

全国计算机等级考试
试题精讲丛书

二级 Visual FoxPro

罗运纶 主编 林定移 编著



国防工业出版社

National Defense Industry Press <http://www.ndip.cn>

全国计算机等级考试试题精讲丛书

二级 Visual FoxPro

罗运纶 主编

林定移 编著

国防工业出版社

·北京·

内 容 简 介

本书是按照教育部考试中心颁布的最新考试大纲和指定教材编写的。

本书共分为6章。前4章内容涵盖数据库基础知识、Visual FoxPro数据库及其操作、关系数据库标准语言SQL、Visual FoxPro可视化应用程序设计,第五章是5套等级考试笔试的模拟试题,第六章是10套等级考试上机模拟试题。附录中给出了Visual FoxPro 6.0常见文件类型。

本书具有针对性强、试题覆盖面广的特点,非常适合广大准备参加全国计算机等级考试二级Visual FoxPro考试的考生进行考前训练,也可作为大学本科教学用书,大中专、高职与各类培训人员的培训教材,有关技术人员的参考用书。

图书在版编目(CIP)数据

二级 Visual FoxPro/ 林定移编著. —北京:国防工业出版社, 2004. 3

(全国计算机等级考试试题精讲丛书/ 罗运纶主编)

ISBN 7-118-03396-0

I. 二... II. 林... III. 关系数据库—数据库
管理系统, Visual FoxPro—水平考试—自学参考资料
IV. TP311.138

中国版本图书馆 CIP 数据核字(2004)第 002809 号

国防工业出版社 出版发行

(北京市海淀区紫竹院南路 23 号)

(邮政编码 100044)

涿中印刷厂印刷

新华书店经售

*

开本 787×1092 1/16 印张 17½ 400 千字

2004 年 3 月第 1 版 2004 年 3 月北京第 1 次印刷

印数:1—4000 册 定价:24.00 元

(本书如有印装错误,我社负责调换)

丛书编委会

主 编 罗运纶

编 委 (以姓氏笔画为序)

王颖欣 尹 乾 平 澄

师书恩 林定移

前 言

为促进我国计算机知识的普及,提高全社会的计算机应用水平,适应国民经济信息化的需要,教育部考试中心自 1994 年起开始推行全国计算机等级考试。计算机等级考试为社会提供了一个统一、公正和客观的考核标准,深受社会各界欢迎,应试者甚众。

为了满足考生复习应考的需要,我们依据新的计算机水平考试大纲,编写了本书,作为 Visual FoxPro 等级考试的辅导教材。

Visual FoxPro 6.0 关系数据库管理系统是新一代小型数据库管理系统的杰出代表,它以强大的性能、完整而又丰富的工具、较高的处理速度、友好的界面以及完备的兼容性等特点,备受广大用户的欢迎。

Visual FoxPro 6.0 及其中文版是可运行于 Windows 95/98/2000/XP/2003 平台的 32 位数据库开发系统,它不仅可以简化数据库管理,而且能使应用程序的开发流程更为合理。Visual FoxPro 6.0 使组织数据、定义数据库规则和建立应用程序等工作变得简单易行。利用可视化的设计工具和向导,用户可以快速创建表单、查询和打印报表。

Visual FoxPro 6.0 还提供了一个集成化的系统开发环境,它不仅支持过程式编程技术,而且在语言方面作了强大的扩充,支持面向对象可视化编程技术,并拥有功能强大的可视化程序设计工具。目前,Visual FoxPro 6.0 是用户收集信息、查询数据、创建集成数据库系统、进行应用系统开发较为理想的工具软件。

本书讲解简明扼要,层次分明,面向应用。在每一章的后面均有等级考试的例题讲解,最后两章是与历年等级考试要求一致的 5 套笔试模拟题和 10 套上机操作题,供读者复习参考。

