Delphi 宝库

Win 32 数据库开发指南

[美] Warren Rachele 著

陈 池 陈 峦 陈建平 译

郑 铭 审校

电子科技大学出版社

需要全本请访问 www.ertongbook.com

内 容 提 要

全书包含三个不同的部分，各部分内容又是密切相关的。第一部分讲解了关系型数据库系统的理论实现。第一章和第二章较为深入地讨论了关系型数据库的理论和实现。完全理解关系型数据库及其优点对开发人员来说必定受益非浅。第三章让你了解 SQL 语言的使用。SQL 语言的使用对信息查询和 SQL 数据库操纵等等都十分重要。

第二部分为本书的核心，着重讲了 Delphi 数据库工具。第四章为 Delphi 数据库功能主体，即 Borland Database Engine。本章除了讨论引擎的工作原理外，还讲了它的支持工具。到第五章才涉及到 Delphi 组件，讲述数据集对象的一般特点，以及每种数据提取组件及其使用。第六章让开发人员了解所有数据意识接口组件，它们是用户与数据表进行对话的工具。第二部分的最后一章，第七章，是关于数据库信息转化为报表的不同应用形式的论述。

第八、九两章和附录 A 是本书的第三部分，它们是关于 API 使用、API 函数参考以及各种编程技巧的综合讨论。

The Tomes of Delphi: Win32 Database Developer's Guide

Warren Rachele

Copyright©1999,Wordware Publishing ,Inc
All Rights Reserved
2320 Los Rios Boulevard Plano,Texas 75074

Reprinted in Chinese by University of Electronic Science and Technology of China PRESS
under a license granted by Wordware Publishing ,Inc. Plano,Texas 75074 U.S.A

本书中文版由 Wordware Publishing,Inc.授权电子科技大学出版社出版。
四川省版权局著作权合同登记章 图字 21-1999-039 号

前　言

感谢各位购买本书！

尽管数据库方面的书层出不穷，但人们对数据库程序开发技巧仍然未能深刻理解。大多数开发人员都是把一些组件和程序连接在一起，尤其是像 Delphi 这样的提供了比较完善的数据库组件的开发工具。他们认为数据库就是在表格上释放一些控件，把它们指向合适的结构，这样就成了数据库。

但数据库远非如此！这只能算是数据提取操作，因为它们通过读写文字文件、处理存储变化就能获得。而数据库包括数据、数据作用方式、操纵数据事实的操作、处理数据输入输出操作的过程和数据接口工具，甚至包括数据形成信息的数据显示和报表生成等等。因此数据库程序必须在掌握了上述所有内容的情况下，通过仔细设计和精心组织才能取得成功。

本书一开始就深入到 Delphi 数据库编程的内幕，而不是单纯地讲解数据库组件和工具的使用。前三章介绍关系型数据库的概念。我认为花些时间理解它们是值得的，有助于编写程序时使用，从而获得较大的收益。

Delphi 是非常优秀的编程工具，依笔者管窥，认为她是目前世界上最好的开发工具，我是极力推崇她的。Borland Database Engine 结构提供了最大的自由度和最方便的工具。如果你花些时间理解 BDE 的结构和功能，你就会为她提供的功能而叹为观止。

Delphi 工具时时更新，功能也越来越强大。所以，本书在编写时并不局限于 Delphi 的某一版书，书中讨论的范围较广，所以本书必将为你长期使用。

著　者

目　录

第一部分　关系型数据库

第一章 关系型数据库介绍 2

1-1　关系模型 2
1-2　什么是数据库 3
1-3　创始人 Codd 4
1-4　数据结构 5
1-5　数据完整性 7
1-6　数据操作 8
1-7　总结 10
1-8　展望 10

第二章 逻辑和物理数据库设计 11

2-1　数据库图解 11
2-1-1　实体 12
2-1-2　关系 12
2-1-3　主键（Primary Key） 14
2-1-4　替换键（Alternative Key） 14
2-1-5　客键（或称外键，Foreign Key） 14
2-1-6　商业准则 15
2-2　个例研究 16
2-2-1　参与实体 16
2-2-2　各种属性 17
2-2-3　找关系 17
2-3　规范化 20
2-3-1　第一范式（The First Normal Form，简写为 1NF） 20
2-3-2　第二范式（The Second Normal Form，2NF） 21
2-3-3　第三范式（The Third Normal Form，3NF） 22
2-3-4　进一步规范化 22
2-3-5　Boyce-Codd 范式（Boyce/Codd Normal Form，BCNF） 23
2-3-6　第四范式（The Fourth Normal Formal，4NF） 23
2-3-7　第五范式（The Fifth Normal Form，5NF） 23
2-3-8　区键范式（Domain-key Normal Form，DK/NF） 24

