

强化互联网时代的管理竞争力

SQL Server 2000数据库基础知识

技术支持部



2005-11-02

KINGDEE 群智互联 智慧企业

主要内容

- 第1章 **SQL Server** 关系数据库简介
- 第2章 服务器管理
- 第3章 数据库管理
- 第4章 数据库对象的操作
- 第5章 **Transact-SQL**语言
- 第6章 SQL Server安全帐户模式
- 第7章 SQL Server代理服务
- 第8章 SQL Server数据复制
- 第9章 SQL Server 数据转换

第1章 SQL Server 关系数据库简介

- 1.1 SQL Server 特点
- 1.2 SQL Server的安装和配置
- 1.3 SQL Server 工具和实用程序

1 SQL Server 关系数据库简介

- SQL Server是由Microsoft开发和推广的关系数据库管理系统（DBMS），它最初是由Microsoft、Sybase和Ashton-Tate三家公司共同开发的，并于1988年推出了第一个OS/2版本。SQL Server近年来不断更新版本，1996年，Microsoft推出了SQL Server 6.5版本；1998年，SQL Server 7.0版本和用户见面；SQL Server 2000是Microsoft公司于2000年推出的最新版本。

1.1 SQL Server 本版

- SQL Server 2000常见的版本
- 企业版（Enterprise Edition）
- 标准版（Standard Edition）
- 个人版（Personal Edition）
- 开 发 者 版 （ Developer Edition）

1.2 SQL Server的安装和配置

- 1.2.1 SQL Server 环境需求
- 1.2.2 SQL Server 2000的安装
- 1.2.3 SQL Server 系统数据库简介
- 1.2.4 SQL Server 系统表简介

1.2.1 安装、运行SQL Server 2000的硬件需求

(1) 计算机

Inter及其兼容计算机，Pentium 166Mz或者更高处理器或DEC Alpha和其兼容系统。

(2) 内存（RAM）

企业版最少64MB内存，其他版本最少需要32MB内存，建议使用更多的内存。

(3) 硬盘空间

完全安装（Full）需要180MB的空间，典型安装（Typical）需要170MB的空间，最小安装（Minimum）需要65MB的空间，只安装管理工具Client tools only 90 MB。

1.2.1 安装、运行SQL Server 2000的软件需求

- SQL Server 2000企业版必须运行于安装Windows NT Server Enterprise Edition 4.0 或者 Windows2000 Advanced Server以及更高版本的操作系统下。
- SQL Server 2000标准版必须运行于安装Windows NT Server Enterprise Edition 4.0、Windows NT Server 4.0、Windows 2000 Server以及更高版本的操作系统下。
- SQL Server 2000 个人版可在多种操作系统下运行，如可运行于Windows 9x， Windows NT 4.0或Windows 2000的服务器版或工作站版的操作系统下。
- SQL Server 2000 开发者版可运行于上述Windows 9x以外的所有操作系统下。

1.2.2 SQL Server 2000的安装

SQL Server 2000的安装过程与其它Microsoft Windows系列产品类似。用户可根据向导提示，选择需要的选项一步一步地完成。

1.2.3 SQL Server 系统数据库简介

SQL Server 2000有6个系统数据库

1. Master数据库是SQL Server系统最重要的数据库，它记录了SQL Server系统的所有系统信息。这些系统信息包括所有的登录信息、系统设置信息、SQL Server的初始化信息和其它系统数据库及用户数据库的相关信息。
2. Model数据库是所有用户数据库和Tempdb数据库的模板数据库，它含有Master数据库所有系统表的子集，这些系统数据库是每个用户定义数据库需要的。

1. 2. 3 SQL Server 系统数据库简介

- 3. Msdb数据库是代理服务数据库，为其警报、任务调度和记录操作员的操作提供存储空间。
- 4. Tempdb是一个临时数据库，它为所有的临时表、临时存储过程及其它临时操作提供存储空间。
- 5. Pubs和Northwind数据库是两个实例数据库，它们可以作为SQL Server的学习工具。

1.2.4 SQL Server 系统表简介

系统目录是由描述SQL Server 系统的数据库、基表、视图和索引等对象的结构系统表组成。SQL Server 经常访问系统目录，检索系统正常运行所需的必要信息。

1.2.4 SQL Server 系统表简介

1. Sysobjects表

SQL Server 的主系统表sysobjects出现在每个数据库中，它对每个数据库对象含有一行记录。

2. Syscolumns表

系统表syscolumns出现在master数据库和每个用户自定义的数据库中，它对基表或者视图的每个列和存储过程中的每个参数含有一行记录。

3. Sysindexes表

系统表sysindexes出现在master数据库和每个用户自定义的数据库中，它对每个索引和没有聚簇索引的每个表含有一行记录，它还对包括文本/图像数据的每个表含有一行记录。

1.2.4 SQL Server 系统表简介

4. Sysusers表

系统表sysusers出现在master数据库和每个用户自定义的数据库中，它对整个数据库中的每个Windows NT用户、Windows NT用户组、SQL Server 用户或者SQL Server 角色含有一行记录。

5. Sysdatabases表

系统表sysdatabases对SQL Server 系统上的每个系统数据库和用户自定义的数据库含有一行记录，它只出现在master数据库中。

序

1.3.1 企业管理器

1.3.2 服务管理器

1.3.3 查询分析器

1.3.4 事件探查器

1.3.5 导入和导出数据

1.3.6 服务器网络实用工具

1.3.7 客户端网络实用工具

1.3.8 联机丛书

1.3.1 企业管理器

- 企业管理器是基于一种新的被称为微软管理控制台（Microsoft Management Console）的公共服务器管理环境，它是SQL Server中最重要的一个管理工具。企业管理器不仅能够配置系统环境和管理SQL Server，而且由于它能够以层叠列表的形式来显示所有的SQL Server对象，因而所有SQL Server对象的建立与管理都可以通过它来完成。

