

# SQL Server 2000 简明教程

黄维通 编著

清华大学出版社

(京) 新登字 158 号

### 内 容 简 介

本书介绍了 SQL Server 2000 网络关系型数据库的基本知识, 包括 SQL Server 2000 的安装、使用、管理、安全以及与不同数据源连接等重要应用。全书内容循序渐进, 通过实例操作力求使读者掌握 SQL Server 2000 的基本操作和应用。通过这个教材的学习, 读者可以快速掌握数据库的基本应用和操作, 并对 SQL Server 应用有较全面的了解。

本书适合初学者作为数据库课程的入门教材, 也可为广大数据库学习爱好者提供必要的参考。

版权所有, 翻印必究。

本书封面贴有清华大学出版社激光防伪标签, 无标签者不得销售。

书 名: SQL Server 2000 简明教程

作 者: 黄维通 编著

责任编辑: 苗建强

出 版 者: 清华大学出版社 (北京清华大学学研大厦, 邮编 100084)

<http://www.tup.tsinghua.edu.cn>

印 刷 者:

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 16.75 字数: 454 千字

版 次: 2002 年 3 月第 1 版 2002 年 3 月第 1 次印刷

书 号: ISBN 7-302-05278-6/TP·3101

印 数: 0001~5000

定 价: 22.00 元

# 前 言

Microsoft SQL Server 2000 是 Microsoft 公司推出的 SQL Server 最新版本,是目前广为使用的大型数据库系统之一,在 IT 技术飞速发展的今天,学习和掌握数据库原理和技术,是新时代 IT 技术发展对人才的知识结构的新要求。

本书从基本概念出发,循序渐进,通过实际应用,介绍 SQL Server 的基本知识及其应用,旨在引导读者走入数据库的世界。本书共 15 章,每一章的内容都融入了实际应用,包括数据库的创建、表的操作、视图的操作、索引的创建、存储过程的建立、触发器的应用以及函数的应用,在掌握这些基本概念以后,进一步介绍 SQL Server 中的程序设计、安全管理、数据备份与恢复以及不同数据源之间的连接等知识。

本书由部分软件开发人员共同编写,其中刘艳慧、刘先言调试了书中的样例代码,孟威、张佳、刘然提供了部分资料并审阅了部分章节内容,刘洋、邓秋霞参与了一些编写工作,张波协助进行书中的图形处理,陈辉协助完成文字的录入排版工作,在此向他们的辛勤劳动表示感谢。

由于作者水平有限,加上对新软件的使用经验不足,因此,书中难免存在一些不妥之处,恳请广大读者朋友谅解,并提出宝贵意见,以便在将来的修订过程中进一步完善。

作者联系信箱 hwt@cic.tsinghua.edu.cn

编者

2002 年 2 月

# 目 录

第 1 章 SQL Server 概述 .....	1
1.1 SQL .....	1
1.2 SQL 的特点 .....	1
1.3 SQL Server 的结构 .....	2
1.3.1 SQL Server 是单一进程多线程的关系型数据库 .....	2
1.3.2 SQL Server 是以 Client/Server 为设计结构 .....	3
1.3.3 Client/Server 数据库的两个 API .....	5
1.3.4 SQL Server 支持在客户端以 Net-Library 或 ODBC 存取服务器端 .....	5
1.3.5 支持分布式数据库结构 .....	7
1.4 SQL Server 的性能 .....	7
1.5 SQL 分布式管理框架介绍 .....	7
1.6 关系数据库模型简述 .....	8
第 2 章 SQL Server 的安装及基本操作 .....	10
2.1 SQL Server 2000 的硬件和软件安装要求 .....	10
2.2 SQL Server 的安装内容 .....	11
2.3 SQL Server 的安全性 .....	12
2.3.1 登录认证 .....	13
2.3.2 数据库用户账号和角色 .....	13
2.3.3 权限有效性的确认 .....	13
2.4 安装 SQL Server 前的准备工作 .....	14
2.5 安装 SQL Server .....	15
2.6 SQL 群组中各项目的作用 .....	24
2.7 SQL Server 服务器端的操作 .....	25
2.7.1 SQL Server 的启动 .....	25
2.7.2 对 sa 账号加入密码 .....	25
2.7.3 SQL Server 的配置 .....	28
2.7.4 查看专用存储器的构造 .....	29
2.8 SQL Server 客户端的操作界面 .....	30
第 3 章 数据库的基本操作 .....	33
3.1 文件和文件组 .....	33

