第7章 SQL Server 2000 & 2005 数据库管理系统

本章要点

本章讲述了利用SQL Server 2000进行数据库管理与使用的方法以及最新版本SQL Server 2005的系统概述。主要介绍了SQL Server 2000企业管理器、查询分析器等的基本使用，创建与管理数据库、数据表、视图、索引、存储过程、触发器等的方法，数据库备份与恢复方法，以及利用Transact-SQL进行数据库服务器端程序设计的技能等。在SQL Server 2000中创建与管理数据库服务器对象的方法与技能，同样适用于SQL Server 2005最新版数据库管理系统。数据库主要对象的交互式界面操作方法及基本的Transact-SQL命令操作方法等是我们实践数据库应用技术所必须要努力去掌握的。

本章逻辑结构

本章逻辑结构

本章逻辑结构

本章逻辑结构

本章逻辑结构

7.1 SQL Server 2000的简介

SQL Server 是一种关系数据库，它除了支持传统关系数据库组件（如数据库，表）和特性（如表的join）外，另外也支持当今关系数据库常用的组件，如存储过程，视图等。SQL Server支持关系数据库国际标准语言——SQL（它称为Transact-SQL）。SQL Server另外的一项重要的特点是它支持数据库复制的功能。

SQL Server 2000包含有6种不同的版本 ：

（1）企业版（Enterprise Edition）作为各种企业、单位或组织的数据库服务器使用

（2）标准版（Standard Edition）用于小型的工作组或部门

7.1 SQL Server 2000的简介

（3）个人版（Personal Edition）用于单机系统或客户机

（4）开发者版（Developer Edition）用于程序员开发应用程序时，将SQL Server 2000作为其数据存储区。

（5）Windows CE版 安装于执行Windows CE的个人数字助理上（PDA），它能复制任何SQL Server 2000版本的数据，使Windows CE数据能与公司的主要数据库同步。

（6）评测版 SQL Server 2000评测版是一个功能基本齐全的版本，该版本主要用来评估SQL Server 2000的功能。

7.1 SQL Server 2000的简介

一、SQL Server 2000的新特性

（1）在关系数据库方面的增强

1. XML支持
2. 用户定义函数和新的数据类型
3. 索引视图，索引增强
4. 分布式查询
5. 备份和还原
6. 级联参考完整性约束
8. 排序规则

（2）图形管理工具增强

1.日志传送功能

2.SQL事件探查器增强

7.1 SQL Server 2000的简介

3.SQL查询分析器增强

4.复制数据库向导

二、SQL Server 2000的主要组件

作为一个完善的数据库管理系统，SQL Server 2000提供了一些功能强大，使用方便的数据库管理工具。下面,对这些组件一个简单的介绍.

1 、企业管理器(Enterprise Manager)

企业管理器是SQL Server中最重要的管理工具

2 、查询分析器(Query Analyzer)

查询分析器用于执行Transact-SQL命令等SQL脚本程序，以查询分析或处理数据库中的数据。

3 、 服务管理器(Service Manager)

7.1 SQL Server 2000的简介

服理器用于启动、暂停或停止SQL Server的四种服务，即分布式事务协调器(Distributed Transaction Coordinator,DTC)、MSSQL Server OLAP service、SQL Server和SQL Server Agent。

4、客户端网络实用工具(Client Network Utility)

5、服务器端网络实用工具(Server Network Utility)

服务器端网络实用工具用于配置服务器端的连接、测定网络库的版本信息

6、导入和导出数据(Import and Export Data)

7、在IIS中配置SQL XML支持(Configure SQL XML Support in IIS)

7.1 SQL Server 2000的简介

8、事件探查器(Profiler)

事务探查器的功能是监视SQL Server数据库系统引擎事件，主要用于监听SQL Server系统的运行性能。

9、联机丛书(Book Online)

SQL Server 2000提供了大量的联机文档，用户可以便捷地查到许多很有价值的信息。一个优秀的SQL Server 管理员必然是使用联机文档的高手。

7.1 SQL Server 2000的简介

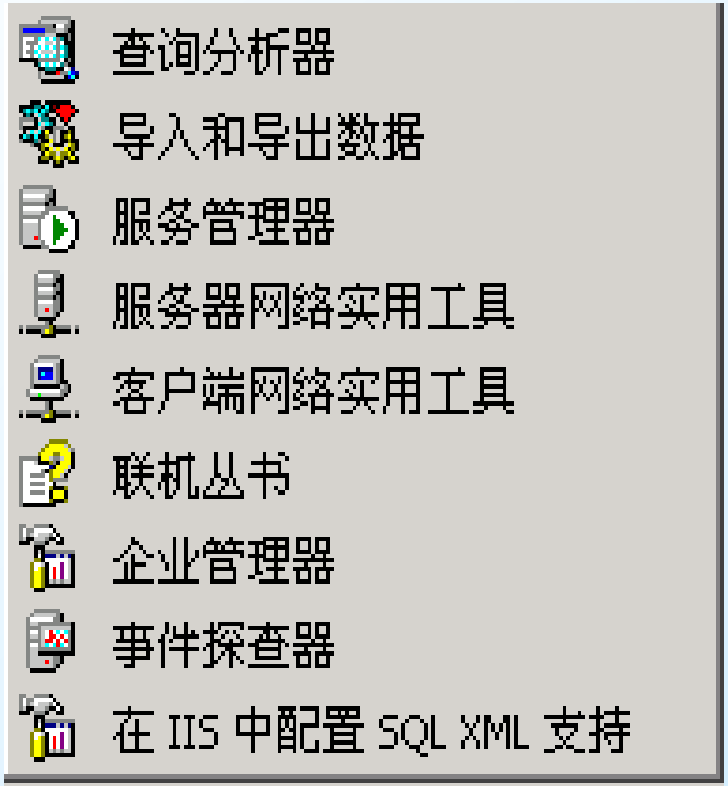
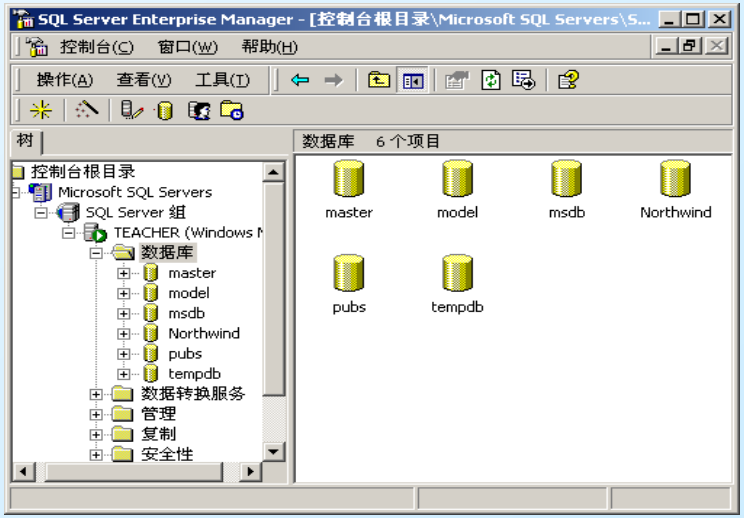


图7.1 SQL Server 2000工具菜单

7.2 企业管理器

- 企业管理器(Enterprise Manager)是SQL SERVER程序组中的最重要的程序之一，是管理服务器和数据库的主要工具。
- 可以从开始菜单命令“开始→程序→Microsoft SQL Server→企业管理器”打开企业管理器。如图7.2



7.2 企业管理器

在企业管理器中包含两个窗口，其中左侧的窗口是以“树状目录”显示的活动窗口，右侧是显示内容的窗口。

1 、树状目录窗口

在左边的树状目录中，根节点是“控制面板根目录”，表示它是所有服务器控制面板的根。

2 、内容窗口

企业管理器右边的窗口为内容窗口。在该窗口中显示的是在树状目录处于“焦点”状态（或选种状态）的条目中包含的内容。企业管理器中的菜单分为上下两行，其中上面一行包括“控制面版”，“窗口”和“帮助”，通过这三个菜单项可以实现退出企业管理器。

7.3 查询分析器

查询分析器（Query Analyzer）是SQL Server提供的使用方便，界面友好的Transact-SQL语句编译工具，是SQL Server 2000客户端应用程序的重要组成部分。

用户可以通过“开始”菜单或从 SQL Server 企业管理器内运行它。还可以通过执行 isqlw 实用工具从命令提示符运行SQL 查询分析器。启动时系统首先打开“连接到SQL Server”对话框，如图7.3所示。

7.3 查询分析器

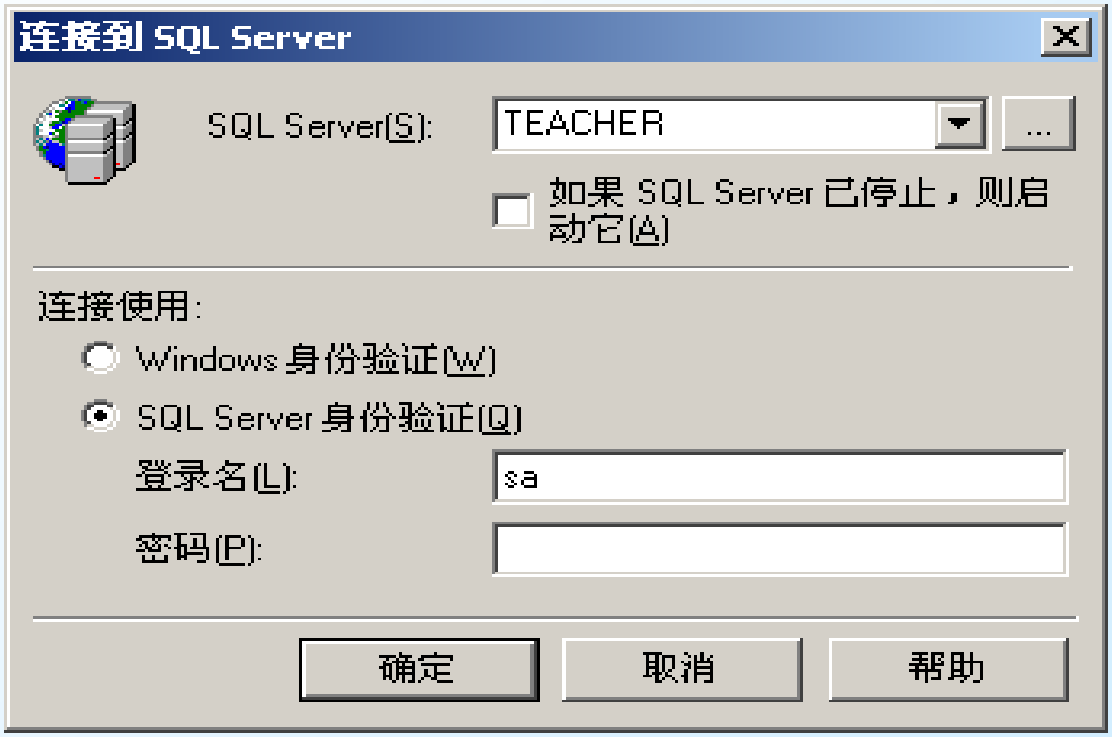


图7.3 启动查询分析器

7.3 查询分析器

选择到哪个SQL Server服务器。

选择服务器并设置正确使用该对话框可以指定连接的身份验证方式，单击“确定”

