



National Computer Rank Examination
全国计算机等级考试
专用辅导教程：
二级Visual FoxPro

希赛教育等考学院 孙鸿飞 武慧娟 主编

电子工业出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

本书由希赛教育等考学院组织编写，作为全国计算机等级考试二级 Visual FoxPro 语言的辅导和培训指定教程。内容紧扣教育部考试中心新推出的考试大纲，通过对历年试题进行科学分析、研究、总结、提炼而成。

本书基于最新的考试大纲和历年试题，内容紧扣大纲，全面实用。全书内容涵盖了考试大纲规定的所有知识点，对考试大纲规定的内容有重点地进行了细化和深化。阅读本书，就相当于阅读了一本详细的、带有知识注释的考试大纲。准备考试的人员可通过阅读本书掌握考试大纲规定的知识，掌握考试重点和难点，熟悉内容的分布。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

全国计算机等级考试专用辅导教程. 二级 Visual FoxPro /孙鸿飞，武慧娟主编. -- 北京：电子工业出版社，2011.1
（全国计算机等级考试专用辅导丛书）
ISBN 978-7-121-11996-5

I. ①全… II. ①孙… ②武… III. ①电子计算机—水平考试—自学参考资料②关系数据库—数据库管理系统，Visual FoxPro—程序设计—水平考试—自学参考资料IV. ①TP3

中国版本图书馆 CIP 数据核字（2010）第 199115 号

责任编辑：李利健

印 刷：北京中新伟业印刷有限公司
装 订：

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本：860×1092 1/16 印张：20.25 字数：745.2 千字

印 次：2011 年 1 月第 1 次印刷

印 数：4 000 册 定价：39.80 元（含光盘 1 张）

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

前言

全国计算机等级考试（NCRE）由教育部考试中心主办，面向社会，用于考查非计算机专业人员的计算机应用知识与能力。考试客观、公正，得到了社会的广泛认可。

本书根据全国计算机等级考试二级 Visual FoxPro 语言的考试大纲编写而成，本书在组织和写作上，倾注了作者的许多精力和心血，相信能够提高考生的通过率，有效地完成“考试过关”提供帮助。考生可通过阅读本书，迅速掌握考试所涉及的知识点，全面进行梳理和系统学习考试大纲中的内容。

作者权威，阵容强大

希赛教育（www.educity.cn）专业从事人才培养、教育产品开发、教育图书出版，在职业教育方面具有极高的权威性。特别是在在线教育方面，稳居国内首位，希赛教育的远程教育模式得到了国家教育部门的认可和推广。

希赛教育等考学院是国内知名的进行计算机等级考试在线教育的大型教育机构，在该领域取得了很好的效果。组织大纲制订者和阅卷组成员编写了考试辅导教材近 20 本，内容涵盖了计算机等级考试的一级、二级、三级和四级的主要级别。组织权威专家和辅导名师录制了考试培训视频教程，对历年考试进行了跟踪研究和比较研究，编写了权威的全真模拟试题。希赛教育的计算机等级考试培训采取统一教材、统一视频、统一认证教师的形式，采取线下培训与线上辅导相结合的方式，确保学员在通过考试的前提下能真正学到有用的知识。

本书由希赛教育等考学院组织编写，参加编写的人员来自大学教学一线和企业研发团队，具有丰富的教学和辅导经验，对等级考试有深入的研究，具有极强的应试技巧、理论知识、实践经验和责任心。

本书由孙鸿飞、武慧娟主编，张友生审核了所有稿件。参加编写工作的有桂阳、胡钊源、谢顺和施游，王冀、符春、唐小娟和王勇参加了稿件的审核和编辑工作。

在线测试，心中有数

上学吧在线测试平台（www.shangxueba.com）为考生准备了在线测试，其中有数十套全真模拟试题和考前密卷，考生可选择任何一套进行测试。测试完毕，系统自动判卷，立即给出分数。

对于考生做错的地方，系统会自动记忆，待考生第二次参加测试时，可选择“试题复习”。这样，系统就会自动把考生原来做错的试题显示出来，供考生重新测试，以加强记忆。

如此，读者可利用上学吧在线测试平台的在线测试系统检查自己的实际水平，加强考前训练，做到心中有数，考试不慌。

诸多帮助，诚挚致谢

在本书出版之际，要特别感谢教育部考试中心计算机等级考试办公室的命题专家，编者在本书中引用了部分考试原题，使本书能够尽量方便读者阅读。在本书的编写过程中，参考了许多相关的文献和书籍，编者在此对这些参考文献的作者表示感谢。

感谢电子工业出版社的田小康老师，他在本书的策划、选题的申报、写作大纲的确定，以及编辑、出版等方面，付出了辛勤的劳动和智慧，给予了我们很多的支持和帮助。

感谢参加希赛教育计算机等级考试辅导和培训的学员，正是他们的想法汇成了本书的源动力，他们的意见使本书更加贴近读者。

由于编者水平有限，且本书涉及的内容很广，书中难免存在错漏和不妥之处，编者诚恳地期望各位专家和读者不吝指正和帮助，对此，我们将十分感激。

互动讨论，专家答疑

希赛教育等考学院（www.csaidk.com）是中国大型的计算机等级考试在线教育网站，该网站论坛是国内人气很旺的计算机等级考试社区。希赛教育等考学院拥有强大的师资队伍，为读者提供全程的答疑服务，在线回答读者的提问。

有关本书的意见反馈和咨询，读者可在希赛教育等考学院论坛“等级考试教材”板块中的“希赛教育等考学院”栏目上与作者进行交流。

希赛教育等考学院

目 录

第 1 章 算法和数据结构	1
1.1 算法与数据结构概述	1
1.1.1 算法的概念	1
1.1.2 算法的复杂度	2
1.1.3 数据结构的定义	3
1.1.4 数据结构的表示	3
1.1.5 线性结构与非线性结构	3
1.2 线性表	4
1.2.1 线性表概述	4
1.2.2 线性表的顺序存储	4
1.3 栈和队列	5
1.3.1 栈的定义与操作	5
1.3.2 队列的定义与操作	6
1.4 线性链表	7
1.4.1 线性表的链式存储	7
1.4.2 双向链表的结构及其基本运算	8
1.5 树与二叉树	8
1.5.1 树的定义	9
1.5.2 二叉树的定义及其性质	9
1.5.3 二叉树的遍历	10
1.6 查找技术	11
1.6.1 顺序查找	12
1.6.2 二分法查找	12
1.7 排序技术	12
1.8 本章习题	14
1.8.1 选择题	14
1.8.2 填空题	15
第 2 章 程序设计结构	16
2.1 程序设计方法与风格	16
2.2 结构化程序设计	17
2.3 面向对象的程序设计	17
2.3.1 面向对象特点	18
2.3.2 类和实例	18
2.3.3 消息	19
2.4 本章习题	19
2.4.1 选择题	19
2.4.2 填空题	19
第 3 章 软件工程基础	20

3.1 软件工程基本概念	20
3.1.1 软件的含义	20
3.1.2 软件工程	21
3.2 结构化分析	22
3.2.1 结构化分析方法	22
3.2.2 软件需求规格说明书	23
3.3 结构化设计方法	24
3.3.1 软件设计的基本内容	24
3.3.2 结构化设计	25
3.3.3 概要设计	26
3.3.4 详细设计	27
3.4 软件测试	27
3.4.1 软件测试概述	27
3.4.2 软件测试技术	28
3.5 程序的调试	29
3.5.1 步骤与方法	29
3.5.2 静态调试	30
3.5.3 动态调试	31
3.6 本章习题	31
3.6.1 选择题	31
3.6.2 填空题	32
第 4 章 数据库设计基础	33
4.1 数据库的基本概念	33
4.1.1 数据和信息	33
4.1.2 数据处理、数据库与数据库管理系统	33
4.1.3 数据库系统的发展	35
4.1.4 数据库系统的内部结构体系	35
4.2 数据模型	36
4.2.1 数据模型概述	37
4.2.2 E-R 模型	37
4.2.3 关系模型	38
4.2.4 数据操作	39
4.2.5 关系中的数据约束	40
4.3 关系代数	40
4.4 数据库设计	41
4.5 本章习题	42
4.5.1 选择题	42
4.5.2 填空题	42
第 5 章 Visual FoxPro 数据库基础	43
5.1 数据库基础知识	43

5.1.1	计算机数据管理的发展	43	6.3.2	字符处理函数	82
5.1.2	数据库系统	45	6.3.3	日期时间处理函数	83
5.1.3	数据库系统的特点	45	6.3.4	数据类型转换函数	83
5.1.4	数据模型	46	6.3.5	测试函数	84
5.2	关系数据库	47	6.4	程序与程序文件	84
5.2.1	关系模型	47	6.4.1	程序的概念	85
5.2.2	关系运算	49	6.4.2	程序文件的建立与执行	85
5.3	数据库设计基础	51	6.4.3	简单的输入/输出命令	87
5.3.1	数据库设计的原则	51	6.5	程序的基本结构	88
5.3.2	数据库设计的步骤	51	6.5.1	选择结构	88
5.3.3	Visual FoxPro 应用系统开发的基本步骤	52	6.5.2	循环结构	90
5.4	Visual FoxPro 系统概述	52	6.6	多模块程序设计	92
5.4.1	Visual FoxPro 6.0 的安装与启动	52	6.6.1	模块的定义和调用	92
5.4.2	Visual FoxPro 6.0 的主界面	54	6.6.2	参数传递	93
5.4.3	Visual FoxPro 6.0 工具栏的使用	55	6.6.3	变量的作用域	93
5.4.4	Visual FoxPro 6.0 的配置	57	6.7	本章习题	94
5.4.5	Visual FoxPro 6.0 的主要文件类型	58	6.7.1	选择题	94
5.5	项目管理器	60	6.7.2	填空题	99
5.5.1	创建项目	60			
5.5.2	使用项目管理器	61	第 7 章 Visual FoxPro 数据库及其操作	101	
5.5.3	定制项目管理器	62	7.1	Visual FoxPro 数据库及其建立	101
5.6	向导、设计器、生成器简介	63	7.1.1	建立数据库	101
5.6.1	向导	63	7.1.2	使用数据库	102
5.6.2	设计器	64	7.2	建立数据库表	103
5.6.3	生成器	65	7.2.1	数据库表的创建	103
5.7	本章习题	68	7.2.2	修改表结构	106
5.7.1	选择题	68	7.3	表的基本操作	106
5.7.2	填空题	70	7.3.1	打开表	106
第 6 章 Visual FoxPro 程序设计基础	72		7.3.2	浏览表	106
6.1	常量与变量	72	7.3.3	增加记录命令	108
6.1.1	常量	72	7.3.4	删除记录命令	108
6.1.2	变量	75	7.3.5	修改记录命令	109
6.2	运算符与表达式	77	7.3.6	显示记录命令	109
6.2.1	算术运算符与算术表达式	77	7.3.7	定位记录命令	110
6.2.2	字符运算符与字符表达式	78	7.4	索引	110
6.2.3	关系运算符与关系表达式	78	7.4.1	基本概念	110
6.2.4	逻辑运算符与逻辑表达式	80	7.4.2	建立索引	111
6.2.5	运算符的优先级别	81	7.4.3	使用索引	112
6.3	常用函数	81	7.5	排序	113
6.3.1	数值函数	81	7.6	数据完整性	114
			7.6.1	实体完整性与主关键字	114
			7.6.2	域完整性与约束规则	114

7.6.3	参照完整性与表之间的关联	114	8.5.1	选择题	137
7.7	自由表	115	8.5.2	填空题	147
7.7.1	数据库表和自由表	115	第 9 章	查询与视图	150
7.7.2	自由表的创建	116	9.1	查询	150
7.7.3	将自由表添加到数据库中	116	9.1.1	查询的概念	150
7.7.4	从数据库中移去表	118	9.1.2	查询设计器	150
7.8	多个表的同时使用	118	9.1.3	建立查询	151
7.8.1	工作区的概念	118	9.1.4	运行查询	154
7.8.2	使用不同工作区的表	119	9.2	视图	155
7.8.3	表的关联操作	119	9.2.1	视图的概念	155
7.9	本章习题	119	9.2.2	建立本地视图	155
7.9.1	选择题	119	9.2.3	远程视图与连接	156
7.9.2	填空题	123	9.2.4	视图与数据更新	157
第 8 章	关系数据库标准语言 SQL	124	9.2.5	使用视图	158
8.1	SQL 概述	124	9.3	本章习题	159
8.2	定义功能	125	9.3.1	选择题	159
8.2.1	表的定义	125	9.3.2	填空题	160
8.2.2	表的删除	127	第 10 章	表单设计及应用	162
8.2.3	表结构的修改	127	10.1	面向对象的概念	162
8.2.4	视图的定义	128	10.1.1	对象和类	162
8.3	操作功能	129	10.1.2	对象的属性、方法和事件	162
8.3.1	插入数据	129	10.1.3	继承与父类、子类	163
8.3.2	更新数据	130	10.2	Visual FoxPro 基类简介	163
8.3.3	删除数据	130	10.2.1	Visual FoxPro 基类	163
8.4	查询功能	130	10.2.2	容器和控件	164
8.4.1	简单查询	131	10.2.3	事件	165
8.4.2	简单的连接查询	132	10.3	创建与运行表单	166
8.4.3	嵌套查询	132	10.3.1	创建表单	166
8.4.4	几个特殊运算符	132	10.3.2	修改表单	168
8.4.5	查询结果排序	133	10.3.3	运行表单	168
8.4.6	简单的计算查询	133	10.4	表单设计器	168
8.4.7	分组与计算查询	133	10.4.1	表单设计器环境	168
8.4.8	利用空值查询	134	10.4.2	控件的操作与布局	170
8.4.9	别名与自连接查询	134	10.4.3	数据环境	172
8.4.10	内、外层相互关联查询	134	10.5	表单属性和方法	174
8.4.11	使用量词和谓词的查询	135	10.5.1	常用的表单属性	174
8.4.12	超连接查询	135	10.5.2	常用的事件和方法	175
8.4.13	集合的并运算	136	10.5.3	添加新的属性和方法	175
8.4.14	Visual FoxPro SQL SELECT 的 几个特殊选项	136	10.6	基本型控件	176
8.5	本章习题	137	10.6.1	标签	176

10.6.2	命令按钮	177	12.3.2	设计多栏报表	219
10.6.3	文本框	177	12.3.3	报表输出	219
10.6.4	编辑框	178	12.4	本章习题	220
10.6.5	复选框	178	12.4.1	选择题	220
10.6.6	列表框	179	12.4.2	填空题	221
10.6.7	组合框	179	第 13 章	应用程序的开发和生成	222
10.7	容器型控件	180	13.1	应用程序项目综合实践	222
10.7.1	命令组	180	13.1.1	系统开发基本步骤	222
10.7.2	选项组	180	13.1.2	连编项目	224
10.7.3	表格	181	13.1.3	应用程序连编及运行	227
10.7.4	页框	183	13.1.4	主程序设计	228
10.8	用户自定义类	183	13.2	使用应用程序生成器	229
10.8.1	使用类设计器创建类	183	13.2.1	使用应用程序向导	229
10.8.2	类库管理	185	13.2.2	应用程序生成器	230
10.8.3	在创建表单时使用用户 自定义类	186	13.3	本章习题	233
10.9	本章习题	187	13.3.1	选择题	233
10.9.1	选择题	187	13.3.2	填空题	234
10.9.2	填空题	190	第 14 章	上机模拟试题与解析	235
第 11 章	菜单设计与应用	191	14.1	上机应试技巧	235
11.1	Visual FoxPro 系统菜单	191	14.1.1	上机考试纪律	235
11.1.1	菜单结构	191	14.1.2	上机考试系统说明	235
11.1.2	系统菜单	192	14.1.3	上机考试步骤	236
11.2	下拉式菜单设计	193	14.2	上机模拟试题一	239
11.2.1	菜单设计的基本过程	193	14.3	上机模拟试题二	240
11.2.2	定义菜单	195	14.4	上机模拟试题三	240
11.2.3	为顶层表单添加菜单	198	14.5	上机模拟试题四	241
11.3	快捷菜单设计	198	14.6	上机模拟试题五	242
11.4	本章习题	199	14.7	上机模拟试题一答案与解析	243
11.4.1	选择题	199	14.8	上机模拟试题二答案与解析	246
11.4.2	填空题	200	14.9	上机模拟试题三答案与解析	248
第 12 章	报表的设计和应用	201	14.10	上机模拟试题四答案与解析	255
12.1	创建报表	201	14.11	上机模拟试题五答案与解析	258
12.1.1	创建报表文件	201	附录 A	习题参考答案	263
12.1.2	报表工具栏	207	附录 B	2010 年 3 月笔试试卷	294
12.2	设计报表	208	附录 C	2010 年 3 月笔试试卷答案及详解	299
12.2.1	报表的数据源和布局	208	附录 D	2010 年 9 月笔试试卷及答案	303
12.2.2	在报表中使用控件	212			
12.3	数据分组和多栏报表	217			
12.3.1	设计分组报表	217			

第1章

算法和数据结构

本章主要介绍算法、线性表、栈和队列、二叉树的概念，介绍几种常见的排序技术。结合计算机等级考试的要求，具体如表 1-1 所示。

表 1-1 考试要求

考试知识点	重要性
算法、线性表基本概念	★
栈和队列	★★★
树和二叉树	★★★★★
查找技术	★
排序技术	★★★★

1.1 算法与数据结构概述

本节的主要考点集中在算法与数据结构的基本概念上，包括算法的基本特征、复杂度，以及数据结构的表示等。

1.1.1 算法的概念

算法（Algorithm）是一系列解决问题的清晰指令，也就是说，能够对一定规范的输入，在有限时间内获得所要求的输出。如果一个算法有缺陷，或不适合于某个问题，执行这个算法将不会解决这个问题。不同的算法可能用不同的时间、空间或效率来完成同样的任务。

1. 算法的基本特征

- （1）有穷性。一个算法必须保证在执行有限步骤之后结束。
- （2）确定性。一个算法有 0 个或多个输入，以刻画运算对象的初始情况，所谓 0 个输入是指算法本身设置了初始条件。
- （3）有穷性。一个算法必须能在有限的时间内完成。
- （4）可行性。算法原则上能够精确地运行，而且人们用笔和纸做有限次运算后即可完成。

2. 算法的基本要素

- （1）算法中对数据的运算和操作：每个算法实际上是按解题要求从环境能进行的所有操作中选择合适的操作所组成的一组指令序列。
计算机可以执行的基本操作是以指令的形式描述的。一个计算机系统能执行的所有指令的集合，称为该计算机系统

的指令系统。计算机程序就是按解题要求从计算机指令系统中选择合适的指令，所组成的指令序列在一般的计算机系统中，基本的运算和操作有以下 4 类。

- 算术运算：主要包括加、减、乘、除等运算。
- 逻辑运算：主要包括“与”、“或”、“非”等运算。
- 关系运算：主要包括“大于”、“小于”、“等于”、“不等于”等运算。
- 数据传输：主要包括赋值、输入、输出等操作。

(2) 算法的控制结构：一个算法的功能不仅取决于所选用的操作，而且还与各操作之间的执行顺序有关。算法中各操作之间的执行顺序称为算法的控制结构。

3. 算法设计的基本方法

计算机算法不同于人工处理的方法，下面是工程上常用的几种算法设计，在实际应用时，各种方法之间往往存在着一定的联系。

(1) 递推法。递推法是利用问题本身所具有的一种递推关系求问题解的一种方法。它把问题分成若干步，找出相邻几步的关系，从而达到目的，此方法称为递推法。

(2) 递归。递归指的是一个过程：函数不断引用自身，直到引用的对象已知。

(3) 穷举搜索法。穷举搜索法是对可能是解的众多候选解按某种顺序进行逐一枚举和检验，并从中找出那些符合要求的候选解作为问题的解。

(4) 贪婪法。贪婪法是一种不追求最优解，只希望得到较为满意解的方法。贪婪法一般可以快速得到满意的解，因为它省去了为找最优解要穷尽所有可能而必须耗费的大量时间。贪婪法常以当前情况为基础作最优选择，而不考虑各种可能的整体情况，所以贪婪法不要回溯。

(5) 分治法。分治法是把一个复杂的问题分成两个或更多的相同或相似的子问题，再把子问题分成更小的子问题，直到最后子问题可以简单地直接求解，原问题的解即子问题解的合并。

(6) 动态规划法。动态规划是一种在数学和计算机科学中使用的，用于求解包含重叠子问题的最优化问题的方法。其基本思想是，将原问题分解为相似的子问题，在求解的过程中通过子问题的解求出原问题的解。动态规划的思想是多种算法的基础，被广泛应用于计算机科学和工程领域。

(7) 迭代法。迭代法是在数值分析中通过从一个初始估计出发寻找一系列近似解来解决问题（一般是解方程或者方程组）的过程，为实现这一过程所使用的方法统称为迭代法。

4. 良好的算法设计的要求

一个好的算法应达到如下目标：

(1) 正确性 (correctness)。算法的计算结果必须是正确的。

(2) 可读性 (readability)。可读性好有助于用户对算法的理解；不易理解的程序易于隐藏较多错误，难以调试和修改。

(3) 健壮性 (robustness)。当输入数据非法时，算法也能适当地做出反应或进行处理，而不会产生莫名其妙的输出结果。

(4) 效率与低存储量需求。效率指的是在程序执行时，对于同一个问题如果有多个算法可以解决，执行时间短的算法效率高；存储量需求指在算法执行过程中所需要的最大存储空间。

1.1.2 算法的复杂度

算法复杂度分为空间复杂度和时间复杂度。

1. 算法的时间复杂度

算法的时间复杂度，是指执行算法所需要的计算工作量。同一个算法用不同的语言实现，或者用不同的编译程序进行编译，或者在不同的计算机上运行，效率均不同。

2. 算法的空间复杂度

算法的空间复杂度是指执行这个算法所需要的内存空间。一个算法所占用的存储空间包括算法程序所占的空间、输入的初始数据所占的存储空间，以及算法执行中所需要的额外空间。

【例题 1】算法的空间复杂度是指_____。(2009 年 9 月)

- A. 算法在执行过程中所需要的计算机存储空间 B. 算法所处理的数据量
C. 算法程序中的语句或指令条数 D. 算法在执行过程中所需要的临时工作单元数

【例题分析】

由以上定义得知，此题选 A。

1.1.3 数据结构的定义

数据结构 (data structure) 是指相互之间存在一种或多种特定关系的数据元素的集合。

数据 (data) 是对客观事物的符号表示，在计算机科学中是指所有能输入到计算机中并被计算机程序处理的符号的总称。

数据元素 (data element) 是数据的基本单位，在计算机程序中通常作为一个整体进行考虑和处理。

在一般情况下，在具有相同特征的数据元素集合中，各个数据元素之间存在某种关系 (即连续)，这种关系反映了该集合中的数据元素所固有的一种结构。在数据处理领域中，通常把数据元素之间这种固有的关系简单地用前后件关系 (或直接前驱与直接后继关系) 来描述。

一般来说，数据元素之间的任何关系都可以用前后件关系来描述。

1. 数据的逻辑结构

数据结构是指反映数据元素之间关系的数据元素集合的表示。通俗地说，数据结构是指带有结构的数据元素的集合。用 D 表示数据元素的集合，用 R 来表示数据元素之间的前后件的关系，用 R 来表示。即一个数据结构可以表示为 $B = (D, R)$ ，其中 B 表示数据结构。这就是一个二元关系的表示方式所谓结构实际上就是指数据元素之间的前后件关系。

一个数据结构应包含以下两方面信息：

- (1) 表示数据元素的信息。
- (2) 表示各数据元素之间的前后件关系。

数据的逻辑结构是对数据元素之间的逻辑关系的描述。它可以用一个数据元素的集合和定义在此集合中的若干关系来表示。

2. 数据的存储结构

数据的逻辑结构在计算机存储空间中的存放形式，称为数据的存储结构 (也称为数据的物理结构)。

由于数据元素在计算机存储空间中的位置关系可能与逻辑关系不同，因此，为了表示存放在计算机存储空间中的各数据元素之间的逻辑关系 (即前后件关系)，在数据的存储结构中，不仅要存放各数据元素的信息，还需要存放各数据元素之间的前后件关系的信息。

一种数据的逻辑结构根据需要可以表示成多种存储结构，常用的结构有顺序、链接、索引等存储结构而采用不同的存储结构，其数据处理的效率是不同的。因此，在进行数据处理时，选择合适的存储结构是很重要的。

1.1.4 数据结构的表示

数据结构的表示除了用二元关系表示外，还可以直观地用图形表示。

在数据结构的图形表示中，对于数据集合 D 中的每一个数据元素用中间标有元素值的方框表示，一般称之为数据结点，并简称为结点；为了进一步表示各数据元素之间的前后件关系，对于关系 R 中的每一个二元组，用一条有向线段从前件结点指向后件结点。

在数据结构中，没有前件的结点称为根结点；没有后件的结点称为终端结点 (也称为叶子结点)。

一个数据结构中的结点可能是动态变化的。根据需要或在处理过程中，可以在一个数据结构中增加一个新结点 (称为插入运算)，也可以删除数据结构中的某个结点 (称为删除运算)。插入与删除是对数据结构的两种基本运算。除此之外，对数据结构的运算还有查找、分类、合并、分解、复制和修改等。

1.1.5 线性结构与非线性结构

根据数据结构中各数据元素之间前后件关系的复杂程度，一般将数据结构分为两大类：线性结构与非线性结构。

线性结构满足如下条件：

- (1) 有且只有一个根结点。
- (2) 每一个结点最多有一个前件，也最多有一个后件。

如果一个数据结构不是线性结构，称之为非线性结构。如果在一个数据结构中一个数据元素都没有，则称该数据结构为空。线性结构与非线性结构都可以是空的数据结构。对于空的数据结构，如果对该数据结构的运算是按线性结构的规则来处理的，则属于线性结构；否则属于非线性结构。

1.2 线性表

本节主要考查线性表的基本概念，以及线性表的顺序存储方式。

1.2.1 线性表概述

线性表是一种常用的数据结构。

在实际应用中，线性表都是以栈、队列、字符串、数组等特殊线性表的形式来使用的。由于这些特殊线性表都具有各自的特性，因此，掌握这些特殊线性表的特性，对于数据运算的可靠性和提高操作效率都是至关重要的。

线性表是一个线性结构，它是一个含有 $n \geq 0$ 个结点的有限序列。对于其中的结点，有且仅有一个开始结点，没有前驱但有一个后继结点；有且仅有一个终端结点，没有后继但有一个前驱结点。其他的结点都有且仅有一个前驱和一个后继结点。一般地，一个线性表可以表示成一个线性序列： $k_1, k_2, \dots, k_n$ ，其中 k_1 是开始结点， k_n 是终端结点。

线性结构的基本特征为：

- (1) 集合中必存在唯一的一个“第一元素”。
- (2) 集合中必存在唯一的一个“最后元素”。
- (3) 除最后一个元素之外，均有唯一的后继（后件）。
- (4) 除第一个元素之外，均有唯一的前驱（前件）。

由 n ($n \geq 0$) 个数据元素（结点） $a_1, a_2, \dots, a_n$ 组成的有限序列。

数据元素的个数 n 定义为表的长度。

当 $n=0$ 时称为空表。

常常将非空的线性表 ($n>0$) 记作： $(a_1, a_2, \dots, a_n)$

1.2.2 线性表的顺序存储

线性表的顺序存储指的是用一组地址连续的存储单元依次存储线性表的数据元素。

1. 线性表的顺序存储基本概念

线性表的顺序存储结构具备如下两个基本特征：

- (1) 线性表中的所有元素所占的存储空间是连续的；
- (2) 线性表中各数据元素在存储空间中是按逻辑顺序依次存放的。

假设线性表的每个元素需要占用 k 个存储单元，并以所占的存储位置 $ADR(a_{i+1})$ 和第 i 个数据元素的存储位置 $ADR(a_i)$ 之间满足下列关系： $ADR(a_{i+1}) = ADR(a_i) + k$

线性表第 i 个元素 a_i 的存储位置为： $ADR(a_i) = ADR(a_1) + (i-1) \times k$

公式中 $ADR(a_1)$ 是线性表的第一个数据元素的存储位置，通常称为线性表的起始位置或基址。

在 C 语言中，通常定义一个一维数组来表示线性表的顺序存储空间。

a[0]	a[1]	a[2]	a[3]	a[4]

图 1-1 顺序存储

在用一维数组存放线性表时，该一维数组的长度通常要定义得比线性表的实际长度大一些，以便对线性表进行各种运算，特别是插入运算。

在线性表的顺序存储结构下，可以对线性表做以下运算：

- (1) 在线性表的指定位置处加入一个新的元素（即线性表的插入）。
- (2) 在线性表中删除指定的元素（即线性表的删除）。
- (3) 在线性表中查找某个（或某些）特定的元素（即线性表的查找）。
- (4) 对线性表中的元素进行排序（即线性表的排序）。
- (5) 将一个线性表分解成多个线性表（即线性表的分解）。
- (6) 将多个线性表合并成一个线性表（即线性表的合并）。
- (7) 复制一个线性表（即线性表的复制）。
- (8) 逆转一个线性表（即线性表的逆转）等。

2. 顺序表的基本操作

顺序表的基本操作包括插入运算和删除运算。

1) 顺序表的插入运算

线性表的插入运算是指在表的第 i ($1 \leq i \leq n+1$) 个位置上，插入一个新结点 x ，使长度为 n 的线性表：

$$(a_1, \dots, a_{i-1}, a_i, \dots, a_n)$$

变成长度为 $n+1$ 的线性表： $(a_1, \dots, a_{i-1}, x, a_i, \dots, a_n)$

现在分析算法的复杂度。设它的值为 n 。该算法的时间主要花费在循环结点后移语句上，该语句的执行次数（即移动结点的次数）是 $n-i+1$ 。由此可看出，所需移动结点的次数不仅依赖于表的长度，而且还与插入位置有关。

当 $i=n+1$ 时，由于循环变量的终值大于初值，结点后移语句将不进行；这是最好情况，其时间复杂度为 $O(1)$ 。

当 $i=1$ 时，结点后移语句，将循环执行 n 次，需移动表中所有结点，这是最坏情况，其时间复杂度为 $O(n)$ 。

综合以上的情况，得出算法的平均时间复杂度为 $O(n)$ 。

2) 顺序表的删除运算

线性表的删除运算是指将表的第 i ($1 \leq i \leq n$) 个结点删除，使长度为 n 的线性表： $(a_1, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_n)$

变成长度为 $n-1$ 的线性表： $(a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_n)$

该算法的时间分析与插入算法相似，结点的移动次数也是由表长 n 和位置 i 决定的。

若 $i=n$ ，则由于循环变量的初值大于终值，前移语句将不执行，无须移动结点。

若 $i=1$ ，则前移语句将循环执行 $n-1$ 次，需移动表中除开始结点外的所有结点。在这两种情况下算法的时间复杂度分别为 $O(1)$ 和 $O(n)$ 。

综合以上的情况得出在顺序表上进行删除运算，平均要移动表中约一半的结点，平均时间复杂度也是 $O(n)$ 。

1.3 栈和队列

栈和队列都是特殊的线性表，其定义符合线性表的定义，其操作也类似于线性表的操作，只不过增加了一些限定而已。

1.3.1 栈的定义与操作

栈（Stack）是一种特殊的线性表。栈是只能在表的一端进行插入和删除运算配线性表，通常称插入、删除的这一端为栈顶（Top），另一端为栈底（Bottom），如图 1-2 所示。当表中没有元素时称为空栈。栈顶元素总是后插入的元素，从

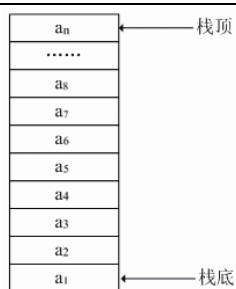


图 1-2 栈

而也是最先被删除的元素；栈底元素总是最先被插入的元素，从而也是最后才能被删除的元素。

假设栈 $S = (a_1, a_2, a_3, \dots, a_n)$ ，则 a_1 称为栈底元素， a_n 为栈顶元素。栈中元素按 $a_1, a_2, a_3, \dots, a_n$ 的次序进栈，退栈的第一个元素应为栈顶元素。换句话说，栈的修改是按后进先出的原则进行的。因此，栈称为先进后出表（FILO, First In Last Out），或“后进先出”表（LIFO, Last In First Out）。

栈的操作主要有入栈运算、退栈运算（出栈）和读栈顶元素。

（1）入栈运算：入栈运算是指在栈顶位置插入一个新元素。首先将栈顶指针加一（即 $\text{top} + 1$ ），然后将元素插入到栈顶指针指向的位置。当栈顶指针已经指向存储空间的一个位置时，说明栈空间已满，不可能再进行入栈操作。这种情况称为栈“上溢”错误。

（2）退栈运算：退栈是指取出栈顶元素并赋给一个指定的变量。首先将栈顶元素（栈顶指针指向的元素）赋给一个指定的变量，然后将栈顶指针减一（即 $\text{top} - 1$ ）。当栈顶指针为“.”时，说明栈空，不可进行退栈操作。这种情况称为栈的“下溢”错误。

（3）读栈顶元素：读栈顶元素是指将栈顶元素赋给一个指定的变量。这个运算不删除栈顶元素，只是将它赋给一个变量，因此栈顶指针不会改变。当栈顶指针为 0 时，说明栈空，读不到栈顶元素。

1.3.2 队列的定义与操作



图 1-3 队列

队列（queue）是只允许在一端删除，在另一端插入的顺序表，允许删除的一端叫做队头（front），允许插入的一端叫做队尾（rear），如图 1-3 所示。

1. 队列的运算

当队列中没有元素时称为空队列。在空队列中依次加入元素 $a_1, a_2, \dots, a_n$ 之后， a_1 是队头元素， a_n 是队尾元素。显然退出队列的次序也只能是 $a_1, a_2, \dots, a_n$ 也就是说队列的修改是依先进先出的原则进行的。因此队列也称为先进先出（FIFO, First In First Out）的线性表，或后进后出（LILO, Last In Last Out）的线性表

（1）入队操作。往队列队尾插入一个元素称为入队运算。

（2）出队操作。从队列的队头删除一个元素称为出队运算。

2. 循环队列的运算

所谓循环队列，就是将队列存储空间的最后一个位置绕到第一个位置，形成逻辑上的环状空间。

在循环队列中，用队尾指针 rear 指向队列中的队尾元素，用队头指针 front 指向排头元素的前一个位置。因此，从排头指针 front 指向的后一个位置到队尾指针 rear 指向的位置之间所有的元素均为队列中的元素。

在循环队列中进行出队、入队操作时，头尾指针仍要加 1，朝前移动。只不过当头尾指针指向向量上界（ $\text{QueueSize}-1$ ）时，其加 1 操作的结果是指向向量的下界 0。

由于入队时尾指针向前追赶头指针，出队时头指针向前追赶尾指针，故队空和队满时头尾指针均相等。因此，我们无法通过 $\text{front}=\text{rear}$ 来判断队列是“空”还是“满”。在实际使用循环队列时，为了能区分队列满还是队列空，通常还需增加一个标志值的定义如下：当 $s=0$ 时表示队列空；当 $s=1$ 时表示队列非空。

（1）入队运算。入队运算是指在循环队列的队尾加入一个新元素。首先将队尾指针进一（即 $\text{rear}=\text{rear}+1$ ），并当 $\text{rear}=\text{m}+1$ 时置 $\text{rear}=1$ ；然后将新元素插入到队尾指针指向的位置。当循环队列非空（ $s=1$ ）且队尾指针等于队头指针时，说明循环队列已满，不能进行入队运算，这种情况称为“上溢”。

（2）退队运算。退队运算是指在循环队列的队头位置退出一个元素并赋给指定的变量。首先将队头指针进一（即 $\text{front}=\text{front}+1$ ），并当 $\text{front}=\text{m}+1$ 时置 $\text{front}=1$ ；然后将排头指针指向的元素赋给指定的变量。当循环队列为空（ $s=0$ ）时，不能进行退队运算，这种情况称为“下溢”。

【例题 2】下列数据结果中，能够按照“先进后出”原则存取数据的是_____。（2009 年 9 月）

- A. 循环队列 B. 栈
C. 队列 D. 二叉树

【例题分析】

按照先进后出的原则存取数据的是栈。

1.4 线性链表

本节主要考查线性链表的存储方式和基本操作。

1.4.1 线性表的链式存储

在定义的链表中，若只含有一个指针域来存放下一个元素地址，称这样的链表为单链表或线性链表。线性表的链式存储如图 1-4 所示。

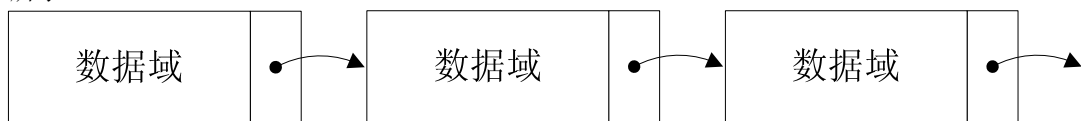


图 1-4 线性表的链式存储

在链式存储方式中, 要求每个结点由两部分组成: 一部分用于存放数据元素值, 称为数据域; 另一部分用于存放指针, 称为指针域。其中指针用于指向该结点的前一个或后一个结点 (即前件或后件)。

1. 线性链表的存储结构

用一组任意的存储单元来依次存放线性表的结点，这组存储单元既可以是连续的，也可以是不连续的，甚至是零散分布在内存中的任意位置上的。因此，链表中结点的逻辑次序和物理次序不一定相同。为了能正确表示结点间的逻辑关系，在存储每个结点值的同时，还必须存储指示其后件结点的地址(或位置)信息，这个信息称为指针(pointer)或链(link)。这两部分组成了链表中的结点结构，

链表通过每个结点的链域将线性表的 n 个结点按其逻辑次序链接在一起。由于上述链表的每一个结点只有一个链域, 所以将这种链表称为单链表 (Single Linked)。

显然,单链表中每个结点的存储地址是存放在其前驱结点 Next 域中的,而开始结点无前驱,故应设头指针 HEAD 指向开始结点。同时,由于终端结点无后继,所以终端结点的指针域为空,即 NULL。

2. 线性链表的基本运算

线性链表的基本运算包括在线性链表中查找指定元素、在线性链表的插入元素、在线性链表的删除元素。

1) 在线性链表中查找指定元素

在对线性链表进行插入或删除的运算中，总是首先需要找到插入或删除的位置，这就需要对线性链表进行扫描查找，在线性链表中寻找包含指定元素的前一个结点。

在链表中，查找是否有结点值等于给定值 x 的结点，如果有，则返回首次找到的其值为 x 的结点的存储位置；否则返回 NULL。查找过程从开始结点出发，顺着链表逐个将结点的值和给定值 x 进行比较。

2) 在线性链表的插入元素

线性链表的插入是指在链式存储结构下的线性链表中插入一个新元素。

插入运算是将值为 x 的新结点插入到表第 $i-1$ 个结点和第 i 个结点之间。因此，我们必须首先找到 a_{i-1} 的存储位置 p ，然后生成一个数据域为 x 的新结点 $*p$ ，并令结点 p 的指针域指向新结点，新结点的指针域指向结点 a_i 。

由线性链表的插入过程可以看出，由于插入的新结点取自于可利用栈，因此，只要可利用栈不空，在线性链表插入时总能取到存储插入元素的新结点，不会发生“上溢”的情况。而且，由于可利用栈是公用的，多个线性链表可以共享它，从而很方便地实现了存储空间动态分配。另外，线性链表在插入过程中不发生数据元素移动的现象，只要改变有关结点的指针即可，从而提高了插入的效率。

3) 在线性链表的删除元素

线性链表的删除是指在链式存储结构下的线性链表中删除包含指定元素的结点。

删除运算是将表的第 i 个结点删去。因为在单链表中结点 a_i 的存储地址是在其直接前趋结点 a_{i-1} 的指针域 Next 中，所以我们必须首先找到 a_{i-1} 的存储位置 p 。然后令 $p \rightarrow \text{Next}$ 指向 a_i 的直接后继结点，即把 a_i 从链上删除，最后释放结点 a 的空间。

从线性链表的删除过程可以看出，从线性链表中删除一个元素后，不需要移动表中的数据元素，只要改变被删除元素所在结点的前一个结点的指针域即可。另外，可利用栈是用于收集计算机中所有的空闲结点，因此，当从线性链表中删除一个元素后，该元素的存储结点就变为空闲，应将空闲结点送回可利用栈。

1.4.2 双向链表的结构及其基本运算

双向链表也叫双链表，是链表的一种，它的每个数据结点中都有两个指针，分别指向直接后继和直接前驱，如图 1-5 所示，所以，从双向链表中的任意一个结点开始，都可以很方便地访问它的前驱结点和后继结点。

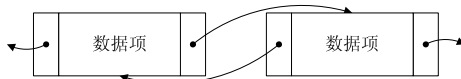


图 1-5 双向链表

1. 双向链表的基本运算

- (1) 插入：在 HEAD 为头指针的双向链表中，在值为 y 的结点之后插入值为 x 的结点，插入结点的指针变化。
- (2) 删除：在以 HEAD 为头指针的双向链表中删除值为 x 的结点，删除算法的指针变化。

2. 双链表的结构及其基本运算

在前面所讨论的线性链表中，其插入与删除的运算虽然比较方便，但还存在一个问题，在运算过程中对于空表和第一个结点的处理必须单独考虑，使空表与非空表的运算不统一。

因此，我们可以考虑建立这样的链表，具有单链表的特征，但又不需要增加额外的存储空间，仅对表的链接方式稍做改变，使得对表的处理更加方便灵活。从单链表可知，最后一个结点的指针域为 NULL，表示单链表已经结束。如果将单链表最后一个结点的指针域改为存放链表中头结点（或第一个结点）的地址，就使得整个链表构成一个环，又没有增加额外的存储空间，而满足这样条件的链表叫做循环链表。

循环链表具有以下两个特点：

- (1) 在循环链表中增加了一个表头结点，其数据域为任意或者根据需要来设置，指针域指向线性表的第一个元素的结点。循环链表的头指针指向表头结点。
 - (2) 循环链表中最后一个结点的指针域不是空，而是指向表头结点。即在循环链表中，所有结点的指针构成了一个环状链。
- 在循环链表中，只要指出表中任何一个结点的位置，就可以从它出发访问到表中其他所有的结点，而线性单链表做不到这一点。

由于在循环链表中设置了一个表头结点，因此，在任何情况下，循环链表中至少有一个结点存在，从而使空表的运算统一。

1.5 树与二叉树

本节要求考生掌握树和二叉树的基本定义，重点考查二叉树的基本性质和二叉树的遍历。

1.5.1 树的定义

树是由 n ($n \geq 0$) 个结点组成的有限集合。若 $n=0$ ，称为空树；若 $n>0$ ，则：

(1) 有一个特定的称为根 (root) 的结点。它只有直接后件，但没有直接前件。

(2) 除根结点以外的其他结点可以划分为 m ($m \geq 0$) 个互不相交的有限集合 $T_0, T_1, \dots, T_{m-1}$ ，每个集合 T_i ($i=0, 1, \dots, m-1$) 又是一棵树，称为根的子树，每棵子树的根结点有且仅有一个直接前件，但可以有 0 个或多个直接后件。

如图 1-6 所示为一棵树的示例。

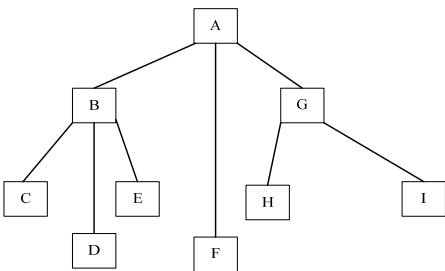


图 1-6 树的示例

1.5.2 二叉树的定义及其性质

二叉树 (binary tree) 是由 n ($n \geq 0$) 个结点的有限集合构成，此集合或者为空集，或者由一个根结点及两棵互不相交的左右子树组成，并且左右子树都是二叉树，如图 1-7 所示。二叉树可以是空集合，根可以有空的左子树或空的右子树。

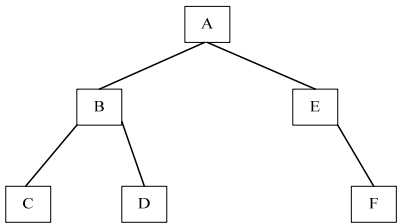


图 1-7 二叉树

1. 二叉树的特点
- 二叉树具有如下两个特点：
- (1) 非空二叉树只有一个根结点。
 - (2) 每一个结点最多有两棵子树，且分别称为该结点的左子树与右子树。

在二叉树中，一个结点可以只有左子树而没有右子树，也可以只有右子树而没有左子树。当一个结点既没有左子树也没有右子树时，该结点即叶子结点。

2. 满二叉树与完全二叉树
- 满二叉树：除最后一层外，每一层上的所有结点都有两个子结点，如图 1-8 所示。
- 完全二叉树是指这样的二叉树：除最后一层外，每一层上的结点数均达到最大值；在最后一层上只缺少右边的若干结点，如图 1-9 所示。

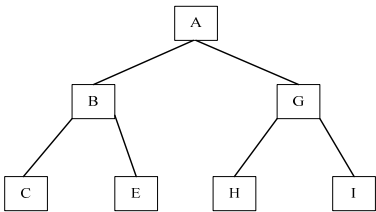


图 1-8 满二叉树

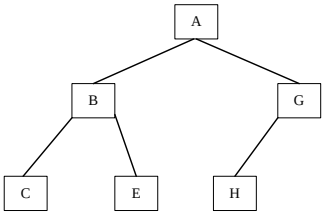


图 1-9 完全二叉树

如果一棵具有 n 个结点的深度为 k 的二叉树，它的每一个结点都与深度为 k 的满二叉树中编号为 $1 \sim n$ 的结点一一对应。

3. 二叉树的基本性质
- 性质 1：在二叉树的第 k 层上至多有 2^{k-1} 个结点 ($k \geq 1$)。
- 性质 2：深度为 m 的二叉树至多有 $2^m - 1$ 个结点。
- 性质 3：对任何一棵二叉树，度为 0 的结点 (即叶子结点) 总是比度为 2 的结点多一个。

性质 4: 具有 n 个结点的完全二叉树的深度至少为 $\lceil \log_2 n \rceil + 1$, 其中 $\lceil \log_2 n \rceil$ 表示 $\log_2 n$ 的整数部分。

性质 5: 具有 n 个结点的完全二叉树深度为 $\lceil \log_2 n \rceil + 1$ 或 $\lceil \log_2 (n+1) \rceil$ 。

性质 6: 如果对一棵有 n 个结点的完全二叉树的结点按层序编号 (从第 1 层到第 $\lceil \log_2 n \rceil + 1$ 层, 每层从左到右), 则对任意结点 i ($1 \leq i \leq n$), 有:

(1) 如果 $i=1$, 则结点 i 无双亲, 是二叉树的根; 如果 $i>1$, 则其双亲是结点 $\lceil i/2 \rceil$ 。

(2) 如果 $2i \leq n$, 则结点 i 为叶子结点, 无左孩子; 否则, 其左孩子是结点 $2i$ 。

(3) 如果 $2i+1 \leq n$, 则结点 i 无右孩子; 否则, 其右孩子是结点 $2i+1$ 。

【例题 3】某二叉树有 5 个度为 2 的结点, 以及 3 个度为 1 的结点, 则该二叉树中共有结点_____个。(2009 年 9 月)

【例题分析】

由二叉树的性质得到: 度数为 2 的结点个数=度数为 0 的结点个数+1, 所以, 度数为 0 的结点为 4 个, 总的结点个数为 12 个。

4. 二叉树的存储结构

在计算机中, 二叉树通常采用链式存储结构。用于存储二叉树中各元素的存储结点由两部分组成: 数据域与指针域。但在二叉树中, 由于每一个元素可以有两个后件 (两个子结点), 因此, 用于存储二叉树的存储结点的指针域有两个: 一个用于指向该结点的左子结点的存储地址, 称为左指针域; 另一个用于指向该结点的右子结点的存储地址, 称为右指针域。

1.5.3 二叉树的遍历

所谓遍历二叉树, 就是遵从某种次序, 访问二叉树中的所有结点, 使得每个结点仅被访问一次。

1. 前序遍历

前序遍历是指在访问根结点、遍历左子树与遍历右子树这 3 者中, 首先访问根结点, 然后遍历左子树, 最后遍历右子树; 并且, 在遍历左右子树时, 仍然先访问根结点, 然后遍历左子树, 最后遍历右子树。

2. 中序遍历

中序遍历是指在访问根结点、遍历左子树与遍历右子树这 3 者中, 首先遍历左子树, 然后访问根结点, 最后遍历右子树; 并且, 在遍历左、右子树时, 仍然先遍历左子树, 然后访问根结点, 最后遍历右子树。

3. 后序遍历

后序遍历是指在访问根结点、遍历左子树与遍历右子树这 3 者中, 首先遍历左子树, 然后遍历右子树, 最后访问根结点, 并且, 在遍历左、右子树时, 仍然先遍历左子树, 然后遍历右子树, 最后访问根结点。

【例题 4】写出图 1-10 所示的二叉树的前序、中序和后序遍历序列。

【例题分析】

从图 1-10 中可以得出:

前序遍历序列: ABCDEF。

中序遍历序列: CBDAEF。

后序遍历序列: CDBFEA。

【例题 5】已知二叉树的中序序列为 ABCDEFG, 后序序列为 BDCAFGE, 则前序序列是什么?

【例题分析】

(1) 从后序中, 我们可以得到根结点是 E, 再看中序的序列: ABCDEFG, 可以发现 ABCD 位于根结点的左边, 而 FG 位于根结点的右边, 于是得到如图 1-11 所示的草图。

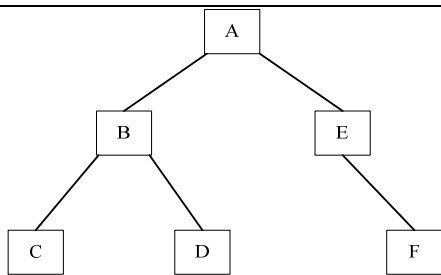


图 1-10 例 4 二叉树

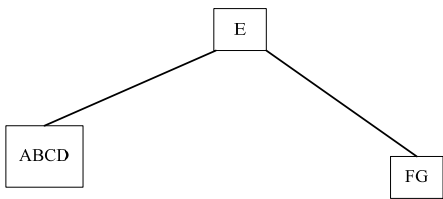


图 1-11 步骤一的草图

(2) 先来看 ABCD 这部分，然后看后序序列，在后序序列中 BDCA 这一部分，可以确定 A 是这部分的根，再看中序序列中的 ABCD，发现 BCD 都在 A 的后面。因此，可以得到如图 1-12 所示的图形。

(3) 再看 BCD 这部分，从后序中看到的顺序是 BDC，所以 C 是这部分的根结点，中序序列是 BCD，可以断定 B 在 C 的左边，D 在 C 的右边。这样，得到图 1-13 所示的图形。

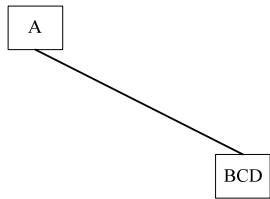


图 1-12 步骤 (2) 的图形

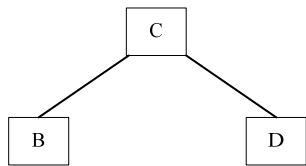


图 1-13 步骤 (3) 的图形

(4) 再看看右边的 FG 这部分，从后序序列 FG 和中序 FG 中，可以推出，G 是这部分的根结点，而 F 位于 G 的左边，如图 1-14 所示。

(5) 根据以上的步骤进行合并成一个二叉树，如图 1-15 所示。

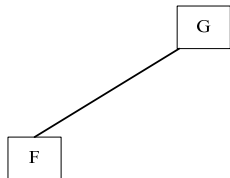


图 1-14 步骤 (4) 的结果

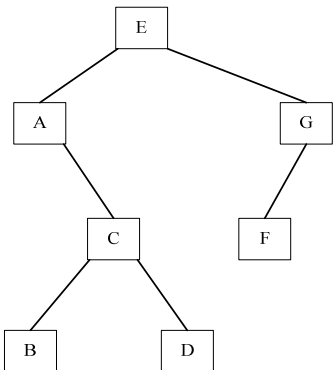


图 1-15 最后的二叉树

(6) 写出前序的序列：EACBDGF。

1.6 查找技术

所谓查找是指在一个给定的数据结构中查找某个指定的元素，在查找的过程中，涉及查找的方法等问题，通常根据不同的数据结构，应采用不同的查找方法。

1.6.1 顺序查找

顺序查找又称顺序搜索。顺序查找一般是指在线性表中查找指定的元素，其基本方法为从线性表的第一个元素开始，依次将线性表中的元素与被查找元素进行比较，若相等则表示找到（即查找成功）；若线性表中所有的元素与被查找元素进行了比较但都不相等，则表示线性表中没有要找的元素（即查找失败）。

在进行顺序查找过程中，如果线性表中的第一个元素就是被查找元素，则只需做一次比较就查找成功，查找效率最高；但如果被查找元素是线性表中的最后一个元素，或者被查找元素根本不在线性表中，则为了查找这个元素需要与线性表中所有的元素进行比较，这是顺序查找的最坏情况。在平均情况下，利用顺序查找法在线性表中查找一个元素，大约要与线性表中一半的元素进行比较。

由此可以看出，对于大的线性表来说，顺序查找的效率是很低的。虽然顺序查找的效率不高，但在下列两种情况下也只能采用顺序查找：

（1）如果线性表为无序线性表（即表中元素的排列是无序的），则不管是顺序存储结构还是链式存储结构，都只能用顺序查找。

（2）即使有序线性表，如果线性表采用链式存储结构，也只能用顺序查找。

1.6.2 二分法查找

二分法查找只适用于顺序存储的有序表。在此所说的有序表是指线性表中的元素按值非递减排列（即从小到大，但允许相邻元素值相等）。

设有序线性表的长度为 n ，被查元素为 x ，则对分查找的方法如下。

将 x 与线性表的中间项进行比较：

- 若中间项的值等于 x ，则说明查到，查找结束。
- 若 x 小于中间项的值，则在线性表的前半部分（即中间项以前的部分）以相同的方法进行查找。
- 若 x 大于中间项的值，则在线性表的后半部分（即中间项以后的部分）以相同的方法进行查找。

这个过程一直进行到查找成功或子表长度为 0（说明线性表中没有这个元素）为止。

1.7 排序技术

在排序技术方面，主要考查插入排序、选择排序、冒泡排序、快速排序和堆排序等方法。

1. 插入排序

每次将一个待排序的数据元素，插入到前面已经排好序的数列中的适当位置，使数列依然有序；直到待排序数据元素全部插入完为止。

【例题 6】使用插入排序对 3、4、2、1、5 按照从小到大的排序。

【例题分析】

第一趟 3 4 2 1 5

第二趟 2 3 4 1 5

第三趟 1 2 3 4 5

第四趟 1 2 3 4 5

2. 选择排序

每一趟从待排序的数据元素中选出最小（或最大）的一个元素，放在已排好序的数列的最后，直到全部待排序的数据元素排完。

【例题 7】使用选择排序对 3、4、2、1、5 按照从小到大的排序。

【例题分析】

第一趟	1	4	2	3	5
第二趟	1	2	4	3	5
第三趟	1	2	3	4	5
第四趟	1	2	3	4	5

3. 冒泡排序

两两比较待排序数据元素的大小，发现两个数据元素的次序相反时即进行交换，直到没有反序的数据元素为止。

设想被排序的数组 $R[1..N]$ 垂直竖立，将每个数据元素看作有重量的气泡，根据轻气泡不能在重气泡之下的原则，从下往上扫描数组 R ，凡扫描到违反本原则的轻气泡，就使其向上“漂浮”，如此反复进行，直至最后任何两个气泡都是轻者在上面，重者在下面为止。

【例题 8】使用冒泡排序对 3、4、2、1、5 按照从小到大的排序。

【例题分析】

第一趟	3	2	1	4	5
第二趟	2	1	3	4	5
第三趟	1	2	3	4	5
第四趟	1	2	3	4	5

4. 快速排序

在当前无序区 $R[1..H]$ 中任取一个数据元素作为比较的“基准”（不妨记为 X ），用此基准将当前无序区划分为左右两个较小的无序区： $R[1..I-1]$ 和 $R[I+1..H]$ ，且左边的无序子区中数据元素均小于等于基准元素，右边的无序子区中数据元素均大于等于基准元素，而基准 X 则位于最终排序的位置上，即 $R[1..I-1] \leq X.Key \leq R[I+1..H]$ ($1 \leq I \leq H$)，当 $R[1..I-1]$ 和 $R[I+1..H]$ 均非空时，分别对它们进行上述的划分过程，直至所有无序子区中的数据元素均已排序为止。

【例题 9】使用快速排序对 7、8、3、4、9、1 进行从小到大的排序，仅写出第一趟的结果，以第一个元素为主。

【例题分析】

第一趟	7	8	3	4	9	1
第一趟	1	8	3	4	9	7
第一趟	1	7	3	4	9	8
第一趟	1	7	3	4	9	8
第一趟	1	4	3	7	9	8
第一趟	1	4	3	7	9	8

一般来说，在考试的时候，只问第一趟的结果，也就是以第一个元素为主排列的结果。

5. 堆排序

堆排序是一树形选择排序，在排序过程中，将 $R[1..N]$ 看成是一颗完全二叉树的顺序存储结构，利用完全二叉树中双亲结点和孩子结点之间的内在关系来选择最小的元素。

N 个元素的序列 $K_1, K_2, K_3, \dots, K_n$ 称为堆，当且仅当该序列满足特性：

$$K_i \leq K_{2i}, K_i \leq K_{2i+1} \quad (1 \leq i \leq [N/2])$$

堆实质上是满足如下性质的完全二叉树：树中任一非叶子结点的关键字均大于等于其孩子结点的关键字。

堆排序正是利用小根堆（或大根堆）来选取当前无序区中关键字最小（或最大）的记录实现排序的。我们不妨利用大根堆来排序。每一趟排序的基本操作是：将当前无序区调整为一个大根堆，选取关键字最大的堆顶记录，将它和无序区中的最后一个记录交换。这样，正好和直接选择排序相反，有序区是在原记录区的尾部形成并逐步向前扩大到整个记录区。

【例题 10】使用堆排序对 7、8、3、4、9、1 进行从小到大的排序，仅写出第一趟排序的结果。

【例题分析】

7 8 3 4 9 1
下标: 1 2 3 4 5 6
因为有 6 个数，首先取 $6/2=3$ ，然后看看 $k_3 \leq k_6$ 是否成立（此处没有 k_7 ）因为 k_3 的值是 3，而 k_6 的值是 1，显然不满足条件，要将 k_3 和 k_6 进行交换，就变成如下的形式：

7 8 1 4 9 3
下标: 1 2 3 4 5 6
然后再看 $k_2 \leq k_4$ 和 $k_2 \leq k_5$ 是否成立。因为 k_2 的值是 8， k_4 的值是 4，而 k_5 的值是 9，所以，将 k_2 的值和 k_4 的值交换，得到如下序列：

7 4 1 8 9 3
下标: 1 2 3 4 5 6
在看 $k_1 \leq k_2$ 和 $k_1 \leq k_3$ ，发现不满足，此时的 k_1 要和 k_2 与 k_3 中的最小的一个进行交换，所以得到序列如下：

1 4 7 8 9 3
下标: 1 2 3 4 5 6
堆建好了吗？检查每个位置是否满足上面的条件？答案是“没有”，因为此时 $k_3 \leq k_6$ 又不成立了，所以要交换，得到如下的序列：

1 4 3 8 9 7
下标: 1 2 3 4 5 6
以上是第一趟排列的结果，如果要进行第二次堆排序，就从剩下的 4、3、8、9、7 开始。

6. 各种排序方法的比较

各种排序方法的比较如表 1-2 所示。

表 1-2 各种排序算法的比较			
方法	平均时间	最坏时间	辅助空间
冒泡、选择、插入	$O(n^2)$	$O(n^2)$	$O(1)$
快速排序	$O(n\log_2 n)$	$O(n^2)$	$O(n\log_2 n)$
堆排序	$O(n\log_2 n)$	$O(n\log_2 n)$	$O(1)$

1.8 本章习题

1.8.1 选择题

1. 下列叙述中错误的是（ ）。
- A. 一种数据的逻辑结构可以有多种存储结构
- B. 数据的存储结构与数据的处理效率无关
- C. 数据的存储结构与数据的处理效率密切相关
- D. 数据的存储结构在计算机中所占的空间不一定是连续的
2. 下列对队列的描述中正确的是（ ）。
- A. 队列属于非线性表
- B. 队列按“先进后出”原则组织数据
- C. 队列在队尾删除数据
- D. 队列按“先进先出”原则组织数据
3. 对长度为 N 的线性表排序，在最坏情况下，比较次数不是 $N(N-1)/2$ 的排序方法是（ ）。
- A. 快速排序
- B. 冒泡排序
- C. 直接插入排序
- D. 堆排序
4. 下列关于栈的描述中错误的是（ ）。

- A. 栈是先进先出的线性表 B. 栈只能顺序存储
C. 栈具有记忆作用 D. 对栈插入删除操作中, 不需要改变栈底指针
5. 某二叉树中有 N 个度为 2 的结点, 则该二叉树中的叶子结点数为 ()。

A. $N+1$ B. $N-1$ C. $2N$ D. $N/2$

6. 对下列二叉树进行前序遍历的结果为 ()。

A. DYBEAFCZX B. YDEBFZXCA
C. ABDYECFXZ D. ABCDEFXYZ

7. 对长度为 n 的线性表进行顺序查找, 在最坏情况下需要比较的次数为 ()。

A. 125 B. $n/2$ C. n D. $n+1$

8. 下列描述中正确的是 ()。

A. 数据的逻辑结构与存储结构必定是一一对应的
B. 由于计算机存储空间是向量式的存储结构, 因此, 数据的存储结构一定是线性结构

C. 利用数组只能处理线性结构

D. 以上 3 种说法都不对

9. 算法分析的目的是 ()。

A. 找出数据结构的合理性 B. 找出算法中输入和输出之间的关系

C. 分析算法的易懂性和可靠性 D. 分析算法的效率以求改进

10. 按照“先进先出”原则组织数据的结构是 ()。

A. 队列 B. 栈 C. 双向链表 D. 二叉树

11. 栈和队列的共同点是 ()。

A. 都是先进先出 B. 都是先进后出

C. 只允许在端点外插入和删除元素 D. 没有共同特点

12. 当线性表采用链式存储时, 结构的存储地址 ()。

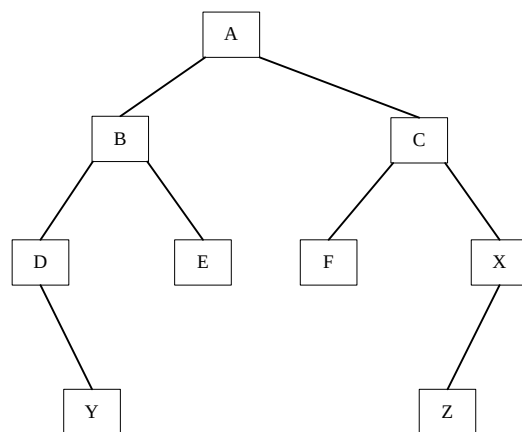
A. 必须是不连续的 B. 连续与否均可
C. 必须是连续的 D. 和头结点的存储地址相连续

13. 在下列数据结构中, 能用二分法进行查找的是 ()。

A. 顺序存储的有序线性表 B. 循环链表
C. 二叉链表 D. 链式存储的有序线性表

14. 线性表进行二分法检索, 其前提条件是 ()。

A. 线性表以顺序方式存储, 并按关键码值排好序
B. 线性表以顺序方式存储, 并按关键码的检索频率排好序
C. 线性表以链式存储, 并按关键码值排好序
D. 线性表以链式存储, 并按关键码的检索频率排好序



1.8.2 填空题

- 在算法的 4 个特征中, 算法必须能在执行有限个步骤之后终止指的是算法的_____特征。
- 在有序列表 (3, 6, 8, 10, 12, 15, 16, 18, 21, 25, 30) 中, 用二分法查找关键值 12 所需的关键码比较次数为_____。
- 排序是计算机程序设计中的一种重要操作, 常见的排序方法有插入排序、_____和选择排序。
- 栈中允许进行插入和删除的一端叫做_____。

第2章

程序设计结构

本章主要介绍了程序设计的方法和风格，以及结构化程序设计的内容，重点描述了面向对象程序设计的风格，结合计算机等级二级 C 语言考试要求，具体如表 2-1 所示。

表 2-1 考试要求

考试知识点	重要性
程序设计方法与风格	★
结构化程序设计	★
面向对象的程序设计	★★

2.1 程序设计方法与风格

程序设计风格指一个人在编制程序时所表现出来的特点、习惯、思路等。

在程序设计中要使程序结构合理、清晰，形成良好的编程习惯，对程序的要求不仅是可以在机器上执行，给出正确的结果，而且要便于程序的调试和维护，这就要求编写的程序不仅自己看得懂，而且也要让别人能看懂。

随着计算机技术的发展，软件的规模增大了，软件的复杂性也增强了。为了提高程序的可阅读性，要建立良好的编程风格。

一般来讲，程序设计风格是指在编写程序时所表现出的特点、习惯和思路。程序是由人来编写的，为了测试和维护程序，往往还要阅读和跟踪程序，因此程序设计的风格总体而言应该强调简单和清晰，程序必须是可以理解的。

1. 数据说明的方法

- (1) 数据说明顺序应规范，使数据的属性更易于查找，从而有利于测试、纠错与维护。
- (2) 当一个语句说明多个变量时，各变量名按字母排序。
- (3) 对于复杂的数据结构，要加注释，说明在程序实现时的特点。

2. 输入和输出

- 无论是批处理的输入和输出方式，还是交互式的输入和输出方式，在设计和编程时都应该考虑如下原则：
- (1) 输入操作步骤和输入格式尽量简单。
 - (2) 应检查输入数据的合法性、有效性，报告必要的输入状态信息及错误信息。
 - (3) 在输入一批数据时，使用数据或文件结束标志，而不要用计数来控制。
 - (4) 在交互式输入时，提供可用的选择和边界值。

- (5) 当程序设计语言有严格的格式要求时，应保持输入格式的一致性。
- (6) 输出数据表格化、图形化。
- (7) 应允许默认值。
- (8) 当程序设计语言对输入格式有严格要求时，应保持输入格式与输入语句的一致性；给所有的输入/出加注释，并设计输出报表格式。

2.2 结构化程序设计

常用的结构化程序设计有如下 3 种形式：

- 1. 顺序结构**
顺序结构是简单的程序设计，它是最基本、最常用的结构，所谓顺序执行，就是按照程序语句行的自然顺序，一条语句一条语句地执行程序，如图 2-1 所示。
- 2. 选择结构**
选择结构又称为分支结构，它包括简单选择和多分支选择结构，这种结构可以根据设定的条件，判断应该选择哪一条分支来执行相应的语句序列，如图 2-2 所示。
- 3. 循环结构**
循环结构根据给定的条件，判断是否需要重复执行某一相同的或类似的程序段，利用重复结构可简化大量的程序行。分为两类：一是先判断后执行，一是先执行后判断，如图 2-3 所示。

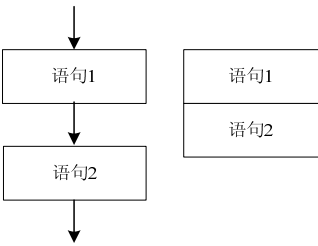


图 2-1 顺序结构程序

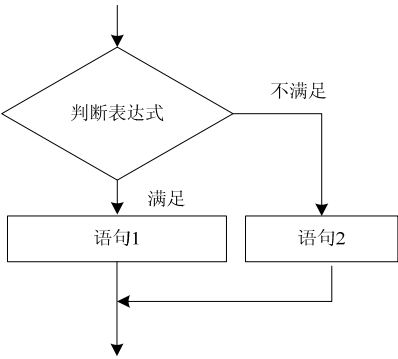


图 2-2 选择结构程序

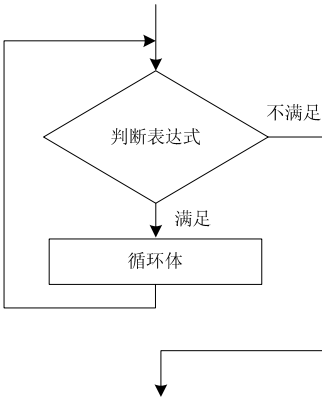


图 2-3 循环结构设计

【例题 1】 符合结构化原则的 3 种基本控制结构的是：选择结构、循环结构和_____。（2009 年 3 月）

【例题分析】
结构化的 3 种基本控制结构为选择结构、循环结构和顺序结构。

2.3 面向对象的程序设计

对象是面向对象方法中最基本的概念。对象可以用来表示客观世界中的任何实体，也就是说，应用领域中有意义的、与所要解决的问题有关系的任何事物都可以作为对象，它既可以是具体的物理实体的抽象，也可以是人为的概念，或者

是任何有明确边界的、有意义的东西。总之，对象是对问题域中某个实体的抽象，设立某个对象就反映软件系统保存有关它的信息并具有与它进行交互的能力。

面向对象的程序设计方法中涉及的对象是系统中用来描述客观事物的一个实体，是构成系统的一个基本单位，它由一组表示其静态特征的属性和它可执行的一组操作组成。

对象可以做的操作表示它的动态行为，在面向对象分析和面向对象设计中，通常把对象的操作也称为方法或服务。

属性即对象所包含的信息，它在设计对象时确定，一般只能通过对象的操作来改变。

操作描述了对象执行的功能，若通过消息传递，还可以为其他对象使用。操作的过程对外是封闭的，即用户只能看到这一操作实施后的结果。这相当于事先已经设计好的各种过程，只需要调用就可以了，用户不必去关心这一过程是如何编写的。事实上，这个过程已经封装在对象中，用户也看不到。对象的这一特性即对象的封装性。

2.3.1 面向对象特点

面向对象具有封装性、继承性和多态性。

1. 封装性

封装是一种信息隐蔽技术，它体现于类的说明，是对象的重要特性。封装性使数据和加工该数据的方法（函数）封装为一个整体，以实现独立性很强的模块，使得用户只能见到对象的外特性（对象能接受哪些消息，具有哪些处理能力），而对象的内特性（保存内部状态的私有数据和实现加工能力的算法）对用户是隐蔽的。封装的目的在于把对象的设计者和对象的使用者分开，使用者不必知晓行为实现的细节，只须用设计者提供的消息来访问该对象。

2. 继承性

继承性是子类自动共享父类之间数据和方法的机制。它由类的派生功能体现。一个类直接继承其他类的全部描述，同时可修改和扩充。继承具有传递性。继承分为单继承（一个子类只有一个父类）和多重继承（一个子类有多个父类）。类的对象是各自封闭的，如果没继承性机制，则类的对象中的数据、方法就会出现大量重复。继承不仅支持系统的可重用性，而且还促进系统的可扩充性。

3. 多态性

对象根据所接收的消息而做出动作。同一消息为不同的对象接受时可产生完全不同的行动，这种现象称为多态性。利用多态性用户可发送一个通用的信息，而将所有的实现细节都留给接受消息的对象自行决定，如是同一消息即可调用不同的方法。例如，Print 消息被发送给一图或表时的调用的打印方法与将同样的 Print 消息发送给一个正文文件而调用的打印方法会完全不同。多态性的实现受到继承性的支持，利用类继承的层次关系，把具有通用功能的协议存放在类层次中尽可能高的地方，而将实现这一功能的不同方法置于较低层次，这样，在这些低层次上生成的对象就能给通用消息以不同的响应。在面向对象的编程语言中可通过在派生类中重定义基类函数（定义为重载函数或虚函数）来实现多态性。

2.3.2 类和实例

类是对象的模板。即类是对一组有相同数据和相同操作的对象的定义，一个类所包含的方法和数据描述一组对象的共同属性和行为，将属性、操作相似的对象归为类，也就是说，类是具有共同属性、共同方法的对象的集合。所以，类是对象的抽象，它描述了属于该对象类型的所有对象的性质，而一个对象则是其对应类的一个实例。

要注意的是，当使用“对象”这个术语时，既可以指一个具体的对象，也可以泛指一般的对象，但是，当使用“实例”这个术语时，必然是指一个具体的对象。

由类的定义可知，类是关于对象性质的描述，它同对象一样，包括一组数据属性和在数据上的一组合法操作。

在 UML 语言中，类的表示如图 2-4 所示。

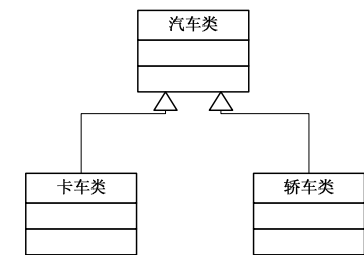


图 2-4 类图

2.3.3 消息

面向对象的世界是通过对象与对象间彼此的相互合作来推动的，对象间的这种相互合作需要一个机制协助进行，这样的机制称为“消息”。消息是一个实例与另一个实例之间传递的信息，它请示对象执行某一处理或回答某一要求的信息，它统一了数据流的控制流。消息的使用类似于函数调用，消息中指定了某一个实例、一个操作名和一个参数表（可空）。接收消息的实例执行消息中指定的操作，并将形式参数与参数表中相应的值结合起来。在消息传递过程中，由发送消息的对象（发送对象）触发操作产生输出结果，作为消息传送至接收消息的对象（接收对象），引发接收消息的对象一系列的操作。所传送的消息实质上是接收对象所具有的操作/方法名称，有时还包括相应参数。

通常，一个消息由下述3部分组成：

- (1) 接收消息的对象名称。
- (2) 消息标识符（也称为消息名）。
- (3) 零个或多个参数。

2.4 本章习题

2.4.1 选择题

1. 下列叙述中正确的是（ ）。
 - A. 在面向对象的程序设计中，各个对象之间具有密切的关系
 - B. 在面向对象的程序设计中，各个对象都是公用的
 - C. 在面向对象的程序设计中，各个对象之间相对独立，相互依赖性小
 - D. 上述3种说法都不对
2. 在面向对象方法中，实现信息隐蔽是依靠（ ）。
 - A. 对象的继承
 - B. 对象的多态
 - C. 对象的封装
 - D. 对象的分类
3. 源程序的文档化不包括（ ）。
 - A. 符号名的命名要有实际意义
 - B. 正确的文档形式
 - C. 良好的视觉组织上
 - D. 正确的程序注释
4. 在面向对象方法中，（ ）描述的是具有相似属性与操作的一组对象。
 - A. 属性
 - B. 事件
 - C. 方法
 - D. 类
5. 下列叙述中正确的是（ ）。
 - A. 程序设计时候不需要讲究风格
 - B. 程序中的注释是可有可无的
 - C. 程序只要求机器读懂就可以了，不需要去关心维护的问题
 - D. 以上说法都不对
6. 结构化程序设计的3种基本结构是（ ）。
 - A. 过程、子程序和分程序
 - B. 顺序、选择和重复
 - C. 递归、堆栈和队列
 - D. 调用、返回和转移

2.4.2 填空题

1. 从程序设计的方法和技术的发展角度来说，程序设计主要经历了结构化设计和_____的程序设计阶段。
2. 在面向对象程序设计中，从外面看只能看到对象有外部特征，而不知道也无须知道具体的数据及实现操作的算法，这称为对象的_____。
3. 在面向对象的程序设计中，类描述的是具有相似性质的一组_____。
4. 在程序设计阶段应该采取_____和逐步求精的方法，把一个模块的功能逐步分解，细化为一系列具体的步骤，继而用某种程序设计语言写成程序。

第3章

软件工程基础

本章主要介绍了软件工程的观念和结构化程序分析、设计，重点介绍了软件测试的内容。结合计算机等级考试要求，具体如表 3-1 所示。

表 3-1 考试要求

考试知识点	重要性
软件工程概念	★
结构化程序分析	★
结构化程序设计	★
软件测试	★★★
程序调试	★

3.1 软件工程基本概念

软件工程是计算机软件学科中一个重要的内容，它对软件产业的形成和发展起着决定性的推动作用，在人类进入信息化社会时成为新信息产业的支柱。

3.1.1 软件的含义

计算机软件（Software）是计算机系统中与硬件相互依存的另一部分，是包括程序、数据及相关文档的完整集合。其中，程序是软件开发人员根据用户需求开发的、用程序设计语言描述的、适合计算机执行的指令（语句）序列。

软件根据应用目标的不同，是多种多样的。软件按功能可以分为应用软件、系统软件、支撑软件（或工具软件）。应用软件是为解决特定领域的应用而开发的软件。系统软件是计算机管理自身资源，提高计算机使用效率并为计算机用户提供各种服务的软件。支撑软件是介于系统软件和应用软件之间，协助用户开发软件的工具性软件，包括辅助、支持开发和维护应用软件的软件。

软件的特点如下：

- （1）软件是一种逻辑实体，是人类智力成果的表现形式。软件的包装、载体、印刷的文档本身都不是软件。
- （2）软件的开发和制造是一个统一的过程。软件开发本身是一个过程，不是突然得到的结果；一旦软件开发完成，剩下的就只是复制分发的过程，对软件本身不再有生产制造活动。
- （3）软件开发是一项经济活动，在一定的成本和时间限制下，满足用户的需求是软件开发的目。那种仅仅为了个人兴趣进行的编程，不是工程意义上的软件开发。
- （4）软件不会磨损，但可能存在错误，需要进行维护。

3.1.2 软件工程

1. 软件工程概念

软件工程是一类工程。工程是将理论和知识应用于实践的科学。就软件工程而言，它借鉴了传统工程的原则和方法，以求高效地开发高质量软件。其中应用了计算机科学、数学和管理科学。计算机科学和数学用于构造模型与算法，工程科学用于制定规范、设计范型、评估成本及确定权衡，管理科学用于计划、资源、质量和成本的管理。

软件工程这一概念，主要是针对 20 世纪 60 年代“软件危机”而提出的。自这一概念提出以来，围绕软件项目开展了有关开发模型、方法及支持工具的研究。具体地说，在软件开发和维护过程中，软件危机主要表现在：

- (1) 软件需求增长得不到满足。
- (2) 软件生产高成本，价格昂贵。
- (3) 软件生产进度无法控制。
- (4) 软件需求定义不准确，易偏离用户需求。
- (5) 软件质量不易保证。
- (6) 软件可维护性差。

分析带来软件危机的原因，宏观方面是由于软件日益深入社会生活的各个层面，对软件需求的增长速度大大超过了技术进步所能带来的软件生产率的提高。而就每一项具体的工程任务来看，许多困难来源于软件工程所面临的任务和其他工程之间的差异，以及软件和其他工业产品的不同。

在软件开发和维护过程中，之所以存在这些严重的问题，一方面与软件本身的特点有关。例如，在软件运行前，软件开发过程的进展难衡量，质量难以评价，因此管理和控制软件开发过程相当困难；在软件运行过程中，软件维护意味着改正或修改原来的设计；另外，软件的显著特点是规模庞大、复杂度超线性增长，在开发大型软件时，要保证高质量是极其复杂困难的，不仅涉及技术问题（如分析方法、设计方法、版本控制），更重要的是必须有严格而科学的管理。另一方面与软件开发和维护方法不正确有关，这是主要原因。

为了消除软件危机，通过认真研究解决软件危机的方法，认识到软件工程是使计算机软件走向工程科学的途径，逐步形成了软件工程的概念，开辟了工程学的新兴领域——软件工程学。软件工程就是试图用工程、科学和数学的原理与方法研制、维护计算机软件的有关技术及管理方法。

2. 软件生命周期

通常，将软件产品从提出、实现、使用维护到停止使用退役的过程称为软件生命周期。也就是说，软件产品从其概念开始，到该软件产品不能使用为止的整个时期都属于软件生命周期。一般可以包括可行性研究与需求分析、设计、实现、测试、交付使用以及维护的活动。

还可以将软件生命周期分为软件定义、软件开发、软件运行维护三个阶段。

(1) 可行性研究与计划制定。确定待开发的软件系统的开发目标和总体要求，给出它的功能、性能、可靠性以及接口等方面的方案。制定完成开发任务的实施计划。

(2) 需求分析。对待开发软件提出的需求进行分析并给出详细定义。编写软件规格说明书以及初步的用户手册，提交评审。

(3) 软件设计。系统设计人员和程序设计人员应该在理解软件需求的基础上，给出软件的结构、模块的划分、功能的分配以及处理流程。通常，把设计阶段可分解为概要设计和详细设计阶段。编写概要设计说明书、详细说明书和测试计划初稿，提交评审。

(4) 软件实现。把软件设计转换成计算机可以接受的程序代码。完成源程序的编码，编写用户手册、操作手册等面向用户的文档，编写单元测试计划。

(5) 软件测试。在设计测试用例的基础上，检验软件的各个组成部分。编写测试分析报告

(6) 运行和维护。将已交付的软件投入运行。并在运行使用中不断的维护，根据新提出的需求进行必要而且可能的扩充和删改。

3.2 结构化分析

需求分析是整个软件开发工作最重要的一步，通过软件需求分析，才能把软件功能和性能的总体概述为具体的需求规格说明，从而形成各个软件开发阶段的基础。

需求分析要做的工作就是深入描述软件的功能和性能，确定软件设计的限制和软件同其他系统元素的接口细节，其步骤如下：

- (1) 获得当前系统的物理模型。
- (2) 抽象出当前系统的逻辑模型。
- (3) 建立目标系统的逻辑模型。
- (4) 对所得到的模型进行进一步的补充。

3.2.1 结构化分析方法

结构化分析方法是需求分析中使用最多的一种方法，它分析的对象是数据流，它采用自顶向下，逐层分解，建立系统的处理流程，强调开发方法的结构合理性，以及所开发软件的结构合理性。结构是指系统内各个组成要素之间相互联系、相互作用的框架。结构化开发方法提出了一组提高软件结构合理性的准则，如分解与抽象、模块独立性、信息隐蔽等。

结构化分析方法通过数据流程图（Data Flow Diagram, DFD）、数据字典（Data Dictionary, DD）、结构化英语、判定表和判定树等工具。

结构化分析的步骤如下：

- (1) 通过对用户的调查，以软件的需求为线索，获得当前系统的具体模型。
- (2) 去掉具体模型中非本质因素，抽象出当前系统的逻辑模型。
- (3) 根据计算机的特点分析当前系统与目标系统的差别，建立目标系统的逻辑模型。
- (4) 完善目标系统并补充细节，写出目标系统的软件需求规格说明。
- (5) 评审直到确认完全符合用户对软件的需求。

结构化分析方法的常用工具主要有数据流程图、数据字典、判定树和判定表。

1. 数据流程图

数据流程图是描述数据处理过程的工具，是需求理解的逻辑模型的图形表示，它直接支持系统的功能建模。

数据流程图从数据传递和加工的角度，来描述数据流从输入到输出的移动变换过程。数据流图中的主要图形元素与说明如图 3-1 所示。

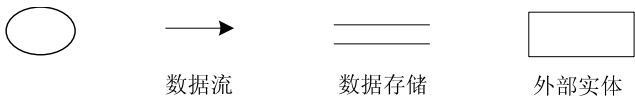


图 3-1 数据流程图各要素

2. 数据字典

数据字典是结构化分析方法的核心。数据字典是对所有与系统相关的数据元素的一个有组织的列表，以及精确的、严格的定义，使得用户和系统分析员对输入、输出、存储成分和中间计算结果有共同的理解。数据字典把不同的需求文档和分析模型紧密地结合在一起，与各模型的图形表示配合，能清楚地表达数据处理的要求。

概括地说，数据字典的作用是对 DFD 中出现的被命名的图形元素的确切解释。通常数据字典包含的信息有名称、别名、何处使用/如何使用、内容描述、补充信息等。例如，对加工的描述应包括加工名、反映该加工层次的加工编号、加工逻辑及功能简述、输入/输出数据流等。

在数据字典的编制过程中，常使用定义式方式描述数据结构。如表 3-2 所示为常用定义式符号。

表 3-2 数据字典定义式方式中出现的符号

符号	含义	符号	含义
=	表示“等于”、“定义为”、“由什么构成”	(...)	表示“可选”，即括号中的项可以没有
[... ...]	表示“或”，即选择括号中用“ ”号分隔的各项中的某一项	**	表示“注释”
+	表示“与”、“和”	..	连接符
n{}m	表示“重复”，即括号中的项要重复若干次，n、m 是重复次数的上、下限		

3. 判定树

在使用判定树进行描述时，应先从问题定义的文字描述中分清哪些是判定的条件，哪些是判定的结论，根据描述材料中的连接词找出判定条件之间的从属关系、并列关系、选择关系，根据它们构造判定树。

例如，在某些货物托运管理系统中，对发货情况的处理要依赖检查发货单，检查发货单受货物托运金额、欠款等条件的约束，可以使用类似分段函数的形式来描述这些约束和处理。对这些约束条件的描述，如果使用自然语言，表达容易出现不准确和不清晰。

