

学用软件起步与进阶丛书

SQL Server 7.0

起步与进阶

李武 姚分喜 等 编著

国防工业出版社



138SQ

19/8

TP311.138.5Q

L35

学用软件起步与进阶丛书

SQL Server 7.0 起步与进阶

李 武 姚分喜 等编著



A0939960

国防工业出版社

·北 京·

图书在版编目(CIP)数据

SQL Server 7.0 起步与进阶/李武等编著. —北京:国防工业出版社, 2000. 7

(学用软件起步与进阶丛书)

ISBN 7-118-02239-X

I. S… II. 李… III. 关系数据库 - 数据管理系统, SQL Server 7.0 IV. TP311.138

中国版本图书馆 CIP 数据核字(2000)第 21342 号

国防工业出版社出版发行

(北京市海淀区紫竹院南路 23 号)

(邮政编码 100044)

北京怀柔新华印刷厂印刷

新华书店经售

*

开本 787×1092 1/16 印张 23 523 千字

2000 年 7 月第 1 版 2000 年 7 月北京第 1 次印刷

印数:1—3000 册 定价:32.00 元

(本书如有印装错误,我社负责调换)

前 言

Microsoft SQL Server 是 Microsoft 公司推出的面向新世纪的数据库开发系统。它是一个基于 Transact - SQL 语言和客户机/服务器模型的关系数据库系统,是 Microsoft BackOffice 家族中最重要的成员之一。

作为目前最新的版本,SQL Server 7.0 在易用性、可靠性以及其他性能上又有了一些改进,使用户能够更加方便地在 SQL Server 7.0 平台上实现进一步的开发,而且它与 Microsoft BackOffice 无缝集成,使用户可以在 Microsoft 公司的各个产品中进行完全自由的数据访问。

Microsoft SQL Server 可以运行在 Windows 95/98、Windows NT 等各个操作系统平台上。作为企业级的关系数据库开发系统,Microsoft SQL Server 来源于 Sybase、Microsoft、Ashton - Tate 联合开发的基于 OS/2 的数据库系统。随着 Windows NT 操作系统的逐步成熟,由 Microsoft 公司往 Windows 平台移植,在其前身 SQL Server 6.5 的基础上,Microsoft 开发出了跨世纪的数据库产品 SQL Server 7.0,在保持向下兼容的同时,又重新设计了服务器的内部结构和图形化的管理界面,这样,用户无论是在使用还是在管理时,都会感到非常方便、简单、稳定和高效。

SQL Server 7.0 数据库的开发包括两个组件:数据库语言 Transact - SQL 和数据库 API,本书的构成正是基于这两个组件。前面两章简要介绍了 SQL Server 7.0 的新特性及其安装过程;第 3 章概述了 SQL Server 数据库开发基础,介绍了有关数据库应用程序开发过程中应当注意的一些概念和细节;第 4 ~ 10 章介绍了 SQL Server 的数据库语言 Transact - SQL 的语法和函数,以及如何使用该语言进行数据库开发和维护;第 11 ~ 17 章详细介绍了使用 ODBC 3.x 在 Visual C++ 6.0 上进行 SQL Server 客户端应用程序开发的方法。

本书细致详尽地介绍了 SQL Server 7.0 的应用与开发。全书有三个特点:实用、完整和可靠。实用是指通过本书的学习,读者可以很方便地掌握基于 SQL Server 7.0 的数据库开发;完整是指本书内容丰富详尽,全面细致,准确清晰;可靠是指本书中的所有实例都由作者亲自编写并调试通过。

除署名作者之外,何也新、倾城、宋昭辉、李明、刘芹、汤建宁、储会玲、贡玉霞、赵文乐、李同、刘永、于晶晶、朱志、赵文新、王伟、马永忠等都为本书的出版付出了不同程度的劳动,在此表示感谢。

由于时间仓促,且编者水平有限,书中难免有疏漏之处,请广大读者批评指正。

编 者

第 1 章

SQL Server 7.0 核心概念与特色

SQL Server 是从 PC 市场上最早的关系数据库系统 Sybase SQL Server 演变而来的。1988 年,由 Sybase、Microsoft 和 Ashton-Tate 联合开发的 OS/2 系统上的 SQL Server 数据库问世。后来,Microsoft 公司开始将 SQL Server 移植到 Microsoft 的拳头产品 Windows NT 上。1993 年,随着 Windows NT 操作系统的成熟,Microsoft 公司和 Sybase 公司的联合开发/认证协议被取消,Microsoft 公司开始全力开发基于 Windows NT 平台的 SQL Server,先后成功地开发出 SQL Server 6.0、SQL Server 6.5 这样一些受到广泛好评的数据库产品。

