

计算机网络基础
与应用系列丛书

由经验丰富的专
家撰写

内容全面、覆盖
面广

教你维护SQL服
务器

(美) Jeffrey L. Byrne 著

李·坚 陈佩 蒋琪 译

Microsoft SQL Server 6.5 管理员手册

计算机网络基础与应用系列丛书

Microsoft SQL Server 6.5 管理员手册

(美) Jeffry L. Byrne 著
李小坚 陈佩 蒋琪 译
李小坚 刘炳兴 审校

机 械 工 业 出 版 社
西蒙与舒斯特国际出版公司

本书介绍了 Microsoft SQL Server 6.5 的最新特点,讲述了作为一名 SQL Server 管理员怎样去维护数据库的完整性、一致性、安全性。

本书语言简洁明了、通俗易懂、方便实用,是数据库管理人员的一本难得的参考手册。

Jeffrey L. Byrne: Microsoft SQL Server What Database Administrators Need to Know

Authorized translation from the English language edition published by Prentice Hall PTR.

Copyright 1997 by Prentice Hall PTR.

All rights reserved. For sale in Mainland China only.

本书中文简体字版由机械工业出版社和美国西蒙与舒斯特国际出版公司合作出版,未经出版者书面许可,本书的任何部分不得以任何方式复制或抄袭。

本书封面贴有 Prentice Hall 防伪标签,无标签者不得销售。

版权所有,翻印必究。

本书版权登记号:图字:01-97-0931

图书在版编目(CIP)数据

Microsoft SQL Server 6.5 管理员手册/(美)白尼(Byrne, J. L.)著;李小坚等译.-北京:机械工业出版社,1997.9

书名原文:Microsoft SQL Server What Database Administrators Need to Know

ISBN 7-111-05890-9

I. M… II. ①白…②李… III. 数据库管理系统, SQL Server 6.5 手册 IV. TP311.13-62

中国版本图书馆 CIP 数据核字(97)第 17496 号

出 版 人:马九荣(北京市百万庄南街 1 号 邮政编码 100037)

责任编辑:江 颖

三河永和印刷有限公司印刷·新华书店北京发行所发行

1997 年 9 月第 1 版 第 1 次印刷

787mm×1092mm 1/16·12.5 印张·296 千字

印数:0001-9000 册

定价:22.00 元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换

译者序

Microsoft SQL Server 是运行在 Windows NT 上的一个著名的高性能数据库管理系统。它基于多线程的客户/服务器体系结构,这一点与其他流行的多进程客户/服务器数据库系统不同。多线程的数据库系统与多进程数据库系统相比,装入 Microsoft SQL Server 时,仅仅启动一个进程,而每个用户有各自的线程。这种结构需要的资源比多进程系统少得多。

Microsoft SQL Server 6.5 扩展了 Microsoft SQL Server 早期版本的性能、可能性和可扩展性,以便满足大规模分布式信息系统用户的需要。SQL Server 6.5 采用了改进的 ANSI SQL 支持和包括声明一致性(DRI)在内的语言强化技术,以期达到继续坚持工业标准的目的。

本书分为 12 章。第 1 章预习 SQL Server,讲述了 SQL Server 应用程序的优点,以及多进程与多线程的区别。第 2、3、4 章的内容包括:常用的 SQL 命令 SELECT、INSERT、UPDATE 和 DELETE;各种数据库对象;SQL Server 的结构。第 5、6 章是关于数据完整性和一致性,讲述各种完整性,怎样用约束和规则来实施完整性,怎样用锁来保证并发事务中数据的一致性。第 7 章是怎样启动和关闭服务器。第 8、9 章内容包括怎样确定数据库空间,扩展和压缩数据库设备,以及怎样优化性能,其中包括内存优化、磁盘优化和查询优化。第 10 章是关于安全性,安全的模式分为集成安全、标准安全和混合安全;以及各种安全级别。第 11、12 章包括数据的备份和恢复以及分布式处理的两阶段托付原则。从基本的 SQL 命令到优化、安全、备份等高级专题部分,作者都深入浅出,娓娓道来,几名话就把一个难懂的问题讲得简明透彻,这也是本书的一大特色。

全书由李小坚、陈佩、蒋琪三人共同翻译,审校由李小坚、刘炳兴完成。其中第 1、7 和第 8 章由李小坚翻译,第 2 章~第 6 章由陈佩翻译,第 9 章~第 12 章由蒋琪翻译。译者由于水平有限,很难表达出原著语言的细微之处;加之时间仓促,错误在所难免,请读者批评指正。

