

第3章

T-SQL语言基础

本章要点：

T-SQL 即 Transact-SQL 的简写,是使用 Microsoft SQL Server 的核心,所有与 Microsoft SQL Server 实例通信的应用程序都通过发送 T-SQL 语句到服务器来完成对数据库的操作。它由数据控制语言(DCL)、数据定义语言(DDL)、数据操作语言(DML) 和附加的语言元素四部分组成。

本章重点介绍：

- 数据类型；
- 常量与变量；
- 运算符；
- 批处理与流程控制语句。

数据库提供的语言一般局限于对数据库的操作,它不是完备的程序设计语言,也不能独立地用来编写应用程序。

SQL 是结构化查询语言 (Structure Query Language) 的英文缩写,Transact-SQL 是由国际标准化组织 (ISO) 和美国国家标准学会 (ANSI) 发布的 SQL 标准中定义的语言的扩展。

3.1 数据类型

SQL Server 2005 提供了一系列系统定义的数据类型,它是数据库对象的一个属性,包括如下几项：

- 数据类别,如字符型、整数型以及数值型等；
- 存储数据值的长度或大小；
- 数值的精度；
- 数值的小数位。

数据类型分为系统数据类型和用户自定义数据类型两类。

3.1.1 系统数据类型

在表 3-1 中给出了 SQL Server 2005 所支持的数据类型。

下面就主要数据类型进行说明。

1. 整型数据

整型数据包括 BigInt、Int、SmallInt、TinyInt,它们的取值范围不同。例如,年龄取值是 0~200 之间,则用 TinyInt 就可以了,若用其他整型数据类型则在内存中占用的空间较大。

Bit 可以取值为 1、0 或 Null。如果表中的列为 8bit 或更少,则这些列作为一个字节存储;如果列为 9~16 bit,则这些列作为两个字节存储,依此类推。字符串值 True 可以转换为 1,False 可以转换为 0。

2. 货币型数据

货币型数据包括 Money 和 SmallMoney 两种,它们的取值范围不同。在使用时,必须在前面上加货币表示的符号(\$),并且数据中间不能有逗号(,);若货币值是负数,则需要“\$”前加上负号(-)。例如 \$ 300、\$ 123.50、-\$ 358.2、-\$ 6859。

3. 浮点型数据

浮点型数据包括 Float、Real,它们是近似数值数据类型,它们的取值范围不同。两者通常用科学记数法表示数据,用 E 表示阶数。例如 1.64E3、-2.84E6、6.3E-10 等。

表 3-1 SQL Server 2005 支持的数据类型

数据类型	名称	取值范围	说明或存储长度
BigInt	大整数	$-2^{63} \sim 2^{63} - 1$	8 字节
Int	整数	$-2^{31} \sim 2^{31} - 1$	4 字节
SmallInt	短整数	$-2^{15} \sim 2^{15} - 1$	2 字节
TinyInt	微短整数	0~255	1 字节
Bit	位型数	1、0 或 Null	
Decimal	精确数值数据类型	$-10^{38} + 1 \sim 10^{38} - 1$	
Numeric	精确数值数据类型	$-10^{38} + 1 \sim 10^{38} - 1$	
Money	货币型数据	$-10^{63} + 1 \sim 10^{63} - 1$	8 字节
SmallMoney	货币型数据	-214748.3648~214748.3647	4 字节
Float	浮点型数据	$-1.79\text{E}+308 \sim 1.79\text{E}+308$	存储长度与数值位数有关
Real	浮点型数据	$-3.40\text{E}+38 \sim 3.40\text{E}+38$	4 字节
DateTime	日期时间型数据	1753 年 1 月 1 日到 9999 年 12 月 31 日的日期和时间数据,精确到 3%秒	8 字节
SmallDateTime	日期时间型数据	1900 年 1 月 1 日到 2079 年 12 月 31 日的日期和时间数据,精确到分钟	4 字节
Char	字符型数据	1~8000,固定长度的非 Unicode 字符数据	一个字符占一个字节
VarChar	字符型数据	1~8000,可变长度的非 Unicode 字符数据	一个字符占一个字节
Nchar	字符型数据	1~4000,固定长度的 Unicode 字符数据	一个字符占两个字节
Nvarchar	字符型数据	1~4000,可变长度的 Unicode 字符数据	一个字符占两个字节
Text	文本型数据	可变长度的非 Unicode 数据,最大长度为 $2^{31} - 1$ 个字符	一个字符占一个字节
Ntext	文本型数据	可变长度的 Unicode 数据,最大长度为 $2^{30} - 1$ 个字符	一个字符占两个字节