由于 SQL Server 是一种涉及到许多全新理论的数据库管理系统,因此,为了方便读者理解后面章节的内容,本章将先介绍 SQL Server 涉及的一些核心概念,然后列举出 SQL Server 7.0 的主要特点。

1.1 数据库

数据库与数据文件有些相似,它们都存储数据,而且与数据文件一样,数据库并不直接把数据显示给用户,而需要运行特定的应用程序来访问数据库中存放的数据,并以某种便于理解的格式显示给用户。

数据库系统比一般的数据文件有更强大的功能,数据库对数据的组织和管理更加高效。一个设计良好的数据库,在用户或应用程序修改数据的同时并没有重复的数据需要额外操作。相关的数据被组合起来形成结构或记录,在结构或记录之间可以定义关系,以表达这些结构或记录之间的相关关系。

当需要处理数据文件时,必须对每种特定格式的数据文件编写不同的应用程序。作为对比,数据库则包含了一个目录,它告诉应用程序数据是如何组织的。一般的应用程序可以根据这个目录动态地从不同的数据库把数据呈现给用户,而不用指定特定的数据格式。

一个典型的数据库一般由两个部分组成:容纳物理数据的数据库和数据库管理系统(DBMS)软件,数据库应用程序通过这个软件来访问数据库中的数据。DBMS 负责实现数据库结构,具体包括:

- (1) 维护数据库中数据之间的关系性。
- (2) 确保数据的正确存储,以及定义数据关系的正确性。
- (3) 在系统发生故障时,在一定程度上恢复数据。

1.2 关系数据库

数据库的数据组织方式有很多种,如层次模型、网状模型和关系模型等。关系数据库模型是由 Dr. E.F. Codd 于 1970 年设计出来用于存储、检索和操作数据的数据库模型,它比层次数据库模型和网络数据库模型更加简单。层次和网络数据库模型设计起来比较困难,而且有时也比较难于书写访问数据的查询,而建立在关系模型上的关系数据库管理系统(RDMS)是其中效率最高的一种。关系数据库管理系统主要是处理储存在表格里的数据,表格之间通过一系列关系联系起来。

在关系数据库模型中是用二维表来组织数据的,表作为一个存储单元来存储某种特定类型的相关数据,如图书管理表、企业账目表、机构组织表等。这样,数据可以通过表格的行和列来定位。一行称为一个记录,一列称为一个域,有时也把每行和每列的交叉点叫作域。每个记录包含某一借阅者、销售项目或机构成员的所有信息,每列包含某个记录的特定部分,如客户编号、客户姓名、单位、购买产品名称、产地、日期等,如表 1-1 所示。

表 1-1 销售管理表

客户编号	客户姓名	单位	购买产品名称	产地	日期
0001	李晓	AA	显示器	苏州	99.10.10
0001	李晓	AA	键盘	广州	99.10.10
0003	于义	CC	主板	台湾	99.10.15

(续)

客户编号	客户姓名	单位	购买产品名称	产地	日期
0011	骆兵	DD	硬盘 1	广州	99.12.17
0011	骆兵	DD	硬盘 2	美国	99.12.17

由于采用了二维表格来表示数据,使得关系数据库模型数据结构非常简单,数据的独立性很高,可以直接处理多对多的联系。

关系数据库还存在一个规范化的问题,虽然我们可以随意地把数据用一个二维表来表示,但是这种表示形式往往不能作为数据库的关系模式。因为不规范的数据库存在着数据冗余度大,易产生操作异常等缺点,而这是数据库设计中必须避免的,下面将举例说明。

观察表 1-1,可以发现客户编号和客户姓名、单位这三项与购买的产品数是联系在一起的,买的各类产品越多,则重复的次数越多,因而造成很大的数据冗余。另外,在这个关系中产品的信息和客户的信息联系在一起,一旦删除某个客户的信息,就可能同时把产品的信息一起删除,容易造成数据丢失。

从上例中可见,为了减少数据的重复,提高数据库的完整性,减少可能造成的信息丢失,必须按照一定的规范来设计数据库。

规范的数据库要求满足:

- (1) 表有唯一的名字,通常标识表中包含的信息的类型。
- (2) 所有的表格由列/域(字段)和行(记录)组成,列/域定义表格行的性质。
- (3) 每行由主关键字唯一确定。
- (4) 主关键字可以由一列或几列组成。
- (5) 主关键字不能为空。

这样规范化的数据库的一个特点就是包含很多只有几列的窄表格,而不同表格之间存在着联系,使得创建的数据库不易出现错误,并且重复的信息也会减少。例如,可以将上例中的销售管理表分成以下三个表,如表 1-2、表 1-3 和表 1-4 所示。

表 1-2 客户信息表

客户姓名	客户编号	单位
李晓	0001	AA
于义	0003	CC
骆兵	0011	DD

表 1-3 产品信息表

产品号	产品名	产地
001	显示器	苏州
002	主板	台湾
003	硬盘 1	广州
004	硬盘 2	美国
005	键盘	广州

表 1-4 销售管理表

产品号	客户编号	产品名	日期
001	0001	显示器	99.10.10
002	0001	键盘	99.10.10
003	0003	主板	99.10.15
004	0011	硬盘 1	99.12.17
005	0011	硬盘 2	99.12.17

关系数据库指一组有某种联系的表格,一个好的关系数据库是组织在表格中的信息,每个表格有唯一的主关键字列,它的值在每一行中存在并且唯一,如表 1-2 中的客户编号、表 1-3 中的产品号、表 1-4 中的编号。某一表格中的主关键字是另一表格的外关键字列,这样不同表格中的数据就联系起来。所谓外关键字,就是与其他表格中的主关键字相匹配的一个域,例如表 1-4 中的客户编号。这样,如果两个表格包含相同的数据,就可以在它们之间建立关系。由于把信息分成独立的表格,关系数据库成为一种存储检索数

据的有效方法。

1.2.1 SQL Server 关系数据库

SQL Server 的核心组件就是关系数据库。在 SQL Server 中,最多可以创建 32,767 个数据库。这个值在创建数据库系统中基本上是不可能达到的,也就是说,我们可以不必考虑创建的数据库数目的限制,这样只需注意磁盘空间的实际大小即可。如果万一需要创建比这个值还多的数据库,则可以借助于网络环境来平衡数据库负载。

在每个数据库中,可以创建多达 2,000,000 个表,但一般情况下只需要几百个表即可。

在每个表中,可以定义多达 250 个列。对于一个数据库而言,这实际上是一个很大的值了。SQL Server 允许在一个查询中组合多达 16 个表中的列。

一个表中行的数目没有限制。因此,它只受磁盘空间的限制。SQL Server 允许数据库扩展到最多包括 32 个物理磁盘。

32 个磁盘的限制是通过附属逻辑结构间接实现的。我们将 SQL Server 数据库、事务日志、索引和表存储在称为段的逻辑结构上。可以通过增加在设备上创建的段来扩展数据库。一个数据库最多可以包括 32 个段。

每个表最多可以定义 250 个索引,它们中间只有一个可以定义为聚簇索引。索引是一个数据结构,它可以使得检索表的行的速度比没有索引时快得多。在聚簇索引中,表的行被排序并以物理上有序的状态存储。也就是说,排序的行以排序的状态存储。索引通常为需要在检索语句中使用的列定义。250 个索引提供了表行的快速检索。

在数据库中,索引需要额外的存储空间存储索引的结构。从性能上考虑,我们应该只定义检索时确实需要的索引,因为它们需要占据存储空间。当然,还是需要定义足够的索引来加快检索速度。当然,对于单个表,需要 250 个索引是不可思议的。

1.2.2 SQL Server 的关系特性

SQL Server 的关系型数据库模型是其最重要的特性。

在关系数据库模式中,虽然数据是存储在二维表中,但它们是可以相互参照的。数据的实际物理存储对于其参照性并不是很重要的。二维表的数据库表模式可以使数据作为表的行和列相互参考。

在关系数据库中,数据是按表的行和列存储并进行相互参照的。可以方便地将数据存储作为二维表的形式进行可视化处理,因为二维表的数据存储在日常生活中是经常遇到的。例如,火车和飞机时刻表常常以二维表的形式给出。

关系数据库模式中的表行是无序的。SQL Server 中使用的关系数据库模式中,数据库表的行一般也是无序的,除非为表创建一个聚簇索引。在创建一个表的聚簇索引后,表行以一系列或多列的升序形式存储。