译者

1997 年 8 月

目 录

译者序

第1章 Microsoft SQL Server 概览1

1.1 客户/服务器和 Microsoft SQL Server.....1

1.2 为什么使用 SQL Server 应用程序.....2

1.2.1 使用方便.....2

1.2.2 简化系统.....3

1.2.3 实施企业规章.....4

1.2.4 设计自定义应用程序.....4

1.2.5 费用因素.....4

1.3 多进程与多线程.....4

1.3.1 多进程系统.....5

1.3.2 多线程系统.....5

1.4 网络.....6

1.5 小结.....7

第2章 SQL 是什么8

2.1 关系数据库.....8

2.1.1 规范数据库.....9

2.1.2 表格、列和行.....11

2.1.3 什么是关系型数据库.....11

2.1.4 设备独立性.....12

2.2 SQL 命令.....12

2.2.1 数据定义语言(DDL).....13

2.2.2 数据操作语言(DML).....13

2.2.3 数据控制语言(DCL).....13

2.3 常用命令.....13

2.3.1 SELECT.....13

2.3.1.1 使用 SELECT 语句.....14

2.3.1.2 用 SELECT 语句连接表格和 排序.....16

2.3.1.3 创建别名.....17

2.3.2 INSERT.....19

2.3.3 UPDATE.....21

2.3.4 DELETE.....22

2.4 事务和 SQL.....22

2.5 小结.....24

第3章 逻辑结构.....25

3.1 设备.....25

3.1.1 数据库设备.....25

3.1.2 备份设备.....28

3.2 数据库对象.....30

3.2.1 表格.....32

3.2.2 视图.....35

3.2.3 游标(cursor).....36

3.2.4 索引.....37

3.2.4.1 聚集索引.....39

3.2.4.2 非聚集索引.....39

3.2.5 缺省.....41

3.2.6 规则.....42

3.2.7 触发器.....43

3.2.8 存储过程.....45

3.2.9 标识符.....46

3.2.9.1 规则.....47

3.2.9.2 对象名字.....47

3.2.9.3 引号标识符.....48

3.3 数据库大小.....49

3.3.1 页.....50

3.3.2 盘区.....51

3.3.3 分配单元.....51

3.4 系统目录.....51

3.5 小结.....52

第4章 Microsoft SQL Server 的结构.....53

4.1 服务器结构.....53

4.1.1 数据库设备.....53

4.1.2 备份设备.....54

4.2 进程.....55

4.3 文件.....57

4.3.1 master 数据库.....57

4.3.2 model 数据库.....57

4.3.3 msdb 数据库.....58

4.3.4 pubs 数据库.....58

4.3.5 事务日志文件.....58

4.3.6 Web 文件.....60

4.4 小结.....66

第5章 数据完整性.....67

5.1 实体完整性.....67

5.1.1 主关键字约束.....67

5.1.2 唯一性约束.....68

5.2 域完整性.....69

5.2.1 域规则.....69

5.2.2 数据类型.....69

5.2.3 CHECK/NOCHECK 约束	72	9.1.2 分配内存	107
5.3 参照完整性	72	9.2 磁盘性能的优化	109
5.3.1 什么是参照完整性	73	9.3 段的使用	110
5.3.2 定义主/外关键字	74	9.4 多线程服务器	111
5.3.3 实施参照完整性	76	9.5 查询优化	112
5.4 用户定义完整性	76	9.5.1 基本成本的优化	112
5.5 小结	78	9.5.1.1 统计数字的收集	113
第6章 数据一致性和并发性	79	9.5.1.2 更新统计数字	115
6.1 语句和事务	79	9.5.2 聚集索引	117
6.2 上锁保证一致性	79	9.5.3 使用非聚集索引	118
6.2.1 理解锁的含义	80	9.5.4 创建有用索引	118
6.2.1.1 共享锁	80	9.6 删除数据库	119
6.2.1.2 修改锁	80	9.7 小结	120
6.2.1.3 排他锁	81	第10章 安全性	122
6.2.2 锁的粒度	81	10.1 识别和确认	122
6.2.2.1 行锁	81	10.1.1 Windows NT 安全	123
6.2.2.2 页锁	82	10.1.2 SQL Server 安全	123
6.2.2.3 表格锁	83	10.1.3 数据库安全	124
6.2.3 死锁	83	10.2 系统管理员的任务	124
6.2.4 在 SELECT 中使用锁	83	10.2.1 选择安全模式	124
6.2.5 设置事务隔离层次	84	10.2.1.1 集成安全	124
6.3 小结	84	10.2.1.2 标准安全	128
第7章 启动和关闭服务器	85	10.2.1.3 混合安全	128
7.1 启动 Microsoft SQL Server	85	10.2.1.4 可信任连接	129
7.2 启动的选项	87	10.2.1.5 安全模式设置	129
7.3 关闭 Microsoft SQL Server	87	10.2.2 创建用户	130
7.3.1 暂停服务器	88	10.2.2.1 设计用户名	130
7.3.2 关闭服务器	88	10.2.2.2 Guest User	132
7.4 小结	89	10.2.2.3 创建组	132
第8章 数据库空间的管理	90	10.2.2.4 所有者	133
8.1 设计所需的数据库空间	90	10.2.2.5 使用别名	135
8.1.1 确定数据库空间	91	10.2.2.6 口令	137
8.1.2 确定事务日志空间	92	10.2.3 授予和废除权限	137
8.2 扩展数据库	93	10.2.3.1 对象权限	138
8.2.1 处于同一数据库设备	93	10.2.3.2 用户权限	139
8.2.2 处于不同的数据库设备	98	10.2.4 使用 SQL Security Manager	141
8.3 压缩数据库	98	10.3 小结	143
8.4 删除数据库设备	101	第11章 备份和恢复	145
8.5 小结	102	11.1 故障类型	145
第9章 性能优化	103	11.1.1 客户应用程序故障	146
9.1 存储区使用优化	103	11.1.2 程序故障	146
9.1.1 系统内存	103	11.1.3 磁盘或介质故障	146

