

图书在版编目(CIP)数据

Visual FoxPro 程序设计/余毅, 单欣主编; 何定华, 吴颖编著. —北京: 人民邮电出版社, 2004.9
(世纪英才高职高专计算机系列教材)

ISBN 7-115-12539-2

I. V... II. ①余... ②单... ③何... ④吴... III. 关系数据库—数据库管理系统, Visual FoxPro—程序设计—高等学校: 技术学校—教材 IV.TP311.138

中国版本图书馆 CIP 数据核字 (2004) 第 095574 号

内 容 提 要

本书是针对高职高专教学特点而编写的基础教材, 其内容包括 Visual FoxPro 的基础知识、项目管理器的使用、数据库的建立、表单的有关操作、数据库的管理与操作、查询与视图、程序设计的常用命令与基本结构及报表与标签设计的方法等, 并详细介绍了常用控件的程序设计、数据表的表单程序设计及菜单等项目的设计方法。

本书的特点是概念清晰、编排合理、实例丰富, 有利于读者对知识的理解与掌握。同时, 本书的每个章节后均附有一定数量的习题, 便于读者复习参考。

本书适合作为各类高职高专院校的“Visual FoxPro 程序设计”课程教材或“数据库应用”课程教材, 也可作为各类培训班的相关课程教材。此外, 对于参加计算机二级 Visual FoxPro 考试的应试者, 也是一本相当实用的参考书。

世纪英才高职高专计算机系列教材

Visual FoxPro 程序设计

◆ 主 编 余 毅 单 颀
编 著 何定华 吴 颖
主 审 龙 翔
责任编辑 赵桂珍

◆ 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 010-67129264
北京密云春雷印刷厂印刷
新华书店总店北京发行所经销

◆ 开本: 787×1092 1/16
印张: 14.5
字数: 355 千字 2004 年 9 月第 1 版
印数: 1-5 000 册 2004 年 9 月北京第 1 次印刷

ISBN 7-115-12539-2/TN · 2328

定价: 22.00 元

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

前 言

近年来, Visual FoxPro 迅速替代其他同类软件, 成为个人计算机上影响最大、应用最广的关系型数据库管理系统。在数据操作方面, 它基本具备了大型 SQL 数据库管理系统的特征, 如视图、关联、数据字典、触发器和存储过程等; 在数据库设计方面, 它提供了各种数据库组件的设计向导, 以及图形化、智能化的设计过程。

Visual FoxPro 安装容易, 运行环境要求不高, 便于学习掌握, 易于为广大用户所接受。更为可取的是, 它不仅支持先进的面向对象程序设计方法, 而且提供了用以开发各种数据库应用系统界面的可视化设计工具。它一改传统程序设计的概念与方法, 使得用户只需编写少量简单的代码, 就可利用鼠标方便、快捷、直观地创建各种 Windows 风格的窗口、对话框、菜单和报表等, 从而极大地提高了应用程序界面的友好性和程序设计的效率, 因此在企事业管理和办公自动化等各方面得到了普遍的应用。

本书共分 11 章, 大致可归纳为 5 部分内容。

第 1、2、3 章主要介绍数据库和数据库管理系统的基本概念, 以及 Visual FoxPro 数据元素的基础知识。其中有关数据类型、常量、变量、函数和表达式等概念至为重要, 是学好 Visual FoxPro 的基础。

第 4、5、6 章主要讲解 Visual FoxPro 数据表和数据库的基本操作, 详细地说明了如何建立、维护、使用数据表和数据库, 如何增删和修改表中的数据记录、按要求统计汇总数据、按指定的关键字对数据记录进行物理或逻辑排序, 以及如何从单个或多个数据表中查询所需的数据等。掌握了这部分内容, 就可利用 Visual FoxPro 做一些简单的数据管理工作, 同时这些知识也是进一步学习程序设计的基础。

第 7 章讲述面向过程的程序设计方法, 这是进一步学习面向对象程序设计的基础。本章在阐述顺序、分支、循环等基本程序结构及相应流程控制语句的同时, 介绍了结构化的程序设计思想。其间通过大量的例子介绍了传统程序设计的常用方法与技巧, 力图使读者在学习理解的基础上, 能够举一反三, 迅速掌握这些方法并学会编写一些简单的应用程序。

