# Mysql学习总结

## 安装与登录

Linux下：参考（<http://www.linuxidc.com/Linux/2016-07/133128.htm>）（最好串行安装）

1. apt-get install mysql-server
2. apt-get install mysql-client
3. apt-get install libmysqlclient-dev
4. 检查：netstat –tap |grep mysql

显示：

命令netstat，参考（<http://www.cnblogs.com/azheng007/p/3159320.html> ）

1. 登录：mysql –u用户名 –p密码 [–P端口 –h服务器地址 –prompt提示符] 注意选项和值之间没有空格

## 卸载

apt-get autoremove --purge mysql-server-5.7

apt-get remove mysql-server

apt-get remove mysql-common

apt-get autoremove mysql-client

清理残留数据

dpkg -l|grep ^rc|awk '{print $2}'|xargs dpkg -P

再安装出错

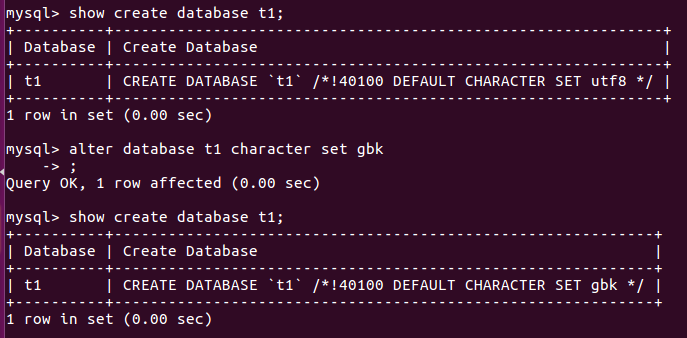
## 查看与创建数据库

1. SHOW DATABASES;
2. CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] 数据库名 [CHARACTE SET [=] 编码方式]
3. 修改数据库编码：

alter database test default character set=utf8;

* 1. ALTER DATABASE 数据库名 CHARACTER SET [=] 编码
  2. SET NAMES UTF8

1. 删除数据库：DROP DATABASE 数据库名
2. 使用数据库：USE 数据库名
3. 查看当前数据库：SELECT DATABASE();
4. 查看数据库结构：SHOW CREATE DATABASE 数据库名



## 查看与创建表

1. 数据类型：
   1. 整型：

|  |  |
| --- | --- |
| TINYINT | 1字节 |
| SMALLINT | 2 |
| MEDIAINT | 3 |
| INT | 4 |
| BIGINT | 8 |

* 1. 浮点型

|  |  |
| --- | --- |
| FLOAT(M,D) | 4 ；M表示总位数，D表示小数点后位数 |
| DOUBLE(M,D) | 8 ；M表示总位数，D表示小数点后位数 |

* 1. 日期

|  |  |
| --- | --- |
| YEAR | 1 |
| DATE | 3 |
| TIME | 3 |
| DATETIME | 8 |
| TIMESTAMP | 4 |

* 1. 字符

|  |  |
| --- | --- |
| CHAR(M) | 最大M字节，不足补空格 |
| VARCHAR(M) | 最大M字节，不会补空格 |
| TINYTEXT |  |
| TEXT |  |
| MEDIATEXT |  |
| BIGTEXT |  |
| ENUM(‘xx’,’xx’,…) |  |
| SET(‘XX’,’XX’,..) |  |

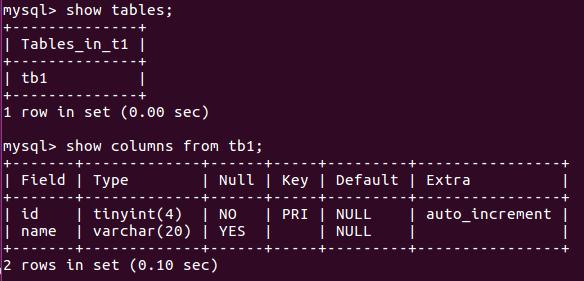
1. 表操作
   1. CREATE TABLE 表名 （

Id 类型 约束，

…

）；

* 1. INSERT INTO 表名[(列名)] VALUES(值)
  2. 查看表结构：SHOW COLUMNS FROM 表名
  3. 约束：空（NULL | NOT NULL）键(PRIMARY KEY , UNIQUE KEY)默认(DEFAULT)额外(AUTO\_INCREMENT)



1. 记录操作
   1. SELECT \* FROM 表名

## 约束以及修改数据表

1. 约束：
   1. 按功能：空（not null）键（primary key ， foreign key，unique key ）默认（default）额外（auto\_increment）
   2. 按数据列数目划分：列级约束、表级约束
2. 修改数据表
   1. 添加、删除、修改列
      1. ALTER TABLE tb1 ADD add\_1 TINYINT UNSIGNED NOT NULL;//添加列名为add\_1的列
      2. ALTER TABLE tb1 DROP add\_1;//删除列add\_1;
      3. ALTER TABLE tb1 MODIFY id SMALLINT UNSIGNED;//修改列id
      4. ALTER TABLE tb1 CHANGE id new\_id SMALLINT;//修改列，更强于modify，可以修改列名
   2. 添加、删除
      1. ALTER TABLE tb1 ADD PRIMARY KEY (id);//添加主键约束
3. ALTER TABLE tb1 DROP PRIMARY KEY;//删除主键约束
4. ALTER TABLE tb1 ADD UNIQUE KEY (uname，..);//添加唯一约束
5. ALTER TABLE tb1 DROP INDEX XXXX,…;//删除唯一约束需要找到它的索引，show index from 表名，注意没有括号。
6. ALTER TABLE tb1 ALTER 列名 SET DEFAULT XXXX;//添加默认属性
7. ALTER TABLE tb1 ALTER 列名 DROP DEFAULT;//删除默认值
   1. 修改表名
      1. ALTER TABLE tb1 RENAME tbb1;//更改列名tb1为tbb1；
      2. RENAME TABLE tbb1 TO/AS tb1;//更改列名tbb1为tb1；
8. 创建外键约束条件：
   1. 数据库引擎INNODB
   2. 参照列和被参照列类型匹配，字符类可以不一致，整型和浮点必须一致。
   3. 自动创建索引
9. 父表和子表对于外键约束四种类型：
   1. CASCODE
   2. SET NULL
   3. RESTRICT
   4. NO ACTION == RESTRICT