本书也可以选作高等院校及其他各类计算机培训班的微机数据库课程教学用书。对于计算机应用人员和计算机爱好者,本教程也是一本实用的自学

参考书。

本书作者有十多年使用数据库的实践经验,并讲授数据库课程多年。参加本书编纂工作的还有杨秀丽、兰耀志、孙太兵等同志。

由于时间仓促与水平有限,书中的疏漏或错误之处在所难免,在此恳请广大读者不吝赐教。

编 者

目 录

第一章 数据库基础知识.....	1
知识点睛.....	1
试题剖析	21
第二章 Visual FoxPro 数据库及其操作	27
知识点睛	27
试题剖析	60
第三章 关系数据库标准语言 SQL	70
知识点睛	70
试题剖析	82
第四章 Visual FoxPro 可视化应用程序设计	91
知识点睛	91
试题剖析.....	125
第五章 笔试模拟试题及答案.....	134
笔试模拟试题一.....	134
答案.....	138
笔试模拟试题二.....	143
答案.....	149
笔试模拟试题三.....	153
答案.....	158
笔试模拟试题四.....	163
答案.....	169
笔试模拟试题五.....	173
答案.....	179
第六章 上机模拟试题及答案.....	184
上机模拟试题一.....	184
答案.....	185
上机模拟试题二.....	196
答案.....	198
上机模拟试题三.....	207
答案.....	208
上机模拟试题四.....	215
答案.....	216

上机模拟试题五.....	225
答案.....	226
上机模拟试题六.....	231
答案.....	232
上机模拟试题七.....	239
答案.....	240
上机模拟试题八.....	249
答案.....	249
上机模拟试题九.....	254
答案.....	255
上机模拟试题十.....	262
答案.....	263
附录 Visual FoxPro 6.0 常见文件类型	270

第一章 数据库基础知识



知识点睛

计算机应用人员应该掌握数据库系统的基础知识,熟悉数据库管理系统的特点,这样才能开发出适用的数据库应用系统。本章首先介绍数据库的基本概念,然后讲述关系数据库以及数据库设计方面的知识,最后介绍 Visual FoxPro 的数据运算。

一、数据库的基本概念

数据库管理系统是处理数据的有效工具,在此应该首先了解数据库等的概念,以及计算机数据管理的发展历程。

(一) 数据库的基本概念

1. 数据

数据(Data)是描述现实世界各种信息的符号记录,它可以有多种不同的表现形式,如数字、文字、图形、图像、声音等。数据是信息的载体,信息的具体表现形式。

2. 数据库

数据库(DataBase, DB)是长期存储在计算机中、有组织、可共享的数据的集合。数据库的特点是:按照一定的数据模型组织;独立、易扩展;较小的冗余度;可共享。

3. 数据库管理系统

数据库管理系统(DataBase Management System, DBMS)是管理数据库的软件(数据库管理软件),用于建立、运用和维护数据库,位于用户和操作系统之间。

4. 数据库系统

数据库系统(DataBase System, DBS)由计算机系统和数据库构成,是计算机化的记录保持系统,其目的是存储信息和产生所需要的信息。数据库系统一般包含数据库(DB)、数据库管理系统(DBMS)、用户(Users)和硬件环境。

5. 数据库技术

数据库技术是研究数据库的结构、存储、设计、管理和使用的一门软件学科,它是在操作系统的文件系统基础上发展起来的。数据库技术要用到数据结构、集合论、数理逻辑等理论知识,因此,数据库技术是一门综合性较强的学科。

(二) 数据管理技术的发展

产生于 20 世纪 60 年代末的数据库技术是计算机软件领域的一个重要分支。数据管理技术大致经过以下 3 个阶段。

1. 人工管理阶段(20 世纪 50 年代中期)