第 8、9、10 章是本书的精华部分, 在阐述了面向对象程序设计的概念、方法与术语之后, 着重介绍了利用各种设计器可视化地设计表单、报表和菜单的方法和步骤。从某种意义上讲, 面向对象程序设计的关键就是对应用程序用户界面的设计, 而组成用户界面的各种窗口和对话框则均为表单的不同表现形式, 因此, 表单设计是这一部分最重要的内容。

第 11 章详细地讲述了开发一个基于数据库的应用系统所需的各个步骤, 包括用户需求分析与目标设计、系统功能分析与模块设计、数据库需求分析与逻辑结构设计、各功能模块的创建与调试、主文件的设置, 以及整个系统的编译、运行等。本章总结前面几章所讲的各种知识, 介绍了一些编程思想与技巧, 并将前面几章的实例内容连贯起来, 通过编译、调试, 形成一个完整的应用程序。

本书采用图文并茂的形式, 结合大量实用、丰富的实例深入浅出地讲述了面向对象编程的概念, 使读者逐步掌握 Visual FoxPro 6.0 的基本操作及面向对象编程技术, 并能

独立进行小型应用系统开发。本书可作为高职高专类院校计算机及相关专业学生的教材，也可作为科研人员的参考用书，同时，它还适合于广大计算机用户和计算机技术初学者使用。

由于作者水平有限，书中难免有错误和不足之处，欢迎广大读者批评指正。

编者

目 录

第 1 章 Visual FoxPro 的基础	1
1.1 数据库基础	1
1.1.1 数据库系统及其特点与发展历史	1
1.1.2 数据模型	2
1.1.3 关系运算	4
1.1.4 数据库管理系统	5
1.2 数据库设计	5
1.2.1 数据库设计原则	5
1.2.2 数据库设计过程	6
1.3 Visual FoxPro 6.0 简介	9
1.3.1 Visual FoxPro 6.0 的特点	9
1.3.2 Visual FoxPro 6.0 的数据类型	9
1.3.3 VFP 的命令构成和文件命名方法	12
1.3.4 文件命名方法和常用文件类型	15
习题 1	16
第 2 章 Visual FoxPro 系统的编程工具介绍	18
2.1 Visual FoxPro 6.0 系统的主界面介绍	18
2.2 项目管理器	19
2.2.1 项目管理器简介	20
2.2.2 管理项目的数据	22
2.2.3 管理项目的文档	22
2.2.4 管理项目的各种操作	23
2.3 设计器	25
2.4 工具栏	26
2.5 生成器简介	26
2.6 向导	27
2.7 表单设计器	28
2.8 Visual FoxPro 编程步骤	30
2.8.1 添加控件	31
2.8.2 由控件创建对象的方法	31
2.8.3 修改属性	32
2.8.4 编写代码	32
2.8.5 运行表单	32
2.8.6 修改表单	33
习题 2	33
第 3 章 Visual FoxPro 数据与数据运算	35

3.1 常量	35
3.2 变量	36
3.2.1 字段变量	36
3.2.2 内存变量	36
3.2.3 系统内存变量	39
3.2.4 数组	39
3.3 运算符及表达式	41
3.3.1 数值表达式及数值运算符	41
3.3.2 字符表达式及字符串运算符	41
3.3.3 日期表达式及日期运算符	41
3.3.4 关系表达式及关系运算符	42
3.3.5 逻辑表达式及逻辑运算符	42
3.3.6 名称表达式	43
3.4 常用函数	43
3.4.1 字符处理函数	43
3.4.2 数值处理函数	46
3.4.3 日期和时间处理函数	48
3.4.4 数据转换函数	49
3.4.5 与表有关的函数	50
3.4.6 显示位置函数	52
3.4.7 文件管理函数	52
3.4.8 数组函数	54
3.4.9 系统函数	56
习题 3	56
第 4 章 Visual FoxPro 数据库及其操作	59
4.1 创建数据库与表	59
4.1.1 数据库和表的基本概念	59
4.1.2 创建数据库	59
4.1.3 数据库的使用及操作	61
4.2 表的设计与操作	63
4.2.1 创建表	63
4.2.2 对表结构的操作	66
4.3 对表记录的操作	67
4.3.1 输入表记录	67
4.3.2 记录指针的定位	68
4.3.3 删除记录	69
4.3.4 修改记录	71
4.4 对表的高级操作	72
4.5 自由表	77
4.5.1 自由表的建立	77