2-4 规范化个例研究 24
2-5 逻辑设计到物理设计的转化 26
2-5-1 字段 26
2-6 总结 27
2-7 展望 28

第三章 结构化查询语言 29

3-1 什么是 SQL 29
3-1-1 学习 SQL 30
3-1-2 SQL 子语言 30
3-1-3 Local SQL 30
3-2 数据操纵语言指令 31
3-2-1 SELECT 语句 32
3-3 数据定义语言（DDL）命令 42
3-3-1 CREATE TABLE 语句 42
3-3-2 ALTER TABLE 语句 44
3-3-3 DROP TABLE 语句 44
3-3-4 CREATE INDEX 语句 44
3-3-5 DROP INDEX 语句 44
3-4 总结 45
3-5 展望 45

第二部分 Delphi 数据库工具

第四章 Borland Database Engine 和数据库应用 47

4-1 结构 47
4-2 数据库驱动器 48
4-3 配置 Borland Database Engine 49
4-3-1 配置标准数据库 49
4-3-2 配置 ODBC 数据库 50
4-3-3 配置数据库引擎 51
4-4 数据库开发平台 54
4-4-1 建立表 54
4-5 BDE 应用程序的销售 58
4-6 数据库浏览器（Database Explorer） 59
4-6-1 数据库浏览器接口（Database Explorer Interface） 59
4-7 数据字典 60
4-7-1 字典菜单 65
4-7-2 引入 SQL 限制 65

需要全本请访问 www.ertongbook.com

4-8 总结 65
4-9 展望 66

第五章 Delphi 数据提取 67

5-1 数据集（Dataset）对象 67
5-1-1 数据集对象（TDataSet） 67
5-1-2 数据集状态 68
5-2 表组件（TTable） 69
5-2-1 关键属性 69
5-2-2 表事件 72
5-3 查询组件（TQuery） 74
5-3-1 关键属性 74
5-4 数据源（DataSource）组件（TDataSource） 76
5-4-1 主要属性 76
5-5 使用数据源、表和查询组件 78
5-5-1 Datasrc 工程 78
5-5-2 华尔街观景（Wall Street Watch） 79
5-5-3 定单工程 82
5-6 存储过程组件（TStoredProc） 83
5-6-1 关键属性 84
5-6-2 TStoredProc 事件 85
5-6-3 应用存储过程 85
5-7 数据库组件(TDatabase) 88
5-7-1 主要属性 88
5-7-2 TDatabase 事件 90
5-7-3 使用 Database 组件 90
5-8 会话（Session）组件（Tsession） 91
5-8-1 主要属性 91
5-8-2 TSession 事件 92
5-9 块移动（BatchMove）组件（TbatchMove） 92
5-9-1 主要属性 93
5-9-2 应用 BatchMove 组件 95
5-10 更新 SQL（UpdateSQL）组件（TUpdateSQL） 97
5-11 嵌入表（NestedTable）组件（TnestedTable） 98
5-12 总结 99
5-13 展望 99

