

数据库技术系列

**DATABASE
PROFESSIONAL'S
LIBRARY**

SQL Server 2000

基 础

北京希望电子出版社 总策划
吴 豪 编 著

红 旗 出 版 社



北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn

数据库技术系列

DATABASE
PROFESSIONAL'S
LIBRARY

SQL Server 2000

基 础

北京希望电子出版社 总策划
吴 豪 编 著

红 旗 出 版 社



北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn

图书在版编目 (CIP) 数据

SQL Server 2000 基础 / 吴豪编著. —北京: 红旗出版社, 2005. 2

ISBN 7-5051-1078-0

I. S... II. 吴... III. 关系数据库—数据库管理系统, SQL Server 2000 IV. TP311.138

中国版本图书馆 CIP 数据核字 (2004) 第 113628 号

内 容 简 介

本书详细介绍了关系数据库 SQL Server 2000 的基础知识、语言基础以及利用 SQL Server 2000 进行数据库开发等内容。全书利用大量示例对重点内容进行讲解、分析。

本书由 13 章和两个附录组成。主要内容包括 SQL Server 2000 的历史及特点、SQL Server 数据库组成、实用工具、T-SQL 基础语句、联接表、创建和修改表、约束、视图、脚本和批处理、存储过程、用户自定义函数、触发器、SQL 游标、用 WMI 编写管理函数以及微软数据转换服务包对象库参考。

本书适合于 SQL Server 数据库设计领域的初学者和具有一定关系数据库经验的读者, 以及那些为微软认证考试而备战的读者, 可满足不同层次读者的切实需要。

本书的部分实例程序请从 www.b-xr.com 下载。

需要本书或需要得到技术支持的读者, 请与北京中关村 083 信箱 (邮编 100080) 发行部联系, 电话: 010-82702660, 82702658, 62978181 转 103 或 238, 62978181 (总机) 传真: 010-82702698, E-mail: tbd@bhp.com.cn。

系 列 名	数据库技术系列
书 名	SQL Server 2000 基础
编 著	吴豪
总 策 划	北京希望电子出版社
责 任 编 辑	李兴旺 雷锋
出 版 发 行	红旗出版社 北京希望电子出版社
地 址	红旗出版社 北京市沙滩北街 2 号 (100727) 电话: (010) 64037138 北京希望电子出版社 北京市海淀区上地 3 街 9 号金隅嘉华大厦 C 座 610
经 销	各地新华书店 软件连锁店
排 版	希望图书输出中心
印 刷	北京东升印刷厂
版 次 / 印 次	2005 年 2 月第 1 版 2005 年 2 月第 1 次印刷
开 本 / 印 张	787 毫米×1092 毫米 1/16 30.25 印张
字 数	702.5 千字
印 数	1~5000 册
书 号	ISBN 7-5051-1078-0
定 价	46.00 元

前 言

SQL Server 2000 在 SQL Server 6.5 和 SQL Server 7.0 的基础上有了飞跃的发展,增加了不少功能:XML 集成、分布式分区视图、索引视图、INSTEAD OF 触发器、用户自定义函数、两阶段数据抽取——用于数据转换服务 (DTS)、数据挖掘等。作为目前应用广泛的一种数据库,它不仅采用了合理的体系结构,而且利用了优越的可视化开发环境,加上其优秀的编辑界面,使其很多性能又远远超过了 Oracle 及其他数据库,从而为广大用户所青睐。目前有不少应用软件利用了 SQL Server 2000 作为后台存储数据库,如部分医院信息管理系统,银行、证券管理系统等。为了使越来越多的用户逐步深入掌握 SQL Server 2000 这种优秀的数据库,笔者编写了这套《数据库技术系列》丛书,与大家分享。

笔者是一位具有多年数据库研究开发经验的资深系统分析员和软件工程师,用过 SQL Server 和 Oracle 等大型数据库,开发过医院信息管理系统和远程医疗视频会议等大型管理系统。在学习和研究这些数据库的过程中,笔者深深地感受到学习过程的艰辛,为把推敲再三的心得殷实详尽地记录下来,为后学者铺出一条广阔的道路,笔者与同事一起系统地编写了这套《数据库技术系列》丛书。

这是一套 SQL Server 进阶的系列丛书,更是一组经验的结晶,可满足不同层次读者的切实需要。本书适合于 SQL Server 数据库设计领域的初学者和高级开发者,但重点面向开发者。这意味着,为了简明扼要,那些适合于数据库管理员而不适合于开发者的内容有时一带而过,甚至避而不谈,但对开发者有影响或在开发过程中需要考虑的管理因素,会在部分章节对相关内容进行简要阐述。对于初学者或具有一定关系数据库经验而不熟悉 SQL Server 的人来说,使用这套丛书可以迅速利用 SQL Server 进行数据库开发,设计出自己需要的数据库结构;对于具有一定基础的中级水平的开发人员,可以利用本书提供的内容,融会贯通、灵活运用,并利用本丛书提供的最新特点和使用方法,帮助自己提高开发和运用工具的水平;对于高级开发人员,这是一套不可多得的参考书,书中讨论的内容表明:本版软件包含了更多内容,SQL Server 将以一个全新的面孔展现出精彩的世界。

