

Microsoft SQL Server 2000 实用教程



SQ-43

人民邮电出版社
www.pptph.com.cn

闪四清 编著

Microsoft SQL Server 2000

实用教程

闪四清 编著

人民邮电出版社

Microsoft SQL Server 2000 实用教程

◆ 编 著 闪四清
责任编辑 刘 涛

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ pptph.com.cn
网址 <http://www.pptph.com.cn>
北京汉魂图文设计有限公司制作
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销

◆ 开本:787×1092 1/16
印张:35.25

字数:877 千字

2000 年 12 月第 1 版

印数:1-6 000 册

2000 年 12 月北京第 1 次印刷

ISBN 7-115-08897-7/TP·1906

定价:51.00 元

Microsoft SQL Server 2000 是新一代的数据库和数据分析系统,可以帮助用户建立他们自己的 Internet 商务体系。微软公司宣称,SQL Server 2000 已经在系统性能和可扩展性方面确立了世界领先的地位,是一套完全的数据库和数据分析解决方案,该产品可以使用户快速地创建新一代的可扩展电子商务和数据仓库解决方案。SQL Server 2000 具有完全的 Web 功能,支持扩展标记语言(XML),并且拥有一个新的、集成的数据挖掘引擎,可以让用户从 B2B 和 B2C 型的电子商务所提供的机会和高效性中获益。SQL Server 2000 采用了业界先进的数据结构,能够为要求苛刻的 Web 应用和企业应用提供高度的可扩展性和高可靠性。在该产品中,微软公司提供了许多技术和工具,作为数据库系统的标准集成组件,使得 SQL Server 2000 可以进行最快速的应用程序开发。Windows 2000 和 SQL Server 2000 群集提供了高度可用性和自动化管理,故障切换和复制这两种技术内置到了 SQL Server 2000 系统中,使得 SQL Server 2000 具有了高度可用性。SQL Server 2000 还提供了强大的管理模型,它以图形化用户界面和向导工具为基础,带有一个用于重复任务执行的作业调度程序以及用于编写特定应用操作脚本的 SQL - DMO。SQL Server 2000 的结构可以适应模块化增长、自动化配置、维护大规模服务器程序开发的需要。

本书全面讲解了 SQL Server 2000 产品的功能。作为一本实用教程,本书详细研究了与 SQL Server 2000 产品有关的各种专题,所有该产品的专题内容在本教程中都给予了详细的理论讲解和实际示例说明。通过本教程的学习,读者可以快速全面地掌握数据库的基本理论和 SQL Server 2000 的各种功能。在内容的安排上,本书采取了理论和实践相结合的方式,一方面详细阐述了数据库的基本理论,另一方面注重培养读者解决实际问题的能力。书中列举了大量的实例,使读者在学习完本书之后可以更加深刻地理解 SQL Server 2000 产品的功能和特点。

本书由康博创作室策划,闪四清博士主编著。在本书的编写过程中,得到了微软(中国)有限公司 SQL Server 产品部门有关技术人员的大力支持,在此深表谢意。

由于时间仓促,加上水平有限,书中错误和疏漏之处在所难免,恳请广大读者批评指正。

作者

2000 年 8 月

SQL Server 2000 系统概述

信息技术是知识经济时代的重要支柱,而网络技术和数据库技术又是信息技术最主要的核心。以 Internet/Intranet/Extranet 为标志的计算机网络技术席卷整个业界,信息传播速度越来越快。以 SQL Server 为代表的成熟的数据库技术的应用越来越广泛,大容量的数据存储、快速的数据查询、方便的向导和工具、友好亲切的用户界面,促进了电子商务、信息处理和管理的发展。

SQL Server 2000 是微软最新推出的 SQL Server 版本,它具有可靠性、可伸缩性、可用性等特点,提供了完整的数据库解决方案。本书将全面介绍 SQL Server 2000 的特点和功能。

本章是 SQL Server 2000 系统概述,将主要介绍下面一些内容:

- 关系型数据库系统的概念和演变进程;
- SQL Server 2000 的体系结构;
- SQL Server 2000 的发展历史;
- SQL Server 2000 的新增功能。

1.1 关系型数据库系统的概念和演变过程

1.1.1 关系型数据库系统的概念

数据库技术是从 20 世纪 60 年代后期产生并发展起来的,这是信息技术中发展最快的领域之一。数据库技术的主要目标是解决数据处理的非数值计算问题,例如仓库管理、档案管理、图书资料管理等数据处理问题。这类数据的特点是数据量比较大、数据检索频繁,数据处理的主要内容是数据的存储、查询、修改、分类排序等,以及支持决策功能。数据库技术就是针对这类数据的处理而产生发展起来的,并且仍在不断地、迅速地发展和完善。

数据库是为了满足某一部门中多个用户的多种应用的需要,按照一定的数据模型在计算机系统中组织存储和使用的互相联系的数据集合。数据库系统是管理大量的、持久的、可靠的、一致的、共享的

数据的工具。从这个简单的定义中可以看出数据库系统是一种管理数据的工具,其管理对象具有以下特征:

(1) 巨量性。这表明数据量很大,不能放在通常的内存中,需要有大容量的外部存储设备支持数据,例如某家银行的业务数据量可能高达太字节(TB)。

(2) 持久性。这表明这些数据必须长久地保持,并不是简单地为某一特定用途的应用准备的,不是当应用一完成数据就随之消失,而是这些数据需要经常使用,如银行的业务数据、单位的人事数据等,这些数据需要保存几年、几十年甚至更长的时间。

