

7.2 Transact-SQL

本节要点

T-SQL语言用于管理SQL Server Database Engine实例，创建和管理数据库对象，以及插入、检索、修改和删除数据。T-SQL是对按照国际标准化组织(ISO)和美国国家标准协会(ANSI)发布的SQL标准定义的语言的扩展。对用户来说，T-SQL是可以与SQL Server数据库管理系统进行交互的唯一语言。

本章主要简单介绍T-SQL语言。介绍该语言的基本概念、语法格式、运算符和表达式以及基本语句和函数等的使用。

本章目录

[7.2.1 Transact-SQL语法约定](#)

[7.2.2 Transact-SQL的新增功能和增强功能](#)

[7.2.3 运算符](#)

[7.2.4 数据类型](#)

[7.2.5 函数](#)

[7.2.6 Transact-SQL变量](#)

[7.2.7 表达式](#)

[7.2.8 控制流](#)

[7.2.9 保留关键字](#)

[7.2.10 批处理](#)

[7.2.11 Transact-SQL游标](#)

[练习题3](#)

[上机实习2](#)

7.2.1 Transact-SQL语法约定

T-SQL是使用SQL Server的核心，与SQL Server实例通信的所有应用程序都通过将T-SQL语句发送到服务器运行（不考虑应用程序的用户界面）来实现使用SQL Server及其数据的，应该说认真学习好T-SQL是深入掌握SQL Server的必经之路。表7-1反映出构成T-SQL的主要内容。

SQL 元素	说明
标识符	表、视图、列、数据库和服务器等对象的名称。
数据类型	定义数据对象（如列、变量和参数）所包含的数据的类型。大多数T-SQL语句并不显式引用数据类型，但它们的结果受语句中所引用对象的数据类型之间的交互操作影响。
函数	语法元素，可以接受零个、一个或多个输入值，并返回一个标量值或表格形式的一组值。示例包括将多个值相加的 SUM 函数、确定两个日期之间相差多少个时间单位的DATEDIFF函数、获取SQL Server 2005实例名称的 @@SERVERNAME 函数或在远程服务器上执行T-SQL语句并检索结果集的OPENQUERY函数。
表达式	SQL Server 可以解析为单个值的语法单位。表达式的示例包括常量、返回单值的函数、列或变量的引用。
表达式中的运算符	与一个或多个简单表达式一起使用，构造一个更为复杂的表达式。例如，表达式 PriceColumn * 1.1 中的乘号 (*)使价格提高百分之十。
注释	插入到T-SQL语句或脚本中、用于解释语句作用的文本段。SQL Server 不执行注释。
保留关键字	保留下来供SQL Server使用的词，不应用作数据库中的对象名。

7.2.1 Transact-SQL语法定约定

1、语法定约定

下表7-2列出了T-SQL 参考的语法关系图中使用的约定，并进行了说明。

表7-2 T-SQL参考的语法定约定

约定	用于
UPPERCASE (大写)	T-SQL 关键字。
Italic	用户提供的T-SQL语法的参数。
bold (粗体)	数据库名、表名、列名、索引名、存储过程、实用工具、数据类型名以及必须按所显示的原样键入的文本。
下划线	指示当语句中省略了包含带下划线的值的子句时应用的默认值。
(竖线)	分隔括号或大括号中的语法项。只能选择其中一项。
[] (方括号)	可选语法项。不要键入方括号。
{ } (大括号)	必选语法项。不要键入大括号。
[, ... n]	指示前面的项可以重复 n 次。每一项由逗号分隔。
[... n]	指示前面的项可以重复 n 次。每一项由空格分隔。
[;]	可选的T-SQL语句终止符。不要键入方括号。
<label> ::=	语法块的名称。此约定用于对可在语句中的多个位置使用的过长语法段或语法单元进行分组和标记。可使用的语法块的每个位置由括在尖括号内的标签指示：<label>。

7.2.1 Transact-SQL语法约定

2、多部分名称

除非另外指定，否则，所有对数据库对象名的T-SQL引用可以是由四部分组成的名称，格式如下：

[server_name. [database_name]. [schema_name] | database_name.

[schema_name] | schema_name.]object_name

其中：server_name：指定链接的服务器名称或远程服务器名称；database_name：如果对象驻留在SQL Server的本地实例中，则指定SQL Server数据库的名称。如果对象在链接服务器中，则database_name将指定OLE DB目录；schema_name：如果对象在SQL Server数据库中，则指定包含对象的架构的名称。如果对象在链接服务器中，则schema_name将指定OLE DB架构名称；object_name：对象的名称。

7.2.1 Transact-SQL语法约定

若要省略中间节点，请使用句点来指示这些位置。表7-3显示了对象名的有效格式。

表7-3 对象名的有效格式

对象引用格式	说明
server.database.schema.object	四个部分的名称
server.database..object	省略架构名称
server..schema.object	省略数据库名称
server...object	省略数据库和架构名称
database.schema.object	省略服务器名
database..object	省略服务器和架构名称
schema.object	省略服务器和数据库名称
object	省略服务器、数据库和架构名称

7.2.1 Transact-SQL语法约定

3、代码示例约定

除非专门说明，否则，在T-SQL参考中提供的示例都已使用 SQL Server Management Studio 及其以下选项的默认设置进行了测试：ANSI_NULLS、ANSI_NULL_DFLT_ON、ANSI_PADDING、ANSI_WARNINGS、CONCAT_NULL_YIELDS_NULL、QUOTED_IDENTIFIER。

T-SQL 参考中的大多数代码示例都已在运行区分大小写排序顺序的服务器上进行了测试。测试服务器通常运行 ANSI/ISO 1252 代码页。

7.2.2 Transact-SQL的新增功能和增强功能

SQL Server 2005扩展并增强T-SQL语法以支持新增功能和特性。本SQL Server版本中提供的T-SQL增强功能分为以下几类：1) 示例数据库增强功能；2) T-SQL 数据类型；3) 数据库控制台命令(DBCC)语句；4) 数据定义语言语句；5) 数据操作语言语句；6) 元数据；7) 复制系统存储过程；8) 其它语句。

7.2.3 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、字符串串联运算符、按位运算符、一元运算符和比较运算符。