4. 判定表

判定表与判定树相似，当数据流图中的加工要依赖于多个逻辑条件的取值时，即完成该加工的一组动作是由于某一组条件取值的组合而引发的，使用判定表描述比较便宜。

判定表或判定树是以图形形式描述数据流图的加工逻辑，它结构简单，易读易懂。尤其遇到组合条件的判定，利用判定表或判定树可以使问题的描述清晰，而且便于直接映射到程序代码。在表达一个加工逻辑时，判定树、判定表都是好的描述工具，根据需要还可以交叉使用。

表 3-3 “检查发货单”判定表

		1	2	3	4
条件	发货单金额	>\$500	>\$500	≤\$500	≤\$500
	赊欠情况	>60 天	≤60 天	>60 天	≤60 天
操作	不发出批准书	√			
	发出批准书		√	√	√
	发出发货单		√	√	√
	发出赊欠报告			√	

3.2.2 软件需求规格说明书

软件需求规格说明书（SRS,Software Requirement Specification）是需求分析阶段的最终成果，是软件开发中的重要文档之一。

1. 软件需求规格说明书的作用

- （1）便于用户、开发人员进行理解和交流。
- （2）反映出用户问题的结构，可以作为软件开发工作的基础和依据。
- （3）作为确认测试和验收的依据。

2. 软件需求规格说明书的内容

软件需求规格说明书是作为需求分析的一部分而指定的可交付文档。该说明把在软件计划中确定的软件范围加以展开，制定出完整的信息描述、详细的功能说明、恰当的检验标准，以及其他与要求有关的数据。其中：

概述是从系统的角度来描述软件的目标和任务的。

数据描述是对软件系统必须解决的问题作出的详细说明。

功能描述视为解决用户问题所需要的每一项功能的过程细节。对每一项功能要给出处理说明和在设计时需要考虑的限制条件。

性能描述是说明系统应达到的性能和应该满足的限制条件、检测的方法和标准、预期的软件响应和可能要考虑的特殊问题。

参考文献目录描述应包括与该软件有关的全部参考文献，其中包括前期的其他文档、技术参考资料、产品目录手册及标准等。

附录部分包括一些补充资料，如列表数据、算法的详细说明、框图、图表和其他材料。

3. 软件需求规格说明书的特点

软件需求规格说明书是确保软件质量的有力措施，衡量软件需求规格说明书质量好坏的标准、标准的优先级及标准的内涵。其特点如下：

- (1) 正确性。体现待开发系统的真实要求。
- (2) 无歧义性。对每一个需求只有一种解释，其陈述具有唯一性。
- (3) 完整性。包括全部有意义的需求，如功能的、性能的、设计的、约束的，属性或外部接口等方面的需求。
- (4) 可验证性。描述的每一个需求都是可以验证的，即存在有限代价的有效过程验证确认。
- (5) 一致性。各个需求的描述不矛盾。
- (6) 可理解性。需求说明书必须简明易懂，尽量少包含计算机的概念和术语，以使用户和软件人员都能接受它。
- (7) 可修改性。SRS 的结构风格在需求有必要改变时是易于实现的。
- (8) 可追踪性。每一个需求的来源，流向是清晰的，当产生和改变文件编制时，可以方便地引证每一个需求。

软件需求规格说明书是一份在软件生命周期中至关重要的文件，它在开发早期就为尚未诞生的软件系统建立了一个可见的逻辑模型，它可以保证开发工作的顺利进行，因而应及时地建立并保证它的质量。

作为设计的基础和验收的依据，软件需求规格说明书应该是精确而无二义性的，需求说明书越精确，则以后出现错误、混淆、反复的可能性越小。用户能看懂需求说明书，并且发现和指出其中的错误是保证软件系统质量的关键，因而需求说明书必须简明易懂，尽量少包含计算机的概念和术语，以使用户和软件人员双方都能接受它。

3.3 结构化设计方法

在软件的需求分析阶段已经完全弄明白了软件的各种需求，较好地解决了要让所开发的软件“做什么”的问题，并已经在软件需求说明书中详尽和充分地阐述了这些需求，下一步就是要着手解决“如何做”的问题，而软件设计就是把软件的需求分析变成软件表示的过程。

3.3.1 软件设计的基本内容

将软件的结构设计按自顶向下方式，对各个层次的过程细节和数据细节逐层细化，直到用程序设计的语言的语句能够实现，从而最后确定整个的体系结构。

模块是指把一个待开发的软件分解成若干小的、简单的部分，在解决一个复杂问题时自顶向下逐层把软件系统划分成若干模块的过程。模块独立性是指，每个模块只完成系统要求独立的子功能，并且与其他模块的联系最少且接口简单。

模块的独立程度是评价设计好坏的重要度量标准。衡量软件的模块独立性使用耦合性和内聚性两个定性的度量标准。

1. 内聚性

内聚性是一个模块内部各个元素间彼此结合的紧密程度的度量。内聚是从功能角度来度量模块内的联系。

内聚具有如下的种类，它们之间的内聚性由弱到强排列为：

- 偶然内聚：指一个模块内的各处理元素之间没有任何联系。
- 逻辑内聚：指模块内执行几个逻辑上相关的功能，通过参数确定该模块完成哪一个功能。
- 时间内聚：把需要同时或顺序执行的动作组合在一起形成的模块为时间内聚模块。比如初始化模块，它按顺序为变量置初值。
- 过程内聚：如果一个模块内的处理元素是相关的，而且必须以待定次序执行则称为过程内聚。
- 通信内聚：指模块内所有处理功能都通过使用公用数据而发生关系。这种内聚也具有过程内聚的特点。
- 顺序内聚：指一个模块中各个处理元素和同一个功能密切相关，而且这些处理必须顺序执行，通常前一个处理元素的输出就是下一个处理元素的输入。
- 功能内聚：指模块内所有元素共同完成一个功能，缺一不可，模块已不可再分。这是最强的内聚。

内聚性是指信息隐蔽和局部化概念的自然扩展。一个模块的内聚性越强则该模块的模块独立性越强。软件结构设计的原则，要求每一个模块的内部都具有很强的内聚性，它的各个组成部分彼此都密切相关。

2. 耦合性

耦合性是模块间互相连接的紧密程度的度量。

耦合性取决于各个模块之间接口的复杂度、调用方式，以及哪些信息通过接口。耦合可以分为下列几种，它们之间的耦合度由高到低排列为：

- 内容耦合：如一个模块直接访问另一个模块的内容，则这两个模块称为内容耦合。
- 公共耦合：若一组模块都访问同一全局数据结构，则它们之间的耦合称为公共耦合。
- 外部耦合：如果一组模块都访问同一全局简单变量（而不是同一全局数据结构），且不通过参数表传递该全局变量的信息，则称为外部耦合。
- 控制耦合：若一模块明显地把开关量、名字等信息送入另一模块，控制另一模块的功能，则为控制耦合。
- 标记耦合：若两个以上的模块都需要其余某一数据结构子结构时，不使用其余全局变量的方式而是用记录传递的方式，即两模块间通过数据结构交换信息，这样的耦合称为标记耦合。
- 数据耦合：若一个模块访问另一个模块，被访问模块的输入和输出都是数据项参数，即两模块间通过数据参数交换信息，则这两个模块为数据耦合。
- 非直接耦合：若两个模块没有直接关系，他们之间的联系完全是通过主模块的控制和调用来实现的，则称这两个模块为非直接耦合。非直接耦合独立性最强。

耦合性与内聚性是模块独立性的两个定性标准，耦合与内聚是相互关联的。在程序结构中，各模块的内聚性越强，则耦合性越弱。一般较优秀的软件设计，应尽量做到高内聚，低耦合，即减弱模块之间的耦合性和提高模块内的内聚性，有利于提高模块的独立性。

【例题 1】软件设计中划分模块的一个准则是_____。（2009 年 9 月）

A. 低内聚低耦合	B. 高内聚低耦合
C. 低内聚高耦合	D. 高内聚高耦合

【例题分析】

本题考查软件设计中划分模块的原则，很显然“高内聚低耦合”是划分的标准，选 B。

3.3.2 结构化设计

结构化设计方法给出一组帮助设计人员在模块层次上区分设计质量的原理与技术。它通常与结构化分析方法衔接起来使用，以数据流图为基础得到软件的模块结构。结构化设计就是采用最佳的可能方法设计系统的各个组成部分，以及各成分之间的内部联系的技术。也就是说，结构化设计是这样一个过程，它决定用哪些方法把哪些部分联系起来，才能解决好某个具体有清楚定义的问题。

结构化设计的步骤如下：

- (1) 评审和细化数据流图。

- (2) 确定数据流图的类型。
- (3) 把数据流图映射到软件模块结构，设计出模块结构的上层。
- (4) 基于数据流图逐步分解高层模块，设计中、下层模块。
- (5) 对模块结构进行优化，得到更为合理的软件结构。
- (6) 描述模块接口。

【例题 2】下列选项中不属于结构化程序设计原则的是_____。(2009 年 9 月)

- A. 可封装
- B. 自顶向下
- C. 模块化
- D. 逐步求精

【例题分析】

结构化程序设计原则包括可封装性、自顶向下分解、逐步求精的原则，所以答案选 C。

3.3.3 概要设计

概要设计的主要任务是把需求分析得到的 DFD 转换为软件结构和数据结构。设计软件结构的具体任务是：将一个复杂系统按功能进行模块划分、建立模块的层次结构及调用关系、确定模块间的接口及人机界面等。数据结构设计包括数据特征的描述、确定数据的结构特性及数据库的设计。在需求分析阶段，主要是分析信息在系统中加工和流动的情况。面向数据流的设计方法定义了一些不同的映射方法，利用这些映射方法可以把数据流图变换成结构图表示的软件结构。首先需要了解数据流图表示的数据处理的类型，然后针对不同类型分别进行分析处理。

典型的数据流类型有两种：变换型和事务型。

1. 面向变换型的数据流

变换型是指信息沿输入通路进入系统，同时由外部形式变换成内部形式，进入系统的信息通过变换中心，经加工处理以后再沿输出通路变换成外部形式离开软件系统。变换数据处理问题的工作过程大致分为 3 步，即取得数据、变换数据和输出数据，如图 3-2 所示。

2. 面向事务型的数据流

在很多软件应用中，存在某种作业数据流，它可以引发一个或多个处理，这些处理能够完成该作业要求的功能，这种数据流就叫做事务。事务型数据流的特点是接受一项事务，根据事务处理的特点和性质，选择分派一个适当的处理单元（事务处理中心），然后给出结果。在一个事务型数据流中，事务中心接收数据，分析每个事务以确定它的类型，根据事务类型选取一条活动通路，如图 3-3 所示。

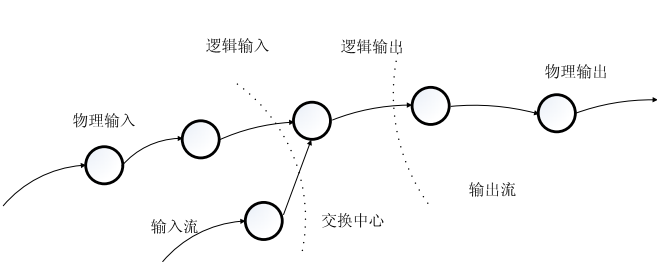


图 3-2 变换型数据流

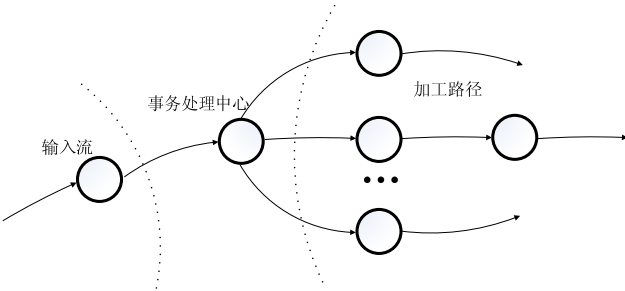


图 3-3 事务型数据流

3. 设计的过程

数据流设计方法的过程如下：

- (1) 精化 DFD。指把 DFD 转换成软件结构图前，设计人员要仔细地研究分析 DFD，并参照数据字典，认真理解其中的有关元素，检查有无遗漏或不合理之处，进行必要的修改。

(2) 确定 DFD 类型。如果是变换型，确定变换中心和逻辑输入、逻辑输出的界线，映射为变换结构的顶层和第一层；如果是事务型，确定事务中心和加工路径，映射为事务结构的顶层和第一层。

(3) 分解上层模块，设计中、下层模块结构。

(4) 根据优化准则对软件结构求精。

(5) 描述模块功能、接口及全局数据结构。

(6) 复查，进入详细设计。

3.3.4 详细设计

详细设计的主要任务是设计每个模块的实现算法、所需的局部数据结构。详细设计的目标有两个：实现模块功能的算法逻辑上正确和算法描述要简明易懂。

1. 详细设计任务

(1) 为每个模块确定采用的算法，选择某种适当的工具表达算法的过程，写出模块详细的过程性描述。

(2) 确定每一模块使用的数据结构。

(3) 确定模块接口的细节，包括对系统外部的接口和用户界面，对系统内部其他模块的接口，以及模块输入数据、输出数据及局部数据的全部细节。

在详细设计结束时，应该把上述结果写入详细设计说明书，并且通过复审形成正式文档，交付给下一阶段（编码阶段）的工作依据。

(4) 要为每一个模块设计出一组测试用例，以便在编码阶段对模块代码（即程序）进行预定的测试，模块的测试用例是软件测试计划的重要组成部分，通常应包括输入数据，期望输出等内容。

2. 详细设计的工具

(1) 图形工具。利用图形工具可以把过程的细节用图形描述出来，如程序流程图、N-S 图、PAD 图和 HIPO 图等。

(2) 表格工具。可以用一张表来描述过程的细节，在这张表中列出了各种可能的操作和相应的条件，如判定表等。

(3) 语言工具。用某种高级语言（称之为伪码）来描述过程的细节，如 PDL（伪码）等。

详细设计的任务是，为软件结构图中的每一个模块确定实现算法和局部数据结构，用某种特定的表达工具表示算法和数据结构的细节。表达工具可以由设计人员自由选择，但它应该具有描述过程细节的能力，而且能够使程序员在编程时便于直接翻译成程序设计语言的源程序。

3.4 软件测试

软件测试就是利用测试工具按照测试方案和流程对产品进行功能和性能测试，甚至根据需要编写不同的测试工具，设计和维护测试系统，对测试方案可能出现的问题进行分析和评估。执行测试用例后，需要跟踪故障，以确保开发的产品适合需求。

3.4.1 软件测试概述

软件测试是为了发现错误而执行程序的过程，成功的测试是为了发现至今尚未发现的错误的测试，测试的目的就是希望能以最少的人力和时间发现潜在的各种错误和缺陷，应根据开发各阶段的需求、设计等文档或程序内部结构精心设计测试用例，并利用这些实例来运行程序，以便发现错误。

1. 软件测试的原则

(1) 软件开发人员即程序员应当避免测试自己的程序。

(2) 应尽早和不断地进行软件测试。

(3) 对测试用例要有正确的态度。

(4) 严格执行测试计划，排除测试的随意性，以避免发生疏漏或者重复无效的工作。

(5) 应当对每一个测试结果进行全面检查。一定要全面地、仔细地检查测试结果，但常常被人们忽略，导致许多错误被遗漏。

(6) 妥善保存测试用例、测试计划、测试报告和最终分析报告，以备回归测试及维护之用。

2. 软件测试的分类

(1) 从软件内部结构和具体实现的角度划分分为白盒测试，黑盒测试和灰盒测试。

(2) 从是否执行程序的角度划分分为静态测试和动态测试。

(3) 从软件开发的过程按阶段划分有单元测试，集成测试，确认测试（验收测试）和系统测试。

3. 软件测试的步骤

(1) 单元测试。单元测试又称模块测试，是针对软件设计的最小单位——程序模块，进行正确性检验的测试工作。其目的在于发现各模块内部可能存在的各种差错。

(2) 集成测试。集成测试（集成测试、联合测试）需要将所有模块按照设计要求组装成为系统而进行的测试。

(3) 确认测试。确认测试又称有效性测试。任务是验证软件的功能和性能及其他特性是否与用户的要求一致。

(4) 系统测试。系统测试，是将通过确认测试的软件，作为整个基于计算机系统的一个元素，与计算机硬件、外设、某些支持软件、数据和人员等其他系统元素结合在一起，在实际运行环境下，对计算机系统进行一系列的组装测试和确认测试。

(5) 验收测试。验收测试是确保软件准备就绪，并且可以让最终用户将其用于执行软件的既定功能和任务。

3.4.2 软件测试技术

1. 静态测试

静态测试包括代码检查、静态结构分析、代码质量度量等。静态测试可以由人工进行，充分发挥人的逻辑思维优势，也可以借助软件工具自动进行。经验表明，使用人工测试能够有效地发现 30%~70% 的逻辑设计和编码错误。

代码检查主要检查代码和设计的一致性，包括代码逻辑表达的正确性，代码结构的合理性等方面。这项工作可以发现违背程序编写标准的问题，如程序中不安全、不明确和模糊的部分，找出程序中不可移植部分、违背程序编写风格的问题，包括变量检查、命名和类型审查、程序逻辑审查、程序语法检查和程序结构检查等内容。代码检查包括代码审查、代码走查、桌面检查、静态分析等具体方式。

2. 动态测试

静态测试不实际运行软件，主要通过人工进行。动态测试是基于计算机的测试，是为了发现错误而执行程序的过程。或者说，是根据软件开发各阶段的规格说明和程序的内部结构而精心设计的一批测试用例（即输入数据及其预期的输出结果），并利用这些测试用例去运行程序，以发现程序错误的过程。

3. 白盒测试方法

白盒测试也称结构测试或逻辑驱动测试，它是按照程序内部的结构测试程序，通过测试来检测产品内部动作是否按照设计规格说明书的规定正常进行，检验程序中的每条通路是否都能按预定要求正确工作。

这一方法是把测试对象看做一个打开的盒子，测试人员依据程序内部逻辑结构相关信息，设计或选择测试用例，对程序所有逻辑路径进行测试，通过在不同点检查程序的状态，确定实际的状态是否与预期的状态一致。

白盒测试的主要方法有逻辑覆盖、基本路径测试等

(1) 逻辑覆盖测试

逻辑覆盖是指一系列以程序内部的逻辑结构为基础的测试用例技术。逻辑表示有判断、分支、条件等几种表示方式。

- 语句覆盖：选择足够的测试用例，使程序中的每一个语句至少都执行一次。
- 路径覆盖：执行足够的测试用例，使程序中所有的可能路径都至少执行一次。

- 判定覆盖：使设计的测试用例保证程序中每个判断的每个取值分支至少经历一次。
- 条件覆盖：设计的测试用例保证程序中每个判断的每个条件的可能取值至少执行一次。
- 判断-条件覆盖：设计足够的测试用例，使判断中每个条件的所有可能取值至少执行一次，同时每个判断的所有取值分支至少执行一次。

(2) 基本路径测试

基本测试路径的思想是根据软件过程描述中的控制流程确定程序的环路复杂性度量，用此度量定义基本路径集合，并由此导出一组测试用例对每一条独立执行路径的测试。

“白盒”测试全面了解程序内部逻辑结构、对所有逻辑路径进行测试。“白盒”测试是穷举路径测试。在使用这一方案时，测试者必须检查程序的内部结构，从检查程序的逻辑出发，得出测试数据。贯穿程序的独立路径数是天文数字。但即使每条路径都测试了仍然可能有错误。第一，穷举路径测试不能查出程序违反了设计规范，即程序本身是个错误的程序。第二，穷举路径测试不可能查出程序中因遗漏路径而出错。第三，穷举路径测试可能发现不了与数据相关的错误。

4. 黑盒测试方法

黑盒测试也称功能测试，它是通过测试来检测每个功能是否都能正常使用。在测试中，把程序看做一个不能打开的黑盒子，在完全不考虑程序内部结构和内部特性的情况下，在程序接口进行测试，它只检查程序功能是否按照需求规格说明书的规定正常使用，程序是否能适当地接收输入数据而产生正确的输出信息。黑盒测试着眼于程序外部结构，不考虑内部逻辑结构，主要针对软件界面和软件功能进行测试。

黑盒测试是以用户的角度，从输入数据与输出数据的对应关系出发进行测试的。很明显，如果外部特性本身设计有问题或规格说明的规定有误，用黑盒测试方法是发现不了的。

黑盒测试方法主要有等价类划分法、边界值分析法、错误推测法、因果法等，主要用于软件确认测试。

黑盒测试法注重于测试软件的功能需求，主要试图发现下列几类错误：

- (1) 功能不正确或遗漏。
- (2) 界面错误。
- (3) 数据库访问错误。
- (4) 性能错误。
- (5) 初始化和终止错误等。

从理论上讲，黑盒测试只有采用穷举输入测试，把所有可能的输入都作为测试情况考虑，才能查出程序中所有的错误。实际上测试情况有无穷多个，人们不仅要测试所有合法的输入，而且还要对那些不合法但可能的输入进行测试。这样看来，完全测试是不可能的，所以我们要进行有针对性的测试，通过制定测试案例指导测试的实施，保证软件测试有组织、按步骤，以及有计划地进行。黑盒测试行为必须能够加以量化，才能真正保证软件质量，而测试用例就是将测试行为具体量化的方法之一。

3.5 程序的调试

所谓程序调试，是在编制的程序投入实际运行前，用手工或编译程序等方法进行测试，修正语法错误和逻辑错误的过程。这是保证计算机信息系统正确性的必不可少的步骤。

3.5.1 步骤与方法

程序调试的步骤如下：

- (1) 用编辑程序把编制的源程序按照一定的书写格式送到计算机中，编辑程序会根据使用人员的意图对源程序进行增、删或修改。
- (2) 把送入的源程序翻译成机器语言，即用编译程序对源程序进行语法检查，并将符合语法规则的源程序语句翻译

成计算机能识别的“语言”。如果经编译程序检查，发现有语法错误，那就必须用编辑程序来修改源程序中的语法错误，然后再编译，直至没有语法错误为止。

(3) 使用计算机中的连接程序，把翻译好的计算机语言程序连接起来，并连接成一个计算机能真正运行的程序。在连接过程中，一般不会出现连接错误，如果出现了连接错误，说明源程序中存在子程序的调用混乱或参数传递错误等问题。这时又要用编辑程序对源程序进行修改，再进行编译和连接，如此反复进行，直至没有连接错误为止。

(4) 将修改后的程序进行试算，这时可以假设几个模拟数据去试运行，并把输出结果与手工处理的正确结果相比较。如有差异，就表明计算机的程序存在有逻辑错误。如果程序不大，可以用人工方法模拟计算机对源程序的这几个数据进行修改处理；如果程序比较大，人工模拟显然行不通，这时只能将计算机设置成单步执行的方式，一步步跟踪程序的运行。一旦找到问题所在，仍然要用编辑程序来修改源程序，接着仍要编译、连接和执行，直至无逻辑错误为止。

程序调试主要有两种方法，即静态调试和动态调试。

3.5.2 静态调试

1. 对程序语法规则进行检查

(1) 语句正确性检查。保证程序中每个语句的正确性是编写程序的基本要求。由于程序中包含大量的语句，书写过程中由于疏忽或笔误，语句写错在所难免。对程序语句的检查应注意以下几点：

- 检查每个语句的书写是否有字符遗漏，包括必要的空格符。
- 检查形体相近的字符是否书写正确，如字母 o 和数字 0，书写时要有明显的分别。
- 检查函数调用时形参和实参的类型、个数是否相同。

(2) 语法正确性检查。每种计算机语言都有自己的语法规则，在书写程序时必须遵守一定的语法规则，否则在编译时程序将给出错误信息。

(3) 语句的配对检查。许多语句都是配对出现的，不能只写半个语句。另外，当语句有多重括号时，每个括号也都应成对出现，不能缺少左或右的。

(4) 注意检查语句顺序。有些语句不仅句法本身要正确，而且语句在程序中的位置也必须正确。例如，变量定义要放在所有可执行语句之前。

2. 检查程序的逻辑结构

(1) 检查程序中各变量的初值和初值的位置是否正确。我们经常遇到的是“累加”、“累乘”，其初值和位置都非常重要。用于累加的变量应取 0 初值或给定的初值，用于累乘的变量应赋初值为 1 或给定的值。因为累加或累乘都是通过循环结构来实现的，因此这些变量赋初值语句应在循环体之外。对于多重循环结构，内循环体中的变量赋初值语句应在内循环之外；外循环体中的变量赋初值语句应在外循环之外。如果赋初值的位置放错了，那么将得不到预想的结果。

(2) 检查程序中分枝结构是否正确。程序中的分枝结构都是根据给定的条件来决定执行不同的路径的，因此在设置各条路径的条件时一定要谨慎，在设置“大于”、“小于”这些条件时，一定要仔细考虑是否应该包括“等于”这个条件，更不能把条件写反。尤其要注意的是，实型数据在运算过程中会产生误差，如果用“等于”或“不等于”对实数的运算结果进行比较，则会因为误差而产生误判断，路径选择也就错了。因此在遇到要判断实数 a 与 b 相等与否作为条件来选择路径时，应该把条件写成： $\text{if } (\text{fabs}(a-b) \leq 1e-6)$ ，而不应该写成 $\text{if}(a=b)$ 。要特别注意条件语句嵌套时，if 和 else 的配对关系。

(3) 检查程序中循环结构的循环次数和循环嵌套的正确性。C 语言中可用 for 循环、while 循环、do-while 循环。在给定的循环条件时，不仅要考虑循环变量的初始条件，还要考虑循环变量的变化规律、循环变量变化的时间，任何一条变化都会引起循环次数的变化。

(4) 检查表达式的合理与否。程序中不仅要保证表达式的正确性，而且还要保证表达式的合理性。尤其要注意表达式运算中的溢出问题，运算数值可能超出整数范围就不应该采用整型运算，否则必然导致运算结果的错误。两个相近的数不能相减，以免产生“下溢”。更要避免在一个分式的分母运算中发生“下溢”，因为编译系统常把下溢做零处理。因此分母中出现下溢时要产生“被零除”的错误。由于表达式不合理而引起的程序运行错误往往很难查找，会增加程序调

试的难度。因此,认真检查表达式的合理性,是减少程序运行错误,提高程序动态调试效率的重要方面。

(5) 程序的静态调试是程序调试非常重要的一步。初学者应培养自己静态检查的良好习惯,在上机前认真做好程序的静态检查工作,从而节省上机时间,使有限的机时充分发挥作用。

3.5.3 动态调试

在静态调试中可以发现和改正很多错误,但由于静态调试的特点,有一些比较隐蔽的错误还不能检查出来。只有上机进行动态调试,才能够找到这些错误并改正它们。

1. 编译过程中的调试

编译过程除了将源程序翻译成目标程序外,还要对源程序进行语法检查。如果发现源程序有语法错误,系统将显示错误信息。用户可以根据这些提示信息查找出错误性质,并对程序中出错之处进行相应的修改。有时我们会发现编译时有几行的错误信息都是一样的,但检查这些行本身没有发现错误,这时要仔细检查与这些行有关的名字、表达式是否有问题。例如,因为程序中数组说明语句有错,这时,那些与该数组有关的程序行都会被编译系统检查出错。在这种情况下,用户只要仔细分析,修改数组说明语句的错误,许多错误就会同时没有了。对于编译阶段的调试,要充分利用屏幕给出的错误信息,对它们进行仔细分析判断。只要注意总结经验,使程序通过编译是不难做到的。

2. 连接过程的调试

编译通过后要连接。连接的过程也有查错的功能,它将指出外部调用、函数之间的联系及存储区设置等方面的错误。如果连接时有这类错误,编译系统也会给出错误信息,用户要对这些信息仔细判断,从而找出程序中的问题并改正。在连接时较常见的错误有以下几类:

(1) 某个外部调用有错,通常系统明确提示了外部调用的名字,只要仔细检查各模块中与该名有关的语句,就不难发现错误。

(2) 找不到某个库函数或某个库文件,这类错误是由于库函数名写错、疏忽了某个库文件的连接等。

(3) 某些模块的参数超过系统的限制,如模块的大小、库文件的个数超出要求等。引起连接错误的原因很多,而且很隐蔽,给出的错误信息也不如编译时给出的直接、具体。因此,连接时的错误要比编译时的错误更难查找,需要仔细分析判断,而且对系统的限制和要求要有所了解。

3. 运行过程中的调试

运行过程中的调试是动态调试的最后一个阶段。这一阶段的错误大体可分为两类:

(1) 运行程序时给出出错信息。运行时出错多与数据的输入/输出格式有关,与文件的操作有关。如果给出数据格式有错,这时要对有关的输入/输出数据格式进行检查,一般容易发现错误。如果程序中的输入/输出函数较多,则可以在中间插入调试语句,采取分段隔离的方法,很快就可以确定错误的位置了。如果是文件操作有误,也可以针对程序中的有关文件的操作采取类似的方法进行检查。

(2) 运行结果不正常或不正确。运行结果不正确,一般来说是程序设计方面出现了一定的问题。

3.6 本章习题

3.6.1 选择题

1. 下列工具中为需求分析常用工具的是()。

A. PAD B. BFD C. N-S D. DFD

2. 软件设计一般分为两步完成,它们是()。

A. 概要设计与详细设计 B. 数据设计与接口设计

- C. 软件结构设计与数据设计 D. 过程设计与数据设计
3. 软件生命周期中所花费最多的阶段是（ ）。
- A. 详细设计 B. 软件编码
- C. 软件测试 D. 软件维护
4. 在软件开发中，需求分析阶段产生的主要文档是（ ）。
- A. 可行性分析报告 B. 软件需求规格说明书
- C. 概要设计说明书 D. 集成测试计划
5. 下列关于软件工程的描述中正确的是（ ）。
- A. 软件工程只是解决软件项目的管理问题
- B. 软件工程只是解决产品的生产率问题
- C. 软件工程的主要思想是强调在软件开发过程中需要应用工程化原则
- D. 软件工程只是解决软件开发中的技术问题
6. 在软件设计中，不属于过程设计工具的是（ ）。
- A. PDL B. PAD 图
- C. N-S 图 D. DFD 图
7. 下列描述中正确的是（ ）。
- A. 软件测试应该由程序开发者来完成 B. 程序经调试后一般不需要测试
- C. 软件维护只包括对程序代码的维护 D. 以上 3 种说法都不对
8. 下面不属于软件工程要素的是（ ）。
- A. 工具 B. 过程 C. 方法 D. 环境
9. 软件工程学一般包括软件开发技术和软件工程管理两方面的内容，软件工程经济学是软件工程管理的技术内容之一，它专门研究（ ）。
- A. 软件开发的方法学 B. 软件开发技术和工具
- C. 软件成本效益分析 D. 计划、进度和预算
10. 软件需要分析阶段的工作，可以分为 4 个方面，即需求获取、需求分析、编写需求规格说明书及（ ）。
- A. 阶段性能报告 B. 需求评审 C. 总结 D. 都不正确
11. 设计软件结构是软件生命周期的（ ）。
- A. 软件定义期 B. 软件开发期
- C. 软件维护期 D. 以下 3 个都不是
12. 数据流图用于抽象描述一个软件的逻辑模型，数据流图由一些特定的图形符号组成，下列不符合数据流图的是（ ）。
- A. 控制流 B. 加工 C. 数据存储 D. 源和潭

3.6.2 填空题

1. 软件产品从提出、实现、使用维护到停止使用的过程称为_____。
2. 在结构化设计方法中，数据流图表达了问题中的数据和加工间的关系，并且每一个_____实际上对应一个处理模块。
3. 等价类划分法是_____测试常用方法。
4. 软件生命周期包括 3 个阶段。为使各时期的任务明确，可以分为以下 3 个时期：软件定义期、软件开发期、软件维护期。编码和测试属于_____期。
5. 软件是程序、数据和_____的集合。
6. 全面支持软件开发过程的软件工具集合称为_____。
7. 软件设计模块的目的是_____程序设计的复杂性。

第4章

数据库设计基础

本章主要介绍了数据库的基本概念、数据模型、关系代数等内容。结合计算机等级考试二级 C 语言要求，具体如表 4-1 所示。

表 4-1 考试要求

考试知识点	重要性
数据库系统的基本概念	★
数据模型	★★★
关系代数	★★
数据库设计与管理	★★

4.1 数据库的基本概念

在计算机系统开发过程中，数据库是非常重要的组成部分，数据库就是存放数据的仓库，人们在收集并抽取了大量数据之后，应将其保存起来，以供进一步的发展，而这些数据保存的地方就是数据库。

4.1.1 数据和信息

数据处理是计算机应用的主流，特别是关系数据库管理系统 DBSEIII推出之后，数据库技术就成为计算机应用者必须掌握的技能。数据和信息是数据处理中两个最基本的概念。

数据是客观事物的属性值，反映客观事物的特征。信息则是由客观事物得到的，使人们能够认知客观事物的各种消息、情报、数据、信号所包括的内容。

数据是物理的，信息是观念性的。信息可以数据化，数据代表信息，但二者是有区别的。信息是现实世界中的事物反映到人们头脑并经过识别、选择、命名和分类等综合分析形成印象和概念并产生的认识。数据是现实世界中的客观现象经过信息抽象后的表示形式。

数据不仅包括数值数据，而且包括各种非数值数据。
对计算机而言，数据则是指能够用计算机处理的数字、字母和符号，包括数值数据和非数值数据。

4.1.2 数据处理、数据库与数据库管理系统

数据处理包括对数据进行收集、存储、传送、整理、检索、计算、输出等各种加工、管理。
数据库（DataBase， DB）是数据的集合，它有统一的结构形式存放于统一的存储介质内，是多种应用数据的集成，

并可被各个应用程序所共享。

数据库中的数据具有“集成”、“共享”特点，即数据库集中了个各种应用的数据，进行统一的构造与存储，而使它们可被不同应用程序所使用。

1. 数据库系统

数据库系统（Database System，DBS）由如下几部分组成：数据库、数据库管理系统、数据库管理员、硬件平台、软件平台。这 5 个部分构成了一个以数据库为核心的完整的运行实体，称为数据库系统。

在数据库系统中，硬件平台包括：

- 计算机：它是系统中硬件的基础平台，目前常用的有微型机、小型机、中型机、大型机。
- 网络：以建立在网络上为主，而其结构形式又以客户、服务器（C/S）方式与浏览器/服务器（B/S）方式为主。

在数据库系统中，软件平台包括：

- 操作系统：它是系统的基础软件平台，目前常用的有各种 UNIX（包括 Linux）与 Windows 两种。
- 数据库系统开发工具：为开发数据库应用程序所提供的工具，包括可视化开发工具 VB.net、Eclipse、SQL2005 等。

2. 数据库管理系统

数据库管理系统（Database Management System，简称 DBMS）是数据库的机构，用来帮助用户建立、使用、管理、加工和维护数据库而配置的专用系统软件，它在操作系统的支持下对数据库进行统一的管理和控制。数据库中的数据有时具有海量级的数据，并且其结构复杂，因此需要提供管理工具。数据库管理系统是数据库系统的核心，它主要有如下几方面的具体功能：

- （1）数据模式定义。数据库管理系统负责为数据库构建模式，也就是为数据库构建起数据框架。
- （2）数据存取的物理构建。数据库管理系统负责为数据模式的物理存取及构建提供有效的存取方法与手段。
- （3）数据操纵。数据库管理系统为用户使用数据库中的数据提供方便，它一般提供查询、插入、修改及删除数据的功能。此外，它自身还具有做简单算术运算及统计的能力，而且还可以与某些过程性语言结合，使其具有强大的过程性操作能力。
- （4）数据的完整性、安全性定义与检查。数据库中的数据具有内在语义上的关联性与一致性，它们构成了数据的完整性，数据的完整性是保证数据库中数据正确的必要条件，因此必须经常检查以维护数据的正确。
- （5）数据库的并发控制与故障恢复。数据库是一个集成、共享的数据集合体，它能为多个应用程序服务，所以存在多个应用程序对数据库的并发操作。在并发操作中如果不加控制和管理，多个应用程序间就会相互干扰，从而对数据库中的数据造成破坏。因此，数据库管理系统必须对多个应用程序的并发操作进行必要的控制以保证数据不受破坏，这就是数据库的并发控制。
- （6）数据的服务。数据库管理系统提供对数据库中数据的多种服务功能，如数据拷贝、转存、重组、性能监测、分析等。

为完成以上 6 个功能，数据库管理系统一般提供相应的数据语言（Data Language），它们是：

- 数据定义语言（Data Definition Language，简称 DDL）。该语言负责数据的模式定义与数据的物理存取构建。
- 数据操纵语言（Data Manipulation Language，简称 DML）。该语言负责数据的操纵，包括查询、增、删、改等操作。
- 数据控制语言（Data Control Language，简称 DCL）。该语言负责数据完整性、安全性的定义与检查，以及并发控制、故障恢复等功能，包括系统初启程序、文件读写与维护程序、存取路径管理程序、缓冲区管理程序、安全性控制程序、完整性检查程序、并发控制程序、事务管理程序、运行日志管理程序、数据库恢复程序等。

3. 数据库应用系统

数据库应用系统（Database Application System，简称 DBAS）是程序员根据用户的需要在数据库管理系统的支持下，用数据库管理系统提供的命令编写、开发并能够在数据库管理系统的支持下运行的程序和数据库的总称。如各种财务管理系统、人事管理系统、物质管理系统等。它包含 5 个主要方面，分别是硬件、操作系统、数据库管理系统、应用开发工具、应用系统。

4.1.3 数据库系统的发展

数据管理发展至今已经经历了3个阶段：人工管理阶段、文件系统阶段和数据库系统阶段。人工管理阶段是在20世纪50年代中期以前，主要用于科学计算，硬件无磁盘，直接存取，软件没有操作系统。20世纪50年代后期到20世纪60年代中期，进入文件系统阶段。20世纪60年代之后，数据管理进入数据库系统阶段。随着计算机应用领域不断扩大，数据库系统的功能和应用范围也愈来愈广，到目前已成为计算机系统的基本及主要的支撑软件。

1. 文件系统阶段

文件系统是数据库系统发展的初级阶段，它提供了简单的数据共享与数据管理能力，但是它无法提供完整的、统一的管理和数据共享的能力。由于它的功能简单，因此它附属于操作系统而不成为独立的软件，目前一般将其看成仅是数据库系统的雏形，而不是真正的数据库系统。

2. 层次数据库与网状数据库系统阶段

从20世纪60年代末期起，真正的数据库系统——层次数据库与网状数据库开始发展，它们为统一管理与共享数据提供了有力支撑，这个时期数据库系统蓬勃发展形成了有名的“数据库时代”。但是这两种关系也存在不足，主要是它们脱胎于文件系统，受文件的物理影响较大，对数据库使用带来诸多不便，同时，此类系统的数据模式构造烦琐不宜于推广使用。

3. 关系数据库系统阶段

关系数据库系统出现于20世纪70年代，在80年代得到蓬勃发展，并逐渐取代前两种系统。关系数据库系统结构简单，使用方便，逻辑性强物理性少，因此在80年代以后一直占据数据库领域的主导地位，但是由于此系统来源于商业应用，适合于事务处理领域，而对非事务领域应用受到限制，因此在80年代末期兴起了与应用技术相结合的各种专用数据库系统。

目前，数据库技术也与其他信息技术一样在迅速发展之中，计算机处理能力的增强和越来越广泛的应用是促进数据库技术发展的重要动力。一般认为，未来的数据库系统应支持数据管理、对象管理和知识管理，应该具有面向对象的基本特征。在关于数据库的诸多新技术中，下面3种是比较重要的：

- 面向对象数据库系统：用面向对象方法构筑面向对象数据模型，使其具有比关系数据库系统更为通用的能力。
- 知识库系统：用人工智能中的方法特别是用谓词知识表示方法构筑数据模型，使其模型具有特别通用的能力。
- 关系数据库系统的扩充：利用关系数据库系统进行进一步扩展，使其在模型的表达能力与功能上有进一步的加强，如网络技术相结合的Web数据库、数据仓库及嵌入式数据库等。

数据库技术是在文件系统基础上发展产生的，两者都以数据文件的形式组织数据，但由于数据库系统在文件系统之上加入了DBMS对数据进行管理，从而使得数据库系统具有以下特点：

- 数据的集成性。
- 数据的高共享性与低冗余性。
- 数据独立性。
- 数据统一管理与控制。

4.1.4 数据库系统的内部结构体系

数据库系统的内部结构主要体现在三级模式和两级映射上。

1. 数据库系统的三级模式

数据模式是数据库系统中数据结构的一种表示形式，它具有不同的层次与结构方式，如图4-1所示。

(1) 概念模式。概念模式（Conceptual Schema）是数据库系统中全局数据逻辑结构的描述，是全体用户（应用）公共数据视图。此种描述是一种抽象的描述，它不涉及具体的硬件环境与平台，也与具体的软件环境无关。

概念模式主要描述数据的概念记录类型及它们间的关系，它还包括一些数据间的语义约束，对它的描述可用DBMS

中的 DDL 语言定义。

(2) 外模式。外模式 (External Schema) 也称子模式 (Subschema) 或用户模式 (User's schema)。它是用户的数据视图, 也就是用户所见到的数据模式, 它由概念模式推导而出。概念模式给出了系统全局的数据描述而外模式则给出每个用户的局部数据描述。一个概念模式可以有若干个外模式, 每个用户只关心与它有关的模式, 这样不仅可以屏蔽大量无关信息而且有利于数据保护。在一般的 DBMS 中都提供有相关的外模式描述语言 (外模式 DDL)。

(3) 内模式。内模式 (Internal Schema) 又称物理模式 (Physical Schema), 它给出了数据库物理存储结构与物理存取方法, 如数据存储的文件结构、索引、集簇及 hash 等存取方式与存取路径, 内模式的物理性主要体现在操作系统及文件级上, 它还未深入到设备级上 (如磁盘及磁盘操作)。内模式对一般用户是透明的, 但它的设计直接影响数据库的性能。DBMS 一般提供相关的内模式描述语言 (内模式 DDL)。

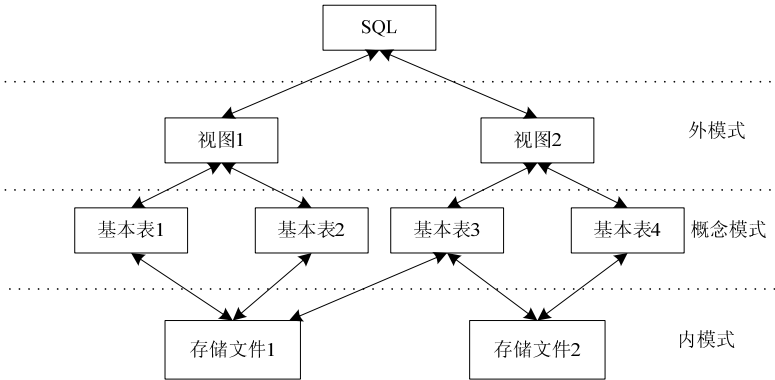


图 4-1 三级模式图

模式的 3 个级别层次反映了模式的 3 个不同环境及它们的不同要求, 其中内模式处于最底层, 它反映数据在计算机物理结构中的实际存储形式, 概念模式处于中层, 它反映了设计者的数据全局逻辑要求, 而外模式处于最外层, 它反映了用户对数据的要求。

2. 数据库系统的两级映射

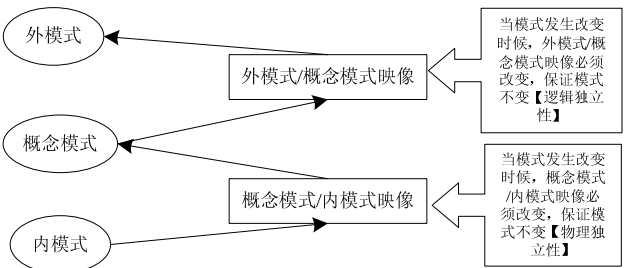


图 4-2 两级映射

数据库系统的三级模式是对数据的 3 个级别抽象, 它把数据的具体物理实现留给物理模式, 使用户与全局设计者不必关心数据库的集体实现与物理背景; 同时, 它通过两级映射建立了模式间的联系与转换, 使得概念模式与外模式虽然并不具备物理存在, 但是也能通过映射而获得其实体。此外, 两级映射也保证了数据库系统中数据的独立性, 即数据的物理组织改变与逻辑概念级改变相互独立, 使得只要调整映射方式而不必改变用户模式, 如图 4-2 所示。

(1) 概念模式到内模式的映射。该映射给出了概念模式中的全局逻辑结构到数据的物理存储结构间的对应关系, 此种映射一般由 DBMS 实现。

(2) 外模式到概念模式的映射。概念模式是一个全局模式而外模式是用户的局部模式。一个概念模式中可以定义多个模式, 而每个外模式是概念模式的一个基本视图。外模式到概念模式的映射给出了外模式与概念模式的对应关系, 这种映射一般也是由 DBMS 来实现的。

4.2 数据模型

数据库中的数据模型可以将复杂的现实世界要求反映到计算机数据库中的物理世界, 从现实世界来看, 用户为了某

种需要,需将现实世界中的部分需求用数据库实现,而数据模型(Data Model)是现实世界中数据特征的抽象,为数据库系统的信息表达与操作提供一个抽象的框架。

4.2.1 数据模型概述

数据模型所描述的内容包括3个部分:数据结构、数据操作、数据约束。

(1) 数据结构。数据模型中的数据结构主要描述数据的类型、内容、性质及数据间的联系等。数据结构是数据模型的基础,数据操作和约束都建立在数据结构上。不同的数据结构具有不同的操作和约束。

(2) 数据操作。数据模型中数据操作主要描述相应的数据结构上的操作类型和操作方式。

(3) 数据约束。数据模型中的数据约束主要描述数据结构内数据间的语法、词义联系、他们之间的制约和依存关系,以及数据动态变化的规则,以保证数据的正确、有效和相容。

数据模型的分类如下:

(1) 概念数据模型简称概念模型,它是一种面向客观世界、面向用户的模型,它与具体的数据库管理系统无关,与具体的计算机平台无关。概念模型着重于对客观世界复杂事物的结构描述及它们之间的内在联系的刻画。概念模型是整个数据模型的基础。

(2) 逻辑数据模型又称数据模型,它是一种面向数据库系统的模型,该模型着重于在数据库系统一级的实现。概念模型只有在转换成数据模型后才能在数据库中得以表示。目前,逻辑数据模型也有很多种,较为成熟并先后被人们大量使用过的有层次模型、网状模型、关系模型、面向对象模型等。

(3) 物理数据模型又称物理模型,它是一种面向计算机物理表示的模型,此模型给出了数据模型在计算机上物理结构的表示。

目前,数据库领域中最常用的数据模型有4种:

- 层次模型(Hierarchical Model):用层次结构表示实体类型及实体间联系的数据模型。
- 网状模型(Network Model):用有向图结构表示实体类型及实体间联系的数据结构模型。
- 关系模型(Relational Model):关系数据模型是以关系数学理论为基础的,用二维表结构来表示实体及实体间联系的模型。
- 面向对象模型(Object Oriented Model):对象模型表示了静态的、结构化的系统数据性质,描述了系统的静态结构,它是从客观世界实体的对象关系角度来描述的,表现了对象的相互关系。

4.2.2 E-R 模型

概念模型中被广泛使用的模型是E-R模型(entity-relationship model),该模型将现实世界的要求转化成实体、联系、属性等几个基本概念,以及它们间的两种基本连接关系,并且可以用一种图非常直观地表示出来。它是数据库系统开发中最重要的模型。

1. E-R 模型的基本概念

(1) 实体。现实世界中的事物可以抽象成为实体,实体是概念世界中的基本单位,它们是客观存在的且又能相互区别的事物。凡是有共性的实体可组成一个集合称为实体集(entity set)。例如,小赵、小李是实体,他们又均是学生而组成一个实体集。

(2) 属性。现实世界中的事物均有一些特性,这些特性可以用属性来表示。属性刻画了实体的特征。一个实体往往可以有若干个属性。每个属性可以有值,一个属性的取值范围称为该属性的值域(value domain)或值集(value set)。例如,小赵年龄取值为17,小李为19。

(3) 联系。现实世界中事物间的关联称为联系。在概念世界中联系反映了实体集间的一定关系。

两个实体集间的联系实际上是实体集间的函数关系,这种函数关系可以有下面几种:

- 一对一(one to one)的关系,简记为1:1。这种函数关系是常见的函数关系之一,例如,学校与校长间的联系,一个学校与一个校长间相互一一对应,如图4-3所示。



图 4-3 一对一关系图

- 一对多（one to many）或多对一（many to one）关系，简记为 1：M（1：m）或 M：1（m：1）。这两种函数关系实际上是一种函数关系，例如，班主任与其学生的联系是多对一的关系（反之，则为一对多关系），即多个学生对应一个班主任，如图 4-4 所示。



图 4-4 一对多关系图

- 多对多（many to many）关系，简记为 M：N 或 m：n。这是一种较为复杂的函数关系，例如，课程与学生这两个实体集间的关系是多对多的，因为一门课程可以由多个学生学习，而一个学生又可以学习多个课程，如图 4-5 所示。



图 4-5 多对多关系图

2. E-R 模型的图示法

E-R 模型可以用一种图非常直观地表示，这种图称为 E-R 图（entity-relationship diagram）。在 E-R 图中我们分别用下面不同的几何图形表示 E-R 模型中的 3 个概念与两个连接关系。

- （1）实体集表示法。在 E-R 图中用矩形表示实体集，在矩形内写上该实体集的名字。如实体集学生（student）、课程（course）可用图 4-6 表示。
- （2）属性表示法。在 E-R 图中用椭圆形表示属性，在椭圆形内写上该属性的名称。如学生有属性：学号（S#）、姓名（Sn）及年龄（Sa），它们可以用图 4-7 表示。
- （3）联系表示法。在 E-R 图中用菱形（内写上联系名）表示联系。如学生与课程间的联系 SC，用图 4-8 表示。



图 4-6 实体集表示法

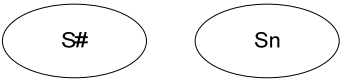


图 4-7 属性表示法

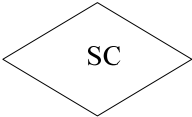


图 4-8 联系表示法

【例题 1】在 E-R 图中，用来表示实体联系的图形是_____。（2009 年 9 月）

- A. 椭圆图 B. 矩形 C. 菱形 D. 三角形

例题分析

实体之间的相互联系，应该用菱形来表示，答案选 C。

4.2.3 关系模型

关系模型采用二维表来表示，简称表。二维表由表框架（Frame）及表的元祖（Tuple）组成。表框架由 n 个命名的属

性 (Attribute) 组成, n 称为属性元数 (Arity)。每个属性有一个取值范围称为值域 (Domain)。表框架对应了关系的模式, 即类型的概念。

1. 关系模型的主要特点

关系数据模型是以集合论中的关系概念为基础发展起来的。关系模型中无论是实体还是实体间的联系均由单一的结构类型——关系来表示。现在流行的数据库就是基于关系模型的。其主要特点是:

- (1) 一个表中不允许出现相同的两个字段名。
- (2) 一个表中不允许出现完全相同的两行。
- (3) 一个表中同一列的数据项必须是类型相同的数据。
- (4) 一个表中每一行与每一列的数据项都是不可拆分的基本数据项。
- (5) 一个表中行或列的顺序改变都不影响表格所描述的内容。

2. 关系模型的重要概念——码

- 候选码: 指的是关系模式中的属性或属性组能唯一区分一条记录。
- 主码 (主键): 从候选码中选一个出来作为主码。
- 主属性: (主属性不是主码) 是指包含在候选码中的任何属性。
- 非主属性: 不包含在任何候选码中的属性。
- 规范化: 在关系数据库中的每一个关系 (数据表, 如学生表、课程表) 满足一定要求, 满足不同要求的为不同范式, 范式缩写 NF。
- 1NF: 每个分量必须是不可再分的。
- 2NF: 每一个非主属性都完全依赖候选码 (非主属性不能只依赖候选码中的部分)。
- 3NF: 每一个非主属性都不传递依赖于候选码 (这里非主属性都不传递依赖候选码是指非主属性之间不存在传递依赖的关系, 至于候选码内是否存在依赖就不用去管了, 我们只盯着非主属性)。
- BCNF: 每一个决定的属性集都包含候选码 (也就是说任何属性对候选码的传递都不行, 请注意, 这里是说任何属性, 也包含主属性, 因为在候选码中, 也有主属性之间相互传递的情况)。

【例题 2】人员基本信息一般包括: 身份证号、姓名、性别、年龄等。其中可以作为主关键字的是_____

例题分析

由于姓名、性别、年龄这 3 个属性可能存在重复的值, 无法达到区别一条记录的作用, 所以, 主关键字为身份证号。

4.2.4 数据操作

关系模型的数据操作是建立在关系上的数据操作, 一般有查询、增加、删除及修改 4 种操作。

1. 数据查询

用户可以查询关系数据库中的数据, 它包括一个关系内的查询, 以及多个关系间的查询。

(1) 对一个关系内查询的基本单位是元组分量, 其基本过程是先定位后操作。所谓定位包括纵向定位与横向定位两部分, 纵向定位即指定关系中的一些属性 (称列指定), 横向定位即选择满足某些逻辑条件的元组 (称为选择)。通过纵向与横向定位后一个关系中的元组分量即可确定了。在定位后即可进行查询操作, 就是将定位的数据从关系数据库中取出并放入指定内存。

(2) 对多个关系间的数据查询则可分为 3 步: 第一步, 将多个关系合并成一个关系; 第二步, 对合并后的一个关系进行定位; 第三步, 操作。其中第二步与第三步为对一个关系的查询。对多个关系的合并可分解成两个关系的逐步合并。

2. 数据删除

数据删除的基本单位是一个关系内的元组, 它的功能是将指定关系内的指定元组删除。它也分为定位与操作两部分,

其中定位部分只需要横向定位而无须纵向定位，定位后即执行删除操作。因此数据删除可以分解为一个关系内的元组选择与关系中元组删除两个基本操作。

3. 数据插入

数据插入仅对一个关系而言，在指定关系中插入一个或多个元组。在数据插入中不需要定位，仅需做关系中元组插入操作，因此数据插入只有一个基本操作。

4. 数据修改

数据修改是在一个关系中修改指定的元组与属性。数据修改不是一个基本操作。它可以分解为删除需修改的元组与插入修改后的元组两个基本的操作。

4.2.5 关系中的数据约束

关系模型允许定义 3 类数据约束，它们是实体完整性约束、参照完整性约束及用户定义的完整性约束，其中前两种完整性约束由关系数据库系统自动支持。对于用户定义的完整性约束，则由关系数据库系统提供完整性约束语言，用户利用该语言写出约束条件，运行时由系统自动检查。

(1) 实体完整性约束 (Entity Integrity Constraint)。该约束要求关系的主键中属性值不能为空值，这是数据库完整性的最基本要求，因为主键是唯一决定元组的，如为空值则其唯一性就成为不可能的了。

(2) 参照完整性约束 (Reference Integrity Constraint)。该约束是关系之间相关联的基本约束，它不允许关系引用不存在的元组，即在关系中的外键要么是所关联关系中实际存在的元组，要么就为空值。比如在关系 S (S#, SN, SD, SA) 与 SC (S#, C#, G) 中，SC 中主键为 (S#, C#) 而外键为 S#，SC 与 S 通过 S# 相关联，参照完整性约束要求 SC 中的 S# 的值必须在 S 中有相应元组值，如有 SC (S13, C8, 70)，则必须在 S 中存在 (S13, ...)。

(3) 用户定义的完整性约束 (User defined Integrity Constraint)。是针对具体数据环境与应用环境由用户具体设置的约束，它反映了具体应用中数据的语义要求。

实体完整性约束和参照完整性约束是关系数据库所必需遵守的规则，在任何一个关系数据库管理系统 (RDBMS) 中均由系统自动支持。

4.3 关系代数

关系代数是一种抽象的查询语言，用对关系的运算来表达查询，作为研究关系数据语言的数学工具。

关系代数的运算对象是关系，运算结果也为关系。关系代数用到的运算符包括 4 类：集合运算符、专门的关系运算符、算术比较符和逻辑运算符。

比较运算符和逻辑运算符是用来辅助专门的关系运算符进行操作的，所以关系代数的运算按运算符的不同主要分为传统的集合运算和专门的关系运算两类。

传统的集合运算是二目运算，包括并、交、差、广义笛卡儿积 4 种运算。

1. 并 (Union)

设关系 R 和关系 S 具有相同的目 n (即两个关系都有 n 个属性)，且相应的属性取自同一个域，则关系 R 与关系 S 的并由属于 R 或属于 S 的元组组成。其结果关系仍为 n 目关系，记为 $R \cup S = \{t | t \in R \vee t \in S\}$ 。

2. 差 (Difference)

设关系 R 和关系 S 具有相同的目 n ，且相应的属性取自同一个域，则关系 R 与关系 S 的差由属于 R 而不属于 S 的所有元组组成。其结果关系仍为 n 目关系，记为 $R - S = \{t | t \in R \wedge t \notin S\}$ 。

3. 交 (Intersection Referential integrity)

设关系 R 和关系 S 具有相同的目 n ，且相应的属性取自同一个域，则关系 R 与关系 S 的交由既属于 R 又属于 S 的元组组成。其结果关系仍为 n 目关系，记为 $R \cap S = \{t | t \in R \wedge t \in S\}$ 。

4. 广义笛卡儿积 (Extended cartesian product)

两个分别为 n 目和 m 目的关系 R 和 S 的广义笛卡儿积是一个 $(n+m)$ 列的元组的集合。元组的前 n 列是关系 R 的一个元组，后 m 列是关系 S 的一个元组。若 R 有 k_1 个元组， S 有 k_2 个元组，则关系 R 和关系 S 的广义笛卡儿积有 $k_1 \times k_2$ 个元组。

广义的笛卡儿积与并、交、差的不同之处在于，进行交、并、差 3 种运算的时候要求具有相同的列的个数和列名，而进行笛卡儿积的时候就不需要了。

5. 基本操作

(1) 选择。是指从一个关系（表）中找出满足一定条件的所有元组（记录），即在二维表中选取若干行。

(2) 投影。是指从一个关系（表）中选择所需要的若干属性（字段），构成一个新的关系（表），即在表中筛选出若干列。

(3) 连接。是指从两个关系中按一定的条件分别选取其中的若干属性（字段），形成新的关系。生成的新关系中行的选取，取决于规定的条件。

4.4 数据库设计

数据库设计 (Database Design) 是指对于一个给定的应用环境，构造最优的数据库模式，建立数据库及其应用系统，使之能够有效地存储数据，满足各种用户的应用需求（信息要求和处理要求）。在数据库领域内，常常把使用数据库的各类系统统称为数据库应用系统。

数据库设计目前一般采用生命周期 (life cycle) 法，即将整个数据库应用系统的开发分解成目标独立的若干阶段。分别是需求分析阶段、概念设计阶段、逻辑设计阶段、物理设计阶段、编码阶段、测试阶段、运行阶段、维护阶段。

1. 需求分析

数据库的需求分析是调查和分析用户的业务活动和数据的使用情况，弄清所用数据的种类、范围、数量，以及它们在业务活动中交流的情况，确定用户对数据库系统的使用要求和各种约束条件等，形成用户需求规约。从以下几个方面进行分析：

(1) 技术可行性：主要从项目实施的技术角度，合理设计技术方案，并进行比选和评价。各行业不同项目技术可行性的研究内容及深度差别很大。对于工业项目，可行性研究的技术论证应达到能够比较明确地提出设备清单的深度；对于各种非工业项目，技术方案的论证也应达到目前工程方案初步设计的深度，以便与国际惯例接轨。

(2) 经济可行性：主要从资源配置的角度衡量项目的价值，评价项目在实现区域经济发展目标、有效配置经济资源、增加供应、创造就业、改善环境、提高人民生活等方面的效益。

(3) 法律可行性：确认待开发的系统可能会涉及的任何侵犯、妨碍、责任等问题。

2. 概念设计

对用户要求描述的现实世界（可能是一个工厂、一个商场或者一个学校等），通过对其中住处的分类、聚集和概括，建立抽象的概念数据模型。这个概念模型应反映现实世界各部门的信息结构、信息流动情况、信息间的互相制约关系，以及各部门对信息储存、查询和加工的要求等。所建立的模型应避开数据库在计算机上的具体实现细节，用一种抽象的形式表示出来。以 E-R 模型的联系模型方法为例，第一步先明确现实世界各部门所含的各种实体及其属性、实体间的联系及对信息的制约条件等，从而给出各部门内所用信息的局部描述（在数据库中称为用户的局部视图）。第二步再将前面得到的多个用户的局部视图集成为一个全局视图，即用户要描述的现实世界的概念数据模型。

3. 逻辑设计

主要工作是将现实世界的概念数据模型设计成数据库的一种逻辑模式，即适应于某种特定数据库管理系统所支持的逻辑数据模式。与此同时，可能还需为各种数据处理应用领域产生相应的逻辑子模式。这一步设计的结果就是所谓“逻辑数据库”。

4. 物理设计

根据特定数据库管理系统所提供的多种存储结构和存取方法等依赖于具体计算机结构的各项物理设计措施，对具体的应用任务选定最合适的物理存储结构（包括文件类型、索引结构和数据的存放次序与位逻辑等）、存取方法和存取路径等。这一步设计的结果就是所谓“物理数据库”。

5. 编码设计和测试

在上述设计的基础上，收集数据并具体建立一个数据库，运行一些典型的应用任务来进行数据库的编码，在完成数据库的编码完成后，进行验证数据库设计的正确性和合理性。一般，一个大型数据库的设计过程往往需要经过多次循环反复测试。当设计的某步发现问题时，可能就需要返回到前面去进行修改。因此，在做上述数据库设计时就应考虑到今后修改设计的可能性和方便性。

6. 运行与维护设计

在数据库系统正式投入运行的过程中，必须不断地对其进行调整与修改。

4.5 本章习题

4.5.1 选择题

1. 数据库管理系统 DBMS 中用来定义模式、内模式和外模式的语言为（ ）。
A. C B. Basic C. DDL D. DML
2. 数据库设计的 4 个阶段是需求分析、概念设计、逻辑设计和（ ）。
A. 编码设计 B. 测试阶段 C. 运行阶段 D. 物理设计
3. 下列描述中错误的是（ ）。
A. 在数据库系统中，数据的物理结构必须与逻辑结构一致
B. 数据库技术的根本目标是要解决数据的共享问题
C. 数据库设计是指在自己有数据库管理系统的基础上建立数据库
D. 数据库系统需要操作系统的支持
4. 下列描述中正确的是（ ）。
A. 数据库系统是一个独立的系统，不需要操作系统的支持
B. 数据库技术的根本目标是要解决数据的共享问题
C. 数据库管理系统就是数据库系统
D. 以上 3 种说法都不对

4.5.2 填空题

1. 数据管理技术的发展经历了人工管理、文件系统和数据库管理系统 3 个阶段，其中数据独立性最高的是_____。
2. 数据的独立性分为逻辑独立性与物理独立性。当数据的存储结构改变时，其逻辑结构可以不变，因此，基于逻辑结构的应用程序不必修改称为_____。

第5章

Visual FoxPro 数据库基础

数据库是为满足日益发展的数据处理需要，在文件系统的基础上发展起来的一种先进的数据管理技术，是人们进行数据存储、共享和数据处理的有效工具，也是计算机科学领域发展最快、应用最广泛的核心技术之一。大多数的计算机应用系统都是在数据库的基础上开发的。因此，学习和掌握数据库的基础知识，熟悉数据库管理系统的特点，是开发数据库应用系统的必要前提和重要保证。

本章主要介绍与 Visual FoxPro 数据库管理系统相关的一些数据库基本概念、关系数据库、数据库设计的基础知识、Visual FoxPro 系统特点与工作方式，以及 Visual FoxPro 的基本数据元素等内容。

5.1 数据库基础知识

在学习数据库相关知识时，首先需要了解数据、数据库等基本概念。这些概念和术语贯穿数据处理的全过程，对于更好地学习和使用数据库管理系统有着重要的意义。只有在充分了解这些基本概念后，学习 Visual FoxPro 数据库管理系统操作和应用，才会更加得心应手。

5.1.1 计算机数据管理的发展

1. 数据与数据处理

1) 数据

数据是指存储在某一媒体上，能够被识别的物理符号。数据的形式可以是多种多样的，不仅有数字、字母、文字和其他特殊字符组成的文本形式的数据，还可以有图形、图像、声音和其他多媒体数据等非文本形式的数据。

数据的概念包括两个方面的内容：

- (1) 描述事物特性的数据内容。
- (2) 存储在某一媒体上的数据形式。

2) 数据处理

数据处理也叫做信息处理，是指将数据转换成信息的过程。数据处理的目的是从大量的、原始的数据中获得所需要的资料并提取有用的数据成分即信息，作为行为和决策的依据。

在计算机中，数据处理就是利用计算机对各种类型的数据进行采集、整理、存储、分类、排序、检索、维护、加工、统计和传输等一系列操作。

2. 计算机数据管理

数据处理的中心问题是数据管理。数据管理技术的发展与计算机硬件、软件的发展，以及用户的应用需求密不可分，随着计算机软件、硬件技术的进步和计算机应用范围的发展，计算机数据处理过程发生了划时代的变革，而数据库技术

的发展，使数据处理跨入了一个崭新的阶段。

数据管理技术的发展经历了人工管理、文件管理和数据库系统管理 3 个阶段。

1) 人工管理阶段

人工管理出现在计算机应用于数据管理的初期。在 20 世纪 50 年代中期以前，计算机主要用于科学计算。在硬件方面，外部存储器只有纸带、卡片、磁带，没有像磁盘这样可以随机访问和直接存取的外部存储设备；在软件方面，还没有操作系统，没有管理数据的软件。因此，用户在应用程序中不仅要有设计数据处理的方法、数据输入/输出的方式，还要阐明数据在存储器上的存储地址、存储结构。

在人工管理阶段，由于数据量少，加上计算机软件、硬件的限制，数据处理中的数据不需要、也不允许长期保存，程序运行结束后就得退出计算机系统；用户的应用程序与数据相互结合不可分割，当数据有所变动时，程序也得随之改变，独立性差；另外，一个程序中的数据无法被其他程序利用，即使两个应用程序都涉及某些共同的数据，也必须各自定义，无法共享，因此程序与程序之间存在大量的重复数据，称为数据冗余。这种管理方式既不灵活，也不安全，编程效率低下。

2) 文件管理阶段

从 20 世纪 50 年代后期到 60 年代中期，计算机的应用已经拓展到数据处理阶段，大量的数据存储、检索和维护成为紧迫的需求。这期间，硬件方面已经有了磁盘、磁鼓等直接存取的存储设备；软件方面，操作系统中的文件系统专门用于管理数据，文件系统不仅有批处理的处理方式，而且能够实现联机实时处理。

在文件管理阶段，文件系统可以把有关的数据组织成一个数据文件，可以长期保存在外部存储器上，可以对数据文件多次存取并实现对数据的查询、修改、插入和删除等操作；文件系统把数据按其内容、结构和用途组织成若干个独立的数据文件，实现了“按文件名访问，按记录进行存取”的数据管理技术；程序与数据分开存储，因此数据与程序之间具有一定的独立性；数据文件一般为某一用户或用户组所有，仅供指定的其他用户进行共享。

虽然文件管理比人工管理有了很大的进步，但是数据文件仍然高度依赖于其对应的特定程序，数据独立性较差；同一数据项可能重复出现在多个文件中，因此，冗余不可避免，浪费了大量的存储空间，增加了更新的开销，而且由于不能统一修改，还容易造成数据的不一致性；此外，文件管理不支持并发访问。

3) 数据库系统管理阶段

从 20 世纪 60 年代以来，计算机用于数据管理的任务更加繁重，应用也越来越广泛，数据量急剧增加，用户对数据共享的要求越来越强烈。文件系统的数据管理方法已经无法适应开发应用系统的需要。为了实现计算机对数据的统一管理，达到数据共享的目的，数据库技术便应运而生，出现了管理数据的专门软件，即数据库管理系统。

数据库系统管理是通过数据库管理系统软件对所用数据实行统一规划和管理，形成一个数据中心，构成一个数据的“仓库”，在数据库中的数据能够满足不同用户的要求，供不同的用户共享。在这一管理方式下，应用程序不再只与一个孤立的数据文件相对应，可以取整体数据集的某个子集作为逻辑文件与其对应，通过数据库管理系统实现逻辑文件与物理数据之间的协调互动，从而实现数据处理。在数据库系统管理的环境下，应用程序对数据管理和访问更加灵活方便，而且数据与应用程序之间完全独立，从而减少了应用程序开发和维护的代价，提高了程序的编制质量和效率。由于数据库文件中的各数据子集之间可以建立关联关系，因此数据的冗余大大减少，数据的共享性显著增强，提高了数据的一致性和完整性。

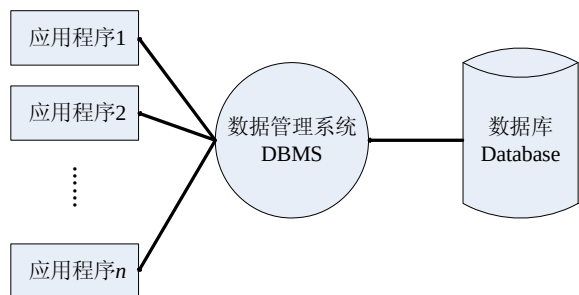


图 5-1 数据库系统中数据与程序的关系

在数据库系统管理阶段，程序与数据的关系如图 5-1 所示。

数据库技术自产生以来，经历了多个发展阶段。目前，数据库技术的应用已经深入到人们生活的各个领域，数据库技术也已经十分成熟，因此，数据库技术的研究已经开始转向新的应用领域所提出的新的要求。这期间，数据库技术最重要的发展是分布式数据库技术和面向对象的数据库技术的产生。

4) 分布式数据库系统

分布式数据库系统是数据库技术与计算机网络技术结合的产物。网络技术的发展为数据库提供了分布式的运行环境，从主

机-终端系统结构发展到客户/服务器（Client/Server）系统结构，进而发展到浏览器/服务器（Browser/Server）系统结构。

数据库技术与网络技术的结合分为紧密式结合与松散式结合两种，因此，分布式数据库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。

5) 面向对象数据库系统

面向对象是一种认识、描述事物的方法论。面向对象的程序设计是 20 世纪引入计算机科学技术领域的一种新的程序设计技术和范型。它的发展十分迅速，成为当前软件开发的主要方法。

面向对象数据库是数据库技术与面向对象程序设计相结合的产物，是面向对象方法在数据库领域的应用。

Visual FoxPro 不但支持标准的过程化程序设计，还提供了面向对象程序设计和网络开发的强大功能和更大的灵活性。

5.1.2 数据库系统

为了加深对数据库系统基础知识的理解，首先介绍几个重要的概念。

1. 数据库

数据库（Database，DB）是存储在计算机存储设备上、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，而且包括相关事物之间的联系。

数据库中的数据具有较高的独立性和较小的冗余性,可以被多个应用程序和用户共享。

2. 数据库管理系统

数据库管理系统 (Database Management System, DBMS), 是一个专门对数据库的建立、使用和维护进行管理的软件。目前, 世界上已经开发出许多数据库管理系统, Visual FoxPro 就是其中的一种。

3. 数据库系统

数据库系统 (Database System, DBS) 是指引进数据库技术后的计算机系统。由 5 个部分组成:

- (1) 硬件系统。
- (2) 数据库集合。
- (3) 数据库管理系统及相关软件。
- (4) 数据库管理员。
- (5) 用户。

从上面的定义可以发现：这些概念的范畴是不同的，它们之间具有包含和被包含的关系。数据库（DB）、数据库系统（DBS）和数据库管理系统（DBMS）之间的关系是：DBS 包括 DB 和 DBMS。

4. 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。

5.1.3 数据库系统的特点

在数据库系统中，数据库管理系统（DBMS）是数据库系统的核心，各层次软件之间的相互关系如图 5-2 所示。

一个数据库系统具有以下的主要特点。

1. 数据高度结构化

数据库系统通过特定的数据模型把整个组织内部的数据结构化,因此,数据库系统不仅可以表示事物内部各数据项之间的联系,而且可以表示事物与事物之间的联系,从而全面反映出现实世界事物之间的联系。

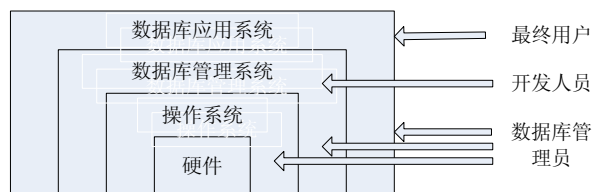


图 5-2 数据库系统层次示意图

2. 数据的独立性强

在数据库系统中，由于数据结构的定义和组织是单独进行的，与应用程序的编写几乎无关，因此实现了应用程序对数据的总体逻辑结构、物理存储结构之间有较高的独立性。改变数据结构并不一定要修改应用程序；当修改程序时，也不必修改数据结构。这就为程序的编写及数据的管理提供了极大的方便。

3. 数据的共享程度高

在数据库系统中，数据是面向系统的，而不是面向某个具体程序的，数据的定义和描述已经从应用程序中分离出来，通过数据库管理系统来统一管理，因此，具有较高的共享性，这也是数据库的重要特性之一。

4. 有统一的数据控制功能

数据库通过具有强大功能的数据库管理系统对数据进行统一控制，以保护数据的安全性和完整性，通过对数据的检验措施，保证了数据的正确性、有效性和相容性，通过并发控制，使多个用户可以同时共享数据库，而不至于互相干扰。

5.1.4 数据模型

数据库需要根据应用系统中数据的性质、内在联系，按照管理的要求来设计和组织。人们把客观存在的事物以数据的形式存储到计算机中，经历了对现实生活中事物特性的认识、概念化到计算机数据库里具体表示的逐级抽象过程。

1. 实体的描述

现实世界存在各种事物，这些事物之间存在着联系。这种联系是客观存在的，是由事物本身的性质所决定的。

(1) 实体。客观存在并且可以相互区别的事物称为实体。实体可以是具体的事物，也可以是抽象的事件，如教师、学生和课程等。

(2) 实体的属性。描述实体的特性称为属性。如学生的属性有学号、姓名、性别、专业、出生日期、简况、照片等。

(3) 实体集和实体型。属性值的集合表示一个实体，而属性的集合表示一种实体的类型，称为实体型。同类型实体的集合称为实体集。

在 Visual FoxPro 中，用“表”存放一类实体，如学生表、课程表等。一个表包含若干“字段”，即实体的属性。字段值的集合组成表中的一个记录，代表一个具体的实体。

2. 实体间的联系及联系的种类

实体之间的对应关系称为“联系”，它反映现实世界事物之间的相互关联。实体之间联系的种类是指一个实体型中可能出现的每一个实体和另一个实体型中多少个具体实体存在联系。

实体间的联系有 3 种类型：

(1) 一对一联系 (1:1)。如果实体集 A 中的每一个实体只与实体集 B 中的一个实体相联系，反之亦然，则说明这种关系是一对一联系。

例如，在大学中，学院和院长是两个实体型，如果一个学院只有一个院长，一个院长不能同时在其他学院中再兼任院长，则学院和院长之间存在一对一联系。

(2) 一对多联系 (1:n)。如果实体集 A 中的每一个实体，在实体集 B 中都有多个实体与之对应；实体集 B 中的每一个，在实体集 A 中只有一个实体与之对应，则称实体集 A 与实体集 B 是一对多联系。

例如，学院和学生是两个实体型，每个学院都有多名学生，但是每名学生却只能属于一个学院，因此，学院和学生的联系就是一对多联系。

(3) 多对多联系 (m:n)。如果实体集 A 中的每一个实体，在实体集 B 中都有多个实体与之对应，反之亦然，则称这种关系是多对多联系。

例如，产品和订单是两个实体型，每个订单中都有多个产品，同样，每个产品也可以出现在多个订单中，因此，产品和订单之间的联系就是多对多联系。

3. 数据模型

数据模型是数据库管理系统中用来表示实体及实体间联系的方法，也是描述数据间相互联系的结构形式。一个具体的数据模型应该正确地反映出数据之间存在的整体逻辑关系。

数据模型在数据库系统的整体结构中占有重要的地位，任何一个数据库管理系统都是基于某种数据模型的。数据库设计的一个核心问题，就是设计一个好的数据模型。

在数据库的发展历程中，产生了 3 种数据模型：层次模型、网状模型和关系模型。其中，关系模型出现最晚，却应用得最广泛。而使用支持某种特定数据模型的数据库管理系统也相应地被称为层次数据库管理系统、网状数据库管理系统、关系数据库管理系统。

1) 层次数据模型

数据的层次模型是由若干基本层次联系组成的以记录型为结点的有向树。所谓基本层次联系，是指两个记录型之间的某种相关关系。它们之间用有向线段连接。处于上层的称为“双亲”，下层称为“子女”，无子女的记录称为“叶子”。

层次模型有两个基本特点：

- (1) 有且仅有一个记录型无双亲，这个记录型称为树的根。
- (2) 其他记录型有且仅有一个双亲。

2) 网状数据模型

网状模型是以记录型为结点的网状结构。它与层次模型有两点非常显著的区别：

- (1) 一个子女可以有多个双亲。
- (2) 两个记录之间可以有多种联系。

网状模型更适于表达客观事物复杂的隶属关系和横向关系，但其结构也更加复杂，记录的存取路径也往往不是唯一的。从网状模型与层次模型的相互关系看，后者仅是前者的一种特殊情况。

3) 关系数据模型

用二维表表示实体及实体之间联系的模型称为关系模型。关系模型是以关系数学理论为基础的。在关系模型中，操作的对象和结果都是二维表，即关系。

关系模型是数学领域的专家、学者总结了层次模型和网状模型设计和使用中的经验教训，并借助于近代数学工具而提出来的。它有效地、较为圆满地解决了过去出现的种种问题，提出了一整套定义、概念、公理、定理、推论及各种实用算法。巧妙地把抽象的数学理论和具体的实际问题结合起来，理论上十分严密并非常实用。它不仅对数据库领域的发展起到巨大的推动作用，而且对整个计算机领域的发展也有很大影响。

5.2 关系数据库

自 20 世纪 80 年代以来，新推出的数据库管理系统几乎都支持关系模型，Visual FoxPro 就是一种关系数据库管理系统。

5.2.1 关系模型

关系模型的用户界面非常简单，一个关系的逻辑结构就是一张二维表。这种用二维表的形式来表示实体和实体之间联系的数据模型称为关系数据模型。

1. 关系术语

在 Visual FoxPro 中，一个“表”就是一个关系。

(1) 关系。在关系数据库中，一个关系就是一张二维表。每个关系有一个关系名，在 Visual FoxPro 中，一个关系存储为一个文件，文件扩展名是“.dbf”，称为“表”。如图 5-3 所示，就是一个“产品”关系，在 Visual FoxPro 中的名字为“产品.dbf”。

产品								
产品id	产品名称	供应商id	类别id	单位数量	单价	库存量	订购量	再订购量
1.00000	苹果汁	1.00000	1.00000	每箱24瓶	18.00000	39.00000	0.00000	10.00000
2.00000	牛奶	1.00000	1.00000	每箱24瓶	19.00000	17.00000	40.00000	25.00000
3.00000	番茄酱	1.00000	2.00000	每箱12瓶	10.00000	13.00000	70.00000	25.00000
4.00000	盐	2.00000	2.00000	每箱12瓶	22.00000	53.00000	0.00000	0.00000
5.00000	麻油	2.00000	2.00000	每箱12瓶	21.35000	0.00000	0.00000	0.00000
6.00000	酱油	3.00000	2.00000	每箱12瓶	25.00000	120.00000	0.00000	25.00000
7.00000	海鲜粉	3.00000	7.00000	每箱30盒	30.00000	15.00000	0.00000	10.00000
8.00000	胡椒粉	3.00000	2.00000	每箱30盒	40.00000	6.00000	0.00000	0.00000
9.00000	鸡	4.00000	6.00000	每袋500克	97.00000	29.00000	0.00000	0.00000
10.00000	蟹	4.00000	8.00000	每袋500克	31.00000	31.00000	0.00000	0.00000
11.00000	民众奶酪	5.00000	4.00000	每袋6包	21.00000	22.00000	30.00000	30.00000
12.00000	德国奶酪	5.00000	4.00000	每箱12瓶	38.00000	86.00000	0.00000	0.00000
13.00000	龙虾	6.00000	8.00000	每袋500克	6.00000	24.00000	0.00000	5.00000
14.00000	沙茶	6.00000	7.00000	每箱12瓶	23.25000	35.00000	0.00000	0.00000
15.00000	味精	6.00000	2.00000	每箱30盒	15.50000	39.00000	0.00000	5.00000
16.00000	饼干	7.00000	3.00000	每箱30盒	17.45000	29.00000	0.00000	10.00000
17.00000	猪肉	7.00000	6.00000	每袋500克	39.00000	0.00000	0.00000	0.00000
18.00000	墨鱼	8.00000	8.00000	每袋500克	62.50000	42.00000	0.00000	0.00000
19.00000	糖果	8.00000	3.00000	每箱30盒	9.20000	25.00000	0.00000	5.00000
20.00000	桂花糕	8.00000	3.00000	每箱30盒	81.00000	40.00000	0.00000	0.00000
21.00000	花生	8.00000	3.00000	每箱30包	10.00000	3.00000	40.00000	5.00000
22.00000	糯米	9.00000	5.00000	每袋3公斤	21.00000	104.00000	0.00000	25.00000
23.00000	燕麦	9.00000	5.00000	每袋3公斤	9.00000	61.00000	0.00000	25.00000
24.00000	汽水	10.00000	1.00000	每箱12瓶	4.50000	20.00000	0.00000	0.00000

图 5-3 “产品”关系

对关系的描述称为关系模式，一个关系模式对应一个关系的结构，其格式为：关系名（属性 1，属性 2，…，属性 *n*）在 Visual FoxPro 中，表示为表结构，其格式为：表名（字段 1，字段 2，…，字段 *n*）

(2) 元组（记录）。在关系表中，一行称为一个元组，也称为一条记录。一个关系有多条记录，即多个元组。

(3) 属性（字段）。二维表中垂直方向的列称为属性，也叫字段，每一列有一个属性名（字段名）。每个字段的数据类型、宽度等在创建表的结构时规定。例如图 5-3 中的产品名称、单价、库存量等。

(4) 域。是指属性（字段）的取值范围。不同类型的字段取值范围不同，例如，产品名称的取值范围是字符，单价的取值范围是数字等。

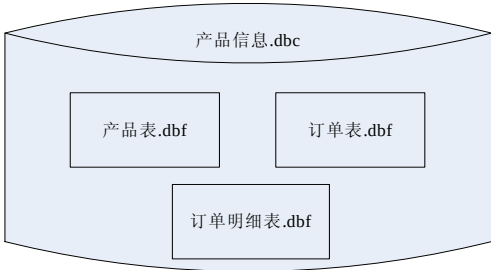


图 5-4 数据库示意图

(5) 关键字。关键字是属性或属性的组合，其值能够唯一地标识一个元组（记录）。在 Visual FoxPro 中，关键字是字段或字段的组合。例如图 5-3 中的产品 id，可以作为标识一条记录的主关键字。

(6) 外部关键字。如果表中的字段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，则称为外部关键字。外部关键字可以与主关键字同名，也可以不同名。

在 Visual FoxPro 中，把相互联系的若干表存放在一个数据库中统一进行管理，即一个数据库中 can 包含许多表，如图 5-4 所示。数据库文件的类型是 “.dbc”。

2. 关系特点

- 关系模型中的表，必须具备以下特点：
- (1) 关系必须规范化。所谓规范化是指关系模型中的每一个关系模式都必须满足一定的要求。关系的规范条件很多，但最基本的条件是，关系的每个属性（字段）必须是不可再细分的数据项，也就是说，表中不允许再包含表。
 - (2) 在一个关系中，不能出现相同的属性名。
 - (3) 在一个关系中，不能有完全相同的元组，即不能存在冗余。
 - (4) 在一个关系中，元组的次序无关紧要，即可以任意交换两行的次序。
 - (5) 在一个关系中，属性的次序无关紧要，即可以任意交换两列的次序。

3. 关系模型的组成

关系模型通过一系列的关系模式来表述数据的结构和属性，它一般有 3 个组成部分：数据结构、数据操作和完整性规则。

- (1) 数据结构。数据库中所有数据及其相互联系都被组织成关系（即二维表）的形式。
- (2) 数据操作。提供一组完备的关系运算，以支持对数据库的各种操作。
- (3) 完整性规则。包括域完整性规则、实体完整性规则、参照完整性规则和用户定义的完整性规则等。

数据完整性是指数据库中数据的准确性、正确性和有效性。数据库中的数据完整性是用户对数据存储和维护的一种需求，它可以指定某些属性或者字段的取值必须限制在一定的范围之内，也可以指定某些数据之间必须满足一定的约束条件。数据完整性根据它所要求的内容可以分成不同的种类。

域完整性规定了属性的取值范围。

实体完整性则要求任意元组的主关键字的值不得为空值，并且必须在所属的关系中唯一。

参照完整性则要求当一个元组的外部关键字的值不为空值时，以该外部关键字的值作为主关键字的值的元组必须在相应的关系中存在。

4. 实际关系模型

一个具体的关系模型由若干个关系模式组成。在 Visual FoxPro 中，一个数据库包含相互联系的多个表，它们之间的联系是通过公共字段（即名称相同的字段）建立的。