按钮，即可打开SQL Server查询分析器，如图7.4所示

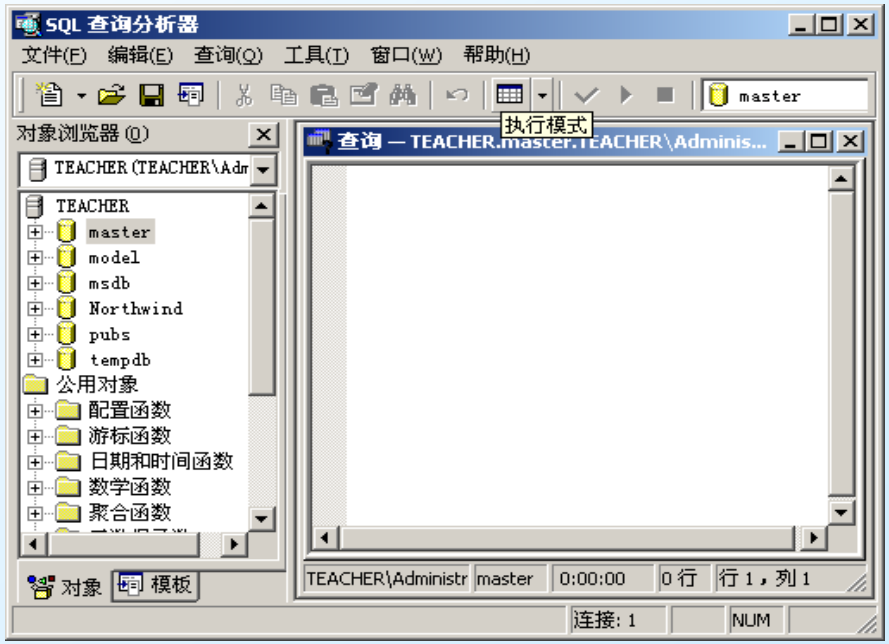


图7.4 查询分析器

7.3 查询分析器

查询分析器左边的窗口是“对象浏览器”，这个窗口是SQL Server 2000中新增的窗口。该窗口包含两个选项卡：

“对象”选项卡—用于浏览SQL Server中所有的数据库对象，内置函数和数据库类型等；

“模版”选项卡—提供一些常用的程序模版，用户可以在这些模版的基础上进行修改，以简化Transact-SQL语言的输入操作。

7.4 系统数据库

7.4.1 系统数据库

SQL Server 2000 系统有四种系统数据库：

1. master 数据库

master 数据库记录 SQL Server 系统的所有系统级别信息

2. tempdb 数据库

tempdb 数据库保存所有的临时表和临时存储过程。

3. model 数据库

model 数据库用作在系统上创建的所有数据库的模板（包括用户数据库和tempdb数据库）。

4. msdb 数据库

SQL Server 企业管理器和SQL Server Agent使用Msdb数据库来执行安排工作和警报以及记录操作者等操作。

7.4.2 实例数据库

也存储在SQL Server 2000默认安装目录的Data 目录中。Pubs和Northwind数据库可以作为SQL Server 2000的学习工具。其中，pubs实例数据库SQL Server 2000有两个实例数据库：pubs和Northwind。与系统数据库一样，实例数据库的文件存储了一个虚构的图书出版公司的基本情况，Northwind实例数据库包含了一个公司的销售数据，是一个虚构的公司，该公司从事世界各地的特产食品进出口贸易

7.4.3 创建数据库

在SQL Server 2000中创建数据库的场所有两处：一是在企业管理器中使用现成的命令和功能交互式创建；二是在查询分析器中书写Transact-SQL语句。

首先介绍如何使用企业管理器创建自己的用户数　据库。使用企业管理器创建数据库，可以采用两种方法：①使用创建数据库向导；②在控制面板树上选择数据库，然后选择“新建数据库”菜单命令直接创建用户数据库。

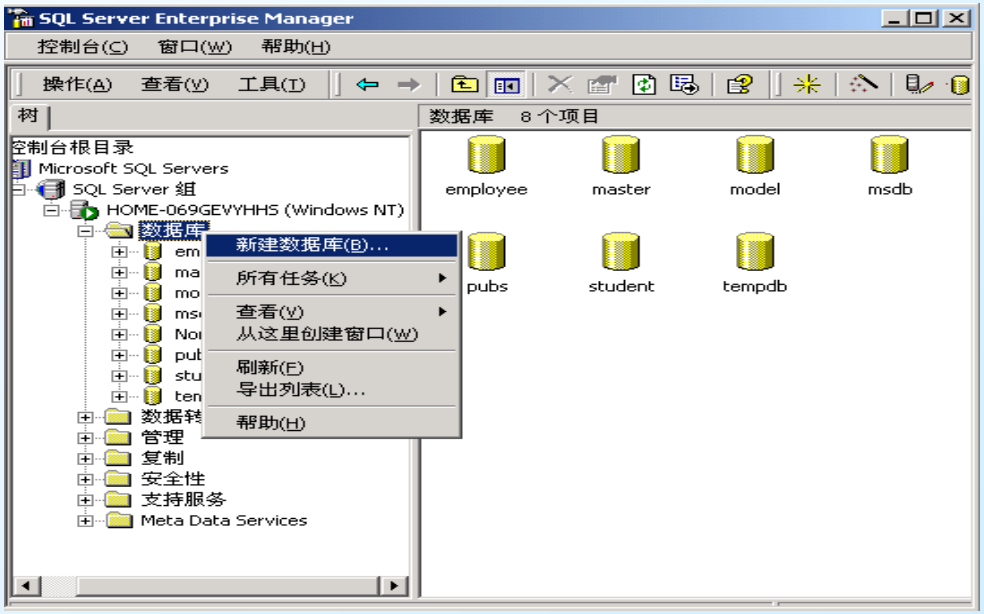
数据库创建向导简单易用，但是它的使用在一定程度上限制了物理数据库的复杂程度，所以在实际操作中并不常用。因此，我们主要讨论第二种方法

7.4.3 创建数据库

在企业管理器中直接创建用户数据库的步骤如下：

（1）打开企业管理器

（2）在控制面板目录中选择“数据库”节点如下图



7.4.3 创建数据库

（3）在“数据库”节点上单击右键，并在弹出的菜单中选择“新建数据库”

（4）在“名称”文本框中输入数据库的名称。

（5）点击“数据文件”选项卡。“数据文件”是SQL Server 2000用于实际存储数据、索引等数据库对象的文件。

（6）点击“日志文件”选项卡指定事务日志文件的名称。

（7）单击“确定”按钮，则创建一个新数据库

7.4.4 查看数据库信息

1. 用企业管理器查看数据库信息

在企业管理器窗口中查看数据库信息的方法如下。

（1）方法1：在企业管理器窗口中的左侧目录树窗口中，展开“数据库”文件夹，在某个数据库名称上先单左键，再单击右键，在出现的快捷菜单中单击“查看”中的“任务板”。

（2）方法2：在企业管理器窗口中的左侧目录树窗口中，展开“数据库”文件夹，单击某个数据库名称，然后单击“查看”菜单，在其下拉菜单中单击“任务板”

7.4.4 查看数据库信息

操作完成后，在企业管理器窗口右侧的“任务板中看到数据库的“常规”信息，“表”信息和“向导”信息。

在打开数据库文件夹目录树后，可以选择各种数据库对象进行信息浏览。

7.4.5 更改数据库

1. 使用企业管理器

修改数据库的一个简单快捷的方法是使用企业管理器，使用它修改数据库结构的步骤下：

（1）打开需要修改的用户数据库的数据库属性对话框。

（2）在属性对话框中对相关内容（如数据文件和事务日志文件的属性）进行修改

7.4.5 更改数据库

2.使用ALTER DATABASE 语句修改

除使用企业管理器外，还可以使用ALTER DATABASE语句修改数据库，语法如下：

```
ALTER DATABASE database
{   ADD   FILE 〈filespec〉 [   ,   ...n]   [TO   FILEGROUP
  filegroup_name]
|ADD LOG  FILE 〈filespec〉 [, ...n]
|REMOVE FILE logical_file_name
|ADD FILEGROUP filegroup_name
|REMOVE FILEGROUP filegroup_name
|MODIFY FILE<filespec>
|MODIFY FILEGROUP file_group_name filegroup_property}
```

7.4.5 更改数据库

<filespec>::=(NAME=logical_file_name[,FILENAME='os_file_name'][,SIZE=size][,MAXSIZE={vmax_size|UNLIMITED}][,FILEGROWTH=growth_increment])

各参数说明如下：

（1）ADD FILE指定要增加文件到哪个文件组。

（2）TO FILEGROUP指定要增加文件到哪个文件组。

（3）ADD LOG FILE 指定要增加的事务日志文件。

（4）REMOVE FILE 从数据库系统表中删除指定文件的定义并且删除其物理文件。文件只有为空时才能被删除。

（5）ADD FILEGROUP指定要增加的文件组。

（6）REMOVE FILEGROUP从数据库中删除指定文件组的定义并且删除其包括的所有数据库文件。

7.4.5 更改数据库

（7）MODIFY FILE修改指定文件的文件名，容量大小，最大容量以及文件增容方式等属性，但一次只能修改一个文件的一个属性。使用此项时应注意：在文件格式filespec中必须NAME明确指定文件的名称，如果文件大小已经确定，那么新定义的size必须比当前的文件容量大

（8）MODIFY FILEGROUP filegroup_name{filegroup_property}用于修改文件组属性。

［例1］ 修改student数据库的主数据文件的大小增加到15MB。

7.4.5 更改数据库

ALTER DATABASE student MODIFY FILE(NAME='员工数据库_dat',SIZE=15)

3.更改数据库名

重命名数据库需要使用系统存储过程SP_renamedb,其语法如下：

sp_renamedb
[@old_name=]'old_name',[@new_name=]'new_name'

[例2] 更改student数据库的名称为study。

exec sp_renamedb 'student','study'

7.4.6 删除数据

1. 用企业管理器删除数据库
2. 用DROP DATABASE命令删除数据库

[例3] 删除数据库student。

```
DROP DATABASE student
```

7.4.7 压缩数据库

数据库在使用一段时间后时常会出现因数据删除而造成数据库中空闲空间太多的情况，这时就需要减少分配给数据库文件和事务日志文件的磁盘空间，以免浪费磁盘空间。当数据库中没有数据时，可以修改数据库文件属性，直接改变其占用空间；但当数据库中有数据时，这样做会破坏数据库中的数据。因此需要使用压缩的、式来缩减数据库空间。 对数据库可以进行自动压缩，也可以进行人工压缩。

7.4.7 压缩数据库

1 自动压缩数据库

2 人工压缩数据库

（1）使用企业管理器压缩

在企业管理器界面下在要被压缩大小的数据库上选择单击鼠标右键，在快捷菜单中选取所有任务→收缩数据库，

（2）用DBCC SHRINKDATABASE命令压缩数据库大小

除了利用企业管理器来压缩数据库大小外，也可以使用DBCC SHRINKDATABASE命令来压缩数据库大小以删除未使用使用空间。其语法格式如下：

7.4.7 压缩数据库

DBCC SHRINKDATABASE(数据库名称[, 百分比][, {NOTRUNCATE|TRUNCATEONLY}])

说明：

百分比：在数据库被压缩后，数据库文件保留的空间比率

[例4] 压缩数据库student的未使用空间为数据库大小50%。

```
dbcc shrinkdatabase(student, 50)
```

（3）使用DBCC SHRINKFILE改变数据库文件大小

DBCC SHRINKFILE命令用于压缩当前数据库中的文件，其语法如下：

7.4.7 压缩数据库

BCC SHRINKFILE({文件名称|file_id}，{[，新的文件大小]｛EMPTYFILE | NOTRUNCATE | TRUNCATEONLDY ｝}))

[例5] 压缩数据库student中的数据文件student到1MB。

dbcc shrinkfile(student,1)

7.5 管理数据表

创建了一个数据库以后，就可以在该数据库中创建表了。表是一种最重要的数据库对象，它在数据库中存储数据，可以创建自己的数据表。同时表也是用来存储数据和操作数据的逻辑结构，其结构和电子表格相似，有行和列组成。

7.5.1 用企业管理器创建数据表

在企业管理器中创建表按以下步骤进行。

（1）在要创建表的数据库中选择“表”对象后，单击右键从快捷菜单中选择“新建表”选项或在工具栏中选择　图标，即会出现定义列对话框，如图7.8所示在此可设定表的列名、数据类型、精度和缺省值等属性。

（2）将数据表中各列定义完毕后，单击工具栏中的保存按钮，即出现输入新建表名的对话框。

（3）输入表名后，单击“确定”按钮，即会将表保存到数据库中。

7.5.2 修改表

1. 表的重新命名

（1）在企业管理器中展开希望重新命名的表所在的数据库节点，然后选中该数据库节点下一级菜单中的“表”节点。

（2）在希望重新命名的表上单击鼠标右键，在弹出菜单上选择命令“重命名”，输入新的表名称。

（3）按回车键，在弹出的对话框内单击“确定”按钮

当表创建好后，可以根据需要对表的列，约束等属性进行添加，删除和修改。这就需要修改表结构

2. 修改数据表结构

打开希望修改表结构的表设计窗口，在该窗口中可以执行下列操作:

7.5.2 修改表

（1）改数据表中字段定义

（2）插入新字段

如果希望为数据表添加新的字段，可以在某字段所在的行上单击鼠标右键，然后在弹 出菜单中选择“插入列”命令，该字段前面会插入一个空白行，在空白行中编辑新字段即可。

（3）删除现有字段

选中该字段并单击鼠标右键，在弹出菜单中选择命令“删除列”。

3. 修改表格属性

在表设计窗口中单击“表和索引属性”按钮，打开表格属性窗口，在该窗口中可以设置与整个表相关的重要属性，如“表”，“关系”，“索引/键”，“CHECK约束”。

7.5.2 修改表

（1）在要创建表的数据库中选择“表”对象后，单击右键从快捷菜单中选择“新建表”选项或在工具栏中选择图　标，即会出现定义列对话框，如图7.8所示在此可设定表的列名、数据类型、精度和缺省值等属性。1）在要创建表的数据库中选择“表”对象后，单击右键从快捷菜单中选择“新建表”选项或在工具栏中选择

