

流程控制语句

- 顺序结构: 程序从上往下依次执行
- 分支结构: 程序从多条或多条路径中選擇一条去执行
- 循环结构: 程序在满足一定条件的基础上, 重复执行一段代码

一、分支结构

1. if 语句

- ① 功能: 实现简单的分支
- ② 语法: `if (表达式1, -- 2, -- 3)`
- ③ 执行顺序: 如果表达式1成立, 则返回 -- 2, 否则返回 -- 3.
- ④ 应用: 任何地方.

2. case 语句

- ① 情况一: 类似于 Java 中的 `switch` 语句, 极用于实现多值判断

语法: `case 变量/表达式/语句`
`when 要判断F值 Then 返回的值1或语句1;`
`when 要判断F值 Then 返回的值2或语句2;`
`...`
`else 要返回的值n或语句n;`
`end case;`

- ② 情况二: `--- 多重if语句, --- 区间判断.`

语法: `case 变量/表达式/语句`
`when 要判断条件1 Then 返回的值1或语句1;`
`when 要判断条件2 Then 返回的值2或语句2;`
`...`
`else 要返回的值n或语句n;`
`end case;`

- ③ 特点: a. 可作为表达式, 嵌套在其他语句中使用, 可放在任何地方, `begin end` 中或 `begin end` 的外面.

可作为独立语句使用, 只能放在 `begin end` 中.

- b. 如果 `when` 中的值满足或条件成立, 则执行对应的 `then`

后面的语句, 并且结束 case; 如果都不满足, 则执行 else 中的语句或值.

c. else 可省略, 若省略且所有 when 事件不满足, 则 select null.

④ 案例:

```
#创建存储过程, 根据传入的成绩, 来显示等级, 比如传入的成绩: 90-100, 显示A, 80-90, 显示B, 60-80, 显示C, 否则, 显示D
CREATE PROCEDURE test_case(IN score INT)
BEGIN
    CASE
        WHEN score >= 90 AND score <= 100 THEN SELECT 'A';
        WHEN score >= 80 THEN SELECT 'B';
        WHEN score >= 60 THEN SELECT 'C';
        ELSE SELECT 'D';
    END CASE;
END $
CALL test_case(95)$
```

3. if 结构

① 功能: 实现了分支

② 语法: if 事件1 then 语句1;

else if 事件2 then 语句2;

[else 语句n;]

end if;

③ 应用: 应用在 begin end 中.

④ 案例:

```
#案例1: 根据传入的成绩, 来显示等级, 比如传入的成绩: 90-100, 返回A, 80-90, 返回B, 60-80, 返回C, 否则, 返回D
CREATE FUNCTION test_if(score INT) RETURNS CHAR
BEGIN
    IF score >= 90 AND score <= 100 THEN RETURN 'A';
    ELSEIF score >= 80 THEN RETURN 'B';
    ELSEIF score >= 60 THEN RETURN 'C';
    ELSE RETURN 'D';
    END IF;
END $
```

二、循环结构

1. 分类: while, loop, repeat

2. 循环控制: iterate. 类似于 continue, 继续, 结束本次循环, 开始下一次
leave. 类似于 break, 跳出, 结束当前所在的所有循环.

3. while 语法: [标签:] while 循环条件 do
循环体;
end while [标签];

Java 中
while (循环条件) {
循环体;
}

4. loop 语法: [标签:] loop
循环体;

end loop [标签];

可用于模拟简单的死循环

5. repeat语法: [标签:] repeat

循环体;

until 结束循环的条件

end repeat [标签];

6 案例:

```
#1. 没有添加循环控制语句
#案例: 批量插入, 根据次数插入到admin表中多条记录
DROP PROCEDURE pro_while1$
CREATE PROCEDURE pro_while1(IN insertCount INT)
BEGIN
    DECLARE i INT DEFAULT 1;
    WHILE i<=insertCount DO
        INSERT INTO admin(username, 'password') VALUES(CONCAT('Rose', i), '666');
        SET i=i+1;
    END WHILE;
END $

CALL pro_while1(100)$
```

```
#2. 添加leave语句
#案例: 批量插入, 根据次数插入到admin表中多条记录, 如果次数>20则停止
TRUNCATE TABLE admin$
DROP PROCEDURE test_while1$
CREATE PROCEDURE test_while1(IN insertCount INT)
BEGIN
    DECLARE i INT DEFAULT 1;
    a:WHILE i<=insertCount DO
        INSERT INTO admin(username, 'password') VALUES(CONCAT('xiaohua', i), '0000');
        IF i>=20 THEN LEAVE a;
        END IF;
        SET i=i+1;
    END WHILE a;
END $

CALL test_while1(100)$
```

```
#3. 添加iterate语句
#案例: 批量插入, 根据次数插入到admin表中多条记录, 只插入偶数次
TRUNCATE TABLE admin$
DROP PROCEDURE test_while1$
CREATE PROCEDURE test_while1(IN insertCount INT)
BEGIN
    DECLARE i INT DEFAULT 0;
    a:WHILE i<=insertCount DO
        SET i=i+1;
        IF MOD(i, 2) != 0 THEN ITERATE a;
        END IF;
        INSERT INTO admin(username, 'password') VALUES(CONCAT('xiaohua', i), '0000');
    END WHILE a;
END $
```