## 函数与过程

1. 函数
   1. 定义
      1. 参数 个数<1024
      2. 返回值
   2. 用法：

DELIMITER //

CREATE FUNCTION 函数名（）括号是必须的，参数可选

RETURNS 类型

BEGIN

函数体

RETURN X

END

//

DELIMITER ;

1. 过程
   1. 优点：
      1. 灵活
      2. 效率高
      3. 减少网络流量
   2. 用法
   3. DELIMITER //

CREATE PROCEDURE 过程名（[IN/OUT/INOUT] 变量名）

BEGIN

SELECT XX INTO 变量;

END

//

DELIMITER ;

* 1. 调用
     1. 过程没有参数：CALL f；CALL f();
  2. 过程与函数区别
     1. 过程复杂，函数针对性
     2. 过程可以右多个返回值
     3. 过程独立，函数是组成部分

## 存储引擎

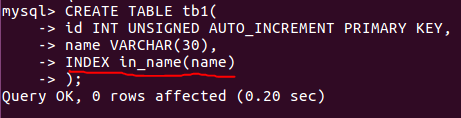
* + - 1. 常用存储引擎
         1. MyISAM
         2. InnoDB
         3. Memery
         4. CSV
         5. Archive
      2. 各个引擎特点

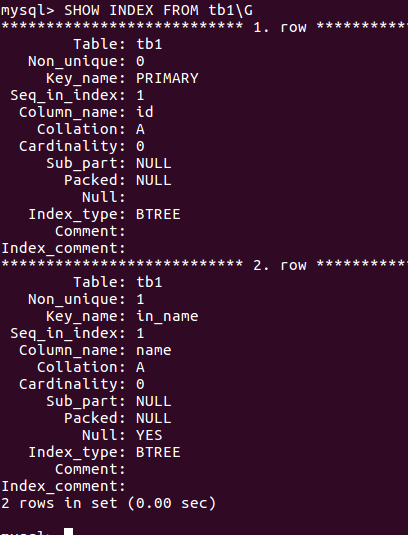
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 大小 | 事务 | 外键 | 索引 | 压缩 | 锁粒度 |
| MyISAM | 256T |  |  | 支持 | 支持 | 表锁 |
| InnoDB | 64T | 支持 | 支持 | 支持 |  | 行锁 |
| Memery | 有限制 |  |  | 支持 |  | 表锁 |
| Archive | 无 |  |  |  | 支持 | 行锁 |

* + - 1. 设置修改引擎
         1. 配置文件
         2. CREATE TABLE 表名（..）ENGINE=XXX
         3. ALTER TABLE 表名 ENGINE=XXX

## 索引

1. 分类
   1. 主键索引
   2. 普通索引
2. 添加索引
   1. CREATE TABLE 表名（

）；



* 1. ALTER TABLE 表名 INDEX index名（字段）
  2. CREATE INDEX 索引名 ON 表名（字段）；

1. 删除索引

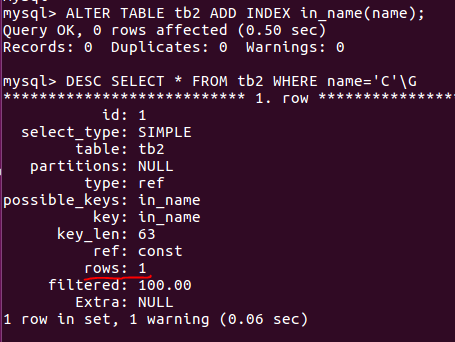
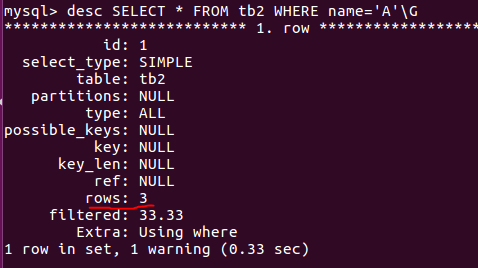
ALTER TABLE 表名 DROP INDEX index名

//查看index名：SHOW INDEX FROM 表名\G 注意不加分号

//查看命令用法: ？ 命令

1. 检测索引

DESC SELECT \* FROM 表名 WHERE XXX=”XXX”\G

查看rows值，即检索次数

## 注意事项

* + - 1. select xx from （A join B on xx=xx ） as T是错误的，join的结果不能取别名
      2. 由外键情况删除：set foreign\_key\_checks=0;
      3. group\_concat(字段 分隔符号 order by ) … group by
      4. 多表删除：delete 表名1，表名2，.. from A join B on xx=xx

当是单表时不用写表名