

“十五”国家重点电子出版物规划项目·计算机知识普及和软件开发系列
数据库开发与应用丛书(1)

陆昌辉 吴晓华 编写

SQL Server 2000 核心技术揭秘



中航出版社



北京希望电子出版社

“十五”国家重点电子出版物规划项目·计算机知识普及和软件开发系列
数据库开发与应用丛书(1)

陆昌辉 吴晓华 编写

SQL Server 2000 核心技术揭秘



中航出版社



北京希望电子出版社

内 容 简 介

本套书由《SQL Server 2000 核心技术揭秘》和《SQL Server 2000 开发人员指南》组成。

《SQL Server 2000 核心技术揭秘》由 19 章组成。第 1 章到第 3 章内容包括 SQL Server 2000 的新特性、整体构架及其兼容性；第 4 章到第 6 章分别介绍了故障转移群集、数据转换技术、数据的备份与恢复；第 7 章到第 12 章讲述了服务器以及客户端的管理，数据的复制，如何监视服务器的性能和服务，Web 助手向导的使用，SQL Server 安全性与用户管理，以及数据仓库的创建和使用等内容；第 13 章的内容是数据仓库的设计创建，使用和维护；SQL Server 2000 数据库管理部分包括第 14 章到第 19 章，其中内容包括数据库设计与管理，SQL Server 的表与视图管理技术，索引管理技术，存储过程管理技术，触发器管理技术，数据库性能优化等。

本书内容丰富、全面，结构合理，重点突出，讲解深入，实用性强，适合于用 SQL Server 2000 进行数据库开发、维护与管理的广大从业人员自学指导书；高等院校相关专业师生教学、自学参考用书。

需要本书或需要得到技术支持的读者，请与北京海淀 083 信箱北京希望电子出版社（邮编 100080）联系。网址：www.bhp.com.cn，E-mail：lwm@bhp.com.cn。电话：010-62562329，62541992，62637101，62637102，62633308，62633309（图书发行，技术支持）；010-62613322-215（门市）；010-62521798（编辑部）。传真：010-62579874。

宇 航 出 版 社
北京希望电子出版社 出版发行
北京市和平里滨河路 1 号（100013）
北京中关村大街 26 号（100080）
发行地址：北京阜成路 8 号（100830）
北京中关村大街 26 号（100080）
北京广益印刷有限公司印刷
新华书店经销

ISBN 7-80034-959-4

2002 年 6 月第 1 版 2002 年 6 月第 1 次印刷

开本：787×1092 1/16 印张：24.75 字数：560 千字

全套定价：78 元（共 2 本）本册定价：39.00 元

前 言

在当今的 IT 行业，变幻莫测，竞争日益激烈，这使得数据库开发及编程人员迫切需要一种有利的工具帮助他们在竞争中处于有利地位。SQL Server 2000 的诞生，使得数据库与操作系统的集成，对分布计算环境的适应，Web 数据发布、收集和大型异构数据库的互连，对决策支持和在线事务的支持等变得那么的方便。

SQL Server 2000 是为创建可伸缩电子商务、在线商务和数据仓储解决方案而设计的真正意义上的关系型数据库管理与分析系统。它完全支持集成的 XML 环境、分析服务中的新的数据开采特性，以及元数据服务中增强的知识库技术，在稳定性、安全性和易操作性等方面都得到了加强。

为了帮助广大开发人员深入了解 SQL Server 2000，并为他们在进行项目开发时提供参考，我们编写了这本 SQL Server 2000 高级管理员指南，对 Microsoft SQL Server 2000 的高级管理功能进行了介绍。

本书由 19 章组成。第 1 章到第 3 章内容包括 SQL Server 2000 的新特性、整体构架及其兼容性；第 4 章到第 6 章分别介绍了故障转移群集、数据转换技术、数据的备份与恢复；第 7 章到第 12 章讲述了服务器以及客户端的管理，数据的复制，如何监视服务器的性能和服务，Web 助手向导的使用，SQL Server 安全性与用户管理，以及数据仓库的创建和使用等内容；第 13 章的内容是数据仓库的设计创建，使用和维护；SQL Server 2000 数据库管理部分包括第 14 章到第 19 章，其中内容包括数据库设计与管理，SQL Server 的表与视图管理技术，索引管理技术，存储过程管理技术，触发器管理技术，数据库性能优化等。

本书内容翔实，结构合理，全面的介绍了 SQL Server 2000 高级管理的内容，适合于广大中、高级的数据库系统管理人员、数据库应用程序开发人员使用，也可供数据库的一般用户和爱好者阅读参考。

本书由吴晓华策划，陆昌辉主编，杨国锴、刘炼、蔡成毅、高平立、刘良江、汪小栋、陆昌辉、周敏龙、吴晓华、温靖邦、唐九阳等编写。因时间和水平有限，一定还有许多不周到和不准确之处，恳请广大读者提出批评和建议。