这一阶段的计算机主要用于科技计算。外存只有磁带、卡片和纸带等;软件只有汇

编语言，尚无数据管理方面的软件；数据处理方式是批处理。

人工管理阶段的数据管理特点是：数据不保存；没有专用的软件对数据进行管理；只有程序概念，没有文件概念；一组数据对应一个程序，即数据是面向程序的。

2. 文件系统阶段（20 世纪 50 年代后期—20 世纪 60 年代中期）

这一阶段的计算机不仅用于科技计算，还用于信息管理。外存已有了磁盘、磁鼓等直接存取的存储设备；软件中已有专门管理数据存储的软件——文件系统，它是操作系统的一部分，有时也称为信息处理模块；数据处理方式有批处理，也有联机实时处理。

文件系统阶段的数据管理特点是：数据可长期保存在外存的磁盘上；数据的逻辑结构与物理结构存在区别，但较简单；文件组织已多样化，具有索引文件、链接文件和直接存取文件等；数据不再属于某个特定的程序，可以重复使用。

3. 数据库系统阶段（20 世纪 60 年代末—至今）

随着数据管理规模一再扩大，数据量急剧增加，为了提高效率，开始只是对文件系统加以扩充，研制成倒排文件系统等。为了克服文件系统的缺陷，在 20 世纪 60 年代末产生了数据库技术，从而对数据进行更高级、更有效的管理。

数据库阶段的数据管理特点如下所述。

(1) 用数据模型表示复杂的数据。数据模型不仅描述数据本身的特点，还应描述数据之间的联系。

(2) 有较高的数据独立性。数据的逻辑结构与物理结构之间差别可以很大。

(3) 数据库系统为用户提供了方便的用户接口，用户可使用查询语言或终端命令操作数据库，也可以用程序方式操作数据库。

(4) 数据库管理系统提供数据安全性、数据完整性、数据库的并发控制和数据库的恢复 4 方面的数据控制功能。

(三) 数据模型

1. 数据的描述

数据处理涉及不同的数据描述领域。从事物的特性到计算机的具体表示，实际上经历了 3 个数据领域：现实世界、信息世界和机器世界。

1) 现实世界

存在于人们头脑之外的客观世界称为现实世界。现实世界的数据是客观存在的各种报表、图表和查询格式等原始数据。从中抽取数据库技术所研究的数据，分门别类，综合出系统所要的数据。

2) 信息世界

信息世界是现实世界在人们头脑中的反映，人们把它用文字或符号记载下来。在信息世界中，数据库技术用到下列术语。

(1) 实体(Entity)：客观存在并且可以相互区别的事物称为实体。例如，一个学生、一门课程等。实体也可以是抽象的事件。例如，选修课程、一次借书活动等。

(2) 实体集(Entity Set)：性质相同的同类实体集合称为实体集。例如，全校所有的学生、学生选修课程的活动等。

(3) 属性(Attribute)：实体有若干特性，每一个特性称为属性。每个属性有一个值域，其类型可以是整型、实型或字符型。例如，学生有学号、姓名、性别、年龄等属性，其

前 3 个属性的类型一般为字符型，后一个属性的类型为整数型。

(4) 键(Key): 惟一标识实体集每个实体的属性或几个属性的组合称为实体的键，有时也称为码、关键字或实体标识符。例如，学生的学号可以作为学生实体的键。

3) 机器世界

信息世界的信息在机器世界中以数据形式存储。机器世界中数据描述的术语有以下 4 个。

(1) 字段(Field): 标记实体属性的符号集称为字段或数据项。它是可以命名的最小数据单位，字段的命名往往与属性名相同。

(2) 记录(Record): 字段的有序集合称为记录。一般用一个记录描述一个实体，所以记录又可以定义为能完整描述实体的符号集。

(3) 文件(File): 同一类记录的集合称为文件。文件是描述实体集的，所以它又可以定义为描述一个实体集的所有符号集。例如，所有的学生记录构成一个学生文件，学生选修课程的记录构成选修文件。

(4) 键(Key): 惟一标识文件中每个记录的字段或字段集称为文件的键，或记录的键、记录的码。这个概念与实体的键概念是一致的。

2. 数据联系的描述

在现实世界中，事物是相互联系的。这种联系必然要在信息世界中有所反映，即实体并不是孤立静止存在的。实体的联系有两类：一类是实体内部的联系，反映在数据上是同一记录内部各字段间的联系；另一类是实体与实体之间的联系，反映在数据上是记录之间的联系。

