宿州职业技术学院

Suzhou Vocational and Technical College

SQL Server 数据库技术与应用

流程控制

流程控制的概念

- ☑ 流程控制主要用在语句块、批处理、用户自定义函数、存储过程与触发器的编程中，用来控制程序执行的顺序，为结构化程序设计提供支持。

常用的流程控制

流程控制结构	关键字		流程控制语句	关键字
复合语句结构	BEGIN…END		无条件返回语句	RETURN
条件判断结构	IF…ELSE		继续循环语句	CONTINUE
多重分支结构	CASE		中止循环语句	BREAK
循环结构	WHILE		无条件转移语句	GOTO
异常捕捉与处理结构	TRY…CATCH		延迟执行语句	WAITFOR

复合语句

- ☑ 复合语句 Begin…End结构用来将一组封装成一个语句块，并将这些语句作为整体进行执行与处理。
- ☑ Begin…End结构的语法格式如下：

```
BEGIN
   {T-SQL语句块}
END[;]
```

流程控制

条件判断语句

- ☑ 当需要根据不同的条件执行不同的操作与运算时，就要使用条件判断结构IF…ELSE。
- ☑ IF…ELSE结构的语法格式如下：

IF <布尔表达式>
　　{ <语句> | <语句块> }
[ELSE
　　{ <语句> | <语句块> }] [;]

流程控制

1 流程控制概述

2 条件判断语句

3 多重分支语句

4 循环语句

5 其他语句

多重分支结构

- ☑ 多重分支结构能够根据表达式的不同取值，转向不同的计算或处理。
- ☑ CASE结构提供了比IF…ELSE结构更多的判断与选择，能够方便地实现多重分支选择，从而替代嵌套的条件判断结构。CASE结构也称为CASE函数。

多重分支结构分类

☑ 根据语法结构的不同，CASE结构可分为以下两种应用格式：

① 简单CASE结构；

② 搜索CASE结构。

简单CASE结构

☑ 简单CASE结构的语法格式如下：

```
CASE <表达式>
WHEN <匹配表达式值1> THEN <结果表达式1>
WHEN <匹配表达式值2> THEN <结果表达式2>
     ……
WHEN <匹配表达式值n> THEN <结果表达式n>
     [ ELSE <结果表达式n+1> ]
END [;]
```

搜索CASE结构

☑ 搜索CASE结构的语法格式如下：

```
CASE
WHEN <布尔表达式1> THEN <结果表达式1>
WHEN <布尔表达式2> THEN <结果表达式2>
     ……
WHEN <布尔表达式n> THEN <结果表达式n>
     [ ELSE <结果表达式n+1> ]
END [;]
```

流程控制

1. 流程控制概述
2. 条件判断语句
3. 多重分支语句
4. 循环语句
5. 其他语句

循环语句

- ☑ WHILE循环结构用来根据特定的条件决定是否重复执行某一语句或语句块。
- ☑ 循环结构的语法格式如下：

WHILE <布尔表达式>
{ <语句> | <语句块> } [;]

循环语句的其他要点

- ☑ 循环结构允许嵌套使用，从而构造出更为复杂的多重循环结构。
- ☑ 当循环体为语句块时，可以使用CONTINUE或BREAK语句控制循环的流程。
- ☑ CONTINUE语句结束当前的一次循环，转到循环结构的条件判断位置，根据当前布尔表达式的值来决定是否执行下一轮循环。
- ☑ BREAK语句结束当前的这一层循环，跳到循环结构之外继续执行其他语句。当BREAK语句位于多层嵌套循环结构中时，它只能退出其所在的那一层循环，回到相邻的外层循环中，而不会跳出整个嵌套循环结构，除非BREAK语句恰好位于最外层的循环结构中。

流程控制

1. 流程控制概述
2. 条件判断语句
3. 多重分支语句
4. 循环语句
5. 其他语句

流程控制的其他语句

☑ 流程控制还包含以下其他的语句：

① 异常捕捉与处理语句；

② 无条件返回语句；

③ 无条件转移语句；

④ 延迟执行语句。

异常处理结构

- ☑ TRY…CATCH结构类似于其他高级编程语言的异常处理结构，当TRY子句中的代码执行时出现错误或异常时，系统将会把空值传递给CATCH子句去处理。
- ☑ 异常捕捉与处理结构的语法格式如下：

```
BEGIN TRY
    { <语句>|<语句块> }
END TRY
BEGIN CATCH
    { <语句>|<语句块> }   --该处的语句或语句块用以处理错误或异常
END CATCH[;]
```

无条件返回语句

- ☑ RETURN语句的作用是从语句块、批处理、存储过程或触发器中无条件地退出并终止RETURN语句所在的程序模块，从而使RETURN语句后面的任何语句都不再被执行。
- ☑ RETURN语句的语法格式如下：

RETURN［〈整型表达式〉］

无条件转移语句

- ☑ 无条件转移语句GOTO必须与位置标识符配合使用。
- ☑ GOTO语句用来控制程序直接标识符所标识的位置处继续执行，而忽略位于GOTO语句和标识符之间的程序段。
- ☑ GOTO语句通常用于语句块、批处理和存储过程中。

GOTO语句语法

☑ GOTO语句的语法形式如下：

GOTO <位置标识符>

…… /* 将被忽略的程序段 */

<位置标识符>:

…… /* 流程将跳转到并被执行的程序段 */。

延迟执行语句

- ☑ WAITFOR语句用于暂时停止执行语句块、批处理、事务或存储过程等程序模块，直到所设定的等待时间已经到达，才会从暂停处置继续执行。
- ☑ WAITFOR语句的语法形式为：

```
WAITFOR { DELAY <等待的时间间隔> |
    TIME <运行的时刻> }
```

下一章再见！