

国家“十三五”重点出版规划项目
上海高校优秀教材奖获得者主编

上海市高校精品课程
特色教材(立体化新形态)

数据库原理及应用

——基于SQL Server 2016

主编 贾铁军 徐方勤
副主编 邓红霞 戴春妮 王佃来 曹锐
参编 陶维天 张野 降爱莲

 机械工业出版社
CHINA MACHINE PRESS

第7章 T-SQL应用编程



第7章 T-SQL编程基础

微软的Transact-SQL简称**T-SQL**，是微软在其数据库管理软件 SQL Server上**对标准SQL进行的一种扩展，也是标准的SQL程序设计语言的增强版**。几乎所有的应用程序都要读取、存储和处理关系数据库中的数据。只要使用 Microsoft SQL Server，就需要了解及学习使用T-SQL，这是Microsoft为ANSI标准SQL数据库查询语言提供的强大的实现方案。

第7章 目录

7.1 T- SQL基础概述

7.2 批处理、脚本及事务

7.3 流程控制结构

*7.4 嵌入式SQL概述

7.5 实验七 流程控制语句

7.6 本章小结

7.7 练习与实践七(注:网上作业)



超链接



教学目标

- 熟悉T-SQL的概念、特点、执行方式 **重点**
- 理解批处理、脚本、事务等概念 **重点**
- 掌握编程中常用流程控制结构与语句 **重点**
- 理解嵌入式SQL的概念、语法及用法

友情
提示



同步实验(上机)



7.1 T-SQL基础概述

案例7-1

T-SQL是对标准SQL的扩展。 ANSI (American National Standards Institute, 美国国家标准协会) 制定的标准SQL是关系数据库中管理和操作数据的标准语言, 具有操作统一、功能丰富, 非过程化等优点。但是这种高度非过程化的语言也有着显著缺点: 缺少流程控制能力。因此, 各大厂商都对标准SQL进行了扩展, 制定了各自的过程化SQL, 例如Microsoft SQL Server的Transact-SQL、Oracle的PL/SQL、IBM DB2的SQL PL和Kingbase的PL/SQL。



7.1 T-SQL基础概述

7.1.1 T-SQL的概念、特点和功能

1. T-SQL的概念及优点
2. SQL与T-SQL的区别与联系
3. T-SQL的功能

7.1.2 T-SQL的类型和执行方式

1. T-SQL的类型
2. T-SQL的执行方式

7.1 T-SQL基础概述

7.1.1 T-SQL的概念、特点和功能

■ 1. T-SQL的概念及优点

作为**标准SQL的扩展**，T-SQL中不仅包含数据定义语言、数据操纵语言和数据控制语言，同时还包括了**程序控制语言**，而且提供了丰富的编程功能，允许使用变量、运算符、表达式、函数、流程控制语句等，其功能非常强大。

T-SQL提供的交互式查询语言，**既允许用户直接查询存储在数据库中的数据，也可以将语句嵌入到某种高级程序设计语言中使用**，例如可嵌入到C#、.Net或Java等语言中。对于数据库中数据集的操作来说，T-SQL比其它高级语言更加简单高效，而且也具有了其它高级语言的特点，因此得到广泛的应用。



7.1 T-SQL基础概述

7.1.1 T-SQL的概念、特点和功能

■ 1. T-SQL的概念及优点

T-SQL集数据查询、数据操纵、数据定义和数据控制于一体，体现了关系数据库语言的特点和优点。主要具备以下几个特点：

- 1) 功能强大，将数据定义、数据操纵、数据控制、事务管理和附加语言元素集于一体。
- 2) 两种使用方式，可直接交互式操作数据库，也可以嵌入高级语言中使用。
- 3) 与人的思维方式接近，简单易学，易于理解和掌握。

7.1 T-SQL基础概述

7.1.1 T-SQL的概念、特点和功能

■ 2. SQL与T-SQL的区别与联系

T-SQL作为标准SQL的扩展，包含了许多SQL所不具备的编程功能。

例如，SQL Server查询分析器中使用了语句：

```
Select * From 表1
```

这样的语句到底是SQL还是T-SQL？

但是比如输入的是语句：

```
BEGIN
```

```
PRINT "Hello World"
```

```
END
```

既是SQL，也是
T-SQL，T-SQL
就包含了标准
SQL

不是SQL，只是T-
SQL，包含了流程
控制语句



7.1 T-SQL基础概述

7.1.1 T-SQL的概念、特点和功能

■ 3. T-SQL的功能

T-SQL的编程功能主要包括基本功能和扩展功能。

(1) 基本功能

根据T-SQL的功能特点，可将其基本功能概括为：数据定义语言功能、数据操纵语言功能、数据控制语言功能和事务管理语言功能等。

(2) 扩展功能

T-SQL的扩展功能主要包括：程序流程控制结构以及T-SQL附加的语言元素，包括标识符、局部变量、系统变量、常量、运算符、表达式、数据类型、函数、错误处理语言和注释等。