（2）将数据表中各列定义完毕后，单击工具栏中的保存按钮，即出现输入新建表名的对话框。

（3）输入表名后，单击“确定”按钮，即会将表保存到数据库中。

7.5.2 修改表

1. 表的重新命名

（1）在企业管理器中展开希望重新命名的表所在的数据库节点，然后选中该数据库节点下一级菜单中的“表”节点。

（2）在希望重新命名的表上单击鼠标右键，在弹出菜单上选择命令“重命名”，输入新的表名称。

（3）按回车键，在弹出的对话框内单击“确定”按钮

当表创建好后，可以根据需要对表的列，约束等属性进行添加，删除和修改。这就需要修改表结构

2. 修改数据表结构

打开希望修改表结构的表设计窗口，在该窗口中可以执行下列操作：

（1）改数据表中字段定义

7.5.2 修改表

（2）插入新字段

（3）删除现有字段

选中该字段并单击鼠标右键，在弹出菜单中选择命令“删除列”。

3. 修改表格属性

4. 用存储过程sp_rename修改表名和列名

sp_rename存储过程可以修改当前数据库中用户对象的名称，如表，列，索引，存储过程等。其语法如下：

7.5.2 修改表

exec sp_rename
[@objname=]'object_name',[@newname=]'new_name'
[,[@objtype=]'object_type'] OBJECT值指代了系统表sysobjects中的所有对象，如表、视图、存储过程、触发器、规则和约束等。OBJECT值为默认值。

[例6] 更改s表中的列sno名称为stusno。

```
exec sp_rename 's.sno', 'stusno','column'
```

7.5.3 查看表

1．查看表的属性

在企业管理器下选择表名，单击鼠标右键，在出现的快捷菜单中选取“属性”，将会出现如图7.9的界面。在此界面中我们可以选择“常规”以查看表结构，从中可以看到表的大部分属性信息，如数据表的名称，所有者，创建日期，文件组，记录的行数，数据表中的字段名称，结构和类型等。

在企业管理器下选择表名，单击鼠标右键，在出现的快捷菜单中选取“属性”，将会出现如图7.9的界面。在此界面中我们可以选择“常规”以查看表结构，从中可以看到表的大部分属性信息，如数据表的名称，所有者，创建日期，文件组，记录的行数，数据表中的字段名称，结构和类型等。

7.5.3 查看表

2.查看数据表中的数据

在企业管理器中，用右键单击要查看数据的表，从快捷菜单中选择“打开表”，再选择其子菜单中的“返回所有行”，如图7.10，图7.11所示

3.用系统存储过程查看表的信息

sp_help存储过程可以提供指定的数据库对象的信息和系统或用户定义的数据类型的信息。其语法如下：

sp_help [[@objname=]name]

sp_help存储过程只用于当前的数据库，其中[@objname=]name子句指定对象的名称。如果不指定对象的名称，sp_help存储过程就会列出当前数据库中的所有对象

7.5.3 查看表

对象的所有者和对象的类型，但触发器的信息需要用sp_helptrigger存储过程来显示。

[例7] 显示当前数据库中的所有对象的信息：

```
exec sp_help。
```

显示student表的信息：

```
exec sp_help student
```

7.5.4 用企业管理器删除表

在企业管理器中，用右键单击要删除的表，从快捷菜单中选择“删除”选项，则会出现删除对象对话框；单击“全部除去”按钮，当有对象依赖于表时就不能删除表了。

当有对象依赖于表时就不能删除表了。

7.5.5 对表建立索引

在SQL Server中，除了使用SQL语句创建索引外，还用企业管理器创建索引。

1. 用索引创建向导创建索引

（1）打开企业管理器，从工具栏中选择快捷菜单“工具→向导”（如图7.12），打开 “选择向导”对话框。

（2）在“选择向导”对话框中选择“数据库→创建索引向导”节点，单击“确定”按钮

（3）在创建索引向导的欢迎对话框中，单击“下一步”，打开“选择数据库和表”对话框。

7.5.5 对表建立索引

（4）在“选择数据库和表”对话框的“数据库名称”一栏中选择数据库，在“对象名”一栏中选择表名。这里选择了数据库Student中表STUDENT（如图7.14），单击“下一步”。

（5）在“选择列”对话框中选择将包含在索引中的字段。这里选择基于“DEPT”字段创建索引（如图7.15），然后单击“下一步”。

（6）在“指定索引选项”对话框（如图7.16）中设置索引选项，各个选项含义如下使其成为聚集索引：指定该索引为聚集索引。

7.5.5 对表建立索引

使其成为唯一性索引：指定将该索引创建为唯一索引。

填充因子：设置填充因子，并设置系统在最初创建索引时索引页的填充程度。如果希望系统自动填充因子，请选择“最佳”；如果希望自己设置填充因子，选择“固定”，并在右边输入一个数字，该数字表示的是百分比。

（7）设置完索引选项以后，单击“下一步”，弹出“正在完成索引向导”对话框，在该对话框中显示了索引名称和包含在索引中的字段。在这个对话框中还可以为索引重新命

2 使用企业管理器创建索引

介绍如何使用企业管理器创建和管理索引。

（1）选择表“STUDENT”，单击鼠标右键。在弹出菜单中选择命令“所有任务→

7.5.5 对表建立索引

（2）在“管理索引”对话框中从“数据库”下拉菜单中选择数据库student，从“表/视图”下拉菜单中选择表[dbo].[STUDENT]（如图7.19）

（3）此时在“现有索引”的列表框中会列出表s中现存的索引。本例在“SNO”字段上创建一个新的索引，所以选择“新建”按钮，弹出“新建索引”对话框。

（4）在“新建索引”对话框的索引名称文本框中输入索引名称，然后选择字段“SNO”并设置索引属性

（5）设置索引选项后，单击“确定”按钮即可。

3. 查看和修改索引

（1）用企业管理器查看和修改索引

（2）用sp_helpindex存储过程查看索引

7.5.5 对表建立索引

[例8] 查看S表的索引。

```
exec sp_helpindex S
```

4. 用sp_rename存储过程更改索引名称

[例9] 更改s表中索引insno改为dxsno。

```
exec sp_rename 's.insno','dxsno'
```

5.删除索引

(1) 用企业管理器删除索引

(2) 用DROP INDEX命令删除索引

DROP INDEX命令可以删除一个或多个当前数据库中的索引，其语法如下：

7.5.5 对表建立索引

DROP INDEX 'tablename.indexname'[,...n]

DROP INDEX 命令不能删除有CREATE TABLE或ALTER TABLE命令创建的PRIMARY KEY或UNIQUE约束索引，也不能删除系统表中的索引。

[例10] 删除表S中的索引dxsno。

```
drop index S.dxsno
```

7.6 用企业管理器管理数据

创建了数据库，也创建了若干数据库表了，接下来就要向表中添加数据，输入有误时会修改数据，不需要的数据要能删除掉，这些维护数据的操作SQL命令能完成的，这里主要介绍在企业管理器中如何交互界面操作的方法。

7.6.1 添加数据

添加的方法是：利用前面介绍的方法打开待加入记录数据的数据表(如图7.11)，在弹出的窗口中，单击最后的空白行，分别向各字段中输入新数据即可。当输入一个新记录的数据后，会自动在最后出现一新的空白行，用户可以继续连续输入多个记录的数据

7.6.2 删除数据

在SQL Server中，除使用DELETE语句删除数据表中的记录数据外，还可以在企业管理器中，在查看数据表的数据时删除数据，但这种方式比较适合于删除单个或少量的记录等简单情况。

删除数据的具体方法是：打开要删除记录数据的数据表，在如图7.22的窗口中，单击记录左侧的小方快，此时该记录为黑色，表示已选择了该记录，如图7.22所示。也可以用鼠标在记录左侧的区域上，下拖动来选择多个记录。此时按下Del键，系统会出现如图7.23所示的对话框，提示用户是否要删除记录，如果用户单击“是”按钮，则记录被删除，如果用户单击“否”按钮，则记录将不被删除。

具体的图见书上

7.6.3 修改数据

在SQL Server中，除使用UPDATE语句修改数据表中的数据外，还可以在企业管理器中，在查看数据库表的数据时修改数据，但这种方式不能应付大量的数据修改。

修改数据的方法是：打开待修改记录数据的数据表，在如图7.22所示的窗口中，单击要修改的记录，分别向各字段中输入新数据即可，原数据被新数据覆盖。

注意：数据一旦交互更新后，将不能撤消操作。除非利用实时的日志信息来恢复。

7.7 数据完整性

数据库的完整性是指数据库数据的正确性、有效性和相容性，SQL Server 2000有全面的措施来保障数据完整性。

7.7.1 数据完整性概述

数据完整性（Data Integrity）是指数据的精确性和可靠性。它是为防止数据库中存在不符合语义规定的数据和防止因错误信息的输入输出造成无效操作或错误信息而提出的。数据完整性分为四类：实体完整性（Entity Integrity），域完整性（Domain Integrity），SQL Server提供了一些工具来帮助用户实现数据完整性，其中最主要的是规则，默认，约束和触发器

7.7.2 规则

所谓规则是指当您对表做新建或操作时该列输入值必须符合预先设置的条件，如果不符合的话SQL Server就不会让该条数据新建至表内或被操作。

一、利用企业管理器来管理规则

首先在企业管理器主界面下利用鼠标选择要管理规则的数据库，将其展开。选中“规则”弹出如图7.24的界面。如要查看某一列规则的内容，在列规则名称处双击鼠标即可如果要更改某一处规则名的话，在该列规则名称处单击鼠标右键，在出现的快捷菜单中选取重命名即可。也可以在查询分析器下执行sp_rename来更改规则的名称，以下是sp_rename的语法：sp_rename旧规则名称, 新规则名称

7.7.2 规则

二、创建规则

在企业管理器主界面下利用鼠标选择要创建规则的数据库，将其展开。选取“规则”单击鼠标右键，在出现的快捷菜单中选取“新建规则”

在“名称”处输入规则的名称，在“文本”处输入该规则运算式。

规则的条件运算式必须以局部变量名称起头（也即第一个字符必须为“@”），该变量名称可以随意取名。

三、使用CREATE RULE命令创建规则

CREATE RULE命令的语法如下：CREATE RULE[拥有者.]规则名称 AS 条件运算式（规则的定义）。

7.7.2 规则

规则的定义可以是用于WHERE条件子句中的任何表达式，可包含算术运算符，关系运算符和谓词（如IN，LIKE，BETWEEN等）。

[例11] 创建学生性别规则。

```
create rule ru_sex as @sex in('男','女')
```

四、用存储过程查看规则

使用sp_helptext存储过程可以查看规则的细节，其语法规则如下：

```
sp_helptext [@objname=]'name'
```

其中，[@objname=]'name'子句指明对象的名称，sp_helptext存储过程查看的对象可以是当前数据库中的规则，默认，触发器，视图或为加密的存储过程。

7.7.2 规则

[例12] 查看名称为“ru_sex”的规则的内容。

```
exec sp_helptext 'ru_sex'
```

五、规则的绑定与松绑

当规则创建好后它必须和表内的列绑定才可以发挥作用。要将规则定义与表列绑定，首先在企业管理器下单击数据库的对象“规则”，即可在右边的窗口中看到已创建的规则名称，在相应的规则名称上面单击鼠标右键，从快捷菜单中选择“属性”选项，会出现“规则属性”对话框

在“规则属性”对话框中，单击“绑定UDT”按钮，则出现“绑定规则到用户自定义数据类型”的对话框，单击“绑定列”按钮则出现“将规则绑定到列”的对话框。

7.7.2 规则

在该对话框中，在“表”所对应的下拉列表中，选择相应的数据表（如STUDENT表），在左边“未绑定的列”列表只能单击要绑定的某个列（如sex），然后单击“添加”按钮，将该列添加到“绑定列”中，最后单击“确定”，这时规则就被绑定到所选定的列上了。

除了用企业管理器绑定规则外，还可以使用存储过程绑定规则。

(1)存储过程sp_bindrule可以将一个规则绑定到表的一列或一个用户自定义数据类型上。其语法如下：

```
sp_bindrule
[@rulename=]'rule',[@objname=]'object_name'[,'futureonly']
```

7.7.2 规则

各参数说明如下：

①[@rulename=]'rule' 指定规则名称。

②[@objname=]'object_name' 指定规则绑定的对象。

③'futureonly'选项仅在绑定规则到用户自定义数据类型上时才可以使用。当指定此选项时，只有以后使用此用户自定义数据类型的列会应用新规则，而当前已经使用此数据类型的列不受影响。

[例13] 绑定规则ru_sex到s表的字段sex。

```
exec sp_bindrule 'ru_sex', 's.sex'
```

注意：规则对已经输入表中的数据不起作用。

与表的列绑定的规则优先于与用户自定义数据类型绑定的规则。因此，如果表的列的数据类型与规则A绑定，同时列又与规则B绑定，则以规则B为列的规则。

7.7.2 规则

（2）用存储过程sp_unbindrule其语法格式如下：

```
sp_unbindrule
[@objname=]'object_name'[,'futureonly']
```