1.3.1 企业管理器

- 利用企业管理器可以完成的操作有
 - 管理SQL Server 服务器；
 - 建立与管理数据库；
 - 建立与管理表、视图、存储过程、触发程序、角色、规则、默认值等数据库对象，以及用户定义的数据类型；
 - 备份数据库和事务日志、恢复数据库，复制数据库；
 - 设置任务调度；
 - 设置警报；
 - 提供跨服务器的拖放控制操作；
 - 管理用户帐户；
 - 建立 Transact-SQL 命令语句以及管理和控制 SQL Mail。

1.3.1 企业管理器

SQL Server 企业管理器（Enterprise Manager）界面



1.3.2 服务管理器

- SQL Server 服务管理器是在服务器端实际工作时最有用的实用程序，其界面如图1-21所示。服务管理器用来启动、暂停、继续和停止数据库服务器的实时服务，其提供的服务包括：SQL Server、SQL Server Agent、MSDTC（Microsoft Distributed Transaction Coordinator，微软分布式事务协调器），microsoft search。

1.3.2 服务管理器

SQL Server 服务管理器 (Service Manager) 界面



1.3.3 查询分析器

- SQL Server 2000新的图形化查询分析器用于输入和执行Transaction-SQL语句，并且迅速查看这些语句的结果，以分析和处理数据库中的数据。这是一个非常实用的工具，对掌握SQL语言，深入理解SQL Server的管理工作有很大帮助。

1.3.4 事件探查器

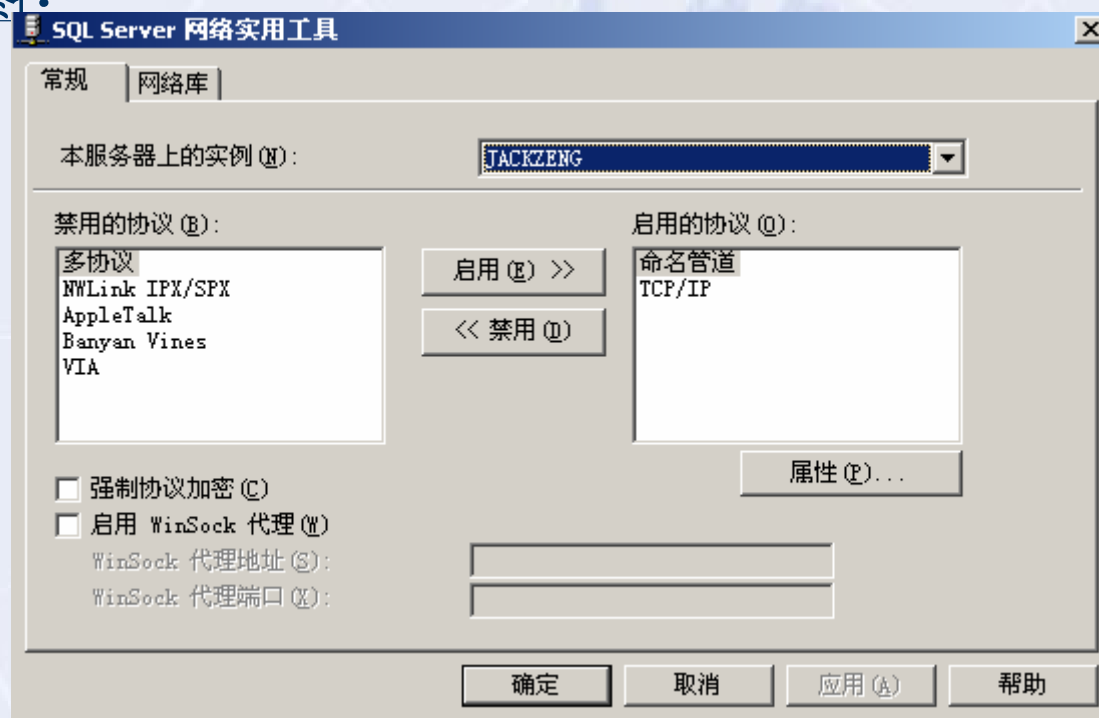
- 是一个图形化的管理工具，用于监督、记录和检查SQL Server 数据库的使用情况。对系统管理员来说，它是一个连续实时地捕获用户活动情况的间谍。

1.3.5 导入和导出数据

- 导入和导出数据采用DTC 导入/导出向导来完成。此向导包含了所有的DTC（Data Transformation Services）工具，提供了在OLE DB数据源之间复制数据的最简捷的方法。