7.2.3 运算符

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类别的一个或多个数据类型。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。如：

```
select getdate(), getdate()-10, cast((getdate()-
cast('2006-01-18' as datetime)) as
int), cast((getdate()-('2006-01-18')) as int)
```

--查询当前日期、10天前日期、日期间隔天数等。

7.2.3 运算符

2、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，见表7-4。其使用详见第6章。

表7-4 逻辑运算符

运算符	含义
<u>ALL</u>	如果一组的比较都为TRUE，那么就为TRUE。
<u>AND</u> 或&&	如果两个布尔表达式都为TRUE，那么就为 TRUE。
<u>ANY</u> 或 <u>SOME</u>	如果一组的比较中任何一个为TRUE，那么就为TRUE。
<u>BETWEEN</u>	如果操作数在某个范围之内，那么就为TRUE。
<u>EXISTS</u>	如果子查询包含一些行，那么就为 TRUE。
<u>IN</u>	如果操作数等于表达式列表中的一个，那么就为TRUE。
<u>LIKE</u>	如果操作数与一种模式相匹配，那么就为TRUE。
<u>NOT</u> 或!	对任何其它布尔运算符的值取反。
<u>OR</u> 或	如果两个布尔表达式中的一个为TRUE，那么就为TRUE。

7.2.3 运算符

3、赋值运算符

等号(=)是唯一的T-SQL赋值运算符。在以下示例中，将创建一个@MyCounter变量，然后赋值运算符将@MyCounter设置为表达式返回的值。

```
DECLARE @MyCounter INT; SET @MyCounter = 10;
```

4、字符串串联运算符

加号(+)是字符串串联运算符，可以用它将字符串串联起来。其它所有字符串操作都使用字符串函数（如 [SUBSTRING](#)）进行处理。

5、位运算符

位运算符在两个表达式之间执行位操作，这两个表达式可以为整数数据类型类别中的任何数据类型。见表7-5。

位运算符的操作数可以是整数或二进制字符串数据类型类别中的任何数据类型（image 数据类型除外），但两个操作数不能同时是二进制字符串数据类型类别中的某种数据类型。表7-6显示所支持的操作数数据类型。

7.2.3 运算符

表7-6 位运算符的操作数要求

左操作数	右操作数
<u>binary</u>	int、smallint或tinyint
<u>bit</u>	int、smallint、tinyint或bit
<u>int</u>	int、smallint、tinyint、binary或varbinary
<u>smallint</u>	int、smallint、tinyint、binary或varbinary
<u>tinyint</u>	int、smallint、tinyint、binary或varbinary
<u>varbinary</u>	int、smallint或tinyint

表7-5 位运算符

运算符	含义
<u>&</u> (位与)	位与 (两个操作数)
<u> </u> (位或)	位或 (两个操作数)
<u>^</u> (位异或)	位异或 (两个操作数)

7.2.3 运算符

6、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 数据类型类别中的任何一种数据类型。具体为：+（正）：数值为正；-（负）：数值为负；~（位非）：返回数字的非。其中：+（正）和-（负）运算符可以用于 numeric 数据类型类别中任一数据类型的任意表达式。~（位非）运算符只能用于整数数据类型类别中任一数据类型的表达式。

7、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）。

7.2.3 运算符

8、运算符优先级和结合性

表达式计算器支持的运算符集中的每个运算符在优先级层次结构中都有指定的优先级，并包含一个计算方向。运算符的计算方向就是运算符结合性。具有高优先级的运算符先于低优先级的运算符进行计算。如果复杂的表达式有多个运算符，则运算符优先级将确定执行操作的顺序。执行的顺序可能对结果值有明显的影响。某些运算符具有相等的优先级。如果表达式包含多个具有相等的优先级的运算符，则按照从左到右或从右到左的方向进行运算。

7.2.3 运算符

下表7-7按从高到低的顺序列出了运算符的优先级。同层运算符具有相等的优先级。

表7-7 运算符的优先级与结合性

运算符	运算类型	结合性	运算符	运算类型	结合性
()	表达式	从左到右	&	位与	从左到右
-, !, ~	一元	从右到左	^	位异或	从左到右
cast as	一元	从右到左		位或	从左到右
*, /, %	乘法性的	从左到右	&&	逻辑与	从左到右
+, -	加法性的	从左到右		逻辑或	从左到右
<, >, <=, >=	关系	从左到右	?:	条件表达式	从右到左
==, !=	等式	从左到右			

7.2.4 数据类型

在 SQL Server 2005 中，每个列、局部变量、表达式和参数都具有一个相关的数据类型。数据类型是一种属性，用于指定对象可保存的数据的类型：整数数据、字符数据、货币数据、日期和时间数据、二进制字符串等。

SQL Server 提供系统数据类型集，该类型集定义了可与 SQL Server 一起使用的所有数据类型。您还可以在 T-SQL 或 .NET Framework 中定义自己的数据类型。别名数据类型基于系统提供的数据类型。用户定义类型从您使用 .NET Framework 支持的编程语言之一创建的类的方法和运算符中获取它们的特征。

7.2.4 数据类型

SQL Server 2005中的数据类型归纳为下列类别：精确数字、Unicode 字符串、近似数字、二进制字符串、日期和时间、其它数据类型、字符串。

精确数字：

bigint, decimal, int, numeric, smallint, money, tinyint, smallmoney, bit

近似数字：float, real；日期和时间：datetime, smalldatetime

字符串1：char, text, varchar, Unicode；

字符串2：nchar, ntext, nvarchar

二进制字符串：binary, image, varbinary

其它类型：

cursor, timestamp, sql_variant, uniqueidentifier, table, xml

在 SQL Server 2005中，根据其存储特征，某些数据类型被指定为属于下列各组：1) 大值数据类型：varchar(max), nvarchar(max) 和 varbinary(max)；2) 大型对象数据类型：

text, ntext, image, varchar(max), nvarchar(max), varbinary(max) 和 xml。

7.2.4 数据类型

1、数据类型优先级

当两个不同数据类型的表达式用运算符组合后，数据类型优先级规则指定将优先级较低的数据类型转换为优先级较高的数据类型。如果此转换不是所支持的隐式转换，则返回错误。当两个操作数表达式具有相同的数据类型时，运算的结果便为该数据类型。