如图 5-5 中所示，数据库“产品信息.dbc”中包含有 3 个表：“产品”表、“订单”表和“订单明细”表。其中，“产品”表和“订单”表之间是多对多的联系，而“订单明细”表则把这个多对多的联系分解成两个一对多的联系。“产品”表与“订单明细”表通过共同字段“产品 id”建立联系，“订单”表和“订单明细”表通过共同字段“订单 id”建立联系。

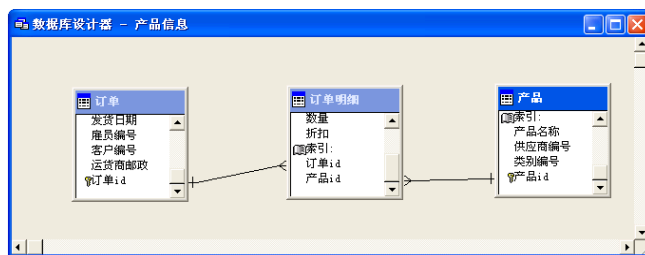


图 5-5 一个关系模型的实例

5.2.2 关系运算

关系的基本运算有两类：一类是传统的集合运算（并、交、差等），另一类是专门的关系运算（选择、投影、连接）。

1. 传统的关系运算

进行并、交、差集合运算的两个关系必须具有相同的结构模式，即相同结构。

- (1) 并。两个相同结构关系的“并”是由属于这两个关系的全部元组组成的集合。
- (2) 交。两个相同结构关系的“交”是由属于这两个关系所共有的元组组成的集合。
- (3) 差。两个相同结构关系的差是由属于前一个关系的元组而不属于后一个关系的元组组成的集合。例如，关系 R 和 S 的差结果是由属于 R 但不属于 S 的元组组成的集合。

如图 5-6 所示为传统关系运算的实例。

R			S			$R \cup S$			$R \cap S$			$R - S$		
A	B	C	A	B	C	A	B	C	A	B	C	A	B	C
a1	b1	c1	a1	b1	c1	a1	b1	c1	a1	b1	c1	a1	b1	c1
a1	b2	c1	a2	b1	c2	a1	b2	c1	a1	b1	c1	a1	b2	c1
a2	b1	c2	a2	b2	c2	a2	b1	c2	a2	b1	c2			

图 5-6 传统关系运算举例

2. 专门的关系运算

- (1) 选择。从关系中找出满足给定条件的元组的操作称为选择，它是原关系的一个子集。选择运算记为 $\sigma_{F(R)}$ ，其中 R 表示一个关系， F 是选择条件， σ 表示选择运算符。

例如，条件 F 为 A=a1，用图 5-6 中的图（a）做选择运算，其结果如图 5-7（a）所示。条件 F 为 B=b2，用图 5-6 中的图（a）做选择运算，其结果如图 5-7（b）所示。

（2）投影。从关系模式中指定若干个属性组成新的关系称为投影。如果新关系中包含重复元组，则去掉重复元组。投影运算记为 $\Pi_x(R)$ ，其中 R 为一个关系， x 为一组属性名或属性序号。

例如，若对图 5-6 中的图（a）做投影运算 $\Pi_{AB}(R)$ 、 $\Pi_{BC}(R)$ ，其结果如图 5-8 所示。

$\sigma_{A=a1}(R)$			$\sigma_{B=b1}(R)$		
A	B	C	A	B	C
a1	b1	c1	a1	b2	c1
a1	b2	c1			

(a) (b)

$\Pi_{AB}(R)$		$\Pi_{BC}(R)$	
A	B	B	C
a1	b1	b1	c1
a1	b2	b2	c1
a2	b1	b1	c2

(a) (b)

图 5-7 选择运算

图 5-8 投影运算

（3）连接。关系的横向结合称为连接，是将两个关系模式拼接成一个更宽的关系模式。连接也称为 θ 连接。它是从两个关系的笛卡儿积中选取属性间满足一定条件的元组。连接记为 $R \bowtie_{A \theta B} S$ ，其中 R 为一个关系， S 为一个关系， A 和 B 分别为 R 和 S 上元数（属性个数）相等且可比的属性组， θ 是比较运算符。连接运算从 R 和 S 的笛卡儿积 $R \times S$ 中选取（ R 关系）在 A 属性组上的值与（ S 关系）在 B 属性组上的值满足比较关系 θ 的元组。

连接运算中有两种最常用的连接，一种是等值连接，另一种是自然连接。其中，以字段值对应相等为条件进行的连接，即 θ 为 “=” 的连接称为等值连接。

（4）自然连接。自然连接是一种特殊的等值连接，是去掉重复属性的等值连接。它要求 R 与 S 中具有相同的属性组，并且在结果中把重复的属性列去掉。自然连接记为 $R \bowtie S$ 。

例如，设图 5-9（a）和图 5-9（b）分别为关系 R 和关系 S ，则图 5-9（c）为 $R \bowtie_{C < D} S$ 的结果，图 5-9（d）为等值连接 $R \bowtie_{R.B=S.B} S$ 的结果，图 5-9（e）为自然连接的结果。

R			S	
A	B	C	B	D
a1	b1	3	b1	3
a1	b2	4	b1	7
a2	b1	5	b2	6

(a) (b)

$R \bowtie_{C < D} S$				
A	B	C	B	D
a1	b1	3	b1	7
a1	b1	3	b2	6
a1	b2	4	b1	7
a1	b2	4	b2	6
a2	b1	5	b1	7
a2	b1	5	b2	6

(c)

$R \bowtie_{R.B=S.B} S$				
A	B	C	B	D
a1	b1	3	b1	3
a1	b1	3	b1	7
a1	b2	4	b2	6
a2	b1	5	b1	3
a2	b1	5	b1	7

(d)

$R \bowtie S$			
A	B	C	D
a1	b1	3	3
a1	b1	3	7
a1	b2	4	6
a2	b1	5	3
a2	b1	5	7

(e)

图 5-9 连接运算

5.3 数据库设计基础

数据库设计是数据应用的核心，是数据库技术的主要内容之一。数据库设计是指对于一个给定的应用环境，构造最优的数据库模式，建立数据库及其应用系统，有效存储数据，满足用户信息要求和处理要求。数据库设计的根本目标是解决数据共享问题。

5.3.1 数据库设计的原则

在进行数据库设计时，需要遵循以下主要原则。

(1) 关系数据库的设计应遵从概念单一化“一事一地”的原则。所谓概念单一化“一事一地”的原则是指：让一个关系描述一个概念、一个实体或者实体之间的一种联系。若一个关系中描述的概念、实体、联系多于一个，就应该对其进行“分离”，即用多个关系进行描述。换言之，数据库中应避免出现大而杂的关系，应将不同的信息分散在不同的关系中。这样做的好处在于可以使数据的组织和维护更简单，避免出现大量的数据冗余，同时也易保证用户访问数据库时具有较高的效率。

(2) 避免在表之间出现重复字段。关系之间的联系依靠外部关键字来实现，除了外部关键字之外，应尽量避免在关系中出现重复属性。其目的是使数据冗余尽量小，同时防止在对数据库操作时造成数据的不一致性。

(3) 表中的字段必须是原始数据和基本数据元素。表中不应该包括通过计算可以得到的属性或多项数据的组合。

(4) 用外部关键字保证有关联的表之间的联系。表之间的关联依靠外部关键字来维系，使得表具有合理的结构，不仅存储了所需要的实体信息，而且能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关系数据模型。

5.3.2 数据库设计的步骤

可以按照以下步骤来设计数据库。

(1) 需求分析。了解管理需求，确定数据库应保存哪些信息。用户需求主要包括以下3个方面：

信息需求：用户需要从数据库中得到信息内容。信息需求定义了数据库应用系统应该提供的所有信息，应注意表述清楚系统中数据的数据类型。例如，需要查询哪些信息？打印什么报表？

处理需求：对数据需要完成的处理功能及方式。处理需求定义了系统数据处理的操作，应注意操作执行的场合、频率、操作对数据的影响等。

安全性及完整性要求：在定义信息需求和处理需求的同时必须注意安全性、完整性的约束。

在进行需求分析时应该与用户充分交流，细致耐心地了解业务处理流程，尽可能收集所有基础资料，如报表、合同、档案、原始单据等。

(2) 确定需求表。根据系统存在的实体及它们之间的联系，确定需要建立哪些表。一般每个实体都要建立一个表，存储它的相关信息。实体之间的联系也要建立一个表。

(3) 确定需求字段。确定每个表中需要保存哪些字段，字段应包括实体的主要属性。确定字段时要同时确定字段的4个要素：字段名、数据类型、宽度，如果是数值型字段，还要确定小数位数。

关系数据库管理系统能够迅速查找存储在多个独立表中的数据并组合这些信息。要求每个表都必须有一个或一组字段可用以唯一确定存储在表中的每个记录，即主关键字。因此需要确定每个表的主关键字。

(4) 确定联系。对每个表进行分析，并确定一个表中的数据和其他表中的数据有何联系。必要时，可以在表中加入一个字段或通过创建一个新表来建立联系。一般是通过主关键字和外部关键字建立联系的。

(5) 设计求精。对设计进一步分析，找出其中的错误并予以改正；创建表，并输入一些示例数据，看是否可以得到预想的结果，必要时还应进行修改。

数据库设计在每一个阶段的后期都要经过用户的确认。如果不能满足用户的要求，则必须返回前面一个或几个阶段进行修改或调整，直到满足用户要求为止。

5.3.3 Visual FoxPro 应用系统开发的基本步骤

学习 Visual FoxPro 的根本目的，是利用它开发各种具体的数据库应用系统。根据结构化的系统开发思想和方法，一个数据库应用系统的开发步骤如下：

- (1) 系统需求分析：确定开发项目，进行实地调查，明确用户需求。
 - (2) 数据库设计：根据用户需求进行数据库设计。
 - (3) 系统功能分析与设计：用结构化系统分析和设计方法，进行系统功能分析和设计。
- 以上 3 个步骤属于系统分析和设计阶段。下面的步骤属于系统实施阶段。
- (4) 创建项目和数据库：按照设计方案创建项目和数据库、表的结构，同时输入若干用于程序调试的数据。
 - (5) 报表设计：按照用户的需要设计打印报表和标签。
 - (6) 编写主程序和建立系统菜单：在学习和掌握程序设计思想和方法的基础上设计应用系统的主程序和系统菜单，搭建系统的框架。
 - (7) 设计系统的主要操作界面——表单：运用面向对象的程序设计方法，设计应用系统的操作界面——表单。这是系统程序设计中工作量最大的部分。
 - (8) 连编项目程序和创建系统安装盘：将以上步骤完成的所有文件进行编译连接，形成系统的可执行文件“.exe”文件，然后利用安装向导制作系统的安装盘。

5.4 Visual FoxPro 系统概述

Visual FoxPro 6.0 是 Microsoft 公司 1998 年发布的可视化编程语言集成包 Visual Studio 6.0 中的一员，是一种用于数据库结构设计和应用程序开发的数据库管理系统。Visual FoxPro 6.0 与其他先前版本相比功能更加强大，它采用了可视化的操作界面、面向对象的程序设计方法，极大地简化了数据库应用系统的开发过程，并且使用 Rushmore 查询优化技术提高了系统性能。

无论是组织信息、运行查询、创建集成的关系型数据库系统，还是为最终用户建立功能全面的数据管理应用系统，Visual FoxPro 6.0 都可以提供管理数据所需的工具。其具有以下新的特性：

- (1) 增强项目与数据库管理。
- (2) 提高应用程序开发的生产率。
- (3) 互操作性和支持 Internet。
- (4) 充分利用已有数据。

5.4.1 Visual FoxPro 6.0 的安装与启动

Visual FoxPro 6.0 的功能很强大，但是它运行的环境对系统的要求并不高。其相关的基本配置要求如表 5-1 所示。

表 5-1 Visual FoxPro 6.0 软硬件基本配置要求

计算机的软硬件	基本配置有求
处理器	486 DX/66 MHz 处理器，推荐使用 Pentium 或更高处理器的 PC
内存	16MB 以上的内存，推荐使用 24MB 内存
硬盘空间	典型安装需要 85MB 硬盘空间，最大安装需要 90MB 硬盘空间
其他硬件	一个鼠标、一个光盘驱动器，推荐使用 VGA 或更高分辨率的监视器
操作系统	需要在 Windows 95/98（中文版）、Windows NT 4.0（中文版）或更高版本的操作系统上运行

1. 安装 Visual FoxPro 6.0
- 安装步骤如下：

(1) 将 Visual FoxPro 6.0 安装盘插入电脑光驱。

(2) 通过“资源管理器”或“我的电脑”打开光盘，运行安装  文件后，出现安装向导界面，如图 5-10 所示。

(3) 按照安装向导提示，单击“下一步”按钮进行安装，在“最终用户许可协议”界面选择“接受协议”复选框之后，单击“下一步”按钮，如图 5-11 所示。



图 5-10 Visual FoxPro 6.0 安装向导



图 5-11 Visual FoxPro 6.0 安装向导——“最终用户许可协议”界面

(4) 在“产品号和用户 ID”界面输入产品的 ID 号和用户信息后，单击“下一步”按钮，如图 5-12 所示。

(5) 为 Visual FoxPro 6.0 应用程序所公用的文件选择具体的安装位置，公用文件一般应该存放在名为“Common”的文件夹中，如果需要用户也可以创建一个新的文件夹，单击“下一步”按钮，开始安装，如图 5-13 所示。



图 5-12 Visual FoxPro 6.0 安装向导——“产品号和用户 ID”界面



图 5-13 Visual FoxPro 6.0 安装向导——“选择公用安装文件夹”界面

(6) 在开始安装 Visual FoxPro 6.0 时，可以选择“自定义安装”或“典型安装”，“自定义安装”可以选取用户需要的文件进行安装，“典型安装”则由系统自动进行安装。当安装完成后，系统会自动提示“安装结束”。

2. 安装示例和联机文档

Visual FoxPro 6.0 的示例和联机文档需要使用“MSDN 安装向导”进行安装，如图 5-14 所示。

3. 启动 Visual FoxPro 6.0

单击 Windows 的“开始”按钮，选择“程序”→“Microsoft Visual FoxPro 6.0”→“Microsoft Visual FoxPro 6.0”命令。在第一次启动 Visual FoxPro 6.0 时，会弹出一个 Visual FoxPro 6.0 欢迎界面，如图 5-15 所示。



图 5-14 Visual FoxPro 6.0 MSDN 安装向导



图 5-15 Visual FoxPro 6.0 欢迎界面

在 Visual FoxPro 6.0 欢迎界面中，可以选择打开或创建一个新的项目，在如图 5-16 所示的“创建”对话框中，输入所创建项目名称后单击“保存”按钮，则创建了一个新的项目文件，同时打开项目管理器，如图 5-17 所示。

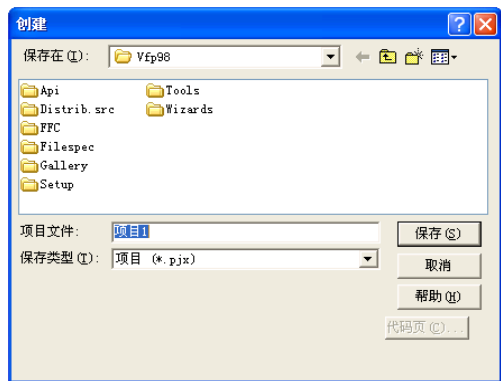


图 5-16 “创建”对话框

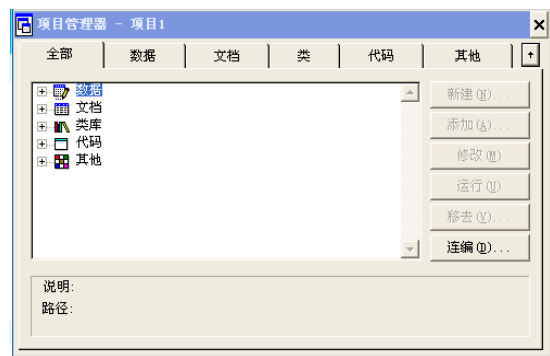


图 5-17 项目管理器

如果在下次启动 Visual FoxPro 6.0 时，不希望再次出现 Visual FoxPro 6.0 欢迎界面，则需要单击“关闭此屏”按钮前选中“以后不再显示此屏”复选框即可。

4. 退出 Visual FoxPro 6.0
- 有 4 种方法可以退出 Visual FoxPro 6.0 返回 Windows，用户可以根据自己的习惯，任选其中一种方法。
- (1) 单击 Visual FoxPro 6.0 标题栏最右面的关闭窗口按钮。
 - (2) 从“文件”下拉菜单中选择“退出”命令。
 - (3) 单击主窗口左上方的狐狸图标，从窗口下拉菜单中选择“关闭”命令，或者按【ALT+F4】组合键。
 - (4) 在命令窗口中输入 QUIT 命令，按【Enter】键。

5.4.2 Visual FoxPro 6.0 的主界面

Visual FoxPro 的主界面即其工作环境，启动 Visual FoxPro 后，Visual FoxPro 6.0 的主界面如图 5-18 所示。

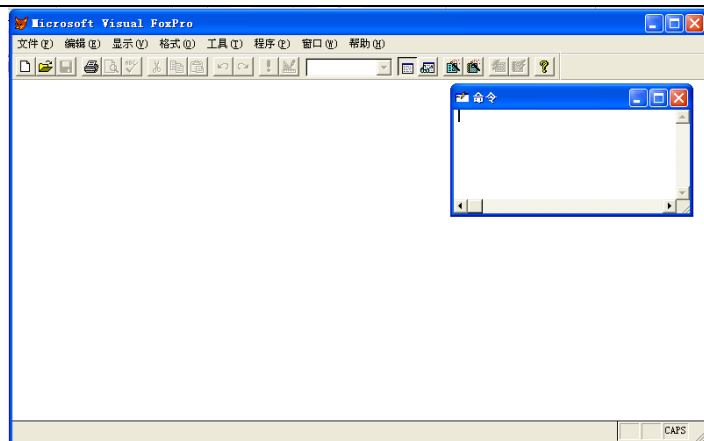


图 5-18 Visual FoxPro 6.0 的主界面

Visual FoxPro 6.0 有 3 种工作方式：利用菜单系统或工具栏按钮执行命令；在命令窗口直接输入命令进行交互式操作；利用各种生成器自动产生程序，或者编写 FoxPro 程序（命令文件），然后执行它。

前两种方式属于交互式工作方式，可以通过这两种方式得到同一结果。执行命令文件为自动化工作方式。其中，菜单工作方式为用户提供了更加便利的操作手段。

1. 菜单操作

Visual FoxPro 6.0 的主菜单包括“文件”、“编辑”、“显示”、“格式”、“工具”、“程序”、“窗口”和“帮助”8 个主菜单项，如图 5-19 所示，均为下拉式菜单，绝大多数操作都可以通过菜单方式实现。

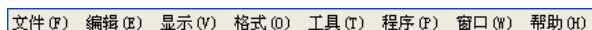


图 5-19 Visual FoxPro 6.0 的主菜单

Visual FoxPro 6.0 的菜单操作有 3 种方法：

- (1) 鼠标操作。利用鼠标左键单击菜单条相应的项目，系统会弹出该项目的下拉菜单，在下拉菜单中选择相应的命令，则可激发与之对应的操作。
- (2) 键盘操作。所有菜单项目的名字后面都有一个带有下画线的字母，该字母就是这个菜单项目的“热键”，按住【Alt】键和相应热键，便可以打开相应项目的下拉菜单。
- (3) 光标操作。在选择子菜单时，按光标键可以将光标移动到所需菜单上，按【Enter】键可激发与之对应的操作。

2. 命令操作

在命令窗口中可以直接输入 Visual FoxPro 的各种命令，按【Enter】键便可执行该命令。命令窗口如图 5-20 所示。

3. 程序执行操作

程序执行操作就是事先编写一个扩展名为.prg 的程序文件，在程序文件中将所要执行的操作按照某种正确的逻辑结构详细地写出来，然后执行该程序文件，便可执行这些操作了。

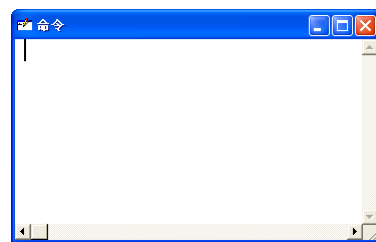


图 5-20 命令窗口

5.4.3 Visual FoxPro 6.0 工具栏的使用

设置工具栏是微软公司流行软件的共同特点，利用各种工具栏比菜单要方便得多，有效地利用工具栏，可以大大提高软件的使用率。工具栏位于菜单栏下面，其默认界面仅包括“常用”工具栏和“表单设计器”工具栏，如图 5-21 所示。

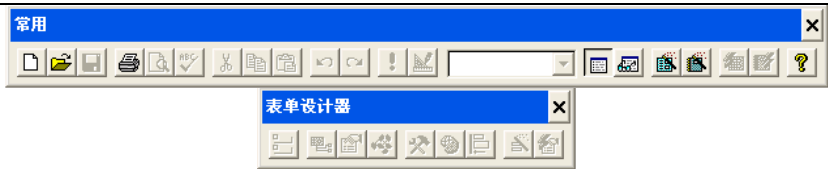


图 5-21 “常用”工具栏和“表单设计器”工具栏

除了“常用”工具栏之外，Visual FoxPro 还提供其他 10 个工具栏，如表 5-2 所示。

表 5-2 Visual FoxPro 6.0 的工具栏

工具栏名称	工具栏名称
报表控件	查询设计器
报表设计器	打印预览
表单控件	调色板
表单设计器	视图设计器
布局	数据库设计器

可以利用鼠标将工具栏拖曳到任何需要的位置，可以随时打开和关闭工具栏，可以重新设置工具栏中的工具，可以定制新的工具栏。所有的工具栏按钮都具有文本提示功能，当把鼠标指针停留在某个按钮上时，系统会用文字的形式显示它的功能。

1. 显示或隐藏工具栏

工具栏中的工具，只有在工具栏打开时才能使用。当某一工具栏打开后，选择其中的某一个按钮，便可以实现对应的操作。

在 Visual FoxPro 中，可以通过以下 3 种方式显示或隐藏工具栏。

- (1) 自动打开。工具栏会随着某一类文件的打开而自动打开。
- (2) 鼠标方式。在任何一个工具栏的空白处，单击鼠标右键，弹出“工具栏”的快捷菜单，从中选择打开或关闭工具栏，如图 5-22 所示。
- (3) 菜单方式。在主菜单“显示”菜单中，选择“工具栏”命令，弹出“工具栏”对话框，从中选择打开或关闭的工具栏，然后单击“确定”按钮，如图 5-23 所示。

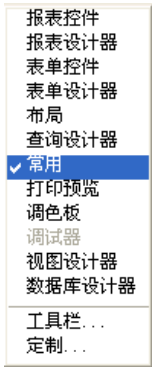


图 5-22 “工具栏”快捷菜单

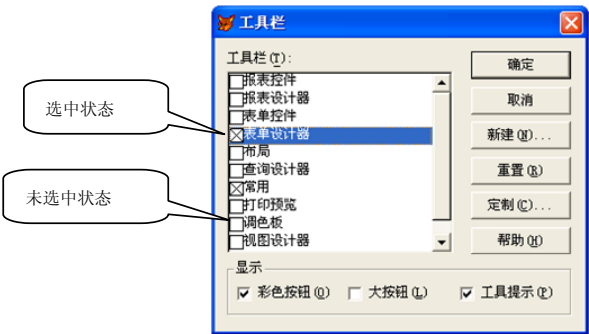


图 5-23 “工具栏”对话框

2. 定制工具栏

在 Visual FoxPro 中，系统提供的工具栏上的按钮，虽然不能删除或添加，但是可以根据工作任务的不同，重新组合这些按钮，建立一个适合具体问题需求的定制化的工具栏。

定制工具栏的步骤如下：

(1) 在“工具栏”对话框中，单击“新建”按钮，弹出“新工具栏”对话框，如图 5-24 所示。

(2) 在“新工具栏”对话框中，输入新工具栏名称，单击“确定”按钮，弹出“定制工具栏”对话框，如图 5-25 所示。



图 5-24 “新工具栏”对话框

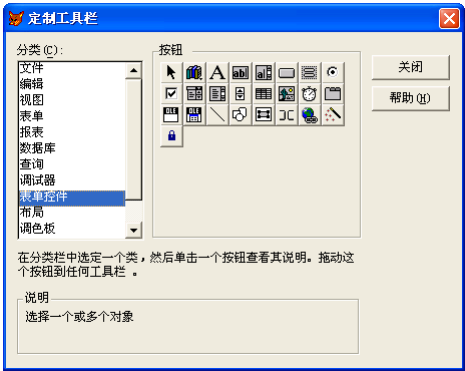


图 5-25 “定制工具栏”对话框

(3) 在“定制工具栏”对话框中，选择工具栏种类，再把所需要的按钮拖到新建的工具栏上，定制完成后，单击“确定”按钮，定制结果如图 5-26 所示。

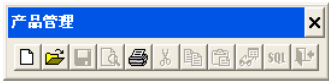


图 5-26 定制的“产品管理”工具栏

3. 修改现有工具栏

修改现有工具栏的步骤如下：

(1) 在“工具栏”对话框中，选择需要修改的工具栏，单击“定制”按钮，弹出“定制工具栏”对话框。

(2) 选择所需要的按钮拖到要修改的工具栏上，或者将要修改的工具栏上的按钮拖曳到工具栏之外，从而增加或删除工具栏上的按钮。

(3) 修改完毕，单击“确定”按钮，退出“定制工具栏”对话框。

4. 重置或删除工具栏

在“工具栏”对话框中，选择系统定义的工具栏时，单击“重置”按钮，可以将用户定制过的工具栏恢复到系统默认状态。

在“工具栏”对话框中，选择用户创建的工具栏时，单击“删除”按钮，可以将用户创建的工具栏删除。

5.4.4 Visual FoxPro 6.0 的配置

在 Visual FoxPro 中，为了使系统能满足个性化的要求，可以通过修改“环境参数”定制个性化的系统环境。

利用“选项”对话框可以方便地完成 Visual FoxPro 工作环境的配置。

1. 使用“选项”对话框

在主菜单“工具”菜单中，选择“选项”命令，弹出“选项”对话框，如图 5-27 所示。

在“选项”对话框中，包括 12 张选项卡，每张选项卡的功能如表 5-3 所示。

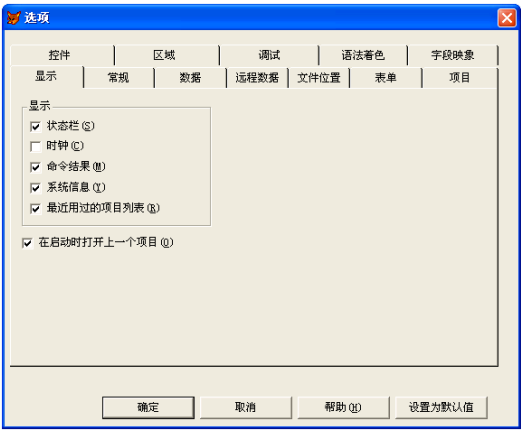


图 5-27 “选项”对话框

表 5-3 “选项”对话框中选项卡及功能

选项卡	功能
显示	界面显示选项，如是否显示标题栏、时钟、命令结果或系统信息等
常规	数据输入与编程选项，如设置警告声音、是否自动填充新记录等
数据	表选项，如是否以独占方式打开、是否忽视已删除记录
远程数据	远程数据访问选项，如共享连接、是否自动事务处理、连接超时设定
文件位置	改变系统默认文件存储位置
表单	表单设计器选项，如是否提示在运行表单之前保存修改、设置网格线等
项目	项目管理器选项，如项目双击操作、是否使用向导等
控件	“表单控件”工具栏中“查看类”按钮所提供的可视类库和 ActiveX 控件选项
区域	时间、日期、货币及数字的格式
调试	调试器显示和跟踪选项
语法着色	确定区分程序元素所用的字体和颜色
字段映像	确定从数据环境设计器、数据库设计器或项目管理器中向表单拖动表或字段时创建何种字段

2. 保存设置
- 在 Visual FoxPro 中，系统工作环境的配置可以是临时的，也可以是永久的。
- (1) 将设置保存为仅在本次系统运行期间有效。在“选项”对话框中完成各项配置后，单击“确定”按钮，退出“选项”对话框。此时所做出的各项配置修改仅在本次系统运行期间有效，再次启动系统，所做的配置修改将丢失。
- (2) 将设置保存为默认设置。在“选项”对话框中完成各项配置后，单击“设置为默认值”按钮，退出“选项”对话框。此时所做出的各项配置修改将会被永久保存。

5.4.5 Visual FoxPro 6.0 的主要文件类型

在 Visual FoxPro 中，文件单独存储在计算机磁盘中，文件扩展名代表了该文件的类型。文件扩展名和文件类型的对应关系如表 5-4 所示。

表 5-4 Visual FoxPro 常用文件扩展名及类型说明

扩展名	文件类型	扩展名	文件类型	扩展名	文件类型
ACT	向导操作图的文档	FRX	报表	PRG	程序
APP	生成的应用程序	FXP	编译后的程序	QPR	生成的查询程序
CDX	复合索引	HLP	图形方式帮助	QPX	编译后的查询程序
DBC	数据库	IDX	索引，压缩索引	SCT	表单备注
DBF	表	LBT	标签备注	SCX	表单
DCT	数据库备注	LBX	标签	FRT	报表备注
DCX	数据库索引	MEM	内存变量保存	TBK	备注备份
DLL	Windows 动态链接库	MNT	菜单备注	TXT	文本
ERR	编译错误	MNX	菜单	VCT	可视类备注
EXE	可执行程序	MPR	生成的菜单程序	VCX	可视类库
FKY	宏	MPX	编译后的菜单程序	LST	向导列表的文档
FLL	FoxPro 动态链接库	OCX	OLE 控制	WIN	窗口文件
FMT	格式文件	PJT	项目备注		
FPT	表备注	PJX	项目		

由表 5-4 可见, Visual FoxPro 系统允许建立的文件有数十种之多, 主要文件类型如图 5-28 所示。图中所示的大多是用户在开发一个应用系统过程中必须建立的重要文件。

1. 项目 (Project)

项目是用户为所开发的应用软件系统建立的文件, 它的扩展名为 “.pjx”。

项目文件是开发应用软件系统首先必须建立的基础文件。它把应用系统的所有其他文件集合成为一个有机整体, 形成一个可视化的管理控制中心。

2. 数据库 (Database)

数据库是应用系统所有数据的管理中心, 是第二个必须建立的重要文件, 其扩展名为 “.dbc”。

数据库统一管理存放数据的表、视图及表之间的联系, 保证数据的完整性。

3. 表 (Table)

表即为存放具体数据的二维表, 其扩展名为 “.dbf”。

Visual FoxPro 中的表分为两种: 一种是归属于数据库管理的表, 称为 “数据库表”, 是存放数据的主要形式; 另一种称为 “自由表”, 其不归数据库管理, 它不具有数据库表的一些特性。

4. 视图 (View)

视图是一个定制的虚拟表或逻辑表, 它是由一个已经存在的或多个相关的数据表根据需要组织起来的查看数据表的一个窗口, 通过它不仅可以查看表中感兴趣的内容, 而且可以更新表的数据。

5. 查询 (Query)

从指定的表或视图中快速地筛选出需要的记录, 为报表或表单组织信息, 或用查询结果创建新的表或图形。

查询是最常用的一种操作, 查询以文件的形式存放在磁盘中, 其扩展名为 “.qpr”。查询同样也是项目数据的组成部分, 建立好的查询随时可以运行。

6. 表单 (Form)

表单是应用系统最主要的操作界面, 是最能体现面向对象程序设计的思想和方法的工具。表单又是实现各种操作功能的 “窗口”, 如数据的输入、维护、查询、报表打印等。Visual FoxPro 面向对象的程序设计主要是各种表单的设计, 其扩展名为 “.scx”。

7. 报表 (Report)

报表是 Visual FoxPro 最重要的打印输出文件, 也是一个应用系统设计的重要部分。其扩展名为 “.frx”。

8. 标签 (Label)

标签是类似 “名片” 形状的打印输出方式, 即把表中的每一条记录分别作为一个输出单元。每张纸可以打印若干个 “标签”。标签上可以打印报表中不宜打印的字段, 如 “备注” 或 “通用型” 字段。标签的扩展名为 “.lbt”。

9. 程序 (Program)

程序即命令文件, 其扩展名为 “.prg”。在面向对象的程序设计中, 需要编写的程序文件不多, 绝大部分都用可视化表单代替。但是程序设计的基本结构和方法, 仍是进行系统开发所需要掌握的基本功。

10. 菜单 (Menu)

菜单能够表示应用系统功能并通过它调用各种程序和表单等, 包括下拉式菜单和快捷菜单。菜单文件的扩展名为 “.mnx”, 经过编译的菜单程序文件扩展名为 “.mpr”。

除此之外, 在 Visual FoxPro 中还包括一些其他类型的文件, 如可执行文件 “.exe”、应用程序文件 “.app” 等。



图 5-28 主要文件类型

5.5 项目管理器

所谓项目就是一项具体的应用，是文件、数据、文档和对象的集合。项目是 Visual FoxPro 中一个重要的文件类型，其扩展名为 .pjx。

创建一个应用系统首先要创建项目，同时也要创建这个项目的项目管理器。项目管理器，是 Visual FoxPro 中最重要的开发平台和控制中心。主要有两个功能：

- (1) 用可视化的方法组织和处理本项目的数据库、表、表单、报表、菜单、程序等一切文件资源，实现对文件的创建、修改、删除等操作。
- (2) 在项目管理器中可以将应用程序编译成一个扩展名为 .app 的应用文件或 .exe 的可执行文件。

5.5.1 创建项目

利用 Visual FoxPro 设计数据库应用系统，首先从创建项目开始。

1. 项目的创建

在 Visual FoxPro 中提供了两种方式创建项目：

- (1) 仅创建一个项目文件，用来分类管理本项目的所有文件资源。
- (2) 使用应用程序向导创建一个项目，同时生成一个 Visual FoxPro 应用程序框架。

创建项目文件的步骤如下：

- (1) 在主菜单“文件”菜单中选择“新建”命令，或者单击“常用”工具栏中“新建”按钮，打开“新建”对话框，如图 5-29 所示。
- (2) 在“新建”对话框中，选中“项目”单选按钮，单击“新建文件”按钮，弹出“创建”对话框，如图 5-30 所示。

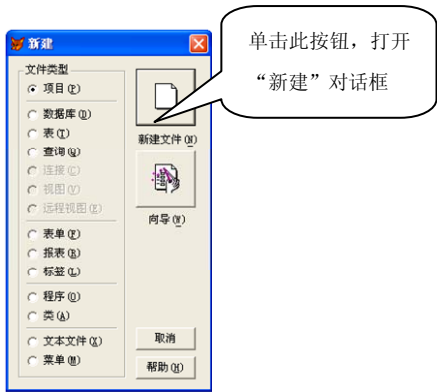


图 5-29 “新建”对话框

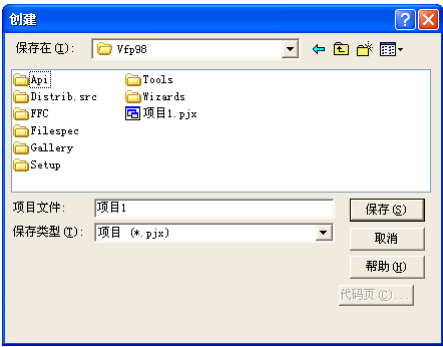


图 5-30 “创建”对话框

- (3) 在“创建”对话框中的“项目文件”文本框中输入项目的名称，在“保存在”组合框中选择存放文件的目录，单击“保存”按钮，完成项目的创建。

另外，也可以在命令窗口中输入“创建项目”命令，如图 5-31 所示。

2. 项目的打开和关闭

可以随时打开或关闭一个项目文件。具体的步骤如下：

在主菜单“文件”菜单中选择“打开”命令，或者单击“常用”工具栏上的“打开”按钮，弹出“打开”对话框，在“文件类型”下拉列表框中选择“项目”选项，在“查找范围”下拉列表框中选择项目所在目录，单击“确定”按钮，打开一个已经存在的项目，如图 5-32 所示。

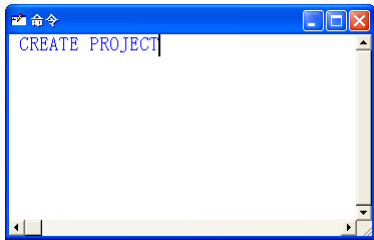


图 5-31 在命令窗口中输入“创建项目”命令

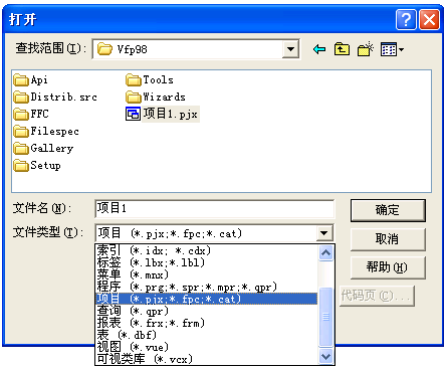


图 5-32 “打开”对话框

(2) 在“项目管理器”对话框中，单击其右上角“关闭”按钮，则关闭当前打开的项目。

3. “项目管理器”对话框

项目管理器是整个开发项目所有文件资源的管理中心，一共有 6 个选项卡，分别管理不同类型的文件。各个选项卡所包含的内容和基本功能如表 5-5 所示。

表 5-5 项目管理器选项卡及功能

选项卡	功能
全部	显示项目中所有的文件，实现对它们的管理
数据	分层次管理数据库、自由表和查询
文档	管理表单、报表和标签
类	管理所有类文件，包括系统基类和用户自定义的对象类
代码	管理扩展名为.prg 的程序文件、.app 的应用程序文件和函数库 API
其他	管理文本文件、菜单文件和其他文件

5.5.2 使用项目管理器

在项目管理器中，可以利用可视化操作在项目中创建、添加、修改、移去和运行指定的文件，如图 5-33 所示。

1. 创建文件

在“项目管理器”中创建一个新的文件，首先选择需要创建的新文件的类型，然后单击“新建”按钮，或者在“项目”菜单中选择“新建文件”命令，即可创建一个新的文件，所创建的文件将自动包含在该项目之中。

2. 添加文件

在“项目管理器”中可以添加一个已经存在的文件到项目之中，具体步骤如下：

- (1) 选择需要添加的文件的类型，单击“添加”按钮或者在“项目”菜单中选择“添加文件”命令，弹出“打开”对话框，在其中选择需要添加的文件。
- (2) 单击“确定”按钮，完成添加文件到项目中。

3. 修改文件

在“项目管理器”中可以修改项目中一个指定的文件，具体步骤如下：

- (1) 选择需要修改的文件，单击“修改”按钮或者在“项目”菜单中选择“修改文件”命令，系统将根据需要修改文件的类型打开相应的设计器。

(2) 在设计器中修改该文件。

4. 移去文件

在“项目管理器”中也可以移去项目中某个不需要的文件，具体步骤如下：

(1) 选择需要移去的文件，单击“移去”按钮或者在“项目”菜单中选择“移去文件”命令，系统将弹出一个提示对话框，如图 5-34 所示。

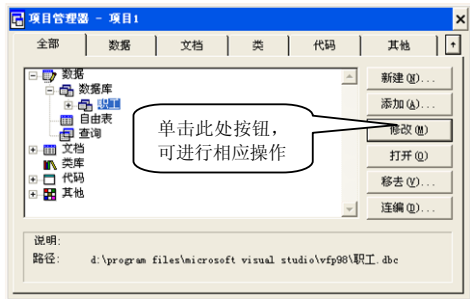


图 5-33 “项目管理器”对话框



图 5-34 提示对话框

(2) 在提示对话框中，若单击“移去”按钮，则系统将所选择文件从项目中移去，但是该文件仍然存在；若单击“删除”按钮，则系统不仅将所选择文件从项目中移去，而且还会彻底删除该文件，该文件将不复存在。

5.其他

在“项目管理器”中，随着所选择的文件类型的不同，会出现一些其他类型的按钮，这些按钮的功能如表 5-6 所示。

表 5-6 “项目管理器”中其他按钮及功能

按钮	功能	选择文件类型
浏览	浏览显示项目中指定的数据表或视图	表、视图
关闭/打开	关闭/打开一个数据库，若选定的数据库已打开，则出现“关闭”按钮，若选定的数据库已关闭，则出现“打开”按钮	数据库
预览	以打印预览方式显示指定的报表或标签	报表、标签
运行	执行指定的查询、表单或程序	查询、表单、程序
连编	连编一个项目或应用程序	

5.5.3 定制项目管理器

用户可以根据个人的喜好对项目管理器进行个性化的定制，可以移动项目管理器的位置、可以放大或缩小项目管理器的窗口大小、可以拆分项目管理器的窗口或者可以选择停放项目管理器的窗口。

1. 移动、缩放和折叠

用户可以随时移动项目管理器的位置和改变项目管理器的窗口大小，具体的方法如下：

- (1) 用鼠标单击项目管理器的“标题栏”，然后拖动鼠标，移动项目管理器到想要的位置即可。
- (2) 将鼠标指针指向项目管理器窗口的四边或四角，拖动鼠标，可以改变项目管理器窗口的大小。
- (3) 用鼠标单击项目管理器窗口右上角的箭头按钮，可以“折叠”或“展开”项目管理器的窗口，单击按钮，是“折叠”项目管理器的窗口，单击按钮，是“展开”项目管理器的窗口。“折叠”后的项目管理器的窗口如图 5-35 所示。

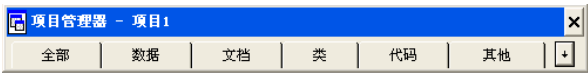


图 5-35 折叠后的项目管理器窗口

2. 拆分项目管理器

将项目管理器的窗口折叠后，还可以对项目管理器的各个选项卡进行拆分，使其成为独立的浮动窗口。在折叠后的项目管理器窗口，选择一个选项卡，用鼠标拖动它离开项目管理器的折叠窗口，该选项卡就成为了一个独立的、浮动的选项卡，如图 5-36 所示。

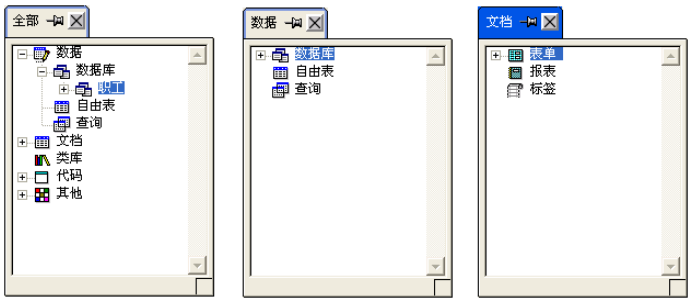


图 5-36 拆分后的选项卡

单击某个浮动的选项卡窗口右上角的“关闭”按钮，可以将该拆分的选项卡返回到项目管理器窗口中。单击某个浮动的选项卡窗口右上角的  按钮，可以将该选项卡窗口置为“顶层窗口”，再次单击该按钮，则取消“顶层窗口”设置。

3. 停放项目管理器

将项目管理器窗口拖曳到 Visual FoxPro 主窗口的顶部，就使其成为工具栏区域的一部分。

5.6 向导、设计器、生成器简介

在 Visual FoxPro 中提供了 3 种界面操作工作：向导、设计器和生成器。在进行系统开发和程序设计的过程中，充分利用这些工具，可以提高工作的效率。

5.6.1 向导

向导是一种交互式程序，用户在向导的指引下，通过选择或输入适当参数，系统即可直接生成相应的文件或者执行相应的任务。

Visual FoxPro 提供了 20 多种向导，主要有数据库向导、表向导、报表向导和表单向导等。

向导自动生成的文件往往具有一定的局限性，如果用户对此不满意，可以在相应的设计器中进行修改和完善。

Visual FoxPro 提供的主要向导及其作用如表 5-7 所示。

表 5-7 Visual FoxPro 常用向导及作用说明

向导	作用	向导	作用
应用程序向导	创建一个应用程序	一对多报表向导	创建一个一对多报表
数据库向导	生成一个数据库	数据透视表向导	创建数据透视表
表单向导	创建一个表单	查询向导	创建一个查询
图形向导	创建一个图形	远程视图向导	创建一个远程视图
导入向导	导入或追加数据	报表向导	创建一个报表
标签向导	创建一个标签	安装向导	创建发布盘
本地视图向导	创建一个本地视图	表向导	创建一个数据表
一对多表单向导	创建一个一对多表单	Web 发布向导	在 HTML 中显示数据

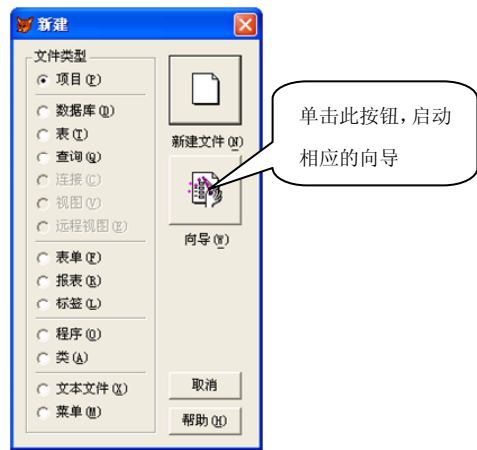


图 5-37 在“新建”对话框启动向导

1. 启动向导

在 Visual FoxPro 中，启动“向导”有以下 4 种方法：

(1) 在主菜单“文件”菜单中选择“新建”命令，或者单击工具栏中的“新建”按钮，弹出“新建”对话框，如图 5-37 所示。

(2) 在“新建”对话框中，选择需要创建的文件类型，单击“新建”按钮，系统就会弹出相应的“新建 XX”对话框，然后单击“向导”按钮，就可以启动相应的向导，如图 5-38 所示。



图 5-38 在“新建 XX”对话框中启动向导

(3) 在主菜单“工具”菜单中选择“向导”命令，在“向导”子菜单中选择需要创建的文件类型，就可以启动相应的向导，如图 5-39 所示。

(4) 单击“工具栏”中“向导”按钮，就可以直接启动相应的向导。

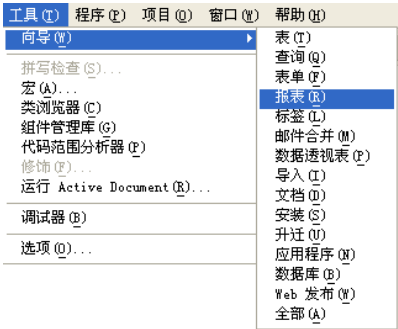


图 5-39 在“工具”菜单中启动向导

2. 使用向导

在使用向导时，需要看清楚系统提供的每一步的提示信息，确定完成好本步的操作后，单击“下一步”按钮，进入到下一步的操作；同时，可以单击“上一步”按钮，重新返回到上一步操作，进行修改或重新操作；若有疑问，可以按【F1】键寻求帮助；单击“取消”按钮将退出向导，并不会产生任何结果；对操作结果满意后，单击“完成”按钮，退出向导。

5.6.2 设计器

设计器是创建和修改应用系统各种组件的“基本可视化工具”。Visual FoxPro 系统提供了许多设计器，利用各种设计器使得创建和修改表、表单、数据库、查询和报表变得轻而易举。

1. 各种设计器

熟练运用各种设计器，是掌握 Visual FoxPro 系统开发的程序设计的基本要求。

Visual FoxPro 系统提供的各种设计器的名称和功能如表 5-8 所示。

表 5-8 Visual FoxPro 设计器及功能

设计器	功能
表设计器	创建并修改数据库表、自由表、字段和索引，可以实现如有效性检查和默认值等高级功能
数据库设计器	管理数据库中包含的全部表、视图和关系，当该窗口活动时，显示“数据库”菜单和“数据库设计器”工具栏
报表设计器	创建和修改打印数据的报表，当该窗口活动时，显示“报表”菜单和“报表控件”工具栏
查询设计器	创建和修改在本地表中运行的查询，当该窗口活动时，显示“查询”菜单和“查询设计器”工具栏
视图设计器	创建和修改在远程数据源上运行的视图，当该窗口活动时，显示“查询”菜单和“视图设计器”工具栏
表单设计器	创建和修改表单和表单集，当该窗口活动时，显示“表单”菜单、“表单设计器”工具栏、“表单控件”工具栏和“属性”窗口
菜单设计器	创建和修改菜单栏和弹出式子菜单
数据环境设计器	定义和修改表单或报表使用的数据源，包括表、视图和关系
连接设计器	为远程视图创建并修改命名连接，因为连接是作为数据库的一部分存储的，所以仅在已打开的数据库中使用“连接设计器”

2. 打开设计器

在 Visual FoxPro 中，可以使用以下 4 种方法打开设计器。

- (1) 在“项目管理器”对话框中打开。利用项目管理器可以快速打开各种 Visual FoxPro 设计器。在“项目管理器”对话框中，选择相应的选项卡，选中需要创建的文件类型，单击“新建”按钮，弹出“新建 XX”对话框，单击“新建 XX”按钮，系统就会打开相应的设计器。
- (2) 在“文件”菜单中打开。在主菜单“文件”菜单中选择“新建”命令，或者单击工具栏中的“新建”按钮，弹出“新建”对话框，选中需要创建的文件类型，单击“新建文件”按钮，系统就会打开相应的设计器。
- (3) 在“显示”菜单中打开。当打开某种类型的文件时，在主菜单“显示”菜单中就会出现相应的“XX 设计器”命令，选择该命令，系统就会打开相应的设计器。
- (4) 使用命令方式打开。在命令窗口输入创建某种类型文件的命令，系统就会打开相应的设计器。

5.6.3 生成器

每个生成器显示一系列选项卡，用于设置选中对象的属性。可以使用生成器在数据库表之间生成控件、表单、设置控件格式和创建参照完整性。生成器是带有选项卡的对话框，用于简化对表单、复杂控件和参照完整性代码的创建。Visual FoxPro 中提供了多种生成器，这些生成器的名称和功能如表 5-9 所示。

表 5-9 Visual FoxPro 生成器及功能

生成器	功能
应用程序生成器	迅速创建一个功能齐全的、完整的应用程序
自动格式生成器	将一组样式应用于选中的相同类型的控件
组合框生成器	设置组合框控件的属性
命令按钮生成器	设置命令按钮控件的属性
编辑框生成器	设置编辑框控件的属性
列表框生成器	设置列表框控件的属性
表格生成器	设置表格控件的属性
选项按钮生成器	设置选项按钮控件的属性
文本框生成器	设置文本框控件的属性
表单生成器	向表单中添加字段作为表单中新的控件
参照完整性生成器	设置触发器，用来控制在相关表中插入、更新或者删除记录，以确保参照完整性

在 Visual FoxPro 中，通常在以下 5 种情况下启动生成器：

- (1) 使用表单生成器来创建和修改表单。
- (2) 对表单中的控件使用相应的生成器。
- (3) 使用自动格式生成器来设置控件格式。
- (4) 使用参照完整性生成器来设置触发器。
- (5) 使用应用程序生成器为开发的项目生成应用程序。

1. 启动表单生成器

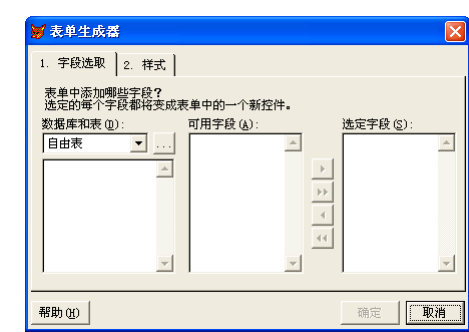


图 5-40 表单生成器

可以使用以下 3 种方法启动表单生成器，打开后的表单生成器如图 5-40 所示。

- (1) 打开“表单设计器”，在当前表单中单击鼠标右键，弹出快捷菜单，从中选择“生成器”命令。
- (2) 打开“表单设计器”，在主菜单“表单”菜单中选择“快速表单”命令，系统就会打开表单生成器。
- (3) 打开“表单设计器”，单击表单设计器工具栏中“表单生成器”按钮，系统就会打开表单生成器。

2. 对表单中的控件使用相应的生成器

对表单中的控件使用相应的生成器的步骤如下：

- (1) 打开“表单设计器”，从“表单控件”工具栏中选择相应的控件添加到表单上。
 - (2) 选中该控件，单击鼠标右键，弹出快捷菜单，从中选择“生成器”命令，系统就会打开相应控件所对应的生成器对话框。
- 各种控件生成器如图 5-41~图 5-47 所示。

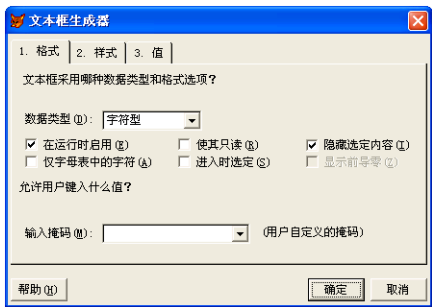


图 5-41 文本框生成器

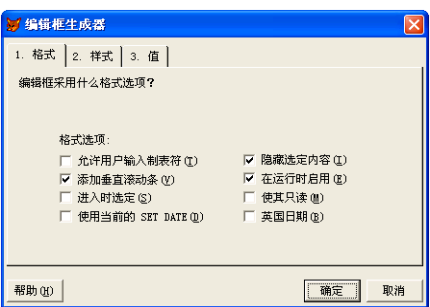


图 5-42 编辑框生成器

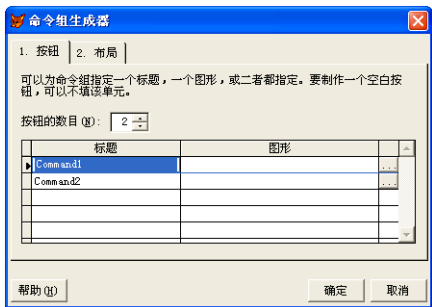


图 5-43 命令按钮组生成器

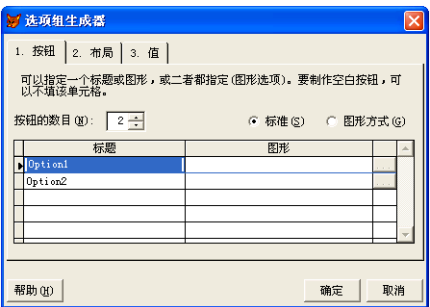


图 5-44 选项组生成器



图 5-45 组合框生成器



图 5-46 列表框生成器



图 5-47 表格生成器

3. 使用自动格式生成器来设置控件格式

为了对表单中的多个同类控件进行格式化，可以使用“自动格式生成器”同时对这些同类控件设置它们的格式。具体步骤如下：

- (1) 在“表单设计器”中，在按住【Shift】键，同时选择多个类型相同的控件。
- (2) 在表单设计器工具栏中，选择“自动格式”按钮，系统就会打开格式生成器。

打开后的格式生成器如图 5-48 所示。

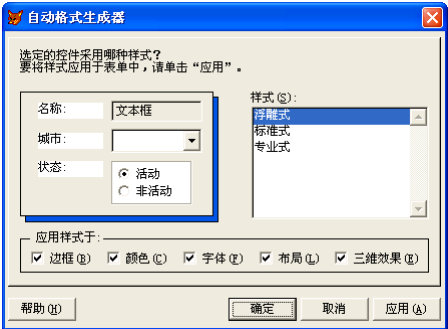


图 5-48 自动格式生成器

4. 使用参照完整性生成器

在参照完整性生成器中设置触发器，用来控制在相关表中插入、更新或者删除记录，以确保参照完整性。可以用以下 3 种方式，打开参照完整性生成器。

- (1) 在“数据库设计器”中，双击两个表之间的关系线，然后在“编辑关系”对话框中选择“参照完整性”选项。
 - (2) 在“数据库设计器”中，单击鼠标右键，选择“编辑参照完整性”命令。
 - (3) 在“数据库设计器”激活的状态下，选择主菜单“数据库”中的“编辑参照完整性”命令。
- 打开后的参照完整性生成器如图 5-49 所示。

5. 使用应用程序生成器

使用应用程序生成器，可以迅速创建一个功能齐全的、完整的应用程序。打开后的应用程序生成器如图 5-50 所示。

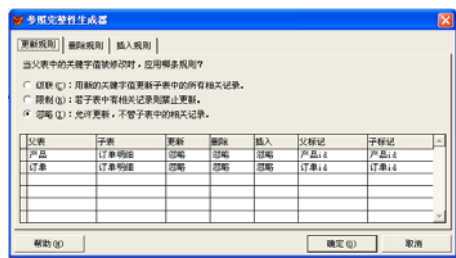


图 5-49 参照完整性生成器

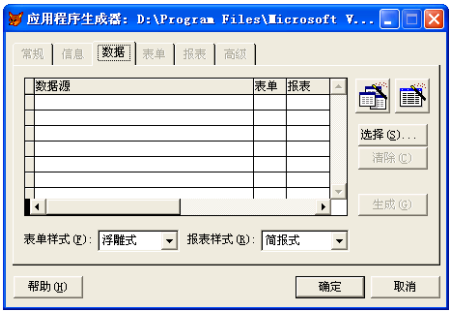


图 5-50 应用程序生成器

5.7 本章习题

5.7.1 选择题

1. 在 Visual FoxPro 中以下叙述错误的是（ ）。
- A. 关系也被称为表

B. 数据库文件不存储用户数据

C. 表文件的扩展名是 .dbf

D. 多个表存储在一个物理文件中
2. 计算机数据管理依次经历的几个阶段为（ ）。
- A. 文件系统、人工管理、数据库系统、分布式数据库系统和面向对象数据库系统

B. 人工管理、文件系统、数据库系统、面向对象数据库系统和分布式数据库系统

C. 人工管理、文件系统、数据库系统、分布式数据库系统和面向对象数据库系统

D. 文件系统、人工管理、数据库系统、面向对象数据库系统和分布式数据库系统
3. 数据库系统的核心是（ ）。
- A. 数据模型

B. 数据库管理系统

C. 数据库

D. 数据库管理员
4. 扩展名为.mnx 的文件是（ ）。
- A. 备注文件

B. 项目文件

C. 表单文件

D. 菜单文件
5. 数据库系统对数据库进行管理的核心软件是（ ）。
- A. DBMS

B. DB

C. OS

D. DBS
6. 关系运算中的选择运算是（ ）。
- A. 从关系中找出满足给定条件的元组的操作

B. 从关系中选择若干个属性组成新的关系的操作

C. 从关系中选择满足给定条件的属性的操作

D. A 和 B 都对
7. DBMS 的含义是：（ ）。
- A. 数据库系统

B. 数据库管理系统

C. 数据库管理员

D. 数据库
8. 对于“关系”的描述，正确的是（ ）。
- A. 同一个关系中允许有完全相同的元组

B. 同一个关系中元组必须按关键字升序存放

C. 在一个关系中必须将关键字作为该关系的第一个属性

D. 同一个关系中不能出现相同的属性名
9. 扩展名为.dbf 的文件是（ ）。
- A. 表文件

B. 表单文件

C. 数据库文件

D. 项目文件

10. 下面有关对数据的理解, 其中说法有误的是 ()。
- A. 数据是指存储在某一种媒体上能够被识别的物理符号
B. 包括描述事物特性的数据内容
C. 包括存储在某一种媒体上的数据形式
D. 数据形式是单一化的
11. 关系数据库管理系统所管理的系统是 ()。
- A. 若干个二维表 B. 一个 DBF 文件 C. 一个 DBC 文件 D. 若干个 DBC 文件
12. 在 Visual FoxPro 中, 用二维表来表示实体与实体间联系的数据模型称为 ()。
- A. 层次模型 B. 网状模型 C. 关系模型 D. E-R 模型
13. 在关系的基本运算中, 下列不属于专门关系运算的是 ()。
- A. 选择 B. 投影 C. 排序 D. 连接
14. 下列关于数据库系统, 说法正确的是 ()。
- A. 数据库中只存在数据项之间的联系
B. 数据库中只存在记录之间的联系
C. 数据库中数据项之间和记录之间都存在联系
D. 数据库中数据项之间和记录之间都不存在联系
15. 在 Visual FoxPro 中, 数据完整性包括 ()。
- A. 实体完整性和参照完整性 B. 实体完整性、域完整性和参照完整性
C. 实体完整性、域完整性和数据库完整性 D. 实体完整性、域完整性和数据表完整性
16. 数据库 (DB)、数据库系统 (DBS) 和数据库管理系统 (DBMS) 之间的关系是 ()。
- A. DB 包括 DBS 和 DBMS B. DBS 包括 DB 和 DBMS
C. DBMS 包括 DB 和 DBS D. 三者平级关系
17. 数据库系统的核心是 ()。
- A. 数据库 B. 操作系统 C. 数据库管理系统 D. 文件系统
18. 控制两个表中数据的完整性和一致性可以设置参照完整性。参照完整性要求这两个表是 ()。
- A. 不同数据库中的表 B. 同一个数据库中的表
C. 一个数据库表和一个自由表 D. 两个自由表文件
19. 参照完整性的前提是 ()。
- A. 建立两个表之间的联系 B. 系统存在两个自由表
C. 系统存在两个数据表 D. 系统中存在一个表
20. 存储在计算机设备上、结构化的相关数据集合称为 ()。
- A. 数据库 B. 数据库系统 C. 数据库管理系统 D. 数据模型
21. 下列关于数据库系统数据冗余的叙述中, 正确的是 ()。
- A. 数据库系统比文件系统出现的冗余多
B. 数据库系统中数据冗余是指数据超出指定数量
C. 数据库系统没有数据冗余
D. 数据库系统减少了数据冗余
22. 下列关于对象的说法, 不正确的是 ()。
- A. 对象可以是具体的实物, 也可以是一些概念
B. 一条命令、一个人、一个桌子等都可以看做是一个对象
C. 一个表单可以看做是一个对象
D. 一次考试不可以看做是一个对象
23. 数据处理的核心问题是 ()。
- A. 数据 B. 数据处理 C. 数据管理 D. 数据计算

24. Visual FoxPro DBMS 是 ()。
- A. 操作系统的一部分 B. 操作系统支持下的系统软件
- C. 一种编译程序 D. 一种操作系统
25. Visual FoxPro 是一种 ()。
- A. 数据库系统 B. 数据库管理系统 C. 数据库 D. 数据库应用系统
26. Visual FoxPro DBMS 基于的数据模型是 ()。
- A. 层次型 B. 关系型 C. 网状型 D. 混合型
27. Visual FoxPro 是一种关系型数据库管理系统，这里的关系通常是指 ()。
- A. 数据库文件 (.dbc 文件) B. 一个数据库中两个表之间有一定的关系
- C. 表文件 (.dbf 文件) D. 一个表文件中两条记录之间有一定的关系
28. 下列关于关系模式的叙述，不正确的是 ()。
- A. 关系模式是对关系的描述
- B. 一个关系模式对应一个关系结构
- C. 关系模式的格式：关系名 (属性名 1, 属性名 2, ..., 属性名 n)
- D. 关系模式的格式：表名 (字段名 1, 字段名 2, ..., 字段名 n)
29. 从关系模式中指定若干个属性组成新的关系的运算称为 ()。
- A. 连接 B. 投影 C. 选择 D. 排序
30. 从表中选择字段形成新关系的操作是 ()。
- A. 选择 B. 连接 C. 投影 D. 并
31. 利用 Visual FoxPro 来开发数据库应用系统，在数据库设计过程中，应遵循的原则是 ()。
- I. 避免在表之间出现重复字段
- II. 用外部关键字保证有关联的表之间的联系
- III. 表中的字段必须是原始数据和基本数据元素
- IV. 关系数据库的设计应遵从概念单一化“一事一地”的原则
- A. 仅 I、II 和 III B. 仅 II、III 和 IV C. 仅 I、II 和 IV D. 全部
32. 规范化理论是关系数据库进行逻辑设计的理论依据。根据这个理论，关系数据库的关系必须满足其每一个属性都是 ()。
- A. 互不相关的 B. 不可分解的 C. 长度可变的 D. 互相关联的

5.7.2 填空题

1. 在关系数据库中，把数据表示成二维表，每一个二维表称为_____。
2. 在奥运会游泳比赛中，一个游泳运动员可以参加多项比赛，一个游泳比赛项目可以有多个运动员参加，游泳运动员与游泳比赛项目两个实体之间的联系是_____联系。
3. Visual FoxPro 中数据库文件的扩展名 (后缀) 是_____。
4. 在关系模型中，“关系中不允许出现相同元组”的约束是通过_____实现的。
5. 在 Visual FoxPro 中，项目文件的扩展名是_____。
6. 在将设计好的表单存盘时，系统生成扩展名分别是.scx 和_____的两个文件。
7. Visual FoxPro 的 3 种基本操作方式为_____、菜单方式和程序方式。
8. Visual FoxPro 是运行于 Windows 平台的数据库管理系统，它在支持标准的面向过程的程序设计方式的同时还支持_____的程序设计方式。
9. 数据库是存储在计算机存储设备上、结构化的相关数据集合，它不仅包括描述事物的数据本身，而且还包括_____。
10. 两个实体间的关系可分为 3 种类型：一对一、一对多和_____。
11. 如果表中的一个字段不是本表的关键字或候选关键字，而是另一个表的关键字或候选关键字，那么这个字段 (属性) 就称为_____。

12. 在查询关系数据库中用户需要的数据时，需要对关系进行一定的_____。
13. _____是存储在某一种媒体上能够识别的物理符号，其概念包含两个方面：一是描述事物特性的数据内容；二是存储在某一种媒体上的数据形式。
14. 对某个关系进行选择、投影或连接运算后，运算的结果仍然是一个_____。
15. 数据库系统不仅可以表示事物内部各数据项之间的联系，而且还可以表示_____之间的关系。
16. 关系数据库管理系统所管理的关系是_____。
17. 任何一个数据库管理系统都是基于_____建立的。
18. 使用_____命令可以直接退出 Visual FoxPro 系统。
19. 在基本表中，要求字段名_____重复。
20. 数据库系统对数据库进行管理的核心软件是_____。

Visual FoxPro 程序设计基础

在进行数据处理时，除了需要处理数据表中的数据之外，还经常要处理其他数据。在 Visual FoxPro 系统开发过程中经常涉及的基本数据元素，包括常量、变量、表达式和函数 4 种形式的数据。常量和变量是数据运算和处理的基本对象，而表达式和函数则体现了语言对数据进行运算和处理的能力。

本章将主要介绍这些基本数据元素、程序的基本结构和多模块程序设计。

6.1 常量与变量

在程序的执行过程中有两种形式的数 据，一种称为常量，另一种称为变量，下面分别予以介绍。

6.1.1 常量

常量，是指在程序的执行过程中其值不能改变的量。在 Visual FoxPro 中，常量包括以下几种类型，不同类型的常量有不同的书写格式。

1. 数值型常量

数值型常量也就是常数，用来表示一个数量的大小。数值型常量由数字 1~9、小数点和正负号构成。

例如，15、45.23、-8.3 等。

另外，也可以用科学记数法表示数值型常量。

例如，2.8E5 表示 2.8×10^5 ，1.23E-6 表示 1.23×10^{-6} 。

系统内存用 8 个字节来表示一个数值型常量，其取值范围是 $-0.9999999999\text{E}+19 \sim 0.9999999999\text{E}+20$ 。

2. 字符型常量

字符型常量又称字符串，由英文状态的单引号、双引号和方括号括起来的字符序列。这里的单引号、双引号和方括号又称为定界符。定界符的作用是确定字符串的起始位置和终止界限，它本身不是字符型常量的内容。定界符必须成对出现，即前后定界符一定要一致。

例如，“系统软件”、“计算机”、[456]等。

在字符型常量中，不包含任何字符的字符串（""）叫空串。空串与包含空格的字符串（" "）不同。

3. 日期型常量

日期型常量是使用一对花括号括起来的日期型数据，用来表示具体日期的常量。花括号内包括年、月、日 3 部分内容，各部分内容之间用分隔符分隔，分隔符可以是斜杠 (/)、连字符 (-)、句点 (.) 和空格。其中斜杠 (/) 是系统在显示日期型数据时使用的默认分隔符。

在 Visual FoxPro 中，日期型常量的格式有两种。

1) 传统的日期格式

系统默认的日期型数据为美国日期格式“mm/dd/yy”(月/日/年)，传统日期格式中的月、日各为 2 位数字，而年份可以是 2 位数字，也可以是 4 位数字。

例如，{10/08/01}、{10-08-01}、{10 08 2001}等。

2) 严格的日期格式

{^yyyy-mm-dd}，格式中的符号“^”表示该日期格式是严格的，并按照 YMD 的格式来解释日期。

严格的日期格式可以在任何情况下使用，用这种格式书写的日期型常量能表达一个确切的日期，它不受 SET DATE 等语句设置的影响。

严格的日期格式的日期型常量在书写时要注意：

- 花括号内第一个字符必须是脱字符 (^)。
- 年份必须用 4 位 (如 2010、1996 等)。
- 年、月、日的次序不能颠倒、不能默认。

日期型数据用 8 个字节表示，取值范围是 0001 年 1 月 1 日~9999 年 12 月 31 日。

严格的日期格式可以在任何情况下使用，而传统的日期格式只能在 SET STRICTDATE TO 0 状态下使用。系统默认的日期格式为严格的日期格式，输入日期型常量时使用严格的日期格式十分方便。

3) 影响日期格式的设置命令

- 命令格式：SET MARK TO [<日期分隔符>]。

命令功能：用于设置显示日期型数据时使用的分隔符，如“-”、“.”等。如果执行 SET MARK TO 没有指定任何分隔符，则表示恢复系统默认的斜杠分隔符。

- 命令格式：SET DATE [TO] AMERICAN|ANSI|BRITISH|FRENCH|GERMAN
|ITALIAN|JAPAN|USA|MDY|DMY|YMD。

命令功能：设置日期显示的格式。

命令中各短语所定义的日期格式如表 6-1 所示。

表 6-1 常用日期格式

短语	格式	短语	格式
AMERICAN	mm/dd/yy	ANSI	yy.mm.dd
BRITISH/FRENCH	dd/mm/yy	GERMAN	dd.mm.yy
ITALIAN	dd-mm-yy	JAPAN	yy/mm/dd
USA	mm-dd-yy	MDY	mm/dd/yy
DMY	dd/mm/yy	YMD	yy/mm/dd

- 命令格式：SET CENTURY ON|OFF|TO [<世纪值> [ROLLOVER <年份参照值>]]。

命令功能：用于设置显示日期型数据时是否显示世纪。“ON”表示显示世纪，即用 4 位数字表示年份；“OFF”表示不显示世纪，即用 2 位数字表示年份。它是系统默认的设置。“TO”选项确定用 2 位数字表示年份所处的世纪，具体地说，如果该日期的 2 位数字年份大于等于<年份参照值>，则它所处的世纪即为<世纪值>；否则为<世纪值>+1。

- 命令格式：SET STRICTDATE TO [0|1|2]。

命令功能：用于设置是否对日期格式进行检查。

- 0：表示不进行严格的日期格式检查，目的是与早期 Visual FoxPro 兼容。
- 1：表示进行严格的日期格式检查，它是系统默认的设置。
- 2：表示进行严格的日期格式检查，并且对 CTOD()和 CTOT()函数的格式也有效。

若在 SET STRICTDATE TO 1 或 2 状态下使用传统日期格式，系统将弹出如图 6-1 所示的提示。

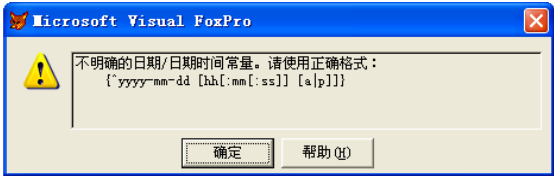


图 6-1 错误提示信息

例：在命令窗口输入如下命令，查看各种日期数据的输出结果。

输入命令：

```
SET STRICTDATE TO 0
SET CENTURY ON
?{12/28/08}
```

主屏幕显示：

```
12/28/2008
```

输入命令：

```
SET CENTURY OFF
```

主屏幕显示：

```
12/28/08
```

输入命令：

```
SET CENTURY TO 1?{^2008-12-28}
```

主屏幕显示：

```
12/28/2008
```

4. 日期时间型常量

日期时间型常量是用花括号括起来的日期时间型数据序列，花括号内包括日期和时间两部分内容：{<日期>,<时间>}。其中，<日期>部分的表示格式与日期型常量相同，<时间>部分的格式为：[hh[: mm[: ss]][a|p]]。其中，hh、mm 和 ss 分别代表时、分和秒，默认值分别为 12、0 和 0。AM（或 A）和 PM（或 P）分别代表上午和下午，默认值为 AM。如果指定的时间大于等于 12，则为下午的时间。

日期时间型数据用 8 个字节存储。日期部分的取值范围与日期型数据相同，时间部分的取值范围是 00: 00: 00 AM~11: 59: 59 PM。

例：在命令窗口输入如下命令，查看各种日期时间数据的输出结果。

```
? {^2008-12-24, 11:23:25 A}
```

主屏幕显示：

```
12/24/08 11:23:25 AM
```

5. 逻辑型常量

逻辑型常量是用来表示逻辑真和逻辑假的常量。逻辑型只有真和假两个值，逻辑真表示形式为：.T.、.t.、.Y.、.y.，逻辑假表示形式为：.F.、.f.、.N.、.n.。逻辑型常量左右两边的句点是定界符，不能省略。

逻辑型常量只占用 1 个字节。

6. 货币型常量

货币型常量用来表示货币值的常量。货币型常量和数值型常量的区别在于前面是否有货币符号（\$）。货币型常量采用 4 位小数存储，超过 4 位小数，系统将会把多于的小数位四舍五入。

货币型常量没有科学记数法形式，在内存中占用 8 个字节，取值范围是－922337203685477.5807～922337203685477.5807。

6.1.2 变量

变量，是指在程序的执行过程中其值可以改变的量。在 Visual FoxPro 中，变量分为字段变量和内存变量两种。

1. 字段变量

字段变量是指数据表中已经定义的任何一个字段名，由于在一个数据表中，字段的具体值与具体的记录有关，是随着记录行的变化而变化的，因此称之为变量。使用字段变量首先要建立数据表，在建立表的过程中创建字段变量。

2. 内存变量

内存变量是内存中的一个存储区域，通过内存变量名来标识该区域，其中存放的数据，称为内存变量的值。内存变量的类型取决于内存变量值的类型。

在 Visual FoxPro 中，内存变量包括简单内存变量和数组。

1) 内存变量的命名

为了区分不同的存储单元，对不同的存储单元需要指定不同的名称，即每个内存变量都应该有自己的名称。在 Visual FoxPro 中，内存变量的命名遵循以下命名规则：

- 内存变量名是由字母、数字和下划线组成的，长度不能超过 128 个字符，并且必须以字母或下划线开头。
- 不能使用系统保留字（系统中的命令）。
- 内存变量名不区分大小写。

2) 内存变量的类型

内存变量的类型有字符型（C）、数值型（N）、货币型（Y）、逻辑型（L）、日期型（D）和日期时间型（T）。

3) 内存变量的赋值

格式 1：STORE 命令格式。

STORE <表达式> TO <内存变量名表>

功能：将表达式计算出来的值赋给<内存变量名表>中的每一个变量。

格式 2：“=”命令格式。

<内存变量>=<表达式>

功能：将表达式计算出来的值赋给以<内存变量>命名的变量。

对两种赋值格式的具体说明如下：

- 格式 2 只能为单个变量赋值，格式 1 可以同时为若干个变量赋值。
- 在 Visual FoxPro 中，一个变量在使用前无须声明或定义，当给变量赋值时，系统会自动创建该变量。
- 可以通过对内存变量重新赋值改变其数据类型，即可以把不同类型的数据赋值给同一个内存变量。

4) 表达式值的显示

格式 1：? <表达式>。

格式 2：?? <表达式>。

功能：先计算表达式的值，然后将该值输出到显示器上。

区别：格式 1 自动换行显示结果，格式 2 在当前光标所在位置显示结果。

5) 内存变量的显示命令

格式 1：LIST MEMORY [LIKE <通配符>][TO PRINTER|TO FILE <文件名>]。

格式 2：DISPLAY MEMORY [LIKE <通配符>][TO PRINTER|TO FILE <文件名>]。

功能：显示内存变量的当前信息，包括变量名、作用范围、类型和值。选用 LIKE 短语可以显示与通配符相匹配的内存变量。通配符包括*和?，*表示任意多个字符，?表示任意一个字符。选用 TO PRINTER 用于在屏幕显示的同时在打印机中打印，选用 TO FILE <文件名>用于存入指定文件名的文本文件中，文件的扩展名为.txt。

区别：格式 1 不分屏显示，格式 2 分屏显示。

6) 内存变量的清除命令

格式 1：CLEAR MEMORY。

格式 2: RELEASE <内存变量名表>。

格式 3: RELEASE ALL [LIKE <通配符>|EXCEPT <通配符>]。

功能: 清除内存变量。

区别: 格式 1 清除所有的内存变量, 格式 2 清除指定的内存变量, 格式 3 清除与通配符相匹配 (LIKE) 的或不相匹配的 (EXCEPT) 内存变量。

例: 清除所有以 X 开头的内存变量。

```
RELEASE ALL LIKE X*
```

3. 字段变量和内存变量的区别

(1) 内存变量是计算机内存中一个命名的存储单元。一般将内存变量分为简单内存变量和数组变量。

(2) 字段变量就是字段, 也是变量。与其他变量不同的是, 字段变量是表中的变量, 它的值既随着表的记录变化, 又是永久性多值变量。字段名就是变量名, 其变量的数据类型为 Visual FoxPro 中的任意数据类型, 字段值就是变量值。

(3) 如果当前表中存在一个和内存变量同名的字段变量, 在访问内存变量时, 必须在变量名前加 M.或者 M->, 否则系统将访问同名的字段变量。

4. 数组

数组是按一定顺序排列的一组内存变量, 在内存中用一片连续的区域来存放, 每一个数组都有自己的数组名, 其中的每一个内存变量都称为数组的元素。当具体使用数组元素时, 用数组名和数组元素在数组中的排列标号来表示, 如用 A(1)、A(2)、A(3)等来表示一维数组中的第 1 个、第 2 个、第 3 个元素。如用 B(2, 3)表示二维数组中的第 2 行第 3 列的元素。

根据下标的个数可以把数组分为一维数组和二维数组。如 A(3)表示一维数组, B(2, 3)表示二维数组。

数组在使用之前必须先定义, 格式如下:

格式 1: DIMENSION <数组名>(<下标 1>[, <下标 2>])。

格式 2: DECLARE <数组名>(<下标 1>[, <下标 2>])。

两种格式的功能完全相同。数组创建之后, 每个数组元素的默认值为逻辑假 (.F.)。

数组的赋值和内存变量的赋值方法相同, 各个数组元素可以分别存放不同类型的数据。

例: 定义一个一维数组 A, 给所有数组元素赋值并输出其值。

```
DIMENSION A(6)
```

```
A(6)="987654"
```

```
STORE 0 TO A(3), A(4), A(5)
```

```
? A(1), A(2), A(3), A(4), A(5), A(6)
```

在使用数组和数组元素时, 具体说明如下:

- 在任何可以使用简单内存变量的地方, 都可以使用数组和数组元素。
- 在同一个运行环境下, 简单内存变量和数组不能同名。
- 在赋值和输入语句中使用数组名, 即省略下标时, 表示将同一个值赋给数组中的所有元素。
- 在赋值语句中数组名不能出现在表达式位置。
- 数组下标的下限为 1, 上限由数组的大小决定。
- 在二维数组中, 数组元素的个数为行下标上限值和列下标上限值的乘积。
- 可以用一维数组的形式访问二维数组。

例如, 数组 y 中含有 6 个元素: y(1, 1)、y(1, 2)、y(1, 3)、y(2, 1)、y(2, 2)、y(2, 3)。其各元素用一维数组形式可依次表示为: y(1)、y(2)、y(3)、y(4)、y(5)、y(6), 其中 y(4)与 y(2, 1)是同一变量。

5. 表中数据与数组中数据之间的交换

数据表中以记录方式存储的数据可以与数组中存储的数据进行相互交换。具体格式如下:

1) 将表中的当前记录复制到数组

格式 1: SCATTER [FIELDS <字段名表>] [MEMO] TO <数组名> [BLANK]

格式 2: SCATTER [FIELDS LIKE <通配符>| FIELDS EXCEPT <通配符>] [MEMO] TO <数组名> [BLANK]

格式 1 的功能是将表的当前记录从指定字段表中的第一个字段内容开始，依次复制到数组名中的从第一个数组元素开始的内存变量中。如果不使用 FIELDS 短语指定字段，则赋值除备注型 M 和通用型 G 之外的全部字段。

如果事先没有创建数组，系统将自动创建；如果已经创建的数组元素个数少于字段数，则系统自动建立其余数组元素；如果已经创建的数组元素个数多于字段数，其余数组元素的值保持不变。

若选用 MEMO 短语，则同时复制备注型字段。若选用 BLANK 短语，则产生一个空数组，各数组元素的类型和大小与表中当前记录的对应字段相同。

格式 2 的功能是用通配符指定包括或排除的字段。FIELDS LIKE <通配符>和 FIELDS EXCEPT <通配符>可以同时使用。

例：将“学生表.dbf”中的当前记录复制到数组 A 中，已知表中有 4 条记录。

```
USE 学生表
SCATTER TO ZG
? A(1), A(2), A (3), A(4)
```

2) 将数组数据复制到表的当前记录

格式 1: GATHER FROM <数组名> [FIELDS <字段名表>] [MEMO]

格式 2: GATHER FROM <数组名> [FIELDS LIKE <通配符>| FIELDS EXCEPT <通配符>] [MEMO]。

格式 1 的功能是将数组中的数据作为一个记录复制到表的当前记录中。从第一个数组元素开始，依次向字段名表指定的字段填写数据。如果不使用 FIELDS 短语指定字段，则依次向各个字段复制，如果数组元素个数多余记录中字段的个数，则多余部分被忽略。若选用 MEMO 短语，则在复制时包括备注型字段；否则备注型字段不予考虑。

格式 2 的功能是用通配符指定包括或排除的字段。FIELDS LIKE <通配符>和 FIELDS EXCEPT <通配符>可以同时使用。

6.2 运算符与表达式

运算是 对数据进行加工的过程，运算符是在代码中对各种数据类型进行运算的符号。按照操作数数目的多少来分，运算符可以分为单目运算符和双目运算符两类。单目运算符一般放在操作对象的前面，双目运算符都放在两个操作数之间。

表达式是指用括号和运算符把常量、变量及函数连接而成的式子，表达式具有计算、判断和数据类型转换等作用。表达式的值是按照规定的运算规则计算出的结果。表达式的类型可以分为数值表达式、字符表达式、日期时间表达式和逻辑表达式。

6.2.1 算术运算符与算术表达式

算术运算符是对数值型数据进行操作的一种符号，算术表达式由算术运算符将数值型数据连接起来形成，其运算结果仍然是数值型数据。算术运算符如表 6-2 所示。

表 6-2 算术运算符

优先级	运算符	说明
1	()	圆括号
2	** 或^	乘方运算符
3	*, /, %	乘、除、求余运算符
4	+, -	加减运算符

例：假设变量 X 的值是 3，计算表达式 6*X+8-4*(X-1)的值。

? 6*X+8-4*(X-1)

结果为：18.00

特殊地加法和减法运算符也可以用于日期（时间）型数据的操作，如表 6-3 所示。

表 6-3 日期时间表达式的格式

格式	结果及类型
<日期>+<天数>	日期型。指定日期若干天后的日期
<天数>+<日期>	日期型。指定日期若干天后的日期
<日期>-<天数>	日期型。指定日期若干天前的日期
<日期>-<日期>	数值型。两个指定日期相差的天数
<日期时间>+<秒数>	日期时间型。指定日期时间若干秒后的日期时间
<秒数>+<日期时间>	日期时间型。指定日期时间若干秒后的日期时间
<日期时间>-<秒数>	日期时间型。指定日期时间若干秒前的日期时间
<日期时间>-<日期时间>	数值型。两个指定日期时间相差的秒数

例：日期时间型数据运算。
? {^2005-1-3 10: 0: 0}-{^2005-1-3 9: 0: 0}
3600
? {^2005-1-13 }-{^2005-1-3}
10
需要注意的是，两个日期型和两个日期时间型数据不能相加。

6.2.2 字符运算符与字符表达式

字符运算符是对字符串进行连接操作的一种符号。字符运算符有两个，它们的优先级相同：

- +: 将前后两个字符串首尾连接形成一个新的字符串。
- -: 用来连接前后两个字符串，并将前面字符串的尾部空格移到合并后的新字符串尾部。

字符表达式由字符串运算符将字符型数据连接起来形成，其运算结果仍然是字符型数据。

例：字符型数据运算。

```
m= " Hello□□ "
n= " everybody! "
?m+n, m-n
Hello□□everybody!Helloeverybody!□□
```

6.2.3 关系运算符与关系表达式

关系运算符是对两个数据进行比较操作的一种符号。关系表达式也称为简单逻辑表达式，它由关系运算符将两个操作数连接起来，运算的结果一定是逻辑值。关系运算符如表 6-4 所示。

表 6-4 关系运算符

运算符	说明	运算符	说明
<	小于	<=	小于等于
>	大于	>=	大于等于
=	等于	==	字符串精确比较
<>、#、!=	不等于	\$	子串包含测试

关系运算符的优先级别相同。其中运算符==和\$仅适用于字符型数据，其他运算符适用于任何类型的数据，但除了日期型和日期时间型数据、数值型和货币型数据可以比较之外，在其他情况下，前后两个操作数的数据类型必须一致。

1. 两个数据的比较规则

(1) 当两个数值型数据比较时，按值的大小直接比较。

例：? 5<3, 6>4

主屏幕显示结果都为逻辑假.F.。

(2) 当两个日期型和日期时间型数据比较时，按年、月、日先后顺序比较，越晚的日期或时间越大。

例：? {^2005-1-12}>{^2005-1-3 0}

主屏幕显示结果都为逻辑假.F.。

(3) 当两个字符型数据比较时，英文字符按 ASCII 码值比较，汉字按机内码值比较，即按拼音顺序的先后比较。

需要注意的是：

- 字符型数据比较有方向性。
- 字符型数据比较可以有非精确比较和精确比较两种形式，若要使比较符“=”进行精确比较，必须使用命令 SET EXACT ON|OFF。

例：

? "刘谦">"赵明"

主屏幕显示结果都为逻辑假.F.。

? "ABD">"ABC"

主屏幕显示结果都为逻辑真.T.。

? "AB"="ABC"

主屏幕显示结果都为逻辑假.F.。

SET EXACT OFF

? "ABC"="AB"

主屏幕显示结果都为逻辑真.T.。

SET EXACT ON

? "ABC"="AB"

主屏幕显示结果都为逻辑假.F.。

(4) 子串包含测试。关系表达式“<字符型表达式 1>\$<字符型表达式 2>”为子串包含测试，如果左边字符串是右边字符串的一个子字符串，则结果为逻辑真 (.T.)，否则为逻辑假 (.F.)。

例：

? "AB"\$"ABC"

主屏幕显示结果都为逻辑真.T.。

2. 设置字符的排序次序

当对两个字符串进行比较时，系统会从左到右逐个字符进行比较，一旦发现两个对应的字符不同，就根据这两个字符的大小决定两个字符串的大小。

在 Visual FoxPro 中，默认的字符排序次序为“PinYin”，但是可以通过设置字符排序次序命令 SET COLLATE TO "<排序次序名>" 重新设置。

排序次序的种类和作用如表 6-5 所示。

表 6-5 排序次序的种类和作用

运算符	说明
Machine 机器次序	按照机器内码排序，西文字符是按 ASCII 码排序：空格在最前面，大写字母在小写字母前面。 汉字的机内码与汉字的国际码一致，即根据汉字的拼音顺序决定大小
PinYin 拼音次序	按照拼音次序排序。西文字符：空格在最前面，小写字母在大写字母前面
Stroke 笔画次序	无论中文、西文，都按照书写笔画的多少排序

3. 字符串精确比较与 EXACT 设置

可以使用“==”运算符对两个字符串进行精确比较，即只有当两个字符串完全相同（包括空格及各字符的位置）时，运算结果才为逻辑真（.T.），否则为逻辑假（.F.）。

在使用“=”运算符比较两个字符串时，运算结果与 SET EXACT ON|OFF 命令的设置有关。

- ON：在进行比较之前，系统先在较短字符串的尾部加上若干空格，使两个字符串的长度相等，然后再进行精确比较。
- OFF：只要右边字符串与左边字符串前面部分内容相匹配，即字符串的比较因右边字符串的结束而终止，此时结果为逻辑真（.T.）。

例：字符串比较与 EXACT 设置。

SET EXACT OFF

S1= " 中华人民共和国 "

S2= " 中华人民共和国□□ "

S3= " 中华 "

```
? S1=S3, S3=S1, S1=S2, S2=S1, S2==S1.T. .F. .F. .T. .F.
SET EXACT ON
? S1=S3, S3=S1, S1=S2, S2=S1, S2==S1
.F. .F. .T. .T. .F.
```

4. 赋值与相等比较的区别

内存变量的赋值命令与相等比较运算都使用“=”符号，两者的区别如下：

- （1）在赋值命令中，等号左边只能是一个内存变量名，其功能是计算等号右边表达式的值，并将计算结果赋值给内存变量。在命令执行之前，内存变量可以存在，也可以不存在。在命令执行后，内存变量的值与类型与表达式的值与类型一致。
- （2）在相等比较运算中，等号左边和右边既可以是表达式，也可以是变量。其计算结果也可以赋值给一个内存变量。

6.2.4 逻辑运算符与逻辑表达式

逻辑运算符是对逻辑型数据进行操作的一种符号，逻辑表达式是由逻辑运算符将逻辑型数据连接起来而形成的，其运算结果仍为逻辑值。

逻辑运算符如表 6-6 所示。

表 6-6 逻辑运算符

优先级	运算符	说明
1	.NOT.或!	逻辑非
2	.AND.	逻辑与
3	.OR.	逻辑或

逻辑运算规则的定义：

- .NOT.：单目运算符，当操作数为真时结果为假，反之结果为真。
 - .AND.：双目运算符，当两个操作数均为真时结果才为真，否则结果为假。
 - .OR.：双目运算符，当两个操作数均为假时结果才为假，否则结果为真。
- 逻辑运算的具体规则如表 6-7 所示。

表 6-7 逻辑运算规则

操作数 A	操作数 B	.NOT.A	A.AND.B	A.OR.B
.T.	.T.	.F.	.T.	.T.
.T.	.F.	.F.	.F.	.T.
.F.	.T.	.T.	.F.	.T.
.F.	.F.	.T.	.F.	.F.

