



数据库应用

loop、repeat循环



LOOP循环结构

语法格式：

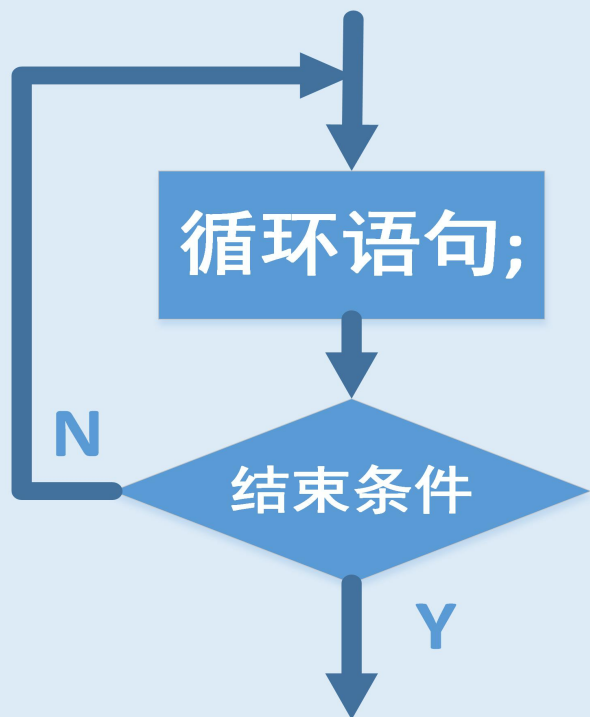
```
[循环语句标签:] LOOP
    循环语句;          #可包含 iterate语句
    IF 结束条件 THEN
        LEAVE [循环语句标签];
    END IF
END LOOP [循环语句标签];
```

功能：当结束条件不成立时，一遍又一遍反复执行循环体，直到结束条件成立为止。



LOOP循环结构

- 流程图:



LOOP执行流程:

①执行1遍循环语句

②求解“结束条件”:

若结果为False, 转第①步,

若为True转第③步;

③结束循环,

执行紧邻的后续语句。



LOOP循环注意事项

- LOOP之前的 “[循环语句标签:]” 和END LOOP 之后的 “[循环语句标签]” 必须一致。
- END LOOP后必须以 “;” 结束。
- “循环语句;” 可以是1条或多条MySQL语句，可包含iterate语句。
- LOOP循环本身没有结束机制，必须使用LEAVE方可终止循环，而LEAVE一般配合IF语句使用。



利用loop循环判断素数

【例1】利用loop循环设计函数IsPrime2判断素数。Loop循环是先执行循环体，再判断条件，对于特殊的素数2，做了单列处理。

注：相对于用WHILE循环判断素数，在第6--9行和第15--18行进行了改动。



利用loop循环判断素数

```
create function IsPrime2( n int ) returns int
```

```
begin
```

```
declare i int default 2;      #i代表从2开始，直到n-1中的一个正整数
```

```
declare yn int default 1;    #先假定yn为 1，代表n是素数
```

```
if n = 2 then                 # 2是素数，没有大于1但小于2的整数
```

```
    return yn ;
```

```
end if ;
```

```
test: LOOP
```

```
    if n % i = 0 then          #i能整除n，反证n不是素数
```

```
        set yn = 0;           #改yn为 0，表示n不是素数
```

```
        leave test;           #结论已提前锁定，不再测试小于n的其它数
```

```
    end if ;
```



利用loop循环判断素数

```
set i = i+1 ;                                #准备下一个用来测试n的正整数
IF i >= n THEN  #测试完毕，结束循环
    LEAVE test ;
END IF ;
END LOOP test ;
return yn ;
end $$
delimiter ;
select IsPrime2( 2 ), IsPrime2( 3 ), IsPrime2( 9 ) ;
```