两个不同实体集的实体间联系主要有以下 3 种情况。

(1) 一对一联系(1:1): 如果实体集 A 中实体至多和实体集 B 中一个实体有联系，反之亦然，那么实体集 A 和实体集 B 的联系称为一对一联系，记为 1:1。例如，班级与班长之间，一个班级只有一个班长。

(2) 一对多联系(1:m): 如果实体集 A 中每个实体与实体集 B 中任意个(零个或多个)实体有联系，而实体集 B 中每个实体至多和实体集 A 中一个实体有联系，那么实体集 A 和实体集 B 的联系称为一对多联系，记为 1:m。例如，班级与学生之间，一个班级有多名学生。

(3) 多对多联系(m:n): 如果实体集 A 中每个实体与实体集 B 中任意个(零个或多个)实体有联系，反之亦然，那么实体集 A 和实体集 B 的联系称为多对多联系，记为 m:n。例如，学生与课程之间，一个学生可以选修多门课程，一门课程可以被多名学生选修。

3. 数据模型的要素

数据模型是严格定义的一组概念的集合，这些概念精确描述系统的静态特性、动态特性和完整性约束条件。数据模型通常由数据结构、数据操作和完整性约束 3 部分组成。

1) 数据结构

数据结构是所研究对象的类型集合，这些对象是数据库的组成成分。它一般可以分为两类：一类是与数据类型、内容有关的对象；另一类是与数据之间联系有关的对象。数据库系统中通常按照数据结构的类型来命名数据模型，例如，层次结构、网状结构和关系结构的模型分别命名为层次模型、网状模型和关系模型。

2) 数据操作

数据操作是指对数据模型中各种对象型的实例所允许执行的所有操作，即操作的集合，包括操作及有关的操作规则。数据库中主要有检索和更新（包括插入、删除和修改）两大类操作。

数据结构是对系统静态特性的描述，数据操作是对系统动态特性的描述。

3) 数据的完整性约束条件

数据的完整性约束条件是完整性规则的集合。完整性规则是提供的数据库模型中数据及其联系所具有的制约和依存规则。这些规则用来限定基于数据库模型的数据状态及状态的变化，以保证数据库中数据的正确、有效和一致。

4. 主要数据模型

数据库领域中最常见的数据模型有 3 种，它们分别是层次模型（Hierarchical Model）、网状模型（Network Model）和关系模型（Relational Model）。其中，层次模型和网状模型统称为非关系模型，在关系模型出现以前，它们是常用的数据模型。

1) 层次模型

在数据库中，把满足 2 个条件的基本层次联系的集合称为层次模型，这两个条件为：有且仅有一个节点无双亲，这个节点称为根节点，其他节点有且仅有一个双亲。在层次模型中，同一双亲的子节点称为兄弟节点，没有子节点的节点称为叶节点。

层次数据库采用层次模型作为数据的组织方式。最典型层次模型的数据库管理系统是 IMS，它是“信息管理系统”的英文缩写，于 1968 年由 IBM 公司推出。

2) 网状模型

在数据库中，把满足 2 个条件的基本层次联系的集合称为网状模型，这两个条件为：允许有一个以上的节点无父节点，一个节点可以有多个父节点。

网状数据库采用网状模型作为数据的组织方式。其典型代表是 CODASYL（Conference On Data Systems Language）系统，这是 20 世纪 70 年代数据系统语言研究会 CODASYL 下属的数据库任务组（DataBase Task Group, DBTG）提出的一个系统方案，所以又称为 DBTG 报告。

相比于层次模型，网状模型适用于描述较为复杂的现实世界。数据库技术的发展历史说明，层次数据库是数据库应用的先驱，而网状数据库则对数据库的概念、方法、技术进行了较全面的发展。