最初的关系数据库模型需要表的每行至少有一列是唯一定义的,即为唯一键。这种唯一性可以用于对表行进行独立的、唯一的访问。用于访问表行的查询语言只能使用存储在每行中的数据来区别各个表行。但是,SQL Server 不需要定义唯一表行,可以创建一个表的两个或多个不能彼此区分的行。虽然这种重复的行可能没有用,但在某些情况

下用户可能会觉得使用这种行比较方便。

在关系数据库模型中,存储在一个或多个数据库中的多个表中的数据可以在某个查询的行访问中结合起来。例如,Employees 表包括列 Name、Department 和 Badge, Pays 表包括列 Hours _ Worked、PayRate 和 Badge。在两个表中都有 Badge 列。这样,通过指定两个表中都有的信息,可以在一个查询中检索两个表的内容。这个例子使用了 SQL Server 可以从有相关信息的表中检索数据行的关系功能。这个 SQL 检索的 SELECT 语句如下所示:

```
Select *
FROM Employees, Pays where
WHERE Employees.Badge = Pays.Badge
```

如果修改或删除 Employees 表中的一个 badge 号码,则在 Pays 表中的相关 badge 号码也必须修改或删除。这种确保相关表的相关值以得到数据的一致维护的处理过程称为相关完整性。如果在某个表中删除了主记录,则其他表中的相关信息也必须删除。

如果表行是唯一的,则维护相关完整性就比较方便。在关系数据库模型中,唯一定义的某列或某些列称为表的关键字或键(key)。

唯一定义表行的关键字称为主关键字(primary key)。如果在一个表中定义为主关键字的列加入到另一个表中,则在另一个表中的这些列称为外关键字(foreign key)。外关键字可以将两个表的信息关联起来。

在以前的数据库系统中,在数据库中要创建和维护内部指针来关联表中的数据行。这种指针结构会引起一个问题,即当创建数据库时,我们必须定义以后检索中要结合的数据。

旧的层次和网络数据库模型不使用行列结构。它们使用自己的术语系统描述数据。例如,关系数据库中的行称为记录类型,而列称为数据项。

在像 SQL Server 这样的关系数据库中,可以随时加入新列到一个表中。这时只要使用 ALTER TABLE 语句来修改已有的数据库表即可。

1.3 SQL 语言

要想读写或修改数据库中的数据,必须使用由 DBMS 软件定义的一套命令或语句(语言)。对于关系数据库,有很多不同的语言可供使用,但其中最常用的是 SQL 语言——结构化查询语言(Structured Query Language)。目前最新的 SQL 标准是 1992 年制定的 SQL _ 92,它的全称是“International Standard ISO/IEC 9075: 1992, Database Language SQL”。

SQL 标准的确定使得大多数数据库厂家纷纷采用 SQL 语言作为其数据库检索语言,这些厂家又在 SQL 标准的基础上进行了部分扩充,形成各自数据库的检索语言。所以 SQL 语言还存在许多版本,如 Oracle 使用 PL/SQL,DB2 使用特定平台的 DLL,而 SQL Server 7.0 使用的是一种叫 Transact - SQL 的版本。Transact - SQL 满足 SQL _ 92 标准,并且提供了满足其他编程要求的一些附加功能。

由于后而有专门的章节来讲述 SQL 语言,所以本节不再赘述。

1.4 客户机/服务器结构

随着计算机工业的不断进步,客户机/服务器正逐渐开始支配所有形式的网络计算,并且由于在商业领域和局域网的实现而成为客观上的标准。

所谓客户机/服务器,就是存在一个中央计算机,也就是服务器,而众多用户的计算机形成客户机。与过去流行的主机系统不同的是,客户的计算机不再是简单的终端,而是具有一定信息处理能力的机器。也就是说,客户机/服务器系统把信息处理能力分散给各个用户,而不是保留在中央计算机上。服务器存储着公用的重要信息,可以对信息的管理和安全进行严格的控制。客户通过自己的前台(即应用程序)向服务器申请对信息的使用,而对信息的处理过程在客户端进行,最后客户机将处理完的信息存储在服务器上,使得其他客户能够使用。这种分工是考虑到由于个人计算机的性能不断提高,已经能够担任一定的信息处理业务,以便使服务器能够集中精力来对信息进行维护和管理。这样,用户还能够自由地设计自己的应用程序,实现对信息的最优化使用。