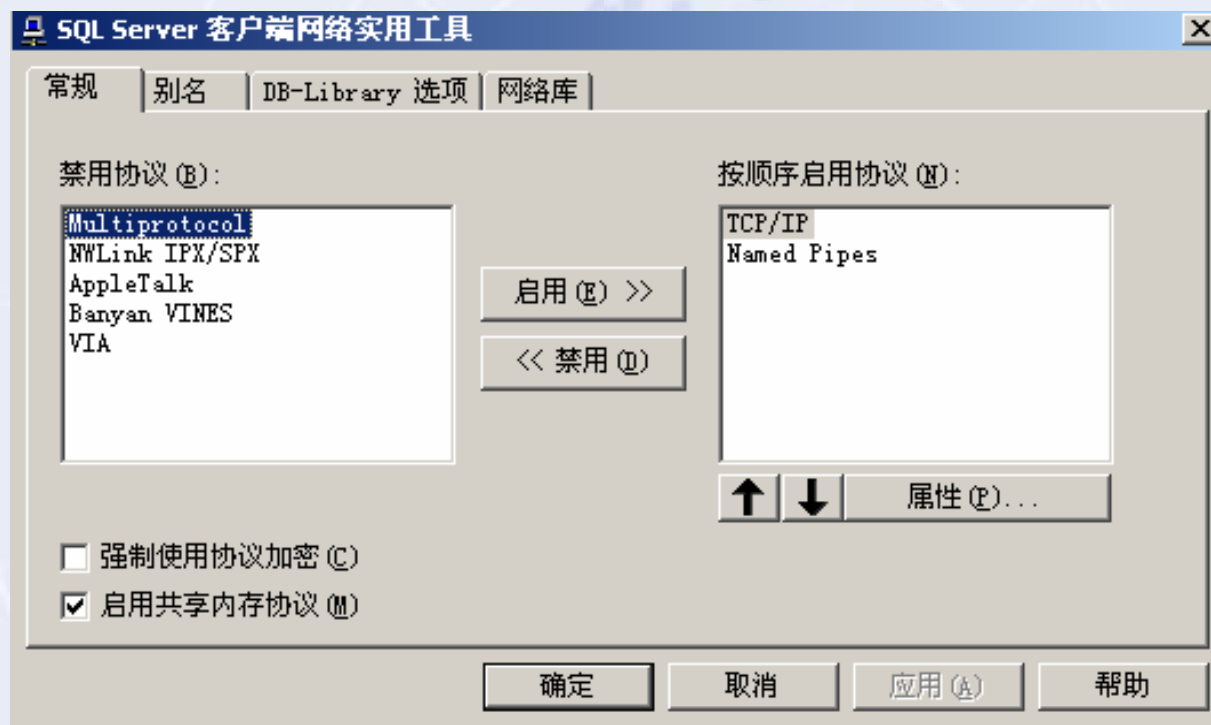
1.3.6 服务器网络实用工具

用来配置服务器端网络连接和设置等相关参数，如图。



1.3.7 客户端网络实用工具

用来配置客户端的网络连接、管理和测定客户端的网络库等，如图：



第2章 服务器管理

- 2.1 创建服务器组
- 2.2 服务器注册
- 2.3 服务器启动、暂停和停止
- 2.4 服务器连接与断开
- 2.5 服务器配置选项设置

2.1 创建服务器组

- 在一个网络系统中，可能有多个SQL Server服务器，可以对这些SQL Server服务器进行分组管理。分组的原则往往是依据组织结构原则。SQL Server分组管理由企业管理器（Enterprise Manager）来进行。首次启动企业管理器时，有一个名字叫“SQL Server 组”的默认服务器组自动被创建，但用户可以创建新的SQL Server组，在企业管理器菜单中单击操作（Action）菜单或者用右键单击SQL Servers组，选择新建SQL Server 组菜单项，接着出现服务器组对话框，输入组的名称并选择组的级别，单击确定即可。

2.2 服务器注册

- 所谓服务器注册是指将网络系统中的其它SQL Server服务器注册到企业管理器中，以便于管理。

2.3 服务器启动、暂停和停止

- 启动服务器，下面介绍常用的三种方法：
- 1. 自动启动服务器
- 2. 用SQL Server 服务管理器启动
- 3. 用企业管理器（Enterprise Manager）启动

2.3 暂停和关闭服务器

- 暂停和关闭服务器的方法与启动服务器的方法类似，只需在相应的窗口中选择“暂停（Pause）”或“停止（Stop）”选项即可。为了保险起见，我们通常会在停止运行SQL Server之前先暂停SQL Server。其主要原因在于，一旦暂停SQL Server，将不再允许任何新的上线者，然而原先已联机到SQL Server的用户仍然能继续作业，这样可以确保原来正在进行的作业不会中断，而可以持续进行并加以完成。

2.4 服务器连接与断开

- 在SQL Server 企业管理器的SQL Server组中用左键单击所要连接的服务器，或者用右键单击所要连接的服务器，从快捷菜单中选择“连接”选项，即可连接。如果在注册服务器时选择了在连接时提示输入SQL Server 帐户信息选项，这时会提示要录入登录用户名和口令。
- 同样，在SQL Server 企业管理器的SQL Server组中用左键单击所要断开的服务器，或者用右键单击所要连接的服务器，从快捷菜单中选择“断开”选项，即可断连接。

2.5 服务器配置选项设置

- 使用SQL Server企业管理器配置服务器的操作方法为：在企业管理器的SQL Server组（Group）中用右键单击所要进行配置的服务器，从快捷菜单中选择“属性（Properties）”选项，就可以进行服务器的属性（配置选项）的设置。

2.5

服务器属性设置对话框



第3章 数据库管理

- 3.1 数据库的存储结构
- 3.2 创建、修改和删除数据库
- 3.3 数据库备份
- 3.4 恢复数据库

3.1

数据库的存储结构

- 3.1.1 数据库文件
- 3.1.2 数据库文件组

3.1.1 数据库文件

- 逻辑存储结构和物理存储结构
 - 一、数据库的逻辑存储结构指的是数据库是由哪些性质的信息所组成，SQL Server的数据库不仅仅只是数据的存储，所有与数据处理操作相关的信息都存储在数据库中。
 - 二、数据库的物理存储结构则是讨论数据库文件是如何在磁盘上存储的，数据库在磁盘上是以文件为单位存储的，由数据库文件和事务日志文件组成，一个数据库至少应该包含一个数据库文件和一个事务日志文件。

3.1.1 数据库文件

1. 主数据库文件（Primary Database File）
2. 辅助数据库文件（Secondary Database File）
3. 事务日志文件

3.1.2 数据库文件组

- 为了便于分配和管理，SQL Server允许将多个文件归纳为同一组，并赋予此组一个名称，这就是文件组。
- 与数据库文件一样，文件组也分为主文件组（Primary File Group）和次文件组（Secondary File Group）。