利用loop循环判断素数

运行结果如下:

```
mysql> delimiter $$
mysql> create function IsPrime2( n int ) returns int
-> begin
->   declare i int default 2 ;           #i代表从2开始, 直到n-1中的一个正整数
->   declare yn int default 1 ;         #先假定yn为 1, 代表n是素数
->   if n = 2 then                       #2是素数, 没有1和2本身以外可以用的检测材料
->     return yn ;
->   end if ;
->   test: LOOP
->     if n % i = 0 then                 #i能整除n, 反证n不是素数
->       set yn = 0 ;                   #改yn为 0, 表示n不是素数
->       leave test ;                   #结论已提前锁定, 不再测试小于n的其它数
->     end if ;
->     set i = i+1 ;                     #准备下一个用来测试n的正整数
->     if i >= n then                     #测试完毕, 结束循环
->       leave test ;
->     end if ;
->   END LOOP test ;
->   return yn ;
-> end $$
```

Query OK, 0 rows affected (0.00 sec)

```
mysql> delimiter ;
mysql> select IsPrime2( 2 ), IsPrime2( 3 ), IsPrime2( 9 ) ;
```

IsPrime2(2)	IsPrime2(3)	IsPrime2(9)
1	1	0



REPEAT循环结构

语法格式：

[循环语句标签:] REPEAT

循环语句；# 可包含ITERATE和LEAVE 语句

UNTIL 结束条件

END REPEAT [循环语句标签]；

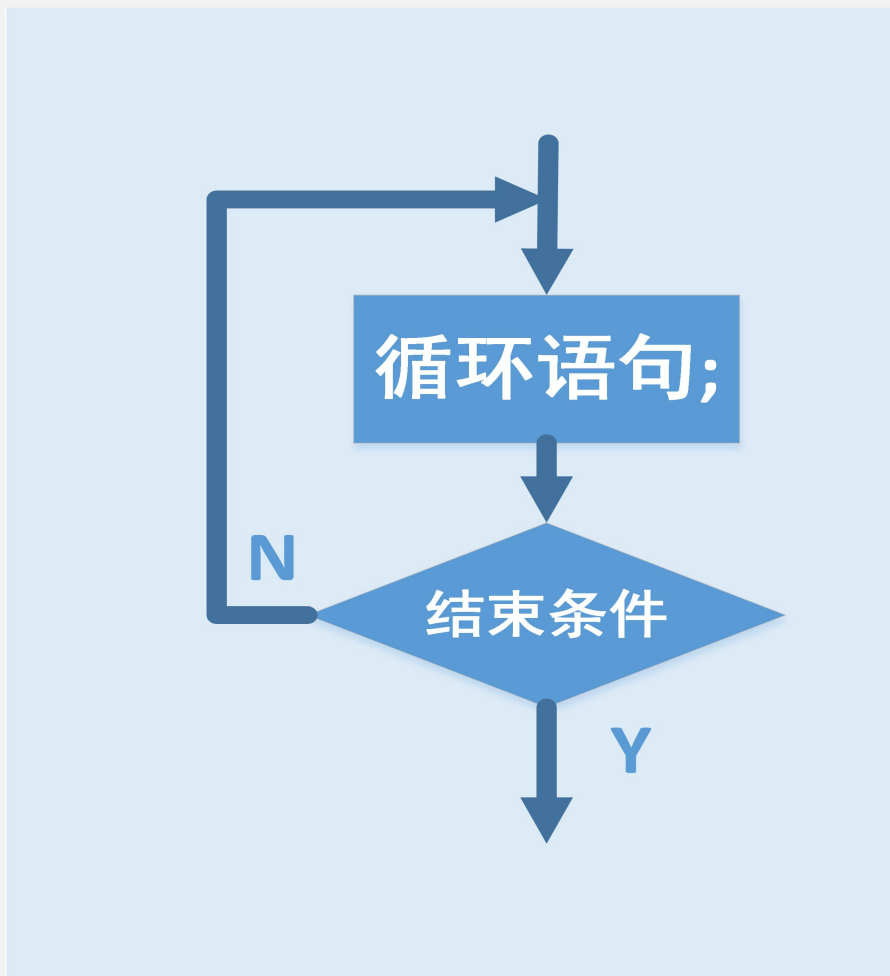
功能：

当结束条件不成立时，一遍又一遍反复执行循环体，直到结束条件成立为止。



REPEAT循环结构

- 流程图：



REPEAT执行流程：

①执行1遍循环语句

②求解“结束条件”：

若结果为False，转第①步，

若为True转第③步；

③结束循环，

执行紧邻的后续语句。



REPEAT循环注意事项

- REPEAT之前的可选项 “[循环语句标签:]” 和 END REPEAT之后的 “[循环语句标签]” 必须一致。
- END REPEAT后必须以 “;” 结束。
- “循环语句;” 可以是1条或多条MySQL语句，可包含 ITERATE和LEAVE语句。
- REPEAT循环本身就带有结束循环的机制，LEAVE语句不是必须的。



用REPEAT循环判断素数

【例2】 请用REPEAT循环设计函数IsPrime3，
判断给定正整数n是不是素数。

本例因REPEAT循环是先执行循环体，再判断条件，
对于特殊的素数2，做了单列处理。



用REPEAT循环判断素数

```
create function IsPrime3( n int ) returns int
```

```
begin