其中，‘futureonly’选项同绑定时一样，仅用于用户自定义数据类型，它指定现有的用此用户自定义数据类型定义的列仍然保持与此规则的绑定。

[例14] 解除已绑定到S表的字段sex的规则ru_sex。

```
exec sp_unbindrule 'ru_sex'
```

六、删除规则

（1）用企业管理器删除规则 。注意：在删除一个规则前必须先将与其绑定的对象解除绑定。

7.7.3 默认

[例15] 删除上面已创建的规则ru_sex。

DROP RULE ru_sex

所谓默认（default）是用户输入记录时没有指定具体数据的列中自动插入的数据。

1 创建默认

（1）用CREATE DEFAULT命令创建默认

CREATE DEFAULT命令用于在当前数据库中创建默认对象，其语法如下：

CREATE DEFAULT default_name AS constant_expression

其中，default_name是要创建的默认的名称， constant

7.7.3 默认

expression子句是默认的定义，该子句可以是数学表达式或函数，也可以包含表的列名或其他数据库对象。

[例16] 创建年龄的默认值age_defa。

```
CREATE DEFAULT age_defa as '20'
```

设定默认值后，当用户在输入记录数据时，如果未提供字段age的值，系统将自动默认其值为“20”岁。

（1）用企业管理器查看默认

（2）用企业管理器创建默认

2. 查看和修改默认

（1）用企业管理器查看默认

（2）用存储过程sp_helptext查看默认

使用sp_helptext存储过程可以查看默认的信息。[例17] 查看sex_defa的默认。exec sp_helptext sex_defa

7.7.3 默认

3. 默认的绑定与松绑

创建默认后，默认仅仅是一个存在于数据库中的对象，并未发生作用，默认同规则一样，需要将它与数据库表或用户自定义对象绑定。

（1）用企业管理器管理默认的绑定

用企业管理器管理默认的绑定，方法如下：在企业管理器的左侧窗格中，展开相应的数据库目录，数据库的对象“默认”上面单击右键，这时在右侧窗格中即可看到已建立的各个默认的信息，在某个默认的名称上面单击右键，从快捷菜单中选择“属性”菜单项，即会出现“默认属性”对话框。**单击“绑定UDT”按钮，则出现“绑定默认到表的列”的** 对话框。用他们来管理默认与表的列以及用户自定义数据类型之间的绑定非常方便。

7.7.3 默认

(2)用存储过程sp_bindefault 绑定默认

存储过程 sp_bindefault可以将一个默认绑定到表的一列或一个用户自定义数据类型上。其语法如下：

sp_bindefault[@defname=]'default',

[@objname=]'object_name'[,'futureonly']

[例18] 绑定默认sex_defa到数据表S的sex上。

exec sp_bindefault sex_defa,'S.[sex]'

4. 删除默认

（1）用企业管理器删除默认

（2）使用DROP DEFAULT命令删除默认

使用DROP DEFAULT命令删除当前数据库中的一个或多个默认，其语如下： DROP DEFAULT{默认名}[,…n]

7.7.3 默认

[例19] 删除学生性别的默认值sex_defa。

```
DROP DEFAULT sex_defa
```

注意：删除一个默认值前必须先将与其绑定的对象解除绑定。

7.8 数据查询

数据库是为更方便有效的管理数据而存在的，人们希望可以随时提供有效的所需要的数据信息。因此，对用户来说，数据查询是数据库最重要的功能。

在数据库中数据查询是通过SELECT语句来完成的。SELECT语句可以从数据库中按用户要求检索数据，并将查询结果以表格的形式返回。[例20] 查询系科为“计算机”系的学生的学号和姓名，并把查询结果存放到变量@sno和@sn中。命令为：

```
declare @sno varchar(10)
declare @sn varchar(10)
select @sno=sno,@sn=sn from s where sept='计算机'
```

SELECT @local_variable 通常用于将单个值返回到变量中。

7.8 数据查询

如果 SELECT 语句返回多个值，则将返回的最后一个值赋给变量。如果 SELECT 语句没有返回行，变量将保留当前值。如果 expression 是不返回值的标量子查询（集合形式），则将变量设为 NULL。

7.9 视图

视图是与基本表同等重要的概念，在第3章已有介绍，这里主要介绍关于它的交互式界面7.9视图操作与管理的内容。

7.9.1 创建视图

一、使用创建视图向导

使用创建视图向导创建视图的步骤如下：

（1）打开企业管理器，选择工具→向导，打开“选择向导”对话框，在“选择向导”

对话框中展开“数据库”节点，并选择它的下一级节点“创建视图向导”，单击“确定”按钮继续（如图7.31所示），此时会出现“欢迎使用创建视图向导”对话框，在该对话框中单击“下一步”。

（2）在如图7.32的窗口中选择需要的数据库，单击“下一步”。这里选择student数据库。

7.9.1 创建视图

（3）在如图7.33所示的窗口中选择数据库的对象，方法是在选择对象的“包含在视图中”中选种复选框，单击“下一步”。

（4）选择将在视图中显示的所有字段，单击“下一步。”

（5）在“定义限制”窗口中输入WHERE语句（如图7.34）所示

（6）在“命名视图”中输入视图名称。

（7）在“完成创建视图向导”对话框中单击“确定”即可。

二、在企业管理器中交互式创建视图

在企业管理器中，展开指定的数据库，点按“视图”，按鼠标右键，从弹出的快捷菜单中选择“新建视图”菜单项。

7.9.2 管理视图

一、查看和修改视图

（1）方法1

① 启动企业管理器，登录到指定的服务器。

② 打开要查看或修改视图的数据库文件夹，选种“视图”图标，此时在右面的窗格中显示当前数据库的所有视图。

③ 在右面窗格中，用右键单击要查看的视图，在弹出的菜单中选择“属性”菜单项，打开“视图属性”对话框。

④在该对话框内可浏览到该视图的SQL文本，也可以对该视图进行修改，然后单击“检查语法”按钮来对语句合法性进行检查。若要对视图的访问权限进行设置，单击“权限”即可。

（2）方法2

7.9.2 管理视图

②打开要查看或修改视图的数据库文件夹，选种“视图”图标，此时在右面的窗格中显示当前数据库的所有视图。

③在右面窗格中，用右键单击要查看的视图，在弹出的菜单中选择“设计视图”菜单项，可即时进入到设计视图的窗口，

④在该窗口中可按照创建视图的方法对原有的视图进行各种修改。

二、使用存储过程

在SQL Server中有三个用于了解视图信息的关键存储程，即sp_depends,sp_help 和sp_helptext。

其语法格式如下：sp_depends [@objname =] 'object'

参数：[@objname =] 'object' 下面的命令列出依赖Student表的数据库对象。

7.9.2 管理视图

EXEC sp_depends 'Student'

三、删除视图

除了SQL命令删除视图外，在SQL Server中，通过企业管理器也可删除视图。

（1）启动企业管理器，登录到指定的服务器。

（2）打开要删除视图的数据库文件夹，选种“视图”，此时在右面的窗格中显示当前数据库的所有视图。

（3）在右窗格中，用右键单击要删除的视图，在弹出菜单中选择“删除”菜单项即可删除视图。

7.10 备份和还原

数据库的备份和还原是维护数据库的安全性和完整性的重要组成部分。通过备份数据库，可以防止因为各种原因而造成的数据破坏和丢失。还原是指在造成数据丢失和破坏以后利用备份来恢复数据的操作。

7.10.1 备份和还原概念

备份和还原是SQL Server的重要组成部分。备份是对SQL Server数据库或事务日志进行复制，数据库备份记录了在进行备份这一操作时数据库中所有数据的状态，如果数据库因意外而损坏，这些备份文件将在数据库还原时用来还原数据库。

一、数据库备份的类型

7.10.1 备份和还原概念

SQL Server2000提供了四种主要的备份方式：数据库备份，事务日志备份，差异备份，文件和文件组备份。下面分别介绍：

（1）数据库备份

数据库备份是指对数据库的完整备份，包括所有的数据以及数据库对象。

（2）事务日志备份

事务日志备份是指对数据库发生的事务进行备份，包括从上次进行事务日志备份，差异备份和数据库完全备份之后，所有已经完成的事务。

（3）差异备份

7.10.1 备份和还原概念

差异备份是指将最近一次数据库备份以来发生的数据变化备份起来。

（4）文件和文件组备份

文件或文件组备份是指对数据库文件或文件组进行部分，但它不象完整的数据库备份那样同时也进行事务日志备份。使用该备份方法可提高数据库还原的速度，因为它仅对遭到破坏的文件或文件组进行还原。

二、备份和还原的策略

（1）如果只进行数据库备份，那么将无法还原自最近一次数据库备份以来数据库中所发生的所有事务。

7.10.1 备份和还原概念

（2）如果在进行数据库备份时也进行事务日志，那么可以将数据库还原到失败点。从以上问题可以看出，对数据库一致的要求程度成为我们选择备份方案的主要 的普遍性的问题。另外需要注意的是我们在备份时要决定应使用哪种备份设备，如磁盘或磁带，并且决定如何在备份设备上创建备份，比如将备份添加到备份设备上或将其覆盖。

7.10.2 创建备份和还原

在进行备份前必须先创建备份设备。备份设备是用来存储数据库、事务日志或文件和文件组备份的存储介质。

1. 如何创建磁盘备份设备（企业管理器）

①展开服务器组，然后展开服务器。

②展开"管理"文件夹，右击"备份"，然后单击"新建备份设备"命令。如图7.36

7.10.2 创建备份和还原

③在“名称”框中输入该命名备份设备的名称。

④单击"文件名"，然后执行下列操作之一：

输入磁盘备份设备所使用的文件名。

单击浏览（“...”）按钮显示“备份设备位置”对话框，再选择磁盘备份设备所使用的本地计算机上的物理文件。

2. 如何删除备份设备（使用企业管理器）

①展开服务器组，然后展开服务器。

②展开"管理"文件夹，然后单击"备份"。

③在详细信息窗格中，右击要删除的命名备份设备。

7.10.2 创建备份和还原

④单击"删除"命令并确认删除。

使用存储过程管理备份设备：

（1）使用sp_addumpdevice创建备份设备

```
sp_addumpdevice[@devtype=]'device_type' ,[ @logic
  alname = ] 'logical_name',[ @physicalname = ]
  'physical_name'[,{[@cntrltype=]controller_type|
  [ @devstatus = ] 'device_status' }]
```

[例21] 下面的示例添加一个名为 MYDISKD 的磁盘备份设备。

```
USE master
EXEC sp_addumpdevice
'disk','mydiskd','c:\dump\dump1.bak'
```

7.10.2 创建备份和还原

[例22] 下面的示例显示一个远程磁盘备份设备。

```
USE master
EXEC sp_addumpdevice
  'disk','networkdevice','\\servername\sharename\
  path\filename.ext'
```

[例23] 下面的示例添加 TAPEDUMP1 设备，其物理名称为 \\.\Tape0。

```
USE master
EXEC sp_addumpdevice 'tape',
  'tapedump1','\\.\tape0'
```

7.10.2 创建备份和还原

sp_dropdevice删除备份设备，其语法格式为：

sp_dropdevice [@logicalname =] '*device*'
[,[@delfile =] '*delfile*']

其中，[@logicalname=]'*device*'数据库设备或备份设备的逻辑名称，*device* 的数据类型为 sysname，没有默认值。[@delfile=]'*delfile*'指出是否应该删除物理备份设备文件。

[例24] 下面的示例从 SQL Server 除去 TAPEDUMP1 磁带转储设备。

sp_dropdevice 'TAPEDUMP1'

7.10.3 数据库备份

在SQL Server中无论是数据库备份，还是事务日志备份，差异备份，文件和文件组备份都执行同样的步骤，。

（1）展开服务器组，然后展开服务器。

（2）展开“数据库”文件夹，右击数据库，指向“所有任务”子菜单，然后单击“备份数据库”命令。如图7.38

（3）在该对话框中的“常规“选项卡中的”备份“选项栏内选择要进行备份的类型。

（4）通过“添加“按钮来选择备份的设备。

（5）若选择了“重写“选项栏中的重写现有媒体，则将原备份覆盖。

（6）“调度”复选框，可对备份的时间表进行设置。

7.10.4 还原数据库

1. 展开服务器组，然后展开服务器。
2. 展开“数据库”文件夹，右击数据库，指向“所有任务”子菜单，然后单击“还原数据库”命令。
3. 在“还原为数据库”框中，如果要还原的数据库名称与显示的默认数据库名称不同，请在其中进行输入或选择。
4. 单击“数据库”。
5. 在"要还原的第一个备份"列表中，选择要还原的备份集。
6. 在"还原"列表中，单击要还原的数据库备份。

7.10.4 还原数据库

单击"选项"选项卡并执行下列操作：（可选）①在"还原为"中输入组成数据库备份的各数据库文件的新名称或新位置。②单击"使数据库可以继续运行，但无法还原其它事务日志"，如果没有其它要应用的事务日志或差异数据库备份。③如果要应用另一个事务日志或差异数据库备份，则单击"使数据库不再运行，但能还原其它事务日志"。