3) 关系模型

关系模型是数据库领域所讨论的模型中最重要的模型。自 20 世纪 80 年代以来，计算机厂商推出的 DBMS 产品几乎都是支持关系模型的。在用户看来，关系模型中数据的逻辑结构（即数据结构）就是一张二维表。

(1) 关系模型：用二维表结构来表示实体及实体间联系的模型称为关系模型。

(2) 属性和值域：在二维表中的列（字段、数据项）称为属性，列值称为属性值，属性值的取值范围称为值域。

(3) 元组与关系：在二维表中的行（记录的值）称为元组，元组的集合称为关系。

(4) 关键字、键或码：在关系的诸属性中，能够用来惟一标识元组的属性（或属性组合）称为关键字，即关系中的元组由关键字的值来惟一确定。在一个关系中，关键字

的值不能为空。有些关系的关键字是由单个属性组成的，还有一些关系的关键字是由若干个属性组合而成的。例如，成绩表（学号、考试科目、成绩），它的关键字由（学号、考试科目）属性的组合构成。

(5) 候选关键字或候选码：如果在一个关系中，存在多个属性或属性组合都能用来惟一标识该关系的元组，这些属性或属性组合都称为该关系的候选关键字或候选码。

(6) 主关键字、主键或主码：在一个关系的若干个候选关键字中指定作为关键字的属性或属性组合称为该关系的主关键字或主码。

(7) 外部关键字、外键或外码：当关系中的某个属性或属性组合虽不是该关系的关键字或只是关键字的一部分，但却是另一个关系的关键字时，称该属性或属性组合为这个关系的外部关键字或外键。

(8) 主表与从表：主表和从表是指与外键相关联的两个表，以外键作为主键的表称为主表，外键所在的表称为从表。

二、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。关系模型与以往的非关系模型最大的不同是，它建立在严格的数学概念的基础上。1970年IBM公司的研究员E. P. Codd发表了题为《大型共享数据库的关系模型》的论文，提出数据库的关系模型，奠定了关系数据库理论基础。

关系数据库产品一问世，就以其简单清晰的概念、易懂易学的数据库语言，使用户不需了解复杂的存取路径细节，不需说明怎么干，只需指出干什么，就能操作数据库，深受广大用户喜爱，涌现出许多性能良好的商品化的关系数据库管理系统，即RDBMS。例如，著名的DB2、ORACLE、SYBASE、INFORMIX等都是关系型数据库管理系统，而且关系数据库产品也从单一的集中式系统发展到可在网络环境下运行的分布式系统，从封闭式系统逐步发展到开放式系统，从联机事务处理到支持信息管理、辅助决策，系统的功能不断完善，数据库的应用领域迅速扩大。

(一) 关系模型

关系模型是最重要的一种数据库模型。自20世纪80年代以来，计算机厂商所推出的数据库管理系统产品几乎都是支持关系模型的。

关系操作通过关系语言实现。关系语言的特点是高度非过程化，即用户只需要说明做什么，而不必说明怎样做，这也是关系数据库的主要优点之一。

早期关系操作有两种表示方式：关系代数与关系演算。关系代数通过对关系的运算来表达查询，其操作对象和结果都是关系，关系代数的运算包括传统的集合操作和专门的关系操作两类。关系演算是用谓词来表达查询要求的方式。

数据库的数据完整性是指数据库中数据的正确性、有效性和一致性，包括以下2个方面。

(1) 与现实世界中应用需求的数据的相容性和正确性。

(2) 数据库内数据之间的相容性和正确性。

例如，学生的学号必须惟一，学生的性别只能是男或女，学生所选修的课程必须是已开设的课程等。可见，数据库中数据是否具备完整性，关系到数据库系统能否真实地

反映现实世界。因此，数据库数据的完整性是十分重要的。

(二) 关系模型的数据完整性

关系模型中的数据完整性是指数据库中数据的正确性、有效性和一致性。数据完整性由数据完整性规则来维护，包括实体完整性、参照完整性和用户定义的完整性。

1. 实体完整性规则

关系对应于现实世界中的实体或联系，而现实世界中实体是可区分的，也就是说每个实体具有惟一性标识，在关系模型中，是由主码做惟一标识的。若主码取空值，则说明这个实体不可标识，即不可区分，这显然是错误的，与现实世界应用环境矛盾。因此不存在这样的无标识实体，从而引入实体完整性的概念。