SQL Server 2005对数据类型使用以下优先级顺序：用户定义数据类型（最高）

→sql_variant→xml→datetime→smalldatetime→float→real→decimal→money→smallmoney→bigint→int→smallint→tinyint→bit→ntext→text→image→timestamp→uniqueidentifier→nvarchar→nchar→varchar→char→varbinary→binary（最低）

7.2.4 数据类型

可以按以下方案转换数据类型：

当一个对象的数据移到另一个对象，或两个对象之间的数据进行比较或组合时，数据可能需从一个对象的数据类型转换为另一个对象的数据类型。

将T-SQL结果列、返回代码或输出参数中的数据移到某个程序变量中时，必须将这些数据从SQLServer 2005系统数据类型转换成该变量的数据类型。

可以隐式或显式转换数据类型：1) 隐式转换：SQL Server会自动将数据从一种数据类型转换为另一种数据类型。例如，将smallint与int进行比较时，在比较之前smallint会被隐式转换为int；2) 显式转换：使用CAST或CONVERT函数来实现类型的显式转换。

7.2.4 数据类型

3、数据类型转换行为

以下说明下列数据类型所表现出来的转换行为：binary和varbinary数据；float和real数据；bit数据；money数据；字符数据；decimal和numeric数据；datetime和smalldatetime数据；使用OLE自动化存储过程转换数据类型。

7.2.4 数据类型

4、排序规则优先顺序

排序规则优先顺序也称为排序规则强制规则，用于确定：1) 计算结果为字符串的表达式的最最终结果排序规则；2) 区分排序规则的运算符所使用的排序规则，这些运算符使用字符串输入但不返回字符串，如LIKE和IN。

排序规则的优先顺序规则只应用于下列字符串数据类型：char、varchar、text、nvarchar、nvarchar和ntext。具有其它数据类型的对象不参与排序规则计算。

在数据类型转换之后确定排序规则优先顺序（具体略）。生成结果排序规则的操作数可以与提供最终结果数据类型的操作数不同。例如，请看下面的批处理：

```
CREATE TABLE TestTb(PrimaryKey int PRIMARY KEY, CharCol
char(10) COLLATE French_CI_AS)
```

```
SELECT * FROM TestTb WHERE CharCol LIKE N'abc'
```

简单表达式N'abc'的Unicode数据类型有更高的数据类型优先级。因此，所生成的表达式将Unicode数据类型分配给N'abc'。但是，表达式CharCol具有隐式排序规则标签，而N'abc'具有级别更低的强制标签，即强制默认。因此，所使用的排序规则是CharCol的French_CI_AS 排序规则。

7.2.5 函数

SQL Server 2005提供了许多内置函数，同时也允许您创建用户定义函数。

函数类型分类表见表7-10。其中最常用的标量函数有：配置函数、游标函数、日期和时间函数、数学函数、元数据函数、安全函数、字符串函数、系统函数、系统统计函数、文本和图像函数等10类。

表7-10 函数分类表

函数	说明
行集函数	返回可在SQL语句中像表引用一样使用的对象。
聚合函数	对一组值进行运算，但返回一个汇总值。
排名函数	对分区中的每一行均返回一个排名值。
标量函数	对单一值进行运算，然后返回单一值。只要表达式有效，即可使用标量函数。

7.2.5 函数

1、系统函数

下列函数对SQL Server 2005中的值、对象和设置进行操作并返回有关信息。它们是：APP_NAME、CASE表达式、CAST和CONVERT、COALESCE、CURRENT_TIMESTAMP、CURRENT_USER、DATALENGTH、@@ERROR、ERROR_LINE、ERROR_MESSAGE、ERROR_NUMBER、ERROR_PROCEDURE、ERROR_SEVERITY、ERROR_STATE、fn_helpcollations、HOST_ID、HOST_NAME、@@IDENTITY、ISDATE、ISNULL、ISNUMERIC、NEWID、PARSENAME、@@ROWCOUNT、SERVERPROPERTY、SESSIONPROPERTY、SESSION_USER、SYSTEM_USER、USER_NAME等。

7.2.5 函数

2、常用函数介绍

1. 统计函数

在SQL Server 2005中的统计函数如下：

(1) STDEV: STDEV函数返回给定表达式中所有值的统计标准偏差。其语法格式如下：STDEV(expression), expression是数字表达式。不允许进行聚合函数运算和子查询。

expression是精确数字或近似数字数据类型分类（bit数据类型除外）的表达式。它的返回类型是float。其它统计函数参数与返回类型情况相同。其它统计函数语法格式类似。

(2) STDEVP: STDEVP函数返回给定表达式中所有值的填充统计标准偏差。

例如：USE AdventureWorks; SELECT STDEVP(Bonus) FROM Sales.SalesPerson

(3) VAR: VAR函数返回给定表达式中所有值的统计方差。

(4) VARP: VARP函数返回给定表达式中所有值的填充的统计方差。

7.2.5 函数

2. 算术函数

算术函数（例如 ABS、CEILING、DEGREES、FLOOR、POWER、RADIANS和SIGN）返回与输入值相同数据类型的值。三角函数和其它函数（包括 EXP、LOG、LOG10、SQUARE和SQRT）将输入值投影到float并返回float值。

除了RAND外，所有数学函数都是确定性函数。每次用一组特定输入值调用它们时，所返回的结果相同。仅当指定种子参数时，RAND才具有确定性。

7.2.5 函数

3. 字符串函数

字符串函数用于对字符和二进制字符串进行各种操作，它们返回对字符数据进行操作后得到的值。以下是这些字符串函数的函数名：ASCII、NCHAR、SOUNDEX、CHAR、PATINDEX、SPACE、CHARINDEX、QUOTENAME、STR、DIFFERENCE、REPLACE、STUFF、LEFT、REPLICATE、SUBSTRING、LEN、REVERSE、UNICODE、LOWER、RIGHT、UPPER、LTRIM、RTRIM。