---

3.1.1	文件.....	33
3.1.2	文件组.....	33
3.2	创建数据库.....	34
3.2.1	用企业管理器以图形化界面建立数据库.....	34
3.2.2	在查询分析器窗口中用 T-SQL 命令创建数据库.....	36
3.2.3	事务日志.....	37
3.2.4	查看数据库信息.....	38
3.3	管理数据库.....	40
3.3.1	打开数据库.....	40
3.3.2	增加数据库容量.....	40
3.3.3	查看目前数据库选项设定及修改.....	42
3.3.4	缩减数据库容量.....	43
3.3.5	更改数据库名称.....	45
3.3.6	查看目前 SQL Server 上共有几个数据库.....	45
3.3.7	数据库的删除.....	46
<b>第 4 章</b>	<b>数据库中表的建立.....</b>	<b>48</b>
4.1	表的概念.....	48
4.2	SQL Server 的数据类型.....	48
4.3	数据库中表的操作.....	54
4.3.1	数据库中表的创建.....	54
4.3.2	数据库中表的删除.....	58
4.3.3	数据库中表的修改.....	59
<b>第 5 章</b>	<b>数据检索.....</b>	<b>70</b>
5.1	SELECT 语句.....	71
5.2	带条件的检索.....	76
5.2.1	WHERE 子句.....	76
5.2.2	WHERE 子句中条件的指定.....	78
5.2.3	HAVING 子句.....	82
5.2.4	COMPUTE BY 及 COMPUTE 子句.....	84
5.3	从多张表中查询数据.....	84
5.4	UNION 操作.....	87
5.5	子查询.....	89
<b>第 6 章</b>	<b>实现 SQL Server 的数据完整性.....</b>	<b>91</b>
6.1	数据完整性的基本概念.....	91
6.1.1	数据完整性的类型.....	91
6.1.2	强制数据完整性.....	92

---

6.2	定义约束.....	92
6.2.1	约束的类型.....	92
6.2.2	约束的创建.....	93
6.2.3	查看约束的定义信息.....	95
6.2.4	删除约束.....	97
6.3	约束类型.....	98
6.3.1	DEFAULT 约束.....	98
6.3.2	CHECK 约束.....	99
6.3.3	PRIMARY KEY 约束.....	101
6.3.4	UNIQUE 约束.....	102
6.3.5	FOREIGN KEY 约束.....	103
6.4	创建约束的其他选项.....	105
6.4.1	使用 WITH NOCHECK 选项.....	105
6.4.2	使用 NOCHECK 选项.....	106
6.5	使用默认.....	107
6.6	使用规则.....	109
6.7	数据完整性强制方法的选择.....	112
第 7 章	索引及其应用.....	114
7.1	有关索引的基础知识.....	114
7.1.1	SQL Server 中数据的存储.....	114
7.1.2	SQL Server 中数据的访问.....	115
7.2	创建索引的原因和选择索引列.....	115
7.2.1	创建索引的考虑因素.....	115
7.2.2	选择创建索引的数据列.....	117
7.3	索引的分类.....	117
7.3.1	聚集索引.....	117
7.3.2	非聚集索引.....	120
7.4	建立可利用的索引.....	122
7.4.1	在查询分析器窗口中用 SQL 命令建立索引.....	122
7.4.2	在企业管理员的图形化界面上建立索引.....	127
7.5	索引信息的查询.....	129
7.6	更改索引的名称.....	130
7.7	删除索引.....	131
7.7.1	在“查询分析器”窗口中用 T-SQL 命令语句删除索引.....	131
7.7.2	在“企业管理器”窗口中删除索引.....	132
7.8	设置创建索引的选项.....	132
7.8.1	设置 FILLFACTOR 选项.....	132