续表

数据类型	名称	取值范围	说明或存储长度
Binary	二进制数据	1~8000,固定长度的二进制数据	
VarBinary	二进制数据	1~8000,可变长度的二进制数据	
Image	图像型数据	可变长度的二进制数据,最大长度为 $2^{31}-1$ 个字节,用于存储图片、照片等	输入数据的实际长度
Cursor	游标	游标的引用	
Sql_variant	特殊数据类型	一种存储 SQL Server 支持的各种数据类型 (text、ntext、timestamp 和 SQL variant 除外)值的数据	
Table	特殊数据类型	一种特殊的数据类型,存储供以后处理的结果集	
TimeStamp	日期时间戳	数据库范围的唯一数字,每次更新行时也进行更新	
UniqueIdentifier	全局唯一标识符	全局唯一标识符 (GUID)	

4. 字符型数据

字符型数据包括 Char(n)、VarChar(n)两种。

Char(n)是定长字符数据类型,其中, n 是定义字符数据的长度, n 在1~8000之间,默认值是1。当表中的列定义为Char(n)类型时,若实际要存储的长度不足 n 时,则在串的尾部添加空格,以达到长度 n ,所以Char(n)的长度是 n 。

VarChar(n)是变长字符数据类型,其中, n 的规定与定长字符型Char(n)中的 n 完全相同,但这里的 n 表示的是字符串可达到的最大长度。例如,列数据类型定义为VarChar(30),而输入的实际值是“shanxi”,则存储的实际长度为6个字符。

5. 文本型数据

当需要存储大量的字符数据时,如较长的备注、日志等信息,可使用文本型数据类型。文本型数据包括Text和Ntext两类,分别对应的是ASCII和Unicode字符。

Text类型存储数据的长度是实际英文字符个数字节,而Ntext类型存储数据的长度是实际英文字符个数的两倍。

6. 二进制数据

二进制数据类型表示的是数据流,包括Binary(固定长度)和VarBinary(可变长度)两种。Binary(n)是固定长度的 n 个字节的二进制数据, n 的取值范围为1~8000,默认值为1,若输入实际长度小于 n ,则不足部分用0填充。VarBinary(n)是 n 个字节的变长的二进制数据, n 的取值范围为1~8000,默认值为1。

7. 日期时间型数据

日期时间型数据用于存储日期和时间信息,包括DateTime和SmallDateTime两类。它们只是时间和存储长度不同,如表3-1所示。

常用的日期表示格式如下：

- 月/日/年：4/1/2010、04/01/10；
- 月-日-年：4-1-2010、04-01-2010；
- 月.日.年：4.1.2010、04.01.2010。

3.1.2 用户定义数据类型

除了使用系统提供的数据类型之外,SQL Server 2005 还允许用户定义数据类型,并可以用此数据类型来声明变量或字段。

在设计数据库时,如果多个表的列中要存储同种类型的数据,并且要确保这些列具有完全相同的数据类型长度时,可使用用户定义数据类型(注意:用户定义数据类型并不是用户可以随意定义一个数据类型,而是依据现有的系统数据类型定义一个新的数据类型)。

创建用户定义数据类型时必须提供以下 3 个参数:数据类型名称、新数据类型所依据的系统数据类型和为空性(若为空性未定义,则系统将依据数据库的默认设置进行指派)。下面举例说明用户定义数据类型的创建及使用。

1. 使用 SQL Server Management Studio 创建用户定义数据类型

【例 3-1】 使用 SQL Server Management Studio 在 college 数据库中创建一个名为 Stuno (学号) 的用户定义数据类型,数据类型为 VarChar,长度为 10,该列不能为 Null。