实体完整性是指关系的主属性，即主码的组成不能为空，也就是关系的主属性不能是空值（NULL）。若所定义的实体是“学生选课程”，由于其主码为组合属性（学号、课程号），则该实体的任一元组中，学号和课程号这2个属性的值均不得为空值，否则就违反了实体完整性规则。

2. 参照完整性规则

参照完整性的定义：如果基本关系 R_2 的属性中，包含与另一基本关系 R_1 的主码 K 相对应的属性组 F ，那么 F 称为外部码。对 R_2 每个元组在 F 上的取值只允许两种可能，一是空值；二是等于 R_1 中某个元组的主码值。 K 与 F 定义在同一域中。

例如，对于学生关系（学号、姓名、性别、院系号）和院系关系（院系号、院系名称）而言，其中院系关系中的院系号是主码。学生关系中，每个学生也有院系号一项，表明这个学生是在哪个院系上学的，学生关系中的“院系号”属性与院系关系中的院系号属性相对应，学生关系中院系号则是外部码。

从上面的例子可看到：在学生关系中每个学生的院系号一项，要么取空值，表示这个学生是新生，还未分配到任何一个院系；要么取值必须与院系关系中的某个元组的院系号相同，表示这个学生分配到某个院系学习。这就是参照完整性。若是学生关系中某个学生的院系号取值不能与院系关系中任何一个元组的院系号值相同，表示这个学生被安排到不属于所在单位的院系学习，这与实际应用环境是不相符的，显然是错误的。

实体完整性与参照完整性是由系统自动支持的，即在建立关系（表）时只要说明谁是主码、谁参照于谁，系统将自动进行此类完整性的检查。

3. 用户定义的完整性规则

用户定义的完整性规则是针对某一应用环境的完整性约束条件，它反映了某一具体应用所涉及的数据应满足的要求，如定义职工表年龄字段的取值范围只能是 18~60 之间。系统提供定义和检验这类完整性规则的机制，其目的是用统一的方式由系统来处理它们，而不再由应用程序来完成这项工作。

在实际系统中，这类完整性规则一般在建立库表的同时进行定义，应用编程人员不需再作考虑。如果某些约束条件没有建立在库表一级，则应用编程人员应在各模块的具体编程中通过程序进行检查和控制。

(三) 关系运算

关系代数是关系操纵语言的一种传统表示方式，它以集合代数为基础，但它的运算对象和运算结果均为关系。关系代数也是一种抽象的查询语言，它通过对关系的运算来

表达查询。

任何一种运算都是将一定的运算符作用于一定的运算对象上，从而得到预期的运算结果，所以运算对象、运算符、运算结果是运算的 3 大要素。关系代数的运算对象是关系，它将一定的关系代数运算符作用于一定的关系上，得到预期的运算结果亦为关系。

关系代数的运算可分为传统的集合运算和专门的关系运算两类。

在关系代数运算中，把基本的关系代数运算经过有限次复合的表达式称为关系代数运算表达式（简称为代数表达式）。这种表达式的运算结果仍是一个关系，可以用关系代数表达式表示所需要进行的各种数据库查询和更新处理需求。

1. 传统的集合运算

这类运算将关系看成是元组的集合，其运算是从关系的水平方向，即从行的角度来进行的，包括并、交、差、广义笛卡儿积。

R 和 S 的并 ($\cup$)，是 R 中的元素和 S 中的元素共同组成的集合。假如某个元素既在 R 中出现，又在 S 中出现，那么它也只能在 $R \cup S$ 中出现一次。

