

```
connection = 'mysql://用户名:密码@主机:端口/数据库/表名'
```

```

mysql> DROP TABLE `users`;
Query OK, 0 rows affected (0.00 sec)

mysql> CREATE TABLE `users` (
  -> `id` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,
  -> `name` VARCHAR(50) NOT NULL,
  -> `sex` ENUM('Y','N') NULL DEFAULT 'Y',
  -> `ctime` TIMESTAMP NOT NULL DEFAULT '0000-00-00 00:00:00',
  -> `mtime` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  -> PRIMARY KEY (`id`)
  -> ) ENGINE=FEDERATED connection = 'mysql://www:qwer123@192.168.2.1:3306/test/t1';
Query OK, 0 rows affected (0.06 sec)

mysql>
mysql> show tables;
+-----+
| Tables_in_test |
+-----+
| users          |
+-----+
1 row in set (0.00 sec)

mysql> desc users;
+-----+-----+-----+-----+-----+-----+
| Field | Type                | Null | Key | Default                | Extra          |
+-----+-----+-----+-----+-----+-----+
| id    | int(10) unsigned   | NO   | PRI | NULL                   | auto_increment |
| name  | varchar(50)        | NO   |     | NULL                   |                |
| sex   | enum('Y','N')      | YES  |     | Y                       |                |
| ctime | timestamp          | NO   |     | 0000-00-00 00:00:00   |                |
| mtime | timestamp          | NO   |     | CURRENT_TIMESTAMP     | on update     |
CURRENT_TIMESTAMP |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select * from users;
+-----+-----+-----+-----+-----+-----+
| id | name | sex | ctime                | mtime                |
+-----+-----+-----+-----+-----+-----+

```

```

+----+-----+-----+-----+-----+-----+
| 1 | neo | Y | 0000-00-00 00:00:00 | 2013-05-17 18:05:09 |
| 2 | zen | Y | 0000-00-00 00:00:00 | 2013-05-17 18:05:11 |
| 3 | lily | N | 0000-00-00 00:00:00 | 2013-05-17 18:05:22 |
+----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)

```

FEDERATED 与 mysqldump 问题!

切记, mysqldump 只会dump出使用FEDERATED引擎表的结构,不会包含数据。

BLACKHOLE

```
CREATE TABLE test(id INT, val CHAR(10)) ENGINE = BLACKHOLE;
```

ARCHIVE

归档(是适用于存放大量数据的存储引擎),仅支持select、insert操作;不支持delete、update、索引等操作;使用zlib无损算法压缩数据,节省磁盘空间;

适用场景:适用于大量可查询但不能删除的历史数据保存;

基于 order 表创建 order_audit 归档表

```
create table order_audit engine=archive as select * from `order`;
```

order_audit 表结构如下

```
CREATE TABLE `order_audit` (
  `id` int(10) unsigned NOT NULL DEFAULT '0' COMMENT '订单ID',
  `name` varchar(45) NOT NULL COMMENT '订单名称',
  `price` float NOT NULL COMMENT '价格',
  `ctime` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间'
) ENGINE=ARCHIVE DEFAULT CHARSET=utf8
```

```
mysql> show table status like 'order_audit';
```

```

+-----+-----+-----+-----+-----+-----+-----+
| Name          | Engine | Version | Row_format | Rows | Avg_row_length | Data_length |
+-----+-----+-----+-----+-----+-----+-----+
| order_audit   | ARCHIVE | 10.0.16 | Dynamic    | 3    | 133              | 133          |
+-----+-----+-----+-----+-----+-----+-----+
| Max_data_length | Index_length | Data_free | Auto_increment | Create_time |
+-----+-----+-----+-----+-----+-----+-----+
| Update_time     | Check_time | Collation | Checksum | Create_options |
+-----+-----+-----+-----+-----+-----+-----+

```

```

Comment |
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
| order_audit | ARCHIVE |      10 | Compressed |      4 |           2215 |           8861 |
0 |           0 |           0 |           NULL | NULL |           |           |
NULL | utf8_general_ci |           NULL |           |           |           |           |
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)

```

CSV

创建表

```

CREATE TABLE `csv_table` (
  `id` int(11) NOT NULL,
  `name` varchar(45) NOT NULL,
  `age` int(11) NOT NULL
) ENGINE=CSV DEFAULT CHARSET=utf8

```

查看表状态

```

mysql> show table status like 'csv_table';
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
| Name      | Engine | Version | Row_format | Rows | Avg_row_length | Data_length |
Max_data_length | Index_length | Data_free | Auto_increment | Create_time |
Update_time | Check_time | Collation      | Checksum | Create_options | Comment |
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
| csv_table | CSV    |      10 | Dynamic    |      2 |           0 |           0 |
0 |           0 |           0 |           NULL | NULL |           NULL |           NULL |
| utf8_general_ci | NULL |           |           |           |           |           |
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

```

插入数据

```

insert into csv_table values (1,'Neo',37),(2,'Jam',40);

```

查看数据

```
mysql> SELECT * FROM test.csv_table;
+----+-----+-----+
| id | name | age |
+----+-----+-----+
|  1 | Neo  |  37 |
|  2 | Jam  |  40 |
+----+-----+-----+
2 rows in set (0.00 sec)
```

CSV 引擎是可以直接将csv文件复制出来的，表存储在 /var/lib/mysql/ 目录

```
root@netkiller /etc/nginx/conf.d % ls -l /var/lib/mysql/test/csv*
/var/lib/mysql/test/csv_table.CSM
/var/lib/mysql/test/csv_table.CSV
/var/lib/mysql/test/csv_table.frm
```

.*CSM,*.frm 是表结构文件，*.CSV 是我们需要的文件，纯文本，可以使用Excel打开。

```
root@netkiller /etc/nginx/conf.d % cat /var/lib/mysql/test/csv_table.CSV
1,"Neo",37
2,"Jam",40
```


3. Partitioning

```
mysql> SHOW VARIABLES LIKE '%partition%';
```

```
+-----+-----+
| Variable_name | Value |
+-----+-----+
| have_partitioning | YES |
+-----+-----+
1 row in set (0.00 sec)
```

3.1. RANGE

18.5.1. Partitioning Keys, Primary Keys, and Unique Keys
This section discusses the relationship of partitioning keys with primary keys and unique keys. The rule governing this relationship can be expressed as follows: All columns used in the partitioning expression for a partitioned table must be part of every unique key that the table may have.

In other words, every unique key on the table must use every column in the table's partitioning expression. (This also includes the table's primary key, since it is by definition a unique key. This particular case is discussed later in this section.) For example, each of the following table creation statements is invalid:

SQL code:

```
mysql> create table tx (
->     id int not null ,
->     info_time date,
->     primary key(id,info_time)
-> )
-> PARTITION BY RANGE(info_time div 100)
-> (
->     PARTITION p_2008_11 VALUES LESS THAN (200812),
```

```
-> PARTITION p_2008_12 VALUES LESS THAN (200901),
-> PARTITION p_2009_01 VALUES LESS THAN (200902),
-> PARTITION p_2009_02 VALUES LESS THAN (200903),
-> PARTITION p_2009_03 VALUES LESS THAN (200904),
-> PARTITION p_2009_04 VALUES LESS THAN (200905),
-> PARTITION p_catch_all VALUES LESS THAN MAXVALUE
-> );
Query OK, 0 rows affected (0.17 sec)

mysql>
```

```
CREATE TABLE t1 (
  year_col INT,
  some_data INT
)
PARTITION BY RANGE (year_col) (
  PARTITION p0 VALUES LESS THAN (1991),
  PARTITION p1 VALUES LESS THAN (1995),
  PARTITION p2 VALUES LESS THAN (1999),
  PARTITION p3 VALUES LESS THAN (2002),
  PARTITION p4 VALUES LESS THAN (2006),
  PARTITION p5 VALUES LESS THAN MAXVALUE
);
```

e.g.2

```
CREATE TABLE rc (
  a INT NOT NULL,
  b INT NOT NULL
)
PARTITION BY RANGE COLUMNS(a,b) (
  PARTITION p0 VALUES LESS THAN (10,5),
  PARTITION p1 VALUES LESS THAN (20,10),
  PARTITION p2 VALUES LESS THAN (MAXVALUE,15),
  PARTITION p3 VALUES LESS THAN (MAXVALUE,MAXVALUE)
);
```

```
CREATE TABLE part_tab
(
    c1 int default NULL,
    c2 varchar(30) default NULL,
    c3 date default NULL
) engine=myisam
PARTITION BY RANGE (year(c3)) (
    PARTITION p0 VALUES LESS THAN (2000) ,
    PARTITION p1 VALUES LESS THAN (2001) ,
    PARTITION p2 VALUES LESS THAN (2002) ,
    PARTITION p3 VALUES LESS THAN (2003) ,
    PARTITION p4 VALUES LESS THAN (2004) ,
    PARTITION p12 VALUES LESS THAN (2012),
    PARTITION p13 VALUES LESS THAN MAXVALUE
);
```

3.2. LIST

```
CREATE TABLE client_firms (
    id INT,
    name VARCHAR(35)
)
PARTITION BY LIST (id) (
    PARTITION r0 VALUES IN (1, 5, 9, 13, 17, 21),
    PARTITION r1 VALUES IN (2, 6, 10, 14, 18, 22),
    PARTITION r2 VALUES IN (3, 7, 11, 15, 19, 23),
    PARTITION r3 VALUES IN (4, 8, 12, 16, 20, 24)
);
```

```

CREATE TABLE lc (
  a INT NULL,
  b INT NULL
)
PARTITION BY LIST COLUMNS(a,b) (
  PARTITION p0 VALUES IN( (0,0), (NULL,NULL) ),
  PARTITION p1 VALUES IN( (0,1), (0,2), (0,3), (1,1), (1,2)
),
  PARTITION p2 VALUES IN( (1,0), (2,0), (2,1), (3,0), (3,1)
),
  PARTITION p3 VALUES IN( (1,3), (2,2), (2,3), (3,2), (3,3)
)
);

```

```

CREATE TABLE th (id INT, name VARCHAR(30), adate DATE)
PARTITION BY LIST(YEAR(adata))
(
  PARTITION p1999 VALUES IN (1995, 1999, 2003)
  DATA DIRECTORY = '/var/appdata/95/data'
  INDEX DIRECTORY = '/var/appdata/95/idx',
  PARTITION p2000 VALUES IN (1996, 2000, 2004)
  DATA DIRECTORY = '/var/appdata/96/data'
  INDEX DIRECTORY = '/var/appdata/96/idx',
  PARTITION p2001 VALUES IN (1997, 2001, 2005)
  DATA DIRECTORY = '/var/appdata/97/data'
  INDEX DIRECTORY = '/var/appdata/97/idx',
  PARTITION p2000 VALUES IN (1998, 2002, 2006)
  DATA DIRECTORY = '/var/appdata/98/data'
  INDEX DIRECTORY = '/var/appdata/98/idx'
);

```

3.3. HASH

```

CREATE TABLE `test` (
  `userid` int(10) unsigned NOT NULL auto_increment,
  `username` int(10) unsigned NOT NULL DEFAULT '0',
  `password` int(10) unsigned NOT NULL DEFAULT '0',

  primary key (`userid`),
  KEY `userid` (`username`)
) ENGINE=InnoDB
PARTITION BY HASH(userid)
PARTITIONS 8;

```

使用HASH (year(created)) 替代 RANGE(year(created))

```

CREATE TABLE stuff (
  id INT AUTO_INCREMENT,
  name varchar(50),
  password varchar(50),
  created DATE,
  PRIMARY KEY (id, created)
)
PARTITION BY RANGE(year(created)) (
  PARTITION p0 VALUES LESS THAN (2010),
  PARTITION p1 VALUES LESS THAN (2012),
  PARTITION p2 VALUES LESS THAN MAXVALUE
);

```

更好的方法

```

CREATE TABLE stuff (
  id INT AUTO_INCREMENT,
  name varchar(50),
  password varchar(50),
  created DATE,
  PRIMARY KEY (id, created)
)
PARTITION BY HASH (year(created)) PARTITIONS 10;

```

我们演示一下

```
mysql> CREATE TABLE stuff (  
-> id INT AUTO_INCREMENT,  
-> name varchar(50),  
-> password varchar(50),  
-> created DATE,  
-> PRIMARY KEY (id, created)  
-> )  
-> PARTITION BY HASH (year(created)) PARTITIONS 10;  
Query OK, 0 rows affected (0.08 sec)
```

```
mysql> insert into stuff (name,password,created)  
values('neo','test','2010-10-1');  
Query OK, 1 row affected (0.06 sec)
```

```
mysql> insert into stuff (name,password,created)  
values('neo1','test','2012-2-1');  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> insert into stuff (name,password,created)  
values('neo2','test','2012-3-5');  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> insert into stuff (name,password,created)  
values('neo4','test','2011-1-5');  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> SELECT  
-> partition_name part,  
-> partition_expression expr,  
-> partition_description descr,  
-> table_rows  
-> FROM  
-> INFORMATION_SCHEMA.partitions  
-> WHERE  
-> TABLE_SCHEMA = schema()  
-> AND TABLE_NAME='stuff';
```

part	expr	descr	table_rows
p0	year(created)	NULL	1
p1	year(created)	NULL	1
p2	year(created)	NULL	2

p3	year(created)	NULL	0
p4	year(created)	NULL	0
p5	year(created)	NULL	0
p6	year(created)	NULL	0
p7	year(created)	NULL	0
p8	year(created)	NULL	0
p9	year(created)	NULL	0

10 rows in set (0.02 sec)

```
mysql> EXPLAIN PARTITIONS SELECT * FROM stuff WHERE
created='2011-01-05'\G
```

```
***** 1. row
*****
```

```
      id: 1
  select_type: SIMPLE
        table: stuff
  partitions: p1
         type: system
possible_keys: NULL
         key: NULL
        key_len: NULL
         ref: NULL
         rows: 1
      Extra:
```

1 row in set (0.08 sec)

```
mysql> EXPLAIN PARTITIONS SELECT * FROM stuff WHERE
created='2012-03-05'\G
```

```
***** 1. row
*****
```

```
      id: 1
  select_type: SIMPLE
        table: stuff
  partitions: p2
         type: ALL
possible_keys: NULL
         key: NULL
        key_len: NULL
         ref: NULL
         rows: 2
      Extra: Using where
```

1 row in set (0.00 sec)

LINEAR HASH

```
CREATE TABLE employees (  
    id INT NOT NULL,  
    fname VARCHAR(30),  
    lname VARCHAR(30),  
    hired DATE NOT NULL DEFAULT '1970-01-01',  
    separated DATE NOT NULL DEFAULT '9999-12-31',  
    job_code INT,  
    store_id INT  
)  
PARTITION BY LINEAR HASH( YEAR(hired) )  
PARTITIONS 4;
```

3.4. KEY分区

按照KEY进行分区类似于按照HASH分区，除了HASH分区使用的用户定义的表达式，而KEY分区的哈希函数是由MySQL服务器提供。MySQL 簇（Cluster）使用函数MD5()来实现KEY分区；

```
CREATE TABLE tk (  
    col1 INT NOT NULL,  
    col2 CHAR(5),  
    col3 DATE  
)  
PARTITION BY LINEAR KEY (col1)  
PARTITIONS 3;
```

3.5. Subpartitioning

```
CREATE TABLE ts (id INT, purchased DATE)
```



```

PARTITION BY RANGE( YEAR(purchased) )
SUBPARTITION BY HASH( TO_DAYS(purchased) )
SUBPARTITIONS 2 (
    PARTITION p0 VALUES LESS THAN (1990),
    PARTITION p1 VALUES LESS THAN (2000),
    PARTITION p2 VALUES LESS THAN MAXVALUE
);

CREATE TABLE ts1 (id INT, purchased DATE)
PARTITION BY RANGE( YEAR(purchased) )
SUBPARTITION BY HASH( MONTH(purchased) )
SUBPARTITIONS 2 (
    PARTITION p0 VALUES LESS THAN (1990),
    PARTITION p1 VALUES LESS THAN (2000),
    PARTITION p2 VALUES LESS THAN MAXVALUE
);

```

3.6. 分区管理

新增分区

mysql 5.5+

为已经存在表添加分区

```
ALTER TABLE tbl_name ADD PARTITION PARTITIONS 6;
```

新增 RANGE 分区

```
ALTER TABLE category ADD PARTITION (PARTITION p4 VALUES IN
(100,200,300,400)
    DATA DIRECTORY = '/data/category'
    INDEX DIRECTORY = '/data/category');
```

新增 LIST 分区

```
CREATE TABLE expenses (  
    expense_date DATE NOT NULL,  
    category VARCHAR(30),  
    amount DECIMAL (10,3)  
);  
  
ALTER TABLE expenses  
PARTITION BY LIST COLUMNS (category)  
(  
    PARTITION p01 VALUES IN ( 'lodging', 'food'),  
    PARTITION p02 VALUES IN ( 'flights', 'ground  
transportation'),  
    PARTITION p03 VALUES IN ( 'leisure', 'customer  
entertainment'),  
    PARTITION p04 VALUES IN ( 'communications'),  
    PARTITION p05 VALUES IN ( 'fees')  
);
```

新增 HASH分区

```
CREATE TABLE t1 (  
    id INT,  
    year_col INT  
);  
  
ALTER TABLE t1  
    PARTITION BY HASH(id)  
    PARTITIONS 8;
```

```
/* 在MySQL 5.1中*/  
CREATE TABLE t2  
(  
    dt DATE  
)
```

```

PARTITION BY RANGE (TO_DAYS(dt))
(
  PARTITION p01 VALUES LESS THAN (TO_DAYS('2007-01-01')),
  PARTITION p02 VALUES LESS THAN (TO_DAYS('2008-01-01')),
  PARTITION p03 VALUES LESS THAN (TO_DAYS('2009-01-01')),
  PARTITION p04 VALUES LESS THAN (MAXVALUE));

SHOW CREATE TABLE t2 \G
***** 1. row
*****
      Table: t2
Create Table: CREATE TABLE `t2` (
  `dt` date DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1
/*!50100 PARTITION BY RANGE (TO_DAYS(dt))
(PARTITION p01 VALUES LESS THAN (733042) ENGINE = MyISAM,
PARTITION p02 VALUES LESS THAN (733407) ENGINE = MyISAM,
PARTITION p03 VALUES LESS THAN (733773) ENGINE = MyISAM,
PARTITION p04 VALUES LESS THAN MAXVALUE ENGINE = MyISAM) */

/*在MySQL 5.5中*/
CREATE TABLE t2
(
  dt DATE
)
PARTITION BY RANGE COLUMNS (dt)
(
  PARTITION p01 VALUES LESS THAN ('2007-01-01'),
  PARTITION p02 VALUES LESS THAN ('2008-01-01'),
  PARTITION p03 VALUES LESS THAN ('2009-01-01'),
  PARTITION p04 VALUES LESS THAN (MAXVALUE));

SHOW CREATE TABLE t2 \G
***** 1. row
*****
      Table: t2
Create Table: CREATE TABLE `t2` (
  `dt` date DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1
/*!50500 PARTITION BY RANGE COLUMNS(dt)
(PARTITION p01 VALUES LESS THAN ('2007-01-01') ENGINE =
MyISAM,
PARTITION p02 VALUES LESS THAN ('2008-01-01') ENGINE =

```

```
MyISAM,  
PARTITION p03 VALUES LESS THAN ('2009-01-01') ENGINE =  
MyISAM,  
PARTITION p04 VALUES LESS THAN (MAXVALUE) ENGINE = MyISAM)  
*/
```

删除分区

删除分区 p0

```
ALTER TABLE users DROP PARTITION p0;
```

重建分区

使用 REORGANIZE 重建分区。

RANGE 分区重建

```
ALTER TABLE users REORGANIZE PARTITION p0,p1 INTO (PARTITION  
p0 VALUES LESS THAN (6000000));
```

将原来的 p0,p1 分区合并起来，放到新的 p0 分区中。

LIST 分区重建

```
ALTER TABLE users REORGANIZE PARTITION p0,p1 INTO (PARTITION  
p0 VALUES IN(0,1,4,5,8,9,12,13));
```

将原来的 p0,p1 分区合并起来，放到新的 p0 分区中。

HASH/KEY 分区重建

```
ALTER TABLE users REORGANIZE PARTITION COALESCE PARTITION 2;  
分区的数量改为2,
```

注意：在这里数量只能减少不能增加。想要增加可以用 ADD PARTITION 方法

调整HASH/KEY分区数量，将分区总数扩展到8个。

```
ALTER TABLE users ADD PARTITION PARTITIONS 8;
```

分区维护

重建分区：这和先删除保存在分区中的所有记录，然后重新插入它们，具有同样的效果。它可用于整理分区碎片。

示例：

```
ALTER TABLE t1 REBUILD PARTITION (p0, p1);
```

- 优化分区：如果从分区中删除了大量的行，或者对一个带有可变长度的行（也就是说，有VARCHAR，BLOB，或TEXT类型的列）作了许多修改，可以使用“ALTER TABLE ... OPTIMIZE PARTITION”来收回没有使用的空间，并整理分区数据文件的碎片。

示例：

```
ALTER TABLE t1 OPTIMIZE PARTITION (p0, p1);
```

在一个给定的分区表上使用“OPTIMIZE PARTITION”等同于在那个分区上运行CHECK PARTITION，ANALYZE PARTITION，和REPAIR PARTITION。

- 分析分区：读取并保存分区的键分布。

示例：

```
ALTER TABLE t1 ANALYZE PARTITION (p3);
```

- 修补分区： 修补被破坏的分区。

示例：

```
ALTER TABLE t1 REPAIR PARTITION (p0,p1);
```

- 检查分区： 可以使用几乎与对非分区表使用CHECK TABLE 相同的方式检查分区。

示例：

```
ALTER TABLE trb3 CHECK PARTITION (p1);
```

3.7. EXPLAIN PARTITIONS

EXPLAIN PARTITIONS

```
mysql> EXPLAIN PARTITIONS SELECT * FROM users\G
***** 1. row *****
      id: 1
  select_type: SIMPLE
        table: users
  partitions: p0,p1,p2,p3,p4,p5,p6
         type: ALL
possible_keys: NULL
          key: NULL
      key_len: NULL
         ref: NULL
         rows: 7
      Extra:
1 row in set (0.03 sec)

mysql> EXPLAIN PARTITIONS SELECT * FROM users WHERE id < 5\G
***** 1. row *****
      id: 1
  select_type: SIMPLE
        table: users
  partitions: p0,p1,p2,p3,p4,p5,p6
         type: range
possible_keys: PRIMARY
          key: PRIMARY
      key_len: 4
         ref: NULL
         rows: 7
      Extra: Using where
1 row in set (0.00 sec)
```

3.8. SHOW CREATE TABLE

SHOW CREATE TABLE

```

mysql> SHOW CREATE TABLE users\G
***** 1. row *****
      Table: users
Create Table: CREATE TABLE `users` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `username` varchar(20) NOT NULL DEFAULT '',
  `birthday` datetime DEFAULT NULL,
  PRIMARY KEY (`id`,`username`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=latin1
/*!50100 PARTITION BY KEY (id,username)
PARTITIONS 7 */
1 row in set (0.00 sec)

```

3.9. INFORMATION_SCHEMA.partitions 表

```

SELECT
  partition_name part,
  partition_expression expr,
  partition_description descr,
  table_rows
FROM
  INFORMATION_SCHEMA.partitions
WHERE
  TABLE_SCHEMA = schema()
  AND TABLE_NAME='employees';

```

```

select
  partition_name part,
  partition_expression expr,
  from_seconds(partition_description) descr,
  table_rows
FROM

```

```
INFORMATION_SCHEMA.partitions
WHERE
    TABLE_SCHEMA = 'test'
    AND TABLE_NAME='t2';
```

3.10. 分区数据操作

指定分区查询

```
SELECT * FROM employees PARTITION (p0, p2);

SELECT count(1) FROM employees PARTITION (p0);
SELECT count(1) FROM employees PARTITION (p0, p2);
SELECT count(1) FROM employees PARTITION (p0, p2, p1);
```

删除分区中的记录

```
DELETE FROM employees PARTITION (p0, p1);
```

更新指定分区

```
UPDATE employees PARTITION (p0) SET store_id = 2 WHERE fname
= 'Jill';
```

指定分区连表查询

```
SELECT e.id, s.city FROM employees AS e JOIN stores PARTITION
(p1) AS s ...;
```


将某个表迁移到分区上

```
ALTER TABLE employees EXCHANGE PARTITION p0 WITH TABLE  
employees2;
```

4. Index

4.1. SHOW INDEX

```
SHOW INDEX FROM tbl_name
```

垂直显示

```
SHOW INDEX FROM tbl_name\G
```

4.2. CREATE INDEX

```
CREATE INDEX index_name  
ON table_name (column_name)
```

CREATE UNIQUE INDEX

```
CREATE UNIQUE INDEX index_name  
ON table_name (column_name)
```

4.3. DROP INDEX

```
DROP INDEX index_name ON tbl_name
```

4.4. rebuild

```
SHOW INDEX FROM tbl_name  
alter index IND_PK rebuild;
```

5. 外键(Foreign Key)

ON DELETE, ON UPDATE 事件触发限制, 可选参数: RESTRICT | CASCADE | SET NULL | NO ACTION

1. RESTRICT (限制外表中的外键改动)
2. CASCADE (跟随外键改动)
3. SET NULL (设空值)
4. SET DEFAULT (设默认值)
5. NO ACTION (无动作, 默认的)

5.1. FOREIGN KEY (RESTRICT)

```
CREATE TABLE `bank_account_group_has_bank_account` (  
    `id` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,  
    `bank_account_group_id` INT(10) UNSIGNED NOT NULL  
DEFAULT '0',  
    `bank_account_id` INT(10) UNSIGNED NOT NULL DEFAULT  
'0',  
    PRIMARY KEY (`id`),  
    INDEX  
`FK_bank_account_group_has_bank_account_bank_account`  
(`bank_account_id`),  
    INDEX  
`FK_bank_account_group_has_bank_account_bank_account_group`  
(`bank_account_group_id`),  
    CONSTRAINT  
`FK_bank_account_group_has_bank_account_bank_account` FOREIGN  
KEY (`bank_account_id`) REFERENCES `bank_account` (`id`),  
    CONSTRAINT  
`FK_bank_account_group_has_bank_account_bank_account_group`  
FOREIGN KEY (`bank_account_group_id`) REFERENCES  
`bank_account_group` (`id`)  
)
```

```
COMMENT='bank_account_group 与 bank_account 的 N:M 关系'  
COLLATE='utf8_general_ci'  
ENGINE=InnoDB  
AUTO_INCREMENT=35;
```

6. 视图(View)

```
CREATE VIEW view_name AS  
SELECT column_name(s)  
FROM table_name  
WHERE condition
```

update view

```
SQL CREATE OR REPLACE VIEW Syntax  
CREATE OR REPLACE VIEW view_name AS  
SELECT column_name(s)  
FROM table_name  
WHERE condition
```

7. 存储过程(PROCEDURE)

7.1. 存储程序

存储过程没有返回数据，需使用call proc()调用

```
CREATE DEFINER=`neo`@`%` PROCEDURE `angelfund`(IN `puid`
VARCHAR(50), IN `ptime` DATETIME)
LANGUAGE SQL
NOT DETERMINISTIC
CONTAINS SQL
SQL SECURITY DEFINER
COMMENT ''
BEGIN

    DECLARE fusername VARCHAR(16) DEFAULT NULL;
    DECLARE fname VARCHAR(16) DEFAULT NULL;
    DECLARE fmembers_date VARCHAR(20) DEFAULT NULL;

    SELECT username,name,FROM_UNIXTIME(createtime) INTO
    fusername,fname,fmembers_date FROM members WHERE username =
    puid;

    IF fusername IS NOT NULL THEN
        INSERT IGNORE INTO
    angelfund(username,name,members_date,accounts_date,endtime,`st
    atus`,op,operator,`description`)
    value(fusername,fname,fmembers_date,ptime,DATE_ADD(ptime,
    INTERVAL +1 MONTH),'N','N','computer','');
    END IF;

END
```

调用过程

```
call angelfund('100','2013-10-10 10:10:10');
```

7.2. EXECUTE 执行 SQL

在过程中运行SQL，下面的例子是文件导出的例子。

```
DROP procedure IF EXISTS `export_file`;  
  
DELIMITER $$  
CREATE DEFINER=`dba`@`%` PROCEDURE `export_file`(IN file_name  
char(64), IN tabname char(64))  
BEGIN  
    set @sql = concat('SELECT * INTO OUTFILE  
' , "/var/lib/mysql-files/" , file_name , "' FROM ' , tabname) ;  
    -- select @sql;  
    PREPARE stmt FROM @sql;  
    EXECUTE stmt;  
    Deallocate prepare stmt;  
END$$  
  
DELIMITER ;
```

call 存储过程

```
call test.export_file('test', 'mytable');
```

7.3. PREPARE 传递参数

```
mysql> PREPARE stmt1 FROM 'SELECT SQRT(POW(?,2) + POW(?,2)) AS  
hypotenuse';  
Query OK, 0 rows affected (0.00 sec)  
Statement prepared
```



```

mysql> SET @a = 3;
Query OK, 0 rows affected (0.00 sec)

mysql> SET @b = 4;
Query OK, 0 rows affected (0.00 sec)

mysql> EXECUTE stmt1 USING @a, @b;
+-----+
| hypotenuse |
+-----+
|           5 |
+-----+
1 row in set (0.00 sec)

mysql> DEALLOCATE PREPARE stmt1;
Query OK, 0 rows affected (0.00 sec)

mysql>

```

7.4. 存储过程返回数据

```

USE `test`;
DROP procedure IF EXISTS `test`;

DELIMITER $$
USE `test`$$
CREATE DEFINER=`dba`@`%` PROCEDURE `test`(in a int, in b int
,out num int)
BEGIN

    set num = a + b;

END$$

DELIMITER ;

```

运行后返回结果 10

```
set @num = 0;
call test(3,7,@num);
select @num;
```

7.5. 结果集转JSON

```
USE `netkiller`;
DROP procedure IF EXISTS `table2json`;

DELIMITER $$
USE `netkiller`$$
CREATE DEFINER=`neo`@`%` PROCEDURE `table2json`(
IN `schema` VARCHAR(32),
IN `table` VARCHAR(32),
IN `id` VARCHAR(10),
OUT rev VARCHAR(1024)
)
BEGIN
    SET @column = NULL;
    SET @str = NULL;

    SELECT
    GROUP_CONCAT(fields) AS col INTO @column FROM (
        SELECT
        COLUMN_NAME) AS fields
    FROM
    INFORMATION_SCHEMA.Columns
    WHERE
    table_name = `table`
    AND table_schema = `schema`
```

```

AS tmpatable;

    -- SELECT @column;

    SET @sql = CONCAT('SELECT json_object(',@column, ' )
as json INTO @str FROM ', `table`,` where id = ', `id`);

    -- SELECT @sql;

    PREPARE stmt FROM @sql;
    EXECUTE stmt;
    Deallocate prepare stmt;

    set rev = @str;

END$$

DELIMITER ;

```

使用实例

```

set @rev = '0';
call netkiller.table2json('test', 'test', '1', @rev);
select @rev;

```

7.6. 例子·过程返回结果

```

USE `netkiller`;
DROP procedure IF EXISTS `trigger2json`;

DELIMITER $$
USE `netkiller`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `trigger2json`(

```

```

IN `schema` VARCHAR(32),
IN `table` VARCHAR(32),
OUT rev VARCHAR(1024)
)
BEGIN
    SET @column = NULL;
    SET @str = NULL;

    SELECT
    GROUP_CONCAT(fields) AS col
INTO @column FROM
    (SELECT
        CONCAT('"' , COLUMN_NAME, '"', NEW.' , COLUMN_NAME) AS
fields
    FROM
        INFORMATION_SCHEMA.Columns
    WHERE
        table_name = `table`
        AND table_schema = `schema`) AS tmp;

-- SELECT @column;

    SET @sql = CONCAT('SELECT json_object(' , @column, ' )
as json INTO @str ');

    -- SELECT @sql;

    PREPARE stmt FROM @sql;
    EXECUTE stmt;
    Deallocate prepare stmt;

    set rev = @str;

END$$

DELIMITER ;

```

```
set @rev = '0';  
call neo.trigger2json('gw', 'member', @rev);  
select @rev;
```

8. 函数

函数会返回数据，调用函数使用 `select fun()`，不能使用 `call` 调用，否则提示

```
mysql> call myfun();
ERROR 1305 (42000): PROCEDURE test.myfun does not exist
```

下面做一个实验

```
CREATE TABLE `t` (
  `id` INT(11) UNSIGNED NOT NULL AUTO_INCREMENT,
  `n` INT(11) UNSIGNED NULL DEFAULT '0',
  PRIMARY KEY (`id`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
AUTO_INCREMENT=5;

CREATE DEFINER=`neo`@`%` FUNCTION `myfun`()
  RETURNS int(11)
  LANGUAGE SQL
  NOT DETERMINISTIC
  READS SQL DATA
  SQL SECURITY DEFINER
  COMMENT ''
BEGIN
  INSERT INTO t (n) VALUES(rand()*100);
  RETURN LAST_INSERT_ID();
END
```

```
mysql> select myfun();
+-----+
| myfun() |
+-----+
|      9 |
+-----+
1 row in set, 2 warnings (0.07 sec)
```

8.1. TIMESTAMP TO ISO8601

```
USE `netkiller`;
DROP function IF EXISTS `timestamp_to_iso8601`;

DELIMITER $$
USE `netkiller`$$
CREATE DEFINER=`neo`@`db.netkiller.cn` FUNCTION
`timestamp_to_iso8601`(dt timestamp) RETURNS varchar(24) CHARSET utf8
BEGIN

    RETURN DATE_FORMAT( CONVERT_TZ(dt, @@session.time_zone,
'+00:00') , '%Y-%m-%dT%T.000Z' );

END$$

DELIMITER ;
```

调用函数

```
mysql> select timestamp_to_iso8601(current_timestamp()) as iso8601;
+-----+
| iso8601 |
+-----+
| 2017-12-07T07:21:22.000Z |
+-----+
1 row in set (0.00 sec)
```

9. 触发器(Trigger)

9.1. create trigger

Update 更新出发

实现 history 历史表功能，BEFORE update 做到数据库更新自动备份

```
CREATE TABLE user_history SELECT * FROM user WHERE 1 <> 1

DELIMITER //
CREATE TRIGGER user_history BEFORE update ON user FOR EACH ROW
BEGIN
insert into user_history SELECT * FROM user WHERE id = OLD.id;
END; //
DELIMITER ;
```

判断某字段数据修改满足条件后出发。

```
CREATE DEFINER=`dba`@`%` TRIGGER
`cms`.`jc_content_BEFORE_UPDATE` BEFORE UPDATE ON `jc_content`
FOR EACH ROW
BEGIN
    IF NEW.status = '1' THEN
        insert into `neo`.elasticsearch_trash(id)
values(OLD.content_id);
    END IF;
    IF NEW.status = '2' THEN
        delete from `neo`.elasticsearch_trash where id
= OLD.content_id;
    END IF;
END
```


Delete 删除出发

```
CREATE DEFINER=`dba`@`%` TRIGGER
`cms`.`jc_content_BEFORE_DELETE` BEFORE DELETE ON `jc_content`
FOR EACH ROW
BEGIN
    insert into `neo`.elasticsearch_trash(id)
values(OLD.content_id);
END
```

Insert 插入出发

9.2. drop trigger

```
DROP TRIGGER admin_user_history;

DELIMITER //
CREATE TRIGGER admin_user_history BEFORE update ON admin_user
FOR EACH ROW
BEGIN
insert into admin_user_history SELECT * FROM admin_user WHERE
user_id = OLD.user_id;
END; //
DELIMITER;
```

9.3. show triggers

```
show triggers;
```

SHOW CREATE TRIGGER

```
mysql> SHOW CREATE TRIGGER ins_sum\G
***** 1. row *****
      Trigger: ins_sum
      sql_mode:
STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION
SQL Original Statement: CREATE DEFINER=`me`@`localhost` TRIGGER
ins_sum
                        BEFORE INSERT ON account
                        FOR EACH ROW SET @sum = @sum +
NEW.amount
      character_set_client: utf8
      collation_connection: utf8_general_ci
      Database Collation: latin1_swedish_ci
      Created: 2013-07-09 10:39:34.96
```

9.4. EXAMPLE

BEFORE/AFTER

例 61.1. BEFORE/AFTER

```
DROP TRIGGER MY_TEST_MONITOR;
DELIMITER //
CREATE TRIGGER MY_TEST_MONITOR BEFORE insert ON MY_TEST FOR
EACH ROW
BEGIN
    INSERT INTO MY_TEST_MONITOR SELECT * FROM MY_TEST WHERE
TICKET = NEW.TICKET;
END; //
DELIMITER;
```

```

DROP TRIGGER MY_TEST_MONITOR;
DELIMITER //
CREATE TRIGGER MY_TEST_MONITOR AFTER insert ON MY_TEST FOR EACH
ROW
BEGIN
    INSERT INTO MY_TEST_MONITOR SELECT * FROM MY_TEST WHERE
TICKET = NEW.TICKET;
END; //
DELIMITER;

```

通过触发器保护数据，防止重复插入数据

```

CREATE DEFINER=`neo`@`%` TRIGGER `members_before_insert` BEFORE
INSERT ON `members` FOR EACH ROW BEGIN
    IF new.username IS NOT NULL THEN
        IF exists(select m.username from members m
where m.username = new.username) THEN
            set new.username = '';
        END IF;
    END IF;
END

```

UUID

例 61.2. uuid()

```

delimiter $$
CREATE TABLE `member` (
  `uuid` char(36) NOT NULL,
  `username` varchar(20) DEFAULT NULL,
  `password` varchar(32) DEFAULT NULL,
  PRIMARY KEY (`uuid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8$$

CREATE
DEFINER=`root`@`%`

```

```

TRIGGER `test`.`member_before_insert`
BEFORE INSERT ON `test`.`member`
FOR EACH ROW
SET new.uuid = uuid()
$$

```

CALL PROCEDURE

```

CREATE DEFINER=`neo`@`%` TRIGGER `accounts_angelfund` AFTER
INSERT ON `accounts` FOR EACH ROW BEGIN

    IF new.paymode = 'angelfund' THEN
        call angelfund(new.name,new.ctime);
    END IF;

END

CREATE DEFINER=`neo`@`%` PROCEDURE `angelfund`(IN `puid`
VARCHAR(50), IN `ptime` DATETIME)
LANGUAGE SQL
NOT DETERMINISTIC
CONTAINS SQL
SQL SECURITY DEFINER
COMMENT ''
BEGIN

    DECLARE fusername VARCHAR(16) DEFAULT NULL;
    DECLARE fchinese_name VARCHAR(16) DEFAULT NULL;
    DECLARE fmembers_date VARCHAR(20) DEFAULT NULL;

    SELECT username,chinese_name,FROM_UNIXTIME(createtime)
INTO fusername,fchinese_name,fmembers_date FROM members WHERE
username = puid;

    IF fusername IS NOT NULL THEN
        INSERT IGNORE INTO
angelfund(username,chinese_name,members_date,accounts_date,endt
ime,`status`,op,operator,`description`)
value(fusername,fchinese_name,fmembers_date,ptime,DATE_ADD(ptim
e, INTERVAL +1 MONTH),'N','N','computer','');
    END IF;

```

END

10. 事件调度器(EVENT)

10.1. 启用 EVENT

```
set GLOBAL event_scheduler=ON;
```

my.cnf 配置

```
event_scheduler=on
```

查看状态

```
mysql> select @@GLOBAL.event_scheduler;
+-----+
| @@GLOBAL.event_scheduler |
+-----+
| ON                        |
+-----+
1 row in set (0.00 sec)

mysql> SHOW VARIABLES LIKE 'event_scheduler';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| event_scheduler | ON    |
+-----+-----+
1 row in set (0.01 sec)
```



```

-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| netkiller | captcha      | neo@%   | SYSTEM   | RECURRING |
NULL       | 5            | MINUTE  |          | 2013-07-08
16:27:03  | NULL | ENABLED |          | 1 | utf8
| utf8_general_ci | utf8_general_ci |
| netkiller | sms_ips_log | neo@%   | SYSTEM   | RECURRING |
NULL       | '0 5'      | DAY_HOUR |          | 2013-07-09
14:39:51  | NULL | ENABLED |          | 1 | utf8
| utf8_general_ci | utf8_general_ci |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

```

```

mysql> show events \G;
***** 1. row
*****
          Db: netkiller
          Name: captcha
          Definer: neo@%
          Time zone: SYSTEM
          Type: RECURRING
          Execute at: NULL
          Interval value: 5
          Interval field: MINUTE
          Starts: 2013-07-08 16:27:03
          Ends: NULL
          Status: ENABLED
          Originator: 1
character_set_client: utf8
collation_connection: utf8_general_ci
  Database Collation: utf8_general_ci
***** 2. row
*****
          Db: netkiller
          Name: sms_ips_log
          Definer: neo@%
          Time zone: SYSTEM
          Type: RECURRING
          Execute at: NULL
          Interval value: '0 5'
          Interval field: DAY_HOUR

```



```
        Starts: 2013-07-09 14:39:51
          Ends: NULL
        Status: ENABLED
      Originator: 1
character_set_client: utf8
collation_connection: utf8_general_ci
  Database Collation: utf8_general_ci
2 rows in set (0.00 sec)

ERROR:
No query specified
```

10.5. 实例·每月创建一个表

每月创建一张新表，适用于分表的场景

```
CREATE DEFINER=`neo`@`netkiller` EVENT `logging`
  ON SCHEDULE
    EVERY 1 MONTH STARTS '2017-12-11 15:51:00'
  ON COMPLETION PRESERVE
  ENABLE
  COMMENT '每月自动创建表'
DO BEGIN
  declare _table_date varchar(10);
  select date_format(date_add(curdate(),interval 1
month),'%Y%m') into _table_date;
  call logging(_table_date);
END
```

```
CREATE DEFINER=`neo`@`netkiller` PROCEDURE `logging`(
  IN `table_date` VARCHAR(10)
)
LANGUAGE SQL
NOT DETERMINISTIC
```

```

CONTAINS SQL
SQL SECURITY DEFINER
COMMENT ''
BEGIN
    set @_table_name = CONCAT('log_',table_date);
    set @_create = "CREATE TABLE If Not Exists ";
    set @_param = "(
        `id` INT(11) NOT NULL AUTO_INCREMENT,
        `type` VARCHAR(255) NULL DEFAULT NULL
COMMENT '日志类型 1: 网站 2: IOS 3:Android',
        `url` VARCHAR(640) NULL DEFAULT NULL
COMMENT '用户访问url',
        `serverIp` VARCHAR(255) NULL DEFAULT
NULL COMMENT '服务器ip',
        `bodyBytesSent` VARCHAR(255) NULL
DEFAULT NULL,
        `bytesSent` VARCHAR(255) NULL DEFAULT
NULL COMMENT '参数字节数',
        `browser` VARCHAR(255) NULL DEFAULT
NULL COMMENT '浏览器信息',
        `ctime` TIMESTAMP NULL DEFAULT
CURRENT_TIMESTAMP,
        `mtime` TIMESTAMP NULL DEFAULT NULL
ON UPDATE CURRENT_TIMESTAMP,
        PRIMARY KEY (`id`),
        INDEX `ctime` (`ctime`,
`deviceType`,`isFirst`),
        INDEX `userIp` (`userIp`),
        INDEX `deviceId` (`deviceId`),
        INDEX `account` (`account`)
    )
    COMMENT='APP 访问记录'
    COLLATE='utf8_general_ci'
    ENGINE=InnoDB
    ;";

    SET @sql = CONCAT(@_create,@_table_name,@_param);
    PREPARE stmt FROM @sql;
    EXECUTE stmt;
    Deallocate prepare stmt;
END

```

第 62 章 DML (Data Manipulation Language)

```
SELECT - retrieve data from the a database
INSERT - insert data into a table
UPDATE - updates existing data within a table
DELETE - deletes all records from a table, the space for the
records remain
CALL - call a PL/SQL or Java subprogram
EXPLAIN PLAN - explain access path to data
LOCK TABLE - control concurrency
```

1. INSERT

1.1. INSERT INTO ... SELECT

```
SET @OLDTMP_SQL_MODE=@@SQL_MODE, SQL_MODE='';
DELIMITER //
CREATE TRIGGER `members_mobile_insert` BEFORE INSERT ON
`members_mobile` FOR EACH ROW BEGIN
    insert into members_location(id,province,city) select
NEW.id,mobile_location.province,mobile_location.city from
mobile_location where mobile_location.id = md5(LEFT(NEW.number,
7));
END//
DELIMITER ;
SET SQL_MODE=@OLDTMP_SQL_MODE;
```

1.2. INSERT IGNORE

INSERT IGNORE 与INSERT INTO的区别就是INSERT IGNORE会忽略数据库中已经存在的数据，如果数据库没有数据，就插入新的数据，如果有数据的话就跳过这条数据。

```
insert ignore into table(name) select name from table2
```

1.3. INSERT...ON DUPLICATE KEY UPDATE

```
create table foo (id serial primary key, u int, unique key
(u));

insert into foo (u) values (10);
insert into foo (u) values (10) on duplicate key update u = 20;

mysql> select * from foo;
+----+-----+
| id | u     |
+----+-----+
|  1 |  20  |
+----+-----+
```

```
DROP TRIGGER IF EXISTS `cms`.`jc_content_BEFORE_DELETE`;

DELIMITER $$
USE `cms`$$
CREATE DEFINER=`5kwords`@`%` TRIGGER `jc_content_BEFORE_DELETE`
BEFORE DELETE ON `jc_content`
FOR EACH ROW BEGIN

    insert into `cms`.elasticsearch_trash(id)
values(OLD.content_id) on duplicate key update ctime = now();
    insert into `cms`.trash(id,`type`, site_id)
values(OLD.content_id, "delete", OLD.site_id) on duplicate key
update `type`="delete", ctime = now();

END$$
DELIMITER ;
```

2. REPLACE

replace 类似 ON DUPLICATE KEY UPDATE，插入过程遇到已经存在的字段，会更新处理。

```
replace into (id) value('1')
```

3. DELETE

3.1. 删除重复数据

```
delete from member group by username having count(username) > 1
```

第 63 章 SQL Statement Syntax

Structured Query Language

1. DISTINCT

```
SELECT DISTINCT user.name FROM user
```

```
SELECT DISTINCT user.name FROM user
```

2. group by

统计重复的手机号吗

```
select * from (select count(mobile) as c, mobile from member
where length(mobile) >= 11 group by mobile) as m where m.c > 1;
```


3. HAVING

```
select * from accounts where paymode='alipay' group by name  
having count(name) >1;
```

4. REGEXP

正则匹配

判断非数字字符

```
select '看89700' regexp '^[0-9]+$'  
select '89看700' regexp '^[0-9]+$'  
select '89700看' regexp '^[0-9]+$'
```

应用到实际工作中

```
select count(*) from accounts a where a.name != '' and not  
a.name regexp '^[0-9]+$';  
select count(*) from accounts a, members m where a.member =  
m.id and a.name != '' and not a.name regexp '^[0-9]+$'  
group by member;  
SELECT * FROM tablename WHERE SUBSTRING(fieldname, 1, 1)  
REGEXP '[[:digit:]]';
```

5. IN / NOT IN

```
select * from members where id in ('1','100','1000');  
select * from members where group_id in (select id from  
members_group);
```

6. ALL / Any

NOT IN 与 \neq ALL 两个语句是相同的:

```
SELECT s1 FROM t1 WHERE s1 <> ALL (SELECT s1 FROM t2);  
SELECT s1 FROM t1 WHERE s1 NOT IN (SELECT s1 FROM t2);
```

IN 与 "= ANY" 两个语句是一样的:

```
SELECT s1 FROM t1 WHERE s1 = ANY (SELECT s1 FROM t2);  
SELECT s1 FROM t1 WHERE s1 IN (SELECT s1 FROM t2);
```

例 63.1. SQL ANY example

```
select * from members where id = any(select members_id from  
accounts where id < 100);
```

7. exists, not exists

```
SELECT c.id, companyname
FROM customers c
WHERE EXISTS(
    SELECT orderid FROM orders o WHERE o.customer_id = cu.id)
```

8. UNION

union 分页问题

```
(SELECT a FROM tbl_name_a WHERE a=10 AND B=1)
UNION
(SELECT a FROM tbl_name_b WHERE a=11 AND B=2)
ORDER BY a LIMIT 10;
```

```
select * from (
    select a from tbl_name_a WHERE a=10 AND B=1
    union all
    select a from tbl_name_b WHERE a=10 AND B=1
) tbl_name
order by a limit 0,1;
```

8.1. UNION ALL

UNION ALL 不会合并重复的记录

```
select a,b from tbl_name_a WHERE a=10 AND B=1
union all
select a,b from tbl_name_b WHERE a=10 AND B=1
```

8.2. 两张表字段不对等解决方法

```
SELECT * FROM
(
    SELECT contract_address, decimals, name, symbol, seq,
```

```
logo FROM token
UNION
SELECT contract_address, decimals, name, symbol, 100,
'https://www.netkiller.cn/images/eth.jpg' FROM user_token
WHERE address = '0xB94054c174995AE2A9E7fcf6c7924635FBa8ECF7'
AND contract_address NOT IN (SELECT contract_address FROM
token)
) AS tmp
ORDER BY seq
```

9. OUTFILE/LOAD DATA INFILE

查询结果输出到文件

```
SELECT * FROM tablename INTO OUTFILE '/tmp/tablename.txt';
```

使用tee将屏幕输出到文件

```
mysql>tee /home/neo/screen.txt  
mysql>select * from user;  
mysql>exit
```

```
SELECT * INTO OUTFILE '/home/mark/Orders.txt'  
  FIELDS  
  TERMINATED BY = ','  
  FROM Orders  
  WHERE Order_Date >= '2000-01-01'
```

```
LOAD DATA INFILE 'data.txt' INTO TABLE db2.my_table;
```

9.1. Export data to CSV from MySQL

```
SELECT *  
INTO OUTFILE '/tmp/products.csv'  
FIELDS TERMINATED BY ','  
ENCLOSED BY '"'
```



```
ESCAPED BY '\\\  
LINES TERMINATED BY '\\n'  
FROM products
```

9.2. Import data from CSV file.

```
LOAD DATA LOW_PRIORITY LOCAL INFILE 'C:\\hx.csv' IGNORE INTO  
TABLE `tmp`.`creditlog`  
CHARACTER SET gbk FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED  
BY '"' ESCAPED BY '"' LINES TERMINATED BY '\\r\\n'  
(`ctime`, `login`, `mode`, `type`, `prevavailcredit`, `change`,  
`newavailcredit`, `comment`);
```

10. CASE Syntax

```
CASE case_value
  WHEN when_value THEN statement_list
  [WHEN when_value THEN statement_list] ...
  [ELSE statement_list]
END CASE
```

Or:

```
CASE
  WHEN search_condition THEN statement_list
  [WHEN search_condition THEN statement_list] ...
  [ELSE statement_list]
END CASE
```

11. MySQL 专有命令

11.1. SQL_NO_CACHE

```
SELECT /*!40001 SQL_NO_CACHE */ * FROM table
```

11.2. SIGNAL Syntax

```
DROP TRIGGER `members_before_insert`;  
CREATE DEFINER=`neo`@`%` TRIGGER `members_before_insert`  
BEFORE INSERT ON `members` FOR EACH ROW BEGIN  
    IF new.username IS NOT NULL THEN  
        IF not exists(select username from  
members_available where username = new.username) THEN  
            /*set new.username = NULL;*/  
            SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT =  
'An error occurred', MYSQL_ERRNO = 1001;  
        END IF;  
    END IF;  
END;
```

12. SQL 92

insert + select

```
insert into product_type_commission select id,5,1,1,0,0,0,0,0,0
from product_type where title='notebook' and is_physical=0;
```

update table1,table2

```
begin;
ALTER TABLE `customer` ADD COLUMN `cutoff_time` TIMESTAMP NOT
NULL default '0000-00-00 00:00:00';
update customer,agent set customer.cutoff_time =
agent.cutoff_time where customer.id = agent.id;
ALTER TABLE `agent` DROP COLUMN `cutoff_time`;
commit;
```

update table1 set field1 = (select value from table2)

```
UPDATE
    transaction
SET
    transaction.total_sold_price = (
        SELECT
            SUM(transaction_item.price)
        FROM
            transaction_item
            WHERE transaction_item.transaction_id = 100
    )
WHERE
    transaction.id = 100
```

update table1, (select * from other) as table2 set table1.field1 =
table2.field1

```

UPDATE
    transaction,(    SELECT
SUM(product_item.bought_price) AS total_bought_price,
transaction_item.transaction_id
                                FROM
transaction_item
                                WHERE
transaction_item.transaction_id IN ( '123','456' )
                                ) as total
SET
    transaction.total_bought_price =
total.total_bought_price
WHERE
    transaction.id = total.transaction_id