本丛书目的在于使读者阅读以后能在前进的道路上掌握更高级的技术,假设你已经是一个有经验的开发者(并不一定在数据库方面)。为了彻底理解本书,虽然需要懂得编程的基础知识,如变量、数据类型、程序化设计等,但并不要求以前就看过有关查询的内容。

从头到尾通读全书,就可以很好地掌握本书介绍的相关内容。本书侧重于帮助读者利用 SQL Server 2000 成功地开发应用程序,并为备战微软认证考试的人提供帮助,因此笔者会简要介绍一些力求更完美的内容。

这套书不是万能工具书,但它能够提供有关 SQL Server 2000 开发的许多信息,如 DTS、英文查询和联机分析处理 (OLAP) 等。

内容结构

本套丛书分为,《SQL Server 2000 基础》和《SQL Server 2000 提高》两本。(以下简称《基础》和《提高》)

为了使读者迅速利用 SQL Server 进行数据库开发,设计出自己需要的数据库结构,笔

者从数据库设计的角度出发,先介绍如何使用 SQL Server 2000 必备的工具,然后讨论如何进行数据库设计和开发应用,最后就高级数据库设计中可能遇到的各种问题做详细剖析。在《基础》中,笔者介绍了 SQL Server 2000 的历史背景、数据对象、常用工具、Transact-SQL、联接、创建和修改表、约束、视图、脚本和批处理、存储过程、用户自定义函数、触发器及游标等内容。在《提高》中,笔者先讨论了如何进行范式化设计,并就数据库设计中不得不讨论的索引结构、事务和锁展开探讨,再结合目前广为应用的 XML 技术,探讨了如何将 SQL Server 2000 与 XML 结合起来,并就如何利用 XML 技术进行高级查询、全文检索、英文查询做了深入介绍,同时就如何设计合理的分布式分区视图展开描述,最后就数据转换服务和数据库之间的数据复制等内容做了详细讨论。

下面简要介绍一下《SQL Server 2000 基础》中各章节的内容。

第 1 章比较了各种 SQL Server 版本和数据访问技术,并提供了 SQL Server 的一些历史背景信息。

第 2 章对 SQL Server 2000 的数据库对象 (database objects) 进行探讨。本章详细讨论了数据库的默认安装,包括存储过程、有效性规则和视图等其他常见的对象,并讨论 SQL Server 2000 中的数据类型。

第 3 章介绍开发者在 SQL Server 2000 中使用的工具,包括企业管理器 (Enterprise Manager)、性能监视器 (Performance Monitor) 和查询分析器 (Query Analyzer) 等工具。

第 4 章根据基本查询的需要,系统性地介绍了 Transact-SQL。

第 5 章基于前 4 章的内容,介绍了联接 (joins: INNER, OUTER, FULL 和 CROSS) 的概念。

第 6 章处理创建和修改表。

第 7 章介绍约束 (constraints)。解释主键 (primary keys) 和外部键 (foreign keys),并提供相关示例,还讨论了规则和默认值,并涵盖了 SQL Server 2000 的级连 (Cascade) 操作。

第 8 章重点讨论简单视图 (views),它是一种虚拟表。同时,笔者还在本章详细讨论了索引视图。

第 9 章讨论 SQL 脚本 (Scripts) 和批处理 (batches)。包括 USE 语句、GO 语句、SQL Server 变量用法、系统函数、OSQL (允许从命令提示符下运行) 以及讨论动态 SQL 语句 (Dynamic SQL)。

第 10 章介绍存储过程 (Stored Procedures),这是数据库开发的基本工具之一。介绍如何创建、修改和删除存储过程,并讨论扩展存储过程和系统存储过程 (Extended and System stored procedures) 以及异常处理,最后简要讨论调试。

第 11 章讨论 SQL Server 2000 中用户自定义函数 (UDF, User Defined Functions),并详细比较 UDF 和存储过程的差别。

第 12 章涵盖了触发器 (triggers) 的内容。介绍如何创建和修改触发器,触发器触发的时机,如何在视图上添加触发器,最后讨论 Instead of 触发器的概念。

第 13 章介绍如何通过游标 (Cursors) 驾驭一个记录集。虽然尽可能避免使用游标,有时候也无法避免,因为采用游标是惟一的解决办法。

运行环境

本书讨论的所有内容几乎都有例子，而且列出了相应的代码。为了能够更好地运行本书提供的示例，需要配备：

- SQL Server 2000
- Windows NT 4.0 和 SP5 补丁，最好使用 Windows 2000（Windows 9X 也可以工作，但在某些方面会失败）

约 定

本书采用许多不同文本方式和显示方式，用于帮助区分不同内容。其中：

- 对于需要特别注意和需要提示的内容，相应的文字使用楷体，并在段落前标以“注意：”或“提示：”字样。例如：

注意：Personal 版本虽然能与其他工具协调使用，但建议不要这样使用，因为微软不支持这种用法。

- 新代码或重要代码的格式设为：

```
SELECT CustomerID, ContactName, Phone  
FROM Customers
```

- 以前见过的代码或者无须讨论的代码格式为：

```
SELECT ProductName FROM Products
```

- 背景信息、附加信息和参考内容的字体格式如下：

20 世纪 80 年代，美国国家标准化组织（American National Standards Institute, ANSI）制订了 SQL 语言规范，ANSI-SQL 诞生了，这是 RDBMS 计算领域的关键时刻。

