**MySQL：**

每个命令之间都用分号（；）分开，分号告诉MySQL执行这些命令，否则不会执行这些命令

SQL语句不区分大小写，但数据库和表的名称则区分大小写

数据库包含DDL（数据定义语言）DML（数据操作语言）

在php中使用查询语句的时候，一定要注意之间的空格是不是有，否则查不出结果的。

**登录：**

mysql -h hostname -u username -p password

-h hostname :指定所希望连接的主机 （运行mysql服务器的主机名称）

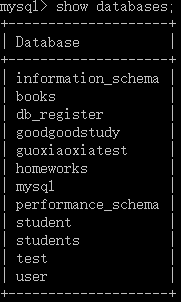
-u username：指定连接数据库时使用的用户名称

-p password：密码

**查看数据库：**

查看数据库列表

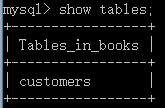
mysql>show databases;



用use dbname跳转到数据库，

显示对应数据库中所有表的清单

mysql>show tables;

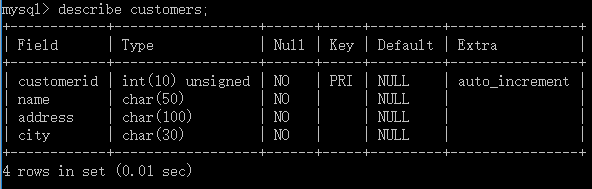


查看特定表的详细信息

mysql>describe customers(tablename);

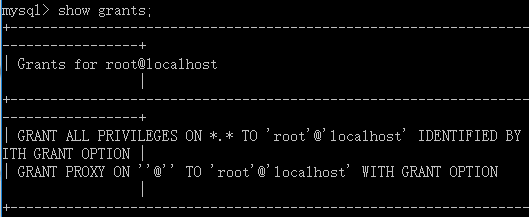
与mysql>show columns from customers;

与mysql>explain books;的效果是一样的



查看用户所拥有的权限

mysql>show grants;



**创建用户并赋予权限：GTANT**

GRANT用来授予MySQL用户的权限，分别为：

**GRANT** privileges(权限组) [columns](对每个列指定权限)

**ON** item（新权限应用于的数据库或表）

**TO** user\_name [**IDENTIFIED BY** ‘password’]

[REQUIRE ssl\_option]（指定用户是否必须通过加密套接字连接）

[WITH [GRANT OPTION | limit\_options]]（允许指定用户向别人授予自己的权限）

Item：

最少权限原则：一个用户应该拥有能执行分配给他的任务的最低级别权限

全局权限：（\*.\*）

数据库权限：dbname.\*

表权限:dbname:tablename

列权限:

user\_name ：用户登录到MySQL使用的用户名，不必与登录系统是使用的用户名一样

创建一个管理员：

mysql>grant all

->on \*

->to guoxiaoxia identified by ‘123456’

->with grant option

**撤回权限：REVOKE**

**REVOKE** privileges(权限组) [columns](对每个列指定权限)

**ON** item（新权限应用于的数据库或表）

**FROM** user\_name

收回：

mysql>revoke all privileges,grant

->from guoxiaoxia

**数据定义语言DDL:**

**创建数据库**

**create database** dbname;

**创建数据库表**

**CREATE TABLE** tablename （

Columnname type limit

）

**创建索引CREATE INDEX:**

**CREATE [UNIQUE|FULLREXT] INDEX** index\_name

**ON** table\_name (index\_column\_name[[length](该字段的前几个字符将被索引) [ASC|DESC],...])

**数据操作语言DML:**

**易错：在php中使用查询语句的时候，一定要注意之间的空格是不是有，否则查不出结果的。**

**插入语句insert：**

**INSERT INTO** tablename [(column1, column2, column3...)] **VALUES**(value1,value2,value3...);

或：

**INSERT INTO** tablename

**set** column1 = value1,

column2 = value2,

column3 = value3;

如：

Insert into customers values (NULL,’Ann’,’BeiJing’,’Airport West’);

或：

Insert into customers

set name=’Ann’,

city=’BeiJing’,

Address=’Airport West’

字符串要包含在一对单引号和双引号中，数字和日期不用

**查询数据select:**

**SELECT** [option] items [INTO file\_details] **FROM** tables

[WHERE conditions]

[GROUP BY group\_type[asc|desc]]

[HAVING where\_definition]

[ORDER BY order\_type]

[LIMIT limit\_criteria]（两个参数：起始行号，返回行数）：limit 2,3;

[PRCEDURE proc\_name(arguments)]

[lock\_options];