11.2 备份和恢复.....147	第 12 章 分布式事务处理、复制、 出版和订阅167
11.2.1 备份的类型.....147	12.1 分布式数据库.....167
11.2.1.1 离线或静态备份.....147	12.1.1 建立服务器.....168
11.2.1.2 在线或动态备份.....148	12.1.1.1 建立分发服务器.....168
11.2.2 建立备份策略.....148	12.1.1.2 建立出版和订阅服务器.....170
11.2.3 创建一个备份.....149	12.1.2 复制.....173
11.3 恢复数据库.....152	12.1.3 分发.....181
11.3.1 重装载丢失的数据库.....152	12.1.3.1 同步过程.....181
11.3.1.1 删除有缺陷的数据库.....153	12.1.3.2 日志阅读过程.....181
11.3.1.2 恢复数据库备份.....154	12.1.3.3 分发服务器和分发数据库.....181
11.3.2 使用事务处理日志.....157	12.1.3.4 复制清除.....182
11.3.3 恢复 master 数据库.....159	12.2 两阶段托付.....184
11.4 重建丢失设备.....161	12.2.1 定义.....184
11.5 重型保护——镜像和 RAID.....162	12.2.2 两阶段托付如何运行.....184
11.5.1 镜像一个设备.....163	12.2.3 配置 MS DTC.....187
11.5.2 RAID 磁盘阵列.....163	12.2.4 问题区域.....188
11.6 使用可移动介质.....165	12.3 小结.....190
11.7 小结.....166	

第1章 Microsoft SQL Server 概览

这章主要包括:

- 客户/服务器和 Microsoft SQL Server
- 为什么使用 SQL 应用程序
- 多进程与多线程
- 网络

拿起这本书你马上会迷上它,因为能够了解更多关于 Microsoft SQL Server 的信息。Microsoft SQL Server 是运行在 Windows NT 上最畅销的数据库服务器。作为一种关系型数据库管理系统(RDBM),是客户/服务器的一部分,一般被看作后台程序;而前台程序,或者说客户程序,一般是指用户设计的应用程序,也有可能是在市场能买到的独立数据库软件,如 Microsoft Access。

这一章里会讲解什么是 Microsoft SQL Server,以及为什么它会适合于客户/服务器方案。

1.1 客户/服务器和 Microsoft SQL Server

主机系统虽然速度快且功能强大,但是在过去和现在总是很昂贵,而且要求专门的 MIS 人员编程以及维护。这样一来,为了获得其他部门的信息必须向 MIS 部门提出申请,然后,他们再判断在主机数据库中是否有这条信息,并决定怎样去查询,以及以什么样的格式向需要信息的部门提交这份报告,这每一步都要花费时间和金钱。如果是每个部门自己支付这笔费用,那么,为了不支付这笔昂贵的费用,在决策时他们很可能就利用一些不完全的信息,即使该决策对企业来说很关键。

相反,对许多小公司和有计算需要的部门来说,个人计算机/局域网(PC/LAN)非常引人注目了。许多公司部门发现,用市场上的数据库或电子表格建立一个应用程序,可以为他们提供所需要的信息,并且花费的时间非常少,费用也相对来说低。记住,在大多数企业中,为部门设计应用程序的全部费用由该部门支付。使用 PC/LAN 之后,就没有必要向 MIS 部门提出申请,并且避免了把时间花在某些事情上,而在主机系统中这些又是必要的:申请获取什么信息,怎样才能获得这条信息,怎样创建一个应用程序或查询,然后又怎样才能把信息送回来——希望信息的格式能够让人满意。

PC/LAN 系统的自由结构既是它的最大优点,同时也是最大缺点。虽然能让个人和部门迅速获取信息,但是,它导致大量的冗余信息,而且还存在一些不相容的信息。这看起来似乎自相矛盾。在大公司中——也许小企业也是这样——基本的信息存储在许多不同的部门,每个部门的信息基本相同,但格式不同,使用的程序和操作系统不同。为了能够共享相互之间的数据,需要解决这些问题,但是要耗费大量的编程时间和维护时间。这也是在浪费金钱和计