亮点工作室

目 录

第 1 章 SQL Server 2000 的新特色 1	
1.1 新的数据类型 1	
1.2 用户自定义函数 2	
1.3 增强的索引 3	
1.4 SQL Server 2000 和 XML 4	
1.5 全文检索功能 6	
1.6 联合服务器 7	
1.6.1 集群服务器与联合服务器的区别 . . . 7	
1.6.2 使用联合服务器 8	
第 2 章 SQL Server 2000 构架 10	
2.1 SQL Server 2000 构架基础知识 10	
2.1.1 SQL Server 构架概述 10	
2.1.2 SQL Server 2000 组件概述 10	
2.2 SQL Server 2000 的特性 11	
2.2.1 与 Internet 集成 11	
2.2.2 可伸缩性和可用性 11	
2.2.3 企业级数据库功能 12	
2.2.4 易于安装、部署和使用 12	
2.2.5 数据仓库 12	
2.3 关系数据库组件 12	
2.3.1 数据库 12	
2.3.2 关系数据库 12	
2.3.3 可伸缩性 12	
2.3.4 结构化查询语言 13	
2.3.5 可扩展标记语言 13	
2.4 数据库构架 13	
2.5 管理构架 14	
2.5.1 DDL 和存储过程 14	
2.5.2 SQL 分布式管理框架 14	
2.5.3 图形工具 14	
2.5.4 自动管理构架 17	
2.5.5 备份/恢复构架 18	
2.5.6 数据导入/导出构架 18	
2.5.7 数据完整性验证 18	
2.6 复制构架 19	
2.7 数据仓库和联机分析处理 19	
2.7.1 将 OLTP 数据转换到 OLAP 数据仓库 20	
2.7.2 SQL Server 2000 数据仓库和 OLAP 组件 20	
2.8 应用程序开发框架 21	
2.8.1 应用程序接口 (API) 或统一 资源定位器 (URL) 21	
2.8.2 数据库语言 21	
第 3 章 SQL Server 2000 的兼容性 . . . 22	
3.1 SQL Server 的向后兼容 22	
3.1.1 客户端网络实用工具和命名实例 . . 22	
3.1.2 升级 SQL Server 6.5 客户端软件 . . 23	
3.1.3 身份验证模式 23	
3.1.4 恢复模型和数据库选项 23	
3.1.5 SQL 查询分析器中的 默认连接选项设置 23	
3.1.6 来自早期版本可视化数据库 设计工具的数据库关系图 24	
3.1.7 数据转换服务 24	
3.1.8 English Query 和 SQL Server 7.0 OLAP 服务 25	
3.2 SQL Server 2000 的不兼容问题 25	
3.2.1 多服务器作业和命名实例 25	
3.2.2 在远程表上所执行操作的 ROWCOUNT 设置 25	
3.2.3 服务器配置选项 25	
3.2.4 保留关键字 26	
3.2.5 SQL 事件探查器扩展存储过程 . . . 26	

3.2.6 SQL-DMO 中的扩展对象	26	5.3 bcp 技术	46
3.2.7 SQL-SCM	26	5.3.1 bcp 的语法	46
3.2.8 bcp 实用工具	26	5.3.2 bcp 的使用	49
第 4 章 故障转移群集	27	5.4 数据转换技术实例	50
4.1 故障转移群集概述	27	5.4.1 实现从 Access 到 SQL Server	
4.2 故障转移群集支持	27	的转移	50
4.3 处理故障转移群集	28	5.4.2 实现从 Sybase Anywhere 到 SQL	
4.3.1 安装故障转移群集前的准备工作	28	Server 的转移	50
4.3.2 安装故障转移群集	28	第 6 章 备份与恢复数据	53
4.3.3 故障转移群集的相关性	29	6.1 理解备份与恢复	53
4.4 创建故障转移群集	30	6.2 使用恢复模型	54
4.4.1 虚拟服务器的元素	30	6.2.1 简单恢复	54
4.4.2 命名虚拟服务器	30	6.2.2 完全恢复	54
4.4.3 创建故障转移群集前的准备	30	6.2.3 大容量记录日志恢复	55
4.4.4 创建故障转移群集	31	6.3 备份与恢复操作	56
4.5 维护故障转移群集	32	6.3.1 数据库备份	56
4.5.1 向现有虚拟服务器中添加节点	32	6.3.2 差异数据库备份	64
4.5.2 从现有故障转移群集中删除节点	32	6.3.3 事务日志备份	67
4.5.3 将 SQLServer 群集实例作为		6.4 管理备份	70
独立实例运行	33	6.4.1 备份介质	71
4.5.4 从故障转移群集失败中恢复	33	6.4.2 查看有关备份的信息	71
4.5.5 更改服务帐户	34	6.4.3 验证备份	71
4.6 在故障转移群集中使用 SQL Server		6.5 了解检查点机制	72
工具	34	第 7 章 管理服务器	73
4.6.1 全文查询	34	7.1 注册服务器	73
4.6.2 SQL Server 企业管理器	34	7.2 设定 sa 密码	73
4.6.3 服务管理器	35	7.3 设置网络支持	74
4.6.4 SQL 事件探查器	35	7.3.1 Net-Library 和网络协议	74
4.6.5 SQL 查询分析器	35	7.3.2 网络实用工具	74
4.6.6 SQL 邮件	35	7.3.3 通过 Internet 连接到 SQL Server	75
第 5 章 数据转换技术	36	7.4 配置链接服务器	76
5.1 数据转换技术简介	36	7.4.1 链接服务器组件	76
5.2 数据转换服务技术	37	7.4.2 链接服务器详细信息	76
5.2.1 DTS 数据导入/导出向导	37	7.4.3 管理链接服务器的定义	76
5.2.2 DTS 包设计器	44	7.4.4 建立链接服务器的安全性	78
5.2.3 DTS Transfer Manager	44	7.5 配置远程服务器	79

