

一、存储过程和函数：类似于 java 中的方法。

1. 优点：① 提高代码的复用性
- ② 简化操作。

二、存储过程。

1. 含义：一组预先编译好的 SQL 语句的集合，通常成批处理 SQL 语句
2. 优点：① 提高代码的复用性

② 简化操作。

③ 减少了编译次数 和 与数据库服务器的连接次数，提高效率

3. 创建语法：
`creat procedure 存储过程名(参数列表)`
`begin`
`存储过程体(一组合法的 SQL 语句)`
`end`

*注意：① 参数列表包含三部分：参数模式 参数名 参数类型

`in`：该参数作为输入，即该参数需要调用方法传入值

`out`：—— 输出。即该参数作为返回值。

`inout`：—— 输入和输出。即既需要传入值，又取返回值。

② 如果存储过程体仅有一句话，`begin end` 可省略。

③ 存储过程体中每条 SQL 语句的结尾必须加分号，
—— 该结尾可用 `delimiter` 重新设置。

4. 调用语法：`call 存储过程名(实参列表)`；

```
#1. 空参列表
#案例：插入到 admin 表中五条记录
SELECT * FROM admin;

DELIMITER $
CREATE PROCEDURE myp1()
BEGIN
    INSERT INTO admin(username, 'password')
    VALUES('john1', '0000'), ('lily', '0000'), ('rose', '0000'), ('jack', '0000'), ('tom', '0000');
END $
```

调用
`call myp1();`

```
#2. 创建带 in 模式参数的存储过程
#案例1：创建存储过程实现 根据女神名，查询对应的男神信息

CREATE PROCEDURE myp2(IN beautyName VARCHAR(20))
BEGIN
    SELECT bo.*
    FROM boys bo
    RIGHT JOIN beauty b ON bo.id = b.boyfriend_id
    WHERE b.name=beautyName;
END $
```

调用
`call myp2('杨颖');`

#案例2：创建存储过程实现，用户是否登录成功

```

CREATE PROCEDURE myp4(IN username VARCHAR(20),IN PASSWORD VARCHAR(20))
BEGIN
    DECLARE result INT DEFAULT 0;#声明并初始化
    SELECT COUNT(*) INTO result#赋值
    FROM admin
    WHERE admin.username = username
    AND admin.password = PASSWORD;
    SELECT IF(result>0,'成功','失败');#使用
END $
#调用
CALL myp3('张飞','8888')$

```

3. 创建in,out模式参数存储过程

#案例1：根据女神名，返回对应的男神名

```

CREATE PROCEDURE myp5(IN beautyName VARCHAR(20),OUT boyName VARCHAR(20))
BEGIN
    SELECT bo.boyName INTO boyName
    FROM boys bo
    INNER JOIN beauty b ON bo.id = b.boyfriend_id
    WHERE b.name=beautyName;
END $
#调用
CALL myp5('小昭')$
SELECT @bName$

```

#案例2：根据女神名，返回对应的男神名和男神魅力值

```

CREATE PROCEDURE myp6(IN beautyName VARCHAR(20),OUT boyName VARCHAR(20),OUT userCP INT)
BEGIN
    SELECT bo.boyName ,bo.userCP INTO boyName,userCP
    FROM boys bo
    INNER JOIN beauty b ON bo.id = b.boyfriend_id
    WHERE b.name=beautyName;
END $
#调用
CALL myp6('小昭',@bName,@usercp)$

```

#4. 创建带inout模式参数的存储过程

#案例1：传入a和b两个值，最终a和b都翻倍并返回

```

CREATE PROCEDURE myp8(INOUT a INT ,INOUT b INT)
BEGIN
    SET a=a*2;
    SET b=b*2;
END $
#调用
SET @m=10$
SET @n=20$
CALL myp8(@m,@n)$
SELECT @m,@n$

```

案例：

#一、创建存储过程实现传入用户名和密码，插入到admin表中

```

CREATE PROCEDURE test_pro1(IN username VARCHAR(20),IN loginPwd VARCHAR(20))
BEGIN
    INSERT INTO admin(admin.username,PASSWORD)
    VALUES(username,loginpwd);
END $

