

//

Visual Basic应用技术系列



Visual Basic

郭琦 编著

数据库系统开发技术

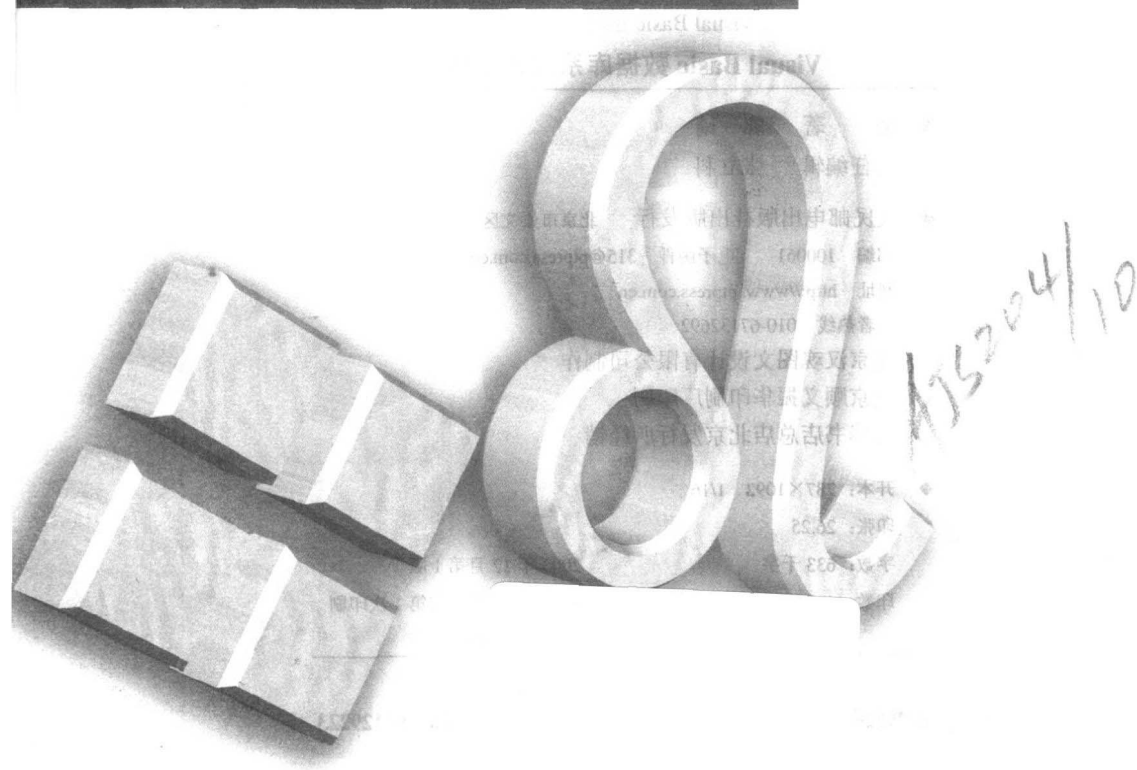


 人民邮电出版社
POSTS & TELECOM PRESS

Visual Basic

郭琦 编著

数据库系统开发技术



人民邮电出版社

图书在版编目(CIP)数据

Visual Basic 数据库系统开发技术/郭琦编著. —北京: 人民邮电出版社, 2003.12

ISBN 7-115-11727-6

I. V... II. 郭... III. BASIC 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2003) 第 099422 号

内 容 提 要

本书介绍了数据库的基本知识, 详细阐述了利用 Visual Basic 6.0 编制数据库操作应用程序的方法。

全书共 14 章内容, 主要包括了数据库基本知识、数据库创建基础知识、Visual Basic 6.0 数据库基本操作技术和高级操作技术, 全面深入地讲解数据库应用基础、Microsoft Access 介绍、SQL 语言、DAO 技术、ADO 技术、Visual Basic 数据访问控件、数据环境、数据报表、数据图表、远程数据库、多层结构数据库应用、复杂数据控件、数据安全性等技术知识, 最后给出了一个详细的综合应用实例。

