

数据库技术

全书内容新颖

组织合理

实例丰富

可供管理软件开发人员

大专院校师生



附赠 CD-ROM

现代数据库技术丛书



(美) Carl Ganz, Jr. 著

李小坚 田英 战凤梅 等译

Visual Basic 5 开发Web数据库 指南

Visual Basic 5 Web Database Development



机械工业出版社



CMP

3B112
现代数据库技术丛书

Visual Basic 5 开发 Web 数据库指南

(美) Carl Ganz, Jr. 著

李小坚 田 英 战凤梅 等译
刘炳兴 审校

机 械 工 业 出 版 社

本书主要介绍了怎样使用 VB 进行数据库开发, 书中详细地介绍了各种数据库操作的对象、方法、属性, 还介绍了一些第三方的数据库控件以及如何用 HTML、VBScript 和 ActiveX 创建基于数据库的各种交互式 Web 页。还解释了许多 Internet 上流行的技术, 使人耳目一新。

本书是为那些对 VB 有基本了解的初学者和专业 VB 程序设计人员而编写的。

Carl Ganz, Jr.: Visual Basic 5 Web Database Development.

Authorized translation from the English language edition published by The Coriolis Group, Inc.

Copyright 1997 by The Coriolis Group, Inc.

All rights reserved.

本书中文简体字版由机械工业出版社出版, 未经出版者书面许可, 本书的任何部分不得以任何方式复制或抄袭。

版权所有, 翻印必究。

本书版权登记号: 图字: 01-98-0742

图书在版编目 (CIP) 数据

Visual Basic 5 开发 Web 数据库指南 / (美) 甘兹 (Ganz, C.) 著; 李小坚等译. - 北京: 机械工业出版社, 1998.7

(现代数据库技术丛书)

书名原文: Visual Basic 5 Web Database Development

ISBN 7-111-06532-8

I. V… II. ①卡… ②李… III. 万维网-数据库系统-BASIC 语言-程序设计-指南 IV. TP393.4

中国版本图书馆 CIP 数据核字 (98) 第 15317 号

出 版 人: 马九荣 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑: 江 颖 张彤漫

北京市密云县印刷厂印刷·新华书店北京发行所发行

1998 年 7 月第 1 版第 1 次印刷

787mm×1092mm¹/₁₆·20.5 印张

印数: 0 001-5 000 册

定价: 56.00 元 (附光盘)

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换

译 者 序

Visual Basic 是新一代面向对象的可视化开发语言，功能强大、简单易学、第三方的支持使它内容更加丰富。随着 VB5 的发行，它的数据库开发功能和 Internet 应用得到了进一步的加强。你也许用 VB 开发过数据库应用，领会过它友好的人机界面，强大而灵活的数据库操作能力，那么它在 Internet 上表现如何呢？事实上，它在 Internet 上的数据库开发能力一点也不逊色。

本书介绍了怎样使用 VB 进行数据库开发，首先从关系型数据库和 SQL 语句开始，比较详细地介绍了各种数据库操作的对象、方法、属性，还介绍了一些第三方的数据库控件。接着讲述了如何用 HTML、VBScript 和 ActiveX 建立基于数据库的各种交互式 Web 页，还解释了许多 Internet 上流行的技术，使人耳目一新。

本书的作者长期从事数据库开发，有很多的实践经验。书中所附的大量例子都有很强的实用性，并提供了一些实际中经常使用，而其他书籍很少涉及或根本不涉及的诀窍和方法。

本书由李小坚、田英、战凤梅、刘炳兴、王丹宇、张松、谢新华、王小媛等人翻译。李小坚对全书作了审校。由于译者水平有限，不切之处，热忱希望读者批评指正。

译者
1998 年 3 月

前 言

每一次推出 Microsoft Visual Basic, 都会提供越来越多的数据访问功能。当 1991 年第一次推出 VB for Windows 时, 就将它作为常规用途的开发语言, 它能向开发人员提供处理 C++ 与 Windows API 的另一方法。那时候, C++ 是创建 Windows 应用程序的唯一方法。而 Visual Basic 开创了新的领地。遗憾的是, 数据库访问是后来开发出的功能。事实上, VB1.0 版本除了提供 ASCII 文件访问之外, 并没有数据库的功能。自那时起, VB 演变成数据库访问的最基本的前端开发语言。意识到这一潮流之后, Microsoft 每次推出新版本, 都把 VB 作为数据库开发工具而不断对其进行改进。Visual Basic 也不例外。如今, Visual Basic 并非设计成只具有强大的数据库访问能力, 它还能在 Internet 上发布数据库应用程序。利用 ActiveX 技术及一套第三方附加工具, 使得一度只限于桌面上使用的 VB 应用程序现在可以在 Internet 上使用。本书把 Visual Basic 具体作为数据库开发工具来进行说明。它主要的目的是帮助读者设计与创建数据库应用程序, 并使用户能够通过 Internet 应用这些程序。

Visual Basic 经常被称为是“胶水”。也就是说, 它是用来与一套第三方附加工具一同使用的, 它们都封装在 ActiveX 控件中, 并由不同的软件厂商创建。作为开发人员, 在开发的每一个应用程序中, 我使用精选过的第三方附加工具。实际上, 没有它们我则无法正确开发软件。为此, 我将对在应用程序中用到的第三方附加工具进行讨论。本书所用到的例子也包括了这些产品, 因此, 在英文原书 CD-ROM 上, 包括了这些产品的试用版。