7.8.2	设置 PAD_INDEX 选项.....	134
7.9	索引的分析和维护.....	134
7.9.1	索引分析.....	134
7.9.2	索引维护.....	137
7.10	使用索引优化向导.....	141
<b>第 8 章</b>	<b>视图及其应用.....</b>	<b>145</b>
8.1	综述.....	145
8.1.1	视图的基本概念.....	145
8.1.2	使用视图的优点和缺点.....	146
8.2	视图的创建和查询.....	146
8.3	视图定义信息查询.....	148
8.3.1	使用企业管理器.....	149
8.3.2	通过执行系统存储过程查看视图的定义信息.....	150
8.4	视图的修改和删除.....	152
8.4.1	视图的修改.....	152
8.4.2	视图的删除.....	153
8.5	通过视图修改数据.....	154
<b>第 9 章</b>	<b>存储过程及其应用.....</b>	<b>157</b>
9.1	存储过程的概念.....	157
9.2	存储过程的优点.....	157
9.3	创建和执行简单存储过程.....	158
9.3.1	创建存储过程.....	158
9.3.2	创建存储过程中的步骤及注意事项.....	159
9.3.3	浏览存储过程信息.....	161
9.4	创建和执行含参数存储过程.....	163
9.4.1	创建含有输入参数的存储过程.....	163
9.4.2	执行含有输入参数的存储过程.....	165
9.4.3	创建含有输出参数的存储过程.....	166
9.4.4	执行含有输出参数的存储过程.....	167
9.5	存储过程的重编译处理.....	168
9.5.1	存储过程的处理.....	168
9.5.2	存储过程的重编译处理.....	169
9.6	修改和删除存储过程.....	170
9.6.1	修改存储过程.....	170
9.6.2	删除存储过程.....	172
9.7	系统存储过程和扩展存储过程.....	172

---

9.7.1 系统存储过程.....	172
9.7.2 扩展存储过程.....	173
第 10 章 触发器及其应用.....	177
10.1 触发器的基本概念和优点.....	177
10.1.1 触发器的基本概念.....	177
10.1.2 使用触发器的优点.....	177
10.2 触发器的创建.....	178
10.2.1 INSERT 触发器.....	179
10.2.2 UPDATE 触发器.....	180
10.2.3 DELETE 触发器.....	182
10.2.4 查看触发器信息.....	182
10.3 使用触发器.....	185
10.3.1 使用触发器强制数据完整性.....	185
10.3.2 使用触发器强制业务规则.....	185
10.4 修改和删除触发器.....	186
10.4.1 修改触发器.....	186
10.4.2 删除触发器.....	187
10.4.3 禁止或启用触发器.....	187
10.5 嵌套触发器和递归触发器.....	188
10.5.1 嵌套触发器.....	188
10.5.2 递归触发器.....	189
第 11 章 用户自定义函数.....	191
11.1 用户自定义函数的基本概念.....	191
11.2 创建用户自定义函数.....	191
11.2.1 创建用户自定义函数.....	191
11.2.2 查看用户自定义函数.....	193
11.3 用户自定义函数类型.....	195
11.3.1 标量函数.....	195
11.3.2 内嵌表值函数.....	196
11.3.3 多语句表值函数.....	198
11.4 修改和删除用户自定义函数.....	199
11.4.1 修改用户自定义函数.....	199
11.4.2 删除用户自定义函数.....	200
11.4.3 设置用户自定义函数的权限.....	201
第 12 章 SQL Server 中的程序设计.....	202
12.1 程序设计中批处理的基本概念.....	202