(3) 可靠性。这是指万一系统发生软硬件故障时,可以有办法迅速可靠地把系统恢复到发生故障前的状态,例如必须可以对数据进行经常性的备份,以便当系统遭到破坏、数据丢失后可以把数据恢复到系统破坏前的状态。

(4) 一致性。这是表示存储在数据库中的数据完整性,例如存储在雇员表中的雇员数据不能出现完全相同的两行,公司中雇员数据和财务数据应该满足一些约束条件等。

(5) 共享性。这是指许多用户能够按照一定有序的方式存取数据库中的数据,避免同步存取时可能会造成的并发性错误,例如在铁路部门或航空部门的售票系统中,经常会出现多个售票点对同一数据源中数据的操作。

数据库中的数据是高度结构化的,即数据库不仅要考虑记录内的数据项之间的联系,还要考虑记录之间的联系。数据模型主要是指描述这种联系的数据结构形式。数据模型是数据库系统中的一个关键概念,是实体与实体之间的一个轮廓视图,是数据库系统用以提供信息表示和操作手段的形式框架。在数据库的发展史上,最有影响的数据库模型有这样几种:层次模型(Hierarchical Model)、网状模型(Network Model)和关系模型(Relational Model)。这三种模型都是以数据结构命名,层次模型和网状模型又称为格式化模型。这类模型的数据结构用图来描述,实体用记录型来表示,一个记录型对应于图中的一个结点,记录型之间的联系对应于结点之间的连接弧。对应于树状图的数据模型为层次模型,对应于网状图的数据模型为网状模型。关系模型为非格式化模型。在关系模型中,使用单一的二维表的结构表示实体与实体之间的关系。

数据库系统是一个由外模式、概念模式和内模式组成的多级系统结构。为了定义各级模式,需要一组软件提供相应的定义工具。数据库为了保证存储在其中的数据的安全和一致,必须有一组软件来完成相应的管理任务,这种软件就是数据库管理系统,简称 DBMS。DBMS 的功能随着各供应商的不同而不同,大型数据库管理系统的功能多且强,小型数据库管理系统相对来说少且弱。但是,一般来说,DBMS 应该包括以下几个方面的功能:

(1) 数据库描述功能。数据库描述功能用于定义数据库的全局逻辑结构、局部逻辑结构和其他各种数据库对象等。

(2) 数据库管理功能。数据库管理功能包括系统配置和管理、数据存取和更新管理、数据完整性管理和数据安全性管理等。

(3) 数据库的查询及操纵功能。该功能包括对数据库的各种检索和修改功能。

(4) 数据库维护功能。该功能包括数据引入引出管理、数据库结构维护、系统恢复和性能监测等。

从内容上来说,数据库管理系统由三部分组成:数据描述语言、数据操纵语言和数据库管理例行程序。现代的数据库系统除了 DBMS 之外,为了提高数据库系统的开发效率,还集

成和提供了各种各样的支持应用开发工具。

关系型数据库的规范化理论是数据库设计的理论基础,其目的是研究关系模式中各个属性之间的依赖关系及其对关系模式的影响。规范化理论不仅能够作为数据库设计优劣的判断依据,而且还可以预测数据库系统可能出现的问题。在关系型数据库理论中,一个二维表就称为一个关系。二维表是由行和列组成,一个列对应于一个字段,称为属性;一行对应于一条记录,称为一个元组。二维表的框架对应于数据库结构,称为关系模式。在一个关系中,必须有一个关键字。这个关键字有时也称为主关键字,或者简称为主键。主键可以唯一地标示出一个元组。例如,在人事档案中,可以选择工作证号或职工编号作为主键,这时,由于每一条记录都具有不同的主键的值,所以可以根据主键的值唯一地确定一条记录。如果在公司中没有姓名相同的员工,那么也可以使用姓名作为主键。但是,实际上存在着重名的可能性和现实性,所以使用姓名就无法唯一地标示出每一条记录,不能使用姓名作为关键字。一般地,关系具有下述性质:每一列上的数据属于同一种属性;没有重复的行(所有的行都不相同);行的顺序不重要;每一个列上有不同的名称,即在一个关系中属性名唯一;列的顺序不重要。

为了避免操作异常,关系模式要满足一定的条件。关系模式要满足的条件称为规范化形式(Normal Form),简称为范式 NF。不同程度的条件称为不同层次的范式,这些范式包括第一规范化形式(1NF)、第二规范化形式(2NF)、第三规范化形式(3NF)、第四规范化形式(4NF)、第五规范化形式(5NF)等。前三个范式是最常用的范式形式。第一规范化形式要求元组中的每一个数据项都不能再分割,都是原子项,记作 1NF。简单地说,1NF 就是要求在二维表中的每一个元素都是单一的、不可再分的数据。如果一个关系模式符合第一规范化形式(1NF),并且每一个非主键属性都完全依赖主键,那么认为该关系模式符合第二规范化形式,简称 2NF。在 2NF 中,非主键属性(从全部属性的集合中除去主键包含的属性)必须与整个主键有函数依赖关系。如果在一个符合第二规范化形式的关系模式中,所有非主键属性之间不存在函数依赖关系,那么称该关系模式符合第三规范化形式,简记为 3NF。3NF 的实质是从符合 2NF 的关系模式中除去非主键属性之间的传递依赖。

