Mysql存储过程：是一组为了完成特定功能的sql语句集，经编译后存储在数据库中，用户通过制定存储过程的名字并给定参数(如果带参数)来调用执行它。

存储过程可以看做是对编程中面向对象方法的模拟，允许控制数据的访问方式

存储过程的优点：

1. 增强了sql语言的功能和灵活性。存储过程可以用流程控制语句编写，有很强的灵活性，可以完成复杂的判断和较复杂的运算
2. 允许标准组件式编程。存储过程被创建后，可以在程序中被多次调用，而不必重写该存储过程。专业人员可以随时对存储过程进行修改，对应用程序源代码毫无影响。
3. 存储过程能实现较快的执行速度。
4. 存储过程能够减少网络流量。
5. 存储过程可被作为一种安全机制来充分利用。系统管理员通过执行某一存储过程进行限制，能够实现对相应的数据的访问权限的限制，避免了非授权用户对数据的访问，保证了数据的安全。

存储过程缺点：

1. 调试比较麻烦，不过使用PL/SQL Developer 调试很方便，能弥补这个缺点
2. 移植问题，但如果是工程型项目，基本不存在移植问题
3. 重新编译问题，因为后端代码是运行前编译的，如果带有引用关系的对象发生改变时，受影响的存储过程、包需要重新编译(不过也可以设置为运行时自动编译)
4. 如果一个程序系统中大量使用存储过程，到程序交付使用时随着用户需求的增加会导致数据结构的变化，而且维护系统很麻烦。

过程结果可用select或者set返回

存储过程的创建：

Delimiter //

Create procedure 过程名称([IN |OUT |INOUT ] 参数 type );

Begin

过程语句;

End //

Delimiter ;

语法解析：

1. Delimiter // 和 Delimiter ;表示声明界定符
2. 三种参数类型[IN |OUT |INOUT ] ：
   1. In：输入参数，表示该参数值必须在调用存储过程时指定，在存储过程中修改该参数的值不能别返回，为默认值
   2. Out：输出参数，该值可在存储过程内部被改变，并可返回
   3. Inout：输入输出参数，调用时指定，并可被改变和返回
3. Type：参数的数据类型，该类型为mysql数据库中的任意类型

调用存储过程：

Call 存储过程名(参数值);

删除存储过程：

Drop procedure [ if exists ] 存储过程名

查询现存的存储过程：

select \*from mysql.proc

存储过程相当于自定义函数

存储过程与触发器的不同点：

1. 存储过程有参数、返回值；触发器没有
2. 触发器由系统自动调用，存储过程需要有用户调用才能使用

自定义存储函数：是非常特殊的存储过程

存储过程与函数的区别：

1. 函数区别于存储过程的唯一特点是：函数必须有返回值
2. 存储函数的参数类型类似于in参数
3. 存储过程的参数类型有三种：in、out、inout