具体操作步骤如下：

- (1) 在 Management Studio 中展开 college 数据库。
- (2) 展开“可编程性”,再展开“类型”。
- (3) 右击“用户定义数据类型”,在弹出的快捷菜单中选择“新建用户定义数据类型”命令,如图 3-1 所示。
- (4) 若要允许此数据类型接受空值,则选中“允许空值”复选框,这里没有选中,表示不允许空值,如图 3-2 所示。
- (5) 在“默认值”和“规则”列表选择一个规则或默认值(如果有的话),以将其绑定到用户定义数据类型上。
- (6) 最后单击“确定”按钮完成。

2. 使用系统存储过程 sp_addtype 创建用户定义数据类型

语法格式如下：

```
sp_addtype [ @TypeName = ]新建数据类型名称,  
[ @phystype = ] 系统数据类型  
[, [ @nulltype = ] 'Null | Not Null | Nonull']
```

说明：

- (1) 系统数据类型是指 SQL Server 中的系统数据类型,但不能使用 TimeStamp 数据类型。

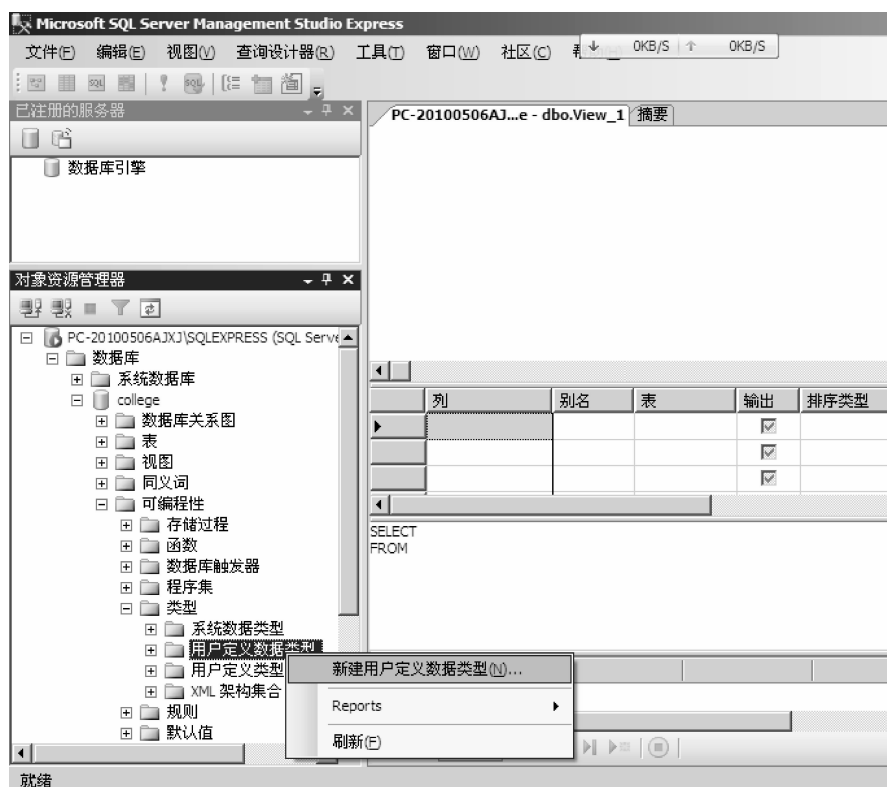


图 3-1 创建用户定义数据类型

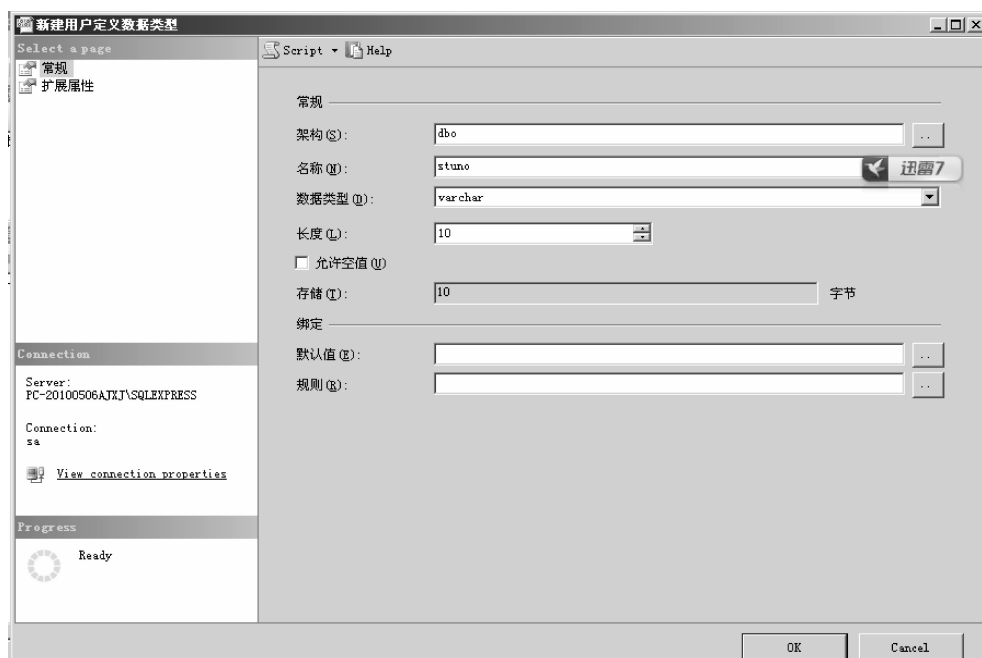


图 3-2 用户定义数据类型

(2) 指定用户定义数据类型的 Null 属性,其值可为 Null、Not Null 或 Nonull。默认时与系统默认的 Null 属性相同。

(3) 用户定义数据类型的名称在数据库中应是唯一的,但不同名称的用户自定义数据类型可以有相同的类型定义。在使用 CREATE TABLE 命令时,用户自定义数据类型的 Null 属性可以改变,但其长度定义不能更改。