几乎所有的客户机/服务器系统都可以分成三部分:客户机、中间件和服务器,如图 1-1 所示。

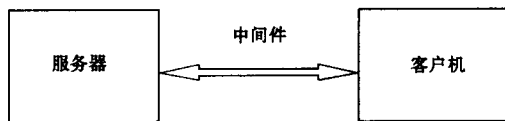


图 1-1 客户机、中间件和服务器模型

客户机/服务器计算是一种分布式模式的数据存储、访问和处理过程。在分布式处理系统中,多个计算机可以一起进行某些操作。一个客户机/服务器系统至少要使用两台计算机。

由微机系统组成的分布式处理系统可以提供大型机的功能。几个微机的数据存储、访问和处理能力合起来可以与一台大型机的某些处理功能相当。

在客户机/服务器模式中,每个系统都可以进行一个或多个数据存储、访问或处理操作。当 PC 机和服务器连接后,整个处理过程就可以在服务器、大型机或微机、客户系统间进行分别处理。

客户机/服务器模式包括处理过程的不对等区分,这种不平等性是因为服务器和客户端的处理不一致性。更大更快的服务器计算机比客户系统有更快的传递速度,更大的数据存储能力,且可以操作更广泛的处理工作。

小的 PC 系统用于客户机/服务器模式的客户系统,因为 PC 机只有较小的数据处理能力,而它主要是依赖于服务器来进行负载较重的数据操作。此外,PC 机的键盘和监视器可以用于数据输入和输出。

客户机和服务器也可以通过数据流的方向和操作职能来定义。如果一个功能强大的 PC 系统从一个或多个 PC 系统接收命令和数据,并进行数据处理以及返回信息到其他 PC 系统,则它可以作为服务器。服务器是从其他计算机系统接收处理要求或信息的计算机系统。

我们可以使用功能较强的 PC 机作为服务器,而用功能较弱的 PC 机作为客户机。这样仍然可以使用客户机/服务器技术。在这样的环境下,PC 服务器通常比平均的 PC 机具有更强的功能,这样它就可以处理来自客户端的处理请求。

1.4.1 SQL Server 的客户机/服务器特性

SQL Server 是一个客户机/服务器系统的完美例子。SQL Server 数据库必须安装在 Windows NT 平台上,Windows NT 操作系统提供了极强功能的处理器系统作为服务器。Windows NT 在基于 I386、PPC、MIPS 和 Alpha AXP RISC 处理器的系统上都得到支持。

SQL Server 提供了安装在服务器系统上的服务器软件和安装在客户机系统上的客户端软件。Windows GUI 应用程序工具可以从客户端创建、维护和访问数据库以及所有的数据库对象。

连接客户机和服务器计算机的网络软件组件是在 Windows NT 系统中提供的。Windows NT 还提供了对客户机和服务器间通信的网络协议的选择。客户机可以运行 Windows 95,它也包括内建的可以与 Windows NT 服务器系统连接的网络软件。Windows 95 客户机和 Windows NT 服务器系统可以支持较大范围的网络功能处理。

在 SQL Server 客户机/服务器数据库系统中,服务器负责创建和维护数据库及其对象,如表和索引。服务器还要维护相关完整性和安全性,并确保操作可以在事件失败时恢复。

客户操作所有的用户交互,包括信息显示和操作有图形化用户接口的应用程序。在检索服务器中的数据之后,应用程序可以创建保留在本地的拷贝并可以处理数据,还可以控制信息访问的类型。只读访问通常是最好的,它可以将用户与他们处理的服务器上的信息的主拷贝隔离。

如果要处理信息的本地拷贝,可以不用与服务器通信就在本地处理信息。完成工作后,可以将修改的信息送回到服务器;如果信息只是简单进行检索的,则可以将其丢弃。当然,如果需要的话,也可以直接在客户端的 SQL Server 数据库中操作数据。我们必须确定用所有的修改信息更新服务器,以便其他用户可以访问到数据库中最新的数据。

还可以直接从服务器访问 SQL Server。直接进行服务器访问是很方便的,尤其是对于隔离操作,但是这并不是客户机/服务器方式的操作,因为操作是在本地而非网络上。

虽然我们可以使客户机应用程序确认新数据或更新数据,这个确认最好还是在服务器端进行。例如,可以检查某列以确定每个新的或更新的数值是唯一的且在某个范围内。对于数据而言,在服务器端作为 SQL Server 定义的机制进行确认比较安全。如果确认定义在服务器端,则不管连接的客户机是否操作了确认都会在服务器端进行确认。