计算机书并非学术性的论文, 因此通常并没有什么专题, 而本书却有专题。这个专题就是为 Internet 开发数据库应用程序, 这与为其他网络开发应用程序并无不同。毕竟, Internet 只不过是世界上最大的广域网络, 具有其独自の通信协议, 这一点与 Novell 或 Windows NT 一样。Internet 上的每台机器都有其各自的操作系统, 并且具有作为公用链接的通信协议。当开发一个在 Internet 上发布的数据库应用程序并不适用于典型的客户/服务器程序时, 需要考虑一些特殊的问题, 但是这些问题很大程度上与硬件及环境有关。

本书面向哪些读者

本书面向至少了解 Visual Basic 基本知识和一般程序设计结构的读者。它是为那些不仅想知道如何在 Internet 上使用其数据库应用程序, 并且还想学习更扎实数据库知识的 VB 编程人员而编写的。这本书给出的许多例子都是取自于生活中的实例, 并且正应用于客户一方。

本书结构

第 1 章对数据库设计和关系理论进行了讨论。第 2 章介绍 SQL。第 3 章与第 4 章解释数据库对象及模型 (DAO), 及如何在应用程序中使用 DAO。第 5 章讲述了错误处理。第 6 章与第 7 章分别介绍了 ODBC 数据库和非 ODBC 数据库。第 8 章介绍了如何使用 Videosoft 的 VS-View 来书写报表, 我以前从未在哪一本书上看到过介绍数据驱动编程这一问题。这让我

很吃惊，因为它对于数据库的开发来说是至关重要的，所以我在第 9 章里专门论述了这一问题。第 10 章，我们开始进入 Internet，并介绍 HTML 语言、VBScript 以及如何操作 IE 对象。第 11 章介绍 ActiveX 的开发，第 12 章的内容是如何在 Web 上使用数据库。第 13 章介绍了如何用 VBA 的强大功能操作 Office 套件。读完此书，你就会了解为什么 VB5 是最受欢迎的开发语言。

目 录

译者序	
前言	
第 1 章 数据库设计概念	1
1.1 SQL 术语	1
1.2 开始	1
1.3 表关系	2
1.3.1 一对多	2
1.3.2 多对多	3
1.3.3 一对一	3
1.4 数据结构与存储	4
1.5 主键与外键	5
1.6 参照完整性 (RI)	7
1.7 索引	8
1.8 数据规范化	8
1.8.1 第一范式	8
1.8.2 第二范式	9
1.8.3 第三范式	9
1.8.4 第四和第五范式	10
1.9 域	10
1.10 事务	11
1.11 永久与临时数据	11
1.12 安全	11
1.13 客户/服务器概念	12
1.14 三层结构	13
1.15 本章小结	13
第 2 章 深入 SQL	14
2.1 什么是 SQL	16
2.2 数据查询语言 (DQL)	17
2.2.1 SELECT、FROM 和 WHERE 语句	17
2.2.2 ORDER BY 语句	18
2.2.3 数学运算符号	20
2.2.4 LIKE 语句	20
2.2.5 列的级连 (AS)	21
2.2.6 逻辑操作符 (AND、OR 和 NOT)	21
2.2.7 UNION 语句	21
2.2.8 BETWEEN 语句	23
2.2.9 IN 语句	24
2.2.10 DISTINCT 语句	25
2.2.11 DISTINCTROW 语句	26
2.2.12 TOP N 语句	26
2.2.13 TOP N PERCENT 语句	27
2.2.14 在 VB 代码中创建 SQL 语句	27
2.2.15 SQL 中的函数	29
2.2.16 MDB 之间的 SQL 语句	29
2.2.17 SQL 合计函数	30
2.2.18 GROUP BY 语句	32
2.2.19 HAVING 语句	32
2.2.20 JOIN 语句	33
2.2.21 子查询语句	38
2.2.22 交叉表查询	39
2.3 数据操作语言 (DML)	40
2.3.1 INSERT	40
2.3.2 INSERT INTO	40
2.3.3 SELECT INTO	40
2.3.4 UPDATE	40
2.3.5 DELETE	41
2.4 数据定义语言 (DDL)	41
2.4.1 创建表	41
2.4.2 删除表	42
2.4.3 修改表	42
2.4.4 建立索引	42
2.4.5 删除索引	42
2.5 本章小结	42
第 3 章 掌握数据库对象模型	43
3.1 DBEngine 对象	44
3.1.1 DBEngine 属性	44
3.1.2 DBEngine 方法	44
3.2 Workspace 对象	46
3.2.1 Workspace 方法	46
3.2.2 Workspace 属性	46
3.3 Database 对象	46
3.3.1 Database 属性	47