算机资源,因为能够把它们作别的用途。

客户/服务器系统突破了主机系统和基于 PC/LAN 的系统的障碍,分散了处理任务,也就是说把处理能力分散给各个用户,而不是全保留在主机上。服务器 RDBM 实实在在存储所有数据,对基本的数据库结构担负主要责任,可以对数据完整性、管理和安全性进行严格的统一控制,因为像这样把数据存储在中心服务器上很容易让系统管理员备份数据和定期维护数据和服务器。客户向服务器提交自己的应用程序——前台——选取的信息再从服务器返回客户。这种分工使得服务器能够集中精力,尽量做好它的本分工作:存储数据、保证安全性和提供数据;客户设计复杂的用户界面,这是非常占用处理器时间的任务,例如,用窗口和图形表示表格和数据。

把客户和服务器紧紧联系在一块的是结构化查询语言(SQL)。在大多数文件分享的 LAN 数据库管理系统(DBMS)中,响应查询时传送大量不必要的数据——通常是整个文件;而客户/服务器不同,只传送查询中指定的信息,因此,是一种更加有效的信息管理系统。服务器数据库系统和客户应用程序用 SQL 相互通信:首先,客户应用程序传送给服务器一个查询;然后服务器解释该查询,接下来用一个称为优化器的过程运行它,判断用哪种方法获取信息最快,最后把数据送回给客户应用程序。与以前的文件分享系统相比,这种方式把更多的处理负担加在保存数据的服务器上。为了减少网络运输量,减轻工作站 CPU 的计算负担,在客户工作站上应该保存少量的数据。

Microsoft SQL Server 功能强大,特点之一是其保证数据的完整性。有了服务器维护数据的完整性,就没有再必要依赖各个部门的程序员,他们能力各异,编写的应用程序也很难统一。虽然可以在应用程序内部努力保证完整性,但是没有必要完全寄希望于此,因为有了 Microsoft SQL Server,可以在整个 LAN 内部保证数据库信息的完整性和一致性。

1.2 为什么使用 SQL Server 应用程序

用于办公自动化的计算机程序包括下面几种类型:字处理器、电子表格、数据库和桌面排版,这些程序是专门针对某一种特定任务的,例如字处理器,可以使用它创建一个复杂的文档,然后按格式编辑它,最后打印,而无需花很长时间先打草稿,也不要重敲这段冗长的文档。

SQL 应用程序本身是面向信息的,包括帐目、工资表、销售量、天气情况、顾客记录、利润、卖主记录,以及有关要收集数据的所有其他信息。大多数 SQL Server 应用程序用来简化数据收集和存储的管理,但 SQL 客户/服务器应用程序还有其他用途。

1.2.1 使用方便

大多数客户/服务器应用程序使用起来很方便,提供直观的界面,屏幕上显示的就是通常形式的文件。因此,销售人员过去也许习惯于自己填写表格,但现在由于使用销售定单应用程序,可以在屏幕上看到相同的表格。除了使用方便外,销售人员不必查找客户或销售项目的信息,应用程序会提供弹出列表,销售人员只需从中选取客户,从另一个列表中选取销售项目。由于防止以错误的价格销售,而且在公司数据库中消除了重复数据,因此能够保证一致性。这样只需简单地敲几下键盘,单击几下鼠标,就完成了—一个表格,然后打印出来。

由于前台应用程序的用户友好界面,销售人员、顾客和库存管理员的效率显著提高,同时提高公司记录和帐目的准确性。

1.2.2 简化系统

许多公司的系统并不遵循客户-服务器模式。虽然他们利用办公自动化的某些方面完成一些特定的工作,但不能保证前台客户应用程序和作为后台的 Microsoft SQL Server 联合起来能够提供的整体性。下面看看绝大多数企业中典型的交易链。

销售人员收到客户的定单,然后也许要用计算器计算金额总数,接着把这个写好的定单放在一个篮子里,送到定单输入员,他收到定单后把它输入计算机。计算机再把定单传送到库房,在那里打印提货单,库房人员用它来结单。结单后,把提货单复制两份,分别送给帐目人员和清单管理员。帐目人员输入订购的数量、价格信息,填写装货清单,并且查看顾客的存款,并把它输入到帐目文件中。同时,清单管理员修改清单文件,表示已经销售了该项目。

这种典型系统至少涉及到 6 种事务,6 种不同的人员。在这条链上的任何一点都可能出现错误,因为系统需要在每一点上都要输入、然后输出数据。虽然信息的用途和格式在事务之间各不相同,但每个事务只包括一些特定的信息:顾客、项目、数量和价格。

如果从客户-服务器的观点看这个事务链,很容易消除或减少了这些冗余的事务。在客户-服务器应用程序中,只需要输入一次定单,应用程序就会自动处理剩下的所有事务。定单输入员收到定单,输入到屏幕上的表格中,应用程序就会自动输出客户名、销售项目以及总金额。如果这个客户由于信用问题而在他的记录上做了一个标记,那是用来提醒接收定单的人员注意这点。如果客户住址、电话号码改变,可以在应用程序中修改他的文件;当然,也可以修改存货清单,在库房打印提货单,输出存货清单。

