

第15章_存储过程与函数

讲师：尚硅谷-宋红康（江湖人称：康师傅）

官网：<http://www.atguigu.com>

存储过程练习

#0.准备工作

```
CREATE DATABASE test15_pro_func;
```

```
USE test15_pro_func;
```

#1. 创建存储过程insert_user(),实现传入用户名和密码，插入到admin表中

```
CREATE TABLE admin(  
id INT PRIMARY KEY AUTO_INCREMENT,  
user_name VARCHAR(15) NOT NULL,  
pwd VARCHAR(25) NOT NULL  
);
```

#2. 创建存储过程get_phone(),实现传入女神编号，返回女神姓名和女神电话

```
CREATE TABLE beauty(  
id INT PRIMARY KEY AUTO_INCREMENT,  
NAME VARCHAR(15) NOT NULL,  
phone VARCHAR(15) UNIQUE,  
birth DATE  
);  
  
INSERT INTO beauty(NAME,phone,birth)  
VALUES  
( '朱茵', '13201233453', '1982-02-12' ),  
( '孙燕姿', '13501233653', '1980-12-09' ),  
( '田馥甄', '13651238755', '1983-08-21' ),  
( '邓紫棋', '17843283452', '1991-11-12' ),  
( '刘若英', '18635575464', '1989-05-18' ),  
( '杨超越', '13761238755', '1994-05-11' );
```

```
SELECT * FROM beauty;
```

#3. 创建存储过程date_diff(),实现传入两个女神生日，返回日期间隔大小

#4. 创建存储过程format_date(),实现传入一个日期，格式化成年xx月xx日并返回

#5. 创建存储过程beauty_limit(),根据传入的起始索引和条目数，查询女神表的记录

#创建带inout模式参数的存储过程

#6. 传入a和b两个值，最终a和b都翻倍并返回

#7. 删除题目5的存储过程

#8. 查看题目6中存储过程的信息

答案:

#0.准备工作

```
CREATE DATABASE test15_pro_func;
```

#1. 创建存储过程insert_user(),实现传入用户名和密码，插入到admin表中

```
CREATE TABLE admin(  
id INT PRIMARY KEY AUTO_INCREMENT,  
user_name VARCHAR(15) NOT NULL,  
pwd VARCHAR(25) NOT NULL  
  
);
```

```
DELIMITER //
```

```
CREATE PROCEDURE insert_user(IN username VARCHAR(20),IN loginPwd VARCHAR(20))  
BEGIN  
    INSERT INTO admin(user_name,pwd)  
    VALUES(username,loginpwd);  
END //
```

```
DELIMITER ;
```

#2. 创建存储过程get_phone(),实现传入女神编号，返回女神姓名和女神电话

```
DELIMITER //
```

```
CREATE PROCEDURE get_phone(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 //
```

```
DELIMITER ;
```

#调用

```
CALL get_phone(1,@name,@phone);  
SELECT @name,@phone;
```

#3. 创建存储过程date_diff(),实现传入两个女神生日，返回日期间隔大小

```
DELIMITER //
```

```
CREATE PROCEDURE date_diff(IN birth1 DATETIME,IN birth2 DATETIME,OUT result INT)  
BEGIN
```

```

    SELECT DATEDIFF(birth1,birth2) INTO result;
END //
DELIMITER ;

#调用
SET @birth1 = '1992-09-08';
SET @birth2 = '1989-01-03';
CALL date_diff(@birth1,@birth2,@result);

SELECT @result;

#4. 创建存储过程format_date(),实现传入一个日期, 格式化成xx年xx月xx日并返回
DELIMITER //

CREATE PROCEDURE format_date(IN mydate DATETIME,OUT strdate VARCHAR(50))
BEGIN
    SELECT DATE_FORMAT(mydate,'%y年%m月%d日') INTO strDate;
END //

DELIMITER ;

#调用
SET @mydate = '1992-09-08';
CALL format_date(@mydate,@strdate);

SELECT @strdate;

#5. 创建存储过程beauty_limit(), 根据传入的起始索引和条目数, 查询女神表的记录
DELIMITER //

CREATE PROCEDURE beauty_limit(IN startIndex INT,IN size INT)
BEGIN
    SELECT * FROM beauty LIMIT startIndex,size;
END //

DELIMITER ;

#调用
CALL beauty_limit(1,3);

#创建带inout模式参数的存储过程
#6. 传入a和b两个值, 最终a和b都翻倍并返回
DELIMITER //

CREATE PROCEDURE add_double(INOUT a INT ,INOUT b INT)
BEGIN
    SET a = a * 2;
    SET b = b * 2;
END //

DELIMITER ;

#调用
SET @a = 3,@b = 5;
CALL add_double(@a,@b);