3.3.2 Database 方法	47	4.4.5 处理多用户冲突	113
3.4 RecordSet 对象	49	4.5 复制	116
3.4.1 表	49	4.5.1 复制如何实现	117
3.4.2 动态集合型	49	4.5.2 Replication Manager 3.5	117
3.4.3 快照型	50	4.5.3 数据库对象复制属性与方法	117
3.4.4 动态型	50	4.5.4 Internet 复制	19
3.4.5 正向型	50	4.6 操作二进制信息	120
3.4.6 记录集合的属性	51	4.7 本章小结	123
3.4.7 记录集合方法	56	第 5 章 错误处理	124
3.5 域对象	60	5.1 捕获错误的方法	124
3.6 TableDef 对象	67	5.2 VB 的 Err 与 Error 对象	125
3.6.1 TableDef 方法	67	5.2.1 Err 与 Error 对象的属性	125
3.6.2 TableDef 属性	68	5.2.2 Err 对象方法	126
3.7 索引	69	5.2.3 DBEngine 的错误集合	126
3.7.1 索引方法	69	5.3 错误捕获语句	128
3.7.2 索引属性	70	5.4 编译错误处理程序	130
3.8 关系	71	5.5 VB 错误传递	132
3.8.1 关系方法	72	5.6 测试错误条件	133
3.8.2 关系属性	72	5.7 集中化的错误处理	134
3.9 QueryDefs	73	5.8 错误日志	136
3.9.1 QueryDef 方法	74	5.9 来自于 Visual Basic 外部的错误	137
3.9.2 QueryDef 属性	75	5.10 本章小结	138
3.10 数据控件	76	第 6 章 使用 ODBC 数据库	139
3.10.1 数据控件属性	77	6.1 超越 Access	139
3.10.2 数据控件方法	77	6.2 Oracle	139
3.10.3 数据控件事件	78	6.3 Microsoft 的 SQL Server	144
3.11 概览	79	6.4 开放式数据库连接 (ODBC)	145
3.12 初始化 Jet	84	6.4.1 ODBC Data Source Administrator (DSA)	146
3.13 本章小结	85	6.4.2 通过 Jet 引擎访问 ODBC 数 据源	148
第 4 章 使用数据库对象模型	86	6.4.3 使用 ODBC API	149
4.1 应用程序分析	86	6.4.4 使用远程数据对象 (RDO)	153
4.1.1 开发工具	86	6.4.5 远程数据控件	156
4.1.2 飞溅屏	87	6.5 批处理更新	156
4.1.3 登录屏	88	6.6 同步和异步查询	158
4.1.4 学生信息窗体	89	6.7 缓存	158
4.1.5 词典的维护	101	6.8 数据等效类型	159
4.2 安全性	105	6.8 本章小结	160
4.3 事务处理	108	第 7 章 使用非 SQL 数据库	161
4.4 多用户问题	109	7.1 ISAM 驱动程序	161
4.4.1 安装 Jet 引擎	109	7.1.1 xBase	162
4.4.2 文件访问	109		
4.4.3 LDB 文件	110		
4.4.4 页锁定与记录锁定	111		

7.1.2 Paradox	162	10.2.4 常量	237
7.1.3 Btrieve	163	10.2.5 使用 HTML 控件	238
7.1.4 Lotus/Excel	163	10.2.6 事件处理	238
7.2 第三方方案——SuccessWare 的 Artemis	163	10.2.7 错误陷阱与调试	240
7.3 输出数据	168	10.3 使用 IE 对象编程	241
7.4 存取外部数据资源	171	10.4 本章小结	247
7.4.1 使用 DAO	171	第 11 章 ActiveX 控件编程	248
7.4.2 使用数据控件	172	11.1 什么是 ActiveX	248
7.4.3 附加表格	173	11.1.1 DCOM (分布部件对象模型)	249
7.5 ASCII 文件和二进制文件	174	11.1.2 ActiveX 的内幕	250
7.5.1 低级文件语句和函数	174	11.1.3 ActiveX 自动化	251
7.5.2 输入 ASCII 数据	181	11.1.4 类型库	251
7.5.3 输出 ASCII 数据	186	11.1.5 ActiveX 安全	251
7.6 本章小结	186	11.1.6 ActiveX 的分类	252
第 8 章 报表书写器	187	11.2 系统注册簿	253
8.1 报表书写器与手工编字代码	187	11.3 控件授权	255
8.1.1 报表书写器最大的缺点	187	11.4 OOP 面向对象的编程入门	255
8.1.2 报表书写器的缺点	188	11.5 在 VB 中创建 ActiveX 控件	257
8.2 进入 VS-View	190	11.5.1 建立 ActiveX 控件	257
8.2.1 用代码设置属性和方法	190	11.5.2 编辑和注册 ActiveX 控件	259
8.2.2 使用更小的运行时文件	190	11.6 使用 ActiveX 接口向导	259
8.2.3 打印报表	190	11.7 使用 ActiveX 文档移植向导	264
8.2.4 错误处理	202	11.8 在 Web 上使用 ActiveX 控件	265
8.3 本章小结	203	11.9 限制	270
第 9 章 数据驱动程序	204	11.10 本章小结	270
9.1 设计概念	204	第 12 章 Web 上的数据库技术	271
9.1.1 窗体实例化	204	12.1 Internet 入门	271
9.1.2 使用控件数组	207	12.1.1 Internet 通信	273
9.2 管理数据存取	211	12.1.2 协议和服务	275
9.3 宏编译	219	12.1.3 Intranet	277
9.4 本章小结	223	12.2 Microsoft 信息服务器 (IIS)	278
第 10 章 使用 VBScript、HTML 和 IE 对象	224	12.2.1 通用网关接口 (CGI)	278
10.1 HTML 入门	224	12.2.2 Internet 服务器应用编程接口 (ISAPI)	278
10.1.1 语言和语法	224	12.2.3 Active 服务器页 (ASP)	284
10.1.2 框架	227	12.2.4 Microsoft dbWeb	288
10.1.3 HTML 控件	227	12.3 高级论题	288
10.2 VBScript 与 Visual Basic	234	12.3.1 OLE DB 与 Active 数据对象 (ADO)	288
10.2.1 VBScript 原码	235	12.3.2 Microsoft 事务服务器	289
10.2.2 函数、对象和逻辑流程	235	12.3.3 Microsoft 代理服务器 (MPS)	290
10.2.3 变量范围和类型	236	12.3.4 远程访问选项	290
		12.4 本章小结	291

第 13 章 集成 Microsoft Office 97

对象	292
13.1 Office 97 对象模型	292
13.1.1 Create Object 和 Get Object 函数	293
13.1.2 Microsoft Word	295
13.1.3 Microsoft Fxcel	297
13.1.4 Microsoft Access	299
13.2 SAS 数据集的自动化	306
13.3 本章小结	307