这里的每一步事务也许是由不同的前台客户应用程序处理,但都在同一个后台 SQL Server 数据库中。SQL Server 把数据合并起来管理,既能够保证数据的完整性,也能够保证安全性。应用程序允许用户用各自的形式处理这些基本数据,但信息只需要输入某一个应用程序中。图 1-1 说明信息是怎样流经各个事务,完成从初始的定单到最后运货、支付现金的一个循环。

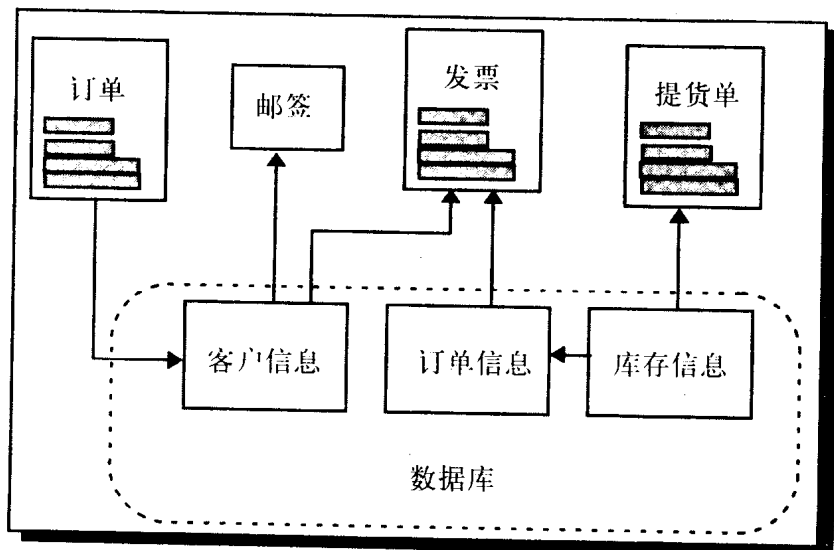


图 1-1 基于服务器的数据库系统很容易减少一些冗余的事务,各个不同的前台应用程序分别满足不同部门的要求

1.2.3 实施企业规章

企业规章包括许多方面,在 Microsoft SQL Server 中称为规则 (rule), 主要与输入字段的数据有关——保证数据的正确性和完整性。如果应用程序有很好的完整性特点,那么能够保证输入数据的正确性,实施其他规则。

例如,在为卖主发票付款之前得保证已经接收到货物。好的帐目系统应该要求订购货物的采购员与给卖主的发票支付支票的不是同一个人,也不是实际接收货物的人员。Microsoft SQL Server 很容易做到这一点,如果购买货物的定单还没有确定下来,或者还没有验收货物,应用程序不允许为发票支付支票。

好的应用程序可以为高级雇员减轻一些重复性的、简单的工作。采购员由于不要艰难地在键盘上敲订购单,能够花更多的时间确定卖主,检查货物。帐目员能够很快批准这个款项,而不要每次在开支票前检查收文。同时也有利于减少可能错误,而在非常依赖于人工检查的日常文书工作中,这种错误经常发生。

1.2.4 设计自定义应用程序

在市场上能够买到的许多软件包带有一组常见的企业规则或规章,它们不一定能够满足你的要求。有的应用程序考虑了这一点,允许修改规章;而有些软件出版商要求为修改程序支付昂贵的费用。虽然许多过时的软件也能够使用,但是,这常常是削足适履,需要改变企业运行的方式,而不是程序满足企业的要求。

为了满足特定的需要,可以利用各种不同的前台应用程序,设计自定义应用程序,例如高级程序设计语言 Visual C++、PowerBuilder,甚至 Visual Basic,或者数据库软件 Microsoft Access,每种语言都有优点和缺点。在短时间内设计出功能强大的应用程序产品,不仅节省系统的开发费用,而且能够打开新市场,提高顾客服务的性能,这样也就增强自身的竞争力。

1.2.5 费用因素

在创建客户/服务器系统时最后一个要考虑的因素是费用。许多客户/服务器系统正在逐渐取代传统的主机系统,因为后者的维护费用太高,这每年昂贵的维护使得已经很贵的价格更加突出。为了节省开支,许多公司正在试图转变为客户/服务器模式。

许多企业正在投资建立 LAN/WAN 系统,他们的目标是某些类型的办公自动化:文件共享、电话服务、传真、共享调制解调和电话会议。在 LAN/WAN 上增加客户/服务器系统不会增加很多消耗——金钱和网络运输量,这就保证不会由于新应用程序带来的负担和网络运输量而使系统负荷过重。

警告: 必须懂得客户/服务器系统能够为企业带来什么好处,这点非常重要,因为只有这样才能明确地告诉管理部门所能达到的实际目标。许多数据库程序员认为,他们已经达到管理部门制订的目标,但是,由于期望值不同,在管理人员眼中他们失败了。