```

```
SELECT @a,@b;
```

#7. 删除题目5的存储过程

```
DROP PROCEDURE beauty_limit;
```

#8. 查看题目6中存储过程的信息

```
SHOW CREATE PROCEDURE add_double;
```

```
SHOW PROCEDURE STATUS LIKE 'add_double';
```

存储函数练习

题目：

#0. 准备工作

```
USE test15_pro_func;
```

```
CREATE TABLE employees
```

```
AS
```

```
SELECT * FROM atguigudb.`employees`;
```

```
CREATE TABLE departments
```

```
AS
```

```
SELECT * FROM atguigudb.`departments`;
```

#无参有返回

#1. 创建函数`get_count()`, 返回公司的员工个数

#有参有返回

#2. 创建函数`ename_salary()`, 根据员工姓名, 返回它的工资

#3. 创建函数`dept_sal()`, 根据部门名, 返回该部门的平均工资

#4. 创建函数`add_float()`, 实现传入两个`float`, 返回二者之和

答案：

#0. 准备工作

```
USE testtesttest;
```

```
CREATE TABLE employees
```

```
AS
```

```
SELECT * FROM atguigudb.`employees`;
```

```
CREATE TABLE departments
```

```
AS
```

```
SELECT * FROM atguigudb.`departments`;
```

#无参有返回

#1. 创建函数get_count(), 返回公司的员工个数

DELIMITER //

```
CREATE FUNCTION get_count() RETURNS INT
BEGIN
    RETURN (SELECT COUNT(*) FROM employees);

END //
```

DELIMITER ;

#调用

```
SELECT get_count();
```

#有参有返回

#2. 创建函数ename_salary(), 根据员工姓名, 返回它的工资

DELIMITER //

```
CREATE FUNCTION ename_salary(emp_name VARCHAR(20)) RETURNS DOUBLE
BEGIN
    RETURN (
        SELECT salary
        FROM employees
        WHERE last_name = emp_name
    );

END //
```

DELIMITER ;

#调用

```
SELECT ename_salary('Abel');
```

#3. 创建函数dept_sal(), 根据部门名, 返回该部门的平均工资

DELIMITER //

```
CREATE FUNCTION dept_sal(dept_name VARCHAR(20)) RETURNS DOUBLE
BEGIN
    RETURN (
        SELECT AVG(salary)
        FROM employees e JOIN departments d
        ON e.department_id = d.department_id
        WHERE d.department_name = dept_name
    );

END //
```

DELIMITER ;

#调用

```
SELECT dept_sal('Marketing');
```

#4. 创建函数add_float(), 实现传入两个float, 返回二者之和

DELIMITER //

```
CREATE FUNCTION add_float(num1 FLOAT, num2 FLOAT) RETURNS FLOAT
BEGIN
    RETURN (SELECT num1 + num2 );
```

```
END //
```

```
DELIMITER ;
```

```
#调用
```

```
SET @num1 := 1.2, @num2 = 3.2;
```

```
SELECT add_float(@num1, @num2);
```