使用服务器端确认的一个主要好处是:它可以保护数据库数据不受可能用非标准形式访问数据库的应用程序的访问。这些应用程序包括可以使用 ODBC 连接到数据库的应用程序(例如 Excel、Access 和 Word)。这时在服务器上仍会实现规则和完整性检查,只是客户应用程序不知道而已。

如果要在客户机应用程序传送到服务器前确认数据,必须确保所有的客户机应用程序操作一致,还需要确保不会在没有定义确认机制的服务器上直接进行修改。在这一方面,实现服务器确认更加简单有效。

除在服务器上确认,我们还可以在客户机应用程序中操作确认。客户机确认对于没有被服务器确认支持的客户机应用程序而言可能比较特殊。当更新传送到服务器时,它还会执行自己的确认机制。

1.4.2 客户机/服务器数据库系统

建立客户机/服务器系统可以使数据库存放在某台计算机即服务器上,被若干用户共享,用户通过客户端或服务器应用程序访问服务器。

(1) 对一个双重客户机/服务器系统,用户在他们的本地计算机——客户机上运行应用程序,这些计算机通过网络连接到运行 SQL Server 的服务器。客户端应用程序运行事务逻辑和程序代码,并把输出结果显示给用户。

(2) 对一个多重客户机/服务器系统,客户端应用程序在两处运行:客户端应用程序运行在用户的本地计算机上,并用来向用户显示结果;客户机向服务器应用程序发出请求。服务器程序本身就是多线程的,能够和很多用户同时工作。这里的服务器应用程序是用来和数据库服务器建立连接的应用程序,它可以运行在该数据库服务器上,也可以通过网络连接到的数据库服务器上。

这一点对 Internet 应用程序来说是很典型的。举例来说,服务器应用程序可以运行在 Microsoft Internet 信息服务(IIS)上,并服务于运行在 Internet 或 Intranet 上的成千上万的客户机。服务器应用程序用连接池和 SQL Server 的副本通信。SQL Server 可以被安装在装有 IIS 的计算机上,也可以安装在网络中的一台单独的服务器上。

在客户机/服务器结构中,数据库文件一般是集中存放在服务器上的,这样做的好处有如下几点:

(1) 每个数据项都存储在服务器上,这样所有的用户可以共享数据。单独的数据项副本不存放在每台客户机上,避免了当所有用户同时操作相同信息时可能引起的错误。

(2) 事务和安全标准可以在服务器上一次性指定,并在所有用户中公平地实施。这些可以通过使用约束、存储过程和触发器来完成,也可以在服务器应用程序中完成。

(3) 关系数据库服务器通过只返回应用程序需要的数据优化网络流通量。

举例来说,一个应用程序需要显示华北地区销售代表名字列表,如果它是和文件服务器一起工作的,那么它就要得到整个员工文件,但如果它是和关系数据库服务器一起工作,它只发送这条命令:

```
SELECT name
FROM employees
WHERE emp_title = 'Sales Representative'
      AND emp_state = 'NorthChina'
```

关系数据库返回的仅是华北地区销售代表名字列表,而不是有关员工的所有信息。

(4) 最小化硬件成本,因为数据不是存储在每个客户机上,客户机也不用考虑如何管理数据,并且服务器也不用耗费资源以显示数据。服务器上可以进行设置,以优化磁盘输入和输出性能,而客户机也可以通过设置,优化对从服务器上获得的数据的格式化和显示。

(5) 维护任务诸如备份和恢复数据被大大简化,因为数据集中在服务器上。

对大型的客户机/服务器系统,可能有几千个用户同时连接到 SQL Server 上,SQL Server 对这些环境有完全的保护,防止了诸如多个用户同时修改同一个数据。SQL Server 还可以在多个用户间有效地分配可用资源,如内存、网络带宽和磁盘 I/O 等。

SQL Server 应用程序可以运行在 SQL Server 所在的计算机上,它通过 Windows 进程间通信(IPC)组件和 SQL Server 连接,这使得 SQL Server 可以用在小型系统上,这时应用程序需要在本地计算机上存储它的数据。