4.5.2 将自由表添加到数据库	78
4.5.3 从数据库中移出表	79
4.6 多个表的同時使用	79
4.6.1 多工作区的概念	80
4.6.2 使用不同工作区的表	80
4.6.3 表之间的关联	81
4.7 排序	83
习题 4	83
第 5 章 查询与视图	86
5.1 查询设计器	86
5.1.1 查询设计器简介	86
5.1.2 查询设计器的查询	90
5.1.3 查询菜单的使用	93
5.2 视图	94
5.2.1 视图的概念	94
5.2.2 视图的创建	95
5.2.3 视图与数据更新	97
5.2.4 视图的使用与维护	98
习题 5	98
第 6 章 关系数据库标准语言 SQL	101
6.1 SQL 语言概述	101
6.2 SQL 语言的定义操作	101
6.2.1 建立表结构	101
6.2.2 修改表结构	105
6.2.3 将表添加到数据库	107
6.2.4 将数据库表从数据库中移去	107
6.3 SQL 语言的查询操作	108
6.3.1 简单查询	109
6.3.2 嵌套查询	111
6.3.3 计算查询	112
6.3.4 分组查询	113
6.4 SQL 语言的修改功能	114
6.4.1 添加记录	114
6.4.2 删除记录	116
6.4.3 更新记录	117
习题 6	117
第 7 章 程序设计基础	121
7.1 程序与程序文件	121
7.1.1 程序文件的建立	121
7.1.2 程序文件的运行	122

7.1.3 在程序中经常使用的基本命令	123
7.2 程序的控制结构	126
7.2.1 顺序结构	126
7.2.2 分支结构	126
7.2.3 循环结构	129
7.3 多模块程序设计	135
7.3.1 子程序	135
7.3.2 自定义函数	136
7.3.3 过程和过程文件	137
7.3.4 变量的作用域	138
7.4 程序的调试	140
7.4.1 错误类型	141
7.4.2 程序的调试	141
习题 7	143
第 8 章 利用表单设计应用程序	147
8.1 创建表单	147
8.1.1 使用表单向导创建表单	147
8.1.2 使用表单设计器创建表单	147
8.2 表单设计器	153
8.2.1 属性窗口	153
8.2.2 表单控件工具栏	153
8.2.3 控件的操作与布局	154
8.3 常用表单控件	156
8.3.1 标签、线条、形状和图像	156
8.3.2 文本框与编辑框	157
8.3.3 列表框与组合框	159
8.3.4 命令按钮与命令按钮组	162
8.3.5 复选框与选项按钮组	164
8.3.6 微调控件	166
8.3.7 计时器	167
8.3.8 表格	169
8.3.9 页框与容器	170
8.4 数据环境	173
8.4.1 数据环境设计器的使用	173
8.4.2 数据信息和数据环境	173
习题 8	175
第 9 章 使用菜单	177
9.1 设计菜单	177
9.1.1 规划菜单系统	177
9.1.2 菜单设计器简介	178

9.1.3 主菜单中的有关选项	179
9.1.4 在顶层表单中添加菜单	182
9.2 建立快捷菜单	183
习题 9	185
第 10 章 报表设计	187
10.1 创建报表和标签	187
10.1.1 创建前的准备工作	187
10.1.2 创建报表布局	188
10.1.3 创建标签	195
10.2 报表设计器	196
10.2.1 报表设计器窗口及其分区	196
10.2.2 报表菜单	196
10.2.3 报表工具栏	197
10.3 定制报表布局	197
10.3.1 添加标题和总结区	197
10.3.2 定义报表的页面	198
10.3.3 设置报表带区高度	199
10.4 报表控件的使用	200
10.4.1 标签控件	200
10.4.2 域控件	201
10.4.3 统计计算和汇总	202
10.4.4 报表变量	202
10.4.5 线条、矩形及圆角矩形控件	203
10.4.6 图片 / ActiveX 绑定控件	203
10.5 报表数据环境	204
10.6 数据分组	205
10.6.1 一级数据分组	205
10.6.2 多级数据分组	206
10.7 报表或标签的输出	207
10.7.1 预览报表或标签	207
10.7.2 打印报表或标签	207
习题 10	208
第 11 章 应用程序开发实例	209
11.1 应用程序开发步骤	209
11.1.1 需求分析	209
11.1.2 数据库设计	209
11.1.3 系统总体构架设计	210
11.1.4 创建各程序模块	211
11.1.5 主文件设计	212
11.1.6 测试、连编与发行	213