本书由吴豪、翟新海、宁义执笔，连平、董纪平、杨前凤、李太杰、吴安青、胡湖、蒲卫、郑辉、李沪建、焦志勇、薛涛、王明升、李国宝、李焕文、邹福荣、张宗然、吴俊、何萍、陈善、卫红权、宋斌参与了本书的编写工作。

由于笔者水平有限，加之时间紧迫，本书难免会有疏漏和不足之处，恳请广大读者提出宝贵意见。

编 者

目 录

第1章 SQL Server 2000 的历史及特点	1	3.2 客户和服务器网络实用工具	29
1.1 数据库发展简史	1	3.2.1 协议	30
1.1.1 数据库类型	1	3.2.2 客户端配置	33
1.1.2 关系数据库发展史	2	3.3 企业管理器	34
1.2 SQL Server 2000 简介	3	3.4 数据转换服务 (DTS)	35
1.2.1 版本说明	4	3.5 大容量拷贝程序	35
1.2.2 对硬件和操作系统的要求	6	3.6 MS DTC 管理控制台	35
1.3 建立数据库连接系统	6	3.7 SQL Server Profiler	36
1.3.1 单层结构 (主机系统)	7	3.8 查询分析器	36
1.3.2 2-层结构 (客户/服务器结构)	7	3.8.1 开始连接	36
1.3.3 3-层结构	9	3.8.2 建立连接	38
1.3.4 N-层结构	10	3.9 OSQL	42
1.4 数据存取模型	11	3.10 服务管理器	43
1.5 小结	12	3.11 sqlmaint 工具	44
第2章 SQL Server 数据库的组成	13	3.12 小结	44
2.1 数据库对象概述	13	第4章 T-SQL 基础语句	45
2.1.1 数据库对象	14	4.1 最基本的 SELECT 语句	45
2.1.2 事务日志	16	4.1.1 SELECT 语句和 FROM 从句	46
2.1.3 表——最基本的数据库对象	17	4.1.2 WHERE 从句	48
2.1.4 文件组	18	4.1.3 ORDER BY 从句	51
2.1.5 关系表	19	4.1.4 利用 GROUP BY 从句聚集数据	53
2.1.6 视图	19	4.1.5 利用 HAVING 从句在分组上 放置条件	60
2.1.7 存储过程	20	4.1.6 用 FOR XML 从句输出 XML	62
2.1.8 用户自定义函数	21	4.1.7 利用 OPTION 从句的提示信息	63
2.1.9 用户和角色	21	4.1.8 DISTINCT 和 ALL 谓词	63
2.1.10 规则	21	4.2 INSERT 语句	65
2.1.11 默认值	21	INSERT INTO...SELECT 语句	68
2.1.12 用户定义数据类型	22	4.3 用 UPDATE 语句更新数据	70
2.1.13 全文目录	22	4.4 DELETE 语句	72
2.2 SQL Server 的数据类型	22	4.5 小结	73
NULL 值	25	第5章 联接表	74
2.3 SQL Server 的对象标识符	26	5.1 联接	74
2.3.1 已经被命名的对象	26	5.2 内部联接	75
2.3.2 命名规则	26	INNER JOIN 类似于 WHERE 从句	78
2.4 小结	27	5.3 外部联接	82
第3章 实用工具	28	5.3.1 简单的 OUTER JOIN	82
3.1 联机丛书	28		

5.3.2	查找落孤记录或不匹配的记录	84
5.3.3	处理更复杂的 OUTER JOIN	87
5.4	完全联接	95
5.5	交叉联接	96
5.6	探讨联接的替代语法	97
5.6.1	INNER JOIN 的替代语法	97
5.6.2	OUTER JOIN 的替代语法	98
5.6.3	CROSS JOIN 的替代语法	98
5.7	联合	99
5.8	小结	103
第 6 章	创建和修改表	105
6.1	SQL Server 的对象名	105
6.1.1	所有者	105
6.1.2	数据库名	107
6.1.3	用服务器命名	107
6.1.4	回顾默认值	107
6.2	CREATE 语句	108
6.2.1	创建数据库	108
6.2.2	创建表	112
6.3	ALTER 语句	121
6.3.1	ALTER DATABASE 命令	121
6.3.2	更改表	123
6.4	DROP 语句	126
6.5	使用 GUI 工具	126
6.5.1	利用 EM 创建数据库	127
6.5.2	返回到编码: 利用 EM 创建脚本的基础	132
6.6	小结	137
第 7 章	约束	138
7.1	约束类型	139
7.1.1	域约束	139
7.1.2	实体约束	139
7.1.3	引用完整性约束	140
7.2	约束名	140
7.3	键约束	141
7.3.1	主键约束	141
7.3.2	外部键约束	144
7.3.3	惟一性约束	159
7.4	CHECK 约束	159