7.在设置完选项后，单击“确定“即可。

7.10.5 备份和还原系统数据库

系统数据库保存了有关SQL Server的许多重要数据相信，这些数据的丢失将给系统带来极为严重的后果。所以我们也必须对数据库进行备份。这样一旦系统或数据库丢失，则可以通过还原来重建系统数据库。在SQL Server中，重要的系统数据库主要有master，msdb，model。虽然tempdb也是系统数据库，但没有必要对其进行备份，因为SQL Servre每次启动都会重新创建该数据库，而当SQL Server停止运行时，tempdb数据库中所有数据都会被自动清除。在这里我们主要讨论master数据库的备份和还原问题。

7.10.5 备份和还原系统数据库

(1) 增加或删除用户数据库。但是如果增加或删除文件或文件组，或用户数据库自动增加来容纳新添加的数据，这些操作并不对master数据库产生影响，所以此时不必对它进行备份。

(2) 创建新的登录或执行与登录有关的操作。但是增加数据库用户并不影响master数据库。

(3) 更改任何服务器范围的配置选项或数据库配置选项。

(4) 创建或删除备份设备。

(5) 为进行分布式查询或远程过程调用而对数据库服务器进行配置，如增加连接服务器或远程登录等

7.10.5 备份和还原系统数据库

使用Rebuild Master Utility 来重建master数据库，主要执行以下步骤。

（1）关掉SQL Server，然后运行位于“.../Microsoft SQL Server/80/tools/Binn”下的Rebuildm.exe文件，出现“重建master”对话框。

（2）在出现的对话框中单击“浏览”按钮来浏览包括Data文件的源目录。

（3）单击“设置”按钮出现“排序规则设置”对话框。

（4）单击“重建”按钮来重建master数据库。

将数据库移到另外的服务器或磁盘上需要进行以下操作。

7.10.5 备份和还原系统数据库

（1）拆开数据库

（2）把数据库文件（数据库文件和事务日志文件）移到另外的服务器或磁盘上。

（3）连接服务器并定义数据库文件的存放位置。

在SQL Server中，拆开数据库使用的Transact-SQL命令是sp_detach_db，其语法格式为：sp_detach_db [@dbname=]'dbname'[,[@skipchecks=]'skipchecks']

其中，@dbname表示数据库名称； @skipchecks如果为True将修改关于数据库的所有统计表，反之则不修改。

7.10.5 备份和还原系统数据库

[例25] 拆开pubs数据库。

```
exec sp_detach_db 'pubs','true'
```

使用sp_attach_db来重新连接数据库，其语法格式为：

```
sp_attach_db
[@dbname=]'dbname',[@filename1=]'filename_n'[...16]
```

[例26]重新连接pubs数据库

```
exec sp_attach_db @dbname =
N'pubs','c:\mssql7\data\pubs.mdf','c:\mssql7\data\pubs.log'
```

7.11 SQL Server程序设计初步*

Transact-SQL语言是SQL Server对标准SQL语言的扩充，如：引入了程序设计的思想，增强了程序的流程控制语句等。因此，在Transact-SQL语言中，标准的SQL语句可畅通无阻。Transact-SQL语言最主要的用途是设计服务器的能够在后台执行的程序块，如：存储过程，触发器等。

7.11.1 Transact-SQL简介

一、在Transact-SQL中可以使用两种变量：局部变量和全局变量

1.局部变量

局部变量是用户可以自定义的变量，它的作用范围仅在程序内部。局部变量在程序中通常用来存储从表中查询到的数据，或当做程序执行过程中暂存变量。局部变量必须以@开头，而且必须先用DECLARE命令说明后才可使用。其语法形式如下：

DECLARE @变量名 变量类型[，@变量名 变量类型.....]

其中变量类型可以是SQL Server2000支持的所有数据类型。

7.11.1 Transact-SQL简介

在Transact-SQL中不能像在一般的程序语言中一样使用“变量=变量值”来给变量赋值，必须使用SELECT或SET命令来设定变量的值。其语法如下：

SELECT @局部变量=变量值

SET @局部变量=变量值

例27] 声明两个变量并赋值。

```
USE Northwind
GO
DECLARE @FirstNameVariable NVARCHAR(20),@RegionVariable NVARCHAR(30)
SET @FirstNameVariable = N'Anne'
SET @RegionVariable = N'WA'
```

7.11.1 Transact-SQL简介

2. 全局变量

全局变量是SQL Server 2000系统内部使用的变量，其作用范围并不局限于某一程序，而是任何程序均可随时调用。全局变量通常存储一些SQL Server 2000的配置设定值和效能统计数据。用户可在程序中用全局变量来测试系统的设定值或Transact-SQL命令执行后的状态值。

全局变量不是由用户的程序定义的，而是有系统定义和维护的，只能使用预先说明及定义的全局变量。引用全局变量时必须以“@@”开头。SQL Server提供的全局变量共有33个，但并不是每个都可用，而且局部变量与全局变量不能同名。否则会在应用中出错。

7.11.1 Transact-SQL简介

3.注释符

在Transact-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。

二、流程控制语句 1.BEGIN...END

1. BEGIN ... END 语句用于将多个 Transact-SQL 语句组合为一个逻辑块。

7.11.1 Transact-SQL简介

其语法格式如下：

```
BEGIN
   <命令行或程序块>
END
```

BEGIN 和 END 语句必须成对使用：任何一条语句均不能单独使用。BEGIN 语句行后为Transact-SQL 语句块。最后，END 语句行指示语句块结束。

BEGIN 和 END 语句用于下列情况：

(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。

7.11.1 Transact-SQL简介

2. IF...ELSEIF...ELSE的语法格式如下：IF<条件表达式><命令行或程序块>ELSE<条件表达式><命令行或程序块> 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. IF...ELSEIF...ELSE的语法格式如下：IF<条件表达式><命令行或程序块>ELSE<条件表达式><命令行或程序块> 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。

7.11.1 Transact-SQL简介

[例29] 下面的示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'
```

7.11.1 Transact-SQL简介

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>

...

7.11.1 Transact-SQL简介

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。

[例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”。

7.11.1 Transact-SQL简介

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

格式2：

```
CASE
WHEN <条件表达式> THEN <运算式>
…
WHEN <条件表达式> THEN <运算式>
END
```

7.11.1 Transact-SQL简介

[例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'
```

7.11.1 Transact-SQL简介

4. WHILE...CONTINUE...BREAK

只要指定的的条件为真，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

<命令行或程序块>

7.11.1 Transact-SQL简介

5.WAITFOR

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

（3）ERROREXIT：直到处理非正常中断

（4）PROCESSEXIT：直到处理正常或非正常中断

（5）MIRROREXIT：直到镜像设备失败

其语法格式如下：WAITFOR { DELAY 'time'|TIME 'time' }

7.11.1 Transact-SQL简介

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：GOTO 标识符

7.11.1 Transact-SQL简介

7. RETURN

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他程序。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

7.11.1 Transact-SQL简介

-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。

7.11.1 Transact-SQL简介

三、其他命令

1．BACKUP

BACKUP命令用于将数据库内容或其他事务处理日志备份到存储介质上（如软盘，硬盘，磁带等）。

2．CHECKPOINT

CHECKPOINT命令用于将当前工作的数据库中被更改过的数据页或日志页从数据库缓冲器中强制写入硬盘。

CHECKPOINT的语法格式如下：CHECKPOINT

3．DBCC

数据库一致性检查程序命令用于验证数据库完整性，查找错误，分析系统使用情况等。DBCC命令后必须加上子命令，系统才知道要做什么。如DBCC CHECKALLOC命令检查数据库

7.11.1 Transact-SQL简介

内所有数据页的分配和使用情况。

4. DECLARE

在批处理或过程的正文中用 DECLARE 语句声明变量，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 data_type } |
{ @cursor_variable_name CURSOR |
{ table_type_definition } } [ ,...n]
```

需要用SELECT或SET命令来给变量赋值。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。

7.11.1 Transact-SQL简介

5. EXECUTE

EXECUTE命令用来执行存储过程

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。

7. PRINT

PRINT 语句用一个字符或 Unicode 字符串表达式作为参数。它把这个字符串作为一个消息返回给应用程序。

PRINT的语法格式如下：

PRINT 'anyASCIItext' | @local_variable | @@FUNCTION | string_expression

7.11.1 Transact-SQL简介

8. RAISERROR

RAISERROR命令用于在SQL Server系统返回错误信息时，同时返回用户指定的信息。

9. READTEXT

读取 text、ntext 或 image 列中的 text、ntext 或 image 值，从指定的偏移量开始读取指定的字节数。其语法格式如下：

READTEXT { table.column text_ptr offset size }[HOLDLOCK]

其中：table.column：是从中读取的表和列的名称。

text_ptr：有效文本指针。

7.11.1 Transact-SQL简介

Offset：开始读取 text、image 或 ntext 数据之前跳过的字节数（使用 text 或 image 数据类型时）或字符数（使用 ntext 数据类型时）。

Size：是要读取数据的字节数（使用 text 或 image 数据类型时）或字符数（使用 ntext 数据类型时）。

HOLDLOCK：使文本值一直锁定到事务结束。

10. RESTORE

RESTORE命令用来将数据库或其他事务处理日志备份文件由存储介质回寸到SQL Server系统中。

11. SELECT

7.11.1 Transact-SQL简介

SELECT命令可用于给变量赋值，其语法格式如下：

SELECT [@Local_variable=expression][,...n]

SELECT命令可以一次给多个变量赋值。当表达式expression为列名时，SELECT命令可利用其查询功能一次返回多个值，变量中保存的是其返回的最后一个值。如果SELECT命令没有返回值，则变量值仍为原来的值。当表达式expression是一个子查询时，如果子查询没有返回值，则变量被设为NULL。

7.11.1 Transact-SQL简介

12. SET

SET命令有两种用法。

（1）用于给局部变量赋值。其语法格式如下：

