

# SQL中的流程控制

原創、平凡世界 SQL數據庫開發 今天

點擊關注上方“  
設為“置頂或星標

SQL專欄

SQL基礎知識第二版

## 流程控制的定義

一般是指用來控制程序執行和流程分支的指令，一般指的是邏輯計算部分的控制。

## 流程控制種類

常見的流程控制有以下8種

BEGIN ... END	WAITFOR	GOTO
WHILE	IF ... ELSE	BREAK
RETURN	CONTINUE	

下面給大家具體介紹每種流程控制的用法。

## BEGIN...END

BEGIN ... END語句用於將多個T-SQL語句合為一個邏輯塊。當流程控制語句必須執行一個包含兩條或兩條以上的T-SQL語句的語句塊時，使用BEGIN ... END語句。

## 語法

```
BEGIN  
sql_statement...  
END
```

## 示例

我們在數據庫中打印出我們公眾號的名稱"SQL數據庫開發"

```
DECLARE @A VARCHAR(20)  
SET @A='SQL数据库开发'  
BEGIN  
SELECT @A  
END
```

結果如下：

(无列名)
SQL数据库开发

這裡的SELECT @A就是一條被執行的命令語句。

## IF [...ELSE]

IF [...ELSE]表示可以只使用IF，也可以IF和ELSE一起使用，表示條件判斷。當滿足某個條件使，就執行IF 下面的語句，否則執行ELSE下面的語句

## IF語法

```
IF <條件表達式>  
{命令行 | 程序塊}
```

## IF示例

如果某字符串的長度大於5，就打印該字符串

```
DECLARE @A VARCHAR(20)  
SET @A='SQL数据库开发'  
IF LEN(@A)>5  
SELECT @A
```

結果：

(无列名)
SQL数据库开发

這裡結果與上面的BEGIN...END一樣，但是如果我們將條件改成大於8，結果可能就不是這樣的了，小伙伴們可以試一下。

## IF...ELSE語法

```
IF <條件表達式>  
{命令行 | 程序塊}  
ELSE {命令行 | 程序塊}
```

## IF...ELSE示例

如果字符串的長度大於10，就打印該字符串，否則打印"字符串長度太短"

```
DECLARE @A VARCHAR(20)
SET @A='SQL数据库开发'
IF LEN(@A)>10
SELECT @A
ELSE
SELECT '字符串长度太短'
```

結果：

(无列名)

字符串长度太短

很明顯字符串"SQL數據庫開發"長度不大於10，所以返回ELSE裡的結果了。

## WHILE

WHILE是循環控制，當滿足WHILE後面的條件後，就可以循環執行WHILE下面的語句。通常與CONTINUE和BREAK一起使用，Break命令讓程序完全跳出循環語句，結束WHILE命令，CONTINUE是讓命令繼續返回執行

## 語法

```
WHILE <條件表達式>
{命令行 | 程序塊}
CONTINUE
{命令行 | 程序塊}
BREAK
```


{命令行 | 程序塊}

## 示例

有1到10這樣一組數字，從1按順序開始，遇到偶數就跳過，遇到奇數就打印出來，當遇到9就結束打印。

```
DECLARE @i int;
SET @i = 0;
WHILE(@i < 10)
BEGIN
    SET @i = @i + 1;
    IF(@i % 2 = 0)
    BEGIN
        PRINT ('跳过偶数' + CAST(@i AS varchar));
        CONTINUE;
    END
    ELSE IF (@i = 9)
    BEGIN
        PRINT ('到' + CAST(@i AS varchar) + '就结束打印');
        BREAK;
    END
    PRINT @i;
END
```

結果如下：

 消息

```
1  
跳過偶數  
3  
跳過偶數  
5  
跳過偶數  
7  
跳過偶數  
到9就結束打印
```

我們只正常打印出來了1-3-5-7其他的不是跳過就是到9就結束了。

## RETURN

RETURN語句用於使程序從一個查詢、存儲過程或批量處理中無條件返回，其後面的語句不再執行。如果在存儲過程中使用return語句，那麼此語句可以指定返回給調用應用程序、批處理或過程的整數；

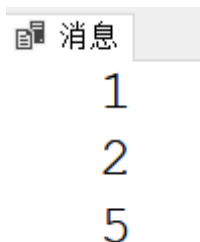
### 語法

```
RETURN [整數表達式]
```

### 示例

```
BEGIN
    PRINT(1);
    PRINT(2);
    RETURN ;
    PRINT(3); -- 在RETURN之后的代码不会被执行，因为会跳过当前批处理
    PRINT(4);
END
GO
BEGIN
    PRINT(5);
END
```

結果如下：



RETURN后面的3-4都没打印，说明在当前批处理的RETURN后都没执行，而新起的BEGIN...END不受上面的RETURN影响，所以打印了5

## GOTO

GOTO命令用来改变程序执行的流程，使程序跳转到标识符指定的程序行再继续往下执行。

GOTO命令虽然增加了程序设计的灵活性，但破坏了程序的结构化，使程序结构变得复杂而且难以测试。

注意：

- 语句标识符可以是数字或者字母的组合，但必须以":"结束。而在GOTO语句后的标识符不必带":"。

- GOTO语句和跳转标签可以在存储过程、批处理或语句块中的任何地方使用，但不能超出批处理的范围。

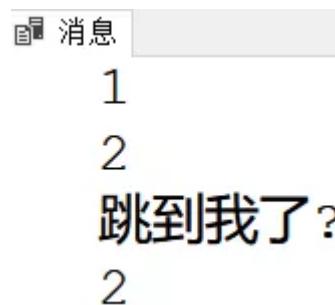
## 语法

GOTO 标识符

## 示例

```
DECLARE @i INT;  
SET @i = 1;  
PRINT @i;  
SET @i = 2;  
PRINT @i;  
GOTO ME;  
SET @i = 3; -- 这行被跳过了  
PRINT @i;  
  
ME:PRINT('跳到了我?');  
PRINT @i
```