```

join + subquery

```

select u.*,t.category,t.items,t.[property] from
tb_sysregchkusers as u left join (select a.items as category,
b.* from (select id, items from tb_sysregchktask where
categoryid=0) as a left join tb_sysregchktask as b on
b.categoryid=a.id ) as t on u.taskID=t.id

select * from tb_sysregchklog where
CONVERT(datetime,CONVERT(varchar(10),checkTime,120)) between
convert(datetime,'2007-12-12') and convert(datetime,'2007-12-
12')

```

```

select DISTINCT user_point_history.user_id,user.username,
(select count(id) from transaction where id =
user_point_history.transaction_id) as transactions,
(SELECT SUM(u_p_h.points) FROM user_point_history as u_p_h
WHERE u_p_h.type != 'RDMP' AND u_p_h.status IN('pr','ac') AND
u_p_h.user_id = user_point_history.user_id) as
total_points_earned,
(SELECT SUM(u_p_h.points) FROM user_point_history as u_p_h

```

```
WHERE u_p_h.type = 'RDMP' AND u_p_h.status IN('pr','ac') AND
u_p_h.user_id = user_point_history.user_id) as
total_points_redeemed
from user_point_history,user where user_point_history.user_id =
user.id;
```

(total_points_earned - total_points_redeemed) as
current_balance_points

```
select user_id, username, transactions, total_points_earned,
total_points_redeemed, (total_points_earned -
total_points_redeemed) as current_balance_points
from (select DISTINCT user_point_history.user_id,user.username,
(select count(id) from transaction where id =
user_point_history.transaction_id) as transactions,
(SELECT SUM(u_p_h.points) FROM user_point_history as u_p_h
WHERE u_p_h.type != 'RDMP' AND u_p_h.status IN('pr','ac') AND
u_p_h.user_id = user_point_history.user_id) as
total_points_earned,
(SELECT SUM(u_p_h.points) FROM user_point_history as u_p_h
WHERE u_p_h.type = 'RDMP' AND u_p_h.status IN('pr','ac') AND
u_p_h.user_id = user_point_history.user_id) as
total_points_redeemed
from user_point_history,user where user_point_history.user_id =
user.id) as user_performance;
```

subquery作为一个字段使用

```
select product_type_attribute.*, (select 'selected' from
product_type_attribute_set where
product_type_attribute_set.product_type_attribute_id =
product_type_attribute.id and
product_type_attribute_set.product_type_id = 26) as selected
from product_type_attribute;
```

第 64 章 Functions and Operators

1. COUNT

count()

```
SELECT (SELECT count(1) FROM ecs_category) as 'Export category  
count',  
       (SELECT count(1) FROM ecs_goods) as 'Goods count',  
       (SELECT count(1) FROM ecs_goods_attr) as 'Attr count';
```

2. group_concat() 列传行

```
SELECT tags FROM neo.article;  
  
linux  
redis  
mysql  
java  
php
```

tags字段专为一行显示

```
SELECT group_concat(tags) FROM neo.article;  
  
linux,redis,mysql,java,php
```

distinct 去除重复数据

```
select group_concat(distinct author) from neo.article;
```

以id分组，把name字段的值打印在一行，分号分隔

```
select id,group_concat(tags separator ';') from neo.article  
group by tags;
```

排序结果

```
select group_concat(distinct author order by author desc) from  
neo.article;
```


3. UUID()

```
SELECT UUID(),LENGTH(UUID()),UUID_SHORT(),  
LENGTH(UUID_SHORT());
```

4. String

4.1. LEFT/RIGHT

LEFT(str,len)

```
mysql> select left(concat('1','0000000'),5) as number;
+-----+
| number |
+-----+
| 10000  |
+-----+
1 row in set (0.00 sec)
```

RIGHT(str,len)

```
mysql> select right(concat('0000000','1'),5) as number;
+-----+
| number |
+-----+
| 00001  |
+-----+
1 row in set (0.00 sec)
```

4.2. RPAD/LPAD

补齐长度用'0'填充

RPAD(str,len,padstr)

```
mysql> select rpad('10',5,'0') as txt;
+-----+
| txt   |
+-----+
| 10000 |
+-----+
1 row in set (0.01 sec)
```

LPAD(str,len,padstr)

```
mysql> select lpad('10',5,'0') as txt;
+-----+
| txt   |
+-----+
| 00010 |
```

```
+-----+  
1 row in set (0.00 sec)
```

4.3. CONCAT

CONCAT(str1,str2,...)

```
mysql> select concat('Neo',' ','Chen') as Name;  
+-----+  
| Name      |  
+-----+  
| Neo Chen  |  
+-----+  
1 row in set (0.00 sec)
```

4.4. CONCAT_WS

```
SELECT CONCAT_WS(',', 'Neo', 'Chen');  
Neo,Chen  
  
SELECT CONCAT_WS('-', 'Neo', 'Chen');  
Neo-Chen
```

使用逗号链接字符串

```
SELECT  
    CONCAT_WS(',', id, name, age)  
FROM  
    mytable
```

4.5. 链接所有字段

当我使用 `select CONCAT_WS(" ", *) as string from tab` 时发现不支持 * 操作。

解决方案如下

```
SET @column = NULL;
```

```

SELECT
    GROUP_CONCAT(COLUMN_NAME) AS fields INTO @column
FROM
    INFORMATION_SCHEMA.Columns
WHERE
    table_name = 'mytable'
    AND table_schema = 'test';

-- select @column;

SET @sql = CONCAT('SELECT CONCAT_WS(",","',@column, ' ') FROM mytable');

select @sql;

PREPARE stmt FROM @sql;
EXECUTE stmt;
DEALLOCATE PREPARE stmt;

```

4.6. GROUP_CONCAT

```

mysql> select GROUP_CONCAT(CONVERT( username , CHAR (16)) order by username
desc) as username from test;
+-----+
| username |
+-----+
| jam,jam2,john,john2,john3,neo,neo1,neo2 |
+-----+
6 rows in set, 1 warning (0.01 sec)

```

4.7. replace

```

select replace(goods_desc,':8000','') from ecs_goods;

update ecs_goods set goods_desc=replace(goods_desc,':8000','');

```

4.8. SUBSTRING

```

mysql> SELECT SUBSTRING('netkiller',4,4);
+-----+
| SUBSTRING('netkiller',4,4) |
+-----+
| kill |
+-----+
1 row in set (0.00 sec)

```

与left,right 相同的用法

```
select right('M2014030615410572307:DEPOSIT', 7);  
SELECT SUBSTRING('M2014030615410572307:DEPOSIT', -7);
```

4.9. SUBSTRING_INDEX

```
SELECT SUBSTRING_INDEX('M2014030615410572307:DEPOSIT', ':', -1);  
SELECT SUBSTRING_INDEX('M2014030615410572307:DEPOSIT', ':', 1);
```

4.10. AES_ENCRYPT / AES_DECRYPT

简单用法

```
mysql> select AES_ENCRYPT('helloworld','key');  
+-----+  
| AES_ENCRYPT('helloworld','key') |  
+-----+  
|                                  |  
+-----+  
1 row in set (0.00 sec)  
  
mysql> select AES_DECRYPT(AES_ENCRYPT('helloworld','key'),'key');  
+-----+  
| AES_DECRYPT(AES_ENCRYPT('helloworld','key'),'key') |  
+-----+  
| helloworld |  
+-----+  
1 row in set (0.00 sec)  
  
mysql>
```

加密数据入库

```
CREATE TABLE `encryption` (  
  `mobile` VARBINARY(16) NOT NULL,  
  `key` VARCHAR(32) NOT NULL  
)  
ENGINE=InnoDB;  
  
INSERT INTO encryption(`mobile`,`key`)VALUES(
```

```
AES_ENCRYPT('13691851789',md5('13691851789')), md5('13691851789'))
select AES_DECRYPT(mobile,`key`), length(mobile) from encryption;
```

5. Date and Time

```
SELECT NOW(),CURRENT_TIMESTAMP(),SYSDATE();
```

5.1. year/month/day hour:minute:second

```
mysql> select year('2012-03-20');
```

```
+-----+  
| year('2012-03-20') |  
+-----+  
|                2012 |  
+-----+  
1 row in set (0.00 sec)
```

```
mysql> select month('2012-03-20');
```

```
+-----+  
| month('2012-03-20') |  
+-----+  
|                    3 |  
+-----+  
1 row in set (0.00 sec)
```

```
mysql> select day('2012-03-20');
```

```
+-----+  
| day('2012-03-20') |  
+-----+  
|                    20 |  
+-----+  
1 row in set (0.00 sec)
```

```
mysql> select hour('12:30:55');
```

```
+-----+  
| hour('12:30:55') |  
+-----+  
|                    12 |  
+-----+  
1 row in set (0.00 sec)
```

```
mysql> select minute('12:30:55');
+-----+
| minute('12:30:55') |
+-----+
|                    30 |
+-----+
1 row in set (0.00 sec)

mysql> select second('12:30:55');
+-----+
| second('12:30:55') |
+-----+
|                    55 |
+-----+
1 row in set (0.00 sec)
```

5.2. Unix time

语法: FROM_UNIXTIME(unix_timestamp,format)

返回表示 Unix 时间标记的一个字符串，根据format字符串格式化。format可以包含与DATE_FORMAT()函数列出的条目同样的修饰符。根据format字符串格式化date值。

下列修饰符可以被用在format字符串中：

%M 月名字(January.....December)

%W 星期名字(Sunday.....Saturday)

%D 有英语前缀的月份的日期(1st, 2nd, 3rd, 等等。)

%Y 年, 数字, 4 位

%y 年, 数字, 2 位

%a 缩写的星期名字(Sun.....Sat)

%d 月份中的天数, 数字(00.....31)

%e 月份中的天数, 数字(0.....31)

%m 月, 数字(01.....12)

%c 月, 数字(1.....12)

%b 缩写的月份名字(Jan.....Dec)

%j 一年中的天数(001.....366)
 %H 小时(00.....23)
 %k 小时(0.....23)
 %h 小时(01.....12)
 %I 小时(01.....12)
 %l 小时(1.....12)
 %i 分钟, 数字(00.....59)
 %r 时间,12 小时(hh:mm:ss [AP]M)
 %T 时间,24 小时(hh:mm:ss)
 %S 秒(00.....59)
 %s 秒(00.....59)
 %p AM或PM
 %w 一个星期中的天数(0=Sunday6=Saturday)
 %U 星期(0.....52), 这里星期天是星期的第一天
 %u 星期(0.....52), 这里星期一是星期的第一天
 %% 一个文字“%”。

```

mysql> SELECT UNIX_TIMESTAMP('2005-03-27 02:00:00');
+-----+
| UNIX_TIMESTAMP('2005-03-27 02:00:00') |
+-----+
|                               1111885200 |
+-----+
mysql> SELECT FROM_UNIXTIME(1111885200);
+-----+
| FROM_UNIXTIME(1111885200) |
+-----+
| 2005-03-27 03:00:00      |
+-----+
  
```

```

SELECT UNIX_TIMESTAMP('2012-01-01 00:00:00');
SELECT UNIX_TIMESTAMP('2012-07-30 00:00:00');
SELECT UNIX_TIMESTAMP();
  
```

```

SELECT UNIX_TIMESTAMP('2009-08-06') ;
SELECT UNIX_TIMESTAMP( curdate( ) );

select FROM_UNIXTIME(UNIX_TIMESTAMP('2012-07-30 00:00:00'),
'%Y-%m-%d');
SELECT FROM_UNIXTIME( 1249488000, '%Y年%m月%d日' );

SELECT FROM_UNIXTIME(time_stamp, '%Y-%m-%d %H:%i:%S') FROM
test.transaction_history;

select FROM_UNIXTIME(createtime, '%m') as month, count(1) as
count from members where createtime BETWEEN
UNIX_TIMESTAMP('2012-01-01 00:00:00') and
UNIX_TIMESTAMP('2012-12-31 00:00:00') group by
FROM_UNIXTIME(createtime, '%m');
select FROM_UNIXTIME(createtime, '%m') as month, count(1) as
count from members where createtime BETWEEN
UNIX_TIMESTAMP('2011-01-01 00:00:00') and
UNIX_TIMESTAMP('2011-12-31 00:00:00') group by
FROM_UNIXTIME(createtime, '%m');

select FROM_UNIXTIME(createtime, '%m-%d') as month, count(1)
as count from members where createtime BETWEEN
UNIX_TIMESTAMP('2011-01-01 00:00:00') and
UNIX_TIMESTAMP('2011-12-31 00:00:00') group by
FROM_UNIXTIME(createtime, '%m-%d');
select FROM_UNIXTIME(createtime, '%m-%d') as month, count(1)
as count from members where createtime BETWEEN
UNIX_TIMESTAMP('2012-01-01 00:00:00') and
UNIX_TIMESTAMP('2012-12-31 00:00:00') group by
FROM_UNIXTIME(createtime, '%m-%d');

```

5.3. DATE_FORMAT

DATE_FORMAT() 函数用于以不同的格式显示日期/时间数据。

语法

DATE_FORMAT(date, format)

date 参数是合法的日期。format 规定日期/时间的输出格式。

可以使用的格式有：

格式	描述
%a	缩写星期名
%b	缩写月名
%c	月, 数值
%D	带有英文前缀的月中的天
%d	月的天, 数值(00-31)
%e	月的天, 数值(0-31)
%f	微秒
%H	小时 (00-23)
%h	小时 (01-12)
%I	小时 (01-12)
%i	分钟, 数值(00-59)
%j	年的天 (001-366)
%k	小时 (0-23)
%l	小时 (1-12)
%M	月名
%m	月, 数值(00-12)
%p	AM 或 PM
%r	时间, 12-小时 (hh:mm:ss AM 或 PM)
%S	秒(00-59)
%s	秒(00-59)
%T	时间, 24-小时 (hh:mm:ss)
%U	周 (00-53) 星期日是一周的第一天
%u	周 (00-53) 星期一是一周的第一天
%V	周 (01-53) 星期日是一周的第一天, 与 %X 使用
%v	周 (01-53) 星期一是一周的第一天, 与 %x 使用
%W	星期名
%w	周的天 (0=星期日, 6=星期六)
%X	年, 其中的星期日是周的第一天, 4 位, 与 %V 使用
%x	年, 其中的星期一是周的第一天, 4 位, 与 %v 使用
%Y	年, 4 位
%y	年, 2 位

实例

下面的脚本使用 `DATE_FORMAT()` 函数来显示不同的格式。我们使用 `NOW()` 来获得当前的日期/时间：

```
DATE_FORMAT(NOW(), '%b %d %Y %h:%i %p')
DATE_FORMAT(NOW(), '%m-%d-%Y')
DATE_FORMAT(NOW(), '%d %b %y')
DATE_FORMAT(NOW(), '%d %b %Y %T:%f')
```

```
SELECT DATE_FORMAT(NOW(), '%Y-%m-%d');
```

```
select DATE_FORMAT(asctime, '%Y-%m-%d') as Date, count(1) as  
Count from logging where tag='www' and facility='login' group  
by DATE_FORMAT(asctime, '%Y-%m-%d') order by asctime desc;
```

5.4. DATE_SUB/DATE_ADD

当前时间向后推10天

```
mysql> select DATE_SUB(now(), INTERVAL 240 HOUR);
```

```
+-----+  
| DATE_SUB(now(), INTERVAL 240 HOUR) |  
+-----+  
| 2012-03-09 10:26:03 |  
+-----+
```

```
1 row in set (0.00 sec)
```

```
mysql> select DATE_SUB(now(), INTERVAL 24 HOUR);
```

```
+-----+  
| DATE_SUB(now(), INTERVAL 24 HOUR) |  
+-----+  
| 2012-03-18 10:28:43 |  
+-----+
```

```
1 row in set (0.00 sec)
```

```
DELETE from Message where created < DATE_sub(now(), INTERVAL  
240 HOUR);
```

```
select * from PRICES_HISTORY where DATE_FORMAT(TIME  
, GET_FORMAT(DATE, 'ISO')) = (  
select if ( WEEKDAY(CURRENT_DATE())=6 ,  
DATE_SUB(CURRENT_DATE(), INTERVAL 1 DAY) , CURRENT_DATE()  
)
```

DATE_ADD

```
SELECT DATE_ADD('1998-01-02', INTERVAL 31 DAY);
```

5.5. datediff / timediff

计算时间差，两个时间相减结果

```
mysql> select timediff('22:20:00','17:30:00');
```

```
+-----+
| timediff('22:20:00','17:30:00') |
+-----+
| 04:50:00 |
+-----+
```

```
1 row in set (0.00 sec)
```

```
mysql> select datediff('2008-08-08 12:00:00', '2008-08-01
00:00:00');
```

```
+-----+
| datediff('2008-08-08 12:00:00', '2008-08-01 00:00:00') |
+-----+
| 7 |
+-----+
```

```
1 row in set (0.00 sec)
```

6. 数值函数

6.1. cast 类型转换

```
mysql> SELECT cast(SUBSTRING('123456789',1,4) as UNSIGNED) *
100;
+-----+
| cast(SUBSTRING('123456789',1,4) as UNSIGNED) * 100 |
+-----+
|                                                    123400 |
+-----+
1 row in set (0.00 sec)
```

6.2. truncate 保留小数位数

```
select profit, deficit, concat(truncate((profit /
deficit)*100,2),'%') as percentage from ((select count(*) as
profit from angelfund where profit > 0) as profit, (select
count(*) as deficit from angelfund where profit < 0) as
deficit);
```

6.3. MOD 求余

```
mysql> select 9 mod 5;
+-----+
| 9 mod 5 |
+-----+
|         4 |
+-----+
```

```
1 row in set (0.00 sec)
```

```
mysql> select mod(5,2);
```

```
+-----+
```

```
| mod(5,2) |
```

```
+-----+
```

```
|          1 |
```

```
+-----+
```

```
1 row in set (0.00 sec)
```

```
mysql> select mod(5,2);
```

7. Control Flow Functions

CASE

```
mysql> SELECT CASE 1 WHEN 1 THEN 'one'
      ->      WHEN 2 THEN 'two' ELSE 'more' END;
      -> 'one'
mysql> SELECT CASE WHEN 1>0 THEN 'true' ELSE 'false' END;
      -> 'true'
mysql> SELECT CASE BINARY 'B'
      ->      WHEN 'a' THEN 1 WHEN 'b' THEN 2 END;
      -> NULL
```

IFNULL

```
mysql> SELECT IFNULL("TEST", 'OK');
+-----+
| IFNULL("TEST", 'OK') |
+-----+
| TEST                  |
+-----+
1 row in set (0.00 sec)

mysql> SELECT IFNULL(NULL, 'OK');
+-----+
| IFNULL(NULL, 'OK')   |
+-----+
| OK                   |
+-----+
1 row in set (0.00 sec)
```

NULLIF()

IF

```
mysql> SELECT IFNULL("TEST", 'OK');
```

```
+-----+  
| IFNULL("TEST", 'OK') |  
+-----+  
| TEST                  |  
+-----+
```

```
1 row in set (0.00 sec)
```

```
mysql> SELECT IFNULL(NULL, 'OK');
```

```
+-----+  
| IFNULL(NULL, 'OK') |  
+-----+  
| OK                  |  
+-----+
```

```
1 row in set (0.00 sec)
```

第 65 章 PostgreSQL GUI

1. pgAdmin III

<http://www.pgadmin.org/>

2. phpPgAdmin

homepage: <http://phppgadmin.sourceforge.net/>

```
$ wget
http://nchc.dl.sourceforge.net/sourceforge/phppgadmin/phpPgAdmin-4.2.2.tar.bz2
$ tar jxvf phpPgAdmin-4.2.2.tar.bz2
$ ln -s phpPgAdmin-4.2.2 phpPgAdmin
```

3. Monitoring

3.1. PgBadger

<http://projects.dalibo.org/pgbadger/example.html>



4. pgModeler - PostgreSQL Database Modeler

<http://pgmodeler.com.br/>

第 66 章 Barman Backup & recovery for PostgreSQL

<http://www.pgbarman.org/>

第 67 章 pgbouncer - lightweight connection pooler for PostgreSQL

<http://pgfoundry.org/projects/pgbouncer/>

1. 安装 pgbouncer

1.1. Ubuntu

Ubuntu 13.04

```
$ apt-cache search pgbouncer
pgbouncer - lightweight connection pooler for PostgreSQL

$ sudo apt-get install pgbouncer
```

```
$ dpkg -L pgbouncer
/.
/usr
/usr/share
/usr/share/man
/usr/share/man/man5
/usr/share/man/man5/pgbouncer.5.gz
/usr/share/man/man1
/usr/share/man/man1/pgbouncer.1.gz
/usr/share/doc
/usr/share/doc/pgbouncer
/usr/share/doc/pgbouncer/config.html
/usr/share/doc/pgbouncer/README
/usr/share/doc/pgbouncer/README.Debian
/usr/share/doc/pgbouncer/README.html
/usr/share/doc/pgbouncer/faq.html
/usr/share/doc/pgbouncer/copyright
/usr/share/doc/pgbouncer/todo.html
/usr/share/doc/pgbouncer/examples
/usr/share/doc/pgbouncer/examples/pgbouncer.ini.gz
/usr/share/doc/pgbouncer/examples/userlist.txt
```

```
/usr/share/doc/pgbouncer/NEWS.gz
/usr/share/doc/pgbouncer/AUTHORS
/usr/share/doc/pgbouncer/usage.html
/usr/share/doc/pgbouncer/changelog.Debian.gz
/usr/sbin
/usr/sbin/pgbouncer
/etc
/etc/init.d
/etc/init.d/pgbouncer
/etc/pgbouncer
/etc/pgbouncer/userlist.txt
/etc/pgbouncer/pgbouncer.ini
/etc/default
/etc/default/pgbouncer
```

1.2. CentOS

```
# yum install pgbouncer
```

```
# rpm -ql pgbouncer.x86_64 0:1.5.4-1.rhel6
/etc/pgbouncer/mkauth.py
/etc/pgbouncer/mkauth.pyc
/etc/pgbouncer/mkauth.pyo
/etc/pgbouncer/pgbouncer.ini
/etc/rc.d/init.d/pgbouncer
/etc/sysconfig/pgbouncer
/usr/bin/pgbouncer
/usr/share/doc/pgbouncer-1.5.4
/usr/share/doc/pgbouncer-1.5.4/AUTHORS
/usr/share/doc/pgbouncer-1.5.4/NEWS
/usr/share/doc/pgbouncer-1.5.4/README
/usr/share/man/man1/pgbouncer.1.gz
/usr/share/man/man5/pgbouncer.5.gz
```


2. 配置 pgbouncer

databases 配置

```
[databases]
main = host=localhost port=5432 dbname=mydb user=myuser
password=mypass connect_query='SELECT 1'
```

pgbouncer

```
[pgbouncer]
logfile = /var/log/pgbouncer.log
pidfile = /var/run/pgbouncer/pgbouncer.pid
listen_addr = 127.0.0.1
listen_port = 6432
auth_type = trust
auth_file = /etc/pgbouncer/userlist.txt
admin_users = postgres
stats_users = stats, postgres
pool_mode = session
server_reset_query = DISCARD ALL
max_client_conn = 100
default_pool_size = 20
```

例 67.1. /etc/pgbouncer/pgbouncer.ini

```
# grep -v '^;' /etc/pgbouncer/pgbouncer.ini | grep -v '^$'
```

```
[databases]
main = host=localhost port=5432 dbname=test user=test
password=test connect_query='SELECT 1'
[pgbouncer]
logfile = /var/log/pgbouncer.log
pidfile = /var/run/pgbouncer/pgbouncer.pid
listen_addr = 127.0.0.1
```

```
listen_port = 1521
auth_type = trust
auth_file = /etc/pgbouncer/userlist.txt
admin_users = postgres
stats_users = stats, postgres
pool_mode = session
server_reset_query = DISCARD ALL
max_client_conn = 100
default_pool_size = 20
```

第 68 章 **Foreign data wrappers**

http://wiki.postgresql.org/wiki/Foreign_data_wrappers

第 69 章 Barman for PostgreSQL

<http://www.pgbarman.org/>

第 70 章 Connector

1. Pomm

<http://pomm.coolkeums.org/>

第 71 章 Replication

1. Bucardo

Asynchronous PostgreSQL Replication System

第 72 章 FAQ

1. Reset root password 重置MySQL root密码

忘记root密码是使用 --skip-grant-tables 启动项

CentOS 6.x

```
# vim /etc/init.d/mysqld

$exec --skip-grant-tables --datadir="$datadir" --socket="$socketfile" \
  --pid-file="$mypidfile" \
  --basedir=/usr --user=mysql >/dev/null 2>&1 &
```

```
# /etc/init.d/mysqld restart
Stopping mysqld:           [ OK ]
Starting mysqld:          [ OK ]

# mysqladmin -u root flush-privileges password "newpassword"
```

1.1. MySQL 5.7.x

CentOS 7.x

添加 skip-grant-tables=1 选项，然后重启mysql

```
# cat /etc/my.cnf
# For advice on how to change settings please see
# http://dev.mysql.com/doc/refman/5.6/en/server-configuration-
defaults.html

[mysqld]
#
# Remove leading # and set to the amount of RAM for the most important
data
```

```
# cache in MySQL. Start at 70% of total RAM for dedicated server, else
10%.
# innodb_buffer_pool_size = 128M
#
# Remove leading # to turn on a very important data integrity option:
logging
# changes to the binary log between backups.
# log_bin
#
# Remove leading # to set options mainly useful for reporting servers.
# The server defaults are faster for transactions and fast SELECTs.
# Adjust sizes as needed, experiment to find the optimal values.
# join_buffer_size = 128M
# sort_buffer_size = 2M
# read_rnd_buffer_size = 2M
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
skip-grant-tables=1
# Disabling symbolic-links is recommended to prevent assorted security
risks
symbolic-links=0

# Recommended in standard MySQL setup
sql_mode=NO_ENGINE_SUBSTITUTION,STRICT_TRANS_TABLES

[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

```
# systemctl restart mysqld
```

```
update mysql.user set authentication_string=password('netkiller') where
user='root' and Host = 'localhost';
flush privileges;
quit;
```

```
# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
```



```
Server version: 5.7.14 MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> update mysql.user set authentication_string=password('netkiller')
where user='root' and Host = 'localhost';
Query OK, 1 row affected, 1 warning (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 1

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

mysql> quit;
Bye
```

删除 skip-grant-tables=1 重启MySQL

1.2. MySQL 8.0

```
[root@localhost log]# vim /etc/my.cnf

[mysqld]
skip-grant-table
```

```
ALTER USER root@localhost identified by 'MQiEgelikst7S_6tlXzB0mt';
ALTER USER root@localhost PASSWORD EXPIRE NEVER;
```

2. 查看错误代码

```
mysql> \! perror 6
OS error code 6: No such device or address
```

2.1. ERROR 1153 (08S01) at line 3168: Got a packet bigger than 'max_allowed_packet' bytes

```
max_allowed_packet=500M
```

2.2. ERROR 1129 (00000): Host 'XXXXXXX' is blocked because of many connection errors; unblock with 'mysqladmin flush-hosts'

连接在中途被中断了的连接请求。在 max_connect_errors 次失败请求后，mysql 阻止该主机进一步的连接，直到管理员执行命令 mysqladmin flush-hosts。

```
mysql> flush hosts;
```

```
set global
max_connect_errors=1000;
```

```
max_connect_errors=10000
```

3. 临时表是否需要建索引

答案：要

4. [Warning] Changed limits: max_open_files: 5000 (requested 20480)

```
2018-01-08T01:34:44.515973Z 0 [Warning] Changed limits:
max_open_files: 5000 (requested 10240)
2018-01-08T01:34:44.516402Z 0 [Warning] Changed limits:
table_open_cache: 1471 (requested 2000)
```

提出出现在 CentOS 7 ulimit 配置没有问题的情况下mysql日志提示 Warning

```
# ulimit -Sa | grep "open files"
open files (-n) 40960
```

```
[root@netkiller ~]# cat /proc/`pidof
mysqld`/limits
Limit Soft Limit Hard Limit Units
Max cpu time unlimited unlimited
seconds
Max file size unlimited unlimited bytes
Max data size unlimited unlimited bytes
Max stack size 8388608 unlimited bytes
Max core file size 0 unlimited bytes
Max resident set unlimited unlimited
bytes
Max processes 63494 63494 processes
Max open files 5000 5000 files
Max locked memory 65536 65536 bytes
Max address space unlimited unlimited
bytes
Max file locks unlimited unlimited
locks
Max pending signals 63494 63494 signals
Max msgqueue size 819200 819200 bytes
```

```
Max nice priority 0 0
Max realtime priority 0 0
Max realtime timeout unlimited
unlimited us
```

动态改变

```
[root@netkiller ~]# egrep '^(Limit|Max
open files)' /proc/`pidof mysqld`/limits
Limit Soft Limit Hard Limit Units
Max open files 5000 5000 files
```

问题的出现原因是systemctl启动脚本覆盖了ulimit配置

```
# cat
/usr/lib/systemd/system/mysqld.service | grep -A2
open_files_limit
# Sets open_files_limit
LimitNOFILE = 5000
```

解决方法，直接修改上面的数值，不建议修改mysqld.service，这样会影响你下次升级。请采用下面的方案完美解决：

```
mkdir /usr/lib/systemd/system/mysqld.service.d
cat >> /usr/lib/systemd/system/mysqld.service.d/override.conf
<<EOF
[Service]
LimitNOFILE=40960
EOF
```

重启 MySQL

```
systemctl daemon-reload
systemctl restart mysqld
```

检查是否生效

```
mysql> show variables like 'open_files_limit';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| open_files_limit | 65535 |
+-----+-----+
1 row in set (0.01 sec)
```

5. this is incompatible with sql_mode=only_full_group_by

ERROR 1055 (42000): Expression #1 of SELECT list is not in GROUP BY clause and contains nonaggregated column 'mydb.contact.id' which is not functionally dependent on columns in GROUP BY clause; this is incompatible with sql_mode=only_full_group_by

```
mysql> select @@version;
+-----+
| @@version |
+-----+
| 5.7.10    |
+-----+
1 row in set (0.00 sec)

mysql> select @@GLOBAL.sql_mode;
+-----+
| @@GLOBAL.sql_mode
|
+-----+
|
| ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_
| DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_S
| UBSSTITUTION |
+-----+
+-----+
1 row in set (0.00 sec)

mysql> SET sql_mode = '';
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> select id,name from contact group by name limit 10;
+-----+-----+
```

id	name
84046	张伟
80259	张磊
784	王岩
87685	杨钊

10 rows in set (0.07 sec)

不建议设置 SET sql_mode = "", 正确方式如下:

```
mysql> set global
sql_mode='STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION';
mysql> set session
sql_mode='STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION';
```

或者采用

Adding only one mode to sql_mode without removing existing ones:

```
SET sql_mode=(SELECT CONCAT(@@sql_mode,',<mode_to_add>'));
```

Removing only a specific mode from sql_mode without removing others:

```
SET sql_mode=(SELECT
REPLACE(@@sql_mode,'<mode_to_remove>',''));
```

In your case, if you want to remove only ONLY_FULL_GROUP_BY mode, then use below command:

```
SET sql_mode=(SELECT REPLACE(@@sql_mode, 'ONLY_FULL_GROUP_BY',
```



```
''));
```

6. ERROR 1071 (42000) at line 25: Specified key was too long; max key length is 767 bytes

这个保存通常出现在高版本数据库向低版本数据导入数据，活着云主机例如阿里云。

```
mysql> show variables like '%innodb_large_prefix%';
+-----+-----+
| Variable_name      | Value |
+-----+-----+
| innodb_large_prefix | OFF   |
+-----+-----+
1 row in set (0.00 sec)
```

解决方案

```
innodb_large_prefix=ON
```

7. ERROR 1086 (HY000): File '/var/lib/mysql-files/order.txt' already exists

SELECT * FROM tablename INTO OUTFILE 不支持覆盖操作，这是MySQL从安全角度考虑的。

```
mysql> SELECT * FROM `order` INTO OUTFILE '/var/lib/mysql-  
files/order.txt';  
ERROR 1086 (HY000): File '/var/lib/mysql-files/order.txt'  
already exists
```

8. Error Code: 1146. Table 'test.CACHE_UPDATE' doesn't exist

问题分析，首先确认表是存在的，但是无法读取。可以判定是 lower_case_table_names=1 选项的问题，开启后表以小写方式打开。

```
Error Code: 1146. Table 'test.CACHE_UPDATE' doesn't exist
```

如果是 MySQL 8.0 之前没有开启 lower_case_table_names=1，现在需要开启，加入配置后将无法启动，解决办法是，你需要重做 mysql data 目录

```
[root@localhost ~]# rm -rf /var/lib/mysql/*
[root@localhost ~]# systemctl restart mysqld
[root@localhost ~]# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.21 Source distribution

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show variables like '%lower_case_table_names%';
+-----+-----+
| Variable_name          | Value |
+-----+-----+
| lower_case_table_names | 1     |
+-----+-----+
1 row in set (0.01 sec)

mysql> SELECT CURRENT_SERIAL FROM CACHE_UPDATE WHERE ID=1;
+-----+
| CURRENT_SERIAL |
+-----+
|                7 |
+-----+
```

```
+-----+  
1 row in set (0.00 sec)
```

9. ERROR 1273 (HY000) at line 3364: Unknown collation: 'utf8mb4_0900_ai_ci'

找出指定字符集的表

```
select TABLE_SCHEMA, TABLE_NAME, TABLE_COLLATION from
information_schema.tables where table_collation =
'utf8mb4_0900_ai_ci' and table_schema = 'your_schema';
```

```
SELECT
    CONCAT(
        'ALTER TABLE ',
        TABLE_NAME,
        ' CONVERT TO CHARACTER SET utf8mb4 COLLATE
utf8mb4_general_ci;'
    )
FROM
    information_schema.`TABLES`
WHERE
    TABLE_SCHEMA = 'DATABASE_NAME';
```

10. ERROR 1290 (HY000): The MySQL server is running with the --secure-file-priv option so it cannot execute this statement

MySQL 不允许向 `secure_file_priv` 意外的目录导出文件。

```
mysql> SELECT * FROM `order` INTO OUTFILE '/tmp/order.txt';
ERROR 1290 (HY000): The MySQL server is running with the --
secure-file-priv option so it cannot execute this statement
```

```
mysql> show variables like '%secure%';
```

Variable_name	Value
require_secure_transport	OFF
secure_auth	ON
secure_file_priv	/var/lib/mysql-files/

```
3 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM `order` INTO OUTFILE '/var/lib/mysql-
files/order.txt';
```

```
Query OK, 3 rows affected (0.00 sec)
```

```
root@netkiller ~ % cat /var/lib/mysql-files/order.txt
```

```
1      Tom      22      2017-11-16 17:23:15
2      Neo      34.65   2017-11-16 17:29:28
3      Cici     34.98   2017-11-16 17:30:29
```

在 `my.cnf` 中加入 `secure-file-priv=/tmp` 可以修改为你需要的目录。

11. ERROR 1364: 1364: Field 'id' doesn't have a default value

```
set
@@SESSION.sql_mode='NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION'
;
SELECT @@GLOBAL.sql_mode;
UPDATE `cms`.`content` SET `source`='test' WHERE
`content_id`='1099';
```


12. ERROR 1415: Not allowed to return a result set from a trigger

触发器中不允许返回结果集，解决方法是顶一个变量，然后将返回值返回给变量。

```
DROP TRIGGER IF EXISTS `test`.`demo_AFTER_INSERT`;  
  
DELIMITER $$  
USE `test`$$  
CREATE DEFINER=`root`@`%` TRIGGER `test`.`demo_AFTER_INSERT`  
AFTER INSERT ON `demo` FOR EACH ROW  
BEGIN  
    set @rev = "";  
    SELECT  
    OUT2FILE('/tmp/demo.log',  
            CONCAT_WS(',',  
                      NEW.id,  
                      NEW.name,  
                      NEW.sex,  
                      NEW.address))  
    INTO @rev;  
END$$  
DELIMITER ;
```

13. ERROR 1503 (HY000): A PRIMARY KEY must include all columns in the table's partitioning function

<http://dev.mysql.com/doc/refman/5.1/en/partitioning-limitations-partitioning-keys-unique-keys.html>

14. ERROR 1819 (HY000): Your password does not satisfy the current policy requirements

MySQL 5.7 密码强度，必须含有0-9，a-z,A-Z以及“-”或“_”

<https://dev.mysql.com/doc/refman/5.7/en/validate-password-options-variables.html>

禁用密码安全策略（早起5.7版本可用，新版已经废弃这个选项）

```
password                                # cat /etc/my.cnf | grep validate-  
                                         validate-password=OFF
```

新的方式修改策略与密码长度

```
mysql> set global validate_password_policy=0;  
mysql> set global validate_password_length=4;  
mysql> grant all privileges on test.* to 'test'@localhost  
identified by 'chen';
```

15. ERROR 1820 (HY000): You must reset your password using ALTER USER statement before executing this statement.

这个错误来自 MySQL 5.7，首次登陆MySQL 5.7 必须修改密码

```
ALTER USER 'root'@'localhost'  
IDENTIFIED BY 'your_password';
```

16. ERROR 1840 (HY000) at line 24: @@GLOBAL.GTID_PURGED can only be set when @@GLOBAL.GTID_EXECUTED is empty.

问题出现在 MySQL 5.7 导入数据库时候

```
[www@testing ~]$ zcat netkiller.2021-08-19.sql.gz | mysql
netkiller
ERROR 1840 (HY000) at line 24: @@GLOBAL.GTID_PURGED can only be
set when @@GLOBAL.GTID_EXECUTED is empty.
```

解决方案

执行 reset master;

```
[www@testing ~]$ mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 25669
Server version: 5.7.35 MySQL Community Server (GPL)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All
rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or
its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current
input statement.

mysql> reset master;
```

```
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> exit
```

```
Bye
```

```
[www@testing ~]$ zcat netkiller.2021-08-19.sql.gz | mysql  
netkiller
```

17. ERROR 3024 (HY000): Query execution was interrupted, maximum statement execution time exceeded

```
mysql> select * from cert;
ERROR 3024 (HY000): Query execution was interrupted, maximum
statement execution time exceeded
mysql> SET GLOBAL max_execution_time=10;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from cert;
Empty set (0.08 sec)

mysql> show variables like 'max_execution_time';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| max_execution_time | 10 |
+-----+-----+
1 row in set (0.01 sec)

mysql> select /*+ max_execution_time(3000)*/ count(*) from
cert;
+-----+
| count(*) |
+-----+
|          0 |
+-----+
1 row in set (0.29 sec)
```

18. Authentication plugin

**'caching_sha2_password' cannot be loaded:
/usr/lib64/mysql/plugin/caching_sha2_password.s
o: cannot open shared object file: No such file or
directory**

这个故障出现在 MySQL 8.0 上，用户使用 mysql client 5.7 链接 MySQL 8.0 提示如下

```
[root@netkiller ~]# mysql -h 193.112.95.53 -uroot -p
Enter password:
ERROR 2059 (HY000): Authentication plugin
'caching_sha2_password' cannot be loaded:
/usr/lib64/mysql/plugin/caching_sha2_password.so: cannot open
shared object file: No such file or directory
```

解决方案，创建用户使用 mysql_native_password 密码

```
mysql> CREATE USER 'root'@'%' IDENTIFIED WITH
mysql_native_password BY 'pMQiEgelikst7S_6tlXzB0mt_4b';
Query OK, 0 rows affected (0.08 sec)

mysql> grant all on *.* to 'root'@'%';
Query OK, 0 rows affected (0.08 sec)
```

重新链接

```
[root@netkiller ~]# mysql -h 193.112.95.53 -uneo -p
```


Enter password:

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 24

Server version: 8.0.11 MySQL Community Server - GPL

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

19. com.mysql.jdbc.exceptions.jdbc4.MySQLNonTransientConnectionException: Public Key Retrieval is not allowed

问题出在现在 MySQL 8.0 版本

解决方法：在连接后面添加 allowPublicKeyRetrieval=true

```
spring.datasource.url=jdbc:mysql://192.168.0.1:3306/test?  
useUnicode=true&characterEncoding=UTF-  
8&serverTimezone=UTC&useSSL=false&allowPublicKeyRetrieval=true
```

20. mysqldump: Couldn't execute 'SELECT COLUMN_NAME,

问题出现在 MySQL 8.0 备份 MySQL 5.7 数据库时。

```
mysqldump: Couldn't execute 'SELECT COLUMN_NAME,  
JSON_EXTRACT(HISTOGRAM, '$."number-of-buckets-specified"')  
FROM information_schema.COLUMN_STATISTICS          WHERE  
SCHEMA_NAME = 'testra' AND TABLE_NAME = 'branch';': Unknown  
table 'column_statistics' in information_schema (1109)
```

解决办法，使用 --column-statistics=0 选项

```
mysqldump -hdb.netkiller.cn -uroot -ptest neo --column-  
statistics=0
```

21. this is incompatible with sql_mode=only_full_group_by

```
Expression #2 of SELECT list is not in GROUP BY clause and contains nonaggregated column 'test.table.username' which is not functionally dependent on columns in GROUP BY clause; this is incompatible with sql_mode=only_full_group_by
```

问题出在 MySQL 5.7 向 MySQL 8.0 迁移。

查询 sql_mode 设置

```
select @@global.sql_mode;
'ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION'
```

临时解决方案，去掉 ONLY_FULL_GROUP_BY 即可

```
set
@@GLOBAL.sql_mode='STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,
NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION';
set
@@SESSION.sql_mode='STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,
NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION';
```

彻底解决，有三种解决方案：

- 第一种，将 MySQL 的版本降回到 5.7
- 第二种，关闭 `only_full_group_by` 检查
- 第三种，修改sql使其遵守`only_full_group_by`语法规则

采用哪个方案？打工人的方案，就是解决眼前问题，如果你对项目负责，最好采用第三种方案。

我的职业生涯遇到无数次，因版本太低同时发现重大漏洞，厂商已经不在维护该版本，此时一边是黑客攻击，一边你又无法短时间内升级到新版本，公司又不能接受停服，你会怎么应对？我用了很多非常规手段，给开发团队争取了一周的时间，升级系统。

22. mysqldump: [Warning] Using a password on the command line interface can be insecure.

```
mysqldump: [Warning] Using a password on the command line interface can be insecure.
```

```
vim ~/.my.cnf  
  
[mysqldump]  
user=root  
password=123456
```

```
[root@netkiller neo]# mysqldump -h 192.168.30.40 neo  
[root@netkiller neo]# mysqldump --defaults-extra-file=~/.my.cnf  
netkiller > netkiller.sql
```

23. mysql: [Warning] Using a password on the command line interface can be insecure.

当使用 `-pnetkiller` 在命令行出现密码的时候，会提示下面信息。

```
[www@testing ~]$ mysql -h127.0.0.1 -uneo -pnetkiller test
mysql: [Warning] Using a password on the command line interface
can be insecure.
Reading table information for completion of table and column
names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 25623
Server version: 5.7.35 MySQL Community Server (GPL)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All
rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or
its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current
input statement.

mysql>
```

创建 `~/.my.cnf` 配置文件，将密码写入配置

```
[www@testing ~]$ cat ~/.my.cnf
[mysql]
host=127.0.0.1
```

```
user=neo  
password=netkiller
```

这时直接使用 mysql 命令即可进入。

```
[www@testing ~]$ mysql  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 25622  
Server version: 5.7.35 MySQL Community Server (GPL)  
  
Copyright (c) 2000, 2020, Oracle and/or its affiliates. All  
rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or  
its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current  
input statement.  
  
mysql>
```


部分 VII. Oracle

第 73 章 Oracle 12c 安装可

```
[root@localhost ~]# hostnamectl
  Static hostname: localhost.localdomain
        Icon name: computer-desktop
        Chassis: desktop
        Machine ID: b4f65cfce3a24bdea03ec5b8ee0a0260
        Boot ID: f6e5e776b7cf47f882f4fcc0c70be2e9
  Operating System: CentOS Linux 7 (Core)
        CPE OS Name: cpe:/o:centos:centos:7
        Kernel: Linux 3.10.0-229.el7.x86_64
  Architecture: x86_64
```

1. Server 安装

```
unzip linuxamd64_12c_database_1of2.zip
unzip linuxamd64_12c_database_2of2.zip

cd database/
./runInstaller
```

Configure Security Updates

- Configure Security Updates**
- Installation Option
- Grid Installation Options
- Install Type
- Typical Installation
- Prerequisite Checks
- Summary
- Install Product
- Finish

Provide your email address to be informed of security issues, install the product and initiate configuration manager. [View details.](#)

Email:

Easier for you if you use your My Oracle Support email address/username.

I wish to receive security updates via My Oracle Support.

My Oracle Support Password:

Help

< Back

Next >

Install

Cancel

Select Installation Option

- [Configure Security Updates](#)
- Installation Option**
- [Grid Installation Options](#)
- Install Type
 - Typical Installation
 - Prerequisite Checks
 - Summary
 - Install Product
 - Finish

Select any of the following install options.

- Create and configure a database
- Install database software only
- Upgrade an existing database

Help

< Back

Next >

Install

Cancel

System Class

- [Configure Security Updates](#)
- [Installation Option](#)
- [System Class](#)**
- [Typical Installation](#)
- [Prerequisite Checks](#)
- [Summary](#)
- [Install Product](#)
- [Finish](#)

Desktop class

Choose this option if you are installing on a laptop or desktop class system. This option includes a starter database and allows minimal configuration.

Server class

Choose this option if you are installing on a server class system, which Oracle defines as a system used in a production data center. This option allows for more advanced configuration options.

Help

< Back

Next >

Install

Cancel

Grid Installation Options

- [Configure Security Updates](#)
- [Installation Option](#)
- [System Class](#)
- [Grid Installation Options](#)**
- [Install Type](#)
- Typical Installation
- Prerequisite Checks
- Summary
- Install Product
- Finish

Select the type of database installation you want to perform.

- Single instance database installation
- Oracle Real Application Clusters database installation
- Oracle RAC On g Node database installation

Help

< Back

Next >

Install

Cancel

Select Install Type

- [Configure Security Updates](#)
- [Installation Option](#)
- [System Class](#)
- [Grid Installation Options](#)
- [Install Type](#)**
- [Typical Installation](#)
- [Prerequisite Checks](#)
- [Summary](#)
- [Install Product](#)
- [Finish](#)

- T**ypical install
Perform full Oracle Database installation with basic configuration.
- A**dvanced install
Allows advanced selections such as different passwords for the SYS, SYSMAN, SYSTEM and DBSNMP accounts, database character set, product languages, automated backups, custom installation, and alternative storage options such as Oracle Automatic Storage Management.

Help

< Back **Next >** Install Cancel

Typical Install Configuration

- [Configure Security Updates](#)
- [Installation Option](#)
- [System Class](#)
- [Grid Installation Options](#)
- [Install Type](#)
- Typical Installation**
- [Prerequisite Checks](#)
- Summary
- Install Product
- Finish

Perform full database installation with basic configuration.

Oracle base:

Software location:

Storage type:

Database file location:

Database edition:

OSDBA group:

Global database name:

Administrative password:

Confirm password:

Service name:

Create as Container database

Pluggable database name:

Create Inventory

- Configure Security Updates
- Installation Option
- System Class
- Grid Installation Options
- Install Type
- Typical Installation
- Create Inventory**
- Prerequisite Checks
- Summary
- Install Product
- Finish

You are starting your first installation on this host. Specify a directory for installation metadata files (for example, install log files). This directory is called the "inventory directory". The installer automatically sets up subdirectories for each product to contain inventory data. The subdirectory for each product typically requires 150 kilobytes of disk space.

Inventory Directory:

Specify an operating system group whose members have write permission to the inventory directory (orainventory).

orainventory Group Name:

Help

< Back

Next >

Install

Cancel

Summary

- [Configure Security Updates](#)
- [Installation Option](#)
- [System Class](#)
- [Grid Installation Options](#)
- [Install Type](#)
- [Typical Installation](#)**
- [Create Inventory](#)
- [Prerequisite Checks](#)
- [Summary](#)**
- [Install Product](#)
- [Finish](#)

Oracle Database 12c Release 1 Installer

- Global settings**
 - Disk space: required 6.4 GB available 282.58 GB [\[Edit\]](#)
 - Source location: /opt/database/install/./stage/products.xml
 - Install method: Typical installation [\[Edit\]](#)
 - Database edition: Enterprise Edition (Create and configure a database) [\[Edit\]](#)
 - Oracle base: /opt/oracle [\[Edit\]](#)
 - Software location: /opt/oracle/product/12.1.0/dbhome_1 [\[Edit\]](#)
 - Privileged Operating System groups: dba (OSDBA), dba (OSOPER), dba (OSBACKUPDBA), dba (OSRMAN)
- Inventory information**
 - Inventory location: /opt/orainventory [\[Edit\]](#)
 - oraInventory group: oinstall [\[Edit\]](#)
- Database information**
 - Configuration: General Purpose / Transaction Processing
 - Global database name: orcl.localdomain [\[Edit\]](#)
 - Oracle system identifier (SID): orcl [\[Edit\]](#)
 - Pluggable database name: pdborcl [\[Edit\]](#)
 - Allocated memory: 746 MB
 - Automatic memory management option: TRUE
 - Database character set : West European (WE8MSWIN1252)
 - Management method: Database express

[Save Response File...](#)

[Help](#)

< Back

Next >

Install

Cancel

Oracle Database 12c Release 1 Installer – Step 10 of 11

Install Product

ORACLE DATABASE 12^c

Configure Security Updates
 Installation Option
 System Class
 Grid Installation Options
 Install Type
 Typical Installation
 Create Inventory
 Prerequisite Checks
 Summary
Install Product
 Finish

Progress

36%

Extracting files to '/opt/oracle/product/12.1.0/dbhome_1'.

Status

→ Oracle Database installation	In Progress
• Prepare	Succeeded
• Copy files	In Progress
• Link binaries	Pending
• Setup	Pending
Setup Oracle Base	Pending
Execute Root Scripts	Pending
Oracle Database configuration	Pending

Details Retry Skip

ORACLE DATABASE 12^c

Help < Back Next > Install Cancel

Execute Configuration scripts

The following configuration scripts need to be executed as the "root" user.

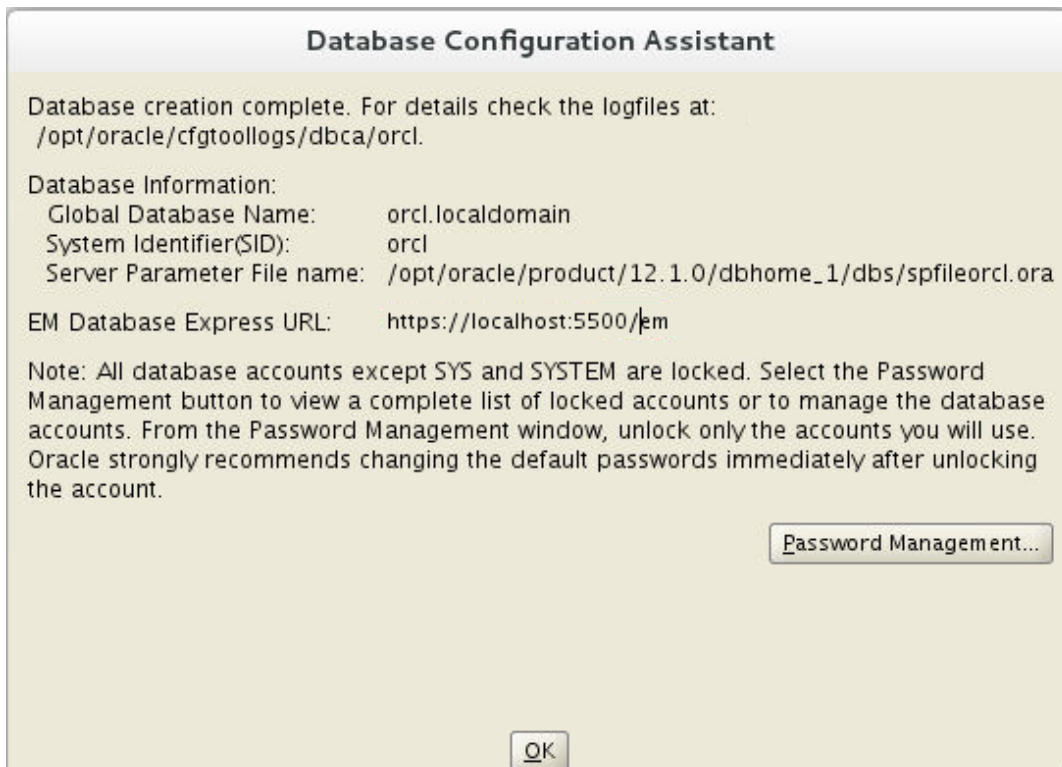
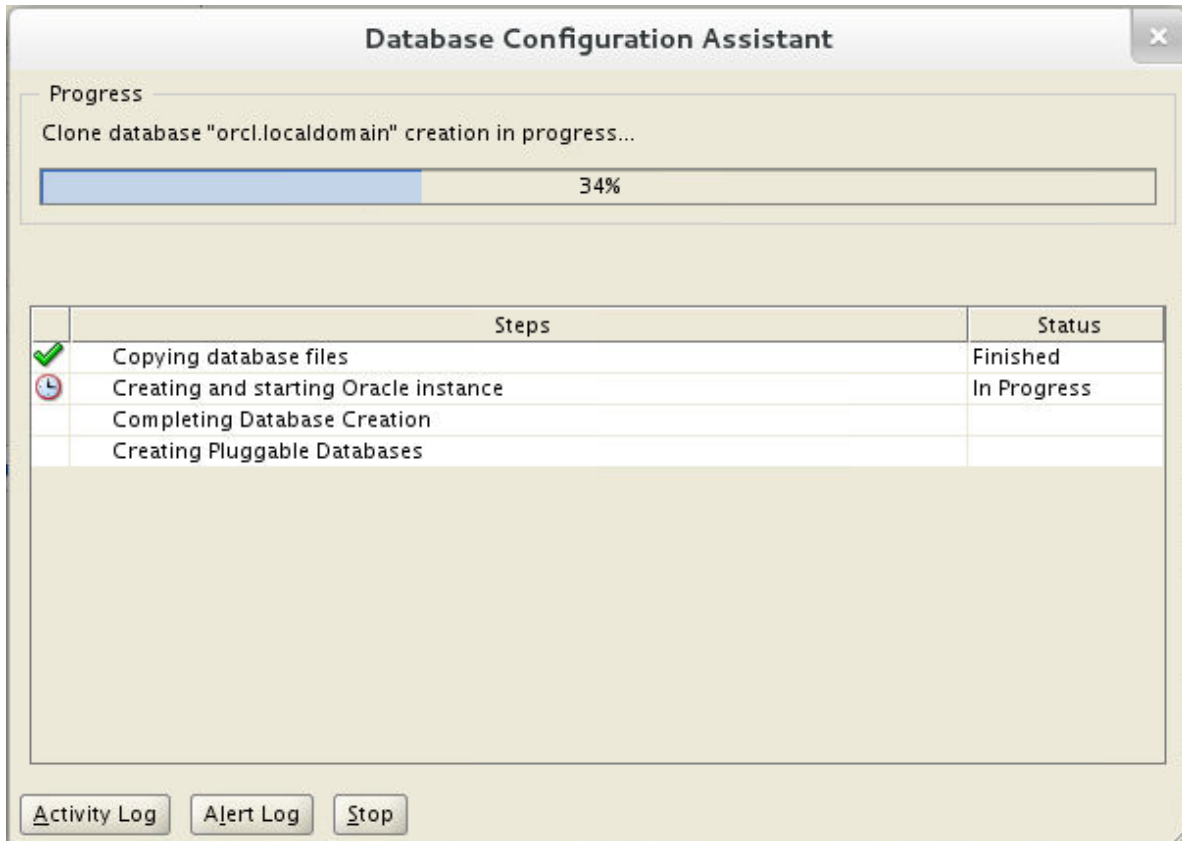
Scripts to be executed:

Number	Script Location
1	/opt/orainventory/orainstRoot.sh
2	/opt/oracle/product/12.1.0/dbhome_1/root.sh

To execute the configuration scripts:

1. Open a terminal window
2. Log in as "root"
3. Run the scripts
4. Return to this window and click "OK" to continue

Help OK



Password Management

Lock / unlock database user accounts and / or change the default passwords:

User Name	Lock Account? ▲	New Password	Confirm Password
SYS		••••	••••
SYSTEM		••••	••••
OUTLN	✓		
LBACSYS	✓		
OLAPSYS	✓		
SI_INFORMTN_SCHEMA	✓		
DVSY	✓		
ORDPLUGINS	✓		
XDB	✓		
ANONYMOUS	✓		
CTXSYS	✓		
ORDDATA	✓		
GSMADMIN_INTERNAL	✓		
APPQOSSYS	✓		
APEX_040200	✓		
WMSYS	✓		

OK Cancel

Database Configuration Assistant

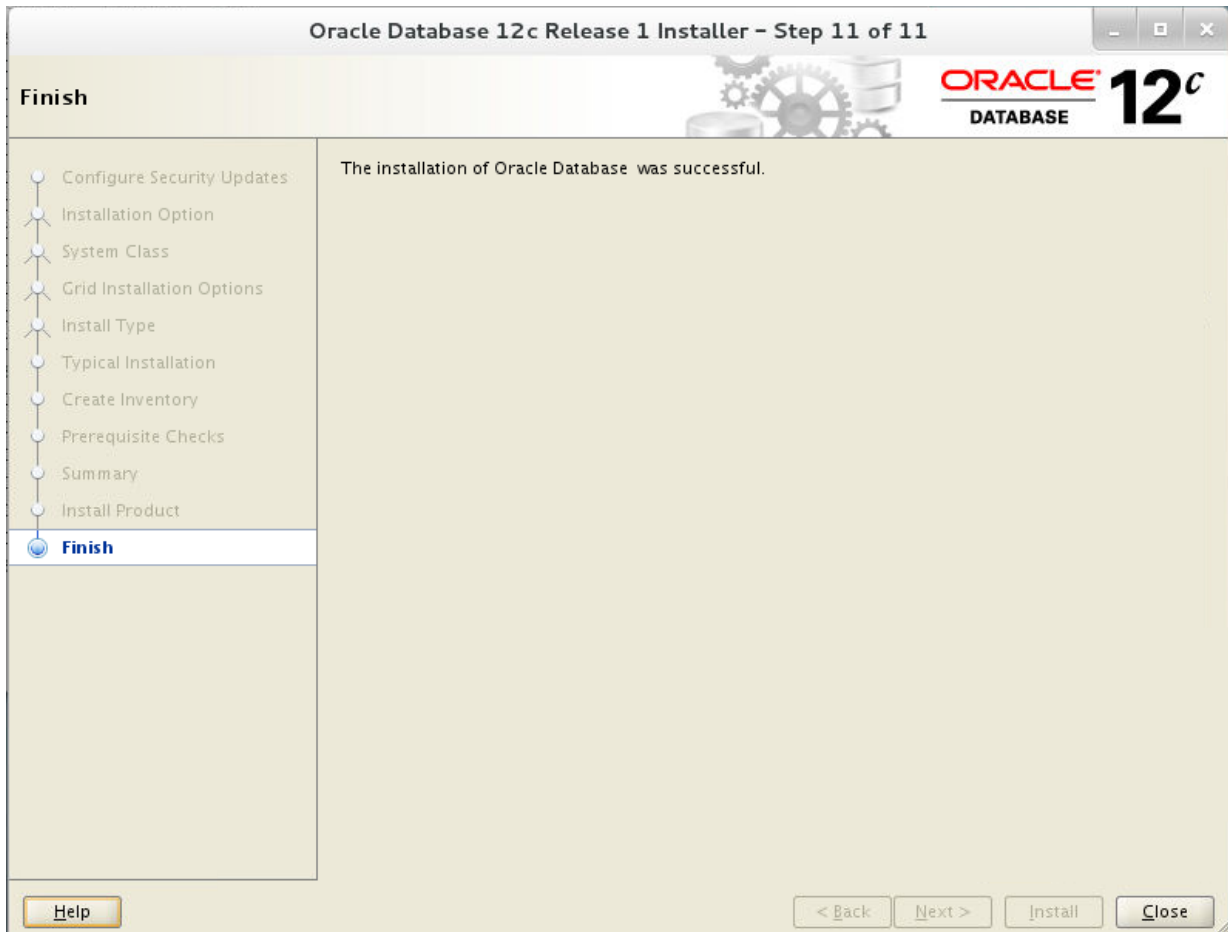


Password for the following users does not satisfy Oracle recommended password complexity policy. A password should have minimum of 8 characters in length. In addition, the password must contain at least one upper case character, one lower case character and one digit:

SYS
SYSTEM

Do you want to continue?

Yes No