附录 A META Solutions 公司的 VB

开发标准	308
A.1 变量类型和作用域	308
A.2 变量命名	308
A.3 通用变量	309
A.4 控件命名	310
A.5 按钮	310
A.6 菜单标准	310
A.7 代码结构	311
A.8 报表	314
A.9 软件测试	315

第 1 章 数据库设计概念

在本章，我们将对正确的数据库设计理论在技巧性与科学性方面进行检验。我们不仅要学会在现实生活中如何设计，而且同样重要的是，我们还将学会如何不去这样设计。

现代数据库设计理论是以 Dr.E.F.Codd 富有创意的工作成果为基础的。Dr.E.F.Codd 是 IBM 公司 San Jose 研究实验室的研究员。1970 年 6 月，Dr.Codd 发表了题为《大型共享数据库的关系型数据模型》（A Relational Model of Data for large Shared Data Banks）的论文。论文中提到的关系型模型成为了大多数现代关系型数据库管理系统（RDBMS）的基础，Codd 把数学中集合理论应用于数据库关系中，并且指出，数据库的逻辑存储方式是独立于物理存储方式存在的。

正确的数据库设计是应用开发的基础，这一过程应先于实际编程。若是试图对一项设计不正确的数据库应用程序进行重新设计，就如同试着往已经盖好的房子下新建地下室一样，这需要拆掉所有的结构，从头做起。我曾经咨询过许多项目，发现许多编写得很好的应用程序却是基于设计很差的数据库结构上，这给程序员和用户都带来了很多问题。程序员们苦于应用程序越来越难以更新与维护，而用户们则面临系统成本增加，性能降低，并且很有可能由于复制而使数据丧失完整性。

1.1 SQL 术语

对于那些了解 xBase 的人来说，我们有必要谈及一些有关数据库术语的问题。在 xBase 中，DBF 文件的集合被称为一组数据库，而每一个 DBF 文件则被称为是一个数据库。每一个数据库，或称每一个 DBF 文件都包含多项记录，而这些记录又由一个或多个字段组成。相比之下，SQL 术语是把所有的表、索引、查询等作为一个数据库。xBase 中的一个“数据库”相当于 SQL 数据库中的一个表。在 RDBMS 中，就是这些表存储了一组组的数据。比如 CD-ROM 里的 SCHOOL.MDB 数据库包含了这样一些表——学生、考试、课程表等等。依次，表中又包含多行信息。每一行又由列组成。在学生表中，关于学生的每一条信息都由一个列来表示——名、姓、地址等等。每一列和行的交叉点称为字段。字段中存储着系统中最小的信息单位——一个数据。例如，一个字段可以存储某一个学生的姓氏。图 1-1 表明了 SQL 数据库的结构层次。

既然我们已掌握了术语，现在我们开始进行数据库设计。

1.2 开始

设计数据库的第一步是确定系统中所含有的主要组成部分，必须把应用程序作为一个整体来验证，并把它抽象成主要部分。如果你熟悉面向对象程序设计中有关抽象过程的知识，那么这里所谈到的方法是与之相似的。首先，要列出应用程序的主要部分，然后建立表模型。客户、雇员、学生、班级、订单、询盘等都是典型数据库系统的组成部分。一旦这些组