7.1 T-SQL基础概述

7.1.2 T-SQL的类型和执行方式

■ 1. T-SQL的类型

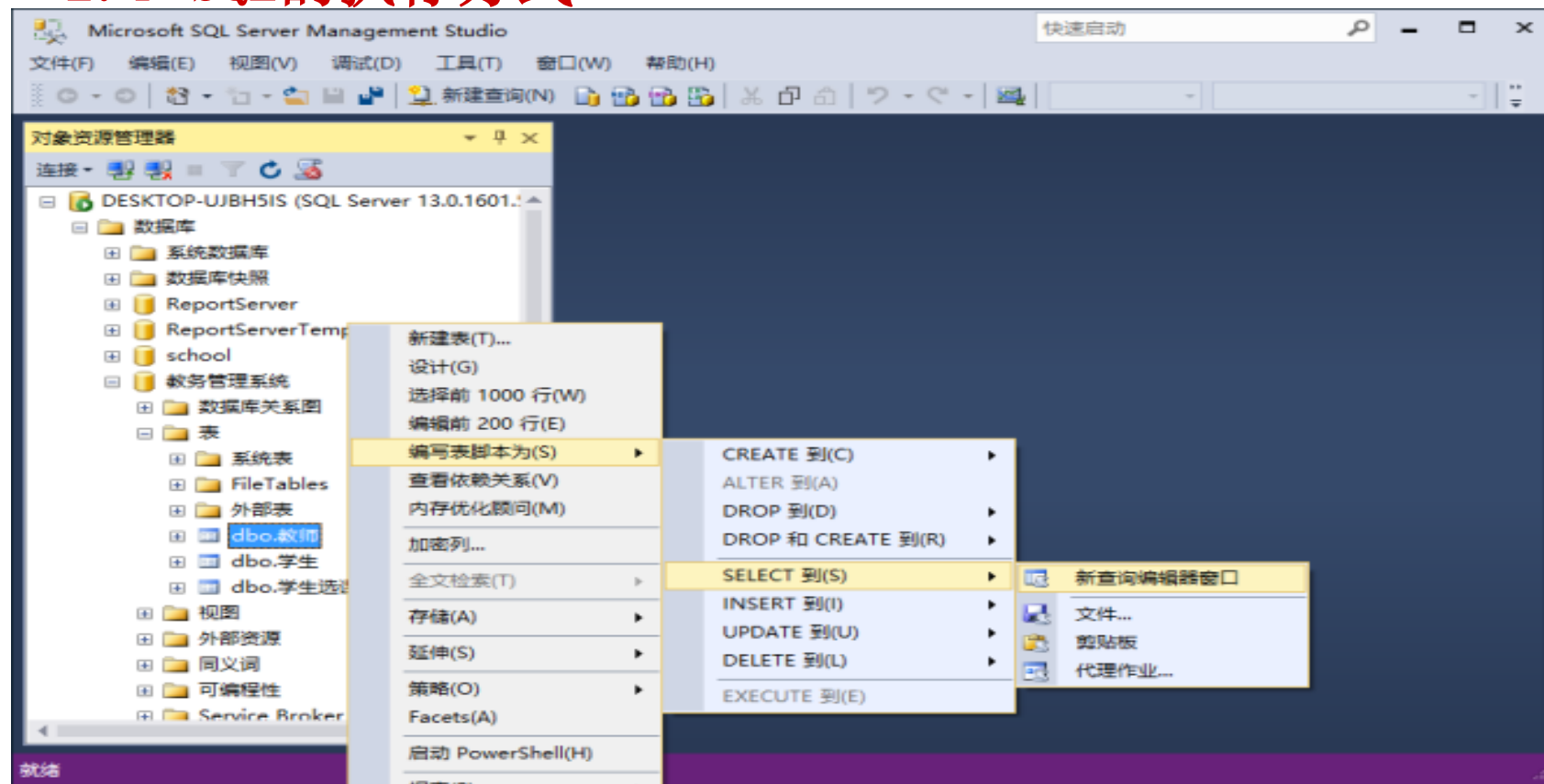
根据SQL Server数据库管理系统具有的功能，**T-SQL语言可分为5种类型**：

- 数据定义语言（CREATE、ALTER、DROP）
- 数据操纵语言（SELECT、INSERT、DELETE、UPDATE）
- 数据控制语言（GRANT、REVOKE、DENY）
- 事务管理语言（COMMIT、ROLLBACK）
- 附加的语言元素（标识符、变量、常量、运算符、表达式、数据类型、函数、流程控制、注释等）

7.1 T-SQL基础概述

7.1.2 T-SQL的类型和执行方式

2. T-SQL的执行方式



7.1 T-SQL基础概述

7.1.2 T-SQL的类型和执行方式

2. T-SQL的执行方式

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The left pane shows the '对象资源管理器' (Object Explorer) with the '教务管理系统' (Teaching Management System) database selected. The right pane shows the 'SQLQuery1.sql' window with the following T-SQL query:

```
USE [教务管理系统]
GO

SELECT [工号]
, [系部编号]
, [教师姓名]
, [性别]
, [年龄]
, [职称]
, [所在院系]
FROM [dbo].[教师]
GO
```

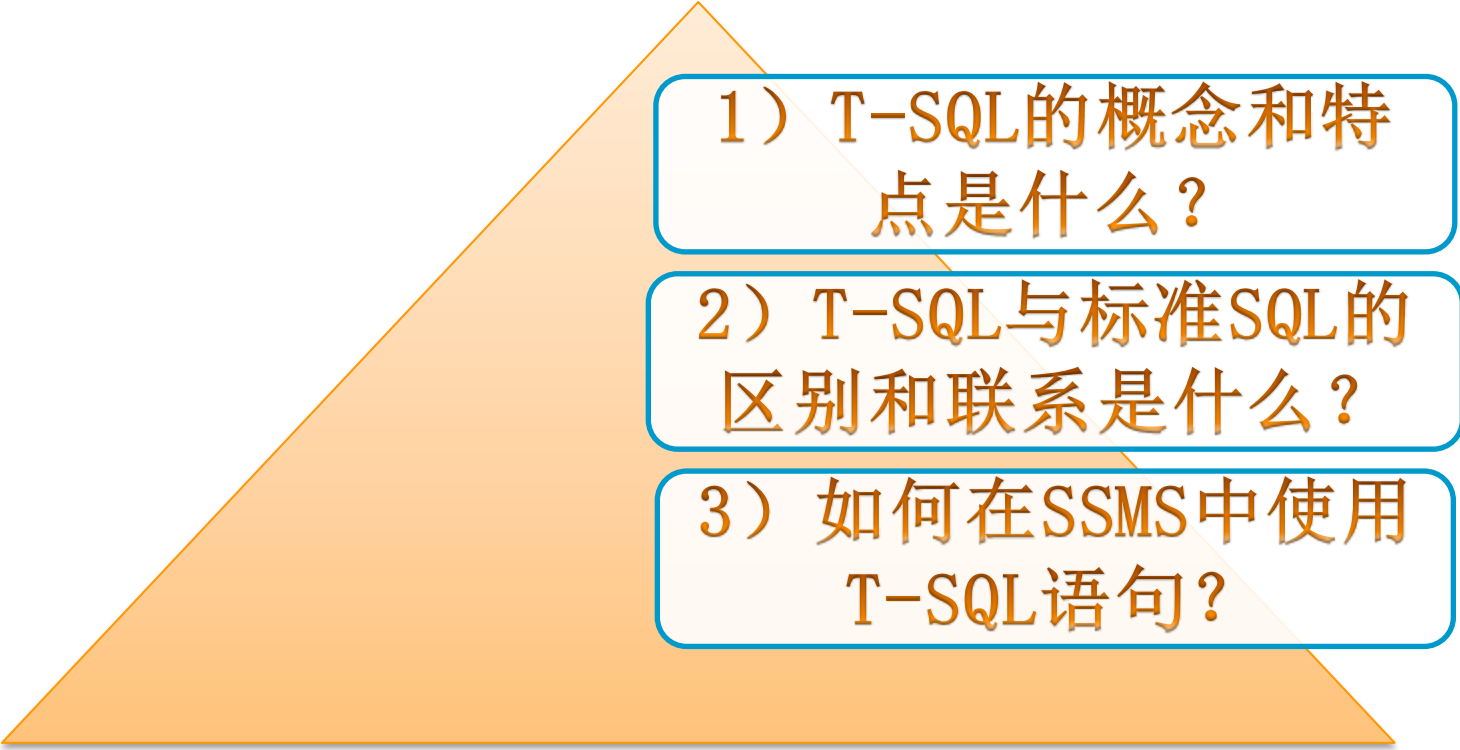
Below the query, the '结果' (Results) tab shows the output of the query as a table with 8 columns and 1 row:

	工号	系部编号	教师姓名	性别	年龄	职称	所在院系
1	A312	B3215	周明	男	43	教授	信息学院网络工程

The status bar at the bottom indicates '查询已成功执行。' (Query executed successfully.) and 'DESKTOP-UJBH5IS (13.0 RTM) | sa (53) | 教务管理系统 | 00:00:00 | 1 行' (1 row).

