

# Transact-SQL编程基础

Email: 18442056@QQ.com

# 学习目标

- ❖ 掌握变量的使用方法
- ❖ 掌握运算符和表达式的使用
- ❖ 数据库内置内置函数

# 常量、变量及运算符

- 现代数据库在**1986**年制订了**SQL**标准，**1992**年对其进行扩展增加了编程能力，包括程序控制结构，变量和其它附加元素。
- 而学习数据库的编程则从最简单的常量、变量、表达式，函数开始。

## 常量

- 常量，也称为文字值或标量值，是表示一个特定数据值的符号，在程序运行过程中其值保持不变。
- 常量的格式取决于它所表示的值的数据类型

- **1. 字符串常量**

- 字符串常量括在单引号内并包含字母、数字字符（a-z、A-Z 和 0-9）以及特殊字符，如!、@ 和#。
- 如：'4711207@qq.com', 'hello123'

- **2. 数值常量**

- 数值常量以没有用引号括起来的数字字符串来表示，包括整数：例如100， 54等；
- 小数：例如123.45， 5.5等；

- **3. 日期时间常量**

日期时间常量使用特定格式的字符日期时间值来表示，并被单引号括起来。

例如：'2010-2-7'，'2011-2-5 14:23:65'

- **4. 空值**

空值是表示值未知，不同于空白或零值，用NULL来表示。

# 变量

- 变量是可以对其赋值并参与运算的一个实体，其值在运行过程中可以发生改变。变量可以分为全局变量和局部变量两类，其中全局变量由系统定义并维护，局部变量由用户定义并赋值。

## 1、全局变量

- 全局变量由系统定义，通常用来跟踪服务器范围和特定会话期间的信息，不能被用户显式地定义和赋值，但是我们可以通过访问全局变量来了解系统目前的一些状态信息，全局变量以两个@符号开头



## SQL Server中较常用的全局变量

变量	说明
@@version	当前SQL Server大的版本信息
@@error	当前SQL Server执行的最后一条语句的返回码
@@rowcount	当前SQL Server执行的最后一条语句影响的行数
@@identity	当前SQL Server执行的最后一条语句影响的行数
@@fetch_status	当前SQL Server执行的最后一条语句影响的行数
@@nestlevel	当前SQL Server执行的最后一条语句影响的行数
@@server_name	当前SQL Server执行的最后一条语句影响的行数
@@spid	当前SQL Server执行的最后一条语句影响的行数
@@cpu_busy	SQL Server 自上次启动后的工作时间

-

select \* from T\_Classes

print @@Rowcount

结果

消息

(6 行受影响)

受影响的行数为:6

## • 2. 声明局部变量

- 局部变量用于在一次查询中临时保存数据，必须在使用前用**Declare** 语句声明：
- 指定局部变量名称。名称的第一个字符必须是@。
- 指定变量的数据类型，可以是系统提供的数据类型或用户自定义数据类型。对于字符型变量，还可以指定长度；数值型变量，指定精度和小数位。赋初值Null
- **Declare**语句的语法如下：
- **Declare @变量名 [AS] 数据类型**
- 如 `declare @stuname varchar(20), @age int`