7.6 使用备用服务器	80	9.6.3 使用请求订阅向导	108
7.6.1 创建备份服务器	80	9.7 在异构环境中复制数据的技术	110
7.6.2 使备用服务器联机	81	第 10 章 监视服务器性能和活动	112
7.7 日志传送	81	10.1 评估服务器性能	112
7.7.1 日志传送模型	81	10.1.1 响应时间与吞吐量	112
7.7.2 使用数据库维护计划向导配置 日志传送	81	10.1.2 影响性能的因素	112
7.8 设置配置选项	88	10.1.3 建立性能基线	112
7.9 管理 SQL Server 消息	89	10.1.4 识别瓶颈	113
第 8 章 管理客户端	90	10.1.5 确定用户活动	113
8.1 客户端 Net-Library 和网络协议	90	10.2 使用系统监视器进行监视	113
8.1.1 多网络协议支持	90	10.3 正确使用 SQL 事件探查器	114
8.1.2 Net-Library 的安装和默认设置	91	10.3.1 使用 SQL 事件探查器的方案	114
8.2 配置客户端之前的准备	91	10.3.2 使用 SQL 事件探查器 监控事件分类	115
8.3 配置客户端网络连接	91	10.3.3 创建并管理跟踪和模板	116
8.4 配置客户端 Net-Library	91	10.4 使用对象和计数器	119
8.4.1 查看网络库信息	92	10.4.1 监视磁盘活动	120
8.4.2 更改默认的客户端配置	92	10.4.2 监视 CPU 的使用	120
8.5 配置 ODBC 数据源	93	10.4.3 监视内存的使用	121
8.5.1 使用 ODBC 数据源管理器	93	10.5 用 SQL Server 企业管理器进行监视	121
8.5.2 使用 ODBC API 函数	94	10.5.1 显示当前服务器的活动内容	121
8.5.3 添加或删除 ODBC 数据源	94	10.5.2 查看有关对象锁的信息	122
8.6 配置 OLE DB 客户端	96	10.5.3 给当前连接的用户发送消息	122
8.7 DB-Library 选项	97	10.5.4 终止进程	123
第 9 章 数据复制	98	10.6 监视错误日志	123
9.1 数据复制的概念和特点	98	10.6.1 查看 SQL Server 错误日志	123
9.2 复制模型	99	10.6.2 查看 Windows 应用程序日志	124
9.3 复制类型	100	10.7 用 SNMP 进行监视	124
9.3.1 快照复制	100	10.7.1 在 SQL Server 上启用 SNMP 支持	124
9.3.2 事务复制	101	10.7.2 启用 SQL Server MIB	124
9.3.3 合并复制	102	第 11 章 使用 Web 助手向导	126
9.4 规划复制	104	11.1 配置 Web 助手向导	126
9.5 建立复制	104	11.1.1 设置权限	126
9.6 使用向导	105	11.1.2 选择要发布的数据库	126
9.6.1 使用配置发布和分发向导	105	11.1.3 创建查询	127
9.6.2 使用创建发布向导	106		