【例 3-2】 定义生日(Birthday)数据类型。

```
Exec sp_addtype Birthday, DateTime, 'Not Null'
```

【例 3-3】 定义身份证号码(Cardid)数据类型。

```
Exec sp_addtype Cardid, 'Char(18)', 'Not Null'
```

系统数据类型与用户定义数据类型的关系:(1)用户定义数据类型以系统数据类型为基础,也就是说用户定义数据类型是系统数据类型的一个特例;(2)用户定义数据类型更方便、更能直观体现所用数据类型的特点。

3.2 常量和变量

常量和变量是程序设计中不可缺少的元素,下面介绍常量和变量的具体使用方法。

3.2.1 Select 语句

无源查询是使用 Select 语句查询不在表中的数据,其实质就是在客户机屏幕上显示的常量、变量或表达式的值。

1. SQL 查询窗口

SQL 查询窗口是一种图形工具,可以对单个 SQL 语句或由 SQL 语句编写的脚本进行编写调试,如图 3-3 所示。

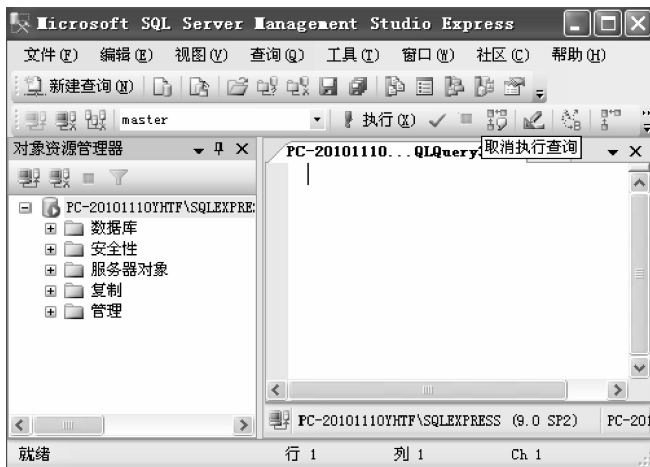


图 3-3 查询窗口

2. Select 语句无源查询

Select 语句无源查询是最简单的语句,用来查询不在表中的数据。

语法格式如下:

Select 常量|变量|函数|表达式[as 别名][, ... n]

【例 3-4】 使用 Select 语句查看常量。

```
Select '我是一名学生'
Select GetDate()           --获取当前系统的日期时间
```