11.2 应用程序向导和应用程序生成器.....	214
11.2.1 使用“应用程序向导”	214
11.2.2 应用程序生成器	216
习题 11	220

第 1 章 Visual FoxPro 的基础

Visual FoxPro 是一种典型的第四代语言，它具有良好的图形用户界面，提出了大量可视化编程对象、丰富的工具按钮、简单的数据存取并与 dBase、FoxBase 和 FoxPro 完全兼容；它具有功能强大的面向对象程序设计工具以及客户与服务器功能。

本章将在讲述数据库相关知识的基础上，对 Visual FoxPro 系统进行简要介绍。

1.1 数据库基础

在数据库技术没有发展起来以前，人们常常用某种计算机高级语言处理数据，这种方法不但执行速度慢、数据冗余大，而且使得程序的设计和修改非常复杂。20 世纪 60 年代末期，数据库技术兴起，才使这一切有所改变。数据库技术是一种数据处理的新方法，也是信息处理中一项非常重要的新技术。由于数据库具有数据结构化、数据冗余度低、程序与数据独立性高而且易于扩充、易于程序设计等优点，因此，它得到了迅速发展和广泛应用，并使计算机在应用方面又开辟了一个新的领域。

1.1.1 数据库系统及其特点与发展历史

1. 基本概念

数据与信息是密切相关的，所谓信息（Information），是客观事物的特征在人脑中的反映。数据（Data）是符号化了的信息。所谓数据库，是按照一定方式组织起来的相互关联的数据的集合。数据库系统（DataBase System, DBS）是以数据库应用为基础的计算机系统。

2. 数据库的特点

数据库一般具有以下几个特点。

（1）数据独立

应用程序不需要了解数据实际的存取方式，通过数据库的存取指令，就可得到所需的数据，因此，当数据的存取结构变更时，仅需更改数据库系统的内部程序，而外部的应用程序完全不需要改变。

（2）数据共享

由数据库的概念可知，数据库中的数据是供所有的程序使用的，因此，数据库具有数据

共享的特点。

(3) 减少冗余

山于非数据库系统中, 每个应用方法都使用自己的数据来处理, 经常会造成数据的重复建立, 而且彼此之间的数据格式也不相同, 无法交互应用。而在数据库系统中, 仅需建立共用的数据库, 其余的应用程序则都使用该数据库, 因此, 大大减少了数据的冗余。

(4) 统一控制

数据库中的数据是各用户的共享资源, 即许多用户同时使用数据库会产生并发操作, 因此, 数据库系统提供了三方面的数据控制功能。

- 安全性控制: 防止数据被破坏和泄密, 设置安全措施, 只允许合法用户进行有限的操作。
- 完整性控制: 保证数据库中的数据在输入、修改过程中始终符合原来的定义和规定, 保证数据正确、有效、相容。
- 并发控制: 当多用户同时存取、修改数据库时, 可能会发生相互干扰而使数据库的完整性遭到破坏, 因此, 数据库系统提供了对并发操作的控制功能。

3. 数据库技术的发展

20 世纪, 随着计算机硬件和软件的发展, 数据管理技术也得到了迅速的发展, 大致经历了三个阶段, 即人工管理、文件系统、数据库系统。20 世纪 60 年代前属于人工管理阶段; 20 世纪 60 年代末属于文件系统管理阶段; 20 世纪 70 年代后期, 随着微机进入市场, dBase 被开发出来并进入了微机世界, 成为相当受欢迎的数据库管理系统, 它在 20 世纪 80 年代才进入我国。随着个人计算机的广泛应用, dBase 的产品也在不断地升级, 由最开始的 dBase II、dBase III, 最后到 dBase IV。它的诞生和发展使数据的规范化、独立化、共享性都得到了进一步的改善, 极大地推动了计算机信息管理科学的进步。

