

教育部、国家民委规划教材

Visual FoxPro

程 序 设 计

主 编 黄洪强
副主编 李作主 张慧丽
徐 薇

广西民族出版社

教育部、国家民委规划教材编委会

主任委员 图道多吉

副主任委员 吴仕民 夏 铸

委 员 李步海 陈 理 张春雨

马 建 张 强 孟立军

教育部、国家民委规划教材
Visual FoxPro Chengxu Sheji
Visual FoxPro 程序设计

主编 黄洪强

出版发行	广西民族出版社(地址:南宁市桂春路3号 邮政编码:530028)
发行电话	(0771)5523216 5523226 传 真(0771)5523246
E-mail	CR@gxmzbook.cn
责任编辑	韦光化
封面策划	张文馨
封面设计	吴左平
责任校对	方 丹
责任印制	蓝剑风
印 刷	南宁市信亨昌印务有限责任公司
规 格	787×1092 毫米 1/16
印 张	18
字 数	430 千字
版 次	2005 年 1 月第 1 版
印 次	2005 年 1 月第 1 次印刷
印 数	1~3 500 册

ISBN 7-5363-4883-5/G · 1889

定价:25.80 元

如发现印装质量问题,影响阅读,请与出版社联系调换 电话:(0771)5523216

编者的话

由于数据库知识及其应用在计算机教学中占有十分重要的地位，为了适应教学改革的需要，更好地优化教材结构，我们对大学计算机基础教学方案进行了结构性调整，将《计算机基础知识及应用》第七章（Visual FoxPro 6.0）的内容从原书中剥离出来，从而形成了这本《Visual FoxPro 程序设计》，极大地丰富了教材的内容，更能够体现教学要求。

本书共分十章，各章的具体内容分布情况如下：

- | | |
|-----------------|---|
| 第一章 数据库的基础知识 | 着重介绍了关于数据库的一些初步知识。 |
| 第二章 数据与数据运算 | 介绍了常量、变量、表达式及函数等基本概念。 |
| 第三章 数据库基本操作 | 介绍了项目、数据库及表的操作方法和命令。 |
| 第四章 结构化查询语言 SQL | 详细介绍了 SQL 语言的语法规则、功能及使用方法。 |
| 第五章 查询与视图 | 着重分析了查询与视图的区别和使用方法。 |
| 第六章 结构化程序设计 | 介绍了程序设计语句和程序的结构化设计方法。 |
| 第七章 表单设计与应用 | 全面介绍了各种表单对象的属性、事件和方法以及面向对象程序设计的基本操作和技巧。 |
| 第八章 报表设计 | 介绍了根据不同的需要来制作各种不同报表的方法。 |
| 第九章 菜单设计 | 介绍了各种不同形式菜单的设计方法。 |
| 第十章 应用系统开发 | 运用实例演示在实际工作中如何用 VFP 开发环境解决实际工作问题的全过程，它是对全书各知识点的总结与综合。 |

本书在中南民族大学领导的关怀和大力支持下，集全体同仁的共同智慧，由黄洪强、李作主、张慧丽、徐薇执笔编写。为追求尽量满足学生求知和实践应用能力的需要，所有的老师都花费了大量的心血，提供了宝贵的意见，在此表示衷心的感谢。

编者

2004年8月

前 言

《Visual FoxPro 程序设计》是全国计算机等级考试的考试科目之一，也是高等院校学生学习数据库程序设计的计算机公共基础课程。为了帮助广大学生深入理解关系数据库的基本概念，灵活运用 Visual FoxPro 的基本知识，熟练掌握 Visual FoxPro 的上机操作技巧，我们结合全国计算机等级考试的考试大纲，编写了这本《Visual FoxPro 程序设计》教材。

本书的读者对象

本书可以作为高等学校各专业计算机程序设计基础 Visual FoxPro 课程的教材，同时也可以作为计算机等级考试培训的教学参考书。本书从基础入手，贴近等级考试，分析精辟，深入浅出，既可以作为高等院校各专业的 Visual FoxPro 程序设计教材，也可以作为计算机等级考试培训的教学参考书；实践教程结合大量的计算机二级考试中所涉及的考试要点及解题方法，能够有效地提高学生上机操作能力。