```
[root@localhost ~]# /opt/oraInventory/orainstRoot.sh
Changing permissions of /opt/oraInventory.
Adding read,write permissions for group.
Removing read,write,execute permissions for world.

Changing groupname of /opt/oraInventory to oinstall.
The execution of the script is complete.
```

```
[root@localhost ~]# /opt/oracle/product/12.1.0/dbhome_1/root.sh
Performing root user operation.

The following environment variables are set as:
    ORACLE_OWNER= oracle
    ORACLE_HOME=  /opt/oracle/product/12.1.0/dbhome_1

Enter the full pathname of the local bin directory:
[/usr/local/bin]:
```

```
Copying dbhome to /usr/local/bin ...  
Copying oraenv to /usr/local/bin ...  
Copying coraenv to /usr/local/bin ...
```

Creating /etc/oratab file...

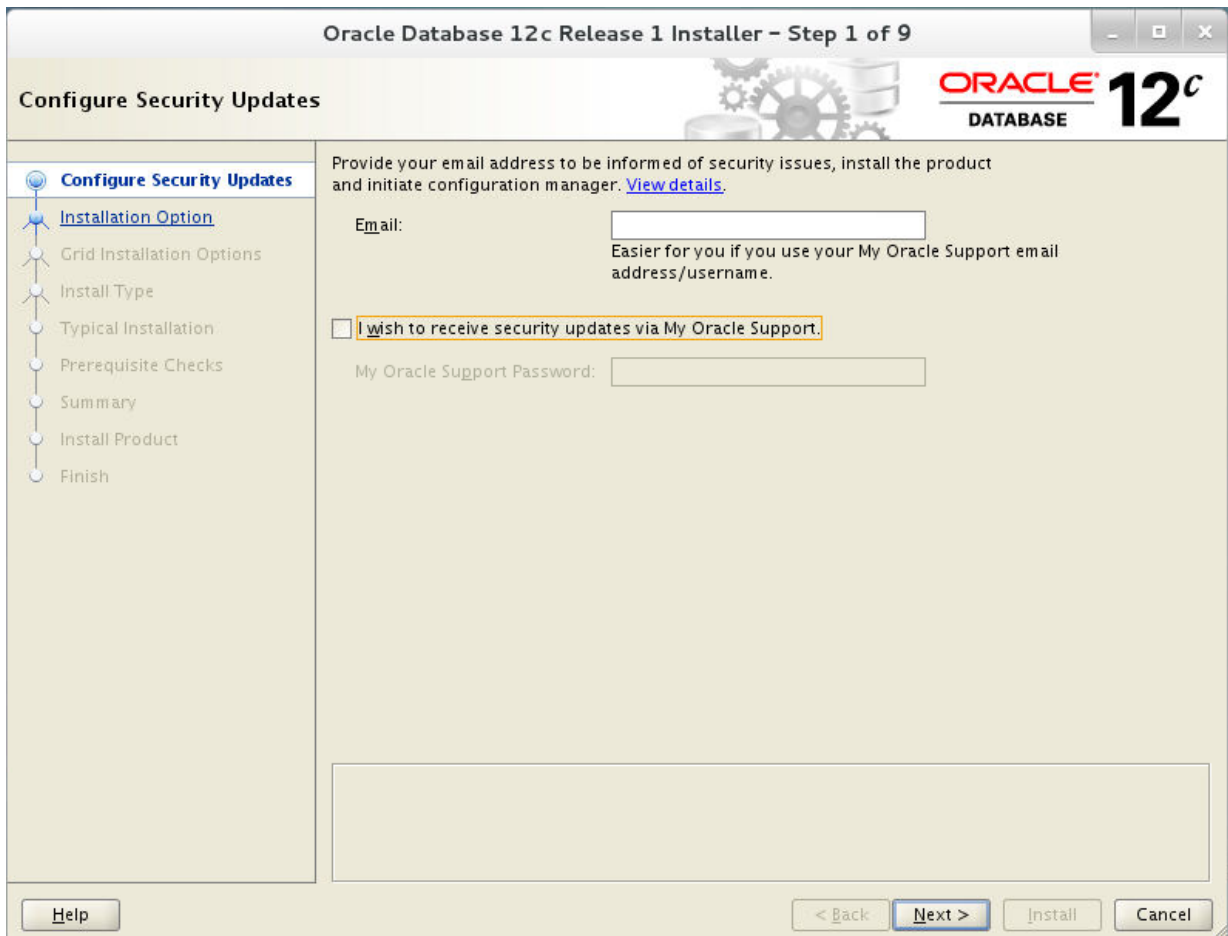
Entries will be added to the /etc/oratab file as needed by
Database Configuration Assistant when a database is created

Finished running generic part of root script.

Now product-specific root actions will be performed.

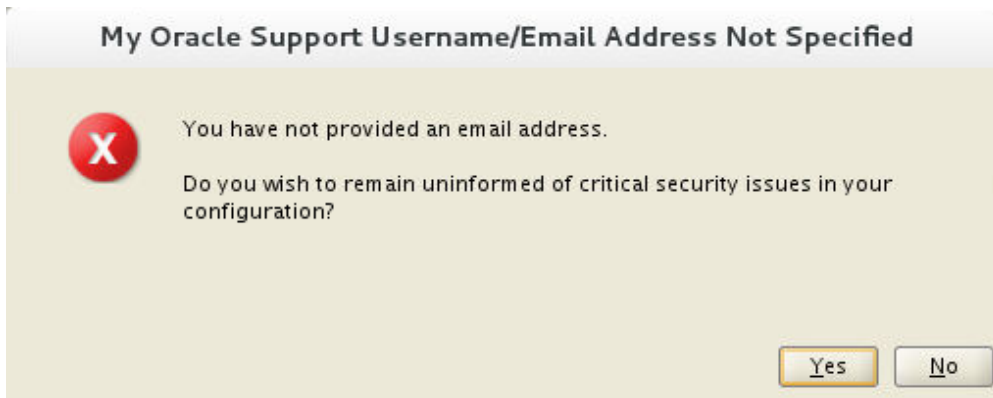
2. Desktop 安装

Desktop 是指安装在桌面计算机，笔记本电脑上的数据库，占用资源比较少。

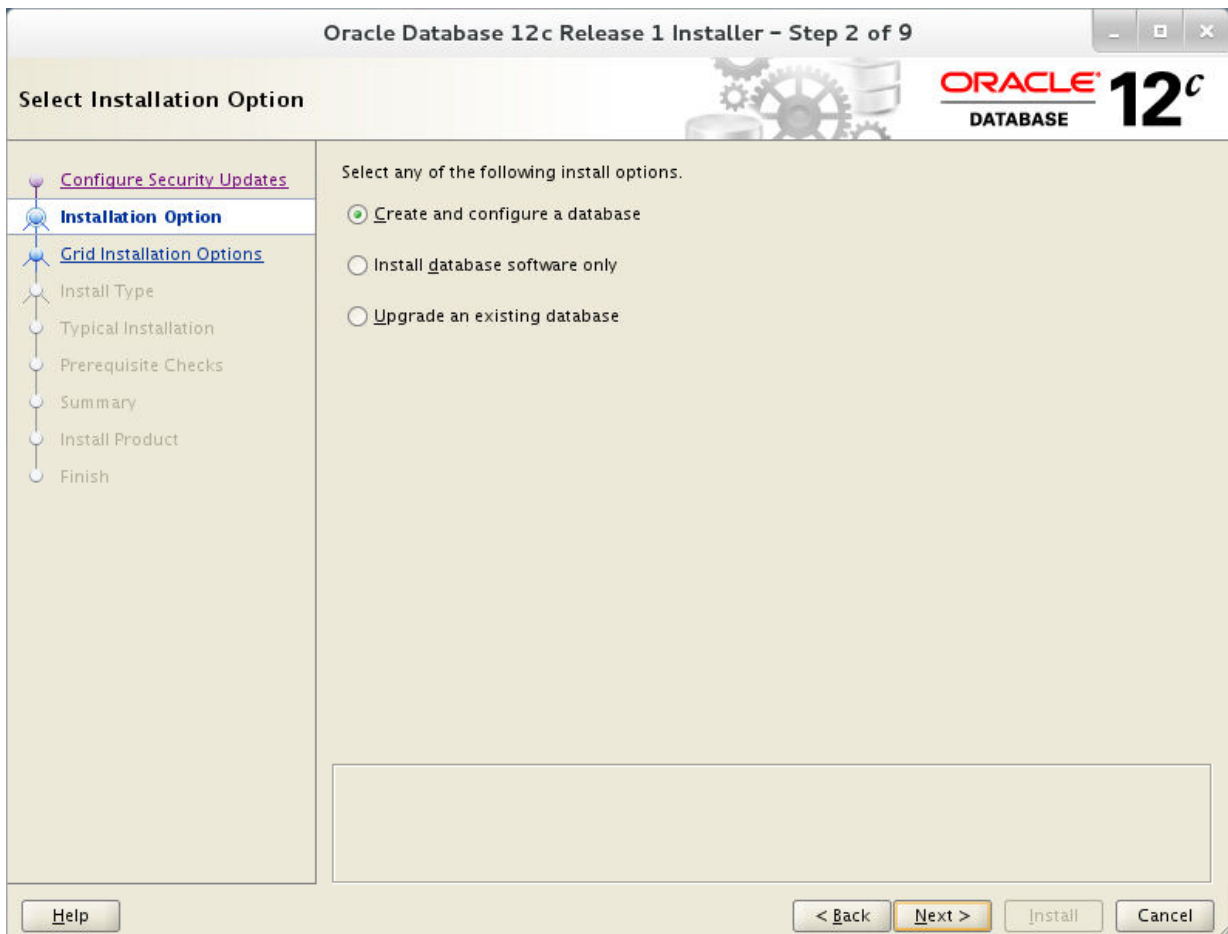


如果你想接收Oracle发来的关于数据库安全的邮件，请填写email地址

勾去选项 I wish to receive security updates via My Oracle Support.



点击Yes按钮继续

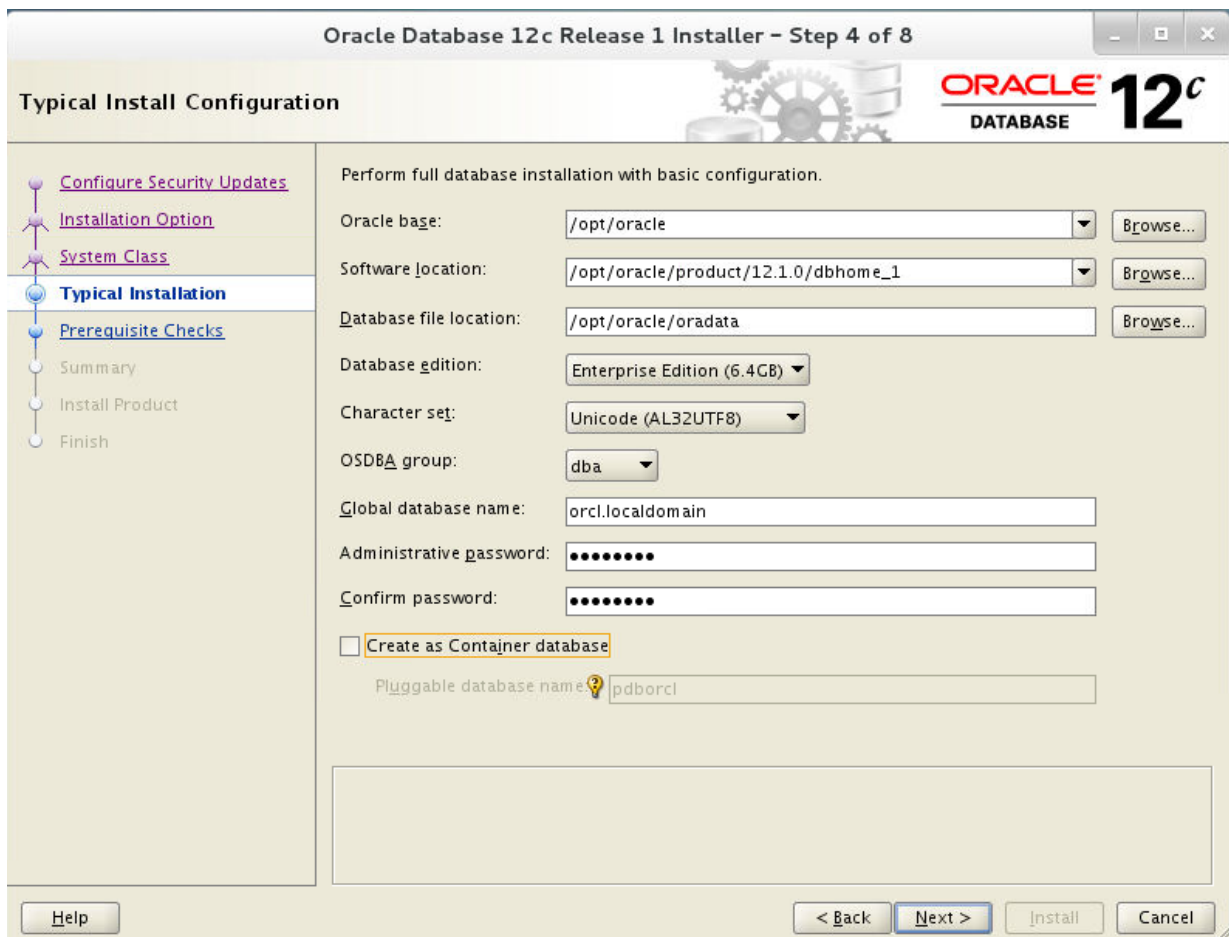




System Class

- [Configure Security Updates](#)
- [Installation Option](#)
- [System Class](#)**
- [Typical Installation](#)
- [Prerequisite Checks](#)
- [Summary](#)
- [Install Product](#)
- [Finish](#)

- D**esktop class
Choose this option if you are installing on a laptop or desktop class system. This option includes a starter database and allows minimal configuration.
- S**erver class
Choose this option if you are installing on a server class system, which Oracle defines as a system used in a production data center. This option allows for more advanced configuration options.



Character set 选择 Unicode(AL32UTF8)

输入 Administrative password 与 Confirm password 两个密码。

勾去选项 Create as Container database

Create Inventory

- Configure Security Updates
- Installation Option
- System Class
- Grid Installation Options
- Install Type
- Typical Installation
- Create Inventory**
- Prerequisite Checks
- Summary
- Install Product
- Finish

You are starting your first installation on this host. Specify a directory for installation metadata files (for example, install log files). This directory is called the "inventory directory". The installer automatically sets up subdirectories for each product to contain inventory data. The subdirectory for each product typically requires 150 kilobytes of disk space.

Inventory Directory

Specify an operating system group whose members have write permission to the inventory directory (orainventory).

orainventory Group Name:

Help

< Back

Next >

Install

Cancel



Summary

- [Configure Security Updates](#)
- [Installation Option](#)
- [System Class](#)
- [Typical Installation](#)
- [Create Inventory](#)
- [Prerequisite Checks](#)
- Summary**
- [Install Product](#)
- [Finish](#)

Oracle Database 12c Release 1 Installer

- Global settings**
 - Disk space: required 6.4 GB available 56.35 GB [\[Edit\]](#)
 - Source location: /opt/database/install/./stage/products.xml
 - Install method: Desktop installation [\[Edit\]](#)
 - Database edition: Enterprise Edition (Create and configure a database) [\[Edit\]](#)
 - Oracle base: /opt/oracle [\[Edit\]](#)
 - Software location: /opt/oracle/product/12.1.0/dbhome_1 [\[Edit\]](#)
 - Privileged Operating System groups: dba (OSDBA), dba (OSOPER), dba (OSBACKUPDBA), db
- Inventory information**
 - Inventory location: /opt/orainventory [\[Edit\]](#)
 - orainventory group: oinstall [\[Edit\]](#)
- Database information**
 - Configuration: General Purpose / Transaction Processing
 - Global database name: orcl.localdomain [\[Edit\]](#)
 - Oracle system identifier (SID): orcl [\[Edit\]](#)
 - Allocated memory: 25696 MB
 - Automatic memory management option: FALSE
 - Database character set : Unicode standard UTF-8 (AL32UTF8) [\[Edit\]](#)
 - Management method: Database express
 - Database storage mechanism: File system

[Save Response File...](#)

[Help](#)

[< Back](#)

[Next >](#)

[Install](#)

[Cancel](#)

Oracle Database 12c Release 1 Installer - Step 8 of 9

Install Product

ORACLE DATABASE 12^c

Configure Security Updates
 Installation Option
 System Class
 Typical Installation
 Create Inventory
 Prerequisite Checks
 Summary
Install Product
 Finish

Progress

23%

Extracting files to '/opt/oracle/product/12.1.0/dbhome_1'.

Status

➔ Oracle Database installation	In Progress
• Prepare	Succeeded
➔ • Copy files	In Progress
• Link binaries	Pending
• Setup	Pending
Setup Oracle Base	Pending
Execute Root Scripts	Pending
Oracle Database configuration	Pending

Details Retry Skip

ORACLE DATABASE 12^c

Help < Back Next > Install Cancel

Execute Configuration scripts

The following configuration scripts need to be executed as the "root" user.

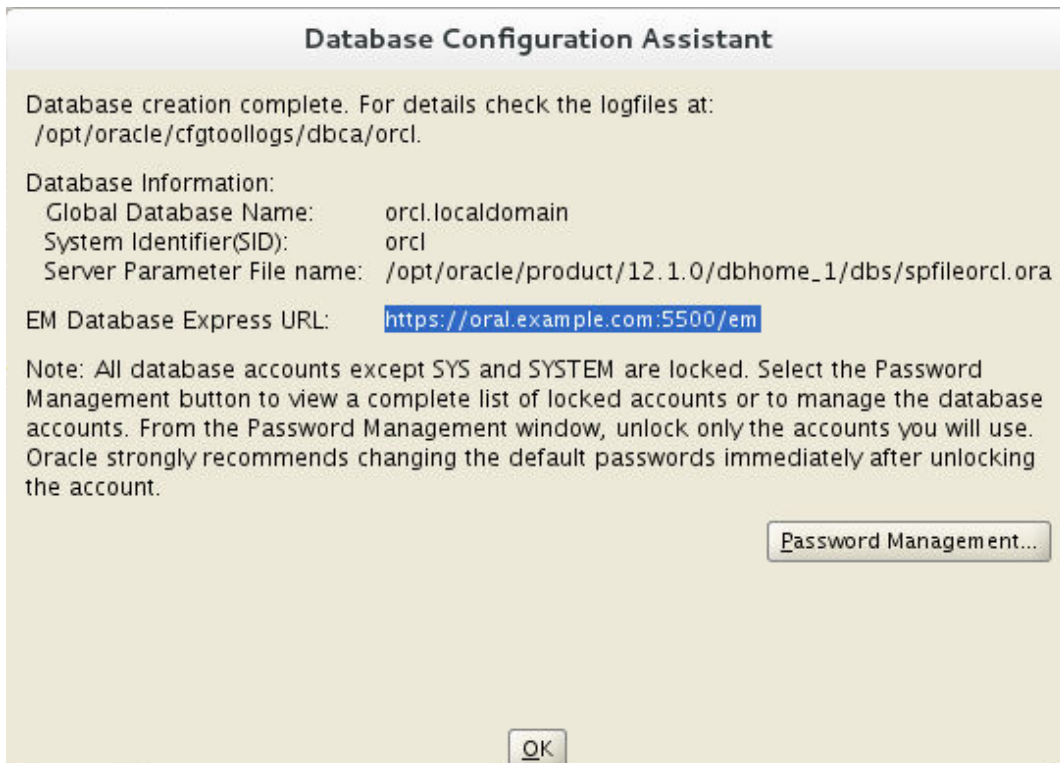
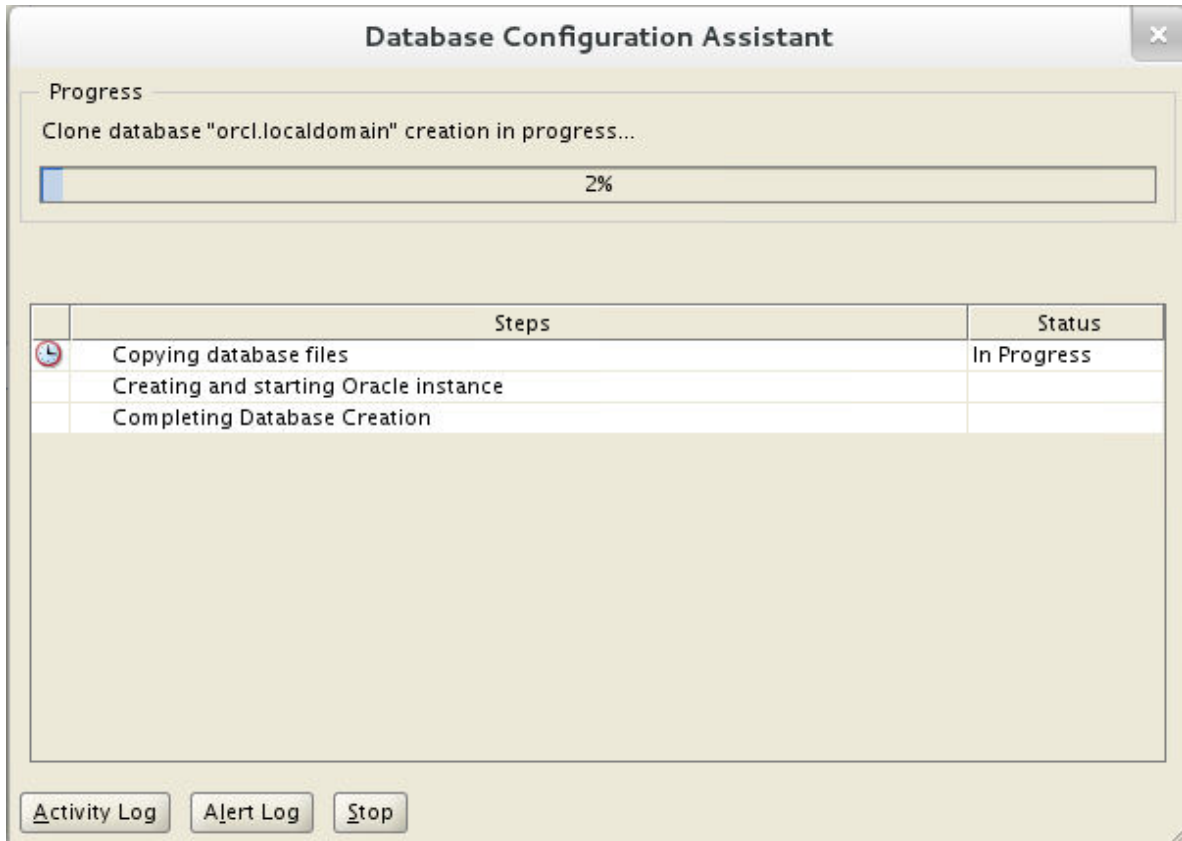
Scripts to be executed:

Number	Script Location
1	/opt/orainventory/orainstRoot.sh
2	/opt/oracle/product/12.1.0/dbhome_1/root.sh

To execute the configuration scripts:

1. Open a terminal window
2. Log in as "root"
3. Run the scripts
4. Return to this window and click "OK" to continue

Help OK



Password Management

Lock / unlock database user accounts and / or change the default passwords:

User Name	Lock Account? ▲	New Password	Confirm Password
SYS		••••	••••
SYSTEM		••••	••••
OUTLN	✓		
LBACSYS	✓		
OLAPSYS	✓		
SI_INFORMTN_SCHEMA	✓		
DVSY	✓		
ORDPLUGINS	✓		
XDB	✓		
ANONYMOUS	✓		
CTXSYS	✓		
ORDDATA	✓		
GSMADMIN_INTERNAL	✓		
APPQOSSYS	✓		
APEX_040200	✓		
WMSYS	✓		

OK Cancel

Database Configuration Assistant



Password for the following users does not satisfy Oracle recommended password complexity policy. A password should have minimum of 8 characters in length. In addition, the password must contain at least one upper case character, one lower case character and one digit:

SYS
SYSTEM

Do you want to continue?

Yes

No



Finish

- Configure Security Updates
- Installation Option
- System Class
- Grid Installation Options
- Install Type
- Typical Installation
- Create Inventory
- Prerequisite Checks
- Summary
- Install Product

Finish

The installation of Oracle Database was successful.

[Help](#)

< Back Next > Install [Close](#)

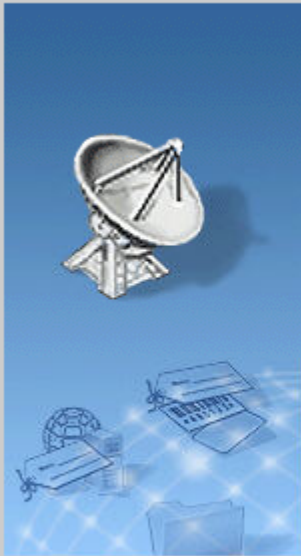
3. Oracle Net Services Configuration

3.1. Listener configuration

```
[oracle@localhost database]$ netca
```



Oracle Net Configuration Assistant: Listener Configuration, Listener



For remote connections to be made to your Oracle database, you must configure a Oracle Net listener. The Oracle Net Configuration Assistant allows you to add, reconfigure, rename or delete a listener.

Select what you want to do:

- Add
- Reconfigure
- Delete
- Rename

Cancel

Help

Back

Next

Oracle Net Configuration Assistant: Listener Configuration, Listener Name



For remote connections to be made to your Oracle database you must have at least one Oracle Net listener. Enter the name of the listener you want to create:

Listener name:

Cancel

Help

Back

Next

Oracle Net Configuration Assistant: Listener Configuration, Select Protoc...



You can configure the listener to accept connections over one or more protocols. Select which protocols you want to configure for this listener. Keep your configuration as simple as possible by configuring only the protocols you need.

Available Protocols

- TCPS
- IPC



Selected Protocols

- TCP



Cancel Help

Back Next

Oracle Net Configuration Assistant: Listener Configuration, TCP/IP Proto...



Which TCP/IP port number should the listener use? The port number selected should not be used by any other software on this computer.

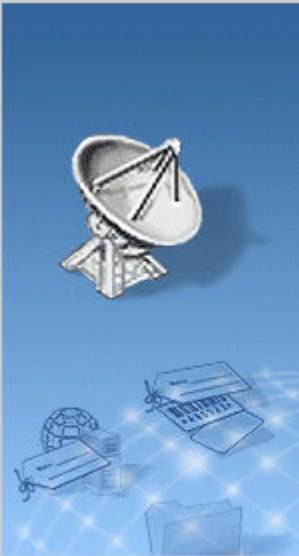
Use the standard port number of 1521

Use another port number:

Cancel Help

Back Next

Oracle Net Configuration Assistant: Listener Configuration, More Listene...



Would you like to configure another listener?

No

Yes

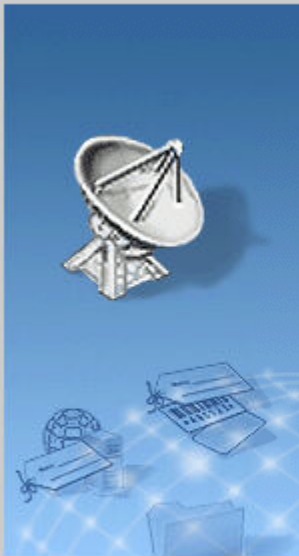
Cancel

Help

< Back

Next >

Oracle Net Configuration Assistant: Listener Configuration, Select Listener



Select a listener you want to start:

LISTENER

Cancel

Help

< Back

Next >



```
cat /opt/oracle/product/12.1.0/db_1/network/admin/listener.ora
# listener.ora Network Configuration File:
/opt/oracle/product/12.1.0/db_1/network/admin/listener.ora
# Generated by Oracle configuration tools.

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = oral.example.com)(PORT
= 1521))
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
    )
  )
```

3.2. Local Net Service Name configuration





4. Oracle 12c Client for Windows 10





选择安装类型

选择安装类型

选择产品语言

执行先决条件检查

概要

安装产品

完成

您需要何种安装类型?

InstantClient (350.0MB)(S)

安装 Instant Client 软件

管理员 (1.5GB)(A)

安装管理控制台, 管理工具, 联网服务, 实用程序和基本客户机软件。

运行时 (1.1GB)(R)

安装用于开发应用程序、网络服务和基本客户机软件的工具。

定制(C)

使您可选择单个组件进行安装。

帮助(H)

< 上一步(B)

下一步(N) >

安装(I)

取消



选择产品语言

- 选择安装类型
- 选择产品语言**
- 执行先决条件检查
- 概要
- 安装产品
- 完成

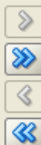
请选择运行产品时使用的语言。

可用语言(A):

- 丹麦语
- 乌克兰语
- 俄语
- 保加利亚语
- 克罗地亚语
- 冰岛语
- 加泰罗尼亚语
- 匈牙利语
- 印度尼西亚语
- 土耳其语
- 埃及语
- 墨西哥西班牙语
- 孟加拉语
- 希伯来语
- 希腊语
- 汉语

所选语言(S):

- 简体中文
- 英语



帮助(H)

< 上一步(B)

下一步(N) >

安装(I)

取消



指定 Oracle 主目录用户

选择安装类型

选择产品语言

Oracle 主目录用户选择

指定安装位置

执行先决条件检查

概要

安装产品

完成

请使用 Windows 内置帐户或指定标准 Windows 用户帐户 (非管理员帐户) 来安装和配置 Oracle 主目录。此帐户用于运行 Oracle 主目录的 Windows 服务。请勿使用此帐户登录来执行管理任务。

使用现有 Windows 用户(X)

用户名(U):

口令(P):

创建新 Windows 用户(C)

用户名(S):

口令(A):

确认口令(Q):

新创建的用户没有 Windows 登录权限。

使用 Windows 内置帐户(L)

帮助(H)

< 上一步(B)

下一步(N) >

安装(I)

取消



指定安装位置

- 选择安装类型
- 选择产品语言
- Oracle 主目录用户选择
- 指定安装位置**
- 执行先决条件检查
- 概要
- 安装产品
- 完成

指定用于放置所有 Oracle 软件以及与配置相关的文件的 Oracle 基目录路径。此位置是 Oracle 基目录。

Oracle 基目录(O): 浏览(B)...

指定用于存储 Oracle 软件文件的位置。此位置是 Oracle 主目录。

软件位置(S): 浏览(W)...

帮助(H)

< 上一步(B)

下一步(N) >

安装(I)

取消



概要

- 选择安装类型
- 选择产品语言
- Oracle 主目录用户选择
- 指定安装位置
- 执行先决条件检查

概要

- 安装产品
- 完成

Oracle Database Client 12c 发行版 1 安装程序

全局设置

- 磁盘空间: 要求 1.5 GB 可用 283.38 GB [\[编辑\]](#)
- 源位置: D:\工具\winx64_12102_client\clientinstall\stage\products.xml
- 安装类型: 管理员 [\[编辑\]](#)
- Oracle 基目录: D:\app\client\neo [\[编辑\]](#)
- Oracle 主目录位置: D:\app\client\neo\product\12.1.0\client_1 [\[编辑\]](#)
- Oracle 主目录用户选择: NT AUTHORITY\LOCAL SERVICE [\[编辑\]](#)

保存响应文件(S)...

帮助(H)

< 上一步(B)

下一步(N) >

安装(I)

取消



完成

- 选择安装类型
- 选择产品语言
- Oracle 主目录用户选择
- 指定安装位置
- 执行先决条件检查
- 概要
- 安装产品

完成

Oracle Client 的安装 已成功。

帮助(H)

< 上一步(B)

下一步(N) >

安装(I)

关闭(C)

5. 卸载 Oracle

卸载或者重装运行下面命令清理Oracle

```
# rm -rf /etc/oratab /etc/oraInst.loc  
# rm -rf /opt/oracle /opt/oraInventory /opt/ORCLfmap/
```

6. Linux 安装 sqlplus

<http://www.oracle.com/technetwork/topics/linuxx86-64soft-092277.html>

```
oracle-instantclient11.2-basic-11.2.0.4.0-1.x86_64.rpm
oracle-instantclient11.2-sqlplus-11.2.0.4.0-1.x86_64.rpm
```

安装

```
yum localinstall oracle-instantclient11.2-*
```

创建sqlnet.ora和tnsnames.ora两个文件

```
mkdir -p /usr/lib/oracle/11.2/client64/network/admin
cd /usr/lib/oracle/11.2/client64/network/admin

[root@localhost admin]# cat sqlnet.ora
NAMES DIRECTORY_PATH=(TNSNAMES, EZCONNECT)
ADR_BASE = /usr/lib/oracle/

[root@localhost admin]# cat tnsnames.ora
XXX =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 172.16.0.2)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = DB)
    )
  )
)
```

~/bash_profile

```
ORACLE_HOME=/usr/lib/oracle/11.2/client64
PATH=$ORACLE_HOME/bin:$PATH
LD_LIBRARY_PATH=$ORACLE_HOME/lib
export ORACLE_HOME
export LD_LIBRARY_PATH
export TNS_ADMIN=$ORACLE_HOME/network/admin
export PATH
```



```
sqlplus "username/password@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)
(HOST=192.168.2.1)(PORT=1521))(CONNECT_DATA=(SID=YOURSID)))"
```

第 74 章 Privilege

1. Account

```
select username,account_status from dba_users;  
  
select username,account_status from dba_users where  
account_status='OPEN';  
  
select username,account_status from dba_users t where  
t.default_tablespace = 'USERS' and t.account_status = 'OPEN';
```

1.1. show user

```
SQL> show user;  
USER is "SYS"  
  
SQL> select user from dual;  
  
USER  
-----  
SYS
```

1.2. DEFAULT_TABLESPACE

用户默认表空间

```
SQL> SELECT DEFAULT_TABLESPACE FROM DBA_USERS WHERE  
USERNAME='WCUSER';
```

```
DEFAULT_TABLESPACE
```

```
-----  
WCSDB
```

```
SQL> SELECT DEFAULT_TABLESPACE FROM DBA_USERS WHERE USERNAME=  
(select user from dual);
```

```
DEFAULT_TABLESPACE
```

```
-----  
SYSTEM
```

```
SQL> select username,account_status from dba_users t where  
t.default_tablespace = 'USERS' and t.account_status = 'OPEN';
```

```
USERNAME                                ACCOUNT_STATUS
```

```
-----  
--
```

```
FINANCE                                OPEN
```

```
NEWS                                    OPEN
```

1.3. unlock/lock

帐号加锁与解锁

alter user scott account unlock /lock;

```
SQL> alter user scott account unlock;
```

```
User altered.
```

```
SQL> select username,account_status from dba_users where  
username='SCOTT';
```

```
USERNAME                                ACCOUNT_STATUS
```

```
-----  
--
```

```
SCOTT                                    EXPIRED
```

```
SQL> alter user scott account lock;
```

User altered.

```
SQL> select username,account_status from dba_users where  
username='SCOTT';
```

USERNAME	ACCOUNT_STATUS
SCOTT	EXPIRED & LOCKED

```
SQL>
```

2. User

2.1. Create user

```
create user userName identified by password;
```

指定表空间

```
create user neo identified by 123456 default tablespace netkiller;
```

2.2. Drop user

```
drop user userName cascade;
```

cascade 删除用户同时删除该用户所创建的对象。

2.3. Alter user

```
alter user neo default tablespace neo;
```

2.4. 修改密码

输入 sqlplus /nolog 回车

```
$ sqlplus /nolog
SQL> conn /as sysdba
SQL> alter user system identified by "123456";
SQL> alter user sys identified by "123456";
SQL> alter user FINANCE identified by "123123" account unlock;
User altered.
```

account unlock 解锁账号

```
alter user backup identified by "passw0rd" account unlock;
```

2.5. Reset Password 忘记密码重置密码

密码文件 orapwORACLE_SID 这里的SID是orcl，所以密码文件是 orapworcl

```
orapwd file=$ORACLE_HOME/dbs/orapworcl password=newpass
```

3. grant

3.1. 列出权限

```
CREATE SESSION  登陆权限, 例如sqlplus
```

```
SQL> SELECT * FROM SYSTEM_PRIVILEGE_MAP;
```

PRIVILEGE NAME	PROPERTY
-3 ALTER SYSTEM	0
-4 AUDIT SYSTEM	0
-5 CREATE SESSION	0
-6 ALTER SESSION	0
-7 RESTRICTED SESSION	0
-10 CREATE TABLESPACE	0
-11 ALTER TABLESPACE	0
-12 MANAGE TABLESPACE	0
-13 DROP TABLESPACE	0
-15 UNLIMITED TABLESPACE	0
-20 CREATE USER	0
-21 BECOME USER	0
-22 ALTER USER	0
-23 DROP USER	0
-30 CREATE ROLLBACK SEGMENT	0
-31 ALTER ROLLBACK SEGMENT	0
-32 DROP ROLLBACK SEGMENT	0
-40 CREATE TABLE	0
-41 CREATE ANY TABLE	0
-42 ALTER ANY TABLE	0
-43 BACKUP ANY TABLE	0
-44 DROP ANY TABLE	0
-45 LOCK ANY TABLE	0
-46 COMMENT ANY TABLE	0
-47 SELECT ANY TABLE	0
-48 INSERT ANY TABLE	0
-49 UPDATE ANY TABLE	0
-50 DELETE ANY TABLE	0
-60 CREATE CLUSTER	0
-61 CREATE ANY CLUSTER	0
-62 ALTER ANY CLUSTER	0
-63 DROP ANY CLUSTER	0
-71 CREATE ANY INDEX	0
-72 ALTER ANY INDEX	0

-73	DROP ANY INDEX	0
-80	CREATE SYNONYM	0
-81	CREATE ANY SYNONYM	0
-82	DROP ANY SYNONYM	0
-83	SYSDBA	0
-84	SYSOPER	0
-85	CREATE PUBLIC SYNONYM	0
-86	DROP PUBLIC SYNONYM	0
-90	CREATE VIEW	0
-91	CREATE ANY VIEW	0
-92	DROP ANY VIEW	0
-105	CREATE SEQUENCE	0
-106	CREATE ANY SEQUENCE	0
-107	ALTER ANY SEQUENCE	0
-108	DROP ANY SEQUENCE	0
-109	SELECT ANY SEQUENCE	0
-115	CREATE DATABASE LINK	0
-120	CREATE PUBLIC DATABASE LINK	0
-121	DROP PUBLIC DATABASE LINK	0
-125	CREATE ROLE	0
-126	DROP ANY ROLE	0
-127	GRANT ANY ROLE	0
-128	ALTER ANY ROLE	0
-130	AUDIT ANY	0
-135	ALTER DATABASE	0
-138	FORCE TRANSACTION	0
-139	FORCE ANY TRANSACTION	0
-140	CREATE PROCEDURE	0
-141	CREATE ANY PROCEDURE	0
-142	ALTER ANY PROCEDURE	0
-143	DROP ANY PROCEDURE	0
-144	EXECUTE ANY PROCEDURE	0
-151	CREATE TRIGGER	0
-152	CREATE ANY TRIGGER	0
-153	ALTER ANY TRIGGER	0
-154	DROP ANY TRIGGER	0
-160	CREATE PROFILE	0
-161	ALTER PROFILE	0
-162	DROP PROFILE	0
-163	ALTER RESOURCE COST	0
-165	ANALYZE ANY	0
-167	GRANT ANY PRIVILEGE	0
-172	CREATE MATERIALIZED VIEW	0
-173	CREATE ANY MATERIALIZED VIEW	0
-174	ALTER ANY MATERIALIZED VIEW	0
-175	DROP ANY MATERIALIZED VIEW	0
-177	CREATE ANY DIRECTORY	0
-178	DROP ANY DIRECTORY	0
-180	CREATE TYPE	0
-181	CREATE ANY TYPE	0
-182	ALTER ANY TYPE	0

-183	DROP ANY TYPE	0
-184	EXECUTE ANY TYPE	0
-186	UNDER ANY TYPE	0
-188	CREATE LIBRARY	0
-189	CREATE ANY LIBRARY	0
-190	ALTER ANY LIBRARY	0
-191	DROP ANY LIBRARY	0
-192	EXECUTE ANY LIBRARY	0
-200	CREATE OPERATOR	0
-201	CREATE ANY OPERATOR	0
-202	ALTER ANY OPERATOR	0
-203	DROP ANY OPERATOR	0

PRIVILEGE	NAME	PROPERTY
-----------	------	----------

-204	EXECUTE ANY OPERATOR	0
-205	CREATE INDEXTYPE	0
-206	CREATE ANY INDEXTYPE	0
-207	ALTER ANY INDEXTYPE	0
-208	DROP ANY INDEXTYPE	0
-209	UNDER ANY VIEW	0
-210	QUERY REWRITE	0
-211	GLOBAL QUERY REWRITE	0
-212	EXECUTE ANY INDEXTYPE	0
-213	UNDER ANY TABLE	0
-214	CREATE DIMENSION	0
-215	CREATE ANY DIMENSION	0
-216	ALTER ANY DIMENSION	0
-217	DROP ANY DIMENSION	0
-218	MANAGE ANY QUEUE	1
-219	ENQUEUE ANY QUEUE	1
-220	DEQUEUE ANY QUEUE	1
-222	CREATE ANY CONTEXT	0
-223	DROP ANY CONTEXT	0
-224	CREATE ANY OUTLINE	0
-225	ALTER ANY OUTLINE	0
-226	DROP ANY OUTLINE	0
-227	ADMINISTER RESOURCE MANAGER	1
-228	ADMINISTER DATABASE TRIGGER	0
-233	MERGE ANY VIEW	0
-234	ON COMMIT REFRESH	0
-235	EXEMPT ACCESS POLICY	0
-236	RESUMABLE	0
-237	SELECT ANY DICTIONARY	0
-238	DEBUG CONNECT SESSION	0
-241	DEBUG ANY PROCEDURE	0
-243	FLASHBACK ANY TABLE	0
-244	GRANT ANY OBJECT PRIVILEGE	0
-245	CREATE EVALUATION CONTEXT	1
-246	CREATE ANY EVALUATION CONTEXT	1
-247	ALTER ANY EVALUATION CONTEXT	1

-248	DROP ANY EVALUATION CONTEXT	1
-249	EXECUTE ANY EVALUATION CONTEXT	1
-250	CREATE RULE SET	1
-251	CREATE ANY RULE SET	1
-252	ALTER ANY RULE SET	1
-253	DROP ANY RULE SET	1
-254	EXECUTE ANY RULE SET	1
-255	EXPORT FULL DATABASE	0
-256	IMPORT FULL DATABASE	0
-257	CREATE RULE	1
-258	CREATE ANY RULE	1
-259	ALTER ANY RULE	1
-260	DROP ANY RULE	1
-261	EXECUTE ANY RULE	1
-262	ANALYZE ANY DICTIONARY	0
-263	ADVISOR	0
-264	CREATE JOB	0
-265	CREATE ANY JOB	0
-266	EXECUTE ANY PROGRAM	0
-267	EXECUTE ANY CLASS	0
-268	MANAGE SCHEDULER	0
-269	SELECT ANY TRANSACTION	0
-270	DROP ANY SQL PROFILE	0
-271	ALTER ANY SQL PROFILE	0
-272	ADMINISTER SQL TUNING SET	0
-273	ADMINISTER ANY SQL TUNING SET	0
-274	CREATE ANY SQL PROFILE	0
-275	EXEMPT IDENTITY POLICY	0
-276	MANAGE FILE GROUP	1
-277	MANAGE ANY FILE GROUP	1
-278	READ ANY FILE GROUP	1
-279	CHANGE NOTIFICATION	0
-280	CREATE EXTERNAL JOB	0
-281	CREATE ANY EDITION	0
-282	DROP ANY EDITION	0
-283	ALTER ANY EDITION	0
-284	CREATE ASSEMBLY	0
-285	CREATE ANY ASSEMBLY	0
-286	ALTER ANY ASSEMBLY	0
-287	DROP ANY ASSEMBLY	0
-288	EXECUTE ANY ASSEMBLY	0
-289	EXECUTE ASSEMBLY	0
-290	CREATE MINING MODEL	0
-291	CREATE ANY MINING MODEL	0
-292	DROP ANY MINING MODEL	0
-293	SELECT ANY MINING MODEL	0
-294	ALTER ANY MINING MODEL	0
-295	COMMENT ANY MINING MODEL	0
-301	CREATE CUBE DIMENSION	0
-302	ALTER ANY CUBE DIMENSION	0
-303	CREATE ANY CUBE DIMENSION	0

-304	DELETE ANY CUBE DIMENSION	0
-305	DROP ANY CUBE DIMENSION	0
-306	INSERT ANY CUBE DIMENSION	0
-307	SELECT ANY CUBE DIMENSION	0
-308	CREATE CUBE	0
-309	ALTER ANY CUBE	0
-310	CREATE ANY CUBE	0
-311	DROP ANY CUBE	0
-312	SELECT ANY CUBE	0
-313	UPDATE ANY CUBE	0
PRIVILEGE NAME		PROPERTY

-314	CREATE MEASURE FOLDER	0
-315	CREATE ANY MEASURE FOLDER	0
-316	DELETE ANY MEASURE FOLDER	0
-317	DROP ANY MEASURE FOLDER	0
-318	INSERT ANY MEASURE FOLDER	0
-319	CREATE CUBE BUILD PROCESS	0
-320	CREATE ANY CUBE BUILD PROCESS	0
-321	DROP ANY CUBE BUILD PROCESS	0
-322	UPDATE ANY CUBE BUILD PROCESS	0
-326	UPDATE ANY CUBE DIMENSION	0
-327	ADMINISTER SQL MANAGEMENT OBJECT	0
-328	ALTER PUBLIC DATABASE LINK	0
-329	ALTER DATABASE LINK	0
-350	FLASHBACK ARCHIVE ADMINISTER	0

208 rows selected.

3.2. 查看用户权限