1.1.2 数据库理论的演变过程

20 世纪 60 年代以前,计算机主要用于科学计算。进入 60 年代以后,在社会大系统中出现了巨大的信息流和相伴随的宏大的数据流,为了适应和满足社会发展的这些需要,必须采用新的技术和手段,对这些数据进行收集、存储、加工、检索、分类、统计和传输等。这就是数据库技术产生的历史背景。60 年代,美国系统发展公司在为美国海军基地研制数据库时,首先使用了“Data Base”一词。1968 年,IBM 公司在数据库管理系统方面率先研制成功了 IDS (Integrated Data Store) 系统,它可以为多个 COBOL 程序提供共享数据库。1969 年 10 月 CODASYL 数据库研制者提出了网络模型数据库系统规范报告 DBTG,使数据库系统开始走向规范化和标准化。因此,许多专家认为数据库技术诞生于 60 年代末。

数据库技术的产生来源于社会的实际需要,而数据库技术的实现必须有理论作为指导,系统的开发和应用又不断地促进数据库理论的发展和完善。1970 年,IBM 公司 San Jose 研究所的 E. F. Code 发表了题为《大型共享数据库的数据关系模型》的论文,开创了数据库的关系方法和关系规范化的理论研究。关系方法由于其理论上的完美和结构上的简单,它的

出现对数据库技术的发展起着至关重要的作用,成功地奠定了关系数据库理论的基石。为此,1981年,E. F. Code 获得了计算机科学的最高奖 ACM 图林奖。1971年,美国数据系统语言协会在正式发表 DBTG 报告中,提出了三级抽象模式,即对应用程序所需的那部分数据结构描述的外模式、对整个实体系统数据结构描述的概念模式、对数据存储结构描述的内模式,解决了数据独立性的问题。1974年,IBM 公司 San Jose 研究所研制成功了关系数据库管理系统 System R,并且投放软件市场。1976年,美籍华人陈平山提出了数据库逻辑设计的实体联系方法。1978年,新奥尔良发表了 DBDWD 报告,他把数据库系统的设计过程分为4个阶段:需求分析、信息分析与定义、逻辑设计和物理设计。1980年,J. D. Ullman 所著的《数据库系统原理》一书正式出版。1984年美国空军推出了 IDEF1X 方法,该方法是一种语义数据模型,可以描述复杂系统的信息模型。David Marer 1984年所著的《关系数据库理论》一书,标志着数据库在理论上的成熟。

1.2 SQL Server 2000 的体系结构

微软公司宣称 SQL Server 2000 是一种新一代的数据库管理系统。作为 Windows 数据库家族中出类拔萃的成员,SQL Server 2000 这种关系型数据库管理系统能够满足各种类型的企业客户和独立软件供应商构建商业应用程序的需要。根据客户的反映和需求,SQL Server 2000 在可伸缩性、可用性和可管理性等方面有了显著的改进和提高。

运行在 Windows 2000 服务器上的微软公司的 SQL Server 和 BackOffice 产品已经开始支持巨型数据库和应用。本节概要说明了 SQL Server 当前的强大功能。这些内容表明,SQL Server 的可伸缩性既可以适应很小的兆字节个人数据库,也可以适应由数以百万用户使用的巨大的太字节数据库。SQL Server 的这种可伸缩性支持在对称多处理器(SMP)上增大规模和允许用户增加处理器、内存、磁盘以及组网能力来创建一个大型的单节点。SQL Server 还支持对多节点群集的规模扩充(scale out),即把一个巨型数据库分区到一个服务器群集中,其中每一个服务器存储了这个数据库的一部分,并且每个服务器都可以执行一部分任务,但是对于用户来说,整个数据库的存储和访问就像是对待一个单独的数据库那样。通过使用这种扩充规模的伸缩样式,SQL Server 2000 达到了任何数据库系统在任何平台上所能达到的顶级 TPC-C 性能。Windows 2000 和 SQL Server 2000 群集提供了高度可用性和自动化管理。故障切换和复制这两种技术内置到了 SQL Server 系统中,使得 SQL Server 具有了高度可用性。SQL Server 还提供了强大的管理模型,它以图形化用户界面和向导工具为基础,带有一个用于执行重复任务的作业调度程序以及用于编写特定应用程序脚本的 SQL-DMO。SQL Server 的结构可以适应模块化增长、自动化配置、维护大规模服务器程序开发等需要。

Windows DNA 2000 是微软推出的下一代 Web 应用程序平台,SQL Server 2000 在这种应用程序平台中位于数据层,起着存储大量数据的作用。这种下一代 Web 应用程序平台的逻辑结构如图 1-1 所示。

SQL Server 2000 是一种客户机/服务器的关系性数据库管理系统,它使用 Transact-SQL 语句在服务器和客户机之间传送请求,这种结构如图 1-2 所示。