1.3 多进程与多线程

客户/服务器数据库结构分为两种类型:多进程与多线程,各自都有支持者和否定者。Microsoft SQL Server 把多线程作为基础结构。

多进程系统是指在同一时刻运行多个程序。而一般情况下,当用户在系统中登录时,装

入整个可执行程序集——也被称作启动一个实例。这种类型的系统使用的资源明显比多线程系统多。

多线程系统是单进程的，Microsoft SQL Server 就是一个多线程数据库系统。装入 Microsoft SQL Server 时，仅仅启动一个进程；而每个用户有各自的线程，但不是都有自己的实例。这种结构需要的资源比多进程系统少得多。

1.3.1 多进程系统

在执行客户查询时多进程数据库系统使用一个可执行的应用程序系统，一个用户在数据库中登录就真正启动这个程序的一个实例。为了协调同一个数据库的不同用户，有其他的全局“协调程序”为每个已经打开的实例调度各种资源。

最流行的多进程 SQL 数据库系统是 Oracle Corporation's Oracle Server。每次一个用户在数据库系统中登录，启动一个 Oracle 实例，他们设计的查询从所使用的前台程序传送到该实例。然后该实例与其他可执行的程序相互协调，返回结果给用户。另外有一些通用的可执行程序用来管理文件、记录锁、写日志文件，把已经修改的文件写入磁盘文件中。如图 1-2 所示，每次一个用户在数据库中登录，就启动一个实例及其附带的一些可执行程序，每个程序都需要很多系统资源。

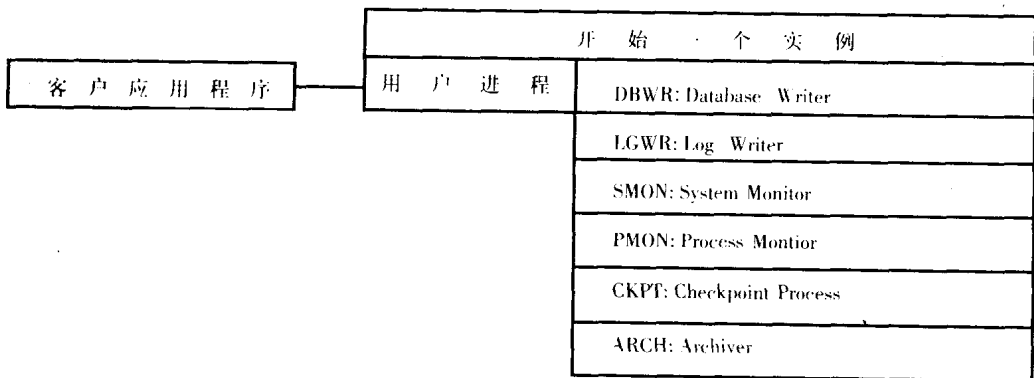


图 1-2 每次一个新用户启动一个实例，需要为它复制很多系统资源，因为这种处理方式是以每个用户为基础的

多进程数据库系统已经使用很多年了，也是一种比较好的多用户数据库。多进程服务器和多线程服务器都有升级性，能够往机器中增加 CPU 和内存。由于操作系统，像 Windows NT，能够使用多个 CPU，SQL Server 开始大显神威。因为可以把线程分配给不同的处理器，保证最高的 CPU 利用率。

1.3.2 多线程系统

多线程数据库亲自管理多个用户，而不是假定操作系统会管理用户。在使用资源方面，比多进程系统节省得多，启动它时，仅仅只有程序的一个实例被打开。每次新用户登录，都被赋予一个线程。线程是一个管道，所有用户的查询和指令都是通过它传送到数据库和操作系统的。

Microsoft SQL Server 使用的本地 Win32 操作线程（见图 1-3）与数据库内核的线程不一样，这是与其他多线程系统一样的。这些本地线程运行在各自的保护内存空间，这种多线程

类型能够保证在单个线程毁坏后不会给整个应用程序造成故障；相反，能够俘获这个线程，并且把它隔离开。

Microsoft SQL Server 有一个工作线程池 (pool)，最大容量是 1024，缺省值是 255 个工作线程。如果并发用户的数量没有超过池中工作线程的数目，那么给每个用户分配一个线程，否则，出现在池 (pooling) 情形，在池就是说用户一完成任务，他的线程就分配给另一个需要访问数据库的用户。