3.2.2 常量

常量是指在程序运行过程中其值不变的量。常量又称为字面值和标量值,是表示一个特定数据的符号。根据常量值的不同类型,T-SQL 的常量主要有以下几种。

1. 字符串常量

字符串常量包含在单引号内,由字母(a~z、A~Z)、数字以及特殊字符(如!、@和#)组成。

如果字符串常量中包含一个单引号,如 I'm a Student,可以使用两个单引号表示这个字符串常量内的单引号,即表示为: I''m a Student。

在字符串常量前面加上字符 N,则表明该字符串常量是 Unicode 字符串常量。

例如,N'Mary'是 Unicode 字符串常量;'Mary'是字符串常量。

2. 二进制常量

二进制常量必须是以 0x 开头的十六进制数字字符串,这些常量可以不用引号括起来,例如 0xAF、0x12、0x(其长度为零)。

3. Bit 常量

Bit 常量只能使用数据 0 或 1 表示,并且不能放在引号中。如果使用一个大于 1 的数字来定义,那么这个数据会强制转换为 1。

虽然在 SQL Server Management Studio 中打开数据库表看到的 Bit 类型的字段内容为 Rrue 或 False,但是在 T-SQL 中使用 Bit 常量时,还是用 1 或 0 来表示 True 或 False。

4. DateTime 常量

DateTime 常量是用单引号括起来的日期或时间字符的字符串,只要输入的日期能明显分辨出生年月日,不论用哪种年月日的表达式都可以视为正确的输入,例如 '2010-2-01'、'02/12/10'、'May 3, 2010'。

5. Int 常量

Int 常量用没有引号括起来的、不包含小数点的数字来表示,例如 123、68、-98 等。

6. Decimal 常量

Decimal 常量用没有引号括起来的、包含小数点的数字来表示,例如 352.6、-78.30、

458.98 等。

7. Float 和 Real 常量

Float 和 Real 常量用科学记数法来表示,例如 122.2E5、-65.2E-3、0.52E6。

8. Money 常量

Money 常量用货币符号开头(如 \$)的数值来表示,例如 \$253、\$630、-\$123。

9. UniqueIdentifier 常量

UniqueIdentifier 常量表示全局唯一标识符(GUID)值的字符串,可以使用字符或二进制字符串格式指定,例如 6DF9B89F-4513-4CAF-A5AD-CEB24CC90CA5。

3.2.3 变量

变量对应内存中的一个存储空间,它与常量不同。变量的值在程序执行过程中可能随时改变。变量有两种类型:局部变量和全局变量。局部变量用来存储指定数据类型的单个数据值的对象。全局变量是由系统提供且声明的用来保存 SQL Server 系统运行状态数据值的变量。

1. 局部变量

局部变量是用户在程序中定义的变量,一次只能保存一个值,它仅在定义的程序范围内有效。局部变量可以临时存储数据。局部变量名总是以@符号开始,最长为 128 个字符。

使用 Declare 语句定义局部变量,内容包括局部变量名、数据类型和长度。

语法格式如下:

```
Declare {@local_variable data_type}[, ...n]
```

例如:

```
Declare @name VarChar(20)
```

上例定义的局部变量的初值为 Null(空)。需要使用 Select 语句或 Set 语句给局部变量赋值。Set 语句一次只能给一个局部变量赋值;Select 语句可以同时给一个或多个局部变量赋值。

【例 3-5】 编写计算 50 与 60 之和的程序。

```
Declare @i int,@j int,@sum int      --定义 3 个局部变量
Set @i = 50                          --给变量@i 赋值 50
Set @j = 60                          --给变量@j 赋值 60
Select @sum = @i + @j                --将@i 和@j 的和赋值给变量@sum
Print @sum                           --显示@sum 变量的内容
Go
```

为了便于理解,程序对每行代码均给出了注释。

通常在以下几种情况下使用局部变量:

(1) 作为计数器计算循环执行的次数或控制循环执行的次数。

- (2) 保存数据值以供语句测试。
- (3) 保存由存储过程返回代码返回的数据值。

2. 全局变量

全局变量是由 SQL Server 系统提供并赋值的变量,名称由“@@”符号开始。用户不能建立全局变量,也不能使用 Set 语句去修改全局变量的值。通常应该将全局变量的值赋给在同一个批中的局部变量,以便保存和处理。

全局变量分为两类:一类是与 SQL Server 连接有关的全局变量,如 @@rowcount 表示受最近一个语句影响的行数;另一类是与系统内部信息有关的全局变量,如 @@version 表示 SQL Server 的版本号。表 3-2 介绍了几个常用的全局变量。