本书的特点

1. 针对计算机二级考试分为笔试和上机考试两部分的特点，本书也分解为理论部分和实践教程两部分。理论部分侧重分析讲解，实践环节侧重操作方法的训练。

2. 本书在命令格式通式表述上，采用了各教材通行的符号约定方案：

< > 为必选项符号，表示必须提供一个值，并应满足 <> 内项的要求。

[] 为可选项符号，表示 [] 内的值，可以视情况进行增删。

| 或 / 为任选项符号，表示在 | 线两边的选项，只能任取其一。

……省略符号，表示继续重复前面的项目，各选项之间用逗号分隔。

以上各种符号在实际应用命令时不需要输入。

3. 本教材力求做到简明扼要，层次分明；既能够通过 Visual FoxPro 深入理解数据库的理论与应用，又能够适应全国计算机二级考试。

4. 本书理论与实践相互呼应，由简入难，使学生在循序渐进中学习，始终围绕计算机等级考试这个中心，保持相应的难度。

使用建议

本书内容丰富，注重理论与实践相结合。由于各大专院校的专业设置及教学计划的学时数量不尽相同，在教材的使用上，教材的使用者可根据实际情况有所侧重。一般读者可以从操作入手，体会操作过程，以此升华，从而理解基础知识和基本概念。

由于编者的水平有限，编写时间仓促，书中的错误和不当之处在所难免，我们热诚地欢迎广大读者提出宝贵意见。

编 者

2004 年 8 月

目 录

第一章 数据库的基本概念

1.1	数据库基础知识	(1)
1.2	数据模型	(4)
1.3	关系数据库	(6)
1.4	Visual FoxPro 简介	(12)

第二章 数据与数据运算

2.1	常量	(15)
2.2	变量	(17)
2.3	表达式	(21)
2.4	函数	(26)
2.5	表达式生成器简介	(38)

第三章 数据库操作

3.1	表的基本操作	(40)
3.2	数据查询与统计	(57)
3.3	数据工作期	(66)
3.4	项目管理器对数据库的管理	(74)
3.5	数据库表的操作	(79)
3.6	数据完整性	(84)
3.7	数据共享与交换	(91)

第四章 结构化查询语言

4.1	SQL 语言概述	(96)
4.2	数据定义	(97)
4.3	数据操纵	(105)
4.4	数据查询	(107)

第五章 查询与视图

5.1 查询	(125)
5.2 视图	(135)

第六章 结构化程序设计

6.1 程序与程序文件	(146)
6.2 分支结构程序设计	(150)
6.3 循环结构的程序设计	(154)
6.4 多模块程序设计	(159)
6.5 变量的作用域	(163)

第七章 表单设计与应用

7.1 面向对象的概念	(166)
7.2 创建表单	(171)
7.3 管理表单及对象	(176)
7.4 表单设计器	(183)
7.5 常用表单控件	(187)

第八章 报表设计

8.1 报表概述	(214)
8.2 使用报表向导创建报表	(215)
8.3 使用报表设计器创建快速报表	(220)
8.4 使用报表设计器自定义报表	(221)
8.5 报表输出	(240)

第九章 菜单设计

9.1 菜单结构	(242)
9.2 菜单设计	(243)
9.3 表单中的菜单定义	(253)

第十章 应用系统开发

10.1 应用项目综合设计	(258)
10.2 应用程序生成器	(269)

第一章 数据库的基本概念

在社会高度发展的时代，需要通过数据管理来处理海量信息。20 世纪 60 年代末期，数据管理由人工管理阶段进入了计算机的数据库管理阶段。在这一阶段，数据库系统经历了两代的演变。第一代是层次与网状数据库系统，第二代是关系数据库系统。由于关系数据库具有易于理解、设计和建立等特点，所以目前在较新的信息管理系统中，关系数据库系统占据着绝对优势的地位。

1.1 数据库基础知识

数据是指存储在某一种媒体上的能够识别的物理符号。数据的概念包括两个方面：其一是描述事物特性的数据内容，其二是存储在某一种媒体上的数据形式。信息是有用的数据，而数据则是信息的具体表现形式。

