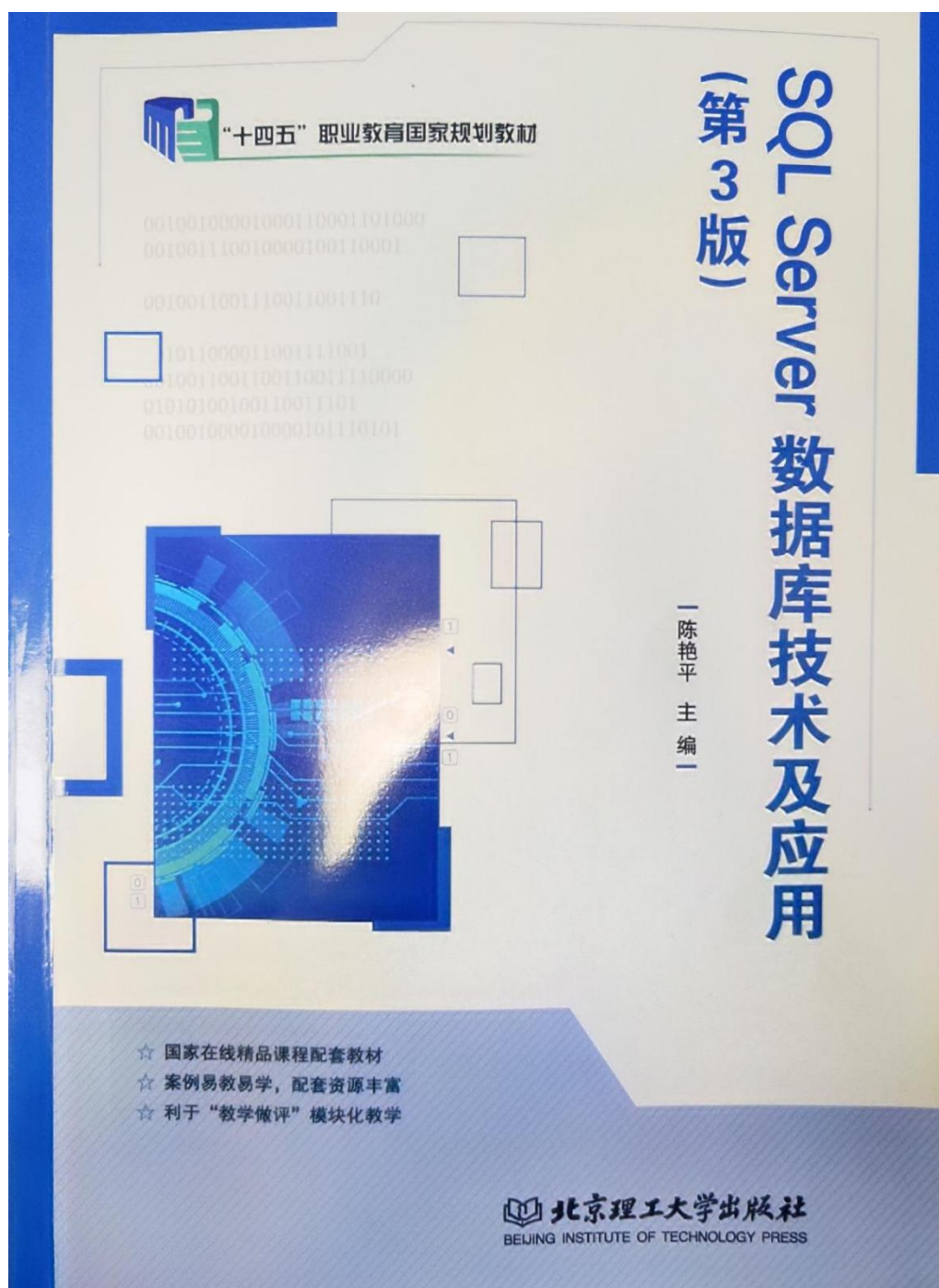


计算机应用技术专业教师岗位试讲内容

教学内容：模块 8 T-SQL 编程与应用，任务 3 认识流程控制语句

教学重点：IF……ELSE……、WHILE、CASE 语句的控制流程和用法。

参考教材：教材名称《SQL Server 数据库技术及应用》，北京理工大学出版社，2021 年 10 月第 3 版，陈艳平主编。



8.3 任务3: 认识流程控制语句

任务目标

- 理解流程控制语句的作用。
- 熟练掌握 IF...ELSE...、WHILE、CASE 语句的控制流程和用法。
- 熟练掌握 GOTO、WAITFOR 和 RETURN 语句的用法。

流程控制语句是指用来控制程序执行和流程分支的语句，在 SQL Server 中，流程控制语句主要用来控制 T-SQL 语句、语句块或者存储过程的执行流程。

8.3.1 顺序语句

顾名思义，顺序语句实际上是指顺序结构控制的语句，就是指按照语句在程序中书写的先后次序逐条顺次执行。上述操作运算语句即顺序语句，包括表达式语句、输入/输出语句等。下面为读者介绍几个顺序语句。

1. SET 语句

SET 语句有两种用法，除了用于给局部变量赋值之外，还可以用于设定用户执行 T-SQL 命令时 SQL Server 的处理选项，一般有以下几种设定方式：

- (1) SET 选项 ON：选项开关打开；
- (2) SET 选项 OFF：选项开关关闭；
- (3) SET 选项值：设定选项的具体值。

SET 语句常用的设置选项见表 8-8。

表 8-8 SET 语句常用的设置选项

序号	选项	说明
1	XACT_ABORT	语法：SET XACT_ABORT { ON OFF } 功能：当 SET XACT_ABORT 为 ON 时，如果 T-SQL 语句产生运行时错误，整个事务将终止并回滚。为 OFF 时，只回滚产生错误的 T-SQL 语句，而事务将继续进行处理。编译错误（如语法错误）不受 SET XACT_ABORT 的影响。