除了工作线程池外，另外还有其他池，用来分配其他系统资源，如磁盘设备、备份带区和并行表格扫描。

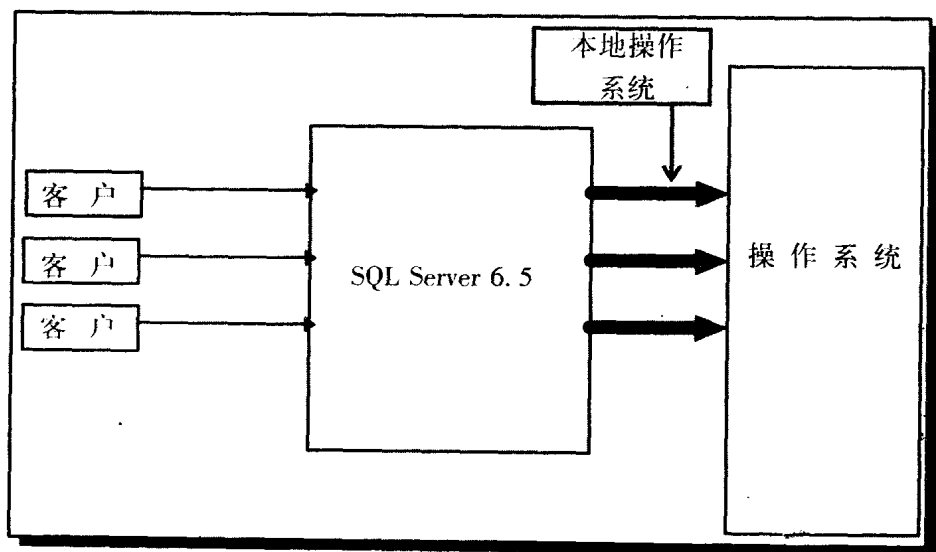


图 1-3 Microsoft SQL Server 从 Microsoft Windows NT Server 操作系统中使用本地 Win32 线程

Microsoft SQL Server 对内存要求取决于系统的功能：

- 如果系统主要用来开发 SQL Server 应用程序，并且没有复制的要求，可以在 16MB 的 RAM 上运行。
- 产品数据库环境中的系统可以使用任何量的 RAM，与所希望的一样，或者说是能够提供的。
- 有复制要求的系统至少得有 32MB 的 RAM，并且至少得有 16MB 专门用于 SQL Server。

1.4 网络

客户/服务器模式假定服务器应用程序 (这里就是 Microsoft SQL Server) 工作在基于 Microsoft Windows NT Server 的计算机中。客户机器通过网络电缆、网卡物理连接到服务器上，当然还得有合适的软件。所有客户必须有与服务器通信的功能，这样 SQL 查询才能传送给服务器，然后再把结果传送回客户。

在 Microsoft SQL Server 中，能够使用 Microsoft Windows NT Server 的网络能力与很多类

型的客户通信:PCs、Macintoshes、UNIX 工作站和 NetWare 客户。因为 Microsoft SQL Server 使用起主机操作系统(Microsoft Windows NT Server)的网络能力,所以可以授权任何连接到服务器的计算机,使之能够访问 SQL 数据库。

1.5 小结

这章讲述了关于 Microsoft SQL Server 的一些基本知识,以及为什么它适合客户/服务器方案,特别是:

- 为什么 Microsoft SQL Server 适合作为客户/服务器网络系统的后台部分。
- 使用客户/服务器数据库应用程序存储信息的原因:使用方便、简化数据库系统、能够实施企业规则和规章,能够把客户应用程序很方便作为客户/服务器的前台使用。
- 多进程系统与多线程系统之间的区别,以及联合使用 Microsoft SQL Server 和 Windows NT 作为一个多线程方案的优点。

第2章 SQL 是什么

本章主要包括:

- 关系数据库
- SQL 命令
- 常用命令
- 事务与 SQL

为了使客户应用程序和服务器数据库系统相互通信,它们必须使用同一种语言——结构化查询语言,通常也称之为 SQL。SQL 有几种不同版本,Microsoft SQL Server 使用一种叫 Transact-SQL 的版本。现在有两种的 SQL 标准:ANSI SQL-89 和 ANSI SQL-92,分别是在 1989 年和 1992 年由美国国家标准局颁布。Transact-SQL 满足 ANSI SQL-89 和 ANSI SQL-92 规定的所有标准,并且提供满足其他编程要求的一些附加功能。

在这章里会讲述怎样创建一个关系型数据库以及怎样组织数据。了解数据库的基本结构后会更加容易理解 Transact-SQL 的工作机理。

2.1 关系型数据库

关系型数据库管理系统(RDMS)是一种处理存储在表格中信息的系统。这些表格通过一系列关系联系起来:某一表格中的数据集与另一表格中相应的数据集之间的关系。

在设计之前,必须清楚数据将会存储在数据库里。提出并回答下面的问题以及其他与要处理的信息相关的问题:

- 存储哪种类型的信息?
- 数据上静态的、永远不变的,还是动态的、经常更新的?
- 怎样保存以及访问记录?
- 当前每个记录是否有唯一的标识特性?
- 怎样把记录中的信息分成域?

