第16章_变量、流程控制与游标

讲师：尚硅谷-宋红康（江湖人称：康师傅）

官网： http://www.atguigu.com

---

1. 变量

题目：

```
#0.准备工作
CREATE DATABASE test16_var_cur;

use test16_var_cur;

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
CREATE DATABASE test16_var_cur;

use test16_var_cur;

CREATE TABLE employees
AS
SELECT * FROM atguigudb.`employees`;

CREATE TABLE departments
AS
SELECT * FROM atguigudb.`departments`;
```

```sql
#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

    DECLARE c INT DEFAULT 0;#定义局部变量
    SELECT COUNT(*) INTO c#赋值
    FROM employees;
    RETURN c;

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(15))
RETURNS DOUBLE
BEGIN
    SET @sal=0;#定义用户变量
    SELECT salary INTO @sal   #赋值
    FROM employees
    WHERE last_name = emp_name;

    RETURN @sal;
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(15))
RETURNS DOUBLE
BEGIN
    DECLARE avg_sal DOUBLE ;
    SELECT AVG(salary) INTO avg_sal
    FROM employees e
    JOIN departments d ON e.department_id = d.department_id
    WHERE d.department_name=dept_name;

    RETURN avg_sal;
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(value1 FLOAT,value2 FLOAT)
RETURNS FLOAT
BEGIN
    DECLARE SUM FLOAT;
    SET SUM=value1+value2;
    RETURN SUM;
END //

DELIMITER ;

#调用
SET @v1 := 12.2;
SET @v2 = 2.3;
SELECT add_float(@v1,@v2);
```

2. 流程控制

题目：

```
#1. 创建函数test_if_case()，实现传入成绩，如果成绩>90,返回A，如果成绩>80,返回B，如果成绩>60,返回C，否则返回D
#要求：分别使用if结构和case结构实现

#2. 创建存储过程test_if_pro()，传入工资值，如果工资值<3000,则删除工资为此值的员工，如果3000 <= 工资值 <= 5000,则修改此工资值的员工薪资涨1000，否则涨工资500

#3. 创建存储过程insert_data(),传入参数为 IN 的 INT 类型变量 insert_count,实现向admin表中批量插入insert_count条记录

CREATE TABLE admin(
id INT PRIMARY KEY AUTO_INCREMENT,
user_name VARCHAR(25) NOT NULL,
user_pwd VARCHAR(35) NOT NULL
);

SELECT * FROM admin;
```

答案：

```
#1. 创建函数test_if_case()，实现传入成绩，如果成绩>90,返回A，如果成绩>80,返回B，如果成绩>60,返回C，否则返回D
#要求：分别使用if结构和case结构实现
#方式1：
DELIMITER //

CREATE FUNCTION test_if_case1(score DOUBLE)
RETURNS CHAR
BEGIN
    DECLARE ch CHAR;

    IF score>90
```

```
        THEN SET ch='A';
    ELSEIF score>80
        THEN SET ch='B';
    ELSEIF score>60
        THEN SET ch='C';
    ELSE SET ch='D';
    END IF;

    RETURN ch;

END //

DELIMITER ;



#调用
SELECT test_if_case1(87);

#方式2：
DELIMITER //

CREATE FUNCTION test_if_case2(score DOUBLE)
RETURNS CHAR
BEGIN
    DECLARE ch CHAR;

    CASE
    WHEN score>90 THEN SET ch='A';
    WHEN score>80 THEN SET ch='B';
    WHEN score>60 THEN SET ch='C';
    ELSE SET ch='D';
    END CASE;

    RETURN ch;
END //

DELIMITER ;

#调用
SELECT test_if_case2(67);

#2. 创建存储过程test_if_pro()，传入工资值，如果工资值<3000,则删除工资为此值的员工，如果3000 <= 工资值 <= 5000,则修改此工资值的员工薪资涨1000，否则涨工资500
DELIMITER //

CREATE PROCEDURE test_if_pro(IN sal DOUBLE)
BEGIN
    IF sal<3000
        THEN DELETE FROM employees WHERE salary = sal;
    ELSEIF sal <= 5000
        THEN UPDATE employees SET salary = salary+1000 WHERE salary = sal;
    ELSE
        UPDATE employees SET salary = salary+500 WHERE salary = sal;
    END IF;

END //
```

```
DELIMITER ;

SELECT * FROM employees;

#调用
CALL test_if_pro(3100);



#3. 创建存储过程insert_data(),传入参数为 IN 的 INT 类型变量 insert_count,实现向admin表中批量插入insert_count条记录

DELIMITER //

CREATE PROCEDURE insert_data(IN insert_count INT)
BEGIN
    DECLARE i INT DEFAULT 1;
    WHILE i <= insert_count DO
        INSERT INTO admin(user_name,user_pwd) VALUES(CONCAT('Rose-',i),ROUND(RAND() * 100000));
        SET i=i+1;
    END WHILE;

END //

DELIMITER ;

#调用
CALL insert_data(100);
```

3. 游标的使用

创建存储过程update_salary()，参数1为 IN 的INT型变量dept_id，表示部门id；参数2为 IN的INT型变量change_sal_count，表示要调整薪资的员工个数。查询指定id部门的员工信息，按照salary升序排列，根据hire_date的情况，调整前change_sal_count个员工的薪资，详情如下。

hire_date	salary
hire_date < 1995	salary = salary*1.2
hire_date >=1995 and hire_date <= 1998	salary = salary*1.15
hire_date > 1998 and hire_date <= 2001	salary = salary *1.10
hire_date > 2001	salary = salary * 1.05

答案：

```
DELIMITER //

CREATE PROCEDURE update_salary(IN dept_id INT,IN change_sal_count INT)
BEGIN
```

```
    #声明变量
    DECLARE int_count INT DEFAULT 0;
    DECLARE salary_rate DOUBLE DEFAULT 0.0;

    DECLARE emp_id INT;
    DECLARE emp_hire_date DATE;


    #声明游标
    DECLARE emp_cursor CURSOR FOR SELECT employee_id,hire_date FROM employees
    WHERE department_id = dept_id ORDER BY salary ;
    #打开游标
    OPEN emp_cursor;

    WHILE int_count < change_sal_count DO
        #使用游标
        FETCH emp_cursor INTO emp_id,emp_hire_date;

        IF(YEAR(emp_hire_date) < 1995)
            THEN SET salary_rate = 1.2;
        ELSEIF(YEAR(emp_hire_date) <= 1998)
            THEN SET salary_rate = 1.15;
        ELSEIF(YEAR(emp_hire_date) <= 2001)
            THEN SET salary_rate = 1.10;
        ELSE SET salary_rate = 1.05;
        END IF;

        #更新工资
        UPDATE employees SET salary = salary * salary_rate
        WHERE employee_id = emp_id;

        #迭代条件
        SET int_count = int_count + 1;

    END WHILE;

    #关闭游标
    CLOSE emp_cursor;

END //

DELIMITER ;

# 调用
CALL update_salary(50,2);
```