```
SELECT * FROM TABLE_PRIVILEGES;
SELECT * FROM DBA_TAB_PRIVS;
```

```
select * from dict where table_name like '%PRIV%';
```

```
SELECT grantee,privilege,admin_option FROM dba_sys_privs WHERE grantee
IN ('SCOTT','SYS','SYSTEM') ORDER BY grantee;
```

```
SQL> SELECT grantee,privilege,admin_option FROM dba_sys_privs WHERE
grantee IN ('SCOTT','SYS','SYSTEM') ORDER BY grantee;
```

GRANTEE	PRIVILEGE
ADM	

SCOTT	UNLIMITED TABLESPACE
NO	
SYS	ADMINISTER ANY SQL TUNING SET
NO	
SYS	ADMINISTER DATABASE TRIGGER
NO	
SYS	ADMINISTER RESOURCE MANAGER
NO	
SYS	ADMINISTER SQL MANAGEMENT OBJECT
NO	
SYS	ADMINISTER SQL TUNING SET
NO	
SYS	ADVISOR
NO	
SYS	ALTER ANY ASSEMBLY
NO	
SYS	ALTER ANY CLUSTER
NO	
SYS	ALTER ANY CUBE
NO	
SYS	ALTER ANY CUBE DIMENSION
NO	
SYS	ALTER ANY DIMENSION
NO	
SYS	ALTER ANY EDITION
NO	
SYS	ALTER ANY EVALUATION CONTEXT
YES	
SYS	ALTER ANY INDEX
NO	
SYS	ALTER ANY INDEXTYPE
NO	
SYS	ALTER ANY LIBRARY
NO	
SYS	ALTER ANY MATERIALIZED VIEW
NO	
SYS	ALTER ANY MINING MODEL
NO	
SYS	ALTER ANY OPERATOR
NO	
SYS	ALTER ANY OUTLINE
NO	
SYS	ALTER ANY PROCEDURE
NO	
SYS	ALTER ANY ROLE
NO	

SYS	ALTER ANY RULE
YES	
SYS	ALTER ANY RULE SET
YES	
SYS	ALTER ANY SEQUENCE
NO	
SYS	ALTER ANY SQL PROFILE
NO	
SYS	ALTER ANY TABLE
NO	
SYS	ALTER ANY TRIGGER
NO	
SYS	ALTER ANY TYPE
NO	
SYS	ALTER DATABASE
NO	
SYS	ALTER PROFILE
NO	
SYS	ALTER RESOURCE COST
NO	
SYS	ALTER ROLLBACK SEGMENT
NO	
SYS	ALTER SESSION
NO	
SYS	ALTER SYSTEM
NO	
SYS	ALTER TABLESPACE
NO	
SYS	ALTER USER
NO	
SYS	ANALYZE ANY
NO	
SYS	AUDIT ANY
NO	
SYS	AUDIT SYSTEM
NO	
SYS	BACKUP ANY TABLE
NO	
SYS	BECOME USER
NO	
SYS	CHANGE NOTIFICATION
NO	
SYS	COMMENT ANY MINING MODEL
NO	
SYS	COMMENT ANY TABLE
NO	
SYS	CREATE ANY ASSEMBLY
NO	
SYS	CREATE ANY CLUSTER
NO	
SYS	CREATE ANY CONTEXT

NO	
SYS	CREATE ANY CUBE
NO	
SYS	CREATE ANY CUBE BUILD PROCESS
NO	
SYS	CREATE ANY CUBE DIMENSION
NO	
SYS	CREATE ANY DIMENSION
NO	
SYS	CREATE ANY DIRECTORY
NO	
SYS	CREATE ANY EDITION
NO	
SYS	CREATE ANY EVALUATION CONTEXT
YES	
SYS	CREATE ANY INDEX
NO	
SYS	CREATE ANY INDEXTYPE
NO	
SYS	CREATE ANY JOB
NO	
SYS	CREATE ANY LIBRARY
NO	
SYS	CREATE ANY MATERIALIZED VIEW
NO	
SYS	CREATE ANY MEASURE FOLDER
NO	
SYS	CREATE ANY MINING MODEL
NO	
SYS	CREATE ANY OPERATOR
NO	
SYS	CREATE ANY OUTLINE
NO	
SYS	CREATE ANY PROCEDURE
NO	
SYS	CREATE ANY RULE
YES	
SYS	CREATE ANY RULE SET
YES	
SYS	CREATE ANY SEQUENCE
NO	
SYS	CREATE ANY SQL PROFILE
NO	
SYS	CREATE ANY SYNONYM
NO	
SYS	CREATE ANY TABLE
NO	
SYS	CREATE ANY TRIGGER
NO	
SYS	CREATE ANY TYPE
NO	

SYS	CREATE ANY VIEW
NO	
SYS	CREATE ASSEMBLY
NO	
SYS	CREATE CLUSTER
NO	
SYS	CREATE CUBE
NO	
SYS	CREATE CUBE BUILD PROCESS
NO	
SYS	CREATE CUBE DIMENSION
NO	
SYS	CREATE DATABASE LINK
NO	
SYS	CREATE DIMENSION
NO	
SYS	CREATE EVALUATION CONTEXT
YES	
SYS	CREATE EXTERNAL JOB
NO	
SYS	CREATE INDEXTYPE
NO	
SYS	CREATE JOB
NO	
SYS	CREATE LIBRARY
NO	
SYS	CREATE MATERIALIZED VIEW
NO	
SYS	CREATE MEASURE FOLDER
NO	
SYS	CREATE MINING MODEL
NO	
SYS	CREATE OPERATOR
NO	
SYS	CREATE PROCEDURE
NO	
SYS	CREATE PROFILE
NO	
SYS	CREATE PUBLIC DATABASE LINK
NO	
SYS	CREATE PUBLIC SYNONYM
NO	
SYS	CREATE ROLE
NO	
SYS	CREATE ROLLBACK SEGMENT
NO	
GRANTEE	PRIVILEGE
ADM	

SYS	CREATE RULE
YES	
SYS	CREATE RULE SET
YES	
SYS	CREATE SEQUENCE
NO	
SYS	CREATE SESSION
NO	
SYS	CREATE SYNONYM
NO	
SYS	CREATE TABLE
NO	
SYS	CREATE TABLESPACE
NO	
SYS	CREATE TRIGGER
NO	
SYS	CREATE TYPE
NO	
SYS	CREATE USER
NO	
SYS	CREATE VIEW
NO	
SYS	DEBUG ANY PROCEDURE
NO	
SYS	DEBUG CONNECT SESSION
NO	
SYS	DELETE ANY CUBE DIMENSION
NO	
SYS	DELETE ANY MEASURE FOLDER
NO	
SYS	DELETE ANY TABLE
NO	
SYS	DEQUEUE ANY QUEUE
YES	
SYS	DROP ANY ASSEMBLY
NO	
SYS	DROP ANY CLUSTER
NO	
SYS	DROP ANY CONTEXT
NO	
SYS	DROP ANY CUBE
NO	
SYS	DROP ANY CUBE BUILD PROCESS
NO	
SYS	DROP ANY CUBE DIMENSION
NO	
SYS	DROP ANY DIMENSION
NO	
SYS	DROP ANY DIRECTORY
NO	
SYS	DROP ANY EDITION

NO	
SYS	DROP ANY EVALUATION CONTEXT
YES	
SYS	DROP ANY INDEX
NO	
SYS	DROP ANY INDEXTYPE
NO	
SYS	DROP ANY LIBRARY
NO	
SYS	DROP ANY MATERIALIZED VIEW
NO	
SYS	DROP ANY MEASURE FOLDER
NO	
SYS	DROP ANY MINING MODEL
NO	
SYS	DROP ANY OPERATOR
NO	
SYS	DROP ANY OUTLINE
NO	
SYS	DROP ANY PROCEDURE
NO	
SYS	DROP ANY ROLE
NO	
SYS	DROP ANY RULE
YES	
SYS	DROP ANY RULE SET
YES	
SYS	DROP ANY SEQUENCE
NO	
SYS	DROP ANY SQL PROFILE
NO	
SYS	DROP ANY SYNONYM
NO	
SYS	DROP ANY TABLE
NO	
SYS	DROP ANY TRIGGER
NO	
SYS	DROP ANY TYPE
NO	
SYS	DROP ANY VIEW
NO	
SYS	DROP PROFILE
NO	
SYS	DROP PUBLIC DATABASE LINK
NO	
SYS	DROP PUBLIC SYNONYM
NO	
SYS	DROP ROLLBACK SEGMENT
NO	
SYS	DROP TABLESPACE
NO	

SYS	DROP USER
NO	
SYS	ENQUEUE ANY QUEUE
YES	
SYS	EXECUTE ANY ASSEMBLY
NO	
SYS	EXECUTE ANY CLASS
NO	
SYS	EXECUTE ANY EVALUATION CONTEXT
YES	
SYS	EXECUTE ANY INDEXTYPE
NO	
SYS	EXECUTE ANY LIBRARY
NO	
SYS	EXECUTE ANY OPERATOR
NO	
SYS	EXECUTE ANY PROCEDURE
NO	
SYS	EXECUTE ANY PROGRAM
NO	
SYS	EXECUTE ANY RULE
YES	
SYS	EXECUTE ANY RULE SET
YES	
SYS	EXECUTE ANY TYPE
NO	
SYS	EXECUTE ASSEMBLY
NO	
SYS	EXPORT FULL DATABASE
NO	
SYS	FLASHBACK ANY TABLE
NO	
SYS	FLASHBACK ARCHIVE ADMINISTER
NO	
SYS	FORCE ANY TRANSACTION
NO	
SYS	FORCE TRANSACTION
NO	
SYS	GLOBAL QUERY REWRITE
NO	
SYS	GRANT ANY OBJECT PRIVILEGE
NO	
SYS	GRANT ANY PRIVILEGE
NO	
SYS	GRANT ANY ROLE
NO	
SYS	IMPORT FULL DATABASE
NO	
SYS	INSERT ANY CUBE DIMENSION
NO	
SYS	INSERT ANY MEASURE FOLDER

NO	
SYS	INSERT ANY TABLE
NO	
SYS	LOCK ANY TABLE
NO	
SYS	MANAGE ANY FILE GROUP
NO	
SYS	MANAGE ANY QUEUE
YES	
SYS	MANAGE FILE GROUP
NO	
SYS	MANAGE SCHEDULER
NO	
SYS	MANAGE TABLESPACE
NO	
SYS	MERGE ANY VIEW
NO	
SYS	ON COMMIT REFRESH
NO	
SYS	QUERY REWRITE
NO	
SYS	READ ANY FILE GROUP
NO	
SYS	RESTRICTED SESSION
NO	
SYS	RESUMABLE
NO	
SYS	SELECT ANY CUBE
NO	
SYS	SELECT ANY CUBE DIMENSION
NO	
SYS	SELECT ANY MINING MODEL
NO	
SYS	SELECT ANY SEQUENCE
NO	
SYS	SELECT ANY TABLE
YES	
SYS	SELECT ANY TRANSACTION
NO	
SYS	UNDER ANY TABLE
NO	
GRANTEE	PRIVILEGE
ADM	

SYS	UNDER ANY TYPE
NO	
SYS	UNDER ANY VIEW
NO	
SYS	UNLIMITED TABLESPACE

```

NO
SYS                UPDATE ANY CUBE
NO
SYS                UPDATE ANY CUBE BUILD PROCESS
NO
SYS                UPDATE ANY CUBE DIMENSION
NO
SYS                UPDATE ANY TABLE
NO
SYSTEM            CREATE MATERIALIZED VIEW
NO
SYSTEM            CREATE TABLE
NO
SYSTEM            GLOBAL QUERY REWRITE
NO
SYSTEM            SELECT ANY TABLE
NO
SYSTEM            UNLIMITED TABLESPACE
YES

206 rows selected.

```

select grantee,privilege,admin_option from dba_sys_privs where grantee in ('SCOTT','SYS') and privilege = 'EXECUTE ANY PROCEDURE' order by grantee;

```

SQL> select grantee,privilege,admin_option from dba_sys_privs where
grantee in ('SCOTT','SYS') and privilege = 'EXECUTE ANY PROCEDURE'
order by grantee;

```

```

GRANTEE                PRIVILEGE
ADM
-----
SYS                EXECUTE ANY PROCEDURE
NO

```

3.3. 吊销权限

```

REVOKE EXECUTE ANY PROCEDURE FROM scott;

```

3.4. System Privileges 系统权限

授予DBA权限，不限制的表空间，查询任何表，查询任何字典

```
grant dba to neo;  
grant unlimited tablespace to neo;  
grant select any table to neo;  
grant select any dictionary to neo;
```

EM 权限

```
grant connect, resource, select_catalog_role to username;
```

Granting and Revoking SYSDBA and SYSOPER Privileges

Granting and Revoking SYSDBA and SYSOPER Privileges

If your server is using an EXCLUSIVE password file, use the GRANT statement to grant the SYSDBA or SYSOPER system privilege to a user, as shown in the following example:

```
GRANT SYSDBA TO user;
```

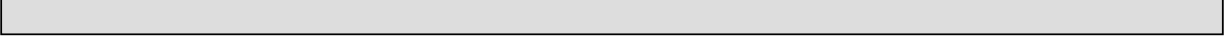
Use the REVOKE statement to revoke the SYSDBA or SYSOPER system privilege from a user, as shown in the following example:

```
REVOKE SYSDBA FROM user;
```

SYSDBA, SYSOPER, SYSASM

```
REVOKE SYSOPER FROM non-SYS-user;  
GRANT SYSOPER TO non-SYS-user;
```

3.5. Object Privileges 对象权限



4. Opening the Encrypted Wallet

```
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Mon Feb 1 15:53:57 2016

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> select * from v$encryption_wallet;

WRL_TYPE
-----
WRL_PARAMETER
-----
STATUS
-----
file
/opt/oracle/admin/orcl/wallet
CLOSED
```

```
          $ orapki wallet create -wallet /opt/oracle/admin/orcl/wallet -
auto_login_local
Oracle PKI Tool : Version 11.2.0.1.0 - Production
Copyright (c) 2004, 2009, Oracle and/or its affiliates. All rights reserved.

Enter password:
Enter password again:

[oracle@orcl ~]$ ll /opt/oracle/admin/orcl/wallet
total 8
-rw----- 1 oracle oinstall 3589 Feb  1 15:58 cwallet.sso
-rw----- 1 oracle oinstall 3512 Feb  1 15:58 ewallet.pl2
```

```
SQL> select * from v$encryption_wallet;

WRL_TYPE
-----
WRL_PARAMETER
-----
STATUS
```

```
-----  
file  
/opt/oracle/admin/orcl/wallet  
OPEN_NO_MASTER_KEY
```

The Wallet must be open before opening the database:

```
SQL> startup mount;  
SQL> alter system set wallet open identified by "password";  
SQL> alter database open;
```

待续.....

第 75 章 数据库管理

1. lsnrctl

启动监听服务:

```
lsnrctl start
```

停止监听服务:

```
lsnrctl stop
```

查看监听状态:

```
lsnrctl status
```

2. listener.ora

```
[root@database ~]# cat
/u01/app/oracle/product/10.2.0.1/network/admin/listener.ora
# listener.ora Network Configuration File:
/u01/app/oracle/product/10.2.0.1/network/admin/listener.ora
# Generated by Oracle configuration tools.

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = PLSExtProc)
      (ORACLE_HOME = /u01/app/oracle/product/10.2.0.1)
      (PROGRAM = extproc)
    )
  )

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1))
      (ADDRESS = (PROTOCOL = TCP)(HOST = database.example.com)
(PORT = 1521))
    )
  )
```

3. TNS 配置

tnsnames.ora 文件默认在 network/admin/tnsnames.ora

有些情况如你没有权限修改network/admin/tnsnames.ora, 你可以在 \$HOME下创建一个.tnsnames.ora文件

```
[oracle@orcl admin]$ cat tnsnames.ora
# tnsnames.ora Network Configuration File:
/opt/oracle/product/11.2.0/dbhome_1/network/admin/tnsnames.ora
# Generated by Oracle configuration tools.

ORCL =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = oral.example.com)(PORT =
1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl.example.com)
    )
  )
)
```

测试TNS

```
$ sqlplus sys/chen@orcl
```

3.1. HOST

HOST可以使用IP地址 HOST = 192.168.0.5 建议改为hostname 例如 HOST = oral.example.com这样方便服务器更换IP。

```
ORCL =
  (DESCRIPTION =
```

```
(ADDRESS_LIST =
  (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.0.5)(PORT =
1521))
)
(CONNECT_DATA =
  (SERVICE_NAME = orcl.example.com)
)
)
```

SERVICE_NAME通過show parameter service_name;查詢

```
$ sqlplus user@orcl
```

3.2. SID

Oracle 10G 之前多采用 SID = oradb

10G之后更多使用 SERVICE_NAME = orcl.example.com

```
oradb10g =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = db1.domain.com)(PORT =
1521))
    )
    (CONNECT_DATA =
      (SID = oradb10g)
    )
  )

oradb =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = db2.domain.com)(PORT =
1521))
    )
    (CONNECT_DATA =
      (SID = oradb)
    )
  )
```

)
)

4. parameter

4.1. db

```
SQL> show parameter db;
```

NAME	TYPE	VALUE
db_16k_cache_size	big integer	0
db_2k_cache_size	big integer	0
db_32k_cache_size	big integer	0
db_4k_cache_size	big integer	0
db_8k_cache_size	big integer	0
db_block_buffers	integer	0
db_block_checking	string	FALSE
db_block_checksum	string	TYPICAL
db_block_size	integer	8192
db_cache_advice	string	ON
db_cache_size	big integer	0

NAME	TYPE	VALUE
db_create_file_dest	string	
db_create_online_log_dest_1	string	
db_create_online_log_dest_2	string	
db_create_online_log_dest_3	string	
db_create_online_log_dest_4	string	
db_create_online_log_dest_5	string	
db_domain	string	example.com
db_file_multiblock_read_count	integer	128
db_file_name_convert	string	
db_files	integer	200
db_flash_cache_file	string	

NAME	TYPE	VALUE
db_flash_cache_size	big integer	0

db_flashback_retention_target	integer	1440
db_keep_cache_size	big integer	0
db_lost_write_protect	string	NONE
db_name	string	orcl
db_recovery_file_dest	string	
/opt/oracle/flash_recovery_are		
		a
db_recovery_file_dest_size	big integer	3882M
db_recycle_cache_size	big integer	0
db_securefile	string	PERMITTED
db_ultra_safe	string	OFF
NAME	TYPE	VALUE
-----	-----	-----
db_unique_name	string	orcl
db_writer_processes	integer	1
dbwr_io_slaves	integer	0
rdbms_server_dn	string	
standby_archive_dest	string	?/dbs/arch
standby_file_management	string	MANUAL
xml_db_events	string	enable

4.2. instance_name

```
SQL> show parameter instance_name;
```

NAME	TYPE	VALUE
-----	-----	-----
instance_name	string	orcl

```
SQL>
```

```
SQL> select instance from v$thread;
```

INSTANCE

orcl

```
orcl
```

4.3. service_name

```
SQL> show parameter service_name;
```

NAME	TYPE	VALUE
service_names	string	orcl.example.com

```
SQL>
```

4.4. global_name

```
SQL> select * from global_name;
```

GLOBAL_NAME
ORCL.EXAMPLE.COM

4.5. db_name

```
SQL> show parameter db_name;
```

NAME	TYPE	VALUE
db_name	string	orcl


```
SQL> select name from v$database;
```

```
NAME
```

```
-----
```

```
ORCL
```

4.6. db_domain

```
SQL> show parameter db_domain;
```

```
NAME
```

```
TYPE
```

```
VALUE
```

```
-----
```

```
db_domain
```

```
string
```

```
example.com
```

```
SQL>
```

4.7. sga

```
SQL> show parameter sga;
```

```
NAME
```

```
TYPE
```

```
VALUE
```

```
-----
```

```
lock_sga
```

```
boolean
```

```
FALSE
```

```
pre_page_sga
```

```
boolean
```

```
FALSE
```

```
sga_max_size
```

```
big integer 6016M
```

```
sga_target
```

```
big integer 0
```

4.8. size

```
SQL> show parameter size
```

NAME	TYPE	VALUE
-----	-----	-----
bitmap_merge_area_size	integer	1048576
client_result_cache_size	big integer	0
create_bitmap_area_size	integer	8388608
db_16k_cache_size	big integer	0
db_2k_cache_size	big integer	0
db_32k_cache_size	big integer	0
db_4k_cache_size	big integer	0
db_8k_cache_size	big integer	0
db_block_size	integer	8192
db_cache_size	big integer	0
db_flash_cache_size	big integer	0
NAME	TYPE	VALUE
-----	-----	-----
db_keep_cache_size	big integer	0
db_recovery_file_dest_size	big integer	3882M
db_recycle_cache_size	big integer	0
global_context_pool_size	string	
hash_area_size	integer	131072
java_max_sessionspace_size	integer	0
java_pool_size	big integer	0
large_pool_size	big integer	0
max_dump_file_size	string	unlimited
object_cache_max_size_percent	integer	10
object_cache_optimal_size	integer	102400
NAME	TYPE	VALUE
-----	-----	-----
olap_page_pool_size	big integer	0
parallel_execution_message_size	integer	16384
result_cache_max_size	big integer	16064K
sga_max_size	big integer	6272M
shared_pool_reserved_size	big integer	36909875
shared_pool_size	big integer	0
sort_area_retained_size	integer	0
sort_area_size	integer	65536

streams_pool_size	big integer	0
workarea_size_policy	string	AUTO

4.9. spfile

```
SQL> show parameter spfile ;
```

NAME	TYPE	VALUE
-----	-----	-----
spfile	string	
/opt/oracle/product/11.2.0/dbh		
ome_1/dbs/spfilewcsdb.ora		

4.10. cache

```
SQL> show parameter cache
```

NAME	TYPE	VALUE
-----	-----	-----
client_result_cache_lag	big integer	3000
client_result_cache_size	big integer	0
db_16k_cache_size	big integer	0
db_2k_cache_size	big integer	0
db_32k_cache_size	big integer	0
db_4k_cache_size	big integer	0
db_8k_cache_size	big integer	0
db_cache_advice	string	ON
db_cache_size	big integer	0
db_flash_cache_file	string	
db_flash_cache_size	big integer	0

NAME	TYPE	VALUE
db_keep_cache_size	big integer	0
db_recycle_cache_size	big integer	0
object_cache_max_size_percent	integer	10
object_cache_optimal_size	integer	102400
result_cache_max_result	integer	5
result_cache_max_size	big integer	16064K
result_cache_mode	string	MANUAL
result_cache_remote_expiration	integer	0
session_cached_cursors	integer	50

4.11. Character Set

```
SQL> select * from V$NLS_VALID_VALUES where
parameter='CHARACTERSET' and value like '%UTF%';
```

PARAMETER	VALUE	CON_ID
CHARACTERSET	AL24UTFFSS	TRUE 0
CHARACTERSET	UTF8	FALSE 0
CHARACTERSET	UTFE	FALSE 0
CHARACTERSET	AL32UTF8	FALSE 0
CHARACTERSET	AL16UTF16	FALSE 0

```
SQL> select userenv('language') from dual;
```

```
USERENV('LANGUAGE')
```

```
-----  
AMERICAN_AMERICA.WE8MSWIN1252
```

```
SQL> select * from nls_database_parameters ;
```

```
PARAMETER
```

```
VALUE
```

```
-----  
-----  
-----  
-----  
NLS_RDBMS_VERSION
```

```
12.1.0.2.0
```

```
NLS_NCHAR_CONV_EXCP
```

```
FALSE
```

```
NLS_LENGTH_SEMANTICS
```

```
BYTE
```

```
NLS_COMP
```

```
BINARY
```

```
NLS_DUAL_CURRENCY
```

```
$
```

```
NLS_TIMESTAMP_TZ_FORMAT
```

```
DD-MON-RR HH.MI.SSXFF AM TZR
```

```
NLS_TIME_TZ_FORMAT
```

```
HH.MI.SSXFF AM TZR
```

```
NLS_TIMESTAMP_FORMAT
```

```
DD-MON-RR HH.MI.SSXFF AM
```

```
NLS_TIME_FORMAT
```

```
HH.MI.SSXFF AM
```

```
NLS_SORT
```

```
BINARY
```

```
NLS_DATE_LANGUAGE
```

```
AMERICAN
```

```
NLS_DATE_FORMAT
DD-MON-RR
NLS_CALENDAR
GREGORIAN
NLS_NUMERIC_CHARACTERS
.,
NLS_NCHAR_CHARACTERSET
AL16UTF16
NLS_CHARACTERSET
WE8MSWIN1252
NLS_ISO_CURRENCY
AMERICA
NLS_CURRENCY
$
NLS_TERRITORY
AMERICA
NLS_LANGUAGE
AMERICAN

20 rows selected.
```

```
SQL> SELECT * FROM v$nls_parameters;
```

```
PARAMETER
VALUE
CON_ID
```

```
-----
-----
-----
NLS_LANGUAGE
AMERICAN
0
NLS_TERRITORY
AMERICA
0
NLS_CURRENCY
$
0
NLS_ISO_CURRENCY
AMERICA
0
```

```
NLS_NUMERIC_CHARACTERS
.,
0
NLS_CALENDAR
GREGORIAN
0
NLS_DATE_FORMAT
DD-MON-RR
0
NLS_DATE_LANGUAGE
AMERICAN
0
NLS_CHARACTERSET
WE8MSWIN1252
0
NLS_SORT
BINARY
0
NLS_TIME_FORMAT
HH.MI.SSXFF AM
0
NLS_TIMESTAMP_FORMAT
DD-MON-RR HH.MI.SSXFF AM
0
NLS_TIME_TZ_FORMAT
HH.MI.SSXFF AM TZR
0
NLS_TIMESTAMP_TZ_FORMAT
DD-MON-RR HH.MI.SSXFF AM TZR
0
NLS_DUAL_CURRENCY
$
0
NLS_NCHAR_CHARACTERSET
AL16UTF16
0
NLS_COMP
BINARY
0
NLS_LENGTH_SEMANTICS
BYTE
0
NLS_NCHAR_CONV_EXCP
FALSE
0
```

19 rows selected.

```
SQL> SELECT * FROM v$nls_valid_values;
```

PARAMETER	VALUE	ISDEP	CON_ID
LANGUAGE AMERICAN	FALSE	0	
LANGUAGE GERMAN			FALSE
0			
LANGUAGE FRENCH			FALSE
0			
LANGUAGE CANADIAN FRENCH			FALSE
0			
LANGUAGE SPANISH			FALSE
0			
LANGUAGE ITALIAN			FALSE
0			
LANGUAGE DUTCH	FALSE	0	
LANGUAGE SWEDISH			FALSE
0			
LANGUAGE NORWEGIAN	FALSE	0	
LANGUAGE DANISH			FALSE
0			

LANGUAGE		
FINNISH		FALSE
0		
LANGUAGE		
ICELANDIC		
FALSE	0	
LANGUAGE		
GREEK		
FALSE	0	
LANGUAGE		
PORTUGUESE		
FALSE	0	
LANGUAGE		
TURKISH		FALSE
0		
LANGUAGE		
BRAZILIAN PORTUGUESE		
FALSE	0	
LANGUAGE		
MEXICAN SPANISH		FALSE
0		
LANGUAGE		
RUSSIAN		FALSE
0		
LANGUAGE		
POLISH		FALSE
0		
LANGUAGE		
HUNGARIAN		
FALSE	0	
LANGUAGE		
CZECH		
FALSE	0	
LANGUAGE		
LITHUANIAN		
FALSE	0	
LANGUAGE		
SLOVAK		FALSE
0		
LANGUAGE		
CATALAN		FALSE
0		
LANGUAGE		
BULGARIAN		
FALSE	0	

LANGUAGE		
ROMANIAN		
FALSE	0	
LANGUAGE		
SLOVENIAN		
FALSE	0	
LANGUAGE		
HEBREW		FALSE
0		
LANGUAGE		
EGYPTIAN		
FALSE	0	
LANGUAGE		
CROATIAN		
FALSE	0	
LANGUAGE		
ARABIC		FALSE
0		
LANGUAGE		
THAI		
FALSE	0	
LANGUAGE		
JAPANESE		
FALSE	0	
LANGUAGE		
KOREAN		FALSE
0		
LANGUAGE		
SIMPLIFIED CHINESE		
FALSE	0	
LANGUAGE		
TRADITIONAL CHINESE		
FALSE	0	
LANGUAGE		
ENGLISH		FALSE
0		
LANGUAGE		
LATIN AMERICAN SPANISH		FALSE
0		
LANGUAGE		
UKRAINIAN		
FALSE	0	
LANGUAGE		
ESTONIAN		
FALSE	0	

LANGUAGE		
GERMAN DIN		
FALSE	0	
LANGUAGE		
MALAY		
FALSE	0	
LANGUAGE		
VIETNAMESE		
FALSE	0	
LANGUAGE		
BENGALI		TRUE
0		
LANGUAGE		
LATVIAN		FALSE
0		
LANGUAGE		
INDONESIAN		
FALSE	0	
LANGUAGE		
HINDI		
FALSE	0	
LANGUAGE		
TAMIL		
FALSE	0	
LANGUAGE		
KANNADA		FALSE
0		
LANGUAGE		
TELUGU		FALSE
0		
LANGUAGE		
ORIYA		
FALSE	0	
LANGUAGE		
MALAYALAM		
FALSE	0	
LANGUAGE		
ASSAMESE		
FALSE	0	
LANGUAGE		
GUJARATI		
FALSE	0	
LANGUAGE		
MARATHI		FALSE
0		

LANGUAGE PUNJABI 0	FALSE
LANGUAGE BANGLA 0	FALSE
LANGUAGE AZERBAIJANI FALSE 0	
LANGUAGE MACEDONIAN FALSE 0	
LANGUAGE CYRILLIC SERBIAN FALSE 0	
LANGUAGE LATIN SERBIAN FALSE 0	
LANGUAGE CYRILLIC UZBEK 0	FALSE
LANGUAGE LATIN UZBEK FALSE 0	
LANGUAGE CYRILLIC KAZAKH 0	FALSE
LANGUAGE ALBANIAN FALSE 0	
LANGUAGE BELARUSIAN FALSE 0	
LANGUAGE IRISH FALSE 0	
LANGUAGE SWAHILI 0	FALSE
LANGUAGE DARI FALSE 0	
LANGUAGE LATIN BOSNIAN FALSE 0	

LANGUAGE		
AMHARIC		FALSE
0		
LANGUAGE		
LAO		
FALSE	0	
LANGUAGE		
MALTESE		FALSE
0		
LANGUAGE		
NEPALI		FALSE
0		
LANGUAGE		
ARMENIAN		
FALSE	0	
LANGUAGE		
KHMER		
FALSE	0	
LANGUAGE		
PERSIAN		FALSE
0		
LANGUAGE		
DIVEHI		FALSE
0		
LANGUAGE		
SINHALA		FALSE
0		
TERRITORY		
AMERICA		FALSE
0		
TERRITORY		
UNITED KINGDOM		FALSE
0		
TERRITORY		
GERMANY		FALSE
0		
TERRITORY		
FRANCE		FALSE
0		
TERRITORY		
CANADA		FALSE
0		
TERRITORY		
SPAIN		
FALSE	0	

TERRITORY		
ITALY		
FALSE	0	
TERRITORY		
THE NETHERLANDS		FALSE
0		
TERRITORY		
SWEDEN		FALSE
0		
TERRITORY		
NORWAY		FALSE
0		
TERRITORY		
DENMARK		FALSE
0		
TERRITORY		
FINLAND		FALSE
0		
TERRITORY		
ICELAND		FALSE
0		
TERRITORY		
GREECE		FALSE
0		
TERRITORY		
PORTUGAL		
FALSE	0	
TERRITORY		
TURKEY		FALSE
0		
TERRITORY		
BRAZIL		FALSE
0		
TERRITORY		
MEXICO		FALSE
0		
PARAMETER		
VALUE		
ISDEP	CON_ID	

- - - - -		

TERRITORY		
CIS		

TRUE	0	
TERRITORY		
CROATIA		FALSE
0		
TERRITORY		
POLAND		FALSE
0		
TERRITORY		
HUNGARY		FALSE
0		
TERRITORY		
CZECHOSLOVAKIA		TRUE
0		
TERRITORY		
LITHUANIA		
FALSE	0	
TERRITORY		
ISRAEL		FALSE
0		
TERRITORY		
BULGARIA		
FALSE	0	
TERRITORY		
ALGERIA		FALSE
0		
TERRITORY		
BAHRAIN		FALSE
0		
TERRITORY		
CATALONIA		
FALSE	0	
TERRITORY		
EGYPT		
FALSE	0	
TERRITORY		
IRAQ		
FALSE	0	
TERRITORY		
JORDAN		FALSE
0		
TERRITORY		
KUWAIT		FALSE
0		
TERRITORY		
LEBANON		FALSE

0		
TERRITORY		
LIBYA		
FALSE	0	
TERRITORY		
MOROCCO		FALSE
0		
TERRITORY		
MAURITANIA		
FALSE	0	
TERRITORY		
OMAN		
FALSE	0	
TERRITORY		
QATAR		
FALSE	0	
TERRITORY		
ROMANIA		FALSE
0		
TERRITORY		
SAUDI ARABIA		
FALSE	0	
TERRITORY		
SOMALIA		FALSE
0		
TERRITORY		
SYRIA		
FALSE	0	
TERRITORY		
DJIBOUTI		
FALSE	0	
TERRITORY		
SLOVENIA		
FALSE	0	
TERRITORY		
TUNISIA		FALSE
0		
TERRITORY		
YEMEN		
FALSE	0	
TERRITORY		
SUDAN		
FALSE	0	
TERRITORY		
SWITZERLAND		

FALSE	0	
TERRITORY		
AUSTRIA		FALSE
0		
TERRITORY		
UNITED ARAB EMIRATES		
FALSE	0	
TERRITORY		
THAILAND		
FALSE	0	
TERRITORY		
CHINA		
FALSE	0	
TERRITORY		
HONG KONG		
FALSE	0	
TERRITORY		
JAPAN		
FALSE	0	
TERRITORY		
KOREA		
FALSE	0	
TERRITORY		
TAIWAN		FALSE
0		
TERRITORY		
CZECH REPUBLIC		FALSE
0		
TERRITORY		
SLOVAKIA		
FALSE	0	
TERRITORY		
UKRAINE		FALSE
0		
TERRITORY		
ESTONIA		FALSE
0		
TERRITORY		
MALAYSIA		
FALSE	0	
TERRITORY		
BANGLADESH		
FALSE	0	
TERRITORY		
LATVIA		FALSE

0		
TERRITORY		
VIETNAM		FALSE
0		
TERRITORY		
INDONESIA		
FALSE	0	
TERRITORY		
CYPRUS		FALSE
0		
TERRITORY		
AUSTRALIA		
FALSE	0	
TERRITORY		
KAZAKHSTAN		
FALSE	0	
TERRITORY		
UZBEKISTAN		
FALSE	0	
TERRITORY		
SINGAPORE		
FALSE	0	
TERRITORY		
SOUTH AFRICA		
FALSE	0	
TERRITORY		
IRELAND		FALSE
0		
TERRITORY		
BELGIUM		FALSE
0		
TERRITORY		
LUXEMBOURG		
FALSE	0	
TERRITORY		
NEW ZEALAND		
FALSE	0	
TERRITORY		
INDIA		
FALSE	0	
TERRITORY		
CHILE		
FALSE	0	
TERRITORY		
COLOMBIA		

FALSE	0	
TERRITORY		
COSTA RICA		
FALSE	0	
TERRITORY		
EL SALVADOR		
FALSE	0	
TERRITORY		
GUATEMALA		
FALSE	0	
TERRITORY		
NICARAGUA		
FALSE	0	
TERRITORY		
PANAMA		FALSE
0		
TERRITORY		
PERU		
FALSE	0	
TERRITORY		
PUERTO RICO		
FALSE	0	
TERRITORY		
VENEZUELA		
FALSE	0	
TERRITORY		
YUGOSLAVIA		
TRUE	0	
TERRITORY		
MACEDONIA		
TRUE	0	
TERRITORY		
RUSSIA		FALSE
0		
TERRITORY		
AZERBAIJAN		
FALSE	0	
TERRITORY		
FYR MACEDONIA		
FALSE	0	
TERRITORY		
SERBIA AND MONTENEGRO		
TRUE	0	
TERRITORY		
ARGENTINA		

FALSE	0	
TERRITORY		
ECUADOR		FALSE
0		
TERRITORY		
PHILIPPINES		
FALSE	0	
TERRITORY		
ALBANIA		FALSE
0		
TERRITORY		
BELARUS		FALSE
0		
TERRITORY		
SERBIA		FALSE
0		
TERRITORY		
MONTENEGRO		
FALSE	0	
TERRITORY		
HONDURAS		
FALSE	0	
TERRITORY		
PAKISTAN		
FALSE	0	
TERRITORY		
SENEGAL		FALSE
0		
TERRITORY		
CAMEROON		
FALSE	0	
TERRITORY		
CONGO BRAZZAVILLE		
FALSE	0	
TERRITORY		
CONGO KINSHASA		FALSE
0		
TERRITORY		
GABON		
FALSE	0	
TERRITORY		
IVORY COAST		
FALSE	0	
TERRITORY		
BAHAMAS		FALSE

```

0
TERRITORY
BERMUDA                                     FALSE
0
TERRITORY
NIGERIA                                     FALSE
0
TERRITORY
UGANDA                                     FALSE
0
TERRITORY
ZAMBIA                                     FALSE
0
TERRITORY
URUGUAY                                    FALSE
0
TERRITORY
KENYA
FALSE      0

PARAMETER
VALUE
ISDEP      CON_ID
-----
-----
-----
TERRITORY
TANZANIA
FALSE      0
TERRITORY
BOLIVIA                                         FALSE
0
TERRITORY
BELIZE                                         FALSE
0
TERRITORY
PARAGUAY
FALSE      0
TERRITORY
AFGHANISTAN
FALSE      0
TERRITORY
BOSNIA AND HERZEGOVINA                        FALSE
0
TERRITORY

```

ETHIOPIA		
FALSE	0	
TERRITORY		
LAOS		
FALSE	0	
TERRITORY		
MALTA		
FALSE	0	
TERRITORY		
NEPAL		
FALSE	0	
TERRITORY		
ARMENIA		FALSE
0		
TERRITORY		
MALDIVES		
FALSE	0	
TERRITORY		
CAMBODIA		
FALSE	0	
TERRITORY		
IRAN		
FALSE	0	
TERRITORY		
SRI LANKA		
FALSE	0	
CHARACTERSET		
US7ASCII		
FALSE	0	
CHARACTERSET		
WE8DEC		FALSE
0		
CHARACTERSET		
WE8HP		
FALSE	0	
CHARACTERSET		
US8PC437		
FALSE	0	
CHARACTERSET		
WE8EBCDIC37		
FALSE	0	
CHARACTERSET		
WE8EBCDIC500		
FALSE	0	
CHARACTERSET		

WE8EBCDIC1140	
FALSE	0
CHARACTERSET	
WE8EBCDIC285	
FALSE	0
CHARACTERSET	
WE8EBCDIC1146	
FALSE	0
CHARACTERSET	
WE8PC850	
FALSE	0
CHARACTERSET	
D7DEC	
FALSE	0
CHARACTERSET	
F7DEC	
FALSE	0
CHARACTERSET	
S7DEC	
FALSE	0
CHARACTERSET	
E7DEC	
FALSE	0
CHARACTERSET	
SF7ASCII	
FALSE	0
CHARACTERSET	
NDK7DEC	
0	
CHARACTERSET	
I7DEC	
FALSE	0
CHARACTERSET	
NL7DEC	
0	
CHARACTERSET	
CH7DEC	
0	
CHARACTERSET	
YUG7ASCII	
FALSE	0
CHARACTERSET	
SF7DEC	
0	
CHARACTERSET	
	FALSE
	FALSE
	FALSE
	FALSE

TR7DEC	FALSE
0	
CHARACTERSET	
IW7IS960	
FALSE 0	
CHARACTERSET	
IN8IScii	
FALSE 0	
CHARACTERSET	
WE8EBCDIC1148	
FALSE 0	
CHARACTERSET	
WE8PC858	
FALSE 0	
CHARACTERSET	
WE8ISO8859P1	
FALSE 0	
CHARACTERSET	
EE8ISO8859P2	
FALSE 0	
CHARACTERSET	
SE8ISO8859P3	
FALSE 0	
CHARACTERSET	
NEE8ISO8859P4	
FALSE 0	
CHARACTERSET	
CL8ISO8859P5	
FALSE 0	
CHARACTERSET	
AR8ISO8859P6	
FALSE 0	
CHARACTERSET	
EL8ISO8859P7	
FALSE 0	
CHARACTERSET	
IW8ISO8859P8	
FALSE 0	
CHARACTERSET	
WE8ISO8859P9	
FALSE 0	
CHARACTERSET	
NE8ISO8859P10	
FALSE 0	
CHARACTERSET	

TH8TISASCII		
FALSE	0	
CHARACTERSET		
TH8TISEBCDIC		
FALSE	0	
CHARACTERSET		
BN8BSCII		
FALSE	0	
CHARACTERSET		
VN8VN3		FALSE
0		
CHARACTERSET		
VN8MSWIN1258		
FALSE	0	
CHARACTERSET		
WE8ISO8859P15		
FALSE	0	
CHARACTERSET		
BLT8ISO8859P13		FALSE
0		
CHARACTERSET		
CEL8ISO8859P14		FALSE
0		
CHARACTERSET		
CL8ISOIR111		
FALSE	0	
CHARACTERSET		
WE8NEXTSTEP		
FALSE	0	
CHARACTERSET		
CL8KOI8U		
FALSE	0	
CHARACTERSET		
AZ8ISO8859P9E		
FALSE	0	
CHARACTERSET		
AR8ASMO708PLUS		TRUE
0		
CHARACTERSET		
AR8EBCDICX		
FALSE	0	
CHARACTERSET		
AR8XBASIC		
TRUE	0	
CHARACTERSET		

EL8DEC		FALSE
0		
CHARACTERSET		
TR8DEC		FALSE
0		
CHARACTERSET		
WE8EBCDIC37C		
FALSE	0	
CHARACTERSET		
WE8EBCDIC500C		
FALSE	0	
CHARACTERSET		
IW8EBCDIC424		
FALSE	0	
CHARACTERSET		
TR8EBCDIC1026		
FALSE	0	
CHARACTERSET		
WE8EBCDIC871		
FALSE	0	
CHARACTERSET		
WE8EBCDIC284		
FALSE	0	
CHARACTERSET		
WE8EBCDIC1047		
FALSE	0	
CHARACTERSET		
WE8EBCDIC1140C		FALSE
0		
CHARACTERSET		
WE8EBCDIC1145		
FALSE	0	
CHARACTERSET		
WE8EBCDIC1148C		FALSE
0		
CHARACTERSET		
WE8EBCDIC1047E		FALSE
0		
CHARACTERSET		
WE8EBCDIC924		
FALSE	0	
CHARACTERSET		
EEC8EUROASCI		
FALSE	0	
CHARACTERSET		

EEC8EUROPA3	
FALSE	0
CHARACTERSET	
LA8PASSPORT	
FALSE	0
CHARACTERSET	
BG8PC437S	
FALSE	0
CHARACTERSET	
EE8PC852	
FALSE	0
CHARACTERSET	
RU8PC866	
FALSE	0
CHARACTERSET	
RU8BESTA	
FALSE	0
CHARACTERSET	
IW8PC1507	
FALSE	0
CHARACTERSET	
RU8PC855	
FALSE	0
CHARACTERSET	
TR8PC857	
FALSE	0
CHARACTERSET	
CL8MACCYRILLIC	FALSE
0	
CHARACTERSET	
CL8MACCYRILLICS	FALSE
0	
CHARACTERSET	
WE8PC860	
FALSE	0
CHARACTERSET	
IS8PC861	
FALSE	0
CHARACTERSET	
EE8MACCES	
FALSE	0
CHARACTERSET	
EE8MACCROATIANS	FALSE
0	
CHARACTERSET	

TR8MACTURKISHS FALSE
0

PARAMETER
VALUE
ISDEP CON_ID

CHARACTERSET
IS8MACICELANDICS

FALSE 0

CHARACTERSET

EL8MACGREEKS

FALSE 0

CHARACTERSET

IW8MACHEBREWS

FALSE 0

CHARACTERSET

EE8MSWIN1250

FALSE 0

CHARACTERSET

CL8MSWIN1251

FALSE 0

CHARACTERSET

ET8MSWIN923

FALSE 0

CHARACTERSET

BG8MSWIN

FALSE 0

CHARACTERSET

EL8MSWIN1253

FALSE 0

CHARACTERSET

IW8MSWIN1255

FALSE 0

CHARACTERSET

LT8MSWIN921

FALSE 0

CHARACTERSET

TR8MSWIN1254

FALSE 0

CHARACTERSET

WE8MSWIN1252

FALSE 0

CHARACTERSET		
BLT8MSWIN1257		
FALSE	0	
CHARACTERSET		
D8EBCDIC273		
FALSE	0	
CHARACTERSET		
I8EBCDIC280		
FALSE	0	
CHARACTERSET		
DK8EBCDIC277		
FALSE	0	
CHARACTERSET		
S8EBCDIC278		
FALSE	0	
CHARACTERSET		
EE8EBCDIC870		
FALSE	0	
CHARACTERSET		
CL8EBCDIC1025		
FALSE	0	
CHARACTERSET		
F8EBCDIC297		
FALSE	0	
CHARACTERSET		
IW8EBCDIC1086		
FALSE	0	
CHARACTERSET		
CL8EBCDIC1025X		FALSE
0		
CHARACTERSET		
D8EBCDIC1141		
FALSE	0	
CHARACTERSET		
N8PC865		FALSE
0		
CHARACTERSET		
BLT8CP921		
FALSE	0	
CHARACTERSET		
LV8PC1117		
FALSE	0	
CHARACTERSET		
LV8PC8LR		
FALSE	0	

CHARACTERSET BLT8EBCDIC1112 0	FALSE
CHARACTERSET LV8RST104090 FALSE 0	
CHARACTERSET CL8KOI8R FALSE 0	
CHARACTERSET BLT8PC775 FALSE 0	
CHARACTERSET DK8EBCDIC1142 FALSE 0	
CHARACTERSET S8EBCDIC1143 FALSE 0	
CHARACTERSET I8EBCDIC1144 FALSE 0	
CHARACTERSET F7SIEMENS9780X 0	FALSE
CHARACTERSET E7SIEMENS9780X 0	FALSE
CHARACTERSET S7SIEMENS9780X 0	FALSE
CHARACTERSET DK7SIEMENS9780X 0	FALSE
CHARACTERSET N7SIEMENS9780X 0	FALSE
CHARACTERSET I7SIEMENS9780X 0	FALSE
CHARACTERSET D7SIEMENS9780X 0	FALSE
CHARACTERSET F8EBCDIC1147 FALSE 0	

CHARACTERSET
WE8GCOS7
FALSE 0
CHARACTERSET
EL8GCOS7
FALSE 0
CHARACTERSET
US8BS2000
FALSE 0
CHARACTERSET
D8BS2000
FALSE 0
CHARACTERSET
F8BS2000
FALSE 0
CHARACTERSET
E8BS2000
FALSE 0
CHARACTERSET
DK8BS2000
FALSE 0
CHARACTERSET
S8BS2000
FALSE 0
CHARACTERSET
WE8BS2000E
FALSE 0
CHARACTERSET
WE8BS2000
FALSE 0
CHARACTERSET
EE8BS2000
FALSE 0
CHARACTERSET
CE8BS2000
FALSE 0
CHARACTERSET
CL8BS2000
FALSE 0
CHARACTERSET
WE8BS2000L5
FALSE 0
CHARACTERSET
WE8DG
FALSE 0

CHARACTERSET		
WE8NCR4970		
FALSE	0	
CHARACTERSET		
WE8ROMAN8		
FALSE	0	
CHARACTERSET		
EE8MACCE		
FALSE	0	
CHARACTERSET		
EE8MACCROATIAN		FALSE
0		
CHARACTERSET		
TR8MACTURKISH		
FALSE	0	
CHARACTERSET		
IS8MACICELANDIC		FALSE
0		
CHARACTERSET		
EL8MACGREEK		
FALSE	0	
CHARACTERSET		
IW8MACHEBREW		
FALSE	0	
CHARACTERSET		
US8ICL		FALSE
0		
CHARACTERSET		
WE8ICL		FALSE
0		
CHARACTERSET		
WE8ISOICLUK		
FALSE	0	
CHARACTERSET		
EE8EBCDIC870C		
FALSE	0	
CHARACTERSET		
EL8EBCDIC875S		
TRUE	0	
CHARACTERSET		
TR8EBCDIC1026S		FALSE
0		
CHARACTERSET		
BLT8EBCDIC1112S		FALSE
0		

CHARACTERSET IW8EBCDIC424S FALSE 0	
CHARACTERSET EE8EBCDIC870S FALSE 0	
CHARACTERSET CL8EBCDIC1025S 0	FALSE
CHARACTERSET TH8TISEBCDICS FALSE 0	
CHARACTERSET AR8EBCDIC420S FALSE 0	
CHARACTERSET CL8EBCDIC1025C 0	FALSE
CHARACTERSET CL8EBCDIC1025R 0	FALSE
CHARACTERSET EL8EBCDIC875R FALSE 0	
CHARACTERSET CL8EBCDIC1158 FALSE 0	
CHARACTERSET CL8EBCDIC1158R 0	FALSE
CHARACTERSET EL8EBCDIC423R FALSE 0	
CHARACTERSET WE8MACROMAN8 FALSE 0	
CHARACTERSET WE8MACROMAN8S FALSE 0	
CHARACTERSET TH8MACTHAI FALSE 0	
CHARACTERSET TH8MACTHAIS FALSE 0	

CHARACTERSET		
HU8CWI2		FALSE
0		
CHARACTERSET		
EL8PC437S		
FALSE	0	
CHARACTERSET		
EL8EBCDIC875		
FALSE	0	
CHARACTERSET		
EL8PC737		
FALSE	0	
CHARACTERSET		
LT8PC772		
FALSE	0	
CHARACTERSET		
LT8PC774		
FALSE	0	
CHARACTERSET		
EL8PC869		
FALSE	0	
CHARACTERSET		
EL8PC851		
FALSE	0	
CHARACTERSET		
CDN8PC863		
FALSE	0	
CHARACTERSET		
HU8ABMOD		
FALSE	0	
PARAMETER		
VALUE		
ISDEP	CON_ID	

CHARACTERSET		
AR8ASMO8X		
FALSE	0	
CHARACTERSET		
AR8NAFITHA711T		TRUE
0		
CHARACTERSET		
AR8SAKHR707T		

TRUE	0	
CHARACTERSET		
AR8MUSSAD768T		
TRUE	0	
CHARACTERSET		
AR8ADOS710T		
TRUE	0	
CHARACTERSET		
AR8ADOS720T		
TRUE	0	
CHARACTERSET		
AR8APTEC715T		
TRUE	0	
CHARACTERSET		
AR8NAFITHA721T		TRUE
0		
CHARACTERSET		
AR8HPARABIC8T		
TRUE	0	
CHARACTERSET		
AR8NAFITHA711		
FALSE	0	
CHARACTERSET		
AR8SAKHR707		
FALSE	0	
CHARACTERSET		
AR8MUSSAD768		
FALSE	0	
CHARACTERSET		
AR8ADOS710		
FALSE	0	
CHARACTERSET		
AR8ADOS720		
FALSE	0	
CHARACTERSET		
AR8APTEC715		
FALSE	0	
CHARACTERSET		
AR8MSWIN1256		
FALSE	0	
CHARACTERSET		
AR8NAFITHA721		
FALSE	0	
CHARACTERSET		
AR8SAKHR706		

FALSE	0	
CHARACTERSET		
AR8ARABICMAC		
FALSE	0	
CHARACTERSET		
AR8ARABICMACS		
FALSE	0	
CHARACTERSET		
AR8ARABICMACT		
TRUE	0	
CHARACTERSET		
LA8ISO6937		
FALSE	0	
CHARACTERSET		
JA16VMS		FALSE
0		
CHARACTERSET		
JA16EUC		FALSE
0		
CHARACTERSET		
JA16EUCYEN		
FALSE	0	
CHARACTERSET		
JA16SJIS		
FALSE	0	
CHARACTERSET		
JA16DBCS		
FALSE	0	
CHARACTERSET		
JA16SJISYEN		
FALSE	0	
CHARACTERSET		
JA16EBCDIC930		
FALSE	0	
CHARACTERSET		
JA16MACSJIS		
FALSE	0	
CHARACTERSET		
JA16EUCTILDE		
FALSE	0	
CHARACTERSET		
JA16SJISTILDE		
FALSE	0	
CHARACTERSET		
KO16KSC5601		

FALSE	0	
CHARACTERSET		
KO16DBCS		
FALSE	0	
CHARACTERSET		
KO16KSCCS		
FALSE	0	
CHARACTERSET		
KO16MSWIN949		
FALSE	0	
CHARACTERSET		
ZHS16CGB231280		FALSE
0		
CHARACTERSET		
ZHS16MACCGB231280		
FALSE	0	
CHARACTERSET		
ZHS16GBK		
FALSE	0	
CHARACTERSET		
ZHS16DBCS		
FALSE	0	
CHARACTERSET		
ZHS32GB18030		
FALSE	0	
CHARACTERSET		
ZHT32EUC		
FALSE	0	
CHARACTERSET		
ZHT32SOPS		
FALSE	0	
CHARACTERSET		
ZHT16DBT		
FALSE	0	
CHARACTERSET		
ZHT32TRIS		
FALSE	0	
CHARACTERSET		
ZHT16DBCS		
FALSE	0	
CHARACTERSET		
ZHT16BIG5		
FALSE	0	
CHARACTERSET		
ZHT16CCDC		

FALSE	0	
CHARACTERSET		
ZHT16MSWIN950		
FALSE	0	
CHARACTERSET		
ZHT16HKSCS		
FALSE	0	
CHARACTERSET		
AL24UTFFSS		
TRUE	0	
CHARACTERSET		
UTF8		
FALSE	0	
CHARACTERSET		
UTFE		
FALSE	0	
CHARACTERSET		
AL32UTF8		
FALSE	0	
CHARACTERSET		
ZHT16HKSCS31		
FALSE	0	
CHARACTERSET		
JA16EUCFIXED		
TRUE	0	
CHARACTERSET		
JA16SJISFIXED		
TRUE	0	
CHARACTERSET		
JA16DBCSFIXED		
TRUE	0	
CHARACTERSET		
KO16KSC5601FIXED		
TRUE	0	
CHARACTERSET		
KO16DBCSFIXED		
TRUE	0	
CHARACTERSET		
ZHS16CGB231280FIXED		
TRUE	0	
CHARACTERSET		
ZHS16GBKFIXED		
TRUE	0	
CHARACTERSET		
ZHS16DBCSFIXED		TRUE

0		
CHARACTERSET		
ZHT32EUCFIXED		
TRUE	0	
CHARACTERSET		
ZHT32TRISFIXED		TRUE
0		
CHARACTERSET		
ZHT16DBCSFIXED		TRUE
0		
CHARACTERSET		
ZHT16BIG5FIXED		TRUE
0		
CHARACTERSET		
AL16UTF16		
FALSE	0	
SORT		
BINARY		FALSE
0		
SORT		
WEST_EUROPEAN		
FALSE	0	
SORT		
XWEST_EUROPEAN		FALSE
0		
SORT		
GERMAN		FALSE
0		
SORT		
XGERMAN		FALSE
0		
SORT		
DANISH		FALSE
0		
SORT		
XDANISH		FALSE
0		
SORT		
SPANISH		FALSE
0		
SORT		
XSPANISH		
FALSE	0	
SORT		
GERMAN_DIN		

FALSE	0	
SORT		
XGERMAN_DIN		
FALSE	0	
SORT		
FINNISH		FALSE
0		
SORT		
FRENCH		FALSE
0		
SORT		
NORWEGIAN		
FALSE	0	
SORT		
SWEDISH		FALSE
0		
SORT		
ITALIAN		FALSE
0		
SORT		
ICELANDIC		
FALSE	0	
SORT		
DUTCH		
FALSE	0	
SORT		
XDUTCH		FALSE
0		
SORT		
SWISS		
FALSE	0	
SORT		
XSWISS		FALSE
0		
SORT		
ARABIC		FALSE
0		
SORT		
HUNGARIAN		
FALSE	0	
SORT		
XHUNGARIAN		
FALSE	0	
SORT		
GREEK		

FALSE	0	
SORT		
CZECH		
FALSE	0	
SORT		
XCZECH		FALSE
0		
SORT		
POLISH		FALSE
0		
SORT		
SLOVAK		FALSE
0		
PARAMETER		
VALUE		
ISDEP	CON_ID	

SORT		
XSLOVAK		FALSE
0		
SORT		
LATIN		
FALSE	0	
SORT		
THAI_DICTIONARY		TRUE
0		
SORT		
THAI_TELEPHONE		TRUE
0		
SORT		
TURKISH		FALSE
0		
SORT		
XTURKISH		
FALSE	0	
SORT		
RUSSIAN		FALSE
0		
SORT		
HEBREW		FALSE
0		
SORT		

LITHUANIAN			
FALSE	0		
SORT			
CROATIAN			
FALSE	0		
SORT			
XCROATIAN			
FALSE	0		
SORT			
ROMANIAN			
FALSE	0		
SORT			
BULGARIAN			
FALSE	0		
SORT			
CATALAN			FALSE
0			
SORT			
XCATALAN			
FALSE	0		
SORT			
SLOVENIAN			
FALSE	0		
SORT			
XSLOVENIAN			
FALSE	0		
SORT			
UKRAINIAN			
FALSE	0		
SORT			
ESTONIAN			
FALSE	0		
SORT			
ASCII7			FALSE
0			
SORT			
JAPANESE			
TRUE		0	
SORT			
MALAY			
FALSE	0		
SORT			
PUNCTUATION			
FALSE	0		
SORT			

XPUNCTUATION		
FALSE	0	
SORT		
CANADIAN_FRENCH		TRUE
0		
SORT		
VIETNAMESE		
FALSE	0	
SORT		
EEC_EURO		
FALSE	0	
SORT		
LATVIAN		FALSE
0		
SORT		
BENGALI		FALSE
0		
SORT		
XFRENCH		FALSE
0		
SORT		
INDONESIAN		
FALSE	0	
SORT		
ARABIC_MATCH		
FALSE	0	
SORT		
ARABIC_ABJ_SORT		FALSE
0		
SORT		
ARABIC_ABJ_MATCH		
FALSE	0	
SORT		
EEC_EUROPA3		
FALSE	0	
SORT		
CZECH_PUNCTUATION		
FALSE	0	
SORT		
XCZECH_PUNCTUATION		
FALSE	0	
SORT		
UNICODE_BINARY		FALSE
0		
SORT		

EBCDIC		FALSE
0		
SORT		
GENERIC_BASELETTER		
FALSE	0	
SORT		
AZERBAIJANI		
FALSE	0	
SORT		
XAZERBAIJANI		
FALSE	0	
SORT		
GENERIC_M		
FALSE	0	
SORT		
SPANISH_M		
FALSE	0	
SORT		
FRENCH_M		
FALSE	0	
SORT		
THAI_M		FALSE
0		
SORT		
CANADIAN_M		
FALSE	0	
SORT		
DANISH_M		
FALSE	0	
SORT		
TCHINESE_RADICAL_M		
FALSE	0	
SORT		
BIG5		
FALSE	0	
SORT		
HKSCS		
FALSE	0	
SORT		
TCHINESE_STROKE_M		
FALSE	0	
SORT		
SCHINESE_PINYIN_M		
FALSE	0	
SORT		

SCHINESE_STROKE_M		
FALSE	0	
SORT		
GBK		
FALSE	0	
SORT		
SCHINESE_RADICAL_M		
FALSE	0	
SORT		
JAPANESE_M		
FALSE	0	
SORT		
KOREAN_M		
FALSE	0	
SORT		
UCA0610_ROOT		
FALSE	0	
SORT		
UCA0610_DUCET		
FALSE	0	
SORT		
UCA0610_SPANISH		FALSE
0		
SORT		
UCA0610_TSPANISH		
FALSE	0	
SORT		
UCA0610_CFRENCH		FALSE
0		
SORT		
UCA0610_DANISH		FALSE
0		
SORT		
UCA0610_THAI		
FALSE	0	
SORT		
UCA0610_JAPANESE		
FALSE	0	
SORT		
UCA0610_KOREAN		FALSE
0		
SORT		
UCA0610_SCHINESE		
FALSE	0	
SORT		