3.2 创建、修改和删除数据库

- 3.2.1 创建数据库
- 3.2.2 修改数据库
- 3.2.3 删除数据库

3.2.1 创建数据库

- 每个数据库都由以下几个部分的数据库对象所组成：
- 关系图、表、视图、存储过程、用户、角色、规则、默认、用户自定义数据类型和用户自定义函数。

3.2.1 创建数据库的方法

- 使用向导创建数据库
- 使用企业管理器（Enterprise Manager）创建数据库
- 使用Transact-SQL语言创建数据库。

3.2.2 修改数据库

- 1. 利用企业管理器修改数据库
- 2. 使用ALTER DATABASE语句修改数据库

3.2.3 删除数据库

- 1.利用企业管理器删除数据库
- 2. 利用Drop语句删除数据库
- Drop语句可以从SQL Server中一次删除一个或多个数据库。其语法如下：
- Drop database database_name[,...n]

3.3

数据库备份

- 3.3.1 备份方式
- 3.3.2 创建备份设备
- 3.3.3 备份的执行

式

- 1、完全数据库备份（Database-complete）
- 2、差异备份或称增量备份（Database-differential）
- 3、事务日志备份（Transaction log）
- 4、数据库文件和文件组备份（File and filegroup）

3.3.2 创建备份设备

- 在进行备份以前首先必须指定或创建备份设备，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，备份设备可以是硬盘、磁带或管道。当使用磁盘时，SQL Server允许将本地主机硬盘和远程主机上的硬盘作为备份设备，备份设备在硬盘中是以文件的方式存储的。

3.3.2 创建备份设备的两种方法

- 1、使用SQL Server 企业管理器创建备份设备
- 2、使用系统存储过程创建备份设备

在SQL Server 中，可以使用sp_addumpdevice 语句创建备份设备，其语法形式如下：

```
sp_addumpdevice {'device_type'}  
[, 'logical_name'][, 'physical_name'][,  
{controller_type|'device_status'}]
```

例11：在磁盘上创建一个备份设备

```
use master  
exec sp_addumpdevice 'disk',  
    'backup_company',  
    'c:\mssql7\backup\bk_company.bak'
```

3.3.2 删除备份设备

- 删除备份设备与创建的过程类似，只须在企业管理器中选中要删除的备份设备，在弹出的菜单中选择删除选项即可删除该备份设备。或者使用sp_dropdevice语句来删除备份设备。其语法如下：

- sp_dropdevice
['logical_name'][, 'delfile']

- 例子：删除上面创建的备份设备

- sp_dropdevice 'backup_company',
'c:\mssql7\backup\bk_company.bak'

3.3.3 备份的执行

- SQL Server系统提供了两种备份操作的方法：
- (1)SQL Server 企业管理器中的备份向导
- (2)Transact-SQL语句。语法形式如下：

Backup database

{database_name|@database_name_var}

to <backup_file>[,...n]

[with [[,]format] [[,]{init|noinit}]

[[,]restart]]

<backup_file>::={backup_file_name|@backup_file_evar}|{disk|tape|pipe}

={temp_file_name|@temp_file_name_evar}

3.4

恢复数据库

- 3.4.1 恢复概述
- 3.4.2 恢复数据库
- 3.4.3 恢复系统数据库

3.4.1 恢复概述

- 数据库备份后，一旦系统发生崩溃或者执行了错误的数据库操作，就可以从备份文件中恢复数据库。数据库恢复是指将数据库备份加载到系统中的过程。系统在恢复数据库的过程中，自动执行安全性检查、重建数据库结构以及完整数据库内容。

3.4.2 恢复数据库

- SQLServer系统有两种恢复数据库的方法：
- (1)SQL Server 企业管理器中的恢复向导
- (2)Transact-SQL语句。

3.4.3 恢复系统数据库

- 1. 关闭SQL Server，运行系统安装目录下的bin子目录下的rebuilem.exe文件，这是个命令行程序，运行后可以重新创建系统数据库。
- 2. 系统数据库重新建立后，启动SQL Server。
- 3. SQL Server启动后，系统数据库是空的，没有任何系统信息。因此，需要从备份数据库中恢复。一般是先恢复master数据库，再恢复msdb数据库，最后恢复model数据库。

第4章 数据库对象

- 4.1 数据类型
- 4.2 表
- 4.3 视图
- 4.4 索引
- 4.5 存储过程
- 4.6 触发器
- 4.7 图 表

4.1

数据类型

- 4.1.1 系统数据类型
- 4.1.2 自定义数据类型

4.1.1 系统数据类型

- 1. 整型数据类型
- 2. 浮点数据类型
- 3. 字符数据类型
- 4. 日期和时间数据类型
- 5. 文本和图形数据类型
- 6. 货币数据类型
- 7. 位数据类型
- 8. 二进制数据类型
- 9. 特殊数据类型
- 10. 新增数据类型

4. 2 表

1. 创建表
2. 增加 修改表
3. 创建、删除和修改约束
4. 查看表格

4.3 视图

1. 视图的概念和作用
2. 创建视图
3. 修改和重命名视图
4. 查看视图信息、删除视图
5. 通过视图修改记录

视图的概念:

- 视图是从一个或者多个表或视图中导出的表，其结构和数据是建立在对表的查询基础上的。和真实的表一样，视图也包括几个被定义的数据列和多个数据行，但从本质上讲，这些数据列和数据行来源于其所引用的表。因此，视图不是真实存在的基础表而是一个虚拟表，视图所对应的数据并不实际地以视图结构存储在数据库中，而是存储在视图所引用的表中。