本书每章先介绍基础原理, 再分析各种具体的技术手段, 最后是工程实践应用。既能让读者系统地学习各方面内容, 又可以直接应用于项目开发。

本书由浅入深、重点突出、实例丰富, 是初中级读者学习的最佳参考书。本书对于高级读者也具很好的参考价值。

Visual Basic 应用技术系列

Visual Basic 数据库系统开发技术

◆ 编 著 郭 琦

责任编辑 张立科

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号

邮编 100061 电子函件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

读者热线 010-67132692

北京汉魂图文设计有限公司制作

北京顺义振华印刷厂印刷

新华书店总店北京发行所经销

◆ 开本: 787×1092 1/16

印张: 26.25

字数: 633 千字

2003 年 12 月第 1 版

印数: 1-5 000 册

2003 年 12 月北京第 1 次印刷

ISBN7-115-11727-6/TP · 3648

定价: 42.00 元 (附光盘)

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

前 言

Visual Basic 6.0 作为 Microsoft 公司开发的 Visual Studio 6.0 套件的一部分，是一个功能丰富、方便实用的程序开发环境，它的一个很大的特色就是功能强大、编程简单的数据库操作能力。可视化的开发环境、丰富的数据库访问控件的支持使得 Visual Basic 成为数据库应用程序项目中重要的开发工具。

本书包括 4 篇内容，总共 14 章。全面详细地分析了数据库基本知识、数据库创建基础知识、Visual Basic 6.0 数据库基本操作技术和高级操作技术等方面的内容。全书从基础原理入手，以各种广泛应用的技术手段为主，最后讲解工程实践。通过本书的学习，读者既可以系统、深入地学习这几个方面的内容，又可以掌握实践技术，直接面向项目开发。

本书内容安排如下。

第 1 章 数据库基础，介绍数据库系统的基本概念和数据库技术的发展历史，并着重详细介绍了关系型数据库的构成和使用，最后介绍了几种常见的数据库产品。

第 2 章 用 Microsoft Access 创建数据库，主要介绍通过 Microsoft Access 创建桌面型数据库的方法。

第 3 章 SQL 语言，主要介绍 SQL 结构化数据库查询语言中常用的几条语句，是数据库操作的基本手段。

第 4 章 DAO 技术，介绍 Visual Basic 操作数据库最基本的对象库 DAO，通过 DAO 进行数据库连接、查询、更新、删除等操作。

第 5 章 ADO 技术，主要介绍目前最有前途的数据库访问对象库 ADO，介绍 ADO 的几个主要对象及其操作数据库的方法。

第 6 章 数据控件和数据感知控件，本章介绍 Visual Basic 中具有特色的数据控件和具有数据感知的控件，二者结合操作数据库的基本技术。

第 7 章 数据环境，主要介绍 Visual Basic 6.0 中自带的最新的一项数据库操作技术，以数据环境为基础可以为其他高层应用提供底层的数据库访问服务。

第 8 章 数据报表和打印，主要介绍在 Visual Basic 环境中利用 Data Report 控件创建各种类型的数据报表并打印的方法。

第 9 章 数据图表，主要介绍利用 MSChart 控件显示二维和三维的棒图、区域图、线形图、饼图等多种常用图表。

第 10 章 设计远程数据库，介绍如何通过 RDO 控件操作远程数据库，主要是 SQL Server 数据库的方法。

第 11 章 多层结构的数据库应用，主要介绍新式的数据库应用程序设计理念，通过多层结构的设计思路使得应用程序整体更清晰，扩展性更强，系统更稳定。

第 12 章 复杂数据控件，介绍各种水晶报表的使用和利用 Web 浏览器控件制作报表打印的方法。

第 13 章 解决数据安全性问题，主要介绍了 Microsoft Access 的安全性构造以及数字认

证 CA 方面的应用。

第 14 章 综合实例,通过详细介绍一个进销存管理系统的编制过程让读者具有一定的工程应用和项目开发知识。

本书每章内容都配以丰富而恰当的实例,可以直接应用于项目开发。实例的完整源代码保存在本书所附带的光盘里,以供感兴趣的读者学习和研究。

光盘的使用说明如下:

(1) 按章节建立目录,章节标题为目录名称,再以实例名称建立子目录,源代码放于其中。例如:

【第 5 章 用 ADO 操作数据库】

【5.5 VB 中使用 ADO 操作数据库】

(2) 将光盘代码拷贝到本机硬盘后,首先取消目录及其所属文件的“只读”属性,然后在 Visual Basic 6.0 中打开各实例的工程文件*.vbp 即可。

在本书编写过程中,得到了很多朋友和同事的大力支持,在此表示衷心的感谢;出版社的编辑、策划为本书也付出了相当的努力,在此也深表感谢。

由于时间较紧,再加上作者的能力和水平所限,本书的疏漏之处在所难免,希望广大读者给予批评指正。欢迎读者登录 <http://www.cs-book.com> 与作者交流。