---

12.2	事务处理.....	204
12.3	SQL Server 的流程控制语句.....	208
12.3.1	声明变量.....	209
12.3.2	预声明的全局变量.....	210
12.3.3	RETURN 语句.....	210
12.3.4	CASE 表达式.....	212
12.3.5	BEGIN...END 块.....	213
12.3.6	游标.....	215
第 13 章	SQL Server 的安全管理.....	218
13.1	SQL Server 的安全机制.....	218
13.1.1	与 Windows 2000 集成的登录验证模式.....	218
13.1.2	基于角色的安全性.....	218
13.2	管理服务器的安全性.....	219
13.2.1	服务器登录账号.....	219
13.2.2	设置安全验证模式.....	220
13.2.3	服务器账号管理.....	220
13.2.4	服务器角色.....	223
13.2.5	管理数据库的用户.....	225
13.3	管理权限.....	228
13.3.1	SQL Server 2000 的权限.....	228
13.3.2	权限设置.....	229
13.3.3	权限和系统表.....	231
13.4	应用程序的安全性与应用程序角色.....	232
第 14 章	备份与还原.....	234
14.1	数据库备份与恢复的基本概念.....	234
14.2	备份的类型.....	235
14.3	备份操作.....	235
14.3.1	用 T-SQL 命令方式进行备份.....	235
14.3.2	使用 SQL Enterprise Manager 进行备份.....	239
14.4	还原及其操作.....	240
14.4.1	用 T-SQL 命令进行还原操作.....	240
14.4.2	使用 SQL Enterprise Manager 进行还原操作.....	242
14.4.3	灾难后的恢复.....	243
14.5	制定备份及恢复计划.....	243
14.5.1	备份计划.....	244
14.5.2	恢复计划.....	245

---

第 15 章 通过 ODBC 连接不同数据源 .....	246
15.1 ODBC 概述.....	246
15.2 ODBC 数据源.....	247
15.3 安装 ODBC 的数据源驱动程序.....	248
15.3.1 添加数据源.....	249
15.3.2 数据源转换.....	251

# 第 1 章 SQL Server 概述

## 1.1 SQL

SQL 是用来对存放在计算机中的数据库进行组织、管理和检索的语言。SQL 一词是“Structured Query Language (结构式查询语言)”的缩写,是 IBM 公司 San Jose 实验室为 System R 而设计的查询语言,从 1982 年开始,美国国家标准协会(ANSI)即着手 SQL 的标准化工作,1986 年 ANSI 的数据库委员会批准了 SQL 作为关系数据库语言的美国标准,这就是第一个 SQL 标准,同时公布了 SQL 标准文本。在此后不久的 1987 年,国际标准化组织(ISO)也作出了同样的决定,目前的 SQL 标准是 1992 年制定的 SQL - 92 标准,是一种用于与数据库进行交互的语言。随着数据库技术的发展和数据库功能的增强,目前,各个 DBMS 厂商都自称采用 SQL 语言,但完全按 ISO 标准实现的并不多。IBM 公司实际上以其 DB2 的 SQL 作为 IBM 的标准,其他厂商所实现的 SQL,由于历史原因,也有不少差异,但总的倾向是向国际标准靠拢,并与 DB2 的 SQL 保持兼容。SQL 语言的极大普及是当今计算机工业中最引人注目的趋势之一。在过去的几年中,SQL 已经发展成为标准计算机数据库查询语言。现在,微机到大型机,有很多数据库产品支持 SQL,SQL 的国际标准已经被采用并被不断扩充。SQL 在所有主要计算机开发商的数据库体系中占有重要的地位。

## 1.2 SQL 的特点

SQL 是一种综合的、通用的、功能极强的关系数据库语言(有关关系数据库的概念,将在本章的最后给读者一个简要的概述),它包括数据定义(Definition)、数据操纵(Manipulation)、数据管理(Management)、存取保护(Access Protection)、处理控制(Control)等多种功能。利用表(table)、索引(index)、码(keys)、行(rows)和列(columns)等来确定存储位置。

SQL 语言本身并不是一个很完整的编程语言,例如它不支持流控制等。一般它都与其他编程语言(如 DELPHI、Powerbuilder、VB、VC 等)结合来使用。

SQL 的主要特点在于:

### (1) 一体化的特点

SQL 语言能完成定义关系模式,录入数据以建立数据库、查询、更新、维护、数据库重构、数据库安全性控制等一系列操作要求,用 SQL 可以实现数据库生命期当中的全部活动。由于关系模型中实体与实体间的联系都是用关系来表示,这种数据结构的单一性保证了操作符的单一性。