11.2 使用 Web 助手向导	127	13.1.1 从数据库到数据仓库	156
11.3 在 Web 站点上查看结果.....	131	13.1.2 数据的仓库化过程	159
11.4 使用 Web 助手向导接收查询结果.....	132	13.1.3 数据仓库的体系结构	160
11.5 使用 Web 助手向导的一个问题.....	132	13.1.4 SQL Server 2000 数据仓库构件.....	162
第 12 章 SQL Server 安全性与用户管理 . 134		13.2 Microsoft 数据仓库框架.....	166
12.1 SQL Server 安全性管理概述.....	134	13.3 OLE DB:信息交换标准.....	167
12.2 安全认证方式	134	13.4 设计和创建数据仓库	169
12.2.1 Windows 身份验证	135	13.4.1 数据仓库系统设计方法概述.....	169
12.2.2 SQL Server 身份验证.....	136	13.4.2 使用 SQL Server 2000 工具 进行维度建模	172
12.2.3 其他安全认证方式.....	136	13.4.3 SQL Server 2000 的事实数据表.....	173
12.3 创建和管理用户帐户.....	138	13.4.4 SQL Server 2000 的聚合表.....	174
12.3.1 创建和管理 Windows 用户或组.....	138	13.4.5 SQL Server 2000 的维度表.....	174
12.3.2 创建和管理 SQL Server 登录用户	140	13.4.6 使用 SQL Server 2000 创建 数据仓库索引	176
12.3.3 特定的数据库用户	141	13.5 使用数据仓库	176
12.3.4 查看用户登录信息.....	142	13.5.1 SQL 查询.....	177
12.3.5 修改用户登录信息.....	142	13.5.2 OLAP 和数据挖掘.....	177
12.3.6 数据库角色管理.....	144	13.5.3 English Query	178
12.4 审核 SQL Server 活动.....	146	13.5.4 Microsoft Office 2000	178
12.4.1 SQL 事件探查器	146	13.5.5 Web 访问和创建报表	179
12.4.2 审核 SQL Server 活动.....	147	13.5.6 脱机 OLAP 多维数据集	179
12.5 权限管理	148	13.5.7 第三方应用程序	179
12.5.1 对 Transant-SQL 语句可以赋予 的权限	148	13.5.8 自定义应用程序	180
12.5.2 对数据库对象进行操作的权限.....	148	13.6 维护数据仓库	180
12.5.3 固定角色的权限.....	148	13.6.1 更新数据仓库数据	180
12.5.4 权限和 ALTER VIEW.....	151	13.6.2 管理数据仓库	181
12.6 其他安全性管理	152	13.6.3 优化数据仓库性能	182
12.6.1 使用视图作为安全机制.....	152	第 14 章 SQL Server 数据库设计与管理.. 184	
12.6.2 使用存储过程作为安全机制.....	153	14.1 数据库基本概念	184
12.6.3 使用加密方法	153	14.1.1 数据库	184
12.6.4 在网络上公布 SQL Server.....	154	14.1.2 事务日志	185
12.6.5 编写 Internet Explorer 中数据 访问控件的脚本.....	154	14.1.3 文件和文件组	188
第 13 章 数据仓库技术 156		14.2 数据库设计	191
13.1 数据仓库概述	156	14.2.1 SQL Server 2000 不再 支持段操作.....	191

14.2.2	规划数据库	192	15.3.6	使用约束的例子	242
14.2.3	数据的规范化	193	15.4	视图的概念	244
14.2.4	数据的完整性	194	15.5	视图的创建	245
14.2.5	数据库性能	195	15.5.1	建立视图	245
14.2.6	数据库维护	196	15.5.2	建立分区视图	248
14.3	创建数据库和日志文件	196	15.5.3	建立索引视图	251
14.3.1	数据库创建	196	15.6	视图的应用	254
14.3.2	在原始分区上创建数据库	202	15.6.1	使用视图检索数据	254
14.3.3	设置数据库选项	203	15.6.2	通过视图修改基表数据	255
14.3.4	创建数据库的限制	210	15.7	修改和删除视图	256
14.4	修改数据库	210	第 16 章 SQL Server 索引管理技术... 257		
14.4.1	改变数据库的大小	212	16.1	索引的工作机理	257
14.4.2	改变数据库文件的数目	215	16.1.1	数据的存储和访问	257
14.4.3	文件组管理	216	16.1.2	链接页面的更新	258
第 15 章 SQL Server 的表与视图			16.1.3	决定是否创建索引	259
管理技术			16.1.4	聚集索引	259
15.1	表或视图中的数据类型	219	16.1.5	非聚集索引	262
15.1.1	数字数据	219	16.2	索引的创建	266
15.1.2	字符数据	220	16.2.1	一般索引的创建	266
15.1.3	二进制数据	220	16.2.2	SET 选项的设置	275
15.1.4	日期和时间数据	220	16.2.3	索引的增强	278
15.1.5	Unicode 数据	220	16.3	查看索引信息	281
15.1.6	货币数据	221	16.4	删除索引	283
15.1.7	特殊数据	221	第 17 章 SQL Server 存储过程		
15.1.8	用户定义的数据类型	222	管理技术		285
15.2	表管理技术	223	17.1	存储过程概述	285
15.2.1	选择数据库	223	17.2	创建存储过程	286
15.2.2	表的创建	223	17.3	存储过程中参数的使用	290
15.2.3	表的修改	229	17.3.1	指定参数的名称	291
15.2.4	特殊的表——临时表	232	17.3.2	指定参数的数据类型	291
15.3	约束的使用	234	17.3.3	指定参数的方向	292
15.3.1	PRIMARY KEY 约束	234	17.3.4	指定参数的默认值	293
15.3.2	UNIQUE 约束	236	17.4	存储过程编程	293
15.3.3	FOREIGN KEY 约束	237	17.4.1	流控制语句	293
15.3.4	CHECK 约束	240	17.4.2	定义和使用变量	295
15.3.5	DEFAULT 约束	241	17.4.3	使用附加过程及批语句关键字	296