件被抽象归纳到表中，就必须创建表与表之间的关系。

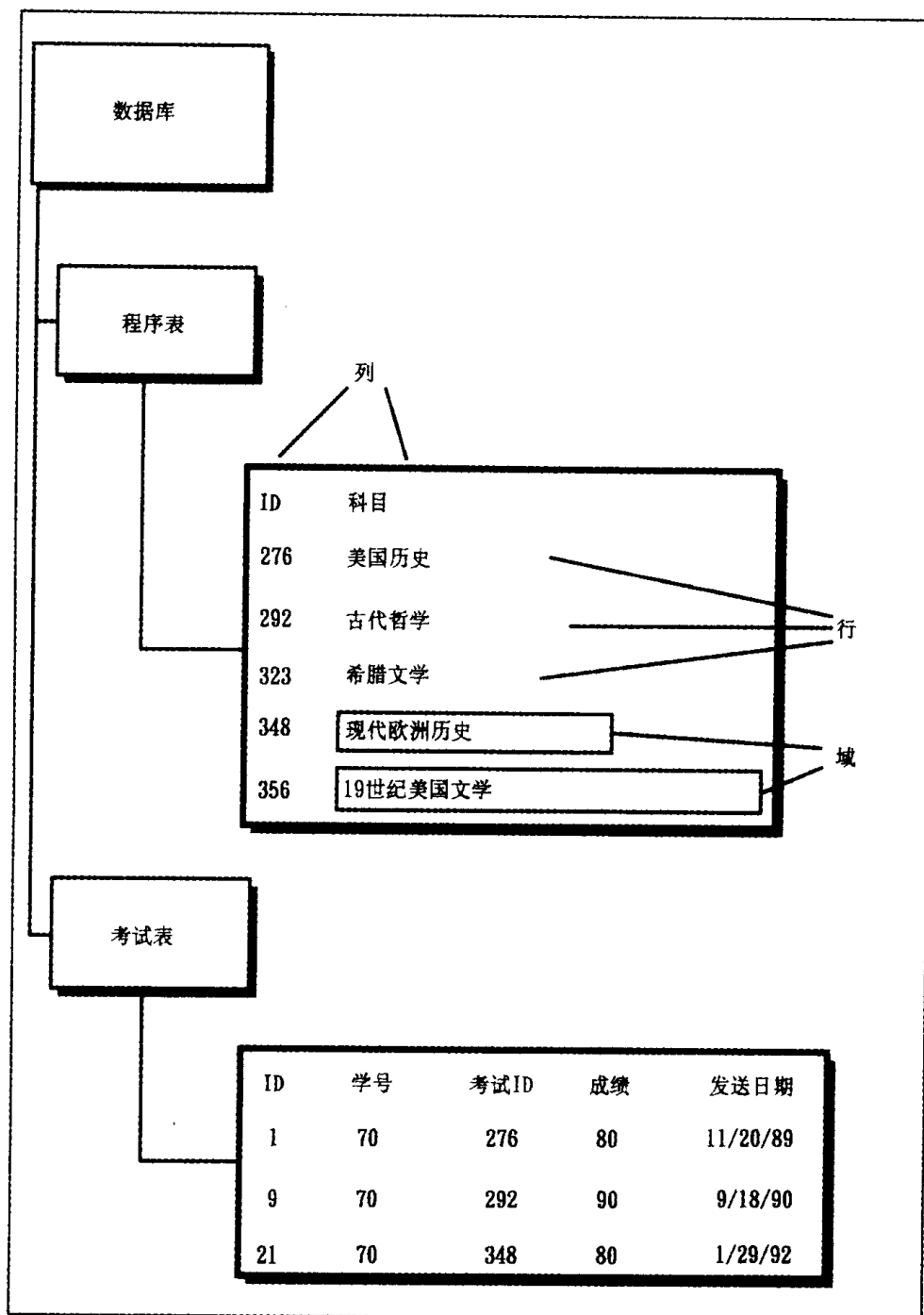


图 1-1 数据库对象层次

1.3 表关系

表与表之间有三种关系：一对多、多对多、及一对一。让我们快速浏览一下每一种关系类型。

1.3.1 一对多

一对多是 RDBMS 中最普通的关系类型。这种关系又被称为是父表/子表关系或主表/分

表关系。在数据库启蒙书籍中经典的例子就是一个应收帐款发票系统。发票标题表中的每一行在发票明细表中都有一行以上的相关行。在应收帐款系统中，发票标题表就是一个父表，而发票明细表为子表。图 1-2 就体现了一种父表/子表关系。这里，学生表与国家表就是一对多的关系。学生表为父表，考试表为子表，学生表中的一行在考试表中可以有零行、一行或许多行与之对应，而考试表中的一行都与那个学生表的唯一行对应。

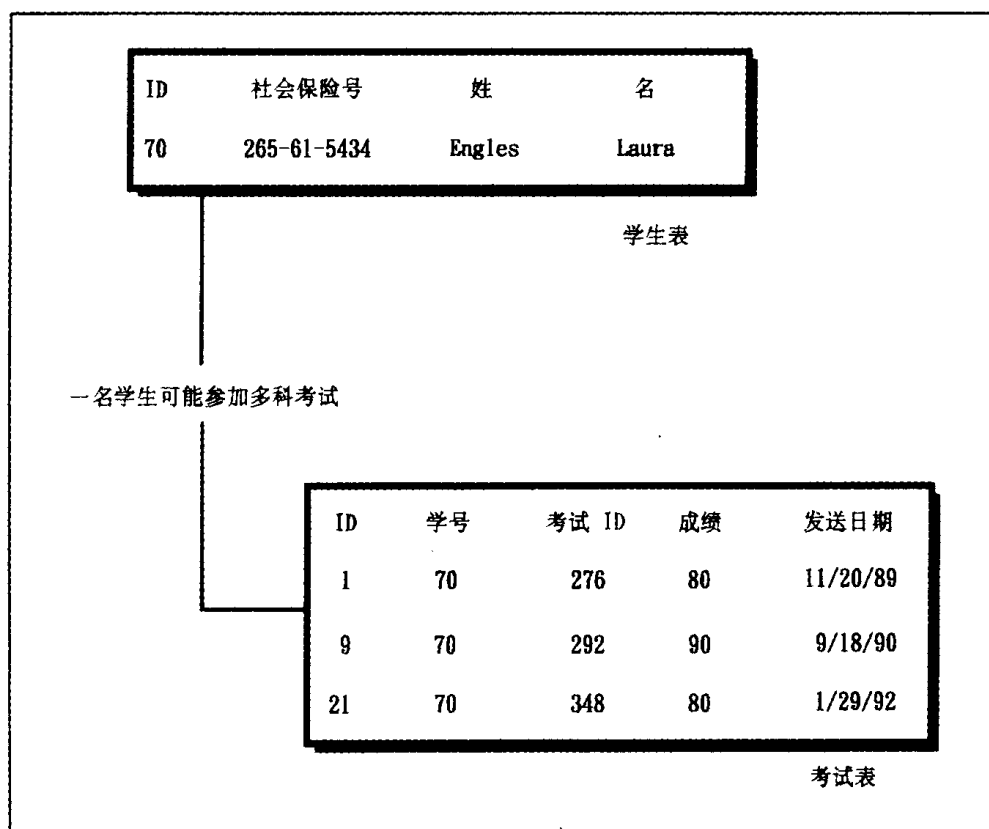


图 1-2 一对多关系

1.3.2 多对多

在多对多关系中，第一张表的每一行可对应第二张表的许多行。同样地，第二张表中的每一行也可与第一张表中的许多行相对应。例如，一个学生可以参加多次考试，一次考试也可由许多学生参加。要想在结构上做到这样，就要建立第三张表，将学生代码与考试代码连接起来，这个表就是前两个表的父表。在图 1-3 里，“考试表”就是两个子表——学生表与课程表——的父表。