20 世纪 80 年代初, 由 Fox SoftWare 公司在个人计算机平台上开发出了一种新型的数据库管理系统 (DBMS) FoxBase, 并于 1984 年正式投放市场。该产品为在 DOS 操作系统平台下完成计算机数据处理发挥了巨大的作用。

20 世纪 90 年代, 由于操作系统平台的更换, FoxBase 已经不能适应在 Windows 操作系统环境下运行, 因此, 又开发出了运行于新平台的数据库管理系统 FoxPro, 该产品的标准版本不但增加了 100 多条全新的命令与函数, 在其他方面的功能, 如 SQL 结构化设计语言和直观的案例关系查询、窗口及菜单操作等都是前所未有的。

1992 年, Microsoft 公司收购了 Fox 公司, 因此也将 FoxPro 纳入了自己的产品系列中。在此基础上, 该公司于 1995 年推出了 Visual FoxPro 5.0 及其中文版, 1998 年发布了可视化编程语言集成包 Visual FoxPro 6.0 版本。

1.1.2 数据模型

数据模型即数据库结构, 可分为三种: 层次型 (Hierarchical DataBase)、网状型 (Network) 和关系型 (Relation)。

1. 层次型数据库（如图 1.1 所示）

层次型数据库如树状结构，具有父子关系，每个父节点（如图 1.1 中的 A、B 层）可以有很多子节点，但每个子节点（B、C、D、E、F）仅允许有一个父节点。若需要子节点有很多父节点或不同的父节点的子节点间产生联系，则无法使用此模式，必须改用其他模式。

以一个国家系统来说，最高层为中央，中央下面管理许多省，省下面又管理许多县，县下面还管理许多区，依此类推。因此，层次型数据库结构的优点是：结构简单、易于操作；由上而下寻找数据比较方便，与日常生活的数据类型相似。其缺点是：寻找非直系的节点非常麻烦，必须通过多个父节点由下而上，再向下寻找，搜寻的效率太低。

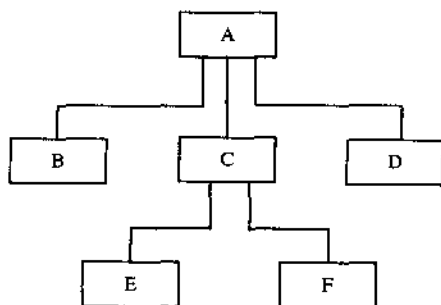


图 1.1 层次型库结构

2. 网状型数据库（如图 1.2 所示）

网状型数据库比层次型数据库更有弹性，它允许子节点有多个父节点，但缺点是路径太多，当添加或删除数据时，牵动的相关数据也很多，重建和维护数据库比较麻烦。

此结构被应用于稳定的数据库中，当数据变动较小时，此方法可以很容易地与其他的数据联系。但缺点是它不适用于动态数据库，因为动态数据库中的数据常常需要变更，数据库重建时就困难一些。

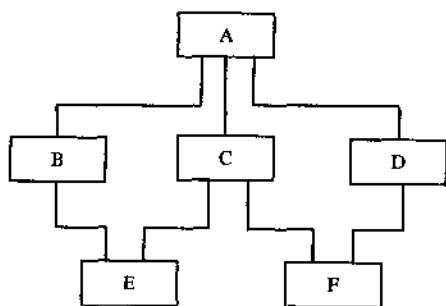


图 1.2 网状型库结构

3. 关系型数据库

关系模型中仅存在惟一的结构类型——关系。

所谓关系型数据库，就是利用关系模型组织的数据结构。我们可将关系型数据库结构看成是一个二维表，这种二维表就是关系，无论是实体本身还是实体之间的联系均用二维表表示，使得实体中的数据本身能自然地反映它们之间的联系。下面以一个由学生情况、学生成绩和组织机构三种数据的集合为例。构成学生情况关系数据模型为学号、班级、姓名……如表 1.1 所示。

表 1.1 学生情况表

学号	班级	姓名	性别	出生日期	团员	籍贯	备注
200001	计算机	于海涛	女	09-12-80	.F.	江西	memo
200002	通信	吕小雅	女	11-02-79	.T.	浙江	memo
200003	机械	齐圆圆	女	09-25-80	.T.	河北	memo
200004	光电子	张富知	女	11-11-79	.T.	四川	memo
200010	经管	史竹	女	12-20-80	.T.	河北	memo
200030	测控	李洋	女	09-10-80	.F.	浙江	memo

