关系型数据库中的基本命令：

DDL

CREATE

ALTER

DROP

DML

INSERT,REPLACE

DELETE

UPDATE

SELECT

DCL

GRANT

REVOKE

DBA:

开发：

数据库设计

代码设计：存储过程，存储函数，触发器

管理：

连接管理及优化

备份及还原

数据库设计

基本语句优化

用户及权限管理

安全管理

数据库软件安装及升级

配置优化

运维主要负责管理工作而不是DBA的代码设计等

5.0

分布式事务（看存储引擎是否支持）

stored procedure，cursors，views，triggers，XA transaction

5.1

（表）分区，基于行的复制、

插件式的API，

5.5

5.6

多线程的复制

6.0

在线备份 falcon和maria存储引擎

缺陷：

子查询，存储过程存储函数还需要优化

安装和升级mysql

安装方式：

1 RPM

2 通用二进制包

3 源码编译安装 //参考mysql编译文章

5.5之后make变为cmake(cross make)先安装cmake

之后再增加配置等

事物日志：

将随机IO转换为顺序IO

重做日志：

undo log 对比









显示全局功能

显示执行信息全局状态信息

mysql的bin目录下有很多二进制程序，各种功能

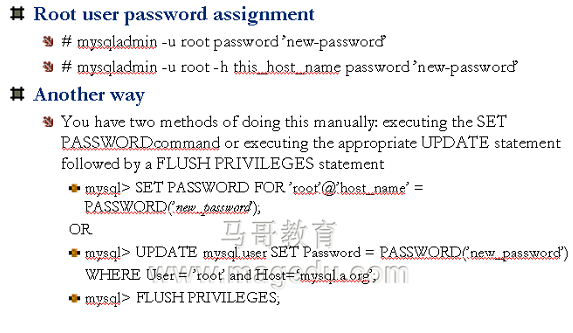
mysql的配置文件：集中式的分段配置

按顺序查找



**安装后的安全设置：**

管理员密码



最后FLUSH 重读授权表

样例：

匿名用户





 两个单引号

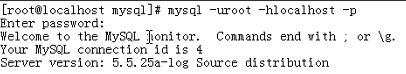








每次登陆输入密码很麻烦，



可以这样，定义自己的配置文件把密码写进去：

#vim .my.cnf



逻辑视角：

表

视图

索引

触发器：事件触发

存储过程

存储函数

事件调度器：时间触发

用户和权限

游标

事务：ACID 原子性 一致性 隔离性 持久性 测试后才成为满足事务的数据库

物理视角：

文件

一表一文件（一表多文件）

多表一文件（表空间,InnoDB）

日志：

错误

查询

事务

二进制

慢查询

中继

mysqld --verbose --help设定的参数 变量

读取配置文件的顺序：

/etc/my.cnf /etc/mysql/my.cnf $MYSQL\_HOME/my.cnf

--default-extra-file= ~/.my.cnf

一些参数可以在配置文件中指定，或者在启动时通过参数指定。

SHOW GLOBAL VARIABLES 配置参数

大多内容可以在配置文件中指定

分为： 动态变量 服务运行可以直接改变的

静态变量 启动前设定好的

SHOW GLOBAL STATUS 状态信息，统计数据

插件式存储引擎：(表级别)

MyISAM：不支持事物

InnoDB：支持事物

CSV：

HEAP:MEMORY 内存存储

MyISAMMRG 多表合并

Federated （6.0以后使用）

Archive

Blackhole:类似于null 复制的时候用到

MySQL命令的工作模式：

交互式模式：连进去执行命令

批处理模式：载入一个sql脚本一次执行多个命令

交互式模式：

选项：

--user=user\_name

-u user\_name 默认root

--password[password], -p[password] (可选)

默认使用空密码

--host=host\_name , -h host\_name

默认localhost

-h localhost方式不同-h 127.0.0.1

localhost基于sock 127.0.0.1基于3306的tcp

--protocol={TCP|SOCKET|PIPE|MEMORY}

--port= port\_num -P port\_num

命令：

客户端命令：本地执行不需要发到远端执行的

都不用使用命令结束符的 如use mysql

服务端命令：

都必须使用命令结束符的； select \*from user

List of all MySQL commands:

Note that all text commands must be first on line and end with ';'

? (\?) Synonym for `help'.