6.2.5 运算符的优先级别

当在一个表达式中出现了各种运算符时，应该按照运算符优先级别的顺序依次来进行计算，Visual FoxPro 中不同种类的运算符的优先级别顺序如表 6-8 所示。

表 6-8 运算符的优先级别

优先级别	运算符
一级	()
二级	算术运算符
三级	字符运算符
四级	关系运算符
五级	逻辑运算符

例：计算表达式 4^2>8.AND.(“张三”\$“张”.OR.6<>4+2)的值。

结果：逻辑假 (.F.)。

运算顺序为：+，<>，\$，.OR.，^，>，.AND.。

6.3 常用函数

为了提高程序设计效率，Visual FoxPro 提供了大量的函数。这些函数是具有特定功能的程序，在程序设计时，灵活地应用这些函数，可以简化应用程序的编写，能够完成相当复杂的工作。

6.3.1 数值函数

数值函数是指函数值为数值的一类函数。数值函数包含了常用的数学函数和适合计算机数据处理的其他函数，如表 6-9 所示。

表 6-9 数值函数

格式	功能说明
ABS(<数值表达式>)	返回数值表达式的绝对值
SIGN(<数值表达式>)	返回数值表达式的符号。当数值表达式的计算结果为正、负和零时，函数值分别为 1、-1 和 0
SQRT(<数值表达式>)	返回数值表达式的平方根。数值表达式的值不能为负
PI()	返回圆周率，该函数没有自变量（参数）
INT(<数值表达式>)	返回指定数值表达式的整数部分
CEILING(<数值表达式>)	返回大于或者等于指定数值表达式的最小整数
FLOOR(<数值表达式>)	返回小于或者等于指定数值表达式的最大整数

续表

格式	功能说明
ROUND(<数值表达式 1>, <数值表达式 2>)	返回指定数值表达式在指定位置四舍五入后的结果。<数值表达式 2>指明四舍五入的位置。若<数值表达式 2>大于或者等于 0，那么它表示的是要保留的小数位数；若<数值表达式 2>小于 0，那么它表示的是整数部分的舍入位数
MOD(<数值表达式 1>, <数值表达式 2>)	返回两个数值相除后的余数。<数值表达式 1>是被除数，<数值表达式 2>是除数。余数的符号和除数相同。如果除数和被除数同号，函数值为两数相除的余数；如果除数和被除数异号，则函数值为两数相除的余数再加上除数的值
MAX(<数值表达式 1>, <数值表达式 2>, ..., <数值表达式 N>)	计算各数值表达式的值，并返回其中的最大值
MIN(<数值表达式 1>, <数值表达式 2>, ..., <数值表达式 N>)	计算各数值表达式的值，并返回其中的最小值

6.3.2 字符处理函数

字符函数是指自变量一般是字符型数据的函数。其处理的对象通常是字符型表达式，但函数值不一定是字符型数据。常见的字符处理函数如表 6-10 所示。

表 6-10 字符处理函数

格式	功能说明
LOWER(<字符表达式>)	将字符表达式中的大写字母转换成小写字母，其他字符不变
UPPER(<字符表达式>)	将字符表达式中的小写字母转换成大写字母，其他字符不变
LEN(<字符表达式>)	返回指定字符表达式的长度，即所包含的字符个数。函数值为数值型
SPACE(<数值表达式>)	返回数值表达式指定数量空格组成的字符串
TRIM(<字符表达式>)	返回字符表达式去掉尾部（右侧）空格后组成的字符串
LTRIM(<字符表达式>)	返回字符表达式去掉首部（左侧）空格后组成的字符串
ALLTRIM(<字符表达式>)	返回字符表达式去掉前后空格后组成的字符串
LEFT(<字符表达式>, <长度>)	返回字符表达式左侧起提取指定长度的字符串
RIGHT(<字符表达式>, <长度>)	返回字符表达式右侧起提取指定长度的字符串
SUBSTR(<字符表达式>, <起始位置>[, <长度>])	返回字符表达式从左侧起始位置开始的指定长度的字符串，如果没有长度参数，则提取字符表达式从左侧起始位置后所有字符作为函数值返回
OCCURS(<字符表达式 1>, <字符表达式 2>)	返回第一个字符串在第二个字符串中出现的次数，函数值为数值型。若第一个字符串不是第二个字符串的子串，函数值为 0
AT(<字符表达式 1>, <字符表达式 2>[, <数值表达式>])	如果<字符表达式 1>是<字符表达式 2>的子串，则返回<字符表达式 1>的首字符在<字符表达式 2>中首次出现的位置；若不是子串，函数返回 0。如果带<数值表达式>参数，则标明要<字符表达式 1>在<字符表达式 2>中第几次出现时的位置。函数值为数值型
ATC(<字符表达式 1>, <字符表达式 2>[, <数值表达式>])	功能与函数 AT 功能基本相同，其在子串比较时不区分大小写
STUFF(<字符表达式 1>, <起始位置>, <字符表达式 2>)	用<字符表达式 2>值替换<字符表达式 1>中由<起始位置>和<长度>指明的一个子串，替换和被替换的字符个数不一定相等。如果<长度>为 0，则<字符表达式 2>插入在<字符表达式 1>的<起始位置>前面。如果<字符表达式 2>是空串，则相当于在<字符表达式 1>中删去由<起始位置>和<长度>指明的一个子串。

续表

格式	功能说明
CHRTAN(<字符表达式 1>， <字符表达式 2>， <字符表达式 3>)	当<字符表达式 1>中一个或多个相同字符与<字符表达式 2>中的某个字符相匹配时，使用<字符表达式 3>中对应字符（与<字符表达式 2>中的那个字符具有相同位置）替换<字符表达式 1>中的与<字符表达式 2>中的对应字符。注意：这里是对应位置的字符，不是对应位置的字符串。如果<字符表达式 3>中包含的字符个数少于<字符表达式 2>中包含的字符个数，导致没有对应字符，则删除<字符表达式 1>中相匹配字符。如果<字符表达式 3>中包含的字符个数多于<字符表达式 2>中包含的字符个数，多于字符被忽略
LIKE(<字符表达式 1>， <字符表达式 2>)	比较两个字符串对应位置字符是否匹配，是则返回逻辑真，否则返回逻辑假。注意：<字符表达式 1>中可以包含通配符

6.3.3 日期时间处理函数

日期和时间函数的自变量一般是日期表达式或日期时间表达式，但是函数值不一定是日期型或日期时间型数据。常见的日期时间函数如表 6-11 所示。

表 6-11 日期和时间函数

格式	功能说明
DATA()	返回当前系统日期，函数值为日期型
TIME()	返回当前系统时间，函数值为字符型
DATETIME()	返回当前系统日期时间，函数为日期时间型
YEAR(<日期表达式> <日期时间表达式>)	从指定的日期表达式或者日期时间表达式中返回年份，函数值为数值型
MONTH(<日期表达式> <日期时间表达式>)	从指定的日期表达式或者日期时间表达式中返回月份，函数值为数值型
DAY(<日期表达式> <日期时间表达式>)	从指定的日期表达式或者日期时间表达式中返回月份里面的天数，函数值为数值型
HOURL(<日期时间表达式>)	从指定的日期时间表达式中返回小时部分（24 小时制），函数值为数值型
MINUTE(<日期时间表达式>)	从指定的日期时间表达式中返回分钟部分，函数值为数值型
SEC(<日期时间表达式>)	从指定的日期时间表达式中返回秒数部分，函数值为数值型

6.3.4 数据类型转换函数

数据类型转换函数可以将某一种类型的数据转换为另一种类型的数据。常见的数据类型转换函数如表 6-12 所示。

表 6-12 数据类型转换函数

格式	功能说明
STR(<数值表达式>[, <长度>[, <小数位数>]])	将<数值表达式>的值转换为字符串，转换时根据需要进行四舍五入
VAL(<字符表达式>)	将由数字符号组成的字符型数据转换成相应的数值型数据，直到非数字字符
CTOD(<字符表达式>)	将<字符表达式>转换为日期型数据
CTOT(<字符表达式>)	将<字符表达式>转换为日期时间型数据
DTOC(<日期表达式> <日期时间表达式>[, 1])	将日期型数据或者日期时间型数据的日期部分转换成字符串
TTOC(<日期时间表达式>[, 1])	将日期时间数据转换成字符串
&<字符型变量>[.]	替换出字符型变量的内容，即函数值是变量中的字符串。如果该函数与其后面的字符无明确分界，则要用“.”作为函数结束标识。该宏替换函数可以嵌套使用

6.3.5 测试函数

常见的测试函数如表 6-13 所示。

表 6-13 测试函数

格式	功能说明
BETWEEN(<表达式 T>, <表达式 L>, <表达式 H>)	判断一个表达式的值是否介于两个表达式的值之间。当<表达式 T>值大于或者等于<表达式 L>且小于或者等于<表达式 H>时，函数值为逻辑真，否则为逻辑假
VARTYPE(<表达式>[, <逻辑表达式>])	测试<表达式>的类型，返回一个大写字母，函数值为字符型。若<表达式>的运算结果是 NULL 值，则函数根据<逻辑表达式>值决定是否返回<表达式>的类型：.T.则返回<表达式>的类型，.F.或默认，则返回 X 表明运算结果是 NULL 值
ISNULL(<表达式>)	判断表达式的运算结果是否为 NULL 值，如果是则返回逻辑真，否则返回逻辑假
EMPTY(<表达式>)	测试空值，如果表达式结果为“空”值，则返回逻辑真，否则返回逻辑假
EOF([<工作区号> <表别名>])	检测指定表文件中记录指针是否指向文件尾，如果是则返回真，否则返回假
BOF([<工作区号> <表别名>])	测试当前表文件或指定表文件中的记录指针是否指向文件首，若是则返回真，否则返回假
RECNO([<工作区号> <表别名>])	返回当前表或者指定表的当前记录号
RECCOUNT([<工作区号> <表别名>])	返回当前表或者指定表的记录个数
IIF(<逻辑表达式>, <表达式 1>, <表达式 2>)	测试<逻辑表达式>的值，如果为真则函数返回<表达式 1>的值；若为逻辑假，则函数返回<表达式 2>的值
DELETED([<工作区号> <表别名>])	测试指定的表当前记录是否有删除标记，如果有则返回真，否则返回假

其中 VARTYPE 函数中，字母的含义如表 6-14 所示。

表 6-14 VARTYPE 函数值所代表的数据类型

字母	数据类型	字母	数据类型
C	字符型或备注型	G	通用型
N	数值型、整型、浮点型或双精度型	D	时期型
Y	货币型	T	日期时间型
L	逻辑型	X	NULL 值
O	对象型	U	未定义

其中 EMPTY 函数中，不同类型表达式的空值如表 6-15 所示。

表 6-15 EMPTY 函数不同类型的空值

数据类型	空值
数值型	0
字符型	空串、空格、回车、换行
日期型、日期时间型	空日期
逻辑型	.F.

6.4 程序与程序文件

计算机对数据的处理过程中，许多任务无法仅靠一条命令来完成，而是需要一组命令来完成。如果采用在命令窗口中直接逐条输入命令进行交互式操作，十分麻烦而且非常容易出错，甚至根本无法完成较为复杂和较大的任务。因此，采用程序的方式是一种必然的选择。

6.4.1 程序的概念

程序是能够完成一定任务的命令的有序集合。这组命令被存放在称为程序文件或命令文件的文本文件中，当用户运行程序时，系统会按照一定的次序自动执行包含在程序文件中的命令。程序可以多次反复运行，比命令交互操作更有效率。

所谓程序方式，就是先根据任务的要求确定能完成该任务的命令序列，即编写程序；然后在磁盘上建立包含程序代码的程序文件；最后通过运行程序，让系统自动执行程序代码。

与在命令窗口逐条输入命令的交互式操作方式相比，采用程序方式有如下好处：

- (1) 可以利用编辑器，方便地输入、修改和保存程序。
- (2) 可以用多种方式、多次运行程序。
- (3) 可以在一个程序中调用另一个程序。
- (4) 在程序中可以出现在命令窗口中无法使用的命令和语句。

下面给出一个程序示例，结合这个具体的例子来了解一下 Visual FoxPro 的程序。

```
*功能说明：求圆的面积
*文件名：area.prg
CLEAR                                &&清除主窗口里的全部内容
SET TALK OFF
INPUT "请输入圆的半径，半径=" TO Radius    &&请用户输入半径
Area=Pi()*Radius^2                        &&计算圆面积
? "圆的面积=", Area
SET TALK ON
```

从上面的程序例子中可以看到，一个 Visual FoxPro 的程序是由若干有序的命令行组成，并且满足以下要求：

- (1) 命令分行。程序中的每条命令都以回车结尾，一行只能写一条命令。如果需要分行书写，应在一行结束时输入续行符“;”，然后再按回车键。
- (2) 命令注释。为了提高程序的可读性，需要在程序中加入注释。注释是非执行代码，不会影响程序的功能。在 Visual FoxPro 中，采用以下两种注释方式：
 - * <注释内容>或者 NOTE <注释内容>。这种注释方式一般用于对下面一段命令代码进行说明。
 - &&<注释内容>。这种注释方式放置在命令行的尾部，对所在行的命令代码进行说明。
- (3) SET TALK ON|OFF 命令。用于显示和隐藏一些命令的执行信息。

6.4.2 程序文件的建立与执行

命令文件的建立与修改一般是通过调用系统内置的文件编辑器来进行的，也可以使用其他编辑器来完成。命令文件的扩展名为.prg。

1. 程序文件的建立与修改

建立程序文件的步骤如下：

- (1) 在主菜单“文件”菜单中选择“新建”命令，打开“新建”对话框，如图 6-2 所示。
- (2) 在“新建”对话框中选择“程序”选项，单击“新建文件”按钮，打开“文本编辑窗口”，如图 6-3 所示。
- (3) 在文本编辑窗口中输入程序内容，如图 6-4 所示。此处所输入的程序命令不会马上被执行。
- (4) 在主菜单“文件”菜单中选择“保存”命令或者按【Ctrl+W】组合键，在弹出的“另存为”对话框中指定程序文件的存放位置和名称，单击“保存”按钮保存程序文件，如图 6-5 所示。程序文件的默认扩展名为“.prg”。



图 6-2 “新建”对话框



图 6-3 “文本编辑”窗口



图 6-4 在“文本编辑”窗口中输入程序内容

打开或修改程序文件的步骤如下：

(1) “文件”菜单中选择“打开”命令，弹出“打开”对话框，如图 6-6 所示。



图 6-5 “另存为”对话框

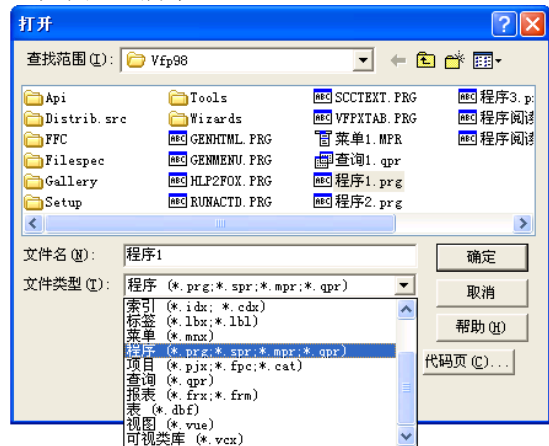


图 6-6 “打开”对话框

(2) 在“打开”对话框中，在“文件类型”下拉列表框中选择“程序”选项，在文件列表框中选择需要修改的文件，单击“确定”按钮。

(3) 编辑修改后，在主菜单“文件”菜单中选择“保存”命令或按【Ctrl+W】组合键保存程序文件；若要放弃修改，可直接按【Esc】键，或从主菜单“文件”菜单中选择“还原”命令。

在 Visual FoxPro 中，也可以用命令方式建立和修改程序文件。

命令格式为：MODIFY COMMAND <程序文件名>

这里，在程序文件名前可以指定保存文件的路径。如果没有给定扩展名，系统会自动加上默认扩展名.prg。

2. 执行程序文件

程序文件建立后，可以使用多种方式、多次执行它，常用的方式有以下两种：

1) 菜单方式

(1) 在主菜单“程序”菜单中选择“运行”命令，弹出“运行”对话框，如图 6-7 所示。

(2) 在“运行”对话框中，选择需要运行的程序文件，单击“运行”按钮。

2) 命令方式



图 6-7 “运行”对话框

命令格式：DO <文件名>。

该命令既可以在命令窗口中发出，也可以出现在某个程序文件中，这样就使得一个程序在执行过程中还可以调用执行另一个程序。

当用 DO 命令执行程序文件时，如果没有指定扩展名，系统将按下列顺序寻找该程序文件的源代码或某种目标代码文件执行：.exe、.app、.fxp、.prg。

如果用 DO 命令执行查询文件、菜单文件，那么<文件名>中必须要包括扩展名.qpr、.mpr。

当程序文件被执行时，文件中包含的命令将被依次执行，直到所有的命令被执行完毕，或者当执行到以下命令时，程序运行会发生变化。

- CANCEL：终止程序运行，释放所有私有变量，返回命令窗口。
- DO：执行另外一个程序。
- RETURN：结束当前程序的执行，返回到调用它的上级程序，如果没有上级程序则返回到命令窗口。
- QUIT：退出 Visual FoxPro，返回操作系统。

6.4.3 简单的输入/输出命令

一个程序一般都包含数据输入、数据处理和数据输出 3 个部分。

1. INPUT 命令

格式：INPUT [<字符表达式>] TO <内存变量>。

功能：该命令等待用户从键盘输入数据，当用户按回车键后，系统将表达式的值存入指定的内存变量，程序继续运行。

注释：

- 如果选用<字符表达式>，那么系统会首先显示该表达式的值，作为提示信息。
- 输入的数据可以是常量、变量，也可以是一般的表达式，但是不能不输入任何内容而直接按回车键。
- 在输入字符时必须加定界符，输入逻辑型常量时要用圆点定界（如.T.、.F.），输入日期时间型常量时要用大括号（如{^2001-03-01}）。

2. ACCEPT 命令

格式：ACCEPT [<字符表达式>] TO <内存变量>。

功能：该命令等待用户从键盘输入字符串，当用户按回车键后，系统将表达式的值存入指定的内存变量，程序继续运行。

注释：

- 如果选用<字符表达式>，那么系统会首先显示该表达式的值，作为提示信息。
- 该命令只能接收字符串。用户在输入字符串时不需要定界符；否则，系统会把定界符作为字符串本身的一部分。
- 如果不输入任何内容而直接按回车键，则系统会把空串赋给指定的内存变量。

3. WAIT 命令

格式：WAIT [<字符表达式>] [TO<内存变量>][WINDOW[AT<行>，<列>]]
[NOWAIT][CLEAR|NOCLEAR][TIMEOUT<数值表达式>]。

功能：该命令显示字符表达式的值作为提示信息，暂停程序的执行，直到用户按任意键或单击鼠标时才继续运行程序。

注释：

- 如果<字符表达式>值为空串，那么不会显示任何提示信息。如果没有指定<字符表达式>，则显示默认的提示信息“按任意键继续…”。
- <内存变量>用来保存用户输入的字符，其类型为字符型。若用户按了【Enter】键或单击了鼠标，那么<内存变量>

>中保存的将是空串。若不选 TO <内存变量>短语，则输入的单字符不保留。

- 在一般情况下，提示信息被显示在 Visual FoxPro 主窗口或当前用户自定义窗口里。如果指定了 WINDOW 子句，则会出现一个 WAIT 提示窗口，用以显示提示信息。提示窗口一般定位于主窗口的右上角，也可用 AT 短语指定其在主窗口中的位置。
- 若同时选用 NOWAIT 短语和 WINDOW 子句，系统将不等待用户按键，直接往下执行。
- 若选用 NOCLEAR 短语，则不关闭提示窗口，直到用户执行下一条 WAIT WINDOW 命令或 WAIT CLEAR 命令为止。
- TIMEOUT 子句用来设定等待时间（秒数）。一旦超时就不再等待用户按键，自动往下执行。

ACCEPT、INPUT 和 WAIT 这 3 个命令的区别如表 6-16 所示。

表 6-16 ACCEPT、INPUT 和 WAIT 这 3 个命令的区别

命令	接收的数据类型	数据的数量	是否需要回车结束输入
ACCEPT	字符型	任意个	是
INPUT	数值型、字符型、日期型、逻辑型	任意个	是
WAIT	字符型	1 个	否

6.5 程序的基本结构

程序结构是指程序中命令或语句执行的流程结构，包括 3 种基本结构：顺序结构、选择结构和循环结构。

顺序结构是最简单的程序结构，它按命令在程序中出现的先后次序依次执行。但是，绝大多数问题仅用顺序结构是无法解决的，还要用到选择结构和循环结构。Visual FoxPro 提供了相应的语句支持选择结构和循环结构的实现。

6.5.1 选择结构

支持选择结构的语句包括条件语句（IF-ENDIF）和分支语句（DO CASE-ENDCASE）。

1. 简单形式的条件语句

格式：

```
IF <条件>
  <语句序列 1>
ENDIF
```

功能：语句执行时，首先计算<条件>表达式的值。如果<条件>成立，则执行<语句序列 1>，然后转向 ENDIF 的下一条语句；否则直接转向 ENDIF 的下一条语句去执行。

例：编写程序，判断输入的数字是奇数还是偶数。

```
INPUT " 请输入数据： " TO number
IF number%2=0
  ? " 该数为偶数。 "
ENDIF
? " 该数为奇数。 "
```

2. 一般形式的条件语句

格式：

```
IF <条件>
  <语句序列 1>
```



```
ELSE
    <语句序列 2>
ENDIF
```

功能：当语句执行时，首先计算<条件>表达式的值。该语句根据<条件>是否成立从两组代码中选择一组执行。

注释：

- 当有 ELSE 子句时，两组可供选择的代码分别是<语句序列 1>和<语句序列 2>。如果<条件>成立，则执行<语句序列 1>；否则，执行<语句序列 2>。然后转向 ENDIF 的下一条语句。
- 无 ELSE 子句时，可看做第二组代码不包含任何命令。如果<条件>成立，则执行<语句序列 1>，然后转向 ENDIF 的下一条语句；否则直接转向 ENDIF 的下一条语句去执行。
- IF 和 ENDIF 必须成对出现，IF 是本结构的入口，ENDIF 是本结构的出口。
- 条件语句可以嵌套，但不能出现交叉。在嵌套时，为了使程序清晰，易于阅读，可按缩进格式书写。

例：编写程序，判断输入的数字是奇数还是偶数。

```
INPUT "请输入数据：" TO number
IF number%2=0
    ? "该数为偶数。"
ELSE
    ? "该数为奇数。"
ENDIF
```

例：编写程序，根据输入考试成绩，显示相应的成绩等级。

```
CLEAR
INPUT "请输入考试成绩：" TO Result
IF Result<60
    Grade= "不及格"
ELSE
    IF Result<90
        Grade= "通过"
    ELSE
        Grade= "优秀"
    ENDIF
ENDIF
```

3. 多分支语句

格式：

```
DO CASE <条件 1>
    <语句序列 1>
CASE <条件 2>
    <语句序列 2>
    .....
CASE <条件 n>
    <语句序列 n>
    [OTHERWISE <语句序列>]
ENDCASE
```

功能：当语句执行时，依次判断 CASE 后面的条件是否成立。分支语句实现的是一种扩展的选择结构，可以根据条件从多组代码中选择一组执行。当发现某个 CASE 后面的条件成立时，就执行该 CASE 和下一个 CASE 之间的命令序列，然后执行 ENDCASE 后面的命令。如果所有的条件都不成立，则执行 OTHERWISE 与 ENDCASE 之间的命令序列，然后转向 ENDCASE 后面的语句。

注释：

- 不管有几个 CASE 条件成立，只有最先成立的那个 CASE 条件的对应命令序列被执行。
- 如果所有的 CASE 条件都不成立，且没有 OTHERWISE 子句，则直接跳出本结构。
- DO CASE 和 ENDCASE 必须成对出现，DO CASE 是本结构的入口，ENDCASE 是本结构的出口。

例：编写程序，用来显示出当前季节。

```
CLEAR
M=MONTH(DATE())
DO CASE
    CASE M=3 OR M=4 OR M=5
        Season=" 春 "
    CASE M=6 OR M=7 OR M=8
        Season=" 夏 "
    CASE M=9 OR M=10 OR M=11
        Season=" 秋 "
    CASE M=12 OR M=1 OR M=2
        Season=" 冬 "
ENDCASE
```

? “当前的季节是：”，Season

6.5.2 循环结构

循环结构是在满足一定的条件下可以重复执行的程序结构，也称为重复结构。被重复执行的语句序列称为“循环体”。循环结构可以使程序简短，提高程序设计效率。

Visual FoxPro 提供了 3 种循环结构语句：DO WHILE-ENDDO、FOR-ENDFOR 和 SCAN-ENDSCAN。

1. DO WHILE-ENDDO 语句

格式：

```
DO WHILE<条件>
    <语句序列 1>
    [LOOP]
    <语句序列 2>
    [EXIT]
    <语句序列 3>
ENDDO
```

功能：首先判断循环条件是否为真，如果为真则执行 DO WHILE 和 ENDDO 之间的命令序列，如果遇到 LOOP 语句，则回到 DO WHILE 处继续执行，如果遇到 EXIT 语句，则退出循环体；当执行到 ENDDO 时，再次返回 DO WHILE 处判断条件，一直到条件为假时，退出循环体，执行 ENDDO 后面的语句。

注释：

- 如果第一次判断条件时，条件即为假，则循环体一次都不执行。
- 如果循环体包含 LOOP 命令，那么当遇到 LOOP 时，就结束循环体的本次执行，不再执行其后面的语句，而是转回 DO WHILE 处重新判断条件。
- 如果循环体包含 EXIT 命令，那么当遇到 EXIT 时，就结束该语句的执行，转去执行 ENDDO 后面的语句。
- 通常 LOOP 或 EXIT 出现在循环体内嵌套的选择语句中，根据条件来决定是 LOOP 回去，还是 EXIT 出去。

例：编写程序，计算 1~100 的累加和。

```
CLEAR
```

```
SUM=0
I=1
DO WHILE I<=100
    SUM=SUM+I
    I=I+1
ENDDO
?SUM
```

2. FOR-ENDFOR 语句

格式：

```
FOR <循环变量>=<初值> TO<终值> [STEP<步长>]
    <循环体>
ENDFOR
```

功能：首先将初值赋给循环变量，然后判断循环条件是否成立（若步长为正值，则循环条件为<循环变量>=<终值>;若步长为负值，则循环条件为<循环变量>=<终值>）。如果成立，则执行循环体，然后循环变量增加一个步长值，并再次判断循环条件是否成立，以确定是否再次执行循环体。一直到条件为假时，退出循环体，执行 ENDFOR 后面的语句。该语句通常用于实现循环次数已知情况下的循环结构。

注释：

- <步长>的默认值为 1。
- <初值>、<终值>和<步长>都可以是数值表达式，但这些表达式仅在循环语句执行开始时被计算一次。在循环语句的执行过程中，初值、终值和步长是不会改变的。
- 可以在循环体内改变循环变量的值，但这会影响循环体的执行次数。

EXIT 和 LOOP 命令同样可以出现在该循环语句的循环体内。当执行到 LOOP 命令时，结束循环体的本次执行，然后循环变量增加一个步长值，并再次判断循环条件是否成立。

例：编写程序，求 1~100 中偶数的和。

```
CLEAR
SUM=0
FOR I=2 TO 100 STEP 2
    SUM=SUM+I
ENDFOR
?SUM
```

3. SCAN-ENDSCAN 语句

格式：

```
SCAN[<范围>][FOR<条件 1>][WHERE<条件 2>]
    <循环体>
ENDSCAN
```

功能：该循环语句一般用于处理表中的记录，语句可指明需要处理的记录范围及应满足的条件。当执行该语句时，记录指针自动、依次地在当前表的指定范围内满足条件的记录上移动，对每一条记录执行循环体内的命令。

注释：

- <范围>的默认值是 ALL。
- EXIT 和 LOOP 命令同样可以出现在该循环语句的循环体内。

例：编写程序，逐条显示 Stu 表中女生的记录。

```
CLEAR
USE Stu
```

```
SCAN FOR Sex= " 女 "  
    DISPLAY  
    WAIT  
ENDSCAN
```

6.6 多模块程序设计

应用程序一般都是多模块程序，包含多个程序模块。模块是一个相对独立的程序段，它可以被其他模块调用，也可以调用其他模块。通常，把被其他模块调用的模块称为子程序，把调用其他模块而没有被其他模块调用的模块称为主程序。

将一个应用程序划分成一个个功能相对简单、单一的模块程序，不仅便于程序的开发，也利于程序的阅读和维护。

6.6.1 模块的定义和调用

在 Visual FoxPro 中，模块可以是命令文件，也可以是过程。

1. 模块及其定义

过程定义的语句格式为：

```
PROCEDURE|FUNCTION <过程名>  
    <命令序列>  
    [RETURN[<表达式>]]  
[ENDPROC|ENDFUNC]
```

功能说明如下：

- PROCEDURE|FUNCTION 命令表示一个过程的开始，并命名过程。过程名必须以字母或下画线开头，可以包含数字、字母和下画线。
- 当程序执行到 RETURN 命令时，控制将转回到调用程序（或命令窗口），并返回表达式的值。若默认 RETURN 命令，系统将在过程结束处自动调用一条隐含的 RETURN 命令。若 RETURN 命令不带<表达式>，则返回逻辑真.T。
- ENDPROC|ENDFUNC 命令表示一个过程的结束。如果默认此语句，那么过程结束于下一条 ENDPROC|ENDFUNC 命令或文件结尾处。

2. 过程文件的打开

过程可以放置在程序文件代码的后面，也可以保存在称为过程文件的单独文件里。过程文件的建立仍使用 MODIFY COMMAND 命令，文件的默认扩展名还是.prg。

过程文件里只包含过程，这些过程能被任何其他程序所调用。但在调用过程文件中的过程之前首先要打开过程文件。

可以打开一个或多个过程文件。一旦一个过程文件被打开，那么该过程文件中的所有过程都可以被调用。如果选用 ADDITIVE，那么在打开过程文件时，并不关闭原先已打开的过程文件。

命令格式如下：

```
SET PROCEDURE TO[<过程文件 1>[, <过程文件 2>, ...]][ADDITIVE]
```

3. 过程文件的关闭

当使用不带任何文件名的 SET PROCEDURE TO 命令时，将关闭所有打开的过程文件。

如果不想一并关闭所有的过程文件，而是关闭个别过程文件，则命令格式如下：

```
RELEASE PROCEDURE [<过程文件 1>[, <过程文件 2>, ...]]
```

4. 模块的调用

过程调用的格式有两种。

(1) 使用 DO 命令，格式如下：

```
DO <文件名>|<过程名>
```

(2) 在名字后加一对小括号，格式如下：

```
<文件名>|<过程名> ()
```

在上面的两种格式里，如果过程是程序文件的代码，则用<文件名>；否则用<过程名>。

格式 2 既可以作为命令使用（返回值被忽略），也可以作为函数出现在表达式里。在这里，<文件名>不能包含扩展名。

存放在命令文件里的过程主要被命令文件中的代码所调用，但也可以被其他程序所调用。当命令文件处于执行（打开）状态时，包含在其中的过程就可以被直接调用；如果命令文件不处于打开状态，那么要调用其中的过程，就需要用 SET PROCEDURE 命令先打开此命令文件。这与打开过程文件的道理是一样的。

总而言之，我们这里讨论的模块主要是指过程和命令文件里的代码。而过程的调用则需要它所在的文件处于打开状态。

6.6.2 参数传递

模块程序可以接收调用程序传递过来的参数，并能够根据接收到的参数控制程序流程或对接收到的参数进行处理，从而大大提高模块程序功能设计的灵活性。

接收参数的命令有 PARAMETERS 和 LPARAMETERS，它们的格式如下：

```
PARAMETERS <形参变量 1>[, <形参变量 2>]
LPARAMETERS <形参变量 1>[, <形参变量 2>]
```

PARAMETERS 命令声明的形参变量被看做是模块程序中建立的私有变量，LPARAMETERS 命令声明的形参变量被看做是模块程序中建立的局部变量。除此之外，两条命令没有什么不同。

不管是 PARAMETERS 命令还是 LPARAMETERS 命令，都应该是模块程序的第一条可执行命令。

相应地，调用模块程序的格式为：

```
格式 1: DO <文件名>|<过程名> WITH <实参 1>[, <实参 2>, ...]
格式 2: <文件名>|<过程名> (<实参 1>[, <实参 2>, ...])
```

实参可以是常量、变量，也可以是一般形式的表达式。当调用模块程序时，系统会自动把实参传递给对应的形参。形参的数目不能少于实参的数目，否则系统会产生运行时错误。如果形参的数目多于实参的数目，那么多余的形参取初值逻辑假.F。

当采用格式 1 调用模块程序时，如果实参是常量或一般形式的表达式，系统会计算出实参的值，并把它们赋值给相应的形参变量。这种情形称为按值传递。如果实参是变量，那么传递的将不是变量的值，而是变量的地址。这时形参和实参实际上是同一个变量（尽管它们的名称可能不同），在模块程序中对形参变量值的改变，同样是对实参变量值的改变。这种情形称为按引用传递。

采用格式 2 调用模块程序时，在默认情况下都以按值方式传递参数。如果实参是变量，可以通过 SET UDFPARMS 命令重新设置参数传递的方式。该命令的格式如下：

```
SET UDFPARMS TO VALUE | REFERENCE
```

- TO VALUE: 按值传递。形参变量值的改变不会影响实参变量的取值。
- TO REFERENCE: 按引用传递。在形参变量值改变时，实参变量值也随之改变。

6.6.3 变量的作用域

变量的一个重要属性是作用域。变量的作用域指的是变量在什么范围内有效或能够被访问。在 Visual FoxPro 中，可

分为公共变量、私有变量和局部变量 3 类。

1. 公共变量

在任何模块中都可使用的变量称为公共变量，公共变量必须先定义后使用，格式如下：

PUBLIC <内存变量表>

说明：该命令用于建立公共的内存变量，并赋以初值逻辑假.F。建立好的公共变量一直有效，即使程序结束返回命令窗口也不会消失，只有执行 CLEAR MEMORY、RELEASE 或 QUIT 等命令后，才能被释放。

在命令窗口中直接使用而由系统自动隐含建立的变量也是公共变量。

2. 私有变量

在程序中直接使用（没有预先声明，是指没有通过 PUBLIC 和 LOCAL 命令事先声明），而由系统自动隐含建立的变量都是私有变量。

私有变量的作用域是建立它的模块及其下属的各层模块。一旦建立它的模块程序运行结束，这些私有变量将自动清除。

3. 局部变量

局部变量只能在建立它的模块中使用，不能在上层或下层模块中使用。当程序结束时，自动释放局部变量。格式如下：

LOCAL <内存变量表>

说明：该变量同样给建立的内存变量赋以初值逻辑假.F，也必须先定义后使用。

当子程序中定义的变量与主程序中的变量同名时，为避免子程序中的变量值改变主程序中的变量值，可以在子程序中使用 PRIVATE 命令隐藏主程序中可能存在的变量，使这些变量在子程序中暂时无效。格式如下：

- PRIVATE <内存变量表>
- PRIVATE ALL[LIKE<通配符>|EXECEPT<通配符>]

说明：该命令并不建立内存变量，它只是隐藏指定的在上层模块中可能已经存在的内存变量，但是当模块程序返回上层模块时，被隐藏的变量就自动恢复有效性，保持原值。

6.7 本章习题

6.7.1 选择题

1. Visual FoxPro 内存变量的数据类型不包括（ ）。
A. 数值型 B. 货币型 C. 备注型 D. 逻辑型
2. 在 Visual FoxPro 中说明数组的命令是（ ）。
A. DIMENSION 和 ARRAY B. DECLARE 和 ARRAY
C. DIMENSION 和 DECLARE D. 只有 DIMENSION
3. 关于 Visual FoxPro 的变量，下面说法中正确的是（ ）。
A. 在使用一个简单变量之前要先声明或定义 B. 数组中各元素的数据类型可以不同
C. 在定义数组后，系统为数组的每个数组元素赋以数值 0 D. 数组元素的下标下限为 0
4. 执行如下命令序列后，最后一条命令的显示结果是（ ）。
DIMENSION M(2,2)
M(1,1)=10
M(1,2)=20
M(2,1)=30
M(2,2)=40
? M(2)

- A. 变量未定义的提示 B. 10 C. 20 D. .F.

5. 在 Visual FoxPro 中, 下面 4 个关于日期或日期时间的表达式中, 错误的是 ()。

A. {^2002.1.3 10: 0: 0 AM}-{^2002.1.3 10: 0: 0 AM}

B. {^01/01/2002} “+” 20

C. {^2001.1.3}+{^2002.1.3 }

D. {^2001.1.3}-{^2002.1.3 }

6. 设 X="11", Y="1122", 下列表达式结果为假的是 ()。

A. NOT (X= Y) AND (X\$Y)

B. NOT (X\$Y) OR (X◇Y)

C. NOT (X>=Y)

D. NOT (X\$Y)

7. 在下面的 Visual FoxPro 表达式中, 运算结果为逻辑真的是 ()。

A. EMPTY (.NULL.) B. LIKE (' xy?', ' xyz')

C. AT (' xy', ' abcxyz') D. ISNULL (SPACE (0))

8. 下列表达式中, 表达式返回结果为.F.的是 ()。

A. AT ("A", "BCD")

B. "[信息]"\$ "管理信息系统"

C. ISNULL (.NULL.) D. SUBSTR ("计算机技术", 3, 2)

9. 设 X=10, 语句? VARTYPE ("X") 的输出结果是 ()。

A. N

B. C

C. 10

D. X

10. 在 Visual FoxPro 中, 宏替换可以从变量中替换出 ()。

A. 字符串

B. 数值

C. 命令

D. 以上 3 种都可能

11. 设 X="11", Y="1122", 下列表达式结果为假的是 ()。

A. NOT (X==Y) AND (X\$Y)

B. NOT (X\$Y) OR (X◇Y)

C. NOT (X>=Y)

D. NOT (X\$Y)

12. 执行如下命令序列后, 最后一条命令的显示结果是 ()。

```
DIMENSION M(2,2)
```

```
M(1,1)=10
```

```
M(1,2)=20
```

```
M(2,1)=30
```

```
M(2,2)=40
```

```
? M(2)
```

A. 变量未定义的提示

B. 10

C. 20

D. .F.

13. 依次执行以下命令后的输出结果是 ()。

```
SET DATE TO YMD
```

```
SET CENTURY ON
```

```
SET CENTURY TO 19 ROLLOVER 10
```

```
SET MARK TO "."
```

```
? CTOD ("49-05-01")
```

A. 49.05.01

B. 1949.05.01

C. 2049.05.01

D. 出错

14. 假设职员表已在当前工作区打开, 其当前记录的“姓名”字段值为“张三”(字符型, 宽度为 6)。在命令窗口输入并执行如下命令:

```
姓名=姓名 "您好"
```

```
?姓名
```

那么主窗口中将显示 ()。

A. 张三

B. 张三 您好

C. 张三您好

D. 出错

15. 下列表达式中, 表达式返回结果为.F.的是 ()。

A. AT ("A", "BCD")

B. "[信息]"\$ "管理信息系统"

C. ISNULL (.NULL.)

D. SUBSTR ("计算机技术", 3, 2)

16. 表达式 LEN (SPACE (0)) 的运算结果是 ()。

A. NULL

B. 1

C. 0

D. ""

17. 命令? VARTYPE(TIME())的结果是 ()。

A. C

B. D

C. T

D. 出错

18. 命令? LEN(SPACE(3)-SPACE(2))的结果是 ()。

- A. 1 B. 2 C. 3 D. 5

19. 想要将日期型或日期时间型数据中的年份用 4 位数字显示, 应当使用设置命令 ()。

- A. SET CENTURY ON B. SET CENTURY OFF C. SET CENTURY TO 4 D. SET CENTURY OF 4

20. 有如下赋值语句: a= " 计算机 "、b= " 微型 " , 结果为 “微型机” 的表达式是 ()。

- A. b+LEFT(a,3) B. b+RIGHT(a,1) C. b+LEFT(a,5,2) D. b+RIGHT(a,2)

21. 在 Visual FoxPro 中, 有如下内存变量赋值语句:

```
X={^2001-07-28 10:15:20 PM}
Y=.F.
M=$123.45
N=123.45
Z= " 123.24 "
```

执行上述赋值语句之后, 内存变量 X、Y、M、N 和 Z 的数据类型分别是 ()。

- A. D、L、Y、N、C B. T、L、Y、N、C C. T、L、M、N、C D. T、L、Y、N、S

22. 如果内存变量和字段变量均有变量名 “姓名”, 那么引用内存的正确方法是 ()。

- A. M.姓名 B. M->姓名 C. 姓名 D. A 和 B 都可以

23. 下列程序段执行后, 内存变量 s1 的值是 ()。

```
s1= " network "
s1=stuff(s1,4,4, " BIOS " )
```

- A. network B. netBIOS C. net D. BIOS

24. 说明数组后, 数组元素的初值是 ()。

- A. 整数 0 B. 不定值 C. 逻辑真 D. 逻辑假

25. 设 a= " 计算机等级考试 " , 结果为 " 考试 " 的表达式是 ()。

- A. Left(a,4) B. Right(a,4) C. Left(a,2) D. Right(a,2)

26. 下列程序段的输出结果是 CLOSE DATA ()。

```
a=0
USE 教师 GO TOP
DO WHILE .NOT. EOF()
    IF 主讲课程="数据结构".OR.主讲课程="C 语言"
        a=a+1
    ENDIF
    SKIP
ENDDO
?a
```

- A. 4 B. 5 C. 6 D. 7

27. 使用 “调试器” 调试程序时, 用于显示正在调试的程序文件的窗口是 ()。

- A. 局部窗口 B. 跟踪窗口 C. 调用堆栈窗口 D. 监视窗口

28. 在 Visual FoxPro 中, 用来建立程序文件的命令是 ()。

- A. CREATE COMMAND <文件名> B. CREATE FILE <文件名>
C. MODIFY FILE <文件名> D. MODIFY COMMAND <文件名>

29. 在 Visual FoxPro 中, 程序文件的扩展名为 ()。

- A. .prg B. .qpr C. .app D. .scx

30. 在 Visual FoxPro 中, 代码片段是指 ()。

- A. 能独立运行并完成指定功能的程序 B. 一个子程序
C. 一组命令, 用于完成某一项操作 D. 一个可调用的函数

31. 下列关于 Visual FoxPro 输入/输出命令的说法不正确的是 ()。

- A. INPUT 命令用来从键盘输入数据
 B. 用 INPUT 命令输入数据时, 若不输入任何数据就直接按回车键时, 则系统会把空字符串赋给指定的内存变量
 C. ACCERT 命令只能接收字符串
 D. WAIT 命令将暂停程序执行, 直到用户按任意键或单击鼠标时继续程序
32. 设内存变量 X1 为数值型, 要从键盘输入数据给 X1 赋值, 应使用命令 ()。
 A. INPUT TO X1 B. ACCEPT TO X1 C. WAIT TO X1 D. 以上均可
33. 结构化程序设计的 3 种基本结构是 ()。
 A. 顺序结构、选择结构和模块结构 B. 选择结构、循环结构和模块结构
 C. 顺序结构、循环结构和模块结构 D. 顺序结构、选择结构和循环结构
34. 下列关于 DO CASE...ENDCASE 语句说法不正确的是 ()。
 A. DO CASE 和 ENDCASE 必须成对出现
 B. 只要 CASE 条件成立, 就执行这个 CASE 条件对应的命令序列
 C. 所有的 CASE 条件都不成立且没有 OTHERWISE 语句, 则直接跳出本结构, 执行 ENDCASE 后面的语句
 D. 在 DO CASE 和第一个 CASE 之间的任何语句都不被执行
35. 在循环结构程序设计中, 在指定范围内扫描数据文件, 查找符合条件的记录并执行循环体中的命令, 应使用的循环结构为 ()。
 A. SCAN...ENDSCAN B. FOR...ENDFOR C. DO WHILE...ENDDO D. DO CASE...ENDCASE
36. 关于过程文件下列说法错误的是 ()。
 A. 过程文件的建立使用 MODIFY COMMAND 命令
 B. 过程文件的默认扩展名为 PRG
 C. 在调用过程文件中的过程之前不必打开过程文件
 D. 过程文件只包含过程, 可以被其他程序所调用
37. 在某个程序模块中用 PRIVATE 语句定义的内存变量 ()。
 A. 可以在该程序的所有模块中使用 B. 只能在定义该变量的模块中使用
 C. 只能在定义该变量的模块及其上层模块中使用 D. 只能在定义该变量的模块及其下属模块中使用
38. 下列关于过程调用的叙述中, 正确的是 ()。
 A. 被传递的参数是变量参数, 则为引用方式 B. 被传递的参数是常量, 则为传值方式
 C. 被传递的参数是表达式, 则为传值方式
 D. 按值传递方式中形参变量值的改变不会影响实参变量的取值, 引用传递方式则刚好相反
39. 只能被本层模块调用, 而不能被上层或下层模块调用的变量类型是 ()。
 A. 公共变量 B. 私有变量 C. 局部变量 D. 局部变量和私有变量
40. 在调试器中, 可以显示程序、过程和方法程序中的变量、数组和对象的名称, 以及当前取值和类型的窗口是 ()。
 A. 跟踪 B. 监视 C. 局部 D. 调用堆栈
41. 在调试器中设置断点的类型有 ()。
 A. 2 种 B. 3 种 C. 4 种 D. 5 种
42. 下列各组语句可以将变量 A, B 的值互换的是 ()。
 A. A=B B. A=(A+B)/2 C. A=A+B D. A=C
 B=A B=(A+B)/2 B=A-B C=B
 A=A-B B=A
43. 设有如下程序段:
 A=10
 B=20
 C=40
 IF A>B

```

IF C>A
    C=A+B
ELSE
    C=A-B
ENDIF
ENDIF
?C

```

执行该程序，显示结果为（ ）。

- A. 30 B. -10 C. 10 D. 40

44. 有如下程序文件：

```

INPUT  请输入 X 的值：  TO  X
INPUT  请输入 Y 的值：  TO  Y
INPUT  请输入 Z 的值：  TO  Z

```

```

IF MAX (X, Y) <Z
    ?MAX (X,Y )
ELSE
    IF MIN (X, Y) <Z
        ? Z
    ELSE
        ? MIN(X,Y)
    EDNIF
ENDIF

```

当运行程序后，输入的 X, Y, Z 的值分别为 3, 4, 10。则程序运行的结果为（ ）

- A. 3 B. 4 C. 10 D. 3, 4, 10

45. 设成绩表当前记录中计算机字段的值为 85，执行下面程序段之后，输出结果为（ ）。

```

DO CASE
CASE 计算机<60
    ? 计算机等级是+不及格
CASE 计算机>=60
    ? 计算机等级是+及格
CASE 计算机>=75
    ? 计算机等级是+良好
CASE 计算机>=85
    ? 计算机等级是+优
ENDCASE

```

- A. 不及格 B. 及格 C. 良好 D. 优

46. 执行如下程序，如果输入 X 的值为 5，则最后 S 的显示值为（ ）。

```

SET TALK OFF
S=0
I=1
INPUT "X="TO X
DO WHILE S<=X
    S=S+1
    I=I+1
ENDDO
? S
SET TALK ON

```

- A. 1 B. 3 C. 5 D. 6

47. 有如下程序:

```
LOCAL X1
? X1
DO P1
PROCEDURE P1
X1=1
?? X1
RETURN
ENDPROC
```

执行程序的结果为 ()。

- A. .F. 1 B. 1 .F. C. .F. .F. D. 1 1

48. 设有如下程序文件:

```
SET TALK OFF
DIMENSION a(2,3)
i=1
DO WHILE i<=2
  J=1
  DO WHILE j<=3
    a(i,j)=i+j
    ??a(i,j)
    j=j+1
  ENDDO
  SET TALK ON
  RETURN
```

执行此程序, 程序的运行结果为 ()。

- A. 2 3 4 B. 1 2 3 C. 1 2 3 D. 2 3 4
3 4 5 3 4 5 2 4 6 4 5 6

49. 在 Visual Foxpro 中, 过程的返回语句是 ()。

- A. GO BACK B. COME BACK C. RETURN D. BACK

50. 下面程序计算一个整数的各位数字之和。在下画线处应填写的语句是 ()。

```
SET TALK OFF
INPUT "x=" TO x
s=0
DO WHILE x!=0
  s=s+MOD(x, 10)
  _____
ENDDO
? s
SET TALK ON
```

- A. x=int(x/10) B. x=int(x%10) C. x=x-int(x/10) D. x=x-int(x%10)

6.7.2 填空题

1. 常量 n 表示的是_____型的数据。
2. 常量 {^2009-10-01,15:30:00} 的数据类型是_____。
3. 执行命令 A=2005/4/2 之后, 内存变量 A 的数据类型是_____型。
4. 把当前表中记录的学号、姓名字段复制到数组 A 的命令是_____。
5. 表达式 {^2005-1-3 10: 0: 0}-{^2005-10-3 9: 0: 0} 的数据类型是_____。
6. 表示“1962 年 10 月 27 日”的日期常量应该写为_____。

7. 常量{^2009-10-01,15:30:00}的数据类型是_____。

8. 在 Visual Foxpro 中，如下程序的运行结果（即执行命令 DO main 后）是_____。

```
*程序文件名: main.prg
```

```
CLEAR ALL
```

```
mX="Visual Foxpro"
```

```
mY="二级"
```

```
DO s1
```

```
?mY+mX
```

```
RETURN
```

```
*子程序文件名: s1.prg
```

```
PROCEDURE s1
```

```
LOCAL mX
```

```
mX="Visual FoxPro DBMS 考试"
```

```
mY="计算机等级"+mY
```

```
RETUREN
```

9. 如下程序显示的结果是_____。

```
s=1
```

```
i=0
```

```
do while i<8
```

```
s=s+i
```

```
i=i+2
```

```
enddo
```

```
?s
```

10. 在 Visual FoxPro 中，可以使用_____语句跳出 SCAN...ENDSCAN 循环体外执行 ENDSCAN 后面的语句。

11. 执行下列程序，显示的结果是_____。

```
one= " WORK "
```

```
two= " "
```

```
a=LEN (one)
```

```
i=a
```

```
DO WHILE i>=1
```

```
Two=two+SUBSTR (one, i, 1)
```

```
i=i-1
```

```
ENDDO
```

```
? two
```

第7章

Visual FoxPro 数据库及其操作

Visual FoxPro 引入数据库的概念，数据库是一个逻辑上的概念和手段，通过一组系统文件将相互联系的数据库表及其相关的数据库系统对象统一组织和管理，使它们成为相互关联的数据集合。

本章主要介绍 Visual FoxPro 数据库的建立和操作等内容，包括建立和管理数据库、建立和使用表，以及索引和数据完整性等方面的内容。

7.1 Visual FoxPro 数据库及其建立

Visual FoxPro 的数据库是一个容器或框架，可以存放一系列数据库对象，如表、视图等。

当建立 Visual FoxPro 数据库时，数据库文件名的扩展名为“.dbc”的文件，在需要时还会自动建立一个与之相关的扩展名为“.dct”的数据库备注（memo）文件和一个扩展名为“.dcx”的数据库索引文件。这 3 个文件是 Visual FoxPro 数据库管理系统管理数据库时使用的，用户一般不能直接使用这些文件。

新建立的数据库是空的，没有数据，也不能输入数据，还需要建立数据库表和其他数据库对象，才能输入数据和实施其他数据库操作。

7.1.1 建立数据库

在 Visual FoxPro 中，可以有以下 3 种方法来建立数据库：

- （1）在项目管理器中建立数据库。
- （2）通过系统菜单的“新建”对话框建立数据库。
- （3）在命令窗口中使用命令建立数据库。

1. 在项目管理器中建立数据库

- （1）建立项目并打开项目管理器，此时建立的数据库属于这个项目。
 - （2）在“项目管理器”对话框中选择“数据”选项卡，如图 7-1 所示。从中选择“数据库”选项，单击“新建”按钮，弹出“新建数据库”对话框，如图 7-2 所示。
 - （3）单击“新建数据库”按钮，建立一个新的数据库，如图 7-3 所示。
- 此时可以看到该数据库是一个“空库”，还没有任何表和视图等数据文件。可以通过“数据库工具栏”中的相应按钮，在数据库中新建、修改或移去表等各种操作。

2. 通过系统菜单的“新建”对话框建立数据库

在主菜单“文件”菜单中选择“新建”命令，弹出“新建”对话框，在“文件类型”下拉列表框中选择“数据库”选项，单击“新建文件”按钮建立数据库。

此时建立的数据库不属于项目。

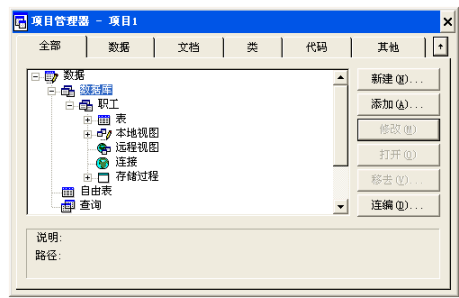


图 7-1 “项目管理器”中的“数据”选项卡



图 7-2 “新建数据库”对话框

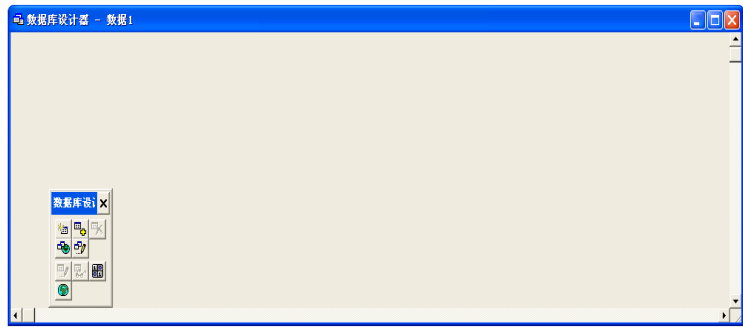


图 7-3 新建立的数据库

3. 在命令窗口中使用命令建立数据库

建立数据库的命令的格式是：CREATE DATABASE <数据库名>。
用命令方式建立的数据库不属于项目。可以通过“项目管理器”的“添加”功能将不属于项目的数据库加入到项目中。

7.1.2 使用数据库

数据库建立以后，可以在“项目管理器”中或利用命令方式对数据库进行各种操作。

1. 打开数据库

在数据库中建立表或视图，以及使用数据库中的表或视图时，都需要首先打开数据库。在 Visual FoxPro 中打开数据库的方法有以下 3 种。

- (1) 在项目管理器中打开数据库。在“项目管理器”对话框中选择“数据”选项卡，从中选择“数据库”选项，选中需要打开的数据库文件，单击“打开”按钮，则将该数据库打开。
- (2) 通过“打开”对话框打开数据库。选择主菜单“文件”菜单中“打开”命令，或者单击工具栏中“打开”按钮，弹出“打开”对话框，在“文件类型”下拉列表框中选择“数据库 (*.dbc)”选项，选中需要打开的数据库文件，单击“确定”按钮，则将该数据库打开。
- (3) 在命令窗口通过命令打开数据库。

打开数据库的命令的格式是：OPEN DATABASE <数据库名> [EXCLUSIVE][SHARED]。
使用该命令打开指定的数据库，可以通过后面的短语确定打开方式，其中，EXCLUSIVE 表示以独占的方式打开数据库，SHARED 则以共享方式打开数据库。

2. 修改数据库

用户可以在数据库设计器中完成各种数据库对象的建立、修改和删除等操作。数据库设计器是交互修改数据库对象的界面和向导。

在 Visual FoxPro 中打开数据库设计器的方法有以下 3 种。

(1) 在项目管理器中打开数据库设计器。

在“项目管理器”对话框中选择“数据”选项卡，从中选择“数据库”选项，选中需要修改的数据库文件，单击“修改”按钮，则将该数据库及数据库设计器打开。

(2) 通过“打开”对话框打开数据库设计器。选择主菜单“文件”菜单中“打开”命令，或者单击工具栏中“打开”按钮，弹出“打开”对话框，在“文件类型”下拉列表框中选择“数据库 (*.dbc)”选项，选中需要打开的数据库文件，单击“确定”按钮，则将该数据库及数据库设计器打开。

(3) 在命令窗口通过命令打开数据库设计器。

打开数据库设计器的命令的格式是：**MODIFY DATABASE <数据库名> [NOEDIT]**

打开指定数据库的设计器，以便进行修改。如果选用 NOEDIT 短语，则只打开数据库设计器，但不允许修改。

3. 关闭数据库

当关闭数据库设计器的时候，并没有把数据库关闭。如果想把数据库关闭就要使用数据库关闭命令。

关闭数据库的命令的格式是：**CLOSE DATABASE <数据库名>**

用于关闭当前打开的或指定的数据库。

4. 删除数据库

如果一个数据库不再使用时，随时都可以删除该数据库。

在 Visual FoxPro 中删除数据库的方法有以下两种。

(1) 在项目管理器中删除数据库。在“项目管理器”对话框中选择“数据”选项卡，从中选择“数据库”选项，选中需要删除的数据库文件，单击“移去”按钮，弹出删除数据库对话框，如图 7-4 所示。

- 移去：从项目管理器中删除数据库，但并不从磁盘中删除相应的数据库文件。
- 删除：从项目管理器中删除数据库，并从磁盘中删除相应的数据库文件。
- 取消：取消当前操作，即不进行删除数据库的操作。

(2) 在命令窗口通过命令删除数据库。

删除数据库的命令的格式是：**DELETE DATABASE <数据库名> [DELETETABLES][RECYCLE]**。

删除指定的数据库，可以通过后面的短语确定删除方式。其中，若选用 DELETETABLES，则将数据库和其中所有的表同时删除，否则，只删除数据库。若选用 RECYCLE，则表明把删除的数据库和表放入回收站，否则直接彻底删除。



图 7-4 删除数据库对话框

7.2 建立数据库表

在 Visual FoxPro 中，表是存储数据的文件，一般一个表对应于一个扩展名为“.dbf”文件。表可以分为数据库表和自由表两种。本节将主要介绍数据库表的创建，自由表的创建将在 7.6 节中介绍。

7.2.1 数据库表的创建

在数据库中创建表首先要建立表的结构，然后再输入数据。

在 Visual FoxPro 中，创建数据库表有以下 3 种方法：

- (1) 通过项目管理器创建数据库表。
- (2) 在数据库设计器中创建数据库表。
- (3) 在命令窗口中通过命令创建数据库表。

1. 通过项目管理器创建数据库表

通过项目管理器创建数据库表的步骤如下：

(1) 在“项目管理器”对话框中，选择“数据”选项卡，从中选择“数据库”选项，选中需要创建数据库表的数据库文件，选中“表”，如图 7-5 所示。单击“新建”按钮，弹出“新建表”对话框，如图 7-6 所示。



图 7-5 在项目管理器中选择“表”



图 7-6 “新建表”对话框

(2) 单击“新建表”按钮，弹出“创建”对话框，通过“保存在”下拉列表框可以设置文件保存的路径，在“输入表名”文本框中输入表名，如图 7-7 所示。单击“保存”按钮，打开“表设计器”对话框，如图 7-8 所示。

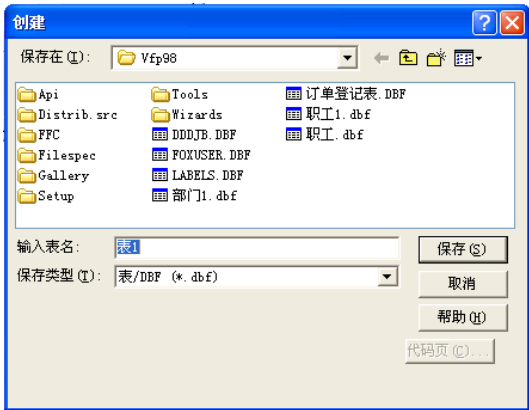


图 7-7 “创建”对话框



图 7-8 “表设计器”对话框

(3) 在“表设计器”对话框中，输入“字段名”、“类型”、“宽度”等内容，单击“确定”按钮，弹出消息提示框，如图 7-9 所示。选择“是”，则立刻可以输入记录，选择“否”，则可以暂时不输入记录。

在“表设计器”对话框中，一些基本内容和概念如下。

● “字段名”。字段名就是关系的属性名，可以通过字段名直接引用表中的数据。

字段名的命名规则如下：

➢ 字段名可以是字母、汉字、数字和下画线。

- 字段名必须以字母或汉字开头。
- 数据库表字段名最长为 128 个字符。
- 自由表字段名最长为 10 个字符。
- 字段名中不能包含空格。

● “类型”。字段的数据类型决定了存储在字段中的值的数据类型，同样的数据类型通过宽度限制可以决定存储的数据的数量和精度。可以选择的数据类型如表 7-1 所示。



图 7-9 消息提示框

表 7-1 字段类型

字段类型	宽度	说明
字符型	用户自定义	可以是汉字、字母、数字等各种字符型文本
数值型	用户自定义	整数或小数
货币型	8	货币单位
浮点型	用户自定义	类似于“数值型”
日期型	8	由年、月、日构成的数据类型
日期时间型	8	由年、月、日、时、分、秒构成的数据类型
双精度型	8	双精度数值类型，一般用于要求精度较高的数据
整型	4	不带小数点的数值类型
逻辑型	1	值为“真”或“假”
备注型	4	不定长的字母数字文本
通用型	4	OLE 对象链接与嵌入
字符型（二进制）	用户自定义	同“字符型”，但是当代码页更改时字符值不变
备注型（二进制）	4	同“备注型”，但是当代码页更改时备注不变

- “空值”。空值即 NULL，它表示是否允许该字段为空值。空值是没有的值或是没有确定的值。一个字段是否允许为空值与实际的应用有关。
- “字段有效性”。在“字段有效性”中可以定义字段的有效性规则、违反规则时的提示信息和字段的默认值。
- “显示”选项组。在“显示”选项组中可以定义字段显示的格式、输入的掩码和字段的标题。
 - “格式”：用于输入格式表达式。格式实质上就是一个输出掩码，可以确定字段在表单、浏览窗口或报表等界面中的显示样式和风格。
 - “输入掩码”：用于输入输入掩码，可以指定字段的输入格式，限制输入数据的范围，控制用户输入的正确性。
 - “标题”：用于设置字段显示时的标题，使用户更容易理解。如果不指定则显示字段名。
- “字段注释”。Visual FoxPro 中，允许为每个字段添加注释，便于日后维护数据库。

2. 在数据库设计器中创建数据库表

在数据库设计器中创建数据表的方法如下：

在“数据库设计器”对话框中，在主菜单“数据库”菜单中选择“新建表”命令，或者在“数据库设计器”对话框中，单击鼠标右键，弹出快捷菜单，选择“新建表”命令，打开“表设计器”对话框，如图 7-10 和图 7-11 所示。



图 7-10 选择“数据库”菜单中“新建表”命令

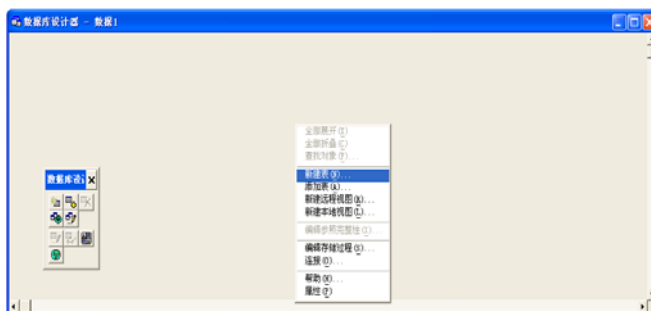


图 7-11 选择快捷菜单中“新建表”命令

3. 在命令窗口通过命令创建数据库表

用命令方式创建数据库表，数据库必须处于打开状态，否则创建的表为自由表。

命令的格式是：CREATE <数据表名>。

7.2.2 修改表结构



图 7-12 在“数据库设计器”中修改表结构

在“表设计器”中，可以增加或删除字段，修改字段名、字段长度和字段数据类型等，也可以增加、删除或者修改索引标志，还可以建立、修改、删除有效性规则等。

修改表的结构首先要打开“表设计器”。打开“表设计器”通常有以下两种方法：

(1) 在“数据库设计器”对话框中，选择要修改的表后，单击鼠标右键，弹出快捷菜单，从中选择“修改”命令，如图 7-12 所示。

(2) 如果当前不在“数据库设计器”中，则首先使用 USE 命令打开需要修改的表文件，然后使用命令 MODIFY STRUCTURE 打开表设计器。

在“表设计器”对话框中，可以进行如下修改操作：

- 修改已有的字段：用户可以直接修改字段的名称、类型和宽度。
- 增加新字段：如果想要在原有的字段后增加新的字段，首先将鼠标指针移动到最后，然后输入新的字段名、类型和宽度即可；如果想要在某个原有字段前插入新的字段，首先将鼠标指针移动到该字段上，单

击“插入”按钮，一个新字段就会插入到当前字段前，然后输入新的字段名、类型和宽度。

- 删除不用的字段：如果想要删除某个字段，首先将鼠标指针定位在需要删除的字段上，然后单击“删除”按钮。

7.3 表的基本操作

对表的基本操作包括在表中添加、删除、修改和查看记录等。

7.3.1 打开表

在 Visual FoxPro 中，打开表有以下 3 种方法。

- 通过项目管理器打开表。
- 通过“打开”对话框打开表。
- 通过命令方式打开表。

打开表的命令格式是：USE <表名>。

7.3.2 浏览表

浏览器采用二维表形式显示表的记录，而且可以使用 Browse 浏览器对表中的数据进行交互式的操作，包括追加、定位、修改和删除等操作。在 Visual FoxPro 中，使用浏览器对表进行浏览有以下 3 种方法：

- (1) 通过项目管理器打开浏览器。
- (2) 在数据库设计器中打开浏览器。
- (3) 使用命令的方式打开浏览器。

1. 通过项目管理器打开浏览器

在“项目管理器”对话框中，将“数据”选项卡中“数据库”选项展开至“表”，并且选择需要操作的表，然后单击“浏览”按钮，如图 7-13 所示。

产品id	产品名称	供应商id	类别id	单位数量	单价	库存量	订购量	再订购量
1.00000	苹果汁	1.00000	1.00000	每箱24瓶	18.00000	39.00000	0.00000	10.00000
2.00000	牛奶	1.00000	1.00000	每箱24瓶	19.00000	17.00000	40.00000	25.00000
3.00000	番茄酱	1.00000	2.00000	每箱12瓶	10.00000	13.00000	70.00000	25.00000
4.00000	盐	2.00000	2.00000	每箱12瓶	22.00000	53.00000	0.00000	0.00000
5.00000	麻油	2.00000	2.00000	每箱12瓶	21.35000	0.00000	0.00000	0.00000
6.00000	酱油	3.00000	2.00000	每箱12瓶	25.00000	120.00000	0.00000	25.00000
7.00000	海鲜粉	3.00000	7.00000	每箱30盒	30.00000	15.00000	0.00000	10.00000
8.00000	胡椒粉	3.00000	2.00000	每箱30盒	40.00000	6.00000	0.00000	0.00000
9.00000	鸡	4.00000	6.00000	每袋500克	97.00000	29.00000	0.00000	0.00000
10.00000	蟹	4.00000	8.00000	每袋500克	31.00000	31.00000	0.00000	0.00000
11.00000	民众奶酪	5.00000	4.00000	每袋8包	21.00000	22.00000	30.00000	30.00000
12.00000	德国奶酪	5.00000	4.00000	每箱12瓶	38.00000	66.00000	0.00000	0.00000
13.00000	龙虾	6.00000	8.00000	每袋500克	6.00000	24.00000	0.00000	5.00000
14.00000	炒茶	6.00000	7.00000	每箱12瓶	23.25000	35.00000	0.00000	0.00000
15.00000	味精	6.00000	2.00000	每箱30盒	15.50000	39.00000	0.00000	5.00000
16.00000	饼干	7.00000	3.00000	每箱30盒	17.45000	29.00000	0.00000	10.00000
17.00000	猪肉	7.00000	6.00000	每袋500克	39.00000	0.00000	0.00000	0.00000
18.00000	墨鱼	9.00000	8.00000	每袋500克	62.50000	42.00000	0.00000	0.00000
19.00000	糖果	8.00000	3.00000	每箱30盒	9.20000	25.00000	0.00000	5.00000
20.00000	桂花糕	8.00000	3.00000	每箱30盒	81.00000	40.00000	0.00000	0.00000
21.00000	花生	8.00000	3.00000	每箱30包	10.00000	3.00000	40.00000	5.00000
22.00000	糯米	9.00000	5.00000	每袋3公斤	21.00000	104.00000	0.00000	25.00000
23.00000	燕麦	9.00000	5.00000	每袋3公斤	9.00000	61.00000	0.00000	25.00000
24.00000	汽水	10.00000	1.00000	每箱12瓶	4.50000	20.00000	0.00000	0.00000

图 7-13 表的浏览窗口

2. 在数据库设计器中打开浏览器

在“数据库设计器”中，选中需要操作的表，单击鼠标右键，在弹出的快捷菜单中选择“浏览”命令，如图 7-14 所示。

3. 使用命令的方式打开浏览器

在命令方式下，首先用 USE 命令打开要操作的表，然后输入 BROWSE 命令。

命令格式为：

```
BROWSE [<范围>][FIELDS <字段名表>][FOR <条件表达式>][NOMODI][NOAPPEND][NODELETE]
```

打开浏览窗口，并按照指定的范围和条件显示记录的指定字段。

- NOMODI：不允许进行修改。
- NOAPPEND：不允许追加记录。
- NODELETE：不允许删除记录。

4. 通过浏览器操作表

打开浏览器窗口以后，可以利用浏览器对表进行的常规操作有：

- (1) 浏览操作。可以利用鼠标对表中记录进行上、下、左、右滚动翻页和定位，除此之外，还可以进行如下操作：
- 上一条记录：【↑】键。
 - 下一条记录：【↓】键。
 - 上一页：【PageUp】键。
 - 下一页：【PageDown】键。
 - 上一字段：【Shift+Tab】组合键。
 - 下一字段：【Tab】键。
- (2) 添加记录。使用【Ctrl+Y】组合键或在主菜单“表”菜单中选择“追加新记录”命令，则系统会在浏览器尾部增加一条新的空白记录。
- (3) 修改记录。将鼠标指针定位到需要修改的记录上，然后就可以直接对该记录进行修改了。
- (4) 删除记录。在浏览器状态下，选用需要删除的记录，在主菜单“表”菜单中选择“删除记录”命令，则对该记录进行逻辑删除，即打上删除标记。如果想要物理删除该记录，就需要在主菜单“表”菜单中选择“彻底删除”命令。
- (5) 退出浏览器。使用【Ctrl+W】组合键或【Esc】键可以退出 Browse 浏览器。



图 7-14 在“数据库设计器”中打开浏览器

7.3.3 增加记录命令

向已打开的表中增加记录的命令有两种：APPEND 命令和 INSERT 命令。

1. APPEND 命令



图 7-15 添加记录

该命令的作用是在表的尾部增加一个或多个新的记录，它有 3 种格式：

- APPEND：一次可以连续输入多条新的记录，如图 7-15 所示。
- APPEND BLANK：在表的尾部增加一条空白记录，然后利用 EDIT、CHANGE 或 BROWSE 命令交互式输入记录值，或用 REPLACE 命令替换该空白记录值。
- APPEND FROM <文件名> [FOR <条件>][FIELDS <字段名表>]：从指定文件中将满足条件的记录追加到当前表的尾部。如果指定了 FIELDS <字段名表>，则只追加指定字段的字段值，否则，将追加全部字段的字段值。

2. INSERT 命令

在表的任意位置插入新的记录，命令格式如下：INSERT [BEFORE][BLANK]。
在表中当前记录的前面或后面插入一条新的记录。

如果不指定 BEFORE，则在当前记录之后插入一条新记录，否则在当前记录之前插入一条新记录。如果不指定 BLANK，则以交互式命令输入记录的值，否则在当前记录之后（或之前）插入一条空白记录。然后利用 EDIT、CHANGE 或 BROWSE 命令交互式输入记录值，或用 REPLACE 命令替换该空白记录值。

7.3.4 删除记录命令

Visual FoxPro 中的记录删除分为逻辑删除和物理删除。逻辑删除是在记录旁打上删除标记，此时表中被打上删除标记的记录并没有真正从表中删除，需要时还可以将删除标记去掉，恢复该记录；物理删除则是将记录真正的从表中删除。

删除表中的记录一般分为两步：

- (1) 对要删除的记录添加删除标志（逻辑删除）。
- (2) 将已打上删除标记的记录从表中彻底删除（物理删除）。

1. 逻辑删除命令

实现逻辑删除或置删除标记，命令是：DELETE [FOR <条件表达式>]。
使用 FOR 短语，则逻辑删除使“逻辑表达式”为真的所有记录；不使用 FOR 短语，则只逻辑删除当前记录。

2. 恢复记录命令

恢复被逻辑删除的记录，命令是：RECALL [FOR <条件表达式>]。
使用 FOR 短语，则恢复使“逻辑表达式”为真的所有记录；不使用 FOR 短语，则只恢复当前记录。

3. 物理删除所有有删除标记的记录

物理删除所有逻辑删除（有删除标记）的记录，不可再被恢复，命令是：PACK。

4. 物理删除表中的所有记录

物理删除表中全部记录，并没有删除表，表结构依然存在，命令是：ZAP。

7.3.5 修改记录命令

在 Visual FoxPro 中，可以交互式的修改记录，也可以用指定值来直接替换原有记录的值。

1. 使用 EDIT 或 CHANGE 命令交互式修改记录值

在 Visual FoxPro 中，使用 EDIT 或 CHANGE 命令可以对当前记录进行交互式的修改和编辑，如图 7-16 所示。

命令的格式为：EDIT [FIELDS <字段名表>][FOR <条件>]。

2. 使用 REPLACE 命令直接用指定表达式或值替换记录值

可以使用 REPLACE 命令直接用指定表达式或值替换当前的记录值，命令格式是：REPLACE <字段名 1> WITH 表达式 1[, <字段名 2> WITH 表达式 2]...[FOR <条件表达式>]。

该命令是用“表达式”的值替换“字段名”的值。如果使用 FOR 短语，则替换使“逻辑表达式”为真的所有记录；不使用 FOR 短语，则只替换当前记录。



图 7-16 修改记录

7.3.6 显示记录命令

向表文件增加记录之后，可以随时查看表中的记录数据。

显示记录时可以使用 LIST 或 DISPLAY 命令，二者区别仅在于不使用条件时，LIST 默认显示全部记录，而 DISPLAY 则默认显示当前记录，如图 7-17 所示。

图 7-17 显示记录

命令格式是：LIST/DISPLAY [[FIELDS] <字段名表>][FOR <条件表达式>][OFF][TO PRINTER|TO FILE <文件名>]。

其中：

- FIELDS: 用逗号分隔的字段名列列表，默认显示全部字段。
- FOR <条件表达式>: 显示满足条件的记录。
- TO PRINTER: 将结果输出到打印机。
- TO FILE: 将结果输出到文件。

7.3.7 定位记录命令

在数据库应用中，有时需要将记录定位到某条想要操作的记录上。记录定位通常使用以下 3 种方法：GOTO/GO 命令、SKIP 命令和 LOCATE 命令。

1. 利用 GOTO/GO 命令直接定位

命令格式是：GOTO/GO 记录号[TOP|BOTTOM]。

其中：

- 记录号：通过记录号可以直接定位。
- TOP：定位到表头，即表中第一条记录。
- BOTTOM：定位到表尾，即表中最后一条记录。

2. 利用 SKIP 命令从当前位置向前或向后移动若干条记录位置

命令格式是：SKIP n。

n 可以是正数或负数，默认值是 1。如果是正数，则向后移动；如果是负数，则向前移动。

3. 使用 LOCATE 命令按条件定位记录

- 命令格式是：LOCATE FOR <条件表达式>。
- <条件表达式>是查询定位的表达式，该命令执行后将记录指针定位在满足条件的第一条记录上，如果没有满足条件的记录，则指向文件结束位置。
- 为了判别 LOCATE 命令是否找到了满足条件的记录，可以使用函数 FOUND()，如果有满足条件的记录，该函数返回逻辑真 (.T.)，否则返回逻辑假 (.F.)。
- 如果想要指针定位到下一个满足条件的记录，可以使用 CONTINUE 命令。如果没有下一个满足条件的记录，则指向文件结束位置。

7.4 索引

数据库的主要功能是对数据的查询，而索引就是在不改变表中记录物理顺序的前提下，按照索引关键字的顺序（升序或降序）排列记录，从而方便快速查找。

7.4.1 基本概念

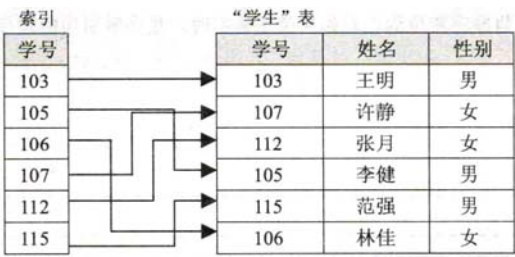


图 7-18 索引

所谓索引，是根据指定字段或字段表达式（即索引关键字）建立的一个索引关键字与记录号之间的一个对照表（即索引文件），如图 7-18 所示。

在 Visual FoxPro 中，根据索引功能的不同，可以将索引分成不同的类别。

1. 按扩展名分类

Visual FoxPro 中支持复合索引和单一索引两类索引文件。前者扩展名为“.cdx”，后者扩展名为“.idx”。

1) 复合索引文件

复合索引文件允许包含多个索引，每个索引都有一个索引标识，代表一种记录逻辑顺序。这种索引以压缩的形式存储，以便减少存储空间。

复合索引文件又分为结构化复合索引文件和非结构化复合索引文件。

- 结构化复合索引文件：索引文件名和表文件名相同，它与表文件同步打开与关闭。
 - 非结构化复合索引文件：索引文件名和表文件名不相同，它不与表文件同步，必须使用专门的命令将其打开与关闭。
- 2) 单一索引文件
- 单一索引文件中仅有一个索引，这种类型是为了与 FoxBASE 早期版本兼容而保留的。

2. 按功能分类

索引除了具有建立记录逻辑顺序、方便查找的作用外，还能控制是否允许相同的索引关键字的值在不同记录中重复出现，或允许在永久关系中建立参照完整性。

1) 主索引

所谓主索引，是指在指定字段或表达式中不允许出现重复值的索引。主索引可以确保字段中输入值的唯一性并决定处理记录的顺序。

建立主索引的字段可以看做是主关键字，一个表只能有一个主关键字，所以也只能创建一个主索引。

2) 候选索引

候选索引与主索引相同，也是一个在指定字段或表达式中不允许出现重复值的索引。候选索引与主索引最大的区别在于，一个数据库表只能建立一个主索引，但是可以建立多个候选索引。

之所以称为候选索引，主要是因为这种索引是作为一个表中主索引的候选者出现的。

3) 唯一索引

唯一索引是为了保持同早期版本兼容而保留的，对于关键字相同的记录，索引中只列出其中第一个记录，即输出时无重复值。在一个表中可以建立多个唯一索引。

4) 普通索引

普通索引也可以决定记录的处理顺序，它不仅允许字段中出现重复值，并且索引项中也允许出现重复值。在一个表中可以建立多个普通索引。

通常，主索引用于主关键字字段，候选索引用于那些不作为主关键字但字段值又必须唯一的字段；普通索引用于提高查询速度；唯一索引用于一些特殊的程序设计。

以上 4 种索引类型的区别如表 7-2 所示。

表 7-2 索引类型

索引类型	关键字重复值	说明	创建、修改命令	索引个数
主索引	不允许	仅适用于数据库表，可用于在永久关系中建立参照完整性	CREATE TABLE ALTER TABLE	仅一个
候选索引	不允许	可用做主关键字，可用于在永久关系中建立参照完整性	INDEX CREATE TABLE ALTER TABLE	允许多个
唯一索引	允许，但输出无重复值	为与早期版本兼容而设置	INDEX	允许多个
普通索引	允许	可以作为一对多关系的“多方”	INDEX	允许多个