```
UCA0610_SCHINESE1
FALSE      0
SORT
UCA0610_SCHINESE2
FALSE      0
SORT
UCA0610_TCHINESE
FALSE      0
SORT
UCA0610_TCHINESE1
FALSE      0
SORT
UCA0620_ROOT
FALSE      0
SORT
UCA0620_DUCET
FALSE      0
SORT
UCA0620_SPANISH
0
SORT
UCA0620_TSPANISH
FALSE      0
SORT
UCA0620_CFRENCH
0
SORT
UCA0620_DANISH
0
SORT
UCA0620_THAI
FALSE      0
SORT
UCA0620_JAPANESE
FALSE      0
SORT
UCA0620_KOREAN
0
SORT
UCA0620_SCHINESE
FALSE      0
SORT
UCA0620_SCHINESE1
FALSE      0
SORT
```

```
UCA0620_SCHINESE2
FALSE      0
SORT
UCA0620_TCHINESE
FALSE      0
SORT
UCA0620_TCHINESE1
FALSE      0
```

571 rows selected.

5. 进程

```
select * from v$process
```


6. 查看用户

查看当前用户

```
select user from dual;
```

查看所有用户

```
select * from all_users
```

```
select username from dba_users;
```

7. 显示表

```
select * from tab where tabtype='SYNONYM';
```

```
select name,type,referenced_name from user_dependencies;
```

8. 显示视图

```
select object_name,created,status from user_objects where  
object_type in ('VIEW')
```

9. PROCEDURE

```
select object_name,created,status from user_objects where  
object_type in ('PROCEDURE')
```

10. FUNCTION

```
select object_name,created,status from user_objects where  
object_type in ('FUNCTION')
```

11. 视图、存储过程、函数

查询数据库中的视图、存储过程、函数

```
select object_name,created,status  
from user_objects  
where object_type in ('PROCEDURE','FUNCTION','VIEW')
```

12. 查看存储过程源代码

```
select text from user_source where name = 'ADD_DEPT';
```

```
select text from all_source where name = 'ADD_DEPT';
```

13. 日期时间格式

```
export NLS_LANG=AMERICAN      ---要注意这一句必须指定，不然下一句不生效。  
export NLS_DATE_FORMAT='yyyy-mm-dd hh24:mi:ss'
```

13.1. Date

修改当前会话的日期格式

```
alter session set nls_date_format='YYYY-MM-DD HH24:MI:SS';
```

查看本次会话定义

```
select * from nls_session_parameters where parameter =  
'NLS_DATE_FORMAT';
```

查看本次会话定义

```
SELECT * FROM v$nls_parameters where parameter =  
'NLS_DATE_FORMAT';
```

13.2. 修改系统日期格式

查看数据库定义

```
alter system set nls_date_format='yyyy-mm-dd hh24:mi:ss'  
scope=spfile;
```



```
select * from nls_database_parameters where parameter =  
'NLS_DATE_FORMAT';
```

13.3. TIMESTAMP

```
ALTER SESSION SET NLS_TIMESTAMP_TZ_FORMAT='DD-MON-RR  
HH:MI:SSXFF AM TZR';
```

13.4.

查看数据库时区信息

```
select dbtimezone from dual;
```

查看session时区信息:

```
select sessiontimezone from dual;
```

Database的timezone可以在创建数据库的时候指定, 如:

```
CREATE DATABASE db01
```

```
...
```

```
SET TIME_ZONE='+08:00';
```

或者在数据库创建之后通过alter database语句修改, 但是只有重启数据库后有效:

```
ALTER DATABASE SET TIME_ZONE='+08:00';
```

session的timezone可以简单通过alter session语句修改:

```
ALTER SESSION SET TIME_ZONE='+08:00';
```

14. 切换字符集

14.1. 切换到 GBK

```
SQL> shutdown immediate;
SQL> startup mount;
SQL> alter system enable restricted session;
SQL> alter system set job_queue_processes=0;
SQL> alter database open;
SQL> alter database character set internal_use ZHS16GBK;
SQL> shutdown immediate;
SQL> startup
SQL> alter system disable restricted session;
```

14.2. 切到 UTF-8

```
sqlplus "/ as sysdba"
SQL> SHUTDOWN IMMEDIATE
SQL> STARTUP MOUNT
SQL> ALTER SYSTEM ENABLE RESTRICTED SESSION;
SQL> ALTER SYSTEM SET JOB_QUEUE_PROCESSES=0;
SQL> ALTER SYSTEM SET AQ_TM_PROCESSES=0;
SQL> ALTER DATABASE OPEN
SQL> ALTER DATABASE NATIONAL CHARACTER SET INTERNAL_USE UTF8;
SQL> SHUTDOWN IMMEDIATE
SQL> STARTUP
```

14.3. 切到 AL32UTF8

```
SQL> shutdown immediate;
```

```
SQL> startup mount;
SQL> alter system enable restricted session;
SQL> alter system set job_queue_processes=0;
SQL> alter database open;
SQL> alter database character set internal_use AL32UTF8;
SQL> shutdown immediate;
SQL> startup
SQL> alter system disable restricted session;
```

14.4. 切换过程实例

例 75.1. Oracle 字符集切换实例

```
C:\Users\Administrator>sqlplus

SQL*Plus: Release 11.2.0.1.0 Production on Mon Nov 30 17:01:10
2015

Copyright (c) 1982, 2010, Oracle. All rights reserved.

Enter user-name: sys as sysdba
Enter password:

Connected to:
Oracle Database 11g Release 11.2.0.1.0 - 64bit Production

SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup mount;
ORACLE instance started.

Total System Global Area 3423965184 bytes
Fixed Size 2180544 bytes
Variable Size 1879050816 bytes
Database Buffers 1526726656 bytes
Redo Buffers 16007168 bytes
Database mounted.
SQL> alter system enable restricted session;
```

System altered.

```
SQL> alter system set job_queue_processes=0;
```

System altered.

```
SQL> alter database open;
```

Database altered.

```
SQL> ALTER DATABASE character set INTERNAL_USE ZHS16GBK;
```

Database altered.

```
SQL> shutdown immediate;
```

Database closed.

Database dismounted.

ORACLE instance shut down.

```
SQL> startup
```

ORACLE instance started.

Total System Global Area 3423965184 bytes

Fixed Size 2180544 bytes

Variable Size 1879050816 bytes

Database Buffers 1526726656 bytes

Redo Buffers 16007168 bytes

Database mounted.

Database opened.

```
SQL> alter system disable restricted session;
```

System altered.

```
SQL>
```

```
SQL> select userenv('language') from dual;
```

USERENV('LANGUAGE')

AMERICAN_AMERICA.ZHS16GBK

```
SQL>
```

15. Oracle 表空间

15.1. 查询空闲表空间

```
select tablespace_name,file_id,block_id,bytes,blocks from
dba_free_space;
```

```
SQL> select file_name from dba_data_files;
```

```
FILE_NAME
```

```
-----
-----
-----
-----
-----
-----
-----
-----
```

```
/opt/oracle/oradata/orcl/users01.dbf
/opt/oracle/oradata/orcl/undotbs01.dbf
/opt/oracle/oradata/orcl/sysaux01.dbf
/opt/oracle/oradata/orcl/system01.dbf
/opt/oracle/oradata/orcl/example01.dbf
/opt/oracle/oradata/orcl/neo.dbf
```

```
6 rows selected.
```

15.2. 创建表空间

```
create tablespace test
datafile '/opt/app/oracle/oradata/test.dbf' size 8M
autoextend on
next 5M
```

```
maxsize 10M;
```

maxsize unlimited 是大小不受限制

```
create tablespace test
datafile '/opt/app/oracle/oradata/test.dbf' size 800M
autoextend on
next 50M
maxsize unlimited
```

uniform表示区的大小相同，默认为1M

```
create tablespace test
datafile '/opt/app/oracle/oradata/test.dbf' size 800M
autoextend on
next 50M
maxsize 1000M
extent management local uniform;
```

uniform size 500K表示区的大小相同，为500K

```
create tablespace test
datafile '/opt/app/oracle/oradata/test.dbf' size 800M
autoextend on
next 50M
maxsize 1000M
extent management local uniform size 500K;
```

autoallocate 表示区的大小由随表的大小自动动态改变

```
create tablespace test
datafile '/opt/app/oracle/oradata/test.dbf' size 800M
autoextend on
next 50M
maxsize 1000M
```

```
extent management local autoallocate;
```

temporary 创建字典管理临时表空间

```
create tablespace test
datafile '/opt/app/oracle/oradata/test.dbf' size 800M
autoextend on
next 50M
maxsize 1000M
temporary;
```

例 75.2. 创建表空间实例

```
SQL> create tablespace ts_b01_def datafile
'/opt/oracle/oradata/orcl/ts_b01_def.dbf' size 100m autoextend
on;
```

Tablespace created.

```
SQL> create tablespace ts_b01_idx datafile
'/opt/oracle/oradata/orcl/ts_b01_idx.dbf' size 100m autoextend
on;
```

Tablespace created.

临时表空间

创建临时表空间，语句中的datafile都换为tempfile

```
create temporary tablespace test
tempfile '/opt/app/oracle/oradata/test.dbf' size 800M
autoextend on
next 50M
maxsize 1000M
```

15.3. 更改表空间属性

更改自动扩展属性

```
alter database datafile
  '/opt/app/oracle/oradata/test.dbf',
  '/opt/app/oracle/oradata/test01.dbf'
  '/opt/app/oracle/oradata/test02.dbf'
  autoextend off;
```

修改表空间大小

先查询数据文件名称、大小和路径的信息，语句如下：

```
select tablespace_name,file_id,bytes,file_name from
dba_data_files;
```

增加表空间，修改文件大小语句如下

```
alter database datafile '需要增加的数据文件路径，即上面查询出来的路径'
  resize 800M;
```

15.4. 删除表空间

```
drop tablespace "空间名" including contents and datafiles
drop tablespace test including contents and datafiles
```


第 76 章 Oracle Gui

1. Create instance

```
[oracle@netkiller ~]$ dbca
```

```
dbca -silent -createDatabase -templateName General_Purpose.dbc  
-gdbname orallg -sid orallg -responseFile NO_VALUE -  
characterSet AL32UTF8  
-memoryPercentage 30 -emConfiguration LOCAL
```

```
Enter SYSTEM user password:
```

```
password
```

```
Enter SYS user password:
```

```
password
```

```
Copying database files
```

```
1% complete
```

```
3% complete
```

```
...
```

2. Oracle Net Configuration Assistant

3. Oracle Enterprise Manager

启动em

emctl start dbconsole

```
[oracle@oracle ~]$ emctl start dbconsole
Oracle Enterprise Manager 11g Database Control Release
11.2.0.1.0
Copyright (c) 1996, 2009 Oracle Corporation. All rights
reserved.
https://oracle.example.com:1158/em/console/aboutApplication
Starting Oracle Enterprise Manager 11g Database Control .....
started.
-----
---
Logs are generated in directory
/opt/oracle/product/11.2.0/dbhome_1/oracle.example.com_wcsdb/sy
sman/log
```

```
https://oracle.example.com:1158/em
```

使用system用户登录

停止em

emctl stop dbconsole

```
[oracle@oracle ~]$ emctl stop dbconsole
Oracle Enterprise Manager 11g Database Control Release
11.2.0.1.0
Copyright (c) 1996, 2009 Oracle Corporation. All rights
reserved.
https://oracle.example.com:1158/em/console/aboutApplication
Stopping Oracle Enterprise Manager 11g Database Control ...
... Stopped.
```

3.1. EM Configuration issue.

/opt/oracle/product/11.2.0/dbhome_1/orcl.example.com_orcl not found.

如果不记得密码请先重置密码

```
alter user sys identified by chen account unlock;
alter user sysman identified by chen account unlock;
alter user dbsnmp identified by chen account unlock;
```

运行emca 重建EM

```
[oracle@orcl ~]$ emca -config dbcontrol db -repos recreate

STARTED EMCA at Dec 2, 2015 8:48:06 AM
EM Configuration Assistant, Version 11.2.0.0.2 Production
Copyright (c) 2003, 2005, Oracle. All rights reserved.

Enter the following information:
Database SID: orcl
Listener port number: 1521
Listener ORACLE_HOME [ /opt/oracle/product/11.2.0/dbhome_1 ]:
Password for SYS user:
Password for DBSNMP user:
Password for SYSMAN user:
Email address for notifications (optional): netkiller@msn.com
Outgoing Mail (SMTP) server for notifications (optional):
-----
--

You have specified the following settings

Database ORACLE_HOME .....
/opt/oracle/product/11.2.0/dbhome_1
```

```
Local hostname ..... orcl.example.com
Listener ORACLE_HOME .....
/opt/oracle/product/11.2.0/dbhome_1
Listener port number ..... 1521
Database SID ..... orcl
Email address for notifications .....
netkiller@msn.com
Outgoing Mail (SMTP) server for notifications .....
```

--

```
Do you wish to continue? [yes(Y)/no(N)]: Y
Dec 2, 2015 8:49:12 AM oracle.sysman.emcp.EMConfig perform
INFO: This operation is being logged at
/opt/oracle/cfgtoollogs/emca/orcl/emca_2015_12_02_08_48_06.log.
Dec 2, 2015 8:49:12 AM oracle.sysman.emcp.EMReposConfig invoke
INFO: Dropping the EM repository (this may take a while) ...
Dec 2, 2015 8:50:15 AM oracle.sysman.emcp.EMReposConfig invoke
INFO: Repository successfully dropped
Dec 2, 2015 8:50:16 AM oracle.sysman.emcp.EMReposConfig
createRepository
INFO: Creating the EM repository (this may take a while) ...
Dec 2, 2015 8:54:56 AM oracle.sysman.emcp.EMReposConfig invoke
INFO: Repository successfully created
Dec 2, 2015 8:54:58 AM oracle.sysman.emcp.EMReposConfig
uploadConfigDataToRepository
INFO: Uploading configuration data to EM repository (this may
take a while) ...
Dec 2, 2015 8:55:35 AM oracle.sysman.emcp.EMReposConfig invoke
INFO: Uploaded configuration data successfully
Dec 2, 2015 8:55:37 AM oracle.sysman.emcp.util.DBControlUtil
configureSoftwareLib
INFO: Software library configured successfully.
Dec 2, 2015 8:55:37 AM oracle.sysman.emcp.EMDBPostConfig
configureSoftwareLibrary
INFO: Deploying Provisioning archives ...
Dec 2, 2015 8:55:55 AM oracle.sysman.emcp.EMDBPostConfig
configureSoftwareLibrary
INFO: Provisioning archives deployed successfully.
Dec 2, 2015 8:55:55 AM oracle.sysman.emcp.util.DBControlUtil
secureDBConsole
INFO: Securing Database Control (this may take a while) ...
Dec 2, 2015 8:56:39 AM oracle.sysman.emcp.util.DBControlUtil
secureDBConsole
```

```
INFO: Database Control secured successfully.
Dec 2, 2015 8:56:39 AM oracle.sysman.emcp.util.DBControlUtil
startOMS
INFO: Starting Database Control (this may take a while) ...
Dec 2, 2015 8:57:06 AM oracle.sysman.emcp.EMDBPostConfig
performConfiguration
INFO: Database Control started successfully
Dec 2, 2015 8:57:06 AM oracle.sysman.emcp.EMDBPostConfig
performConfiguration
INFO: >>>>>>>>> The Database Control URL is
https://orcl.example.com:1158/em <<<<<<<<<<<
Dec 2, 2015 8:57:11 AM oracle.sysman.emcp.EMDBPostConfig invoke
WARNING:
***** WARNING *****

Management Repository has been placed in secure mode wherein
Enterprise Manager data will be encrypted. The encryption key
has been placed in the file:
/opt/oracle/product/11.2.0/dbhome_1/orcl.example.com_orcl/sysma
n/config/emkey.ora. Please ensure this file is backed up as
the encrypted data will become unusable if this file is lost.

*****
Enterprise Manager configuration completed successfully
FINISHED EMCA at Dec 2, 2015 8:57:11 AM
```

查看EM是否启动

```
$ ss -lnt | grep 1158
0          0                *:1158
*:*
```

<https://orcl.example.com:1158/em>

4. Other GUI - phpOraAdmin

<http://phporaadmin.sourceforge.net>



第 77 章 CLI

1. SQL*Plus

sqlplus /nolog

```
[oracle@wcs ~]$ sqlplus /nolog

SQL*Plus: Release 11.2.0.1.0 Production on Sat May 28 18:19:53
2011

Copyright (c) 1982, 2009, Oracle. All rights reserved.

SQL> conn / as sysdba;
Connected to an idle instance.
SQL> exit
```

sqlplus / as sysdba

```
[oracle@wcs ~]$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Sat May 28 18:31:25
2011

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
64bit Production
With the Partitioning, OLAP, Data Mining and Real Application
Testing options

SQL>
```


1.1. conn

```
SQL>conn system/manger@orcl as sysdba
```

```
conn sys/sys@ip:1521/orainstance as sysdba
```

1.2. startup/shutdown

```
[oracle@localhost ~]$ sqlplus
SQL*Plus: Release 10.2.0.1.0 - Production on Tue Jan 5 09:44:13
2010
Copyright (c) 1982, 2005, Oracle. All rights reserved.
Enter user-name: sys as sysdba
Enter password:
```

```
SQL> conn / as sysdba;
```

startup

```
SYS@orcl> startup
```

shutdown

```
SYS@orcl> shutdown immediate
```

1.3. \$ORACLE_HOME/sqlplus/admin/glogin.sql

```
set line 2000
set linesize 2000
set pagesize 100
col ename format a30
col sal format 999,999.999
alter session set nls_date_format = 'yyyy-mm-dd hh24:mi:ss';
```

1.4. @运行SQL

```
SQL> @ /home/oracle/your.sql
```

```
set pagesize 0
set linesize 80
set term off
set feed off
set echo off
set show off
set veri off
set head off

spool outputfile
select * from dba_users;
/
```

```
spool off
```

1.5. 链接数据库

Example: Connect to database using Net Service Name and the database net service name is ORCL.

```
sqlplus myusername/mypassword@ORCL
```

```
sqlplus "user/password@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)  
(Host=192.168.4.9)(Port=1521))(CONNECT_DATA=(SERVER =  
DEDICATED)(SERVICE_NAME = orcl.example.com)))"
```

2. exp/imp

创建导入，导出用户

```
sqlplus /nolog
conn system/manager

GRANT CREATE USER,DROP USER,ALTER USER ,CREATE ANY VIEW ,
      DROP ANY VIEW,EXP_FULL_DATABASE,IMP_FULL_DATABASE,
      DBA,CONNECT,RESOURCE,CREATE SESSION TO 用户名字;
```

2.1. exp

导出命令

```
exp USER/PASSWORD@SID FILE=/opt/oracle/backup/***.DMP //保存数据;
exp USER/PASSWORD@SID FILE=/opt/oracle/backup/***.DMP full=y //全部导出;
```

将数据库中system用户与sys用户的表导出

```
exp system/password@TEST file=d:\data.dmp owner=(system,sys)
```

指定导出表

```
exp system/password@DB1 file= d:\data.dmp tables=
(table1,table2)
```

将数据库中的表table1中的字段filed1以"130"打头的的数据导出

```
exp system/password@TEST file=d:\data.dmp tables=(table1)
query=" where filed1 like '130%'"
```

压缩导出文件

```
exp system/password@TEST file=d:\data.dmp full=y compress=y
```

2.2. imp

导入命令用法

```
imp user/password@orcl file=/opt/oracle/backup/***.dmp full=y
ignore=y;
```

```
imp system/password full=y file=database.dmp
```

A用户导出**B**用户导入

创建parfile文件内容如下

```
vim parfile.par

FROMUSER=USERA
TOUSER=USERB
ROWS=Y
INDEXES=Y
GRANTS=Y
CONSTRAINTS=Y
```

```
BUFFER=409600  
file=data.dmp  
log=data.log
```

导入命令如下

```
imp parfile=/filepath/parfile.par
```

3. expdp/impdp

创建 dump 目录与用户

```
create directory mydump as '/u01/oracle';
```

```
grant read, write on directory mydump to dumpuser  
grant read,write on directory backup to finance;
```

3.1. expdp

导出用户的数据

```
expdp dumpuser/password directory=mydump dumpfile=user.dmp
```

导出指定表数据

```
expdp dumpuser/password directory=mydump dumpfile=table.dmp tables=test1,test2
```

按用户导

```
expdp scott/tiger@orcl schemas=scott dumpfile=expdp.dmp DIRECTORY=backup;
```

并行进程进程数

```
expdp scott/tiger@orcl directory=backup dumpfile=scott3.dmp parallel=40  
job_name=scott3
```

指定查询条件

```
expdp scott/tiger@orcl directory=backup dumpfile=expdp.dmp Tables=emp  
query='WHERE age=20';
```

按表空间导

```
expdp system/manager DIRECTORY=backup DUMPFILE=tablespace.dmp
```

```
TABLESPACES=temp,example;
```

导整个数据库

```
expdp system/manager DIRECTORY=backup DUMPFILE=full.dmp FULL=y;
```

3.2. impdp

导入该用户数据

```
impdp dumpuser/password directory=mydump dumpfile=user.dmp
```

导出表数据

```
impdp dumpuser/password directory=mydump dumpfile=table.dmp
```

```
$ impdp \'/ as sysdba\' directory=backup dumpfile=user.dmp logfile=user.log  
tablespaces=dave,bl table_exists_action=replace;
```

导到指定Schemas下

```
impdp scott/tiger DIRECTORY=backup DUMPFILE=expdp.dmp SCHEMAS=scott;
```

改变表的所有者

```
impdp system/manager DIRECTORY=backup DUMPFILE=expdp.dmp TABLES=scott.dept  
REMAP_SCHEMA=scott:system;
```

导入表空间

```
impdp system/manager DIRECTORY=backup DUMPFILE=tablespace.dmp  
TABLESPACES=example;
```


导入数据库

```
impdb system/manager DIRECTORY=dump_dir DUMPFILE=full.dmp FULL=y;
```

追加数据

```
impdp system/manager DIRECTORY=backup DUMPFILE=expdp.dmp SCHEMAS=system  
TABLE_EXISTS_ACTION
```

3.3. 数据泵演示

创建目录

```
mkdir /opt/oracle/backup
```

创建用户

```
create user backup identified by passw0rd;  
GRANT resource,connect,CREATE SESSION,CREATE TABLE,SELECT ANY TABLE,UPDATE ANY  
TABLE,DELETE ANY TABLE,INSERT ANY TABLE,ALTER ANY TABLE,DROP ANY TABLE TO  
backup;  
  
create directory backup as '/opt/oracle/backup';  
grant read,write on directory backup to backup;
```

例 77.1. expdp

```
$ expdp backup/passw0rd dumpfile=backup.dmp directory=backup  
Export: Release 11.2.0.1.0 - Production on Tue Dec 1 16:19:56 2015  
Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.  
Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit  
Production  
With the Partitioning, OLAP, Data Mining and Real Application Testing options  
Starting "BACKUP"."SYS_EXPORT_SCHEMA_01": backup/***** dumpfile=backup.dmp  
directory=backup  
Estimate in progress using BLOCKS method...  
Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA  
Total estimation using BLOCKS method: 0 KB  
Processing object type SCHEMA_EXPORT/PRE_SCHEMA/PROCACT_SCHEMA  
Processing object type SCHEMA_EXPORT/TABLE/TABLE
```

```

Processing object type SCHEMA_EXPORT/TABLE/INDEX/INDEX
Processing object type SCHEMA_EXPORT/TABLE/CONSTRAINT/CONSTRAINT
Processing object type SCHEMA_EXPORT/TABLE/INDEX/STATISTICS/INDEX_STATISTICS
Processing object type SCHEMA_EXPORT/TABLE/COMMENT
Master table "BACKUP"."SYS_EXPORT_SCHEMA_01" successfully loaded/unloaded
*****
Dump file set for BACKUP.SYS_EXPORT_SCHEMA_01 is:
/opt/oracle/backup/backup.dmp
Job "BACKUP"."SYS_EXPORT_SCHEMA_01" successfully completed at 16:20:28

```

运行完成后会在/opt/oracle/backup/目录生成backup.dmp文件

例 77.2. impdp

首先将dmp文件复制到恢复目录中，确认oracle用户有权限访问该文件。

```

mv backup.dmp /opt/oracle/backup/
chown oracle:oinstall /opt/oracle/backup/backup.dmp

```

如果不知道目录可以使用下面语句查看

```

SQL> SELECT directory_name, directory_path FROM dba_directories WHERE
directory_name='BACKUP';

```

```

DIRECTORY_NAME
-----

```

```

DIRECTORY_PATH
-----

```

```

BACKUP
/opt/oracle/backup

```

```

$ impdp backup/passw0rd directory=backup dumpfile=backup.dmp

```

```

Import: Release 11.2.0.1.0 - Production on Tue Dec 1 16:32:34 2015

```

```

Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.

```

```

Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit
Production

```

```

With the Partitioning, OLAP, Data Mining and Real Application Testing options

```

```

ORA-31655: no data or metadata objects selected for job

```

```

ORA-39154: Objects from foreign schemas have been removed from import

```

```

Master table "BACKUP"."SYS_IMPORT_FULL_01" successfully loaded/unloaded

```

```

Starting "BACKUP"."SYS_IMPORT_FULL_01": backup/***** directory=backup
dumpfile=backup.dmp

```

```
Job "BACKUP"."SYS_IMPORT_FULL_01" successfully completed at 16:32:36
```

3.4. 查看dmp文件的表空间

```
impdp \'/ as sysdba\' dumpfile=backup:expdp.dmp sqlfile=backup:expdp.sql  
grep 'TABLESPACE' /opt/oracle/backup/expdp.sql | awk '{print $2}' | sort -u  
TABLESPACE  
"TS_DATA_DEF"  
"TS_DATA_IDX"  
UNLIMITED
```

4. RMAN

4.1. 数据库模式

数据库必须是归档模式，使用下面SQL查询当前数据库模式

```
SQL> archive log list;
Database log mode                Archive Mode
Automatic archival                Enabled
Archive destination                USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence        66
Next log sequence to archive      68
Current log sequence               68
```

如果已经是归档模式可跳过此步,下面是切换服务器到归档模式的方法:

```
[oracle@oracle ~]$ sqlplus /nolog (启动sqlplus)
SQL> conn / as sysdba (以DBA身份连接数据库)
SQL> shutdown immediate; (立即关闭数据库)
SQL> startup mount (启动实例并加载数据库,但不打开)
SQL> alter database archivelog; (更改数据库为归档模式)
SQL> alter database open; (打开数据库)
SQL> alter system archive log start; (启用自动归档)
SQL> exit (退出)
```

rman 采用块备份，查看块信息使用下面SQL语句

```
select * from dba_extents
```

4.2. 完全备份

```
run {
allocate channel d1 type disk;
backup format='/opt/oracle/backup/%d_%N_%s.bk' tablespace
users;
release channel d1;
}
```

下面让我来演示给你看

```
[oracle@oracle ~]$ mkdir /opt/oracle/backup
[oracle@oracle ~]$ rman target sys/passw0rd nocatalog

Recovery Manager: Release 11.2.0.1.0 - Production on Mon Jun 13
12:03:20 2011

Copyright (c) 1982, 2009, Oracle and/or its affiliates. All
rights reserved.

connected to target database: WCSDB (DBID=2970836713)
using target database control file instead of recovery catalog

RMAN> run {
allocate channel d1 type disk;
backup format='/opt/oracle/backup/%d_%N_%s.bk' tablespace
users;
release channel d1;
} 2> 3> 4> 5>

allocated channel: d1
```

```

channel d1: SID=36 device type=DISK

Starting backup at 13-JUN-11
channel d1: starting full datafile backup set
channel d1: specifying datafile(s) in backup set
input datafile file number=00004
name=/opt/oracle/oradata/wcsdb/users01.dbf
channel d1: starting piece 1 at 13-JUN-11
channel d1: finished piece 1 at 13-JUN-11
piece handle=/opt/oracle/backup/WCSDB_USERS_1.bk
tag=TAG20110613T120325 comment=NONE
channel d1: backup set complete, elapsed time: 00:00:01
Finished backup at 13-JUN-11

released channel: d1

RMAN>

```

查看备份结果

```

RMAN> list backup of tablespace users;

List of Backup Sets
=====

BS Key   Type LV Size           Device Type Elapsed Time Completion
Time
-----  -
1        Full  1.30M           DISK        00:00:01    13-JUN-11
        BP Key: 1   Status: AVAILABLE Compressed: NO Tag:
TAG20110613T120325
        Piece Name: /opt/oracle/backup/WCSDB_USERS_1.bk
List of Datafiles in backup set 1
File LV Type Ckp SCN      Ckp Time  Name
-----  -
4        Full 1561686      13-JUN-11
/opt/oracle/oradata/wcsdb/users01.dbf

```

备份目录下面是刚刚生成的备份文件

```
$ ls /opt/oracle/backup
WCSDDB_USERS_1.bk
```

例 77.3. full backup

```
# crontab -u oracle -l
0 1 * * * /opt/oracle/rman/backup.sh
```

```
# cat .bash_profile
export ORACLE_BASE=/opt/oracle
export ORACLE_HOME=$ORACLE_BASE/product/11.2.0/dbhome_1
export ORACLE_SID=orcl
export PATH=$PATH:$HOME/bin:$ORACLE_HOME/bin
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/usr/lib

# cat /opt/oracle/rman/backup.sh
rman target sys/passw0rd@orcl msglog=/opt/oracle/rman/log/`date
+%Y%m%d%H%M`.log cmdfile=/opt/oracle/rman/orcl.rman
```

```
# cat /opt/oracle/rman/orcl.rman
run {
  allocate channel c1 type disk;
  allocate channel c2 type disk;
  configure retention policy to recovery window of 6 days;
  configure controlfile autobackup on;
  configure backup optimization on;
```

```

    configure device type disk parallelism 4 backup type to
compressed backupset;
    configure controlfile autobackup format for device type disk
to '/opt/oracle/backup/rman/%F.ctl';
    sql 'alter system switch logfile';
    backup full database format
'/opt/oracle/backup/rman/df_%t_%s_%p.bak' tag='full' include
current controlfile;
    sql 'alter system archive log current';
    backup archivelog all format
'/opt/oracle/backup/rman/arc_%U_%s.bak' delete all input;
    release channel c1;
    release channel c2;
}
crosscheck backup;
delete noprompt expired backup;
delete noprompt obsolete;
exit

```

4.3. 增量备份

```

RMAN> BACKUP INCREMENTAL LEVEL 0 DATABASE;
RMAN> BACKUP INCREMENTAL LEVEL 1 TABLESPACE SYSTEM DATAFILE
'ora_home/oradata/ tools01.dbf';
RMAN> BACKUP INCREMENTAL LEVEL = 1 CUMULATIVE TABLESPACE users;

```

网上发现的地步，署名不详

```

run{
allocate channel c1 type disk;
allocate channel c2 type disk;
allocate channel c3 type disk;
backup full tag 'dbfull' format
'/u01/oradata/backup/full%u_%s_%p' database include current
controlfile;
sql 'alter system archive log current';
backup filesperset 3 format '/u01/oradata/backup/arch%u_%s_%p'
archivelog all delete input;
release channel c1;

```



```
release channel c2;
release channel c3;
}
```

零级备份脚本

```
run{
allocate channel c1 type disk;
allocate channel c2 type disk;
allocate channel c3 type disk;
backup incremental level 0 tag 'db0' format
'/u01/oradata/backup/db0%u_%s_%p' database skip readonly;
sql 'alter system archive log current';
backup filesperset 3 format '/u01/oradata/backup/arch%u_%s_%p'
archivelog all delete input;
release channel c1;
release channel c2;
release channel c3;
}
```

一级备份脚本

```
run{
allocate channel c1 type disk;
allocate channel c2 type disk;
allocate channel c3 type disk;
backup incremental level 1 tag 'db1' format
'/u01/oradata/backup/db1%u_%s_%p' database skip readonly;
sql 'alter system archive log current';
backup filesperset 3 format '/u01/oradata/backup/arch%u_%s_%p'
archivelog all delete input;
release channel c1;
release channel c2;
release channel c3;
}
```

网上发现的地步，没有署名，我也没有测试过，仅供参考

```
do_rman.sh
```

```
#!/bin/bash
```

```
#set env
```

```
export NLS_LANG=AMERICAN_AMERICA.ZHS16GBK
```

```
export PATH=$ORACLE_HOME/bin:$PATH
```

```

TARGET_SID=$TARGET_SID
RMAN_SID=$RMAN_SID

export PATH=$ORACLE_HOME/bin:$PATH
DATE=`date +%w`
DATE_2=`date +%Y%m%d`
BACKUP_PATH=$ORACLE_BASE/admin/$ORACLE_SID/rman/backup
LEVEL=$@
BIN=$ORACLE_HOME/bin
# Delete the data backed up last time

rm -rf $BACKUP_PATH/data/$DATE/*

if [ $# != 1 ]; then
    echo "usage: do_rman.sh n
    where n is the rman backup level(0,1,2 is permitted)."
    exit 1
fi
if [ $@ -ne 0 -a $@ -ne 1 -a $@ -ne 2 ]; then
    echo "usage: do_rman.sh n
    where n is the rman backup level(Only 0,1,2 is
permitted)."
    exit 2
fi
echo "[do_rman] rman is starting."
if [ $LEVEL = 0 ]; then
    $BIN/rman log
$BACKUP_PATH/log/level.$TARGET_SID.$LEVEL.$DATE_2.log < connect
target /;
    connect catalog rman/rman@$RMAN_SID;
    resync catalog;
    run{
        allocate channel c1 type disk ;
        crosscheck backupset of archivelog all ;
        backup filesperset 3 format
'$BACKUP_PATH/data/$DATE/arch.%d.live.$LEVEL.%t'(archivelog
from time 'sysdate-7' all delete input) ;
        delete noprompt expired backupset of archivelog
all ;
        release channel c1 ;
    }

    run{
        allocate channel c2 type disk ;

```

```

                crosscheck backupset of database ;
                backup incremental level $LEVEL filesperset 3
format
'$BACKUP_PATH/data/$DATE/data.%d.live.$LEVEL.%t' (database
include current controlfile) ;
                delete noprompt expired backupset of database ;
                delete noprompt obsolete ;
                release channel c2 ;
        }
        exit;
EOF
else
        $BIN/rman log
$BACKUP_PATH/log/level.$TARGET_SID.$LEVEL.$DATE_2.log < connect
target sys/sys202;
        connect catalog rman/rman@$RMAN_SID;
        resync catalog;
        run{
                allocate channel c1 type disk ;
                crosscheck backupset of archivelog all ;
                backup filesperset 3 format
'$BACKUP_PATH/data/$DATE/arch.%d.live.$LEVEL.%t' (archivelog
from time 'sysdate-1' all) ;
                delete noprompt expired backupset of archivelog
all ;
                release channel c1 ;
        }

        run{
                allocate channel c2 type disk ;
                crosscheck backupset of database ;
                backup incremental level $LEVEL filesperset 3
format '$BACKUP_PATH/data/$DATE/data.%d.live.$LEVEL.%t'
(database include current controlfile) ;
                delete noprompt expired backupset of database ;
                delete noprompt obsolete ;
                release channel c2 ;
        }

        exit;
EOF
fi

echo "[do_rman] rman is success."

```

4.4. 恢复数据库

```
%rman target=rman/rman@mydb  
  
RMAN> startup nomount  
  
RMAN> restore database;  
  
RMAN> recover database;  
  
RMAN> alter database open;
```

网上找到的文档，署名不详，我没有测试过是否可以运行

设定参数：

```
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO  
'/arch/rman/controlfile%F.ctl';  
CONFIGURE CHANNEL 1 DEVICE TYPE DISK FORMAT  
'/arch/rman/full%t.bak';
```

数据库rman 全备

```
rman>backup database plus archivelog delete input;
```

备份产生的三个文件

```
-rw-r----- 1 oracle oinstall    7143424 Jan 28 18:05  
controlfilec-2719028776-20100128-01.ctl  
-rw-r----- 1 oracle oinstall    41074688 Jan 28 18:03  
full1709495428.bak  
-rw-r----- 1 oracle oinstall    763379712 Jan 28 18:05  
full1709495432.bak  
-rw-r----- 1 oracle oinstall      17920 Jan 28 18:05  
full1709495518.bak
```

rman恢复

1、启动数据库到 nomount 状态

```
$sqlplus / as sysdba  
SQL> startup nomount
```

2、spfile 恢复

```
$rman nocatalog  
rman> connect target /  
run {  
allocate channel c1 DEVICE TYPE DISK format  
'/arch/rman/controlfile%F.ctnl';  
restore spfile to pfile '/arch/pfile.ora' from  
'/arch/rman/controlfilec-2719028776-20100128-01.ctnl';  
release channel c1;  
}
```

3、控制文件恢复

```
run {  
allocate channel c1 DEVICE TYPE DISK format  
'/arch/rman/controlfile%F.ctnl';  
restore controlfile from '/arch/rman/controlfilec-2719028776-  
20100128-01.ctnl';  
release channel c1;  
}
```

4、全库恢复

在恢复控制文件的情况下，可以修改数据到 mount 状态，进行全库的恢复

```
rman> alter database mount;  
run {  
allocate channel c1 device type disk format  
'/arch/rman/full%t.bak';  
restore database;  
release channel c1;  
}
```

5、恢复archivelog

```
run {  
allocate channel c1 device type disk format  
'/arch/rman/full%t.bak';  
restore archivelog all;  
}  
run {  
allocate channel c1 device type disk format  
'/arch/rman/full%t.bak';  
restore archivelog from logseq=72 until logseq=73;  
}
```

6、redolog 恢复

```
SQL>recover database using backup controlfile until cancel;  
SQL>alter database open resetlogs; //现在有redolog 产生了, 还有  
temp表空间文件也生成了或者分开两步执行
```

```
SQL>select * from dual;  
全库成功恢复
```

4.5. 是用tar打包rman文件

```
# find /opt/oracle/rman/ -type f -mtime 1 -printf "%CY-%Cm-%Cd  
%Cr %s %f\n"  
  
tar --newer="2011-07-04" -zcvf backup.tar.gz /opt/oracle/rman/  
  
find /opt/oracle/rman/ -type f -mtime 1 | xargs tar zcvf  
oracle_2011-07-04.tgz  
  
rsync -azP `find /opt/oracle/rman/ -type f -mtime 1`  
test@172.16.0.5:/home/test
```

4.6. 打包 rman 备份文件

```
find -type f -mtime 1 |xargs ls -l|wc -l  
  
find /u01/backup/rman/ -type f -mtime 1 |xargs tar -zcvf  
oracle.2011-6-6.tgz
```

第 78 章 DDL

1. 字段类型

```
CREATE TABLE "NEO"."TEST_DATE_TABLE"
(
  "COLUMN1" DATE DEFAULT sysdate NOT NULL ENABLE,
  "COLUMN2" TIMESTAMP (6) DEFAULT current_timestamp
) SEGMENT CREATION IMMEDIATE
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS
LOGGING
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS
2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL
DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "TS_NEO_DEF" ;

INSERT INTO "NEO"."TEST_DATE_TABLE" VALUES (DEFAULT, DEFAULT)
```

1.1. Date

首先说明Oracle Datetime 这样的字段 Date中就包含日期。

有时你会遇到只有日期，没有显示时间，可以使用下来两种方法解决。

方法一：使用 TO_CHAR

```
select TO_CHAR(COLUMN1, 'YYYY-MM-DD HH24:MI:SS') from
test_date_table;
```

方法二：使用

```
ALTER SESSION SET NLS_DATE_FORMAT='YYYY-MM-DD HH24:MI:SS';  
select COLUMN1 from test_date_table;
```

1.2. TIMESTAMP



2. 索引

修改表字段长度

```
ALTER TABLE  
  customer  
MODIFY  
  ( lastloginip varchar2(60) )  
;
```

例 78.1.

3. 表

Oracle 主键与其他数据库不同，例如MySQL中PK带有唯一索引功能，而Oracle需要自己创建唯一索引，否则PK数据是可能重复

```
CREATE TABLE "NEO"."PK_TEST_TABLE"
(
  "ID" VARCHAR2(20 BYTE) NOT NULL ENABLE,
  CONSTRAINT "PK_TEST_TABLE_PK" PRIMARY KEY ("ID")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 NOCOMPRESS
  LOGGING
  TABLESPACE "TS_NEO_DEF" ENABLE
) SEGMENT CREATION DEFERRED
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS
LOGGING
TABLESPACE "TS_NEO_DEF" ;
```

下面表，可以重复插入ID

```
CREATE TABLE "NEO"."PK_TEST_TABLE"
(
  "ID" VARCHAR2(20 BYTE) NOT NULL ENABLE,
  PRIMARY KEY ("ID")
)
```

3.2. 约束检查

实现MySQL那样的枚举类型

```
CREATE TABLE sizes (
  size ENUM('small', 'medium', 'large')
);
```

具体实现

```
CREATE TABLE sizes (  
  size VARCHAR2(10) CHECK( size IN ('small','medium','large') )  
);
```

4. SEQUENCE 序列

查看序列

```
SELECT  
SEQUENCE_NAME,MIN_VALUE,MAX_VALUE,INCREMENT_BY,LAST_NUMBER FROM  
USER_SEQUENCES;
```

创建序列

INCREMENT BY: 指定序列号之间的间隔, 该值可为正的或负的整数, 但不可为0. 序列为升序. 忽略该子句时, 缺省值为1.

START WITH: 指定生成的第一个序列号. 在升序时, 序列可从比最小值大的值开始, 缺省值为序列的最小值. 对于降序, 序列可由比最大值小的值开始, 缺省值为序列的最大值.

MAXVALUE: 指定序列可生成的最大值.

NOMAXVALUE: 为升序指定最大值为1027, 为降序指定最大值为-1.

MINVALUE: 指定序列的最小值.

NOMINVALUE: 为升序指定最小值为1. 为降序指定最小值为-1026.

NOCYCLE: 一直累加, 不循环

```
CREATE SEQUENCE "NEO"."NEO_SEQUENCE_ID";  
or  
CREATE SEQUENCE "NEO"."NEO_SEQUENCE_ID" MINVALUE 1 MAXVALUE  
99999999999999999999999999999999 INCREMENT BY 1 START WITH 1 CACHE  
20 NOORDER NOCYCLE ;
```

使用序列

下一个值

```
SELECT NEO_SEQUENCE_ID.NEXTVAL FROM DUAL;
```

当前值

```
SELECT NEO_SEQUENCE_ID.CURRVAL FROM DUAL;
```

删除缓存

```
DROP SEQUENCE "NEO"."NEO_SEQUENCE_ID";
```

第 79 章 程序包

1. 执行包中的过程

```
select package_name.function_name (15000) from dual;
```

```
declare  
    result number;  
begin  
    -- Call the function  
    result := package_name.function_name (15000);  
end;
```

第 80 章 数据库链接

Oracle 提供数据库链接功能，实现从当前数据库链接到其他数据库，从当前数据库操作其他数据库上的数据。

创建链接

```
CREATE DATABASE LINK dblinkname CONNECT TO dbuser IDENTIFIED BY
your_password USING 'orcl';

CREATE PUBLIC DATABASE LINK 数据链名称 CONNECT TO 登陆用户名
IDENTIFIED BY 密码 USING '(DESCRIPTION = (ADDRESS_LIST =
(ADDRESS = (PROTOCOL = TCP)(HOST = 对方Oracle服务器的IP地址)(PORT
= 端口号)) ) (CONNECT_DATA =(SERVICE_NAME = 对方Oracle服务器服务名)
))'
```

查询

```
select * from v$dblink;
select * from user_db_links;
select owner, db_link from dba_db_links;
```

删除链接

```
drop database link 数据链名称;
```

例 80.1. DB LINK

```
CREATE DATABASE LINK WEB
CONNECT TO WEB IDENTIFIED BY 0CfJly7y5YB060Cf5YBCf
USING 'orcl';

CREATE VIEW ACCOUNT AS SELECT username, password FROM
member@WEB;
```

```
select * from ACCOUNT where username="netkiller";
```


第 81 章 Oracle 监控

1. Oracle 日志的路径

```
SQL> select * from v$logfile;

   GROUP# STATUS   TYPE      MEMBER
-----
IS_
-----
          3      ONLINE  /opt/oracle/oradata/orcl/redo03.log
NO
          2      ONLINE  /opt/oracle/oradata/orcl/redo02.log
NO
          1      ONLINE  /opt/oracle/oradata/orcl/redo01.log
NO
SQL>
```

2. SQL 日志监控

```
SQL> select * from v$sql;

SQL> select * from v$sqlarea;
```

```
SQL> select sql_text from v$sql where rownum<10;

SQL_TEXT
-----
/* OracleOEM */          SELECT SEVERITY_INDEX,
CRITICAL_INCIDENTS, WARNING_INCIDENTS from v$incmeter_summary
SELECT LAST_LOAD_TIME FROM MGMT_TARGETS WHERE TARGET_GUID=:B1
SELECT LAST_LOAD_TIME FROM MGMT_TARGETS WHERE TARGET_GUID=:B1
SELECT LAST_LOAD_TIME FROM MGMT_TARGETS WHERE TARGET_GUID=:B1
SELECT LAST_LOAD_TIME FROM MGMT_TARGETS WHERE TARGET_GUID=:B1
SELECT LAST_LOAD_TIME FROM MGMT_TARGETS WHERE TARGET_GUID=:B1
SELECT LAST_LOAD_TIME FROM MGMT_TARGETS WHERE TARGET_GUID=:B1
SELECT LAST_LOAD_TIME FROM MGMT_TARGETS WHERE TARGET_GUID=:B1
SELECT LAST_LOAD_TIME FROM MGMT_TARGETS WHERE TARGET_GUID=:B1

9 rows selected.