```

#二、创建存储过程实现传入女神编号，返回女神名称和女神电话

```

CREATE PROCEDURE test_pro2(IN id INT,OUT NAME VARCHAR(20),OUT phone VARCHAR(20))
BEGIN
    SELECT b.name ,b.phone INTO NAME,phone
    FROM beauty b
    WHERE b.id = id;
END $

```

#三、创建存储过程或函数实现传入两个女神生日，返回大小

```

CREATE PROCEDURE test_pro3(IN birth1 DATETIME,IN birth2 DATETIME,OUT result INT)
BEGIN
    SELECT DATEDIFF(birth1,birth2) INTO result;
END $

```

5. 删除存储过程： drop procedure 存储过程名;

6. 查看 --- : show creat procedure 存储过程名;

案例：

#四、创建存储过程或函数实现传入一个日期，格式化成xx年xx月xx日并返回

```

CREATE PROCEDURE test_pro4(IN mydate DATETIME,OUT strDate VARCHAR(50))
BEGIN
    SELECT DATE_FORMAT(mydate,'%y年%m月%d日') INTO strDate;
END $
CALL test_pro4(NOW(),@str)$
SELECT @str $

```

#五、创建存储过程或函数实现传入女神名称，返回：女神 and 男神 格式的字符串
如 传入：小昭
返回：小昭 AND 张无忌

```

DROP PROCEDURE test_pro5 $
CREATE PROCEDURE test_pro5(IN beautyName VARCHAR(20),OUT str VARCHAR(50))
BEGIN
    SELECT CONCAT(beautyName,' and ',IFNULL(boyName,'null')) INTO str
    FROM boys bo
    RIGHT JOIN beauty b ON b.boyfriend_id = bo.id
    WHERE b.name=beautyName;
END $
CALL test_pro5('柳岩',@str)$
SELECT @str $

```

#六、创建存储过程或函数，根据传入的条目数和起始索引，查询beauty表的记录

```

DROP PROCEDURE test_pro6$
CREATE PROCEDURE test_pro6(IN startIndex INT,IN size INT)
BEGIN
    SELECT * FROM beauty LIMIT startIndex,size;
END $
CALL test_pro6(3,5)$

```

1. 含义: 一组预先编译好的 SQL 语句的集合, 编译成批处理语句

2. 优点: ① 提高代码的复用性

② 简化操作.

③ 减少编译次数和与数据库服务器的连接次数, 提高效率

3. 函数与存储过程的区别.

存储过程: 可以有 0 个返回, 也可以有 1 个返回. 适合做批量插入、更新

函数: 有且仅有 1 个返回, 适合做处理数据后返回一个结果.

4. 创建语法: `creat function 函数名(参数列表) returns 返回类型.
begin
 函数体
end`

注意: ① 参数列表包含两部分: 参数名 参数类型

② 函数体: 肯定会有 return 语句, 如果没有会报错.

如果 return 不放在最后不报错, 但不建议

③ 函数体仅有一句话, 可省略 begin end.

④ 使用 delimiter 语句设置结束标识

5. 调用语法: `select 函数名(参数列表)`

案例:

```
#1. 无参有返回
#案例: 返回公司的员工个数
CREATE FUNCTION myf1() RETURNS INT
BEGIN
    DECLARE c INT DEFAULT 0; #定义变量
    SELECT COUNT(*) INTO c #赋值
    FROM employees;
    RETURN c;
END $
SELECT myf1();
```

```
#2. 有参有返回
#案例1: 根据员工名, 返回它的工资
CREATE FUNCTION myf2(empName VARCHAR(20)) RETURNS DOUBLE
BEGIN
    SET @sal=0; #定义用户变量
    SELECT salary INTO @sal #赋值
    FROM employees
    WHERE last_name = empName;
    RETURN @sal;
END $
SELECT myf2('k_ing');
```

```
#案例2: 根据部门名, 返回该部门的平均工资
CREATE FUNCTION myf3(deptName VARCHAR(20)) RETURNS DOUBLE
BEGIN
    DECLARE sal DOUBLE;
    SELECT AVG(salary) INTO sal
    FROM employees e
    JOIN departments d ON e.department_id = d.department_id
    WHERE d.department_name=deptName;
    RETURN sal;
END $
SELECT myf3('IT');
```

6. 查看函数: `show creat function 函数名;`

7. 删除项: drop function 函数名;