17.5 调用存储过程	301	19.3 磁盘 I/O 性能	340
17.5.1 使用 EXECUTE 语句执行 存储过程	301	19.3.1 标准的磁盘传输率与 SQL Server 的关系	340
17.5.2 存储过程的自动执行	302	19.3.2 连续和不连续的磁盘 I/O 操作 ..	341
17.6 修改和重新命名存储过程	303	19.3.3 磁盘 I/O 传输率/RAID 控制器 传输率/PCI 总线带宽	341
17.7 删除存储过程	305	19.3.4 RAID	342
17.8 系统存储过程	306	19.3.5 磁盘 I/O 并行	346
17.8.1 管理类系统存储过程	306	19.4 SQL Server 索引	348
17.8.2 辅助类系统存储过程	307	19.4.1 聚集索引	349
17.8.3 配置类系统存储过程	308	19.4.2 非聚集索引	350
17.8.4 监控类系统存储过程	309	19.4.3 覆盖索引	350
17.9 扩展存储过程	309	19.4.4 自动覆盖索引或隐蔽查询	351
第 18 章 触发器管理技术	311	19.4.5 索引选择	351
18.1 触发器的基本概念及用途	311	19.4.6 聚集索引的选择、连续键数据 以及如何查找热点	352
18.2 创建触发器	312	19.4.7 FILLFACTOR 和 PAD_INDEX 的重要性	354
18.2.1 CREATE TRIGGER 语句	312	19.5 SQL Server 性能优化工具	356
18.2.2 指定触发器的类型	314	19.5.1 数据和工作负荷示例	356
18.2.3 测试表列是否被修改	319	19.5.2 SQL 事件探查器	356
18.2.4 使用企业管理器创建触发器	323	19.5.3 查询分析器	361
18.3 触发器的使用	324	19.5.4 性能监视器	364
18.3.1 触发器的限制	324	19.6 其他性能主题	369
18.3.2 多行的考虑	325	19.6.1 减少网络流量和数据库服务器 的资源消耗	369
18.3.3 条件 INSERT 触发器	327	19.6.2 死锁	369
18.3.4 使用嵌套触发器	328	19.6.3 只要有任何可能就应 避免的 SQL	369
18.3.5 加强数据约束	331	19.6.4 灵巧化标准	370
18.4 修改触发器	332	19.6.5 分区视图	370
第 19 章 SQL Server 数据库性能优化 ..	334	19.6.6 复制和备份性能	370
19.1 SQL Server 性能优化的原则	334	19.6.7 特殊的磁盘 I/O 调整方案: EMC 对 称集成高速缓存磁盘队列	370
19.2 耗费 CPU 和磁盘 I/O 资源的组件	335		
19.2.1 工作线程	335		
19.2.2 LazyWriter	336		
19.2.3 Checkpoint	337		
19.2.4 Log Manager	338		
19.2.5 Read-Ahead Manager	339		

第 1 章 SQL Server 2000 的新特色

使用 SQL Server 2000 的最新增强功能开发数据库解决方案。建立在 SQL Server 7.0 可扩展基础上的 SQL Server 2000 代表着下一代 Microsoft .NET Enterprise Servers (企业服务器) 数据库的发展趋势。SQL Server 2000 是为创建可伸缩电子商务、在线商务和数据仓储解决方案而设计的真正意义上的关系型数据库管理与分析系统。这一章主要介绍了 SQL Server 2000 的新特色, 主要包括:

- 新的数据类型
- 用户自定义函数
- 增强的索引
- SQL Server 2000 和 XML
- 全文检索功能
- 联合服务器

1.1 新的数据类型

继 SQL Server 7.0 增加了四种新的数据类型之后, SQL Server 2000 又提供了三种新的数据类型, 分别是 `bigint`、`sql_variant` 和 `table`。在数据量越来越大的今天, 用 `int` (-2^{31} (-2,147,483,648) 到 $2^{31}-1$ (2,147,483,647)) 来表示整数有时已经显得不太够用了, `FLOAT` 也不是很合适, 所以 VB 7 和 SQL Server 2000 都有了新的整数类型, 这就是 `bigint`, 它是 8bytes 的整数类型, 范围是 -2^{63} (-9223372036854775808) 到 $2^{63}-1$ (9223372036854775807)。熟悉 VBScript 的人都知道, VBScript 使用的是一种叫做 `variant` 的数据类型。它可以存放任何数据类型, 数字、字符、对象等。SQL Server 2000 里面的 `sql_variant` 还没有那么强大的功能, 它不能存放 `blob` 的数据, 如 `text`, `ntext` 和 `image`, 还有 `timestamp`, 但是已经给编程带来了很大的方便了。