2	ROWCOUNT	语法：SET ROWCOUNT { number @number_var } 功能：使 SQL Server 在返回指定的行数之后停止处理查询。number @number_var 是在停止给定查询之前要处理的行数（整数）。

续表

序号	选项	说明
3	NOCOUNT	<p>语法: SET NOCOUNT { ON OFF }</p> <p>功能: 使返回的结果中不包含有关受 T-SQL 语句影响的行数的信息。当 SET NOCOUNT 为 ON 时, 不返回计数 (表示受 T-SQL 语句影响的行数)。当 SET NOCOUNT 为 OFF 时, 返回计数</p>
4	DEADLOCK_PRIORITY	<p>语法: SET DEADLOCK_PRIORITY { LOW NORMAL @ deadlock_var }</p> <p>功能: 控制在发生死锁情况时会话的反应方式。LOW 指定当前会话为首选死锁牺牲品, SQL Server 自动回滚死锁牺牲品的事务, 并给客户端应用程序返回 1205 号死锁错误信息。NORMAL 指定会话返回到默认的死锁处理方法。@ deadlock_var 是指定死锁处理方法的字符变量。如果指定 LOW, 则 @ deadlock_var 为 3; 如果指定 NORMAL, 则 @ deadlock_var 为 6</p>
5	LOCK_TIMEOUT	<p>语法: SET LOCK_TIMEOUT timeout_period</p> <p>功能: 指定语句等待锁释放的毫秒数。timeout_period 是在 SQL Server 返回锁定错误前经过的毫秒数。值为 -1 (默认值) 时表示没有超时期限 (即无限期等待)。当锁等待超过超时值时, 将返回错误。值为 0 时表示根本不等待, 并且一遇到锁就返回信息。可以使用全局变量 @@ LOCK_TIMEOUT 返回当前会话的当前锁超时设置, 单位为毫秒</p>