数据处理是指将数据转换成信息的过程。从数据处理的角度而言，信息是一种被加工成特定形式的数据，这种数据形式对于数据接收者来说是有意义的。

数据库是相互关联的数据的集合。它安全地存储信息并对其进行组织，以便于用户可以快速地检索到所需数据。

1.1.1 计算机数据管理的发展

数据处理的中心问题是数据管理。计算机对数据的管理是指以计算机作为组织、分类、编码、存储、检索和维护数据的操作手段。

计算机数据管理随着计算机硬件技术、软件技术和计算机应用范围的发展而不断发展，多年来经历了人工管理、文件系统管理、数据库系统管理、分布式数据库系统管理和面向对象数据库系统管理等几个阶段。

1. 人工管理

20 世纪 50 年代中期以前，外存储器还没有磁盘，只是用磁带、纸带等进行信息存储，也没有专门的数据管理软件，数据包含在处理 and 计算它的程序之中。此阶段的特点是：数据与程序是不可分割的统一体，数据由计算机完成，而处理结果则由人工管理；由于一个程序中的数据不能被另外一个程序使用，从而造成大量的数据重复，被称之为数据冗余。

2. 文件系统管理

20 世纪 50 年代后期至 60 年代中后期，计算机开始大量地用于数据管理。在计算机的

硬件方面已经有了磁盘、磁鼓等直接存取的存储设备；在软件方面，操作系统中已经有了专门的数据管理软件——文件系统，如图 1—1 所示。此阶段的特点是：

- (1) 程序和数据分别按文件的形式存储，分别称为程序文件和数据文件。
- (2) 程序和数据虽然具有一定的独立性，但两者之间有一定的依赖性。数据可被特定的应用程序多次使用。
- (3) 文件系统管理会造成数据不一致和冗余度大。

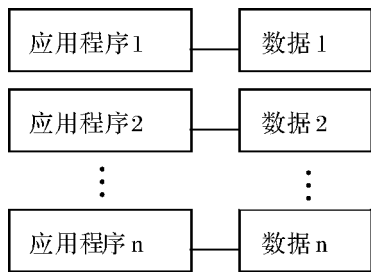


图 1—1 文件管理模型

3. 数据库系统

20 世纪 60 年代后期以后，计算机用于管理的规模越来越大，人们对数据共享的需求也日益增强。虽然这时硬件的价格在下降，但软件的价格上升很快，为编制和维护系统软件及应用程序所需的成本在相对增加。在处理方式上，联机实时处理的需求很多，人们已开始提出并考虑分布处理。为了解决多用户、多应用共享数据的问题，使数据为尽可能多的应用服务，数据库技术便应运而生，如图 1—2 所示。此阶段的特点是：

- (1) 数据由数据库管理系统统一管理和控制。数据库管理系统还提供了数据的安全性、完整性、并发控制和数据库恢复等功能。
- (2) 数据高度集成和数据结构化（这是数据库与文件系统的根本区别）。
- (3) 数据具有一致性并且冗余度低，易扩充。
- (4) 实施统一标准实现数据充分共享。

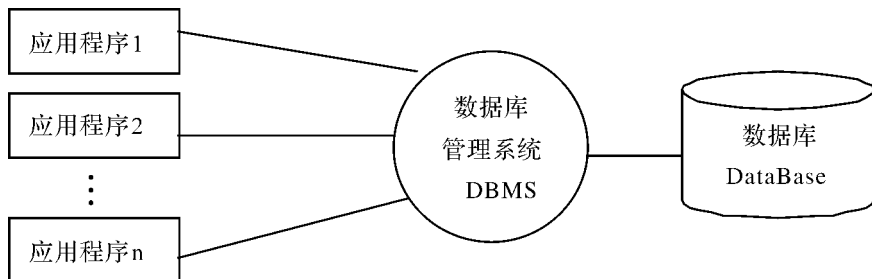


图 1—2 数据库管理模型

4. 分布式数据库系统

分布式数据库系统是数据库技术和计算机网络技术紧密结合的产物。分布式数据库是把数据分散存储在网络的多个节点上，各个节点上的计算机可以利用网络的通信功能访问其他结点上的数据库资源。分布式数据库管理系统分为物理上分布、逻辑上集中的分布式数据库结构和物理上分布、逻辑上分布的分布式数据库结构两种。