第六章 Delphi Data-Aware 组件 100

6-1 共同属性 100

6-1-1 主要属性 100
6-2 DBGrid 组件 102
6-2-1 主要属性 102
6-2-2 DBGrid 事件 102
6-3 DBNavigator 组件 104
6-3-1 主要属性 104
6-3-2 DBNavigator 事件 105
6-3-3 应用 DBGrid 和 DBNavigator 组件 105
6-4 DBImage 组件 109
6-4-1 主要属性 110
6-5 DBMemo 组件 111
6-5-1 主要属性 111
6-5-2 DBMemo 事件 112
6-5-3 使用 DBMemo 和 DBImage 组件 112
6-6 DBText 组件 114
6-6-1 主要属性 115
6-7 DBEdit 组件 115
6-7-1 主要属性 115
6-8 DBCheckBox 组件 116
6-8-1 主要属性 116
6-9 DBRichEdit 组件 117
6-9-1 主要属性 117
6-9-2 DBRichEdit 事件 118
6-9-3 使用 DBText、DBEdit 和 DBRichEdit 119
6-10 DBListBox 组件 122
6-10-1 主要属性 122
6-10-2 DBListBox 事件 122
6-11 DBComboBox 组件 123
6-11-1 主要属性 123
6-11-2 DBComboBox 事件 123
6-12 DBLookupListBox 和 DBLookupComboBox 组件 124
6-12-1 主要属性 124
6-12-2 查找字段 125
6-12-3 应用 DBLookUpListBox 和 DBLookUpComboBox 126
6-13 DBRadioGroup 组件 128
6-13-1 主要属性 128
6-14 DBCtrlGrid 组件 130
6-14-1 主要属性 130
6-14-2 DBCtrlGrid 事件及其相关方法 132

6-14-3 使用 DBCtrlGrid 132
6-15 总结 135
6-16 展望 135

第七章 报表和打印 136

7-1 报表分析 136
7-2 报表格式 138
7-3 提出报表 138
7-4 使用 TPrinter 对象 138
7-5 列式报表 142
7-6 组件报表 149
7-7 格式报表 152
7-8 标签报表 154
7-9 交叉制表报表 156
7-10 外部报表生成器 162
7-11 总结 162

第三部分 周密的应用

第八章 诀窍与技术 164

8-1 Access 数据库 164
8-1-1 开放式数据库连接——ODBC 164
8-1-2 建立别名 165
8-2 Delphi 中的查找 168
8-2-1 他们称我为查找专家 169
8-3 建立表索引 171
8-3-1 次序仍没变 171
8-4 建立表 172
8-4-1 在设计时建立表 172
8-4-2 在运行时建立表 173
8-5 在运行时使用 SQL 174
8-5-1 从另一字符串列表对象中装入 SQL 属性 174
8-5-2 从外部数据源中取得 SQL 属性 175
8-5-3 直接输入 SQL 字符串列表 177
8-6 DBChart 组件 178
8-7 替换 Borland 数据库引擎 181
8-7-1 Local Interbase Server (LIBS) 181
8-7-2 Advantage Database Server 182
8-7-3 FlashFiler 182

第九章 Borland 数据库引擎 API 183
9-1 API 结构 183
9-2 使用 API 184
9-2-1 初始化 BDE 184
9-2-2 打开数据库对象 184
9-2-3 把数据库对象指向包含表的子目录 185
9-2-4 为临时对象设置子目录 185
9-2-5 打开一个表，创建一个光标对象 185
9-2-6 获得表的属性 185
9-2-7 为记录缓冲区分配内存 185
9-2-8 把记录指针放到适当的记录上 186
9-2-9 从光标处开始检索记录 186
9-2-10 从记录中取得字段 186
9-2-11 善后处理 186
9-2-12 包装 dBASE 表 191
9-2-13 包装 Paradox 表 192
9-2-14 检查 Paradox 表 194
9-3 API 函数 195
9-3-1 功能或格式函数 195
9-3-2 光标函数 196
9-3-3 数据提取函数 197
9-3-4 数据库函数 198
9-3-5 日期/时间/数字函数 198
9-3-6 环境函数 199
9-3-7 错误函数 200
9-3-8 索引函数 200
9-3-9 锁定函数 200
9-3-10 查询函数 201
9-3-11 会话函数 201
9-3-12 表函数 202
9-3-13 交易函数 202
9-4 总结 203
附录 A BDE API 函数速查 204
函数参考 204

第一部分　关系型数据库

本部分包括：

- 第一章　关系型数据库介绍
- 第二章　逻辑和物理数据库设计
- 第三章　结构化查询语言

第一章　关系型数据库介绍

本章内容： 1. 关系模型；2. 什么是数据库；3. 创始人 Codd；4. 数据结构；5. 数据完整性；6. 数据操作。

现代信息处理中，数据库无处不在。数据收集在几乎所有的票据应用中。数据库表现在大量数据的自动收集及其随之而来的归类、查找和维护所有记录的能力。记录的增多迫使广泛应用的计算机硬件设施飞速发展。数据功能强大，各种组织都意识到信息资源控制不仅具有竞争性，更危系着生死存亡。