SQL Server 客户机/服务器数据库系统的组成可以用图 1-2 来表示。

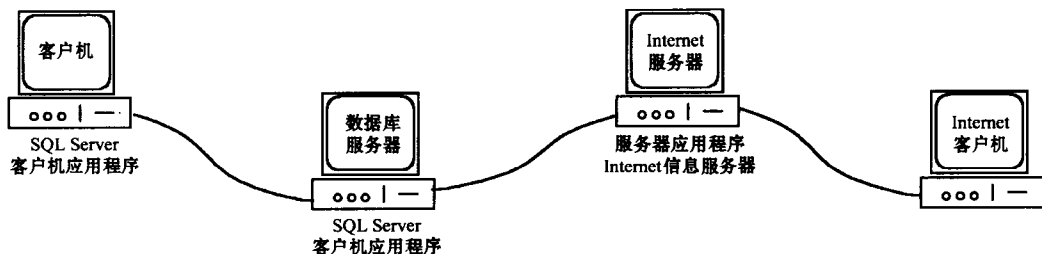


图 1-2 客户机/服务器系统的组成

1. 登录

要连接到 SQL Server,应用程序需要知道运行 SQL Server 的服务器的网络名和登录账号(ID)。

登录账号用来控制对 SQL Server 系统进行访问的权限,只有在 SQL Server 验证身份后才能够连接到 SQL Server 上。SQL Server 使用两种类型的验证方式:SQL Server 验证和 Windows NT 验证。

(1) SQL Server 验证:SQL Server 系统管理员首先设定所有有效的 SQL Server 登录账号和口令。注意,这不同于 Windows 账号或网络账号。只有提供了正确的 SQL Server 账号和口令,应用程序才能连接到 SQL Server 上。

(2) Windows NT 验证:系统管理员必须首先指定可以连接到 SQL Server 上的 Windows NT 账号。使用这种验证方式,在连接到 SQL Server 时不用提供登录账号和口令,对 SQL Server 的访问由 Windows NT 账号和组来控制,这已经在登录到 Windows NT 时得到了身份验证。

2. 客户端组件与通信组件

所谓客户端组件就是客户应用程序,用户并不直接访问 SQL Server,而通过应用程序客户来访问位于 SQL Server 的数据。应用程序通过数据库应用程序接口(API)来访问 SQL Server,一种数据库应用程序 API 包括两个部分:传递给数据库的语句和一组函数或面向对象的接口或方法。

SQL Server 支持两类主要的数据库 API:OLE DB 和 ODBC。有关 API 的内容将在第三章详细介绍。

SQL Server 支持多种方法用于客户端应用程序和服务器之间的通信。当该应用程序位于 SQL Server 服务器上时(如桌面级数据库系统),Windows 进程通信组件(IPC)被使用;当应用程序位于单独的客户机上时,则使用网络进程通信组件。

一个进程通信组件包括两部分:应用程序接口(API)和协议。API 定义了一组函数,

应用程序可以调用这些函数来使用进程通信组件;而协议则定义了通过进程通信组件进行通信的两个组件之间信息(包)的发送格式。

1.5 SQL Server 7.0 的主要特点

SQL Server 是一种基于结构化查询语言(SQL)和多线程的客户机/服务器型关系数据库管理系统。它具有客户机/服务器体系结构,能够满足大规模分布式计算环境的需要,可以与 Windows NT 及 Windows 2000 操作系统紧密结合,从而全面使用 Windows NT 和 Windows 2000 的管理和维护功能。所以,SQL Server 已经成为 Windows NT 平台上最重要、同时也是使用最广泛的数据库管理系统。该数据库系统包含 SQL Server 和 SQL Workstation 两种产品。SQL Workstation 相当于一个单用户许可的 SQL Server,最多只能同时连接 15 个数据库,但它提供了一些客户端数据库应用程序开发工具,如 DB_Library for C、DB_Library for Visual Basic、ODBC(Open DataBase Connectivity,开放式数据库互连)和 ODS(Open Data Services,开放数据服务)编程接口(API)等。除了这些差异之外,SQL Workstation 和 SQL Server 具有同样的性能和功能。

1.5.1 SQL Server 7.0 的不同版本

SQL Server 7.0 以三个版本发行:标准版、企业版和小型商业服务器版(SBS),表 1-5 比较了这三个版本的功能。

表 1-5 SQL Server 7.0 不同版本比较

功 能	SBS	标准版	企业版
能否运行在 Microsoft Backoffice 小型商业服务器上	是	是	否
能否运行在 Microsoft Windows NT Server	否	是	否
能否运行在 Microsoft Windows NT Server 企业版	否	是	是
最大数据库容量	3GB	无限制	无限制
支持对称多处理器个数	4	4	32
是否支持扩展内存	否	否	是
是否提供 SQL Server failover 支持	否	否	是
是否支持 Microsoft 全文本目录和索引检索服务	是	是	是
是否支持 Microsoft SQL Server OLAP 服务	否	是(不含用户定义的 cube 分区)	是(包含用户定义的 cube 分区)