**clear** (\c) Clear command. //相当于此句不执行相当于cancle

connect (\r) Reconnect to the server. Optional arguments are db and host.

**delimiter** (\d) Set statement delimiter. NOTE: Takes the rest of the line as new delimiter.设置命令结束符

edit (\e) Edit command with $EDITOR.

**ego (\G) Send command to mysql server, display result vertically.竖排 \g横排**

**exit (\q) Exit mysql. Same as quit.**

go (\g) Send command to mysql server.

help (\h) Display this help.

nopager (\n) Disable pager, print to stdout.

notee (\t) Don't write into outfile.

pager (\P) Set PAGER [to\_pager]. Print the query results via PAGER.

print (\p) Print current command.

prompt (\R) Change your mysql prompt.

**rehash (\#) Rebuild completion hash.**

**source** (\.) **Execute an SQL script file.** Takes a file name as an argument.

status \s) Get status information from the server.

**system (\!) Execute a system shell command.**

tee (\T) Set outfile [to\_outfile]. Append everything into given outfile.

**use (\u) Use another database. Takes database name as argument.**

charset (\C) Switch to another charset. Might be needed for processing binlog with multi-byte charsets.

**warnings (\W) Show warnings after every statement. 打开警告**

**nowarning (\w) Don't show warnings after every statement. 关闭警告**

批处理模式：

vim test.sql

creat database taetdb;

creat table testdb

mysql中：

\. test.sql **source模式**

或者这样执行：



mysql客户端的功能：

历史

命令补全 在配置问文件中将no 什么hush注释掉，但每次连上去可能会扫描造成卡顿 mysql -A 启动时不补全

命令行编辑 shell中的都管用

ctrl a行首 ctrl e行尾 等

mysql -e “指定mysql command”

如：mysql -e “SHOW DATABASES”

####################################

mysqladmin的用法：

mysqladmin command

例：mysqladmin -uroot ping

子命令：

create 名称 创建数据库

drop

管理相关的命令：

flush-privileges 刷新授权表

flush-status 重置大多数的表

flush-tables 关闭已经打开的表

flush-thread 清空线程池 很可能会清空准备好的缓存

kill

password 改密码，默认改当前用户的密码



flush-hosts 清空dns缓存hostname信息等

**flush-log** 日志滚动 备份的时候很有用

/data/mysql下的mysql-bin会越来越多，而且每次重启都会生成一个。连上之后可以使用#SHOW BINLOGS 查看日志文件，使用#SHOW MASTER STATUS查看当前正在使用的二进制日志

**refresh=flush-host+flush-logs**

status 全局状态信息类似于status的最后一行信息。

--sleep=n 刷新时间 --count=n 刷新几次

extended-status 扩展的状态信息

variables 全局变量信息

version

shutdown

processlist

################################

客户端工具：

mysqldump

mysqlimport

mysqlcheck

非客户端命令：必须在服务器上运行的

myisamchk

myisampack：打包myisam表的数据文件

GUI tools：

phpMyAdmin

SQLyog

MySQL Front

Navicat for MySQL

有的限定管理员账户不能远程连接

mysql开发类的 存储过程 游标等不暂不讲解

################################################

MySQL的数据类型：

字符型：

CHAR(255), VARCHAR(65535): 不区分字母的大小写；有字符集和排序规则

BINARY(255), VARBINARY(65535)：区分字母大小写；而且没有字符集和排序规则

TEXT：TINYTEXT, TEXT, MEDIUMTEXT, LONGTEXT

BLOB:

ENUM, 要么这个，要么另外一个

SET

BIT

数值型：

整数：

TINYINT, 1Byte, 8bit: -128-127, 0-255

SMALLINT, 2Byptes, 16bit: -32768-32767, 0-65535

MEDIUMINT, 3Bytes

INT, 4Bytes

BIGINT 8字节

浮点数：

12.34： 1.234\*10^1, 0.1234\*10^2

二进制：2/5

Decimal: 定点 DECIMAL(g,f) f后面小数点位数，g总位数

日期时间型： 时间修饰符

日期

时间

日期时间

timestamp

YEAR(2), YEAR(4)

字符类型修饰符：

NOT NULL

NULL

DEFAULT ‘value’