7.1 T-SQL基础概述

☺讨论思考

- 
- 1) T-SQL的概念和特点是什么？
 - 2) T-SQL与标准SQL的区别和联系是什么？
 - 3) 如何在SSMS中使用T-SQL语句？



7.2 批处理、脚本及事务

7.2.1 批处理概述

1. 批处理的概念
2. 批处理的规则
3. 指定批处理的方法
4. 批处理的结束与退出

7.2.2 脚本及事务

1. 脚本
2. 事务



7.2 批处理、脚本及事务

7.2.1 T-SQL的概念、特点和功能

■ 1. 批处理的概念

批处理是指一个或多个T-SQL语句的集合，由客户端应用程序一次性发送到SQL Server服务器以完成执行，它表示用户提交给数据库引擎的工作单元。

批处理是作为一个单元进行分析和执行的，它要经历的处理阶段有：分析（语法检查）、解析（检查引用的对象和列是否存在、是否具有访问权限）、优化（作为一个执行单元）。

SQL Server将批处理语句编译为单个可执行的单元，称为执行计划，执行计划中的语句每次执行一条，这种批处理方式这有助于节省执行时间。

7.2 批处理、脚本及事务

7.2.1 T-SQL的概念、特点和功能

■ 2. 批处理的规则

下面的规则适用于批处理的使用：

- 1) 创建默认、函数、过程、规则、模式、触发器、视图的语句不能在批处理中与其他语句组合使用。
- 2) 批处理必须以CREATE语句开始，所有跟在该批处理后的其他语句将被解释为第一个CREATE语句定义的一部分。
- 3) 不能在同一个批处理中更改表结构（例如修改字段名、新增字段，新增或更改约束等），然后引用新列。因为SQL Server可能还不知道架构定义发生了变化，导致出现解析错误



7.2 批处理、脚本及事务

7.2.1 T-SQL的概念、特点和功能

■ 2. 批处理的规则（续）

- 4) 不能在同一个批处理中删除一个对象之后，再次引用该对象。
- 5) 不可将规则和默认值绑定到表字段或自定义字段上之后，立即在同一个批处理中使用。
- 6) 使用SET语句设置的某些SET选项不能应用于同一个批处理中的查询。
- 7) 如果批处理中的第一个语句是执行某个存储过程的EXECUTE语句，则EXECUTE关键字可以省略。如果EXECUTE语句不是批处理中第一条语句，则必须保留。



7.2 批处理、脚本及事务

7.2.1 T-SQL的概念、特点和功能

■ 3. 指定批处理的方法

指定批处理的方法有4种。

- 1) 应用程序作为一个执行单元发出的所有SQL语句构成一个批处理，并生成单个执行计划。
- 2) 存储过程或触发器内的所有语句构成一个批处理，每个存储过程或者触发器都编译为一个执行计划。
- 3) 由EXECUTE语句执行的字符串是一个批处理，并编译为一个执行计划。
- 4) 由sp_executesql 存储过程执行的字符串是一个批处理，并编译为一个执行计划。



7.2 批处理、脚本及事务

7.2.1 T-SQL的概念、特点和功能

■ 3. 指定批处理的方法（续）

需要注意以下几点：

- 1) 若应用程序发出的批处理过程中含有EXECUTE语句，则已执行字符串或存储过程的执行计划讲和包含EXECUTE语句的执行计划分开执行。
- 2) 若sp_executesql存储过程所执行的字符串生成的执行计划与sp_executesql调用的批处理执行计划分开执行。
- 3) 若批处理中的语句激活了触发器，则触发器的执行将和原始的批处理分开执行。



7.2 批处理、脚本及事务

7.2.1 T-SQL的概念、特点和功能

■ 4. 批处理的结束与退出

(1) 执行批处理语句

用EXECUTE语句执行标量值的用户自定义函数、系统过程、用户自定义存储过程或扩展存储过程。同时支持T-SQL批处理内字符串的执行。

(2) 批处理结束语句

在SSMS、sqlcmd实用工具和osql实用工具中都使用GO命令作为批处理语句的结束标记，即当编译器执行到GO时会把之前的所有语句当做一个批处理来执行。但是GO并不是一个T-SQL语句，它只是供客户端工具识别的一个命令。

7.2 批处理、脚本及事务

7.2.1 T-SQL的概念、特点和功能

■ 4. 批处理的结束与退出（续）

GO命令和T-SQL命令不能在同一行，否则无法识别。但在GO命令行中可包含注释。用户必须遵照使用批处理的规则。

SQL Server 2005以及更高的版本中对GO命令这个客户端工具进行了增强，让它可以支持一个正整数参数，表示GO之前的批处理将执行指定的次数。当需要重复执行批处理是，就可以使用这个增强后的命令。该命令使用的语法具体格式为：

GO [count]

其中，count为一个正整数，²²指定批处理将执行的次数。

7.2 批处理、脚本及事务

7.2.1 T-SQL的概念、特点和功能

■ 4. 批处理的结

案例7-2

Use 教务管理

```
CREATE view vw_teacher
```

```
AS select * from 教师
```

```
GO
```

```
Insert into 教师 default values
```

```
GO 10
```

将这些语句在编辑器窗口中执行后，会发现数据库“教务管理系统”中出现一个视图“vw_teacher”；表“教师”中新增元组10个，每个元组中的属性值都为NULL（前提是该表未定义主码，即元组可重复）

```
/*创建视图“vw_teacher”*/
```

```
/*批处理执行前面语句1次*/
```

```
/*在“教师”表中插入*/
```

```
/*批处理执行插入元组语句10次*/
```



7.2 批处理、脚本及事务

7.2.1 T-SQL的概念、特点和功能

■ 4. 批处理的结束与退出（续）

（3）批处理退出语句

批处理退出语句的基本语法格式为：

RETURN [整型表达式]

该语句可无条件终止查询、存储过程或批处理的执行。存储过程或批处理不执行RETURN之后的语句。当存储过程使用该语句时，RETURN语句不能返回空值。可用该语句指定返回调用应用程序、批处理或过程的整数值。

7.2 批处理、脚本及事务

7.2.1 T-SQL的概念、特点和功能

■ 4. 批处理的结束与退出（续）

案例7-3 RETURN语句的使用。

USE 教务管理系统

GO

CREATE PROCEDURE checkstate @param char (10)