1.3.3 一对一

一对一的关系将一表中的一行与另一表中的一行连接起来。这种关系非常少见，因为如果是这种关系的话，将第二张表取消而将其中的列并入第一张表会更合理。建立一对一的关系通常是为了克服表对列数的限制。在 xBase 系统中这种情况更为常见，由于静态分配的磁盘空间有限，文件长度不断增大会带来问题。在 xBase 文件中，对于 100 字符的文本字段，无论其字段的实际内容的大小如何，它都会占用 100 字符的磁盘空间。因此，许多有注释字段的系统经常会以一对一的关系创建起一个注释表，这样在并非每一记录都使用注释的情况

下，避免了主要文件字节不断增大，数据库系统诸如 Access 和 Oracle 对数据存储采用磁盘空间的动态分配。例如，如果一个字段具有 100 字节的空间而只需要其中的 10 个，那么只有 10 个字节的空间被用掉。总的来说，就不必创建一对一的关系。

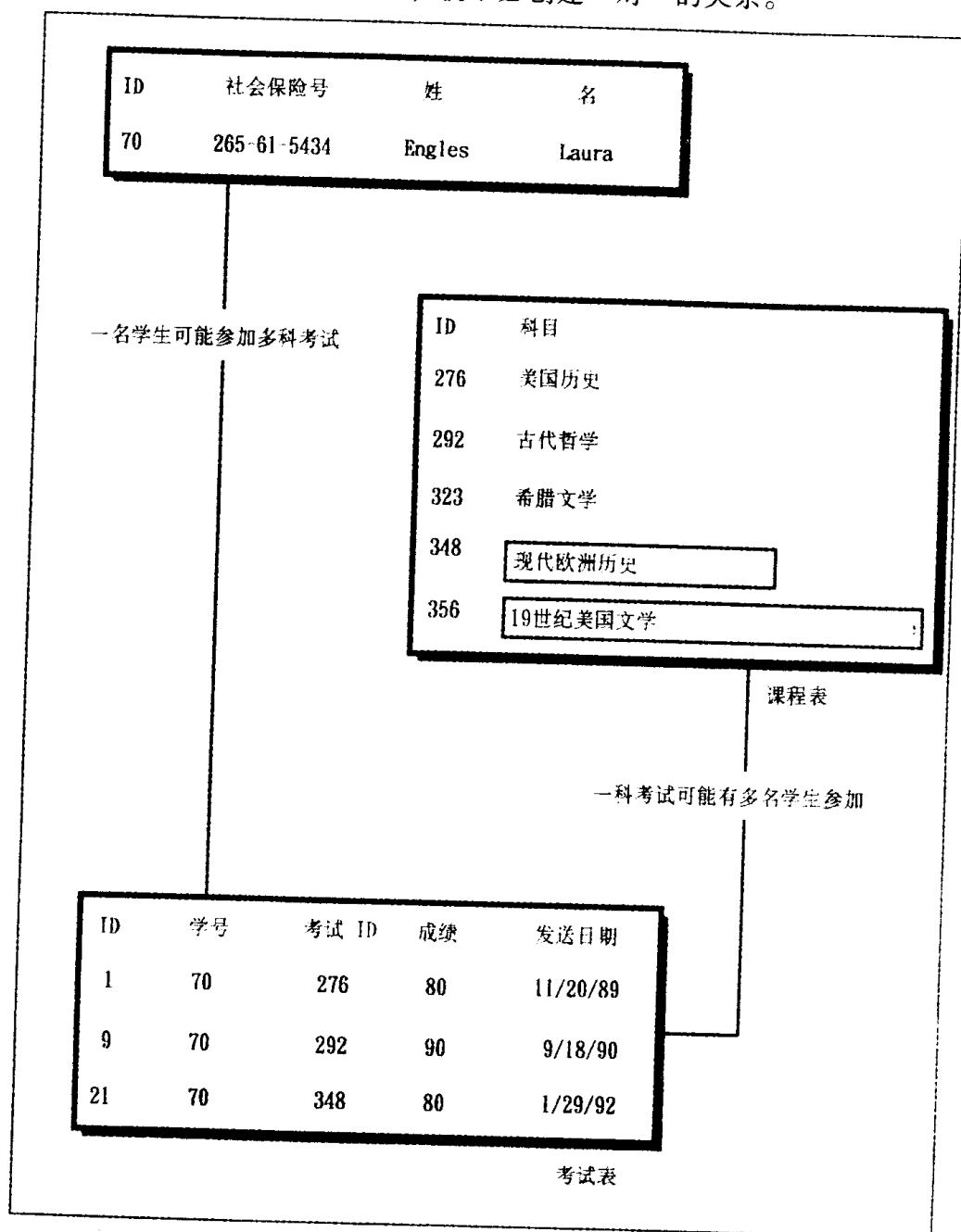


图 1-3 多对多关系

1.4 数据结构与存储

由于数据库的逻辑结构必须独立于其物理结构，在设计表格、字段和关系时就需记住一些规则。行与列的顺序并不起什么作用，虽然在表中，列会有一定的顺序位置，但它们并不以此来代表引用顺序。记录的顺序只有在它们从表中取出并送给用户时才起作用。SQL 会通过参照数据库索引来处理记录的提取顺序。