BINARY //注意不能BINARY CHAR连用 直接使用BINARY(20)即可

CHARACTER SET 字符集

COLLATION 排序规则 默认从表、库继承

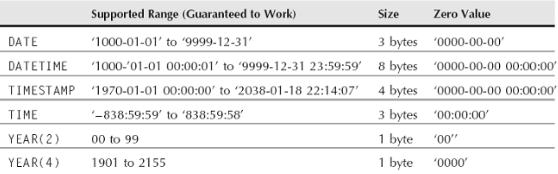
UNSGINED 无符号

数值型修饰符：NOT NULL NULL； DEFAULT # ;UNSGINED

AUTO\_INCREMENT 必须定义为主键或惟一键

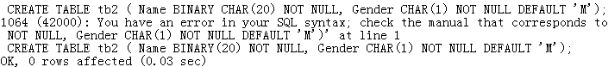
主键：PRIMARY KEY

惟一键： UNIQUE KEY



################################################

样例：









ENUM('M','F')

CHAR(3):Mon, Tue, aaa

ENUM('Mon','Tue','Wen','Thu','Fri','Sat','Sun')

DNS: 数据

ENUM('A', 'AAAA','CNAME','PTR','MX','NS','SOA','SRV','TEXT')

如果修改其中一个顺序会变，大数据量不适合这样存储

SET('a','b','c')

a:100

ab:110

abc:111

ac:101

BIT(2)

sql\_mode:

ANSI

TRADITIONAL

STRICT\_TRANS\_TABLES

STRICT\_ALL\_TRABLES

############################

LIKE:

%: 任意长度的任意字符

\_: 任意单个字符

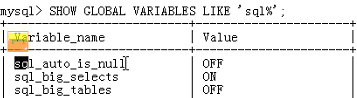
参数：

全局：GLOBAL

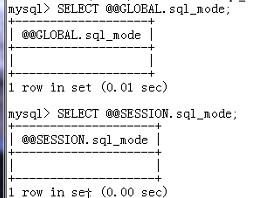
会话：SESSION

SHOW GLOBAL VARIABLES LIKE ''; 显示所有的 全局变量

SHOW SESSION VARIABLES LIKE ''; 会话级别的变量



某种级别的变量值：



SET 设置变量：

SESSION, 立即生效；

GLOBAL: 改变后的值只对新建立的会话；用户有SUPER权限；



永久生效：

#vim /etc/my.cnf

mysqld段中指定变量值

//mysql对数据库、表、视图是否区分大小写取决于文件系统

//存储函数、事件名称、函数名称、关键字不区分大小写

//触发器、表别名区分大小写

//字符串是否区别大小写取决于字符集

//有一个库information-schema，类似于prco目录一样

//在数据目录下创建一个目录在mysql中即有体现

########################

DDL：

CREATE DROP ALTER

这几个的对象主要有

CREATE:

DATABASE

TABLE

INDEX

LOGFILE GROUP

TABLESPACE

TRIGGER

VIEW

EVENT

FUNCTION

PROCEDURE

##################

数据库相关的命令

CREATE DATABASE //HELP加关键词CREATE DATABASE

CREATE SCHEMA IF NOT EXISTS mydb;

SHOW WARNINGS; //这样避免脚本执行到这里中断

改的话也就能改一个字符集和排序规则了

删除 直接删除不可逆

数据库如何改名呢？一种数据库停了，改文件夹名字

其实没有直接的命令

##################

表创建：

HELP CREATE TABLE 这个选项要比库多的多

三种创建表的语法：

CREATE TABLE [IF NOT EXISTS] tb\_name;

CREATE TABLE [IF NOT EXISTS] tb\_name SELECT statement;