AS

IF(SELECT 职称 FROM 教师 where 工号=@param)='教授'

RETURN 1

ELSE

RETURN 2

7.2 批处理、脚本及事务

7.2.1 T-SQL的概念、特点和功能

■ 4.批处理的结束与退出（续）

以上语句创建了过程checkstate检查教师的职称状态。如果该教师职称是“教授”，将返回状态码1；其他情况下，返回状态码2。

再通过下面语句显示通过过程checkstate检查工号为“A312”的教师职称状态，并返回执行结果。

```
DECLARE @return_status int
EXEC @return_status = checkstate 'A312'
SELECT '返回值' = @return_status
GO
```

执行该批处理语句后，结果窗口显示返回值为1，表示工号为“A312”的教师职称是教授。



7.2 批处理、脚本及事务

7.2.2 脚本及事务

■ 1. 脚本

脚本是存储在文件中的一系列SQL语句，脚本文件保存时的扩展名为.sql，该文件是一个纯文本文件。

脚本文件中可包含一个或多个批处理，GO作为批处理结束语句，若脚本中无GO语句，则作为单个批处理。

使用脚本可以将创建和维护数据库时进行的操作保存在磁盘文件中，方便以后重复使用该段代码，还可以将此代码复制到其他计算机上执行。因此，对于经常操作的数据库，保存相应的脚本文件是一个良好的使用习惯。

7.2 批处理、脚本及事务

7.2.2 脚本及事务

■ 2. 事务

事务的概念主要是为了保持数据的一致性而提出的。

例如，在一次银行交易中，需要将账户A中的一万元转账至账户B中，那么这个操作序列就应该是两步来完成。第一步从账户A中减去一万元，第二步向账户B中加入一万元。可以设想，如果第一步完成后，比如突然断电等故障导致第二个动作没来得及完成，那么故障恢复后，会发现账户A中金额少了一万，但是账户B中并没有增加1万，总金额发生错误，造成数据不一致现象。这就需要用到事务的概念。。

7.2 批处理、脚本及事务

7.2.2 脚本及事务

■ 2. 事务

(1) 事务的定义

事务 (transaction) 是用户定义的数据操作序列，这些操作可作为一个完成的不可分割的工作单元，一个事务内的操作要么全部执行，要么全部不执行。

事务的开始和结束可以由用户显式控制，在SQL中常用的定义事务的语句为3条：

```
BEGIN TRANSACTION;
```

```
COMMIT;
```

```
ROLLBACK;
```


7.2 批处理、脚本及事务

7.2.2 脚本及事务

■ 2. 事务

(1) 事务的定义

```
BEGIN TRANSACTION;  
COMMIT;  
ROLLBACK;
```

其中，BEGIN TRANSACTION表示事务的开始，以COMMIT或ROLLBACK结束事务。COMMIT表事务的提交，即将所有对数据库的更新写会到磁盘上的物理数据库中，正常结束该事务。ROLLBACK表示事务的回滚，用于事务执行过程中发生了某种故障，事务不能继续执行，系统将该事务中对数据库的所有已完成更新操作全部撤销，回滚到事务开始时的状态。

7.2 批处理、脚本及事务

7.2.2 脚本及事务

■ 2. 事务

(1) 事务的定义

```
BEGIN TRANSACTION;  
COMMIT;  
ROLLBACK;
```

上述银行转账示例中，如果把整个过程当作一个事务，那么两步操作要么都执行，要么都不执行。也就是说，当发生故障造成事务中断时，系统会自动撤销已完成的第一个动作，这样当系统恢复时，账户A和账户B的金额还是正确的。

7.2 批处理、脚本及事务

7.2.2 脚本及事务

■ 2. 事务

(1) 事务的定义

- 事务不同于程序：一般地，一个程序中包含多个事务；一个事务则可以是一条SQL语句、一组SQL语句或整个程序。
- 事务不同于批处理：事务是工作的原子工作单位，而一个批处理可以包含多个事务，一个事务也可以在多个批处理多种的某些部分提交。当事务在执行中途被取消或回滚时，SQL Server会撤销自事务开始以来进行的部分活动，而不考虑批处理是从哪里开始的。

7.2 批处理、脚本及事务

7.2.2 脚本及事务

■ 2. 事务

(2) 事务的特征

事务定义了一个或多个数据库操作的序列，但是并非任意数据库操作序列都能成为事务，为了保护数据的完整性，一般要求事务具有以下4个特征。

- 原子性 (Atomic)
- 一致性 (Consistent)
- 隔离性 (Isolated)
- 持久性 (Durable)

ACID特性



7.2 批处理、脚本及事务

7.2.2 脚本及事务

■ 2. 事务

(3) 事务的类型

任何对数据的修改都是在事务环境中进行的。SQL Server 使用三类事务模式来管理数据的修改。

- 1) **显式事务**。事务中存在显式的BEGIN TRANSACTION语句开始，以COMMIT或ROLLBACK语句显式结束。
- 2) **隐式事务**。在前一个事务完成时新事务就隐式启动，但是每个事务仍然以COMMIT或ROLLBACK语句显式结束。

7.2 批处理、脚本及事务

7.2.2 脚本及事务

■ 2. 事务

(3) 事务的类型 (续)

3) 自动提交事务。如果数据修改语句是在没有显式或者隐式事务的数据库中执行的，就称为自动提交事务。简言之，它一次仅执行一个操作，每条单独的语句都可看作一个事务。

4) 批处理级事务。只能应用于多个活动结果集 (MARS)，在MARS回话启动的T-SQL显式或隐式事务变为批处理级事务。当批处理完成时没有提交或回滚的批处理级事务自动由SQL Server进行回滚。

7.2 批处理、脚本及事务

7.2.2 脚本及事务

■ 2. 事务

(3) 事务的类型 (续)

3) 自动提交事务。如果数据修改语句是在没有显式或者隐式事务的数据库中执行的，就称为自动提交事务。简言之，它一次仅执行一个操作，每条单独的语句都可看作一个事务。

4) 批处理级事务。只能应用于多个活动结果集 (MARS)，在MARS回话启动的T-SQL显式或隐式事务变为批处理级事务。当批处理完成时没有提交或回滚的批处理级事务自动由SQL Server进行回滚。

7.2 批处理、脚本及事务

7.2.2 脚本及事务

案例7-4 通过以下银行转账的算法理解事务的概念。

```
BEGIN TRANSACTION
BALANCE1=账户A的余额;
BALANCE=BALANCE-AMOUNT; /*AMOUNT为转账金额*/
IF (BALANCE<0) THEN
    {PRINT ‘账户金额不足，不能转账’ ;
    ROLLBACK;}
ELSE
    {BLANCE2=账户B的余额;
    BALANCE2=BALANCE2+AMOUNT;
    账户B的余额=BALANCE2;
    COMMIT;}
```