```
SET
{{@local_varable=expression}|{@cursor_variable={@cursor_variable
|cursor_name|{CURSOR[FROWARD_ONLY|SCROLL][STATIC|KEYSET|DYNAMIC|FAST_FORWARD][READ_ONLY|SCROLL_LOCKS|OPTIMISTIC][TYPE_WARNING]
FOR select_statement [FOR{READ ONLY|UPDATE[OF column_name[,...n]]}]}}}}
```

在用DECLARE命令申明之后，所有的变量都被赋予初值

7.11.1 Transact-SQL简介

所有的变量都被赋予初值NULL。需要用SET命令来给变量赋值，但与SELECT命令不同的是SET命令一次只能给一个变量赋值，不过由于SET命令功能更强且更严密，因此，SQL Server推荐使用SET命令来给变量赋值。

（2）用于用户执行SQL命令时，处理选项的设定。起设定方式为：

SET 选项 {ON|OFF} 或 SET 选项 选项值

13. SHUTDOWN

除非 sysadmin 固定服务器角色成员指定 WITH NOWAIT 选项，否则 SHUTDOWN 尝试关闭 SQL Server 时的顺序方式为：

7.11.1 Transact-SQL简介

1. 禁用登录（sysadmin 固定服务器角色成员除外）
2. 等待当前正在执行的 Transact-SQL 语句或存储过程执行完毕。
3. 在每个数据库执行CHECKPOINT命令
4. 停止SQL Server命令执行

SHUTDOWN 的语法格式如下：SHUTDOWN [WITH NOWAIT]

当使用NOWAIT参数时，SHUTDOWN命令立即停止，在终止所有的用户过程并对每一现行的事务发生一个回滚后，退出SQL Server。

7.11.1 Transact-SQL简介

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。该语句将彻底重写受其影响的列中的任何现有数据。WRITETEXT 语句不能用在视图中的 text、ntext 和 image 列上。

其语法格式如下：WRITETEXT { table.column text_ptr }[WITH LOG] { data }

一般使用 WRITETEXT 来替换 text、ntext 和 image 数据，而用 UPDATETEXT 来修改 text、ntext 和 image 数据。UPDATETEXT 更灵活，因为它仅更改 text、ntext 或 image 列的某一部分，而不是整个列。

7.11.1 Transact-SQL简介

15. USE

USE命令用来改变当前使用的数据库为指定的数据库。用户必须是目标数据库的用户成员，或者在目标数据库中建有GUEST用户帐号时，使用USE命令才能成功切换到目标数据库。

四、常用函数

1.统计函数

在SQL Server 2000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)

7.11.1 Transact-SQL简介

（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。

其语法格式如下：STDEVP （ expression ）

其中参数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

（3）VAR

VAR函数返回给定表达式中所有值的统计方差。

其语法格式如下：VAR （expression ）

7.11.1 Transact-SQL简介

（4）VARP

VARP函数返回给定表达式中所有值的填充的统计方差。

其语法格式如下：VARP （ expression ）

其中参数expression表示精确数字或近似数字数据类型类别的表达式（bit 数据类型除外）。不允许使用聚合函数和子查询。它的返回类型是float

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。

7.11.1 Transact-SQL简介

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。

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时通常所需要的值。大多数字符串函数只能用于char、nchar、varchar 和 nvarchar 数据类型，或隐性转换为上述数据类型的数据类型。少数字符串函数也可用于 binary 和varbinary 数据。

（1）ASCII函数

ASCII函数返回字符表达式最左端字符的 ASCII 代码值。其语法格式如下：

ASCII (character_expression)

7.11.1 Transact-SQL简介

（2）CHAR函数

CHAR函数将 int ASCII 代码转换为字符的字符串函数。

其语法格式如下：CHAR (integer_expression)

其值介于 0 和 255 之间的整数。如果整数表达式不在此范围内，将返回 NULL 值。

（3）LOWER函数

LOWER函数将大写字符数据转换为小写字符数据后返回字符表达式。

其语法格式如下：LOWER (character_expression)

（4）STR函数

STR函数由数字数据转换来的字符数据。

7.11.1 Transact-SQL简介

其语法格式如下：STR （ float_expression [, length [, decimal]] ）

STR函数由数字数据转换来的字符数据

（5）LTRIM函数

LTRIM函数删除起始空格后返回字符表达式。

其语法格式如下：LTRIM （ character_expression ）

（6）LEFT函数

LEFT函数返回从字符串左边开始指定个数的字符。

其语法格式如下：LEFT （ character_expression , integer_expression ）

（7）RIGHT函数

7.11.1 Transact-SQL简介

RIGHT函数返回字符串中从右边开始指定个数的integer_expression字符。

其语法格式如下：RIGHT (character_expression , integer_expression)

（8）SUBSTRING函数

SUBSTRING函数返回字符、binary、text 或 image 表达式的一部分。

其语法格式如下：SUBSTRING (expression , start , length)

（9）CHARINDEX函数

CHARINDEX函数返回字符串中指定表达式的起始位置。

其语法如下：CHARINDEX (expression1 , expression2

7.11.1 Transact-SQL简介

CHARINDEX函数返回字符串中指定表达式的起始位置。

其语法如下：CHARINDEX(expression1,expression2 [, start_location])

（10）PATINDEX函数

PATINDEX函数返回指定表达式中某模式第一次出现的起始位置；如果在全部有效的文本和字符数据类型中没有找到该模式，则返回零。

其语法格式如下：PATINDEX('%pattern%',expression)

（11）SOUNDEX函数

SOUNDEX函数返回由四个字符组成的代码 (SOUNDEX) 以评估两个字符串的相似性。

7.11.1 Transact-SQL简介

（12）DIFFERENCE函数

DIFFERENCE函数以整数返回两个字符表达式的 SOUNDEX 值之差。

语法格式如下：DIFFERENCE （ character_expression , character_expression ）

（13）QUOTENAME函数

QUOTENAME函数返回带有分隔符的 Unicode 字符串，分隔符的加入可使输入的字符串成为有效的分隔标识符。

其语法格式如下：QUOTENAME （ 'character_string' [, 'quote_character'] ）

7.11.1 Transact-SQL简介

（14）REPLICATE函数

REPLICATE函数以指定的次数重复字符表达式。

其语法格式如下：REPLICATE （ character_expression , integer_expression ）

（15）REPLACE函数

REPLACE函数代表用第三个表达式替换第一个字符串表达式中出现的所有第二个给定字符串表达式。

其语法格式如下：REPLACE （ 'string_expression1' , 'string_expression2' , 'string_expression3' ）

（16）SPACE函数

7.11.1 Transact-SQL简介

（17）STUFF函数

STUFF函数用于删除指定长度的字符并在指定的起始点插入另一组字符。

其语法格式如下：

STUFF (character_expression , start , length , character_expression)

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

7.11.1 Transact-SQL简介

（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])

data_type为SQL Server系统定义的数据类型，用户自定义的数据类型不能在此使用。

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。与其他函数一样，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。

7.11.1 Transact-SQL简介

(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35] 此示例返回从日期 03/12/1998 后的天数。

```
SELECT DAY('03/12/1998') AS 'Day Number'
GO
```

7.11.1 Transact-SQL 简介

（2）MONTH函数

MONTH函数返回代表指定日期月份的整数。

其语法格式如下：MONTH （ date ）

参数date表示datetime 或 smalldatetime 值或日期格式字符串的表达式。仅对 1753 年 1 月 1 日后的日期使用 datetime 数据类型。此函数的返回类型为int。

MONTH 等价于 DATEPART(mm, date)

[例36] 下面的示例从日期 03/12/1998 中返回月份数。

```
SELECT "Month Number" = MONTH('03/12/1998')
```

7.11.1 Transact-SQL简介

GO

（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37] 下例从日期 03/12/1998 中返回年份数。

SELECT "Year Number" = YEAR('03/12/1998')

7.11.1 Transact-SQL简介

GO

（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返回新值的参数。下表7.3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)

7.11.1 Transact-SQL简介

[例38] 此示例确定在 pubs 数据库中标题发布日期和当前日期间的天数。

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

（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

7.11.1 Transact-SQL 简介

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)

7.11.1 Transact-SQL简介

8）GETDATE函数

GETDATE函数的语法格式如下：GERDATE()

GETDATE函数以DATETIME的缺省格式返回系统当前的日期和时间，它常作为其他函数或命令的参数使用。

6.TEXT函数和IMAGE函数

（1）TEXTPTR函数

TEXTPTR函数的语法格式如下：TEXTPTR(column)

TEXTPTR函数返回一个指向存储文本的第一个数据库页的指针。其返回值是一个VARBINARY(16)类型的二进制字符串。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。

7.11.1 Transact-SQL 简介

（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。

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 2000中用户自定义函数作为一个数据库对象来管理，可以使企业管理器或Transact-SQL命令来创建，修改和删除。

7.11.2 存储过程应用初步

在大型数据库中存储过程和触发器具有很重要的作用。存储过程提供了一种高效和安全的访问数据库的方法。经常被用来访问数据和管理被修改的数据。触发器可有助于更好地维护数据库中数据的完整性。就本质而言，触发器也是一种存储过程。存储过程在运算时生成执行方式，所以，以后对其再运行时，其执行速度很快。

一、存储过程概述

存储过程可以使得对数据库的管理、以及显示关于数据库及其用户信息的工作容易得多。

7.11.2 存储过程应用初步

在SQL Server的系列版本中，存储过程分为两类：系统提供的存储过程和用户自定义存储过程，系统存储过程主要存储在master数据库中，并以sp_为前缀，并且系统存储过程主要是从系统表中获取信息，从而为系统管理员管理SQL Server提供支持。

二、创建存储过程

在SQL Server 2000中创建一个存储过程有两种方法：一种是使用Transact-SQL命令，另一种是使用企业管理器。用Transact-SQL创建存储过程是一种较为快速的方法，但对于初学者，使用企业管理器更易理解，更为简单。

7.11.2 存储过程应用初步

当创建存储过程时，需要确定存储过程的三个组成部分：

（1）所有的输入参数以及传给调用者的输出参数

（2）被执行的针对数据库的操作语句，包括调用其他存储过程的语句。

（3）返回给调用者的状态值以指明调用是成功还是失败。

1.用CREATE PROCEDURE命令创建存储过程

用CREATE PROCEDURE命令能够创建存储过程，在创建存储过程之前应考虑到以下几个方面：

（1）在一个批处理中，CREATE PROCEDURE语句不能与其他SQL语句合并在一起。

7.11.2 存储过程应用初步

（2）数据库所有者具有默认的创建存储过程的权限，它可把该权限传递给其他用户。

（3）存储过程作为数据库对象，其命名必须符合命名规则。

（4）只能在当前数据库中创建属于当前数据库的存储过程。

用CREATE PROCEDURE创建存储过程的语法格式如下：

```
CREATE PROCDURE procedure_name [ ; number ][ { @parameter data_type }[ VARYING ] [ = default ] [ OUTPUT ]] [ ,...n ]
[ WITH{ RECOMPILE | ENCRYPTION | RECOMPILE ,
```

7.11.2 存储过程应用初步

AS sql_statement [...n]

各参数含义如下：

（1）procedure_name

新存储过程的名称。过程名必须符合标识符规则，且对于数据库及其所有者必须唯一。要创建局部临时过程，可以在 procedure_name 前面加一个编号符(#procedure_name)，要创建全局临时过程，可以在 procedure_name 前面加两个编号符(##procedure_name)。完整的名称（包括 # 或 ##）不能超过 128 个字符

7.11.2 存储过程应用初步

（2）;number

是可选的整数，用来对同名的过程分组，以便用一条 DROP PROCEDURE 语句即可将同组的过程一起除去。

（3）@parameter

过程中的参数。在 CREATE PROCEDURE 语句中可以声明一个或多个参数。用户必须在执行过程时提供每个所声明参数的值（除非定义了该参数的默认值）。

（4）data_type

参数的数据类型。

说明：对于可以是 cursor 数据类型的输出参数，没有最大数目的限制

7.11.2 存储过程应用初步

（5）VARYING

指定作为输出参数支持的结果集（由存储过程动态构造，内容可以变化）。仅适用于游标参数。

（6）default

参数的默认值。如果定义了默认值，不必指定该参数的值即可执行过程。默认值必须是常量或 NULL。

（7）OUTPUT

表明参数是返回参数。**该选项的值可以返回给** EXEC[UTE]。

（8）RECOMPILE

RECOMPILE 表明 SQL Server 不会保存该过程的计划。

7.11.2 存储过程应用初步

（9）ENCRYPTION

SQL Server 利用存储在 syscomments 中的加密注释来重新创建加密过程。

（10）RECOMPILE, ENCRYPTION

FOR REPLICATION指定不能在订阅服务器上执行为复制创建的存储过程。

（11）AS

指定过程要执行的操作。

（12）sql_statement

过程中要包含的任意数目和类型的 Transact-SQL 语句。

7.11.2 存储过程应用初步

[例39] 在student数据库中，创建一个名称为sproc的存储过程，该存储过程的功能是从数据表S中查询所有女同学的信息。

```
USE student
GO
CREATE PROCEDURE sproc AS
SELECT * FROM S WHERE sex='女'
GO
```

2. 使用企业管理器创建存储过程

使用企业管理器创建存储过程的步骤如下：

7.11.2 存储过程应用初步

（1）展开服务器组，然后展开服务器。

（2）展开"数据库"文件夹，再展开要在其中创建过程的数据库。单击“存储过程”文件夹，此时在右窗格中显示该数据库的所有存储过程。

（3）右击“存储过程”，然后在弹出快捷菜单中单击“新建存储过程”命令。此时打开“存储过程属性”对话框

（4）输入存储过程的文本。

（5）若要检查语法，单击"检查语法"命令。

（6）若要设置权限，单击"权限"命令

（7）单击“确定”按钮保存

三、管理存储过程

1、查看存储过程

7.11.2 存储过程应用初步

（1）通过企业管理器查看存储过程

步骤如下：

1. 展开服务器组，然后展开服务器。

2. 在企业管理器的左窗格中，展开要查看存储过程的"数据库"文件夹，展开存储过程所属的数据库，然后单击"存储过程"文件夹，此时在右窗格中显示该数据库的所有存储过程。

3. 在右窗格中，右击要查看源代码的存储过程，然后单击"属性"命令或双击该存储过程，此时便可看到存储过程的源代码。

（2）使用sp_helptext系统存储过程查看存储过程源代码

7.11.2 存储过程应用初步

[例43] 查看数据库student中存储过程sproc的源代码。

```
exec sp_helptext sproc
```

如果在创建时使用了WITH ENCRYPTION选项，那么无论是使用企业管理器还是使用系统存储过程sp_helptext都无法查看到存储过程的源代码。

2、重命名存储过程

用系统存储过程sp_rename更改当前数据库中用户创建对象（如表、列或用户定义数据类型）的名称。

其语法格式如下：sp_rename 旧存储过程名，新存储过程名

7.11.2 存储过程应用初步

3、执行存储过程

执行已创建的存储过程可使用EXECUTE命令，执行系统过程、用户定义存储过程或扩展存储过程。

其语法格式如下：

```
[ [ EXEC [ UTE ] ] { [ @return_status =]
  { procedure_name [ ;number ] |
  @procedure_name_var }[ [ @parameter = ] { value |
  @variable [ OUTPUT ] | [ DEFAULT ] } [ ,...n ]
  [ WITH RECOMPILE ]
```

@return_status：是一个可选的整型变量，保存存储过程的返回状态。这个变量在用于 EXECUTE 语句前，必须在批处理、存储过程或函数中声明过。

7.11.2 存储过程应用初步

[例46] 执行数据库student中的存储过程sproc。

```
EXECUTE sproc
```

[例47] 执行数据库student中的存储过程InsertRecord。

```
EXECUTE InsertRecord @tno='t1',@tname='张小平',@sex='
   男',@prof='教授', @sal=1000,@comm='300',@dept='计算
   机'
```

[例48] 执行数据库student中的存储过程InsertRecordDefa。