构成学生成绩关系数据模型为学号、班级、英语、数学、计算机……，如表 1.2 所示。

表 1.2 学生成绩表

学 号	班 级	英 语	数 学	计 算 机
200001	计算机	88	79	78
200002	通 信	87	89	88
200003	机 械	78	76	67
200004	光电子	90	88	69
200010	经 管	69	96	93
200030	测 控	86	87	82

构成课程关系数据模型为编号、学院、专业、班级……，如表 1.3 所示。

表 1.3 组织机构表

编 号	学 院	专 业	班 级
130601	机械工程	轧 钢	200201
130602	信息工程	计算机	200202
130603	经济管理	经 贸	200203
130604	电气工程	测 控	200204

在关系型数据库系统中，表中的一列称为字段，一行称为一条记录，一条记录由多个字段组成，多条记录的集合则构成一个数据库表文件。

关系型数据库结构主要的优点是：表格式的关系有利于数据库的建立；表格式的关系便于用户理解；表格式的结构便于修改和查询。

基于以上这些优点，关系型数据库在计算机中得到了广泛的应用，尤其是在个人计算机中使用得特别多。

1.1.3 关系运算

在对关系型数据库中的数据进行操作和处理时，往往要找到用户所需要的数据，此时就要对数据库中的关系进行一定的关系运算。常用的关系基本运算有三种。

1. 选择

所谓选择，是指在关系中找出满足给定条件的数据操作。选择的条件以逻辑表达式给出，使逻辑表达式条件为真的数据被选取。例如，在学生成绩表中要查找出英语成绩在 86 分以上、计算机成绩在 70 分以上学生的操作就是选择。

选择是从行的角度进行运算，在水平方向选取记录。

2. 投影

所谓投影，是指在关系中选取某些属性列组成新的关系，相当于对关系进行垂直分解，而在新关系中包含的属性个数要比原关系少。例如，在机构组成表中要查找出涉及学院和专业的数据的操作就是投影。

3. 联接

所谓联接,是将两个关系模式合并成一个更宽的关系模式,新关系中包含满足条件的元素。两个关系模式的联接是通过联接条件实现的,联接的结果是满足条件的所有记录。例如,要在学生情况表和学生成绩表中查询英语和计算机成绩都大于 80 分的学生。由于学生情况表中没有学生成绩,而学生成绩表中没有学生姓名,因此就需要将两个表联接起来,联接的条件应是这两个表中共有的学号,然后根据学号找出满足条件的记录。

1.1.4 数据库管理系统

数据库管理系统 DBMS (DataBase Management System) 是使数据库语言能正常运行的支持软件。正如使用高级语言需要解释或编译程序一样,使用数据库语言也需要有特定环境和软件的支持。数据库管理系统一般应具有下列功能。

(1) 数据定义功能

DBMS 应向用户提供“数据定义语言”DDL (Data Definition Language),用于定义、建立、修改或删除数据库结构等。

(2) 控制和管理功能

DBMS 应具有必要的控制和管理功能,其中包括:在多用户使用中对数据的并发控制、备份、恢复和转储等。

(3) 对数据库的操作功能

对数据库进行查询和检索,是数据库的主要作用。DBMS 应提供“数据操作语言”DML (Data Manipulation Language),支持用户对数据库的数据进行查询、编辑(增、删、改)等操作。例如,关系数据库中的 SQL (结构化查询语言)就具有查询语句:

```
select <字段名>
from <库表名>
where <查询条件>
```

目前,在个人计算机上使用的关系数据库管理系统都具有 DDL、DML 和 DCL (数据控制语言)为一体的各项功能。

1.2 数据库设计

一个高效的数据库应用系统还必须要有一个或多个设计合理的数据库的支持。本节将探讨数据库设计的原则和数据库设计的过程。

1.2.1 数据库设计原则

与其他计算机应用系统相比,数据库应用系统具有数据量大、数据关系复杂、用户需求多样化等特点。这就要求对应用系统的数据库和数据表进行合理的结构设计,使其不仅能够有效地存储信息,而且能够反映出数据之间存在的客观联系。