SQL Server 2000 里面是没有数组类型的, 但是新的表格类型允许应用软件暂时存储那些将要用到的结果。它不能作为表的一个列出现, 而只能在 T-SQL 程序中使用。作为自定义函数的返回值很好用。

例如:

```
DECLARE @TableVar TABLE
(Cola int PRIMARY KEY, Colb char(3))
INSERT INTO @TableVar VALUES (1, 'abc')
INSERT INTO @TableVar VALUES (2, 'def')
SELECT * FROM @TableVar
GO
```

注意: 以前版本的 ADO 对这几种数据类型的支持都不完全, 要在程序中使用这两种数

据, 请使用 MDAC 2.6。

1.2 用户自定义函数

在 SQL Server 2000 里面, 用户可以建立自定义的函数, 函数返回值可以是一个值, 也可以是一个表。自定义函数到底有什么作用, 很多人可能还不是太清楚。在优化数据库时, 尽量不要使用游标, 因为这样会带来更大的系统开销。但是有的时候又必须使用游标, 举一个例子, 比如希望得到是一段汉字的拼音。要想把汉字转化为拼音, 必须通过查表来完成, 那么就必须先定义一个游标, 然后再对字段中的每一个字进行查表。但是现在可以使用自定义函数来完成同样的操作, 就大大的节省了系统开销。下面举了一个在两个日期之间的休息日的 SQL Server 的自定义函数。

例 1.1

```
CREATE FUNCTION GetNumDay(@Start datetime,@End datetime)
RETURNS integer AS
BEGIN
declare
    @intNum integer,
    @intAllDay integer,
    @i integer,
    @intWeekDay integer,
    @tempDateTime datetime,
    @intVac integer
select @intAllDay=DateDiff(dd,@Start,@End)
select @i=0
select @intVac=0
while (@i<@intAllDay)
begin
    select @tempDateTime=DateAdd(dd,@i,@Start)
    select @intWeekDay=datepart(dw,@tempDateTime)-1
    if (@intWeekDay=6) or (@intWeekDay=0)
    begin
        select @intVac=@intVac+1
    end
    select @i=@i+1
end
return @intVac
END
```

在这个函数生成以后, 我们就可以使用这个函数了, 比如:

```
select aa=dbo. GetNumDay ('2001-2-15 00:00:00.000','2001-2-28 00:00:00.000')
```

得到的结果是 4。

使用 SQL Server 2000 中的自定义函数,不但执行效率提高了,代码的可读性也好多了。虽然其他数据库已经提供了这样的功能,但是 MS 的跟随速度是越来越快了,提高也是越来越多了,这也是它的市场份额越来越大的原因。

1.3 增强的索引

在以前版本的 SQL Server 里,视图是不可以有索引的,所以视图一般也就没有什么用处了,除了偶尔用它来做一些权限上的管理以外。查询一个视图和使用一个连接语句在执行效率上没有什么区别。但是,在 SQL Server 2000 中可以在视图上创建索引。这样,现有的应用程序就可以不再修改而得到很大的效率提高了。一般在编程我们都会用到一些复杂的连接,这样必须在连接的两端加上索引才能获得较好的效率,而且对连接的结果进行排序或者分组也是很慢的,现在一切都不一样了,程序中将不再出现复杂的 SQL 语句,所有的操作都交给自定义函数、存储过程和视图去完成了,我们只需要在程序中写 `select * from table` 就可以了,别的事情由数据库管理员去处理。

另外 SQL Server 2000 还可以给计算列加索引。下面举例来说明:

例 1.2

```
CREATE TABLE t2 (a int, b int, c int, x float,
y as CASE x
WHEN 0 THEN a
WHEN 1 THEN b
ELSE c
END)
CREATE TABLE mytable (
low int,
high int,
myavg AS (low + high)/2)
```

现在虽然可以给计算列加索引了,但是还是受到一些限制的,比如说,像 avg、sum 这样和整张表都有关的计算值就不可以加索引。还有,具有不确定计算值的也不可以加索引,表达式中使用的字段也必须是这张表中的。下面简单的列出了不能确定返回值的函数:

```
@@ERROR
@@IDENTITY
@@ROWCOUNT
@@TRANCOUNT
APP_NAME
CURRENT_TIMESTAMP
CURRENT_USER
DATENAME
FORMATMESSAGE
```

```
GETANSINULL  
GETDATE  
GetUTCDate  
HOST_ID  
HOST_NAME  
IDENT_INCR  
IDENT_SEED  
IDENTITY  
NEWIDPERMISSIONS  
SESSION_USER  
STATS_DATE  
SYSTEM_USER  
TEXTPTR  
TEXTVALID  
USER_NAME
```