```

```
declare i int default 2;    #i代表从2开始，直到n-1中的一个正整数
```

```
declare yn int default 1;  #先假定yn为1，代表n是素数
```

```
if n = 2 then              #2是素数，没有大于1但小于2的整数
```

```
    return yn;
```

```
end if;
```

```
test: REPEAT
```

```
    if n % i = 0 then
```

```
        set yn = 0;
```

```
        leave test;
```

```
    end if;
```

```
#i能整除n，反证n不是素数
```

```
#改yn为0，表示n不是素数
```

```
#结论已提前锁定，不再测试小于n的其它数
```



用REPEAT循环判断素数

```
set i = i+1 ;  
UNTIL i >= n  
END REPEAT test ;  
return yn ;  
end $$  
delimiter ;  
select IsPrime3( 2 ), IsPrime3( 3 ), IsPrime3( 9 ) ;
```

#准备下一个用来测试n的正整数



用REPEAT循环判断素数

运行结果如下：

```
mysql> delimiter $$
mysql> create function IsPrime3( n int ) returns int
-> begin
->     declare i int default 2 ;
->     declare yn int default 1 ;
->     if n = 2 then
->         return yn ;
->     end if ;
->     test: REPEAT
->         if n % i = 0 then
->             set yn = 0 ;
->             leave test ;
->         end if ;
->         set i = i+1 ;
->     UNTIL i >= n
->     END REPEAT test ;
->     return yn ;
-> end $$
```

#i代表从2开始，直到n-1中的一个正整数
#先假定yn为 1，代表n是素数
2是素数，没有1和2本身以外可以用的检
#i能整除n，反证n不是素数
#改yn为 0，表示n不是素数
#结论已提前锁定，不再测试小于n的其它数
#准备下一个用来测试n的正整数
#测试完毕，结束循环

Query OK, 0 rows affected (0.00 sec)

```
mysql> delimiter ;
mysql> select IsPrime3( 2 ), IsPrime3( 3 ), IsPrime3( 9 ) ;
```

IsPrime3(2)	IsPrime3(3)	IsPrime3(9)
1	1	0



总结与思考

- 本节介绍了MySQL结构家族之LOOP和REPEAT循环，相比While循环，两者都是先执行循环体再判断条件，而且是条件成立时就结束循环。

其中REPEAT循环自带结束机制，相比LOOP循环更加精简、干练。

Loop循环相比REPEAT来说，在于其循环结束点的具体位置比较灵活，可以放到循环体前部或者中间。
- 请思考：如何用循环来处理数据库表里面的数据？