分配列的大小时要慎重，如果一列用来存储一个整数，可以使用整型和长整型。同样

地，如果够用的话，最好建立 100 字符的文本字段而不要用备注字段。在分配列的内存时要与分配变量内存一样慎重。表 1-1 列出了 Access 列的类型。

表 1-1 列数据类型

数据类型	注 释	最大长度
文本型	文本或字母数据	255 字符
备注型	自由形式文本	64, 000 字符
字节型	0 至 255 之间的值	1 字节
整型	- 32, 768 至 32, 768 之间的值	2 字节
长整型	- 2, 147, 483, 648 至 2, 147, 483, 647 之间的值	4 字节
单精度型	- 3.402823E38 至 3.402823E38 之间的值	4 字节
双精度型	- 1.79769313486232E308 至 1.79769313486232E308 之间的值	8 字节
复制 ID	向表中的每一行提供保证唯一值，用以进行数据复制	16 字节
日期/时间	100AD 至 9999AD 之间的值	8 字节
货币型	数学计算中用到的数字数据，精确到小数点后 15 位	8 字节
自动编码	从前称为计数器，是当表中加入新数据时，Access 设定的唯一数字；自动编码字段不能被更新	4 字节
Yes/No	布尔值及只包含 True/False, Yes/No 的字段	1 位
OLE 对象	链接或嵌入的对象，如 Word 文档或图象	千兆字节

1.5 主键与外键

表中每一行必须是唯一的，即必须可以将之与同一表中的其他行区别开来。可通过主键做到这一点。主键是具有唯一值的一个单列或多列的组合，并以此标识出表中每一行。因此，主键不能具有空值。甚至，由多列组成的主键的某一部分也不能具有空值。在 Access 中，通常是把一列设计成自动编号列（原先是计数器），而在其他的 RDBMS 中建立主键，通常是通过系统中的累加器来生成一长整型列并赋值。无论哪种方式，每一行都要与其他行区别开来。

还有其他生成主键的方式。一些开发人员利用了表中根据定义具有唯一值的某一列。所有具有唯一值的某一单列或多列的组合都称为候选键，而主键是由候选键生成的。如前所述，主键可以是简单到只包含一列，但也可以包含多列。无论你选择了哪种主键，一定要注意以下两点：一，尽可能选择最少的列。二，如果键有变化，选择变化少的那个键。例如，社会保险号码，或学生号可以特殊区分每一名学生。注意：这种方法是假定社会保险管理局或学生注册办公室从不因疏忽而给两人注册同一号码的。为确保安全起见，要使用自动编号列，使系统生成唯一数字，该数字对用户来说是显而易见的。

Access 并不要求你一定要给每一个表建一个主键，但不论怎样，你都应该建立它，哪怕你认为你根本不会用到它。在一个关联着发票标题表和列有明细项目表的应收帐款系统中，你会认为连接两个表只需发票号码就足够了。实际上也确实如此。然而，如果没有主键，就

无法保证发票明细表中每一行都能区别出来。因此，为坚持关系模式的原则，要确保表中包含主键列。

提示 使用数字列—这样比较快捷

你应该用数字列来作为主键，因为在内存中数值间要比串间易于比较。用长整型做主键是最理想的。对于四字节数据来说，可生成多于 210 亿的唯一值。就连像社会保险号码这样本身就是唯一值的列，也不要用它做主键。代之以专门设定的数字，这样可以很好地实现目标并且数字不与系统数据相混淆。