使用视图的优点和作用

- ①可以使视图集中数据、简化和定制不同用户对数据库的不同数据要求。
- ②使用视图可以屏蔽数据的复杂性，用户不必了解数据库的结构，就可以方便地使用和管理数据，简化数据权限管理和重新组织数据以便输出到其他应用程序中。
- ③视图可以使用户只关心他感兴趣的某些特定数据和他们所负责的特定任务，而那些不需要的或者无用的数据则不在视图中显示。
- ④视图大大地简化了用户对数据的操作。

使用视图的优点和作用

- ⑤视图可以让不同的用户以不同的方式看到不同或者相同的数据集。
- ⑥在某些情况下，由于表中数据量太大，因此在表的设计时常将表进行水平或者垂直分割，但表的结构的变化对应用程序产生不良的影响。
- ⑦视图提供了一个简单而有效的安全机制。

4.4

索引

1. 索引的概念和作用
2. 创建索引
3. 查看、修改和删除索引

4.4.1 什么叫索引

- 数据库中的索引与书籍中的索引类似，在一本书中，利用索引可以快速查找所需信息，无须阅读整本书。在数据库中，索引使数据库程序无须对整个表进行扫描，就可以在其中找到所需数据。书中的索引是一个词语列表，其中注明了包含各个词的页码。而数据库中的索引是某个表中一列或者若干列值的集合和相应的指向表中物理标识这些值的数据页的逻辑指针清单。

4.4.1 聚集索引与非聚集索引

- 聚集索引对表的物理数据页中的数据按列进行排序，然后再重新存储到磁盘上，即聚集索引与数据是混为一体的，它的叶节点中存储的是实际的数据。
- 非聚集索引具有完全独立于数据行的结构，使用非聚集索引不用将物理数据页中的数据按列排序。非聚集索引的叶节点存储了组成非聚集索引的关键字值和行定位器。

4.4.1 索引的作用

- 通过创建唯一索引，可以保证数据记录的唯一性。
- 可以大大加快数据检索速度。
- 可以加速表与表之间的连接，这一点在实现数据的参照完整性方面有特别的意义。
- 在使用ORDER BY和GROUP BY子句中进行检索数据时，可以显著减少查询中分组和排序的时间。
- 使用索引可以在检索数据的过程中使用优化隐藏器，提高系统性能。

4.4.2 创建索引

- SQL Server 2000创建索引的方法
- （1）利用企业管理器中的索引向导创建索引。
- （2）利用企业管理器直接创建索引。
- （3）利用Transact-SQL语句中的
CREATE INDEX命令创建索引。
- （4）利用企业管理器中的索引优化向导创建索引。

4.4.3 查看、修改和删除索引

1. 利用企业管理器查看、修改和删除索引

要查看和修改索引的详细信息，可以在企业管理器中，展开指定的服务器和数据库项，用右键单击要查看的表，从弹出的快捷菜单中选择所有任务子菜单中的“管理索引”选项，则会出现管理索引对话框。选择要查看或者修改的索引，单击“编辑”按钮，就会出现修改索引对话框。在该对话框中，可以修改索引的大部分设置，还可以直接修改其SQL脚本，只需按下“编辑SQL...”按钮，即可出现编辑SQL脚本对话框，其中可以编辑、测试和运行索引的SQL脚本。

要删除索引，可以在企业管理器中，从管理索引对话框中或者表的属性对话框中，选择要删除的索引，单击“删除”按钮，即可删除索引。

4.4.3 查看、修改和删除索引

- 2. 用系统存储过程查看和更改索引名称

系统存储过程sp_helpindex可以返回表的所有索引信息，其语法形式如下：

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

其中[@objname='name'参数用于指定当前数据库中的表的名称。

另外，系统存储过程sp_rename可以用来更改索引的名称，其语法形式如下：

```
sp_rename [@objname=] 'object_name',  
          [@newname=] 'new_name'  
[ , [ @objtype = ] 'object_type' ]
```

4.4.3 查看、修改和删除索引

- 3. 使用Transact-SQL语句中的DROP INDEX命令删除索引

当不再需要某个索引时，可以将其删除，DROP INDEX命令可以删除一个或者多个当前数据库中的索引，其语法形式如下：

```
DROP INDEX 'table.index | view.index' [ ,...n ]
```

其中，table | view用于指定索引列所在的表或索引视图；index用于指定要删除的索引名称。注意，DROP INDEX命令不能删除由CREATE TABLE或者ALTER TABLE命令创建的主键或者唯一性约束索引，也不能删除系统表中的索引。

4.5 存储过程

- 4.5.1 存储过程的概念
- 4.5.2 创建存储过程
- 4.5.3 执行存储过程
- 4.5.4 查看和修改存储过程
- 4.5.5 重命名和删除存储过程

4.5.1 存储过程的概念

- SQL Server提供了一种方法，它可以将一些固定的操作集中起来由SQL Server数据库服务器来完成，以实现某个任务，这种方法就是存储过程。
- 在SQL Server中存储过程分为两类：即系统提供的存储过程和用户自定义的存储过程。

4.5.2 创建存储过程

- 在SQL Server中，可以使用三种方法创建存储过程：
 - ①使用创建存储过程向导创建存储过程。
 - ②利用SQL Server 企业管理器创建存储过程。
 - ③使用Transact-SQL语句中的CREATE PROCEDURE命令创建存储过程。

4.5.2 创建存储过程

- 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，包括调用其它存储过程的语句。
- ③返回给调用者的状态值，以指明调用是成功还是失败。

4.5.3 执行存储过程

- 直接执行存储过程可以使用EXECUTE命令来执行，其语法形式如下：

- [[EXEC[UTE]]
 {
 [@return_status=
 {procedure_name[;number
]|@procedure_name_var}
 [[@parameter={value|@variable[0
 UTPUT]|[DEFAULT]}
 [,...n]
 [WITH RECOMPILE]