2. SELECT 语句

SELECT 语句除了作为查询语句以外, 也可以作为输出语句使用, 它作为输出语句时的语法如下:

```
SELECT 表达式 1, 表达式 2, ..., 表达式 n
```

3. PRINT 语句

PRINT 语句是专门的输出语句, 其语法格式如下:

```
PRINT 表达式
```

说明: SELECT 语句和 PRINT 语句都可以作为输出语句使用, 但两者是有区别的:

- (1) SELECT 语句可以输出任意的 SQL Server 数据类型、字段、标量函数等。
- (2) PRINT 语句只能用于输出字符型数据, 如: 字符串或 Unicode 字符串常量、任何有

效的字符数据类型 (char 或 varchar) 的变量、能够隐式转换为这些数据类型的变量或者返回值为字符串的表达式。PRINT 语句输出的字符串最大长度为 8 000 个字符, 超过该值以后的任何字符会被截断。

8.3.2 IF...ELSE 语句

在介绍流程控制语句之前, 首先介绍一下什么是语句块。一个语句块以 BEGIN 语句开始, 以 END 语句作为终止, 中间可以包含若干条 T-SQL 语句。语句块在流程控制语句之中是作为一个整体的逻辑单元一起执行的, 语法格式如下:

```
BEGIN
    语句 1
    语句 2
    ...
    语句 n
END
```

IF...ELSE 语句是条件判断语句, 其中, ELSE 子句是可选的, 最简单的 IF 语句没有 ELSE 子句部分。IF...ELSE 语句用来判断当某一条件成立时执行某段程序, 条件不成立时执行另一段程序。SQL Server 允许嵌套使用 IF...ELSE 语句, 而且嵌套层数没有限制。IF...ELSE 语句的语法格式如下:

```
IF <逻辑表达式>
    语句或语句块           -- 逻辑表达式为真时执行
[ ELSE
    语句或语句块 ]       -- ELSE 子句为可选
                          -- 逻辑表达式为假时执行
```

若有多条语句, 则使用语句块, 其语法格式如下:

```
IF <逻辑表达式>
    BEGIN
        语句 1
        语句 2
        ...
    END
[ ELSE ...]
```

【例 8-15】 使用 IF...ELSE 语句判断学生成绩表“SC”中, “C 语言程序设计”这门课程有没有成绩高于 90 分的学生, 如果有则统计成绩高于 90 分的学生人数, 如果没有则打印“成绩不理想!”。

```

USE EDUC
GO
DECLARE @no int
IF EXISTS( SELECT * FROM SC INNER JOIN Course
           ON SC.CID = Course.CID
           WHERE Cname = 'C 语言程序设计' AND Score >=90)
BEGIN
    SELECT @no = COUNT( * ) FROM SC INNER JOIN Course
           ON SC.CID = Course.CID
           WHERE Cname = 'C 语言程序设计' AND Score >=90
    PRINT "C 语言程序设计"高于 90 分的学生人数为: ' + CAST(@no AS char(2))
END
ELSE
    PRINT '成绩不理想!'

```

执行效果如图 8-18 所示。

消息
"C语言程序设计"高于90分的学生人数为:1

图 8-18 “C 语言程序设计”课程成绩在 90 分以上的学生人数统计

8.3.3 WHILE 语句

WHILE 语句用于设置重复执行 T-SQL 语句或语句块的条件。只要指定的条件为真，就重复执行语句，可以使用 BREAK 和 CONTINUE 关键字在循环内部控制 WHILE 循环中语句的执行。其语法格式如下：

```

WHILE <逻辑表达式> -- 逻辑表达式为真时循环
BEGIN
    语句或语句块
    [BREAK] -- 终止整个循环语句的执行
    [CONTINUE] -- 终止本次循环体的执行,继续下一次循环
END