外键是一张表中引用另一张表中主键的列，见图 1-4，学生表中的国籍列就是一种外键，因为它对应着国家表中的主键列。

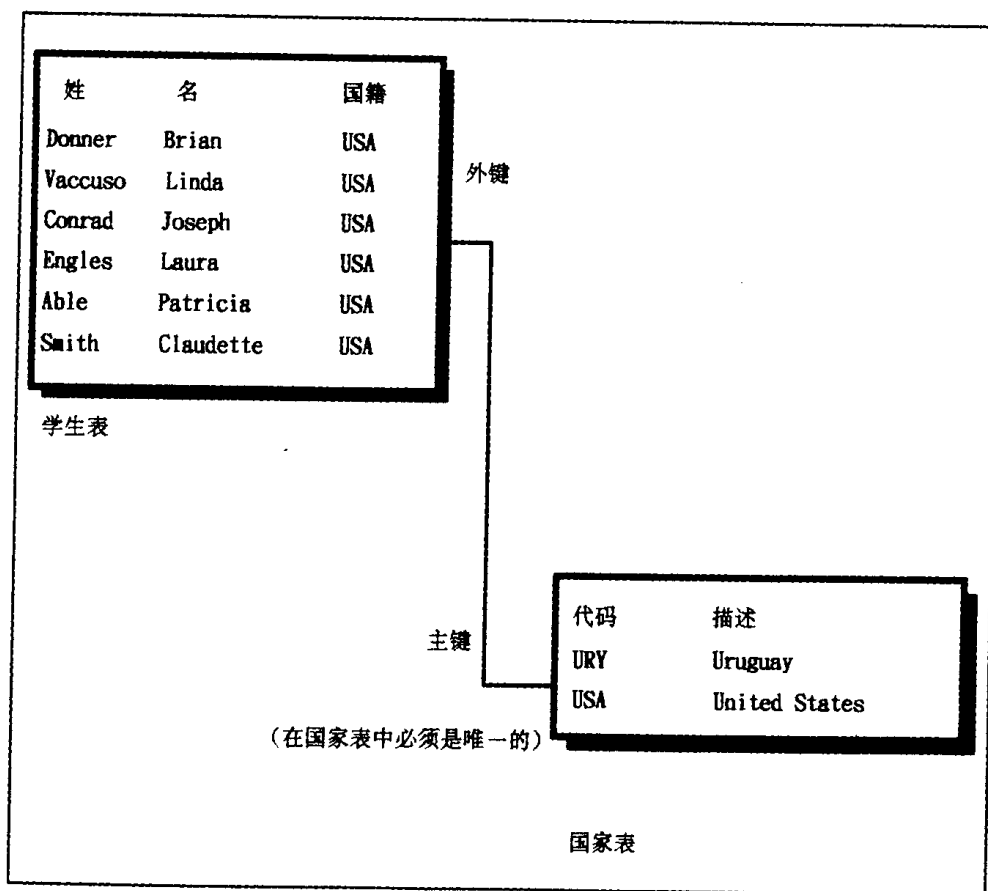


图 1-4 主键和外键

提示 在词典表中存储查找值代码

如果限定一条数据必须是一些有限的选择而不是静态的，把这些选择存于词典表中。传统上讲，这些词典表至少包含两列：代码列，它是自动编号列；描述列，它是一个文本（不要用 Desc 作为列名，因为它是 SQL 中保留的一个字）。引用系统的长整型码列所占用的磁盘空间要少于存储 35 个或更多的字符的描述所占用的空间。例如，在学生表中，把国名存为 USA 或者是一个唯一的无意义的数字，要好于把国名存为 United States of America。如果预期描述只有两三个字符，也要用代码或描述的方法来保持与应用程序其他词典的一致性。在一些应用程序中，允许

用户创造他们自己的缩写字符号码，而不是生成唯一的数字符号。缺乏经验的用户经常用报表生成器转储表格，而不明白要与其他表连接起来取出其中值。因此，这些用户需要根据代码本身决定外键的含义。

1.6 参照完整性 (RI)

参照完整性的规则是所有外键必须与同一主键相匹配。比如：学生表中的国籍代码 USA 必须与含有主键代码 USA 的国家表中的那一行相对应。如果没有对应的行，就违反了参照完整性原则，并出现了孤立记录，如图 1-5 所示。

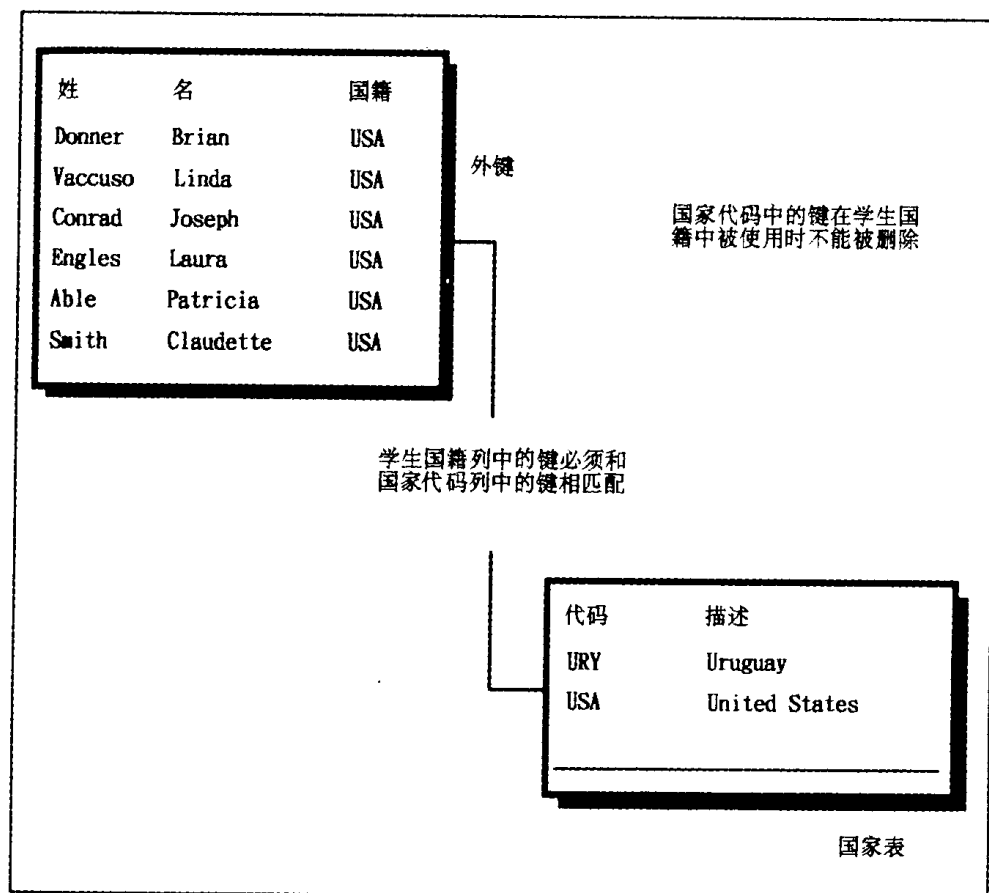


图 1-5 参照完整性

幸运的是，是 RDBMS 而不是应用程序维护参照完整性规则，例如单层文件系统。如果应用程序要进行某一违反参照完整性规则的操作，RDBMS 会引发一个错误，并向应用程序返回一个标识错误类型的代码。程序员只需确定这个错误，并显示正确的警告提示就可以了。

在 RDBMS 中可以选择使用级联更新和删除功能。如果你使用级联更新和删除功能，某一主键的变化会自动引起相应表中主键的变化。比如，如果学生表中的学号 214 被删除，级联删除功能会自动清除试题一中所有与学号 214 有关的行。如果试题表中存在相应行，RDBMS 还可以按指令禁止在学生表中进行任何删除。选择何种参照完整性方法取决于应用程序所要建立的模型规则。在任何情况下，RDBMS 都会指出违反参照完整性的作法。你不需要对比检查程序代码。图 1-6 显示在 Access 中如何建立参照完整性。