4.5.4 查看和修改存储过程

- 存储过程被创建之后，它的名字就存储在系统表 `sysobjects` 中，它的源代码存放在系统表 `syscomments` 中。可以使用使用企业管理器或系统存储过程来查看用户创建的存储过程。

4. 6

触发器

- 4.6.1 触发器的概念和用途
- 4.6.2 创建触发器
- 4.6.3 查看、修改和删除触发器
- 4.6.4 触发器的应用

4.6.1 触发器的概念

- 触发器是一种特殊类型的存储过程，它不同于前面介绍过的存储过程。触发器主要是通过事件进行触发而被执行的，而存储过程可以通过存储过程名称而被直接调用。触发器是一个功能强大的工具，它使每个站点可以在有数据修改时自动强制执行其业务规则。触发器可以用于 SQL Server 约束、默认值和规则的完整性检查。

4.6.1 触发器的用途

- ❖ 触发器是自动的：当对表中的数据作了任何修改（比如手工输入或者应用程序采取的操作）之后立即被激活。
- ❖ 触发器可以通过数据库中的相关表进行层叠更改。
- ❖ 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。

4.6.2 创建触发器

- 创建触发器应该考虑以下几个问题：
- ①CREATE TRIGGER 语句必须是批处理中的第一个语句。
- 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。
- 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。

4.6.2 创建触发器

- 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。
- ⑧WRITETEXT 语句不会引发 INSERT 或 UPDATE 触发器。
- ⑨当创建一个触发器时必须指定：(1)名称；(2)在其上定义触发器的表；(3)触发器将何时激发；(4)激活触发器的数据修改语句。

4.6.3 查看、修改和删除触发器

- 1、查看触发器

- (1) 使用企业管理器查看触发器信息

- (2) 使用系统存储过程查看触发器

4.6.3 查看、修改和删除触发器

- 2、修改触发器

可以使用企业管理器来修改触发器正文：在企业管理器中，展开指定的服务器和数据库，选择指定的数据库和表，用右键单击要修改的表，从弹出的快捷菜单中选择所有任务子菜单下的管理触发器选项，则会出现触发器属性对话框。在名称选项框中选择要修改的触发器的名称，然后在文本框中修改触发器的SQL语句，单击“检查语法”按钮，可以检查语法是否正确。

4.6.3 查看、修改和删除触发器

- 3. 删除触发器

（1）使用系统命令DROP TRIGGER删除指定的触发器，其语法形式如下：DROP TRIGGER { trigger } [,...n]

（2）删除触发器所在的表时，SQL Server将会自动删除与该表相关的触发器。

（3）在企业管理器中，用右键单击要删除的触发器所在的表，从弹出的快捷菜单中选择所有任务子菜单下的管理触发器选项，则会出现触发器属性对话框。在名称选项框中选择要删除的触发器，单击“删除”按钮，即可删除该触发器。

4.6.4 触发器的应用

- 使用嵌套的触发器

如果一个触发器在执行操作时引发了另一个触发器，而这个触发器又接着引发下一个触发器……这些触发器就是嵌套触发器。触发器可嵌套至 32 层，并且可以控制是否可以通过“嵌套触发器”服务器配置选项进行触发器嵌套。如果允许使用嵌套触发器，且链中的一个触发器开始一个无限循环，则超出嵌套级，而且触发器将终止。在执行过程中，如果一个触发器修改某个表，而这个表已经有其它触发器，这是就要使用嵌套触发器。

4.7

图表

- 图表（又称关系图）是SQL Server中一类特殊的数据库对象，它提供给用户直观地管理数据库表的方法。通过图表，用户可以直观地创建、编辑数据库表之间的关系，也可以编辑表及其列的属性。

第5章 Transact-SQL语言

- 5.1 数据定义语言 (DDL)
- 5.2 数据操纵语言 (DML)
- 5.3 数据控制语言 (DCL)
- 5.4 系统存储过程
- 5.5 其它语言元素

5.1 数据定义语言（DDL）

数据定义语言是指用来定义和管理数据库以及数据库中的各种对象的语句，这些语句包括CREATE、ALTER和DROP等语句。在SQL Server 2000中，数据库对象包括表、视图、触发器、存储过程、规则、缺省、用户自定义的数据类型等。这些对象的创建、修改和删除等都可以通过使用CREATE、ALTER、DROP等语句来完成。

例1: 创建一个数据库表

```
CREATE TABLE publishers
(
  pub_id  char(4)  NOT NULL
  CONSTRAINT UPKCL_pubind PRIMARY KEY CLUSTERED
  CHECK(pub_id IN ('1389','0736','0877','1622',
    '1756') OR pub_id LIKE '99[0-9][0-9]'),
  pub_name varchar(40) NULL,
  city  varchar(20) NULL,
  state char(2)  NULL,
  country  varchar(30) NULL DEFAULT('USA')
)
```

例2：修改一个视图

```
USE pubs
GO
ALTER VIEW yourview
AS
SELECT    title,      mycount=  @@      ROWCOUNT,
          ytd_sales
          FROM titles
          WHERE type = 'mod_cook'
GO
SELECT *
FROM yourview
GO
```

例3：删除一个触发器

/*下面的例子将删除触发器employee_insupd。*/

```
USE pubs
IF EXISTS (SELECT name FROM sysobjects
           WHERE      name
           'employee_insupd'
           AND type = 'TR')
DROP TRIGGER employee_insupd
GO
```

5.2 数据操纵语言（DML）

数据操纵语言是指用来查询、添加、修改和删除数据库中数据的语句，这些语句包括SELECT、INSERT、UPDATE、DELETE等。在默认情况下，只有sysadmin、dbcreator、db_owner或db_datawriter等角色的成员才有权利执行数据操纵语言。