7.2.5 函数

4. 数据类型转换函数

在一般情况下，SQL Server会自动完成数据类型的转换，例如：可以直接将字符数据类型或表达式与DATETIME数据类型或表达式比较；当表达式中用了INTEGER，SMALLINT或TINYINT时，SQL Server也可将INTEGER数据类型或表达式转换为SMALLINT数据类型或表达式，这称为隐式转换。如果不能确定SQL Server是否能完成隐式转换或者使用了不能隐式转换的其它数据类型，就需要使用数据类型转换函数做显式转换了。此类函数有两种：

(1) CAST函数：CAST的语法格式如下：CAST(expression AS data_type)

(2) CONVERT函数：CONVERT的语法格式如下：
 CONVERT(data_type[(length)],
 expression[, style])

7.2.5 函数

5. 日期函数

日期函数用来操作DATETIME和SMALLDATETIME类型的数
据执行算术运算。与其它函数一样，可以在SELECT语句的
SELECT和WHERE子句以及表达式中使用日期函数。其使用方
法如下：日期函数(参数)。其中参数的个数随函数的不同
而不同。

(1) DAY函数：DAY函数返回代表指定日期的天的日期部
分的整数。

其语法格式如下：DAY(date)。其返回类型为int，此函
数等价于DATEPART(dd, date)。

例7-20 此示例返回从日期 01/28/2007中返回的天数。

```
SELECT DAY('01/28/2007') AS '几号'
```

(2) MONTH函数：MONTH函数返回代表指定日期月份的整
数。

7.2.5 函数

(3) YEAR函数: YEAR函数返回表示指定日期中的年份的整数。

其语法格式如下: YEAR(date), 参数date表示datetime或smalldatetime类型的表达式。其返回类型为int。此函数等价于DATEPART(yy, date)。

例7-22 本例从日期01/28/2007中返回年份数: SELECT "年份"
"=YEAR('01/28/2007')

(4) DATEADD函数: DATEADD函数表示在向指定日期加上一段时间的基础上, 返回新的 datetime 值。其语法格式如下:
DATEADD(datepart, number, date)

其中参数datepart是规定应向日期的哪一部分增加新值的参数。下表7-13列出了SQL Server识别的日期部分和缩写。

(5) DATEDIFF函数: DATEDIFF函数返回跨两个指定日期的日期和时间边界数。

其语法格式如下: DATEDIFF(datepart, startdate, enddate)

例7-23 此示例确定在 pubs 数据库中标题发布日期和当前日期期间的天数。

```
USE pubs; SELECT DATEDIFF(day, pubdate, getdate()) AS  
no_of_days FROM titles
```

7.2.5 函数

(6) DATENAME函数：DATENAME函数返回代表指定日期的指定日期部分的字符串。

其语法格式如下：DATENAME(datepart, date)

(7) DATEPART函数：DATEPART函数表示返回代表指定日期的指定日期部分的整数。

其语法格式如下：DATEPART(datepart, date)。DATEPART函数以整数值的形式返回日期的指定部分，此部分由datepart来指定。DATEPART(dd, date)等同于DAY(date)；DATEPART(mm, date)等同于MONTH(date) DATEPART(yy, date)等同于YEAR(date)。

(8) GETDATE函数：GETDATE函数的语法格式如下：GETDATE()。GETDATE函数以DATETIME的缺省格式返回系统当前的日期和时间，它常作为其它函数或命令的参数使用。

(9) GETUTCDATE函数：其语法格式如下：GETUTCDATE()。GETUTCDATE函数返回表示当前的UTC 时间（通用协调时间或格林尼治标准时间）的 datetime 值。当前的 UTC 时间得自当前的本地时间和运行SQL Server 实例的计算机操作系统中的时区设置。

7.2.5 函数

表7-13 日期函数中datepart参数的取值

日期部分	缩写	取值区段
Year	yy, yyyy	1753-9999年份
Quarter	qq, q	1-4刻
Month	mm, m	1-12月
dayofyear	dy, y	1-366日
Day	dd, d	1-31日
Week	wk, ww	1-54周
weekday	dw	1-7周几
Hour	hh	0-23小时
minute	mi, n	0-59分钟
second	ss, s	0-59秒
millisecond	Ms	0-999毫秒

7.2.5 函数

6. 文本与图像函数

以下标量函数可对文本或图像输入值或列执行操作，并返回有关该值的信息：PATINDEX（见前面介绍）、TEXTVALID、TEXTPTR。

（1）TEXTPTR函数：TEXTPTR函数的语法格式如下：
 TEXTPTR(column)

返回对应于varbinary格式的text、ntext或image列的文本指针值。如果数据类型为TEXT，NTEXT或IMAGE的列没有赋予初值，则TEXTPTR函数返回一个NULL指针。检索到的文本指针值可用于READTEXT、WRITETEXT和UPDATETEXT语句。

（2）TEXTVALID函数：TEXTVALID函数的语法格式如下：
 TEXTVALID('table.column',

text-pointer)。TEXTVALID函数用于检查指定的文本指针是否有效。如果有效，则返回1，无效则返回0，如果列未赋予初值，则返回NULL值。请注意，text列的标识符必须包含表名。在没有有效的文本指针的情况下，不能使用UPDATETEXT、WRITETEXT或READTEXT。

7.2.5 函数

7. 用户自定义函数

从SQL Server 2000开始，用户可以自定义函数，在SQL Server 2005中用户自定义函数作为一个数据库对象来管理，可以在Management Studio查询窗口中利用T-SQL命令来创建(CREATE FUNCTION)，修改(ALTER FUNCTION)和删除(DROP FUNCTION)它。举一例如下：

```

CREATE FUNCTION dbo.DaysBetweenDates (@D1
datetime, @D2 datetime) RETURNS INT
AS BEGIN RETURN (SELECT cast((@d2-@d1) as int))
END -- 定义

select dbo.DaysBetweenDates (getdate(), cast(' 2006-
01-28' as datetime)) -- 使用
    
```

7.2.6 Transact-SQL变量

T-SQL局部变量是可以保存单个特定类型数据值的对象。批处理和脚本中的变量通常用于：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) 保存数据值以供控制流语句测试；3) 保存存储过程返回代码要返回的数据值或函数返回值。