7.4.2 建立索引

在 Visual FoxPro 中，建立和修改索引有以下两种方法：

- (1) 使用表设计器建立和修改索引。
- (2) 使用命令建立索引。

1. 使用表设计器建立和修改索引

使用表设计器建立和修改索引，具体步骤如下：

- (1) 打开“表设计器”对话框，在“字段”选项卡中选择要建立或修改索引的字段，然后在“索引”下拉列表框，确定升序或降序，如图 7-19 所示。
- (2) 打开“索引”选项卡，确定索引类型。在“类型”下拉列表中，有 4 种索引类型：主索引、候选索引、唯一索

引和普通索引，如图 7-20 所示。



图 7-19 建立索引

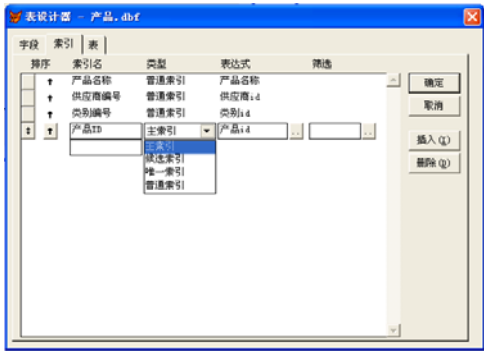


图 7-20 确定索引类型



图 7-21 “表达式生成器”对话框

2. 使用命令建立索引

建立索引的命令是 INDEX 命令，命令的格式是：

```
INDEX ON <索引表达式> TO <索引文件名>|TAG <索引标识名> [OF <索引文件名>]
[FOR <条件表达式>]
[ASCENDING|DESCENDING]
[UNIQUE|CANDIDATE]
[ADDITIVE]
```

其中：

- <索引表达式>：可以是字段名，或包含字段名的表达式。
- TO <索引文件名>：用来建立单一索引文件。
- TAG <索引标识名>：用来建立结构化的复合索引文件。
- OF <索引文件名>：用来建立非结构化的复合索引文件。
- FOR <条件表达式>：给出索引过滤条件，只对满足条件的记录进行索引。
- ASCENDING|DESCENDING：升序或降序，默认是升序的。
- UNIQUE|CANDIDATE：唯一索引或候选索引。
- ADDITIVE：在建立索引时是否关闭当前的索引，默认关闭当前正在使用的索引，使新建索引成为当前索引。

7.4.3 使用索引

建立索引之后就要使用索引文件，使用索引文件包括打开索引、设置主控索引、使用索引快速定位、删除索引等操作。

1. 打开索引

索引文件同表文件一样，必须打开后才能使用。结构化复合索引文件在打开和关闭表文件时会随着表文件自动打开和关闭。但是对于非结构化复合索引文件（.cdx）和单一索引文件（.idx），则必须在使用之前打开索引文件。

打开索引文件的命令格式为：SET INDEX TO <索引文件名>。

2. 设置主控索引

对于一个表，允许打开多个索引文件。但是在某一时刻，在使用某个特定索引项进行查询或需要记录按某个特定索引项的顺序显示时，则只能有一个索引条目为主控索引，它确定了表记录的显示和访问顺序。

主控索引可以使用命令方式进行设置，命令格式是：

```
SET ORDER TO [索引序号| [TAG] <索引标识名>] [ASCENDING|DESCENDING]
```

其中，可以按索引序号或索引标识名指定索引项；不管索引是按升序还是降序建立的，在使用时都可以用 ASCENDING 或 DESCENDING 重新指定升序或降序，无此选项则默认按建立索引时规定的顺序显示；默认索引序号或索引标识名时，表示无主控索引，系统按物理顺序显示记录。

3. 使用索引快速定位

可以使用 SEEK 命令进行使用索引快速定位，命令格式如下：

```
SEEK <表达式> [ORDER 索引序号| [TAG] <索引标识名>] [ASCENDING|DESCENDING]
```

其中，可以按照索引序号或索引标识名指定对哪个索引进行快速定位；表达的值是索引项或索引关键字的值；还可以使用 ASCENDING 或 DESCENDING 说明按升序或降序定位。

4. 删除索引

如果某个索引不再使用了则可以删除它。

删除结构化索引的命令格式如下：DELETE TAG <索引标识名>。

删除全部索引的命令格式如下：DELETE TAG ALL。

7.5 排序

所谓排序就是就将表中的记录按照某一字段值的大小重新排列，然后复制产生一个新的表文件。

排序和索引的区别是：

- 索引是通过索引关键字实现按照用户的需要以某种顺序浏览或查找表中的记录，这个顺序是逻辑顺序。
- 排序则是将表中的记录按照新的物理顺序进行排列。

使用 SORT 命令可以实现物理排序，命令格式是：

```
SORT TO <新表名> ON <字段名 1> [/A|/D][/C] [, <字段名 2> [/A|/D][/C]...
    [ASCENDING| DESCENDING]
    [FOR <条件表达式>]
    [FIELDS <字段名列表>]
```

其中：

- <新表名>：为排序后的表名，SORT 命令对当前表进行排序，排序结果存放到一个新表中。
- <字段名 1>、<字段名 2>：为排序的字段，可以在多个字段上进行排序。
- [/A|/D][/C]：其中/A 说明按升序排序，默认是升序；/D 说明按降序排序；/C 说明排序时不区分大小写字母，默认是区分大小写。
- [ASCENDING|DESCENDING]：指出了除用/A 或/D 指明排序方式的字段外，所有其他排序字段按升序或降序排列。
- FOR <条件表达式>：给出参加排序字段需要满足的条件，只有使<条件表达式>为真的记录参加排序。
- FIELDS <字段名列表>：给出排序以后的表所包含的字段列表，默认是原表的所有字段。

7.6 数据完整性

数据的完整性就是指保证数据库中数据正确的特性。完整性控制的主要目的在于防止不正确的数据进入数据库。在 Visual FoxPro 中，对数据库的操作一般通过 3 类完整性规则来保证数据库中数据的正确性：实体完整性、域完整性和参照完整性。

7.6.1 实体完整性与主关键字

实体完整性指的是在一个表中不允许有重复的记录，是保证表中记录唯一的特性。在 Visual FoxPro 中利用主关键字或候选关键字来保证实体完整性。

如果一个字段的值或几个字段的值能够唯一标识表中的一条记录，则这样的字段称为候选关键字。在一个表上可能会有几个具有这种特征的字段或字段的组合，这时从中选择一个作为主关键字。

在 Visual FoxPro 中，将主关键字称为主索引，将候选关键字称为候选索引。

7.6.2 域完整性与约束规则



图 7-22 设置字段有效性规则

域完整性是指限制表中字段取值的有效性规则。实现域完整性的方法主要有数据类型定义和域约束规则。

域约束规则也称为字段有效性规则，可以在“表设计器”中建立字段有效性规则，Visual FoxPro 中提供的字段有效性规则有 3 类：规则、信息和默认值。

- 规则：字段有效性规则。
- 信息：违背字段有效性规则时的提示信息。
- 默认值：字段的默认值。

具体的操作步骤是：

- (1) 在“表设计器”中，选中需要定义字段有效性规则的字段。
- (2) 在“字段有效性”文本框中输入编辑规则、信息及默认值等内容。设置字段有效性规则的结果如图 7-22 所示。

7.6.3 参照完整性与表之间的关联

参照完整性是指当插入、删除或修改一个表中的数据时，需要参照引用相互关联的另一个表中的数据，借以检查对表的数据操作是否正确。

参照完整性属于表间操作，并且与表之间的联系有关，因此在 Visual FoxPro 中为了建立参照完整性，必须首先建立表之间的联系，然后才能设置参照完整性规则。

1. 建立表之间的联系

表之间的联系可以利用“数据库设计器”对话框来建立，具体步骤如下：

- (1) 在“数据库设计器”对话框中，在建立表之间的联系时，首先要在父表中建立主索引，在子表中建立普通索引，然后通过父表的主索引和子表的普通索引建立两个表之间的联系。
- (2) 用鼠标选中父表中的主索引字段，按住鼠标左键不放，拖至子表中的普通索引字段，然后释放鼠标，即可建立两表之间的联系，如图 7-23 所示。

从图 7-23 可以看到，主索引字段一边是 1 条线，而另外普通索引一边是 3 条线，说明它们是一对多的关系。

如果需要删除表之间的联系，可以用鼠标指针对准连线单击鼠标右键，弹出快捷菜单，选择“删除关系”命令即可。

如果需要修改表之间的联系，可以用鼠标指针对准连线单击鼠标右键，弹出快捷菜单，选择“编辑关系”命令，打

开“编辑关系”对话框，如图 7-24 所示。

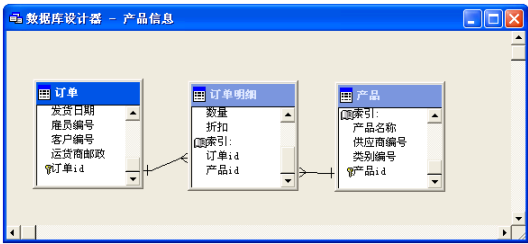


图 7-23 建立表之间的联系



图 7-24 “编辑关系”对话框

2. 设置参照完整性规则

设置参照完整性规则的步骤如下：

(1) 清理数据库。所谓清理数据库，就是物理删除数据库各个表中所有带有删除标记的记录。在“数据库设计器”中，在主菜单“数据库”菜单中选择“清理数据库”命令，即可实现清理数据库。

(2) 设置参照完整性规则。清理完数据库后，在主菜单“数据库”菜单中选择“编辑参照完整性”命令，或者在“编辑关系”对话框中，单击“参照完整性”按钮，打开“参照完整性生成器”对话框，如图 7-25 所示。

在 Visual FoxPro 中的参照完整性约束有 3 种：更新规则、删除规则和插入规则。

- 更新规则：规定了当更新父表中的连接字段（主关键字）值时，如何处理相关子表中的记录。
- 删除规则：规定了当删除父表中的记录时，如何处理子表中相关的记录。
- 插入规则：规定当插入子表中的记录时，是否进行参照完整性检查。

各种规则的各个选项的具体含义如表 7-3 所示。

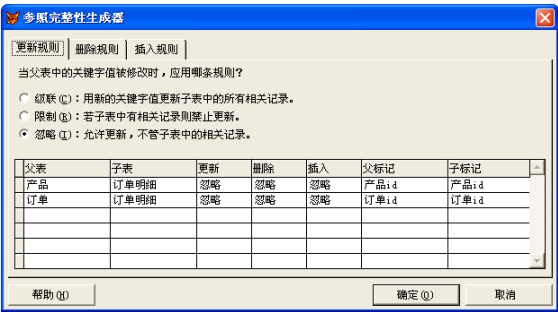


图 7-25 “参照完整性生成器”对话框

表 7-3

参照完整性约束	级联	限制	忽略
更新规则	当更改父表关键字段值时，被关联子表相关记录对应的字段值自动修改	当子表中有相关记录时，则禁止修改父表中主关键字段的值	不做参照完整性检查，可以随意更新父表中主关键字段的值
删除规则	当删除父表记录时，被关联子表相关记录自动删除	当子表中有相关的记录时，则禁止删除父表中的记录	不做参照完整性检查，可以随意删除父表中的记录
插入规则	无	当父表中没有相关记录时，则禁止在子表中插入相关记录	不做参照完整性检查，可以随意在子表中插入记录

7.7 自由表

在 Visual FoxPro 中，表分为数据库表和自由表。自由表不属于任何一个数据库。本节将重点介绍自由表的使用。

7.7.1 数据库表和自由表

所谓数据库表，就是隶属于某一个数据库的数据表；所谓自由表，就是不隶属于任何数据库的数据表。

数据库表和自由表相比，具有以下一些优点：

- (1) 可以使用长表名和长字段名。
- (2) 可以为字段指定标题和添加注释。
- (3) 可以为字段指定默认值和输入掩码。
- (4) 可以规定字段级规则和记录级规则。
- (5) 支持主关键字、参照完整性和表之间的联系。
- (6) 支持 SQL 的 INSERT、UPDATE、DELETE 事件的触发器定义。
- (7) 字段有默认的控制类。

在 Visual FoxPro 中，如果当前没有打开数据库，则创建的表就是自由表。可以将自由表添加到数据库中，使之成为数据库表；也可以将数据库表从数据库中移出，使之成为自由表。

7.7.2 自由表的创建

自由表的创建过程和数据库表的创建过程基本相同，主要有以下 3 种方法。

1. 在项目管理器中创建自由表

在“项目管理器”对话框中，在“数据”选项卡中选择“自由表”选项，单击“新建”按钮，弹出“新建表”对话框。单击“新建表”按钮，则打开“表设计器”对话框，创建一个新的自由表，如图 7-26、图 7-27 和图 7-28 所示。



图 7-26 选择“自由表”选项

图 7-27 “新建表”对话框

图 7-28 “表设计器”对话框

2. 通过菜单方式创建自由表

在主菜单“新建”菜单中，选择“新建”命令，打开“新建”对话框，如图 7-29 所示。选择文件类型为“表”，单击“新建文件”按钮，打开“新建表”对话框，单击“新建表”按钮，打开“表设计器”对话框。

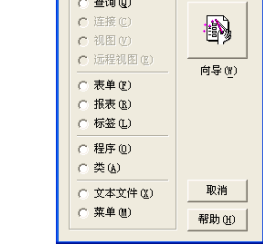


图 7-29 “新建”对话框

3. 使用命令方式创建自由表

使用命令方式创建自由表，命令的格式是：CREATE <数据表名>。

7.7.3 将自由表添加到数据库中

可以把自由表添加到数据库中，使之成为数据库表。一个表只能属于一个数据库，当一个自由表添加到某个数据库

中,就不再是自由表了,因此,不能添加已经属于某个数据库的数据库表到当前数据库中。

1. 使用项目管理器向数据库中添加表

在“项目管理器”对话框中,选择“数据”选项卡,将要添加自由表的数据库展开至“表”选项,单击“添加”按钮,如图 7-30 所示。

弹出“打开”对话框,选择要添加到当前数据库的自由表(即.dbf 文件),即可将该自由表添加到当前数据库中,如图 7-31 所示。



图 7-30 使用项目管理器向数据库中添加表

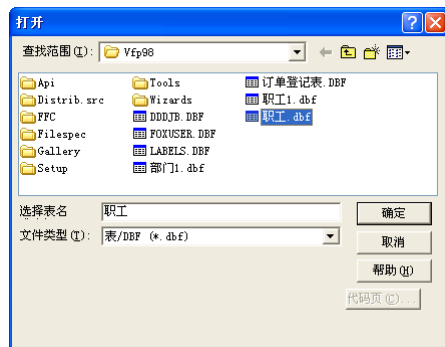


图 7-31 “打开”对话框

2. 使用数据库设计器向数据库中添加表

在“数据库设计器”对话框中,在主菜单“数据库”菜单中选择“添加表”命令,或者单击鼠标右键,在快捷菜单中选择“添加表”命令,弹出“打开”对话框,选择要添加到当前数据库的自由表(即.dbf 文件),即可将该自由表添加到当前数据库中,如图 7-32 和图 7-33 所示。

3. 使用命令向数据库中添加表

可以使用 ADD TABLE 命令添加一个自由表到当前数据库中,命令格式是:

ADD TABLE [<表名>|?][NAME <长表名>]

其中:

- <表名>: 要添加到当前数据库的自由表的表名。
- 问号“?”: 将弹出“打开”对话框,从中选择要添加到当前数据库中的自由表文件。
- NAME <长表名>: 为表指定一个长表名,最多可以有 128 个字符。



图 7-32 在“数据库”菜单中选择“添加表”命令



图 7-33 在快捷菜单中选择“添加表”命令

7.7.4 从数据库中移去表

当数据库不再使用某个表，可以将该表从当前数据库中移去，使之成为自由表。

1. 使用项目管理器从数据库中移去表

在“项目管理器”对话框中，选择“数据”选项卡，将所要移去表的数据库展开至“表”选项，选择所要移去的表，单击“移去”按钮，如图 7-34 所示。

弹出提示信息框，单击“移去”按钮，即可将该数据库表从当前数据库中移去，成为自由表，如图 7-35 所示。

2. 使用数据库设计器从数据库中移去表

在“数据库设计器”对话框中，选中所要移去的表，然后在主菜单“数据库”菜单中选择“移去”命令，弹出提示信息框，单击“移去”按钮，即可将该数据库表从当前数据库中移去，成为自由表，如图 7-36 所示。



图 7-34 使用项目管理器从数据库中移去表



图 7-35 提示信息框

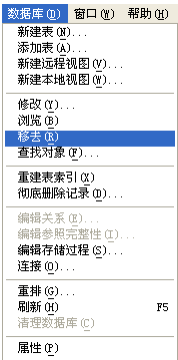


图 7-36 在“数据库”菜单中选择“移去”命令

3. 使用命令从数据库中移去表

可以使用 REMOVE TABLE 命令将一个数据库表从当前数据库中移去，使之成为自由表，命令格式是：

```
REMOVE TABLE [<表名>|?][DELETE][RECYCLE]
```

其中：

- <表名>：要从当前数据库中移去的数据库表的表名。
- 问号“？”：将弹出“移去”对话框，从中选择要从当前数据库中移去的数据库表文件。
- [DELETE]：把所选从数据库中移去的表从磁盘中删除。
- [RECYCLE]：把所选从数据库中移去的表放入回收站中。

7.8 多个表的同時使用

在 Visual FoxPro 中，一次可以打开多个数据库，在每个数据库中打开多个表，另外还可以打开多个自由表。

7.8.1 工作区的概念

所谓工作区，就是在内存中开辟的一块区域，用于存放表文件。如果需要在同一时刻打开多个表，则需要不同的工作区中打开不同的表。

在 Visual FoxPro 中，同一时刻最多允许打开 32767 个工作区，这些工作区分别用数字 1~32767 来标识，称为工作区号。其中，前 10 个工作区还可以使用字母 A~J 来标识，称为工作区的别名。系统默认的工作区为 1 号工作区，用户可以

使用工作区的选择命令选择工作区，当前被选中的工作区称为当前工作区。

选择工作区的命令是：**SELECT <工作区号>|<工作区别名>**。

该命令用来选择指定的工作区为当前工作区，工作区号从 1~32767，如果使用 **SELECT 0** 时，则表示要选择当前未使用的最小工作区号；每一个工作区在某一时刻只能打开一个表文件；可以使用 **SELECT()** 函数来测定当前的工作区号。

7.8.2 使用不同工作区的表

如果要在当前工作区中调用另外一个工作区中的表的记录数据时，可以使用如下调用格式：**<工作区别名>.<字段名>**或者**<工作区别名>-><字段名>**。

另外，还可以通过短语 **IN** 在命令中指定要引用的工作区号、表名或表别名。

格式如下：**IN <工作区号>|<表名>|<表别名>**。

7.8.3 表的关联操作

在使用多个表时，经常需要在移动一个表中记录指针的同时，其他相关表中的记录指针也能够自动调整到相应的位置上。在 **Visual FoxPro** 中，可以通过表的关联来实现这种操作。

所谓关联，就是不同工作区的表记录指针之间建立一种临时的联动关系，使一个表的记录指针移动时，另一个表的指针也随之移动。此时把第一个表称为主动表，第二个表称为被动表。

建立表间临时关联的命令格式如下：

SET RELATION TO <关联表达式> INTO [<别名>|<工作区号>][ADDITIVE]

其中：

- **<关联表达式>**：可以是数值表达式，也可以是索引关键字。如果使用“数值表达式”则主动表中的记录和被动表中的记录通过记录号建立关联；如果使用“索引关键字”则主动表与被动表以公共字段为索引关键字建立关联，此时要求被动表必须以此关键字建立索引，并且将其设定为主控索引。
- **<别名>|<工作区号>**：用来说明临时联系是由当前工作区中的表到其他工作区的哪个表的。
- **[ADDITIVE]**：一个主动表可以同时与多个被动表建立关联，如果指定 **ADDITIVE** 则保留以前建立的关联，否则删除以前建立的关联。当建立多个关联时，主动表的指针移动时，其他所有被动表的记录指针也随之移动。

撤销表间临时关联的命令格式如下：**SET RELATION TO**。

该命令将当前表到所有表的临时联系全都取消。

如果只是取消某个具体的临时联系，命令格式如下：**SET RELATION OFF INTO <别名>|<工作区号>**。

7.9 本章习题

7.9.1 选择题

1. 在 **Visual FoxPro** 中，数据库文件的扩展名为（ ）。
A. .dct B. .dbc C. .dcx D. .idx
2. 打开数据库的命令是（ ）。
A. USE B. USE DATABASE C. OPEN D. OPEN DATABASE
3. **CREATE DATABASE** 命令用来建立（ ）。
A. 数据库 B. 关系 C. 表 D. 数据文件
4. 在 **Visual FoxPro** 中，下面描述正确的是（ ）。
A. 数据库表允许对字段设置默认值 B. 自由表允许对字段设置默认值
C. 自由表或数据库表都允许对字段设置默认值 D. 自由表或数据库表都不允许对字段设置默认值
5. 下列与修改表结构相关的命令是（ ）。

- A. INSERT B. ALTER C. UPDATE D. CREATE
6. 当前打开的图书表中有字符型字段“图书号”，要求将图书号以字母 A 开头的图书记录全部打上删除标记，通常可以使用命令（ ）。
 A. DELETE FOR 图书号="A" B. DELETE WHILE 图书号="A"
 C. DELETE FOR 图书号="A*" D. DELETE FOR 图书号 LIKE "A%"
7. 假设有 student 表，可以正确添加字段“平均分数”的命令是（ ）。
 A. ALTER TABLE student ADD 平均分数 F(6,2) B. ALTER DBF student ADD 平均分数 F6,2
 C. CHANGE TABLE student ADD 平均分数 F(6,2) D. CHANGE TABLE student INSERT 平均分数 6,2
8. 要为当前表所有性别为“女”的职工增加 100 元工资，应使用命令（ ）。
 A. REPLACE ALL 工资 WITH 工资+100 B. REPLACE 工资 WITH 工资+100 FOR 性别="女"
 C. CHANGE ALL 工资 WITH 工资+100 D. CHANGE ALL 工资 WITH 工资+100 FOR 性别="女"
9. 有一学生表文件，且通过表设计器已经为该表建立了若干普通索引。其中一个索引的索引表达式为姓名字段，索引名为 XM。现假设学生表已经打开，且处于当前工作区中，那么可以将上述索引设置为当前索引的命令是（ ）。
 A. SET INDEX TO 姓名 B. SET INDEX TO XM
 C. SET ORDER TO 姓名 D. SET ORDER TO XM
10. 使用 SQL 语句增加字段的有效性规则，是为了能保证数据的（ ）。
 A. 实体完整性 B. 表完整性 C. 参照完整性 D. 域完整性
11. 参照完整性规则的更新规则中“级联”的含义是（ ）。
 A. 在更新父表中连接字段值时，用新的连接字段自动修改子表中的所有相关记录
 B. 若子表中有与父表相关的记录，则禁止修改父表中连接字段值
 C. 父表中的连接字段值可以随意更新，不会影响子表中的记录
 D. 父表中的连接字段值在任何情况下都不允许更新
12. 在 Visual FoxPro 中，有关参照完整性的删除规则正确的描述是（ ）。
 A. 如果删除规则选择的是“限制”，则当用户删除父表中的记录时，系统将自动删除子表中的所有相关记录
 B. 如果删除规则选择的是“级联”，则当用户删除父表中的记录时，系统将禁止删除与子表相关的父表中的记录
 C. 如果删除规则选择的是“忽略”，则当用户删除父表中的记录时，系统将不负责检查子表中是否有相关记录
 D. 上面 3 种说法都不对
13. 在 Visual FoxPro 中，下列关于表的叙述正确的是（ ）。
 A. 在数据库表和自由表中，都能给字段定义有效性规则和默认值
 B. 在自由表中，能给表中的字段定义有效性规则和默认值
 C. 在数据库表中，能给表中的字段定义有效性规则和默认值
 D. 在数据库表和自由表中，都不能给表中的字段定义有效性规则和默认值
14. 对表 SC（学号 C（8），课程号 C（2），成绩 N（3），备注 C（20）），可以插入的记录是（ ）。
 A. (' 20080101' , ' cl' , ' 90' ,NULL) B. (' 20080101' , ' cl' , ' 90' , ' 成绩优秀')
 C. (' 20080101' , ' cl' , ' 90' , ' 成绩优秀') D. (' 20080101' , ' cl' , ' 79' , ' 成绩优秀')
15. 在 Visual FoxPro 中，要想实现物理排序，需要使用下述（ ）命令。
 A. SORT B. REPLACE C. INDEX D. SELECT
16. 在 Visual FoxPro 中，系统默认的工作区编号为（ ）。
 A. 0 B. 1 C. 2 D. 3
17. 在 Visual FoxPro 中，每一个工作区中最多能打开数据库表的数量是（ ）。
 A. 1 个 B. 2 个 C. 任意个，根据内存资源而确定 D. 35535 个
18. 下面有关表间永久联系和关联的描述中，正确的是（ ）。
 A. 永久联系中的父表一定有索引，关联中的父表不需要有索引
 B. 无论是永久联系还是关联，子表一定有索引

- C. 永久联系中子表的记录指针会随父表的记录指针的移动而移动
D. 关联中父表的记录指针会随子表的记录指针的移动而移动
19. 执行 USE sc IN 0 命令的结果是 ()。
A. 选择 0 号工作区打开 sc 表 B. 选择空闲的最小号工作区打开 sc 表
C. 选择第 1 号工作区打开 sc 表 D. 显示出错信息
20. 在表设计器的“字段”选项卡中可以创建的索引是 ()。
A. 唯一索引 B. 候选索引 C. 主索引 D. 普通索引
21. 在创建数据库表结构时, 为该表中一些字段建立普通索引, 其目的是 ()。
A. 改变表中记录的物理顺序 B. 为了对表进行实体完整性约束
C. 加快数据库表的更新速度 D. 加快数据库表的查询速度
22. 设有两个数据表, 父表和子表之间是一对多的联系, 为控制子表和父表的关联, 可以设置“参照完整性规则”, 为此要求这两个表 ()。
A. 在父表连接字段上建立普通索引, 在子表连接字段上建立主索引
B. 在父表连接字段上建立主索引, 在子表连接字段上建立普通索引
C. 在父表连接字段上不需要建立任何索引, 在子表连接字段上建立普通索引
D. 在父表和子表的连接字段上都要建立主索引
23. 在指定字段或表达式中不允许出现重复值的索引是 ()。
A. 唯一索引 B. 唯一索引和候选索引
C. 唯一索引和主索引 D. 主索引和候选索引
24. 打开数据库 abc 的正确命令是 ()。
A. OPEN DATABASE abc B. USE abc C. USE DATABASE abc D. OPEN abc
25. Visual FoxPro 的“参照完整性”中“插入规则”包括的选择是 ()。
A. 级联和忽略 B. 级联和删除 C. 级联和限制 D. 限制和忽略
26. 在 Visual FoxPro 中, 使用 LOCATE FOR <条件表达式>命令按条件查找记录, 当查找到满足条件的第一条记录后, 如果还需要查找下一条满足条件的记录, 应使用 ()。
A. 再次使用 LOCATE FOR <条件表达式>命令 B. SKIP 命令
C. CONTINUE 命令 D. GO 命令
27. 在 Visual FoxPro 中, 如果在表之间的联系中设置了参照完整性规则, 并在删除规则中选择了“限制”, 则当删除父表中的记录时, 系统反应是 ()。
A. 不做参照完整性检查 B. 不准删除父表中的记录
C. 自动删除子表中所有相关的记录 D. 若子表中有相关记录, 则禁止删除父表中记录
28. 允许出现重复字段值的索引是 ()。
A. 候选索引和主索引 B. 普通索引和唯一索引
C. 候选索引和唯一索引 D. 普通索引和候选索引
29. 不允许出现重复字段的索引是 ()。
A. 候选索引和主索引 B. 普通索引和唯一索引
C. 唯一索引和主索引 D. 唯一索引
30. 打开表并设置当前有效索引的正确命令是 ()。
A. order student in 2 index 学号 B. use student in 2 order 学号
C. index 学号 order student D. use student in 2
31. 数据库表的字段有效性规则是一个 ()。
A. 字符表达式 B. 数值表达式 C. 日期表达式 D. 逻辑表达式
32. 以下关于索引的正确叙述是 ()。
A. 使用索引可以提高数据查询速度和数据更新速度

- B. 使用索引可以提高数据查询速度，但会降低数据更新速度
C. 使用索引可以提高数据查询速度，对数据更新速度没有影响
D. 使用索引对数据查询速度和数据更新速度均没有影响
33. 无论索引是否生效，定位到相同记录上的命令是（ ）。
A. GO TOP B. GO BOTTOM C. GO 1 D. SKIP
34. 某校学生的学号由 7 位数字组成，则学号字段的正确输入掩码是（ ）。
A. ##### B. ***** C. 7 D. 9
35. 以下叙述正确的是（ ）。
A. 自由表不能被加入到数据库中
B. 数据库表可以建立字段级规则和约束，而自由表不能
C. 可以在自由表之间建立参照完整性规则，而数据库表不能
D. 可以为自由表字段设置默认值，而数据库表字段不能设置默认值
36. 在数据库中数据完整性是指保证数据正确的特性，数据完整性一般包括（ ）。
A. 实体完整性、域完整性和参照完整性
B. 更新规则、删除规则和插入规则
C. 数据默认值、数据取值范围和数据类型
D. 主索引和候选索引
37. 关系数据库中，在表之间建立永久性联系是通过联结两个表的字段来完成和体现的，这种联结是（ ）。
A. 子表中的主关键字与父表中的外部关键字联结
B. 主表中的主关键字与子表中的外部关键字联结
C. 主表中的普通关键字与子表中的外部关键字联结
D. 主表中的唯一关键字与子表中的普通关键字联结
38. 参照完整性规则包括更新规则、删除规则和插入规则。删除规则中选择“级联”的含义是：当删除父表中的记录时（ ）。
A. 系统自动备份父表中被删除记录到一个新表 B. 若子表中有相关记录，则禁止删除父表中记录
C. 会自动删除子表中所有相关记录 D. 不做参照完整性检查，删除父表记录与子表无关
39. 在 Visual FoxPro 中，假定数据库表 S（学号，姓名，性别，年龄）和 SC（学号，课程号，成绩）之间使用“学号”建立了表之间的永久联系，在参照完整性的更新规则、删除规则和插入规则中选择设置了“限制”，如果表 S 中所有的记录在表 SC 中都有相关联的记录，则（ ）。
A. 允许修改表 S 中的学号字段值 B. 允许删除表 S 中的记录
C. 不允许修改表 S 中的学号字段值 D. 不允许在表 S 中增加新的记录
40. 在 Visual FoxPro 中，对于字段值为空值（NULL）的叙述正确的是（ ）。
A. 空值等同于空字符串 B. 空值表示字段还没有确定值
C. 不支持字段值为空值 D. 空值等同于数值 0
41. 在 Visual FoxPro 中，下面关于索引的正确描述是（ ）。
A. 当数据库表建立索引以后，表中记录的物理顺序将被改变
B. 索引的数据将与表的数据存储在一个物理文件中
C. 建立索引是创建一个索引文件，该文件包含有指向表记录的指针
D. 使用索引可以加快对表的更新操作
42. 在 Visual FoxPro 中，在数据库中创建表的 CREATE TABLE 命令中定义主索引，实现实体完整性规则的短语是（ ）。
A. FOREIGN KEY B. DEFAULT C. PRIMARY KEY D. CHECK
43. 在 Visual FoxPro 中，数据库表的字段或记录的有效性规则的设置可以在（ ）。
A. 项目管理器中进行 B. 数据库设计器中进行

- C. 表设计器中进行 D. 表单设计器中进行
44. 在 Visual FoxPro 的数据库表中只能有一个（ ）。
- A. 候选索引 B. 普通索引 C. 主索引 D. 唯一索引
45. 已知表中有字符型字段职称和性别，要建立一个索引，要求首先按职称排序，职称相同时再按性别排序，正确的命令是（ ）。
- A. INDEX ON 职称+性别 TO ttt B. INDEX ON 性别+职称 TO ttt
C. INDEX ON 职称，性别 TO ttt D. INDEX ON 性别，职称 TO ttt
46. 命令 SELECT 0 的功能是（ ）。
- A. 选择编号最小的未使用工作区 B. 选择 0 号工作区
C. 关闭当前工作区的表 D. 选择当前工作区

7.9.2 填空题

1. 在 Visual FoxPro 中, 主索引可以保证数据的_____完整性。
2. 在定义字段有效性规则时, 在规则框中输入的表达式类型是_____。
3. 使数据库表变为自由表的命令是_____TABLE。
4. 当删除父表中的记录时, 若子表中的所有相关记录也能自动删除, 则相应的参照完整性的删除规则为_____。
5. 在 Visual FoxPro 中, 可以在表设计器中为字段设置默认值的表是_____表。
6. 在 Visual FoxPro 中, 数据库表中不允许有重复记录是通过指定_____来实现的。
7. 在 Visual FoxPro 中, 数据库表 S 中的通用型字段的内容将存储在_____文件中。
8. Visual FoxPro 6.0 中结构复合索引文件的扩展名是_____。
9. 可以为字段建立字段有效性规则的表是_____表。
10. 使用“SET RELATION TO”命令可以取消表之间已经存在的_____联系
11. Visual FoxPro 6.0 数据库表的参照完整性是通过表之间的_____联系建立的。
12. 在 Visual FoxPro 中利用_____或候选关键字来保证表中记录(元组)的唯一, 即实体完整性。
13. Visual FoxPro 中数据库文件的扩展名(后缀)是_____。
14. Visual FoxPro 中所谓自由表就是那些不属于任何_____的表。
15. 一个数据库表只能有一个_____索引。
16. 在数据库中数据完整性是指保证数据正确的特性, 数据完整性一般包括实体完整性、_____和参照完整性。
17. 参照完整性规则包括更新规则、_____和插入规则。
18. 可以伴随着表的打开而自动打开的索引文件是_____文件。
19. 数据库表上字段有效性规则是一个_____表达式。
20. 在 Visual FoxPro 中, 通过建立数据库表的主索引可以实现数据的_____完整性。

第8章

关系数据库标准语言 SQL

SQL 是 Structure Query Language（结构化查询语言）的缩写，它首先由 ANSI（美国国家标准协会）提出，后被 ISO 采纳为国际标准，是当今关系数据库的标准语言，现在所有的关系数据库管理系统都支持 SQL。

学习和掌握 SQL，不仅对学习 Visual FoxPro 有重要作用，而且能为学习其他关系数据库奠定扎实的基础。

8.1 SQL 概述

SQL 语言于 1974 年由 Boyce 和 Chamberlin 提出，1975~1979 年 IBM 公司研制了关系数据库管理系统原型 System R，并实现了这种语言。1986 年 10 月，美国国家标准协会（ANSI）数据库委员会批准 SQL 作为关系数据库语言的标准，同年公布了 SQL 标准文本（简称 SQL-86）。1987 年 6 月国际标准化组织（ISO）也通过了这一标准。此后 ANSI 不断修改和完善 SQL 标准，并于 1989 年 4 月公布了 SQL-89 标准，1992 年 11 月又公布了 SQL-92 标准。目前，SQL 仍在不断地修改、扩充和完善中。

自从 SQL 成为关系数据库的国际标准语言后，数据库厂家纷纷推出各自的 SQL 软件或 SQL 接口软件。这就使大多数数据库管理系统均用 SQL 作为共同的数据存取语言 and 标准接口，使不同数据库系统之间的互操作有了共同的基础。

SQL 语言具有如下主要特点：

- （1）SQL 是一种一体化的语言，它包括了数据定义、数据查询、数据操作和数据控制等功能，可以完成数据库活动中的全部工作，而数据查询是它的核心。
- （2）SQL 语言是一种高度非过程化的语言，在它的命令语句中，只需要指出“做什么”，而不需指明“怎样做”，可以将要求交由系统自动完成全部工作即可。
- （3）SQL 语言非常简洁，但功能强大，而且语法简单，接近自然语言，易于掌握。SQL 的命令动词如表 8-1 所示。
- （4）SQL 语言可以直接以命令方式交互使用，又可以嵌入到程序设计语言中以程序方式使用。

表 8-1 SQL 命令动词

SQL 功能	命令动词
数据定义	CREATE、DROP、ALTER
数据操作	INSERT、UPDATE、DELETE
数据查询	SELECT
数据控制	GRANT、REVOKE

现在大多数的关系数据库管理系统都支持 SQL 语言。Visual FoxPro 支持其数据定义、数据查询和数据操作 3 大功能，但在具体实现方面与标准 SQL 相比尚存在一些差异。

SQL 语言的核心功能是查询，熟练掌握它是学习 SQL 的重点。

8.2 定义功能

标准 SQL 的数据定义功能非常广泛，包括数据库的定义、表的定义、存储过程的定义、视图的定义、规则的定义和索引的定义等。

本节主要介绍 Visual FoxPro 支持的表的定义和视图的定义。

8.2.1 表的定义

所谓表的定义功能，即创建表的结构。利用 SQL 命令可以完成表设计器所能完成的所有功能。

创建表的命令的基本格式如下：

```
CREATE TABLE|DBF <表名> [NAME<长表名>][FREE]
(<字段名 1><类型>[(<字段宽度>[, <小数位数>])]
[NULL|NOTNULL]
[CHECK <逻辑表达式 1>[ERROR <字符型文本信息 1>]]
[DEFAULT<表达式 1>]
[PRIMARY KEY|UNIQUE]
[REFERENCES <表名 2>[TAG<标识名 1>]]
[NOCPTRANS][, <字段名 2>...]
[, PRIMARY KEY <表达式 2> TAG <标识名 2>]
[, UNIQUE <表达式 3> TAG <标识名 3>]
[, FOREIGN KEY <表达式 4> TAG <标识名 4> [NODUP]
[REFERENCES <表名 3>[TAG<标识名 5>]]
[, CHECK <逻辑表达式 2>[ERROR <字符型文本信息 2>]])
|FROM ARRAY <数组名>
```

其中：

- TABLE 和 DBF 选项等价，都是建立表文件。
- <表名>：为新建表指定表名。
- NAME<长表名>：为新建表指定一个长表名。只有打开数据库，在数据库中创建表时，才能指定一个长表名。长表名最多可以包含 128 个字符。
- FREE：建立的表是自由表，不加入到打开的数据库中。当没有打开数据库时，建立的表都是自由表。
- <字段名 1><类型>[(<字段宽度>[, <小数位数>]]：指定字段名、字段类型、字段宽度及小数位数。字段类型可以用一个字符表示，如表 8-2 所示。

表 8-2 数据类型说明

字段类型	字段宽度	小数位	说明
C	n	-	字符型字段的宽度为 n
D	-	-	日期类型
T	-	-	日期时间类型
N	n	d	数值字段类型，宽度为 n，小数位为 d
F	n	d	浮点数值字段类型，宽度为 n，小数位为 d
I	-	-	整数类型
B	-	d	双精度类型
Y	-	-	货币类型
L	-	-	逻辑类型
M	-	-	备注类型
G	-	-	通用类型

- NULL|NOTNULL: NULL 允许该字段值为空。NOTNULL 字段值不能为空。默认值为 NOT NULL。
- CHECK <逻辑表达式 1>: 指定字段的合法值及字段值的约束条件。
- ERROR <字符型文本信息 1>: 指定在浏览或编辑窗口中该字段输入的值不符合 CHECK 子句的合法值时, Visual FoxPro 显示的错误信息。
- DEFAULT<表达式 1>: 为字段指定一个默认值, 表达式的数据类型与字段的数据类型要一致。即每添加一条记录时, 字段自动取该默认值。
- PRIMARY KEY|UNIQUE: PRIMARY KEY 为字段创建一个主索引, 索引标识名与字段名相同。主索引字段值必须唯一。UNIQUE 为字段创建一个候选索引, 索引标识名与字段名相同。
- REFERENCES <表名 2>[TAG<标识名 1>]: 指定建立永久关系的父表, 同时以指定字段为索引关键字建立外索引, 用字段名作为索引标识名。表名为父表名, 标识名为父表中的索引标识名。如果省略索引标识名, 则用父表的主索引关键字建立关系, 否则不能省略。如果指定了索引标识名, 则在父表中存在索引标识字段上建立的关系。父表不能是自由表。
- CHECK <逻辑表达式 2>[ERROR <字符型文本信息 2>]: 由逻辑表达式指定表的合法值。不合法时, 显示由字符型文本信息指定的错误信息。该信息只有在浏览或编辑窗口中修改数据时显示。
- FROM ARRAY <数组名>: 由数组创建表结构。数组名指定的数组包含表的每一个字段的字段名、字段类型、字段宽度及小数位数。

例: 建立一个学生情况表 (Std)。

```
CREATE TABLE Std (
    St_class C(8),:
    St_no C(10) PRIMARY KEY NOT NULL,:
    St_name C(8) NOT NULL,:
    St_sex C(2),:
    St_age N(2))
```

例: 建立一个课程情况表 (Sub)。

```
CREATE TABLE Sub (
    Su_no C(4) PRIMARY KEY NOT NULL,:
    Su_subject C(20) NOT NULL,:
    Su_credit N(6,2),:
    Su_period N(3),:
    Su_preno C(4))
```

例: 建立一个学生选课表 (Sco)。

```
CREATE TABLE Sco (
    St_no C(10),:
    Su_no C(4),:
    Sc_score N(3) NULL,:
    FOREIGN KEY St_no TAG St_no REFERENCE Std,:
    FOREIGN KEY Su_no TAG Su_no REFERENCE Sub)
```

注意:

- 用 SQL CREATE 命令新建的表自动在最低可用工作区打开, 并可以通过别名引用, 新表的打开方式为独占方式 (忽略 SET EXCLUSIVE 的当前设置)。
- 如果建立自由表 (当前没有打开的数据库或使用了 FREE), 则很多选项在命令中不能使用, 如 NAME、CHECK、DEFAULT、FOREIGN KEY、PRIMARY KEY 和 REFERENCES 等。

创建表的命令在书写时要注意下述原则:

- 在一对圆括号里列出完整的字段清单。
- 字段名间用逗号隔开。
- 字段名间的逗号后要加一个空格。

- 最后一个字段名后不用逗号。
- 所有的 SQL 陈述都以分号 “;” 结束。

8.2.2 表的删除

删除数据库表的命令如下：

```
DROP TABLE [<数据库名>.]<表名>
```

其中：

<表名>：是待删除的数据库表名。

使用该命令，可以直接从磁盘上删除<表名>所对应的“.dbf”文件。如果<表名>是数据库中的表，并且相应的数据库是当前数据库，则从数据库中删除该表；否则虽然从磁盘上删除了“.dbf”文件，但是记录在数据库“.dbc”文件中的信息却没有删除，此后会出现错误提示。所以要删除数据库中的表时，最好应使数据库是当前打开的数据库，在数据库中进行操作。

8.2.3 表结构的修改

在基本表建立并使用一段时间后，可能需要根据实际要求对基本表的结构进行修改。例如，增加或删除字段，定义和修改表的字段有效性规则等，而这些操作都可以直接通过 SQL 语句实现。

修改表结构的 SQL 命令为 ALTER TABLE 命令，我们可以通过以下 3 种命令格式修改表结构。

1. 格式 1

该格式可以添加（ADD）新的字段或修改（ALTER）已有的字段，各个参数的意义和创建表的命令相似。

```
ALTER TABLE <表名 1>
    ADD|ALTER [COLUMN] <字段名><类型>[( <字段宽度>[, <小数位数>])
    [NULL|NOT NULL]
    [CHECK <逻辑表达式>[ERROR <字符型文本信息>]]
    [DEFAULT <表达式>]
    [PRIMARY KEY|UNIQUE]
    [REFERENCES <表名 2>[TAG <标识名>]]
```

其中：

ADD|ALTER [COLUMN] <字段名>：向表中添加新的字段或者修改已有的字段。

使用该命令，可以修改字段的类型、宽度、有效性规则、错误信息、默认值，定义主关键字和联系等。但是不能修改字段名，不能删除字段，也不能删除已经定义的字段有效性规则等。

2. 格式 2

该格式可以定义、修改和删除字段有效性规则和默认值定义。

```
ALTER TABLE <表名>
    ALTER [COLUMN] <字段名>
    [NULL|NOT NULL]
    [SET DEFAULT <表达式>]
    [SET CHECK <逻辑表达式>[ERROR <字符型文本信息>]]
    [DROP DEFAULT]
    [DROP CHECK]
```

其中：

- ALTER [COLUMN] <字段名>：指出要修改列的字段名。
- NULL|NOT NULL：指定字段可以为空或不能为空。

- SET DEFAULT <表达式>: 重新设置字段的默认值。
 - SET CHECK <逻辑表达式>[ERROR <字符型文本信息>]: 重新设置该字段的合法值, 要求该字段的原有数据满足合法值。
 - DROP DEFAULT: 删除默认值。
 - DROP CHECK: 删除该表的合法值限定。
- 使用该命令, 不能修改字段名, 不能删除字段。

3. 格式 3

该格式可以删除字段 (DROP [COLUMN]), 可以修改字段名 (RENAME COLUMN), 可以定义、修改和删除表一级的有效性规则等。

```
ALTER TABLE <表名 1>
    [DROP [COLUMN] <字段名 1>]
    [SET CHECK <逻辑表达式 1>[ERROR <字符型文本信息>]]
    [DROP CHECK]
    [ADD PRIMARY KEY <表达式 1> TAG <标识名 1> [FOR <逻辑表达式 2>]]
    [DROP PRIMARY KEY]
    [ADD UNIQUE <表达式 2> [TAG <标识名 2> [FOR <逻辑表达式 3>]]]
    [DROP UNIQUE TAG <标识名 3>]
    [ADD FOREIGN KEY [<表达式 3>] TAG <标识名 4> [FOR <逻辑表达式 4>]
    REFERENCES <表名 2> [TAG <标识名 5>]]
    [DROP FOREIGN KEY TAG <标识名 6> [SAVE]]
    [RENAME COLUMN <字段名 2> TO <字段名 3>]
```

其中:

- DROP [COLUMN] <字段名 1>: 从指定表中删除指定的字段。
- SET CHECK <逻辑表达式>[ERROR <字符型文本信息>]: 重新设置该字段的合法值, 要求该字段的原有数据满足合法值。
- DROP CHECK: 删除该表的合法值限定。
- ADD PRIMARY KEY <表达式 1> TAG <标识名 1>: 为该表建立主索引, 一个表只能有一个主索引。
- ADD UNIQUE <表达式 2> [TAG <标识名 2>]: 为该表建立候选索引, 一个表可以有多个候选索引。
- ADD FOREIGN KEY [<表达式 3>] TAG <标识名 4>: 为该表建立外索引 (非主索引), 与指定的父表建立关系, 一个表可以有多个外索引。
- DROP PRIMARY KEY: 删除主索引。
- DROP UNIQUE TAG <标识名 3>: 删除候选索引。
- DROP FOREIGN KEY TAG <标识名 6>: 删除外索引 (外部关键字), 取消与父表的关系, SAVE 子句将保存该索引。
- RENAME COLUMN <字段名 2> TO <字段名 3>: 修改字段名, <字段名 2>指定要修改的字段名, <字段名 3>指定新的字段名。

8.2.4 视图的定义

在 Visual FoxPro 中, 视图是一个定制的虚拟表, 可以是本地的、远程的或带参数的。视图可以引用一个或多个表 (此时表称作基本表), 或者引用其他视图。视图是可更新的, 它可以引用远程表。

视图是根据对表的查询定义的, 其命令格式如下:

```
CREATE VIEW <视图名>
    AS SELECT 语句
```

其中:

SELECT 语句: 可以是任意的 SELECT 查询语句, 它说明和限定了视图中的数据; 视图的字段名将与 SELECT 语句

中指定的字段名或表中的字段名同名。

视图可以从单个表派生，也可以从多个表派生，此时视图定义语句中的 SELECT 语句采用多表查询或嵌套查询。

1. 从单个表派生出的视图

例：从 Stu 表中定义一个包括男生情况的视图。

```
CREATE VIEW male AS;
    SELECT * FROM Stu;
    WHERE St_Sex= " 男 "
```

2. 从多个表派生出的视图

例：从 Stu 表、Sco 表中定义一个分数在 80 分以上的女学生的姓名、班级和成绩的视图。

```
CREATE VIEW women_score AS;
    SELECT St_Name, St_Class, So_Score;
FROM Stu, Sco;
WHERE St_Sex= " 女 " AND So_Score>=80;
AND St_Sno= So_Sno
```

3. 视图中的虚字段

创建视图的 SELECT 子句可以包含算术表达式或函数，它们与视图中的其他字段一样对待，由于它们是计算得来的，并不存储在表内，所以称为虚字段。

例：定义一个视图，包括职工号、基本工资、奖金、津贴、总工资等字段。

```
CREATE VIEW salary AS;
    SELECT 职工号, 基本工资, 奖金, 津贴, 基本工资+奖金+津贴 AS 总工资;
FROM 职工;
```

4. 视图的删除

视图由于从表中派生出来，所以不存在修改结构的问题，但是视图可以删除，删除视图的命令格式如下：

```
DROP VIEW <视图名>
```

5. 关于视图的说明

在关系数据库中，对视图可以进行查询，但是进行插入、更新和删除操作却有一定限制。在一般情况下，当一个视图是由单个表导出时可以进行插入和更新操作，但不能进行删除操作；当视图是由多个表导出时，插入、更新和删除操作都不允许进行。

8.3 操作功能

Visual FoxPro 中 SQL 的操作功能，包括数据的插入、数据的更新和数据的删除等 3 个方面的功能。

8.3.1 插入数据

INSERT 命令用于在一个表中添加新记录，然后给新记录的字段赋值。Visual FoxPro 支持下述两种 SQL 插入命令格式。

1. 标准格式

```
INSERT INTO <表名> [( <字段名 1>[, <字段名 2>, ...])]
VALUES (<表达式 1>[, <表达式 2>, ...])
```

其中：

- <表名>：指明要插入新记录的表名。
- (<字段名 1>[, <字段名 2>, ...])：字段名表，当插入的不是完整的记录时，用其指定字段。如果省略字段名表，则按照表结构中字段定义的顺序一次指定每个字段的值。
- VALUES：给出新记录中指定字段的字段值。插入的字段值类型要与表中对应字段的类型相一致。

2. 特殊格式

```
INSERT INTO <表名> FROM ARRAY <数组名>|FROM MEMVAR
```

其中：

- FROM ARRAY <数组名>：从指定的数组中插入记录值。
- FROM MEMVAR：根据同名的内存变量来插入记录值，如果同名的变量不存在，那么相应的字段为默认值或空值。

3. 关于插入操作的说明

一般而言，使用上述两种格式都可以插入数据，但是如果一个表定义了主索引或候选索引后，由于相应的字段具有关键字的特性，即不能为空，所以只能用第二种格式插入记录。

8.3.2 更新数据

SQL 的数据更新命令格式如下：

```
UPDATE <表名>  
SET <字段名 1>=<表达式 1>[, <字段名 2>=<表达式 2>...]  
WHERE <条件>
```

其中：

- <表名>：指定需要更新数据所在的表名。
- SET：指定需要更新的字段及它们的新值。
- <字段名 1>：指定需要更新的字段名称。
- <表达式 1>：给出更新后对应字段的值。
- WHERE <条件>：指定更新字段满足的条件。用来更新满足条件的一些记录的字段值，一次可以更新多个字段；如果不使用 WHERE 子句，则更新全部记录。

8.3.3 删除数据

SQL 的数据删除命令格式如下：

```
DELETE FROM <表名> [WHERE <条件>]
```

其中：

- <表名>：指定需要删除数据所在的表名。
- WHERE <条件>：指定被删除记录满足的条件。用来删除满足条件的一些记录，一次可以删除多个记录；如果不使用 WHERE 子句，则删除全部记录。

在 Visual FoxPro 中，SQL DELETE 命令仅仅是逻辑删除记录，如果要物理删除记录，那么还需要继续使用 PACK 命令。

8.4 查询功能

数据查询是指根据给定的条件从关系中找到符合条件的元组。查询功能是 SQL 语言的核心，SQL 语言的查询命令又称为 SELECT 命令，它的基本形式是 SELECT-FROM-WHERE。

SELECT 命令的语法格式如下：

```
SELECT [ALL|DISTINCT][TOP <数值表达式> [PERCENT]]
[<别名>.]<SELECT 查询项> [AS <列名>], [<别名>.]<SELECT 查询项> [AS <列名>]...
FROM [FORCE] [<数据库名>!]<表名> [[AS]<别名>]
[[INNER|LEFT[OUTER]|RIGHT[OUTER]]|FULL[OUTER] JOIN [<数据库名>!]<表名> [[AS]<别名>]
ON <连接条件>...]
[[INTO <目标>]|[TO FILE <文件名> [ADDITIVE]]|TO PRINTER [PROMPT]]|TO SCREEN]]
[PREFERNCE <>]
[NOCONSOLE]
[PLAIN]
[NOWAIT]
[WHERE <连接条件>[AND <连接条件>]][AND|OR <筛选条件>[AND|OR <筛选条件>...]]]
[GROUP BY <分组条件>[, <分组条件>...]]
[HAVING <筛选条件>]
[UNIO[ALL] SELECT 命令]
[ORDER BY <排序表达式>[ASC|DESC][, <排序表达式>[ASC|DESC]...]]
```

其中：

- SELECT：说明要查询的数据。
- FROM：说明要查询的数据来自哪个表或哪些表，可以基于单个表或多个表进行查询。
- DISTINCT：去掉查询结果中的重复记录。
- WHERE：说明查询条件，即选择元组的条件。
- GROUP BY：用于对查询结果进行分组，可以利用它进行分组汇总。
- HAVING：与 GROUP BY 同时使用，用来限定分组的条件。
- ORDER BY：用来对查询结果进行排序。

SELECT 命令从格式上看似乎非常复杂，实际上只要了解命令中各个短语的含义，难度应该不是很大。以其功能为导向，从功能实现角度分解命令，各个击破，由浅入深，循序渐进，是一个不错的办法。

8.4.1 简单查询

简单查询是基于单个表的查询，可以是由 SELECT 和 FROM 短语构成的无条件查询，或者是由 SELECT、FROM 和 WHERE 短语构成的条件查询。如果有 WHERE 子句，系统首先根据指定条件依次检验关系中的每个元组；如果没有指定 WHERE 子句，则不进行这样的检验。执行完检验后，系统选出满足条件的元组（相当于关系的选择操作），并显示 SELECT 子句中指定属性的值（相当于关系中的投影操作）。

基本格式：

```
SELECT <字段名列表> FROM <表名> WHERE <查询条件>
```

其中：

- <字段名列表>：指明查询结果中包含的字段名，多个字段名之间需要用逗号分隔。
- <查询条件>：是一个逻辑表达式，由多个表达式通过逻辑运算符 NOT、AND 和 OR 连接而成。

例：从 Stu 表中查询年龄大于 20 岁的女学生的学号、班级、姓名。

```
SELECT St_no, St_class, St_name FROM Stu;
WHERE St_age>=20
```

关系表达式中可以使用的关系运算符如表 8-3 所示。

表 8-3 SELECT 关系运算符

运算符	含义	运算符	含义
=	等于	<	小于
<>, !=, #	不等于	<=	小于等于
==	精确等于	BETWEEN...AND	在两组值之间
>	大于	IN	在一组值范围内
>=	大于等于	LIKE	字符匹配运算法，%表示与若干个任意字符匹配； _表示与一个任意字符匹配
IS NULL	为空值		

8.4.2 简单的连接查询

连接是关系的基本操作之一，连接查询是一种基于多个关系的查询。在这种查询中，检索命令的 FROM 之后有多个关系，并且这些多个关系之间存在一些联系，否则无法构成检索表达式。当 FROM 之后的多个关系中含有相同的属性名时，这时必须用关系前缀 “.” 直接指明属性所属的关系。

基本格式：

```
SELECT <字段名列表> FROM <表名 1, 表名 2, ...> WHERE <连接条件>
```

其中：

- <字段名列表>：指定查询显示字段。
- <连接条件>：指定连接条件。
- FROM <表名 1, 表名 2, ...>：指定需要连接的表。

例：查询所有学生的姓名、课程号和成绩。

```
SELECT St_name, Su_no, Sc_score;  
FROM Stu, Sco;  
WHERE Stu.St_no=Sco.St_no
```

8.4.3 嵌套查询

嵌套查询是一种基于多个关系的查询，这类查询所要求的结果出自一个关系，但相关的条件却涉及多个关系。这类查询命令中一般含有两个 SELECT-FROM-WHERE 查询块，即内层查询块和外层查询块。内层查询块和外层查询块中的 WHERE 相结合，确定查询条件；外层查询块确定查询时选择的属性。

例：查询和学号为“114”的同学年龄一样的所有学生的学号、课程号和成绩。

```
SELECT St_name, Su_no, Sc_score;  
FROM Stu, Sco;  
WHERE Stu.St_no=Sco.St_no AND Stu.St_no=" 114 " AND Stu.St_age=;  
(SELECT Stu.St_age FROM Stu WHERE Stu.St_no=" 114 " )
```

8.4.4 几个特殊运算符

在实际应用中，恰当使用下述特殊运算符，可以增强 SQL SELECT 查询的功能。

- (1) BETWEEN...AND。适用于查询的条件在什么范围之内，即“在……之间”，可以使表达条件更清晰、更简洁。
- (2) LIKE。字符串匹配运算符，适用于字符串匹配的查询，其常用的通配符有下面两种。
 - “%”：表示 0 个或多个字符；
 - “_”：表示 1 个字符。
- (3) NOT。否定运算符，适用于“不等于”或求相反的请求。

例：查询所有年龄在 18~20 岁之间学生的信息。

```
SELECT * FROM Stu WHERE St_age BETWEEN 18 AND 20
```

例：查询所有名字中有“红”字的学生信息。

```
SELECT * FROM Stu WHERE St_name LIKE "%红%"
```

8.4.5 查询结果排序

使用 SQL SELECT 可以将查询结果排序，排序的短语是 ORDER BY。可以按升序（ASC）或降序（DESC）排序，也可以按一列或多列排序。

具体格式如下：

```
ORDER BY <排序项 1> [ASC|DESC] [, <排序项 1> [ASC|DESC]...]
ORDER BY 是对最终的查询结果进行排序，不可以再在子查询中使用该短语。
```

例：以班级号降序、学号升序显示 Stu 中所有记录。

```
SELECT * FROM Stu ORDER BY St_class DESC, St_no ASC
```

8.4.6 简单的计算查询

SQL 不仅具有一般查询能力，还具有计算方式查询能力。其形式可以是数值表达式、字符串常量、函数等。其中，用于计算查询的函数如表 8-4 所示。

表 8-4 SQL 计算函数

函数名	含义	函数名	含义
COUNT	用于计数计算	SUM	用于求和计算
AVG	用于求平均值计算	MAX	用于求最大值
MIN	用于求最小值		

例：在产品表和订单明细表中查询订单的数量和所有订单的平均金额。

```
SELECT COUNT(DISTINCT 订单号), AVG(数量*单价);
FROM 产品 JOIN 订单明细 ON 产品.产品号=订单明细.产品号
```

例：在 Sco 表中查询最低的成绩分数。

```
SELECT MIN(Sc_score) AS 最低分 FROM Sco
```

8.4.7 分组与计算查询

利用 GROUP BY 子句，可以进行分组计算查询。该子句可以按一列或多列分组，还可以用 HAVING 进一步限定分组的条件。

该子句的格式如下：

```
GROUP BY <分组条件>[, <分组条件>...] [HAVING <筛选条件>]
```

其中：

- GROUP BY 子句一般跟在 WHERE 子句之后，当没有 WHERE 子句时，跟在 FROM 子句之后。
- 可以根据多个属性进行分组。
- 当要求分组满足某个条件时才查询，可以用 HAVING 子句来限定分组。
- HAVING 子句总是跟在 GROUP BY 子句之后，不可以单独使用。

- HAVING 子句和 WHERE 子句并不矛盾，在查询中首先用 WHERE 子句限定分组，然后进行分组，最后再用 HAVING 子句限定分组。

例：统计每个班级的学生人数。

```
SELECT St_class, COUNT(St_no) FROM Std GROUP BY St_class
```

例：统计至少有 30 个学生的班级的平均分数。

```
SELECT St_class, AVG(Sc_score) AS 平均分 FROM Std, Sco;
WHERE Stu.St_no=Sco.St_no;
GROUP BY St_class HAVING COUNT(*)>=30
```

8.4.8 利用空值查询

SQL 支持空值，可以利用空值查询，通常在 WHERE 子句中采用“字段名 IS NULL”或“字段名 IS NOT NULL”来限定条件。

查询空值时要使用“IS NULL”，而“=NULL”是无效的，因为空值不是一个确定的值，所以不能用“=”这样的运算符进行比较。

例：查询尚未确定课程成绩的学生记录。

```
SELECT * FROM Sco WHERE Sc_score IS NULL
```

例：查询已经确定课程成绩的学生记录。

```
SELECT * FROM Sco WHERE Sc_score IS NOT NULL
```

8.4.9 别名与自连接查询

SQL 允许在 FROM 短语中为关系名定义别名，其格式为：

<关系名><别名>

在连接查询中，使用别名可以简化查询语句。在一般的连接查询中，别名并不是必需的，但是在关系的自连接操作中，别名必不可少。

SQL 不仅可以对多个关系实行连接操作，也可以将同一关系与其自身进行连接，这种连接就称为自连接。自连接关系存在着一种特殊的递归联系，即关系中的一些元组可以根据出自同一值域的两个不同的属性，与另外一些元组有一种对应关系（一对多的联系）。

例：在 Sub 表中查询所有先修课的课程及其先修课名称。

```
SELECT A.Su_subject, "的先修课是", B.Su_subject FROM Sub A, Sub B;
WHERE Su_no=Su_preno
```

8.4.10 内、外层相互关联查询

在这种查询中，内、外层查询相互关联，内层查询的条件需要外层查询提供值，外层查询的条件需要内层查询的结果。

例：查询课程成绩低于该课程平均成绩的学生课程成绩。

```
SELECT St_no, Su_no, Sc_score FROM Sco A;
WHERE Sc_score<=;
(SELECT AVG(Sc_score) FROM Sco B;
WHERE a.Su_no=b.Su_no)
```

8.4.11 使用量词和谓词的查询

这种查询和子查询相关，有下面两种形式。

- (1) <表达式> <比较运算符> [ANY|ALL|SOME] (子查询)
- (2) [NOT] EXISTS (子查询)

其中，ANY、ALL 和 SOME 是量词。ANY 和 SOME 二者同义，在进行比较运算时只要子查询中有一行能使结果为真，则结果就为真；而 ALL 则要求子查询中的所有行都使结果为真时，结果才为真。

EXISTS 是谓词，EXISTS 或 NOT EXISTS 用来检查在子查询中是否有结果返回，即是否存在元组，但它本身没有任何运算或比较。

只有当内层查询引用外层查询的表时，使用谓词 EXISTS 或 NOT EXISTS 才有意义。

例：查询所有任课教师的姓名和所在院系。

```
SELECT Tea_name, Tea_depart FROM Teacher A;
WHERE EXIST;
(SELECT * FROM Sub B;
WHERE A.Tea_no=B.Tea_no)
以上查询等价于：
SELECT Tea_name, Tea_depart FROM Teacher A;
WHERE A.Tea_no=;
(SELECT * FROM Sub B;
WHERE A.Tea_no=B.Tea_no)
```

8.4.12 超连接查询

在一般的 SQL 中超连接运算符是“*=”和“=*”。其中“*=”称为左连接，含义是在结果表中包含第一个表中满足条件的所有记录；如果有在连接条件上匹配的元组，则第二个表返回相应值，否则第二个表返回空值。而“=*”称为右连接，含义是在结果表中包含第二个表中满足条件的所有记录；如果有在连接条件上匹配的元组，则第一个表返回相应值，否则第一个表返回空值。

而 Visual FoxPro 不支持超连接运算符“*=”和“=*”，它有专门的连接运算语法格式来支持超连接查询，具体语法格式如下：

```
SELECT ...
FROM <表名 1> INNER|LEFT|RIGHT|FULL JOIN <表名 2>
ON <连接条件>
WHERE ...
```

其中：

- INNER：“INNER JOIN”等价于 JOIN，为普通的连接，在 Visual FoxPro 中称为内部连接。
- LEFT：“LEFT JOIN”称为左连接。
- RIGHT：“RIGHT JOIN”称为右连接。
- FULL：“FULL JOIN”称为全连接，即两个表中的记录不管是否满足连接条件都将在目标表或查询结果中出现，不满足连接条件的记录对应部分为 NULL。
- ON <连接条件>：指定两个表的连接条件。

在这种查询中，连接条件在 ON 短语中给出，而不在 WHERE 短语中；连接类型在 FROM 短语中给出。

JOIN 连接格式在连接多个表时的书写方法要特别注意，在这种格式中 JOIN 的顺序和 ON 的顺序（相应的连接条件）正好相反。

例：从 Std 和 Sco 表中查询所有有考试成绩的学生的学号、姓名、班级、课程号和成绩。

```
SELECT Std.St_no, Std.St_name, Std.St_class, Sco.Su_no, Sco.Sc_score FROM Std INNER JOIN Sco;
ON Std.St_no=Sco.St_no
```

例：从 Std 和 Sco 表中查询所有学生的学号、姓名、班级、课程号和成绩，包括没有考试成绩的学生。

```
SELECT Std.St_no, Std.St_name, Std.St_class, Sco.Su_no, Sco.Sc_score FROM Std LEFT JOIN Sco;
ON Std.St_no=Sco.St_no
```

例：从 Sub 和 Sco 表中查询所有学生的学号、课程号、课程名和成绩，包括没有考试成绩的课程。

```
SELECT Sco.St_no, Sco.Su_no, Sub.Su_subject, Sco.Sc_score FROM Sco RIGHT JOIN Sub;
ON Sco.Su_no=Sub.Su_no
```

8.4.13 集合的并运算

SQL 中集合的并（UNION）运算，可以将两个 SELECT 语句的查询结果通过并运算合并成一个查询结果。为了进行并运算，要求这样的两个查询结果具有相同的字段个数，并且对应字段的值要出自同一个值域，即具有相同的数据类型和取值范围。

例：查询信息管理 09 班的男生和外语 08 班的女生的名单。

```
SELECT * FROM Std;
WHERE St_class=" 管理 09 " AND St_sex=" 男 ";
UNION;
SELECT * FROM Std WHERE St_class=" 外语 08 " AND St_sex=" 女 "
```

8.4.14 Visual FoxPro SQL SELECT 的几个特殊选项

在 Visual FoxPro 的 SELECT 语句中，可以使用具有下述功能的特殊选项。

1. 显示部分结果

使用 TOP <数值表达式> [PERCENT] 短语可以显示满足条件的前几条记录。

其中，当不使用 PERCENT 时，<数值表达式> 是 1~32767 间的整数，说明显示前几个记录；当使用 PERCENT 时，<数值表达式> 是 0.01~0.99 之间的实数，说明显示结果中前百分之几的记录。

TOP 短语要与 ORDER BY 短语同时使用才有效。

例：查询成绩最高的 10 位同学的信息。

```
SELECT * TOP 10 FROM Sco ORDER BY Sc_score DESC
```

2. 将查询结果存放到数组中

使用 INTO ARRAY <数组名> 短语可以将查询结果存放到数组中。

其中，<数组名> 可以是任意的数组变量名。一般将存放查询结果的数组作为二维数组来使用，每行一条记录，每列对应查询结果的一列。

例：查询班级为“管理 09”的学生信息并将结果存入 ARRAY1 数组中。

```
SELECT * FROM Std INTO ARRAY ARRAY1;
WHERE St_class=" 管理 09 "
```

3. 将查询结果存放到临时文件中

使用 INTO CURSOR <临时文件名> 短语可以将查询结果存放到临时表文件中。

其中，<临时文件名> 是临时的表文件名，该短语产生的临时文件是一个只读的 .dbf 文件。查询结束后该临时文件是当前文件，可以像一般的 .dbf 文件一样使用，但仅是只读。当关闭文件时该文件将自动删除。

例：将查询到的课程信息存放到临时文件 Temp1 中。

```
SELECT * FROM Sub INTO CURSOR Temp1
```

4. 将查询结果存放到永久表中

使用短语 INTO DBF|TABLE <表名>可以将查询结果存放到永久表中（.dbf 文件），该表是自由表。

例：将 Std 表中女生的记录存放到表 Women 中。

```
SELECT * FROM Std INTO Women WHERE St_sex="女"
```

5. 将查询结果存放到文本文件中

使用短语 TO FILE <文件名>[ADDITIVE]可以将查询结果存放到文本文件中。

其中，<文件名>给出了文本文件名（默认扩展名是.txt）；如果使用 ADDITIVE，结果将追加到原文件的尾部，否则将覆盖原有文件。

例：将 Std 表中女生的记录存放到文本文件 Women.txt 中。

```
SELECT * FROM Std TO FILE Women WHERE St_sex="女"
```

6. 将查询结果直接输出到打印机

使用短语 TO PRINTER [PROMPT]可以直接将查询结果输出到打印机。

如果使用了 PROMPT 选项，则在开始打印之前会打开打印机设置对话框。

8.5 本章习题

8.5.1 选择题

第 1~8 题基于下面内容。

当前盘当前目录下有数据库：学院.dbc，其中有“教师”表和“学院”表。

“教师”表：

职工号	系号	姓名	工资	主讲课程
11020001	01	肖海	3408	数据结构
11020002	02	王岩盐	4390	数据结构
11020003	01	刘星魂	2450	C 语言
11020004	03	张月新	3200	操作系统
11020005	01	李明玉	4520	数据结构
11020006	02	孙民山	2976	操作系统
11020007	03	钱无名	2987	数据库
11020008	04	呼延军	3220	编译原理
11020009	03	王小龙	3980	数据结构
11020010	01	张国梁	2400	C 语言
11020011	04	林新月	1800	操作系统
11020012	01	乔小廷	5400	网络技术
11020013	02	周兴池	3670	数据库
11020014	04	欧阳秀	3345	编译原理

“学院”表：

系号	系名

- 01 计算机
- 02 通信
- 03 信息管理
- 04 数学

1. 为“学院”表增加一个字段“教师人数”的 SQL 语句是 ()。
 - A. CHANGE TABLE 学院 ADD 教师人数 I
 - B. ALTER STRU 学院 ADD 教师人数 I
 - C. ALTER TABLE 学院 ADD 教师人数 I
 - D. CHANGE TABLE 学院 INSERT 教师人数 I
2. 将“欧阳秀”的工资增加 200 元的 SQL 语句是 ()。
 - A. REPLACE 教师 WITH 工资=工资+200 WHERE 姓名="欧阳秀"
 - B. UPDATE 教师 SET 工资=工资+200 WHEN 姓名="欧阳秀"
 - C. UPDATE 教师工资 WITH 工资+200 WHERE 姓名="欧阳秀"
 - D. UPDATE 教师 SET 工资=工资+200 WHERE 姓名="欧阳秀"
3. 有 SQL 语句: SELECT * FROM 教师 WHERE NOT(工资>3000 OR 工资<2000)
与如上语句等价的 SQL 语句是 ()。
 - A. SELECT * FROM 教师 WHERE 工资 BETWEEN 2000 AND 3000
 - B. SELECT * FROM 教师 WHERE 工资>2000 AND 工资<3000
 - C. SELECT * FROM 教师 WHERE 工资>2000 OR 工资<3000
 - D. SELECT * FROM 教师 WHERE 工资<=2000 AND 工资>=3000
4. 为“教师”表的职工号字段添加有效性规则: 职工号的最左边 3 位字符是 110, 正确的 SQL 语句是 ()。
 - A. CHANGE TABLE 教师 ALTER 职工号 SET CHECK LEFT(职工号,3)="110"
 - B. ALTER TABLE 教师 ALTER 职工号 SET CHECK LEFT(职工号,3)="110"
 - C. ALTER TABLE 教师 ALTER 职工号 CHECK LEFT(职工号,3)="110"
 - D. CHANGE TABLE 教师 ALTER 职工号 SET CHECK OCCURS(职工号,3)="110"
5. 有 SQL 语句:
SELECT DISTINCT 系号 FROM 教师 WHERE 工资>=;
ALL(SELECT 工资 FROM 教师 WHERE 系号="02")
该语句的执行结果是系号 ()。
 - A. "01"和"02"
 - B. "01"和"03"
 - C. "01"和"04"
 - D. "02"和"03"
6. 有 SQL 语句:
SELECT 主讲课程, COUNT(*)FROM 教师 GROUP BY 主讲课程
该语句执行结果含有记录个数是 ()。
 - A. 3
 - B. 4
 - C. 5
 - D. 6
7. 有 SQL 语句:
SELECT 学院, 系名,COUNT(*)AS 教师人数 FROM 教师,学院;
WHERE 教师.系号=学院.系号 GROUP BY 学院.系名
与如上语句等价的 SQL 语句是 ()。
 - A. SELECT 学院.系名, COUNT(*)AS 教师人数; FROM 教师 INNER JOIN 学院;
教师.系号=学院.系号 GROUP BY 学院.系名
 - B. SELECT 学院.系名, COUNT(*)AS 教师人数; FROM 教师 INNER JOIN 学院;
ON 系号 GROUP BY 学院.系名
 - C. SELECT 学院.系名, COUNT(*)AS 教师人数; FROM 教师 INNER JOIN 学院;
ON 教师.系号=学院.系号 GROUP BY 学院.系名
 - D. SELECT 学院.系名, COUNT(*)AS 教师人数; FROM 教师 INNER JOIN 学院;
ON 教师.系号=学院.系号

8. 有 SQL 语句:

SELECT DISTINCT 系号 FROM 教师 WHERE 工资>=;

ALL(SELECT 工资 FROM 教师 WHERE 系号="02")

与如上语句等价的 SQL 语句是 ()。

A. SELECT DISTINCT 系号 FROM 教师 WHERE 工资>=

(SELECT MAX(工资) FROM 教师 WHERE 系号="02")

教师.系号 = 学院.系号 GROUP BY 学院.系名

B. SELECT DISTINCT 系号 FROM 教师 WHERE 工资>=

(SELECT MIN(工资) FROM 教师 WHERE 系号="02")

C. SELECT DISTINCT 系号 FROM 教师 WHERE 工资>=

ANY(SELECT 工资 FROM 教师 WHERE 系号="02")

D. SELECT DISTINCT 系号 FROM 教师 WHERE 工资>=

SOME(SELECT 工资 FROM 教师 WHERE 系号="02")

9. 在 Visual FoxPro 中, 以下关于删除记录的描述, 正确的是 ()。

A. SQL 的 DELETE 命令在删除数据库表中的记录之前, 不需要用 USE 命令打开表

B. SQL 的 DELETE 命令和传统 Visual FoxPro 的 DELETE 命令在删除数据库表中的记录之前, 都需要用 USE 命令打开表

C. SQL 的 DELETE 命令可以物理删除数据库表中的记录, 而传统的 Visual FoxPro 的 DELETE 命令只能逻辑删除数据库表中的记录

D. 传统 Visual FoxPro 的 DELETE 命令在删除数据库表中的记录之前不需要用 USE 命令打开表

10. 使用 SQL 语句将学生表 S 中年龄 (AGE) 大于 30 岁的记录删除, 正确的命令是 ()。

A. DELETE FOR AGE>30

B. DELETE FROM S WHERE AGE>30

C. DELETE S FOR AGE>30

D. DELETE S WHERE AGE>30

11. 使用 SQL 语句向学生表 S (SNO, SN, SEX, AGE) 中添加一条新记录, 字段学号 (SNO)、姓名 (SN)、性别 (SEX)、年龄 (AGE) 的值分别为 0401、王芳、女、18, 正确的命令是 ()。

A. APPEND INTO S(SNO,SN,SEX,AGE) VALUES (' 0401' , ' 王芳' , ' 女' ,18)

B. APPEND S VALUES (' 0401' , ' 王芳' ,18, ' 女')

C. INSERT INTO S(SNO,SN,SEX,AGE) VALUES (' 0401' , ' 王芳' , ' 女' ,18)

D. APPEND S VALUES (' 0401' , ' 王芳' ,18, ' 女')

第 12~16 题基于下面内容。

有如下 3 个表:

职员.DBF: 职员号 C (3), 姓名 C (6), 性别 C (2), 组号 N (1), 职务 C (10)

客户.DBF: 客户号 C (4), 客户名 C (36), 地址 C (36), 所在城市 C (36)

订单.DBF: 订单号 C (4), 客户号 C (4), 职员号 C (3), 签订日期 D, 金额 N (6.2)

12. 查询金额最大的前 10% 订单的信息。正确的 SQL 语句是 ()。

A. SELECT * TOP 10 PERCENT FROM 订单

B. SELECT TOP 10% * FROM 订单 ORDER BY 金额

C. SELECT * TOP 10 PERCENT FROM 订单 ORDER BY 金额

D. SELECT TOP 10 PERCENT * FROM 订单 ORDER BY 金额 DESC

13. 查询订单数在 3 个以上、订单的平均金额在 200 元以上的职员号。正确的 SQL 语句是 ()。

A. SELECT 职员号 FROM 订单 GROUP BY 职员号 HAVING COUNT(*)>3 AND AVG_金额>200

B. SELECT 职员号 FROM 订单 GROUP BY 职员号 HAVING COUNT(*)>3 AND AVG(金额)>200

C. SELECT 职员号 FROM 订单 GROUP BY 职员号 HAVING COUNT(*)>3 WHERE AVG(金额)>200

D. SELECT 职员号 FROM 订单 GROUP BY 职员号 WHERE COUNT(*)>3 AND AVG_金额>200

14. 显示没有签订任何订单的职员信息（职员号和姓名），正确的 SQL 语句是（ ）。
A. SELECT 职员.职员号,姓名 FROM 职员 JOIN 订单
ON 订单.职员号=职员.职员号 GROUP BY 职员.职员号 HAVING COUNT(*)=0
B. SELECT 职员.职员号,姓名 FROM 职员 LEFT JOIN 订单
ON 订单.职员号=职员.职员号 GROUP BY 职员.职员号 HAVING COUNT(*)=0
C. SELECT 职员号,姓名 FROM 职员
WHERE 职员号 NOT IN (SELECT 职员号 FROM 订单)
D. SELECT 职员.职员号,姓名 FROM 职员
WHERE 职员.职员号 <> (SELECT 订单.职员号 FROM 订单)
15. 从订单表中删除客户号为“1001”的订单记录，正确的 SQL 语句是（ ）。
A. DROP FROM 订单 WHERE 客户号="1001" B. DROP FROM 订单 FOR 客户号="1001"
C. DELETE FROM 订单 WHERE 客户号="1001" D. DELETE FROM 订单 FOR 客户号="1001"
16. 将订单号为“0060”的订单金额改为 169 元，正确的 SQL 语句是（ ）。
A. UPDATE 订单 SET 金额=169 WHERE 订单号="0060"
B. UPDATE 订单 SET 金额 WITH 169 WHERE 订单号="0060"
C. UPDATE FROM 订单 SET 金额=169 WHERE 订单号="0060"
D. UPDATE FROM 订单 SET 金额 WITH 169 WHERE 订单号="0060"
17. SQL 的数据操作语句不包括（ ）。
A. INSERT B. UPDATE C. DELETE D. CHANGE
18. 从“订单”表中删除签订日期为 2004 年 1 月 10 日之前（含）的订单记录，正确的 SQL 语句是（ ）。
A. DROP FROM 订单 WHERE 签订日期<={^2004-1-10}
B. DROP FROM 订单 FOR 签订日期<={^2004-1-10}
C. DELETE FROM 订单 WHERE 签订日期<={^2004-1-10}
D. DELETE FROM 订单 FOR 签订日期<={^2004-1-10}
19. 在 SQL 的 SELECT 查询结果中，消除重复记录的方法是（ ）。
A. 通过指定主关系键 B. 通过指定唯一索引
C. 使用 DISTINCT 子句 D. 使用 HAVING 子句
20. 在 Visual FoxPro 中，以下有关 SQL 的 SELECT 语句的叙述中，错误的是（ ）。
A. SELECT 子句中可以包含表中的列和表达式 B. SELECT 子句中使用别名
C. SELECT 子句规定了结果集中的列顺序 D. SELECT 子句中列的顺序应该与表中列的顺序一致
21. 下列关于 SQL 中 HAVING 子句的描述，错误的是（ ）。
A. HAVING 子句必须与 GROUP BY 子句同时使用 B. HAVING 子句与 GROUP BY 子句无关
C. 使用 WHERE 子句的同时可以使用 HAVING 子句 D. 使用 HAVING 子句的作用是限定分组的条件
22. 使用 SQL 命令查询选修的每门课程的成绩都高于或等于 85 分的学生的学号和姓名，正确的命令是（ ）。
A. SELECT 学号,姓名 FROM S WHERE NOT EXISTS
(SELECT * FROM SC WHERE SC.学号=S.学号 AND 成绩<85)
B. SELECT 学号,姓名 FROM S WHERE NOT EXISTS
(SELECT * FROM SC WHERE SC.学号=S.学号 AND 成绩>=85)
C. SELECT 学号,姓名 FROM S,SC
WHERE S.学号=SC.学号 AND 成绩>=85
D. SELECT 学号,姓名 FROM S,SC
WHERE S.学号=SC.学号 AND ALL 成绩>=85
23. 假设“订单”表中有订单号、职员号、客户号和金额字段，正确的 SQL 语句只能是（ ）。
A. SELECT 职员号 FROM 订单

GROUP BY 职员号 HAVING COUNT(*)>3 AND AVG_金额>200

B. SELECT 职员号 FROM 订单

GROUP BY 职员号 HAVING COUNT(*)>3 AND AVG(金额)>200

C. SELECT 职员号 FROM 订单

GROUP,BY 职员号 HAVING COUNT(*)>3 WHERE AVG(金额)>200

D. SELECT 职员号 FROM 订单

GROUP BY 职员号 WHERE COUNT(*)>3 AND AVG_金额>200

24. 假设同一名称的产品有不同的型号和产地, 则计算每种产品平均单价的 SQL 语句是 ()。

A. SELECT 产品名称,AVG(单价)FROM 产品 GROUP BY 单价

B. SELECT 产品名称,AVG(单价)FROM 产品 ORDERBY 单价

C. SELECT 产品名称,AVG(单价)FROM 产品 ORDER BY 产品名称

D. SELECT 产品名称,AVG(单价)FROM 产品 GROUP BY 产品名称

25. SQL SELECT 语句的功能是 ()。

A. 定义

B. 查询

C. 修改

D. 控制

26. 在 SQL 语句中用于限定条件的短语是 ()。

A. FOR

B. WHILE

C. WHERE

D. CONDITION

第 27~35 题基于下面内容。

stock.dbf 的内容是:

股票代码	股票名称	单价	交易所
600600	青岛啤酒	7.48	上海
600601	方正科技	15.20	上海
600602	广电电子	10.40	上海
600603	兴业房产	12.76	上海
600604	二纺机	9.96	上海
600605	轻工机械	14.59	上海
000001	深发展	7.48	深圳
000002	深万科	12.50	深圳

27. 执行如下 SQL 语句后, ()。

SELECT*FROM stock INTO DBF stock ORDER BY 单价

A. 系统会提示语句出错

B. 会生成一个按“单价”升序排序的表文件, 将原来的 stock.dbf 文件覆盖

C. 会生成一个按“单价”降序排序的表文件, 将原来的 stock.dbf 文件覆盖

D. 不会生成排序文件, 只在屏幕上显示一个按“单价”升序排序的结果

28. 与 SELECT*FROM stock WHERE 单价 BETWEEN 12.76 AND 15.20 等价的语句是 ()。

A. SELECT*FROM stock WHERE 单价<=15.20 AND 单价>=12.76

B. SELECT*FROM stock WHERE 单价<1 5.20 AND 单价>12.76

C. SELECT*FROM stock WHERE 单价>=1 5.20.AND.单价<=12.76

D. SELECT*FROM stock WHERE 单价>15.20.AND.单价<12.76

29. 执行如下 SQL 语句的结果是 ()。

SELECT MAX (单价) INTO ARRAY arr FROM stock

A. arr[1]内容为 15.20

B. arr[1]的内容为 6

C. arr[0]内容为 15.20

D. arr[0]的内容为 6

30. 将 stock 的字段“股票名称”的宽度由 8 改为 10, 应使用 SQL 语句 ()。

SELECT MAX (单价) INTO ARRAY arr FROM stock

- A. ALTER TABLE stock 股票名称 WITH c (10)
- B. ALTER TABLE stock 股票名称 c (10)
- C. ALTER TABLE stock ALTER 股票名称 c (10)
- D. ALTER stock ALTER 股票名称 c (10)

31. 使用如下 SQL 语句:

```
CREATE VIEW stock_view AS
SELECT*FROM stock_temp WHERE 交易所="深圳"
建立的视图含有的记录个数是 ( )。
```