数据库通过书面层级结构建模而得，经计算机化后以代替书面文件。数据库经历了无数表现形式，以期找到最有效地维持数据集完整性和高效性的方法。20 世纪 70 年代初，Codd（E. F. Codd）博士提出了一种数据库模型，它是建立在被称作关系的数学对象和能应用于这些关系的处理过程基础上的。这种智能化的策略就成了所谓的关系数据库模型（Relational Database Model），从而引发了数据管理软件设计和开发的革命。

同 Borland Database Engine(Borland 数据库引擎，BDE)紧密相连的 Delphi 给开发人员史无前例的自由，让他们在几乎各种操作平台上设计和开发同数据库资源共同作用的软件。为获得这种自由，人们付出了沉重的代价。数据库开发人员要想从功能强大的操作环境中获益，最好是把这种自由和关系模型结合起来。在投入到简单的 Delphi 编程前，不妨着眼于这种结合的理论和推理过程上，打下牢固的知识基础。数据库开发说来容易做起来却繁。本书前三章将讲述一些背景知识，为你提供高效设计选择以及 Borland 公司提供的实施选择，让你的软件从各种工具中获取最大收益。

1-1　关 系 模 型

学习本书之前，要清楚地建立起关系模型的概念。建立在数学精度概念上的关系模型的定义准则描述了一种在保持数据完整性的首要前提下，能在各种条件下都能取得一致和精确结果的系统。值得一提的是，关系模型并不是数据存储的物理实施方法。应用者具体能多紧密地应用这准则是由准则决定的。各种关系型数据库管理系统及其开发工具在应用关系准则上千差万别，尽管各自都缘于从储存和检索机制中提高性能。Delphi 作为一种开发工具，和 BDE 这一具体数据库引擎，给应用该工具进行数据库开发极大的自由度。理解本文中的准则，应用关系模型和驱动数据库的物理引擎的内部功能所提供的数据完整性概念，就会给你追求数据完整的努力以回报。

需要全本请访问 www.ertongbook.com

1-2 什么是数据库

“数据库”这个术语有各种不同的应用方式。许多应用都特指某种数据库。我们不会理会那些促销术语，简单说来，数据库就是相关数据的集合。关系性是组成集合的数据项或记录之间的自然的或强加的关系。

图 1-1 是一个票据文件。各票据之间是通过它们本身表示的信息和票据中共有的信息而相关的。

计算机化的数据库是这种文件的电子表示，是数据记录的电子储藏室。每一个记录表达一个模式化项目的一套独立的数据元素。

最简单的计算机化数据库是包含了所有数据的单个文件。同存储所有记录的书面文件类似，使用该文件要进行如下操作：

(1) 给数据库中加入新的空记录；

(2) 在数据库中指定位置上插入若干记录；

(3) 删除不需要的记录；

(4) 检索数据库中某些记录；

(5) 更改数据库中一些记录。

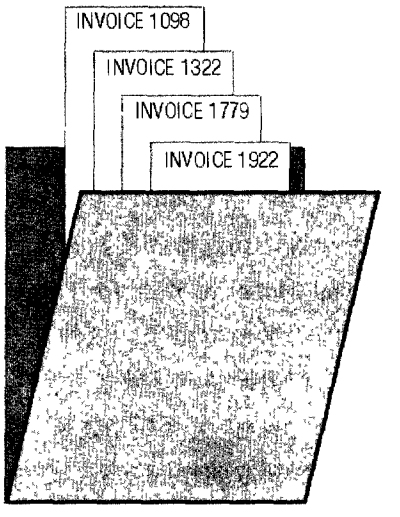


图 1-1 第一个数据库系统模拟的书面文件系统

图 1-2 是我们朋友姓名的简单数据库。表中每条记录表达了一个关于我们认识的朋友的各信息条目。尽管该数据库简洁，但它却给开发人员和用户带来这样或那样的问题。例如：因为我们需把关于某朋友的所有事例记录在一个单个文件中，更改变化的情况时就大大影响了数据的完整性。图上有两个记录来表示 Geurii 家公用的电话号码。因为 Tom、Sharon 都是我们的朋友，所以要求也必须用两个记录来表示他们的名字。修改两者共用的电话号码就需两个不同的操作：一个是改 Tom 的，另一个是改 Sharon 的。两人电话数据的重复就可能造成并排两者都改正，或者两者都未改正。从而导致数据不完整。下次我们查这电话数据库打电话给 Sharon 时，我们查得的就可能是一个不相关的电话号码了。