```
EXEC  InsertRecordDfa @tno='t4',@tname='王立',
@sex='女',@sal=1300,@comm=450,@dept='信息'
```

[例49] 执行数据库student中的存储过程query_teacher。

```
DECLARE @tname char(20),
```

7.11.2 存储过程应用初步

4、修改存储过程

使用ALTER PROCEDURE语句更改先前通过执行 CREATE PROCEDURE 语句创建的过程，但不会更改权限，也不影响相关的存储过程或触发器。

其语法格式如下：

```
ALTERPROC[EDURE]procedure_name[;number ][ { @parame
ter data_type }[ VARYING ] [ = default ] [ OUTPUT ]]
[,...n][WITH{RECOMPILE|ENCRYPTION|RECOMPILE,ENCRYPT
ION}] [ FOR REPLICATION ] AS sql_statement [ ...n ]
```

有关 ALTER PROCEDURE 语句所用参数的信息，请参见 CREATE PROCEDURE语句。

7.11.2 存储过程应用初步

另一种方法使用企业管理器修改存储过程

（1）展开服务器组，然后展开服务器。

（2）展开"数据库"文件夹，再展开过程所属的数据库，然后单击"存储过程"文件夹。

（3）在详细信息窗格中，右击存储过程，然后单击"属性"命令。

（4）在"文本"框中，按需要更改存储过程的文本。

（5）若要检查语法，请单击"检查语法"命令。

（6）若要更改权限，单击"权限"命令。

7.11.2 存储过程应用初步

5、删除存储过程

使用DROP PROCEDURE语句删除存储过程，从当前数据库中删除一个或多个存储过程或过程组。

其语法规则如下：DROP PROCEDURE { procedure } [,...n]

其中参数procedure代表是要删除的存储过程或存储过程组的名称。

[例45] 将存储过程sproc从数据库中删除。

```
DROP PROCEDURE sproc
GO
```

7.11.2 存储过程应用初步

另外一种方法还可以使用企业管理器删除存储过程

（1）展开服务器组，然后展开服务器。

（2）展开"数据库"文件夹，展开存储过程所属的数据库，然后单击"存储过程"文件夹。

（3）在详细信息窗格中右击要删除的存储过程，然后单击"删除"命令。

（4）若要查看删除此存储过程对数据库的影响，单击"显示相关性"命令。

（5）单击"全部除去"按钮。

7.11.3 触发器应用初步

1.触发器的概念和作用

触发器是一种特殊类型的存储过程，当使用下面的一种或多种数据修改操作在指定表中对数据进行修改时，触发器会生效：UPDATE、INSERT 或 DELETE。触发器可以查询其它表，而且可以包含复杂的 SQL 语句。它们主要用于强制复杂的业务规则或要求。

使用触发器的优点有：

（1）触发器在对表的数据作了任何修改之后立即被激活。

7.11.3 触发器应用初步

（2）触发器可以通过数据库中的相关表进行层叠更改。

（3）触发器可以强制限制，这些限制比用 CHECK 约束所定义的更复杂。与 CHECK 约束不同的是，触发器可以引用其它表中的列。

（4）存储过程的调用

为了响应数据库更新，触发器可以调用一个或多个存储过程，甚至可以通过外部过程的调用而在DBMS之外进行操作。

由此可见，触发器可以解决高级形式的业务规则或复杂行为限制以及实现定制就等问题。

7.11.3 触发器应用初步

触发器的种类

SQL Serve 2000支持两种类型的触发器：AFTER触发器和INSTEAD OF触发器。

AFTER触发器

3.触发器的原理

触发器中使用了两种特殊的表：deleted 表和 inserted 表。SQL Server 2000 自动创建和管理这些表。可以使用这两个表测试某些数据修改的效果及设置触发器操作的条件。

inserted 和 deleted 表主要用于触发器中：

7.11.3 触发器应用初步

（1）扩展表间引用完整性。

（2）在以视图为基础的基表中插入或更新数据。

（3）检查错误并基于错误采取行动。

（4）找到数据修改前后表状态的差异，并基于此差异采取行动。

4.创建触发器

创建触发器前应考虑下列问题：

（1）CREATE TRIGGER 语句必须是批处理中的第一个语句。将该批处理中随后的其它所有语句解释为 CREATE TRIGGER 语句定义的一部分。

（2）创建触发器的权限默认分配给表的所有者，且不能将该权限转给其他用户。

7.11.3 触发器应用初步

（3）触发器为数据库对象，其名称必须遵循标识符的命名规则。

（4）虽然触发器可以引用当前数据库以外的对象，但只能在当前数据库中创建触发器。

（5）虽然不能在临时表或系统表上创建触发器，但是触发器可以引用临时表。不应引用系统表，而应使用信息架构视图。有关更多信息，请参见信息架构视图。

（6）在含有用 DELETE 或 UPDATE 操作定义的外键的表中，不能定义 INSTEAD OF 和 INSTEAD OF UPDATE 触发器。

7.11.3 触发器应用初步

（7）虽然 TRUNCATE TABLE 语句类似于没有 WHERE 子句（用于删除行）的 DELETE 语句，但它并不会引发 DELETE 触发器，因为 TRUNCATE TABLE 语句没有记录。

（8）WRITETEXT 语句不会引发 INSERT 或 UPDATE 触发器。

方法一：使用CREATE TRIGGER命令创建触发器

CREATE TRIGGER其语法格式如下：

```
CREATE TRIGGER trigger_name ON { table | view }[ WITH
  ENCRYPTION ]
{{ { FOR | AFTER | INSTEAD OF } { [ INSERT ] [ , ]
  [ UPDATE ] } [ WITH APPEND ][ NOT FOR REPLICATION]
```

7.11.3 触发器应用初步

[例50] 创建一个触发器，当向S表中插入一条记录时，自动显示S表中的记录

```
CREATE TRIGGER Change_Display
 ON S
FOR INSERT, UPDATE, DELETE
 AS  SELECT * FROM S
```

方法二：使用企业管理器创建触发器

用企业管理器创建触发器的步骤如下：

7.11.3 触发器应用初步

（1）展开服务器组，然后展开服务器。

（2）展开“数据库”文件夹，展开含触发器的表所属的数据库，然后单击“表”文件夹。

（3）在详细信息窗格中，右击将在其上创建触发器的表，指向"所有任务"菜单，然后单击“管理触发器”命令，出现图7.42。

（4）在“名称”中，单击〈新建〉。

（5）在“文本”框中输入触发器的文本。

（6）若要检查语法，单击“检查语法”命令。

（7）单击“确定”按钮即

7.11.3 触发器应用初步

5. 管理触发器

如果要显示作用与表上的触发器究竟对表有哪些操作，必须查看触发器信息。在SQL Serve中，有许多方法查看触发器信息。我们这里介绍两种常用的方法，即通过企业管理器和系统存储过程来查看触发器信息。

（1）使用企业管理器显示触发器信息

①展开服务器组，然后展开服务器。

②展开“数据库”文件夹，展开含触发器的表所属的数据库，然后单击“表”文件夹。

③在详细信息窗格中，右击触发器所在的表，指向“所有任务”菜单，然后单击“管理触发器”命令。

7.11.3 触发器应用初步

④在“名称”下拉列表中选择所要查看的触发器的名称，在“文本”编辑框中显示出该触发器的文本命令。

（2）使用系统存储过程查看触发器

①系统过程sp_depends、sp_helptext已分别在本章介绍了，请参阅。

[例51] 查看已建立的Change_Display触发器所涉及的表。