☺讨论思考

1) 什么是批处理？

2) 什么是脚本？脚本有何作用？

3) 什么是事务？如何通过事务进行数据一致性控制？

7.3 流程控制结构

7.3.1 顺序结构

1. BEGIN...END

2. SET

3. SELECT

4. PRINT

7.3.2 选择结构

1. IF...ELSE

2. CASE

7.3.3 循环结构

1. WHILE

7.3.3 循环结构

1. GOTO

2. WAIT...FOR

3. RETURN

7.3 流程控制结构

表7-1 T-SQL提供的主要流程控制语句

语句	具体描述
BEGIN...END	定义语句块
BREAK	退出最内层的WHILE循环
CASE	允许表达式按照条件返回不同值
CONTINUE	重新开始WHILE循环
GOTO	将程序的执行跳到相关标签处
IF...ELSE	判断条件是否成立，执行相应分支
RETURN	无条件退出
WAITFOR	为语句的执行设置延迟
WHILE	当指定条件为真时重复执行循环体

7.3 流程控制结构

7.3.1 顺序结构

■ 1 . BEGIN ... END语句

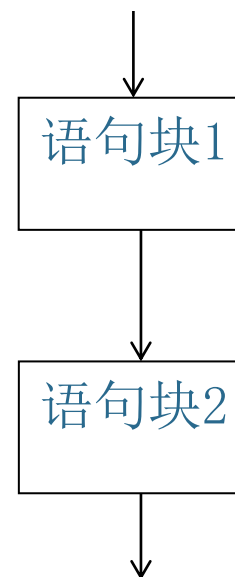
BEGIN

<语句1>

<语句2>

...

END



注意：

- 1) BEGIN和END必须成对出现
- 2) 语句功能类似于程序设计语言中的{...}

图7-3 顺序结构流程图

7.3 流程控制结构

7.3.1 顺序结构

■ 2 . SET语句

案例7-5

如果标量子查询返回多个值，SET语句赋值失败。

```
DECLARE @student_name AS char(8)
SET @student_name=(SELECT 姓名
FROM 学生
WHERE 年龄>20)
SELECT @student_name AS 学生姓名
GO
```

注意： SET语句给局部变量赋值时，一次只能操作一个变量，若要给多个变量赋值，则需要多个SET语句

7.3 流程控制结构

7.3.1 顺序结构

■ 3 . SELECT语句

SELECT语句可作为输出语句使用，其具体语法格式为：

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

该语句的功能是输出指定表达式的结果，默认为字符型。另外，SQL Server还支持一种非标准的赋值SELECT语句，允许在单个语句中既查询数据，又同时把从同一行中获取的多个值分配给多个变量。

7.3 流程控制结构

7.3.1 顺序结构

■ 3 . SELECT语句

案例7-6

将学号为BX15120的学生姓名和专业分别赋值给两个变量。

```
DECLARE @Name AS NVARCHAR(20)  
        AS NVARCHAR(20)
```

```
SELECT
```

```
    @Name=姓名,
```

```
    @Major=专业
```

```
FROM 学生
```

```
WHERE 学号='BX15120'
```

```
GO
```

注意：当满足条件的查询结果只有一行时，赋值SELECT语句的执行过程符合预期希望。但是当查询返回多个满足条件的结果行时，这段代码又该如何执行呢？

在执行过程中，对于每个满足条件的结果行，都会给相应变量进行赋值。但是，每次会用当前行的值覆盖掉变量中的原有值。这样当语句执行结束时，变量中的值是SQL Server访问到的最后一行中的值。

7.3 流程控制结构

7.3.1 顺序结构

■ 4 . PRINT输出语句

PRINT <表达式> 或者

PRINT msg_str | @local_variable | string_expr

说明：

- 1) 若表达式的值不是字符型，则需要先用Convert函数转换为字符型。
- 2) msg_str为字符串或Unicode字符串常量。
- 3) @local_variable为字符类型的局部变量，此变量必须是CHAR、NCHAR、VARCHAR或NVARCHAR类型的变量，或者是能够隐式转换为这些数据类型的变量。

7.3 流程控制结构

7.3.1 顺序结构

■ 4 . PRINT输出语句

PRINT <表达式> 或者

PRINT msg_str | @local_variable | string_expr

说明：

4) string_expr为返回字符串的表达式。可包括串联（即字符串拼接，T-SQL用“+”号实现）的文本值、函数和变量。

5) 消息字符串为非Unicode字符串时，最长为8000；如果为Unicode字符串是，最长为4000。超过该长度的字符串会被截断。

7.3 流程控制结构

7.3.1 顺序结构

■ 4 . PRINT输出语句

案例7-7

查看教师表中职称为教授的教师人数。

```
DECLARE @count int
IF EXISTS (SELECT 工号 FROM 教师 WHERE 职称='教授' )
BEGIN
SELECT @count=count(工号) FROM 教师 WHERE 职称='教授'
PRINT '教师表中职称为教师的人数为： ' +
CONVERT(CHAR(5), @count) + '人'
```

执行结果为： 教师表中职称为教师的人数为： 2 人

7.3 流程控制结构

7.3.2 选择结构

■ 1 . IF ... ELSE语句

```
IF <条件表达式>  
    <语句块1>  
[ELSE  
    <语句块2>]
```

注意：

如果<条件表达式>中包含SELECT语句，则必须将SELECT语句用圆括号括起来

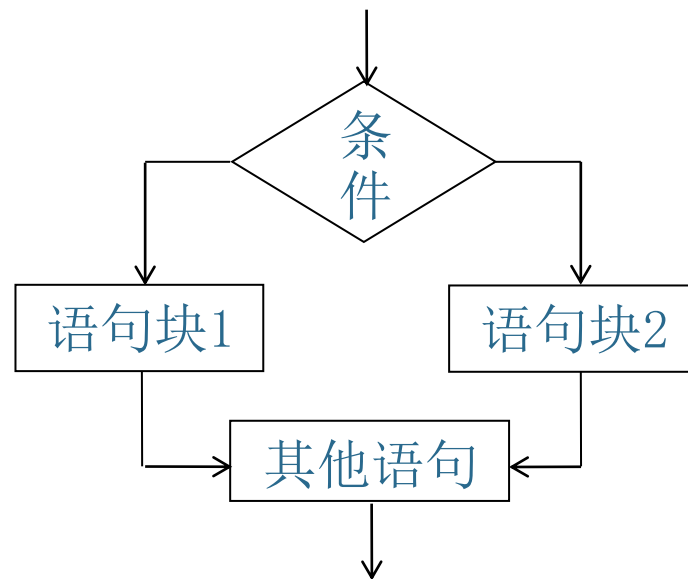


图7-4 选择结构流程图

7.3 流程控制结构

7.3.2 选择结构

■ 1 . IF ... ELSE语句

案例7-8

通过IF...ELSE...语句判断今天是否是今年的最后一天。

```
IF YEAR(SYSDATETIME( )) <> YEAR (DATEADD (day, 1,  
SYSDATETIME( )))
```

```
/*SYSDATETIME( )函数可以获得当前的系统时间*/
```

```
PRINT '今天是今年的最后一天!';
```

```
ELSE
```

```
PRINT '今天不是今年的最后一天!';
```

```
GO
```