```



WHILE 语句

其中，CONTINUE 语句可以使程序跳过 CONTINUE 语句后面的语句，回到 WHILE 循环的第一行命令，继续下一次循环。BREAK 语句则使程序完全跳出循环，结束 WHILE 语句的执行。

说明：WHILE 循环可以嵌套，如果嵌套了两个或多个 WHILE 循环，则内层的 BREAK 语句将退出到下一个外层循环，将首先运行内层循环结束之后的所有语句，然后重新开始下

一个外层循环。

【例 8-16】 使用 WHILE 语句对图书价格进行调整, 如果图书平均价格低于 40 元, 则循环每次使每本图书价格增加 1 元, 直到所有图书的平均价格大于 40 元或者有的图书价格超过 100 元为止。代码如下:

```
USE Library
GO
-- 根据是否存在图书的平均价格低于 40 元而确定循环是否继续执行
WHILE (SELECT AVG(Price) FROM Book) < 40
BEGIN
    UPDATE Book Set Price = Price + 1
    -- 如果有的图书的价格已经超过了 100 元, 则跳出循环
    IF (SELECT MAX(Price) FROM Book) > 100
        BREAK
    ELSE
        CONTINUE
END
-- 循环结束
```

执行前的图书信息表“Book”如图 8-19 所示。

BID	Bname	Author	PubComp	PubDate	Price	ISBN	Class
G448-01	教育心理学	斯莱文	人民邮电出版社	2011-04-01	78.0000	7-115-24710-2	教育类
G455-01	大学生创新能力开发与应用	何静	同济大学出版社	2011-04-06	32.0000	7-560-84516-0	教育类
TP311-011	Java 信息系统设计与开发实例	黄明	机械工业出版社	2004-01-01	22.0000	7-111-14186-5	计算机类
TP311-012	Java 信息管理系统开发	求是科技	人民邮电出版社	2005-04-01	34.0000	7-115-13214-3	计算机类
TP311-051	软件工程	张海藩	人民邮电出版社	2003-07-01	27.0000	7-115-11258-4	计算机类
TP311-052	软件工程案例开发与实践	刘竹林	清华大学出版社	2009-08-01	29.0000	7-81123-508-1	计算机类
TP392-01	数据库系统概论	萨师煊	高等教育出版社	2000-08-01	25.0000	7-04-007494-1	计算机类
TP392-02	数据库应用技术 (SQL Server 2005)	周慧	人民邮电出版社	2009-03-01	29.0000	7-115-19345-2	计算机类
TP392-03	SQL Server 2005 应用教程	梁庆枫	北京大学出版社	2010-08-01	25.0000	7-301-17605-4	计算机类
TP945-08	计算机组装与维护	孙中胜	中国铁道出版社	2003-07-01	24.0000	7-113-05287-8	计算机类

图 8-19 图书价格更新前

执行后的图书信息表“Book”如图 8-20 所示。

8.3.4 其他控制语句

1. WAITFOR 语句

WAITFOR 语句用于暂时停止执行 T-SQL 语句、语句块或者存储过程等, 直到所设置的时间已过或者所设定的时间已到才继续执行, 其语法格式如下:

```
WAITFOR { DELAY '延迟时间' | TIME '到达时间' }
```