### (2) 统一的语法结构，多种使用方式

SQL 有两种使用方式，一种是联机使用方式，另一种是嵌入程序方式。大多数的程序接口都采用嵌入的 SQL 语言。虽然使用方式不同，SQL 语言的语法结构是一致的。这使得用户与程序员之间的通信得以改善。

### (3) 高度非过程化

在 SQL 中，只需用户提出“干什么”，而无须指出“怎么干”，存取路径的选择和 SQL 语句操作的过程由系统自动完成。

### (4) 语言简洁

SQL 语言十分简洁，语法简单。标准 SQL 中，完成核心功能只用了 6 个动词（如表 1-1 所示），因此简单易学，SQL 按其功能可以分为三大部分：

- ◇ 数据定义语言（Data Definition Language，简称 DDL），用于定义、撤消和修改数据库对象
- ◇ 数据操纵语言（Data Manipulation Language，简称 DML），用于数据库中数据的修改和检索
- ◇ 数据控制语言（Data Control Language，简称 DCL），用于数据访问权限的控制

表 1-1 标准 SQL 的 6 个核心动词

SQL 功能	动 词
数据定义	CREATE
数据操纵	INSERT、UPDATE、DELETE、SELECT
数据控制	GRANT

### (5) 客户机/服务器（Client/Server）结构

SQL 能使应用程序采取分布式客户机/服务器结构。交互式查询、报表打印和应用程序称为数据库的“前端”，在个人机上运行，存储和数据管理的后端数据库引擎在服务器上运行，在此情况下，SQL 作为用于用户交互的前端工具和用于数据库管理的后端引擎之间通信的桥梁。

### (6) 支持异类复制

它可以将 SQL Server 数据复制到其他的数据库中，包括 Access、Oracle、Sybase 和 DB2，并采用 ODBC 作为其连接机制。

### (7) Internet 数据库功能的集成

支持数据库信息自动发布到 HTML 文档，同时结合 Microsoft Internet Information Server 和 SQL Server Internet Connector 这两个产品/技术，使用户得到完整的 Internet 数据发布的能力。

## 1.3 SQL Server 的结构

### 1.3.1 SQL Server 是单一进程多线程的关系型数据库

它与 ORACLE 不同之处在于 ORACLE 是多进程数据库，每一用户所打开的进程上需

有一个协调机构，互相与其他进程上的协调机构做沟通，以协调多个用户对资源的存取，SQL Server 是由执行核心来分配多个用户对数据库的存取，减少多个进程对数据库存取的沟通、协调时间，进而提高执行效率，SQL Server 单进程（Single Process）、多线程（Multi-Thread）结构如图 1-1 所示。

它是依赖于同一个应用程序内的多线程工作的，而不是为每一个任务运行不同的可执行程序或应用程序，它的优点是在一定的性能水平上，其硬件要求很低，不像多进程，会消耗可观的系统资源，多线程数据库引擎以一种不同的方式，处理多用户访问，它不依赖于多任务操作系统来为 CPU 安排应用程序，而是自动担当这个重任，在理论上讲，数据库引擎自动处理的能力将提供更大的移植性，因此，数据库要管理多个任务的调度执行、内存和硬盘的访问。

由于是单进程，就没有必要进程之间的通信机制。多线程任务由数据库执行体本身进行管理，线程的操作由数据库引擎来制定，并在最终执行时把这些指令发送给操作系统，在这种方式下，数据库时间片为不同的操作系统采用不同的线程，在合适的时候，把这些线程中的用户指令送给操作系统，它不采用操作系统的时间分片应用程序，而是利用 DBMS 时间片线程。