编 者
2003 年 11 月

目 录

第 1 章 数据库基础	1
1.1 数据库的概念	1
1.1.1 数据库的发展历史	1
1.1.2 数据库系统	3
1.1.3 数据仓库	4
1.2 关系型数据库	4
1.2.1 数据模型	4
1.2.2 实体之间的联系	6
1.2.3 E-R 模型	7
1.2.4 规范化设计理念和方法	7
1.2.5 关键字	9
1.3 当前流行的几种数据库	12
1.3.1 Access 数据库	13
1.3.2 SQL Server 数据库	13
1.3.3 ORACLE 数据库应用服务器	14
1.3.4 MySQL 数据库	14
1.4 本章小结	15
第 2 章 用 Microsoft Access 创建数据库	16
2.1 Microsoft Access 简介	16
2.2 Access 的组成部分	17
2.3 创建一个新数据库	19
2.3.1 数据库需求分析	20
2.3.2 数据库结构设计	22
2.4 用 Access 自带工具创建数据表	23
2.4.1 创建新数据库	24
2.4.2 使用表向导创建表	25
2.4.3 使用“设计视图”创建表	26
2.5 设置表间关系	27
2.6 编辑管理数据记录	28
2.7 建立 Access 查询	29
2.8 Access 数据库导入 SQL Server 数据库	31
2.9 本章小结	32
第 3 章 使用 SQL 查询语句访问数据库	33
3.1 SQL 查询语言介绍	33

3.2 常用的 SQL 语句	34
3.2.1 综合查询语句 SELECT	35
3.2.2 构造新表语句 CREATE TABLE	39
3.2.3 修改已有表 ALTER TABLE	41
3.2.4 删除表 DROP TABLE	41
3.2.5 建立索引 CREATE INDEX	42
3.2.6 删除索引 DROP INDEX	42
3.2.7 插入语句 INSERT	42
3.2.8 修改数据 UPDATE	43
3.2.9 删除语句 DELETE	44
3.3 数据库权限控制 SQL	45
3.3.1 授权语句 GRANT	45
3.3.2 收权语句 REVOKE	46
3.4 嵌入式的 SQL 语句	47
3.5 本章小结	47
第 4 章 数据库访问对象 (DAO) 技术	48
4.1 DAO 访问 Access 数据库介绍	48
4.2 DAO 操作数据库的方式	49
4.2.1 DAO 操作数据库的步骤	49
4.2.2 Visual Basic 项目工程添加 DAO 引用	50
4.2.3 DBEngine 初始化数据库环境	50
4.2.4 打开数据库	51
4.2.5 打开数据集	51
4.3 DAO 数据库对象	52
4.3.1 DBEngine 对象	53
4.3.2 Workspace 对象	54
4.3.3 Connection 对象	55
4.3.4 Database 对象	57
4.3.5 Recordset 对象	58
4.3.6 一个使用 DAO 查询的 Visual Basic 程序	62
4.4 DAO 数据访问控制	67
4.4.1 Microsoft Jet (DAO) 的多客户模型	67
4.4.2 刷新 (Refresh) 和重新查询 (ReQuery)	80
4.4.3 DAO 对象的关闭	82
4.5 DAO 压缩 Access 数据库	82
4.6 DAO 实现数据库的同步	84
4.7 DAO 操作文本形式数据源	89
4.8 使用 ODBC API 操作数据库	93
4.9 本章小结	98

第 5 章 用 ADO 操作数据库	99
5.1 ADO 技术简介	99
5.1.1 ADO 技术的主要功能与特性	99
5.1.2 ADO 模型的主要组成	99
5.1.3 ADO 的编程模型	100
5.2 ADO 连接数据库	101
5.2.1 通过 ODBC 设置数据源	101
5.2.2 ADO 连接数据库的两种方式	102
5.3 ADO 对象详解	103
5.3.1 Connection 对象	103
5.3.2 Recordset 对象	112
5.3.3 Command 对象	125
5.3.4 Errors 集合	132
5.4 ADO 的数据构型 SHAPE	133
5.4.1 数据构型的意义	133
5.4.2 ADO 构型 SHAPE 的简单介绍	134
5.4.3 SHAPE 使用方法	135
5.5 VB 中使用 ADO 操作数据库	138
5.5.1 VB 环境对 ADO 组件的引用	138
5.5.2 VB 使用 ADO 操作数据库的例子	140
5.6 ADO 的物理机制	144
5.7 本章小结	144
第 6 章 数据控件和数据感知控件	146
6.1 VB 操作控件的方法	146
6.1.1 VB 自带控件和 Windows 标准控件	146
6.1.2 符合 COM 规范的可视化控件 (ActiveX 控件) 的注册	147
6.1.3 控件的属性、方法和事件	147
6.2 数据 (Data) 控件的使用	148
6.2.1 Data 控件功能介绍	148
6.2.2 Data 控件的属性、方法和事件	150
6.2.3 Data 控件的使用方法	154
6.3 数据感知 (Data-Aware) 控件	155
6.3.1 DataSource 属性	155
6.3.2 DataField 属性	156
6.3.3 DataFormat 属性	156
6.3.4 DataMember 属性	156
6.3.5 数据感知控件列表	157
6.3.6 数据感知控件的使用实例	157
6.4 ADO Data 控件	162