设计一个高效、合理的数据库，应遵循以下原则。

1. 概念单一化原则

概念单一化原则是指一个数据表仅描述一个实体或实体间的联系，避免设计过全的数据表。首先应分离那些需要作为单个主题而独立保存的信息，然后分析和确定这些主题之间的联系，以便在需要时将所需的信息联系在一起。通过将不同主题的信息分散在不同的数据表中，可以使数据的组织工作和维护工作更加简单，同时也有利于所开发应用程序的性能的提高。

2. 减少重复字段

在各个数据表之间，除了那些在表之间作为纽带关系的关键字字段外，应尽量避免出现重复的字段。这样做不仅能够减少数据的冗余，而且能够降低在数据插入、删除和更新操作时造成表之间数据不一致的可能性。

3. 表中字段应是基本数据元素

数据表中的字段不应包括通过计算就可以得到的“二次数据”或“多项数据”的组合，同时能够通过计算从其他字段中推导出来的字段也应尽量避免。例如，在有“出生日期”字段的数据表中，不应再包括“年龄”字段。当需要查询年龄值时，完全可以通过对“出生日期”字段的简单处理而得到准确的年龄。

在某些情况下可以保留必要的计算字段，但与此同时也必须保证数据的同步更新。例如，在成绩表中的“总分”字段，其值是通过计算各门课程成绩之和得到的，在每次更改相关课程成绩的时候，都应确保“总分”字段的值得到重新计算。

4. 用外部关键字保证表之间的关联

数据表之间的联系是靠外部关键字来维系的。数据库中的数据表不仅存储了各自所需的信息，而且还要通过外部关键字反映出与其他表之间存在的客观联系。

1.2.2 数据库设计过程

1. 数据需求分析

首先需要明确创建数据库的目的，即需要明确数据库设计的信息需求、处理需求以及对数据安全性与完整性的要求。

- 信息需求：即用户需要从数据库中获得哪些信息。信息需求决定了一个数据库应用系统应该提供的所有信息及这些信息的类型。

- 处理需求：即需要对这些数据完成什么样的处理及其处理的方式。处理需求决定了在进行数据库应用系统的数据处理操作时，应考虑执行操作的场合、操作对象、操作频率及对数据的影响等因素。

- 安全性与完整性的要求：在定义信息需求和处理需求的同时必须考虑相应的数据安

全性和完整性的要求，并确定其约束条件。

在整个应用系统设计和数据库设计中，需求分析都是一项十分重要的基础工作。必须与实际使用人员多加交流，耐心细致地了解现行业务的处理流程，收集能够收集到的全部数据资料，包括各种报表、单据、合同、档案和计划等。

2. 确定所需表

确定数据库中所应包含的表是数据库设计过程中技巧性最强的步骤。尽管在需求分析中已经基本确定了所设计的数据库应包含的内容，但在具体设计过程中仍需要仔细推敲应建立多少个独立的数据表，以及如何将这些信息分门别类地放入各自的表中。事实上，根据用户想从数据库中得到的信息，包括要查询的信息、要打印的报表和要使用的表单等，仍不能直接决定数据库中所需的表及这些表的结构，应该从分析数据库应用系统的整体需求出发，对所收集到的数据进行归纳，同时还要防止丢失有用的信息。仔细研究需要从数据库中提取的信息，遵从概念单一化的原则，将这些信息分成各种基本主题，每个主题对应一个独立的表，即用一个表描述一个实体或实体间的联系。例如，在一个销售管理系统的数据库中，可将客户、员工、商品、订单和供应商等每个实体设计成一个独立的数据表。