某些T-SQL系统函数的名称以两个at符号(@@)打头。在SQL Server的早期版本中，@@functions被称为全局变量，在SQL Server 2005中不再这样认为。因为它们不具备变量的行为，它们的语法遵循函数的规则。为此@@functions被称为是系统函数。当然使用@@functions时，它们具有全局变量的某些特性。

局部变量是在批处理或过程的主体中用DECLARE语句声明的，并用SET或SELECT语句赋值。游标变量可使用此语句声明，并可用于其它与游标相关的语句。所有变量在声明后均初始化为 NULL。DECLARE @local_variable的语法为：

```

DECLARE { { @local_variable [AS] data_type } |
{ @cursor_variable_name CURSOR }
      | { @table_variable_name < table_type_definition
> } } [ ,...n]
<table_type_definition> ::= TABLE ( { <column_definition> | <table_constraint> } [ ,... ] )
    
```

DECLARE 语句通过以下操作初始化T-SQL变量：1) 指定名称。名称的第一个字符必须为一个@；2) 指定系统提供的或用户定义的数据类型和长度。对于数值变量还指定精度和小数位数。对于XML 类型的变量，可以指定一个可选的架构集合；3) 将值设置为NULL。

7.2.6 Transact-SQL变量

变量的作用域就是可以引用该变量的T-SQL语句的范围。变量的作用域从声明变量的地方开始到声明变量的批处理或存储过程的结尾。例如，下面的脚本存在语法错误，因为在一个批处理中引用了在另一个批处理中声明的变量：

```

USE AdventureWorks; DECLARE @MyVariable int;SET @MyVariable = 1;
GO          --中断批处理，表示批处理结束
-- @MyVariable已离开其批处理的作用域，为此使用它的SELECT语句将
发生语法错
SELECT      *          FROM          HumanResources.Employee          WHERE
EmployeeID=@MyVariable;
    
```

变量具有局部作用域，只在定义它们的批处理或过程中可见。在本例中，为执行 `sp_executesql` 创建的嵌套作用域不能访问在更高作用域中声明的变量，从而返回错误。

```

DECLARE @MyVariable int; SET @MyVariable=1;
EXECUTE sp_executesql N' select @MyVariable' --这命令产生一个错误
    
```

那么如何为T-SQL变量设置值呢？第一次声明变量时，其值设置为 `NULL`。若要为变量赋值，请使用 `SET` 语句。这是为变量赋值的首选方法。也可以通过 `SELECT` 语句的选择列表中当前所引用值为变量赋值。

7.2.7 表达式

符号和运算符的一种组合，SQL Server 2005 Database Engine 将处理该组合以获得单个数据值。简单表达式可以是一个常量、变量、列或标量函数。可以用运算符将两个或更多的简单表达式联接起来组成复杂表达式。语法为：

```
{constant|scalar_function|[ alias.]column|local_variable|(expression)|  
(scalar_subquery)|{unary_operator}expression|expression{binary_operator}expression|ranking_windowed_function|aggregate_windowed_function}。其说明见表7-14。
```

7.2.7 表达式

表7-14 表达式语法说明表

术语	定义
constant	表示单个特定数据值的符号。constant 是一个或多个字母数字字符（字母 a-z、A-Z 和数字 0-9），也可以是符号（感叹号 (!)、at 符 (@)、井号 (#) 等等）。字符和日期时间值要用引号括起来，但二进制字符串和数字常量可以不用引号。
scalar_function	一个T-SQL语法单元，用于提供特定服务并返回单值。scalar_function 可以是内置标量函数（如 SUM、GETDATE或CAST 函数），也可以是标量用户定义函数。
[alias.]	由 FORM 子句中的 AS 关键字分配给表的别名或者相关名称。
Column	列的名称。在表达式中只允许有列的名称；不能指定由四部分组成的名称。
local_variable	用户定义变量的名称。有关详细信息，请参阅 DECLARE @local_variable 。
(expression)	本主题中定义的任意一个有效表达式。括号是分组运算符，用于确保先运算括号内表达式中的运算符，然后再将结果与别的表达式组合。
(scalar_subquery)	返回一个值的子查询。例如：SELECT MAX(UnitPrice) FROM Products
{unary_operator}	只有一个数字操作数的运算符：+ 指示正数。- 指示负数。~ 指示一的补数运算符。一元运算符只能用于计算结果数据类型属于数字数据类型类别的表达式。
{binary_operator}	用于定义如何组合两个表达式以得到一个结果的运算符。binary_operator 可以是算术运算符、赋值运算符 (=)、位运算符、比较运算符、逻辑运算符、字符串连接运算符 (+) 或一元运算符。
ranking_windowed_function	任意T-SQL排名函数。
aggregate_windowed_function	任意包含Transact-SQLOVER 子句的聚合函数。

7.2.7 表达式

如果没有支持的隐式或显式转换，则两个表达式将无法组合。

任何计算结果为字符串的表达式排序规则都应遵循排序规则优先顺序规则。

表达式结果：对于由单个常量、变量、标量函数或列名组成的简单表达式，其数据类型、排序规则、精度、小数位数和值就是它所引用的元素的数据类型、排序规则、精度、小数位数和值。

用比较运算符或逻辑运算符组合两个表达式时，生成的数据类型为 Boolean，并且值为下列类型之一：TRUE、FALSE或 UNKNOWN。

用算术运算符、位运算符或字符串运算符组合两个表达式时，生成的数据类型取决于运算符。

由多个符号和运算符组成的复杂表达式的计算结果为单值结果。生成的表达式的数据类型、排序规则、精度和值由进行组合的两个表达式决定，并按每次两个表达式的顺序递延，直到得出最后结果。表达式中元素组合的顺序由表达式中运算符的优先级决定。

7.2.8 控制流

T-SQL提供称为控制流语言的特殊关键字，用于控制T-SQL语句、语句块和存储过程的执行流。这些关键字可用于临时T-SQL语句、批处理和存储过程中。

不使用控制流语言，则各T-SQL语句按其出现的顺序分别执行。控制流语言使用与程序设计相似的构造使语句得以互相连接、关联和相互依存。

当您需要T-SQL进行某种操作时，这些控制流关键字非常有用。例如，当在一个逻辑块中包含多个T-SQL语句时，请使用BEGIN...END语句对。使用IF...ELSE 语句对的情况是：IF（如果）满足某条件，则执行某些语句或语句块，而如果不满足此条件（ELSE 条件）则执行另一条语句或语句块。控制流语句不能跨越多个批处理或存储过程。