R 和 S 的交 ($\cap$)，是既出现在 R 中又出现在 S 中的元素组成的集合。

R 和 S 的差 ($-$)，是只在 R 中出现，不在 S 中出现的元素组成的集合。

2 个关系 R 和 S 的笛卡儿积记作 $R \times S$ ，它是一个新关系，其关系模式是 R 和 S 的模式 的并集。 $R \times S$ 是把 R 和 S 的元组以所有可能的方式组合起来，因此， $R \times S$ 拥有的元组数量应该是 R 的元组数与 S 的元组数的乘积。

2. 专门的关系运算

这类运算不仅涉及行而且涉及列，包括投影选择、连接、除。

投影运算符是 Π ，该运算符作用于关系 R 将产生一个新关系 S ， S 只具有 R 的某几个属性列。

选择运算符是 σ ，该运算符作用于关系 R 也将产生一个新关系 S ， S 的元组集合是 R 的一个满足某条件 C 的子集。

连接也称为 θ 连接，它是从 2 个关系的笛卡儿积中选取它们的属性间满足一定条件的元组，关系 R 和 S 基于条件 C 来连接。

连接运算中有 2 个最为重要同时也是最为常用的连接：一种是等值连接，另一种是自然连接。若 θ 为 “=” 的连接运算称为等值运算，它是从关系 R 与 S 的笛卡儿积中选取属性值相等的那些元组。自然连接是一种特殊的等值连接，它要求 2 个关系中进行比较的分量必须是相同的属性组，并要求在结果中将重复的属性去掉，即 R 和 S 具有相同的属性组。

三、数据库设计基础

如果使用较好的数据库设计过程，就能迅速、高效地创建一个设计完善的数据库，为访问所需信息提供方便。在设计时打好坚实的基础，设计出结构合理的数据库，将会节省日后整理数据库所需的时间，并能更快地得到精确的结果。

(一) 数据库设计原则

数据库应用系统与其他计算机应用系统相比，一般都具有数据量庞大、数据保存时间长、数据关联比较复杂、用户要求多样化等特点。设计数据库的目的实质上是设计出

满足实际应用需求的实际关系模型。

为了合理组织数据，应遵从以下基本设计原则。

(1) 关系数据库的设计应遵从概念单一化“一事一地”的原则。

一个表描述一个实体或实体间的一种联系，避免设计大而杂的表。通过将不同的信息分散在不同的表中，可以使数据的组织工作和维护工作更简单，同时也易保证建立的应用程序具有较高的性能。例如，应当把学生信息保存到学生表中，把有关课程的信息保存到课程表中，把学生选课的有关信息，包括所选课程的成绩保存到选课表中。

(2) 避免在表之间出现重复字段。

除了保证表中有反映与其他表之间存在联系的外部关键字之外，尽量避免在表之间出现重复字段。这样做目的是使数据冗余尽量小，防止在插入、删除和更新时造成数据的不一致。例如，在课程表中有了课程名字段，在选课表中就不应有课程名字段，必要时可以通过两个表的连接找到。

(3) 表中的字段必须是原始数据和基本数据元素。

表中不应包括通过计算可以得到的二次数据或多项数据的组合。例如，在职工表中应当包括出生日期字段，而不应包括年龄字段。在特殊情况下可以保留计算字段，但是必须保证数据的同步更新，这可通过 Visual FoxPro 6.0 的触发器来保证重复字段的同步更新。