7.3 流程控制结构

7.3.2 选择结构

■ 2 . CASE语句

CASE语句也称为多分支结构，可通过计算条件列表并返回多个可能结果表达式中的一个。CASE表达式有两种不同形式：简单CASE表达式和搜索式CASE表达式。

1) 简单CASE表达式

```
CASE <判断表达式>  
WHEN <简单表达式1> THEN <结果表达式1>  
WHEN <简单表达式2> THEN <结果表达式2>  
...  
WHEN <简单表达式n> THEN <结果表达式n>  
[ELSE 结果表达式n+1]
```

7.3 流程控制结构

7.3.2 选择结构

■ 2 . CASE语句

1) 简单CASE表达式

```
CASE <判断表达式>  
WHEN <简单表达式1> THEN <结果表达式1>  
WHEN <简单表达式2> THEN <结果表达式2>  
...  
WHEN <简单表达式n> THEN <结果表达式n>  
[ELSE 结果表达式n+1]
```

说明如下：

- 1) CASE表达式中可以有n个WHEN...THEN...语句，且至少有一个。
- 2) <判断表达式>可以是一个变量名、字段名、函数或子查询。

7.3 流程控制结构

7.3.2 选择结构

■ 2 . CASE语句

1) 简单CASE表达式

```
CASE <判断表达式>  
WHEN <简单表达式1> THEN <结果表达式1>  
WHEN <简单表达式2> THEN <结果表达式2>  
...  
WHEN <简单表达式n> THEN <结果表达式n>  
[ELSE 结果表达式n+1]
```

3) 简单表达式不能包含比较运算符，而是给出被比较的表达式或值，其数据类型必须与测试表达式的数据类型相同，或者可以隐式转换为测试表达式的数据类型。

7.3 流程控制结构

7.3.2 选择结构

■ 2 . CASE语句

1) 简单CASE表达式

```
CASE <判断表达式>  
WHEN <简单表达式1> THEN <结果表达式1>  
WHEN <简单表达式2> THEN <结果表达式2>  
...  
WHEN <简单表达式n> THEN <结果表达式n>  
[ELSE 结果表达式n+1]
```

4) 该语句执行过程为：将一个测试表达式和一组简单表达式进行比较，如果某个WHEN语句中简单表达式的值与测试表达式的值相等，则返回相应结果表达式的值；如果都没有则执行ELSE语句相应的结果表达式的值；如果没有指定ELSE子句则返回NULL值。

7.3 流程控制结构

7.3.2 选择结构

■ 2. CASE语句

1) 简单CASE表达式

案例7-9

利用简单CASE表达式显示“网上购物”数据库“会员”数据表中的会员等级及其对应会员卡类型。

SELECT 会员ID, 会员等级, 会员卡类型=

CASE 会员等级

WHEN 1 THEN '钻石卡会员'

WHEN 2 THEN '白金卡会员'

WHEN 3 THEN '金卡会员'

WHEN 4 THEN '银卡会员'

ELSE '其它'

END

FROM 会员

7.3 流程控制结构

7.3.2 选择结构

■ 2 . CASE语句

2) 搜索式CASE表达式

```
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
```

7.3 流程控制结构

7.3.2 选择结构

■ 2 . CASE语句

2) 搜索式CASE表达式

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

说明如下：

- 1) CASE关键字后面没有任何表达式。
- 2) 各WHEN关键字后面的都是布尔表达式，表达式中可以用到比较运算符，也可以使用逻辑运算符。

7.3 流程控制结构

7.3.2 选择结构

■ 2. CASE语句

2) 搜索式CASE表达式

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

3) 该语句执行过程为：按照从上到下顺序为每个**WHEN**子句后面的布尔表达式求值，然后返回第一个取值为真的布尔表达式所对应的结果表达式的值，若所有**WHEN**后面的布尔表达式值没有为真的则返回**ELSE**子句中结果表达式的值，若无**ELSE**子句则返回**NULL**值。

7.3 流程控制结构

7.3.2 选择结构

■ 2. CASE语句

2) 搜索式CASE表达式

案例7-10

利用搜索式CASE表达式，修改完成案例7-8的操作。

SELECT 会员ID, 会员等级, 会员卡类型=
CASE

WHEN 会员等级=1 THEN '钻石卡会员'

WHEN 会员等级=2 THEN '白金卡会员'

WHEN 会员等级=3 THEN '金卡会员'

WHEN 会员等级=4 THEN '银卡会员'

ELSE '其它'

END

FROM 会员

7.3 流程控制结构

7.3.3 循环结构

■ WHILE语句

WHILE <布尔表达式>
{ SQL语句|语句块|BREAK|CONTINUE }

注意：

- 1) BREAK用于终止整个循环的执行；
- 2) CONTINUE用于结束本次循环返回WHILE开始处，重新判断条件以决定是否重复执行下一次循环。

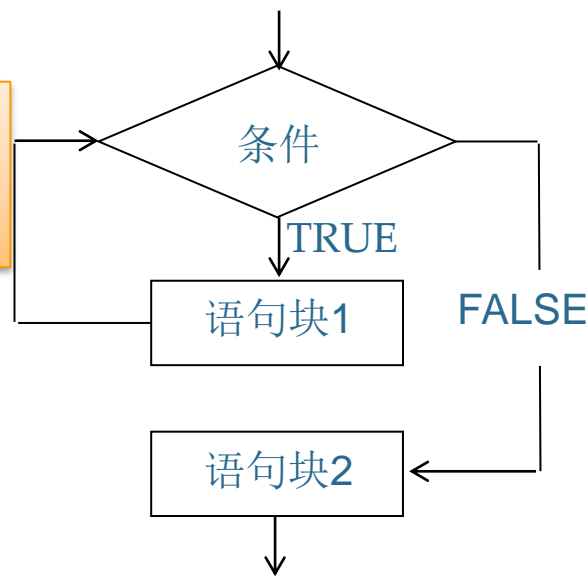


图7-6 循环结构流程图

7.3 流程控制结构

案例7-11

WHILE循环语句及BREAK和CONTINUE的使用。

/*WHILE语句用法*/

```
DECLARE @i AS INT=1
WHILE @i<=6
BEGIN
PRINT @i
SET @i=@i+1
END
GO
```

示例 (a)

循环执行结果:

1
2
3
4
5
6

/*BREAK语句用法*/

```
DECLARE @i AS INT=1
WHILE @i<=6
BEGIN
IF @i=3 BREAK
PRINT @i
SET @i=@i+1
END
GO
```

示例 (b)

1
2
3

/*CONTINUE语句用法*/

```
DECLARE @i AS INT=1
WHILE @i<=6
BEGIN
IF @i=3 BREAK
PRINT @i
SET @i=@i+1
END
GO
```

示例 (c)

1
2
4
5
6

7.3 流程控制结构

7.3.4 其他语句

■ 1 . 转移语句

GOTO label

说明：

1) 用于将程序执行流程更改到标签处。即程序执行到GOTO语句时，跳过其后面的T-SQL语句，并从相应标签位置继续执行。

2) label表示程序转到的相应标签处（起点）。标签的命名必须符合标识符规则，其定义格式为：

Label : <程序行>

3) GOTO语句可嵌套使用。

7.3 流程控制结构

7.3.2 选择结构

■ 1. 转移语句

案例7-12

利用GOTO语句求 $1+2+3+\dots+100$ 的和。

```
DECLARE @i int, @sum int
```

```
SET @i=1
```

```
SET @sum=0
```

```
Label1:
```

/*跳转到此处

```
SET @sum=@sum+@i
```

```
SET @i=@i+1
```

```
IF @i<=100
```

```
GOTO Label1
```

/*跳转到label1处

```
PRINT '累加和为: ' +STR(@sum)
```

7.3 流程控制结构

7.3.4 其他语句

■ 2 . 等待语句

等待语句的功能是在达到指定时间或时间间隔之前，或者指定语句至少修改或返回一行之前，阻止批处理、存储过程或事务的执行。

```
WAITFOR DELAY ' <等待时间长度> ' | TIME ' <执行时间> '  
[TIMEOUT timeout]
```

说明：

1) DELAY指明SQL Server等候的时间长度，最长为24h。
TIME指明SQL Server需要等待的时刻。DELAY与TIME使用的时间格式都是：hh:mm:ss。

7.3 流程控制结构

7.3.4 其他语句

■ 2 . 等待语句

**WAITFOR DELAY ' <等待时间长度> ' | TIME ' <执行时间> '
[TIMEOUT timeout]**

说明：

2) **实际的时间延迟可能与命令中指定的等待时间长度或执行时间不同，它依赖于服务器的活动级别。**时间计数器在计划完成与WAITFOR语句关联的线程后启动。如果服务器忙碌，则可能不会立即计划线程；因此，时间延迟可能比指定的时间要长。

3) **WAITFOR不更改查询的语义。**如果查询不能返回任何行，WAITFOR将一直等待，或等到满足TIMEOUT条件（如果已指定），TIMEOUT timeout指定消息到达队列前等待的时间（以毫秒为单位）。

7.3 流程控制结构

7.3.2 选择结构

■ 2. 等待语句

案例7-13 延迟20秒执行查询“学生”表命令。

```
WAITFOR DELAY '00:00:20'  
SELECT * From 学生
```

案例7-14 在时间12:30:00执行查询“学生”表命令。

```
WAITFOR TIME '12:30:00'  
SELECT * From 学生
```

7.3 流程控制结构

7.3.4 其他语句

■ 3 . 返回语句

RETURN语句用于从查询或过程中无条件退出，并返回到调用它的程序中。RETURN之后的语句是不执行的。

RETURN [<整型表达式>]

返回值	描述	返回值	描述
0	过程已成功返回	-7	资源出错，如没有空间
-1	对象丢失	-8	遇到非致命内部问题
-2	数据类型出错	-9	达到系统界限
-3	选定过程出现死锁	-10	出现致命内部矛盾
-4	许可权限出错	-11	出现致命内部矛盾
-5	语法出错	-12	表或索引损坏
-6	各种用户出错	-14	硬件出错

7.3 流程控制结构

☺讨论思考

1) T-SQL提供了哪些常用流程控制语句？

2) 选择结构通过哪两种形式可以实现？

3) BREAK和CONTINUE语句作用有何区别？



7.4 嵌入式SQL概述

SQL提供了两种使用方式：一种为独立交互式；另一种为嵌入式。作为独立的语言时，SQL能够独立地用于联机交互式的使用方式，用户可以在终端键盘上直接键入SQL命令对数据库进行操作；作为嵌入式语言时，SQL能够嵌入到高级语言（例如C、C++、Java等）程序中，供程序员编程序使用。

7.4.1 嵌入式SQL的概念

7.4.2 嵌入式SQL语法规则及用法

1 嵌入式SQL的语法规则

2. 建立和关闭数据库连接

7.4 嵌入式SQL概述

7.4.1 嵌入式SQL的概念

嵌入式SQL (Embedded SQL , **ESQL**) 就是将SQL语句直接嵌入到某种高级语言中, 被嵌入的高级语言称为**主语言**或者宿主语言。这种使用嵌入式SQL和主语言相结合的方式设计应用程序时, 一方面利用了主语言的过程控制和交互式优点, 另一方面保留了SQL强大的数据库管理功能, 两者相结合使得对数据库的编程更加功能完善和应用方便。

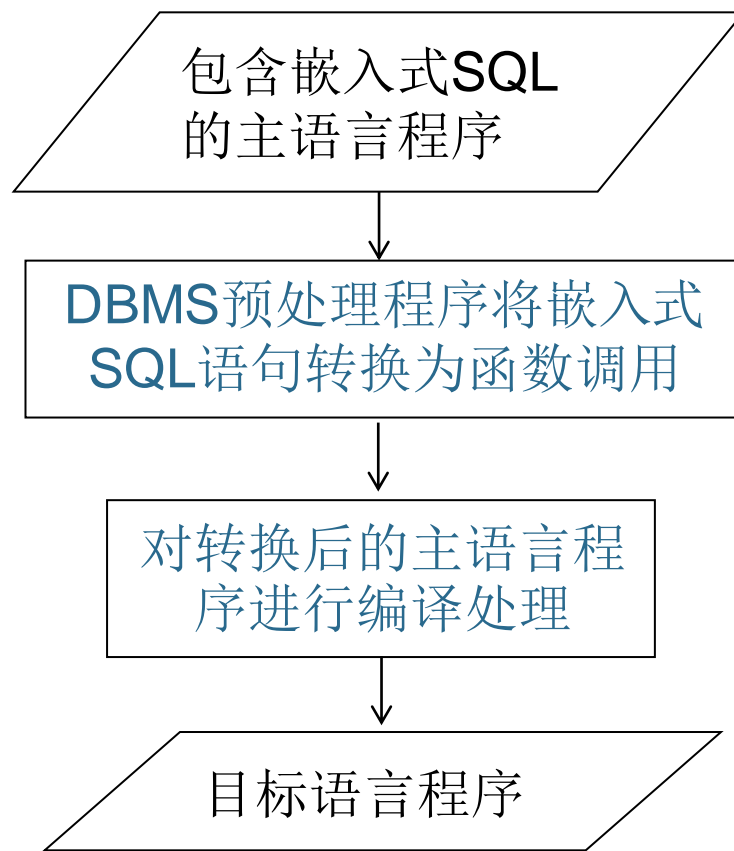


图7-7 ESQL处理过程

7.4 嵌入式SQL概述

7.4.2 嵌入式SQL语法规则及用法

■ 1. 嵌入式SQL的语法规则

1) 在嵌入式SQL中，为了能够区分SQL语句与主语句，所有的SQL语句前面都必须**加上前缀EXEC SQL**；结束标记则和主语言有关，例如在C语言中则以“;”作为结束标记。预编译程序根据开始标记和结束标记识别嵌入的SQL语句。当主语言为C语言时，嵌入式SQL的语法格式为：

