**prepare语法格式：**

1. PREPARE stmt from 'sql语句';
2. *#EXECUTE stmt; #适用sql无参数*
3. EXECUTE stmt USING xxx,xxx; *#sql有参数 // 这里USING的只能是会话变量(提前定义好的便令作为参数)*
4. DEALLOCATE PREPARE stmt; *#释放数据库连接*

**作用：**

1. 可以在存储过程中写一些ddl语句（不建议，因为ddl操作会锁表，所以不建已在存储过程中更改表结构）
2. 可以在存储过程中动态的拼接表名，字段名，来达到动态查询的效果
3. sql语句中还可以用?来代表参数，这样可以有效的防止sql注入

示例：

1. delimiter //
2. create procedure myTest()
3. begin
4. set @\_sql = 'select ? + ?';
5. set @a = 5;
6. set @b = 6;
7. PREPARE stmt from @\_sql; // 预定义sql
8. EXECUTE stmt USING @a,@b;// 传入两个会话变量来填充sql中的 ?
9. DEALLOCATE PREPARE stmt; // 释放连接
10. end //

调用上面的存储过程，会得到11的结果，就是这么简单，值得一提的是，如果是要动态的选择表名，表名并不能用 ? 来当占位符。我们只能采用字符串拼接的方法

1. delimiter //
2. create procedure myTest(in columnName varchar(32)) // 传入一个字符串
3. BEGIN
4. drop table if exists tmpTable; // 如果临时表存在先删除掉
5. set @\_sql = concat('create temporary table if not exists tmpTable( ', columnName, ' varchar(32), id int(11), \_name varchar(32));'); // 创建临时表的语法，我们把传入的参数拼接进来
6. PREPARE stmt from @\_sql;
7. EXECUTE stmt;
8. DEALLOCATE PREPARE stmt; // 执行
9. desc tmpTable;
10. end //

 以上存储过程我们可以看到我们传入的字符串可以动态的添加到临时表里面去。

创建临时表时还可以直接从结果集创建。

create table table\_tempname select \* from table\_name where 1=2; (where条件不成立是复制表，条件成立是复制表和数据)

或者create table table\_tempname like table\_name;