7.5	默认约束	160
7.5.1	在创建表的语句中定义默认约束	161
7.5.2	在现有表上添加默认约束	162
7.6	禁用约束	162
7.6.1	创建约束时忽略坏数据	162
7.6.2	临时禁用现有约束	166
7.7	规则和默认值	168
7.7.1	规则	168
7.7.2	默认值	169
7.7.3	决定哪张表和哪种数据类型使用给定规则或默认值	170
7.8	维护数据完整性的触发器	170
7.9	比较	170
7.10	小结	171
第 8 章	视图	172
8.1	简单视图	172
8.2	视图作为过滤器	178
8.3	更复杂的视图	180
8.3.1	DATEADD 和 CONVERT 函数	184
8.3.2	利用视图修改数据	184
8.4	利用 T-SQL 编辑视图	187
8.5	删除视图	188
8.6	在 EM 中创建和编辑视图	188
8.6.1	在 EM 中创建视图	188
8.6.2	在 EM 中编辑视图	191
8.7	审核	192
8.8	保护代码——加密视图	193
8.9	架构绑定	195
8.10	使用 VIEW_METADATA	195
8.11	索引视图	195
8.12	小结	198
第 9 章	脚本和批处理	199
9.1	编写脚本的基础	199
9.1.1	USE 语句	203
9.1.2	声明变量	203
9.1.3	使用 @@IDENTITY	207
9.1.4	使用 @@ROWCOUNT	215
9.2	批处理	216
9.2.1	另起一行	217

9.2.2	每个批处理都被独立地发送到服务器上.....	217
9.2.3	GO 不是一个 T-SQL 命令.....	220
9.2.4	批处理中的错误.....	220
9.2.5	何时使用批处理.....	221
9.3	OSQL.....	225
9.4	动态 SQL.....	226
9.5	EXEC 的限制.....	228
9.5	小结.....	231
第 10 章	存储过程.....	232
10.1	创建存储过程.....	232
10.1.1	基本语法.....	232
10.1.2	基本存储过程的例子.....	233
10.2	利用 ALTER 改变存储过程.....	233
10.3	删除存储过程.....	234
10.4	参数化.....	234
10.5	流控制语句.....	239
10.5.1	IF...ELSE 语句.....	239
10.5.2	CASE 语句.....	253
10.5.3	利用 WHILE 语句循环.....	256
10.5.4	WAITFOR 语句.....	257
10.6	存储过程的返回值.....	258
怎样使用 RETURN.....		258
10.7	异常处理.....	260
10.7.1	处理内嵌错误.....	260
10.7.2	在错误发生之前处理错误.....	268
10.7.3	手工提示错误.....	272
10.7.4	添加自己定制的错误消息.....	275
10.8	过程的用途.....	279
10.8.1	创建可调用的处理.....	279
10.8.2	因为安全性使用 spoc.....	281
10.8.3	存储过程和性能.....	282
10.9	扩展的存储过程 (XPs).....	284
10.9.1	xp_cmdshell.....	284
10.9.2	xp_msver.....	285
10.10	系统存储过程.....	287
10.11	递归.....	289
10.12	调试.....	291
10.12.1	为调试安装 SQL Server.....	292

12.12.2	启动调试器.....	292
10.12.3	调试器的组成部分.....	294
10.12.4	使用调试器.....	296
10.13	小结.....	300
第 11 章	用户自定义函数.....	301
11.1	UDF 的基本概念.....	301
11.2	返回标量值的 UDF.....	302
11.3	返回表的 UDF.....	305
11.4	理解确定论.....	310
11.5	“系统”函数.....	312
11.5.1	创建“系统”函数.....	312
11.5.2	在创建之后删除“系统”函数.....	314
11.5	小结.....	314
第 12 章	触发器.....	315
12.1	触发器定义 HECK.....	315
12.1.1	ON 从句.....	316
12.1.2	WITH ENCRYPTION 选项.....	316
12.1.3	FOR AFTER 从句与 INSTEAD OF 从句.....	317
12.1.4	WITH APPEND 选项.....	318
12.1.5	NOT FOR REPLICATION 选项.....	319
12.1.6	AS 关键词.....	319
12.2	使用触发器维护引用的完整性规则.....	319
12.2.1	使用触发器维护简单引用的完整性.....	319
12.2.2	使用触发器得到更灵活的引用完整性.....	325
12.3	使用触发器维护数据的完整性规则.....	336
12.3.1	处理其他表的请求.....	336
12.3.2	使用触发器检查被更新的中间数据.....	337
12.3.3	使用触发器定制错误信息.....	339
12.4	触发器的其他常见用途.....	340
12.4.1	更新摘要信息.....	340
12.4.2	为产生报表把数据插入到降低范式化的表中.....	340
12.4.3	设置条件标志.....	340
12.5	其他触发器问题.....	342
12.5.1	触发器可以嵌套.....	342