按照图 1-1 的方式，数据库为各种操作命令如用户命令、锁定数据页、磁盘输入输出、缓存输入输出等均使用一个有限的工作单元。

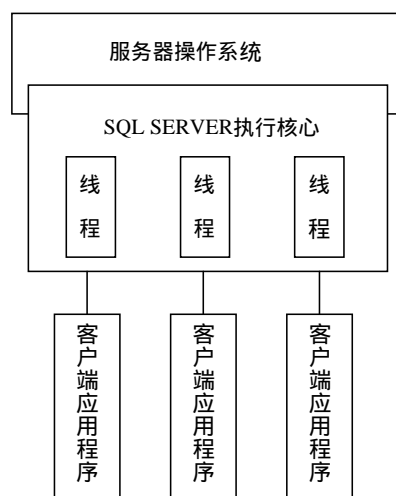


图 1-1 SQL Server 单进程、多线程结构

### 1.3.2 SQL Server 是以 Client/Server 为设计结构

就目前的 Client/Server 体系中有两种编写数据库引擎的体系结构，一种是“多进程引擎”，另一种是“单进程、多线程体系结构”。“单进程、多线程体系结构”多用于 SQL Server，这种结构依赖于同一应用程序内的多线程工作，而不是为每个任务运行不同的可执行程序或应用程序。下面简要介绍一下这两种在 Client/Server 体系结构下的数据库引擎。

### (1) 多进程数据库引擎

有的数据库引擎依靠几个可执行应用程序来完成用户查询工作,在这种体系结构下,用户每次登录,实际上都启动了几个数据库引擎本身的不同实例,为了协调许多用户访问同一个数据集,那些可执行程序和其他全局协调器同时工作以规划在不同用户间的操作,在这种类型数据库下的应用程序使用一种专门的内部进程通信(Internal Process Communication, IPC)设备进行通信,较典型的多进程数据库引擎的例子如 Oracle Server。

多进程数据库引擎具有如下的优点:

- ◇ 一个数据库可以同时支持多个用户
- ◇ 通过在物理机器上增加更多的 CPU 来支持扩展性

在多任务操作系统中,操作系统给每个任务均分配了特定比例的 CPU 时间,因此操作系统把时间分配给了多个应用程序,从而保证了在同一时间内只有一个任务在运行。

多进程数据库引擎的缺点是占用较大的系统资源。

### (2) 单进程、多线程数据库引擎

多线程数据库引擎不依赖于多任务操作系统来为 CPU 安排应用程序,因此,数据库引擎自身处理的能力将提供更大的可移植性。

单进程、多线程数据库引擎具有对服务器内存需求不高,节省硬件投资的优点,但也存在一个缺点,那就是多线程 DBMS 的可扩展性还有待于进一步提高。

### (3) Microsoft SQL Server 的线程服务

SQL Server 中最大的变化在于使用了线程,它不是模拟数据库内核中的线程,而是使用 Win32 操作系统线程,这就意味着这些线程运行于内存保护之下,它是由 Windows NT 操作系统内核进行抢先式调度。

在 Microsoft SQL Server 中,存在一个包含 1024 个“工作线程”的线程池,用以响应用户的连接请求。由于每个连接均对应一个线程,所以用户连接的最大数目理论上是 1024,但实际上数目上限是 32767,SQL Server 动态地分配可用的线程,也就是使用户连接的数目超出了可用线程的数目,如果在单个 SQL Server 上同时支持 1000 个用户,显然系统性能将会变得很差,但毕竟功能是存在的。

SQL Server 为不同的用途准备了不同的线程池,包括并行表格搜索、磁盘备份管理和用户连接等。

SQL Server 按 Client/Server 结构而设计,客户机/服务器结构减小了网络拥挤程度,并将数据库操作分开运行,像输入和显示数据这样需要用户频繁干预的任务集中在作为客户端的 PC 机上完成,客户端可分别处理数据,但若需读取共享数据,必须依据网络管道向服务器端做存取申请,而文件 I/O 和查询处理等频繁需要数据的任务则集中在数据库服务器上完成,共享数据存放在服务器端,从而使每个系统各显其能。SQL 还允许将个人机用作网络服务器或小型、大型机数据库的前端机,从而可使用户在个人机上存取大型数据库的内容。服务器端可先对客户端所要求的数据条件做处理,而仅传给客户端所要的结果,这样就大大减少了在网络上大量数据的来回传送,客户端可分担服务器端的部分工作,从而进一步提高系统的执行效率。SQL Server 把 Microsoft Windows NT 操作系统的功能、可扩展性及易管理性与高可靠性、高级的高端性能、客户机/服务器数据库管理结合在一起,