SQL Server 2000 里面对索引作了一些优化, 在多 CPU 的机器上建立索引的时候可以使用平行多线程来扫描和分类数据。其实现的技术细节是, 服务器在建立索引之前对表作一个快速的随机扫描, 把数据按照设置的平行度分为几块, 然后每个线程利用过滤器对基表进行查询, 分别建立索引结构, 每个线程都完成之后再把这索引连接起来。SQL Server 2000 中很多地方对多 CPU 和 Clustered Server 做了优化。

1.4 SQL Server 2000 和 XML

现在 XML 很流行, 各种软件都开始支持 XML 了。在这一点上, Microsoft 公司自然不会落后, 浏览器 Office, SQL Server, MDAC 都能够支持 XML。去年 MS 就在 MDAC2.0/2.1 里面提供了一个功能, 把一个结果集保存为 XML, 在 SQL Server 2000 里面, 使用 XML 更加方便了, SQL 查询可以直接返回一个 XML, 并且还可以通过 XML 来修改数据库中的数据。

SQL Server 2000 对 XML 新增了一些支持的功能, 这些特色让 SQL Server 2000 成为 XML-enabled 的数据库, 这些新功能包括:

- 可以透过 URL 存取 SQL Server;
- 支持 XML-Data Schemas 和可以定义 XPath 对这些 schema 查询;
- 可以读取和写入 XML 资料;
- 读取 SQL Server 的资料以 XML 格式输出: 透过使用 SELECT 语法加上 FOR XML 子句;
- 将 XML 文件读入: 透过 OpenXML rowset provider 来写入或是运作 XML 资料 OpenXML 会将 XML 文件以 rowset 的方式来传回, 所以可以用在 T-SQL 语法中取代 table、view 或是 OpenRowset 的位置;
- 对 Microsoft SQL Server 2000 OLE DB provider(SQLOLEDB) 的存取能力。

要使 SQL 查询直接返回一个 XML 非常的简单, 只需要在 SQL 语句后面加一个 FOR XML AUTO 就可以了。

例如:

SELECT top 1 Employeeid, FirstName, BirthDate FROM Employees FOR XML AUTO

如果使用 FOR XML RAW 的话, 就会返回

<row Employeeid="1" FirstName="Nancy" BirthDate="1948-12-08T00:00:00"/>

如果用 FOR XML AUTO, ELEMENTS, 结果是

<Employees><Employeeid>1</Employeeid><FirstName>Nancy</FirstName><BirthDate>1948-12-08T00:00:00</BirthDate></Employees>

使用 FOR XML EXPLICIT 特别麻烦, 前面要定义一堆东西, 但是功能也很强大, 这里就不详细介绍了。

上面是如何返回 XML 数据, 下面来看一下如何把 XML 当表使用, OPEN XML 可以把 XML 的数据打开成为一个表。

例 1.3

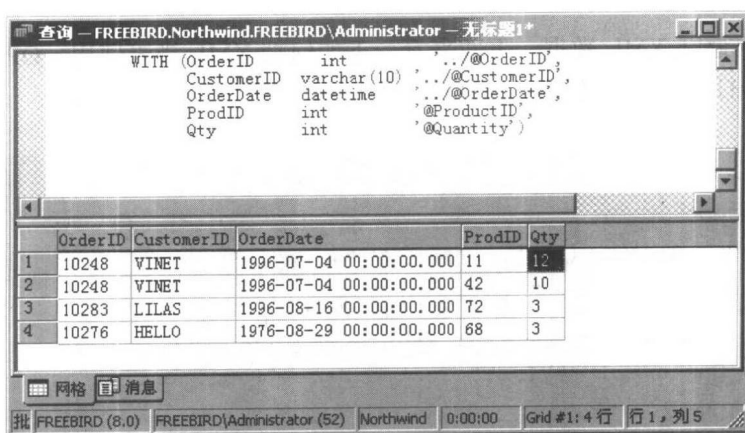
```
declare @l int
declare @idoc int
declare @doc varchar(1000)
set @doc = '
<ROOT>
  <Customer CustomerID="VINET" ContactName="Paul Henriot">
    <Order OrderID="10248" CustomerID="VINET" EmployeeID="5"
      OrderDate="1996-07-04T00:00:00">
      <OrderDetail ProductID="11" Quantity="12"/>
      <OrderDetail ProductID="42" Quantity="10"/>
    </Order>
  </Customer>
  <Customer CustomerID="LILAS" ContactName="Carlos Gonzlez">
    <Order OrderID="10283" CustomerID="LILAS" EmployeeID="3"
      OrderDate="1996-08-16T00:00:00">
      <OrderDetail ProductID="72" Quantity="3"/>
    </Order>
  </Customer>
  <Customer CustomerID="HELLO" ContactName="Carlos Gonzlez">
    <Order OrderID="10276" CustomerID="HELLO" EmployeeID="4"
      OrderDate="1976-08-29T00:00:00">
      <OrderDetail ProductID="68" Quantity="3"/>
    </Order>
  </Customer>
</ROOT>'
```

```

--Create an internal representation of the XML document.
exec sp_xml_preparedocument @idoc OUTPUT, @doc
-- SELECT stmt using OPENXML rowset provider
SELECT *
FROM OPENXML (@idoc, '/ROOT/Customers/Order/OrderDetail',2)
WITH (OrderID int '@OrderID',
      CustomerID varchar(10) '@CustomerID',
      OrderDate datetime '@OrderDate',
      ProdID int '@ProductID',
      Qty int '@Quantity')