EXEC SQL <SQL语句>;

案例7-15

将教师表中所有教师年龄增加1岁。

```
EXEC SQL UPDATE 商品
```

```
SET 年龄=年龄+1;
```

7.4 嵌入式SQL概述

7.4.2 嵌入式SQL语法规则及用法

■ 1. 嵌入式SQL的语法规则

2) 如果一条嵌入式SQL语句占用多行，在C语言中可以用**续行符** “\”，其他语言中也有相应规定。

3) **嵌入式SQL语句中可以使用主语言的程序变量（也称为主变量）来输入或者输出数据。**使用主变量之前，必须在SQL语句的BEGIN DECLARE SECTION...END DECLARE SECTION这两条预编译说明语句之间对主变量进行说明。说明之后，可以在SQL语句中任何一个地方使用主变量进行输入和输出数据。为了加以区分，SQL语句中如果使用的是主变量，那么主变量之前要加 “:” 号；如果使用的是SQL自身的数据库对象名（表名、视图名、列名等），则不用加 “:” 号。

7.4 嵌入式SQL概述

7.4.2 嵌入式SQL语法规则及用法

■ 1. 嵌入式SQL的语法规则

4) SQL是面向集合的操作方式，使用嵌入式SQL语句返回的结果记录可能是单条，也可能是多条，而主语言中的主变量一次只能存放一条记录。为此，嵌入式SQL引入了游标（cursor）的概念，用游标来协调这两种不同的处理方式。**游标是指系统为用户开设的一个缓冲区，存放SQL的执行结果，每个游标区都有一个名字，用户可以通过游标逐一获取记录并赋给主变量，交给主语言进一步处理。**

7.4 嵌入式SQL概述

7.4.2 嵌入式SQL语法规则及用法

■ 1.嵌入式SQL的语法规则

5) 在包含嵌入式SQL的应用程序中，SQL语句负责管理数据库，主语言语句负责控制流程和其他功能。数据库的工作单元和主语言程序的工作单元之间的通信主要包括以下几点。

- ◆ **主语言通过主变量向SQL语句提供参数。**
- ◆ **DBMS通过SQL通信区（SQL Communication Area，SQLCA）向应用程序返回SQL语句的执行状态信息。**
- ◆ **SQL语句操作数据库的结果，可通过主变量和游标进行处理。**

7.4 嵌入式SQL概述

7.4.2 嵌入式SQL语法规则及用法

■ 2. 建立和关闭数据库连接

1) 嵌入式SQL程序要访问数据库必须先连接数据库，关系数据库管理系统根据用户信息对连接请求进行合法验证，只有通过了身份验证，才能建立一个可用的合法连接。

建立数据库连接的ESQL语句是：

```
EXEC SQL CONNECT TO target [AS connection-name]  
[USER user-name];
```

其中：target是要连接的数据库服务器，它可以是一个常见的服务器标志串，如<DBname>@<hostname>:<port>，可以是包含服务器标志的SQL串常量，也可以是DEFAULT。
connection-name是可选的连接名。

7.4 嵌入式SQL概述

7.4.2 嵌入式SQL语法规则及用法

■ 2. 建立和关闭数据库连接

2) 程序运行过程中可以建立多个连接，执行的所有数据库单元的工作都在该操作提交时所选择的当前连接上。程序运行过程中可以修改当前连接，对应的ESQL语句为：

```
EXEC SQL SET CONNECTION connection-name|DEFAULT;
```

3) 当某个连接上的数据库操作完成后，应用程序应该主动关闭数据库连接，释放所占用的连接资源。关闭数据库连接的ESQL语句是：

```
EXEC SQL DISCONNECT [connection];
```

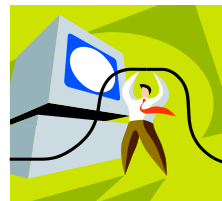
☺讨论思考

1) SQL语句的两种使用方式是什么？

2) 什么是嵌入式SQL？其基本处理过程是什么？

3) 嵌入式SQL的语法规定有哪些？

7.5实验七 流程控制语句



■ 7.5.1 实验目的

- (1) 掌握局部变量和全局变量的赋值与输出。
- (2) 熟悉常用函数以及流程控制语句。
- (3) 掌握T-SQL中顺序结构、选择结构与循环结构的实现

■ 7.5.2 实验内容及步骤

- (1) 变量的定义与输出
- (2) 练习PRINT与SELECT两种输出语句的使用
- (3) 选择结构
- (4) 循环结构

■ 7.5.3 实验练习



7.6 本章小结

- ◆ **T-SQL是对标准SQL的扩展**，包含了许多SQL所不具备的编程功能。本章系统地介绍了T-SQL应用编程中的相关概念、理论和操作。
- ◆ **批处理**是一次性将多个T-SQL语句发送给服务器以完成执行的工作方式，这有助于节省语句的执行时间。**脚本**是指存储在文件中的一系列SQL语句，将常用的T-SQL语句保存为脚本文件，可方便以后重复使用或复制到其他计算机上执行。**事务**是用户定义的数据操作序列，这些操作作为一个完整的不可分隔的工作单元，一个事务内的操作要么都做，要么都不做。利用事务可以保持数据的一致性。
- ◆ T-SQL中提供了一些常用的**流程控制语句**，通过这些语句使得T-SQL除了具备标准SQL的优点之外，又实现了**顺序、选择、循环**等程序结构的流程控制。
- ◆ **嵌入式SQL**是将SQL语句嵌入到某种高级语言中使用，一方面利用了SQL强大的数据库管理功能，另一方面利用了**主语言**的流程控制能力及数据处理能力，两者相结合来实现复杂应用的需求。

国家“十三五”重点出版规划项目
上海高校优秀教材奖获得者主编

上海市高校精品课程
特色教材

诚挚谢意！



数据库原理及应用

——基于SQL Server 2016