12.5.2	触发器可以递归.....	343
12.5.3	触发器不改变结构.....	343
12.5.4	不经删除就可以关闭触发器.....	343
12.5.5	触发器的触发顺序.....	344
12.6	INSTEAD OF 触发器.....	345
12.6.1	INSTEAD OF INSERT 触发器.....	345
12.6.2	INSTEAD OF UPDATE 触发器.....	351
12.6.3	INSTEAD OF DELETE 触发器.....	352
12.7	性能考虑.....	353
12.7.1	触发器是被动的而不是主动的.....	353
12.7.2	触发器与激活它们的进程之间 没有并发性问题.....	354
12.7.3	使用 UPDATE()函数和 UPDATED_COLUMNS()函数.....	354
12.7.4	保持触发器简洁.....	356
12.7.5	在选择索引时不要忘记触发器.....	356
12.7.6	7.0 以前联接触发器对表 数量的限制.....	356
12.7.7	尽量不要在触发器中回滚.....	357
12.8	删除触发器.....	357

12.9	调试触发器.....	357
12.10	小结.....	358
第 13 章	SQL 游标	359
13.1	游标的定义.....	359
13.2	游标的范围.....	360
13.3	游标类型和扩展声明语法.....	363
13.3.1	范围.....	363
13.3.2	滚动性.....	367
13.3.3	游标类型.....	370
13.3.4	并发性选项.....	383
13.3.5	检测游标类型转换: TYPE_WARNING.....	386
13.3.6	用于 SELECT.....	388
13.3.7	用于 UPDATE.....	388
13.4	用 FETCH 语句操作游标.....	388
13.5	在游标中改变数据.....	389
13.6	小结.....	392
附录 A	用 WMI 编写管理函数.....	393
附录 B	微软数据转换服务包对象库参考.....	443

第 1 章 SQL Server 2000 的历史及特点

本质上，数据库是一种用来管理信息的系统工具，它在现实中几乎无处不在。本章将简要讨论数据库的种类，比较它们的优缺点，并讨论数据库与现实生活的联系。随后，将针对 SQL Server 2000，讨论目前可用的各种版本的性能差异。

另外，将用点时间探讨数据库开发流程，并简要讨论系统结构方面的一些入门问题，这些问题将对数据库设计产生重大影响。

1.1 数据库发展简史

SQL Server 是一个关系数据库管理系统 (Relational Database Management System, RDBMS)。本节将回顾一下数据库的发展史，并了解过去使用的其他数据库，从而更好地理解数据库技术的来源和发展现状，熟悉数据库为什么会如此辉煌。

1.1.1 数据库类型

不要将数据库的概念局限于计算机术语，其意义远不止这些。数据库实际上是有组织 (Organized) 的数据集合。

在某种程度上，办公室里有序的文件抽屉就是一个数据库。实际上，数据库几乎贯穿了“文明”世界的所有历史阶段。

数据库与计算机的关系如此紧密是有原因的。关键在于计算机对大多数数据库而言不仅速度快，而且存储数据的效率也高。数据库被分成以下几类：

- 纸张形式 (Paper-based)。尽管目前人们不认为这种形式是数据库的一种，但它是当今数据库的最大组成部分。当今世界上，精心组织保存的成千上万的纸张实际上就是数据库的一种形式。
- 遗留主机 (Legacy Mainframe)。通常是 VSAM (Virtual Storage Access Method, 虚拟存储存取方法，在 IBM 的主机上最普遍) 数据库，不要低估了遗留主机的存在数量及其重要性。本书很多章节采用了主机系统解决方案，而没有采用客户/服务器或基于网页的模型。
- Dbase 和其他基于文件的数据库。通常，这种数据库包括早期的索引顺序存取方法 (Indexed Sequential Access Method, 缩写为 ISAM) 数据库。虽然 ISAM 通常使每张表使用不同的文件，但是其概念来源于数据存储和物理访问途径，而不是文件。目前，ISAM 数据库仍然具有相当广泛的应用范围，有时甚至用于一些新开发，包括一些小型数据库等。在 RDBMS 系统出现之前，这些小型数据库曾相当火爆，它们在独立的小数据库上应用很广，其缺点是同时访问这些数据库的用户数有限制。
- RDBMS 系统。该系统是大量数据的集合，而且具有更好的数据完整性。这种系统所做的工作不仅仅是存储和提取数据，而且特别关心数据的完整性。虽然 VSAM 和 ISAM 可以用于存储数据，但其本身并不控制数据的输入和输出，而是使用这些

数据库的应用程序负责实现数据完整性规则。如果 5 个程序都在访问数据，最好确信这些程序都能正常运行。尽管如此，大家在编写程序中仍然需要了解数据完整性规则，从而尽可能避免数据库产生出错消息。现有数据库会提示一部分出错消息，从而可以保证数据更加安全。

- 面向对象数据库 (Object-oriented Database)。这种数据库已经存在了一段时间了，但直到最近才得到飞速发展。这种数据库以完全不同的方式管理数据，而且到目前为止只找到了一些特殊的用途，这一点与文件管理系统相似。但是，它不是将文件存储在几张表中，而是存储在单一对象中，该对象具有可以维护其状态的属性。