6.4.1	ADO Data 控件的功能	162
6.4.2	ADO Data 控件的使用方法	163
6.5	数据感知控件使用举例	168
6.5.1	DataList 控件和 DataCombo 控件	168
6.5.2	DataGrid 控件	169
6.5.3	DataList 和 DataCombo 控件的级联查询	173
6.6	VB 数据库编程方式综合比较	177
6.7	本章小结	179
第 7 章	数据环境	180
7.1	数据环境 (DataEnvironment) 对象	180
7.1.1	创建数据环境	180
7.1.2	Connection 对象	181
7.1.3	Command 对象	183
7.1.4	Command 对象的层次结构	187
7.1.5	Command 对象的字段映射	192
7.1.6	给数据对象添加 ADO 代码	194
7.2	数据环境的使用方法	194
7.2.1	数据环境使用的基本方法	194
7.2.2	将数据敏感控件绑定到数据环境中	195
7.2.3	拖动对象创建数据绑定控件	196
7.2.4	在程序中访问数据环境中的对象	196
7.3	Microsoft Hierarchical FlexGrid 控件的使用方法	198
7.3.1	Microsoft Hierarchical FlexGrid 功能介绍	199
7.3.2	将 Hierarchical FlexGrid 控件与数据环境绑定	199
7.3.3	Hierarchical FlexGrid 控件的详细设置	202
7.3.4	给 Hierarchical FlexGrid 控件增加编辑功能	205
7.4	本章小结	207
第 8 章	数据报表设计和打印	209
8.1	DataReport 的功能	209
8.1.1	DataReport 介绍	209
8.1.2	DataReport 的组成部分	210
8.1.3	DataReport 的创建方法	212
8.2	DataReport 的高级用法	217
8.2.1	数据报表中的统计	217
8.2.2	RptFunction 控件的使用	219
8.2.3	报表的分组	221
8.2.4	数据报表分页	223
8.2.5	数据报表添加日期、时间、页号和标题	224
8.2.6	导出数据报表	225

8.2.7	数据报表的打印	227
8.2.8	DataReport 的事件	228
8.3	数据报表设计器上控件的位置	230
8.4	数据报表实例	232
8.5	本章小结	234
第 9 章	数据图表	236
9.1	Microsoft Chart Control 介绍	236
9.2	Microsoft Chart 控件使用方法	237
9.2.1	项目添加 Microsoft Chart 控件	237
9.2.2	使用数组作 Microsoft Chart 控件的数据引擎	237
9.2.3	Microsoft Chart 控件的外观类型	240
9.2.4	Microsoft Chart 控件的 3D 特性	245
9.2.5	Microsoft Chart 控件与数据的绑定	247
9.2.6	Microsoft Chart 控件与 Excel 工作表的绑定	249
9.3	图表设计实例	251
9.4	本章小结	252
第 10 章	设计远程数据库应用系统	253
10.1	RDO 远程数据库访问控件	253
10.1.1	RDO 介绍	253
10.1.2	RDO 对象的创建和连接数据库	255
10.1.3	RDO 进行查询	264
10.1.4	RDO 执行存储过程	269
10.1.5	RDO 处理结果集	271
10.1.6	RDO 执行开放式的批更新	274
10.1.7	RemoteData 控件	277
10.2	RDO 使用实例	279
10.3	本章小结	283
第 11 章	多层结构数据库应用体系	286
11.1	Internet 应用程序	286
11.1.1	Visual Basic 在 Internet 应用程序中的使用	286
11.1.2	Internet 应用程序的本质	287
11.1.3	Visual Basic 开发 Internet 应用程序的优点	290
11.2	Windows DNA 体系结构	290
11.2.1	大型电子商务站点的特点	290
11.2.2	Windows DNA 体系结构介绍	292
11.2.3	分层式组件化结构	294
11.2.4	Windows DNA 体系结构原理	295
11.3	Windows DNA 分层式系统的构造实例	297
11.4	本章小结	311