以下是流程控制语句基本语法格式及使用的介绍：

7.2.8 控制流

1、BEGIN ... END

BEGIN ... END语句用于将多个T-SQL语句组合为一个逻辑块。当控制流语句执行一个包含两条或两条以上T-SQL语句的语句块时，可以使用BEGIN和END语句。其语法格式如下：

```
BEGIN  
{ 命令行 | 程序块 }  
END
```

BEGIN和END语句必须成对使用，BEGIN或END语句均不能单独使用。BEGIN语句行后为T-SQL语句块。最后，END 语句行指示语句块结束。

BEGIN和END语句主要用于下列情况：1) WHILE 循环需要包含语句块；2) CASE 函数的元素需要包含语句块；3) IF或ELSE子句需要包含语句块。

BEGIN...END 语句块允许嵌套。

7.2.8 控制流

2、IF…ELSE

IF…ELSE的语法格式如下：

```
IF <条件表达式> { <命令行> | <程序块> }  
[ ELSE { <命令行> | <程序块> } ]
```

IF 语句用于条件的测试。结果流的控制取决于是否指定了可选的 ELSE 语句：

(1)指定 IF 而无 ELSE: IF 语句取值为 TRUE 时, 执行 IF 语句后的语句或语句块。IF 语句取值为 FALSE 时, 跳过 IF 语句后的语句或语句块。

(2)指定 IF 并有 ELSE: IF 语句取值为 TRUE 时, 执行 IF 语句后的语句或语句块。然后控制跳到 ELSE 语句后的语句或语句块之后的点。IF 语句取值为 FALSE或NULL 时时, 跳过 IF 语句后的语句或语句块, 而执行 ELSE 语句后的语句或语句块。

7.2.8 控制流

例7-29 本例显示带有语句块的 IF 条件。如果DB原理书的平均价格不低于15元，那么就显示文本：DB原理书的总价高于15元。

```
USE pubs;  
IF (SELECT AVG(price) FROM titles WHERE  
name='DB原理') < 15  
BEGIN  
    PRINT '书价不正确！'  
END  
ELSE PRINT 'DB原理书的总价高于15元'
```


7.2.8 控制流

3、CASE

计算条件列表并返回多个可能结果表达式之一。

CASE 具有两种格式：1)简单 CASE 函数将某个表达式与一组简单表达式进行比较以确定结果；2)CASE 搜索函数计算一组布尔表达式以确定结果。

两种格式都支持可选的 ELSE 参数。CASE的语法格式为：

格式1：CASE <运算式>

WHEN <运算式> THEN <运算式>

...

WHEN <运算式> THEN <运算式>

[ELSE <运算式>]

END

该语句的执行过程是：将CASE后面表达式的值与各WHEN子句中的表达式的值进行比较，如果二者相等，则返回THEN后的表达式的值，然后跳出CASE语句，否则返回ELSE子句中的表达式的值。ELSE子句是可选项。当CASE语句中不包含ELSE子句时，如果所有比较失败，CASE语句将返回NULL。

7.2.8 控制流

例7-30 从学生表S中，选取SNO，SEX，如果SEX为“男”则输出“M”，如果为“女”则输出“F”。

```
SELECT SNO, SEX=CASE SEX
        WHEN '男' THEN 'M'
        WHEN '女' THEN 'F'
END
FROM S
```

7.2.8 控制流

格式2:

CASE

WHEN <条件表达式> THEN <运算式>

...

WHEN <条件表达式> THEN <运算式>

ELSE <运算式>

END

该语句的执行过程是：首先测试WHEN后的表达式的值，如果其值为真，则返回THEN后面的表达式的值；否则测试下一个WHEN子句中的表达式的值。如果所有WHEN子句后的表达式的值都为假，则返回ELSE后表达式的值。如果在CASE语句中没有ELSE子句，则CASE表达式返回NULL。注意：CASE命令可以嵌入到SQL命令中。

7.2.8 控制流

例7-31 从SC表中查询所有同学选课成绩情况，凡成绩为空者输出“缺考”，小于60分的输出“不及格”，60分至70分输出“及格”，70分至90分输出“良好”，大于或等于90分的输出“优秀”。

```
SELECT SNO, CNO, SCORE=
      CASE
        WHEN SCORE IS NULL THEN '未考'
        WHEN SCORE <60 THEN '不及格'
        WHEN SCORE BETWEEN 60 AND 69 THEN '及格'
        WHEN SCORE BETWEEN 70 AND 89 THEN '良好'
        WHEN SCORE >=90 THEN '优秀'
      END
FROM SC
```

7.2.8 控制流

4、WHILE...CONTINUE...BREAK

只要指定的条件为真，则WHILE语句重复语句或语句块。BREAK 语句退出最内层WHILE循环，CONTINUE语句重新开始WHILE循环。如果没有其它行可以处理，则程序可能执行BREAK语句。如果要继续执行代码，则可执行CONTINUE语句。

WHILE...CONTINUE...BREAK的语法格式如下：

```

WHILE <条件表达式>
BEGIN
    { <命令行> | <程序块> }
    [ BREAK ]
    { <命令行> | <程序块> }
    [ CONTINUE ]
    { <命令行> | <程序块> }
END
    
```

7.2.8 控制流

例7-32 判断是否有员工的奖金(规定工资的30%为奖金)少于300, 如果有, 则将所有员工的工资增加500, 直到所有员工的奖金都多于300或有员工的工资超过了3000。

```

WHILE EXISTS (SELECT * FROM EMPLOYEE WHERE
SALARY*0.3<300)
BEGIN
    UPDATE EMPLOYEE SET SALARY=SALARY+500
    IF (SELECT MAX(SALARY) FROM EMPLOYEE) > 3000
        BREAK
    ELSE
        CONTINUE
END
    
```

7.2.8 控制流

5、WAITFOR

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，阻止执行批处理、存储过程或事务。

```

WAITFOR {DELAY 'time_to_pass' | TIME
'time_to_execute' | (receive_statement) [, TIMEOUT timeout]}
    
```