5. 面向对象数据库系统

面向对象数据库是支持面向对象开发模式的较新型的系统。所谓对象是包含现实世界物体特征的抽象实体，它反映了系统为之保存信息和（或）与它交互的能力。它是一些属性及服务的一个封装体，在程序设计领域，可以用下列公式来表达：

对象 = 数据 + 作用于这些数据上的操作

面向对象的主要思想是提高抽象的水平，从而使设计和建立信息管理系统更为接近自然。例如，在面向对象数据库中，数据库不是存储学生、学生选课等的表，而是存储学生和

学生选课实例。存储在数据库中的相关方法则描述如何增加、更改和删除学生对象和学生选课对象。

面向对象数据库是面向对象的方法在数据库领域中的实现和应用，它既是一个面向对象的系统，又是一个数据库系统。

1.1.2 数据库系统的组成

数据库、数据库应用系统、数据库管理系统和数据库系统经常被作为同义词来使用，但严格地讲，它们是几个相互关联，但却是截然不同的概念。

1. 数据库

数据库是存储在计算机存储设备上的结构化的相关数据的集合。它不仅包括描述事物的数据本身，还包括相关事物之间的联系。

2. 数据库应用系统

数据库应用系统是指系统开发人员利用数据库系统资源开发出来的，面向某一类实际应用的应用软件系统。例如，学生档案管理系统、图书管理系统等。

3. 数据库管理系统

数据库管理系统 DBMS (DataBase Management System) 是大多数管理数据类应用软件的开发平台，它可以对数据库的建立、使用和维护进行有效的管理，包括存储管理、安全性和完整性管理等。例如，Visual FoxPro 就是一个数据库管理系统。

4. 数据库系统

数据库系统是指引进数据库技术后的计算机系统，它可以实现有组织地、动态地存储大量相关数据，以提供数据处理和信息资源共享的便利手段。数据库系统的组成如图 1—3 所示。

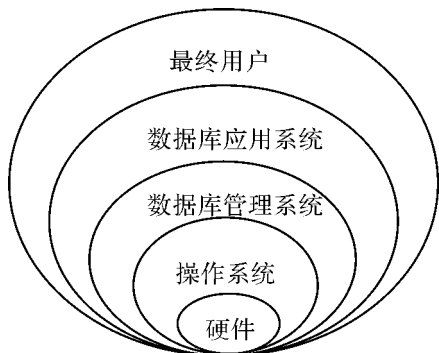


图 1—3 数据库系统层次结构

1.1.3 数据库的三级模式

从数据库管理系统的角度看，数据库系统通常采用三级模式结构，并提供两级映像功能。这是数据库管理系统内部的系统结构。

如图 1—4 所示，数据库系统的三级模式结构是指数据库系统是由外模式、模式和内模式三级构成的。

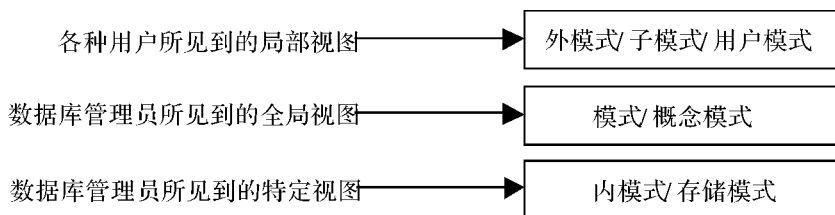


图 1—4 数据库系统的三级模式结构

1. 外模式

外模式也称子模式或用户模式，它是数据库用户（包括应用程序员和最终用户）能够看见和使用的局部数据的逻辑结构和特征描述，是数据库用户的数据视图，是与某一应用有关的数据的逻辑表示。

外模式通常是模式的子集。一个数据库可以有多个外模式。由于它是各个用户的数据视图，如果不同的用户在应用需求、看待数据的方式、对数据保密的要求等方面存在差异，则其外模式描述是不同的。另一方面，同一外模式也可以为某一用户的多个应用系统所使用，但一个应用程序只能使用一个外模式。