第 12 章 复杂数据控件设计数据库程序	312
12.1 水晶报表 (Crystal Report) 控件使用	312
12.1.1 水晶报表 (Crystal Report) 控件的特点	312
12.1.2 用 Crystal Report 创建报表	313
12.2 Web 浏览器控件做报表设计	318
12.2.1 动态 HTML (DHTML) 的对象模型基础	318
12.2.2 基于 DHTML 的程序设计	323
12.2.3 Web 浏览器控件	326
12.2.4 用 Web 浏览器制作 HTML 报表	330
12.2.5 Web 浏览器报表用户控件使用示例	339
12.3 本章小结	345
第 13 章 解决数据安全问题	346
13.1 Access 数据库的密码体系	346
13.1.1 共享数据库的安全准则	346
13.1.2 单个 Access 数据库的密码保护	347
13.1.3 Access 的组 (GROUP) 和用户 (USER)	348
13.1.4 Access 的 ODBC 数据源权限设置	356
13.1.5 通过编程设置 Access 数据库密码	357
13.1.6 Access 数据库的加密	362
13.2 SQL Server 的安全系统	363
13.3 编写 SQL 语句不当引起的安全性问题	365
13.4 高级安全问题——数字认证 (CA) 系统	366
13.5 本章小结	369
第 14 章 综合实例——进销存管理系统	370
14.1 实例需求说明	370
14.1.1 软件需求描述	370
14.1.2 用户群体描述	370
14.1.3 功能列表	371
14.2 系统分析设计	371
14.2.1 系统架构	371
14.2.2 功能分析及模块设计	371
14.3 数据库设计	372
14.3.1 数据库设计说明书	372
14.3.2 E-R 模型设计	373
14.4 界面与报表设计	376
14.4.1 程序主界面	376
14.4.2 管理主界面	376
14.4.3 入库管理和出库管理界面	378
14.4.4 查询报表	378

14.5 业务逻辑层设计	379
14.6 模块调用	382
14.7 数据报表	387
14.7.1 数据环境设计	387
14.7.2 数据报表设计	388
14.8 运行结果演示	390
14.9 本章小结	391
附录	393

第 1 章 数据库基础

20 世纪 70 年代后, 在社会的信息化过程中出现了巨大的信息流和作为其载体的宏大的数据流, 为了适应和满足社会发展的需要, 必须采用新的技术和手段才能对这些数据进行处理。数据库技术就是最近 30 多年间才发展起来的大容量数据的规范化存储和处理技术。数据库技术对数据信息的收集、记载、分类、排序、储存, 以及提取、分析处理提供了完整的科学化的管理过程, 它的发明和应用极大地推进了电子计算机的应用进程。

本章将比较全面地介绍数据库的基础知识, 让读者了解数据库技术的发展历史, 系统地学习关系型数据库的基本元素和构造方法, 并简单介绍几种常用的数据库管理系统。

1.1 数据库的概念

数据库是人们快速全面管理数据信息的有效工具, 它是在计算机存储设备上合理存放的结构化的相关数据集合, 并与其他部分共同构成数据库系统。数据库在物理上是在计算机存储设备上存储了信息数据的实体, 一般以文件的形式存在。数据库系统在软件上包括数据库管理系统, 支持数据管理系统的操作系统, 各种语言编译程序, 用户的应用程序等。数据库系统相对于其他数据管理技术有如下特点: 数据共享、减少了数据的冗余、保持数据的一致性和数据独立性、有数据安全与完整性保障。

1.1.1 数据库的发展历史

20 世纪 50 年代, 计算机主要应用于科学计算方面, 对数据的管理仅依靠人工方式组织。也就是说, 程序员在进行软件设计的时候, 不但要针对某个应用问题设计算法, 还要对数据的结构、存储方法、输入输出等方面进行设计。因而, 数据与程序不具有独立性, 当数据存储结构发生改变的时候, 就必须修改相应的应用程序。如果几个程序都使用同一部分数据, 这一部分的数据就需重复存储, 因而数据冗余量较大。