3. 确定所需字段

确定每个表所需的字段时应考虑以下几个原则：

- 每个字段直接和表的实体相关：即描述另一个实体的字段应属于另一个表。必须确保一个表中的每个字段直接描述本表的实体。如果多个表中重复同样的信息，则表明表中有不必要的字段。
- 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，而不应是多项数据的组合。如果一个字段中结合了多种数据，应尽量把信息分解为较小的逻辑单位，以避免日后获取单独数据时产生困难。
- 表中字段必须是原始数据：即不要包含可由推导或计算得到的字段。多数情况下，不必将计算结果存储在表中。例如，库存表中有商品号、商品名称、单价和数量等字段，而商品总价可根据单价和数量计算后得到，不必包含在库存表中。若要在表单或报表中输出商品总价，可临时通过计算而获得。
- 包括所需的全部信息：在确定所需字段时不要遗漏有用的信息，应确保所需的信息都已包括在某个数据表中，或者可由其他字段计算出来。同时在大多数情况下，应确保每个表中都有一个可以惟一标识各记录的字段。
- 确定主关键字字段：关系型数据库管理系统能够迅速地查询并组合存储在多个独立的数据表中的信息。为使其更有效地工作，数据库中的每一个表都必须至少有一个字段可用来惟一地确定表中的一个记录，这样的字段被称为主关键字字段。Visual FoxPro 能够利用主关键字字段迅速关联多个表中的数据，并按照需要把有关数据组织在一起。主关键字字段不允许有重复值或 NULL 值。例如，在员工数据表中，通常可将员工号作为主关键字字段，而不能将姓名作为主关键字字段。

4. 确定所需关系

设计数据库的一个重要步骤是确定库中各个数据表之间的关系。所确定的关系必须能够

反映出表之间客观存在的联系，同时也应使各个表的结构更加合理。

数据表之间的关系可分为3种，即：一对一关系、一对多关系和多对多关系。（假设有两个数据库表，以下简称表A、表B）

- 一对一关系：在一对一关系中，表A的一个记录在表B中只有一个记录与之对应，而表B中的一个记录在表A中也只有一个记录与之对应。如果存在一对一关系，首先应考虑是否可以把这两个表的信息合并成一个表。如果不适合合并，可在两个表中使用同样的主关键字字段建立一对一关系。例如，职工档案表和职工工资表都可以使用职工号作为主关键字字段建立联系。

- 一对多关系：一对多关系是关系型数据库中最普遍的联系。在一对多关系中，表A的一个记录在表B中可以有多条记录与其对应，而表B中的一个记录在表A中最多只有一个记录与之对应。要建立这种关系，可以将“一方”的主关键字字段拖放到“多方”的表中。“一方”应该使用主索引关键字或候选索引关键字，而“多方”可使用普通索引关键字。

- 多对多关系：在多对多关系中，表A的一个记录在表B中可以有多条记录与其对应，而表B中的一个记录在表A中也可以有多条记录与之对应。例如，在销售管理数据库中，对于订单表中的每个记录，在商品表中可以有多个记录与之对应；同样对于商品表中的每个记录，在订单表中也可以有多个记录与之对应。对于这种复杂的多对多关系，通常需要改变数据库的设计，把多对多的联系分解为两个一对多的联系。方法是创建第三个表，所创建的第三个表应包含两个表的主关键字字段，然后分别与两个表建立一对多的联系。由于这第三个表在两个表之间起着纽带作用，因而被称为“纽带表”。

5. 设计求精

数据库设计的过程实际上是一个不断返回修改、不断调整的过程。在设计中的每一个阶段都需要测试其是否能满足用户的需要，不能满足时就需要返回到前一个或前几个阶段进行修改和调整。

在确定了所需的表、字段和它们之间的联系后，应该再回过头来仔细研究和检查设计方案，看看它是否符合用户的需求，是否易于使用和维护，是否存在某些缺陷和需要改进的地方。经过反复论证和修改之后，才可以在此数据库的基础上开始进行应用系统程序代码的开发工作。下面是需要检查的几个方面：

- 在建立数据库时，是否遗忘了字段？是否有需要的信息没有包含到数据库中？如果是，它们是否包含在已创建的表中？如果它们本身不包含在已创建的表中，那就需要另外创建一个表。

- 在设计表格时，是否有包含了同样字段的表？如果是，需要考虑将与同一实体有关的所有信息合并成一个表。

- 表中是否带有大量的不属于本表的实体信息的字段？例如，在销售表中既带有销售信息字段又带有客户信息的若干字段，此时必须修改设计，确保每个表包含的字段只与一个实体有关。

- 每个表是否选择了合适的主关键字？在使用这个主关键字查找具体记录时，它是否很容易被记忆和键入？同时应确保主关键字的值不会重复。

- 某个表是否重复输入了同样的信息？如果是，则需要将该表分成两个一对多关系。