- A. 1
- B. 2
- C. 3
- D. 4

32. 使用如下 SQL 语句:

```
CREATE VIEW view_stock AS
SELECT 股票名称 AS 名称, 单价 FROM stock
建立的视图含有的字段名是 ( )。
```

- A. 股票名称、单价
- B. 名称、单价
- C. 名称、单价、交易所
- D. 股票名称、单价、交易所

33. 有如下 SQL 语句:

```
SELECT DISTINCT 单价 FROM stock
WHERE 单价=(SELECT MIN (单价) FROM stock) INTO DBF stock_x
执行该语句后, stock-x 中的记录个数是 ( )。
```

- A. 1
- B. 2
- C. 3
- D. 4

34. 求每个交易所的平均单价的 SQL 语句是 ()。

- A. SELECT 交易所, AVG (单价) FROM stock GROUP BY 单价
- B. SELECT 交易所, AVG (单价) FROM stock ORDER BY 单价
- C. SELECT 交易所, AVG (单价) FROM stock ORDER BY 交易所
- D. SELECT 交易所, AVG (单价) FROM stock GROUP BY 交易所

35. 有如下 SQL 语句:

```
SELECT 交易所, AVG (单价) AS 均价 FROM stock
GROUP BY 交易所 INTO DBF temp
执行该语句以后, temp 表中第二条记录的均价字段的内容是 ( )。
```

- A. 7.48
- B. 9.99
- C. 11.73
- D. 15.20

第 36~41 题基于下面内容。

为管理岗位业务培训信息, 建立 3 个表:

S (S#,SN,SD,SA): S#,SN,SD,SA 分别代表学号、学员姓名、所属单位、学员年龄

C (C#,CN): C#,CN 分别代表课程编号、课程名称

SC (S#,C#,G): S#,C#,G 分别代表学号、所选修的课程编号、学习成绩

36. 下述 SQL 语句的作用是 ()。

```
SELECT SN,SD FROM S
WHERE [S#] IN(
    SELECT [S#] FROM C,SC
    WHERE C.[C#]=SC.[C#]
    AND CN='税收基础')
```

- A. 查询选修课程名称为“税收基础”的学员学号和姓名
- B. 删除选修课程名称为“税收基础”的学员学号和姓名
- C. 查询选修课程名称为“税收基础”的学员学号
- D. 查询选修课程名称为“税收基础”的学员姓名

37. 查询选修课程编号为“C2 “的学员姓名和所属单位的 SQL 语句是 ()。

- A. SELECT S.SN,S.SD FROM SC WHERE S.[S#]=SC.[S#] AND SC.[C#]=' C2'
 B. SELECT S.SN,S.SD FROM S,SC WHERE S.[S#]=SC.[S#] AND SC.[C#]=' C1'
 C. SELECT S.SN,S.SD FROM S,SC WHERE S.[S#]=SC.[S#] AND SC.[C#]=' C2'
 D. SELECT SN,SD FROM S,SC WHERE S.[S#]=SC.[S#] AND SC.[C#]=' C2'

38. 下述 SQL 语句的作用是 ()。

```
SELECT SN,SD FROM S
WHERE [S#] NOT IN(
  SELECT [S#] FROM SC
  WHERE [C#]='C5')
```

- A. 查询不选修课程编号为“C5”的学员所属单位 B. 查询不选修课程编号为“C5”的学员姓名和所属单位
 C. 查询选修课程编号为“C5”的学员姓名和所属单位 D. 查询不选修课程编号为“C5”的学员姓名

39. 使用下述 SQL 语句查询选修全部课程的学员姓名和所属单位, 请选出 () 中应该填写的内容。

```
SELECT SN,SD FROM S
WHERE [S#] IN(
  SELECT [S#] FROM SC
  RIGHT JOIN
  C ON SC.[C#]=C.[C#] ( ) [S#]
  HAVING COUNT(*)=COUNT([S#]))
```

- A. WHERE B. HAVING C. ORDER BY D. GROUP BY

40. 使用下述 SQL 语句查询选修课程超过 5 门的学员学号和所属单位, 请选出 () 中应该填写的内容。

```
SELECT SN,SD FROM S
WHERE [S#] IN(
  SELECT [S#] FROM SC
  GROUP BY [S#]
  ( ) COUNT(DISTINCT [C#])>5)A) WHERE
```

- A. HAVING B. WHERE C. ORDER BY D. GROUP BY

41. 查询选修了课程的学员人数的 SQL 语句是 ()。

- A. SELECT 学员人数=COUNT([S#]) FROM SC B. SELECT 学员人数=COUNT(DISTINCT [S#]) FROM SC
 C. SELECT 学员人数=COUNT(*) FROM SC D. SELECT 学员人数=COUNT(DISTINCT [S#])

42. 在 SQL 的 SELECT 语句中, “HAVING<条件表达式>” 用来筛选满足条件的 ()。

- A. 列 B. 行 C. 关系 D. 分组

43. 设有关系 SC (SNO,CNO,GRADE), 其中 SNO、CNO 分别表示学号、课程号 (两者均为字符型), GRADE 表示成绩 (数值型), 若要把学号为“S101”的同学, 选修课程号为“C11”, 成绩为 98 分的记录插到表 SC 中, 正确的语句是 ()。

- A. INSERT INTO SC(SNO,CNO,GRADE)value S(' S101' , ' C11' , ' 98')
 B. INSERT INTO SC(SNO,CNO,GRADE)value S(S101,C11,98)
 C. INSERT(' S101' , ' C11' , ' 98')INTO SC
 D. INSERT INTO SC value S(' S101' , ' C11' ,98)

44. 在 SQL 语句中, 与表达式“年龄 BETWEEN 12 AND 46”功能相同的表达式是 ()。

- A. 年龄>=12 OR <=46 B. 年龄>=12 AND <=46
 C. 年龄>=12 OR 年龄<=46 D. 年龄>=12 AND 年龄<=46

45. 在 Visual FoxPro 中, 如果要将学生表 S (学号, 姓名, 性别, 年龄) 中“年龄”属性删除, 正确的 SQL 命令是 ()。

- A. ALTER TABLE S DROP COLUMN 年龄 B. DELETE 年龄 FROM S

- C. ALTER TABLE S DELETE COLUMN 年龄 D. ALTER TABLE S DELETE 年龄
46. 设有学生表 S (学号, 姓名, 性别, 年龄), 查询所有年龄小于等于 18 岁的女同学, 并按年龄进行降序生成新的表 WS, 正确的 SQL 命令是 ()。
- A. SELECT * FROM S
WHERE 性别="女" AND 年龄<=18 ORDER BY 4 DESC INTO TABLE WS
- B. SELECT * FROM S
WHERE 性别="女" AND 年龄<=18 ORDER BY 年龄 INTO TABLE WS
- C. SELECT * FROM S
WHERE 性别="女" AND 年龄<=18 ORDER BY ' 年龄' DESC INTO TABLE WS
- D. SELECT * FROM S
WHERE 性别="女" OR 年龄<=18 ORDER BY ' 年龄' ASC INTO TABLE WS
47. 设有学生选课表 SC (学号, 课程号, 成绩), 用 SQL 检索同时选修课程号为 "C1" 和 "C5" 的学生学号的正确命令是 ()。
- A. SELECT 学号 FROM SC
WHERE 课程号=' C1' AND 课程号=' C5'
- B. SELECT 学号 FROM SC
WHERE 课程号=' C1' AND 课程号=(SELECT 课程号 FROM SC WHERE 课程号=' C5')
- C. SELECT 学号 FROM SC
WHERE 课程号=' C1' AND 学号=(SELECT 学号 FROM SC WHERE 课程号=' C5')
- D. SELECT 学号 FROM SC
WHERE 课程号=' C1' AND 学号 IN (SELECT 学号 FROM SC WHERE 课程号=' C5')
48. 设有学生表 S (学号, 姓名, 性别, 年龄)、课程表 C (课程号, 课程名, 学分) 和学生选课表 SC (学号, 课程号, 成绩), 检索学号、姓名和学生所选课程名和成绩, 正确的 SQL 命令是 ()。
- A. SELECT 学号, 姓名, 课程名, 成绩 FROM S, SC, C
WHERE S.学号=SC.学号 AND SC.学号=C.学号
- B. SELECT 学号, 姓名, 课程名, 成绩
FROM (S JOIN SC ON S.学号=SC.学号) JOIN C ON SC.课程号=C.课程号
- C. SELECT S.学号, 姓名, 课程名, 成绩
FROM S JOIN SC JOIN C ON S.学号=SC.学号 ON SC.课程号=C.课程号
- D. SELECT S.学号, 姓名, 课程名, 成绩
FROM S JOIN SC JOIN C ON SC.课程号=C.课程号 ON S.学号=SC.学号
49. 在 SQL SELECT 语句中, 为了将查询结果存储到临时表应该使用短语 ()。
- A. TO CURSOR B. INTO CURSOR C. INTO DBF D. TO DBF
50. 在 SQL 的 ALTER TABLE 语句中, 为了增加一个新的字段应该使用短语 ()。
- A. CREATE B. APPEND C. COLUMN D. ADD
- 第 51~56 题使用如下数据表:
- 学生.DBF: 学号 (C, 8), 姓名 (C, 6), 性别 (C, 2), 出生日期 (D)
- 选课.DBF: 学号 (C, 8), 课程号 (C, 3), 成绩 (N, 5, 1)
51. 查询所有 1982 年 3 月 20 日以后 (含) 出生、性别为男的学生, 正确的 SQL 语句是 ()。
- A. SELECT * FROM 学生 WHERE 出生日期>={ ^ 1982-03-20 } AND 性别="男"
- B. SELECT * FROM 学生 WHERE 出生日期<={ ^ 1982-03-20 } AND 性别="男"
- C. SELECT * FROM 学生 WHERE 出生日期>={ ^ 1982-03-20 } OR 性别="男"
- D. SELECT * FROM 学生 WHERE 出生日期<={ ^ 1982-03-20 } OR 性别="男"
52. 计算刘明同学选修的所有课程的平均成绩, 正确的 SQL 语句是 ()。

- A. SELECT AVG(成绩) FROM 选课 WHERE 姓名="刘明"
- B. SELECT AVG(成绩) FROM 学生, 选课 WHERE 姓名="刘明"
- C. SELECT AVG(成绩) FROM 学生, 选课 WHERE 学生.姓名="刘明"
- D. SELECT AVG(成绩) FROM 学生, 选课 WHERE 学生.学号=选课.学号 AND 姓名="刘明"
53. 假定学号的第3、4位为专业代码。要计算各专业学生选修课程号为“101”课程的平均成绩, 正确的SQL语句是 ()。
- A. SELECT 专业 AS SUBS(学号,3,2),平均分 AS AVG(成绩) FROM 选课 WHERE 课程号="101" GROUP BY 专业
- B. SELECT SUBS(学号,3,2) AS 专业,AVG(成绩) AS 平均分 FROM 选课 WHERE 课程号="101" GROUP BY 1
- C. SELECT SUBS(学号,3,2) AS 专业,AVG(成绩) AS 平均分 FROM 选课 WHERE 课程号="101" ORDER BY 专业
- D. SELECT 专业 AS SUBS(学号,3,2),平均分 AS AVG(成绩) FROM 选课 WHERE 课程号="101" ORDER BY 1
54. 查询选修课程号为“101”课程得分最高的同学, 正确的SQL语句是 ()。
- A. SELECT 学生.学号,姓名 FROM 学生,选课 WHERE 学生.学号=选课.学号 AND 课程号="101" AND 成绩 >=ALL(SELECT 成绩 FROM 选课)
- B. SELECT 学生.学号,姓名 FROM 学生,选课 WHERE 学生.学号=选课.学号 AND 成绩 >=ALL(SELECT 成绩 FROM 选课 WHERE 课程号="101")
- C. SELECT 学生.学号, 姓名 FROM 学生,选课 WHERE 学生.学号=选课.学号 AND 成绩 >=ANY(SELECT 成绩 FROM 选课 WHERE 课程号="101")
- D. SELECT 学生.学号,姓名 FROM 学生,选课 WHERE 学生.学号=选课.学号 AND 课程号="101" AND 成绩 >=ALL(SELECT 成绩 FROM 选课 WHERE 课程号="101")
55. 插入一条记录到“选课”表中, 学号、课程号和成绩分别是“02080111”、“103”和“80”, 正确的SQL语句是 ()。
- A. INSERT INTO 选课 VALUES("02080111","103",80)
- B. INSERT VALUES("02080111","103",80)TO 选课(学号,课程号,成绩)
- C. INSERT VALUES("02080111","103",80)INTO 选课(学号,课程号,成绩)
- D. INSERT INTO 选课(学号,课程号,成绩) FORM VALUES("02080111","103",80)
56. 将学号为“02080110”、课程号为“102”的选课记录的成绩改为92, 正确的SQL语句是 ()。
- A. UPDATE 选课 SET 成绩 WITH 92 WHERE 学号="02080110" AND 课程号="102"
- B. UPDATE 选课 SET 成绩=92 WHERE 学号="02080110" AND 课程号="102"
- C. UPDATE FROM 选课 SET 成绩 WITH 92 WHERE 学号="02080110" AND 课程号="102"
- D. UPDATE FROM 选课 SET 成绩=92 WHERE 学号="02080110" AND 课程号="102"
57. SQL 语句中删除视图的命令是 ()。
- A. DROP TABLE B. DROP VIEW C. ERASE TABLE D. ERASE VIEW
58. 设有订单表 order (其中包括字段: 订单号,客户号,职员号,签订日期,金额), 查询2007年所签订单的信息, 并按金额降序排序, 正确的SQL命令是 ()。
- A. SELECT * FROM order WHERE YEAR(签订日期)=2007 ORDER BY 金额 DESC
- B. SELECT * FROM order WHILE YEAR(签订日期)=2007 ORDER BY 金额 ASC
- C. SELECT * FROM order WHERE YEAR(签订日期)=2007 ORDER BY 金额 ASC
- D. SELECT * FROM order WHILE YEAR(签订日期)=2007 ORDER BY 金额 DESC
59. 设有订单表 order (其中包括字段: 订单号,客户号,客户号,职员号,签订日期,金额), 删除2002年1月1日以前签订的订单记录, 正确的SQL命令是 ()。
- A. DELETE TABLE order WHERE 签订日期<{^2002-1-1}
- B. DELETE TABLE order WHILE 签订日期>{^2002-1-1}
- C. DELETE FROM order WHERE 签订日期<{^2002-1-1}
- D. DELETE FROM order WHILE 签订日期>{^2002-1-1}

下表是用 list 命令显示的“运动员”表的内容和结构，60~62 题使用该表。

记录号	运动员号	投中 2 分球	投中 3 分球	罚球
1	1	3	4	5
2	2	2	1	3
3	3	0	0	0
4	4	5	6	7

60. 为“运动员”表增加一个字段“得分”的 SQL 语句是 ()。

- A. CHANGE TABLE 运动员 ADD 得分 I B. ALTER DATA 运动员 ADD 得分 I
C. ALTER TABLE 运动员 ADD 得分 I D. CHANGE TABLE 运动员 INSERT 得分 I

61. 计算每名运动员的“得分”(33 题增加的字段)的正确 SQL 语句是 ()。

- A. UPDATE 运动员 FIELD 得分=2*投中 2 分球+3*投中 3 分球+罚球
B. UPDATE 运动员 FIELD 得分 WITH 2*投中 2 分球+3*投中 3 分球+罚球
C. UPDATE 运动员 SET 得分 WITH 2*投中 2 分球+3*投中 3 分球+罚球
D. UPDATE 运动员 SET 得分=2*投中 2 分球+3*投中 3 分球+罚球

62. 检索“投中 3 分球”小于等于 5 个的运动员中“得分”最高的运动员的“得分”，正确的 SQL 语句是 ()。

- A. SELECT MAX(得分) 得分 FROM 运动员 WHERE 投中 3 分球<=5
B. SELECT MAX(得分) 得分 FROM 运动员 WHEN 投中 3 分球<=5
C. SELECT 得分=MAX(得分) FROM 运动员 WHERE 投中 3 分球<=5
D. SELECT 得分=MAX(得分) FROM 运动员 WHEN 投中 3 分球<=5

63. 在 SQL SELECT 查询中，为了使查询结果排序应该使用短语 ()。

- A. ASC B. DESC C. GROUP BY D. ORDER BY

64. 在 SQL SELECT 语句中与 INTO TABLE 等价的短语是 ()。

- A. INTO DBF B. TO TABLE C. INTO FORM D. INTO FILE

65~68 使用如下关系：

客户 (客户号，名称，联系人，邮政编码，电话号码)

产品 (产品号，名称，规格说明，单价)

订购单 (订单号，客户号，订购日期)

订购单明细 (订单号，序号，产品号，数量)

65. 查询单价在 600 元以上的主机板和硬盘的正确命令是 ()。

- A. SELECT * FROM 产品 WHERE 单价>600 AND (名称=' 主机板' AND 名称=' 硬盘')
B. SELECT * FROM 产品 WHERE 单价>600 AND (名称=' 主机板' OR 名称=' 硬盘')
C. SELECT * FROM 产品 FOR 单价>600 AND (名称=' 主机板' AND 名称=' 硬盘')
D. SELECT * FROM 产品 FOR 单价>600 AND (名称=' 主机板' OR 名称=' 硬盘')

66. 查询客户名称中有“网络”二字的客户信息的正确命令是 ()。

- A. SELECT * FROM 客户 FOR 名称 LIKE " %网络% "
B. SELECT * FROM 客户 FOR 名称= " %网络% "
C. SELECT * FROM 客户 WHERE 名称= " %网络% "
D. SELECT * FROM 客户 WHERE 名称 LIKE " %网络% "

67. 查询订购单的数量和所有订购单平均金额的正确命令是 ()。

- A. SELECT COUNT(DISTINCT 订单号),AVG(数量*单价)
FROM 产品 JOIN 订购单明细 ON 产品.产品号=订购单明细.产品号
B. SELECT COUNT(订单号),AVG(数量*单价)
FROM 产品 JOIN 订购单明细 ON 产品.产品号=订购单明细.产品号
C. SELECT COUNT(DISTINCT 订单号),AVG(数量*单价)

FROM 产品, 订购单名细 ON 产品.产品号=订购单名细.产品号

D. SELECT COUNT(订单号), AVG(数量*单价)

FROM 产品,订购单名细 ON 产品.产品号=订购单名细.产品号

68. 假设客户表中有客户号(关键字) C1~C10 共 10 条客户记录, 订购单表有订单号(关键字)OR1—OR8 共 8 条订购单记录, 并且订购单表参照客户表。如下命令可以正确执行的是 ()。

A. INSERT INTO 订购单 VALUES(' OR5' , ' C5' ,{'^2008/10/10})

B. INSERT INTO 订购单 VALUES(' OR5' , ' C11' ,{'^2008/10/10})

C. INSERT INTO 订购单 VALUES(' OR9' , ' C11' ,{'^2008/10/10})

D. INSERT INTO 订购单 VALUES(' OR9' , ' C5' ,{'^2008/10/10})

69. SQL 语言的查询语句是 ()。

A. INSERT

B. UPDATE

C. DELETE

D. SELECT

70. 在 Visual FoxPro 中, 下列关于 SQL 表定义语句 (CREATE TABLE) 的说法中错误的是 ()。

A. 可以定义一个新的基本表结构

B. 可以定义表中的主关键字

C. 可以定义表的域完整性、字段有效性规则等

D. 对自由表, 同样可以实现其完整性、有效性规则等信息的设置

71. 在 Visual FoxPro 中, 假设教师表 T (教师号, 姓名, 性别, 职称, 研究生导师) 中, 性别是 C 型字段, 研究生导师是 L 型字段。若要查询 “是研究生导师的女老师” 信息, 那么 SQL 语句 “SELECT * FROM T WHERE <逻辑表达式>” 中的<逻辑表达式>应是 ()。

A. 研究生导师 AND 性别= " 女 "

B. 研究生导师 OR 性别= " 女 "

C. 性别= " 女 " AND 研究生导师=.F.

D. 研究生导师=T. OR 性别=女

第 72~76 题基于学生表 S 和学生选课表 SC 两个数据库表, 它们的结构如下:

S (学号, 姓名, 性别, 年龄), 其中学号、姓名和性别为 C 型字段, 年龄为 N 型字段。

SC (学号, 课程号, 成绩), 其中学号和课程号为 C 型字段, 成绩为 N 型字段 (初始为空值)。

8.5.2 填空题

1. 为 “学生” 表增加一个 “平均成绩” 字段的正确命令是:

ALTER TABLE 学生 ADD _____ 平均成绩 N(5,2)。

第 2~4 题基于下面内容。

有如下 3 个数据库表:

金牌榜.DBF: 国家代码 C(3), 金牌数 I, 银牌数 I, 铜牌数 I

获奖牌情况.DBF: 国家代码 C(3), 运动员名称 C(20), 项目名称 C(30), 名次 I

国家.DBF: 国家代码 C(3), 国家名称 C(20)

“金牌榜” 表中一个国家一条记录; “获奖牌情况” 表中每个项目中的各个名次都有一条记录, 名次只取前 3 名。例如:

国家代码	运动员名称	项目名称	名次
001	刘翔	男子 110 米栏	1
001	李小鹏	男子双杠	3
002	菲尔普斯	游泳男子 200 米自由泳	3
002	菲尔普斯	游泳男子 400 米混合个人泳	1
001	郭晶晶	女子 3 米板跳板	1
001	李婷/孙甜甜	网球女子双打	1

2. 为 “金牌榜” 表增加一个字段 “奖牌总数”, 同时为该字段设置有效性规则: 奖牌总数>=0, 应使用 SQL 语句:

ALTER TABLE 金牌榜 _____ 奖牌总数 I _____ 奖牌总数>=0

3. 使用 “获奖牌情况” 和 “国家” 两个表查询 “中国” 所获金牌 (名次为 1) 的数量, 应使用 SQL 语句:

SELECT COUNT(*) FROM 国家 INDEX JOIN 获奖牌情况

_____国家.国家代码=获奖牌情况.国家代码

WHERE 国家.国家名称="中国"AND 名次=1

4. 将金牌榜.DBF 中新增加的字段奖牌总数设置为金牌数、银牌数、铜牌数 3 项的和，应使用 SQL 语句：

_____金牌榜_____奖牌总数=金牌数+银牌数+铜牌数

5. 在 Visual FoxPro 中，使用 SQL 的 CREATE TABLE 语句建立数据库表时，使用_____子句说明有效性规则（域完整性规则或字段取值范围）。

6. 在 SQL 的 SELECT 查询中使用_____子句消除查询结果中的重复记录。

7. SQL 的操作语句包括 INSERT、UPDATE 和_____PROJECT。

8. 在 SQL SELECT 语句中，要对查询结果的记录个数计数应该使用_____函数。

9. 在 ALTER TABLE 命令中用于删除字段的短语是_____。

第 10~12 题基于下面两个表。

“教师”表：

职工号	姓名	职称	年龄	工资	系号
11020001	肖天海	副教授	35	2000.00	01
11020002	王岩盐	教授	40	3000.00	02
11020003	刘星魂	讲师	25	1500.00	01
11020004	张月新	讲师	30	1500.00	03
11020005	李明玉	教授	34	2000.00	01
11020006	孙民山	教授	47	2100.00	02
11020007	钱无名	教授	49	2200.00	03

“学院”表：

系号	系名
01	英语
02	会计
03	工商管理

10. 使用 SQL 语句将一条新的记录插入学院表：_____。

11. 使用 SQL 语句求“工商管理”系的所有职工的工资总和。

SELECT _____(工资) FROM 教师 WHERE 系号 IN (SELECT _____FROM 学院 WHERE 系名="工商管理")

12. 使用 SQL 语句完成将所有教授的工资提高 5%：_____。

13. 当 SUM、COUNT 等函数用在 SQL SELECT 语句中作为条件时，必须使用的短语是_____。

第 14~17 题基于下面内容。

已知关系模式：

S (SNO,SNAME)，学生关系：SNO 为学号，SNAME 为姓名

C (CNO,CNAME,CTEACHER)，课程关系：CNO 为课程号，CNAME 为课程名，CTEACHER 为任课教师

SC(SNO,CNO,SCGRADE)，选课关系：SCGRADE 为成绩

14. 使用 SQL 语句查询没有选修过“李明”老师讲授课程的所有学生姓名。

SELECT SNAME FROM S

WHERE _____(

SELECT * FROM SC,C

WHERE SC.CNO=C.CNO AND CNAME='李明' AND SC.SNO=S.SNO)

15. 使用 SQL 语句查询有两门以上（含两门）不及格课程的学生姓名及其平均成绩。

SELECT S.SNO,S.SNAME,AVG_SCGRADE=AVG(_____) FROM S,SC,(SELECT SNO FROM SC
WHERE SCGRADE<60 GROUP BY SNO HAVING COUNT(DISTINCT CNO)>=2)A

WHERE S.SNO=A.SNO AND SC.SNO=A.SNO GROUP BY S.SNO,S.SNAME

16. 使用 SQL 语句查询既学过“1”号课程，又学过“2”号课程的所有学生姓名。

SELECT S.SNO,S.SNAME FROM S,(SELECT SC.SNO FROM SC,C WHERE SC.CNO=C.CNO AND C.CNAME
IN(_____) GROUP BY SNO HAVING COUNT(DISTINCT CNO)=2) SC WHERE S.SNO=SC.SNO

17. 使用 SQL 语句查询“1”号课成绩比“2”号课该门课成绩高的所有学生的学号。

SELECT S.SNO,S.SNAME FROM S,(SELECT SC1.SNO FROM SC SC1,C C1,SC SC2,C C2 WHERE
SC1.CNO=C1.CNO AND C1.NAME='1' AND SC2.CNO=C2.CNO AND C2.NAME='2' AND _____)SC WHERE
S.SNO=SC.SNO

18. “歌手”表中有“歌手号”、“姓名”和“最后得分”3个字段，“最后得分”越高名次越靠前，查询前10名歌手的 SQL 语句是：SELECT * _____ FROM 歌手 ORDER BY 最后得分 _____。

19. 已有“歌手”表，将该表中的“歌手号”字段定义为候选索引，索引名是 temp，正确的 SQL 语句是：_____ TABLE
歌手 ADD UNIQUE 歌手号 TAG temp。

20. 如下命令查询雇员表中“部门号”字段为空值的记录：SELECT * FROM 雇员 WHERE 部门号 _____。

21. 在 SQL 的 SELECT 查询中，HAVING 子句不可以单独使用，总是跟在 _____ 子句之后一起使用。

22. 在 SQL 中，插入、删除、更新命令依次是 INSERT、DELETE 和 _____。

23. 在 SQL 的 WHERE 子句的条件表达式中，字符串匹配（模糊查询）的运算符是 _____。

24. 使用 SQL 的 CREATE TABLE 语句定义表结构时，用 _____ 短语说明关键字(主索引)。

25. 在 SQL 语句中要查询表 s 在 AGE 字段上取空值的记录，正确的 SQL 语句为：SELECT * FROM s WHERE _____。

第9章

查询与视图

在 Visual FoxPro 中处理的基本数据对象是表，同时为了提高对数据的处理效率，引入了查询和视图。查询和视图是 Visual FoxPro 中操作和检索数据的基本工具和手段，都可以从数据表中快速、方便地获取所需要的数据。视图和查询是一对既相似又不同的两个概念。

本章主要介绍查询和视图的概念、建立和使用。

9.1 查询

查询可以根据用户的需要从一个或多个数据表或视图中提取一组相关的数据子集。使用“查询向导”或“查询设计器”可以创建查询。

9.1.1 查询的概念

所谓查询，就是一个预先定义好的，可以从指定的表或视图中查找满足条件的记录，然后按照希望得到的输出类型定向输出结果的 SQL SELECT 语句。

查询以扩展名为.qpr 的文件保存在磁盘中，可以根据需要直接或反复使用，从而可以提高数据使用的效率。运行查询文件，即可浏览查询的结果。

9.1.2 查询设计器

在 Visual FoxPro 中建立查询有以下几种方法：

(1) 在主菜单“文件”菜单中选择“新建”命令，弹出“新建”对话框，如图 9-1 所示，选择“查询”选项，单击“新建文件”按钮，打开“新建查询”对话框，如图 9-2 所示。单击“新建查询”按钮，弹出“查询设计器”对话框。

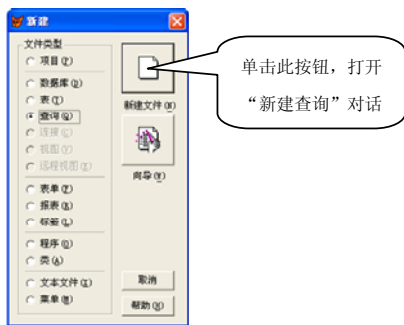


图 9-1 “新建”对话框

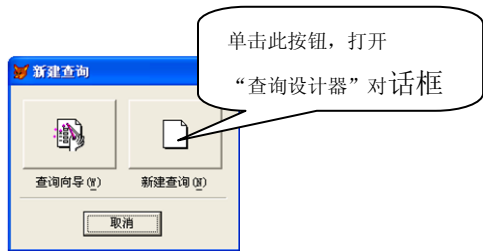


图 9-2 “新建查询”对话框

(2) 在项目管理器中的“数据”选项卡中选择“查询”选项，单击“新建”按钮，打开“新建查询”对话框，如图 9-3 所示。单击“新建查询”按钮，弹出“查询设计器”对话框。

(3) 利用 CREATE QUERY 命令打开“查询设计器”对话框建立查询。

(4) 利用 SQL SELECT 语句直接编辑.qpr 文件建立查询。

在打开“查询设计器”对话框时，首先进入“添加表或视图”对话框，如图 9-4 所示。此时选择所需要的数据库，以及数据库中的数据库表或视图，然后单击“添加”按钮，则将所选择的表或视图加入到“查询设计器”中。单击“其他”按钮，可以选择“自由表”加入到“查询设计器”中。

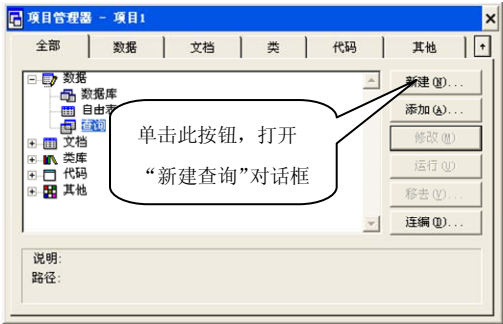


图 9-3 在项目管理器中打开“新建查询”对话框

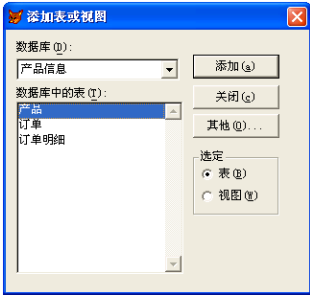


图 9-4 “添加表或视图”对话框

单击“关闭”按钮，进入“查询设计器”对话框中，如图 9-5 所示。

当建立一个基于多个数据表或视图的查询时，必须首先建立这些表或视图之间的联系。

在“查询设计器”对话框中，有 6 个选项卡：“字段”、“联接”、“筛选”、“排序依据”、“分组依据”、“杂项”。这些选项卡与 SQL SELECT 语句的各个短语相互对应，其对应关系如表 9-1 所示。



图 9-5 “查询设计器”对话框

表 9-1 “查询设计器”选项卡与 SQL SELECT 语句的对应关系

选项卡	SQL 语句	说明
字段	SELECT	用来指定包含在查询结果中的字段，在“函数和表达式”文本框中还可以输入或编辑可计算的表达式
联接	JOIN ON	用来确定表或视图之间的联接条件
筛选	WHERE	用来指定查询条件
排序依据	ORDER BY	用来确定查询结果中记录的排序字段和排序方式
分组依据	GROUP BY、HAVING	用来进行分组
杂项	DISTINCT、TOP	用来指定是否要重复记录，以及列在前面的记录

9.1.3 建立查询

1. 建立简单查询

利用“查询设计器”建立查询的步骤如下：

- (1) 在“字段”选项卡中选择需要的字段，作为查询输出的字段，如图 9-6 所示。
- (2) 在“联接”选项卡中选择联接类型，如图 9-7 所示。

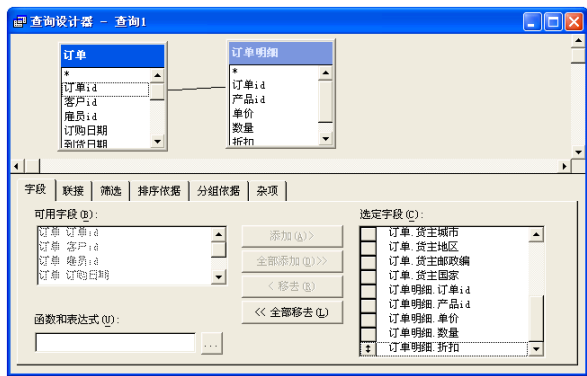


图 9-6 选择查询输出字段

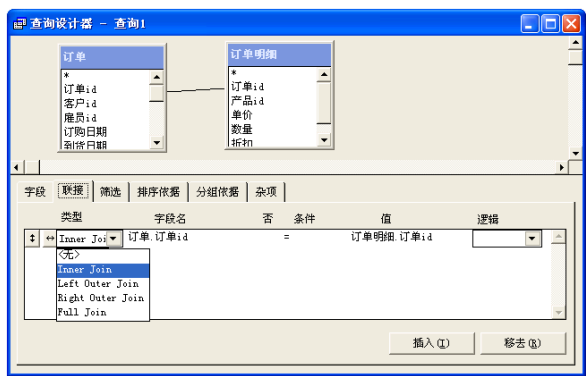


图 9-7 选择联接类型

- (3) 在“筛选”选项卡中设置筛选条件，如图 9-8 所示。
- (4) 在“排序依据”选项卡中选择排序条件和排序方式，如图 9-9 所示。

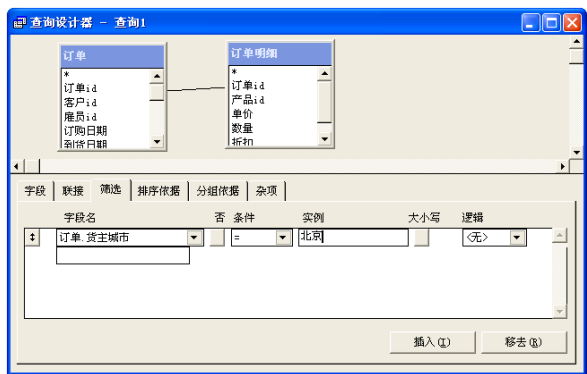


图 9-8 设置筛选条件



图 9-9 设置排序依据

- (5) 在“分组依据”选项卡中选择分组字段，如图 9-10 所示。
- (6) 在“杂项”选项卡中设置是否无重复记录和设置列在前面的记录，如图 9-11 所示。

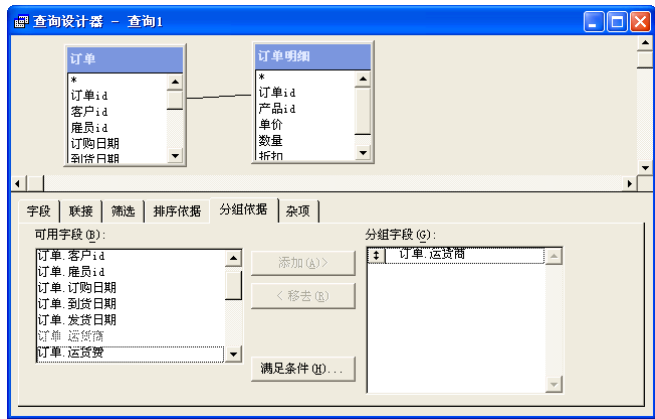


图 9-10 设置分组依据



图 9-11 设置杂项

（7）在主菜单“查询”菜单项中选择“查询去向”命令，弹出“查询去向”对话框，设置查询去向，如图 9-12 所示。

（8）在主菜单“文件”菜单项中选择“保存”命令，把设计好的查询保存为.qpr 查询文件。

2. 查询中常用功能说明

在查询建立过程中，需要设置“查询设计器”中选项卡的内容，下面介绍这些内容的功能。

1) “查询设计器”中“联接”选项卡的联接类型（见表 9-2）

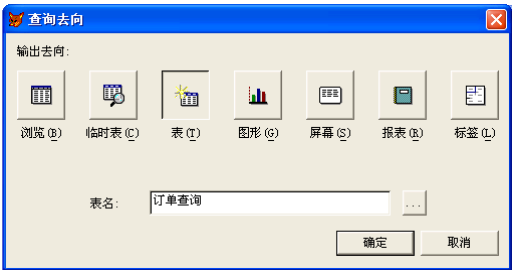


图 9-12 “查询去向”对话框

表 9-2 “联接”选项卡的联接类型

联接类型	说明
INNER JOIN 内部联接	指定只有满足联接条件的记录包含在结果中，是默认的联接类型，也是最常用的联接类型
RIGHT OUTER JOIN 右联接	指定满足联接条件的记录，以及不满足联接条件右侧表中的记录也包含在结果中
LEFT OUTER JOIN 左联接	指定满足联接条件的记录，以及不满足联接条件左侧表中的记录也包含在结果中
FULL JOIN 完全联接	指定满足和不满足联接条件的记录都包含在结果中

2) “查询设计器”中“筛选”选项卡的筛选条件

筛选是指选择表中符合条件的一部分记录，而非全部记录。筛选条件是指实现筛选功能的逻辑表达式，包括字段、条件和实例。

设置筛选条件所包含的内容及其说明如表 9-3 所示。

表 9-3 筛选条件和说明

筛选条件	说明
=	字段值与实例值相等
LIKE	字段值与实例值之间执行不完全匹配，专门针对字符型字段
==	字段值与实例值严格相等，专门针对字符型字段
>	字段值大于实例值
>=	字段值大于或等于实例值
<	字段值小于实例值
<=	字段值小于或等于实例值
IS NULL	指定字段必须包含 NULL 值
BETWEEN	字段值在实例值的最大值和最小值之间
IN	字段值必须是实例值中的一个，实例值各值之间用逗号分隔

3) 查询去向

在 Visual FoxPro 中，用户可以根据需要将查询输出定位到各种输出方式。选择主菜单“查询”菜单中的“查询去向”命令，或者在“查询设计器”工具栏中单击“查询去向”按钮，则可打开“查询去向”对话框，可以在其中选择所需要的查询去向，如图 9-13 所示。



图 9-13 “查询去向”对话框

查询去向的具体含义如表 9-4 所示。

表 9-4 查询去向和含义

查询去向	含义
浏览	默认设置，在浏览窗口中显示查询结果
临时表	将查询结果存储在一个临时只读表中
表	将查询结果存储在一个数据表中
图形	使查询结果可用于 Microsoft Graph 应用程序
屏幕	将查询结果在 Visual FoxPro 主窗口或当前活动输出窗口中显示
报表	将查询结果输送到一个报表文件中
标签	将查询结果输送到一个标签文件中

9.1.4 运行查询

建立查询并把设计好的查询保存为.qpr 查询文件，需要通过运行查询才能看到查询的结果。

在 Visual FoxPro 中运行查询有以下 3 种方法：

- (1) 在“项目管理器”对话框的“查询”选项中，选择需要运行的查询文件，单击“运行”按钮，如图 9-14 所示。
- (2) 在“查询设计器”对话框中，单击鼠标右键，在快捷菜单中选择“运行查询”命令，如图 9-15 所示。

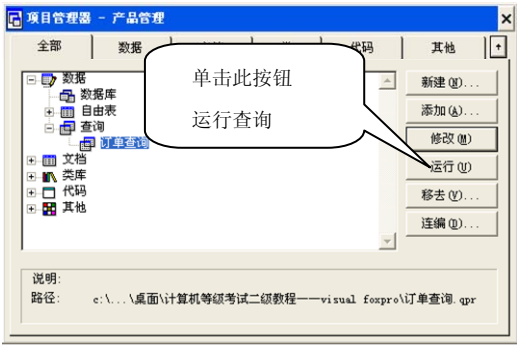


图 9-14 在“项目管理器”中运行查询



图 9-15 在“查询设计器”中运行查询

- (3) 在命令窗口中输入“运行查询”命令，如图 9-16 所示。

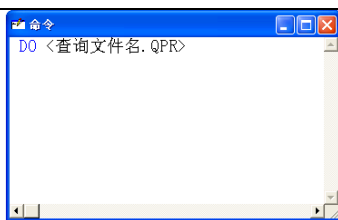


图 9-16 在命令窗口输入“运行查询”命令

9.2 视图

视图与查询一样，也是根据用户的需要从一个或多个数据表或视图提取一组数据子集，但是二者也存在区别。

9.2.1 视图的概念

所谓视图，是一个定制的虚拟表或逻辑表，它是从一个已经存在或多个相关联的数据表根据需要组织起来的查看数据的一个窗口，通过它可以查看表中有用的内容。之所以称之为虚拟表，是因为只能在数据库中看到而在磁盘上看不到它。因此，视图是数据库中的一个特有功能，只有在包含视图的数据库打开时，才能使用视图。

虽然视图是虚拟表，但是完全可以像数据表一样方便地使用。对于数据表的操作命令大多数也同样适用于视图。

与查询不同的是，使用视图从表中提取一组记录，如果改变了记录的值，变更的结果会被送回到基本表中；而查询却是“只读”的，不能更新数据。

9.2.2 建立本地视图

按照建立视图的数据源的不同，可以分为本地视图和远程视图。如果视图中有取自远程数据源（如网络服务器）的数据，则称为远程视图，否则为本地视图。

建立本地视图有以下 4 种方法：

(1) 在主菜单“文件”菜单中选择“新建”命令，弹出“新建”对话框，选择“视图”选项，单击“新建文件”按钮，打开“视图设计器”对话框，如图 9-17 所示。

(2) 在“项目管理器”对话框中的“数据”选项卡中选择“数据库”选项，将要建立视图的数据库分支展开，并选择“本地视图”选项，单击“新建文件”按钮，打开“视图设计器”对话框，如图 9-18 所示。

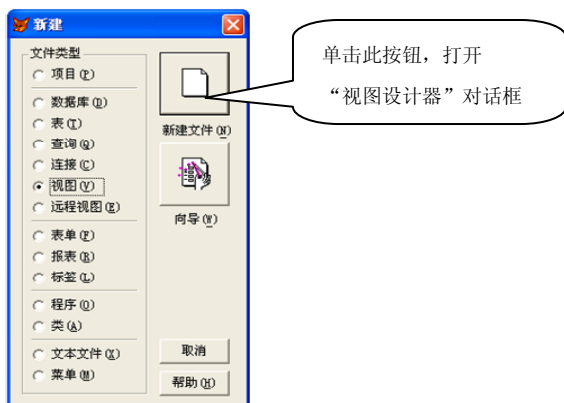


图 9-17 “新建”对话框

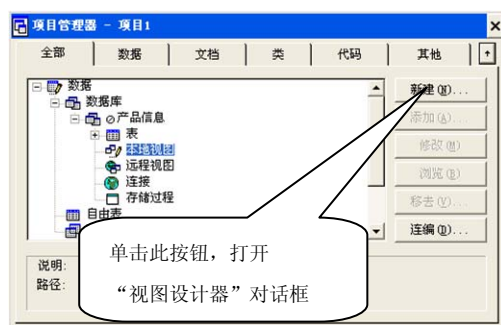


图 9-18 在“项目管理器”中打开“新建查询”对话框

- (3) 利用 CREATE VIEW 命令打开“视图设计器”创建视图。
 - (4) 利用 SQL 命令 CREATE VIEW ... AS ... 建立视图。
- 在打开“视图设计器”对话框时，首先进入“添加表或视图”对话框，如图 9-19 所示。此时选择所需要的数据库，以及数据库中的数据库表或视图，然后单击“添加”按钮，则将所选择的表或视图加入到“视图设计器”中。单击“其他”按钮，可以选择“自由表”加入到“视图设计器”中。
- 单击“关闭”按钮，进入“视图设计器”对话框中，如图 9-20 所示。

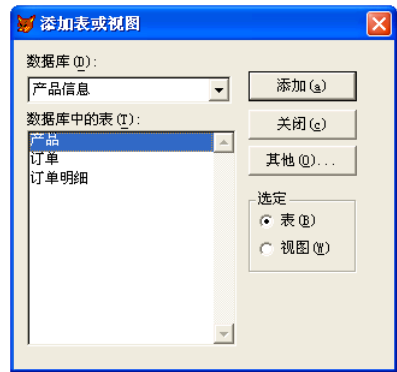


图 9-19 “添加表或视图”对话框



图 9-20 “视图设计器”对话框

- 对比“视图设计器”对话框和“查询设计器”对话框，可以看出二者非常接近，使用方式也几乎完全一样。但是二者还是有区别的，主要区别在于：
- 在“视图设计器”对话框中没有“查询去向”选项。
 - 在“视图设计器”对话框中比“查询设计器”对话框多了一个“更新条件”选项卡，因为视图可以进行更新，即使用视图从表中提取一组记录，如果改变了记录的值，变更的结果将会被送回到基本表中。

9.2.3 远程视图与连接

创建远程视图首先必须建立连接远程数据库的“连接”，“连接”是 Visual FoxPro 数据库中的一种对象。

1. 定义数据源
- 定义数据源指的是定义 ODBC（Open Database Connect，开放数据库连接）数据源，这需要首先安装 ODBC 驱动程序，利用 ODBC 驱动程序既可以定义本地数据库数据源，也可以定义远程数据库数据源。
2. 建立连接
- 所谓连接，指的是根据数据源创建并保存在数据库中的一种对象，它的作用主要是在创建远程视图时可以按其名称进行引用，同时如果设置了命名连接的属性，还可以对 Visual FoxPro 与远程数据源的通信进行优化。
- 建立连接有如下 3 种方法：
- (1) 选择主菜单“文件”菜单中的“新建”命令，弹出“新建”对话框，选择“连接”选项，单击“新建文件”按钮，打开“连接设计器”对话框，如图 9-21 所示。
 - (2) 在“项目管理器”对话框中的“数据”选项卡中选择“数据库”选项，将要建立连接的数据库分支展开，并选择“连接”选项，单击“新建文件”按钮，打开“连接设计器”对话框，如图 9-22 所示。

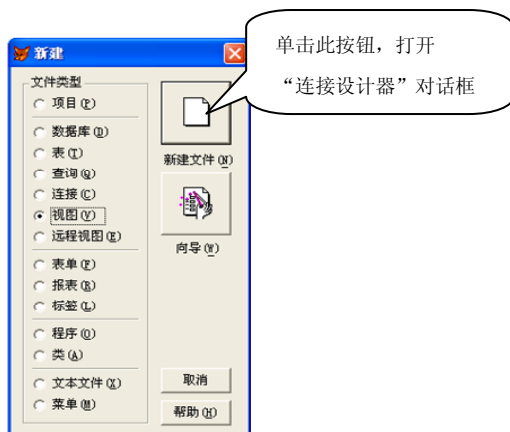


图 9-21 “新建”对话框

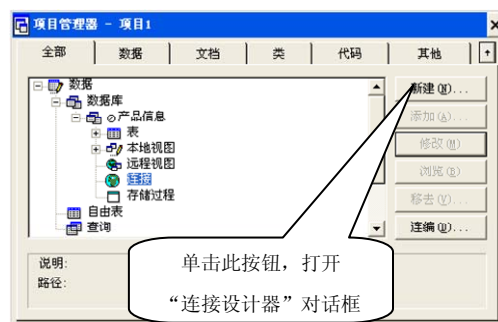


图 9-22 在“项目管理器”中打开“连接设计器”对话框

(3) 利用 CREATE CONNECTION 命令打开“连接设计器”建立连接。

“连接设计器”对话框如图 9-23 所示，选择需要的数据源后，单击“验证连接”按钮进行验证，检验是否能够成功与远程数据库连接，若成功连接，则单击工具栏中“保存”按钮将该连接保存。

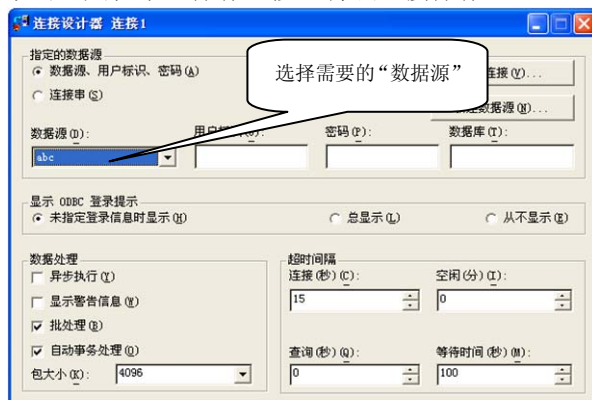


图 9-23 “连接设计器”对话框

9.2.4 视图与数据更新

视图是根据基本表派生而来的，在打开数据库和使用视图时，视图和基本表分别在两个工作区中打开。

系统默认对视图的更新不反映在基本表中，但是可以通过在“视图设计器”对话框中设置更新条件，从而使本地视图或远程视图在更新数据时，也能同时更新基本表。

设置数据更新的步骤如下：

(1) 指定可更新的表。默认该数据库中所有的表都是可以更新的。如果需要指定更新某个表，则从下拉列表框中进行选择，如图 9-24 所示。

(2) 指定可更新的字段。在“字段名”列表框选项左侧前方有两个符号：“钥匙”和“铅笔”。其中，“钥匙”表示关键字段，一般关键字段是不允许更新的；“铅笔”表示可更新字段，默认所有的非关键字段都是可以更新的。单击相应的列，可以改变当前状态，如图 9-25 所示。

(3) 更新的方式。选择“使用更新”选项组，其中，SQL DELETE 然后 INSERT 表示先删除记录，再插入记录；SQL UPDATE 表示使用更新命令，如图 9-26 所示。

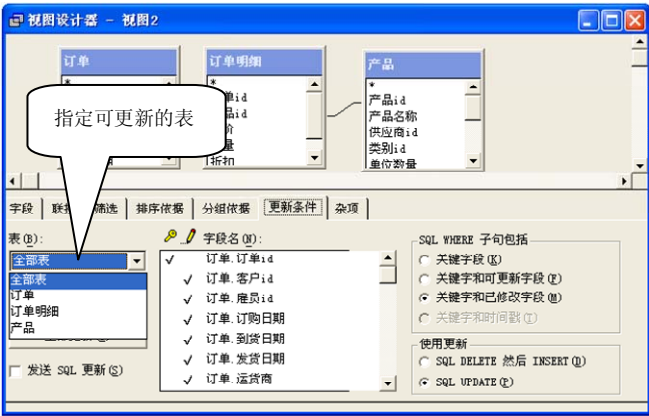


图 9-24 指定可更新的表

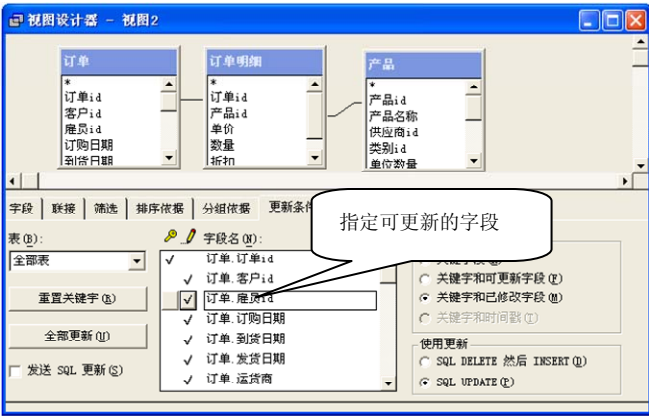


图 9-25 指定可更新的字段

(4)控制更新的合法性。使用远程视图更新数据时，由于是在多用户的工作环境下，因此在允许更新之前，Visual FoxPro 会检查视图所操作的数据是否被其他用户修改过，如果被提取到远程视图的数据已经被修改，则不允许更新操作。

在“视图设计器”对话框中，是通过“SQL WHERE 子句包括”选项组来决定哪些字段包含在 UPDATE 或 DELETE 语句的 WHERE 子句中。

“SQL WHERE 子句包括”选项各选项组的含义如下：

- 关键字段：当基本表中的关键字段被改变时，更新失败。
- 关键字和可更新字段：当基本表中任何标记为可更新的字段被改变时，更新失败。
- 关键字和已修改字段：当在视图中改变的任意字段值在基本表中被改变时，更新失败。
- 关键字和时间戳：当远程表上记录的时间戳在首次检索之后被改变时，更新失败。

(5) 确定发送更新。如果需要向基本表发送更新，则选中“视图设计器”对话框中的“发送 SQL 更新”复选框。如图 9-27 所示。

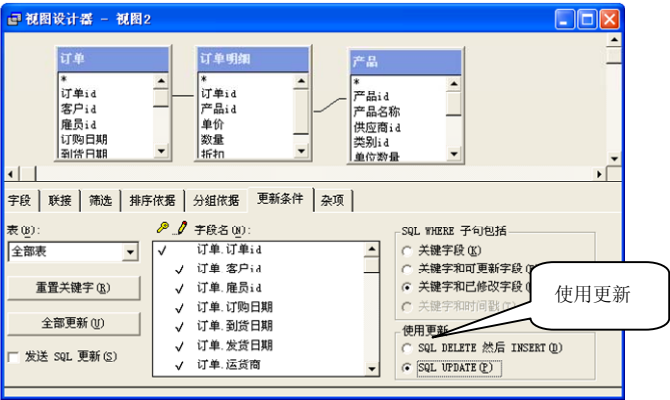


图 9-26 使用更新

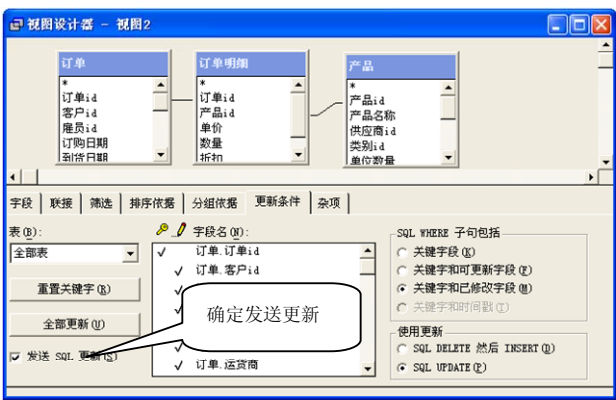


图 9-27 确定发送更新

9.2.5 使用视图

视图作为虚拟表，在打开数据库之后，可以像基本表一样进行操作，同时还可以在表单的数据控件文本框或表格中将其作为数据源，以及在创建报表时将其作为数据源。

9.3 本章习题

9.3.1 选择题

- 在 Visual FoxPro 中，以下关于视图描述中错误的是（ ）。
 - 通过视图可以对表进行查询
 - 通过视图可以对表进行更新
 - 视图是一个虚表
 - 视图就是一种查询
- 关于视图和查询，以下叙述正确的是（ ）。
 - 视图和查询都只能在数据库中建立
 - 视图和查询都不能在数据库中建立
 - 视图只能在数据库中建立
 - 查询只能在数据库中建立
- 建立一个视图 salary，该视图包括了系号和（该系的）平均工资两个字段，正确的 SQL 语句是（ ）。
 - CREATE VIEW salary AS 系号, AVG(工资) AS 平均工资 FROM 教师 GROUP BY 系号
 - CREATE VIEW salary AS SELECT 系号, AVG(工资) AS 平均工资 FROM 教师 GROUP BY 系名
 - CREATE VIEW salary SELECT 系号, AVG(工资) AS 平均工资 FROM 教师 GROUP BY 系号
 - CREATE VIEW salary AS SELECT 系号, AVG(工资) AS 平均工资 FROM 教师 GROUP BY 系号
- 删除视图 salary 的命令是（ ）。
 - DROP salary VIEW
 - DROP VIEW salary
 - DELETE salary VIEW
 - DELETE salary
- 有关查询设计器，正确的描述是（ ）。
 - “联接”选项卡与 SQL 语句的 GROUP BY 短语对应
 - “筛选”选项卡与 SQL 语句的 HAVING 短语对应
 - “排序依据”选项卡与 SQL 语句的 ORDER BY 短语对应
 - “分组依据”选项卡与 SQL 语句的 JOIN ON 短语对应
- 以下关于视图的描述正确的是（ ）。
 - 视图保存在项目文件中
 - 视图保存在数据库文件中
 - 视图保存在表文件中
 - 视图保存在视图文件中
- 在 Visual FoxPro 中，要运行查询文件 query1.qpr，可以使用命令（ ）。
 - DO query1
 - DO query1.qpr
 - DO QUERY query1
 - RUN query1
- 以下关于“查询”的描述正确的是（ ）。
 - 查询保存在项目文件中
 - 查询保存在数据库文件中
 - 查询保存在表文件中
 - 查询保存在查询文件中
- 在 Visual FoxPro 中，关于查询的正确叙述是（ ）。
 - 查询与数据库表相同，用来存储数据
 - 可以从数据库表、视图和自由表中查询数据
 - 查询中的数据是可以更新的
 - 查询是从一个或多个数据库表中导出来为用户定制的虚拟表
- 在查询设计器中，选择查询去向是“表”，则原有的 SQL SELECT 语句后面增加的短语是（ ）。
 - TO TABLE <表名.dbf>
 - INTO TABLE <表名.dbf>
 - INTO CURSOR <表名.dbf>
 - TO CURSOR <表名.dbf>
- 在查询设计器中没有，但在视图设计器中含有的选项卡是（ ）。
 - 筛选
 - 分组依据
 - 更新条件
 - 排序依据
- 视图设计器中包括的选项卡有（ ）。
 - 字段、筛选、排序依据、更新条件
 - 字段、条件、分组依据、更新条件
 - 条件、排序依据、分组依据、更新条件
 - 条件、筛选、杂项、更新条件

13. 在 Visual FoxPro 中, 关于视图的正确叙述是 ()。
- A. 视图与数据库表相同, 用来存储数据 B. 视图不能同数据库表进行联表操作
C. 视图中的数据不能进行更新 D. 视图是从一个或多个数据库表中导出来为用户定制的虚拟表
14. 在 Visual FoxPro 中, 关于建立视图的正确说法是 ()。
- A. 视图通过视图设计器建立
B. 视图通过 CREATE VIEW <视图名> AS <查询块> 命令建立
C. 视图通过 CREATE TABLE <视图名> AS <查询块> 命令建立
D. A 和 B 都对
15. 视图设计器是帮助用户创建 ()。
- A. 各种复杂的视图 B. 一般要求的视图
C. 特殊要求的表 D. 临时表
16. 在 Visual FoxPro 中, 以下关于查询的描述正确的是 ()。
- A. 不能用自由表建立查询 B. 只能使用自由表建立查询
C. 不能用数据库表建立查询 D. 可以用数据库表和自由表建立查询
17. 在视图设计器中有, 而在查询设计器中没有的选项卡是 ()。
- A. 排序依据 B. 更新条件 C. 分组依据 D. 杂项
18. 在使用查询设计器创建查询时, 为了指定在查询结果中是否包含重复记录 (对应于 DISTINCT), 应该使用的选项卡是 ()。
- A. 排序依据 B. 联接 C. 筛选 D. 杂项
19. 可以运行查询文件的命令是 ()。
- A. DO B. BROWSE C. DO QUERY D. CREATE QUERY
20. 在 Visual FoxPro 中, 以下叙述正确的是 ()。
- A. 利用视图可以修改数据 B. 利用查询可以修改数据
C. 查询和视图具有相同的作用 D. 视图可以定义输出去向
21. 在 Visual FoxPro 中, 关于查询的说法错误的是 ()。
- A. 查询的对象可以是数据库表, 也可以是已经存在的视图
B. 查询文件中的内容是一些用 SQL 命令定义的查询条件和规则
C. 执行查询文件与执行该文件包含的 SQL 命令的效果是一样的
D. 执行查询文件查询数据表中的数据时, 必须事先打开有关的数据库
22. 在 Visual FoxPro 的查询设计器中“筛选”选项卡对应的 SQL 语句是 ()。
- A. WHERE B. JOIN C. SET D. ORDER BY
23. 在 Visual FoxPro 中, 以下关于视图的描述正确的是 ()。
- A. 可以用自由表建立视图 B. 可以用查询建立视图
C. 可以用数据库表建立视图 D. 可以用数据库表和自由表建立视图
24. 查询设计器中包括的选项卡有 ()。
- A. 字段、筛选、排序依据 B. 字段、条件、分组依据
C. 条件、排序依据、分组依据 D. 条件、筛选、杂项
25. 删除视图 myview 的命令是 ()。
- A. DELETE myview VIEW B. DELETE myview
C. DROP myview VIEW D. DROP VIEW myview

9.3.2 填空题

1. 查询设计器的筛选选项卡用来指定查询的_____。
2. 使用当前数据库中的表建立的视图是本地视图, 使用当前数据库之外的数据源 (如 SQL Server 或其他 ODBC 数据

源)中的表建立的视图是_____视图。

3. 为了通过视图更新基本表中的数据, 需要在视图设计器界面的左下角选中_____复选框。
4. 查询设计器的“排序依据”选项卡对应 SQL SELECT 语句的_____短语。
5. 查询设计器的“连接”选项卡对应于 SQL SELECT 语句的_____短语, 用于编辑连接条件。
6. 查询设计器的“筛选”选项卡对应于 SQL SELECT 语句的_____短语。
7. 查询设计器的“分组”选项卡对应于 SQL SELECT 语句的_____短语和 HAVING 短语, 用于分组。
8. 查询设计器的结果是将 SQL SELECT 语句以_____扩展名的文件保存在磁盘文件中。
9. 在数据库中可以设计视图和查询, 其中_____不能独立存储为文件(存储在数据库中)。

第 10 章

表单设计及应用

表单是面向对象程序设计的主要工具，也是学习 Visual FoxPro 的重点和难点。表单中可以包含命令按钮、文本框、列表框、组合框等各种界面元素，是建立应用程序界面的最主要的工具之一。

本章主要介绍 Visual FoxPro 中表单的建立和管理、表单设计器及一些常用的表单控件。

10.1 面向对象的概念

在 Visual FoxPro 的表单设计中，处处体现着面向对象的思想和方法。面向对象（Object Oriented，OO）的程序设计是 20 世纪 80 年代出现的一种全新的系统开发和程序设计方法。Visual FoxPro 不仅可以应用结构化程序设计的思想和方法，而且全面支持面向对象的程序设计。

10.1.1 对象和类

对象和类是面向对象方法的两个最基本的概念。

1. 对象（Object）

对象是面向对象方法中最基本的概念，是构成面向对象程序设计的基本单位和运行实体。所谓对象，是指任何有意义的、与需要解决的问题有关系的任何事物。由此可见，客观世界中的任何事物都可以看做是对象。

对象可以是具体的事物，也可以是抽象的概念。在 Visual FoxPro 中，任何一个控件都是一个对象。

2. 类（Class）

所谓类，是指一组具有相同属性和方法的对象的集合。基于类就可以生成这类对象中的任何一个对象，称为这个类的“实例”。这些“实例”具有相同的属性和方法，但是它们的属性取值却并不相同，同一个方法由不同的对象去执行，也会产生不同的结果。

10.1.2 对象的属性、方法和事件

每个对象都有一定的属性和适用的方法。

属性用于描述对象的状态、特征，如颜色、大小、位置等。方法是用于指明对象的操作、行为，如清除、刷新、显示等。

改变属性和方法可在设计时设置，也可以在运行时由命令控制。属性、方法集可以扩展。

与对象紧密相连的还有事件。所谓事件，是一种由系统预先定义而由用户或系统发出的动作，如单击、双击等。事件作用于对象，对象识别事件并做出相应的反应。事件集是固定的，用户不能定义新的事件。

使用面向对象设计方法解决问题，就是先抽象出所需要的对象及其属性和方法，再解决发生某种事件时，对象应该

做出何种反映。为选择的事件编写代码是面向对象程序设计的主要任务。

10.1.3 继承与父类、子类

在面向对象程序设计中，可以基于现有类建立一个新类，新类称为现有类的子类，现有类称为新类的父类。

继承表达了从一般到特殊的进化过程，是指基于现有类创建子类时，子类自动继承其父类的所有属性和方法。除了继承的属性和方法外，还可以为子类添加新的属性和方法，如果有必要，继承的方法也可以改变，如图 10-1 所示。

- 因此，在一个子类中，一般包括：
- (1) 从它的父类继承的属性、方法。
 - (2) 由子类自己定义的属性、方法。

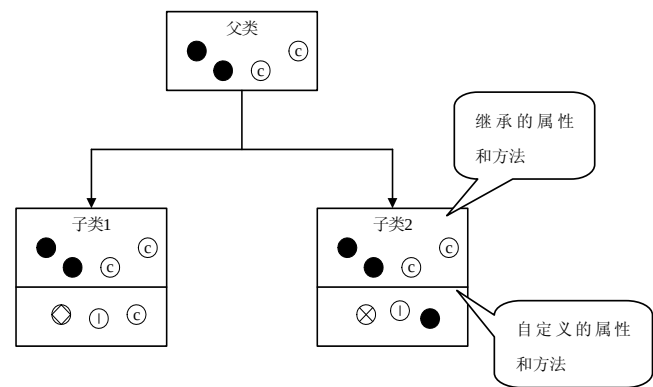


图 10-1 父类与子类的关系

10.2 Visual FoxPro 基类简介

在 Visual FoxPro 中，要进行面向对象的程序设计或创建应用程序，必然要用到 Visual FoxPro 系统的基类。

10.2.1 Visual FoxPro 基类

Visual FoxPro 中的基类就是系统本身内含的、并不存放在类库中的类。用户可以基于 Visual FoxPro 基类生成所需要的对象，如果有特殊要求，也可以扩展基类创建自己的类。

Visual FoxPro 中的基类如表 10-1 所示。

表 10-1 Visual FoxPro 基类

类名	含义	类名	含义	类名	含义
ActiveDoc	活动文档	FormSet	表单集	OptionGroup	选项按钮组
CheckBox	复选框	Grid	表格	Page	页
Column	（表格）列	Header	（列）标头	PageFrame	页框
ComboBox	组合框	HyperLink	超级链接	ProjectHook	项目挂钩
CommandButton	命令按钮	Image	图像	Separator	分隔符
CommandGroup	命令按钮组	Label	标签	Shape	形状
Container	容器	Line	线条	Spinner	微调按钮
Control	控件	ListBox	列表框	TextBox	文本框
Custom	定制	OleControl	OLE 容器控件	Timer	计时器
EditBox	编辑框	OleBoundControl	OLE 绑定控件	ToolBar	工具栏
Form	表单	OptionButton	选项按钮		

Visual FoxPro 中的基类每个都有自己的一组属性、方法和事件。当扩展某个基类创建用户自定义类时，该基类就是用户自定义类的父类，用户自定义类继承该基类中的属性、方法和事件。

所有基类都包含的属性叫做基类的最小属性集，如表 10-2 所示。
所有基类都包含的事件叫做基类的最小事件集，如表 10-3 所示。

表 10-2 Visual FoxPro 基类的最小属性集

属性	说明
Class	类名，当前对象基于哪个类而生成
BaseClass	基类名，当前类从哪个基类派生而来
ClassLibrary	类库名，当前类存放在哪个类库中
ParentClass	父类名，当前类从哪个类直接派生而来

表 10-3 Visual FoxPro 基类的最小事件集

事件	说明
Init	当对象生成时引发
Destroy	当对象从内存中释放时引发
Error	当方法或事件代码运行错误时引发

在编程方式里，对象的生成通常使用 CREATEOBJECT()函数完成，其格式为：
CREATEOBJECT (<类名>[, <参数 1>, <参数 2>, ...])
该函数用于基于指定的类生成一个对象，并返回对象的引用。通常，可以把函数返回的对象引用赋值给某一个变量，然后通过这个变量来标识、访问对象属性及调用对象方法。
访问对象属性及调用对象方法的基本格式：
<对象引用>.<对象属性>
<对象引用>.<对象方法>[(...)]

10.2.2 容器和控件

从包容关系上来划分，Visual FoxPro 中，类可以分为两种类型：容器类和控件类。从而可以分别生成容器对象和控件对象，简称为容器和控件。

1. 容器和控件的概念
- 容器是一种能够包容其他的控件或容器的特殊控件，如表单、表格等。
控件是指能以图形化的方式显示出来并能与用户进行交互的对象，如标签、线条、命令按钮、文本框等。
控件通常被放置在一个容器里。
2. 常用容器能包容的控件
- 常用容器及其所能包容的对象如表 10-4 所示。
容器内的包容关系形成了对象的嵌套层次关系。

表 10-4 常用容器及其所能包容的对象

容器	能包容的对象
表单集	表单、工具栏
表单	任意控件及页框、Container 对象、命令按钮组、选项按钮组、表格
表格	列
列	标头和除表单集、表单、工具栏、定时器及其他列之外的对象
页框	页

续表

容器	能包容的对象
页	任意控件及 Container 对象、命令按钮组、选项按钮组、表格等
命令按钮组	命令按钮
选项按钮组	选项按钮
Container 对象	任意控件及页框、命令按钮组、选项按钮组、表格等对象

3. 对象的引用

在对象的嵌套层次关系中，要引用其中的某个对象，需要指明对象在嵌套层次中的位置。
在对象引用时经常使用的属性或关键字如表 10-5 所示。

表 10-5 对象引用属性或关键字

属性或关键字	引用类型	说明
父对象名	绝对引用	直接在对象名前加上其容器父对象名
Parent	相对引用	当前对象的直接容器对象
This	相对引用	当前对象
ThisForm	相对引用	当前对象所在的表单
ThisFormSet	相对引用	当前对象所在的表单集

对象引用并设置对象属性的示例如表 10-6 所示。

表 10-6 对象引用示例

示例	说明
ThisFormSet.frm1.cmd1.Caption	设置当前表单集中表单 frm1 中控件 cmd 的标题属性
ThisForm.cmd1.Caption	设置当前表单中控件 cmd 的标题属性
This.Caption	设置当前对象的标题属性
This.Parent.Caption	设置当前对象的直接容器对象的标题属性

10.2.3 事件

事件是一种由系统预先定义而由用户或系统发出的动作。它作用于对象，对象识别事件并做出相应的反应。与方法集可以无限扩展不同的是，事件集是固定的，用户不能定义新的事件。

在 Visual FoxPro 中，一些常用的事件如表 10-7 所示。

表 10-7 Visual FoxPro 中常用的一些事件和方法

事件	功 能
Init 事件	在对象建立时引发
Destroy 事件	在对象释放时引发
Error 事件	当对象方法或事件代码在运行过程中产生错误时引发
Load 事件	在表单对象建立之前引发，即运行表单时，先引发 Load 事件，再引发 Init 事件
Unload 事件	在表单释放时引发，表单对象释放时最后一个要引发的事件。在 Destroy 事件之后引发
GotFocus 事件	当对象获得焦点时引发
Click 事件	用鼠标单击对象时引发
DblClick 事件	用鼠标双击对象时引发
RightClick 事件	用鼠标右键单击对象时引发

10.3 创建与运行表单

在 Visual FoxPro 中，可以通过表单设计器或者表单向导来创建表单，并通过运行表单文件来生成表单对象。

10.3.1 创建表单

在 Visual FoxPro 中，创建表单一般有两种方式：

- (1) 使用表单向导创建表单。
- (2) 使用表单设计器创建表单。

1. 使用表单向导创建表单

在 Visual FoxPro 中，在项目管理器中调用表单向导的步骤是：

- (1) 在“项目管理器”对话框中，选择“文档”选项卡，再选择其中的“表单”选项，如图 10-2 所示。
 - (2) 单击“新建”按钮，系统弹出“新建表单”对话框，如图 10-3 所示。
 - (3) 单击“表单向导”图标按钮，打开“向导选取”对话框，从列表框中选择要使用的向导，如图 10-4 所示。
- “向导选取”对话框中，有两种表单向导：
- 表单向导：用来创建基于一个表的表单。
 - 一对多表单向导：用来创建基于两个具有一对多关系的表的表单。
- (4) 单击“确定”按钮，打开“表单向导”对话框，如图 10-5 所示。
 - (5) 在“表单向导”对话框中，根据提示信息，一步一步地自动创建一个表单。

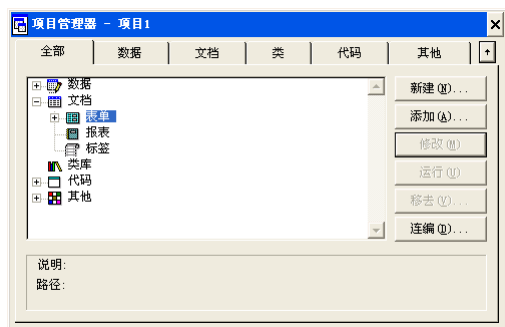


图 10-2 在“项目管理器”对话框中选择“表单”选项



图 10-3 “新建表单”对话框

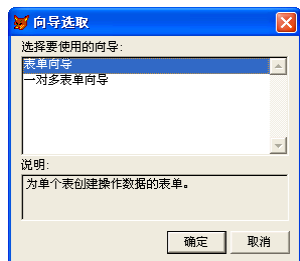


图 10-4 “向导选取”对话框

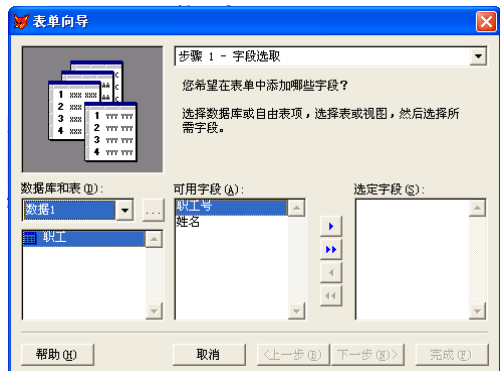


图 10-5 “表单向导”对话框

另外，也可以通过菜单操作打开“表单向导”，具体步骤如下：

- (1) 在主菜单“文件”菜单中，选择“新建”命令，打开“新建”对话框，如图 10-6 所示。
- (2) 在“文件类型”中选择“表单”选项，单击“向导”按钮。
- (3) 或者选择“工具”→“向导”→“表单向导”命令。

2. 使用表单设计器创建表单

使用表单设计器创建表单，首先需要打开表单设计器。打开表单设计器有以下 3 种方法。

1) 在项目管理器中打开表单设计器

在“项目管理器”对话框中，选择“文档”选项卡，选中“表单”选项，单击“新建”按钮，弹出“新建表单”对话框，如图 10-7 所示。单击“新建表单”按钮，则打开“表单设计器”对话框，如图 10-8 所示。

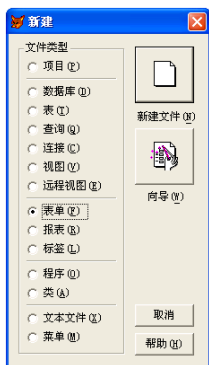


图 10-6 “新建”对话框



图 10-7 “新建表单”对话框

2) 使用菜单方式打开表单设计器

在主菜单“新建”菜单中，选择“新建”命令，打开“新建”对话框，从中选择“表单”文件类型，单击“新建文件”按钮，则打开“表单设计器”对话框。

3) 使用命令方式打开表单设计器

在命令窗口中输入 CREATE FORM 命令打开表单设计器。

在“表单设计器”对话框中，可以通过调用“表单生成器”快速、方便地生成表单。调用“表单生成器”有以下 3 种方式。

- (1) 在主菜单“表单”菜单中，选择“快速表单”命令。
- (2) 单击“表单设计器”工具栏中“表单生成器”按钮。
- (3) 在“表单设计器”中，单击鼠标右键，在快捷菜单中选择“生成器”命令。

通过以上 3 种方法，即可打开“表单生成器”对话框，如图 10-9 所示。

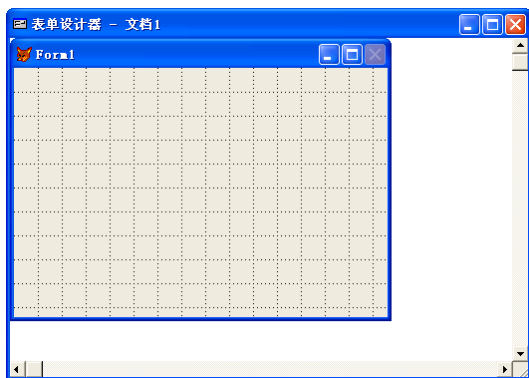


图 10-8 “表单设计器”对话框

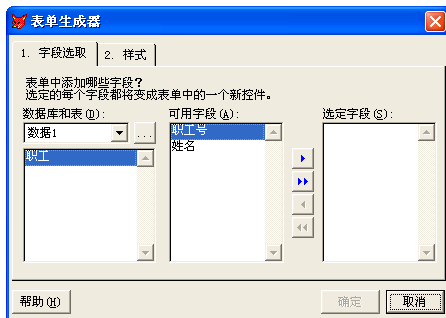


图 10-9 “表单生成器”对话框

10.3.2 修改表单

无论通过哪种方法创建的表单，都可以使用表单设计器进行编辑、修改。当要修改某个表单时，有以下 3 种方式打开“表单设计器”。

(1) 在“项目管理器”对话框中，选择“文档”选项卡，选中需要修改的表单文件，单击“修改”按钮，即可进入“表单设计器”对话框。

(2) 在主菜单“文件”菜单中，选择“打开”命令，弹出“打开”对话框，在“文件类型”选项组中选择“表单”选项，然后选择需要修改的表单文件，单击“确定”按钮，则打开“表单设计器”对话框。

(3) 在命令窗口中，输入 `MODIFY FORM <表单文件名>` 命令，打开“表单设计器”对话框。

10.3.3 运行表单

所谓运行表单，就是根据表单文件及表单备注文件的内容生成表单对象。有以下 4 种方法运行表单。

(1) 在“项目管理器”对话框中，选择“文档”选项卡，选中需要运行的表单，单击“运行”按钮，运行该表单。

(2) 在“表单设计器”对话框中，在主菜单“表单”菜单中，选择“执行表单”命令，或者单击系统工具栏中“运行”按钮，运行该表单。

(3) 在主菜单“程序”菜单中，选择“运行”命令，打开“运行”对话框，从中选择需要运行的表单文件，单击“运行”按钮，运行该表单。

(4) 在命令窗口中，输入运行表单命令，命令格式是：

```
DO FORM <表单文件名> [NAME<变量名>]
WITH <实参>[, <实参>, ...] [LINKED] [NOSHOW]
```

其中：

- **NAME<变量名>**：系统将建立指定名字的用来指向表单对象的变量，否则，系统将建立一个与表单同名的用来指向表单对象的变量。
- **WITH <实参>**：当表单运行，触发 Init 事件时，系统将所列各实参的值传递给该事件代码 PARAMETERS 或 LPARAMETERS 子句中的各形参。
- **[LINKED]**：表单对象将随着指向它的变量的清除而被释放（关闭），否则，表单对象依然存在。
- **[NOSHOW]**：表单运行时将不会显示，否则，默认状态是显示该表单。

10.4 表单设计器

在“表单设计器”中，可以添加表单控件、管理表单控件和设置表单数据环境，从而设计出一个完善的表单对象。

10.4.1 表单设计器环境

打开“表单设计器”对话框后，如图 10-10 所示，Visual FoxPro 主窗口中将出现“表单设计器”对话框、“属性”窗口、“表单控件”工具栏、“表单设计器”工具栏及“表单”菜单。

1. “表单设计器”对话框

“表单设计器”对话框用来显示新建或修改的表单，用户可以利用可视化的方式添加和修改控件。关闭该对话框，即关闭“表单设计器”，如图 10-11 所示。

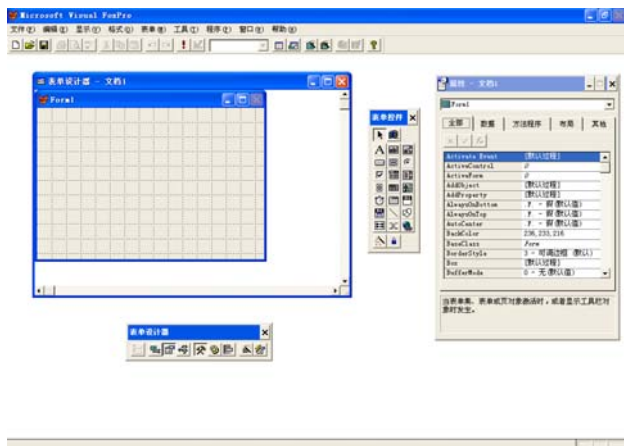


图 10-10 “表单设计器”环境

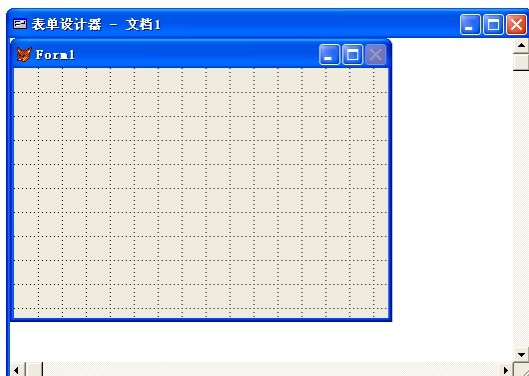


图 10-11 “表单设计器”对话框

2. “属性”窗口

“属性”窗口用来设置属性，包括对象框、属性设置框和属性、方法、事件列表框。如图 10-12 所示。

在“表单设计器”对话框中，显示“属性”窗口的方法有如下 3 种：

- (1) 单击“表单设计器”工具栏中“属性窗口”按钮。
- (2) 选择主菜单“显示”菜单中“属性”命令。
- (3) 在“表单设计器”对话框中，选中对象，单击鼠标右键，在快捷菜单中选择“属性”命令。

“属性”窗口在表单设计中具有重要的作用，任何控件（包括表单）都是通过该窗口设置属性的。

在“属性”窗口中设置属性的步骤如下：

- (1) 在“表单设计器”中选定对象，或者在“属性”窗口的“对象框”中选定对象。
- (2) 在“属性”列表框中选定属性。
- (3) 在“属性设置框”中，输入或选择属性设置，如图 10-13 所示。

3. “表单控件”工具栏

“表单控件”工具栏中含有各种控件，可以方便地向表单中添加控件，如图 10-14 所示。

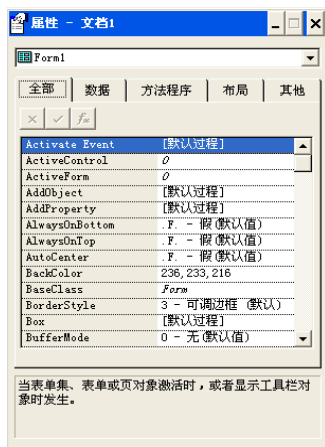


图 10-12 “属性”窗口

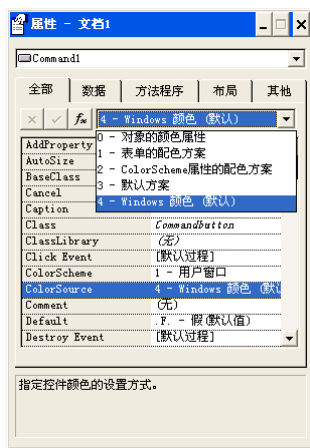


图 10-13 设置属性



图 10-14 “表单控件”工具栏

利用“表单控件”工具栏向表单中添加控件的方法是：单击“表单控件”工具栏中相应的控件按钮，然后在表单中

适当的位置单击鼠标，或者拖曳鼠标以确定控件的大小。

“表单控件”工具栏可以通过单击“表单设计器”工具栏中的“表单控件”按钮，或者在主菜单“显示”菜单中选择“工具栏”命令来打开或关闭。

“表单控件”工具栏中包括控件按钮和 4 个辅助按钮。“表单控件”工具栏中辅助按钮的功能如表 10-8 所示。

表 10-8 “表单控件”工具栏中辅助按钮的功能

辅助按钮	功能
选定对象	当按钮处于按下状态时，表示不可创建控件，此时可以对已经创建的控件进行编辑；当按钮处于未按下状态时，表示允许创建控件
按钮锁定	当按钮处于按下状态时，可以从“表单控件”工具栏中单击选定某种控件按钮，然后在表单窗口中连续添加这种类型的多个控件
生成器锁定	当按钮处于按下状态时，每次往表单中添加控件，系统都会自动打开相应的生成器对话框，以便用户对该控件的常用属性进行设置
查看类	向“表单控件”工具栏中添加类

4. “表单设计器”工具栏

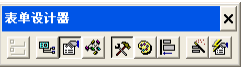


图 10-15 “表单设计器”工具栏

“表单设计器”工具栏可以用来打开各种工具和窗口，如图 10-15 所示。

“表单设计器”工具栏中包括“设置 Tab 键次序”、“数据环境”、“属性窗口”、“代码窗口”、“表单控件工具栏”、“调色板工具栏”、“布局工具栏”、“表单生成器”和“自动格式”等按钮。

可以通过选择主菜单“显示”菜单中“工具栏”命令，在打开的“工具栏”对话框中选择“表单设计器”选项来打开或关闭。

5. “表单”菜单

“表单”菜单用来创建、编辑表单或表单集，为表单增加新的属性和方法等。当“表单设计器”对话框打开时，“表单”菜单自动显示。

10.4.2 控件的操作与布局

在“表单设计器”对话框中，可以对表单中的控件进行移动、复制、布局等操作，也可以为控件设置 Tab 键次序。

1. 控件的基本操作

对表单上的控件的常用操作和实现方式如表 10-9 所示。

表 10-9 控件的常用操作和实现方式

常用操作	实现方式
选定控件	用鼠标单击控件即可选定该控件，被选中控件四周出现 8 个控点；通过鼠标拖动选取多个相邻控件；通过【Shift】键和依次单击鼠标，可以选取多个不相邻控件
移动控件	选定控件，然后用鼠标将控件拖曳到指定位置；或者使用方向键
改变大小	选定控件，然后拖动控件四周的某个控点就可以改变控件的大小
复制控件	选定控件，通过“编辑”菜单中的“复制”命令进行复制
删除控件	选定控件，然后按【Delete】键

2. 控件布局

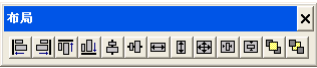


图 10-16 “布局”工具栏

利用“布局”工具栏，可以调整表单中控件的位置和相对大小。布局工具栏如图 10-16 所示。

“布局”工具栏可以通过单击“表单设计器”工具栏中的“布局工具栏”按钮或者选择主菜单“显示”菜单中的“布局工具栏”命令来打开或关闭。

“布局”工具栏上的按钮及其功能如表 10-10 所示。

表 10-10 “布局”工具栏个按钮功能

按钮	功能
左边对齐	让选定的所有控件按照其中最左边控件的左侧对齐
右边对齐	让选定的所有控件按照其中最右边控件的右侧对齐
顶边对齐	让选定的所有控件按照其中最顶端控件的顶边对齐
底边对齐	让选定的所有控件按照其中最下端控件的底边对齐
垂直居中对齐	让选定的所有控件的中心处在一条垂直线上
水平居中对齐	让选定的所有控件的中心处在一条水平线上
相同宽度	调整被选定的所有控件的宽度，使其与其中最宽控件的宽度相同
相同高度	调整被选定的所有控件的高度，使其与其中最高控件的高度相同
相同大小	使被选定的所有控件具有相同的大小
水平居中	使被选定的所有控件在表单内水平居中
垂直居中	使被选定的所有控件在表单内垂直居中
置前	将被选定控件移至最前面，可能会把其他控件覆盖
置后	将被选定控件移至最后面，可能会被其他控件覆盖

3. 设置 Tab 键次序

当表单运行时，用户可以按【Tab】键选择表中的控件，使焦点在控件间移动。控件的【Tab】键次序决定了选择控件的次序。

Visual FoxPro 提供了两种方式来设置 Tab 键次序：交互方式和列表方式。

可以通过以下步骤选择想要使用的设置方式：

- (1) 在主菜单“工具”菜单中选择“选项”命令，弹出“选项”对话框。如图 10-17 所示。
- (2) 选择“表单”选项卡，在“Tab 键次序”下拉列表框中选择“交互”或“按列表”方式，如图 10-18 所示。

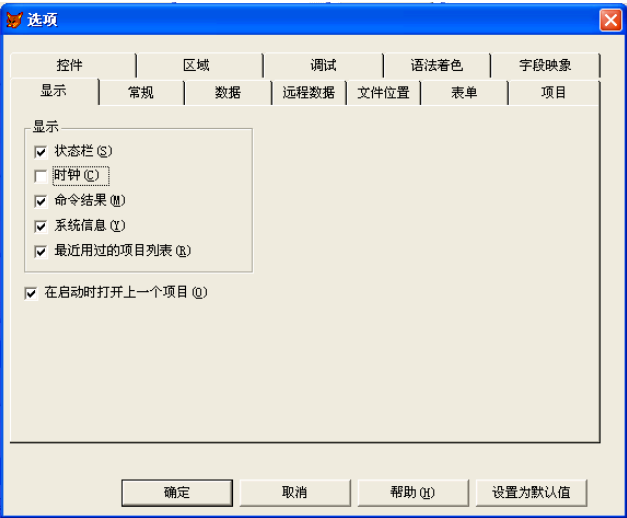


图 10-17 “选项”对话框

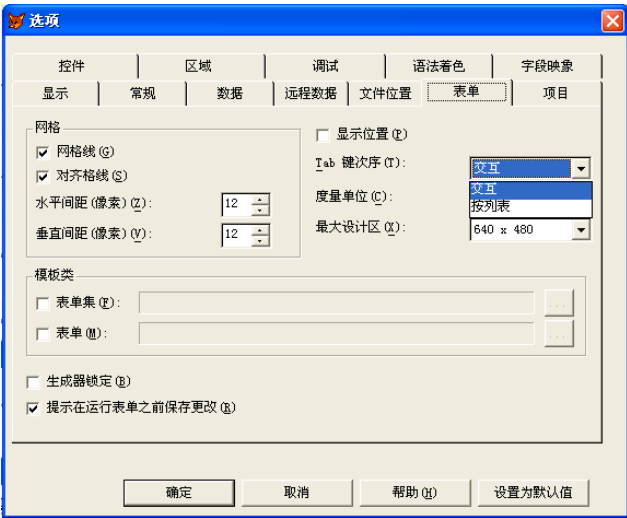


图 10-18 设置“Tab 键次序”

在交互方式下，设置 Tab 键次序的步骤如下：

- (1) 在“表单设计器”对话框中，在主菜单“显示”菜单中选择“Tab 键次序”命令，或者单击“表单设计器”工

具栏中“设置 Tab 键次序”按钮，进入 Tab 键次序设置状态，如图 10-19 所示。

- (2) 此时，在控件左上方出现深色的小方块，称为 Tab 键次序盒，显示出该控件的 Tab 键次序号码。
- (3) 双击某个控件的 Tab 键次序盒，则该控件成为 Tab 键次序中的第一个控件，然后按照希望的顺序依次单击其他控件的 Tab 键次序盒。
- (4) 单击表单空白处，确认设置；按【Esc】键，放弃设置。

在列表方式下，设置 Tab 键次序的步骤如下：

- (1) 在“表单设计器”对话框中，在主菜单“显示”菜单中选择“Tab 键次序”命令，或者单击“表单设计器”工具栏中“设置 Tab 键次序”按钮，进入 Tab 键次序设置状态，如图 10-20 所示。
- (2) 在列表框中按 Tab 键次序显示各控件，通过拖动控件左侧的移动按钮移动控件，改变控件的 Tab 键次序。
- (3) 单击“按行”按钮，将按各控件在表单上的位置从上到下、从左到右自动设置各控件的 Tab 键次序；单击“按列”按钮，将按各控件在表单上的位置从左到右、从上到下自动设置各控件的 Tab 键次序。

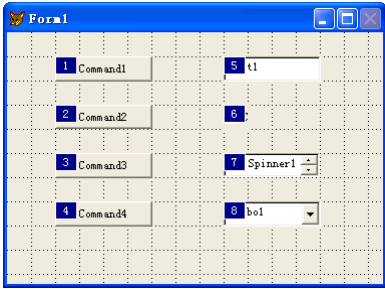


图 10-19 交互方式设置 Tab 键次序



图 10-20 列表方式设置 Tab 键次序

10.4.3 数据环境

数据环境是表单要处理的数据的存放场所，通过为表单建立数据环境，可以方便地设置控件与数据之间的绑定关系。数据环境中包含表单所需要的数据，包括表、视图及表之间的关联，它们随着表单的运行而打开和建立，并随着表单的关闭或释放而关闭。

在 Visual FoxPro 中，可以通过“数据环境设计器”可视化地设置表单的数据环境。

1. 数据环境的常用属性

数据环境也是 Visual FoxPro 中的一个对象，因此数据环境也有自己的属性、方法和事件。数据环境常用的两个属性是：AutoOpenTables 和 AutoCloseTables，它们的设置如表 10-11 所示。

表 10-11 常用的数据环境属性

属性名	含义	默认值
AutoOpenTables	当运行或打开表单时，是否打开数据环境中的表和视图	.T.
AutoCloseTables	当释放或关闭表单时，是否关闭数据环境中的表和视图	.T.