ODBMS 系统通常提供了面向对象的概念，如继承和封装。

目前，RDBMS 系统处于数据库领域的主宰地位。这些系统设计的场合不是基于单表工作的概念，而是在数据相关的截然不同表上进行操作。这些系统以多种方式存储组合数据，尽可能避免重复存储，从而加快了事务处理速度。

1.1.2 关系数据库发展史

关系数据库的发展史可以追溯到 1960 年，IBM 公司的 E.F.Codd 第一次引入了关系数据库结构和结构化英文查询语言 (Structured English QUery Language, 缩写为 SQL) 的基本原理。这个概念实际上相当简单，就是减少重复数据降低成本，增加数据完整性，并改进当时其他数据库中相当普遍的问题。

关系数据库在很长时间内都没有发生实质性变化，直到 20 世纪 70 年代中后期，才出现了真正的关系数据库产品，如 Oracle 和 Sybase。这些系统不仅提供了一种新的管理数据库结构的方法，而且还可以在多平台上运行，可以实现在多系统之间共享数据。

20 世纪 80 年代，美国国家标准化组织 (American National Standards Institute, ANSI) 制订了 SQL 语言规范，ANSI-SQL 诞生了，这是 RDBMS 计算领域的关键时刻。这意味着不同厂家的产品将具有更好的兼容性，也意味着相互竞争的系统可以在一个 RDBMS 中嵌入多种专业技术。这使得有更多的开发者加入到 SQL 领域中来。ANSI 规范定义了各种系统之间的兼容性。

微软 SQL Server (在本书中简称为 SQL Server) 最初起源于 Sybase 的 SQL Server (简称 Sybase)。1989 年，微软与 Sybase 联合开发了一个用于 OS/2 版的 SQL Server。1993 年，他们将 4.2 版的 SQL Server 移植到 Windows NT 上。6.0 版推出后，这种合作关系解除了。6.5 及以后版本的 SQL Server 已经成为微软的专有产品。在 7.0 版中，微软对该产品进行了彻底重写，并取得了高度成功，从而使该产品成为第一个可以用于 Windows 95/98 的产品 (目前，SQL Server 中几乎没有保留 Sybase 的原有代码)，最终导致了 SQL Server 2000 的问世。

虽然 SQL Server 的 4.2 版与 2000 版有着明显的相似性，但它们在实现和性能支持上存在着本质上的区别。6.0 版增加了诸多细节，如游标支持等。6.5 版增加了分布式事务、复制和 ANSI 兼容性。

1.2 SQL Server 2000 简介

SQL Server 2000 的版本主要有企业版 (Enterprise Edition)、标准版 (Standard Edition)、个人版 (Personal Edition)、开发版 (Developer Edition) 等。

全套 SQL Server 2000 由表 1-1 组成。

不同版本的 SQL Server 2000 之间的差异有:

- 对称性多重处理 (Symmetric Multiprocessing, 缩写为 SMP)。在 SQL Server 2000 的不同版本中, 增加了支持 SMP 的功能 (尽管 Windows 98 和 Windows NT4 的工作站不支持 SMP)。如果在 NT 服务器或企业服务器上安装标准版, 则最多可以支持 4 个处理器, 如果在 Windows 2000 的数据中心服务器上安装企业版, 则可以支持多达 32 个处理器。

SMP 将服务器工作量均衡分配在多个处理器上, 也就是说, 尽量不在一个 CPU 上处理, 而是在多个 CPU 之间均衡分配任务。

- 支持聚类 (Clustering Support, 只有企业版和开发版支持)。允许通过服务器均衡加载聚类, 并支持自动接管功能 (当一个服务器死机后, 另一个服务器自动接管这台服务器留下的任务)。当前, 对所有操作系统而言, 只能在两个服务器之间支持聚类, 除了 Windows NT 企业版以外, Windows 2000 企业版和 Windows 数据中心版最多能支持 4 个聚类服务器。

表 1-1 全套 SQL Server 2000

系统/子系统	描述	版本
SQL Server 2000 (主 RDBMS)	这是系统的“核心”, 是其他组件工作的必备内容, 是一个非常具有生命力的关系数据库系统。除了桌面工程版 (Desktop Engine) 只有一个主 RDBMS 外, 它还包括许多服务和用途, 例如 SQL Server 日程表、分布式事务协同管理器 (Distributed Transaction Coordinator, 缩写为 DTC)、SQL Server Profiler 和企业管理器 (Enterprise Manager)	桌面工程版
		个人版
		标准版
		开发版
		企业版
全文检索 (Full-text Search)	这是主安装的可选部分。它不是默认安装的一部分。全文检索 (Full-Text Search) 提供了更具生命力的字查询功能。如果用过 Internet 搜索引擎, 那么全文检索将成为一个强有力的工具。它既能在大文本体内定位小短语, 又能知道原形和过去式相似, 还能区分动词的过去式, 但 SQL Server 的个人版 (Windows 95/8) 不能使用这种功能	个人版 (除 Windows 95/98)
		标准版
		开发版
		企业版
英文查询 (English Query, 又叫 EQ)	英文查询允许用户以普通英语提问或发出命令, 然后将这些命令翻译成可以被 SQL Server 使用的查询。这是一个伟大的工具, 但要记住, 该工具需要独立进行安装	个人版
		标准版
		开发版
		企业版