WAITFOR命令用来暂时停止程序执行，直到所设定的等待时间已过或所设定的时间已到才继续往下执行。其中“时间”必须为DATETIME类型的数据，但不能包括日期。

各关键字含义如下：1) DELAY：用来设定等待的时间，最多可达24小时；2) TIME：用来设定等待结束的时间点。

例7-33 等待1小时3分24秒后才执行SELECT语句。

```

WAITFOR DELAY '01:03:24' ; SELECT * FROM S;
    
```


7.2.8 控制流

6. GOTO

GOTO命令用来改变程序执行的流程，使程序跳到标有标识符的指定的程序行再继续往下执行。作为跳转目标的标识符可为数字与字符的组合。但必须以“:”结尾。在GOTO命令行，标识符后不必跟“:”。GOTO语句的语法格式如下：

<标识符>:

{ <命令行> | <程序块> }

GOTO <标识符>

例7-34 如果员工的工资大于10000，那么将跳过其它语句，而直接降低执行降低工资的语句。

DECREASE_SALARY:

.....

IF (SELECT MAX(SALARY) FROM EMPLOYEE)>10000

GOTO DECREASE_SALARY

7.2.8 控制流

7、RETURN

从查询或过程中无条件退出。RETURN 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 之后的语句是不执行的。其语法为：

```
RETURN [integer_expression]
```

RETURN命令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它程序。在括号内可指定一个返回值。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，如：

- | | | |
|--------------|--------------------|-------------|
| 0 程序执行成功 | -1 找不到对象 | -2 数据类型错误 |
| -3 死锁 | -4 违反权限原则 | -5 语法错误 |
| -6 用户造成的一般错误 | -7 资源错误 | -8 非致命的内部错误 |
| -9 已达到系统的极限 | -10,-11 致命的内部不一致错误 | |
| -12 表或指针破坏 | -13 数据库破坏 | -14 硬件错误 |

如果运行过程产生了多个错误，则返回绝对值最大的数值。

7.2.8 控制流

8、其它命令

1. **BACKUP**: 该命令用于将数据库内容或其它事务处理日志备份到存储介质上（如软盘，硬盘，磁带等）。

2. **CHECKPOINT**: 将当前数据库的全部脏页写入磁盘。“脏页”是已输入缓存区高速缓存且已修改但尚未写入磁盘的数据页。**CHECKPOINT** 可创建一个检查点，在该点保证全部脏页都已写入磁盘，从而在以后的恢复过程中节省时间。语法为：

```
CHECKPOINT [checkpoint_duration]
```

3. **DBCC**: T-SQL 编程语言提供 **DBCC** 语句作为 SQL Server 2005 的数据库控制台命令。数据库控制台命令语句可分为以下类别：1) 维护：对数据库、索引或文件组进行维护的任务；2) 杂项：杂项任务，如启用跟踪标志或从内存中删除 DLL；3) 信息：收集并显示各种类型信息的任务；4) 验证：对数据库、表、索引、目录、文件组或数据库页的分配进行的验证操作。

7.2.8 控制流

4. DECLARE: 在批处理或过程的正文中用 DECLARE 语句声明各种变量, 并用 SET或SELECT 语句给其指派值。游标变量可通过该语句声明, 并且可用在其它与游标相关的语句中。所有变量在声明后均初始化为NULL。

```

DECLARE { { @local_variable [AS] data_type } | {
@cursor_variable_name CURSOR }
| { @table_variable_name < table_type_definition
> } } [,...n]

```

```

<table_type_definition> ::= TABLE
({ <column_definition> | <table_constraint> } [,...])

```

变量类型可为系统定义的类型或用户定义的类型, 但不能为TEXT, NTEXT和IMAGE类型。CURSOR指明变量是局部的游标变量。如果变量为字符型, 那么在data_type表达式中应指明其最大长度, 否则系统认为其长度为1。

7.2.8 控制流

5. EXECUTE: EXECUTE命令用来执行T-SQL批中的命令字符串、字符串或执行下列模块之一: 系统存储过程、用户定义存储过程、标量值用户定义函数或扩展存储过程。

SQL Server 2005扩展了EXECUTE 语句, 以使其可用于向链接服务器发送传递命令。此外, 还可以显式设置执行字符串或命令的上下文。语法如下:

1) 执行存储过程或函数

```
{ EXEC | EXECUTE } [ { [ @return_status= ] { module_name [
;number ] | @module_name_var } [ [ @parameter= ] { value | @variable
[ OUTPUT ] | [ DEFAULT ] } ] [ , ... n ] [ WITH RECOMPILE ] } [ ; ]
```

2) 执行字符串命令

```
{ EXEC | EXECUTE } ( { @string_variable | [ N ] ' tsql_string ' } [ +
... n ] ) [ AS { LOGIN | USER } = ' name ' ] [ ; ]
```

3) 向链接服务器发送传递命令

```
{ EXEC | EXECUTE } ( { @string_variable | [ N ] ' command_string '
} [ + ... n ] [ { , { value | @variable [ OUTPUT ] } } [ ... n ] ] ) [ AS
{ LOGIN | USER } = ' name ' ] [ AT linked_server_name ] [ ; ]
```

7.2.8 控制流

6. KILL: KILL命令用于终止某一过程的执行。默认情况下, sysadmin和processadmin 固定数据库角色的成员具有KILL的默认权限, KILL权限不可转让。

语法: KILL {spid|UOW} [WITH STATUSONLY]

7. PRINT: 向客户端返回用户定义消息。语法: PRINT msg_str | @local_variable | string_expr

8. RAISERROR: 生成错误消息并启动会话的错误处理。

9. READTEXT: 读取text、ntext或image列中的text、ntext或image值

10. RESTORE: RESTORE命令用来将数据库或其它事务处理日志备份文件由存储介质恢复到SQL Server系统中。

11. SELECT: SELECT命令可用于给变量赋值

12. SET: (1) 用于给局部变量赋值。(2) 用于用户执行SQL命令时, 处理选项的设定。

13. SHUTDOWN: SHUTDOWN 尝试关闭 SQL Server

14. WRITETEXT: 允许对现有的 text、ntext或image 列进行无日志记录的交互式更新。

15. USE: 将数据库上下文更改为指定数据库或数据库快照。

7.2.9 保留关键字

SQL Server 2005使用保留关键字来定义、操作或访问数据库。保留关键字是 SQL Server 使用的T-SQL语言语法的一部分，用于分析和理解T-SQL语句和批处理。尽管在T-SQL脚本中使用 SQL Server 保留关键字作为标识符和对象名在语法上是可行的，但规定只能使用分隔标识符，用中括号括起来，如 “[USER]”。