20 世纪 50 年代末至 60 年代中期, 由于计算机的操作系统不断发展, 以及磁盘和磁鼓等大容量辅助存储器的出现, 为计算机数据处理进入文件系统提供了条件。在这个阶段里, 数据以文件的形式进行组织, 并能长期保留在外存储器上, 用户能对数据文件进行查询、修改、插入和删除等操作。然而数据冗余量大的缺点并没有得到根本性的改变, 数据还是面向应用, 当不同的应用程序所需要的数据中有部分数据甚至全部数据是相同的, 也必须建立各自的数据文件, 因而不能共享相同的数据。正是由于这些相同数据的重复存储、各自管理, 给数据的修改和维护带来了困难, 往往容易造成数据的不一致性。

20 世纪 60 年代后期, 计算机的硬件和软件都有了进一步发展, 信息量的爆炸式膨胀带来了数据量急剧增长, 为了解决日益增长的数据量带来的数据管理上的严重问题, 数据库技术逐渐发展和成熟起来。

- 网状数据库

最早出现的数据库是网状数据库。网状模型中以记录为数据的存储单位。记录包含若干

数据项。网状数据库的数据项可以是多值的和复合的数据。每个记录有一个惟一标识它的内部标识符,称为码(DatabaseKey,DBK),它在一个记录存入数据库时由系统自动赋予。DBK可以看作记录的逻辑地址,可作记录的替身或用于寻找记录。网状数据库是导航式(Navigation)数据库,用户在操作数据库时不但要说明做什么,还要说明怎么做。例如在查找语句中不但要说明查找的对象,而且要规定存取路径。

世界上第一个网状数据库管理系统也是第一个数据库管理系统是美国通用电气公司 Bachman 等人在 1964 年开发成功的 IDS (Integrated Data Store)。在数据库发展史上,网状数据库占有重要地位。

● 层次数据库

层次型数据库管理系统是紧随着网络型数据库而出现的。现实世界中很多事物是按层次组织起来的。层次数据模型的提出,首先是为了模拟这种按层次组织起来的事物。层次数据库也是按记录来存取数据的,层次模型是树状的。层次数据模型中最基本的数据关系是基本层次关系,它代表两个记录型之间一对多的关系,也叫做双亲子女关系(PCR)。数据库中有且仅有一个记录型无双亲,称为根节点。其他记录型有且仅有一个双亲。在层次模型中从一个节点到其双亲的映射是惟一的,所以对每一个记录型(除根节点外)只需要指出它的双亲,就可以表示出层次模型的整体结构。

最典型的层次数据库系统是 IBM 公司的 IMS (Information Management System),这是 IBM 公司研制的最早的大型数据库系统程序产品。

● 关系型数据库

网状数据库和层次数据库已经很好地解决了数据的集中和共享问题,但是在数据独立性和抽象级别上仍有很大欠缺。用户在对这两种数据库进行存取时,仍然需要明确数据的存储结构,指出存取路径。而后来出现的关系数据库较好地解决了这些问题。关系数据库理论出现于 20 世纪 60 年代末到 70 年代初。1970 年,IBM 的研究员 E.F.Codd 博士发表《大型共享数据库的关系模型》一文提出了关系模型的概念。后来 Codd 又陆续发表多篇文章,奠定了关系数据库的基础。关系模型有严格的数学基础,抽象级别比较高,而且简单清晰,便于理解和使用。但是当时也有人认为关系模型是理想化的数据模型,用来实现数据库是不现实的,尤其担心关系数据库的性能难以接受,更有人视其为当时正在进行中的网状数据库规范化工作的严重威胁。为了促进对问题的理解,1974 年 ACM 牵头组织了一次研讨会,会上开展了一场分别以 Codd 和 Bachman 为首的支持和反对关系数据库两派之间的辩论。这次著名的辩论推动了关系数据库的发展,使其最终成为现代数据库产品的主流。

关系数据模型提供了关系操作的特点和功能要求,但不给数据库系统的语言给出具体的语法要求。对关系数据库的操作是高度非过程化的,用户不需要指出特殊的存取路径,路径的选择由数据库系统的优化机制来完成。Codd 在 20 世纪 70 年代初期的论文论述了范式理论和衡量关系系统的 12 条标准,用数学理论奠定了关系数据库的基础。Codd 博士也以其对关系数据库的卓越贡献获得了 1983 年 ACM 图灵奖。

关系数据模型是以集合论中的关系概念为基础发展起来的。关系模型中无论是实体还是实体间的联系均由单一的结构类型——关系来表示。在实际的关系数据库中的关系也称表。一个关系数据库就是由若干个表组成。