(4) 用外部关键字保证有关联的表之间的联系。

表之间各关联依靠外部关键字来维系，使得表具有合理结构，不仅存储了所需要的实体信息，并且反映出实体之间客观存在的联系，最终设计出满足应用需求的实际关系模型。

(二) 数据库设计过程

在 Visual FoxPro 6.0 中设计数据库的过程，首先必须通过对用户需求进行详尽分析，才有可能设计出满足用户应用需要的数据库应用系统。

1. 需求分析

用户需求主要包括以下 3 方面。

(1) 信息需求：用户要从数据库获得的信息内容。信息需求定义了数据库应用系统应该提供的所有信息，应注意描述清楚系统中数据的数据类型。

(2) 处理需求：需要对数据完成什么处理功能及处理的方式。处理需求定义了系统的数据处理的操作，应注意操作执行的场合、频率、操作对数据的影响等。

(3) 安全性和完整性要求：在定义信息需求和处理需求的同时必须相应确定安全性、完整性约束。这需要在做需求分析时与数据库用户多交流。尽管收集资料阶段的工作非常繁琐，但必须耐心细致地了解现行业务处理流程，收集全部数据资料，如报表、合同、档案、单据、计划等，所有这些信息在后面的设计步骤中都要用到。

2. 确定数据表

确定数据库中的表是数据库设计过程中技巧性最强的一步。因为根据用户想从数据库中得到的结果（包括要打印的报表、要使用的表单、要数据库回答的问题）不一定能得到如何设计表结构的线索，还需要分析对数据库系统的要求，推敲那些需要数据库回答的问题。

仔细研究需要从数据库中取出的信息，遵从概念单一化“一事一地”的原则，即一个表描述一个实体或实体间的一种联系，并把这些信息分成各种基本实体。例如，在教学管理数据库中，把学生、教师、教材、选修等每个实体设计成一个独立的表。

3. 确定字段

下面是确定字段时需要注意的问题。

(1) 每个字段直接和表的实体相关。

首先必须确保一个表中的每个字段直接描述该表的实体，然后表示表之间的联系，确定描述另一个实体的字段是否为该表的外部关键字。

(2) 以最小的逻辑单位存储信息。

表中的字段必须是基本数据元素，而不是多项数据的组合。例如，商品名、商品类别、商品描述应创建不同的字段。

(3) 表中的字段必须是原始数据。

在通常情况下，不必把计算结果存储在表中，推导的数据可通过计算得到。例如，在库存清单中有字段：商品号、商品名称、数量、单价、总价。其中，总价是通过计算得到的二次数据，不必作为基本数据在数据库里存储。

(4) 确定主关键字字段。

数据库的每个表都必须有一个或一组字段可用以惟一确定存储在表中的每个记录，即主关键字。主关键字能迅速关联多个表中的数据，不允许在主关键字字段中有重复值或空值。例如，在销售管理数据库中，把客户编码、职工号、商品号、订单号分别指定为客户表、职工表、商品表、订单表的主关键字段。

4. 确定联系

确定联系的目的是使表的结构合理。数据库不仅存储了所需要的实体信息，并且反映出实体之间客观存在的关联。要建立 2 个表的联系，可以把其中一个表的主关键字添加到另一个表中，使两个表都有该字段。具体方法如下所述。

(1) 一对多联系。

在一对多联系中，表 A 的一个记录在表 B 中可以有多个记录与之对应，但表 B 中的一个记录最多只能有一个表 A 的记录与之对应。在联系中，“一方”使用主关键字或候选索引关键字，而“多方”使用普通索引关键字。例如，在职工管理数据库中，部门表和职工表之间就存在一对多的联系，应将部门表中的部门编码字段添加到职工表中。

(2) 多对多联系。

在多对多联系中，表 A 的一个记录在表 B 中可以对应多个记录，而表 B 的一个记录在表 A 中也可以对应多个记录。这种情况下，需要改变数据库的设计。例如，在销售管理数据库中，订单表和商品表之间的联系存在多对多的联系。为了避免数据重复存储，又要保持多对多联系，方法是创建第三个表。把多对多的联系分解成 2 个一对多联系。所创建的第三个表包含 2 个表的主关键字，在 2 表之间起着纽带的作用，称之为纽带表。

(3) 一对一联系。

如果存在一对一联系的表，首先要考虑一下是否可以把这些字段合并到一个表中。如果两个表有同样的实体，可在 2 个表中使用同样的主关键字字段。如果 2 个表有不同的实体及不同的主关键字，选择其中一个表，把它的主关键字字段放到另一个表中作为