SQL>
```

```
SQL> select sql_text from v$sqlarea where rownum<100;

SQL_TEXT
-----
/* OracleOEM */          SELECT SEVERITY_INDEX,
```

```

CRITICAL_INCIDENTS, WARNING_INCIDENTS from v$incmeter_summary
SELECT LAST_LOAD_TIME FROM MGMT_TARGETS WHERE TARGET_GUID=:B1
SELECT BLACKOUT_GUID, START_TIME, END_TIME, STATUS FROM
MGMT_BLACKOUT_WINDOWS WHERE TARGET_GUID=:B2 AND START_TIME <=
:B1
UPDATE MGMT_TARGETS SET LAST_LOAD_TIME=:B2 WHERE TARGET_GUID =
:B1 AND (LAST_LOAD_TIME < :B2 OR LAST_LOAD_TIME IS NULL)
SELECT ROWID FROM EMDW_TRACE_DATA WHERE LOG_TIMESTAMP < :B2 AND
CONTEXT_TYPE_ID = NVL(:B1 ,CONTEXT_TYPE_ID) ORDER BY ROWID ASC
SELECT SYSTEM_JOB, JOB_NAME, JOB_OWNER FROM MGMT_JOB WHERE
JOB_ID=:B1
update sys.job$ set this_date=:1 where job=:2
...
...
...
SELECT * FROM AQ_MNTR_MSGS_PERSQSUBS where rownum <=1

SQL_TEXT
-----
select timestamp, flags from fixed_obj$ where obj#=:1
SELECT STEP_STATUS FROM MGMT_JOB_EXECUTION WHERE STEP_ID=:B1

99 rows selected.

SQL>

```

查看 module

```

SQL> select module from v$sql group by module;

MODULE
-----
SQL Developer
Oracle Enterprise Manager.string history purge
Data Pump Worker
OEM.BoundedPool
SEVERITY EVALUATION
STREAMS

```

```

emagent_SQL_oracle_database
OEM.SystemPool
sqlplus@orcl.example.com (TNS V1-V3)
OMS
DBMS_SCHEDULER
Oracle Enterprise Manager.rollup
OEM.CacheModeWaitPool
OEM.DefaultPool
Oracle Enterprise Manager.metric error purge
SQL*Plus
Oracle Enterprise Manager.purge system performan
Oracle Enterprise Manager.purge system error log
MMON_SLAVE
emagent_AQMetrics
perl@orcl.example.com (TNS V1-V3)
JDBC Thin Client
Oracle Enterprise Manager.Metric Engine

EM_PING
Oracle Enterprise Manager.current metric purge

27 rows selected.

```

查询JDBC SQL操作日志

```

SQL> select module,first_load_time,sql_text from v$sql where
MODULE='JDBC Thin Client' and rownum<10 order by
first_load_time desc;

```

MODULE	FIRST_LOAD_TIME	SQL_TEXT
JDBC Thin Client	2016-02-29/16:47:35	INSERT INTO CUSTOMER VALUES (LPAD(CUSTOMER.NEXTVAL, 27, 0), :B8 , :B7 , :B6 , :B5 , SYSDATE, :B4 , :B3 , :B2 , :B1)
JDBC Thin Client	2016-02-29/10:08:25	SELECT * FROM LOTTERYS WHERE ID = :B1 FOR UPDATE NOWAIT
JDBC Thin Client	2016-02-29/10:08:25	SELECT COUNT(1) FROM ADMIN WHERE (FUNCTIONRIGHTS LIKE '%,' :B2

```
' ,%' ) AND LOGINNAME = :B1
JDBC Thin Client      2016-02-29/09:52:41      SELECT
SUM(C.AMOUNT) AMOUNT FROM LOTTERYS C WHERE FLAG = :1
JDBC Thin Client      2016-02-29/09:51:32      SELECT
COUNT(1) COUNT FROM CUSTOMER WHERE LOGINNAME = :1
JDBC Thin Client      2016-02-29/09:29:06      BEGIN ...
JDBC Thin Client      2016-02-29/09:29:06      SELECT ...
JDBC Thin Client      2016-02-29/09:28:15      SELECT ...
JDBC Thin Client      2016-02-29/09:25:34      SELECT FLAG
FROM ADMIN WHERE LOGINNAME = :B1

9 rows selected.
```

查看 SQL Developer 操作日志

```
SQL> select module,first_load_time,sql_text from v$sql where
MODULE='SQL Developer' and rownum<10 order by first_load_time
desc;

MODULE                                FIRST_LOAD_TIME                       SQL_TEXT
-----                                -
-----
SQL Developer      2016-02-29/16:18:42                       BEGIN
DBMS_OUTPUT.ENABLE() ; END;
SQL Developer      2016-02-29/14:14:18                       select * from
customer order by id desc
SQL Developer      2016-02-29/09:47:31                       SELECT TEXT FROM
SYS.DBA_SOURCE WHERE TYPE = :TYPE AND OWNER = :OWNER AND NAME
= :NAME ORDER BY LINE
SQL Developer      2016-02-29/09:30:44                       SELECT
/*OracleDatabaseImpl.ALL_PARTITIONING_QUERY*/ VALUE FROM
V$OPTION WHERE PARAMETER='Partitioning'
SQL Developer      2016-02-29/09:27:31                       select 1 from
dba_dependencies where 1=2
SQL Developer      2016-02-29/09:20:00                       ...
SQL Developer      2016-02-29/09:19:22                       ...
SQL Developer      2016-02-29/09:19:16                       select 1 from
dba_triggers where 1=2
SQL Developer      2016-02-29/09:19:14                       select
```

```
sys_context('USERENV','SESSIONID') from dual  
  
9 rows selected.
```

SQL*Plus 操作日志

```
SQL> select module,first_load_time,sql_text from v$sql where  
MODULE='SQL*Plus' order by first_load_time desc;
```

MODULE	FIRST_LOAD_TIME	SQL_TEXT
--------	-----------------	----------

```
-----  
-----  
-----  
-----  
-----  
-----  
-----  
-----  
SQL*Plus          2015-12-02/08:54:58          insert into  
sys.aud$( sessionid,entryid,statement,ntimestamp#,  
userid,userhost,terminal,action#,returncode,  
logoff$lread,logoff$pread,logoff$lwrite,logoff$dead,  
logoff$time,comment$text,spare1,clientid,sessioncpu,proxy$sid,u  
ser$guid, instance#,process#,auditid,dbid)  
values(:1,:2,:3,SYS_EXTRACT_UTC(SYSTIMESTAMP),  
:4,:5,:6,:7,:8, :9,:10,:11,:12,  
cast(SYS_EXTRACT_UTC(systimestamp) as  
date),:13,:14,:15,:16,:17,:18,          :19,:20,:21,:22)
```

2.1. 查询性能分析

```
SQL> SELECT * FROM (select PARSING_USER_ID,EXECUTIONS,SORTS,  
COMMAND_TYPE,DISK_READS,sql_text FROM v$sqlarea order BY  
disk_reads DESC )where ROWNUM<10 ;
```

PARSING_USER_ID	EXECUTIONS	SORTS	COMMAND_TYPE	DISK_READS
-----------------	------------	-------	--------------	------------


```
TO_CHAR((TO_DATE('2016-02-28', 'YYYY-MM-DD') + 1), 'YYYY-MM-DD')
```

```
9 rows selected.
```

2.2. IO性能分析

```
SELECT se.sid,se.serial#,pr.SPID,se.username,se.status,  
se.terminal,se.program,se.MODULE,se.sql_address,st.event,st.  
pltext,si.physical_reads, si.block_changes FROM v$session  
se,v$session_wait st,  
v$sess_io si,v$process pr WHERE st.sid=se.sid AND st.sid=si.sid  
AND se.PADDR=pr.ADDR AND se.sid>6 AND st.  
wait_time=0 AND st.event NOT LIKE '%SQL%' ORDER BY  
physical_reads DESC
```


3. session

查看用户正在执行的SQL

```
SQL> SELECT osuser, username, sql_text from v$session a, v$sqltext b
where a.sql_address =b.address order by address, piece;
```

OSUSER	USERNAME	SQL_TEXT
oracle	SYSMAN	BEGIN
EMD_NOTIFICATION.QUEUE_READY(:1, :2, :3);		END;
oracle	SYS	SELECT
osuser, username, sql_text from v\$session a, v\$sqltext b		where
oracle	SYS	
a.sql_address =b.address order by address, piece		
linea	Neo	UPDATE
gt_scheduler_config SET report_date = TO_DATE(:1 , 'yyyy-M		
linea	Neo	M-dd
HH24:mi:ss') WHERE id = :2		

```
SQL> select sid,username,sql_text from v$session
a,v$sqltext_with_newlines b where a.SQL_ADDRESS=b.ADDRESS;
```

SID	USERNAME	SQL_TEXT
177	SYSMAN	BEGIN
EMD_NOTIFICATION.QUEUE_READY(:1, :2, :3);		END;
95	SYS	lines b where
a.SQL_ADDRESS=b.ADDRESS		
95	SYS	select sid,username,sql_text
from v\$session a,v\$sqltext_with_new		

4. SQL Monitoring Report

监控级别

```
SQL> show parameter statistics_level
```

NAME	TYPE	VALUE
statistics_level	string	TYPICAL

```
SQL> select
statistics_name,session_status,system_status,activation_level,session_se
ttable from v$statistics_level where statistics_name = 'SQL Monitoring';
```

STATISTICS_NAME	SESSION_	SYSTEM_S	ACTIVAT	SES
SQL Monitoring	ENABLED	TYPICAL	YES	ENABLED

```
SQL> show parameter control_manage
```

NAME	TYPE	VALUE
control_management_pack_access	string	DIAGNOSTIC+TUNING

强制对某个SQL使用实时监控

```
select /*+ monitor */ count(*) from emp where sal > 5000;
```

禁止实时监控:

```
select /*+ no_monitor */ count(*) from emp where sal > 5000;
```

SQL监控报表

```
SQL> select dbms_sqltune.report_sql_monitor from dual;
```

```
REPORT_SQL_MONITOR
```

```
-----  
-----
```

```
SQL Monitoring Report
```

```
SQL Text
```

```
-----
```

```
DECLARE job BINARY
```

第 82 章 Installing Oracle Database

1. Installing Oracle Database 10g Release 2 on Linux x86

reference:

http://www.oracle.com/technology/pub/articles/smiley_10gdb_install.html

To make these changes, cut and paste the following commands as root:

过程 82.1. Configure linux step by step

1. Verifying Your Installation

```
rpm -q binutils compat-db control-center gcc gcc-c++ glibc
glibc-common \
gnome-libs libstdc++ libstdc++-devel make pdksh sysstat
xscreensaver libaio openmotif21
```

installing package

```
yum install compat-gcc-32 compat-gcc-32-c++ compat-gcc-32-
g77 compat-libf2c-32 compat-libstdc++-296 compat-
libstdc++-33 compat-db compat-readline43
```

2. Verifying System Requirements

```
grep MemTotal /proc/meminfo
grep SwapTotal /proc/meminfo
```

Swap = mem * 2

3. Create the Oracle Groups and User Account

```
groupadd oinstall
groupadd dba
useradd -m -g oinstall -G dba oracle
passwd oracle
id oracle
```

4. Create Directories

```
mkdir -p /u01/app/oracle
chown -R oracle:oinstall /u01/app/oracle
chmod -R 775 /u01/app/oracle
```

5. Configuring the Linux Kernel Parameters

```
cat >> /etc/sysctl.conf <<EOF
kernel.shmall = 2097152
kernel.shmmax = 536870912
kernel.shmmni = 4096
kernel.sem = 250 32000 100 128
fs.file-max = 65536
net.ipv4.ip_local_port_range = 1024 65000
net.core.rmem_default=262144
net.core.wmem_default=262144
net.core.rmem_max=262144
net.core.wmem_max=262144
EOF
/sbin/sysctl -p
```

Run the following commands as root to verify your settings:

```
/sbin/sysctl -a | grep shm
/sbin/sysctl -a | grep sem
```

```
/sbin/sysctl -a | grep file-max  
/sbin/sysctl -a | grep ip_local_port_range  
/sbin/sysctl -a | grep rmem_default  
/sbin/sysctl -a | grep rmem_max  
/sbin/sysctl -a | grep wmem_default  
/sbin/sysctl -a | grep wmem_max
```

6. Setting Shell Limits for the oracle User

```
cat >> /etc/security/limits.conf <<EOF  
oracle soft nproc 2047  
oracle hard nproc 16384  
oracle soft nofile 1024  
oracle hard nofile 65536  
EOF
```

7. /etc/profile

```
cat >> /etc/profile <<EOF  
if [ \${USER} = "oracle" ]; then  
  if [ \${SHELL} = "/bin/ksh" ]; then  
    ulimit -p 16384  
    ulimit -n 65536  
  else  
    ulimit -u 16384 -n 65536  
  fi  
  umask 022  
fi  
EOF
```

```
cat >> /etc/csh.login <<EOF  
if ( \${USER} == "oracle" ) then  
  limit maxproc 16384  
  limit descriptors 65536  
  umask 022  
endif  
EOF
```

8. .bash_profile

```
# su - oracle
$ vim .bash_profile

export ORACLE_BASE=/u01/app/oracle
export ORACLE_HOME=$ORACLE_BASE/product/10.2.0.1
export ORACLE_SID=orcl
export PATH=$PATH:$HOME/bin:$ORACLE_HOME/bin
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/usr/lib
```

过程 82.2. Installing Oracle

1. 编辑 /10201_database_linux32/database/install/oraparam.ini 添加

```
vim 10201_database_linux32/database/install/oraparam.ini

### #[Certified Versions]

Linux=redhat-3,SuSE-9,redhat-4,centos-5,UnitedLinux-
1.0,asianux-1,asianux-2

[Linux-centos-5.1-optional]
TEMP_SPACE=80
SWAP_SPACE=150
MIN_DISPLAY_COLORS=256
```

2. install

```
gunzip xxxx.cpio.gz
cpio -idmv < xxxx.cpio
export LANG=en_US
./runInstaller
```

3. dbstart

```
# su - oracle  
# dbstart
```

提示打开 /ade/vikrkuma_new/oracle/bin/tnslsnr 失败

编辑 /u01/app/oracle/product/10.2.0.1/bin/dbstart

```
# Set this to bring up Oracle Net Listener  
ORACLE_HOME_LISTNER=/ade/vikrkuma_new/oracle
```

应该是在78行，将其改为：

```
# Set this to bring up Oracle Net Listener  
ORACLE_HOME_LISTNER=$ORACLE_HOME
```

/etc/oratable 将最后一行的最后一个字符由“N”改为“Y”

```
orcl: /u01/app/oracle/product/10.2.0.1:Y
```

过程 82.3. Configuring Storage

1. Partition the Disks

```
fdisk -l /dev/sdb
```

2. Filesystems

ZFS or btrfs

3. Create the Mount Point

```
mkdir /u02
```

Add the New Filesystem to /etc/fstab

```
/dev/sdb1 /u02 xfs defaults 1 1
```

Mount the New Filesystem

```
mount /u02  
df -h /u02
```

4. Create Oracle Directories and Set Permissions

```
mkdir -p /u02/oradata/demo1  
chown -R oracle:oinstall /u02/oradata  
chmod -R 775 /u02/oradata
```

5. Create a New Tablespace in the New Filesystem

```
Ex:  
$ sqlplus  
  
SQL*Plus: Release 10.2.0.1.0 - Production on Sun Nov 27
```

15:50:50 2005

Copyright (c) 1982, 2005, Oracle. All rights reserved.

Enter user-name: system

Enter password:

Connected to:

Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 -
Production

With the Partitioning, OLAP and Data Mining options

```
SQL> create tablespace data1
  2  datafile '/u02/oradata/demo1/data1_01.dbf' size 100m
  3  extent management local
  4  segment space management auto;
```

Tablespace created.

Now you can use the new tablespace to store database
objects such as tables and indexes.

Ex:

```
SQL> create table demotab (id number(5) not null primary
key,
  2  name varchar2(50) not null,
  3  amount number(9,2))
  4  tablespace data1;
```

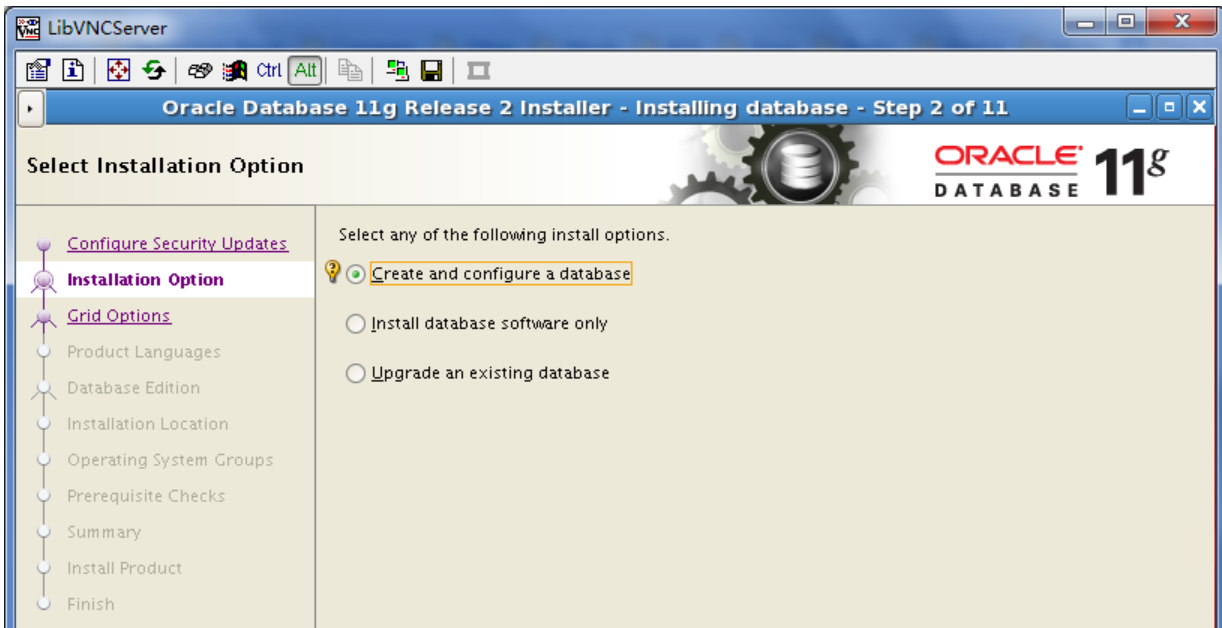
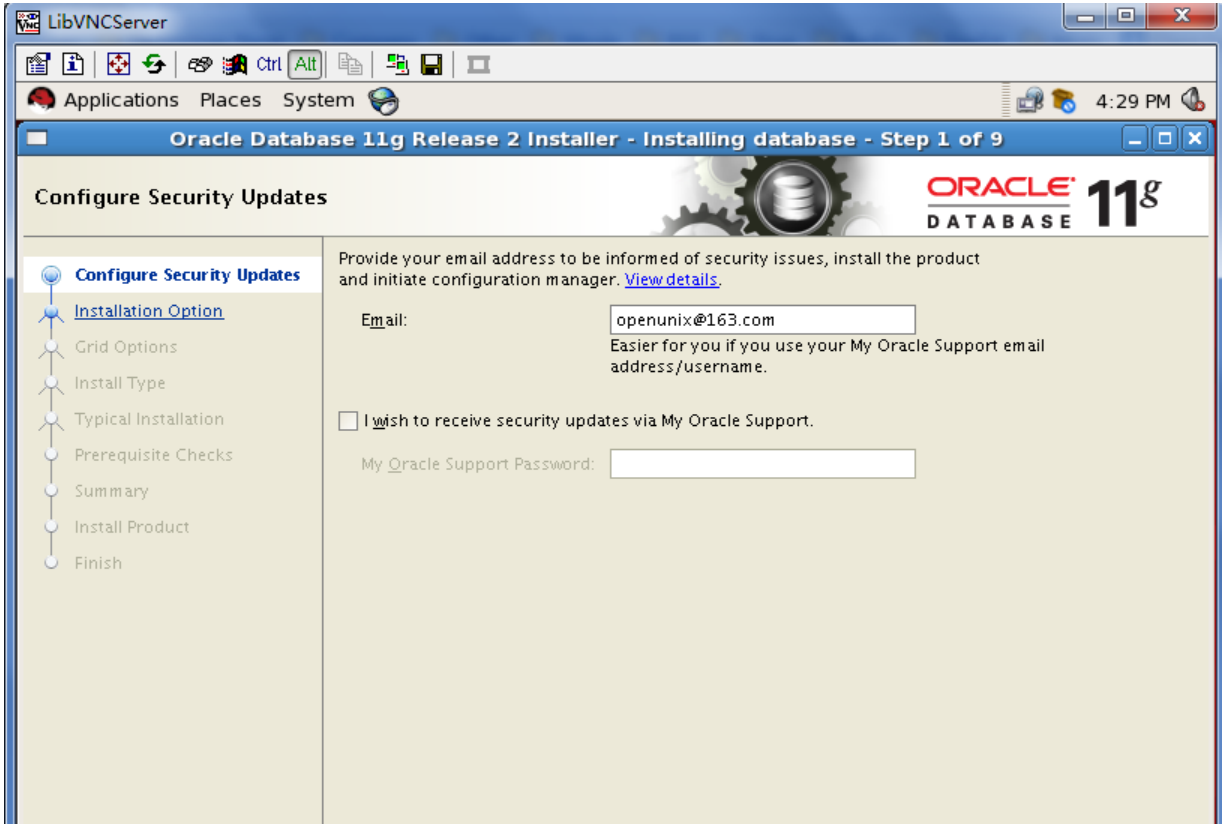
Table created.

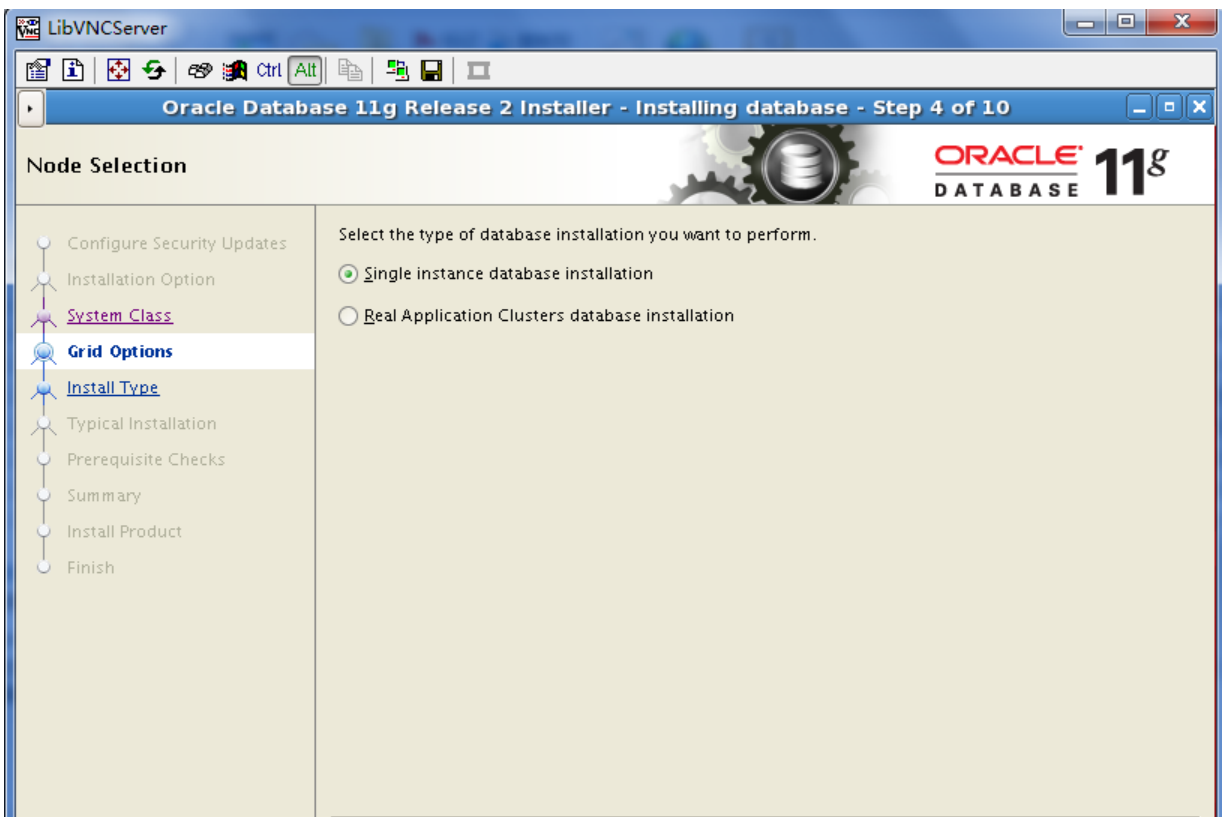
2. 11gR2

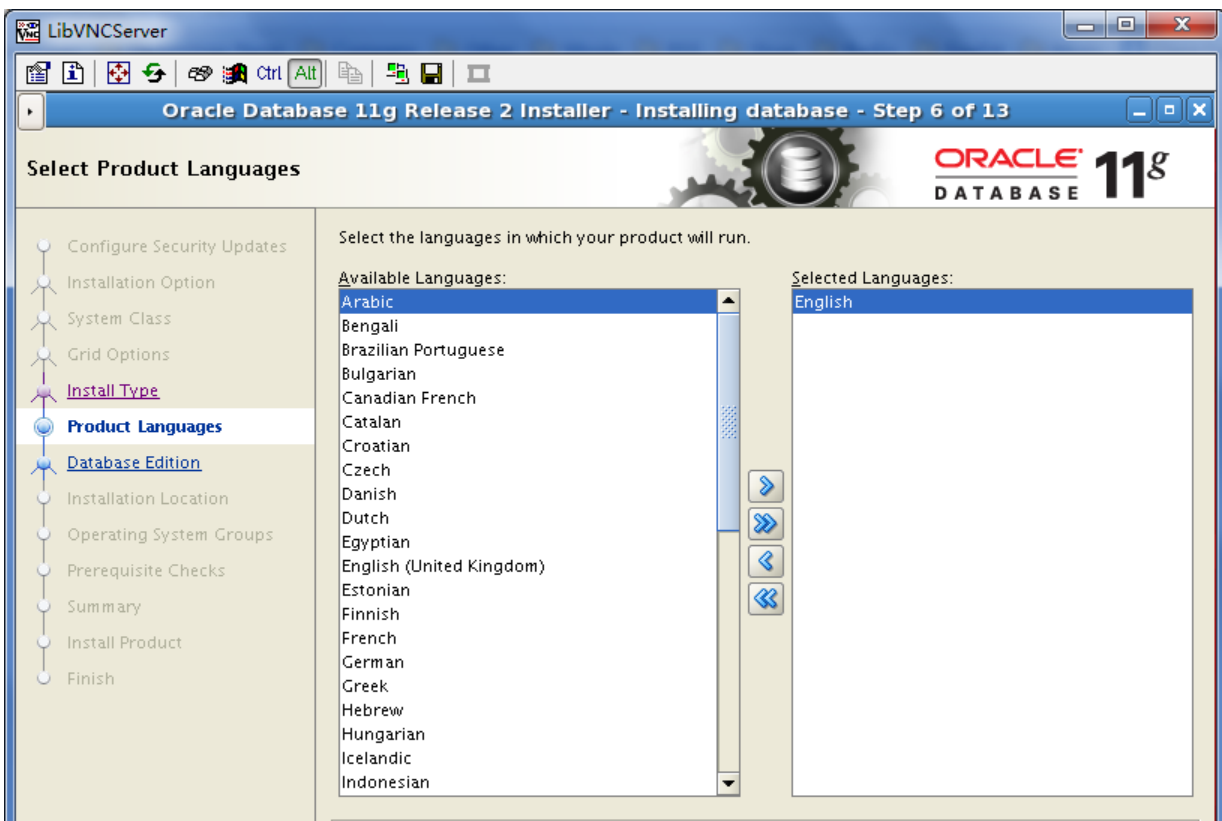
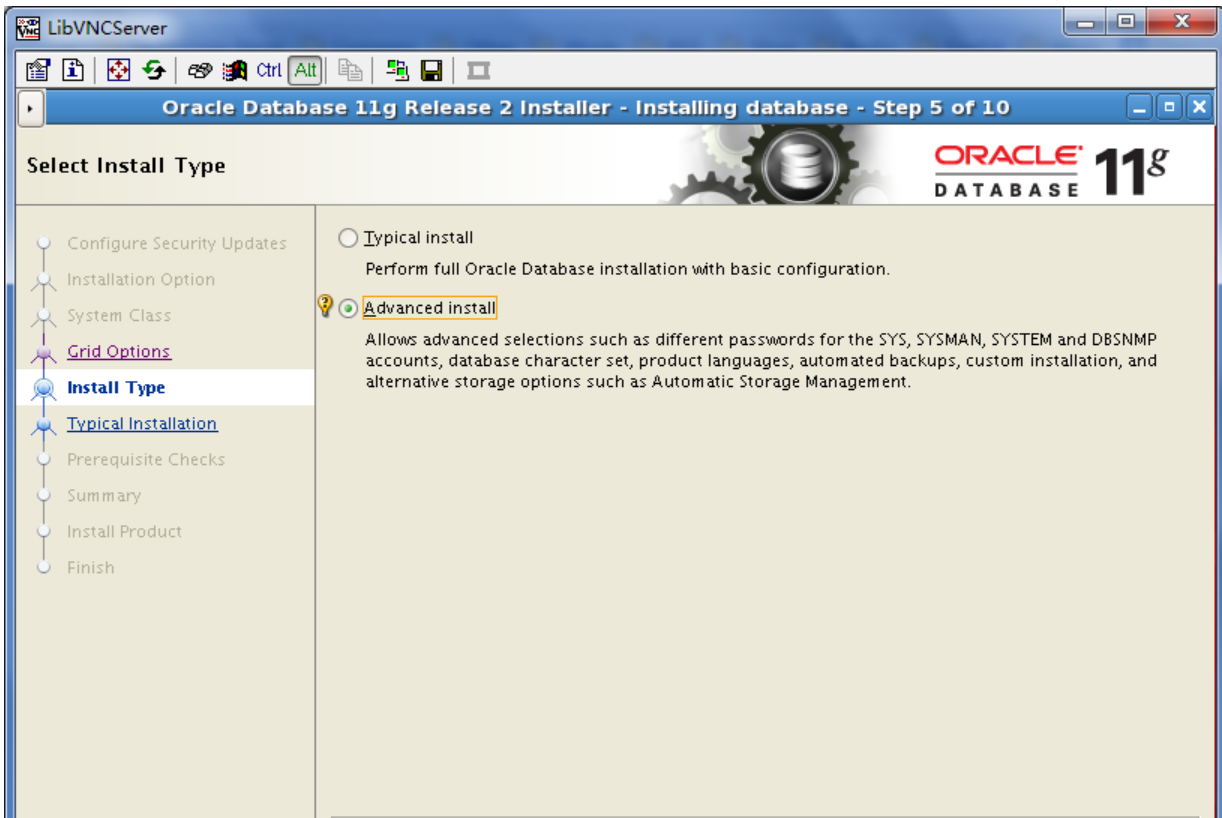
```
unzip linux.x64_11gR2_database_1of2.zip
unzip linux.x64_11gR2_database_2of2.zip