说到关系型数据库,不能不提 SQL (Structured Query Language, 结构化查询语言)的发展历史。

1974年, IBM的Ray Boyce和Don Chamberlin将Codd关系数据库的12条准则的数学定义以简单的关键字语法表现出来, 并提出了SQL (Structured Query Language) 语言。SQL语言的功能包括查询、操纵、定义和控制, 是一个综合的、通用的关系数据库语言, 同时又是一种高度非过程化的语言, 只要求用户指出做什么而不需要指出怎么做。SQL集成实现了数据库生命周期中的全部操作。自产生之日起, SQL语言便成了检验关系数据库的试金石, 而SQL语言标准的每一次变更都指导着关系数据库产品的发展方向。

现在, 任何一个数据库管理系统都需要通过SQL语言来查询和管理, 虽然各个不同的数据库厂家都对SQL标准做了一定程度的增强和定制。

• 未定型的后关系数据库

随着Internet的发展, 数据库面临着信息爆炸的挑战, 而数据对象也不仅仅限于文本数据等简单的数据类型, 还需要描述和保存各种各样的媒体和真实事件。在这种需求下, 后关系型数据库应运而生。

所谓后关系数据库, 实质上是在关系数据库的基础上融合了面向对象技术和Internet网络应用开发背景的发展。它结合了传统数据库如网状、层次和关系数据库的一些特点, 以及Java、ActiveX等新的编程工具环境, 并应用了新兴的XML技术, 适应于新的以Internet Web为基础的应用, 开创了关系数据库的新时代。后关系型数据库的主要特征是将多维处理和面向对象技术结合到关系数据库上。这种数据库使用功能强大而灵活的对象技术, 将经过处理的多维数据模型的速度和可调整性结合起来。

• 异构数据库系统

由于各种原因, Internet上的数据库系统不少是分布、异构的。Internet上大量信息必须通过数据库系统才能有效管理。那么, 在Internet环境下分布式海量信息情况下如何建立合理高效的海量数据库, 成为我们亟待解决的问题。针对目前关系型数据库占据了大多数市场的情况, 要实现网络环境下的海量信息共享, 就必须联合各个异构数据库, 使得数据库之间能够通过主动式的超文本链接, 实现相互连接, 使得交叉引用的数据可以很容易地被检索到。

相互关联的数据库可以很容易地被归纳在一起、创建一个单一的虚拟数据库, 也叫做异构数据库系统。

目前, 异构数据库系统的集成以及建立此基础之上的数据仓库、数据挖掘已经成为网络数据库技术研究的重点之一。国际上, 著名的数据库厂商, 包括Oracle, Microsoft等公司也将异构数据库系统——数据仓库技术作为竞争的焦点, 研究如何将原来传统的、可能分布于各地的多个关系数据库集成起来, 进行改进和发展, 形成虚拟异构数据库系统和数据仓库, 更好地为企业信息化、电子商务服务。

1.1.2 数据库系统

数据库管理系统(Database Management System, DBMS)是建立、使用和维护以及管理数据库的软件系统。数据库系统是数据库、数据库管理系统(DBMS)、用户应用程序和支持它们的计算机系统的总称。

简单地说, 数据库管理系统实质就是管理数据库的软件, 通常包括一系列的软件套件, 包括物理存储管理、访问量管理、网络通信、用户管理等。对于数据库管理系统的基本要求是能够及时准确地满足多个用户的并发存取操作, 另外还有能够保证事务的完整性、时刻保

持数据的一致性、要求在硬件和操作系统正常工作的情况下独立地并发操作互不影响、不丢失数据。

目前处于主流地位的数据库管理系统主要有 IBM 公司的 DB2/DB2 Universal Database、Informix 公司的 InformixIDS2000、Sybase 公司的 Sybase ASE、Oracle 公司的 Oracle 8i/9i 系列、Microsoft 公司的 SQL Server 7/2000 系列、Microsoft 公司的 Access 系列等。

1.1.3 数据仓库

数据库技术经过几十年的发展和广泛应用,社会各行各业的大量信息和数据多年积累使得数据库在不断膨胀。如何从海量的数据中提取,检索出有用的信息就越来越显得重要。因此,数据仓库技术,包括数据挖掘技术近年来成为数据库技术发展的热门。

随着 Client/Server 技术的成熟和并行数据库的发展,信息处理技术的发展趋势是从大量的事务型数据库中抽取数据,并将其清理、转换为新的存储格式,即为决策目标把数据聚合在一种特殊的格式中。随着该过程的发展和不断完善,这种支持决策的、特殊的数据存储即被称为数据仓库(Data Warehouse, DW)。