SQL Server 2005保留关键字约有174个，请你自己从帮助中查阅。

7.2.10 批处理

批处理是包含一个或多个T-SQL语句的组，从应用程序一次性地发送到 SQL Server 2005进行执行，因此可以节省系统开销。SQL Server 将批处理的语句编译为一个可执行单元，称为执行计划，批处理的结束符为“GO”。

编译错误（如语法错误）可使执行计划无法编译。因此未执行批处理中的任何语句。

运行时错误（如算术溢出或违反约束）会产生以下两种影响之一：1）大多数运行时错误将停止执行批处理中当前语句和它之后的语句；2）某些运行时错误（如违反约束）仅停止执行当前语句。而继续执行批处理中其它所有语句。

在遇到运行时错误之前执行的语句不受影响。唯一的例外是如果批处理在事务中而且错误导致事务回滚。在这种情况下，回滚运行时错误之前所进行的未提交的数据修改。

7.2.10 批处理

1、关于注释

在T-SQL中可使用两类注释符：1)ANSI标准的注释符“--”用于单行注释；2)与C语言相同的程序注释符号，即“/*.....*/”，“/*”用于程序注释开头，“*/”用语程序注释结尾，可以在程序中多行文字标示为注释。

批中的注释没有最大长度限制。一条注释可以包含一行或多行。下面是一些有效注释的示例。

```
USE AdventureWorks;
```

```
-- 单行注释.
```

```
SELECT EmployeeID, Title FROM HumanResources.Employee;
```

```
GO
```

```
/* 多行注释的第一行
```

```
   多行注释的第二行   */
```

```
SELECT Name, ProductNumber, Color FROM Production.Product;
```

```
GO
```

```
-- 在调试T-SQL命令时使用注释
```

```
SELECT ContactID, /* FirstName, */ LastName FROM Person.Contact;
```

```
-- 在代码行后使用注释
```

```
USE AdventureWorks;
```

```
UPDATE Production.Product SET ListPrice=ListPrice *0.9; -- 降低  
价格，赢得市场
```

7.2.11 Transact-SQL游标

T-SQL 游标主要用于存储过程、触发器和 T-SQL脚本中，它们使结果集的内容可用于其它 T-SQL 语句。详细情况请参阅相关章节，此处略。

练习题3

一、填空题

1) 在SQL Server 2005中的标识符中有一种限定标识符，它是通过_____或_____符号来限定标识符的。

2) SQL Server 2005支持两种类型的注释，它们是_____和_____，它们的符号分别是_____和_____。

3) T-SQL的系统数据类型有哪几类_____？共多少个_____？

4) 在T-SQL的模式匹配中，使用哪个符号表示匹配任意长度的字符串_____？

练习题3

二、简答题

- 1、简述T-SQL语言的概况。
- 2、什么是标识符？T-SQL中有几种标识符？他们的区别是什么？
- 3、如何表示数据库对象名？
- 4、什么是局部变量？什么是全局变量？如何表示它们？
- 5、以下变量名中，哪些是合法的变量名，哪些是不合法的变量名？
A1, 1a, @x, @@y, &变量1, @姓名, 姓名, #m, ##n, @@@abc##, @my_name
- 6、SQL Server 2005所使用的运算符类别有哪些？
- 7、计算下列表达式：(1) $9-3*5/2+6\%4$ (2) $5\&2|4$ (3) '你们'+'好' (4) ~ 10
(5) DECLARE @d SMALLDATETIME; SET @d='2007-1-26'; SELECT @d+10, @d-10
- 8、举例说明，如果表达式@b=@b - @c 成立，则变量@b、@b、@c可以是什么数据类型？
- 9、用CASE函数，求当前日期是否是闰年？
- 10、符号“=”可以是关系运算符等于，也可以是赋值运算符。那么什么情况下是关系运算符，什么情况下是赋值运算符？
- 11、什么是批处理？使用批处理有何限制？批处理的结束符是什么？
- 12、用T-SQL流程控制语句编写程序，求两个数的最大公约数和最小公倍数。
- 13、用T-SQL流程控制语句编写程序，求斐波那契数列中小于100的所有数（斐波那契数列1, 2, 3, 5, 8, 12, ...）。
- 14、计算下列表达式：
1) $ABS(-5.5)+SQRT(16)*SQUARE(2)$ 2) $ROUND(456.789, 2)-ROUND(345.678, -2)$
3) SUBSTRING(REPLACE('北京大学', '北京', '清华'), 3, 2)
4) 计算今天距离2007年7月7日，还有多少年，多少月，多少日？
- 15、定义一个用户标量函数，用以实现判断并返回三个数中的最大数。
- 16、定义一个用户标量函数，用以实现判断并返回一个日期时间数据位于该年的第几天。
- 17、一般SQL脚本文件的后缀是什么？SQL脚本的执行结果有哪几种形式？

上机实习2

实验目的

1) 了解T-SQL 语言的各元素；2) 实践T-SQL 语言的基本概念、语法规式、运算符和表达式以及基本语句和函数等的使用。

实验内容

- 1、在Management Studio中的查询窗口中，利用SELECT 语句（形如：SELECT XXX），检验练习题中常量、变量、函数或表达式等的值。
- 2、使用WHILE语句求1-100之间的累加和，并输出结果。再通过自定义函数实现之。
- 3、请写出实现下面查询操作的T-SQL语句：从SQL Server 2005实例数据库AdventureWorks中，查询出销售商品编号(ProductNumber)为BK-M68B-42的雇员的姓名(LastName和FirstName) (通过游标程序或SELECT查询命令实现，本题选做)。
- 4、实践本章各例题，查验各题执行情况。