[oracle@wcs ~]$ vi .bash_profile
export TMP=/tmp
export TMPDIR=$TMP
export ORACLE_BASE=/opt/oracle/
export ORACLE_HOME=$ORACLE_BASE/product/11.2.0/dbhome_1
export ORACLE_SID=orcl
export ORACLE_TERM=xterm
export ORACLE_HOME_LISTNER=$ORACLE_HOME
export PATH=$ORACLE_HOME/bin:$PATH
export
LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib64:/usr/lib64:/usr/local/li
b64:/usr/X11R6/lib64/
export
CLASSPATH=$ORACLE_HOME/JRE:$ORACLE_HOME/jlib:$ORACLE_HOME/rdbms/
jlib
export LD_ASSUME_KERNEL=2.6.18
export NLS_LANG="AMERICAN_AMERICA.ZHS16GBK"
```

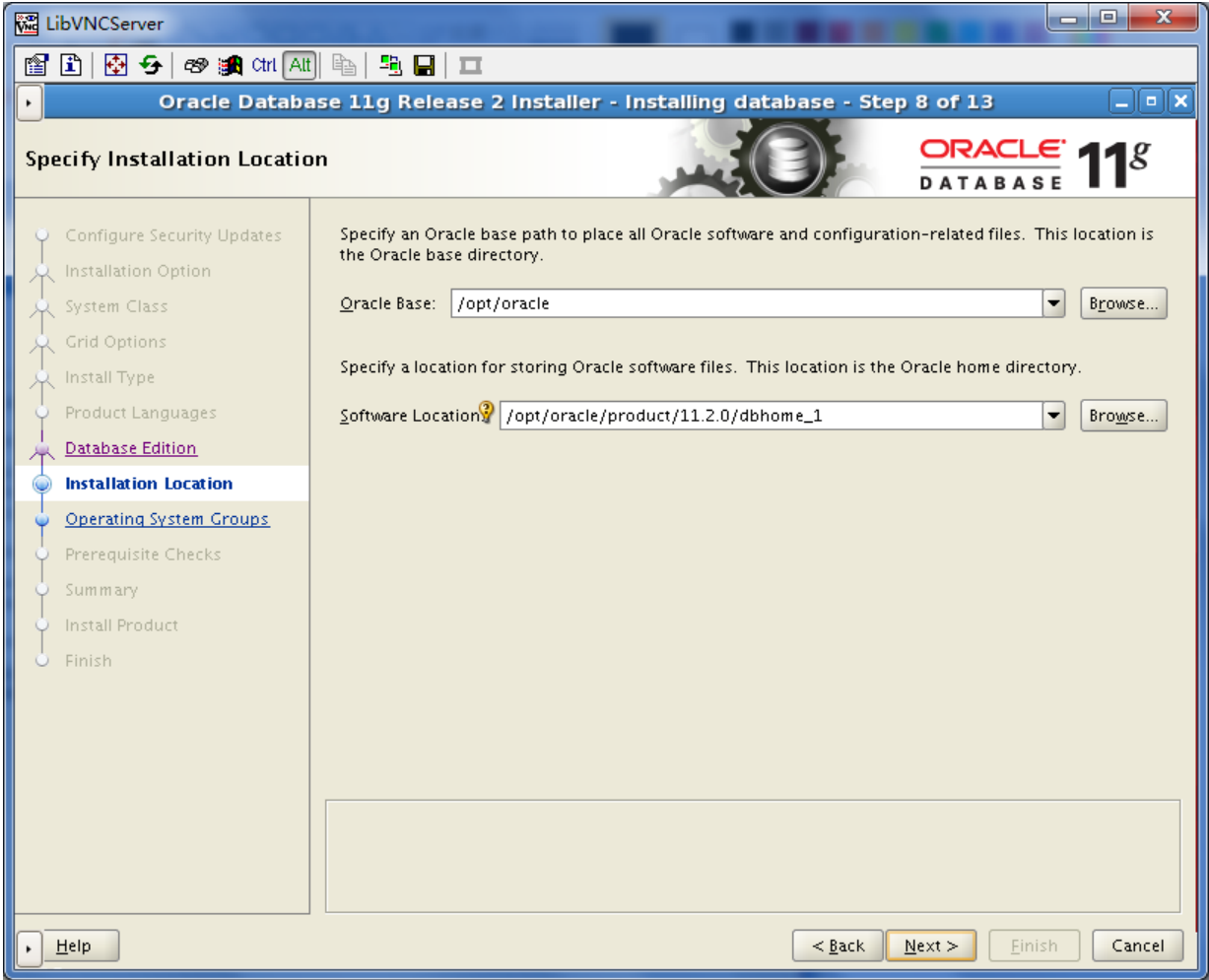
```
cat >> /etc/sysctl.conf <<EOF
fs.aio-max-nr = 3145728
fs.file-max = 6815744
kernel.shmall = 1073741824
kernel.shmmax = 4398046511104
kernel.shmmni = 4096
kernel.sem = 250 32000 100 142
net.ipv4.ip_local_port_range = 9000 65500
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 1048576
EOF
/sbin/sysctl -p
```

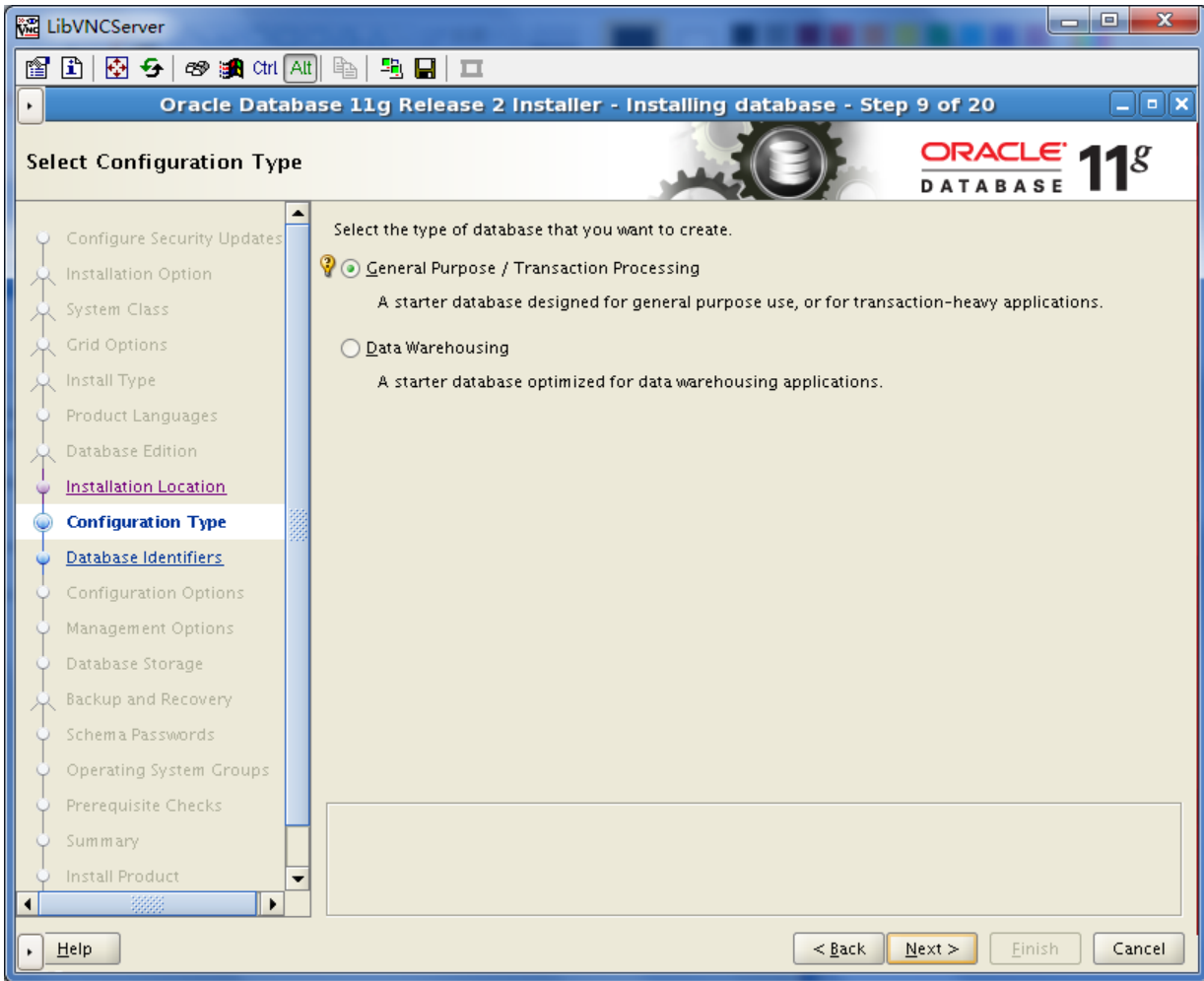


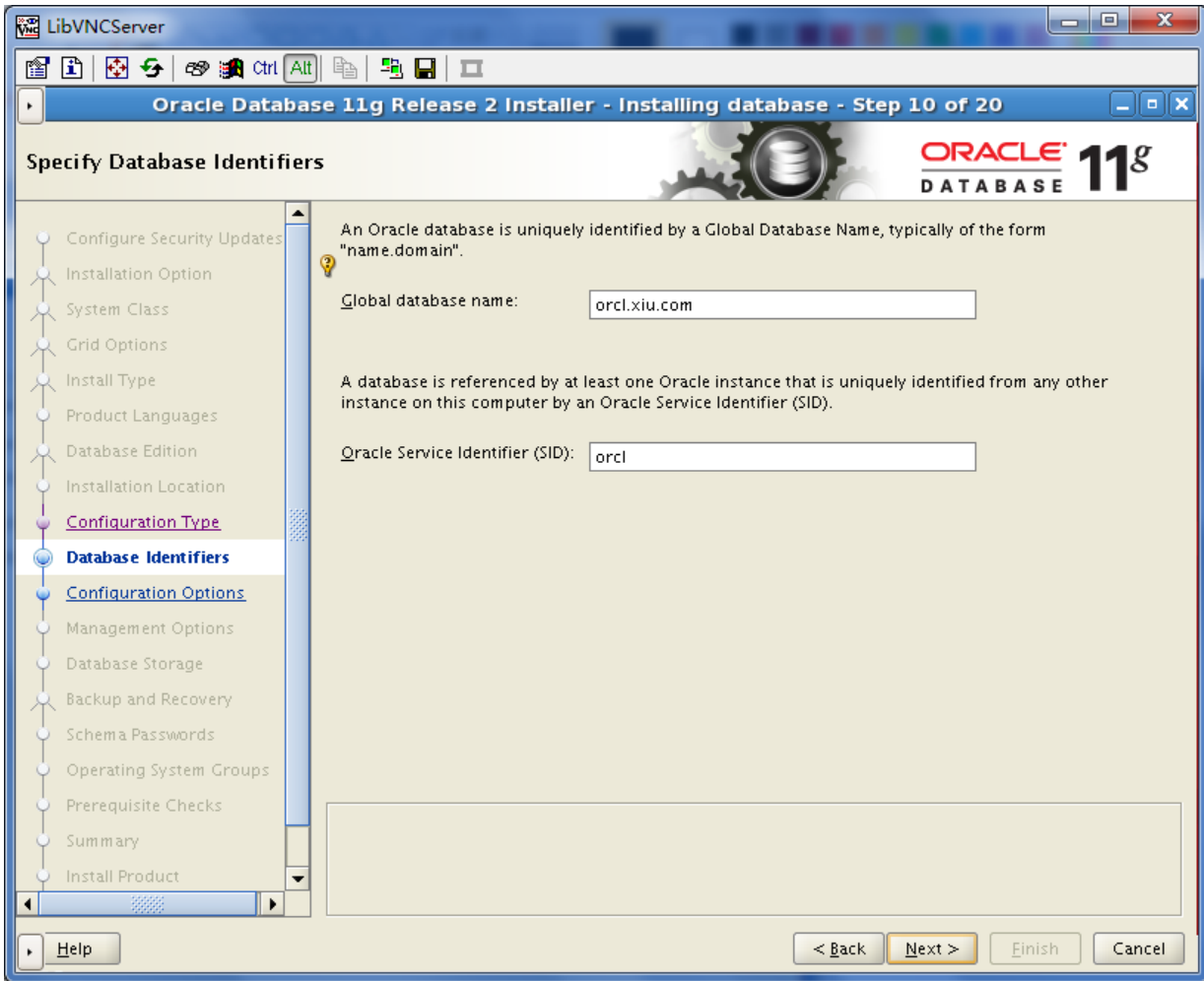


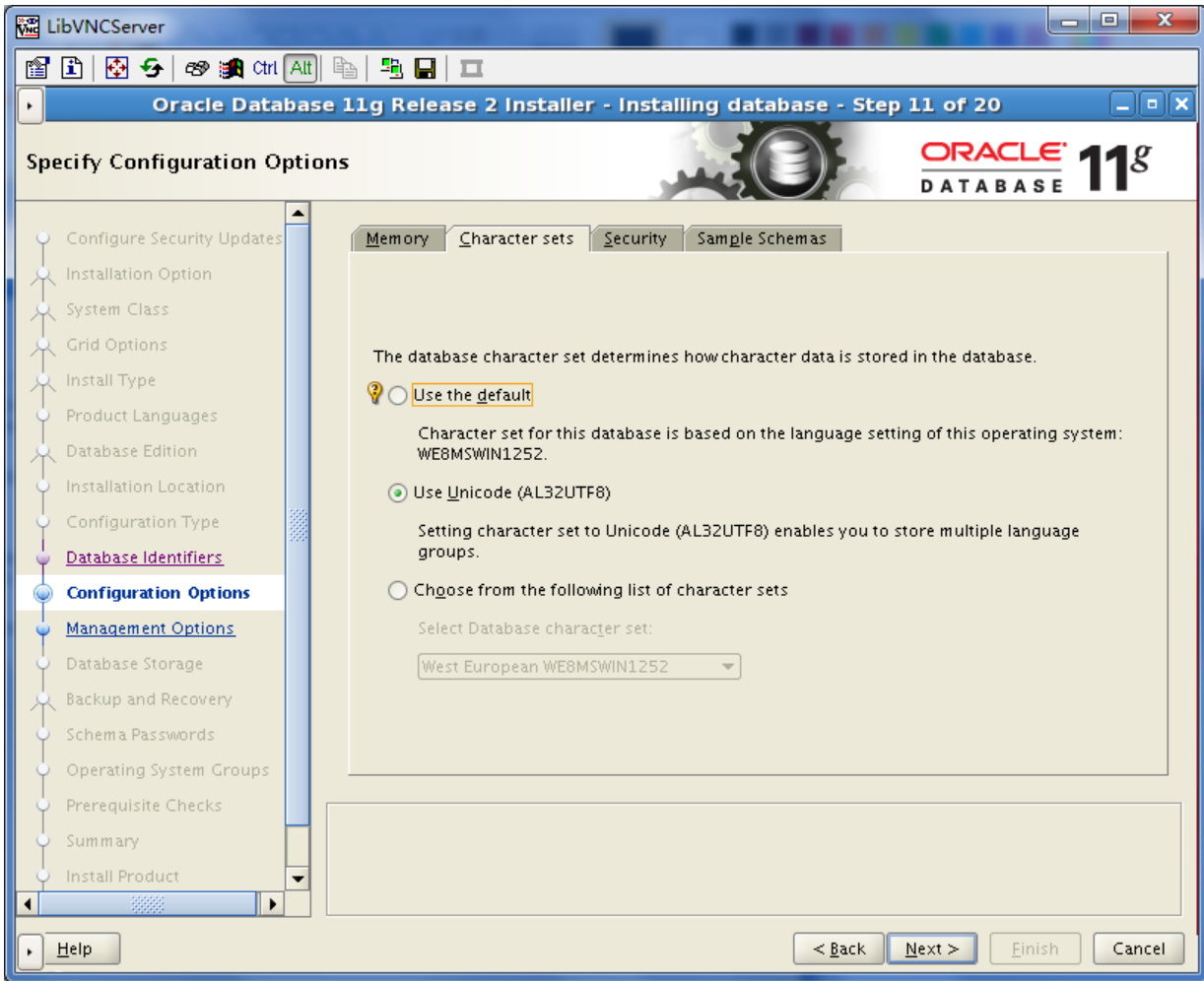


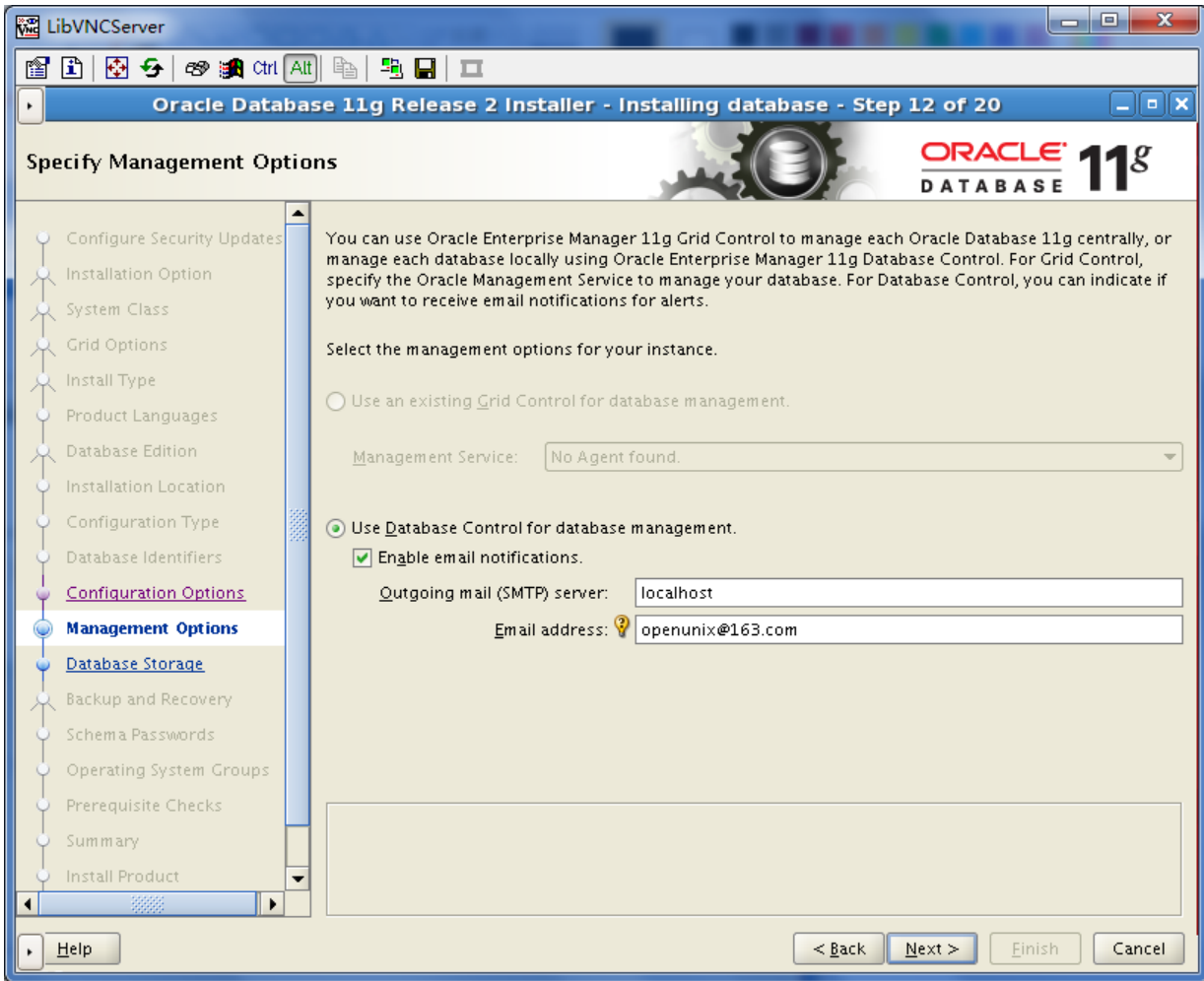


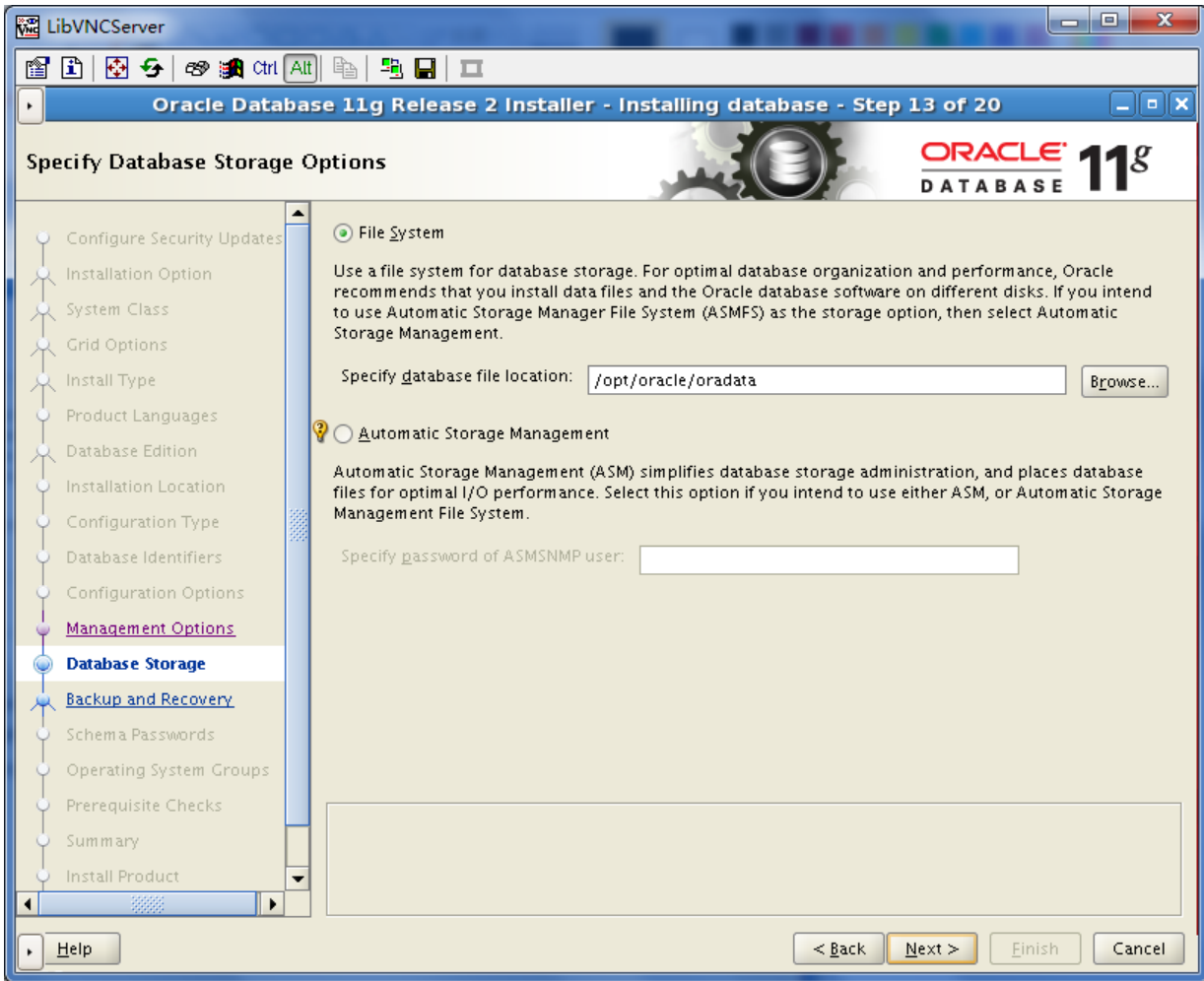


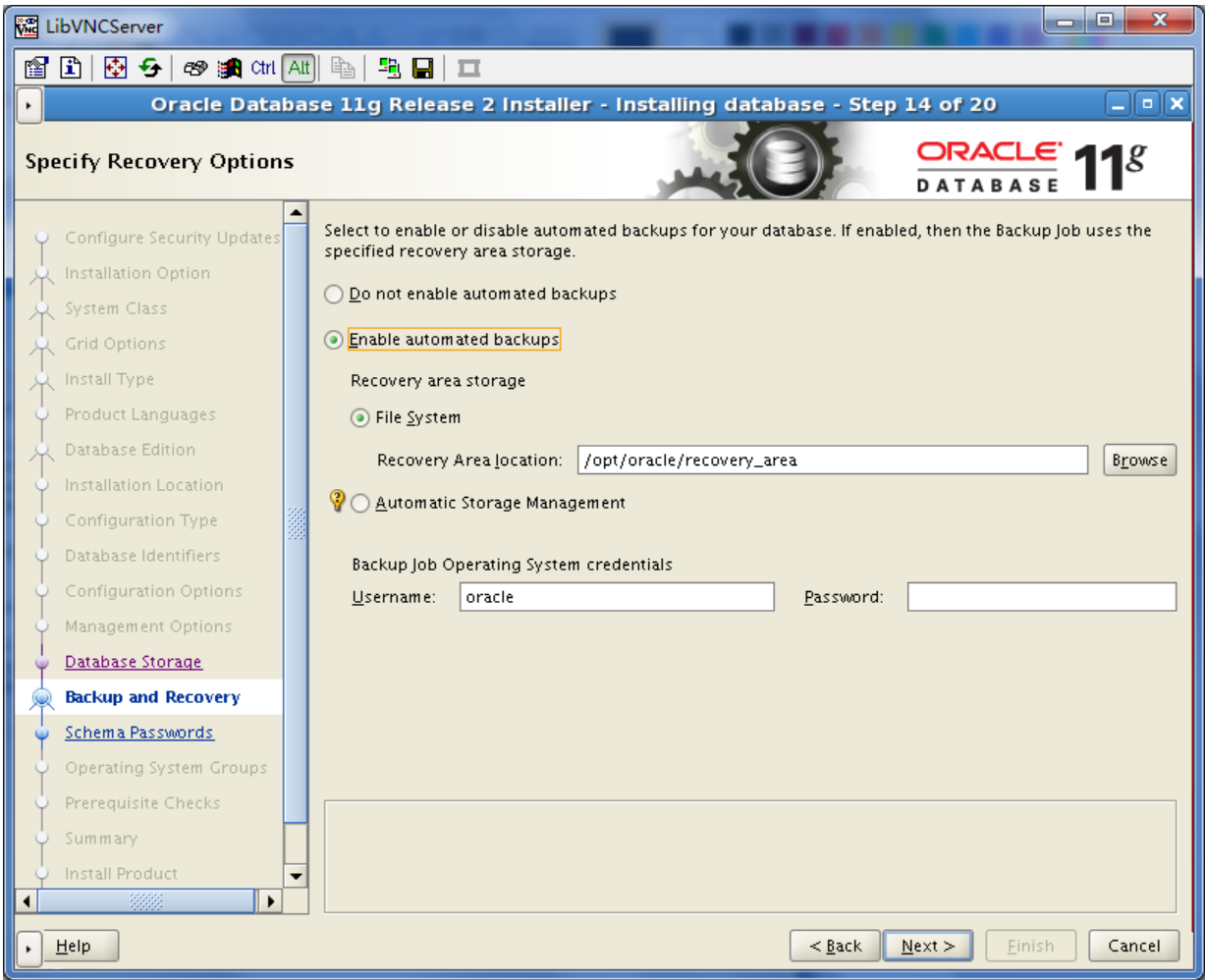


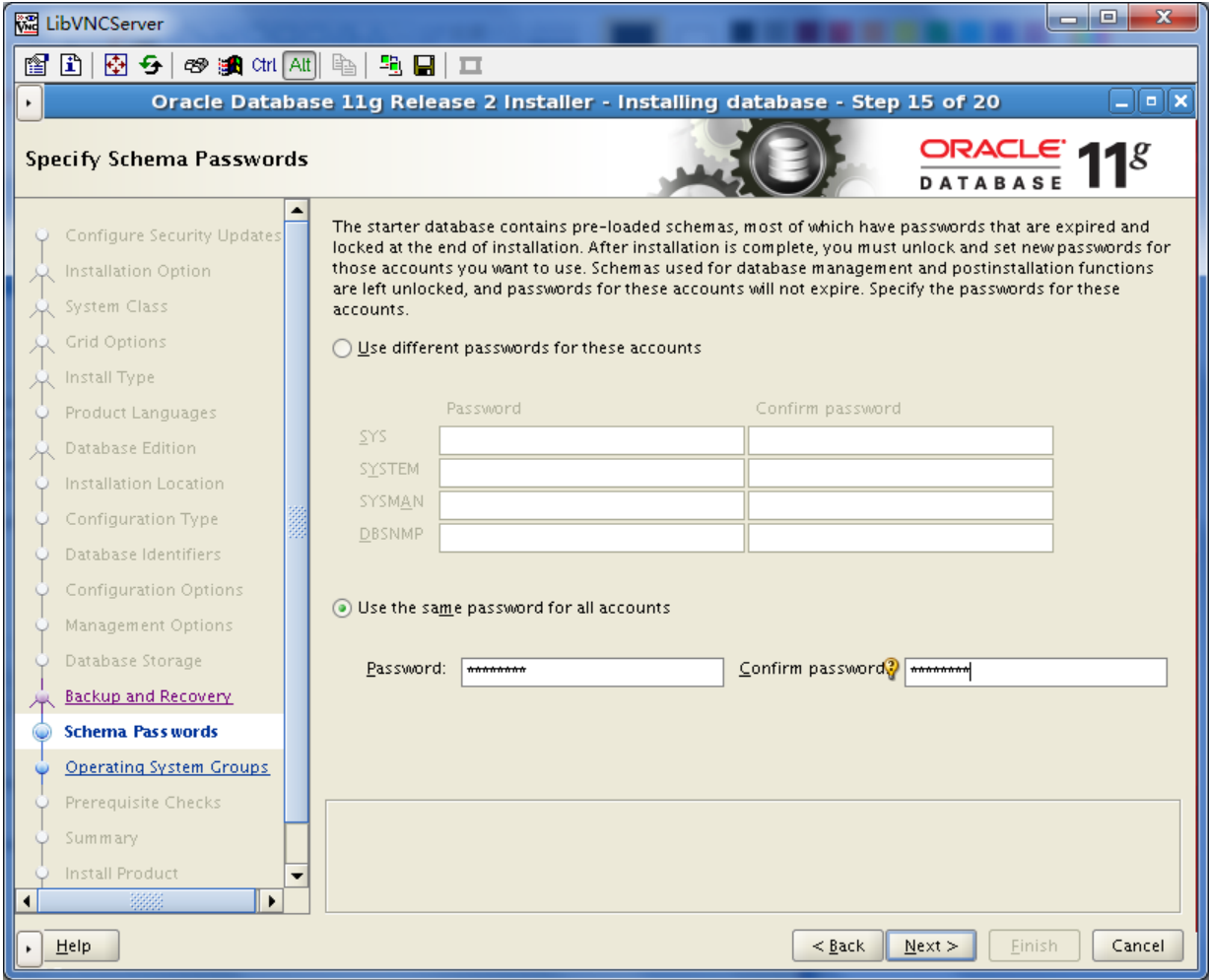


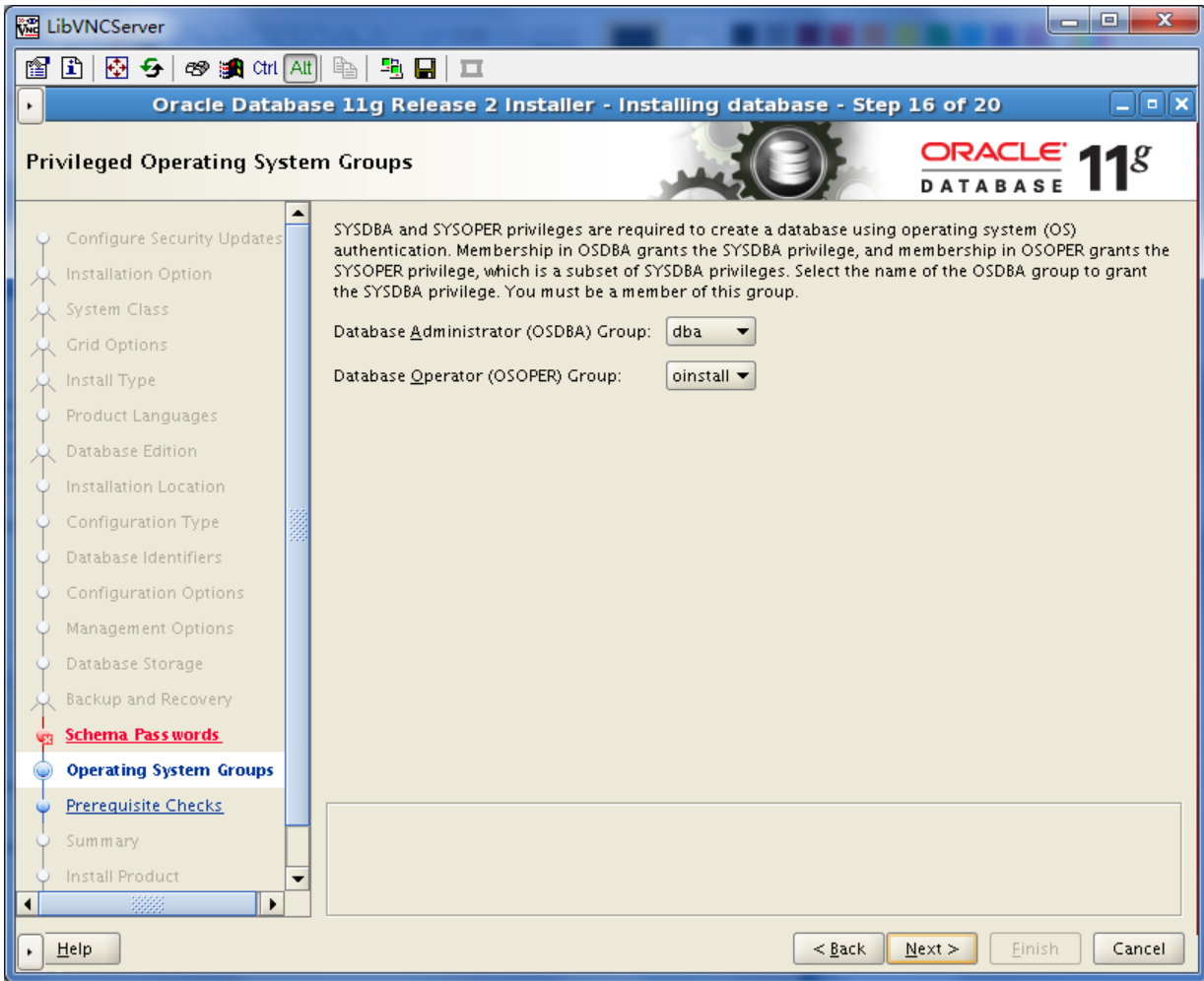


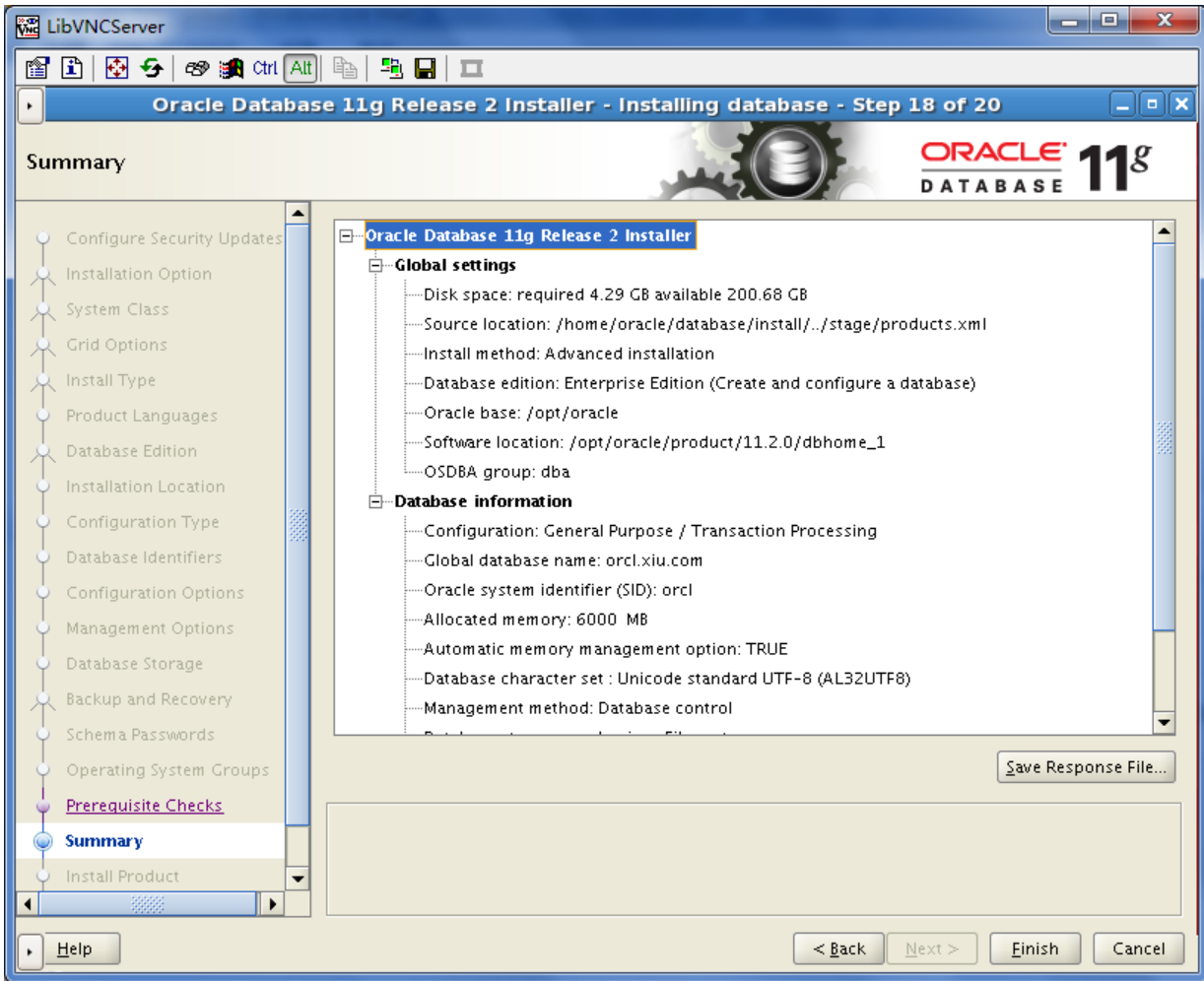


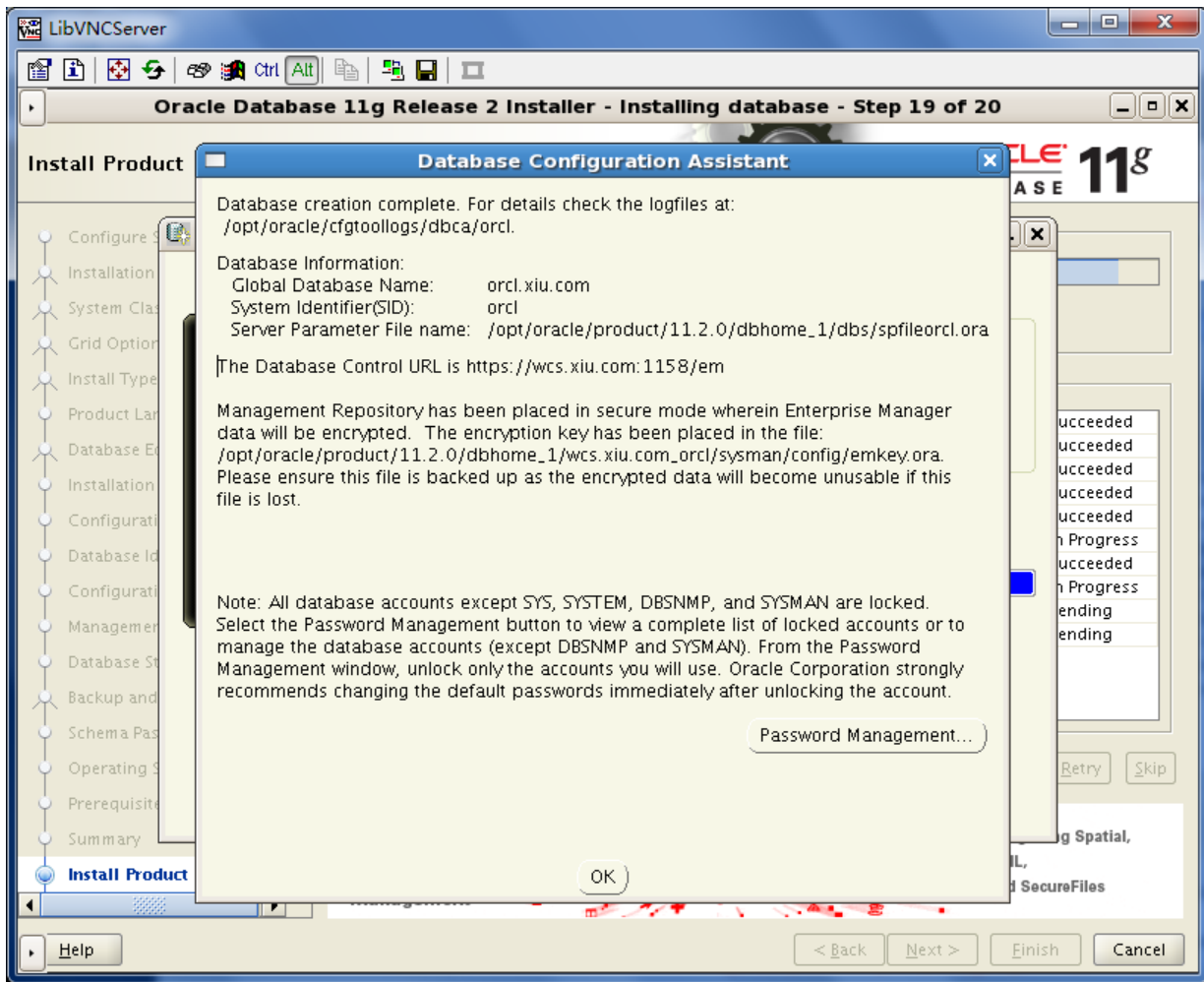


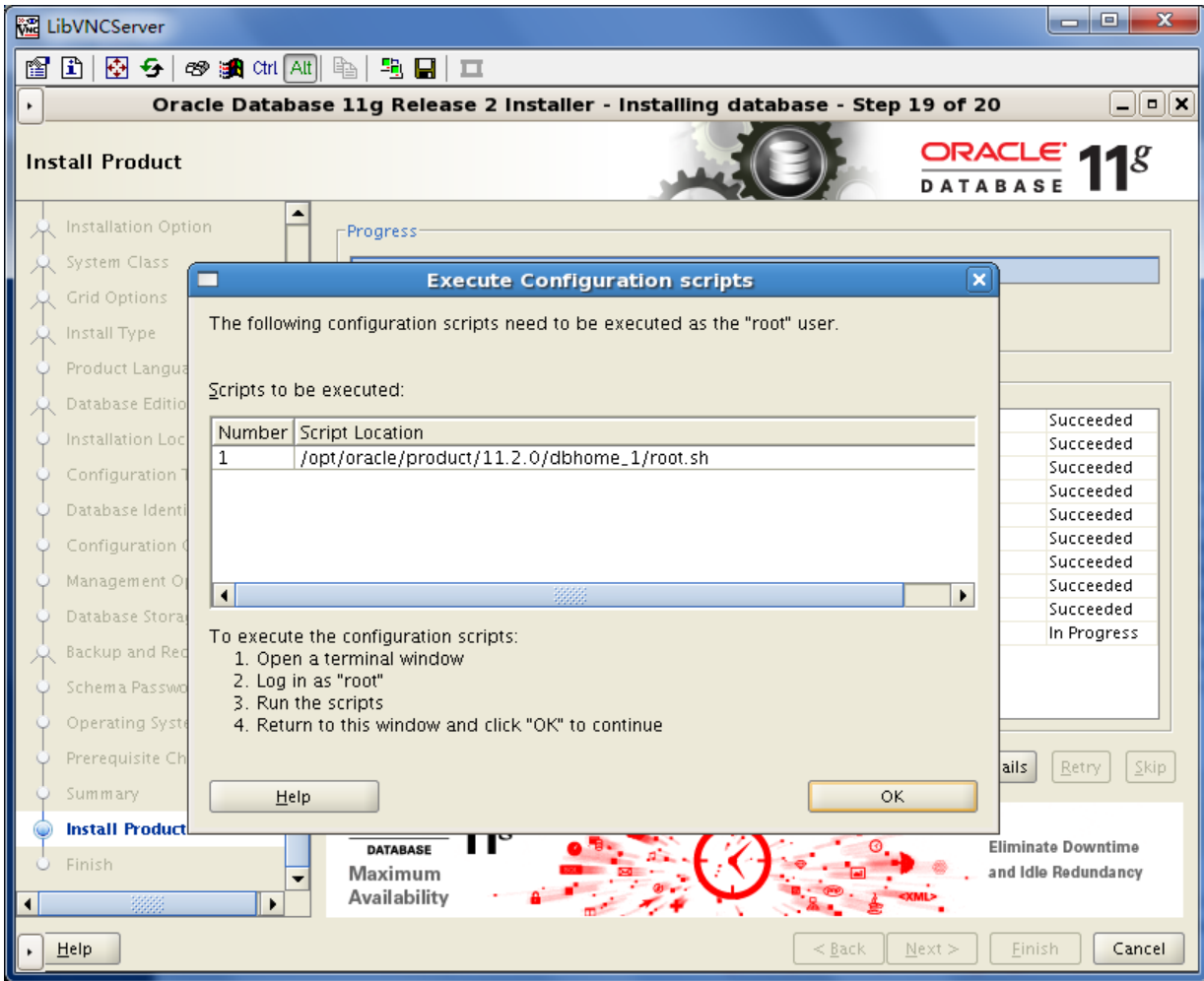


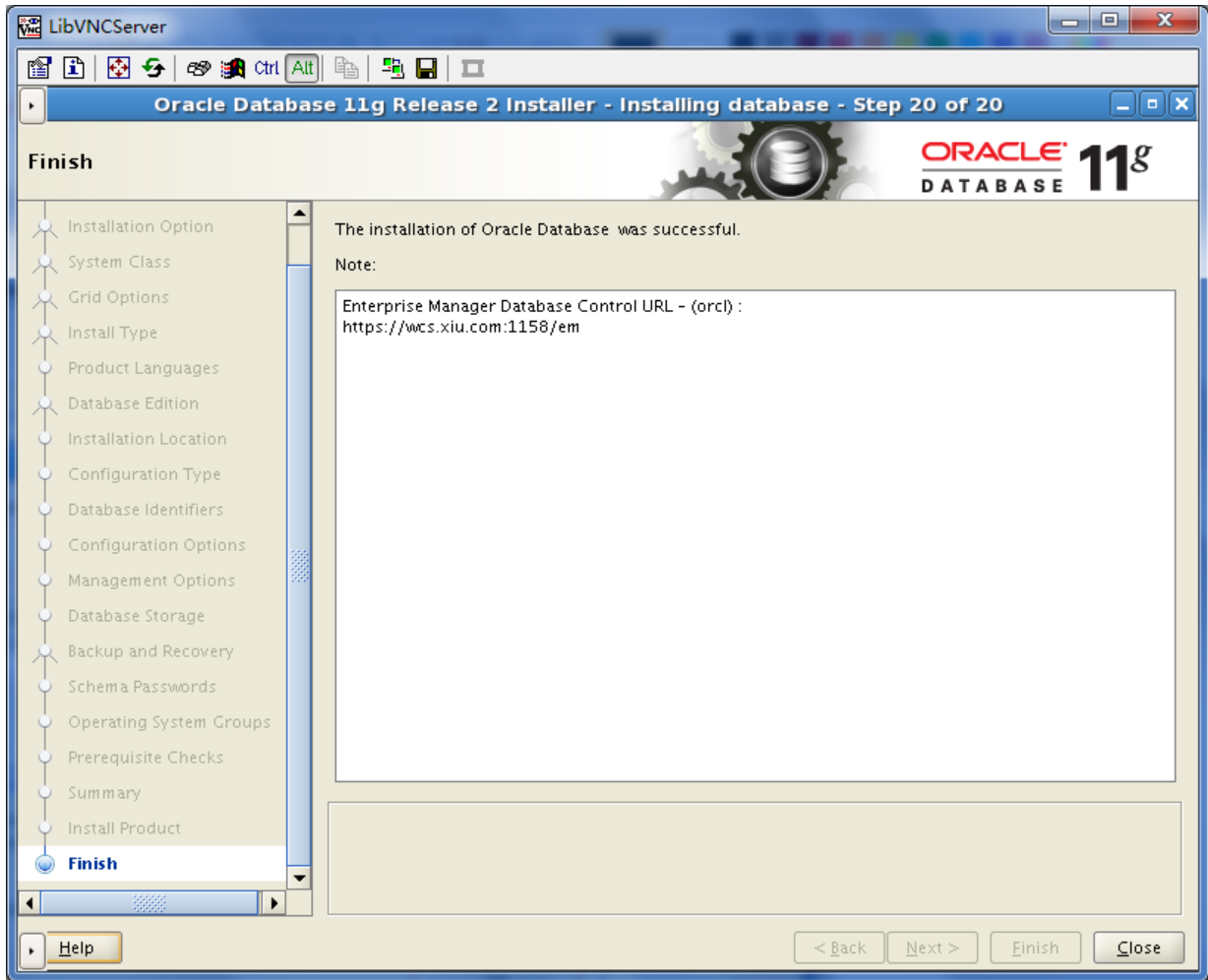












切换到root用户运行下面脚本

```
/opt/oracle/oraInventory/orainstRoot.sh  
/opt/oracle/product/11.2.0.1/client/root.sh
```

```
[root@oracle ~]# /opt/oracle/oraInventory/orainstRoot.sh  
Changing permissions of /opt/oracle/oraInventory.  
Adding read,write permissions for group.  
Removing read,write,execute permissions for world.  
  
Changing groupname of /opt/oracle/oraInventory to oinstall.  
The execution of the script is complete.  
  
[root@oracle ~]# /opt/oracle/product/11.2.0/dbhome_1/root.sh
```

Running Oracle 11g root.sh script...

The following environment variables are set as:

ORACLE_OWNER= oracle

ORACLE_HOME= /opt/oracle/product/11.2.0/dbhome_1

Enter the full pathname of the local bin directory:

[/usr/local/bin]:

Copying dbhome to /usr/local/bin ...

Copying oraenv to /usr/local/bin ...

Copying coraenv to /usr/local/bin ...

Creating /etc/oratab file...

Entries will be added to the /etc/oratab file as needed by
Database Configuration Assistant when a database is created

Finished running generic part of root.sh script.

Now product-specific root actions will be performed.

Finished product-specific root actions.

3. Installing Oracle Client 11g

orains.sh

```
#!/bin/bash
groupadd oinstall
groupadd dba
useradd -m -g oinstall -G dba oracle
echo "oracle:oracle" | chpasswd
id oracle
mkdir -p /opt/oracle
chown oracle.oinstall /opt/oracle
```

```
cat >> /etc/sysctl.conf <<EOF
kernel.shmall = 2097152
kernel.shmmax = 536870912
kernel.shmmni = 4096
kernel.sem = 250 32000 100 128
fs.file-max = 65536
net.ipv4.ip_local_port_range = 32768 61000
net.core.rmem_default=262144
net.core.wmem_default=262144
net.core.rmem_max=4194304
net.core.wmem_max=262144
EOF
/sbin/sysctl -p
```

Run the following commands as root to verify your settings:

```
/sbin/sysctl -a | grep shm
/sbin/sysctl -a | grep sem
/sbin/sysctl -a | grep file-max
/sbin/sysctl -a | grep ip_local_port_range
/sbin/sysctl -a | grep rmem_default
/sbin/sysctl -a | grep rmem_max
/sbin/sysctl -a | grep wmem_default
```

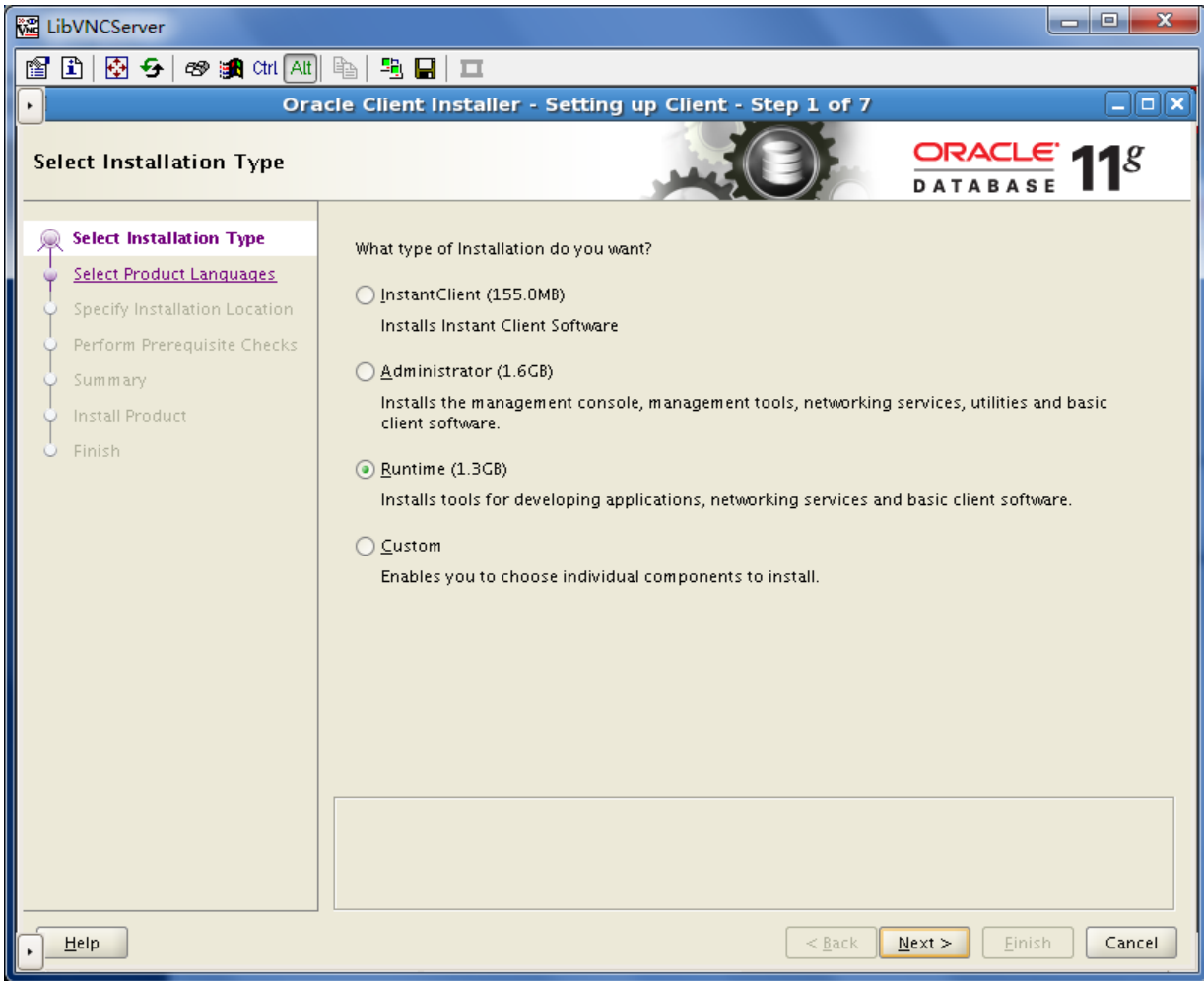
```
/sbin/sysctl -a | grep wmem_max
```

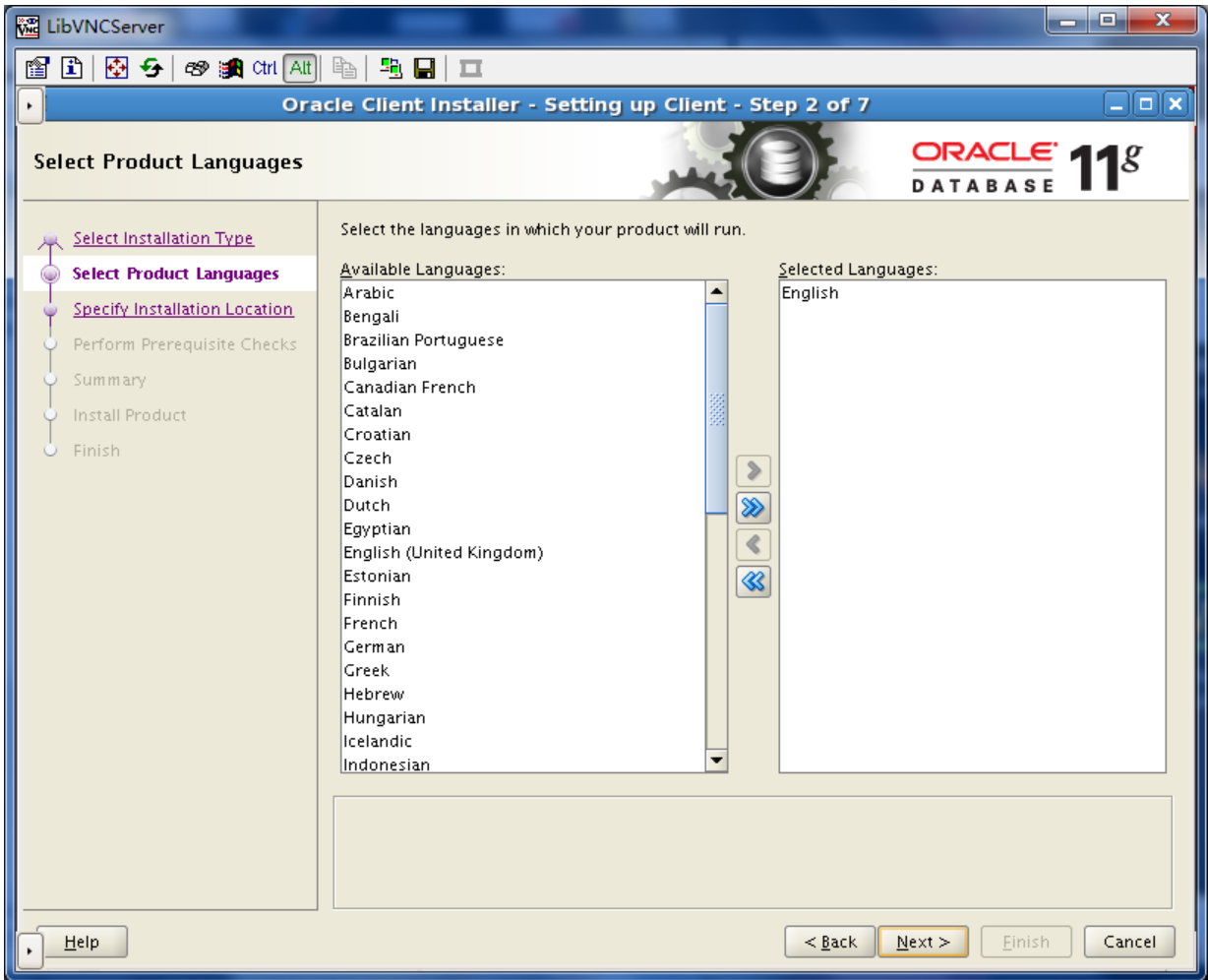
```
export TMP=/tmp
export TMPDIR=/tmp
export ORACLE_BASE=/opt/oracle
export ORACLE_HOME=$ORACLE_BASE/product/11.2.0.1/client
export PATH=$ORACLE_HOME/bin:$PATH
```

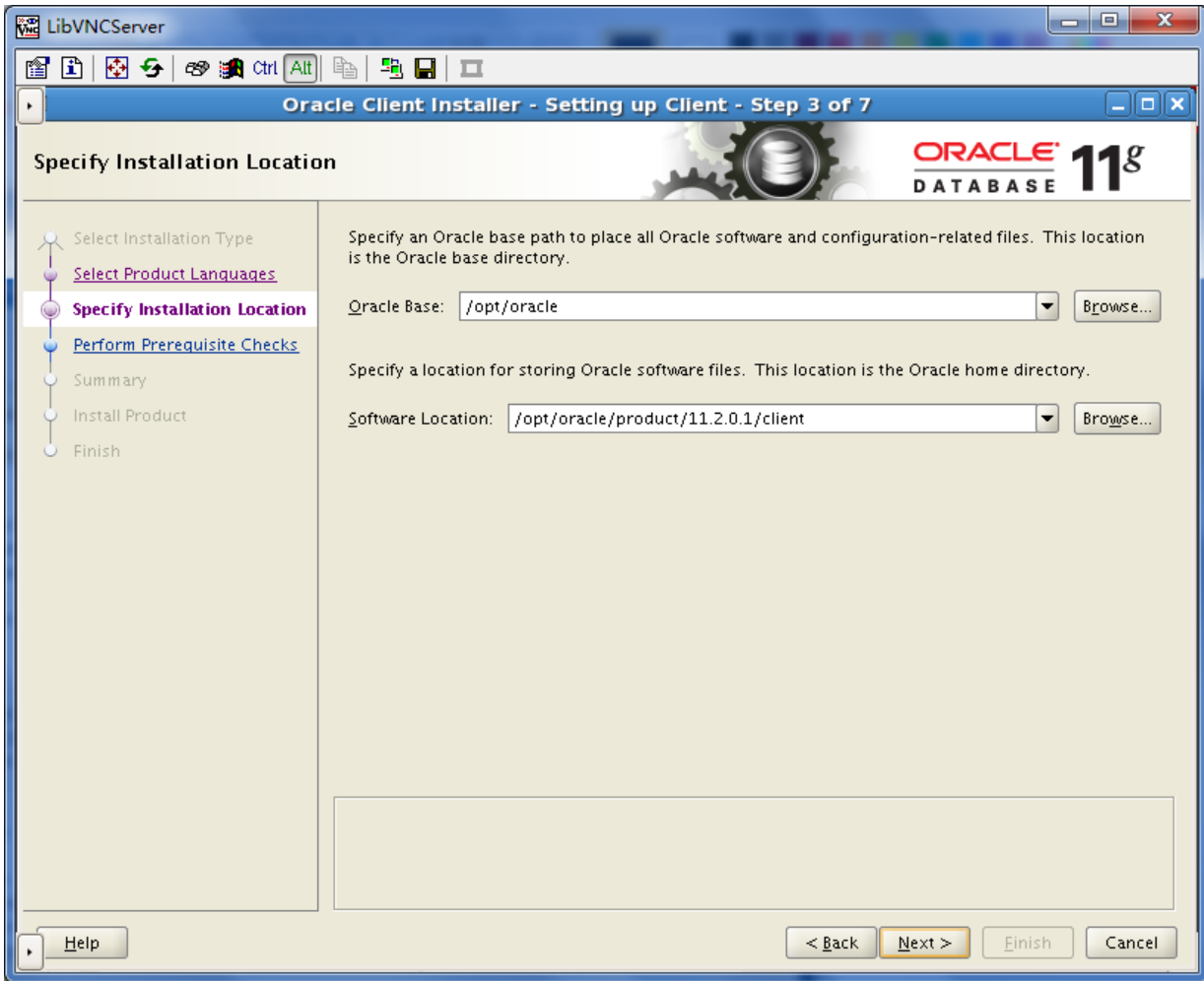
```
yum install sysstat libaio-devel glibc-devel elfutils-libelf-
devel unixODBC unixODBC-devel
```

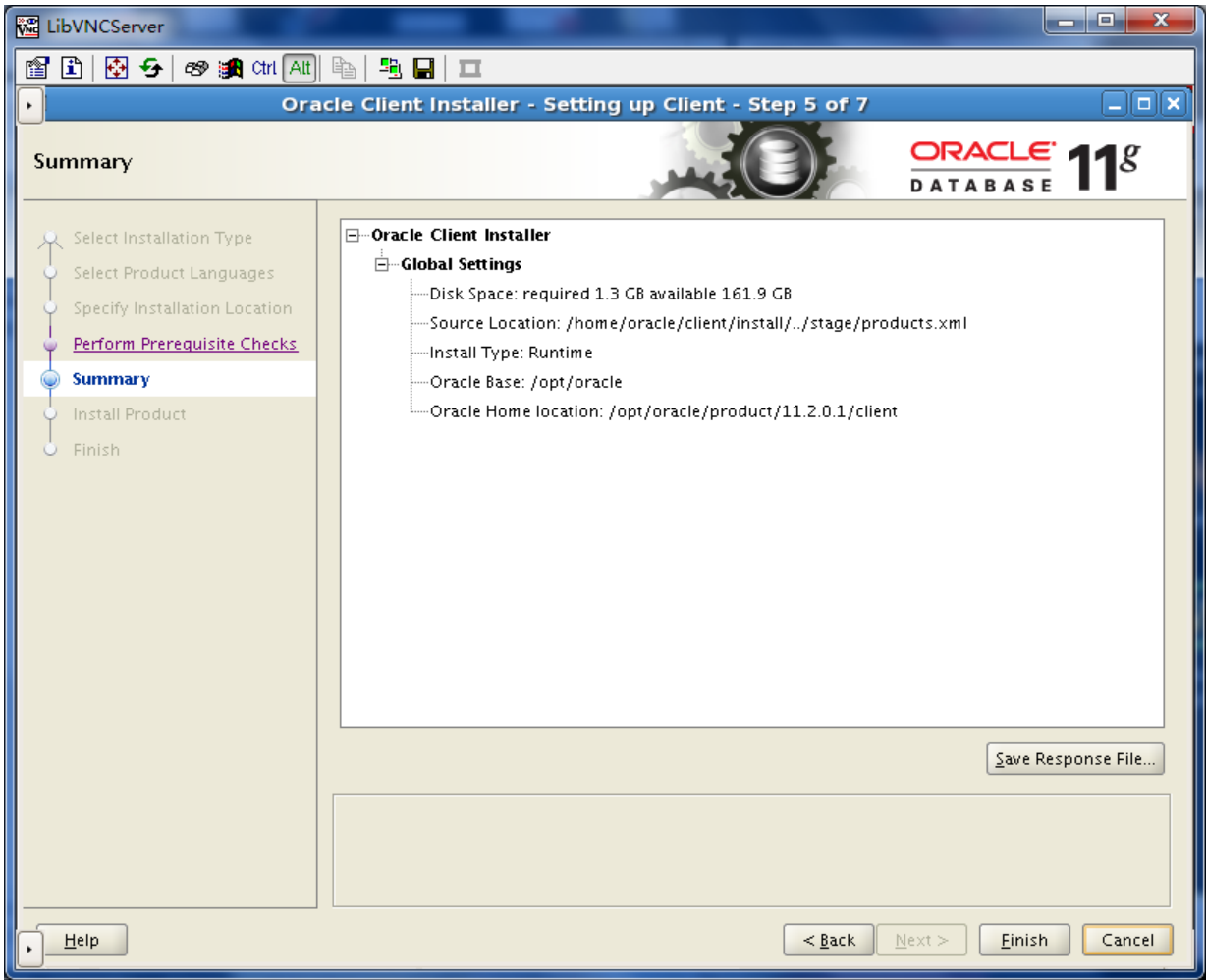
```
unzip linux.x64_11gR2_client.zip
```

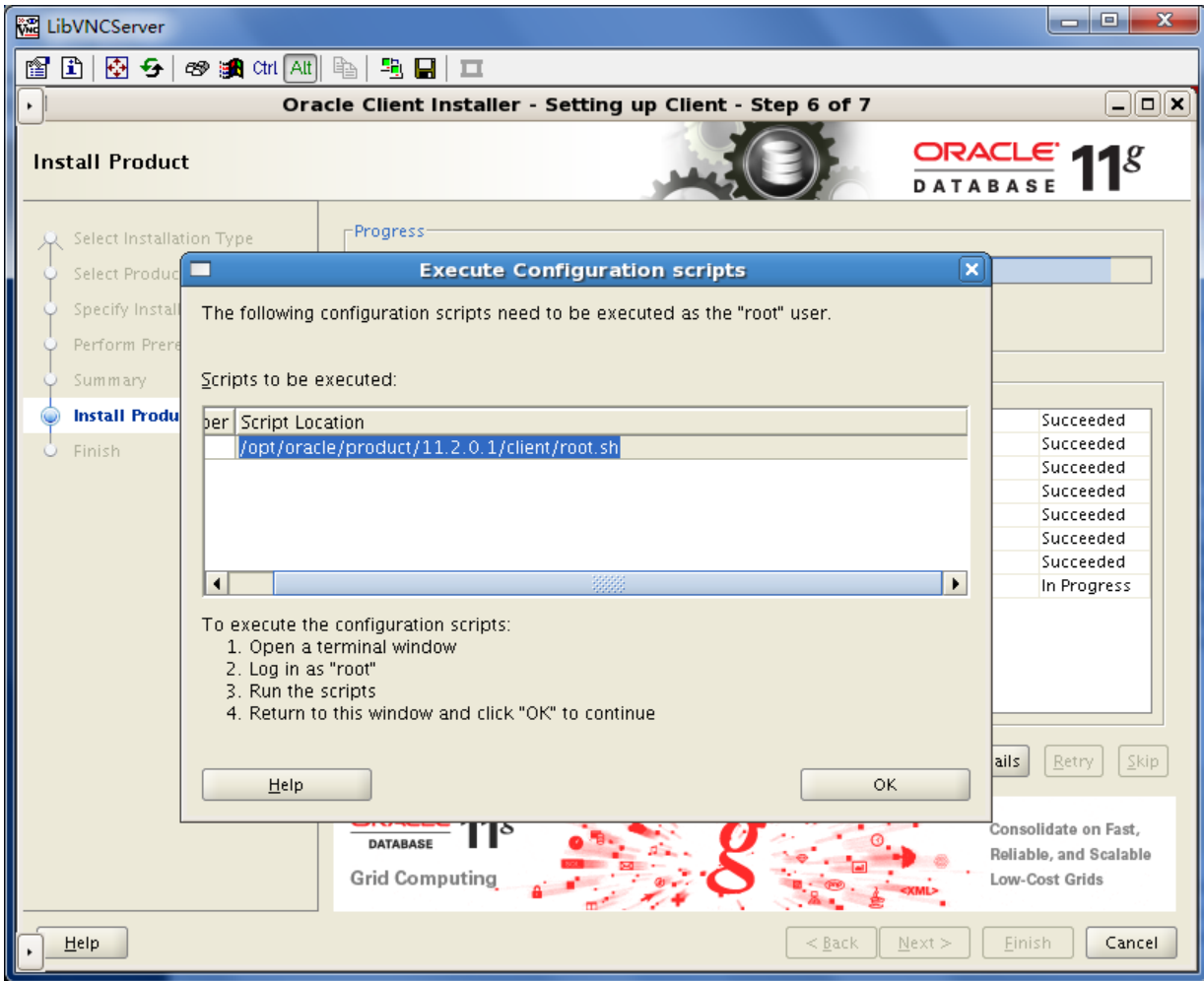
```
[oracle@wcs ~]$ cd client/
[oracle@wcs client]$ ./runInstaller
```

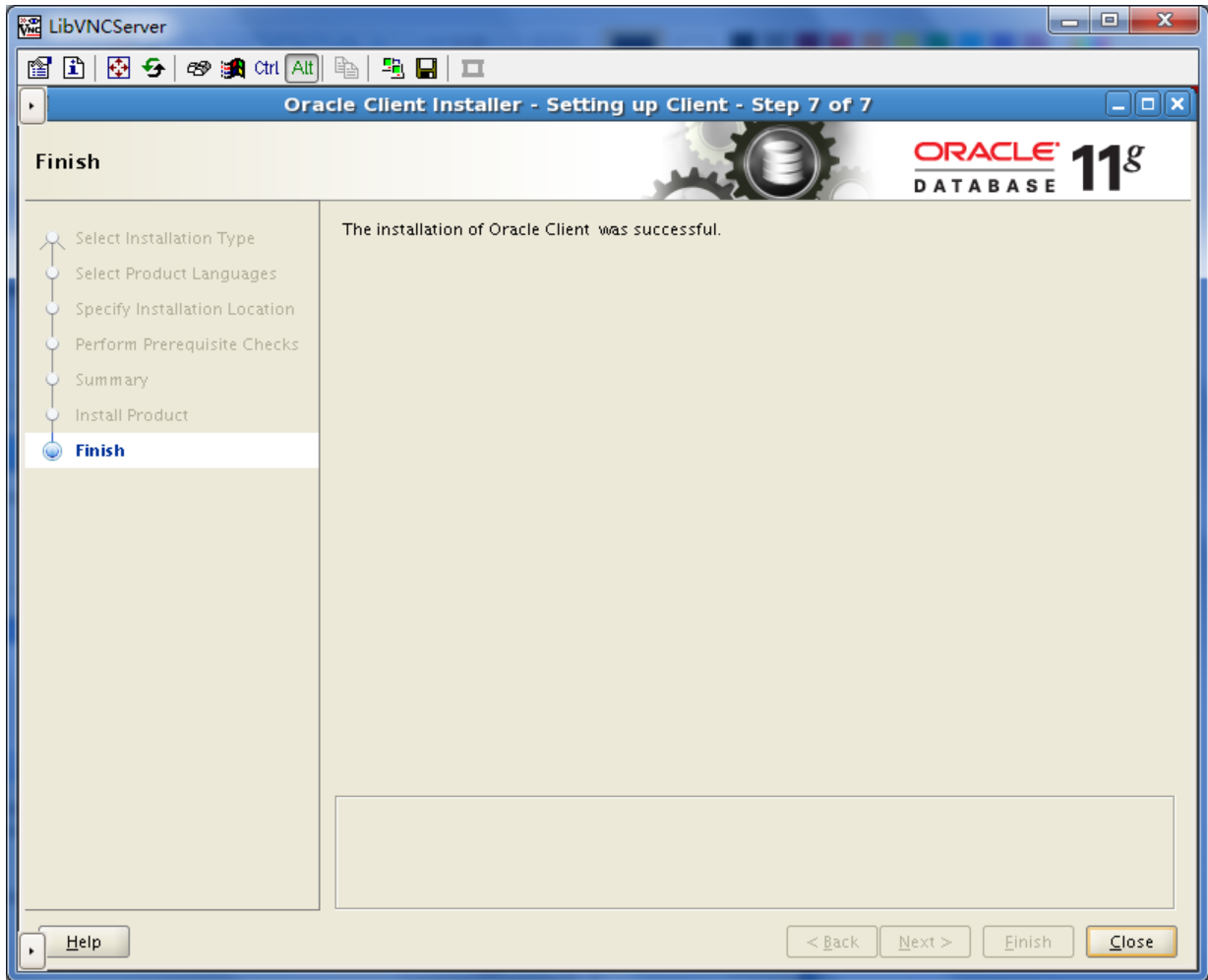












登录到root用户运行下面脚本

```
/opt/oracle/oraInventory/orainstRoot.sh  
/opt/oracle/product/11.2.0.1/client/root.sh
```

```
[root@wcs oracle]# /opt/oracle/oraInventory/orainstRoot.sh  
Changing permissions of /opt/oracle/oraInventory.  
Adding read,write permissions for group.  
Removing read,write,execute permissions for world.  
  