BID	Bname	Author	PubComp	PubDate	Price	ISBN	Class
G448-01	教育心理学	斯莱文	人民邮电出版社	2011-04-01	86.0000	7-115-24710-2	教育类
G455-01	大学生创新能力开发与应用	何静	同济大学出版社	2011-04-06	40.0000	7-560-84516-0	教育类
TP311-011	Java信息系统设计与开发实例	黄明	机械工业出版社	2004-01-01	30.0000	7-111-14186-5	计算机类
TP311-012	Java信息管理系统开发	求是科技	人民邮电出版社	2005-04-01	42.0000	7-115-13214-3	计算机类
TP311-051	软件工程	张海藩	人民邮电出版社	2003-07-01	35.0000	7-115-11258-4	计算机类
TP311-052	软件工程案例开发与实践	刘竹林	清华大学出版社	2009-08-01	37.0000	7-81123-508-1	计算机类
TP392-01	数据库系统概论	萨师煊	高等教育出版社	2000-08-01	33.0000	7-04-007494-1	计算机类
TP392-02	数据库应用技术 (SQL Server 2005)	周慧	人民邮电出版社	2009-03-01	37.0000	7-115-19345-2	计算机类
TP392-03	SQL Server 2005应用教程	梁庆枫	北京大学出版社	2010-08-01	33.0000	7-301-17605-4	计算机类
TP945-08	计算机组装与维护	孙中胜	中国铁道出版社	2003-07-01	32.0000	7-113-05287-8	计算机类

图 8-20 图书价格更新后

其中, DELAY 用于指定时间间隔, 即可以继续执行批处理、存储过程或事务之前必须经过的指定时段, 最长可为 24 小时。TIME 用于指定某一时刻, 即运行批处理、存储过程或事务的时间。'延迟时间' 和 '到达时间' 必须为日期型数据, 可以为 datetime 数据可接受的格式之一。

【例 8-17】 WAITFOR 语句实验。代码如下:

```
USE EDUC
GO
WAITFOR DELAY '00:00:03' -- 指定在执行 SELECT 语句之前需要等待的秒数
SELECT * FROM Student WHERE Specialty = '软件技术' -- 到达等待时间之后执行查询
```

执行结果请读者上机观察。

2. GOTO 语句

GOTO 语句可以使程序直接跳到指定的标有标识符的位置处继续执行, 而位于 GOTO 语句和标识符之间的程序将不会被执行。GOTO 语句和标识符可以用在语句块、批处理和存储过程中, 标识符可以为数字与字符的组合, 但必须以“:”结尾, 如“a1:”。在 GOTO 语句行, 标识符后面不用跟“:”。GOTO 语句的语法格式如下:

```
GOTO label(标识符名称)
...
label:
```

【例 8-18】 GOTO 语句实验: 利用 GOTO 语句求出从 1 加到 5 的总和。代码如下:

```
DECLARE @sum int,@count int
SELECT @sum=0,@count=1
label_1: -- 定义标识符
SET @sum=@sum+@count
SET @count=@count+1
```

```

IF @count <=5
GOTO label_1          --GOTO 语句跳转到 label_1 处执行
SELECT @count,@sum
    
```

执行结果如图 8-21 所示。

说明：在使用中应尽可能避免使用 GOTO 语句，因为过多的 GOTO 语句可能会造成 T-SQL 语句的逻辑混乱而难以理解，影响程序的可读性。另外，标签仅标示了跳转的目标，它并不隔离其前后的语句。只要标签前面的语句本身不是流程控制语句，标签前后的语句将按照顺序正常执行，就如同没有使用标签一样。

结果	消息
(无列名)	(无列名)
1	6 15

图 8-21 GOTO 语句的应用

3. RETURN 语句

RETURN 语句用于结束当前程序的执行，返回到上一个调用它的程序或其他程序，其语法格式如下：

```
RETURN 整数值或变量
```

说明：RETURN 语句要指定返回值，如果没有指定返回值，SQL Server 系统会根据程序执行的结果返回一个内定值。RETURN 语句的执行是即时且完全的，可在任何时候用于从过程、批处理或语句块中退出，RETURN 语句之后的语句将不会被执行。如果用于存储过程，则 RETURN 不能返回空值。具体 RETURN 语句在存储过程中的应用可以参见模块 9。

8.3.5 CASE() 函数

CASE() 函数可以计算多个条件式，并将其中一个符合条件的结果表达式返回，相当于多重 IF 语句及 IF 语句的嵌套，但是条理更加清晰。CASE() 函数按照使用形式的不同，可以分为简单 CASE() 函数和搜索 CASE() 函数，两者可以实现相同的功能。简单 CASE() 函数的写法相对比较简洁，但是和搜索 CASE() 函数相比，在功能方面会有些限制。