2. 模式

模式也称概念模式，是数据库在逻辑级上的视图，它是数据库中全体数据的逻辑结构和特征的描述，是所有用户的公共数据视图。一个数据库只有一个模式。

模式是数据库系统模式结构的中间层，既不涉及数据的物理存储细节和硬件环境，也与具体的应用程序、所使用的应用开发工具及高级程序设计语言无关。定义模式时不仅要定义数据的逻辑结构，而且要定义数据之间的联系，定义与数据有关的安全性、完整性要求。

3. 内模式

内模式也称存储模式，一个数据库只有一个内模式。它是数据物理结构和存储方式的描述，是数据在数据库内部的表示方式。

数据库管理系统在以上三级模式之间提供了两层映像：外模式/模式映像和模式/内模式映像。正是这两层映像保证了数据库系统中的数据能够具有较高的独立性。

1.2 数据模型

数据模型是一组描述数据库的概念。这些概念精确地描述了数据、数据之间的联系、数据的语义和完整性约束。大部分数据模型还包括一个操作集合，这些操作用来说明对数据库进行存取和更新的方法。

1.2.1 基本概念

1. 实体的描述

(1) 实体。

客观存在并且可以相互区别的事物称为实体。它可以是具体的人、事、物，如一个学生、一棵树；也可以是抽象的概念或联系，如比赛活动、学生与成绩的关系等。

(2) 实体的属性。

描述实体的某一方面的特性称为属性。一个实体可以由若干个属性来描述。例如，学生实体可以由学号、姓名、性别、学习成绩等属性组成。

(3) 实体型和实体集。

实体可以用型和值来表示。实体型是属性的集合。例如：反映一个学生全部特征的所有属性之和，就是学生这个实体的型。将这些属性落实到某个学生身上而得到的所有数据就是实体的值。同类型的实体的集合称为实体集。如全班学生就是一个实体集。

在 Visual FoxPro 中，用“表”来存放同一类实体，即实体集。每一条记录表示一个实体，每个字段表示一个属性。如图 1—5 所示的 Student 表就是一个表示学生基本特征的实体集。

学号	班号	姓名	出生日期	性别	入学分数	班级	备注	照片
040001	2	李强	02/18/81	女	568.0	T		
030002	4	谭安琪	03/13/82	女	572.0	F		
020003	1	陶翠翠	05/09/83	女	568.0	T		
030004	2	于文博	08/08/84	男	602.0	F		
020005	5	张磊	NULL	女	564.0	F		
040006	3	陈露露	12/28/83	女	584.0	F		
030007	4	王帅	10/22/84	男	572.0	T		
020008	3	李斯文	NULL	男	564.0	T		

图 1—5 Student 实体集

2. 实体间的联系及联系的种类

实体与实体之间的对应关系称为联系。实体—联系方法通常使用 E—R 图表示。E—R 图直观地表达了概念结构数据模型，可方便地转换成表的逻辑结构。在 E—R 图中，用矩形表示实体，菱形表示联系，椭圆形表示实体的属性。

在数据库应用中，一个实体集中的每一个实体与另一个实体集中具体实体存在着如图 1—6 所示的三种联系类型。

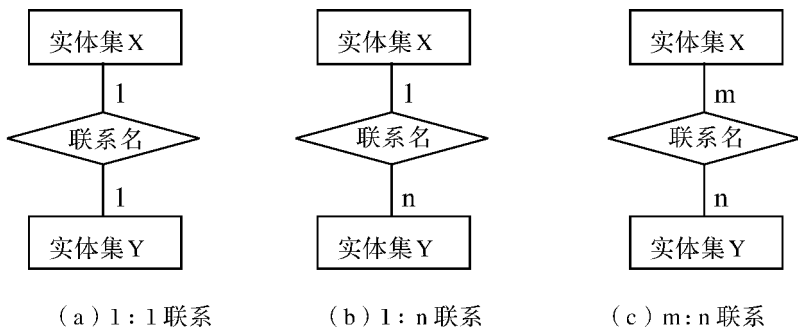


图 1—6 实体—联系的 E-R 图

(1) 一对一联系 (1:1)。

若 X 中的任一实体至多对应 Y 中的唯一实体，反之亦然，则称 X 与 Y 是一对一联系。如一个班只能有一个班长，而一个班长只能属于一个班，则班长和班之间具有一对一联系。