```

执行结果如图 1.1 所示。



	OrderID	CustomerID	OrderDate	ProdID	Qty
1	10248	VINET	1996-07-04 00:00:00.000	11	12
2	10248	VINET	1996-07-04 00:00:00.000	42	10
3	10283	LILAS	1996-08-16 00:00:00.000	72	3
4	10276	HELLO	1976-08-29 00:00:00.000	68	3

图 1.1 把 XML 数据打开为一个表

这些功能意味着可以使用 SQL Server 2000 来生成一个 XML 并且通过 XML 得到客户端的返回值（不一定要通过 ASP 之类的工具，可以通过 URL 来访问 SQL Server 2000 返回的 XML 的）。

1.5 全文检索功能

SQL Server 2000 的全文检索有了一些很好的改进，主要体现在以下几个方面：

(1) 就是不需要重建全文检索索引就能够更新数据。索引更新有好几种方法：可以手动更新，也可以在 job 里面更新，或者使用后台更新选项（background update index option）在数据发生变化的同时更新索引。这样的话，就使得全文检索可以用于频繁更新的数据的实时检索，比如用于论坛贴子的检索。

(2) 全文检索索引可以用于 image 类型的字段。当在 image 字段中存储了指定文档类型的文件时，全文检索可以调用该类型文档对应的过滤器来取得文件的信息。举例说明，在一个 image 字段中放了一个 Word 文件，并且把这个字段加入了全文检索索引，那么可以得到这个文件的作者、文件大小、修改时间等信息。SQL Server 2000 支持的文件类型有 Office

文档、文本文件、HTML 文档和邮件。

例 1.4

```
SELECT Q.FileName, Q.Size, Q.DocAuthor, W.Citizenship
FROM OpenQuery(MyLinkedServer,
' SELECT FileName, Size, DocAuthor
FROM SCOPE('' "D:" '')
WHERE CONTAINS('' "SQL Server"
NEAR() text'')
AND FileName LIKE '%.doc%'
) AS Q,
writers AS W
WHERE Q.DocAuthor = W.writer_name
```

相对来说, SQL Server 7 的全文检索功能又怎样呢? 下面对其作简单的说明:

- (1) 巨耗硬盘资源, 约需使用源内容大小的 40%~50% 的磁盘空间。
- (2) 不是随时更新的 (和 Index Server 不一样, 它有一个 CiDaemon, 随时监测文件变化) 不过也可以设置成每天晚上更新或者事件触发都可以的。
- (3) 支持中文, 但是不太好, 分词技术不完善, 做普通的查询还可以, 不适合用做搜索引擎。
- (4) 查询速度还可以, 感觉足够快。联合 SQL Server 2000 服务器 (Federated SQL Server 2000 Servers)

1.6 联合服务器

1.6.1 集群服务器与联合服务器的区别

下面简单说说集群服务器与联合服务器之间的区别。

SQL Server 2000 的数据库能够被分散在一组独立的数据库服务器上以支持大规模的 Web 站点的访问需求和企业级的数据处理系统。面对日益增长的需求, 只需要简单的添加一个节点服务器就可以了。

Windows 2000 的 COM+ 组件就是被设计来用于在集群的 Windows 2000 服务器上实现商业逻辑层的。每个服务器上都有有一套同样的 COM+ 组件, Windows 2000 自己来平衡各个服务器之间的负载。使用 Cluster Administrator 可以很方便的配置集群服务器。

但是 SQL Server 2000 并不支持这样的集群。SQL Server 2000 支持可更新的分布式视图 (distributed partitioned views), 用它来水平的将数据分割在一组服务器上。这使得用户可以把一个数据库服务器添加到一组数据库服务器中, 这些数据库相互合作, 以提供和集群数据库服务器相同的性能标准。虽然它们合作管理这些数据, 但是它们之间仍然是最高的。

数据库服务器和商业逻辑服务器不一样, 运行商业逻辑的时候每个服务器都可以完成