FRIENDS

Geurii	Sharon	722 Boston Ln	909 221-4456
Rollins	Hank	PO Box 89	818 986-2091
Geurii	Tom	722 Boston Ln	909 221-4456
Wayne	Marion	Warhawk Dr	303 458-8095

图 1-2 FRIENDS 简单文件数据库

同书面文件遇到的问题一样，这种简单文件也会产生大错。如果记录是以姓归档的，简单的错误拼写就使本来匹配的记录分离开来。上面的例子中 Tom 的姓可能正确的写成 Geurii，但 Sharon 的姓却很容易写为 Guerii。这种由于单个更新造成的错误很显然不只出现在简单文件上，而是出现在任何数据更新过程中。如果某次只凭姓来修改地址，就使得

其中一人的地址更改正确，但却导致寄给另一人的信件寄到老地方而可能丢失。

简单的文件数据库实施与记录到磁盘上的物理存储方法紧密相关。由此，用户对表中的存取操作与实际的实施方法相同。在使用者看来，这种简单文件是通过逐个记录存取方法实现的，数据库的获取方法是各记录分开获得，而并没注意到这些相关记录的联系。

这种简单文件数据库是从纸式记录保存向电子数据储存跨越的逻辑中介解决办法。跟随磁盘数据的错误书面记录问题，以及技术使该问题出现的速度，要求采用一种新方法。

1-3 创始人 Codd

1968 年，当 Codd（E.F. Codd）博士仍在 IBM 当研究员时，他就开始探求运用数学均匀性、一致性到无规则的数据库管理领域的可能性。他的研究结果著成了经典论文："A Relational Model of Data for Large Shared Data Banks"，发表于 1970 年。文中勾勒了数据库管理的关系模型思想。这种开先河的原理给人们带来了一种全新的视野，即让人们在数学集合论和谓词逻辑基础之上看待数据。

Codd 提出的关系模型之所以新，是因为他采用了描述数据实际存在的形式准则及用于数据操作和获取的相应方法。这种理论化的数据管理方法不同于当时采用的方法。Codd 强烈地认为数据逻辑表现和其物理储存方法之间有极大的区别。他不想两者之间相互决定。关系模型最重要的也是最有持久意义的贡献是一种概念，即对用户来讲所有数据都好像存在于关系中，也就是数学上说的表中。

通过与表中数据表示对比，Codd 提出了好几种数据管理程序中数据获取方法，使获取的数据最大可能地保持一致、重复。数学关系性质使数据集合获取法与简单文件的逐条记录扫描的方法背道而驰。其集合方法由一个关系的两个关键特征实现。第一，关系中每一行代表数据集中的独立实体（Entity）；第二，关系中每一列表示行中的一种属性(Attribute)。这两种特性的联合确保数据管理系统查询和更新符合逻辑，从而不需要访问每一个记录来决定其内容。关系中每个记录的真正的惟一就不需要采用各种准则来处理众多数据。

关系模型的理论基础很快就成了数据管理的标准。正确的概念和一致的结果使得关系型数据库及其组成部分成了现今几乎所有数据库管理系统的首选。鉴于本章后面将详细讲述关系系统，故先总结关系型数据库模型的概念整体。

数据库管理的关系模型是应用数学逻辑和可预测性的数据集管理的智能化概念，这是因为：

(1) 模型阐述了数据的限制和操作方法，从而可获得数据完整性和管理的能动性；

(2) 数据在使用者看来是储存在关系中，或表中；

(3) 所有对基表或不变表的操作都产生新表。由于所有数据的获取、检索和操作运算都产生新表，所以对使用者来说就维持了一致性；

(4) 关系模型注意了数据的三个方面：数据结构、数据完整性、数据操作。

1-4 数 据 结 构

概念上讲，关系型数据库中数据对用户来说是采取的表或关系的形式。关系是二维表，由一套有名列和未知数目的行组成。每一有名列代表一个单一的区（Domain），每一行代表由列确定的实体的一种属性。关系表的缜密性的、学术性的描述可简单、完整地用关系的六个独特性质来解释。