SQL Server 2000 使用客户机/服务器体系结构把所有的工作负荷分解成在服务器机器

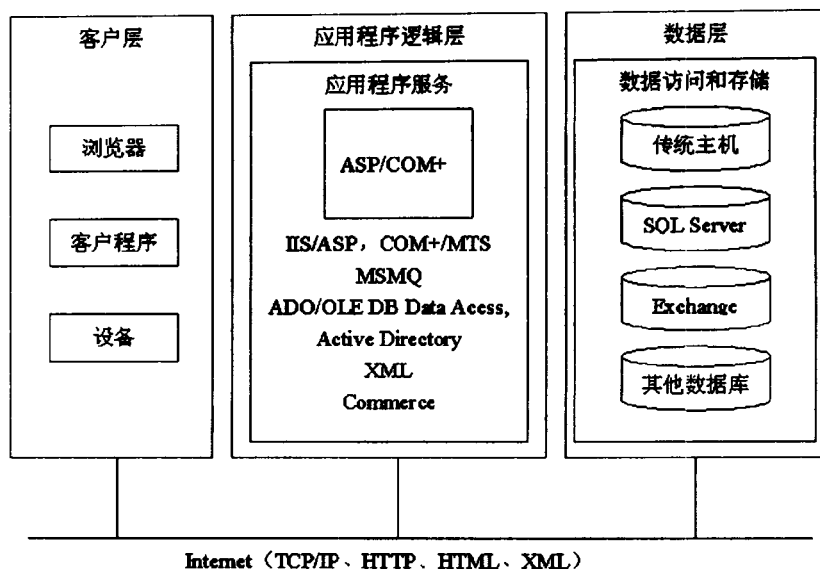


图 1-1 下一代 Web 应用程序平台 Windows DNA 2000

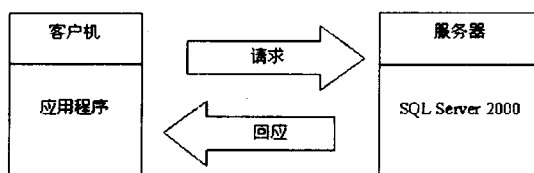


图 1-2 SQL Server 2000 客户机/服务器结构示意图

上的任务和在客户机机器上的任务。客户机应用程序负责商业逻辑和向用户提供数据，一般运行在一个或者多个客户机机器上，但是也可以运行在服务器上。服务器负责管理数据库和分配可用的服务器资源，例如内存、网络带宽和磁盘操作。客户机应用程序界面允许应用程序在单独的客户机上运行，并且通过网络与服务器通信。

关系数据库管理系统简称 RDBMS，负责管理数据库的结构，其内容主要包括维护数据库中数据之间的关系、确保数据存储的正确性以及系统在失败时恢复全部数据。

Transact-SQL 是 SQL Server 2000 使用的一种数据库查询和编程语言。SQL 是结构化查询语句的缩写形式，是由美国国家标准协会(ANSI)和国际标准化组织(ISO)定义的标准。最新的 SQL 版本称为 ANSI SQL-99，于 1999 年出版发行。Transact-SQL 支持最新的 SQL 标准，并且在标准基础上增加了许多扩展。使用 Transact-SQL 语句可以访问数据和查询、修改、管理关系数据库系统。

SQL Server 2000 可以在许多操作系统上运行。SQL Server 2000 的服务器环境可以是 Windows 2000、Windows NT 或者 Windows 9x，其客户机环境可以是 Windows 2000、Windows NT、Windows 9x、Windows 3.x、MS-DOS、第三方平台和 Internet 浏览器。

SQL Server 2000 是与 Windows NT 完全集成的，并且利用了 NT 的许多功能。SQL Server 2000 可与 Windows NT 的安全性系统集成、支持多协议、作为 NT 的服务运行、利用 NT 的性能监测器和使用 Index Server 2000。

SQL Server 2000 可以很好地与 Microsoft BackOffice 产品集成。可以与 SQL Server 一起使用的 Microsoft BackOffice 产品包括 Microsoft Internet Information Server(IIS)、Microsoft Exchange Server、Microsoft SNA Server 和 Microsoft System Management Server。

SQL Server 2000 包括了三个服务,他们是 MSSQLServer、SQL Server Agent 和 Microsoft Distributed Transaction Coordinator(MS DTC)服务。MSSQLServer 服务就是 SQL Server 的 RDBMS。SQL Server Agent 服务,在 SQL Server 6.5 中称为 SQL Executive 服务,用于管理任务作业、警报、复制和操作者。MS DTC 服务是分布式事务协调器,用于保证在多个服务器之间的事务同时完成或者取消,协调多服务器之间事务的一致性,确保多服务器之间数据的完整性。

1.3 SQL Server 的发展历史

下面使用时间表的形式给出 SQL Server 的开发历史:

- 1987 年——Sybase 公司发行了用于 UNIX 平台的 SQL Server;
- 1988 年——Microsoft、Sybase 和 Ashton-Tate 三家公司把 SQL Server 移植到 OS/2 平台上;
- 1989 年——Microsoft、Sybase 和 Ashton-Tate 发行了用于 OS/2 的 SQL Server 1;
- 1990 年——发行了支持 Windows 3 客户机的 SQL Server 1.1;
 - Ashton-Tate 公司中途退出了 SQL Server 的开发;
- 1991 年——微软公司和 IBM 公司终止了用于 OS/2 平台的 SQL Server 的开发;
- 1992 年——发行了用于 16 位 OS/2 1.3 的 SQL Server 4.2;
 - 微软公司和 Sybase 公司把 SQL Server 移植到了 Windows NT 平台;
- 1993 年——发行了 Windows NT 3.1;
 - 微软公司和 Sybase 公司发行了用于 Windows NT 的 SQL Server 4.2;
- 1994 年——微软公司和 Sybase 公司的 SQL Server 共同开发阶段正式结束;
 - 微软公司继续开发用于 NT 平台的 SQL Server 版本;
 - Sybase 公司继续开发 SQL Server 的 UNIX 版本;
- 1995 年——微软公司发行了 SQL Server 6 版本;
- 1996 年——微软公司发行了 SQL Server 6.5 版本;
 - 4 月,发布了 SQL Server 6.5 的标准版本;
- 1997 年 11 月——发布了 SQL Server 6.5 的企业版本;
- 1998 年——微软公司发行了 SQL Server 7 版本;
- 2000 年——微软公司发行了 SQL Server 2000 版本。