(续表)

系统/子系统	描述	版本
分析服务 (Analysis Services)	<p>尽管这个工具不是安装主体的一部分,但是它可以带来额外功能。它由在线分析处理(On line Analytical Processing, 又叫 OLAP)、数据仓库和数据提炼工具组成。</p> <p>尽管只有企业版和开发版具有完整的分析服务功能,但右边所列的版本都支持 OLAP。标准版和个人版只包含主要功能,它由分析服务、定制积累(custom rollups)、数据挖掘和终止进程(结束用户在数据上的操作)组成。另外,分析服务只能安装在 Windows NT/2000 上</p>	个人版
		标准版
		开发版
		企业版
复制 (Replication)	<p>这种功能允许将数据复制到另一个 SQL Server 实例中,通常用作另一个服务器的恢复策略,也可以减少把数据复制到另一个远程服务器上,从而减少数据传输量</p>	桌面版
		个人版
		标准版
		开发版
		企业版
数据转换服务 (Data Transformation Services)	<p>在 SQL Server 2000 中,数据转换服务(DTS)得到极大扩展。无论是在数据库中还是在转移信息输入或输出上,包括定制任务和工作流在内的很多功能都可用数据转换服务。DTS 是一个非常有用的产品,它使公司与公司之间不必利用编程方法(甚至采用大容量拷贝程序)就能进行数据转换</p>	个人版
		标准版
		开发版
		企业版

1.2.1 版本说明

1. Windows CE 版

这个版本将用于 Windows CE 设备,其功能完全限制在给定范围内,显然,这些设备的容量极其有限。目前,使用 Windows CE 和 SQL Server 的应用程序非常有限,实际只可能在更昂贵的 CE 产品上拥有更有用的应用程序。

2. 桌面工程版

SQL Server 2000 的桌面工程版在 SQL Server 7.0 中以 Microsoft Data Engine (MSDE) 著称。不要将它与 SQL Server 7.0 中的桌面版混淆为同一版本,因为它们确实不是一回事。SQL Server 7.0 桌面版目前是 SQL Server 2000 的个人版。桌面工程版仅由主 RDBMS 组成,而且没有管理工具,甚至连企业管理器和查询分析器都没有。该版本不仅小,而且可以自由分发。

3. Personal 版

该版本即 SQL Server 7.0 的桌面版的重命名。创建该版本有两个目的:提供一个比 Access 数据库更具生命力的桌面数据库解决方案(甚至在 Windows 95/98 上);提供一个可

以在“独立”状态下使用的 SQL Server。后者是其主要目的。实践证明，在远程环境下采用这种版本很流行，如果通过网络连接，它们还可以拥有自己的客户端数据库，并可以使用复制实现同步。

当需要一个小的独立数据库或者需要与中心数据源断开连接，但又想得到其中一些数据时，个人版就是最好选择。在 Windows NT/2000 上，可以在一个小服务器上安装个人版，这种最新配置甚至可以支持多处理器。记住，即使是在多处理进程激活的状况下，该系统也不支持并行查询（即同时运行相同查询的不同部分）。

注意：个人版本虽然能与其他工具协调使用，但建议不要这样使用，因为微软不支持这种用法。

不过，个人版不能销售，只有在购买标准版或企业版时才能作为赠品得到。该版本是客户访问授权（Client Access License，又叫 CAL）和客户端软件的一部分。如果在企业版的 Windows 2000 或 Windows 98 上运行个人版，则用处非常大，但需要 GUI 管理工具。

4. 标准版

标准版是 SQL Server 的主流版本，而且是大多数 SQL Server 用户要安装的版本。标准版支持多进程处理，可以支持多达 4 个 CPU 和 2GB 的 RAM。为了安装标准版实例，客户需要为每个标准版的实例购买独立许可证。

5. 开发版

在后面将会看到，系统的默认安装是开发版。企业版和标准版应视为在服务器上的解决方案。在开发版上，可以开发和测试应用程序。它具有企业版的所有特性，因此，一旦一个解决方案在开发版上开发成功，则移到产品环境上将不会有任何问题。

这个版本与企业版之间的惟一差别是：开发版只能用作开发环境。

6. 企业版

运行企业版，必须安装 Windows NT 企业版、Windows 2000 Advanced Server 或 Windows 2000 Datacenter Server。SQL Server 企业版支持多达 32 个 CPU 的多进程处理，而且支持聚类（两个独立服务器之间提供自动接管功能并分担工作量），允许 HTTP 访问 OLAP 多维集（cubes）。

采用企业版通常是一个简单选择，可以根据是否需要企业版特性、价值和许可（企业版每个处理器许可权的价钱是标准版的 4 倍）等因素而自由决定。如果需要聚类，则需要企业版。企业版的特征包括：