```
sp_depends 'Change_Display'
```

[例52] 查看已建立的Change_Display的命令文本。

```
sp_helptext 'Change_Display'
```

7.11.3 触发器应用初步

6.修改与删除触发器

（1）修改触发器

①展开服务器组，然后展开服务器

②展开“数据库”文件夹，展开含触发器的表所属的数据库，然后单击“表”文件夹。

③在详细信息窗格中，右击触发器所在的表，指向“所有任务”菜单，然后单击“管理触发器”命令。

④在“名称”框中选择触发器的名称。

⑤按需要在“文本”字段中更改触发器的文本。按CTRL+TAB 缩进 SQL Server 企业管理器触发器的文本。

⑥若要检查触发器的语法，单击“检查语法”命令。

除此之外，还可使用ALTER TRIGGER命令来修改触发器。

7.11.3 触发器应用初步

（2）删除触发器

①可以使用DROP TRIGGER语句删除触发器

DROP TRIGGER从当前数据库中删除一个或多个触发器。

其语法格式如下：DROP TRIGGER { trigger } [,...n]

参数：trigger是要删除的触发器名称。

例如：DROP TRIGGER Change_Display。能删除Change_Display触发器。

7.11.3 触发器应用初步

②删除触发器所在的表时，SQL Server将自动删除与该表相关的触发器。

③在“触发器属性”对话框中选择要删除的触发器，然后单击“删除”按钮即可。

7.11.3 触发器应用初步

④在“名称”下拉列表中选择所要查看的触发器的名称，在“文本”编辑框中显示出该触发器的文本命令。

（2）使用系统存储过程查看触发器

①系统过程sp_depends、sp_helptext已分别在本章7.9.2、7.7.2中介绍了，请参阅。

7.12 SQL Server 2005系统概述*

Microsoft SQL Server 2005可谓微软“五年研发、十年一剑”的重量级产品，SQL Server 2005这款让微软花去内部一千多名工程师、历时五年才完成更新研发的新产品，是微软具有里程碑意义的企业级数据库产品，SQL Server 2005在企业级支持、商业智能应用、管理开发效率等方面有了显著增强。它提供的集成的数据管理和分析平台，可以帮助组织更可靠的管理来自关键业务的信息、更有效的运行复杂的商业应用；而通过其中集成的报告和数据分析工具，企业可从信息中获得更出色的商业表现力和洞察力。

7.12.1 SQL Server 2005系统简介

1、概况

- （1）什么是 SQL Server 2005
- SQL Server 2005 是一个全面的数据库平台，使用集成的商业智能(Business Intelligence，BI)工具，提供了企业级的数据管理。SQL Server 2005 数据库引擎为关系型数据和结构化数据提供了更安全可靠的存储功能，使您可以构建和管理用于业务的高可用和高性能的数据应用程序。

7.12.1 SQL Server 2005系统简介

SQL Server 2005这款数据库产品是Microsoft 在仔细倾听多方意见反馈，对行业进行了认真研究，在全世界的 Microsoft 研究团队共同努力下，经过创造性思索才最终完成的，这款数据库产品引入了上百种新增功能或改进功能，这些功能将有助于您在下面三个主要方面提高业务：

1）企业数据管理，SQL Server 2005 针对行业和分析两类应用程序提供了一种更安全可靠和更高效的数据平台。SQL Server 的最新版本不仅是迄今为止 SQL Server 的最大发行版本，而且是最为安全的版本。

2）开发人员生产效率，SQL Server 2005 提供了一种端对端的开发环境，其中涵盖了多种新技术，可帮助开发人员大幅度提高生产效率。

3）商业智能，SQL Server 2005 的综合分析、集成和数据迁移功能使各个企业无论采用何种基础平台都可以扩展其现有应用程序的价值。构建于SQL Server 2005的 BI 解决方案使所有员工可以及时获得关键信息，从而在更短的时间内制定更好的决策。

7.12.1 SQL Server 2005系统简介

（2）SQL Server 2005 产品组件功能概述

SQL Server 2005 是用于大规模联机事务处理(OLTP)、数据仓库和电子商务应用的数据库和数据分析平台。其组成部分及功能特点如下表7.4。

（3）SQL Server 2005版本分类

SQL Server 2005根据适用场合、功能规模等的不同，可分为如下六个版本：

1）SQL Server学习版或称精简版（即SQL Server 2005 Express Edition）

2）SQL Server工作组版（即SQL Server 2005 Workgroup Edition）

3）SQL Server开发版（即SQL Server 2005 Developer Edition，功能同SQL Server企业版）

4）SQL Server标准版（即SQL Server 2005 Standard Edition）

5）SQL Server企业版（即SQL Server 2005 Enterprise Edition）

6）SQL Server移动版（即SQL Server 2005 Mobile Edition）

7.12.1 SQL Server 2005系统简介

2、特性

（1）SQL Server 2005的新增或增强功能

SQL Server 2005 扩展了SQL Server 2000的性能、可靠性、可用性、可编程性和易用性。SQL Server 2005包含了多项新功能，这使它成为大规模联机事务处理(OLTP)、数据仓库和电子商务应用程序的优秀数据库平台，新功能如表7.5。

（2）SQL Server 2005的30个最重要特点

以下表格介绍了SQL Server 2005的30个主要特点，它们可以分为三类各10个特点，如下三表。

7.12.2 安装 SQL Server 2005

1、安装 SQL Server 2005

SQL Server 2005安装向导基于Windows安装程序，并提供一个功能树用于安装所有如下SQL Server 2005组件：“数据库引擎”、“Analysis Services”、“Reporting Services”、“Notification Services”、“Integration Services”、“复制”、“管理工具”、“连接组件”、“示例数据库、示例和 SQL Server 2005文档”。

为了成功安装 SQL Server，在安装计算机上需要下列软件组件：1）.NET Framework 2.0；2）SQL Server 本机客户端；3）SQL Server 2005 安装程序支持文件。

这些软件组件将由SQL Server安装程序安装，下面让我们开始安装 SQL Server 2005（评估版）吧。

7.12.2 安装 SQL Server 2005

2、如何验证SQL Server 2005服务的安装成功

若要验证SQL Server 2005安装成功，请确保安装的服务正运行于计算机上。检查SQL Server服务是否正在运行的方法：在“控制面板”中，双击“管理工具”，双击“服务”，然后查找相应的服务显示名称。下表7.9列出服务显示名称及其提供的服务。

7.12.3 SQL Server 2005的主要组件及其初步使用

1、认识安装后的SQL Server 2005

SQL Server 2005安装成功后会在开始菜单中生成类似如下图7.45程序组与程序项：

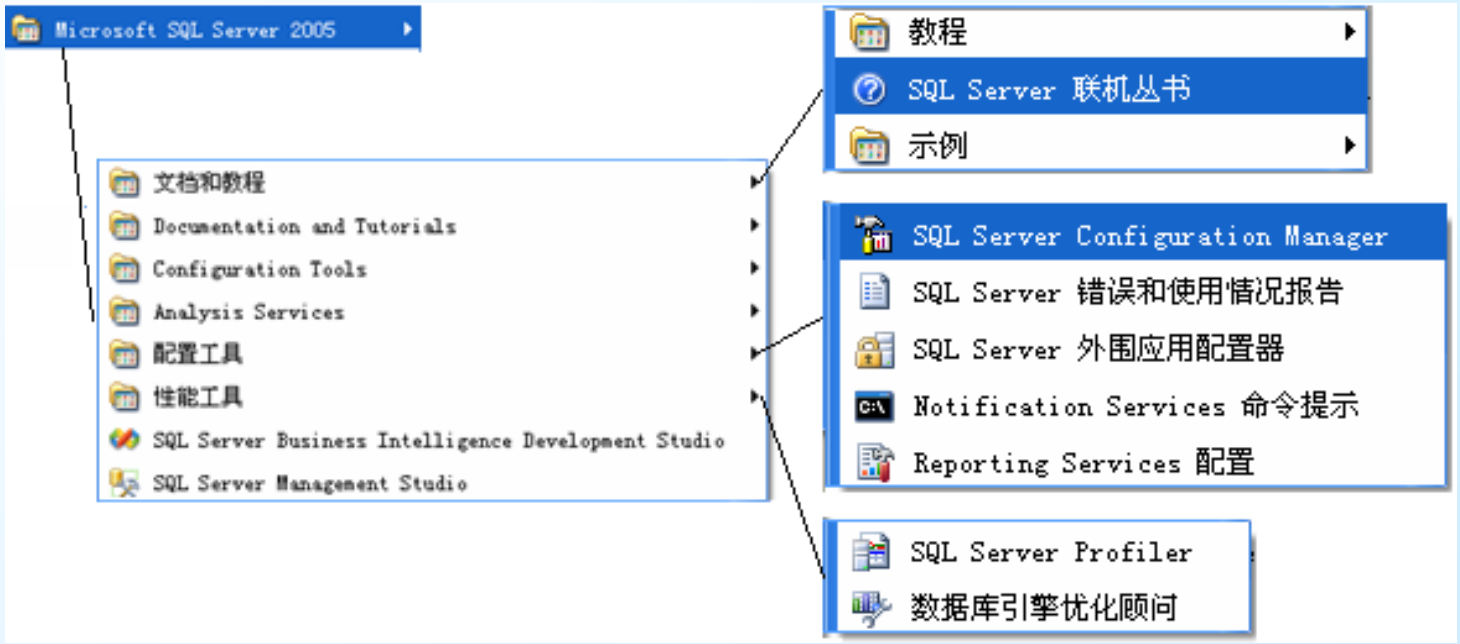


图7.45　安装后SQL Server 2005程序菜单情况

7.12.3 SQL Server 2005的主要组件及其初步使用

2、SQL Server Management Studio（SQL Server集成管理器，SSMS）的使用

SQL Server Management Studio(可称为SQL Server集成管理器，简写为Management Studio，可缩写为SSMS)是为SQL Server数据库管理员和开发人员提供的新工具。此工具由Microsoft Visual Studio 内部承载，它提供了用于数据库管理的图形工具和功能丰富的开发环境。SSMS将SQL Server 2000企业管理器、Analysis Manager和SQL 查询分析器的功能集于一身，还可用于编写MDX、XMLA和XML语句。

7.12.3 SQL Server 2005的主要组件及其初步使用

7.12.3 SQL Server 2005的主要组件及其初步使用

7.12.3 SQL Server 2005的主要组件及其初步使用

3、SQLCMD 实用工具

您可以使用 sqlcmd 实用工具（Microsoft Win32 命令提示实用工具）来运行特殊的 T-SQL 语句和脚本。若要以交互方式使用 sqlcmd，或要生成可使用 sqlcmd 来运行的脚本文件，则需要了解 T-SQL。通常以下列方式使用 sqlcmd 实用工具：

在 sqlcmd 环境中，以交互的方式输入 T-SQL 语句，输入方式与在命令提示符下输入的方式相同。命令提示符窗口中会显示结果（选择其它方式除外）。

您可以通过下列方式提交 sqlcmd 作业：指定要执行的单个 T-SQL 语句，或将实用工具指向包含要执行的 T-SQL 语句的脚本文件。

7.12.3 SQL Server 2005的主要组件及其初步使用

4、SQL Server Configuration Manager（SQL Server配置管理器）

SQL Server 配置管理器管理与SQL Server 2005相关的服务。尽管其中许多任务可以使用 Windows 服务对话框来完成，但值得注意的是 SQL Server 配置管理器还可以对其管理的服务执行更多的操作（例如，在服务帐户更改后应用正确的权限）。使用普通的 Windows 服务对话框配置任何SQL Server 2005服务都可能会造成服务无法正常工作。使用 SQL Server 配置管理器可以完成下列服务任务：1）启动、停止和暂停服务；2）将服务配置为自动启动或手动启动，禁用服务，或者更改其它服务设置；3）更改 SQL Server 服务所使用的帐户的密码；4）使用跟踪标志（命令行参数）启动 SQL Server ；5）查看服务的属性。

7.12.3 SQL Server 2005的主要组件及其初步使用

5、SQL Server 2005外围应用配置器

外围应用减少操作将涉及停止或禁用未使用的组件以增加系统的安全性。对新的 Microsoft SQL Server 2005安装，一些功能、服务和连接将被禁用或停止，以减少 SQL Server 外围应用。对于升级的安装，所有功能、服务和连接将保持其升级前的状态。

使用 SQL Server 外围应用配置器，可以启用、禁用、开始或停止SQL Server 2005安装的一些功能、服务和远程连接。可以在本地和远程服务器中使用 SQL Server 外围应用配置器。

7.12.3 SQL Server 2005的主要组件及其初步使用

6、SQL Server Profiler（事件探查器）

SQL Server Profiler 是一个功能丰富的界面，用于创建和管理跟踪，并分析和重播跟踪结果。对SQL Server Profiler 的使用取决于您出于何种目的监视SQL Server Database Engine 实例。例如，如果您正处于生产周期的开发阶段，则您会更关心如何尽可能地获取所有的性能详细信息，而不会过于关心跟踪多个事件会造成多大的开销。相反，如果您正在监视生产服务器，则会希望跟踪更加集中，并尽可能占用较少的时间，以便尽可能地减轻服务器的跟踪负载。

7.12.3 SQL Server 2005的主要组件及其初步使用

7、数据库引擎优化顾问（Database Engine Tuning Advisor）

数据库引擎优化顾问是 SQL Server 2005中的新工具，使用该工具可以优化数据库，提高查询处理的性能。数据库引擎优化顾问检查指定数据库中处理查询的方式，然后建议如何通过修改物理设计结构（例如索引、索引视图和分区）来改善查询处理性能。

7.12.3 SQL Server 2005的主要组件及其初步使用

8、SQL Server联机丛书

启动方法：“开始”菜单→“所有程序”→“Microsoft SQL Server 2005”→“文档和教程”→“SQL Server联机丛书”，SQL Server联机丛书启动后界面如下图7.73：

该帮助窗体有菜单与工具栏可直观操作，窗口区域分为左右两部分，左边是树型目录结构，能展开各级分类帮助项，右边为含文档链接或超链接的帮助信息，一般有帮助信息的文档页以选项卡形式呈现。

SQL Server 2005 帮助的搜索功能除了可以搜索联机丛书外，还可以搜索网站。如果启用该功能，用户就可以搜索所有 MSDN Online 以及着重于 SQL Server 的社区站点。可以通过搜索页面的“帮助选项”链接或Management Studio选项页面，配置搜索引擎应访问的站点。

7.13 小结

本章是数据库技术的操作与实践内容，可以说是与前面几章所讲述的理论内容相对应的。

在SQL Server 2000下，通过企业管理器或查询分析器及Transact-SQL命令等，可以完成如数据库，数据表，存储过程，视图，触发器，约束和默认等多种数据库对象的管理工作（包括创建，修改，查看，删除等）。

在SQL Server 2005中管理数据库对象同样便捷高效，SQL Server 2005系统的性能与功能等有着跨越式提高，它将会逐步替代SQL Server 2000。

显然，本章要掌握的内容较多，读者应以掌握基本操作为主，而全面深入地使用好SQL Server 2000 & 2005还需在实际工作中逐步积累来实现的。

习　题

1、创建数据库COMPANY和数据表（项目数据表和员工数据表）

使用企业管理器创建下面两个数据表：表B1是项目数据表，表B2是员工数据表，并根据表C1、C2给出的具体内容，进一步对表的结构修改和约束设置。向两表中添加数据。数据如表C1、C2所示。

表B1 项目数据表

字段名	数据类型	注释
项目编号	INT	主键
名称	VARCHAR	长度为50
负责人	INT	
客户	INT	
开始日期	DATETIME	
结束日期	DATETIME	

习　题

表B2 员工数据表

字段名	数据类型	注释
编号	INT	主键
姓名	VARCHAR	长度为50
性别	CHAR	长度为2
所属部门	VARCHAR	长度为50
工资	MONEY	长度为8

习　题

C1 项目数据表

项目编号	名称	负责人	客户	开始日期	结束日期
1	SIS	7	1	06/12/2002	09/01/2004
2	SIS项目2	9	1	04/06/2000	06/12/2002
3	PET	19	2	09/01/2004	04/18/2005
4	PET项目2	7	2	06/17/2005	12/21/2005
5	CCH	8	3	03/21/2003	08/01/2004
6	CCH—LXF	7	3	01/23/2004	09/01/2004
7	CCH-ZHS	19	4	04/18/2005	10/17/2005
8	CCH-LY	9	5	08/01/2004	04/18/2005
9	PETER	4	6	07/11/2004	05/21/2005
10	NBA	4	6	03/06/2005	09/04/2005

习　题

C2员工数据表

编号	姓名	性别	所属部门	工资
1	成有朋	男	项目部	2000
2	张小青	女	项目部	3000
3	孙晓风	男	录入部	1000
4	慕容雪	男	检验部	1500
5	陈秋平	女	检验部	1000
6	王理冰	男	检验部	2000
7	周晴晴	女	办公室	3000
8	杨亭亭	女	项目部	2500
9	马明宇	男	项目部	4000
10	刘燕	女	项目部	3000

习　题

2、在建立的数据库COMPANY中，基于表“项目数据表”和“员工数据表”创建视图，要求为：

（1）视图名为“员工项目”。

（2）包含字段“编号”，“姓名”，“名称”和“开始日期”。

（3）字段别名分别是“员工编号“，“员工姓名”，“项目名称”，“项目开始日期”。

（4）使用INSERT语句通过视图向员工数据表中添加一条记录，要求“姓名”字段值为“马中新”。

（5）执行系统存储过程将视图更名为EMPLOYEEVIEW。

3、在“员工数据表”中基于“姓名”创建索引，要求索引名为IDX_NAME，索引为非聚集索引。

习　题

4、在数据库COMPANY中创建存储过程procedure_salarybydept，要求返回某一特定部门所有员工的工资总和，特定部门的名称以存储过程的输入参数进行传递。

5、表“员工数据表”创建INSERT触发器trigger_newemployeesalary，将插入员工的工资额限制在5000以内。

6、使用企业管理器备份和还原COMPANY数据库。

7、利用所学的数据库后台知识，结合以前学习的前台开发工具，创建一个小型的数据库系统，来操作与管理COMPANY数据库。