如图 1-2 所示。

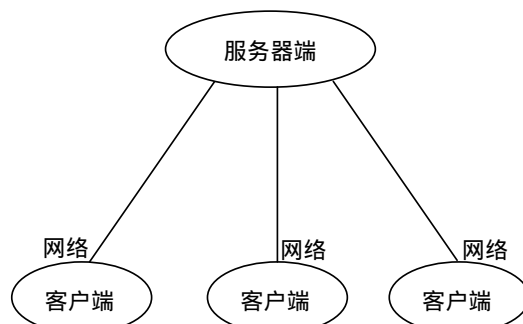


图 1-2 Client/Server 结构

### 1.3.3 Client/Server 数据库的两个 API

API (Application Program Interface) 即为应用程序接口, 在 SQL Server 应用程序中, 与其他应用程序不同, 它不存在预编译的程序, 取而代之的是数据库特有的调用, 只是以字符串的形式驻留在应用程序的源代码中。

接口类型之间的基本的差异定义了一个调用级的接口, DB-Library 和 ODBC 均为调用级接口。在接口内部, 主应用程序使本机语言调用转变成一个数据库函数, 数据操作语句以字符串的形式存储, 而不是像嵌入语言中那样为预编译做好标记。

对 SQL Server 来说, 存在两个每个应用程序都要使用的不同的 API, 传统上一直作为 SQL Server 标准的客户主语言 API 为 DB-Library, 另外一个就是 Transact-SQL (T-SQL), API 就是指向操作系统资源的语句途径或者是提供一系列特定服务的语言程序, T-SQL 以函数、数据类型、逻辑操作符和分支逻辑等形式增强了 SQL 标准。

尽管所有的 Client/Server 数据库为了开发语言程序, 均使用了两种 API, 但只有 SQL Server 使用 DB-Library 和 T-SQL (或者 ODBC), 这提供了利于 SQL Server 强大功能的两种途径。

### 1.3.4 SQL Server 支持在客户端以 Net-Library 或 ODBC 存取服务器端

SQL Server 允许用下列两种方式作客户端和服务器的连接管道。

#### (1) ODBC (Open DataBase Connection)

ODBC 实际上是一个数据库的访问库, 如 SQL Server 群组中的 MS Query、Access、Word、Excel、FoxPro、VB、C 等, 都可依据 ODBC 和 SQL Server 连接 (如图 1-3 所示), ODBC 可以使应用程序直接操纵数据库中的数据。ODBC 的独特之处在于使应用程序不随数据库的改变而改变。

ODBC 通过使用驱动程序来提供数据库的独立性。驱动程序与具体的数据库有关, 如

操作 Access，就需要使用 Access 的 ODBC 驱动程序。

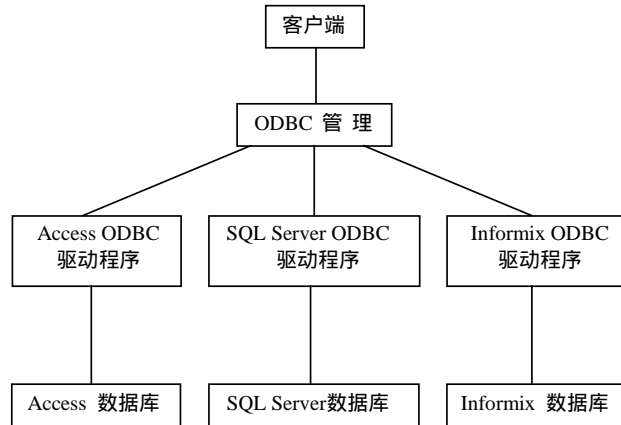


图 1-3 客户端用 ODBC 与 SQL Server 连接

驱动程序是一个用以支持 ODBC 函数调用的模块（通常是 DLL），应用程序通过调用驱动程序所支持的函数来操纵数据库。若想使应用程序操纵不同类型的数据库，就要动态地连接到不同的驱动程序上。