(2) 一对多联系 (1:n)。

若 X 中至少有一个实体对应 Y 中一个以上的实体，且 Y 中任一实体至少对应 X 中的一个实体，则称 X 对 Y 是一对多联系。如一个班对应多个学生，而一个学生只属于一个班，则班与学生之间具有一对多联系。

(3) 多对多联系 (m:n)。

若 X 中至少有一个实体对应 Y 中一个以上的实体，且 Y 中也至少有一个实体对应 X 中一个以上的实体，则称 X 与 Y 是多对多联系。如一个学生可以学习多门课程，而一门课程又可以由多个学生来学习，则学生与课程之间具有多对多联系。

实际上，一对一联系是一对多联系的特例，而一对多联系又是多对多联系的特例。

1.2.2 数据模型简介

数据模型是数据库管理系统用来表示实体及实体间联系的方法。一个具体的数据模型应当能正确地反映出数据之间存在的整体逻辑关系。

任何一个数据库管理系统都必须基于某种特定的数据模型。数据库管理系统所支持的数据模型分为三种：层次模型、网状模型、关系模型。因此，使用支持某种特定数据模型的数据库管理系统开发出来的应用系统相应地称为层次数据库系统、网状数据库系统和关系数据库系统。

1. 层次模型

层次模型使用树型结构来表示各类实体以及实体间的联系，其特点如下：

- (1) 有且只有一个节点没有双亲节点，这个结点称为根节点。
- (2) 根以外的其他节点有且只有一个双亲节点。

层次模型不能直接表示多对多联系。

2. 网状模型

网状模型是层次模型的扩展，它是一种比层次模型更具普遍性的结构，层次模型可看作是网状模型的特例。其特点如下：

- (1) 允许一个以上节点没有双亲。
- (2) 一个节点可以有多个的双亲节点。

3. 关系模型

关系模型是目前最重要的一种数据模型。关系模型中的关系是一张由行和列组成的二维表。Visual FoxPro 就是一个以关系数据模型进行数据管理的系统，下面侧重讨论关于关系数据库的基本概念。

1.3 关系数据库

关系数据库采用关系模型作为数据的组织方式。关系数据库对用户隐藏了数据访问的复杂性，使应用程序的开发相对其他类型的数据库系统来说要简单一些。

1.3.1 关系模型术语

用二维表的形式表示实体和实体间联系的数据模型称为关系数据模型，它是目前多数商品化数据库系统所采用的数据模型。

关系数据库是通过关系模型表示的，在关系模型的描述上，涉及下面一些术语。

1. 关系

一个关系就是一张符合一定要求的二维表，每个关系有一个关系名。对关系的描述称为关系模式，一个关系模式对应一个关系结构。其格式为：关系名（属性名 1，属性名 2，……，属性名 n）。例如，图 1—7 为表示院系基本情况的 Dep 关系。

院系号	院系名称	电话号码
1	计算机科学院	(027)-62147822
2	管理学院	(027)-58746014
3	法律系	(027)-68475235
4	外语系	(027)-53241825
5	计算中心	(027)-67842113
6	经济学院	NULL

图 1—7 关系的二维表形式

2. 元组

在一个用二维表方式表示的具体关系中，水平方向的行称为元组，每一行是一个元组。元组对应存储文件中的一条具体记录。它能反映关系中某个实体的全部属性。例如，在图 1—7 所示的 Dep 关系中，每一行反映一个院系的基本情况。

3. 属性

二维表中垂直方向的列称为属性，每一列有一个属性名。与实体属性相同，属性对应存储文件中的一个字段。它能反映关系中所有实体集的某一方面特征。例如，在图 1—7 所示的 Dep 关系中，每一列反映关于院系某一个方面的基本情况。

4. 域

属性的取值范围，即不同元组对同一个属性的取值所限定的范围称为域。它是观察属性取值是否合理或正确的标准。例如，在图 1—8 所示的 Stu_kc 关系中，院系号的值域受控于 Dep 关系所给出的院系号。如果在 Stu_kc 关系中的院系号大于 6，则无法获取该院系号的名称。此外，在 Stu_cj 关系中，学生成绩在通常情况下的值域范围在 0~100 之间。负分或超出 100 分都是错误的。