由于 SQL Server 7 是一个变化巨大的版本,下面对其开发历史进行一下总结:

1997 年 6 月,发布了 SQL Server 7 的 Beta1 版本,大约有 200 个用户,包括有限数量的独立软件供应商、电脑书籍作者、教学软件设计者、OEM 商和一些公司的财务部门。推出这个版本的目的在于测试一些低层功能和编程界面。

1997 年 11 月,发布了 SQL Server 7 的 Beta2 版本,大约有 3000 个用户。这些用户包括广泛增加的公司财务部门和 ISV 团体,还有一些新加入的国际财务部门。

1998 年 6 月,发布了 SQL Server 7 的 Beta3 版本,这是 SQL Server 7 的第一个面向广大用

户的版本。

1998年12月,SQL Server 7产品正式发布。

SQL Server 7有3个里程碑的版本:1996年发布的SQL Server 6.5、1998年发布的SQL Server 7.0、2000年发布的SQL Server 2000。这3种版本都得到了广泛的应用。

1.4 SQL Server 2000 的新增功能

SQL Server 2000扩展了SQL Server 7的功能、可靠性和易用性,增加了许多功能。经过这次版本的变化,SQL Server 2000成了在大规模在线事务处理(OLTP)、数据仓库和电子商务等应用程序方面一个非常出色的数据库平台。

在SQL Server 7中,有一个服务是OLAP Services,而现在SQL Server 2000中,该服务的名称变成了Analysis Services。这样改名以后,使得该服务的功能和名称真正是名副其实。不仅仅是改名,在Analysis Services服务中,还增加了一个新的数据挖掘组件。

在SQL Server 7中,有一个Repository组件,而在SQL Server 2000中,该组件被称为Meta Data Services。

除此之外,还有许多新增强内容,例如:

- 关系型数据库方面的增强;
- 支持XML语言;
- 图形管理工具方面的增强;
- 复制方面的增强;
- Data Transformation Services 方面的增强。

下面详细介绍这些新增强内容。当然,了解这些内容的前提是了解SQL Server 7版本的内容。

1.4.1 关系型数据库方面的增强

在关系型数据库方面,SQL Server 2000有了许多提高和增强。

1. 支持XML

XML是可扩展标记语言的简称,可以用来描述一组数据的内容并且确定如何输出这些数据或者显示在Web页面上。在SQL Server 2000中,关系型数据库引擎可以使用XML文档返回数据。除此之外,还可以使用XML插入、删除、修改数据库中的数据。在SQL Server 2000中,对XML的支持包括下列内容:

- 可以通过URL访问SQL Server 2000中的数据;
- 支持XML-Data模式,并且可以指定这些模式的Xpath查询;
- 可以检索XML数据,例如使用SELECT语句和FOR XML子句检索XML数据;
- 可以写XML数据,例如使用OpenXML行集提供器写XML数据;
- 对SQL Server 2000 OLE DB provider (SQLOLEDB 方面的增强)允许把XML文档作为命令设置和作为数据流返回结果集。

2. 联邦式数据库服务器

SQL Server 2000 支持分布式分区视图,允许把表水平分区在多个服务器上。这样,就允许扩充一个数据库服务器到一组数据库服务器,这组数据库服务器作为一个数据库服务器的群集,这个群集提供的性能就像一个数据库服务器提供的性能一样。这种联邦式数据库服务器可分解服务器负荷到多个服务器上,以满足最大的 Web 站点数据存储和企业数据处理系统的需要。

3. 用户定义的函数

SQL Server 2000 支持通过创建自己的 Transact-SQL 函数,来扩展 Transact-SQL 语言的编程性。用户定义的函数可以有零个或多个输入参数,既可以返回一个数字或字符等标量值,也可以返回一个表。可以使用 CREATE FUNCTION 语句定义用户自己的函数。

4. 索引视图

索引视图可以显著地提高应用程序的性能,例如频繁执行一些连接或合计函数的查询。索引视图就是允许在视图上创建索引,这些视图和结果集存储在视图所在的数据库中。为了利用索引视图的优点,已有的应用程序没有必要进行修改就可以自动地利用这些优点。索引视图的方法与索引表的方法是一样的。

5. 新的数据类型

SQL Server 2000 引入了一些新的数据类型。这些新增加的数据类型包括 bigint、sql_variant 和 table。Bigint 是一个 8 字节的整数类型,sql_variant 是一种允许存储不同数据类型数据值的数据存储,table 数据类型允许应用程序临时存储以后使用的结果。这样,SQL Server 2000 提供的数据类型有 25 个,如表 1-1 所示。