ODBC 还有一个驱动程序管理器（driver manager），驱动程序管理器包含在 ODBC.DLL 中，可连接到所有的应用程序中，它负责管理应用程序中 ODBC 函数与 DLL 中函数的绑定（Binding）。

对于大型 Client/Server 数据库管理系统，其所支持的 ODBC 驱动程序并不直接访问数据库，这些驱动程序实际上是数据库用于远程操作的网络通信协议的一个界面。

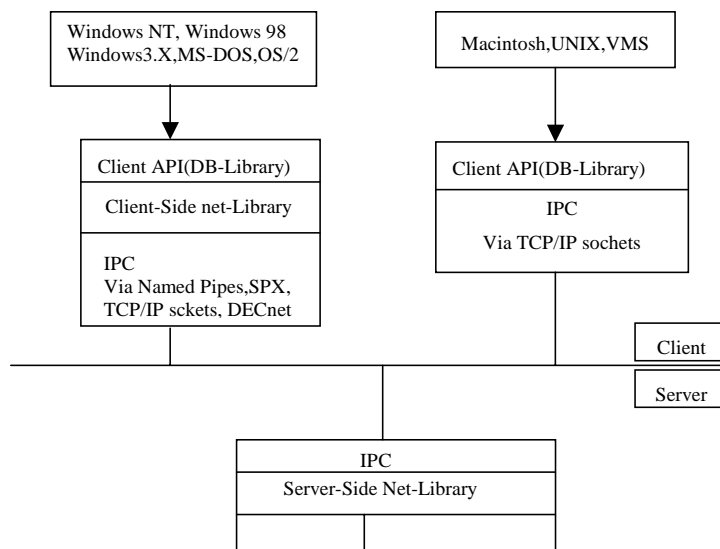


图 1-4 客户端用 Net-Library 与 SQL Server 服务器端连接

## (2) Net-Library

在 Client/Server 的最低层，DB-Library 必须通过网络来发送它的请求，这就要由 Net-Library 来完成这些操作，Net-Library 并不是由语言程序员和开发人员直接使用的，Net-Library 提供了客户端与服务器端的连接工具。如 ISQL/w、Enterprise Manager、Security Manager、VB、C 等客户端应用程序，都是借助 Net-Library 与 SQL Server 连接的（如图 1-4 所示）。

### 1.3.5 支持分布式数据库结构

在一个或多个网络中可有多个 SQL Server，用户可以将数据分别存放在各个 SQL Server 上，成为分布式数据库结构，客户端可向多个 SQL Server 存取数据，或同时向多个 SQL Server 存取数据，这样可以降低单个 SQL Server 处理过多的负担，提高系统的执行效率。

## 1.4 SQL Server 的性能

表 1-2 列出了各种 SQL Server 对象的系统范围，实际的范围将以应用的不同而有所变动。

表 1-2 SQL Server 的性能

对 象	范 围
数据库	32767 个数据库，最小为 1MB，最大为 1TB
表	每个数据库最多有 20 亿个表，每列的最大字节数为 8060（文本和图像列除外）
列	每表最多 1024 个列
索引	每表一个簇式索引，249 个非簇式索引，一个复合索引最多有 16 个索引关键字
触发器	每表最多有 3 个触发器，分别用于 insert、update 和 delete
存储过程	一个存储过程可以有 1024 个参数和最多 32 级嵌套
用户连接	32767 个
锁定及打开的对象	20 亿
打开的数据库	32767 个

## 1.5 SQL 分布式管理框架介绍

SQL 分布式管理框架（SQL Distributed Management Framework，SQL-DMF）是管理员用于管理 Microsoft SQL Server 的，是 Windows NT、SQL Server 服务和对象可编程库的集合。SQL-DMF 提供了一个与特定需要相匹配的灵活和可扩展的管理框架。通过采用与 SQL Server 直接交互的服务方法，管理员可以通过 SQL Server Executive 和 SQL Distributed Management Objects（分布式管理对象）在中央控制台处理多个服务器，SQL-DMF 减轻了