**创建存储过程：**

**语法：create procedure 名([in|out|inout] 参数名 类型,…)[特性…] 过程题**

**Delimiter //**

**Create procedure proce1(out s int)**

**Begin**

**Select count(\*) into s from user;**

**End;**

**//**

**Delimiter;**

注意：Mysql默认以“;”为分隔符，如果没有声明分隔符，则编译器会把存储过程当成sql语句处理，所以事先要用delimiter //声明当前段分隔符，让编译器把两个//之间的内容当做存储过程的代码，delimiter的意思是把分隔符还原

参数：

**in:**参数的值必须在调用存储过程时指定，在存储过程各种修改该参数的值不能被返回为默认值

**OUT**：可以在存储过程中改变，，并可以返回

**INOUT**调用时指定，并且可以改变和返回

例子：

1. delimiter //

create procedure indemo1(in s int)

begin

select s;

set s=2;

select s;

end; //

delimiter;

调用：

Set @s=1;

Call indemo1(@s);

Select @s;

结果：s=1 s=2 @s=1

S的值在存储过程中被修改，但是@s的值是不会改变的。

1. delimiter //

create procedure outdemo2(out s int)

begin

select s;

set s=2;

select s;

end; //

delimiter;

调用：

Set @s=1;

Call outdemo2(@s);

Select @s;

结果 s=null s=2 @s=2

1. delimiter //

create procedure inoutdemo3(inout s int)

begin

select s;

set s=2;

select s;

end; //

delimiter;

调用：

Set @s=1;

Call inoutdemo3(@s);

Select @s;

**结果：s=1 s=2 @s=2**

**存储过程的查询：**

**Select name from mysql.proc where db=’@dbName’**

**Select rount\_name from information\_schema.rountines where ront\_schema=’@dbName’**

**Show procedure status where db=’@dbName’**

**查看详细信息**

**Show create procedure 数据库.存储过程名**

**存储过程的删除**

**Drop procedure 名,……..**

**DDL数据定义语言：**

**Create database @dbName [character set utf8 collate utf8\_bin];**

**创建一个数据库,使用utf8@charSet，并带有校对规则**

**Show [create] database @dbName;**

**查看数据库定义的信息**

**Show databases;**

**查看有哪些数据库**

**Drop database @dbName;**

**删除数据库、**

**Alter database @dbName characher set @charSet collate @rule;**

**修改数据库信息**

**Use @dbName;**

**切换数据库**

**Create table @tableName(@fieldName @fieldType(@length) [约束],…..)[characher set @charSet collate @rule];**

**创建一张表（如果类型是varchar的话 必须指定长度）**

**约束：**

1. **主键约束：primary key(唯一)，如果主键是整形的话，可以使用auto\_increment（自增长）**
2. **唯一约束：unique，不允许重复**
3. **非空约束：not null，这个字段的值不能为空**

**Show tables;**

**查看数据库中有哪些表**

**Desc @tableName;**

**查看表的结构**

**Alter table @tableName add @fieldName 类型(@length) 约束;**

**给表添加字段**

**Alter table @tableName modify @fieldName 类型(@length) 约束;**

**修改该字段的信息**

**Alert table @tableName drop @fieldName;**

**删除字段**

**Alter table @tableName change @old\_field @new 类型(长度) 约束 after @fieldName /first;**

**更改字段名并且放到某个字段之后 、第一个**

**Rename table @old to @new;**

**更改表名**

**Drop table @tablename;**

**删除表**

**DML数据处理语言：**

**Insert into @tablename [(@fieldname,…..)] values(@value,….);**

**插入表数据**

**Update @tableName set @fieldname=@value,…. [Where @case];**

**修改表数据**

**Delete from @table [where @case];**

**删除表数据**

**Truncate table @table;**

**删除表中所有数据**

**Select [distinct] \* [@field [as @alias],….] from @table [where @case];**

**查询表数据（distinct去重）**

**下面是条件：where**

**@case order by @field asc/desc**

**根据某个字段升序(默认)/降序**

**@case group by @field;**

**对某个字段分组**

**以上语句的执行顺序：**

**1)from 将硬盘上的表文件加载到内存.**

**2)where 将符合条件的数据行 摘取出来，生成一张新的零时表.**

**3)group by 根据列表中的数据种类，将当前临时表划分为若干个新的临时表.**

**4)having 可以过滤掉group by 生成的不符合条件的临时表.**

**5)select 对当前临时表进行整列读取.**

**6)order by 对select生成的临时表，进行重新排序，生成新的临时表.**

**7)limit 对最终生成的临时表的数据行，进行截取.**

**数据库备份**

**Cmd》》》 进入mysql的bin目录下**

**Mysqldump –u root –p @database >d:/XXX.sql (不能以分号结尾)**