5.2.1 SELECT语句

SELECT语句的语法形式如下：

```
SELECT select_list
[ INTO new_table ]
FROM table_source
[ WHERE search_condition ]
[ GROUP BY group_by_expression ]
[ HAVING search_condition ]
[ ORDER BY order_expression [ ASC | DESC ] ]
[ COMPUTE clause ]
[ FOR BROWSE ]
```

几种常用语句的用法1

- SELECT子句用于指定所选择的要查询的特定表中的列，它可以是星号（*）、表达式、列表、变量等。
- INTO子句用于指定所要生成的新表的名称。
- FROM子句用于指定要查询的表或者视图，最多可以指定16个表或者视图，用逗号相互隔开。
- WHERE子句用来限定查询的范围和条件。
- GROUP BY子句是分组查询子句。
- HAVING子句用于指定分组子句的条件。

几种常用语句的用法 2

- GROUP BY子句、HAVING子句和集合函数一起可以实现对每个组生成一行和一个汇总值。
- ORDER BY子句可以根据一个列或者多个列来排序查询结果，在该子句中，既可以使用列名，也可以使用相对列号。
- ASC表示升序排列，DESC表示降序排列。
- COMPUTE子句使用集合函数在查询的结果集中生成汇总行。
- COMPUTE BY子句用于增加各列汇总行。

例4：显示居住在加利福尼亚州且姓名不为 **McBadden** 的作者列

```
USE pubs
SELECT au_fname, au_lname, phone AS Telephone
FROM authors
WHERE state = 'CA' and au_lname <> 'McBadden'
ORDER BY au_lname ASC, au_fname ASC
```

例5：使用两个相关子查询查找作者姓名，这些作者至少参与过一本受欢迎的计算机书籍的创作。

```
USE pubs
SELECT au_lname, au_fname
FROM authors
WHERE au_id IN
    (SELECT au_id
     FROM titleauthor
     WHERE title_id IN
         (SELECT title_id
          FROM titles
          WHERE type = 'popular_comp'))
    )
```

例6：显示含有本年度截止到现在的当前销售额的行，然后按type以递减顺序计算书籍的平均价格和预付款总额。

```
USE pubs
SELECT CAST(title AS char(20)) AS
       title, type, price, advance
FROM titles
WHERE ytd_sales IS NOT NULL
ORDER BY type DESC
COMPUTE AVG(price), SUM(advance)
       BY type
COMPUTE SUM(price), SUM(advance)
```

例7：先删除表T1，然后又创建一个带有四个列的表T1。最后利用INSERT语句插入一些数据行，这些行只有部分列包含值。

```
IF EXISTS(SELECT TABLE_NAME FROM
          INFORMATION_SCHEMA.TABLES
          WHERE TABLE_NAME = 'T1')
  DROP TABLE T1
GO
CREATE TABLE T1
( column_1 int identity,
  column_2 varchar(30)
  CONSTRAINT default_name DEFAULT
  ('column default'),
```

续

```
column_3 int NULL,
column_4 varchar(40)
)
INSERT INTO T1 (column_4)
VALUES ('Explicit value')
INSERT INTO T1 (column_2,column_4)
VALUES ('Explicit value', 'Explicit
value')
INSERT INTO T1 (column_2,column_3,column_4)
VALUES ('Explicit value',-44,'Explicit
value')
SELECT *
FROM T1
```

例8：一个简单的修改更新语句

```
UPDATE publishers  
SET city = 'Atlanta', state = 'GA'
```

例9：一个带有WHERE条件的修改语句

```
UPDATE authors  
SET state = 'PC', city = 'Bay City'  
WHERE state = 'CA' AND city = 'Oakland'
```

例10: 通过 UPDATE 语句使用来自另一个表的信息, 其修改表 titles 中的 ytd_sales 列, 以反映表 sales 中的最新销售记录

```
UPDATE titles
SET ytd_sales = titles.ytd_sales + sales.qty
FROM titles, sales
WHERE titles.title_id = sales.title_id
      AND sales.ord_date = (SELECT
                           MAX(sales.ord_date) FROM sales)
```

5.3 数据控制语言（DCL）

数据控制语言（DCL）是用来设置或者更改数据库用户或角色权限的语句，这些语句包括GRANT、DENY、REVOKE等语句，在默认状态下，只有sysadmin、dbcreator、db_owner或db_securityadmin等角色的成员才有权利执行数据控制语言。

5.4

系统存储过程

- 系统存储过程是SQL Server系统创建的存储过程，它的目的在于能够方便地从系统表中查询信息，或者完成与更新数据库表相关的管理任务或其它的系统管理任务。系统存储过程可以在任意一个数据库中执行。系统存储过程创建并存放于系统数据库master中，并且名称以sp_或者xp_开头。

下：

- **sp_addtype**：用于定义一个用户定义数据类型。
- **sp_configure**：用于管理服务器配置选项设置。
- **xp_sendmail**：用于发送电子邮件或寻呼信息。
- **sp_stored_procedures**：用于返回当前数据库中的存储过程的清单。

5.4.1 系统存储过程的部分示例如下：

- **sp_help**：用于显示参数清单和其数据类型。
- **sp_depends**：用于显示存储过程依据的对象或者依据存储过程的对象。
- **sp_helptext**：用于显示存储过程的定义文本。
- **sp_rename**：用于修改当前数据库中用户对象的名称。

5.4.2 局部变量

局部变量是一个能够拥有特定数据类型的对象，它的作用范围仅限制在程序内部。局部变量可以作为计数器来计算循环执行的次数，或是控制循环执行的次数。另外，利用局部变量还可以保存数据值，以供控制流语句测试以及保存由存储过程返回的数据值等。局部变量被引用时要在其名称前加上标志“@”，而且必须先用**DECLARE**命令定义后才可以使用。