若出现了前后的顺序不可以乱了

**从多个表中获取数据：**

（完全关联，交叉关联，笛卡尔乘积）：没有where子句

如：

select orders.orderid,orders.amout（当列名不具有唯一性，用点号表示法：tablename.columnname）

From customers,orders（表名之间的逗号等价于INNER JOIN(两个表的笛卡尔乘积）

Where customers.customerid = orders.customerid;（关联条件，其实已经将关联转化为另一种类型，等价关联）

左关联

以左边行为标准，如果右边没有，结果会增加一行，右边的列为NULL

select orders.orderid,orders.amout

From customers **left join** orders

**on** customers.customerid = orders.customerid;

WHERE子句中实用的比较运算符：

=,>,<,>=,<=,!=

is not null,is null

between：WHERE column BETWEEN value1 AND value2

in,not in: 集合： city IN(‘BeiJing’,’ShangHai’,’GuangZhou’);

like,not like: 用于简单SQL模式匹配（由文本content加上百分号%或下划线\_组成）

%：可以匹配任何数量的任何字符

\_:只匹配一个字符

如：%A：以A结尾的， A%:以A开始的， %A%:包含A的

regexp:用于正则表达式的匹配，MySQL使用POSIX正则表达式，

GROUP BY子句：对根据出现在SELECT子句中的数据进行排序

合计函数：

用于select子句，作用于一个表或表中的一组数据（有**group子句**的时候）

AVG(列):列的平均值

COUNT(项目)：列：所在列的非空行的行数，\*：所在列的包含非空行在内的行数

MIN(列),MAX(列)：列种的最大，最小值

SUN(列)：列种所有值的和

HAVING子句：

用来测试合计函数的结果，类似于用来筛选合计函数结果的where子句，所以一定在group子句后面

LIMIT子句：

指定输出中的哪些行应该返回，如：应用于分页技术

**更新数据update:**

**UPDATE** [LOW\_PRIORITY] [IGNORE] tablename

**SET** column1 = expression1,

column2 = expression2,

column3 = expression3,

[WHERE conditions]

[GROUP BY order\_criteria]

[LIMIT number]（限制受影响的总行数，一个参数：行数）

若出现了前后的顺序不可以乱了

**删除数据delete:**

**DELETE** [LOW\_PRIORITY] [QUICK] [IGNORE]

**FROM** tablename

[WHERE conditions]

[GROUP BY order\_cols]

[LIMIT number]（限制受影响的总行数，一个参数：行数）

若出现了前后的顺序不可以乱了

**事务：**

**事务安全性的4个需求：**

**原子性：**作为一个整体，要么全部执行，要么全部不执行

**一致性：**使数据库处于一致性

**孤立性：**未完成（没有提交）的事务，不能被数据库的其他用户所见

**持续性：**一旦写入，事务必须使永久而其是持续的。

在命令行中输入：mysql>set autocommit=0; //表示在当前会话中关闭自动提交，就是执行的每一个语句都不会被立即写入到数据库中，利用commit可以提交。

如果自动提交被打开：mysql>set autocommit=1;

则需要使用mysql>start transaction; //开启一个事务

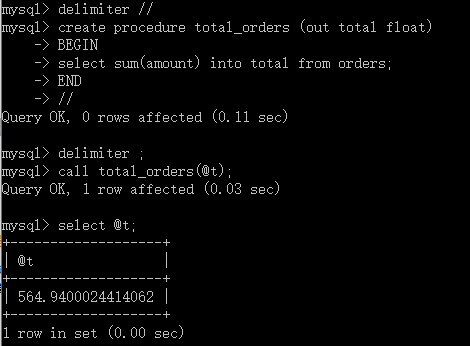
完成事务后，可以用mysql>commit; //进行事务提交

或mysql>rollback; //实现事务回滚

**存储过程：**

一个存储过程是一个可编程的函数，它在MySQL中创建并保存。数据库中的存储过程可以看作是对编程中面向对象方法的模拟。

如：



delimiter:默认的语句末尾为分号(;)时执行语句，改为当语句末尾是双斜杠（//）时执行语句

create procedure:代表创建存储过程

total\_orders:存储过程名

（out（表示参数将被传出或返回，还有in和inout） total（参数名） float（参数类型））

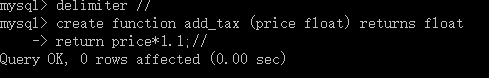
BEGIN和END相当于一对大括号{ }，中间的内容为过程体

call关键字：调用过程

@t：传入一个用来保存结果的变量

通过select @t：查看该变量，就是该存储过程的结果

**MySQL中的函数：**



create function:代表创建函数

add\_tax:函数名

（price(参数名) float（参数类型））

returns float:说明返回值的类型

return ：返回一个值

通过select add\_tax(100)来调用

函数和存储过程可以通过

show create function/procedure 函数名/存储体名：查看他们的代码

drop function/procedure 函数名/存储体名：删除