使用**Set**语句或**SELECT**语句 给变量赋值

格式如下：

**SET @局部变量 = 表达式**

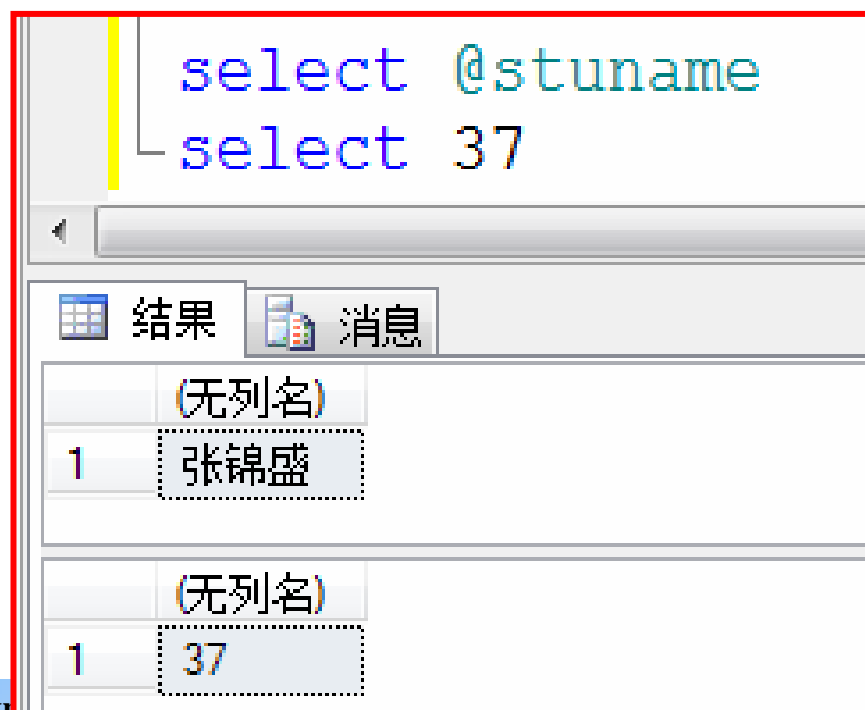
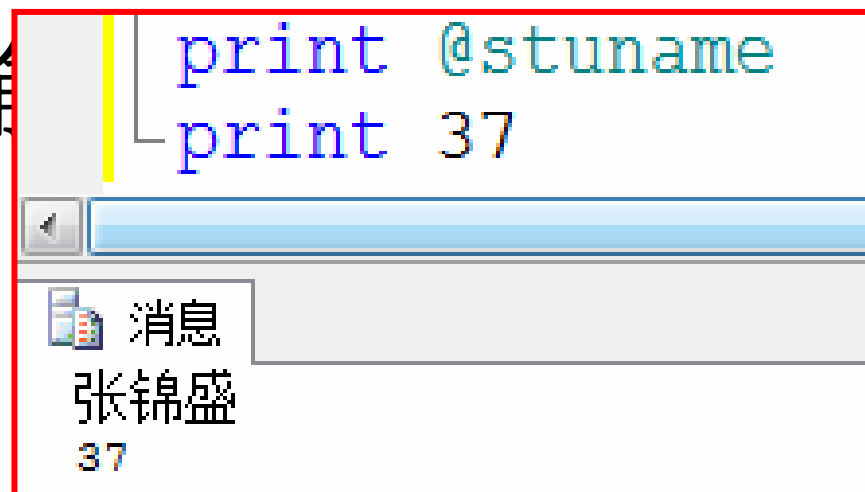
- **SELECT @局部变量=表达式 [,...n]**

**SELECT**与**SET**不同的是：**SELECT**可以一次为多个变量赋值。

```
declare  @stuname  varchar(20),  @age int  
  
set @stuname = '张锦盛'  
  
select @stuname = '张锦盛', @age = 1
```

# T-SQL 输出

- **print** @局部变量 | 常量
- 以消息的形式输出
- **select** @局部变量 | 常量
- 以查询结果的形式输出



# 运算符

- **Microsoft SQL Server 2008**提供了类似高级程序语言的**7**中类型的运算符，分别是**算术运算符、赋值运算符、位运算符、比较运算符、逻辑运算符、字符串运算符和一元运算符**。

# 1. 算术运算符

算术运算符对两个表达式执行运算。操作数必须是数值数据类型或能够转换为数值数据类型。  
SQL Server 2008提供的算术运算符如下表所示。

运算符	名称	
+	加	
-	减	
*	乘	
/	除	
%	取余	

```
print 17 + 5
print 17 - 5
print 17 * 5
print 17 / 5
print 17.0 / 5
print 17 % 5
```