表 1-1

SQL Server 2000 支持的数据类型

bigint	binary	bit	char	cursor
datetime	decimal	float	image	int
money	nchar	ntext	nvarchar	real
smalldatetime	smallint	smallmoney	table	text
timestamp	tinyint	varbinary	varchar	uniqueidentifier

6. INSTEAD OF 和 AFTER 触发器

SQL Server 2000 新增加了两种类型的触发器,即 INSTEAD OF 触发器和 AFTER 触发器。在 SQL Server 2000 中,可以使用 INSTEAD OF 来替代以前那些触发器的操作,例如可以替代 INSERT、UPDATE 和 DELETE 触发器的操作。另外,在 SQL Server 2000 中,触发器可以建立在视图上,这是以前 SQL Server 2000 版本不能实现的功能。通过在视图上建立触发器,可以大大增强通过视图修改表中数据的功能。当某些触发器的操作执行之后,可以触发执行 AFTER 触发器的操作。

7. 级连参考完整性的约束

在 SQL Server 2000 中,当企图修改那些被某些外键所参考的关键列时,就可以控制 SQL Server 2000 的操作。这种新功能可以在 CREATE TABLE 和 ALTER TABLE 语句中使用。在 CREATE TABLE 和 ALTER TABLE 语句的 REFERENCES 子句中,新增加了两个子句,即 ON DELETE 子句和 ON UPDATE 子句。详细内容在本书后面有关章节中介绍。

8. 字符排列顺序方面的增强

以前的 SQL Server 版本,只能在服务器端指定系统使用的代码页和排列顺序,这种指定可以应用到服务器的所有数据库中。但是,在 SQL Server 2000 中,情况有了很大的改变。在 SQL Server 2000 中,除了支持以前版本中大多数的字符排列顺序之外,还包括了一种新的基于 Windows 排列顺序的排列顺序集。现在,在 SQL Server 2000 中,可以在数据库级上或列集上指定字符的排列顺序。这种功能大大增强了 SQL Server 2000 产品的兼容性和可移植性。

9. 全文搜索方面的增强

在 SQL Server 2000 中,全文搜索方面又增加了两种新功能,即修改跟踪和图像过滤。修改跟踪可以维护对全文索引数据所有修改的数据日志。在 SQL Server 2000 中,可以使用许多方法,例如手工修改日志、根据调度的时间修改日志,或者修改自动发生时修改全文索引的数据。这是以前版本中全文索引不能达到的功能。图像过滤功能就是允许索引和查询存储在图像列(image 列)中的文档。如果这种文档存储在文件系统中,那么用户必须在某个列中提供包含了该文档扩展名的文档类型。通过使用这种信息,为了索引的目的,全文索引可以加载合适的文档过滤器来提取文本性信息。

10. 多个 SQL Server 2000 实例

在 SQL Server 2000 中,又增加了一种以前版本没有过的信息。SQL Server 2000 支持在同一台计算机上运行多个关系型数据库引擎实例。在 SQL Server 6.5 或 7.0 版本中,每一台计算机只能运行一个关系型数据库实例,而在 SQL Server 2000 版本中,每一台计算机都可以运行一个或多个关系型数据库实例。在 SQL Server 2000 版本中,每一个实例系统和用户数据库集都是独立的。当应用程序连接到运行在一个计算机上的每一个实例时,连接的方式就像连接到运行在不同计算机上的 SQL Server 2000 实例一样。

11. 索引方面的增强

在 SQL Server 2000 中,可以在经过计算得到的列上创建索引。在创建索引时,既可以指定索引按照升序排列,也可以指定索引按照降序排列,而数据库引擎既可以使用并行扫描,也可以使用并行排序。

12. 故障切换群集功能的增强

现在,对于故障切换群集的管理得到了大大的提高,使得安装、配置、维护 SQL Server 2000 的故障切换变得非常容易。除此之外,还增加了从任何一个 SQL Server 群集到另一个群集故障切换和失败反馈的功能,通过 SQL Server 2000 Setup 从一个群集中增加或删除一个节点的功能,在一个群集中的任何服务器上重新安装或重新创建一个群集实例而不影响该群集中其他节点实例的功能。

13. 网络库方面的增强

在 SQL Server 2000 中,网络库(Net-Libraries)的配置得到了增强。当使用 SQL Server 2000 的客户端程序连接到 SQL Server 2000 时,不必在客户机上对网络库的配置进行管理,系统就可以自动地进行配置。新的网络库还支持对同一台计算机上多个 SQL Server 2000 实例的连接,并且支持在所有的网络库上进行安全加密。

14. 分布式查询功能的增强

在 SQL Server 2000 中,新增加了一个函数 OPENROWSET。该函数可以用来指定分布式查询语句中的连接信息。在 SQL Server 2000 中,OLE DB 提供器可以用来报告由该提供器支持的 SQL 语法和在数据源中关键字的分布信息。然后,分布式查询优化器可以使用这种信息来减少从 OLE DB 数据源中发送的数据量。在 SQL Server 2000 中,增加了比以前版本多得多的对 OLE DB 数据源的操作。分布式查询还支持在 SQL Server 2000 中引入的其他功能,例如多个实例、在结果集中混合不同排列顺序的列以及新的 bigint 和 sql _ variant 数据类型。