关系表与简单文件的区别在于关系表中采用了定义表的性质的一些准则，简单文件数据库缺少这些定义特征。六个准则如下：

(1) 各列单值；

(2) 每列实体来自同一个区；

(3) 各行惟一；

(4) 列序无意义；

(5) 行序无意义；

(6) 各列命名惟一。

尽管这种严密的结构不太容易正确实施，但却很容易被证明。关系模型的这些准则的应用会给用户在数据使用和结果准确性方面增加灵活性。

1．各列单值

这一性质有首要的作用，它决定关系型数据库的后属性质和方法。这条准则意味着表中无重复的组(Group)，一个重复的组代表着一实体的重复。数据库模型的高效、灵活、完整就是由这条严厉的标准决定的。这条单值准则的应用确保表中每一行是惟一的，从而数据操作和查询就有数学精度，也无不明区域需解释。

表 1-1 的 VENDOR 表中显示了各供货商所提供的各项目。

每一行代表一供货商可提供的商品。这种设计是有问题的，因为要想考虑各种可能项目的列的存取逻辑是不现实的，查找某个项目及其提供商也会显得特别困难。较好的方法是通过如表 1-2 所示的关系范例来实施。在这个关系（表）中，没有重复列。要把某一品名与提供商相连就很简单了。

表 1-1 有重复列的 VENDOR 表

VENDOR	ITEM	ITEM2	ITEM3
Kingfish, Inc.	Kipper	Smelt	Sardine
Tidepool Company	Crab	Clam	
The Crab Walk	Tuna	Shark	Swordfish
Devil Ray Foods	Oyster	Abalone	Clam

2．每列实体来自同一个区

区是一列中抽取出的数据的值集，决定整个可接受数据元素的范围。例如：Enployee_Id 这列所成的区是有效的雇员 ID 号码范围。不属于该值集的项目，如雇员的姓或超出了该有

效范围的 ID 号都不在这个列区。

表 1-2　移去重复列的 VENDOR 表

VENDOR	ITEM
Kingfish, Inc.	Kipper
Kingfish, Inc.	Smelt
Kingfish, Inc.	Sardine
Tidepool Company	Crab
Tidepool Company	Clam
The Crab Walk	Tuna
The Crab Walk	Shark
The Crab Walk	Swordfish
Devil Ray Foods	Oyster
Devil Ray Foods	Abalone
Devil Ray Foods	Clam

列区的定义给用户带来数据的一致性。因为列成为加入表中的行的属性或描述，用户就知道同一列上每行都带有同样的性质。应用这条准则也简化了数据的有效性判断。经常用这条准则，商业上的准则就能更容易地应用于各数据元素上。

3．各行惟一

关系表中各行都不相同，但至少有一列要惟一确定每行的内容，这种性质的作用不可低估。在一个非关系数据集中，很可能有重复的信息，检索操作就不得不设计成逐记录查找、比较表中所有列以满足查找要求。修改和操作同样有危险，概念上相匹配的记录的微小不同就可能导致它们不能被选出更新。

行的惟一是通过一个键值（Key Value）来实施的，第二章将详细介绍它。在这里只定义主键。它是惟一描述一行的一列或几列。对关系中的主键有两条简单准则：(1) 它非空，否则就破坏了该准则，也不能指代行；(2) 表中主键惟一，重复的键值在其他字段(Field)中就意味着重复的数据，妨碍关系操作的完整性。

键值的存在给设计数据存取、操作运算的人员带来快速、简便查找方法的自由。当找到一个与查找要求匹配的键值时，所有满足要求的行就可以找到并返回给用户。这种获取方法是键惟一性特点保证的。

仍参见表 1-1，可看到 VENDOR 这一列惟一地确定了每一行。经过转换得到的表 1-2 就因数据库中存在重复行而带来危险。我们在下一章会看到，很容易设计一个解决办法满足两条标准。

4．列序无意义

对各列次序不加注意，就使得关系表中每行数据的次序没有隐含意义。不考虑列的次序，数据库用户就可用各种组合或次序来自由地检索，像数据本身适合的次序一样去看它们，而不靠数据库本身来指定。