5.4.3 全局变量

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

5.5 其他语言元素

- 5.5.1 函数
- 5.5.2 流程控制语句

5.5.1 函数

- 在Transact-SQL语言中，函数被用来执行一些特殊的运算以支持SQL Server的标准命令。Transact-SQL 编程语言提供了三种函数：
- 行集函数：行集函数可以在Transact-SQL语句中当作表引用。
- 聚合函数：聚合函数用于对一组值执行计算并返回一个单一的值。
- 标量函数：标量函数用于对传递给它的一个或者多个参数值进行处理和计算，并返回一个单一的值。

5.5.1 SQL Server中最常用的几种函数

- 字符串函数
- 日期和时间函数
- 数学函数
- 转换函数
- 系统函数
- 聚合函数
- 其它函数

5.5.2 流程控制语句

- 流程控制语句是指那些用来控制程序执行和流程分支的命令，在SQL Server 2000中，流程控制语句主要用来控制SQL语句、语句块或者存储过程的执行流程。

5.5.2 流程控制语句

1. **IF...ELSE**语句
2. **BEGIN...END**语句
3. **CASE**函数
4. **WHILE...CONTINUE...BREAK**语句
5. **GOTO**语句
6. **WAITFOR**语句
7. **RETURN**语句

第6章 SQL Server安全帐户模式

- 安全帐户认证是用来确认登录SQL Server的用户的登录帐号和密码的正确性，由此来验证其是否具有连接SQL Server的权限。SQL Server 2000提供了两种确认用户的认证模式：
 - （一）Windows NT认证模式。
 - （二）混合认证模式。

6.1 Windows NT认证模式

- SQL Server数据库系统通常运行在Windows NT服务器平台上，而NT作为网络操作系统，本身就具备管理登录、验证用户合法性的能力，因此Windows NT认证模式正是利用了这一用户安全性和帐号管理的机制，允许SQL Server也可以使用NT的用户名和口令。在这种模式下，用户只需要通过Windows NT的认证，就可以连接到SQL Server，而SQL Server本身也就不需要管理一套登录数据。

6.2

混合认证模式

- 混合认证模式允许用户使用Windows NT安全性或SQL Server安全性连接到SQL Server，这就意味着用户可以使用他的帐号登录到Windows NT，或者使用他的登录名登录到SQL Server系统。NT的用户既可以使用NT认证，也可以使用SQL Server认证。

6.3 利用企业管理器设置认证模式

其主要过程如下：

1. 打开企业管理器，用右键单击要设置认证模式的服务器，从快捷菜单中选择“属性（properties）”选项，则出现SQL Server属性对话框。
2. 在SQL Server属性对话框中选择安全性选项。
3. 在安全性选项栏中，身份验证中可以选择要设置的认证模式，同时审核级别中还可以选择跟踪记录用户登录时的哪种信息，例如登录成功或登录失败的信息等。
4. 在启动服务帐户中设置当启动并运行SQL Server时默认登录者中哪一位用户。

第7章 SQL Server代理服务

- 7.1 SQL Server代理服务的概念
- 7.2 SQL Server代理服务配置
- 7.3 定义操作员
- 7.4 作业管理
- 7.5 警报管理

7.1 SQL Server 代理服务概念

- SQL Server 代理是一个任务规划器和警报管理器，在实际应用时，可以将那些周期性的工作定义成一个任务，在 SQL Server 代理的帮助下自动执行；在自动执行时，若出现故障，则 SQL Server 代理自动通知操作员，操作员获得通知后及时排除故障。如此在任务、操作员、警报三者之间既相互独立，又相互联系、相互补充，构成了自动完成某些任务的有机整体。

7.2 SQL Server 代理服务的配置

- SQL Server 代理是一个任务规划器和警报管理器，在实际应用时，可以将那些周期性的工作定义成一个任务，在 SQL Server 代理的帮助下自动执行；在自动执行时，若出现故障，则 SQL Server 代理自动通知操作员，操作员获得通知后及时排除故障。如此在任务、操作员、警报三者之间既相互独立，又相互联系、相互补充，构成了自动完成某些任务的有机整体。

7.3 SQL Server 作业管理

- SQL Server 代理是一个任务规划器和警报管理器，在实际应用时，可以将那些周期性的工作定义成一个任务，在 SQL Server 代理的帮助下自动执行；在自动执行时，若出现故障，则 SQL Server 代理自动通知操作员，操作员获得通知后及时排除故障。如此在任务、操作员、警报三者之间既相互独立，又相互联系、相互补充，构成了自动完成某些任务的有机整体。

7.4 SQL Server 警报管理

- SQL Server 代理是一个任务规划器和警报管理器，在实际应用时，可以将那些周期性的工作定义成一个任务，在 SQL Server 代理的帮助下自动执行；在自动执行时，若出现故障，则 SQL Server 代理自动通知操作员，操作员获得通知后及时排除故障。如此在任务、操作员、警报三者之间既相互独立，又相互联系、相互补充，构成了自动完成某些任务的有机整体。

第8章 SQL Server数据复制

- SQL Server提供了强大的内置复制能力，在复制这一支持分布式数据处理能力的重要技术帮助下，我们可以完成服务器之间的信息复制，这样就具有多个域的信息源同步处理的功能。
- 复制最直接的定义就是将数据复制到另一地址，然后将数据同步，最终所有备份数据都有相同的值。

8 SQL Server 2000支持三种复制类型

- 快照复制（Snapshot replication）
- 事务复制（Transaction replication）
- 合并复制（Merge replication）

第9章 SQL Server 数据转换

- 9.1 数据的导入和导出
- 9.2 使用DTS设计器

Thanks!