15. Kerberos 和安全性代表

SQL Server 2000 支持在客户机和服务器之间共同的认证,以及在两个计算机之间传送安全证书的功能,以便在远程服务器上的工作可以处理使用伪装客户机的证书。通过与 Microsoft Windows 2000 的集成,SQL Server 2000 使用 Kerberos 和代表来支持两种认证方式,即集成认证方式和 SQL Server 认证方式。

16. 备份和恢复方面的增强

在 SQL Server 2000 中,又增加了一种新的用于指定备份和恢复的更加容易理解的模型。这种新的模型可以使用户更加清晰地平衡增加或减少丢失的工作量和不同计划的性能与日志空间的需求。在 SQL Server 2000 中,通过使用事务日志中的命名日志标志可以指定工作点,以及执行数据库的部分恢复。

17. 用于实用程序操作方面的可伸缩性增强

在实用程序操作方面,SQL Server 2000 增强了许多功能,例如更加快速的增量备份、并行的数据库一致性检查(DBCC)和并行扫描。现在,增量备份可以备份上一次完全备份之后的数据改变,并且可以在更短的时间内完成。DBCC 可以运行并且运行在不使用共享扫描锁的状况,这样就允许在运行 DBCC 时执行对表修改的活动。另外,DBCC 现在可以利用多个处理器,因此其性能将随着 CPU 数量的增加而呈现几乎非线性增强(当然,I/O 不能是影响系统性能的瓶颈因素)。

18. 行中的文本

在 SQL Server 2000 中支持新的 text in row 表选项。该选项可以指定比较小的 text、ntext 和 image 数据类型数据直接放在表的行中,而不必放在单独的页面中。增加这种功能之后,可以减少用来存储比较小的 text、ntext 和 image 数据类型数据的空间量,并且还可以降低需要处理这些数据的磁盘 I/O 量。

1.4.2 XML 和关系型数据的集成

SQL Server 2000 关系型数据库引擎支持本地的扩展标记语言(Extended Markup Language,

XML),可以使用流行的统一资源定位器(Universal Resource Locator ,URL)来访问 SQL Server 2000。可以在 Microsoft Internet Information Server(IIS)上定义一个虚拟根,参考该虚拟根的 URL 就可以访问 SQL Server 2000 的 XML 功能模块。IIS 应用程序可以使用 URL、ADO 或 OLE DB 在 SQL Server 2000 中实现 XML 功能。下面详细介绍这几种功能。

1. 定义 XML-Data 模式

用户可以定义 XML-Data 模式来提供针对 SQL Server 2000 表和视图中数据的基于 XML 的视图。然后,可以在 Xpath 查询中参考这种模式,以便给出这些表或视图的查询访问。

2. 使用 SELECT 语句返回 XML 文档

在 SQL Server 2000 中,SELECT 语句的结果可以返回成 XML 文档。SQL Server 2000 的 Transact-SQL SELECT 语句支持 FOR XML 子句,该子句可以指定 SELECT 语句的结果返回成 XML 文档的哪一种形式,而不是以前的关系型结果集。复杂的查询语句,或希望保证安全的查询,可以作为模板存储在 IIS 的虚拟目录中,通过参考该模板的名称来执行这种 XML 文档。

3. 使用系统存储过程存储 XML 文档

现在,用户可以使用系统存储过程在 SQL Server 2000 的数据库中存储 XML 文档。然后,可以使用新增加的 OPENXML 行集函数从 XML 文档中打开数据,使得这些数据就像是关系型行集一样。OPENXML 函数可以用在行集函数可以使用的任何地方,例如可以用在指定表名的 FROM 子句中。这种函数允许用户使用 XML 文档中的数据在数据库的表中插入或修改数据。

1.4.3 图形化工具方面的增强

在 SQL Server 2000 中,图形化工具也增加了一些新内容。主要增强的功能包括下面几个方面。

1. 日志传输

在 SQL Server 2000 中,新增加了日志传输功能。日志传输功能就是允许事务日志从一个正常备份的源数据库中加载到另外一个服务器上的目标数据库中。这种功能对于维护热备份服务器或者从源服务器中通过执行查询处理到一个只读的目标服务器是非常重要的。

2. SQL Profiler 工具的增强

在 SQL Server 2000 中,SQL Profiler 工具支持基于尺寸和基于时间的跟踪,并且包括了许多新的事件,例如 Data File Auto Grow、Data File Auto Shrink、Log File Auto Grow、Log File Auto Shrink、Show Plan All、Show Plan Statistics 和 Show Plan Text。详细内容见本书后面的内容。

3. SQL Query Analyzer 工具的增强

对于非常常用的工具 SQL Query Analyzer,SQL Server 2000 又增加了一些新功能。现在,SQL Query Analyzer 包括 Object Browser 功能,该工具允许用户浏览和获取有关数据库对象的参数和依赖性信息,这些数据库对象包括用户和系统表、视图、存储过程、扩展存储过程和函数。其他增强内容包括服务器跟踪和客户机统计,以便显示服务器端和客户机端的查询语句对系统性能的影响。

1.4.4 复制方面的增强

复制是关系型数据库管理系统中非常重要的一种强大功能。在 SQL Server 2000 以前的版本中,已经在复制方面具有了相当强大的功能。为了继续强化这种功能,SQL Server 2000 又在复制方面进行了增强。新增加的功能如下所述。