表 1-3 的 VENDOR 表以两种形式表示。表中的列序倒了个，但数据仍保持一样。尽管列序变了，每行仍代表同一实体。在数据库和程序设计中，维持这种状况很重要。如果想靠人为地给定数据库表中的列序来检索，就会产生关系数据库设计所不容许的维护困难。

表 1-3 列序无意义

CUSTOMER	STREET	CITY	STATE	ZIP
Kingfish, Inc.	366 Old Salt Lane	Coos Bay	OR	90333
Tidepool Company	Keiko Avenue	Reedsport	OR	90421
The Crab Walk	El Perro Blvd	Lincoln City	OR	90231
Devil Ray Foods	Navy Way	Rockaway Beach	OR	90311

ZIP	STATE	CITY	STREET	CUSTOMER
90333	OR	Coos Bay	366 Old Salt Lane	Kingfish, Inc.
90421	OR	Reedsport	Keiko Avenue	Tidepool Company
90231	OR	Lincoln City	El Perro Blvd	The Crab Walk
90311	OR	Rockaway Beach	Navy Way	

5．行序无意义

表中列序无意义，行序就更没什么意义了。没有行序的限制，用户可按数据本身的意义来重排序，则任一列数据就可决定所得表格的查找顺序。

6．各列命名惟一

列序无意义，用户就需参考关系表中列的命名而不是列的位置来指定某列，因此，表中各列要求性质惟一。这种特征可谓一箭双雕：(1) 保证所得的数据即为所要的数据；(2) 由于表中列的物理位置不重要，用户可在不妨碍先前行为的情况下容易修改。靠表中特定的列序来存取、操作的方法就没有这些特征。

1-5 数据完整性

数据库管理方法的最关键的方面之一是数据的完整性及其保证方法。关系型数据库模型的完整性准则主要对表中各列的数据值进行限制，没有这些限制，列就不可能正确赋值。对数据值的限制，简单是关键。在整个完整限制方面只有两条准则：实体完整准则和参考完整准则。

这两个关系完整准则是商业准则，而不是出于技术考虑。关系数据库定义要求用于维持数据库完整的工具是完整的。用户也不必从技术的观点来考虑它们。关系完整性，一旦以技术意义来衡量，应该对终端用户来说是透明的。

1．实体完整

第一条准则只是前面所述表中每行惟一结构要求的重述。实体完整准则要求表中主键

的任何部分都不能为空。空白数据段不能构成行的完整的描述，所以要求组成行键的各列中的数据元素有准确的意义。各行键的惟一完成了关系模型的使命。

2．参考完整

参考完整准则是新东西，牵涉到外键（Foreign Key）。外键是指另一表中起主键作用的一列或几列。对讨论表，应用外键的原因是要建立两表之间的联系方法。这两表的关系常称为父子关系，如表 1-4 中所示，这关系无论何时都要求保持一致。

表 1-4　父子关联

VENDOR(父表)

VENDOR	VEND ID
Kingfish, Inc	8213
Tidepool Company	9471
The Crab Walk	7790
Devil Ray Foods	9022

PRODUCT(子表)

ITEM ID	VENDOR ID	PRODUCT
100	8213	Kipper
120	9022	Abalone
130	8213	Smelt
230	9471	Crab
…	…	…

表 1-4 表示了父子关系。VENDOR 表包含我们的客户实体，每一行是惟一的。通常从这些提供商处购买的商品不止一件，在 PRODUCTS 表中每一项目都与其父记录通过 VENDOR ID 字段相联系，而 VENDOR ID 又是父记录中包含的供货商信息的传递。正如实际生活中，没有父亲就不会有儿子一样，没有 VENDOR ID 就没有商品项目，而只能从 VENDOR 表中的一个实体中产生。而且，这也防止了有产品项目却删除了该供货商的错误。

1-6　数 据 操 作

关系数据库模型关注的第三个方面是数据操作。该模型通过关系可进行两类操作：(1) 指定一个关系到其他关系；(2) 使用 8 种定义的操作方法对数据进行操作。

这两类操作实际上是相互交织的。如 Select 数据操作运算使被选择的数据存储在新表中。

这 8 个关系数据库操作有两个共同特点：(1) 关系操作是集合处理命令，它应用于关系表并获得关系表；(2) 操作不受数据的物理存储形式的影响，因为关系模型要求逻辑实施和物理实施的严格区分。讨论的方法如下：

需要全本请访问 www.ertongbook.com