消息

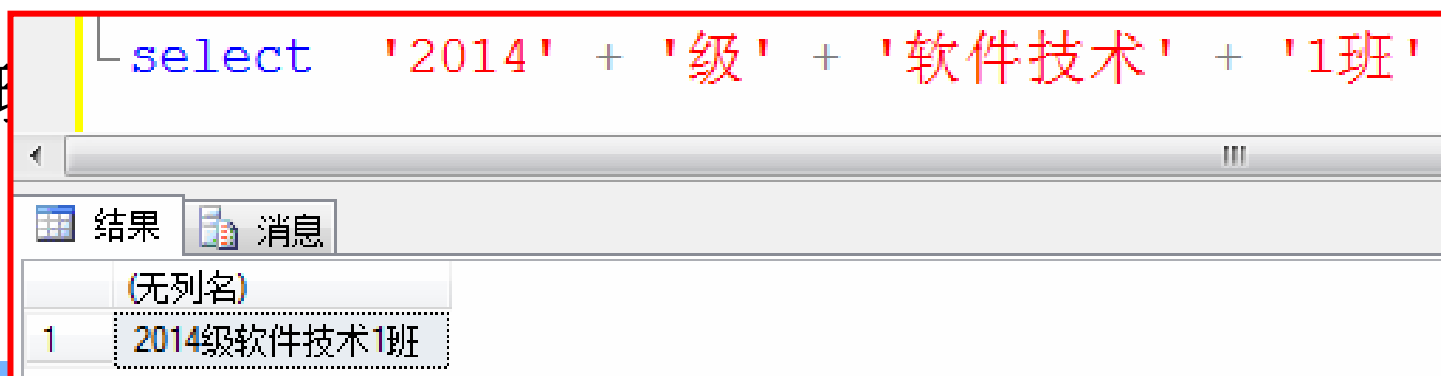
```
22
12
85
3
3.400000
2
```

## • 2. 赋值运算符

- 等号（=）是唯一的 Transact-SQL 赋值运算符。

## • 3. 字符串连接运算符

- 加号（+）是字符串串联运算符，可以用它将字符串串联起来。其他所有字符串操作都使用字符串函数进行处理



## 4. 比较运算符

- 比较运算符用来比较两个表达式值之间的大小关系，可以用于除了 **text**、**ntext** 或 **image** 数据类型之外的所有数据类型。运算的结果为 **True**、**False**,

运算符	名称	语法
<b>=</b>	等于	<b>stuname = '张锦盛'</b>
<b>&gt;</b>	大于	<b>age &gt; 50</b>
<b>&gt;=</b>	大于等于	<b>age &gt;= 50</b>
<b>&lt;</b>	小于	<b>age &lt; 20</b>
<b>&lt;=</b>	小于等于	<b>age &lt;= 20</b>
<b>&lt;&gt;或!=</b>	不等于	<b>poli &lt;&gt; '中共党员'</b>
<b>!&gt;</b>	不大于	<b>stucount !&gt; 15</b>
<b>!&lt;</b>	不小于	<b>stucount !&lt; 60</b>



## 5. 逻辑运算符

- 逻辑运算符用来对多个比较运算符条件结果进行运算，运算的结果任然为True或False，通常用来表示复杂的条件表达式。

运算符	说明	语法
<b>Not</b>	对表达式的值取反	<b>Not</b> age < 20
<b>And</b>	与，如果表达式的值都为True，结果为True，否则为False	age >= 18 <b>and</b> age <= 45
<b>Or</b>	或，如果表达式的值都为False，结果为False，否则为True	addr = '曲靖' <b>or</b> addr = '宣威'
<b>Between...and</b>	如果操作数在某个范围内，结果为True	age <b>between</b> 18 <b>and</b> 45
<b>In</b>	如果操作数等于值列表中的任何一个，结果为True	addr <b>in</b> ('曲靖', '宣威')
<b>Like</b>	如果字符型操作数与某个模式匹配，结果为True	coursename <b>like</b> '%程序%'
<b>Exists</b>	如果子查询结果不空，结果为True	<b>Exists</b> (子查询)
<b>Any或Some</b>	如果操作数与一系列值中的任何一个比较结果为True，结果为True	age > any(15, 16, 22, 21, 17, 18, 19)
<b>All</b>	如果操作数与一系列值中所有值的比较结果为True，结果为True	age > all (15, 16, 22, 21, 17, 18, 19)

# 常用函数

- 函数是能够完成特定功能并返回处理结果的一组**Transact-SQL**语句，处理结果称为“返回值”，处理过程称为“函数体”。函数可以用来构造表达式，可以出现在**Select**语句的选择列表中，也可以出现在**Where**子句的条件中。**SQL Server**提供了许多系统内置函数，同时也允许用户根据需要自己定义函数。
- **SQL Serve**提供的常用的内置函数主要有以下几类：**数学函数、字符串函数、日期函数、转换函数、聚合函数**等。