Changing groupname of /opt/oracle/oraInventory to oinstall.  
The execution of the script is complete.  
[root@wcs oracle]# /opt/oracle/product/11.2.0.1/client/root.sh  
Running Oracle 11g root.sh script...
```

```
The following environment variables are set as:
  ORACLE_OWNER= oracle
  ORACLE_HOME=  /opt/oracle/product/11.2.0.1/client

Enter the full pathname of the local bin directory:
[/usr/local/bin]:
  Copying dbhome to /usr/local/bin ...
  Copying oraenv to /usr/local/bin ...
  Copying coraenv to /usr/local/bin ...

Creating /etc/oratab file...
Entries will be added to the /etc/oratab file as needed by
Database Configuration Assistant when a database is created
Finished running generic part of root.sh script.
Now product-specific root actions will be performed.
```

至此，Oracle客户端安装完毕

4. Silence Install - Database

4.1. Response File

创建response文件 /home/oracle/db.rsp

```
[oracle@oracle ~]$ cat db.rsp
#-----
# Do not change the following system generated value.
#-----
oracle.install.responseFileVersion=/oracle/install/rspfmt_dbinst
all_response_schema_v11_2_0
#-----
# The installation option can be one of the following
# 1. INSTALL_DB_SWONLY
# 2. INSTALL_DB_AND_CONFIG
# 3. UPGRADE_DB
#-----
oracle.install.option=INSTALL_DB_AND_CONFIG
#-----
# This variable holds the hostname of the system as set by the
user.
# It can be used to force the installation to use an
alternative
# hostname rather than using the first hostname found on the
system
# (e.g., for systems with multiple hostnames and network
interfaces).
#-----
ORACLE_HOSTNAME=oracle.example.com
```

```
#-----  
#-----  
# Unix group to be set for the inventory directory.  
#-----  
#-----  
UNIX_GROUP_NAME=oinstall  
#-----  
#-----  
# Inventory location.  
#-----  
#-----  
INVENTORY_LOCATION=/opt/oracle/oraInventory  
#-----  
#-----  
# Specify the languages in which the components will be  
installed.  
#  
# en      : English          ja      : Japanese  
# fr      : French          ko      : Korean  
# ar      : Arabic          es      : Latin American Spanish  
# bn      : Bengali         lv      : Latvian  
# pt_BR   : Brazilian Portuguese lt      : Lithuanian  
# bg      : Bulgarian       ms      : Malay  
# fr_CA   : Canadian French es_MX   : Mexican Spanish  
# ca      : Catalan         no      : Norwegian  
# hr      : Croatian        pl      : Polish  
# cs      : Czech           pt      : Portuguese  
# da      : Danish          ro      : Romanian  
# nl      : Dutch           ru      : Russian  
# ar_EG   : Egyptian        zh_CN   : Simplified Chinese  
# en_GB   : English (Great Britain) sk      : Slovak  
# et      : Estonian        sl      : Slovenian  
# fi      : Finnish         es_ES   : Spanish  
# de      : German          sv      : Swedish  
# el      : Greek           th      : Thai  
# iw      : Hebrew          zh_TW   : Traditional Chinese  
# hu      : Hungarian       tr      : Turkish  
# is      : Icelandic        uk      : Ukrainian  
# in      : Indonesian      vi      : Vietnamese  
# it      : Italian  
#  
# Example : SELECTED_LANGUAGES=en,fr,ja  
#-----
```



```
-----  
SELECTED_LANGUAGES=en
```

```
#-----
```

```
-----  
# Complete path of the Oracle Home
```

```
#-----
```

```
-----  
ORACLE_HOME=/opt/oracle/product/11.2.0/dbhome_1
```

```
#-----
```

```
-----  
# Complete path of the Oracle Base.
```

```
#-----
```

```
-----  
ORACLE_BASE=/opt/oracle
```

```
#-----
```

```
-----  
# Installation Edition of the component.
```

```
#
```

```
# The value should contain only one of these choices.
```

```
# EE      : Enterprise Edition
```

```
# SE      : Standard Edition
```

```
# SEONE   : Standard Edition One
```

```
# PE      : Personal Edition (WINDOWS ONLY)
```

```
#-----
```

```
-----  
oracle.install.db.InstallEdition=EE
```

```
#-----
```

```
-----  
# This property is considered only if InstallEdition is EE.
```

```
#
```

```
# true  : Components mentioned as part of 'customComponents'  
property
```

```
#         are considered for install.
```

```
# false : Value for 'customComponents' is not considered.
```

```
#-----
```

```
-----  
oracle.install.db.isCustomInstall=false
```

```
#-----
```

```
-----  
# This property is considered only if 'IsCustomInstall' is set
```

```

to true
#
# Description: List of Enterprise Edition Options you would
like to install.
#
#           The following choices are available. You may
specify any
#           combination of these choices.  The components
you choose should
#           be specified in the form "internal-component-
name:version"
#           Below is a list of components you may specify to
install.
#
#           oracle.oraolap:11.2.0.0.2 - Oracle OLAP
#           oracle.rdbms.dm:11.2.0.0.2 - Oracle Data Mining
RDBMS Files
#           oracle.rdbms.dv:11.2.0.0.2 - Oracle Database
Vault option
#           oracle.rdbms.lbac:11.2.0.0.2 - Oracle Label
Security
#           oracle.rdbms.partitioning:11.2.0.0.2 - Oracle
Partitioning
#           oracle.rdbms.rat:11.2.0.0.2 - Oracle Real
Application Testing
#           oracle.clrintg.ode_net:11.2.0.0.2 - Oracle
Database Extensions for .NET 1.x (Windows)
#           oracle.clrintg.ode_net_2:11.2.0.0.2 - Oracle
Database Extensions for .NET 2.0 (Windows)
#-----
-----
oracle.install.db.customComponents=
#-----
-----
oracle.install.db.DBA_GROUP=dba
#-----
-----
oracle.install.db.OPER_GROUP=dba
#-----
-----
# This variable represents the cluster node names selected by
the .

```

```
# user for installation
#-----
-----
oracle.install.db.CLUSTER_NODES=
#-----
-----
# One of the following
# - GENERAL_PURPOSE
# - TRANSACTION_PROCESSING
# - DATAWAREHOUSE
#-----
-----
oracle.install.db.config.starterdb.type=GENERAL_PURPOSE
#-----
-----
# Global Database Name
#-----
-----
oracle.install.db.config.starterdb.globalDBName=orcl.example.co
m
#-----
-----
# The Starter Database SID
#-----
-----
oracle.install.db.config.starterdb.SID=orcl
#-----
-----
# Database character set
#
# One of the following
# AL32UTF8, WE8ISO8859P15, WE8MSWIN1252, EE8ISO8859P2,
# EE8MSWIN1250, NE8ISO8859P10, NEE8ISO8859P4, BLT8MSWIN1257,
# BLT8ISO8859P13, CL8ISO8859P5, CL8MSWIN1251, AR8ISO8859P6,
# AR8MSWIN1256, EL8ISO8859P7, EL8MSWIN1253, IW8ISO8859P8,
# IW8MSWIN1255, JA16EUC, JA16EUCTILDE, JA16SJIS,
JA16SJISTILDE,
# KO16MSWIN949, ZHS16GBK, TH8TISASCII, ZHT32EUC,
ZHT16MSWIN950,
# ZHT16HKSCS, WE8ISO8859P9, TR8MSWIN1254, VN8MSWIN1258
#-----
```

```
-----
oracle.install.db.config.starterdb.characterSet=AL32UTF8
#-----
-----
# Specify the total memory allocation for the database. (in MB)
# Value should be at least 256 MB, and should not exceed the
# total physical memory available on the system.
# Example: oracle.install.db.config.starterdb.memoryLimit=40
#-----
-----
oracle.install.db.config.starterdb.memoryLimit=6218
oracle.install.db.config.starterdb.memoryOption=true
#-----
-----
# This variable controls whether to load Example Schemas onto
# the starter database or not.
#-----
-----
oracle.install.db.config.starterdb.installExampleSchemas=true
#-----
-----
# This include enabling audit settings, configuring password
# profiles and revoking some grants to public. These settings
# are provided by default. You may choose to disable all.
#-----
-----
oracle.install.db.config.starterdb.enableSecuritySettings=true
#-----
-----
oracle.install.db.config.starterdb.password.ALL=
#-----
-----
oracle.install.db.config.starterdb.password.SYS=
#-----
-----
oracle.install.db.config.starterdb.password.SYSTEM=
#-----
-----
oracle.install.db.config.starterdb.password.SYSMAN=
```

```
#-----  
-----  
oracle.install.db.config.starterdb.password.DBSNMP=  
#-----  
-----  
# Can be one of the following  
# 1. GRID_CONTROL  
# 2. DB_CONTROL  
#  
oracle.install.db.config.starterdb.control=DB_CONTROL  
#-----  
-----  
# Determines the Management Service to use if Grid Control  
# is selected to manage the database.  
#-----  
-----  
oracle.install.db.config.starterdb.gridcontrol.gridControlServiceURL=  
#-----  
-----  
# Determines whether to receive email notification for  
# critical alerts when using DB control.  
#-----  
-----  
oracle.install.db.config.starterdb.dbcontrol.enableEmailNotification=false  
#-----  
-----  
oracle.install.db.config.starterdb.dbcontrol.emailAddress=  
#-----  
-----  
oracle.install.db.config.starterdb.dbcontrol.SMTPServer=  
#-----  
-----  
oracle.install.db.config.starterdb.automatedBackup.enable=false  
#-----  
-----
```

```
oracle.install.db.config.starterdb.automatedBackup.osuid=
#-----
oracle.install.db.config.starterdb.automatedBackup.ospwd=
#-----
# Can be one of the following
# - FILE_SYSTEM_STORAGE
# - ASM_STORAGE
#-----
oracle.install.db.config.starterdb.storageType=FILE_SYSTEM_STORAGE
#-----
# Database file location:
# directory for datafiles, control files, redo logs.
#
# Applicable only when
oracle.install.db.config.starterdb.storage=FILE_SYSTEM_STORAGE
#-----
oracle.install.db.config.starterdb.fileSystemStorage.dataLocation=/opt/oracle/oradata
#-----
# Backup and recovery location
#
# Applicable only when
oracle.install.db.config.starterdb.storage=FILE_SYSTEM_STORAGE
#-----
oracle.install.db.config.starterdb.fileSystemStorage.recoveryLocation=
#-----
# Name of ASM disk group to be used for storage.
#
# Applicable only when
oracle.install.db.config.starterdb.storageType=ASM_STORAGE
```

```
#-----  
-----  
oracle.install.db.config.asm.diskGroup=  
  
#-----  
-----  
# Password for ASMSNMP user of the ASM instance.  
#  
# Applicable only when  
oracle.install.db.config.starterdb.storage=ASM_STORAGE  
#-----  
-----  
oracle.install.db.config.asm.ASMSNMPPassword=  
  
#-----  
-----  
# Specify the My Oracle Support Account Username.  
#  
# Example : MYORACLESUPPORT_USERNAME=metalink  
#-----  
-----  
MYORACLESUPPORT_USERNAME=neo.chen@msn.com  
  
#-----  
-----  
# Specify the My Oracle Support Account Username password.  
#  
# Example : MYORACLESUPPORT_PASSWORD=password  
#-----  
-----  
MYORACLESUPPORT_PASSWORD=  
  
#-----  
-----  
# Specify whether to enable the user to set the password for  
# My Oracle Support credentials. The value can be either true  
or false.  
# If left blank it will be assumed to be false.  
#  
# Example : SECURITY_UPDATES_VIA_MYORACLESUPPORT=true  
#-----  
-----  
SECURITY_UPDATES_VIA_MYORACLESUPPORT=false  
#-----
```

```

-----
# Specify whether user wants to give any proxy details for
connection.
# The value can be either true or false. If left blank it will
be assumed
# to be false.
#
# Example      : DECLINE_SECURITY_UPDATES=false
#-----
DECLINE_SECURITY_UPDATES=false
#-----
# Specify the Proxy server name. Length should be greater than
zero.
#
# Example      : PROXY_HOST=proxy.domain.com
#-----
PROXY_HOST=
#-----
# Specify the proxy port number. Should be Numeric and at least
2 chars.
#
# Example      : PROXY_PORT=25
#-----
PROXY_PORT=

```

4.2. OS 配置脚本

/home/oracle/orains.sh

```

#!/bin/bash
ORACLE_BASE=/opt/oracle
ORACLE_PASSWORD="oracle"

```



```
groupadd oinstall
groupadd dba
useradd -m -g oinstall -G dba oracle
echo "oracle:$ORACLE_PASSWORD" | chpasswd

mkdir -p $ORACLE_BASE
chown -R oracle:oinstall $ORACLE_BASE
chmod -R 775 $ORACLE_BASE

cat >> /etc/sysctl.conf <<EOF
fs.aio-max-nr = 3145728
fs.file-max = 6815744
kernel.shmall = 1073741824
kernel.shmmax = 4398046511104
kernel.shmmni = 4096
kernel.sem = 250 32000 100 142
net.ipv4.ip_local_port_range = 9000 65500
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 1048576
EOF

cat >> /etc/security/limits.conf <<EOF
oracle soft nproc 2048
oracle hard nproc 16384
oracle soft nofile 1024
oracle hard nofile 65536
EOF

cat >> /home/oracle/.bash_profile <<\EOF
export TMP=/tmp
export TMPDIR=$TMP
export ORACLE_HOSTNAME=$(hostname)
export ORACLE_BASE=/opt/oracle
export ORACLE_HOME=$ORACLE_BASE/product/11.2.0/dbhome_1
export ORACLE_SID=orcl
export ORACLE_TERM=xterm
export PATH=$ORACLE_HOME/bin:$PATH
export
LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib64:/usr/lib64:/usr/local/l
ib64
export
CLASSPATH=$ORACLE_HOME/JRE:$ORACLE_HOME/jlib:$ORACLE_HOME/rdbms
/jlib
```

```
export LD_ASSUME_KERNEL=2.6.18
export NLS_LANG="american_america.UTF8"
export NLS_LANG="AMERICAN_AMERICA.US7ASCII"
#export NLS_LANG="AMERICAN_AMERICA.ZHS16GBK"
#export NLS_LANG="SIMPLIFIED CHINESE_CHINA.ZHS16GBK"
#export NLS_LANG="TRADITIONAL CHINESE_TAIWAN.ZHT16MSWIN950"
#export NLS_LANG="JAPANESE_JAPAN.WE8MSWIN1252"
EOF

cat >> /home/oracle/.bashrc <<\EOF
alias sysdba='sqlplus "/ as sysdba"'
EOF

cat >> /etc/oraInst.loc <<EOF
inventory_loc=$ORACLE_BASE/oraInventory
inst_group=oinstall
EOF
chmod 664 /etc/oraInst.loc

cat >> /etc/yum.repos.d/rhel-source-local.repo <<\EOF
[rhel-source-local]
name=Red Hat Enterprise Linux $releasever - Source
baseurl=file:///media/cdrom0/Server
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
EOF

yum install gcc gcc-c++ glibc-devel libstdc++ libstdc++-devel
libaio-devel sysstat libaio-devel elfutils-libelf-devel
unixODBC unixODBC-devel
```

```
su - root
/home/oracle/orains.sh
```

4.3. 运行 runInstaller

```
unzip linux.x64_11gR2_database_1of2.zip
```

```
unzip linux.x64_11gR2_database_2of2.zip
cd database
./runInstaller -silent -noconfig -responseFile
/home/oracle/db.rsp
```

检查安装日志

```
ls /opt/oracle/oraInventory/logs/
```

5. Silence Install - Client

```
#####  
#####  
## Copyright(c) Oracle Corporation 1998,2008. All rights  
reserved.          ##  
##  
##  
## Specify values for the variables listed below to customize  
##  
## your installation.  
##  
##  
## Each variable is associated with a comment. The comment  
##  
## can help to populate the variables with the appropriate  
##  
## values.  
##  
##  
##  
#####  
#####  
  
#-----  
-----  
# Do not change the following system generated value.  
#-----  
-----  
oracle.install.responseFileVersion=http://www.oracle.com/2007/i  
ninstall/rspfmt_clientinstall_response_schema_v11_2_0  
  
#-----  
-----  
# This variable holds the hostname of the system as set by the  
user.  
# It can be used to force the installation to use an  
alternative  
# hostname rather than using the first hostname found on the  
system
```

```

# (e.g., for systems with multiple hostnames and network
interfaces).
ORACLE_HOSTNAME=wcs.example.com
#-----
# Unix group to be set for the inventory directory.
UNIX_GROUP_NAME=oinstall
#-----
# Inventory location.
INVENTORY_LOCATION=/opt/oracle/oraInventory
#-----
# Specify the languages in which the components will be
installed.
#
# en      : English          ja      : Japanese
# fr      : French          ko      : Korean
# ar      : Arabic          es      : Latin American Spanish
# bn      : Bengali         lv      : Latvian
# pt_BR   : Brazilian Portuguese lt      : Lithuanian
# bg      : Bulgarian       ms      : Malay
# fr_CA   : Canadian French es_MX   : Mexican Spanish
# ca      : Catalan         no      : Norwegian
# hr      : Croatian        pl      : Polish
# cs      : Czech           pt      : Portuguese
# da      : Danish          ro      : Romanian
# nl      : Dutch           ru      : Russian
# ar_EG   : Egyptian        zh_CN   : Simplified Chinese
# en_GB   : English (Great Britain) sk      : Slovak
# et      : Estonian        sl      : Slovenian
# fi      : Finnish         es_ES   : Spanish
# de      : German          sv      : Swedish
# el      : Greek           th      : Thai
# iw      : Hebrew          zh_TW   : Traditional Chinese
# hu      : Hungarian       tr      : Turkish
# is      : Icelandic        uk      : Ukrainian
# in      : Indonesian       vi      : Vietnamese
# it      : Italian
#
# Example : SELECTED_LANGUAGES=en,fr,ja
SELECTED_LANGUAGES=en
#-----
# Complete path of the Oracle Home

```

```

ORACLE_HOME=/opt/oracle/product/11.2.0.1/client
#-----
-----
# Complete path of the Oracle Base.
ORACLE_BASE=/opt/oracle
#-----
-----
#Name          : INSTALL_TYPE
#Datatype     : String
#Description: Installation type of the component.
#
#             The following choices are available. The value
should contain
#             only one of these choices.
#             InstantClient : InstantClient
#             Administrator  : Administrator
#             Runtime        : Runtime
#             Custom         : Custom
#
#Example       : INSTALL_TYPE = "Administrator"
#-----
-----
oracle.install.client.installType=Administrator
#-----
-----
# Name          : oracle.install.client.customComponents
# Datatype     : StringList
#
# This property is considered only if INSTALL_TYPE is set to
"Custom"
#
# Description: List of Client Components you would like to
install
#
# The following choices are available. You may specify any
# combination of these choices. The components you choose
should
# be specified in the form "internal-component-name:version"
# Below is a list of components you may specify to install.
#
# oracle.sqlj:11.2.0.1.0 -- "Oracle SQLJ"
# oracle.rdbms.util:11.2.0.1.0 -- "Oracle Database Utilities"
# oracle.javavm.client:11.2.0.1.0 -- "Oracle Java Client"
# oracle.sqlplus:11.2.0.1.0 -- "SQL*Plus"
# oracle.dbjava.jdbc:11.2.0.1.0 -- "Oracle JDBC/THIN"

```

```

Interfaces"
# oracle.ldap.client:11.2.0.1.0 -- "Oracle Internet Directory
Client"
# oracle.rdbms.oci:11.2.0.1.0 -- "Oracle Call Interface (OCI)"
# oracle.precomp:11.2.0.1.0 -- "Oracle Programmer"
# oracle.xdk:11.2.0.1.0 -- "Oracle XML Development Kit"
# oracle.network.aso:11.2.0.1.0 -- "Oracle Advanced Security"
# oracle.assistants.oemlt:11.2.0.1.0 -- "Enterprise Manager
Minimal Integration"
# oracle.oraolap.mgmt:11.2.0.1.0 -- "OLAP Analytic Workspace
Manager and Worksheet"
# oracle.network.client:11.2.0.1.0 -- "Oracle Net"
# oracle.ordim.client:11.2.0.1.0 -- "Oracle Multimedia Client
Option"
# oracle.ons:11.2.0.0.0 -- "Oracle Notification Service"
# oracle.odbc:11.2.0.1.0 -- "Oracle ODBC Driver"
# oracle.has.client:11.2.0.1.0 -- "Oracle Clusterware High
Availability API"
# oracle.dbdev:11.2.0.1.0 -- "Oracle SQL Developer"
# oracle.rdbms.scheduler:11.2.0.1.0 -- "Oracle Scheduler Agent"
#
# Example      :
oracle.install.client.customComponents="oracle.precomp:11.2.0.1
.0","oracle.ons:11.2.0.0.0","oracle.oraolap.mgmt:11.2.0.1.0","o
racle.rdbms.scheduler:11.2.0.1.0"
#-----
-----
oracle.install.client.customComponents=
#-----
-----
#Name          : MTS_PORT
#Datatype     : int
#Description: Port number to be used for by the Oracle MTS
Recovery Service to listen
#              for requests. This needs to be entered in
case oracle.ntonamts is
#              selected in the list of custom components in
custom install
#
#
#Example      : MTS_PORT = 2030
#-----
-----
oracle.install.client.oramtsPortNumber=

```

```
#-----  
-----  
# Host name to be used for by the Oracle Scheduler Agent.  
# This needs to be entered in case oracle.rdbms.scheduler is  
selected in the  
# list of custom components during custom install  
#  
# Example      : oracle.install.client.schedulerAgentHostName =  
acme.domain.com  
#-----  
-----  
oracle.install.client.schedulerAgentHostName=  
  
#-----  
-----  
# Port number to be used for by the Oracle Scheduler Agent.  
# This needs to be entered in case oracle.rdbms.scheduler is  
selected in the  
# list of custom components during custom install  
#  
# Example: oracle.install.client.schedulerAgentPortNumber =  
1500  
#-----  
-----  
oracle.install.client.schedulerAgentPortNumber=
```

```
#!/bin/bash  
groupadd oinstall  
groupadd dba  
useradd -m -g oinstall -G dba oracle  
echo "oracle:oracle" | chpasswd  
id oracle  
  
mkdir -p /opt/oracle  
chown oracle.oinstall /opt/oracle  
  
cat >> /etc/sysctl.conf <<EOF  
kernel.shmall = 2097152  
kernel.shmmax = 536870912  
kernel.shmmni = 4096  
kernel.sem = 250 32000 100 128
```



```
fs.file-max = 65536
net.ipv4.ip_local_port_range = 32768 61000
net.core.rmem_default=262144
net.core.wmem_default=262144
net.core.rmem_max=4194304
net.core.wmem_max=262144
EOF
/sbin/sysctl -p

cat >> /home/oracle/.bash_profile <<\EOF
export TMP=/tmp
export TMPDIR=/tmp
export ORACLE_BASE=/opt/oracle
export ORACLE_HOME=$ORACLE_BASE/product/11.2.0.1/client
export PATH=$ORACLE_HOME/bin:$PATH
EOF

cat >> /home/oracle/.bashrc <<\EOF
alias sysdba='sqlplus "/ as sysdba"'
EOF
```

```
yum install sysstat libaio-devel glibc-devel elfutils-libelf-
devel unixODBC unixODBC-devel

unzip linux.x64_11gR2_client.zip

cd client/
./runInstaller -silent -noconfig -responseFile
/home/oracle/client.rsp
```

```
# /opt/oracle/oraInventory/orainstRoot.sh
# /opt/oracle/product/11.2.0.1/client/root.sh
```

6. oracle 817 script

Oracle 817 数据库启动脚本

```
#!/bin/bash
#####
# Script to startup and shutdown Oracle and listener
# File:/etc/rc.d/init.d/orcl
#####
# Setup environment for script execution
#./home/oracle/.bash_profile
#
ORACLE_HOME=/u01/app/oracle/product/8.1.7
# Determine and execute action based on command line parameter
case "$1" in
start)
echo "Starting Oracle database(s) listed in /etc/oratab ..."
sleep 2
su - oracle -c "$ORACLE_HOME/bin/dbstart"
echo "Starting TNS listener ..."
sleep 2
su - oracle -c "$ORACLE_HOME/bin/lsnrctl start"
touch /var/lock/subsys/orcl
;;
stop)
echo "Shutting down TNS listener ..."
sleep 2
su - oracle -c "$ORACLE_HOME/bin/lsnrctl stop"
echo "Shutting down Oracle database(s) listed in /etc/oratab
..."
su - oracle -c "$ORACLE_HOME/bin/dbshut"
rm -f /var/lock/subsys/orcl
;;
status)
ps -ax | grep -e ora_ -e tnslnr
;;
*)
echo "Usage: $1 {start|stop|status}"
;;
esac
```

```
exit 0
```

7. Script for automatic startup on boot

```
#!/bin/bash
#
# Run-level Startup script for the Oracle Instance and Listener
#
# chkconfig: 345 91 19
# description: Startup/Shutdown Oracle listener and instance

ORA_HOME="/u01/app/oracle/product/9.2.0.1.0"
ORA_OWNER="oracle"

# if the executables do not exist -- display error

if [ ! -f $ORA_HOME/bin/dbstart -o ! -d $ORA_HOME ]
then
    echo "Oracle startup: cannot start"
    exit 1
fi

# depending on parameter -- startup, shutdown, restart
# of the instance and listener or usage display

case "$1" in
    start)
        # Oracle listener and instance startup
        echo -n "Starting Oracle: "
        su - $ORA_OWNER -c "$ORA_HOME/bin/lsnrctl start"
        su - $ORA_OWNER -c $ORA_HOME/bin/dbstart
        touch /var/lock/subsys/oracle
        echo "OK"
        ;;
    stop)
        # Oracle listener and instance shutdown
        echo -n "Shutdown Oracle: "
        su - $ORA_OWNER -c "$ORA_HOME/bin/lsnrctl stop"
        su - $ORA_OWNER -c $ORA_HOME/bin/dbshut
        rm -f /var/lock/subsys/oracle
        echo "OK"
        ;;
    reload|restart)
```

```
    $0 stop
    $0 start
    ;;
*)
    echo "Usage: $0 start|stop|restart|reload"
    exit 1
esac
exit 0
```

8. Run level shell script to start Oracle 10g services on RedHat Enterprise Linux (RHAS 4)

```
#!/bin/bash
#####
# Script to startup and shutdown Oracle and listener
# Author: neo - http://netkiller.8800.org
# File:/etc/rc.d/init.d/oracle
# chmod 750 /etc/init.d/oracle
# chkconfig --add oracle --level 0356
#####
# Setup environment for script execution
export ORACLE_BASE=/u01/app/oracle
export ORACLE_HOME=/u01/app/oracle/product/10.2.0.1/
export
PATH=$ORACLE_HOME/bin:$ORACLE_HOME/Oracle/Oracle/bin:$PATH
export NLS_LANG='croatian_croatia.ee8iso8859p2'
export ORACLE_SID=orcl
export DISPLAY=:0
export USER=oracle
if [ -f ./home/oracle/.bash_profile ]; then
    ./home/oracle/.bash_profile
fi

# Determine and execute action based on command line parameter

# check Oracle db status
function chkdb_status() {

    # set username
    SUSER="scott"
    # set password
    SPASS="123456"

    sqlplus -s /nolog > /dev/null 2>&1 <<EOF
whenever sqlerror exit failure
connect $SUSER/$SPASS
exit success
EOF
```

```

        if [ $? -ne 0 ]; then
            echo "Connection failed : DB is down"
            exit 1
        else
            echo "Connection succeeded : DB is up"
        fi
    }

function isql {
    case "$1" in
        start)
            echo "*** Starting Oracle iSQL Plus
*** "
            su - $USER -c
"$ORACLE_HOME/bin/isqlplusctl start"
            echo "*** Note: You can access service
at url: http://$(hostname):5560/isqlplus"
            ;;
        stop)
            echo "*** Stopping Oracle iSQL Plus
*** "
            su - $USER -c
"$ORACLE_HOME/bin/isqlplusctl stop"
            ;;
        *)
            echo "Usage: $1 isql {start|stop}"
            ;;
    esac
}

function sqlplus {
    case "$1" in
        start)
            su - "$oracle_user"<<EEO
            lsnrctl start
            apachectl start
            sqlplus /nolog<<EOS
            connect / as sysdba
            startup
EOS
EEO
            ;;
    esac
}

```

```

                stop)
su - "$oracle_user"<<E00
    lsnrctl stop
    apachectl stop
    sqlplus /nolog<<EOS
        connect / as sysdba
        shutdown immediate
EOS
E00

                ;;
            *)
                echo "Usage: $1 emctl {start|stop}"
                ;;
        esac
}
function emctl {
    case "$1" in
        start)
            echo "*** Starting Oracle Enterprise
Manager 10g Database Control ***"
            su - $USER -c "$ORACLE_HOME/bin/emctl
start dbconsole"
            echo "*** Note: You can access service
at url: http://$(hostname):1158/em"
            ;;
        stop)
            echo "*** Stopping Oracle Enterprise
Manager 10g Database Control ***"
            su - $USER -c "$ORACLE_HOME/bin/emctl
stop dbconsole"
            ;;
        *)
            echo "Usage: $1 emctl {start|stop}"
            ;;
    esac
}
case "$1" in
    start)
        echo "Starting Oracle database(s) listed in
/etc/oratab ..."
        sleep 2
        su - $USER -c "$ORACLE_HOME/bin/dbstart"
        echo "Starting TNS listener ..."
        sleep 2

```



```

        su - $USER -c "$ORACLE_HOME/bin/lsnrctl start"
        touch /var/lock/subsys/orcl
        ;;
    stop)
        echo "Shutting down TNS listener ..."
        sleep 2
        su - $USER -c "$ORACLE_HOME/bin/lsnrctl stop"
        echo "Shutting down Oracle database(s) listed
in /etc/oratab ..."
        su - $USER -c "$ORACLE_HOME/bin/dbshut"
        rm -f /var/lock/subsys/orcl
        ;;
    status)
        chkdb_status
        ps -ax | grep -e ora_ -e tnslnr
        ;;
    isql)
        isql $2
        ;;
    sqlplus)
        sqlplus $2
        ;;
    emctl)
        emctl $2
        ;;
    *)
        echo "Usage: $1 {start|stop|status}"
        echo
        echo "Usage: $1 [isql | sqlplus | emctl]
{start|stop}"
        ;;
esac
exit 0

```

第 83 章 FAQ

1. Reset root password 重置MySQL root密码

忘记root密码是使用 --skip-grant-tables 启动项

CentOS 6.x

```
# vim /etc/init.d/mysqld

$exec --skip-grant-tables --datadir="$datadir" --socket="$socketfile" \
  --pid-file="$mypidfile" \
  --basedir=/usr --user=mysql >/dev/null 2>&1 &
```

```
# /etc/init.d/mysqld restart
Stopping mysqld:           [ OK ]
Starting mysqld:          [ OK ]

# mysqladmin -u root flush-privileges password "newpassword"
```

1.1. MySQL 5.7.x

CentOS 7.x

添加 skip-grant-tables=1 选项，然后重启mysql

```
# cat /etc/my.cnf
# For advice on how to change settings please see
# http://dev.mysql.com/doc/refman/5.6/en/server-configuration-
defaults.html

[mysqld]
#
# Remove leading # and set to the amount of RAM for the most important
data
```

```
# cache in MySQL. Start at 70% of total RAM for dedicated server, else
10%.
# innodb_buffer_pool_size = 128M
#
# Remove leading # to turn on a very important data integrity option:
logging
# changes to the binary log between backups.
# log_bin
#
# Remove leading # to set options mainly useful for reporting servers.
# The server defaults are faster for transactions and fast SELECTs.
# Adjust sizes as needed, experiment to find the optimal values.
# join_buffer_size = 128M
# sort_buffer_size = 2M
# read_rnd_buffer_size = 2M
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
skip-grant-tables=1
# Disabling symbolic-links is recommended to prevent assorted security
risks
symbolic-links=0

# Recommended in standard MySQL setup
sql_mode=NO_ENGINE_SUBSTITUTION,STRICT_TRANS_TABLES

[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

```
# systemctl restart mysqld
```

```
update mysql.user set authentication_string=password('netkiller') where
user='root' and Host = 'localhost';
flush privileges;
quit;
```

```
# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
```

```
Server version: 5.7.14 MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> update mysql.user set authentication_string=password('netkiller')
where user='root' and Host = 'localhost';
Query OK, 1 row affected, 1 warning (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 1

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

mysql> quit;
Bye
```

删除 skip-grant-tables=1 重启MySQL

1.2. MySQL 8.0

```
[root@localhost log]# vim /etc/my.cnf

[mysqld]
skip-grant-table
```

```
ALTER USER root@localhost identified by 'MQiEgelikst7S_6tlXzB0mt';
ALTER USER root@localhost PASSWORD EXPIRE NEVER;
```

2. 查看错误代码

```
mysql> \! perror 6
OS error code 6: No such device or address
```

2.1. ERROR 1153 (08S01) at line 3168: Got a packet bigger than 'max_allowed_packet' bytes

```
max_allowed_packet=500M
```

2.2. ERROR 1129 (00000): Host 'XXXXXXX' is blocked because of many connection errors; unblock with 'mysqladmin flush-hosts'

连接在中途被中断了的连接请求。在 max_connect_errors 次失败请求后，mysql 阻止该主机进一步的连接，直到管理员执行命令 mysqladmin flush-hosts。

```
mysql> flush hosts;
```

```
set global
max_connect_errors=1000;
```

```
max_connect_errors=10000
```

3. 临时表是否需要建索引

答案：要

4. [Warning] Changed limits: max_open_files: 5000 (requested 20480)

```
2018-01-08T01:34:44.515973Z 0 [Warning] Changed limits:
max_open_files: 5000 (requested 10240)
2018-01-08T01:34:44.516402Z 0 [Warning] Changed limits:
table_open_cache: 1471 (requested 2000)
```

提出出现在 CentOS 7 ulimit 配置没有问题的情况下mysql日志提示 Warning

```
# ulimit -Sa | grep "open files"
open files (-n) 40960
```

```
[root@netkiller ~]# cat /proc/`pidof
mysqld`/limits
Limit Soft Limit Hard Limit Units
Max cpu time unlimited unlimited
seconds
Max file size unlimited unlimited bytes
Max data size unlimited unlimited bytes
Max stack size 8388608 unlimited bytes
Max core file size 0 unlimited bytes
Max resident set unlimited unlimited
bytes
Max processes 63494 63494 processes
Max open files 5000 5000 files
Max locked memory 65536 65536 bytes
Max address space unlimited unlimited
bytes
Max file locks unlimited unlimited
locks
Max pending signals 63494 63494 signals
Max msgqueue size 819200 819200 bytes
```

```
Max nice priority 0 0
Max realtime priority 0 0
Max realtime timeout unlimited
unlimited us
```

动态改变

```
[root@netkiller ~]# egrep '^(Limit|Max
open files)' /proc/`pidof mysqld`/limits
Limit Soft Limit Hard Limit Units
Max open files 5000 5000 files
```

问题的出现原因是systemctl启动脚本覆盖了ulimit配置

```
# cat
/usr/lib/systemd/system/mysqld.service | grep -A2
open_files_limit
# Sets open_files_limit
LimitNOFILE = 5000
```

解决方法，直接修改上面的数值，不建议修改mysqld.service，这样会影响你下次升级。请采用下面的方案完美解决：

```
mkdir /usr/lib/systemd/system/mysqld.service.d
cat >> /usr/lib/systemd/system/mysqld.service.d/override.conf
<<EOF
[Service]
LimitNOFILE=40960
EOF
```

重启 MySQL


```
systemctl daemon-reload
systemctl restart mysqld
```

检查是否生效

```
mysql> show variables like 'open_files_limit';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| open_files_limit | 65535 |
+-----+-----+
1 row in set (0.01 sec)
```

5. this is incompatible with sql_mode=only_full_group_by

ERROR 1055 (42000): Expression #1 of SELECT list is not in GROUP BY clause and contains nonaggregated column 'mydb.contact.id' which is not functionally dependent on columns in GROUP BY clause; this is incompatible with sql_mode=only_full_group_by

```
mysql> select @@version;
+-----+
| @@version |
+-----+
| 5.7.10    |
+-----+
1 row in set (0.00 sec)

mysql> select @@GLOBAL.sql_mode;
+-----+
| @@GLOBAL.sql_mode
|
+-----+
|
| ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_
| DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_S
| UBSSTITUTION |
+-----+
+-----+
1 row in set (0.00 sec)

mysql> SET sql_mode = '';
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> select id,name from contact group by name limit 10;
+-----+-----+
```

id	name
84046	张伟
80259	张磊
784	王岩
87685	杨钊

10 rows in set (0.07 sec)

不建议设置 SET sql_mode = "", 正确方式如下:

```
mysql> set global
sql_mode='STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION';
mysql> set session
sql_mode='STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION';
```

或者采用

Adding only one mode to sql_mode without removing existing ones:

```
SET sql_mode=(SELECT CONCAT(@@sql_mode,',<mode_to_add>'));
```

Removing only a specific mode from sql_mode without removing others:

```
SET sql_mode=(SELECT
REPLACE(@@sql_mode,'<mode_to_remove>',''));
```

In your case, if you want to remove only ONLY_FULL_GROUP_BY mode, then use below command:

```
SET sql_mode=(SELECT REPLACE(@@sql_mode, 'ONLY_FULL_GROUP_BY',
```

```
''));
```

6. ERROR 1071 (42000) at line 25: Specified key was too long; max key length is 767 bytes

这个保存通常出现在高版本数据库向低版本数据导入数据，活着云主机例如阿里云。

```
mysql> show variables like '%innodb_large_prefix%';
+-----+-----+
| Variable_name      | Value |
+-----+-----+
| innodb_large_prefix | OFF   |
+-----+-----+
1 row in set (0.00 sec)
```

解决方案

```
innodb_large_prefix=ON
```

7. ERROR 1086 (HY000): File '/var/lib/mysql-files/order.txt' already exists

SELECT * FROM tablename INTO OUTFILE 不支持覆盖操作，这是MySQL从安全角度考虑的。

```
mysql> SELECT * FROM `order` INTO OUTFILE '/var/lib/mysql-  
files/order.txt';  
ERROR 1086 (HY000): File '/var/lib/mysql-files/order.txt'  
already exists
```

8. Error Code: 1146. Table 'test.CACHE_UPDATE' doesn't exist

问题分析，首先确认表是存在的，但是无法读取。可以判定是 lower_case_table_names=1 选项的问题，开启后表以小写方式打开。

```
Error Code: 1146. Table 'test.CACHE_UPDATE' doesn't exist
```

如果是 MySQL 8.0 之前没有开启 lower_case_table_names=1，现在需要开启，加入配置后将无法启动，解决办法是，你需要重做 mysql data 目录

```
[root@localhost ~]# rm -rf /var/lib/mysql/*
[root@localhost ~]# systemctl restart mysqld
[root@localhost ~]# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.21 Source distribution

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show variables like '%lower_case_table_names%';
+-----+-----+
| Variable_name          | Value |
+-----+-----+
| lower_case_table_names | 1     |
+-----+-----+
1 row in set (0.01 sec)

mysql> SELECT CURRENT_SERIAL FROM CACHE_UPDATE WHERE ID=1;
+-----+
| CURRENT_SERIAL |
+-----+
|                7 |
+-----+
```

```
+-----+  
1 row in set (0.00 sec)
```


9. ERROR 1273 (HY000) at line 3364: Unknown collation: 'utf8mb4_0900_ai_ci'

找出指定字符集的表

```
select TABLE_SCHEMA, TABLE_NAME, TABLE_COLLATION from
information_schema.tables where table_collation =
'utf8mb4_0900_ai_ci' and table_schema = 'your_schema';
```

```
SELECT
    CONCAT(
        'ALTER TABLE ',
        TABLE_NAME,
        ' CONVERT TO CHARACTER SET utf8mb4 COLLATE
utf8mb4_general_ci;'
    )
FROM
    information_schema.`TABLES`
WHERE
    TABLE_SCHEMA = 'DATABASE_NAME';
```

10. ERROR 1290 (HY000): The MySQL server is running with the --secure-file-priv option so it cannot execute this statement

MySQL 不允许向 `secure_file_priv` 意外的目录导出文件。

```
mysql> SELECT * FROM `order` INTO OUTFILE '/tmp/order.txt';
ERROR 1290 (HY000): The MySQL server is running with the --
secure-file-priv option so it cannot execute this statement
```

```
mysql> show variables like '%secure%';
```

Variable_name	Value
require_secure_transport	OFF
secure_auth	ON
secure_file_priv	/var/lib/mysql-files/

```
3 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM `order` INTO OUTFILE '/var/lib/mysql-
files/order.txt';
```

```
Query OK, 3 rows affected (0.00 sec)
```

```
root@netkiller ~ % cat /var/lib/mysql-files/order.txt
```

```
1      Tom      22      2017-11-16 17:23:15
2      Neo      34.65   2017-11-16 17:29:28
3      Cici     34.98   2017-11-16 17:30:29
```

在 `my.cnf` 中加入 `secure-file-priv=/tmp` 可以修改为你需要的目录。

11. ERROR 1364: 1364: Field 'id' doesn't have a default value

```
set
@@SESSION.sql_mode='NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION'
;
SELECT @@GLOBAL.sql_mode;
UPDATE `cms`.`content` SET `source`='test' WHERE
`content_id`='1099';
```

12. ERROR 1415: Not allowed to return a result set from a trigger

触发器中不允许返回结果集，解决方法是顶一个变量，然后将返回值返回给变量。

```
DROP TRIGGER IF EXISTS `test`.`demo_AFTER_INSERT`;  
  
DELIMITER $$  
USE `test`$$  
CREATE DEFINER=`root`@`%` TRIGGER `test`.`demo_AFTER_INSERT`  
AFTER INSERT ON `demo` FOR EACH ROW  
BEGIN  
    set @rev = "";  
    SELECT  
    OUT2FILE('/tmp/demo.log',  
            CONCAT_WS(',',  
                      NEW.id,  
                      NEW.name,  
                      NEW.sex,  
                      NEW.address))  
    INTO @rev;  
END$$  
DELIMITER ;
```

13. ERROR 1503 (HY000): A PRIMARY KEY must include all columns in the table's partitioning function

<http://dev.mysql.com/doc/refman/5.1/en/partitioning-limitations-partitioning-keys-unique-keys.html>

14. ERROR 1819 (HY000): Your password does not satisfy the current policy requirements

MySQL 5.7 密码强度，必须含有0-9，a-z,A-Z以及“-”或“_”

<https://dev.mysql.com/doc/refman/5.7/en/validate-password-options-variables.html>

禁用密码安全策略（早起5.7版本可用，新版已经废弃这个选项）

```
password                                # cat /etc/my.cnf | grep validate-  
                                         validate-password=OFF
```

新的方式修改策略与密码长度

```
mysql> set global validate_password_policy=0;  
mysql> set global validate_password_length=4;  
mysql> grant all privileges on test.* to 'test'@localhost  
identified by 'chen';
```

15. ERROR 1820 (HY000): You must reset your password using ALTER USER statement before executing this statement.

这个错误来自 MySQL 5.7，首次登陆MySQL 5.7 必须修改密码

```
ALTER USER 'root'@'localhost'  
IDENTIFIED BY 'your_password';
```

16. ERROR 1840 (HY000) at line 24: @@GLOBAL.GTID_PURGED can only be set when @@GLOBAL.GTID_EXECUTED is empty.

问题出现在 MySQL 5.7 导入数据库时候

```
[www@testing ~]$ zcat netkiller.2021-08-19.sql.gz | mysql
netkiller
ERROR 1840 (HY000) at line 24: @@GLOBAL.GTID_PURGED can only be
set when @@GLOBAL.GTID_EXECUTED is empty.
```

解决方案

执行 reset master;

```
[www@testing ~]$ mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 25669
Server version: 5.7.35 MySQL Community Server (GPL)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All
rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or
its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current
input statement.

mysql> reset master;
```



```
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> exit
```

```
Bye
```

```
[www@testing ~]$ zcat netkiller.2021-08-19.sql.gz | mysql  
netkiller
```

17. ERROR 3024 (HY000): Query execution was interrupted, maximum statement execution time exceeded

```
mysql> select * from cert;
ERROR 3024 (HY000): Query execution was interrupted, maximum
statement execution time exceeded
mysql> SET GLOBAL max_execution_time=10;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from cert;
Empty set (0.08 sec)

mysql> show variables like 'max_execution_time';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| max_execution_time | 10 |
+-----+-----+
1 row in set (0.01 sec)

mysql> select /*+ max_execution_time(3000)*/ count(*) from
cert;
+-----+
| count(*) |
+-----+
|          0 |
+-----+
1 row in set (0.29 sec)
```

18. Authentication plugin

**'caching_sha2_password' cannot be loaded:
/usr/lib64/mysql/plugin/caching_sha2_password.s
o: cannot open shared object file: No such file or
directory**

这个故障出现在 MySQL 8.0 上，用户使用 mysql client 5.7 链接 MySQL 8.0 提示如下

```
[root@netkiller ~]# mysql -h 193.112.95.53 -uroot -p
Enter password:
ERROR 2059 (HY000): Authentication plugin
'caching_sha2_password' cannot be loaded:
/usr/lib64/mysql/plugin/caching_sha2_password.so: cannot open
shared object file: No such file or directory
```

解决方案，创建用户使用 mysql_native_password 密码

```
mysql> CREATE USER 'root'@'%' IDENTIFIED WITH
mysql_native_password BY 'pMQiEgelikst7S_6tlXzB0mt_4b';
Query OK, 0 rows affected (0.08 sec)

mysql> grant all on *.* to 'root'@'%';
Query OK, 0 rows affected (0.08 sec)
```

重新链接

```
[root@netkiller ~]# mysql -h 193.112.95.53 -uneo -p
```

Enter password:

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 24

Server version: 8.0.11 MySQL Community Server - GPL

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

19. com.mysql.jdbc.exceptions.jdbc4.MySQLNonTransientConnectionException: Public Key Retrieval is not allowed

问题出在现在 MySQL 8.0 版本

解决方法：在连接后面添加 allowPublicKeyRetrieval=true

```
spring.datasource.url=jdbc:mysql://192.168.0.1:3306/test?  
useUnicode=true&characterEncoding=UTF-  
8&serverTimezone=UTC&useSSL=false&allowPublicKeyRetrieval=true
```

20. mysqldump: Couldn't execute 'SELECT COLUMN_NAME,

问题出现在 MySQL 8.0 备份 MySQL 5.7 数据库时。

```
mysqldump: Couldn't execute 'SELECT COLUMN_NAME,  
JSON_EXTRACT(HISTOGRAM, '$."number-of-buckets-specified"')  
FROM information_schema.COLUMN_STATISTICS          WHERE  
SCHEMA_NAME = 'testra' AND TABLE_NAME = 'branch';': Unknown  
table 'column_statistics' in information_schema (1109)
```

解决办法，使用 --column-statistics=0 选项

```
mysqldump -hdb.netkiller.cn -uroot -ptest neo --column-  
statistics=0
```

21. this is incompatible with sql_mode=only_full_group_by

```
Expression #2 of SELECT list is not in GROUP BY clause and contains nonaggregated column 'test.table.username' which is not functionally dependent on columns in GROUP BY clause; this is incompatible with sql_mode=only_full_group_by
```

问题出在 MySQL 5.7 向 MySQL 8.0 迁移。

查询 sql_mode 设置

```
select @@global.sql_mode;
'ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION'
```

临时解决方案，去掉 ONLY_FULL_GROUP_BY 即可

```
set
@@GLOBAL.sql_mode='STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,
NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION';
set
@@SESSION.sql_mode='STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,
NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION';
```

彻底解决，有三种解决方案：

- 第一种，将 MySQL 的版本降回到 5.7
- 第二种，关闭 `only_full_group_by` 检查
- 第三种，修改sql使其遵守`only_full_group_by`语法规则

采用哪个方案？打工人的方案，就是解决眼前问题，如果你对项目负责，最好采用第三种方案。

我的职业生涯遇到无数次，因版本太低同时发现重大漏洞，厂商已经不在维护该版本，此时一边是黑客攻击，一边你又无法短时间内升级到新版本，公司又不能接受停服，你会怎么应对？我用了很多非常规手段，给开发团队争取了一周的时间，升级系统。

22. mysqldump: [Warning] Using a password on the command line interface can be insecure.

```
mysqldump: [Warning] Using a password on the command line interface can be insecure.
```

```
vim ~/.my.cnf  
  
[mysqldump]  
user=root  
password=123456
```

```
[root@netkiller neo]# mysqldump -h 192.168.30.40 neo  
[root@netkiller neo]# mysqldump --defaults-extra-file=~/.my.cnf  
netkiller > netkiller.sql
```

23. mysql: [Warning] Using a password on the command line interface can be insecure.

当使用 `-pnetkiller` 在命令行出现密码的时候，会提示下面信息。

```
[www@testing ~]$ mysql -h127.0.0.1 -uneo -pnetkiller test
mysql: [Warning] Using a password on the command line interface
can be insecure.
Reading table information for completion of table and column
names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 25623
Server version: 5.7.35 MySQL Community Server (GPL)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All
rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or
its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current
input statement.

mysql>
```

创建 `~/.my.cnf` 配置文件，将密码写入配置

```
[www@testing ~]$ cat ~/.my.cnf
[mysql]
host=127.0.0.1
```

```
user=neo  
password=netkiller
```

这时直接使用 mysql 命令即可进入。

```
[www@testing ~]$ mysql  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 25622  
Server version: 5.7.35 MySQL Community Server (GPL)  
  
Copyright (c) 2000, 2020, Oracle and/or its affiliates. All  
rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or  
its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current  
input statement.  
  
mysql>
```

第 84 章 数据库迁移

1. MySQL to PostgreSQL

```
[root@localhost ~]# mysqldump --default-character-set=utf8 -h
mysql.netkiller.cn -u root -p -t --skip-extended-insert --
compact mydb mytable | sed 's/`//g' | psql -h
pgsql.netkiller.cn -U neo -W
Password for user lzyun:
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
```