
8.1.3 流程控制语句

在数据库编程中常常需要使用流程控制语句来控制程序的流程。MySQL 中流程控制语句主要有：IF 语句、CASE 语句、LOOP 语句、LEAVE 语句、ITERATE 语句、REPEAT 语句和 WHILE 语句等。

1. IF 语句

IF 语句用来进行条件判断,根据是否满足条件(可包含多个条件),来执行不同的语句,是流程控制中最常用的判断语句。其语法的基本形式如下:

```
IF search_condition THEN statement_list
  [ELSEIF search_condition THEN statement_list]...
  [ELSE statement_list]
END IF;
```

其中, search_condition 参数表示条件判断语句,如果返回值为 TRUE,相应的 SQL 语句列表 statement_list 被执行;如果返回值为 FALSE,则 ELSE 子句的语句列表被执行。statement_list 可以包括一个或多个语句。

注意:上节中的 IF()函数不同于这里的 IF 语句。

【例 8-7】统计各个年龄阶层的人数。年龄分为:小于 20 岁,在 20-40 岁之间的,大于 40 岁。各个年龄阶层的人数分别存储在变量@count1、@count2、@count3 中。代码如下:

```
IF age>20 THEN SET @count1=@count1+1;
  ELSEIF age>=20 and age<=40 THEN SET @count2=@count2+1;
  ELSE SET @count3=@count3+1;
END IF;
```

2. CASE 语句

CASE 语句也是用来进行条件判断的,它提供了多个条件进行选择,可以实现比 IF 语句更复杂的条件判断。CASE 语句的基本形式如下:

```
CASE
  WHEN search_condition THEN statement_list
  [WHEN search_condition THEN statement_list] ...
  [ELSE statement_list]
END CASE;
```

其中, search_condition 参数表示条件判断语句; statement_list 参数表示不同条件的执行语句。该语句中的 WHEN 语句将被逐个执行,直到某个 search_condition 表达式为真,则执行对应 THEN 关键字后面的 statement_list 语句。如果没有条件匹配,ELSE 子句里的语句被执行。CASE 语句都要使用 END CASE 结束。

【例 8-8】使用 CASE 统计各个年龄阶层的人数。年龄分为:小于 20 岁,在 20-40 岁之间的,大于 40 岁。各个年龄阶层的人数分别存储在变量@count1、@count2、@count3 中。代码如下:

```
CASE
  WHEN age>20 THEN SET @count1=@count1+1;
  WHEN age>=20 and age<=40 THEN SET @count2=@count2+1;
  ELSE SET @count3=@count3+1;
END CASE;
```

3. LOOP LEAVE 语句

LOOP 语句可以使某些特定的语句重复执行。与 IF 和 CASE 语句相比，LOOP 只实现了一个简单的循环，并不进行条件判断。LOOP 语句本身没有停止循环的语句，必须使用 LEAVE 语句等才能停止循环，跳出循环过程。LOOP 语句的基本形式如下：

```
[label:]LOOP
    statement_list
LEAVE label
END LOOP [label]
```

其中，label 参数表示循环开始和结束的标志，可以省略；statement_list 参数表示需要循环执行的语句。

【例 8-9】统计循环次数，当循环到 100 次时退出循环。代码如下：

```
add_num:LOOP
    SET @count=@count+1;
    IF @count=100 THEN
        LEAVE add_num;
    END LOOP add_num;
```

4. ITERATE 语句

ITERATE 是“再次循环”的意思，用来跳出本次循环，直接进入下一次循环。ITERATE 语句的基本语法形式如下：

```
ITERATE label
```

其中，label 参数表示循环的标志，ITERATE 语句必须跟在循环标志前面。

【例 8-10】请统计从 1 到 1000 之间数字为 7 的倍数的个数。代码如下：

```
add_num:LOOP
    SET @count=@count+1;
    IF @count=1000 THEN
        LEAVE add_num;
    ELSE IF MOD(@count,7)=0 THEN
        ITERATE add_num;
        @count1=@count1+1;
    END LOOP add_num;
select @count-@count1;
```

该示例循环执行@count 加 1 的操作，@count 值为 1000 时结束循环。如果@count 的值能够整除 7，则跳出本次循环，不再执行下面的@count1 加 1 的操作。

说明：LEAVE 语句和 ITERATE 语句都用来跳出循环语句，但两者的功能是不一样的。LEAVE 语句是跳出整个循环，然后执行循环后面的程序。而 ITERATE 语句是跳出本次循环，然后进入下一次循环。

6. REPEAT 语句

REPEAT 语句是有条件控制的循环语句，每次语句执行完毕后，会对条件表达式进行判断，如果表达式返回值为 TRUE，则循环结束，否则重复执行循环中的语句。REPEAT 语句的基本语法形式如下：

```
[label:] REPEAT
```

```
statement_list
UNTIL search_condition
END REPEAT [label]
```

其中：

label 为 REPEAT 语句的标注名称，该参数可以省略；

REPEAT 语句内的语句被重复，直至 search_condition 返回值为 TRUE。

statement_list 参数表示循环的执行语句；

search_condition 参数表示结束循环的条件，满足该条件时循环结束。

REPEAT 循环用 END REPEAT 结束。

【例 8-11】请使用 REPEAT 完成**【例 8-9】**。代码如下：

```
REPEAT
    SET @count=@count+1;
    UNTIL @count=100
END REPEAT;
```

7. WHILE 语句

WHILE 语句也是有条件控制的循环语句。WHILE 语句和 REPEAT 语句不同的是，WHILE 语句是当满足条件时，执行循环内的语句，否则退出循环。WHILE 语句的基本语法形式如下：

```
[label:] WHILE search_condition DO
statement list
END WHILE [label]
```

其中，search_condition 参数表示循环执行的条件，满足该条件时循环执行；statement_list 参数表示循环的执行语句。WHILE 循环需要使用 END WHILE 来结束。

【例 8-12】请使用 WHILE 完成**【例 8-9】**。代码如下：

```
WHILE @count<100 DO
    SET @count=@count+1;
END WHILE;
```