# 1、数学函数

- **Abs ( numeric\_expression )**: 返回指定数值表达式的绝对值
- **Round ( numeric\_expression , length [ ,function ] )**: 返回一个舍入到指定的长度或精度的数值
- **Floor ( numeric\_expression )**: 返回小于或等于指定数值表达式的最大整数
- **Ceiling ( numeric\_expression )**: 返回大于或等于指定数值表达式的最小整数
- **Power ( float\_expression , y )**: 返回指定表达式的指定幂的值
- **Sqrt ( float\_expression )**: 返回指定表达式的平方根
- **Square ( float\_expression )**: 返回指定表达式的平方
- **Exp ( float\_expression )**: 返回指定的表达式的指数值
- **Log ( float\_expression )**: 返回指定表达式的自然对数
- **Log10 ( float\_expression )**: 返回指定表达式的以10为底的对数
- **Sin ( float\_expression )**: 返回指定角度（以弧度为单位）的三角正弦值
- **Cos ( float\_expression )**: 返回指定角度（以弧度为单位）的三角余弦值

## 2、字符串函数

- **Ascii ( character\_expression )**: 返回字符表达式中最左侧的字符的 **ASCII** 代码值
- **Char ( integer\_expression )**: 将 **int**型的 **ASCII** 代码转换为字符
- **Str ( float\_expression [ , length [ , decimal ] ] )**: 返回由数字数据转换来的字符数据
- **Len ( string\_expression )**: 返回指定字符串表达式的字符数，其中不包含尾随空格
- **Substring ( value\_expression , start\_expression , length\_expression )**: 返回字符表达式的从 **start\_expression** 位置开始的长度为 **length\_expression** 的子串
- **Left ( character\_expression , integer\_expression )**: 返回字符串中从左边开始指定个数的字符
- **Right ( character\_expression , integer\_expression )**: 返回字符串中从右边开始指定个数的字符
- **Ltrim ( character\_expression )**: 返回删除了前导空格之后的字符表达式
- **Rtrim ( character\_expression )**: 截断所有尾随空格后返回一个字符串

### 3、日期和时间函数

Transact-SQL语言中提供下列日期时间函数：

- **getdate ( )**：返回系统当前的日期和时间
- **year ( date )**：返回表示指定参数 **date** 的“年”部分的整数
- **month ( date )**：返回表示指定参数 **date** 的“月”部分的整数
- **day ( date )**：返回表示指定参数 **date** 的“日”部分的整数
- **datetime ( datepart , date )**：返回表示指定参数 **date** 的指定 **datepart** 参数 的字符串
- **datepart ( datepart , date )**：返回表示指定参数 **date** 的指定 **datepart** 参数 的整数
- **datediff ( datepart , startdate , enddate )**：根据指定 **datepart** 参数返回两个指定日期之间的差值
- **dateadd ( datepart , number , date )**：根据 **datepart** 参数将一个时间间隔与指定 **date** 参数相加，返回一个新的 **datetime** 值

## 4、 类型转换函数——Cast函数

- **Cast**函数可以将一种数据类型的表达式强制转换为另一种数据类型的表达式。**Cast**函数的语法格式为：
- **Convert ( 表达式 as 数据类型)**

参数说明：

- **表达式**：任何有效的表达式。
- **数据类型**：目标数据类型。
- 如：**CONVERT(varchar(20), getDate(),111 )**

## 4、类型转换函数——Convert函数

- **Convert**函数可以将一种数据类型的表达式强制转换为另一种数据类型的表达式。**Convert**函数的语法格式为：
- **Convert ( 数据类型[ ( length ) ], 表达式 [, style ] )**

参数说明：

- **表达式**：任何有效的表达式。
- **数据类型**：目标数据类型。
- 如：**CAST( '102' as int )**



# 时间转换字符串的格式类型

下表是 convert 函数中对于日期类型进行转换所使用的 style 参数，左侧的两列表示将 **datetime** 或 **smalldatetime** 转换为字符数据的 style 值。给 style 值加 100，可获得包括世纪数位的四位年份 (yyyy)。

不带世纪数位 (yy)	带世纪数位 (yyyy)	标准	输入/输出**
-	0 或 100 (*)	默认值	mon dd yyyy hh:miAM (或 PM)
1	101	美国	mm/dd/yyyy
2	102	ANSI	yy.mm.dd
3	103	英国/法国	dd/mm/yy
4	104	德国	dd.mm.yy
5	105	意大利	dd-mm-yy
6	106		dd-mm-yy hh:mi