学号	课程号	成绩	院系号
040001	e2	85.0	
040001	e3	75.0	
040001	e4	58.0	
030002	e1	47.0	
030002	e4	80.0	
030002	e3	NULL	
020008	e1	80.0	
020008	e5	87.0	
020003	e1	75.0	
020003	e2	NULL	
020003	e3	85.0	
020003	e4	86.0	
020004	e1	83.0	

课程号	课程名	任课教师	院系号	教师课程号
e1	英语	李敏惠	4	
e2	C语言	王博琴	1	e1
e3	计算机基础	李天河	5	e1
e4	法律基础	胡超	3	
e5	经济法	王博琴	2	e4
e6	会计电算化	王敏	2	e3

图 1—8 关系的值域

5. 关键字

关键字是用来标识一条记录的属性、属性组合或表达式，又称为键。在用于排序或索引时，它是确立排序或索引的依据。通常将关键字分为以下几种类型：

(1) 候选关键字。

如果一个关键字能够唯一地标识表中的一个元组，则称该关键字为候选关键字。通常如果某个属性的所有值都是确定的并且都不重复，则称该属性为主属性，在关系中主属性都可以充当候选关键字，而含有不确定或重复值的属性为非主属性。非主属性不能独立充当候选关键字，但可以通过属性组合而形成候选关键字。

例如，在如图 1—5 所示的 Student 关系中，(学号)、(姓名)等都是该关系的候选关键字。但(性别)和(院系号)因为有重复值，所以是非主属性，不能独立充当候选关键字，而将它们的属性组合成为(性别+院系号)或(院系号+性别)，则可以充当候选关键字，因为此时不再有重复值。

(2) 主关键字。

主关键字就是被确立用于唯一标识元组的候选关键字。在一个表中最多只能有一个主关键字，但可以有多于一个候选关键字。当某一个候选关键字被确立为主关键字后，则其他的候选关键字就不能成为主关键字了。

(3) 外部关键字。

如果表中的一个属性不是本表的主关键字或候选关键字，而是另外一个表的主关键字或候选关键字，这个属性就称为该表的外部关键字。例如，在如图 1—8 所示的 Stu_cj 关系中，(学号)和(课程号)是分别依赖于 Student 表和 Stu_kc 表的外部关键字。

在 Visual FoxPro 中，一个关系存储为一个文件，文件扩展名为“.dbf”，称为“表”。Visual FoxPro 中的术语与数据模型中的术语以及关系术语间的对应关系如表 1.1 所示。

表 1.1 术语比较

Visual FoxPro 术语	数据模型术语	关系术语
表	实体集	关系
表结构	实体型	关系模式
记录	实体	元组
字段	实体属性	属性

1.3.2 关系数据库设计

并非所有的二维表形式都可以构成关系，构成关系的二维表由于表与表之间的复杂关系及相互影响，可能出现数据冗余、插入异常、删除异常以及更新复杂等问题。因此，一个好的关系模式必须符合某些规则。

1. 关系规范化

所谓关系规范化实际上就是在设计关系数据库的过程中尽量使各个表满足规范要求。建

立关系数据库的第一步就是必须按照关系规范化的要求来设计满足数据库操作的关系。也就是说必须使数据库中的所有表都符合关系规范化的要求。

关系模式满足的确定约束条件称为范式。根据满足约束条件的级别不同，范式由低到高分为第一范式（1NF）、第二范式（2NF）、第三范式（3NF）、BC范式（BCNF）、第四范式（4NF）和第五范式（5NF）等。将一个低级的关系模式转化为若干个高级范式的过程称为关系模式的规范化过程，简称为规范化。一般情况下，数据库中的表满足 3NF 即可，因此在此只介绍前三种范式及其规范化的方法。

2. 关系规范化过程

(1) 不符合范式表。

在日常工作中，经常可以看到如表 1.2 所示的数据表，这种表中由于存在着某个数据项（如“学生成绩”）可以继续拆分的表中表，所以是一种不符合范式的表，在关系数据库中不允许非规范化的表存在。

表 1.2 非规范化关系示例

学号	姓名	学生成绩				院系情况		
		C1	C2	C3	总分	院系号	院系名称	电话号码
040001	李琪		85	75	160	2	管理学院	(027)-58746014
020003	斯琴卓玛	75	. NULL.	85	. NULL.	1	计算机科学院	(027)-62147822
...