表 3-2 SQL 常用的全局变量

名 称	说 明
@@connections	返回当前服务器的连接数目
@@rowcount	返回上一条 T-SQL 语句影响的数据行数
@@error	返回上一条 T-SQL 语句执行后的错误号
@@procid	返回当前存储过程的 ID 号
@@remserver	返回登录记录中远程服务器的名称
@@spid	返回当前服务器进程的 ID 标识
@@version	返回当前 SQL Server 服务器的版本和处理器类型
@@language	返回当前 SQL Server 服务器的语言

【例 3-6】 显示到当前日期和时间为止试图登录 SQL Server 的次数。

```
Select GetDate() As '当前的日期和时间', @@CONNECTIONS AS '试图登录的次数'
```

3.3 运算符

运算符是一种符号,用来指定要在一个或多个表达式中执行的操作。SQL Server 2005 所使用的运算符类别有算术运算符、逻辑运算符、赋值运算符、字符串连接运算符、位运算符、一元运算符和比较运算符。

3.3.1 算术运算符

算术运算符对两个表达式执行数学运算,这两个表达式可以是数值数据类型中的一个或多个。

算术运算符有+(加)、-(减)、*(乘)、/(除)、%(取模)。%(取模)返回一个除法运算的整数余数。例如,12%5=2,这是因为 12 除以 5 余数为 2。+(加)和-(减)运算符也可用于对 DateTime 和 SmallDateTime 值执行算术运算,例如:

```
Select GetDate(),GetDate()-10, Cast((GetDate()-cast('2006-01-18'As DateTime))As Int)
--查询当前日期、10 天前日期、日期间隔天数等
```

3.3.2 一元运算符

一元运算符只对一个表达式执行操作,该表达式可以是 Numeric 数据类型中的任何一种数据类型,如表 3-3 所示。

表 3-3 一元运算符

运 算 符	描 述
+(正)	返回数值表达式的正值
-(负)	返回数值表达式的负值
~(位非)	将指定的整型数值转换为二进制形式,然后按位进行逻辑非运算

例如,Select +(-9),-(-9),~(-9),~9,其查询结果为-9、9、8、-10。

3.3.3 比较运算符

比较运算符测试两个表达式是否相同。除了 Text、Ntext 或 Image 数据类型的表达式外,比较运算符可以用于所有的表达式。T-SQL 比较运算符有=(等于)、>(大于)、<(小于)、>=(大于等于)、<=(小于等于)、<>(不等于)、!= (不等于,非 SQL-92 标准)、!<(不小于,非 SQL-92 标准)、!>(不大于,非 SQL-92 标准)。

具有 Boolean 数据类型的比较运算符的结果有 3 个值: True、False 和 Unknown。返回 Boolean 数据类型的表达式称为布尔表达式。

与其他 SQL Server 数据类型不同,Boolean 数据类型不能被指定为表列或变量的数据类型,也不能在结果集中返回。

当 Set Ansi Nulls 为 On 时,带有一个或两个 Null 表达式的运算符返回 Unknown。当 Set Ansi Nulls 为 Off 时,上述规则同样适用,但是两个表达式均为 Null,则等号(=)运算符返回 True。例如,如果 Set Ansi Nulls 为 Off,则 Null=Null 返回 True。

3.3.4 逻辑运算符

逻辑运算符对某些条件进行测试,以获得其真实情况。逻辑运算符和比较运算符一样,返回带有 True 或 False 值的 Boolean 数据类型,如表 3-4 所示。

表 3-4 逻辑运算符

运 算 符	含 义
All	如果一组的比较都为 True,那么就为 True
And 或 &&	如果两个布尔表达式都为 True,那么就为 True
Any 或 Some	如果一组的比较中任何一个为 True,那么就为 True
Between	如果操作数在某个范围之内,那么就为 True
Extsts	如果子查询包含一些行,那么就为 True
In	如果操作数等于表达式列表中的一个,那么就为 True
Like	如果操作数与一种模式相匹配,那么就为 True
Not 或 !	对任何其他布尔运算符的值取反
Or 或	如果两个布尔表达式中的一个为 True,那么就为 True