把从另一个表中检索的数据创建一个新表,不过这样创建的约束会丢失

CREATE TABLE [IF NOT EXISTS] tb\_name LIKE old\_table\_name；

仅复制表结构不复制数据

如果想复制数据和结构可以使用insert into （后面再说）



竖排使用\G

表选项：

ENGINE [=] engine\_name 指定引擎，不指定使用默认的引擎

AUTO\_INCREMENT [=] value 自动增长



max\_rows 最多多少行

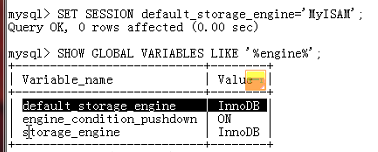
min\_rows 最少多少行

row\_format

tablespace

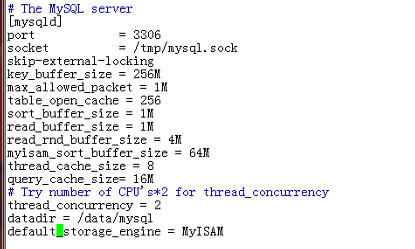
#########################

改变存储引擎



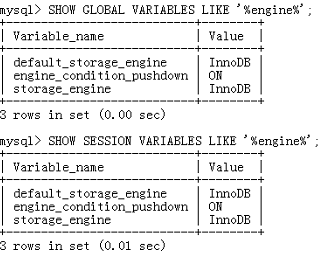
无效。运行时不可修改

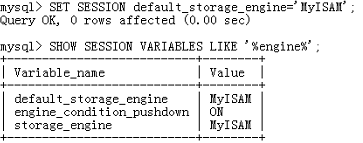
配置文件增加myisam：

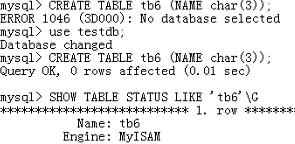


重启服务，这样再看就是myisam的

再改回去：







已经改变了。

改变默认的：



##################################

表更改：例如改字段、存储引擎等

ALTER TABLE：实质就是create然后drop

对一个大表不要这样用

HELP ALTER TABLE:

添加一个字段: ADD

ALTER TABLE tb7 ADD AGE TINYINT UNSIGNED NOT NULL(FIRST);

ALTER TABLE tb7 ADD GENDER ENUM(‘F’,’M’) NOT NULL DEFAULT ‘M’;

指定位置第一个就是FIRST 或者使用AFTER关键字

ALTER TABLE tb7 ADD SALARY FLOAT(10，2) AFTER AGE;

删除一个字段: DROP

ALTER TABLE tb7 DROP SALARY;

修改一个字段:

改字段名称和字段定义： CHANGE

只改字段的定义：MODIFY

ALTER TABLE tb7 MODIFY AGE SMALLINT UNSIGNED NOT NULL;

ALTER TABLE tb7 CHANGE AGE OLD SMALLINT UNSIGNED NOT NULL;

改默认存储引擎

USE testdb;

SHOW TABLE STATUS LIKE ‘tb7’;

ALTER TABLE tb7 ENGINE=’MyISAM’

重命名：

ALTER TABLE tb7 RENAME TO mytb7;

RENAME TABLE mytb7 TO tb7;

删除表：

DROP TABLE ;

如果有外键的时候会影响外部的数据；

###################################################

DML:

1)INSERT

2)REPLACE

3)UPDATE

4)DELETE FROM

1)INSERT (INTO)

INSERT INTO tb\_name (col1,col2) VALUES|VALUE (val1, val2)

INSERT INTO tb\_name VALUES|VALUE (val1, val2) //每个字段都给的话可省略字段

INSERT INTO tb\_name (col1,col2) VALUES|VALUE (val1, val2),(val21,val22),(val31,val32),...

主键：一个或者多个，如果插入的出现了重复的话，不允许插入。

还可以使用SET关键词：

INSERT INTO tb1 SET name=”redhat”,GENDER=’M’；

INSERT INTO tb1 SELECT NAME,GENDER FROM tb4 WHERE ID>=2; //确保字段一样

2)REPLACE INTO

主键如果有些值一样的话可以使用这个替换。

3)UPDATE

一定要指定where否则会改变整张表；



还用where命令；

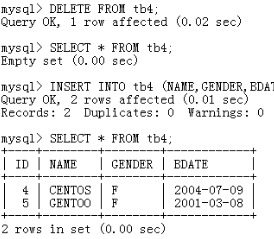
4)DELETE FROM

DELETE FROM tb\_name WHERE condition

DELETE FROM tb\_name ORDER BY 'col' LIMIT n;

DELETE FROM tb\_name LIMIT n; 删除前多少个

 降序排序删除第一个；

DELETE FROM tb4; 会直接清空表！！！！再插入就会从之前的ID接着向下排序。 这个插入还是从4开始的。

想重置顺序的话就是这就从0开始了；

清空整个表：这样就清空表了，并且恢复到初始状态

TRUNCATE TABLE tb\_name;

//有个什么mysql --safe-updates选项 会阻止没有条件的修改和更新。每次生效写入配置文件。