CASE() 函数

1. 简单 CASE() 函数

其语法格式如下：

```

CASE <输入表达式 >
    WHEN <表达式的值 1 > THEN 结果 1
    WHEN <表达式的值 2 > THEN 结果 2
    ...
    [ ELSE 其他结果 ]
END
    
```


功能：计算 CASE 输入表达式，将其值按指定顺序与 WHEN 表达式的值进行比较运算，当 CASE 输入表达式的值等于 WHEN 表达式的值时，函数得到第一个满足条件的 THEN 返回的结果值。如果比较运算结果都不为真，则函数返回 ELSE 后的表达式的值，如果省略此参数并且比较运算的计算结果都不为真，函数将返回 NULL。

【例 8-19】 简单 CASE() 函数举例。根据学生成绩表“SC”中的学生成绩，显示对应的五级制中的等级。代码如下：

```
USE EDUC
GO
SELECT SID,CID,Score,等级 =
CASE CAST(Score/10 AS int)
    WHEN 10 THEN 'A'
    WHEN 9 THEN 'A'
    WHEN 8 THEN 'B'
    WHEN 7 THEN 'C'
    WHEN 6 THEN 'D'
    ELSE 'E'
END
FROM SC
```

执行结果如图 8-22 所示。

2. 搜索 CASE() 函数

其语法格式如下：

```
CASE
    WHEN 条件 1 THEN 结果 1
    WHEN 条件 2 THEN 结果 2
    ...
    [ELSE 其他结果]
END
```

功能：按指定顺序对每个 WHEN 子句求逻辑表达式的值，当计算结果为真时，函数得到第一个满足条件的 THEN 返回的结果值。如果运算结果都不为真，则函数返回 ELSE 后的结果值。如果省略 ELSE 并且计算结果都不为真，函数将返回 NULL。

说明：搜索 CASE() 函数语法格式中所计算的逻辑表达式可以是任何有效的布尔表达式。

【例 8-20】 搜索 CASE() 函数举例。根据学生成绩表“SC”中的学生成绩，显示对应的五级制中的等级。代码如下：

```

USE EDUC
GO
SELECT SID, CID, Score, 等级 =
CASE
    WHEN Score >=90 THEN 'A'
    WHEN Score BETWEEN 80 AND 89 THEN 'B'
    WHEN Score BETWEEN 70 AND 79 THEN 'C'
    WHEN Score BETWEEN 60 AND 69 THEN 'D'
    ELSE 'E'
END
FROM SC

```

执行结果同例 8-19, 如图 8-22 所示。

	SID	CID	Score	等级
1	2020059999	16020010	96	A
2	2020059999	16020011	80	B
3	2020051002	16020012	78	C
4	2020051002	16020013	87	B
5	2020051002	16020014	85	B
6	2020051003	16020014	89	B
7	2020051003	16020015	90	A
8	2020051202	16020010	67	D
9	2020051002	16020014	89	B

图 8-22 成绩从百分制转换成五级制

说明: CASE() 函数只返回第一个符合条件的值, 剩下的 CASE 部分将被自动忽略。

8.4 任务 4: 认识批处理

任务目标

- 理解批处理的概念及用途。
- 掌握批处理的用法。

批处理是 T-SQL 语句集合的逻辑单元。批处理中的所有语句被整合成一个执行计划。一个批处理内的所有语句要么被放在一起通过解析并执行, 要么一句都不执行。

1. 批处理的概念

批处理是指包含一条或多条 T-SQL 语句的语句组, 这组语句从应用程序一次性地发送到 SQL Server 服务器执行, SQL Server 服务器将批处理语句编译成一个执行单元, 然后作为一个整体来执行。



认识批处理