1. 实现复制方面的增强

通过增加一些内容,SQL Server 2000 增强了快照复制、事务复制和合并复制等复制类型的功能。这些新增加的功能包括:可以保存快照文件到某个网络位置即 FTP 站点或者到一个可移动介质上,以便不用使用网络就可以把出版物传送到订阅服务器中;可附加的订阅数据库,允许把复制的数据库中的数据和个或多个订阅物从一个订阅服务器传送到另外一个订阅服务器;出版数据库上的模式改变,允许在出版的表中增加或者删除列,并且把这些变化传送到订阅服务器中;根据需要执行脚本,允许把将要执行的通用的 SQL 脚本传送到所有的订阅服务器中;预快照脚本和后快照脚本允许把初始化的快照应用到订阅服务器之前或之后运行的脚本;更多的快照复制选项,可以传输索引、扩展属性和约束等。

2. 合并复制方面的增强

合并复制是从出版服务器到订阅服务器双向分布数据的进程,允许出版服务器和订阅出版器在连接或断开时进行修改操作,当这些服务器连接之后,可以在这些站点之间合并这些修改。这些方面的增强功能包括:Merge Agent 具有更多的并行性,用于提高服务器到服务器的性能;确定对订阅服务器相关数据修改的优化功能;动态的快照复制提供了更加有效的动态过滤器;用于合并出版物的垂直过滤器;由用户定义的更加强大的动态过滤器。

3. 事务复制方面的增强

事务复制就是首先使用快照复制把数据应用到订阅服务器中,然后当在出版服务器上修改数据时,这些事务就被捕捉并且传送到订阅服务器中。新增强的功能包括:并行的快照复制进程,以便当生成初始化快照时,那些在出版表中的数据可以进行连续的修改;改善了错误处理功能,可以跳过指定的错误然后执行连续的复制;在订阅服务器中验证复制的数据;出版经过索引的视图;queued updating 选项可以存储在订阅服务器上的数据修改;transforming published data 选项可以在出版数据时进行数据的转换。

1.4.5 数据转换服务方面的增强

SQL Server 2000 不但对数据转换服务(Data Transformation Services, DTS)原有的功能进行了增强,而且还增加了新的功能。

1. 对键和约束的支持

现在,在新的 SQL Server 2000 版本中,支持对键和约束的传输。可以使用引入和引出向导,除了从源表到目标表转换数据之外,还可以把主键、外键和约束传送到目标表中。

2. 新的定制任务

现在 SQL Server 2000 DTS Designer 和 DTS 对象模型引入了新的定制功能,允许创建执行任务的包或设置基于运行环境变量的属性。这些功能包括从网络站点引入数据和发送完整

的数据包、异步运行数据包、创建发送消息的数据包以及连接多个数据包来作为事务的一部分。

3. 保存 DTS 数据包到 Visual Basic 文件

在 SQL Server 2000 中,可以把 DTS 数据包保存成 Visual Basic 文件,这种功能允许把 DTS Import Wizard、DTS Export Wizard 或 DTS Designer 创建的数据包合并到 Visual Basic 程序中。详细内容见本书后面的有关内容。

1.5 SQL Server 2000 系统安全性介绍

在 SQL Server 2000 系统中,对用户的认证有两种方式,一种方式是 SQL Server 本身对用户的验证,另外一种方式是 Windows NT 认证方式。SQL Server 认证方式就是用户在使用 SQL Server 时,必须提供用户的登录帐户和口令,这是 SQL Server 本身的安全性机制。Windows NT 认证方式就是 Windows NT 的用户和用户组可以使用 SQL Server 的资源。但是,并不是说 Windows NT 的用户和用户组可以直接使用 SQL Server 的资源,而是必须对这些 Windows NT 的用户和用户组进行授权之后,才能对 SQL Server 的资源进行各种访问操作。图 1-3 就是选择登录方式的一个窗口。

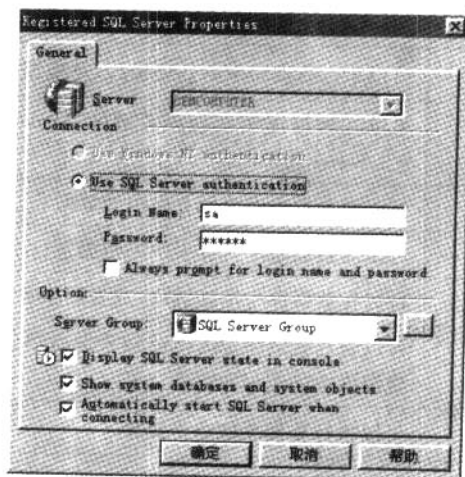


图 1-3 选择登录方式的一个窗口

SQL Server 系统的用户包括三种形式:系统登录帐户(login)和口令、数据库用户和角色。系统登录帐户(login)是用户使用 SQL Server 系统时应该首先提供的用户标识符,口令则是可选的。用户登录到系统之后访问数据库时,还必须在所访问的数据库中有对应的数据库用户名。可以对系统登录帐户和数据库用户授予相应的许可,然后他们才能在系统中执行相应的操作。角色是具有相同许可的用户集合。在 SQL Server 系统中,有二种类型的角色:固定的服务器角色、固定的数据库角色和用户定义的角色。固定的服务器角色就是将具有相同许可的登录帐户集合在一起。系统提供了 7 个固定的服务器角色,如表 1-2 所示。