Bill Inmon, 数据仓库之父,对数据仓库(Dataware House)的定义是这样的,面向主题的、集成的、不易失的、时变的数据集合,用于支持管理决策。这里面关键的几个概念是主题、时变和决策。主题是数据归类的标准,每个主题对应一个客观分析领域,如客户、商店等,它可为辅助决策集成多个部门不同系统的大量数据。数据仓库包含了大量的历史数据,经集成后进入数据仓库的数据是极少更新的。数据仓库内的数据时限为 5 年至 10 年,主要用于进行时间趋势分析。数据仓库的数据量很大,一般为 10GB 左右。它是一般数据库(100MB)数据量的 100 倍,大型数据仓库达到 TB 级。

数据仓库主要应用在两个方面即使用浏览分析工具在 DW 中寻找有用的信息和数据仓库系统支持在 DW 上的应用,形成决策支持系统(DSS)。

数据仓库理论和技术的发展是数据库技术应用 30 多年来的升华,它为人们对日益膨胀的海量信息的处理要求提供有力的手段。

1.2 关系型数据库

上一节介绍了数据库技术发展的历史以及发展趋势,相应描述了各个发展阶段各类型数据库的一些特点。下面将对目前数据库技术中应用最广泛的关系型数据库的基本概念、必要元素和相关法则作介绍,并使用实际的建立关系型数据库的例子向读者阐明关系型数据库建立的过程以及说明建库的规范。关系型数据库最重要的特点就是“实体-关系”的模型,通称为 E-R 模型(Entity-Relationship)。本节还将向读者介绍几个建立 E-R 模型的实用软件工具,这些工具也是笔者在多年的数据库管理经验中筛选出的最好用的 E-R 模型工具,希望能对读者有所帮助和启发。

1.2.1 数据模型

数据模型是描述数据、数据之间的联系的结构模式。数据模型是数据库系统中用于提供

信息表示和操作手段的形式构架。不同的数据模型提供了模型化数据和信息不同工具，根据模型应用的不同目的，可以将模型分为两类或两个层次，一是概念模型（也称信息模型），一是数据模型（层次模型、网状模型和关系模型）。前者是按用户的观点来对数据和信息建模，后者是按计算机系统的观点对数据建模。

1. 概念模型

概念模型是对客观事物及其联系的抽象，用于信息世界的建模，它强调其语义表达能力，以及能够较方便、直接地表达应用中各种语义知识。这类模型概念简单、清晰、易于被用户理解，是用户和数据库设计人员之间进行交流的语言。这种信息结构并不依赖于具体的计算机系统，不是某一个 DBMS 支持的数据模型，而是概念级的模型，然后再把概念模型转换为计算机上某一 DBMS 支持的数据模型。

2. 数据模型

(1) 层次模型

层次模型是用树型结构来描述实体及它们之间的联系方式。下面用一个能反应中学班级设立情况的树形结构来说明，所有的班级结点是实体，结点之间的通路是联系。它的特征是：有且仅有一个结点向上没有联系，称为根结点（如中学结点）；有若干个结点向下没有联系，称为叶结点（如初一级、初三级、初一1班、初一2班、初一3班、高一级、高二级和高三级结点），其余结点向上有惟一的联系，向下可能有多个联系，称为中间结点（初中部、高中部和初二级结点）；从根结点至任何一个中间结点或叶结点，有且仅有一条通路，如图 1-1 所示。

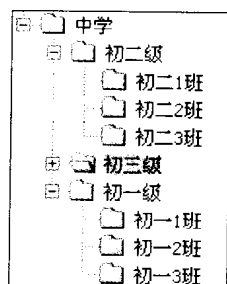


图 1-1 中学层次模型示意图

(2) 网状模型

网状模型又称网络模型，是用网络结构表示实体及它们之间的联系。用城市之间的公路来做例子，以城市结点表示实体，以城市之间的公路表示实体之间的联系。它的特征是：至少有一个结点向上有两个或多个联系；两结点之间也可以有两个或多个联系。网状模型描述多对多的网络结构，能表示多种复杂的联系。

(3) 关系模型

关系模型是用平面二维表格表示实体之间的联系，它沿着行和列两个方向把数据排列成二维表，用以描述实体及其联系。行称为记录，列称为字段。表 1-1 所示为一张描述公司员工档案的关系二维表，表的名称“员工档案”称为关系名；二维表的列表头名称“员工编号”、“姓名”、“性别”、“年龄”为字段名。