回答这些问题后就会熟悉所要使用的信息,了解人们又是怎样使用它们。这个过程会帮助理解数据库的逻辑设计。

一旦熟悉了数据,就会明白它们不同部分之间的关系,一些优秀的数据库设计人员用流程图说明这些关系。随着更深一步的了解就会明白怎样以及为什么数据的相关部分开在不同的表格中。

把数据库的逻辑设计得非常好是没有必要的,因为最后在创建存储数据的表格和列时所用到的物理设计也许与开始的逻辑设计不同。许多数据库设计人员认为,一旦逻辑设计用流程图表示出来,就不能改变。不要局限于某一种设计,必要时可以做些修改。

最后的物理设计要特别注意下面三个问题:

- 数据完整性。

- 创建数据的结构应该是用户容易访问的。
- 性能或速度。

物理数据库的设计应该保证表格中数据的一致性和完整性。虽然对数据库设计人员和管理人员来说这是最常见的要求,但是,用户的看法有点不同,对他们来说,速度和访问途径才上最重要的,还有,他们常希望快速浏览表格并且能自由访问数据。最后,对某一特定数据库来说成败的关键是数据更新速度和检索速度——如果数据库运行很慢,用户肯定会要求提高速度或者重新设计一个新系统。

下面三个标准是用来检验数据库设计者能力的:一致性/整体性、性能/速度和易访问而相互之间排他(exclusive)。满足某一种标准也许会削弱另外一种或两种标准,在最后计划中怎样权衡这些标准。

2.1.1 规范数据库

设计的数据库应该向规范数据库靠拢。规范数据库必须满足下面几个条件:

- 所有实体(表格)由属性(列/域)构成,而属性定义实体中行(记录)的性质(数据类型)。
- 表格的每一行表示一个事件或一项。
- 每行由主关键字唯一确定。
- 主关键字能由一列或多列组成。
- 主关键字不能是空值。

Customer Database	
Name	
Address	
City	
State	
ZipCode	
Phone	
OrderDate	
Ordered	
ShipVia	
TotalAmount	

图 2-1 非规范化数据库包含有重复行,其中有一些列应分解开来,分别存储在不同的表格中

规范数据库的一个特点是包括很多窄(narrow)表格,就是只有几列的表格,相反,有很多列的表格被认为是宽(wide)表格,通常是许多行含有重复数据。图 2-1 所示为一个非规范化数据库。

这个非规范化数据库有以下缺点,在刚接触这个概念的人员所设计的数据库中它们是常见的:

- 没有单列或多列主关键字。
- Name 域应分为 First Name 域和 Last Name 域,这样允许只查询顾客的 last name。
- 存在重复信息的列,如 Ordered 和 ShipVia,表格中不应该包含这种列,因为用户每订购某一项目,都要把所有的用户信息重复存储在一个新行中。想象一下,这样会有多少成千上万的行增加到表格中。
- TotalAmount 列是没有必要的,因为能由其他已有的列推导出来:

$$(((Quantity * Price) \text{ for each item ordered}) + ShippingCharges \text{ for entire order.})$$

设计成规范数据库有以下优点:减少重复数据,并且这样会减少存储要求;提高数据完整性的层次;减少重复数据能够保证不犯简单的错误,如输入错误的数据库。

如果输入的数据存储在几个窄表格中,那么 SQL Server 的每个存储页能够存储更多数据行。这样有利于提高查询和表格流览的速度,提高数据库的整体性能。

这个数据库的简化规范化的版本有点像图 2-2 所示:

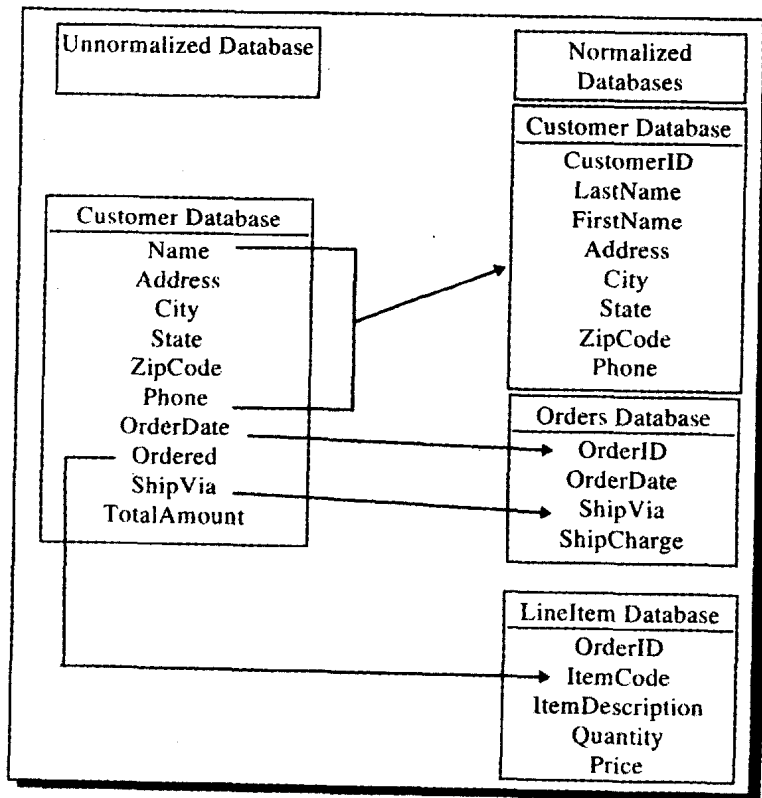


图 2-2 原来那个非规范数据库分解成三个规范数据库