虽然 SBS 也可以安装在少于 50 个用户的情况,但由于 SBS 的性能优势主要体现在吞吐量为 50 个用户时,因此用户少于 50 个时不能完全发挥 SBS 的性能优势。标准版和企业版的性能仅决定于硬件和操作系统。

1.5.2 SQL Server 的特点

作为一种功能强大的数据库系统,SQL Server 主要有以下特点:

(1) 完全的客户机/服务器体系结构。

这一结构可以更有效地使用网络。在客户机/服务器运算模式下,数据库的查询操作都集中在服务器方进行,在网络上传输的是用户的请求命令和服务器的检索结果,而不是整个数据库文件,这样可以减少网络上的信息流量,从而提高网络的使用效率。

(2) 简单的图形化管理工具,使系统管理更为直观方便。

SQL Enterprise Manager(SQL 企业级管理器)是一个基于 Windows 图形界面(GUI)的集成管理工具,利用它可以完成 SQL Server 的配置管理工作,如建立设备(devices)和数据库、备份和恢复数据、执行查询操作、调度和警报管理、数据复制等。SQL Server 的管理体系建立在 SQL 分布式管理对象(SQL Distributed Management Object,简称为 SQL_DMO)基础上,形成 SQL Server 的分布式管理框架。

(3) 丰富的编程接口工具。

SQL Server 提供了 Transact_SQL、DB_Library for C 和 DB_Library for Visual Basic 等专用开发工具。Transact_SQL 与工业标准 SQL 语言兼容,并在其基础上加以扩充,使它更符合事务处理的需要。此外,SQL Server 还支持 ODBC 规范,可以使用 ODBC 访问数据库。

(4) 多线程体系结构。

SQL Server 支持多线程操作,在多用户并发访问时,系统在产生较小额外负担的情况下能够进行并行处理,从而减少内存需求,提高系统吞吐量。在用户数量增加时,SQL Server 的运行速度也不会明显减慢。

(5) 隐含的并发控制能力。

SQL Server 利用动态锁定功能,防止用户在查询和更新并发操作时相互间发生冲突。动态锁定是隐含的,用户不必操心锁定过程。

(6) 良好的兼容性。

SQL Server 是 Microsoft 服务器套件 BackOffice 的成员之一,它与其他软件(Windows NT、Internet Information Server、Exchange Server 等)有机结合,并充分利用它们所提供的服务或功能(如安全管理、事件日志、性能监视器、内存管理和异步 I/O 等),从而增强了 SQL Server 数据库系统的功能,同时减少了系统资源的占用。

1.5.3 7.0 版的新特性

SQL Server 7.0 是 SQL Server 系列产品的最新版本,在符合 ANSI(美国国家标准局)标准并保持与 SQL Server 以前版本兼容的基础上,增强了服务器的结构功能,发展了图形化的管理界面,并优化了 SQL Server 结构体系和工具栏,这使得无论是管理还是使用都更加方便和简单。

新一代的 SQL Server 7.0 较前一代的版本在易用性、可扩展性、可靠性以及数据仓库等方面有了显著的改进和提高。Microsoft 公司宣称 SQL Server 7.0 是面向下一世纪的数据库,也是在于 SQL Server 7.0 版本功能的增加和提高。下面将简要介绍这些新特性,

从本书后面的章节中,读者可以感受到这些特性。

1. 服务器管理的增强和新功能

- 减少了管理负荷和费用。
- 动态锁定功能。
- 为复杂查询提供了重新处理模型。
- 增加了一些新的 Transact - SQL 语句。
- 触发器的增强。
- 游标的增强。
- 提高了容量。
- 分布式查询和修改。
- 使用了新的页和行的格式。
- 为庞大的数据库支持提高了实用程序性能。
- 采用了新的 SQL Server 安全模型。
- 编程界面。
- 增强的数据类型支持。
- 其他一些增强功能。

2. 图形管理界面的增强和新功能

- Microsoft Management Console 界面
- Web Assistant
- SQL Server Agent 服务
- SQL Server Query Analyzer
- SQL Server Profiler
- Data Transformation Services
- 索引调整向导
- Microsoft English Query

第 2 章

SQL Server 7.0 的安装、 启动、注册与配置

本章针对最常见的安装方式,一步步地详细讲解 SQL Server 7.0 的安装过程和服务器的注册及配置,同时对于安装后的平台环境和基本启动方法也将作一些介绍。