(2) 第一范式（1NF）表。

在第一范式关系模式中每个属性的值域都是不可分的简单数据项，它是建立关系模式的最基本范式。可将表 1.2 所示的表转换成如表 1.3 所示的满足 1NF 表。

表 1.3 第一范式（1NF）关系示例

学号	姓名	C1	C2	C3	总分	院系号	院系名称	院系电话号码
040001	李琪		85	75	160	2	管理学院	(027)-58746014
020003	斯琴卓玛	75	. NULL.	85	. NULL.	1	计算机科学院	(027)-62147822
...

第一范式（1NF）表中尽管不存在可再分的数据项，但是在第一范式表中，存在着部分函数依赖关系。例如，在表 1.3 中，（总分）完全依赖于学生的（学号）或（姓名），而（院系电话号码）则依赖于（院系号）或（院系名称）。由于这种情况的存在，在表的操作上就会出现以下四个方面的问题：

① 插入异常：即在插入元组时，插入不成功的情况。例如，新成立的院系单位，或者是没有学生的教学单位，由于在插入元组时无法获取主属性值（学号），则无法插入。这是由于教学单位的各属性不依赖于主属性值（学号）而造成的。

② 删除异常：是指在删除元组时，删除了其他不该被删除的有用信息的情况。例如，某个教学单位的学生全部毕业以后，自然要注销这些学生的学号，因为（学号）是主属性，

所以要删除学号信息，必须删除整个元组，这样一来该教学单位的院系信息也将全部被删除。

③ 数据冗余度大：是指在同一表中会出现大量的重复数据。例如，在表 1.3 所列关系中，由于院系信息与每个学生信息绑定，这样，每个院系的信息都多次重复出现，从而产生了大量的冗余数据，这样，不仅造成了存储空间的浪费，也给数据库的维护带来了困难。

④ 更新复杂：冗余信息的出现，必然导致更新复杂。例如，当某个院系更名或更改电话号码时，必须对表中所有涉及该院系的（院系名称）属性值或电话号码属性值逐个进行修改，才能保证数据的一致性。在数据量较大的数据表中，很容易出现漏改或错改的现象。

如果将表 1.3 中的关系进行分解，剔除 1NF 表中不依赖于某个主属性的所有属性，就形成了如表 1.4 所示的 2NF 关系。

(3) 第二范式 (2NF) 表。

表 1.4 第二范式 (2NF) 关系示例

学号	姓名	C1	C2	C3	总分
040001	李琪		85	75	160
020003	斯琴卓玛	75	. NULL.	85	. NULL.
...

在第二范式表中，每一个非主属性的数据项都完全依赖于它的主属性。但是，在 2NF 中还存在着数据传递函数依赖问题。例如，在表 1.4 中，由于（总分）是由各科成绩求和计算得出的，所以说（总分）实际上依赖于各科成绩。而各科成绩又依赖于（学号），因此非主属性（总分）传递地依赖于主属性（学号）。

由于这种传递依赖关系的存在，2NF 并不能解决所有问题。由于（总分）的非独立性，使其成为一个冗余属性，当对（总分）的属性值进行修改时，会造成修改值与计算值不一致的异常情况，要进一步解决问题，最根本的方法就是对表进行处理，消除这种非主属性对主属性的传递函数依赖关系，使之满足更高的范式要求。

(4) 第三范式 (3NF) 表

在第三范式表中，如表 1.5 所示，任何一个非主属性都是独立地依赖于主属性，而不存在传递依赖关系。

表 1.5 第三范式 (3NF) 关系示例

学号	姓名	C1	C2	C3	院系号	院系名称	院系电话号码
040001	李琪		85	75	2	管理学院	(027)-58746014
020003	斯琴卓玛	75	. NULL.	85	1	计算机科学院	(027)-62147822
...

(a)

(b)

在关系数据库中，数据库中的各表范式越高，建立的表就越多，表之间的关系也就越复杂。一般情况下，达到第三范式就能够较好地解决插入异常、删除异常、数据冗余度大和更新复杂等问题。

尽管表 1.5 中的学生成绩表满足了第三范式要求，但由于每个学生选学课程不同，因此，随着学生数的增加，关系中的空白数据也会逐步增多，对表的操作和维护带来一定的麻