结果如下：



消息

1  
2  
跳到了我?  
2

从上面可以看出，当跳到ME的时候，GOTO之前的数有打印，之后的数就跳过了



## WAITFOR

用于挂起语句的执行，直到指定的时间点或者指定的时间间隔。

### 注意：

WAITFOR常用语某个特定的时间点或时间间隔自动执行某些任务。在WAITFOR语句中不能包含打开游标，定义视图这样的操作。在包含事务的语句中不要使用WAITFOR语句，因为WAITFOR语句在时间点或时间间隔执行期间将一直拥有对象的锁，当事务中包含WAITFOR语句，事务的其他语句又需要访问被锁住的数据对象事就容易发生死锁现象。

### 指定时间点的语法

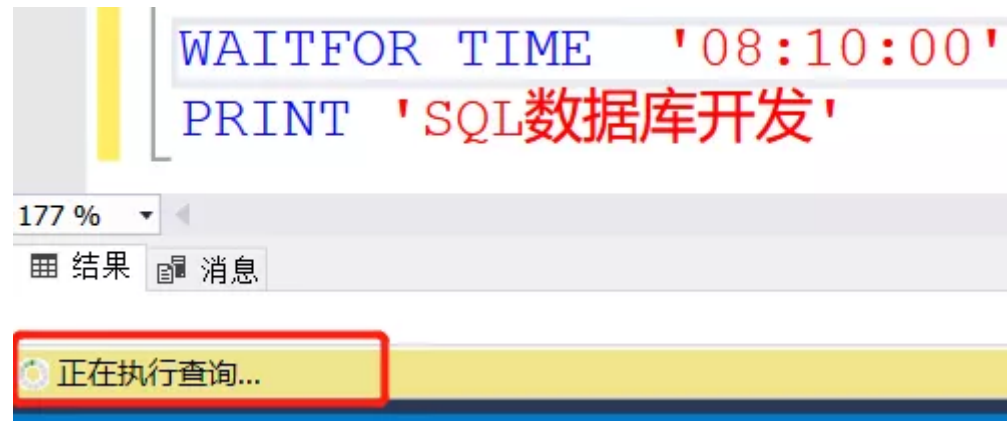
```
WAITFOR TIME <具体时间>
```

### 示例

在'08:10:00'执行打印字符串"SQL数据库开发"

```
WAITFOR TIME '08:10:00'  
PRINT 'SQL数据库开发'
```

如果你执行这句话，那如果在今天这个点之前，那么等到这个时候它就会打印字符串，如果在今天这个点之后，那你需要等到第二天的这个时间点才会打印。在未执行之前查询窗口是一直"正在执行查询..."状态



### 指定等待时间间隔的语法

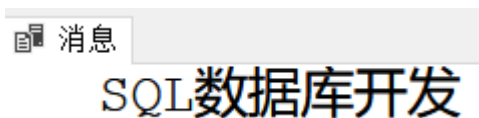
```
WAITFOR DELAY 'INTERVAR'
```

INTERVAR为时间间隔，指定执行WAITFOR 语句之前需要等待的时间，最多为24小时。

### 示例

```
WAITFOR DELAY '00:00:03'  
PRINT 'SQL数据库开发'
```

在等到3秒钟后，会打印出字符串



### 总结

流程控制是SQL开发中经常需要使用到的，特别是条件判断IF...ELSE，循环执行WHILE是经常使用的，对于想在SQL开发中有所提高的同学，务必要掌握这几个流程控制的用法。



最后给大家分享我写的SQL两件套：《SQL基础知识第二版》和《SQL高级知识第二版》的PDF电子版。里面有各个语法的解释、大量的实例讲解和批注等等，非常通俗易懂，方便大家跟着一起来实操。

有需要的读者可以下载学习，在下面的公众号「数据前线」(非本号)后台回复关键字：**SQL**，就行

**数据前线**



后台回复关键字：**1024**，获取一份精心整理的技术干货

后台回复关键字：**进群**，带你进入高手如云的交流群。

## 推荐阅读

- 知乎：国家何时整治程序员的高薪现象？

- SQL触发器实战
- 一款SQL自动检查神器，再也不用担心SQL出错了！
- SQL如何求解连续年份的问题？
- SQL 语法速成手册

喜欢此内容的人还喜欢

知乎：国家何时整治程序员的高效现象？

SQL数据库开发

互联网

国家何时整治程序员  
相同的应届毕业生，程序员相  
有程序员高薪认知。对于这种

关注问题

写回答

如何将一根意面折成两段？费曼也没整明白 | 科学60秒

科研圈



蔡廷鍇跑了、陳浴新叛變，戰場成了“血肉磨坊”……

瞭望智庫