- 聚类
- 分布式分区视图
- 索引视图
- 分区多维集
- 支持超过 4GB 的 RAM
- 日志传输（一种自动接管策略）
- 支持多于 4 个 CPU

另外, 还有一些模糊特性只能在企业版中支持。但是, 很少有人不需要上面的这些特性, 从根本上说, 如果需要这些特性之一, 就要采用企业版。

1.2.2 对硬件和操作系统的要求

SQL Server 最低硬件要求如下:

- PentiumII 500 以上的服务器。
- 至少 64MB 内存, 如果使用企业版, 建议使用 128MB 内存。
- 对桌面工程版和个人版, 也可以仅使用 32M 内存, 但对于 Windows 2000 系统, 则需要 64M 内存。
- 95MB (最小安装) ~270MB (完全安装) 的硬盘空余空间。
如果想安装分析服务管理器, 则再需要 50MB~130MB 的空间; 如果需要安装英文查询器, 则另外再加 80MB 的空间。
桌面工程版仅需要 44MB 的空间。

• 企业版或标准版可在以下操作系统中运行:

- (1) Windows NT Server 4.0 版和 SP5 或以上的补丁。
- (2) Windows NT Server 4.0 企业版和 SP5 或以上的补丁。
- (3) Windows 2000 Server。
- (4) Windows Advanced Server。
- (5) 微软 Windows 2000 Datacenter Server。

开发版可以在上面所列的企业版和标准版的操作系统下运行, 如 Windows 2000 专业版或 Windows NT 工作站 4.0 版和 SP5 或以上的补丁。

个人版和桌面工程版可以在上面所列的操作系统下运行, 也可以在 Windows 98 (如果计算机没有网卡, 则需要第 2 版下)、Windows ME、Windows 2000 专业版和 Windows NT4.0 工作站和 SP5 或以上的补丁上运行。

- VGA 显示器, 并在 800×600 像素模式下 (一些图形工具的需要) 工作。
- IE 5.0 以上的浏览器 (包括 IE 5.0)。

实际上, 需要采用比推荐的配置更好的机器。即使是在独立开发服务器上, 也建议采用至少 128MB 的内存和 Pentium II 500 或者更好的处理器。对于应用系统, 至少要配备 256MB 的内存, 最好是 512MB~2GB。

1.3 建立数据库连接系统

近几年产生了一些系统模型, 通常根据事情的处理方式而将这些模型分为 3 组服务 (services)。

- 用户服务 (User Services)。通常包括描绘用户接口 (the User Interface, 缩写为 UI,) 和基本格式及字段规则等方面的内容。用户服务需要处理的一个实例是日期格式, 包括让用户知道给定字段是日期字段, 而且在输入数字到该字段时必需预先校验并确定该数字为日期型。用户服务是一种表达式, 在每个字段中至少有一种预定的数据类型。

- 业务服务 (Business Services)。这部分就是众所周知的业务规则。在 3-层结构或 n-层结构中, 业务服务对象可以驻留在自身服务器上, 也可以分为多个服务器, 或者以最小安装的方式与数据服务 (Data Services) 共享服务器。
- 数据服务 (Data Services)。该服务完全用于数据存储和提取。数据服务通常以数据完整性规则著称 (如出库数量不能为负数)。这就是 SQL Server 的优势。

下面介绍一些传统的结构。

1.3.1 单层结构 (主机系统)

这是早期的主机和小型机模型。这种系统实质上没有桌面逻辑, 而只有一个无声的终端。所有通过电缆发送到终端的内容就是显示器所显示的信息, 当然包括显示的数据。其优缺点如表 1-2 所示。

表 1-2 单层结构的优点和缺点

优点	缺点
为了在网络上得到快速响应, 所需带宽较少, 在广域网或国际互联网带宽相当昂贵的情况下非常有用	硬件非常昂贵
相当可靠, 主机可以持续工作多年而不停机	本质上是典型的独占, 因此很难与其他系统共享信息
开发新软件十分轻松。只要在主机系统上安装, 就可以使每一个用户拥有新版软件, 不需对逐个机器进行更新	可用的“复活”软件包非常有限。由于潜在客户数量较少, 所以这些软件包的价格非常昂贵

1.3.2 2-层结构 (客户/服务器结构)

2-层结构, 或者叫客户/服务器系统, 流行于 20 世纪 90 年代早期。实际上这种结构有两种类型: 客户中心和服务器中心。

1. 客户中心形式

基于 PC 机便宜的角度 (这是大多数人使用客户/服务器开发的原因所在) 和在分配计算任务时尽可能得到最大效率的观念, 一般应用客户/服务器的客户中心形式。因此, 这种模式下通常只有数据业务部分在服务器上处理, 正常事务和 UI 端的事情都在客户端执行, 从而确保各个计算机公平分担所有任务 (至少, 这就是客户中心形式的思想)。其原理如图 1.1 所示。

以客户端为中心的客户/服务器结构的最大问题是带宽 (目前仍然存在)。如果所有业务逻辑都发生在客户端, 则客户端与服务器端将存在大量交互 (网络发射和接收)。大量的数据经常被发送到客户端, 最终导致网络阻塞, 从而降低了个人读取和写入的速度。其优缺点如表 1-3 所示。