2. 打开数据环境设计器

在 Visual FoxPro 中，打开“数据环境设计器”有 3 种方法。

- (1) 在“表单设计器”对话框中，单击“表单设计器”工具栏中“数据环境”按钮，即可打开“数据环境设计器”对话框，如图 10-21 所示。
- (2) 在主菜单“显示”菜单中选择“数据环境”命令，即可打开“数据环境设计器”对话框。
- (3) 在“表单设计器”对话框中，单击鼠标右键，在快捷菜单中选择“数据环境”命令，即可打开“数据环境设计器”对话框。

3. 向数据环境中添加表或视图

向数据环境中添加表或视图的步骤如下：

(1) 在“数据环境设计器”对话框中，单击鼠标右键，在快捷菜单中选择“添加”命令，或者在主菜单“数据环境”菜单中选择“添加”命令，弹出“添加表或视图”对话框，如图 10-22 所示。

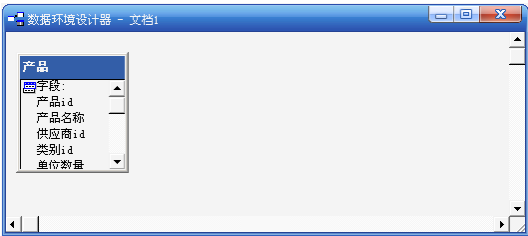


图 10-21 “数据环境设计器”对话框



图 10-22 “添加表或视图”对话框

(2) 选择需要添加的表或视图，单击“添加”按钮。如果单击“其他”按钮，则弹出“打开”对话框，从中选择需要添加的表或视图。

如果添加到表单数据环境的表在其所属数据库中存在永久关系，则这些关系将如同表或视图一样，作为一个对象自动添加到数据环境中。

4. 从数据环境中移去表或视图

从数据环境中移去表或视图的步骤如下：

(1) 在“数据环境设计器”对话框中，选择需要移去的表或视图。

(2) 在主菜单“数据环境”菜单中选择“移去”命令，或者在“数据环境设计器”对话框中，用鼠标右键单击需要移去的表或视图，在快捷菜单中选择“移去”命令。

当表从表单数据环境中移去时，如果存在与表关联的永久或临时关系，则与这个表有关的所有关系也将随之消失。

5. 在数据环境中设置关联

如果添加到数据环境的表之间具有在数据库中设置的永久关系，则这些关系也会自动添加到数据环境中。如果表之间没有永久关系，则可以根据需要在数据环境设计器下为这些表设置关系。

设置关系的方法是：将主表的某个字段（作为关联表达式）拖曳到子表的相匹配的索引标记上即可。如果子表中没有与主表字段相匹配的索引，也可以将主表字段拖曳到子表的某个字段上，系统会提示确认创建索引。

在“数据环境设计器”对话框中，表之间的关联用一条线来表示，如图 10-23 所示。

如果需要删除表之间的关联，可以单击选中表示关联的连线，然后按【Del】键。

6. 在数据环境中编辑关联

在数据环境中，表间关系也是一个独立的对象，它有自己的属性、方法和事件。可以在“属性”窗口中选择这个关系对象来设置它的属性。

常用的关系属性如表 10-12 所示。



图 10-23 在数据环境中设置关联

表 10-12 常用的关系属性

属性名	含义
RelationalExpr	用于指定基于主表的关联表达式
ParentAlias	用于指定主表的别名
ChildAlias	用于指定子表的别名
ChildOrder	用于指定表中与关联表达式相匹配的控制索引
OneToMany	用于指定关系是否为一对多关系

在数据环境中，编辑关系主要是通过设置关系属性来完成。具体的方法是：单击关系连线，选定关系，然后在“属性”窗口中设置其属性。

7. 向表单添加字段

在 Visual FoxPro 中，不仅可以使用“表单控件”工具栏向表单中添加控件，还可以从“数据环境设计器”对话框中直接将字段、表或视图拖曳进表单，系统将产生相应的控件并与字段相联系。

在默认情况下，如果拖曳的是字符型字段，将产生文本框控件；如果拖曳的是备注型字段，将产生编辑框控件；如果拖曳的是表或视图，将产生表格控件。

另外，可以通过选择主菜单“工具”菜单中“选项”命令，打开“选项”对话框，从中选择“字段映像”选项卡，从中修改默认情况下的这种映像关系。

10.5 表单属性和方法

表单中包含许多固定的属性和方法，另外，还可以根据需要为表单添加新的属性和方法。

10.5.1 常用的表单属性

表单中常用的属性如表 10-13 所示。

表 10-13 表单中常用的属性

属性名	含义	默认值
AlwaysOnTop	指定表单是否总是位于其他打开的窗口之上	.F.
AutoCenter	指定表单初始化时是否自动在 Visual FoxPro 主窗口内居中显示	.F.
BackColor	指定表单窗口的颜色	255, 255, 255
BorderStyle	指定表单边框的风格	3
Caption	指定表单标题栏上的文本	Form1
Closable	指定是否可以通过单击“关闭”按钮或双击控制菜单框来关闭表单	.T.
DataSession	指定表单里的表在哪个工作区打开：1 全局工作区、2 私有工作区	1
MaxButton	确定表单是否有最大化按钮	.T.
MinButton	确定表单是否有最小化按钮	.T.
Movable	确定表单是否能够移动	.T.
ScrollBars	指定表单滚动条类型：0 无、1 水平、2 垂直、3 既水平又垂直	0
WindowState	指定表单的状态：0 正常、1 最小化、2 最大化	0
WindowType	指定表单的类型：0 非模式表单、1 模式表单	0

10.5.2 常用的事件和方法

事件的触发依赖于一定的触发条件，这个条件可以是由用户触发的，也可以是由系统触发的。
表单中常用的事件如表 10-14 所示。

表 10-14 表单中常用的事件

类型	事件	含义
运行	Load	在表单对象建立之前触发，即运行表单时，先触发 Load 事件，然后再触发 Init 事件
	Init	在表单对象建立时触发，即运行表单时，先触发表单中控件对象的 Init 事件，然后再触发表单的 Init 事件，因此在表单 Init 事件代码中能够访问它所包含的所有控件对象
关闭	Destroy	在表单对象释放时触发，即关闭表单时，先发表单的 Destroy 事件，然后再触发表单中控件对象的 Destroy 事件，因此在表单 Destroy 事件代码中能够访问它所包含的所有控件对象
	Unload	在表单对象释放时触发，是表单对象释放时最后一个触发的事件，即关闭表单时，先触发 Destroy 事件，再触发 Unload 事件
交互	GotFocus	在对象获得焦点时触发
	Click	在用鼠标单击对象时引发
	DbClick	在用鼠标双击对象时引发
	RightClick	在用鼠标右键单击对象时引发
	InteractiveChange	在用鼠标或键盘交互式改变对象的值时引发，每次更改对象时，都要触发此事件
错误	Error	在对象方法或事件代码运行过程中产生错误时触发

表单中常用的方法如表 10-15 所示。

表 10-15 表单中常用的方法

方法	作用	说明
Show	显示表单	将表单的 Visible 属性设置为.T.，并使其成为活动对象
Hide	隐藏表单	将表单的 Visible 属性设置为.F.
Release	释放表单	将表单从内存中释放（清除）
Refresh	刷新表单或控件	重新绘制表单或控件，并刷新它的所有值
SetFocus	设置焦点	让控件获得焦点，使其成为活动对象

10.5.3 添加新的属性和方法

在 Visual FoxPro 中，可以根据需要向表单中添加新的属性和方法。

1. 创建新属性

向表单中添加新属性时，表单必须处于打开状态。

向表单中添加新属性的步骤如下：

- (1) 在主菜单“表单”中选择“新建属性”命令，打开“新建属性”对话框，如图 10-24 所示。
- (2) 在“名称”文本框中输入属性名称，该新建的属性也会在“属性”窗口的列表框中显示。
- (3) 可以根据需要在“说明”编辑框中输入新建属性的说明信息，这些信息将会在“属性”窗口的底部显示。

2. 创建新方法

向表单中添加新方法时，表单也必须处于打开状态。

向表单中添加新方法的步骤如下：

(1) 在主菜单“表单”中选择“新建方法程序”命令，弹出“新建方法程序”对话框，如图 10-25 所示。



图 10-24 “新建属性”对话框

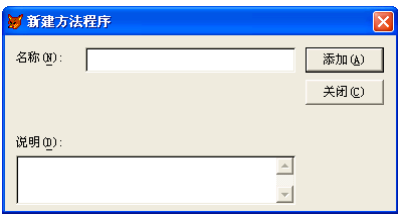


图 10-25 “新建方法程序”对话框

- (2) 在“名称”文本框中输入方法名称，该新建的方法也会在“属性”窗口的列表框中显示。
- (3) 可以根据需要在“说明”编辑框中输入新建方法的说明信息，这些信息也将会在“属性”窗口的底部显示。
- 如果需要删除用户创建的属性和方法，可以在主菜单“表单”菜单中选择“编辑属性/方法程序”命令，弹出“编辑属性/方法程序”对话框，选中需要删除的属性或方法，单击“移去”按钮，如图 10-26 所示。

3. 编辑方法或事件代码

- 在“表单设计器”对话框中，编辑方法或事件代码的步骤如下：
- (1) 在主菜单“显示”菜单中选择“代码”命令，打开代码编辑窗口，如图 10-27 所示。或者双击表单或表单中的某个控件，也可以打开代码编辑窗口。
- (2) 从“对象”组合框中选择方法或事件所属的对象，即表单或表单中的控件。
- (3) 从“过程”组合框中指定需要编辑的方法或事件。
- (4) 在“编辑区”中输入或修改方法或事件的代码。



图 10-26 “编辑属性/方法程序”对话框

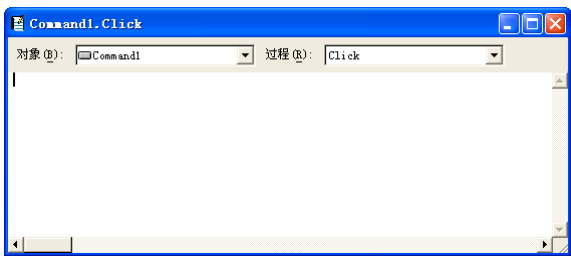


图 10-27 代码编辑窗口

10.6 基本型控件

表单设计离不开控件对象，因此要更好地使用控件，就必须了解其属性、方法和事件。本节将主要介绍各种控件的使用和设计。

控件可以分为基本型控件和容器型控件。基本型控件是指不能包含其他控件的控件，如标签、命令按钮、文本框、编辑框和列表框等；容器型控件是指可包含其他控件的控件，如选项组、命令组和表格等。

10.6.1 标签

标签是用以显示一段固定文本信息字符串的图形控件。标签的标题文本不能在屏幕上直接编辑修改，但可以在代码

中通过重新设置 Caption 属性间接修改。另外，标签不能获得焦点。
标签控件的常用属性如表 10-16 所示。

表 10-16 标签控件的常用属性

属性	说明
Caption	指定标签的标题文本，并且可以将其中的某个字符设置为访问键，方法是在该字符前插入一个反斜杠和一个小于号(\<)。标签标题文本最多为 256 个字符
Alignment	指定标题文本在控件中的对齐方式，取值为 0，左对齐；1，右对齐；2，居中对齐。

10.6.2 命令按钮

命令按钮可以用来启动某个事件代码从而完成特定功能。一般将实现其功能的代码放置在 Click 事件中，用户通过单击命令按钮执行其代码。
命令按钮控件的常用属性与事件如表 10-17 所示。

表 10-17 命令按钮控件的常用属性与事件

属性与事件	说明
Caption 属性	设置命令按钮标题。若在该属性的某字符前加“\<”，该字符就成为热键
Enabled 属性	用于确定命令按钮对象是否响应用户事件，默认值为.T.
Visible 属性	用于确定命令按钮对象是可见还是隐藏，默认值为.T.
Default 属性	将命令按钮设为默认命令按钮，当所有命令按钮都未获得焦点时，用户按回车键，则执行默认命令按钮的 Click 事件；表单中只能有一个默认命令按钮
Cancel 属性	属性值为.T.时，用户按【Esc】键执行该命令按钮的 Click 事件；默认值为.F.
Picture 属性	设置命令按钮上显示的图形
ToolTipText 属性	设置命令按钮的提示文本；在设置该属性值之前，必须设置表单的 ShowTip 属性值为.T.
Click 事件	命令按钮的单击鼠标左键事件
RightClick 事件	命令按钮的单击鼠标右键事件

10.6.3 文本框

文本框用于编辑一行文字、数字或其他输入信息。文本框可以编辑任何类型的数据。文本框的常用属性如表 10-18 所示。

表 10-18 文本框的常用属性

属性	说明
Value	指定文本框对象的值，并在框中显示。同时，文本框的编辑结果也保存到该属性中。如果遇到长数据能自动换行，若按下回车键，则终止数据输入
ControlSource	设置文本框对象的数据源。文本框控件的数据源可以是字段和变量两种，若是字段，则必须是来自数据环境中的表。当使用字段映像功能创建对象时，系统将自动设置该对象的数据源属性
Format	指定 Value 属性输入和显示时的格式
PasswordChar	指定文本框控件内显示用户输入的是字符还是占位符；指定用做占位符的字符。无论输入任何字符，均显示此属性设置的字符。默认值是空串，无占位符。通常在设计登录口令框时使用
InputMask	用于确定控件中如何输入和显示数据。这与 Format 属性不同，Format 属性规定了整个文本框的格式，而 InputMask 属性规定了同一位置字符的格式

文本框控件的各种模式符及功能如表 10-19 所示。

表 10-19 模式符及其功能

模式符	说明
X	允许输入任何字符
9	允许输入数字和正负号
#	允许输入数字、空格和正负号
\$	在固定位置上显示当前货币符号（由 SET CURRENCY 命令指定）
\$\$	在数值前面相邻的位置上显示当前货币符号（浮动货币符）
*	在数值左边显示星号*
.	指定小数点的位置
,	分隔小数点左边的数字串

10.6.4 编辑框

编辑框可以输入和修改多行文本，如备注字段的内容。编辑框具有以下特点：

- （1）编辑框实际上是一个完整的字处理器，利用它能够选择、剪切、复制和粘贴正文，实现自动换行。
- （2）编辑框有自己的垂直滚动条，可以用箭头键在正文里面移动鼠标指针。
- （3）在编辑框中，只能输入、编辑字符型数据，包括字符型内存变量、数组元素、字段及备注字段里的内容。

编辑框的常用属性如表 10-20 所示。

表 10-20 编辑框的常用属性

属性	说明
Value	用于指定编辑框中的值，该属性值的类型只能是字符型。按回车键不能终止数据的输入
Readonly	指定用户能否编辑编辑框中的内容，默认值是.F.。若设置为.T.，则不能编辑，但是仍然可以移动焦点到编辑框上，并能够使用滚动条
ScrollBar	指定编辑框是否具有滚动条。其默认值为 2，表示当编辑框包含的数据超出编辑框所提供的空间时，显示垂直滚动条；为 0 时则无滚动条
ControlSource	设置编辑框对象的数据源，一般是数据环境表的某一备注型字段
SelText	返回用户在编辑框中所选定的文本内容，如果没有选定任何文本，则返回空串。该属性在设计时不可用，在运行时可读写
SelStart	返回用户在编辑框中所选文本的起始点位置或插入点位置，也可以用于指定要选择文本的起始位置或插入点位置。属性的有效取值范围在 0 与编辑区中字符总数之间。该属性在设计时不可用，在运行时可读写
SelLength	返回用户在控件的文本输入区中所选字符的数目，或指定要选定的字符数目。若没有文本被选定，则返回 0。属性的有效取值范围在 0 与编辑区中字符总数之间。该属性在设计时不可用，在运行时可读写
HideSelection	指定当编辑框失去焦点时，编辑框中选定的文本是否仍显示为选定状态，默认值是.T.

10.6.5 复选框

复选框用于标记一个二值状态，如真（.T.）或假（.F.）。当处于真状态时，复选框内显示一个对勾，否则复选框内为空白。

复选框的常用属性如表 10-21 所示。

表 10-21 复选框的常用属性

属性	说明
Caption	用来指定复选框旁边的文字
Alignment	用来指定复选框是显示在标题右边还是左边。默认值是 0，表示复选框显示在标题左边
Value	用来指定复选框的当前状态。默认值是 0（.F.），表示未选；1（.T.），表示被选；2(NULL.)，表示不确定
ControlSource	指定与复选框建立联系的数据源。作为数据源的字段变量或内存变量的数据类型是数值型或逻辑型

10.6.6 列表框

列表框提供一组条目（数据项），用户可以从选择一个或多个条目。在一般情况下，列表框显示其中的若干条目，用户可以通过滚动条浏览其他条目。该控件的常用属性如表 10-22 所示。

表 10-22 列表框的常用属性

属性	说明
RowSourceType	指明列表框中条目的数据源的数据类型
RowSource	指定列表框中条目的数据源
List	用以存取列表框中数据条目的字符串数组。该属性在设计时不可用，在运行时可读写
ListCount	指明列表框中数据条目的数目。该属性在设计时不可用，在运行时只读
ColumnCount	指定列表框的列数，即一个条目中包含的数据项数目
Value	返回列表框中被选中的条目。该属性值可以是数值型也可以是字符型。字符型，其值是被选条目的内容；数值型，其值是被选条目在列表框中的序号。对于列表框，该属性只读
BoundColumn	指明列上的数据项。对于列表框，该属性只读。该属性的域值及类型总是与 ControlSource 属性所指定的字段或内存变量的取值及类型保持一致
ControlSource	用户可以通过该属性指定的一个字段或变量用以保存用户从列表框中选择的结果，其类型可以是字符型或数值型。字符型，其值是被选条目的内容；数值型，其值是被选条目在列表框中的序号
Selected	指定列表框内的某个条目是否处于选定状态。该属性在设计时不可用，在运行时可读写
MultiSelect	指定用户能否在列表框控件内进行多重选定。默认值是 0 或.F.，不允许多重选择；1 或.T.，允许多重选择

其中，RowSourceType 和 RowSource 属性配合使用，其取值和配合说明如表 10-23 所示。

表 10-23 RowSourceType 取值和 RowSource 属性配合说明

属性值	含义	说明
0	无（默认值）	程序运行时通过 AddItem 方法添加条目，通过 RemoveItem 方法移去条目
1	值	通过 RowSource 属性手工指定条目
2	别名	RowSource=表名，用 ColumnCount 指定该表中字段数，即列表框的列数
3	SQL 语句	RowSource=“SQL 语句”，将执行结果作为列表框条目的数据源
4	查询	RowSource=“查询文件全名”，将查询结果作为列表框条目的数据源
5	数组	RowSource=数组名，将数组中的内容作为列表框条目的数据源
6	字段	RowSource=“字段名列表”，将字段内容作为列表框条目
7	文件	把 RowSource 设置为文件类型，将文件名作为列表框条目
8	结构	将表中的字段名作为列表框的条目，由 RowSource 指定表
9	弹出菜单	将弹出式菜单作为列表框条目的数据源

10.6.7 组合框

与列表框相似，组合框也是提供一组条目供用户选择，其属性、方法也大多与列表框类似，建立和使用的方法也基本相同。

组合框与列表框的主要区别如下：

- （1）组合框通常只显示一个条目，其他条目需要单击下拉箭头才能展开，节省屏幕空间。
- （2）组合框不提供多重选择的功能，无 MultiSelect 属性。
- （3）组合框有两种形式：下拉组合框和下拉列表框，通过设置 style 属性可选择想要的形式。
 - 0：下拉组合框。用户既可以从列表中选择，也可以在编辑区内输入，输入的内容可以从 Text 属性中获得。

- 1：下拉列表框。用户只能从列表中选择。
该控件的常用属性如表 10-24 所示。

表 10-24 组合框的常用属性

属性	说明
RowSourceType	指明组合框中条目的数据源的数据类型
RowSource	指定组合框中条目的数据源
List	用以存取组合框中数据条目的字符串数组。该属性在设计时不可用，在运行时可读写
ListCount	指明组合框中数据条目的数目。该属性在设计时不可用，在运行时只读
ColumnCount	指定组合框的列数，即一个条目中包含的数据项数目
Value	返回组合框中被选中的条目。该属性值可以是数值型也可以是字符型。字符型，其值是被选条目的内容；数值型，其值是被选条目在组合框中的序号。对于组合框，该属性只读
BoundColumn	指明列上的数据项。对于组合框，该属性只读。该属性的域值及类型总是与 ControlSource 属性所指定的字段或内存变量的取值及类型保持一致
ControlSource	用户可以通过该属性指定的一个字段或变量用以保存用户从组合框中选择的结果，其类型可以是字符型或数值型。字符型，其值是被选条目的内容；数值型，其值是被选条目在组合框中的序号
Selected	指定组合框内的某个条目是否处于选定状态。该属性在设计时不可用，在运行时可读写

10.7 容器型控件

容器型控件简称容器。容器与其所包含的控件都具有自己的属性、方法和事件。在“表单设计器”中，为了给容器内的某个控件设置属性、方法或事件，需要首先选中该控件。选中容器中某个控件，有以下两种方法：

- (1) 在“属性窗口”中的对象下拉列表框中选择容器内的某个需要设置的控件。
- (2) 用鼠标右键单击容器，然后在快捷菜单中选择“编辑”命令，此时容器进入编辑状态，可以通过鼠标单击选择容器内的某个控件。

10.7.1 命令组

命令组控件是包含一组命令按钮的容器控件，命令组和命令组中的每个命令按钮都有自己的属性、方法和事件。既可以对命令组中单个命令按钮进行设置，也可以作为一组操作其所包含的命令按钮。

命令组控件的常用属性如表 10-25 所示。

表 10-25 命令组控件的常用属性

属性	说明
ButtonCount	指定命令组中命令按钮的个数，默认值是 2，即包括两个命令按钮
Buttons	用于存取命令组中各按钮的数组，该数组在创建命令组时建立。用户可以利用该数组为命令组中的命令按钮设置属性或调用方法，属性数组下标的取值范围应该在 1 和 ButtonCount 属性值之间。该属性在设计时不可用
Value	指定命令按钮当前状态，即用户选择了哪个按钮。该属性的类型可以是数值型，也可以为字符型。默认是数值型，表示第几个命令按钮被选中；若是字符型，则表示 Caption 值为该字符的按钮被选中

10.7.2 选项组

选项组是包含选项按钮的容器。通常，选项按钮允许用户指定对话框中几个操作选项中的一个，而不是输入数据。用户只能从中选择一个按钮，被选中的选项按钮中会显示一个圆点，而选项组中的其他选项按钮，则变为未选中状态。

该控件的常用属性如表 10-26 所示。

表 10-26 选项组控件的常用属性

属性	说明
ButtonCount	指定选项组中选项按钮的个数，默认值是 2，即包括两个选项按钮
Value	指定选项按钮当前状态，即用户选择了哪个按钮。该属性的类型可以是数值型，也可以为字符型。默认是数值型，表示第几个选项按钮被选中；若是字符型，则表示 Caption 值为该字符的按钮被选中
ControlSource	指明与选项组建立联系的数据源。作为选项组数据源的字段变量或内存变量，其类型可以是数值型或字符型。操作结果会自动存储到数据源变量及 Value 属性中
Buttons	用于存取选项组中各按钮的数组，该数组在创建选项组时建立。用户可以利用该数组为选项组中的选项按钮设置属性或调用方法，属性数组下标的取值范围应该在 1 和 ButtonCount 属性值之间。该属性在设计时不可用

10.7.3 表格

表格是一个容器对象，其外形与 Browse 窗口相似。与表单集包含表单一样，表格也能包含列。这些列除了包含标头和控件外，每一个列还拥有自己的一组属性、事件和方法程序。

1. 表格设计基本操作

最简单的表格设计方法是直接把表拖曳到表单中。另外，表格还可以调用“表格生成器”来进行设计。调用“表格生成器”的步骤如下：

（1）单击“表单控件”工具栏中“表格”按钮，然后用鼠标在表单上单击拖曳，生成表格。

（2）用鼠标右键单击表格对象，弹出快捷菜单，从中选择“生成器”命令，打开“表格生成器”对话框，如图 10-28 所示。

（3）在“表格生成器”对话框中设置有关选项参数，单击“确定”按钮，完成表格设计。

在“表格生成器”对话框中包括 4 张选项卡，其具有以下作用：

- “表格项”选项卡：指明在表格中显示的字段。
- “样式”选项卡：指定表格样式，如标准型、专业型、账务型等。
- “布局”选项卡：指明各列标题、控件类型和各列列宽。
- “关系”选项卡：设置一对多关系，指明父表的关键字段与子表的相关索引。

2. 表格的常用属性

表格控件的常用属性如表 10-27 所示。

表 10-27 表格控件的常用属性

属性	类别	说明
Caption	标头属性	指定标头对象的标题文本，显示于列顶部
Alignment	标头属性	指定标题文本在对象中显示的对齐方式
ControlSource	列属性	指定在列中显示的数据源，常见的是表中的一个字段
Sparse	列属性	用于确定 CurrentControl 属性是影响列中的活动单元格还是影响所有单元格，默认值是.T.
CurrentControl	列属性	指定列对象中的一个控件，该控件用以显示和接收列中活动单元格的数据。列中非活动单元格的数据将在默认的 TextBox 中显示



图 10-28 “表格生成器”对话框

续表

属性	类别	说明
RecordSouceType	表格属性	指定表格数据源的类型
RecordSource	表格属性	指定表格数据源
ColumnCount	表格属性	指定表格列的数目，默认值为-1
LinkMasker	表格属性	用于指定表格控件中所显示的子记录的父表名称
ChildOrder	表格属性	用于指定为建立一对多的关联关系，子表所要用的索引
RelationalExpr	表格属性	确定基于主表字段的关联表达式

关于 Alignment 属性,在默认方式下，数值型数据右对齐，其他类型数据左对齐。其具体设置值如表 10-28 所示。

表 10-28 Alignment 属性的设置

属性值	说明	属性值	说明
0	居中靠左	5	右上对齐
1	居中靠右	6	中上对齐
2	居中	7	左下对齐
3	自动（默认值）	8	右下对齐
4	左上对齐	9	中下对齐

关于 CurrentControl 属性，在默认情况下，表格中的一个具体列对象包含一个标头对象（Header1）和一个文本框对象（Text1），而 CurrentControl 默认值就是文本框 Text1。用户可以根据需要往列对象中添加所需要的控件，并将 CurrentControl 属性设置为其中的某个控件。

- 在“表单设计器”对话框中，交互式地往表格列中添加控件，步骤如下：
- （1）在“属性”窗口的对象框中选择表格中需要添加控件的列。
 - （2）在“表单控件”工具栏中选择所需控件，然后用鼠标单击表格中需要添加该控件的列，如图 10-29 所示。
- 同样，也可以从表格列中移去控件，步骤如下：
- （1）在“属性”窗口的对象框中选择不需要的控件。
 - （2）单击“表达设计器”对话框标题栏，激活“表单设计器”。
 - （3）按【Delete】键。

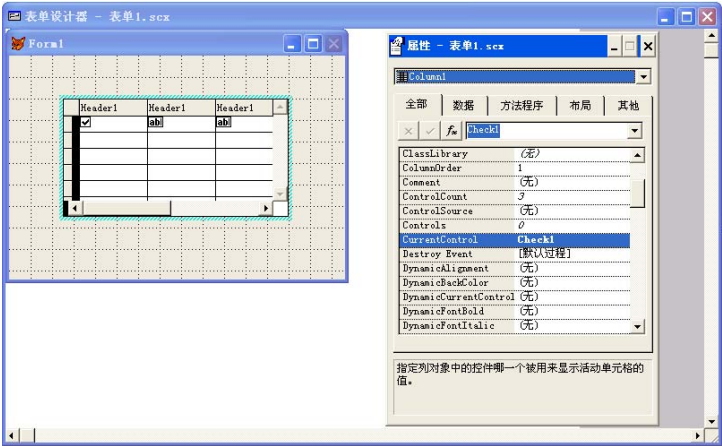


图 10-29 设置 CurrentControl 属性

RecordSouceType 和 RecordSource 属性的取值范围及含义如表 10-29 所示。

表 10-29 RecordSouceType 属性的设置值

属性值	含义	说明
0	表	数据来源于 RecordSource 指定的表，该表能自动打开
1	别名（默认值）	数据来源于 RecordSource 指定别名的表，该表已经打开
2	提示	运行时，由用户根据提示选择表格数据源
3	SQL 语句	数据来源于 RecordSource 指定的 SQL 语句的运行结果
4	查询	数据来源于 RecordSource 指定的查询

ColumnCount 属性的默认值为-1，此时，用户无法对其中的列、标头等进行设置，表格会自动创建足够多的列来显示数据源中的所有字段。如果将 ColumnCount 属性值设置为一个正值，用户就可以对其中的列、标头等进行设置了。

10.7.4 页框

页框是包含页面的容器对象，同时，页面也包含容器，其中又包含其他控件，可以利用它们建立选项卡对话框，其中选项卡就是页面。

向页框中添加控件的步骤如下：

- （1）用鼠标右键单击页框，弹出快捷菜单，从中选择“编辑”命令，然后单击相应页面的标签，使该页面处于激活状态。或者在“属性”窗口的对象框中直接选择相应的页面。
- （2）在“表单控件”工具栏中选择需要的控件，将其添加到页面中，结果如图 10-30 所示。

可以在页框、页面或控件上设置属性，页框的常用属性如表 10-30 所示。

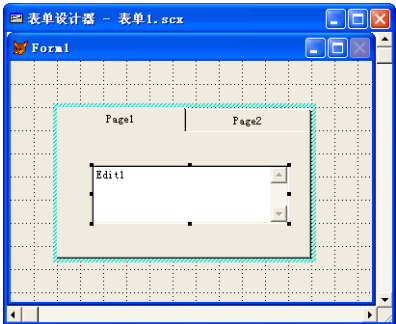


图 10-30 向页框中添加控件

表 10-30 页框的常用属性

属性	说明
Tabs	指定页框中是否显示页面标签栏，默认值为.T.
Pages	该属性是一个数组，用于存取页框中的某个页对象。该属性仅在运行时可用
PageCount	指明一个页框对象中的页面数目
TabStretch	如果页面标题文本太长，则可以通过设置该属性多行显示。0：多行显示。1：单行显示（默认值）
ActivePage	返回页框中活动页的页号，或使页框中的指定页成为活动的

10.8 用户自定义类

可以通过调用类设计器采用可视化的方法创建类。用类设计器创建、定义的类保存在类库文件中，其扩展名为“.vcx”。

10.8.1 使用类设计器创建类

用户自定义类的设计是在“类设计器”中完成的。

1. 调用类设计器

在 Visual FoxPro 中，有以下 3 种方法调用“类设计器”，创建用户自定义类。

- （1）在“项目管理器”对话框中，选择“类”选项卡，单击“新建”按钮，如图 10-31 所示。

(2) 在主菜单“文件”菜单中，选择“新建”命令，打开“新建”对话框，选择“类”文件类型，单击“新建文件”按钮。

(3) 在命令窗口中，输入 CREATE CLASS 命令。
通过以上 3 种方法，都会打开“新建类”对话框，如图 10-32 所示。

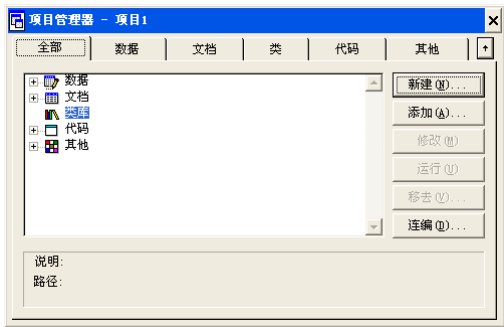


图 10-31 在“项目管理器”中创建类

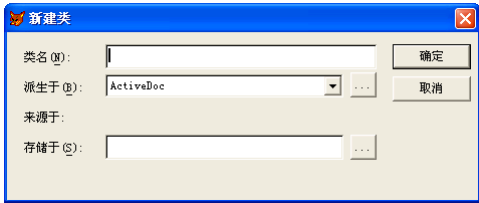


图 10-32 “新建类”对话框

在“新建类”对话框中，在“类名”文本框中输入新创建类的名称，在“派生于”组合框中，选择新创建类派生于哪个父类，在“存储于”文本框中保存新创建类到类库中。

单击“确定”按钮，进入“类设计器”对话框，如图 10-33 所示。

2. 添加属性

可以为所创建的类添加新的属性，具体的步骤如下：

(1) 在主菜单“类”菜单中选择“新建属性”命令，打开“新建属性”对话框。如图 10-34 所示。

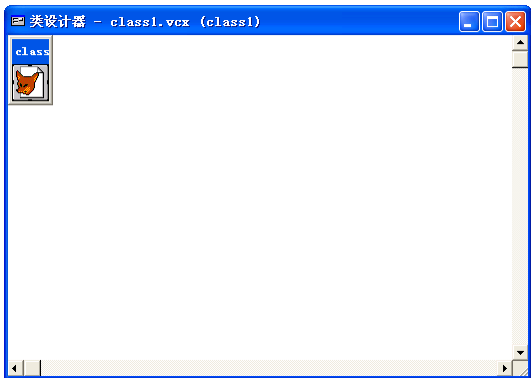


图 10-33 “类设计器”对话框

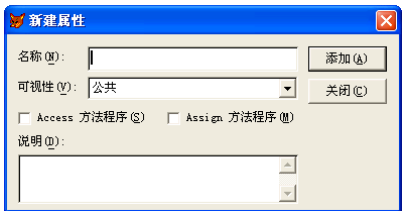


图 10-34 “新建属性”对话框

(2) 在“名称”文本框中，输入新建属性的名称。
(3) 在“可视性”组合框中，选择该属性的可视性。

属性的可视性有 3 种：

- 公共：所谓“公共”属性，是指该属性可以在应用程序的任何地方被访问。
- 保护：所谓“保护”属性，是指该属性只能被在该类中定义的方法和子类中定义的方法所访问。
- 隐藏：所谓“隐藏”属性，是指该属性只能被在该类中定义的方法所访问，即使是子类中定义的方法也不能访问。

(4) 选中“Access”选项，系统会自动在类中添加一个 Access 方法，其名称与相应的 Access 属性名相同，只是多出一个后缀 _Access。Access 方法代码在相应的 Access 属性被查询时自动执行。

(5) 选中“Assign”选项，系统会自动在类中添加一个 Assign 方法，其名称与相应的 Assign 属性名相同，只是多出

一个后缀 Assign。Assign 方法代码在相应的 Access 属性被重新设置时自动执行。

(6) 单击“添加”按钮。

3. 添加方法

可以为所创建的类添加新的方法，具体的步骤如下：

(1) 在主菜单“类”菜单中，选择“新建方法程序”命令，打开“新建方法程序”对话框，如图 10-35 所示。

(2) 在“名称”文本框中，输入新建方法的名称。

(3) 在“可视性”组合框中，选择该方法的可视性。

方法的可视性有 3 种：

- 公共：所谓“公共”方法，是指该方法可以在应用程序的任何地方被访问。
- 保护：所谓“保护”方法，是指该方法只能被在该类中定义的方法和子类中定义的方法所访问。
- 隐藏：所谓“隐藏”方法，是指该方法只能被在该类中定义的方法所访问，即使是子类中定义的方法也不能访问。

(4) 单击“添加”按钮。

通过上述方法，仅仅是在类中声明了一些方法，还需要定义这些方法的代码。定义的方法是：在“属性”窗口中找到该方法并双击，打开“代码”窗口，在其中输入该方法的代码。

方法代码可以访问类中的属性和方法，被访问的属性和方法名字之前加上 This 关键字。

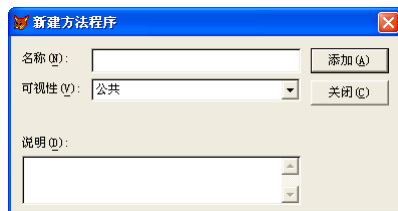


图 10-35 “新建方法程序”对话框

4. 修改类定义

对已经存在的类，可以通过“类设计器”修改它。

修改类定义有以下 3 种方法：

(1) 在“项目管理器”对话框中，选择“类”选项卡，选中需要修改的类，然后单击“修改”按钮。

(2) 在主菜单“文件”菜单中选择“打开”命令，弹出“打开”对话框，从中选择“可视类库”文件类型，选中需要修改的类文件，单击“打开”按钮。

(3) 在命令窗口中输入命令：MODIFY CLASS <类名> OF <类库名>。

10.8.2 类库管理

类库文件扩展名为“.vcx”，对应的备注文件的扩展名为“.vct”。类库中包含了一系列类的定义，使用命令或者利用可视化方式对类库中的类定义进行维护和修改。

1. 创建类库

在使用“类设计器”创建类时，需要指定存放该类的类库，如果该类库不存在，则系统将自动创建该类库。

另外，还可以在命令窗口中输入命令：CREATE CLASSLIB <类库名>。

该命令用于创建一个新的类库。

2. 复制类

利用 CREATE CLASSLIB 所创建的新的类库是一个空的类库，可以通过命令 CREATE CLASS 在类库中创建新的类。

另外，还可以在命令窗口中输入命令：ADD CLASS <类名> [OF <类库名 1>] TO <类库名 2> [OVERWRITE]

该命令用于将一个类库中的某个类复制到另一个类库中。OVERWRITE 表示将原有的同名类覆盖。

使用“项目管理器”实现类的复制的步骤如下：

(1) 在“项目管理器”对话框中，选择“类”选项卡。

(2) 选择“类库”选项，完全将其展开。

(3) 将类从源类库中拖曳到目标类库中。

3. 删除类

使用“项目管理器”实现类的删除的步骤如下：

- (1) 在“项目管理器”对话框中，选择“类”选项卡。
- (2) 选择“类库”选项，完全将其展开。
- (3) 选中想要删除的类，然后单击“移去”按钮。

可以在命令窗口中输入命令：**REMOVE CLASS <类名> OF <类库名>**。

该命令用于将一个已经存在的类从类库中删除。

4. 重命名类

可以在命令窗口中输入命令：**RENAME CLASS <类名 1> OF <类库名> TO <类名 2>**。

该命令用于将一个类的名字进行重命名。

5. 打开类库

当使用基于类创建对象时，需要首先打开类所在的类库文件。可以在命令窗口中输入命令：**SET CLASSLIB TO <类库名> [ADDITIVE][ALIAS <别名>]**。

该命令用于打开某个类库文件；**ADDITIVE** 表示在打开新的类库时，不会关闭当前处于打开状态的其他类库，否则，在打开新的类库时，系统会自动关闭当前处于打开状态的其他类库；可以使用 **ALIAS <别名>**，为打开的类库指定一个别名；如果使用 **SET CLASSLIB TO** 而没有指定<类库名>，则关闭当前所有打开的类库文件。

另外，还可以使用命令：**RELEASE CLASSLIB <类库名>[ALIAS <别名>]**。

该命令用于关闭当前处于打开状态的类库文件。

10.8.3 在创建表单时使用用户自定义类

在创建表单时，如果需要使用用户自定义的类，首先需要注册用户自定义类所在的类库，然后将用户自定义类显示在“表单控件”工具栏中。

1. 注册类库

在 Visual FoxPro 中，注册类库的步骤如下：

- (1) 在主菜单“工具”菜单中选择“选项”命令，弹出“选项”对话框，如图 10-36 所示。

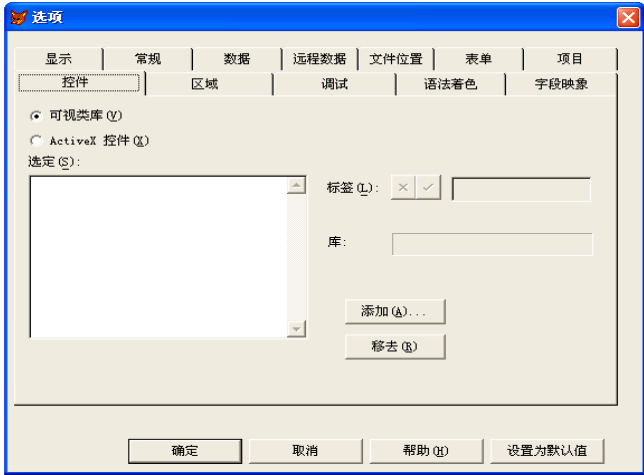


图 10-36 “选项”对话框

- (2) 选择“控件”选项卡，选中“可视类库”选项。

(3) 单击“添加”按钮，弹出“打开”对话框，从中选择需要注册的类型库文件，然后单击“打开”按钮，则将该类库进行注册。

2. 显示用户自定义类

在 Visual FoxPro 中，在“表单控件”工具栏中显示用户自定义类的步骤如下：

(1) 在“表单设计器”对话框中，单击“表单控件”工具栏中的“查看类”按钮，在弹出的快捷菜单中选择自定义类所在的类库，如图 10-37 所示。

(2) 如果没有所需要的类库，选择“添加”命令，弹出“打开”对话框，在其中选择所需要的类库文件，单击“确定”按钮。

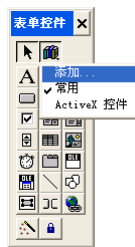


图 10-37 显示用户自定义类

10.9 本章习题

10.9.1 选择题

- 下面关于类、对象、属性和方法的叙述中，错误的是（ ）。
 - 类是对一类相似对象的描述，这些对象具有相同种类的属性和方法
 - 属性用于描述对象的状态，方法用于表示对象的行为
 - 基于同一个类产生的两个对象可以分别设置自己的属性值
 - 通过执行不同对象的同名方法，其结果必然是相同的
- 让控件获得焦点，使其成为活动对象的方法是（ ）。
 - Show
 - Release
 - SetFocus
 - GotFocus
- 下面属于表单方法名（非事件名）的是（ ）。
 - Init
 - Release
 - Destroy
 - Caption
- 假设某个表单中有一个命令按钮 CmdClose，为了实现当用户单击此按钮时能够关闭该表单的功能，应在该按钮的 Click 事件中写入语句（ ）。
 - ThisForm.close
 - ThisForm.Erase
 - ThisForm.Release
 - ThisForm.Return
- 已知表单名为 Form1，表单中有两个命令按钮（Command1 和 Command2）、两个标签、两个文本框（Text1 和 Text2）。如果在运行表单时，要使表单的标题显示“登录窗口”，则可以在 Form1 的 Load 事件中加入语句（ ）。
 - ThisForm.Caption="登录窗口"
 - Form1.Caption="登录窗口"
 - ThisForm.Name="登录窗口"
 - Form1.Name="登录窗口"
- 在 Visual FoxPro 中，假设表单上有一选项组：☉男☉女，初始时该选项组的 Value 属性值为 1。若选项按钮“女”被选中，该选项组的 Value 属性值是（ ）。
 - 1
 - 2
 - “女”
 - “男”
- 以下叙述与表单数据环境有关，其中正确的是（ ）。
 - 当表单运行时，数据环境中的表处于只读状态，只能显示不能修改
 - 当表单关闭时，不能自动关闭数据环境中的表
 - 当表单运行时，自动打开数据环境中的表
 - 当表单运行时，与数据环境中的表无关
- 假定一个表单里有一个文本框 Text1 和一个命令按钮组 CommandGroup1。命令按钮组是一个容器对象，其中包含

Command1 和 Command2 两个命令按钮。如果要在 Command1 命令按钮的某个方法中访问文本框的 Value 属性值，正确的表达式是（ ）。

- A. ThisForm.Parent.Text1.Value B. This.Parent.Parent.Text1.Value
C. Parent.Parent.Text1.Value D. This.Parent.Text1.Value
9. 在表单中为表格控件指定数据源的属性是（ ）。
- A. DataSource B. DataFrom C. RecordSource D. RecordFrom
10. 以下关于表单数据环境叙述错误的是（ ）。
- A. 可以向表单数据环境设计器中添加表或视图
B. 可以从表单数据环境设计器中移出表或视图
C. 可以在表单数据环境设计器中设置表之间的联系
D. 不可以在表单数据环境设计器中设置表之间的联系
11. 如果想在运行表单时，向 Text2 中输入字符，回显字符显示的是“*”号，则可以在 Form1 的 Init 事件中加入语句（ ）。
- A. Form1.Text2.Passwordchar="*" B. Form1.Text2.Password="*" C. ThisForm.Text2.Password="*" D. ThisForm.Text2.Passwordchar="*"
12. 如果文本框的 InputMask 属性值是#99999，允许在文本框中输入的是（ ）。
- A. +12345 B. abc123 C. \$12345 D. abcdef
13. 下列表单的（ ）属性设置为真时，表单运行时将自动居中。
- A. AutoCenter B. AlwaysOnTop C. ShowCenter D. FormCenter
14. 下面关于命令 DO FORM XX NAME YY LINKED 的陈述中，正确的是（ ）。
- A. 产生表单对象引用变量 XX，在释放变量 XX 时自动关闭表单
B. 产生表单对象引用变量 XX，在释放变量 XX 时并不关闭表单
C. 产生表单对象引用变量 YY，在释放变量 YY 时自动关闭表单
D. 产生表单对象引用变量 YY，在释放变量 YY 时并不关闭表单
15. 表格控件的数据源可以是（ ）。
- A. 视图 B. 表 C. SQL SELECT 语句 D. 以上 3 种都可以
16. 以下所列各项属于命令按钮事件的是（ ）。
- A. Parent B. This C. ThisForm D. Click
17. 下面关于表单若干常用事件的描述中，正确的是（ ）。
- A. 当释放表单时，Unload 事件在 Destroy 事件之前引发
B. 当运行表单时，Init 事件在 Load 事件之前引发
C. 单击表单的标题栏，引发表单的 Click 事件
D. 上面的说法都不对
18. 在面向对象的程序设计中，程序运行的最基本实体是（ ）。
- A. 对象 B. 事件 C. 属性 D. 方法
19. 在 Visual FoxPro 中，表单是（ ）。
- A. 一个窗口界面 B. 一个表中各个记录的清单
C. 数据库中各个表的清单 D. 数据库查询的列表
20. 下列对控件类和容器类的说法正确的是（ ）。
- A. 可以对控件类和容器类对象中的组件单独进行修改或操作

- B. 控件类一般作为容器类的控件
C. 控件类的封装性比容器类更加严密, 灵活性更好
D. 控件类和容器类只能用来进行一种相关的控制
21. 下列关于在子类的方法程序中继承父类的方法程序的叙述错误的是 ()。
A. 在子类中重新定义父类的方法和事件时, 就用新定义的方法来取代父类中原有的代码
B. 用函数 DODEFAULT() 来继承父类的方法和事件
C. 可使用<父类名>.<方法>的命令格式继承父类的方法和事件
D. 可使用<父类名>.:<方法>的命令格式继承父类的方法和事件
22. 下列关于基类的说法错误的是 ()。
A. Visual FoxPro 提供的基础类即为基类
B. Visual FoxPro 基类存放在指定的类库中
C. Visual FoxPro 基类是系统本身自带的
D. 可以基于类生成所需的对象, 也可以扩展基类创建自己的类
23. 下列关于“类”的叙述中, 错误的是 ()。
A. 类是对象的集合, 而对象是类的实例
B. 一个类包含了相似对象的特征和行为方法
C. 类并不实行任何行为操作, 它仅仅表明该怎样做
D. 类可以按其定义的属性、事件和方法进行实际的行为操作
24. 在 Visual FoxPro 中, 关键字 Parent 的含义是指 ()。
A. 当前对象
B. 当前对象的直接容器对象
C. 当前对象所在的表单
D. 当前对象所在的表单集
25. 在 Visual FoxPro 中, 基类的最小事件集包含的事件是 ()。
A. Init, Error, Desorty
B. Init, Load, Unload
C. Load, Error, Unload
D. Load, Destroy, Click
26. 在一般情况下, 当运行表单时, 如果要重新绘制表单或控件, 将调用表单对象的 ()。
A. Release 方法
B. Refresh 方法
C. Show 方法
D. Hide 方法
27. 在 Visual FoxPro 中, 当创建一个对象时, 将产生 ()。
A. Init 事件
B. Load 事件
C. Error 事件
D. Destroy 事件
28. 下列关于“表单”窗口的说法错误的是 ()。
A. “表单”窗口包含在“表单设计器”窗口中
B. “表单”窗口可以在主窗口内任意移动
C. 可以在“表单”窗口中可视化地添加和修改控件
D. “表单”窗口只能在表单设计器中移动
29. 要运行表单文件 form1, 下列命令正确的是 ()。
A. DO FORM1.SCX
B. DO FORM FORM1
C. RUN FORM1.SCX
D. RUN FORM FORM1
30. 数据环境泛指定义表单或表单集时使用的 ()。
A. 数据
B. 数据库
C. 数据源
D. 数据项
31. 如果要为控件设置焦点, 则下列属性值必须为.T.的是 ()
A. Enabled 和 Default
B. Enabled 和 Visible
C. Default 和 Cancel
D. Visible 和 Default
32. 在 Visual FoxPro 中调用表单 mf1 的正确命令是 ()。

- A. DO mf1 B. DO FROM mf1 C. DO FORM mf1 D. RUN mf1
33. 在 Visual FoxPro 中, 释放表单时会引发的事件是 ()。
- A. Unload 事件 B. Init 事件 C. Load 事件 D. Release 事件
34. 在 Visual FoxPro 中, Unload 事件的触发时机是 ()。
- A. 释放表单 B. 打开表单 C. 创建表单 D. 运行表单
35. 假设在表单设计器环境下, 表单中有一个文本框且已经被选定为当前对象。现在从属性窗口中选择 Value 属性, 然后在设置框中输入: =^{^2001-9-10}-{^2001-8-20}。请问以上操作后, 文本框 Value 属性值的数据类型为 ()。
- A. 日期型 B. 数值型 C. 字符型 D. 以上操作出错
36. 在表单设计中, 经常会用到一些特定的关键字、属性和事件。下列各项中属于属性的是 ()。
- A. This B. ThisForm C. Caption D. Click
37. 下列表单的 () 属性设置为真时, 表单运行时将自动居中。
- A. AutoCenter B. AlwaysOnTop C. ShowCenter D. FormCenter
38. 表单里有一个选项按钮组, 包含两个选项按钮 Option1 和 Option2。假设 Option2 没有设置 Click 事件代码, 而 Option1 及选项按钮和表单都设置了 Click 事件代码。那么当表单运行时, 如果用户单击 Option2, 系统将 ()。
- A. 执行表单的 Click 事件代码 B. 执行选项按钮组的 Click 事件代码
- C. 执行 Option1 的 Click 事件代码 D. 不会有反应

10.9.2 填空题

1. 在 Visual FoxPro 中, 表单的 Load 事件发生在 Init 事件之_____。
2. 在 Visual FoxPro 中, 如果要想改变表单上表格对象中当前所显示的列数, 应该设置表格的_____属性值。
3. 在 Visual FoxPro 中释放和关闭表单的方法是_____。
4. 在 Visual FoxPro 的表单设计中, 为表格控件指定数据源的属性是_____。
5. 在 Visual FoxPro 中为表单指定表题的属性是_____。
6. 为使表单运行时在主窗口中居中显示, 应设置表单的 AutoCenter 属性值为_____。
7. 在 Visual FoxPro 中, 在运行表单时最先引发的表单事件是_____事件。
8. 在 Visual FoxPro 表单中, 当用户使用鼠标单击命令按钮时, 会触发命令按钮的_____事件。
9. 在 Visual FoxPro 中, 假设表单上有一选项组: ☐男 ☐女, 该选项组的 Value 属性值赋为 0。当其中的第一个选项按钮“男”被选中时, 该选项组的 Value 属性值为_____。

第11章

菜单设计与应用

在数据库应用系统中，菜单起着组织和协调其他对象的关键作用，一个良好的菜单系统会给用户带来一个十分友好的操作界面，并带来操作上的便利。常见的菜单有两种：下拉式菜单与快捷菜单。

本章主要介绍 Visual FoxPro 的系统菜单，以及在开发数据库应用系统过程中菜单的设计及应用。

11.1 Visual FoxPro 系统菜单

在 Visual FoxPro 中，系统的大部分功能都是通过系统菜单实现的，因此在学习设计菜单之前，应该首先了解 Visual FoxPro 系统菜单的结构、特点和功能。

11.1.1 菜单结构

在 Visual FoxPro 中，系统提供了两种类型的菜单：条形菜单和弹出式菜单。无论是哪一种类型的菜单，都具有一个内部名字和一组菜单选项，每一个菜单选项都具有一个名称（标题）和内部名字（内部序号）。菜单项名称用来供用户识别，因此显示在屏幕上；而菜单和菜单项的内部名字或内部序号则是用来在代码中引用。

1. 菜单选项的设置

每一个菜单项都可以设置一个热键和一个快捷键。

（1）热键。又叫访问键，通常是利用【Alt】键和一个字符键组合起来完成一项特定任务。当菜单被激活后，可以通过热键快速选择该菜单项。菜单项名称后面加下画线的字符就是该菜单项的热键。

（2）快捷键。通常是利用【Ctrl】键和一个字符键组合起来完成一项特定任务，无论菜单是否被激活，都可以直接通过快捷键快速选择该菜单项。

2. 菜单选项的功能

当选择菜单中某个菜单项时，系统都会激发一个动作，可能是激活另一个菜单，也可能是执行一个过程，还可能是执行一个命令。因此，菜单项设计的一个重要任务，就是设置该菜单项具体执行哪一种功能。

3. 菜单类别

常见的菜单有两种：下拉式菜单与快捷菜单。下拉式菜单主要是用来列出一个应用程序所具有的所有功能，而快捷菜单主要是用来列出其所从属的界面对象的一些功能。

Visual FoxPro 的主菜单就是一个下拉式菜单，由一个条形菜单和一组弹出式菜单组成，当选择一个条形菜单的选项时，就会激活相应的弹出式菜单。Visual FoxPro 的快捷菜单则是由一个或一组具有上下级关系的弹出式菜单组成。

11.1.2 系统菜单

Visual FoxPro 的系统主菜单是一个典型的菜单系统，其主菜单是一个条形菜单。

1. 主菜单

Visual FoxPro 的系统主菜单的内部名字为 _MSYSMENU，其选项的名称及内部名字如表 11-1 所示。

表 11-1 系统主菜单选项名称及内部名字

选项名称	内部名字	选项名称	内部名字
文件	_MSM_FILE	程序	_MSM_PROG
编辑	_MSM_EDIT	窗口	_MSM_WINDOW
显示	_MSM_VIEW	帮助	_MSM_SYSTM
工具	_MSM_TOOLS		

2. 弹出式菜单

选择条形菜单中的每一个菜单项都会激活一个弹出式菜单，各弹出式菜单的内部名字如表 11-2 所示。

表 11-2 弹出式菜单的内部名字

弹出式菜单	内部名字	选项名称	内部名字
“文件”菜单	_MFILE	“程序”菜单	_MPROG
“编辑”菜单	_MEDIT	“窗口”菜单	_MWINDOW
“显示”菜单	_MVIEW	“帮助”菜单	_MSYSTM
“工具”菜单	_MTOOLS		

3. “编辑”菜单

“编辑”菜单中常用选项的名称和内部名字如表 11-3 所示。

表 11-3 弹出式菜单的内部名字

弹出式菜单	内部名字	选项名称	内部名字
撤销	_MED_UNDO	清除	_MED_CLEAR
重做	_MED_REDO	全部选定	_MV_SLCTA
剪切	_MED_CUT	查找	_MED_FIND
复制	_MED_COPY	替换	_MED_REPL
粘贴	_MED_PASTE		

4. 系统菜单配置命令

在 Visual FoxPro 中，可以通过 SET SYSMENU 命令允许或者禁止在程序执行时访问系统菜单，同时也可以重新配置系统菜单。

具体格式如下：

```
SET STSNENU ON|OFF|AUTOMATIC
| TO[<弹出式菜单名表>]
| TO[<条形菜单项名表>]
| TO[DEFAULT] | SAVE|NOSAVE
```

功能说明如表 11-4 所示。

表 11-4 SET SYSMENU 命令功能说明

命令选项	功能
ON	允许程序执行时访问系统菜单
OFF	禁止程序执行时访问系统菜单
AUTOMATIC	可以使系统菜单显示出来，可以访问系统菜单
TO [<弹出式菜单名表>]	重新配置系统菜单，以内部名字列出可用的弹出式菜单
TO [<条形菜单项名表>]	重新配置系统菜单，以内部名字列出可用的子菜单
TO [DEFAULT]	将系统菜单恢复为默认设置
SAVE	将当前系统菜单设置设为默认设置
NOSAVE	将当前系统菜单默认设置恢复为标准设置

不带任何参数的 SET SYSMENU TO 命令将屏蔽系统菜单，使系统菜单不可用。

11.2 下拉式菜单设计

Visual FoxPro 提供了创建应用系统菜单的工具，用户利用菜单设计器可以设计出与 Visual FoxPro 系统菜单媲美的面向具体问题的应用系统菜单。

本节主要介绍下拉式菜单的设计方法。

11.2.1 菜单设计的基本过程

用菜单设计器设计下拉式菜单的基本过程如图 11-1 所示。

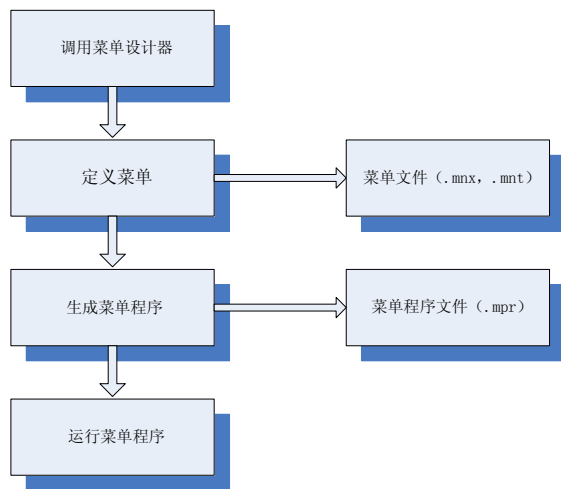


图 11-1 菜单设计的基本过程

1. 调用菜单设计器

打开“菜单设计器”对话框有 3 种方法：

(1) 在主菜单“文件”菜单中选择“新建”命令，弹出“新建”对话框，如图 11-2 所示。在“新建”对话框中，选择“菜单”选项，然后单击“新建文件”按钮，弹出“新建菜单”窗口，如图 11-3 所示，单击“菜单”按钮，进入“菜单设计器”对话框，如图 11-4 所示。

(2) 在主菜单“文件”菜单中选择“打开”命令，弹出“打开”对话框，选择一个菜单文件 (.mnx 文件)，从而在“菜

单设计器”中打开一个已有菜单。

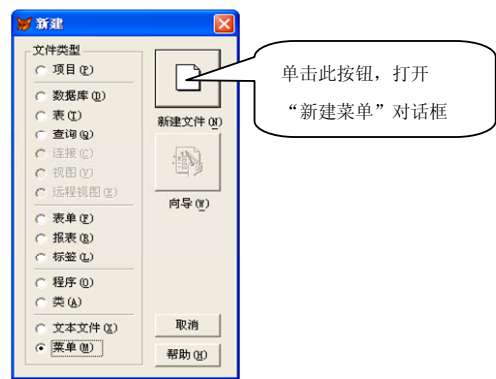


图 11-2 “新建”对话框

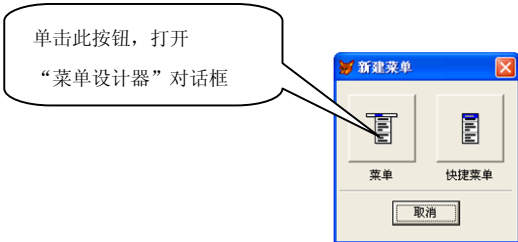


图 11-3 “新建菜单”对话框

(3) 在命令窗口中输入“打开菜单设计器”命令，如图 11-5 所示。

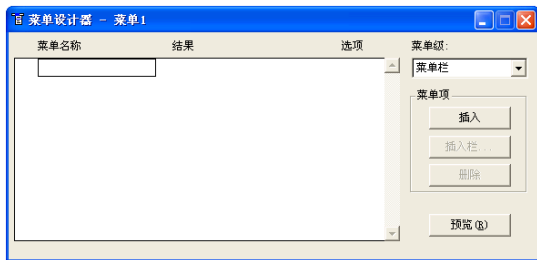


图 11-4 “菜单设计器”对话框



图 11-5 在命令窗口中输入“打开菜单设计器”命令

2. 定义菜单

在“菜单设计器”对话框中定义菜单，指定菜单的各项内容，如菜单项的名称、快捷键等。

定义完菜单的各项内容后，选择主菜单“文件”菜单中“保存”命令，或者利用【Ctrl+W】组合键，将所定义的菜单保存为.mnx 文件。

3. 生成菜单程序

菜单文件.mnx 必须生成成为菜单程序文件.mpr 才能够被运行。在“菜单设计器”对话框被激活的环境下，选择主菜单“菜单”中“生成”命令，弹出“生成菜单”对话框，如图 11-6 所示。在“输出文件”文本框中可以指定菜单程序文件的名称和存放路径，然后单击“生成”按钮，则由菜单文件.mnx 生成了可执行的菜单程序文件.mpr。

4. 运行菜单程序

在命令窗口中输入 DO 命令运行菜单程序文件，其中菜单程序文件的扩展名.mpr 不可省略，如图 11-7 所示。

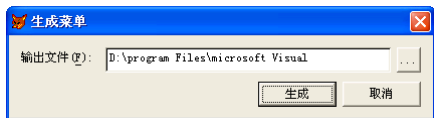


图 11-6 “生成菜单”对话框

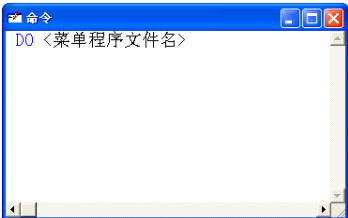


图 11-7 在命令窗口中输入“运行菜单程序文件”命令

11.2.2 定义菜单

在“菜单设计器”对话框中定义菜单，指定菜单的各项内容。

1. “菜单设计器”窗口

“菜单设计器”窗口每页显示和定义一个菜单，既可以设计条形菜单，也可以设计弹出式菜单。在“菜单设计器”窗口中左侧是一个列表框，其中的每一行定义一个当前菜单的菜单项，包括“菜单名称”、“结果”和“选项”3列内容。

1) 菜单名称

指定菜单的名称（标题），同时还可以设置菜单项的访问键，方法是在要作为访问键的字符前加“\<”。如图 11-8 所示，第一个菜单项的访问键为“P”。

同时，根据各菜单项功能的相似性或相近性，可以对菜单项进行分组，方法是在相应位置的“菜单名称”处输入“\-”。如图 11-9 所示，在菜单项中间有一条水平的分组线。

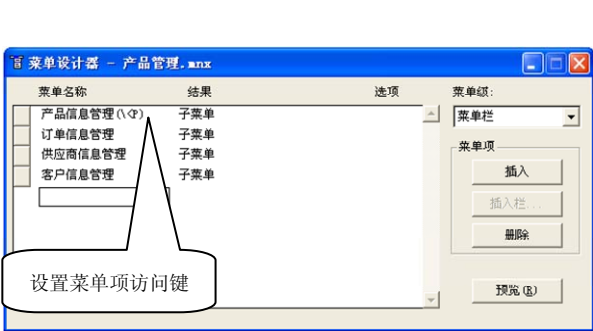


图 11-8 设置“菜单名称”和“菜单项访问键”

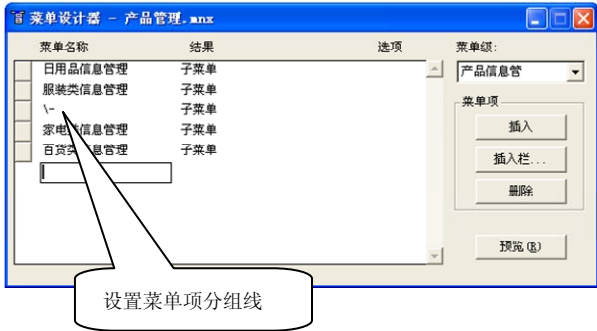


图 11-9 设置“菜单项分组线”

2) 结果

用于指定当用户选择该菜单项时的动作。单击对应的组合列表框，即可看到有 4 个选项，如图 11-10 所示。

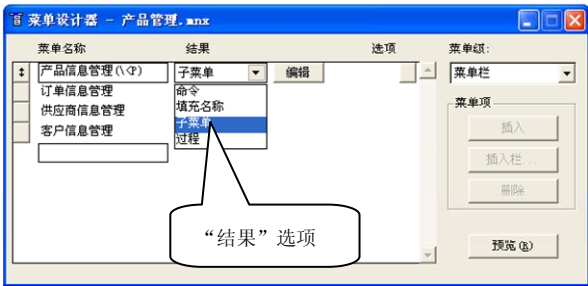


图 11-10 “结果”选项

这 4 个选项的具体含义如表 11-5 所示。

表 11-5 “结果”选项含义

选项	含义
命令	可在其右侧文本框中输入一条命令，当选择该菜单项时，将执行该条命令
填充名称	可在其右侧文本框中输入菜单项的内部名字或序号
子菜单	默认选项，可单击其右侧“创建”或“编辑”按钮，可以创建或编辑修改其下级子菜单
过程	可单击其右侧“创建”或“编辑”按钮，可以创建或编辑修改过程代码，当选择该菜单项时，将执行该过程代码

3) 选项

每个菜单项的“选项”都有一个无符号按钮，单击该按钮，弹出“提示选项”对话框，供用户定义菜单项的其他属性，如图 11-11 所示。

在“提示选项”对话框中，可以定义快捷键、跳过条件（即本菜单不能使用的条件）等属性。当定义过属性后，“选项”的无符号按钮上就会出现“√”符号，如图 11-12 所示。

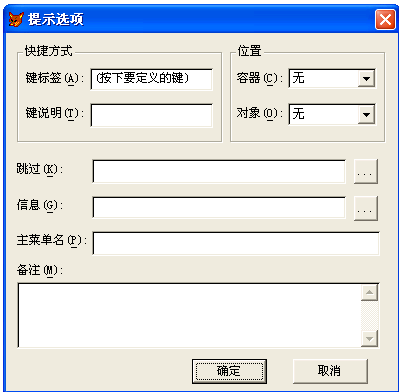


图 11-11 “提示选项”对话框

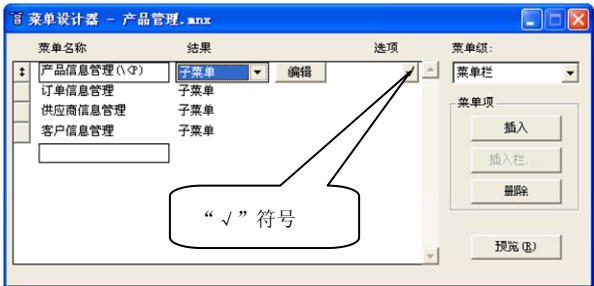


图 11-12 “√”符号

在“提示选项”对话框中，各个属性的具体含义如表 11-6 所示。

表 11-6 “提示选项”对话框中各个属性的含义

属性	含义
快捷方式	用来指定该菜单项的快捷键，方法是：在键盘上直接按下想要设置的快捷键，则“键标签”文本框中会自动出现相应的内容
跳过	用来设置菜单项的跳过条件，方法是：通过一个表达式的值来判定该菜单项是否可选，不可选的菜单项以灰色显示
信息	用来设置菜单项的说明信息，方法是：通过一个字符串或表达式，当鼠标指针指向该菜单项时，显示其内容或表达式的值
主菜单名	用来指定菜单项的内部名字或序号，当“结果”选项为“填充名称”时，该属性不可用

此外，在“菜单设计器”对话框中，各个按钮的具体作用如表 11-7 所示。

表 11-7 “菜单设计器”对话框中各个按钮的功能

按钮	功能
插入	可以在当前菜单项之前插入一个新的菜单项行
插入栏	可以在当前菜单项之前插入一个 Visual FoxPro 系统菜单命令，该按钮仅在定义弹出式菜单时有效
删除	可以删除当前菜单项行
预览	可以预览菜单设计效果
移动	拖动位于菜单项左侧的移动按钮，可以移动对应菜单项的位置

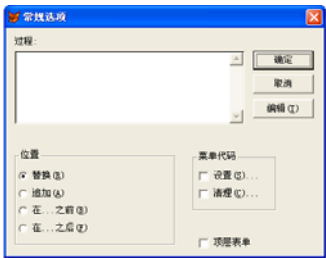
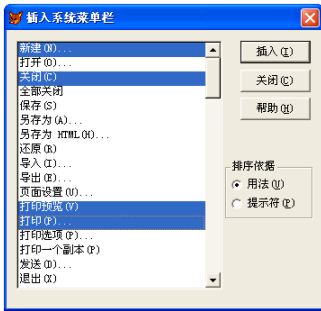
其中，单击“插入栏”按钮，弹出“插入系统菜单栏”对话框，如图 11-13 所示，在其中选择所需的一个或多个菜单命令，最后单击“插入”按钮，完成系统菜单栏的插入工作。

2. “显示”菜单

在激活“菜单设计器”后，在主菜单“显示”菜单中有两个菜单项：“常规选项”和“菜单选项”，单击它们，将弹出各自相应的对话框对菜单进行设置。

1) “常规选项”对话框

选择“常规选项”命令，弹出“常规选项”对话框，可以设置整个下拉菜单的总体属性，如图 11-14 所示。



在“常规选项”对话框中，各个属性的具体含义如表 11-8 所示。

表 11-8 “常规选项”对话框中各个属性的含义

属性	含义
过程	用来为条形菜单中的某个没有定义子菜单的菜单项指定一个过程代码，当选择此菜单项时，执行该过程代码；单击“编辑”按钮，则可以打开“过程代码编辑”对话框
位置	用来指定所定义的下拉式菜单与当前系统的关系，具体关系及含义如表 11-9 所示
菜单代码	选择“设置”选项，弹出“设置代码编辑”对话框，该代码放置在菜单程序文件中菜单定义代码前面，在菜单产生之前执行；选择“清理”选项，弹出“清理代码编辑”对话框，该代码放置在菜单程序文件中菜单定义代码后面，在菜单显示之后执行
顶层表单	用来将所定义的下拉式菜单添加到一个顶层表单中，否则，将该菜单作为一个定制的系统菜单

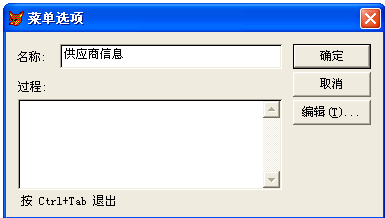
表 11-9 “位置”选项的含义

选项	含义
替换	用定义的菜单内容替换当前系统菜单原有的内容
追加	将定义的菜单内容添加到当前系统菜单原有的内容的后面
在……之前	将定义的菜单内容插到当前系统菜单某个弹出式菜单之前
在……之后	将定义的菜单内容插到当前系统菜单某个弹出式菜单之后

其中，选中“在……之前”或“在……之后”单选按钮时，其右侧会出现一个下拉列表框，可以选择当前系统菜单中一个弹出式菜单，如图 11-15 所示。

2) “菜单选项”对话框

选择“菜单选项”命令，弹出“菜单选项”对话框，可以为当前弹出式菜单的菜单项或所有的弹出式菜单的菜单项定义过程代码，如图 11-16 所示。



在“菜单选项”对话框中，在“名称”文本框中定义当前弹出式菜单的内部名字，在“过程”编辑框中或者单击“编辑”按钮，在弹出的“代码编辑”窗口中输入定义的过程代码，如果当前弹出式菜单的某个菜单项没有指定具体动作时，则执行该过程代码。

11.2.3 为顶层表单添加菜单

将菜单添加到顶层表单中，菜单才能脱离 Visual FoxPro 单独显示，而不是固定地显示在 Visual FoxPro 系统菜单栏中，才会随着表单的打开而打开，随着表单的关闭而清除。

- 为顶层表单添加菜单的方法如下：
- (1) 打开“菜单设计器”对话框，在菜单设计时，选择主菜单“显示”菜单中“常规选项”命令，弹出“常规选项”对话框，选中“顶层表单”复选框，如图 11-17 所示。
 - (2) 将表单的 ShowWindow 属性值设置为“2：作为顶层表单”，将其值为顶层表单。
 - (3) 在表单的 Init 事件代码中，添加调用菜单程序文件命令：DO <文件名> WITH This[, "<菜单名>"]。
 - (4) 在表单的 Destroy 事件代码中，添加清除菜单命令：RELEASE MENU <文件名> [EXTENDED]。
 - (5) 保存表单修改，并运行表单。
- 为顶层表单添加菜单的结果如图 11-18 所示。

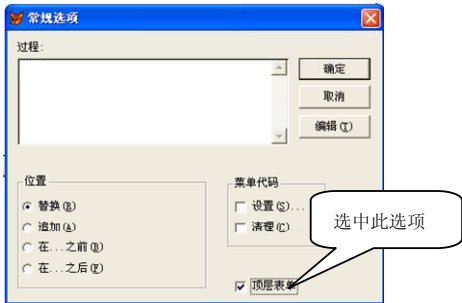


图 11-17 选中“顶层表单”

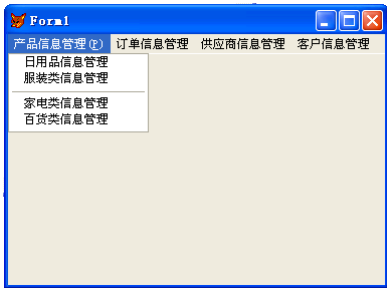


图 11-18 菜单在顶层表单中运行结果

11.3 快捷菜单设计

Visual FoxPro 中的快捷菜单一般是一个弹出式菜单，是由一个或一组具有上下级关系的弹出式菜单组成的。设计快捷菜单的方法如下：

- (1) 在主菜单“文件”菜单中选择“新建”命令，弹出“新建”对话框，在其中选择“菜单”选项，然后单击“新建文件”按钮，弹出“新建菜单”对话框，单击“快捷菜单”按钮，如图 11-19 所示，弹出“快捷菜单设计器”对话框，如图 11-20 所示。

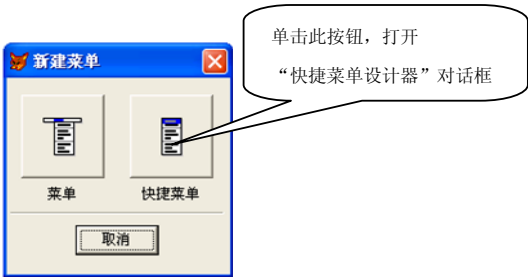


图 11-19 “新建菜单”对话框

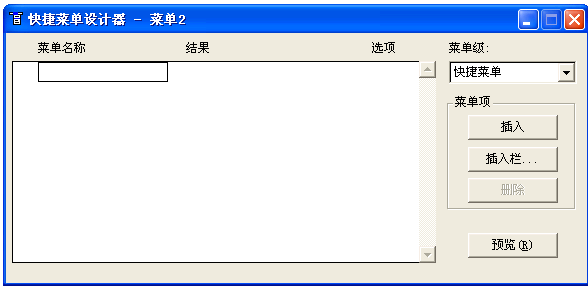


图 11-20 “快捷菜单设计器”对话框

(2) 与设计下拉式菜单方法类似, 在“快捷菜单设计器”对话框中设计快捷菜单。

(3) 在设计快捷菜单过程中, 选择主菜单“显示”菜单中“常规选项”命令, 弹出“常规选项”对话框, 在“菜单代码”选项中选中“设置”复选框, 弹出“设置代码设计器”窗口, 如图 11-21 所示。

(4) 在“设置代码设计器”窗口输入一条命令 `PARAMETERS mfRef`, 定义一个用来接收当前对象引用的参数。

(5) 类似的, 在“常规选项”对话框中的“菜单代码”选项组中选中“清理”复选框, 弹出“清理代码设计器”窗口, 如图 11-22 所示。

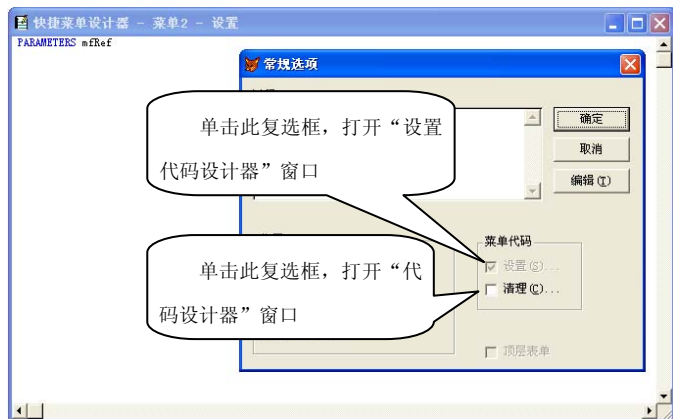


图 11-21 “设置代码设计器”窗口



图 11-22 “清理代码设计器”窗口

(6) 在“清理代码设计器”窗口输入一条命令 `RELEASE POPUPS <快捷菜单名> [EXTENDED]`, 使得在选择、执行菜单命令后能及时清除菜单, 释放其所占用的内存空间。

(7) 与设计下拉式菜单方法类似, 对设计好的快捷菜单生成可执行的菜单程序文件。

(8) 在表单设计器中, 选定需要添加快捷菜单的对象, 在其 `RightClick` 事件中添加调用快捷菜单程序文件的命令:

`DO <快捷菜单程序文件名>`

快捷菜单的运行结果如图 11-23 所示。

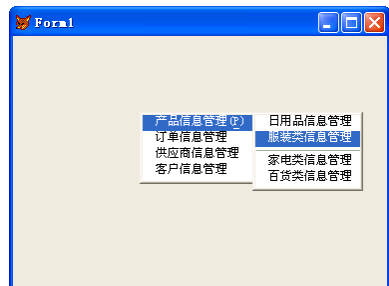


图 11-23 快捷菜单运行结果

11.4 本章习题

11.4.1 选择题

1. 以下是与设置系统菜单有关的命令, 其中错误的是 ()。
 - A. SET SYSMENU DEFAULT
 - B. SET SYSMENU TO DEFAULT
 - C. SET SYSMENU NOSAVE
 - D. SET SYSMENU SAVE
2. 在菜单设计中, 可以在定义菜单名称时为菜单项指定一个访问键。规定了菜单项的访问键为“x”的菜单名称定义是 ()。
 - A. 综合查询\<(x)
 - B. 综合查询/<(x)
 - C. 综合查询(\<x)
 - D. 综合查询(/<x)
3. 在 Visual FoxPro 中, 要运行菜单文件 `menu1.mpr`, 可以使用命令 ()。
 - A. DO menu1
 - B. DO menu1.mpr
 - C. DO MENU menu1
 - D. RUN menu1
4. 在 Visual FoxPro 中, 扩展名为 `.mnx` 的文件是 ()。

- A. 备注文件 B. 项目文件 C. 表单文件 D. 菜单文件
5. 如果菜单项名称为“统计”，热键是 T，则在菜单名称一栏应输入（ ）。
- A. 统计（\<T） B. 统计（Ctrl+T） C. 统计（Alt_T） D. 统计（T）
6. 为表单建立了快捷菜单 mymenu，调用快捷菜单的命令代码 DO mymenu.mpr WITH THIS 应该放在表单的哪个代码中？（ ）
- A. Destroy 事件 B. Init 事件 C. Load 事件 D. RightClick 事件
7. 下列关于快捷菜单的说法，正确的是（ ）。
- A. 快捷菜单中只有条形菜单 B. 快捷菜单中只有弹出式菜单
- C. 快捷菜单中不包含条形菜单或弹出式菜单 D. 快捷菜单中同时包含条形菜单和弹出式菜单
8. 在 Visual FoxPro 中，菜单设计器保存的文件类型为（ ）。
- A. MNX B. MNT C. MPR D. FRX
9. 在 Visual FoxPro 中支持两种类型的菜单，分别是（ ）。
- A. 条形菜单和弹出式菜单 B. 条形菜单和下拉式菜单
- C. 弹出式菜单和下拉式菜单 D. 复杂菜单和简单菜单
10. Visual FoxPro 的系统菜单，其子菜单是一个（ ）。
- A. 条形菜单 B. 弹出式菜单 C. 下拉式菜单 D. 组合菜单
11. 在 Visual FoxPro 中，条形菜单本身的内部名字为（ ）。
- A. _MSM_FILE B. _MVIEW C. _MSM_WINDO D. _MSYSMENU
12. 下列（ ）命令，使系统菜单不可用。
- A. SET SYSMENU NOSAVE B. SET SYSMENU SAVE
- C. SET SYSMENU TO D. SET SYSMENU TO DEFAULT
13. SET SYSMENU 命令的功能是（ ）。
- A. 允许在程序执行时访问系统菜单 B. 禁止在程序执行时访问系统菜单
- C. 重新配置系统菜单 D. 以上功能均可实现
14. “编辑”菜单中，“清除”的内部名称为（ ）。
- A. _MED_CLEAR B. _MED_CUT C. _MED_COPY D. _MED_FIND
15. 在“菜单设计器”窗口中，可用于上下级菜单之间切换的是（ ）。
- A. 菜单项下拉框 B. 菜单级下拉框 C. 结果下拉框 D. 插入命令按钮
16. 在菜单中选择任何一个选项时都执行一定的动作，这个动作可以是（ ）。
- A. 一条命令 B. 一个过程 C. 激活另一个菜单 D. 以上 3 种均可

11.4.2 填空题

1. 为了从用户菜单返回到默认的系统菜单，应该使用命令 SET_____ TO DEFAULT。
2. 快捷菜单与下拉式菜单比较，在快捷菜单中，只能设计_____。
3. 激活条形菜单和弹出式菜单的命令都包含关键字_____。
4. Visual FoxPro 的系统菜单是一个下拉式菜单，它是由一个_____和一个弹出式菜单组成的。
5. 将 Visual FoxPro 的系统菜单恢复成系统标准的配置，首先可执行_____命令，然后再执行_____命令。
6. 用来设置访问键的字符标志是_____。
7. 在顶层表单中添加下拉式菜单，首先应该在_____对话框中选顶层表单复选框。

第12章

报表的设计和应用

报表是 Visual FoxPro 最重要的打印输出文件，在实际应用 Visual FoxPro 工作中需要打印各种报表，用来显示和总结数据，以便进行分析或保存。

本章主要介绍各种报表的创建和设计过程，以及运行报表的方法。

12.1 创建报表

报表创建包括数据源和布局两个基本部分。报表的数据源也称为数据环境，是报表数据的来源，通常是数据库表、自由表、临时表、视图或者由查询生成。报表的布局则是报表的打印格式。

Visual FoxPro 提供了 3 种创建报表的方法：

- (1) 利用报表向导创建报表。
- (2) 利用快速报表创建简单的报表。
- (3) 利用报表设计器创建个性化的报表。

12.1.1 创建报表文件

创建报表的过程包括：

- (1) 定义报表样式。
- (2) 指定所需要的字段。
- (3) 确定打印文本及信息在报表页面上的位置。

最后生成一个扩展名为“.frx”的报表文件和一个相关的“.frt”文件。

无论报表如何复杂，无论是哪一种类型的报表，利用 Visual FoxPro 设计报表可以把其设计过程分为 4 个步骤：

- (1) 确定创建报表的类型。
- (2) 创建报表的布局文件。
- (3) 修改、定制报表布局文件。
- (4) 预览和打印报表。

1. 报表的布局

在利用 Visual FoxPro 创建报表之前，首先应该根据需要选择报表的布局类型，Visual FoxPro 提供了 4 种基本布局类型，如图 12-1 所示。

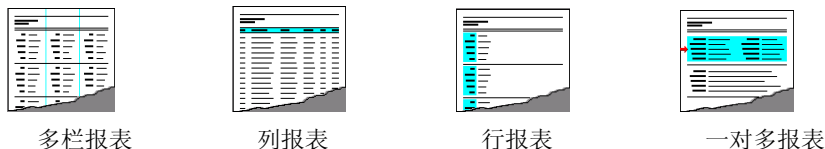


图 12-1 报表布局类型

针对图 12-1，表 12-1 中对每个报表布局类型进行了详细说明。

表 12-1 报表布局类型说明

布局类型	说明	举例
列报表	每行一条记录，每个记录的字段在页面上按水平方向放置	学生名单、财务报表、成绩报告、分组/总计报表
行报表	一行一条记录，每个记录的字段在页面上按垂直方向放置	列表、清单
一对多报表	报表的数据来自一对多的父表和子表	发票、账目或会计报表
多栏报表	多列的记录，每条记录的字段沿分栏的左侧边缘垂直放置，类似多栏排版的文本	电话号码簿、名片

2. 使用报表向导创建报表

在 Visual FoxPro 中，提供了 4 种启动报表向导的方法：

(1) 在“项目管理器”对话框中，选择“文档”选项卡，选择“报表”选项，单击“新建”按钮，弹出“新建报表”对话框，单击“报表向导”按钮，即可启动报表向导，如图 12-2、图 12-3 所示。

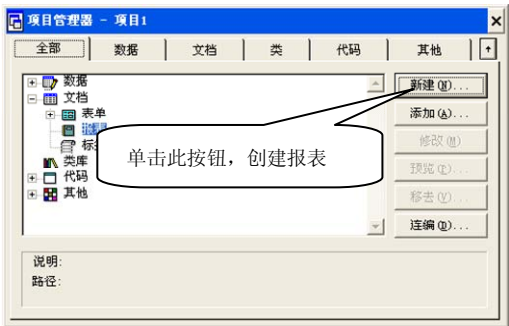


图 12-2 利用项目管理器创建报表



图 12-3 启动报表向导

(2) 在主菜单“文件”菜单中选择“新建”命令，或者单击工具栏上的“新建”按钮，打开“新建”对话框，并在“文件类型”中选择“报表”，然后单击“向导”按钮，如图 12-4 所示。

(3) 在主菜单“工具”菜单中的“向导”子菜单中选择“报表”命令，如图 12-5 所示。

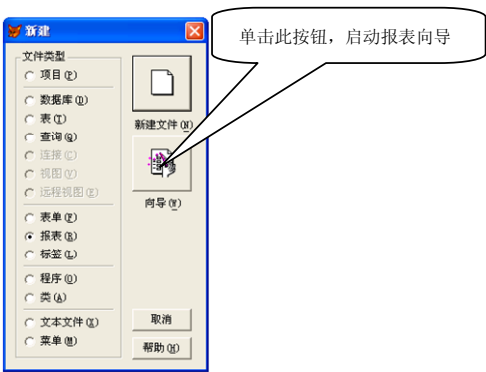


图 12-4 利用“文件”菜单启动报表向导

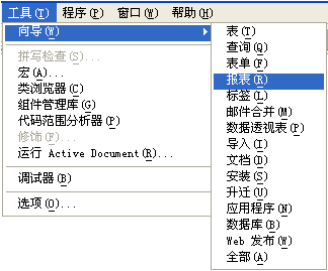


图 12-5 利用“工具”菜单启动报表向导

(4) 直接单击“常用”工具栏上的“报表向导”图标按钮启动报表向导，如图 12-6 所示。

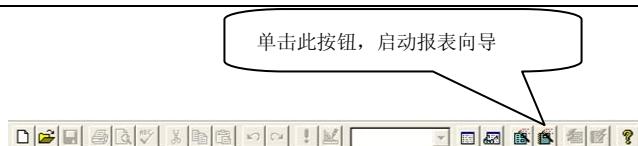


图 12-6 利用“常用”工具栏启动报表向导

报表向导启动后，弹出“向导选取”对话框，如图 12-7 所示，根据数据源的不同，可以选择不同的报表向导。如果数据源是一个表，应选择“报表向导”，如果数据源包括父表和子表，则应选择“一对多报表向导”，最后，单击“确定”按钮。

下面通过一个实例，利用“报表向导”对自由表“产品.dbf”创建一个报表。

首先打开自由表“产品.dbf”，以该自由表作为报表的数据源。按照前面所介绍的步骤启动报表向导，然后根据报表向导中的提示进行操作，具体操作步骤如下：

(1) 字段选取。在“向导选取”对话框中选择“报表向导”选项，弹出“报表向导”对话框，从“数据库和表”列表框中选择“产品”选项，“可用字段”列表框中会自动出现“产品”的所有字段。如果选择全部字段，单击向右的双箭头按钮；如果选取部分字段，可以直接双击选中字段，或者选中一个字段，单击向右的单箭头按钮，被选中的字段就会移动到“选定字段”列表框中，如图 12-8 所示。

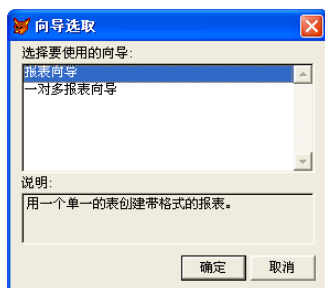


图 12-7 “向导选取”对话框

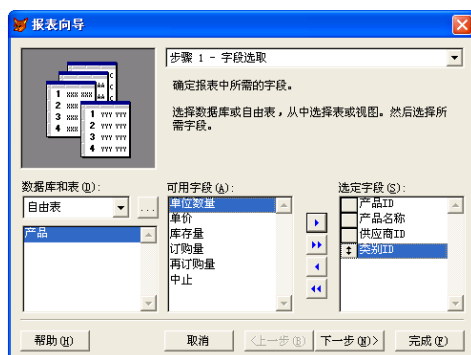


图 12-8 字段选取

(2) 分组记录。所谓分组，是指在打印记录时按照某个字段划分成数据组进行输出。只有数据表的分组字段建立索引以后才能正确分组。最多可建立 3 层分组。若多级分组，当前索引必须是“一级分组字段+二级分组字段+……”。本例中暂不分组，如图 12-9 所示。

(3) 选择报表样式。在“样式”列表框中，有多种样式可以选择，在左上角有放大镜的图中，会显示其相应的具体样式。本例选择“账务式”，如图 12-10 所示。

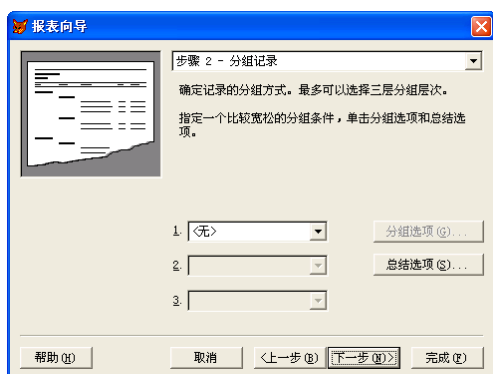


图 12-9 分组记录



图 12-10 选择报表样式

(4) 定义报表布局。该步骤确定报表的布局，本例确定“列数”为 1，“字段布局”为“列”，打印方向为“纵向”，如图 12-11 所示。

(5) 排序记录。确定记录在报表中出现的顺序，排序字段必须已经建立索引。在“可用的字段或索引标识”列表框中，选择排序字段，本例选择“产品 ID”字段进行排序，同时选择“升序”选项，如图 12-12 所示。

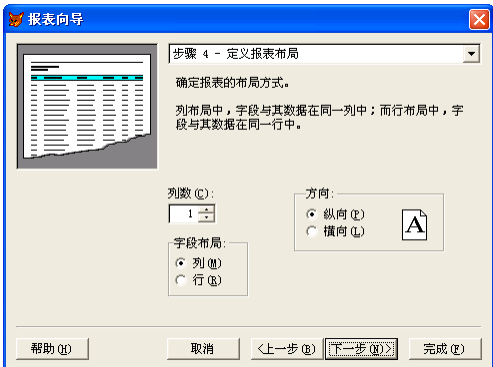


图 12-11 定义报表布局

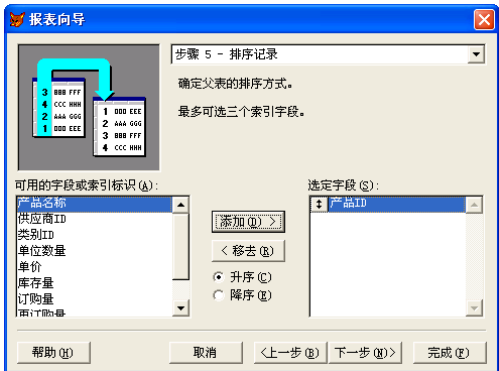


图 12-12 排序记录

(6) 完成。在此对话框中可以输入“报表标题”并选择存盘方式，还可以预览报表的设计结果，本例选择“保存报表以备将来使用”，如图 12-13 所示。

此例的预览结果如图 12-14 所示。从预览窗口中可以看到该报表的布局，虽然具备了报表的基本格式，但是仍然有许多不尽人意的地方，这也是报表向导的缺点，因此，利用报表向导创建的报表还需要进一步利用报表设计器进行修改和完善。

在预览窗口中可以利用打印预览工具栏，改变显示的百分比、退出预览、翻页或直接打印报表。最后单击报表向导上的“完成”按钮，弹出“另存为”对话框，指定报表文件存放的位置和名称，将报表保存为“.frx”报表文件。

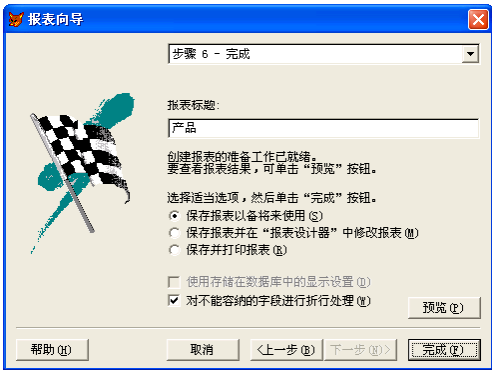


图 12-13 完成

产品ID	产品名称	单位数量	订购量	单价	供应商ID
1.00000	苹果汁	每箱24瓶	0.00000	18.00000	1.00000
2.00000	牛奶	每箱24瓶	40.00000	19.00000	1.00000
3.00000	香彩椒	每箱12瓶	70.00000	10.00000	1.00000
4.00000	盐	每箱12瓶	0.00000	22.00000	2.00000
5.00000	麻油	每箱12瓶	0.00000	21.35000	2.00000
6.00000	酱油	每箱12瓶	0.00000	25.00000	3.00000

图 12-14 报表预览效果

3. 使用报表设计器创建报表

报表设计器可以直接设计和修改报表。可以通过以下 3 种方法调用报表设计器：
(1) 在“项目管理器”对话框中选择“文档”选项卡，选择“报表”选项，然后单击“新建”按钮，在弹出的“新建报表”对话框中，单击“新建报表”按钮，即可打开报表设计器来创建一个新的报表，如图 12-15 所示。

(2) 在主菜单“文件”菜单中选择“新建”命令，或者单击“常用”工具栏中“新建”按钮，在“新建”对话框中选择文件类型为“报表”，然后单击“新建文件”按钮，即可打开报表设计器来创建一个新的报表，如图 12-16 所示。

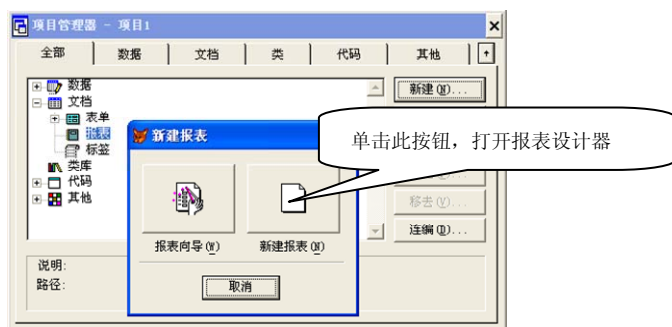


图 12-15 在“项目管理器”中打开报表设计器

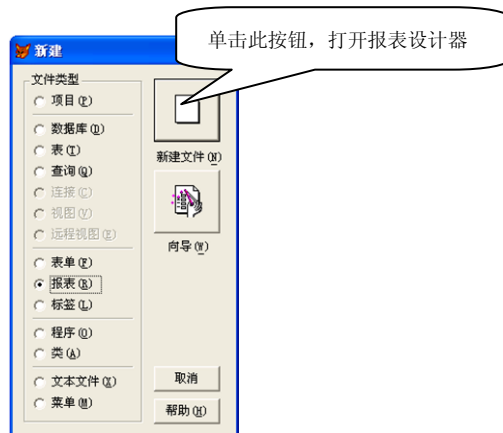


图 12-16 利用“文件”菜单打开报表设计器

(3) 在命令窗口中输入“创建报表”命令，如图 12-17 所示。

通过以上方法可以打开报表设计器来创建一个新的报表，如图 12-18 所示。在图中可以看到，报表设计器包括 4 个基本部分：

- 标题栏：显示整个报表的标题，如果报表有多页，标题只在第一页中显示。
- 页标题：显示每一页的标题。
- 细节栏：显示报表主体内容，在报表设计器中虽然仅能看到一行，但是在打印或预览时显示的行数与实际的记录数相同。
- 页注脚：显示报表页面下方的有关内容，如页码、制表人等一些必要的说明信息。

以上这些内容仅仅是报表设计器的基本组成部分，实际上根据使用的需要还可以增加一些栏目，如设计分组报表等。

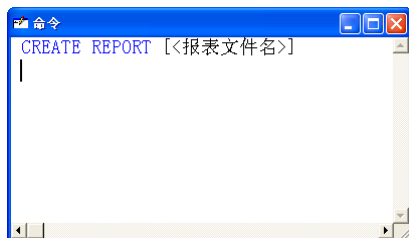


图 12-17 在命令窗口中输入“创建报表”命令

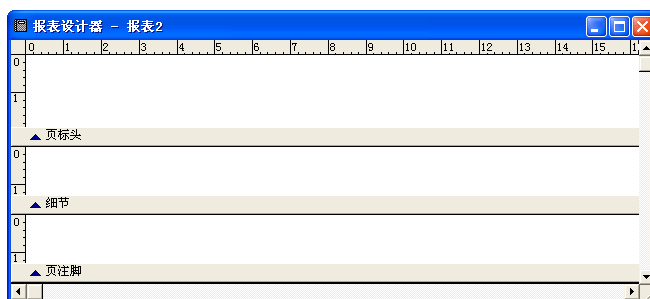


图 12-18 报表设计器

4. 创建快速报表

在 Visual FoxPro 中除了利用“报表向导”和“报表设计器”之外，还可以利用“快速报表”功能快速创建一个格式简单的报表，然后在此基础上用“报表设计器”进行修改，达到快速创建报表的目的。

创建快速报表的步骤如下：

(1) 单击“常用”工具栏中“新建”按钮，选择文件类型为“报表”，单击“新建文件”

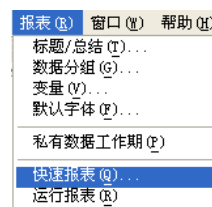


图 12-19 快速报表

按钮，启动“报表设计器”，出现一个空白报表，如图 12-18 所示。

- (2) 在“报表设计器”中，选择主菜单“报表”菜单中的“快速报表”命令，如图 12-19 所示。
- (3) 如果先打开数据源，系统会弹出“打开”对话框，从中选择需要的数据源文件，如图 12-20 所示。
- (4) 选定数据源文件之后，系统弹出“快速报表”对话框，在该对话框中选择“字段布局”、“标题”、“添加别名”、“将表添加到数据环境中”等选项，如图 12-21 所示。

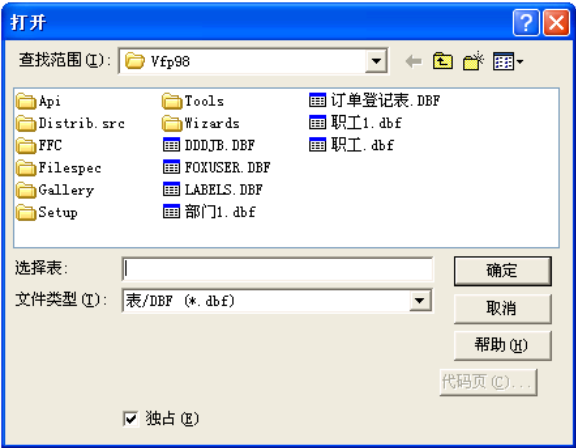


图 12-20 “打开”对话框选择数据源

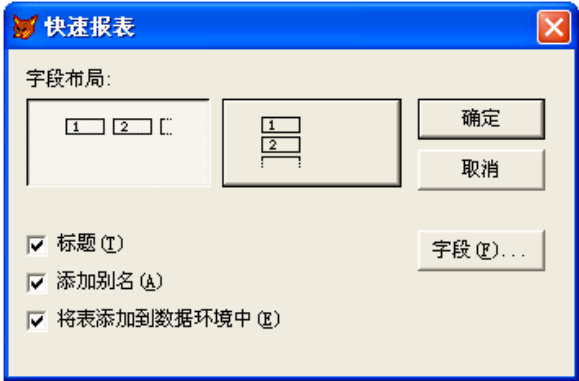


图 12-21 “快速报表”对话框

“快速报表”对话框主要按钮和选项的功能如表 12-2。

表 12-2 “快速报表”对话框功能说明

选项及按钮	功能说明
字段布局	用于设计报表的字段布局，单击左侧按钮产生列报表，单击右侧按钮产生行报表
标题	是否为报表中每个字段添加一个字段标题
添加别名	是否在每个字段前添加表的别名，若数据源是一个表则不选此项
将表添加到数据环境中	是否将打开的表文件添加到报表的数据环境中作为报表的数据源
字段	可以打开“字段选择器”，为报表选择可用的字段

“字段选择器”对话框如图 12-22 所示。



图 12-22 “字段选择器”对话框

通常在默认情况下，在“字段选择器”中显示表文件中除通用型字段以外的全部字段，单击“添加”或“全部”按钮，可以选择报表中需要显示的字段，单击“确定”按钮，则关闭“字段选择器”对话框，返回“快速报表”对话框。

(5) 单击“快速报表”对话框中“确定”按钮,所创建的“快速报表”就出现在“报表设计器”中,可以通过“报表设计器”对所创建的“快速报表”进行必要的修改,如图 12-23 所示。

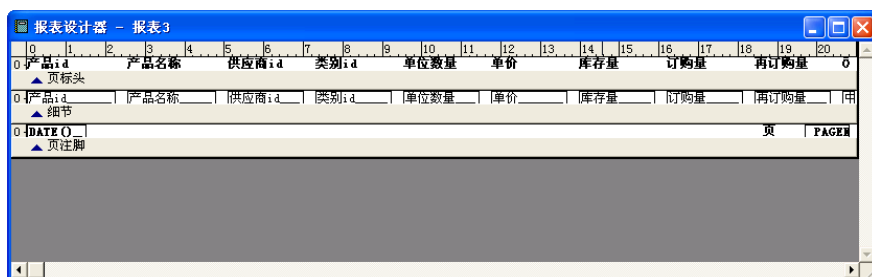


图 12-23 生成“快速报表”

(6) 单击“工具栏”中“打印预览”按钮,或者在主菜单“显示”菜单中选择“预览”命令,可以对所创建的“快速报表”进行预览,预览结果如图 12-24 所示。

产品id	产品名称	供应商id	类别id	单位数量	单价	库存量	订购量	再订购量	0
1.00000	苹果汁	1.00000	1.00000	每箱24瓶	18.00000	39.00000	0.00000	10.00000	Y
2.00000	牛奶	1.00000	1.00000	每箱24瓶	19.00000	17.00000	40.00000	25.00000	N
3.00000	番茄酱	1.00000	2.00000	每箱12瓶	10.00000	13.00000	70.00000	25.00000	N
4.00000	盐	2.00000	2.00000	每箱12瓶	22.00000	53.00000	0.00000	0.00000	N
5.00000	蒜油	2.00000	2.00000	每箱12瓶	21.35000	0.00000	0.00000	0.00000	Y
6.00000	酱油	3.00000	2.00000	每箱12瓶	25.00000	120.00000	0.00000	25.00000	N
7.00000	海鲜粉	3.00000	7.00000	每箱30盒	30.00000	15.00000	0.00000	10.00000	N
8.00000	胡椒粉	3.00000	2.00000	每箱30盒	40.00000	6.00000	0.00000	0.00000	N
9.00000	鸡	4.00000	6.00000	每袋500克	97.00000	29.00000	0.00000	0.00000	Y
10.00000	蟹	4.00000	6.00000	每袋500克	31.00000	31.00000	0.00000	0.00000	N
11.00000	民众奶酪	5.00000	4.00000	每袋6包	21.00000	22.00000	30.00000	30.00000	N
12.00000	德国奶酪	5.00000	4.00000	每箱12瓶	38.00000	68.00000	0.00000	0.00000	N
13.00000	龙蝦	6.00000	6.00000	每袋500克	6.00000	24.00000	0.00000	5.00000	N
14.00000	沙茶	6.00000	7.00000	每箱12瓶	23.25000	35.00000	0.00000	0.00000	N
15.00000	味精	6.00000	2.00000	每箱30盒	15.50000	39.00000	0.00000	5.00000	N
16.00000	饼干	7.00000	3.00000	每箱30盒	17.45000	29.00000	0.00000	10.00000	N

图 12-24 “快速报表”预览

(7) 单击“工具栏”中“保存”按钮,将所创建的“快速报表”保存为“.frx”文件。

12.1.2 报表工具栏

在 Visual FoxPro 中,报表设计器包含了两个与报表设计有关的工具栏:“报表设计器”工具栏和“报表控件”工具栏。可以通过选择主菜单的“显示”菜单中“工具栏”命令,弹出“工具栏”对话框,选择或清除相应的工具栏,如图 12-25 所示。

1. “报表设计器”工具栏

在 Visual FoxPro 中,打开“报表设计器”时,系统会在主窗口中自动出现“报表设计器”工具栏,如图 12-26 所示。

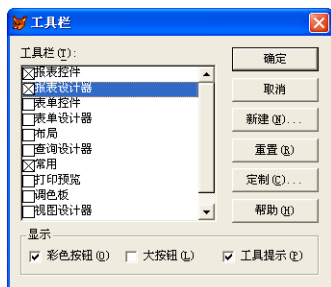


图 12-25 “工具栏”对话框



图 12-26 “报表设计器”工具栏

“报表设计器”工具栏上各个按钮的功能如表 12-3 所示。

表 12-3 “报表设计器” 工具栏按钮功能描述

按钮	功能描述
	“数据分组” 按钮 显示“数据分组”对话框，用于创建数据分组及指定其属性
	“数据环境” 按钮 显示报表的“数据环境设计器”窗口
	“报表控件” 工具栏按钮 显示或关闭“报表控件”工具栏
	“调色板” 工具栏按钮 显示或关闭“调色板”工具栏
	“布局” 工具栏按钮 显示或关闭“布局”工具栏

2. “报表控件” 工具栏

在 Visual FoxPro 中，打开“报表设计器”时，系统也会同时在主窗口中自动出现“报表控件”工具栏，如图 12-27 所示。

“报表控件”工具栏上各个按钮的功能如表 12-4 所示。



图 12-27 “报表控件” 工具栏

表 12-4 “报表控件” 工具栏按钮功能描述

按钮	功能描述
	“选定对象” 按钮 移动或更改控件对象的大小
	“标签” 按钮 用于输入数据记录之外的信息
	“域控件” 按钮 用于显示字段、内存变量或者其他表达式的内容
	“线条” 按钮
	“矩形” 按钮
	“圆角” 按钮
	“图片/ActiveX 绑定控件” 按钮 用于显示图片或者通用型字段的内容
	“按钮锁定” 按钮 允许添加多个相同类型的控件而不需要多次选中该控件按钮

12.2 设计报表

创建报表文件之后，需要进一步改进报表。报表的设计工作是在“报表设计器”中完成的，可以利用“打开”对话框，将报表类型文件.frx 打开，也可以通过使用命令 MODIFY REPORT <报表文件名>打开需要修改的报表文件。

在报表设计器中可以设置报表数据源、更改报表布局、添加报表控件及设计数据分组。

12.2.1 报表的数据源和布局

在 Visual FoxPro 中，报表总是与一定的数据源相联系的，在设计报表时，首先需要确定报表的数据源。报表的数据源通常是数据库表、自由表、临时表、视图或者查询，当数据源中的数据变化时，报表的输出内容也会随之变化，但是报表的格式是不变的。

1. 设置报表数据源

报表的数据源在数据环境中设置，数据环境是存放报表数据源的一个容器，报表的输出内容就来自数据环境。当每次运行报表时，数据源都会被自动打开。

可以通过以下 3 种方法打开报表数据环境设计器：

- (1) 在“报表设计器”的任何空白处单击鼠标右键，弹出快捷菜单，从中选择“数据环境”命令。
- (2) 在“报表设计器”中，选择主菜单的“显示”菜单中的“数据环境”命令。
- (3) 在“报表设计器”中，单击“报表设计器”工具栏中的“数据环境”按钮。

打开后的数据环境设计器如图 12-28 所示。

数据环境可以通过以下 3 种方式管理报表的数据源：

- (1) 打开或运行报表时打开数据表或视图。
- (2) 基于相关数据表或视图收集报表所需数据集合。
- (3) 关闭或释放报表时关闭数据表。

当打开数据环境设计器之后，就可以为报表添加和删除制定的数据源，具体操作步骤如下：

- (1) 通过“报表设计器”创建一个空白报表，利用上面介绍的 3 种方法之一打开“数据环境设计器”窗口。
- (2) 在“数据环境设计器”窗口中，单击鼠标右键，弹出快捷菜单，从中选择“添加”命令，系统则弹出“添加表或视图”对话框，如图 12-29、图 12-30 所示。

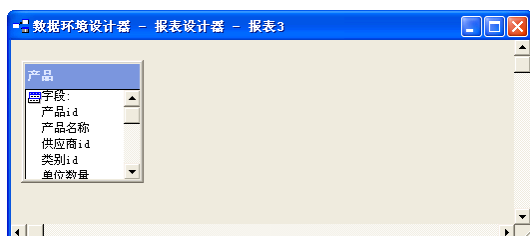


图 12-28 数据环境设计器

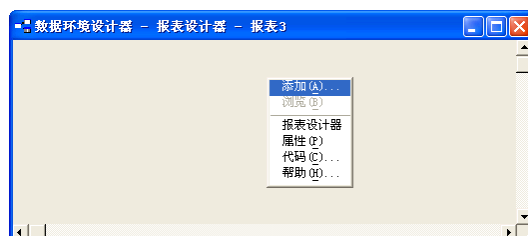


图 12-29 向“数据环境设计器”添加数据源

- (3) 在“添加表或视图”对话框中选择需要的表或视图作为数据源，单击“添加”按钮，作为数据源的表或视图就被加到报表的数据环境中，如图 12-31 所示。

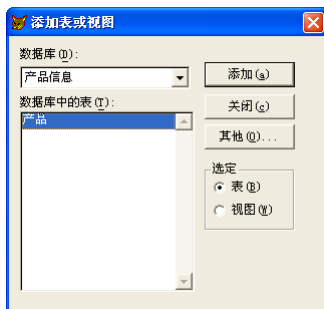


图 12-30 “添加表或视图”对话框

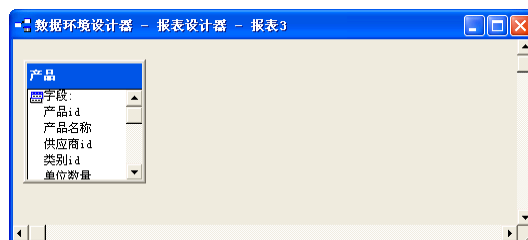


图 12-31 将数据表添加到数据环境中

- (4) 单击“关闭”按钮，退出“添加表或视图”对话框。

- (5) 如果需要移去数据环境中的表或视图，选中需要移去的表或视图，单击鼠标右键，弹出快捷菜单，选中“移去”命令，或者选中需要移去的表或视图后，按【Del】键将其移去，如图 12-32 所示。

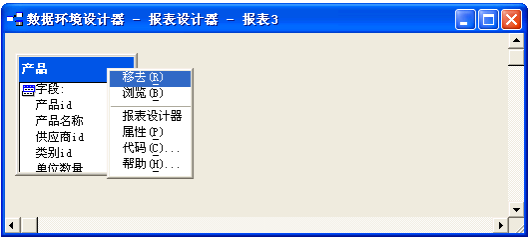


图 12-32 将数据表从数据环境中移去

下。

当通过“快速报表”创建报表时，会出现“页标头”、“细节”和“页注脚”3个默认的基本带区，若需要使用其他带区，则需要用户自己设置。报表带区及作用如表 12-5 所示。

表 12-5 报表带区及作用

带区	作用
标题	每张报表开头打印一次或单独占用一页，如报表名称
页标头	每一页打印一次，如记录的字段名
细节	每条记录打印一次，如各记录的字段值
页注脚	每一页下面打印一次，如页码
总结	每张报表最后一页打印一次或单独占用一页
组标头	如果有数据分组，每组打印一次
组注脚	如果有数据分组，每组打印一次
列标头	在分栏报表中，每列打印一次
列注脚	在分栏报表中，每列打印一次

具体设置报表带区的方法如下：

1) 设置“标题”和“总结”带区

在主菜单“报表”菜单中选择“标题/总结”命令，弹出“标题/总结”对话框，如图 12-33 所示。

在“标题/总结”对话框中，选中“标题带区”和“总结带区”复选框，系统会在报表中添加一个“标题”带区和一个“总结”带区，“标题”带区会被自动地放在报表的顶部，“总结”带区会被自动地放在报表的尾部。如果想把“标题”或“总结”内容单独打印一页，则需要选择“标题/总结”对话框中“新页”复选框。

设置后的报表界面如图 12-34 所示。

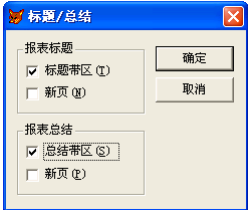


图 12-33 “标题/总结”对话框

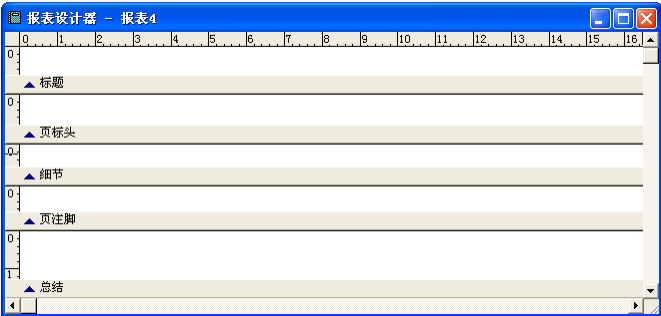


图 12-34 设置“标题”和“总结”带区的报表界面

2) 设置“列标头”和“列注脚”带区

如果需要创建多栏报表就需要设置“列标头”和“列注脚”带区。操作方法如下：

在主菜单“文件”菜单中选择“页面设置”命令，会弹出“页面设置”对话框，如图 12-35 所示。

在“页面设置”对话框中，可以将“列数”微调器中的数值进行调整，调整为大于 1 的整数，报表将添加一个“列标头”带区和一个“列注脚”带区。

设置后的报表界面如图 12-36 所示。

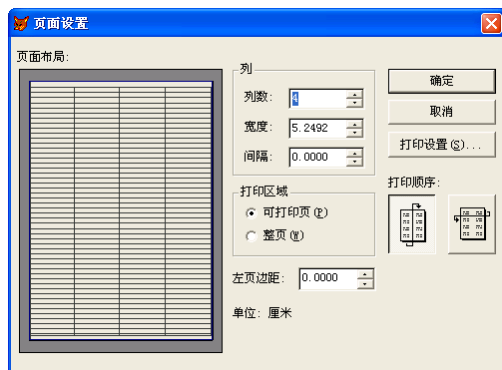


图 12-35 “页面设置”对话框

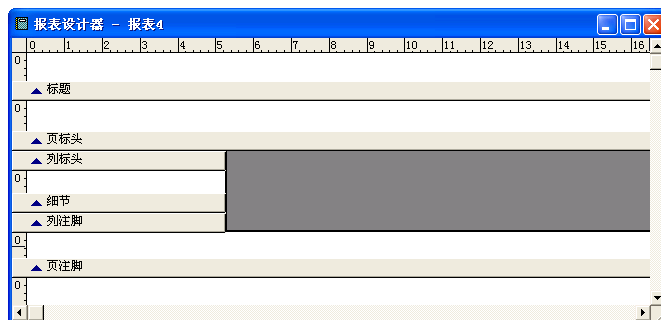


图 12-36 设置“页标头”和“页注脚”带区的报表界面

3) 设置“组标头”和“组注脚”带区

如果需要对报表中的数据进行分组，首先需要对数据源中数据表建立索引，然后才能对数据表中建立索引的字段在报表中进行分组。

在主菜单“报表”菜单中选择“数据分组”命令，或者单击“报表设计器”工具栏中“数据分组”按钮，系统会弹出“数据分组”对话框，如图 12-37 所示。

在“数据分组”对话框中，单击“分组表达式”右侧...按钮，弹出“表达式生成器”对话框，从中选择分组表达式，如图 12-38 所示。

设置“分组表达式”后，报表设计器中将添加一个或多个“组标头”和“组注脚”带区，带区的数目与分组表达式的数目有关。

设置后的报表界面如图 12-39 所示。

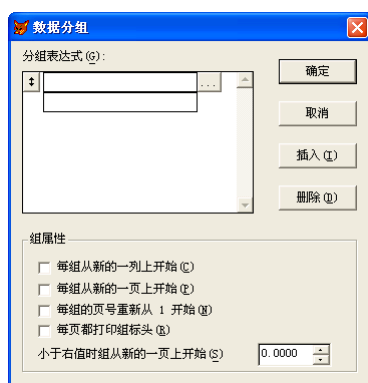


图 12-37 “数据分组”对话框

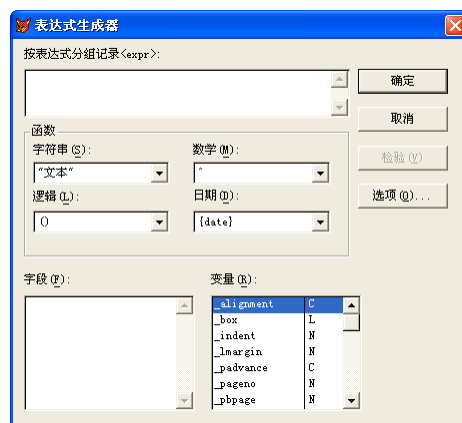


图 12-38 “表达式生成器”对话框

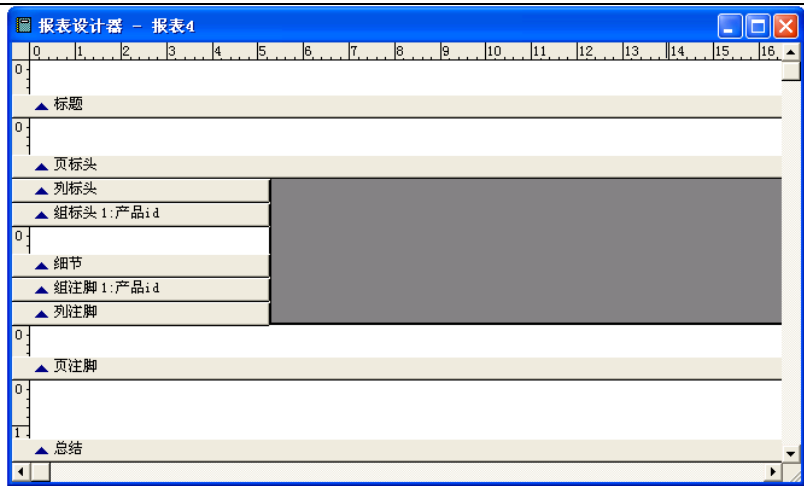


图 12-39 设置“组标头”和“组注脚”带区的报表界面

3. 调整带区高度

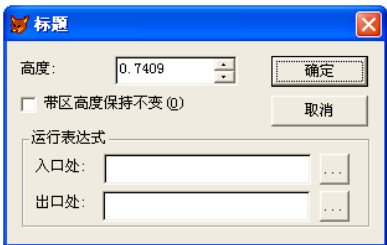


图 12-40 设置带区高度对话框

在该对话框中，各选项的作用如下所述：

- 高度：可以直接输入所需高度的数值，或者调整“高度”微调器中的数值。
- 带区高度保持不变：可以防止报表带区因为容纳过长的数据或者从中删除数据而发生位置的移动。
- 运行表达式入口处：系统将在打印该带区内容之前计算该表达式。
- 运行表达式出口处：系统将在打印该带区内容之后计算该表达式。

12.2.2 在报表中使用控件

可以在“报表设计器”中向报表添加所需的控件，以此来安排所需要打印的内容。在报表中可以添加的控件有标签控件、域控件、线条、矩形、圆角矩形、图片/ActiveX 绑定控件等。

1. 标签控件

标签控件用来设计一些说明性文字或报表标题文本等，用途十分广泛。

1) 插入标签控件

在“报表控件”工具栏中单击“标签”按钮，然后在报表的指定位置上单击鼠标，便出现一个插入点，即可在当前位置上输入文本。

2) 更改字体、字形和字号

标签控件系统默认的字体、字号是宋体常规小 5 号，可以更改每个标签控件中文本的字体、字形和字号。用鼠标选定需要更改的控件，在主菜单“格式”菜单中选择“字体”命令，弹出“字体”对话框，在“字体”对话框中选择适当的字体、字形、字号大小，然后单击“确定”按钮。

3) 更改默认字体

如果需要更改标签控件的默认字体，需要在主菜单“报表”菜单中选择“默认字体”命令，弹出“字体”对话框，然后在“字体”对话框中设置字体、字形和字号大小，然后单击“确定”按钮，所做选择就作为标签控件的默认字体。

2. 线条、矩形和圆角矩形控件

“报表控件”工具栏中提供线条、矩形或圆角矩形控件，用来在报表适当的位置上添加相应的图形线条，使其效果更美观。

1) 添加控件

在“报表控件”工具栏上单击“线条”按钮、“矩形”按钮或“圆角矩形”按钮，然后在报表的一个带区中拖曳鼠标，将分别生成线条、矩形或圆角矩形。

2) 更改样式

可以更改垂直或水平线条、矩形和圆角矩形所用线条的粗细，从细线到 6 磅粗的线，也可以更改线条的样式，从点线到点线和虚线的组合。对于圆角矩形，还可以设置圆角矩形的圆角样式。

设置的方法如下：

(1) 从主菜单“格式”菜单中选择“绘图笔”命令，再从“绘图笔”子菜单中选择适当的大小或样式。

(2) 用鼠标双击圆角矩形控件，弹出“圆角矩形”对话框。在“样式”区域选择想要的圆角样式；如果需要，还可以选择“对象位置”选项和“向下伸展”选项；单击“打印条件”按钮，可以打开“打印条件”对话框设置相应的打印条件；最后单击“确定”按钮完成设置，如图 12-41 所示。

3) 调整控件

选定控件，然后拖动控件四周的某个控点改变控件的宽度和高度。也可以在选定控件之后，按下【Shift】键，再利用光标移动键（↑、↓、←、→）来改变控件的长度。

如果需要制作完全相同的控件，首先需要选定控件，接着单击工具栏上的“复制”按钮，再单击“粘贴”按钮即可，也可以选择主菜单“编辑”菜单中的“复制”、“粘贴”命令。

对于不需要的控件，可以在选定后按【Del】键，或者单击工具栏上的“剪切”按钮，或者选择主菜单“编辑”菜单中的“剪切”命令就可以删除该控件。

4) 同时选定多个控件

如果需要对多个控件进行移动、复制、删除和设置操作，可以同时选择多个控件。方法是拖动鼠标将需要选择的多个控件进行圈选，或者在选定一个控件后，按下【Shift】键，同时再选定其他控件。

同时选定的多个控件，可以作为一组内容来移动、复制、设置或删除。

5) 设置控件布局

利用“布局”工具栏中的按钮可以方便地调整“报表设计器”中被选中控件的相对大小与位置。

“布局”工具栏可以通过选择主菜单“显示”菜单中的“布局工具栏”命令打开，或者单击“报表设计器”工具栏上的“布局工具栏”按钮打开。“布局”工具栏如图 12-42 所示。

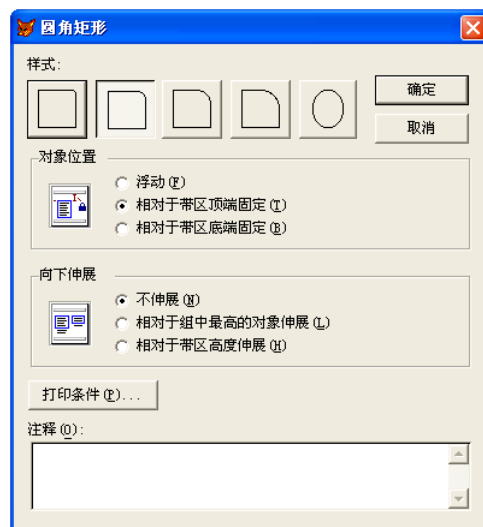


图 12-41 “圆角矩形”对话框

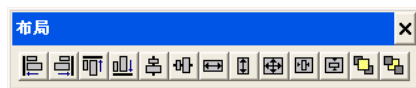


图 12-42 布局工具栏

在“布局”工具栏中一共有 13 个按钮，具体的功能如表 12-6 所示。

表 12-6 “布局”工具栏中各按钮的功能

按钮	功能	作用
左边对齐	使选定的所有控件向其中最左边的控件左侧对齐	主菜单“格式”菜单中“对齐”命令
右边对齐	使选定的所有控件向其中最右边的控件右侧对齐	
顶边对齐	使选定的所有控件向其中最顶端的控件顶边对齐	
底边对齐	使选定的所有控件向其中最下端的控件底边对齐	
垂直居中对齐	使选定的所有控件的中心在一条垂直线上	
水平居中对齐	使选定的所有控件的中心在一条水平线上	
水平居中	使选定控件的中心处于区带水平方向的中间位置	主菜单“格式”菜单中“大小”命令
垂直居中	使选定控件的中心处于区带垂直方向的中间位置	
相同宽度	将选定的所有控件宽度调整到与其中最宽控件相同	
相同高度	将选定的所有控件高度调整到与其中最高控件相同	
相同大小	使选定的所有控件具有相同的大小	主菜单“格式”中“置前”和“置后”命令
置前	将选定的控件移至其他控件的最上层	
置后	将选定的控件移至其他控件的最下层	

3. 域控件

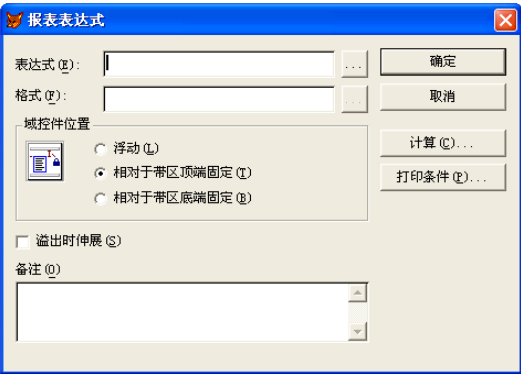


图 12-43 “报表表达式”对话框

域控件用于打印表或视图中的字段、变量和表达式的计算结果。例如，通过设置域控件，可以自动地给报表添加日期、时间、页码等。

1) 添加域控件

在报表中添加域控件有两种方法：

(1) 在“报表设计器”中选择“数据环境”，在“数据环境设计器”中选择所需要的表或视图，然后把相应的字段拖曳到报表指定的带区中相应的位置即可。

(2) 在“报表控件”工具栏中单击“域控件”按钮，然后在报表指定的带区中相应的位置上单击鼠标，系统弹出“报表表达式”对话框，如图 12-43 所示。

在“报表表达式”对话框中，主要选项和按钮的功能如表 12-7 所示。

表 12-7 “报表表达式”对话框选项和按钮功能

选项或按钮	功能
“表达式”文本框	可以输入字段名，或单击右侧按钮打开“表达式生成器”对话框设计表达式
“计算”按钮	可以打开“计算字段”对话框，对于添加的可计算字段创建一个计算结果
“域控件位置”选项组	制定域控件根据周围控件大小浮动，或固定域控件在报表中的位置
“备注”编辑框	可以输入备注文本，文本内容添加到.frx 文件中，但不会在当前报表中显示

其中，在“报表表达式”对话框中单击“表达式”文本框右侧按钮，将打开“表达式生成器”对话框，在“字段”列表框中双击所需字段名，表名和字段名会自动出现在“报表字段的表达式”编辑框中，如图 12-44 所示。

如果“表达式生成器”对话框中的“字段”列表框中为空，则表明当前没有设置数据源，应该首先设置数据源。

如果在“报表表达式”对话框中单击“计算”按钮，将打开“计算字段”对话框，添加可计算字段创建一个计算结果，如图 12-45 所示。



图 12-44 “表达式生成器”对话框

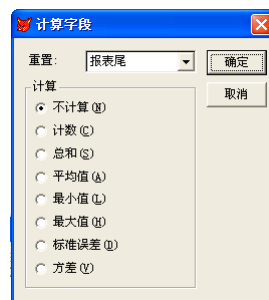


图 12-45 “计算字段”对话框

在“计算字段”对话框中，“计算”选项中有 8 种计算方式，可以制定在报表表达式中执行的计算。

2) 域控件格式设置

在报表中添加域控件之后，还需要设置该控件的数据类型和打印格式。在“报表表达式”对话框中，单击“格式”文本框右侧...按钮，弹出“格式”对话框，如图 12-46 所示。

在“格式”对话框中，首先选择域控件的数据类型，域控件的数据类型可以是字符型、数值型或日期型，每一种数据类型都有自己的格式选项，因此，选定不同的数据类型，“编辑选项”将显示相应数据类型可用的格式选项。

3 种数据类型所对应的“编辑选项”及其含义如表 12-8 所示。



图 12-46 “格式”对话框

表 12-8 编辑选项及其含义

数据类型	编辑选项	含义
字符型	全部大写	将所有的字符转化为大写
	忽略输入掩码	显示但不存储不符合格式的字符
	SET DATE 格式	显示所定义格式的日期数据
	英国日期格式	显示英国日期格式的日期数据
	左对齐	从选定控件位置的最左端显示字符
	右对齐	从选定控件位置的最右端显示字符
	居中对齐	将字符置于中间
数值型	左对齐	从选定控件位置的最左端显示字符
	如果为零保持为空	如果控件输出为零则不打印
	负数加括号	将负数放入括号内
	SET DATE 格式	显示所定义格式的日期数据
	英国日期格式	显示英国日期格式的日期数据
	如果为正，加 CR（收入）	在正数后显示 CR（贷方）
	如果为负，加 DB（支出）	在负数后显示 DB（借方）
	前导零	打印全部前导零
	货币型	按指定格式显示货币格式
	科学计数法	按科学计数法显示数据
日期型	SET DATE 格式	显示所定义格式的日期数据
	英国日期格式	显示英国日期格式的日期数据

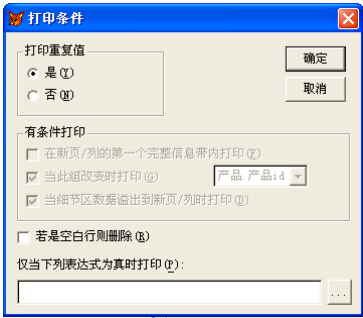


图 12-47 “打印条件”对话框

3) 打印条件设置

当打印报表时，可以对每个控件的输出进行控制。在“报表表达式”对话框中单击“打印条件”按钮，弹出“打印条件”对话框。对于不同类型的对象，该对话框所显示的内容各不相同，可以根据需要进行设置，如图 12-47 所示。

“打印条件”对话框中各个选项的含义如表 12-9 所示。

表 12-9 “打印条件”各选项含义

选项	含义
打印重复值	当某一字段出现多个重复值时，是否打印重复值
有条件打印	如果“打印重复值”选项选择“否”，则该选项 3 个选项都可用；如果选择“是”，则该选项中仅最后一个选项可用
若是空白行则删除	是否打印输出数据源中的空白记录
仅当下列表达式为真时打印	设置一个打印表达式，在打印之前被计算，若结果为“真”则打印该字段，否则不打印

其中，设置打印表达式需要在“仅当下列表达式为真时打印”文本框中输入表达式，或者单击其文本框右侧按钮，弹出“表达式生成器”对话框，输入或选择打印表达式。

4. OLE 对象

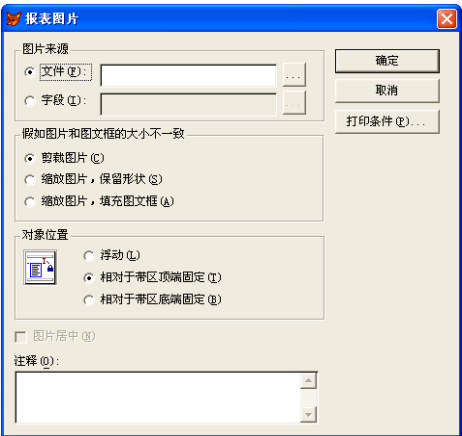


图 12-48 “报表图片”对话框

为了美化报表同时能够更加直观地反映报表表示的意义，需要在报表中加入图片等对象，这就是 OLE——对象链接与嵌入技术。一个 OLE 对象可以是图片、声音、文档等。

1) 添加图片

在“报表控件”工具栏中单击“图片/ActiveX 绑定控件”按钮，在报表指定带区内单击并拖曳鼠标，弹出“报表图片”对话框，如图 12-48 所示。

在“报表图片”对话框中，“图片来源”选项有以下两种形式：

- (1) 将一个图片文件加入报表中。
- (2) 将一个包含 OLE 对象的通用型字段加入报表中。

2) 调整图片

可以调整加入报表的图片大小以适应报表图文框的大小。在“报表图片”对话框中“假如图片和图文框的大小不一致”选项组中有 3 个选项：

- 裁剪图片：可以根据图文框的大小自动裁剪图片。
- 缩放图片，保留形状：可以在图文框中放置一个完整的、保留原来形状的图片。
- 缩放图片，填充图文框：可以将图片填满整个图文框。

3) 对象位置

在“报表图片”对话框中“对象位置”选项组中也有 3 个选项：

- 浮动：图片可以相对于周围控件的位置浮动。
- 相对于带区顶端固定：可以将图片放置在与带区顶端一定距离的位置上。
- 相对于带区底端固定：可以将图片放置在与带区底端一定距离的位置上。

4) 注释

在“报表图片”对话框中的“注释”编辑框中输入对 OLE 对象的注释文本以供参考。

12.3 数据分组和多栏报表

所谓数据分组，是指按照某种标志把报表中需要打印的记录进行分组，同时还可以对每组记录中的数据进行统计、计算并在报表中显示。

12.3.1 设计分组报表

在一个报表中可以设置一个或多个数据分组，在 Visual FoxPro 中最多允许设置 3 层分组，组的分隔基于分组表达式。

1. 设置报表的记录顺序

设计分组报表首先需要在报表的数据源中按分组标志建立索引，一个表可以有多个索引，通过在“命令窗口”输入指定当前索引的命令设置主控索引，如图 12-49 所示。

另外，在“数据环境设计器”中也可以指定当前索引。具体步骤如下：

(1) 在主菜单“显示”菜单中选择“数据环境”命令，或者单击“报表设计器”工具栏上的“数据环境”按钮，或者在“报表设计器”中单击鼠标右键，在快捷菜单中选择“数据环境”命令，弹出“数据环境设计器”对话框。

(2) 在“数据环境设计器”中单击鼠标右键，在快捷菜单中选择“属性”命令，弹出“属性”窗口，如图 12-50 所示。

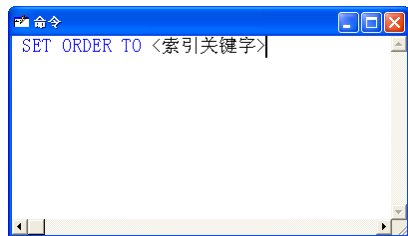


图 12-49 在命令窗口中输入“指定当前索引”命令



图 12-50 “属性”窗口

(3) 选择“Order”属性，输入索引名，或者在索引列表中选定一个索引，如图 12-51 所示。

2. 设计单级分组报表

一个单级分组报表可以基于选择表达式进行一级数据分组。具体步骤如下：

(1) 利用“报表设计器”创建一个新报表，设置数据环境，然后在主菜单“报表”菜单中选择“数据分组”命令，或者单击“报表设计器”工具栏中的“数据分组”按钮，或者在“报表设计器”中单击鼠标右键，在快捷菜单中选择“数据分组”命令，弹出“数据分组”对话框，如图 12-52 所示。

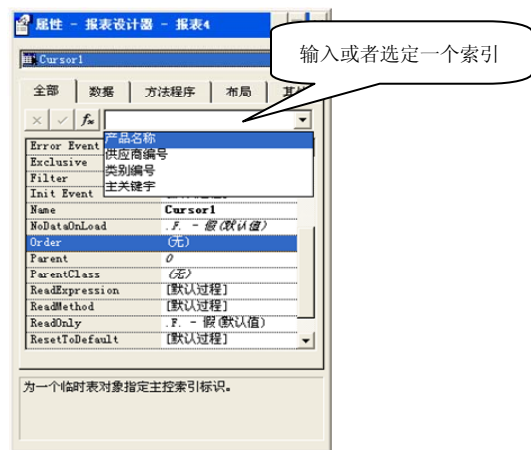


图 12-51 设置“Order”属性

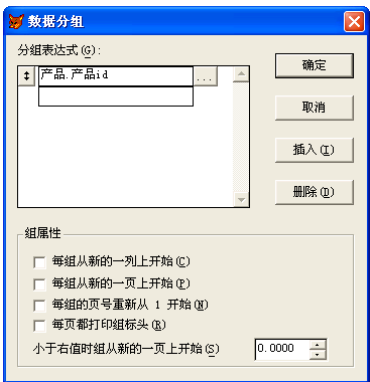


图 12-52 “数据分组”对话框

- (2) 在“分组表达式”文本框中输入分组表达式，或者单击其右侧...按钮，利用“表达式生成器”对话框创建分组表达式。
- (3) 在“组属性”选项组中选择需要的属性，“组属性”各选项含义如表 12-10 所示。

表 12-10 “组属性”各选项含义

选项	含义
每组从新的一列上开始	是否把新出现的一组内容打印到下一列上
每组从新的一页上开始	是否把新出现的一组内容打印到下一页上
每组的页号重新从 1 开始	是否把新出现的一组内容打印到下一页上，并把页号重置为 1
每页都打印组标头	当一组内容出现在多页上时，是否在每一页都打印组标头

- (4) 单击“确定”按钮，完成单级分组报表设计。

3. 设计多级数据分组报表

在设计多级数据分组报表时，需要注意分组的级与多重索引的关系。

1) 多个数据分组基于多重索引

多级数据分组报表的数据源必须可以分出级别，根据实际应用的需要建立复合索引（多重索引）。

2) 分组层次

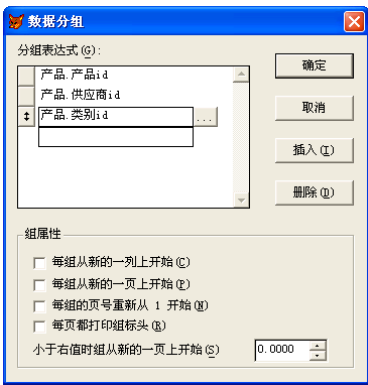


图 12-53 设计多级数据分组报表

一个数据分组对应于一组“组标头”和“组页脚”带区。在“报表设计器”中将数据分组按照创建的顺序进行编号，分组的级别越细，分组的编号越大，编号越大的分组，距离“细节”带区越近。在“报表设计器”中，随时可以改变分组的顺序、重复组标头，以及删除或更改组带区。

3) 设计多级数据分组报表

设计多级数据分组报表步骤与设计单级数据分组报表完全一致，只需要在“数据分组”对话框中连续输入多个分组表达式即可，如图 12-53 所示。

4) 更改分组

如果需要更改已经创建好的数据分组，可以通过“数据分组”对话框对分组表达式进行修改，重新设置组属性。如需改变数据分组的顺序则可以通过分组表达式左侧的移动按钮调整分组表达式的顺序，单击“删除”按钮删除已有分组表达式。

12.3.2 设计多栏报表

多栏报表是一种分为多个栏目进行打印输出的报表。

1. 设置“列标头”和“列注脚”带区

在主菜单“文件”菜单中选择“页面设置”命令，弹出“页面设置”对话框，在“列”选项中，利用“列数”微调器调整栏目数，同时在“报表设计器”中将自动添加一个“列标头”和“列注脚”带区，同时“细节”带区宽度相应缩短，如图 12-54 所示。

2. 添加控件

可以向多栏报表中添加各种控件，但控件不要超过报表设计器带区的宽度，否则打印内容会相互重叠。

3. 设置页面

系统默认报表的打印顺序为“自上而下”，而多栏报表需要把打印顺序设置为“自左向右”，如图 12-55 所示。

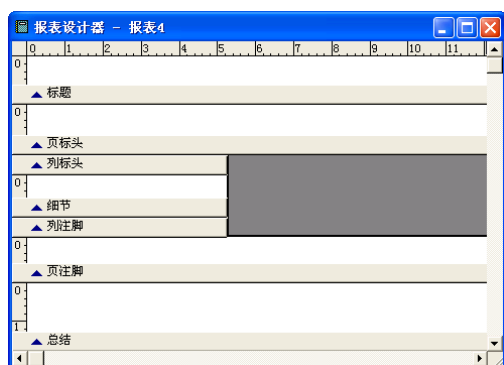


图 12-54 设计多栏报表



图 12-55 设置打印顺序为“自左向右”

12.3.3 报表输出

设计报表的最终目的是要按照一定的格式输出符合要求的数据，即打印报表。

1. 设置报表的页面

在打印报表之前，需要首先设置报表的页面。在主菜单“文件”菜单中选择“页面设置”命令，弹出“页面设置”对话框，在“左页边距”数值框中输入数值或通过“微调器”调整数值来设置报表的页边距。在“页面设置”对话框中，单击“打印设置”按钮，弹出“打印设置”对话框，设置纸张的大小、来源及纸张的方向，如图 12-56 所示。

2. 预览报表

为确保正确输出报表，需要利用“预览”功能在屏幕上检查一下报表页面设置是否符合要求。可以在“报表设计器”中单击鼠标右键，在快捷菜单中选择“预览”命令，或者在主菜单“显示”菜单中选择“预览”命令，或者单击“报表工具栏”中“打印预览”按钮实现预览报表。

3. 打印输出报表

在“报表设计器”中单击鼠标右键，在快捷菜单中选择“打印”命令，或者单击“常用”工具栏上的“运行”按钮，或者从主菜单“文件”菜单

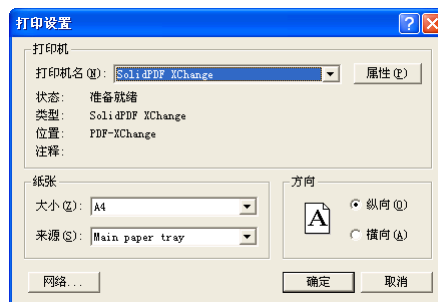


图 12-56 “打印设置”对话框

中选择“打印”命令，弹出“打印”对话框，如图 12-57 所示。

另外，在命令窗口中输入打印或预览报表命令，也可以实现报表的打印输出和预览操作，如图 12-58 所示。

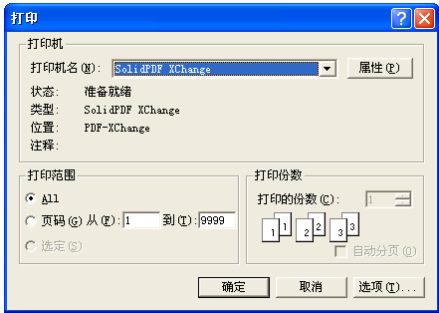


图 12-57 “打印”对话框

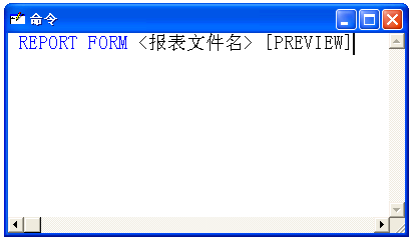


图 12-58 在命令窗口中输入“打印或预览”命令

12.4 本章习题

12.4.1 选择题

1. 报表的数据源可以是（ ）。
A. 表或视图
B. 表或查询
C. 表、查询或视图
D. 表或其他报表
2. Visual FoxPro 的报表文件.frx 中保存的是（ ）。
A. 打印报表的预览格式
B. 已经生成的完整报表
C. 报表的格式和数据
D. 报表设计格式的定义
3. 在 Visual FoxPro 中，报表的数据源不包括（ ）。
A. 视图
B. 自由表
C. 查询
D. 文本文件
4. 为了在报表中打印当前时间，这时应该插入一个（ ）。
A. 表达式控件
B. 域控件
C. 标签控件
D. 文本控件
5. 在“报表设计器”窗口中，若要数据分组，则依据为（ ）。
A. 查询
B. 排序
C. 分组表达式
D. 以上都不是
6. 下列方法中，不能启动报表向导的是（ ）。
A. 在命令窗口中输入 CREATE REPORT 命令
B. 在“新建”对话框中启动报表向导
C. 在“工具”菜单中选择“向导”子菜单，再选择“报表”命令
D. 直接单击工具栏上的“报表向导”按钮
7. 在 Visual FoxPro 中，报表的基本组成部分是（ ）。
A. 视图和布局
B. 数据库和布局
C. 数据表和布局
D. 数据源和布局
8. 在 Visual FoxPro 中，在屏幕上预览报表的命令是（ ）。
A. PREVIEW REPORT
B. REPORT FORM PREVIEW
C. DO REPORT PREVIEW
D. RUN REPORT PREVIEW
9. 组标头的打印方式是（ ）。
A. 每个页面打印一次
B. 每列打印一次
C. 每组打印一次
D. 以上都不是
10. 在创建报表时，基本带区包括（ ）。

- A. 组标头、细节和组注脚 B. 页标头、细节和总结
C. 页标头、细节和页注脚 D. 报表标题、细节和页注脚
11. 在“报表设计器”中，要添加标题或其他说明文字，应使用的控件是（ ）。
A. 域控件 B. 标签 C. 文本框 D. 列表框

12.4.2 填空题

1. 为修改已建立的报表文件，打开报表设计器的命令是_____REPORT。
2. 设计报表通常包括两部分内容：_____和_____。
3. 如果已对报表进行了数据分组，则报表中会自动包含_____和_____带区。
4. 多栏报表的栏目数据可以通过_____对话框来设置。
5. 报表中的域控件用于打印字段、_____和_____。
6. 在报表设计器中添加“标签”的目的是_____。

第13章

应用程序的开发和生成

数据库应用系统开发是学习 Visual FoxPro 的最终目的。在进行数据库应用系统的开发过程中，需要综合地运用数据库、表单、报表、菜单等知识和设计技巧，通过项目管理器把构成数据库应用系统的这些独立的文件连编成一个完整的应用程序，组成一个功能强大的数据库应用系统，最终形成一个扩展名为.app 的应用文件或.exe 的可执行文件。

本章主要介绍开发数据库应用系统的方法和步骤，学习如何设计一个完整的数据库应用系统。

13.1 应用程序项目综合实践

在使用 Visual FoxPro 开发数据库应用系统时，可以利用“项目管理器”将应用系统的各个部分组织起来，用集成化的方法建立应用系统项目，并进行项目测试。

13.1.1 系统开发基本步骤

一个数据库应用系统根据其系统的重点和复杂性的不同，通常分为以下 3 种类型：

- 输入密集型。
- 输出密集型。
- 处理密集型。

无论哪一种类型的数据库应用系统，利用 Visual FoxPro 所开发的系统一般都包括以下几个基本组成部分，如图 13-1 所示。

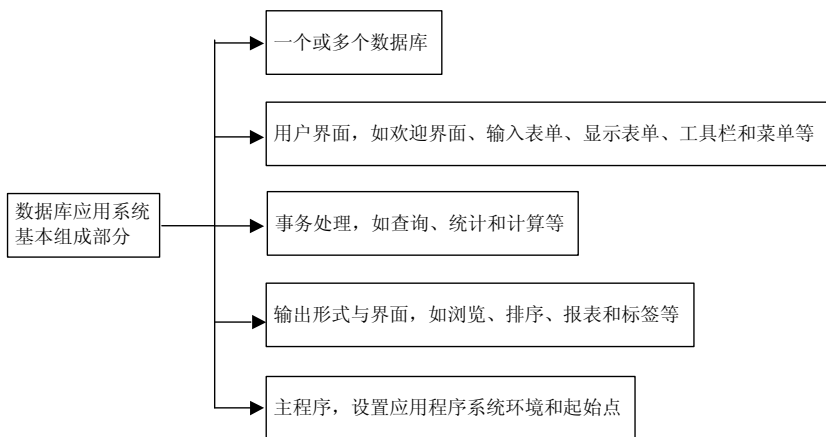


图 13-1 数据库应用系统基本组成部分

利用 Visual FoxPro 开发数据库应用系统的过程如图 13-2 所示。

1. 建立应用程序的目录结构

利用 Visual FoxPro 开发的数据库应用系统，无论规模大小，都包含了各种类型的文件，如数据库、数据表、表单、菜单、报表等文件。为了便于今后修改、维护，需要创建一个层次分明、结构清晰的文件目录结构，通过创建不同的文件夹，将各种不同类别的文件分类存放。

2. 用项目管理器组织应用系统

在利用 Visual FoxPro 设计应用程序时，首先要仔细设计每一个组件，可以每次只建立一部分组件，实现组件功能的同时确定与其他组件之间的关系，然后进行检验，最后当完成了所有的组件之后，就可以进行应用程序的集成和连编了。

“项目管理器”是 Visual FoxPro 开发数据库应用系统的工作平台，用项目管理器组织应用程序的过程如下：

- (1) 新建或打开已有的项目文件，如图 13-3 所示。
- (2) 将已经设计好的组件或模块添加到“项目管理器”中，如图 13-4 所示。

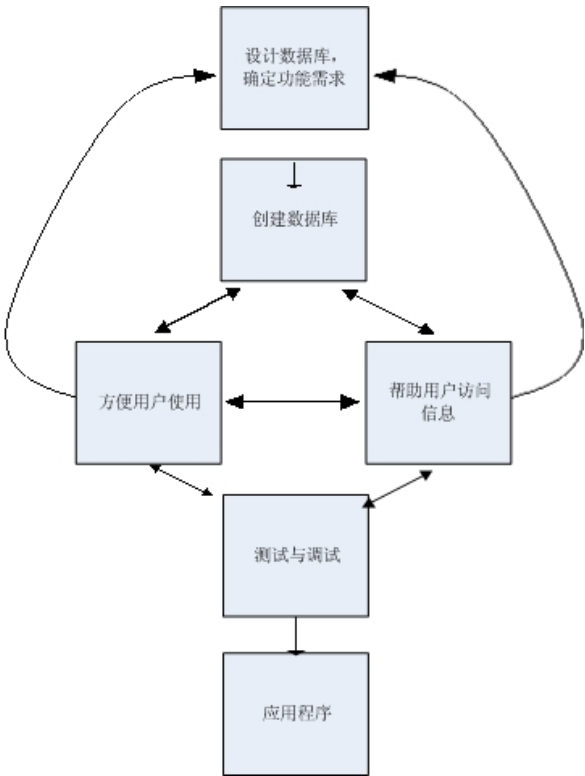


图 13-2 数据库应用系统的开发过程

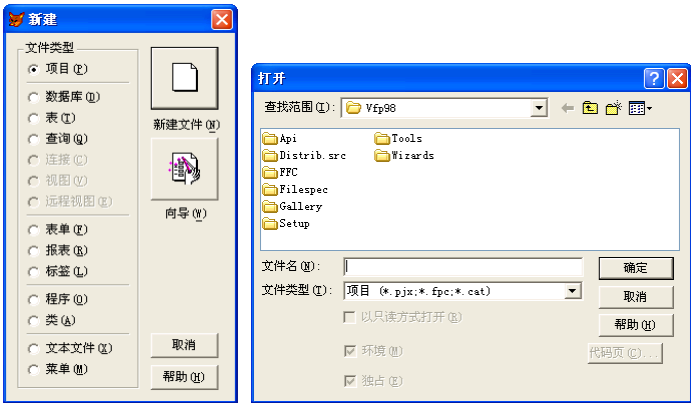


图 13-3 新建或打开已有的项目文件

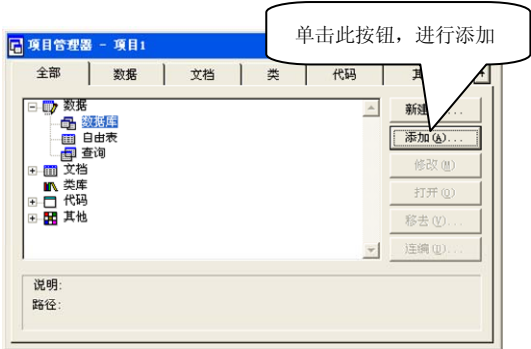


图 13-4 添加得到“项目管理器”

(3) 在“项目管理器”中调试各个组件或模块，采用自下而上、逐层调试的方法。例如，通过主表调用若干子表单，因此需要先调试各个子表单，最后再调试主表单。

对各个模块进行自下而上、逐层调试是为了便于错误代码的正确定位和修改，为了最后的连编工作做好必要的准备。

3. 加入项目信息

在主菜单 “项目” 菜单选择 “项目信息” 命令，可以打开 “项目信息” 对话框，如图 13-5 所示。

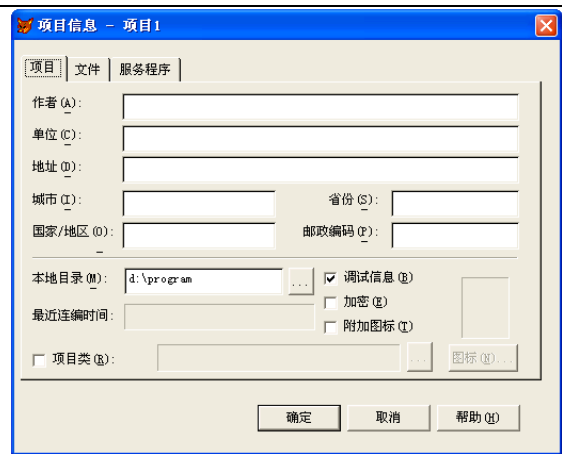


图 13-5 “项目信息”对话框

在“项目信息”对话框中，一共有 3 个选项卡，分别是“项目”选项卡、“文件”选项卡和“服务程序”选项卡。在“项目”选项卡中可以进行如下设置：

- (1) 开发者的姓名、单位、地址等信息。
- (2) 定位项目的主目录。
- (3) 在应用程序中是否包含调试信息。
- (4) 是否对应用程序进行加密。
- (5) 为所生成的文件指定新的图标。

在“文件”选项卡中可以进行如下设置：

- (1) 可以统一查看项目管理器中的所有文件，文件可以按照类型、名称或修改时间进行排列。
- (2) 可以更新本地代码页。
- (3) 当完成所有的设置工作后，单击“确定”按钮，则可以关闭“项目信息”对话框。

13.1.2 连编项目

对已经设计好的组件或模块进行调试确定无误后，就需要对整个项目进行联合调试和编译，称为“连编项目”。

1. 设置主程序

主程序是整个应用程序的入口点，任何应用程序都必须包含一个主程序文件，是应用程序首先要执行的程序文件。主程序设置的主要任务包括：

- (1) 设置应用程序的起始点。
- (2) 初始化环境。
- (3) 显示初始的用户界面。
- (4) 控制事件循环。
- (5) 当退出应用程序时，恢复原始开发环境。

在 Visual FoxPro 中，主程序可以是程序文件、菜单、表单或者查询，“项目管理器”中以黑体显示设置好的主文件名称。

设置主程序有两种方法：

- (1) 在“项目管理器”中选中要设置为主程序的文件名称，然后单击鼠标右键弹出快捷菜单，选择“设置主文件”命令，或者选择主菜单“项目”菜单中的“设置主文件”命令设置主程序，如图 13-6 所示。
- (2) 在“项目信息”的“文件”选项卡中选中要设置的主程序文件，单击鼠标右键弹出快捷菜单，选择“设置主文件”命令，如图 13-7 所示。

在利用 Visual FoxPro 开发的数据库应用系统中，主程序是唯一的，即应用系统中只有一个起始点，当重新设置主文件时，原来的主文件设置就自动解除。

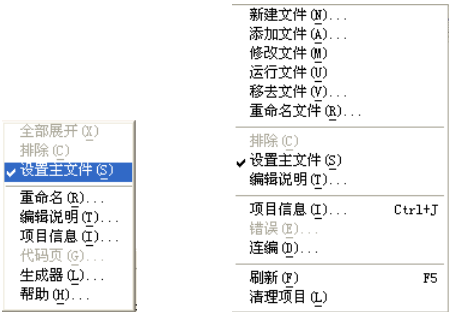


图 13-6 设置主文件方法 1

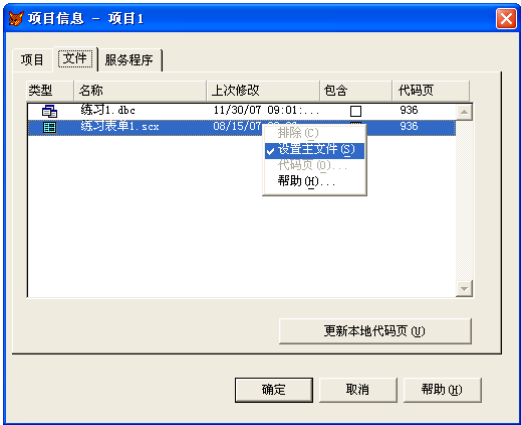


图 13-7 设置主文件方法 2

2. 设置文件的“排除”和“包含”

在如图 13-8 所示的“项目管理器”中，在数据库文件左侧有一个⊖符号，这是一个排除符号，表示该文件从项目中“排除”。

1) 文件的“排除”和“包含”

在项目管理器中，“排除”和“包含”是两个相互对应的操作。当对一个项目进行“连编”时，在项目中的所有标记为“包含”的文件将组合成为一个单一的应用程序文件，其中所有标记为“包含”的文件都变为只读文件，不能再进行任何修改。如果应用程序文件中包含需要修改的文件，就必须将这些文件标记为“排除”。

在 Visual FoxPro 中，被设置为主程序的文件会被项目管理器自动地设置为“包含”，在连编结束后，该文件变为只读文件。

2) 文件的“排除”和“包含”操作

在项目管理器中，文件的“包含”和“排除”操作如图 13-9、图 13-10 所示。

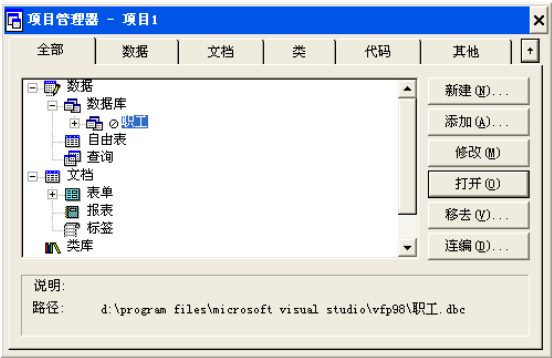


图 13-8 项目管理器

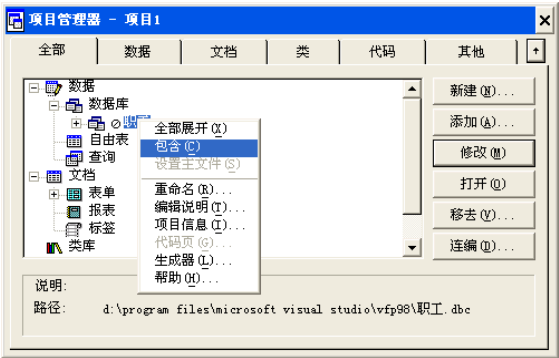


图 13-9 文件的“包含”操作

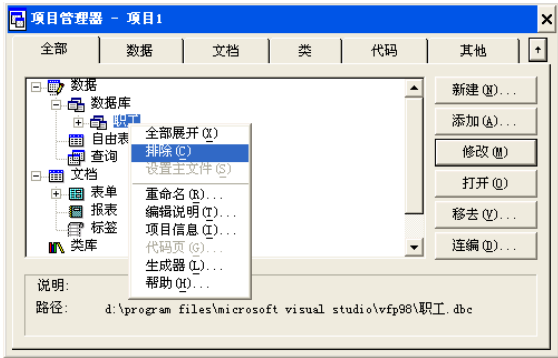


图 13-10 文件的“排除”操作

在项目管理器中，选用要标记为“包含”或者“排除”的文件，单击鼠标右键在弹出的快捷菜单中选择“包含”或

者“排除”命令。

3. 连编项目

项目连编的目的是为了将项目中所有引用的文件组合在一起，使其作为一个整体来协同工作，同时通过连编可以进一步发现错误、排除故障，最后生成一个完整的应用程序文件或可执行文件。

项目连编的最终结果是将项目中所有引用的文件，除去那些被标记为“排除”的文件外，合成为一个应用程序文件。最后将所有数据文件、应用程序文件，以及被标记为“排除”的文件一并交给最终用户。

在项目管理器中进行项目连编的步骤如下：

- (1) 选中被设置为主程序的文件，单击项目管理器中的“连编”按钮，弹出“连编选项”对话框，如图 13-11 所示。
- (2) 在“连编选项”对话框中，选中“重新连编项目”单选按钮，如图 13-11 所示。如果选中“显示错误”复选框，则可以查看错误文件。



图 13-11 在项目管理器中进行项目连编

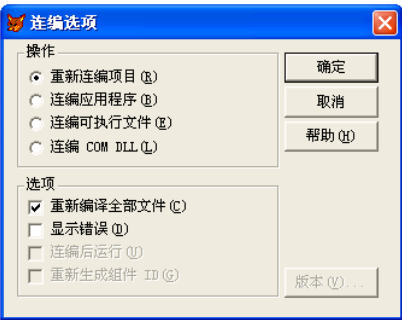


图 13-12 “连编选项”对话框

- (3) 选择所需要的选项后，单击“确定”按钮。
- 同时，项目连编也可以采用在命令窗口中输入“项目连编”命令，进行项目连编，如图 13-13 所示。

4. 连编的其他选项

在“连编选项”对话框中，还包括以下一些需要连编的其他选项内容：

- 1) 连编 COM DLL
利用项目文件中的类信息，创建一个动态链接库“.DLL”文件。
- 2) “版本”按钮
当选中“连编可执行文件”或“连编 COM DLL”单选按钮时，激活“版本”按钮，单击“版本”按钮，弹出“EXE 版本”对话框，在其中设置版本号、版本信息、版本类型等信息，如图 13-14 所示。

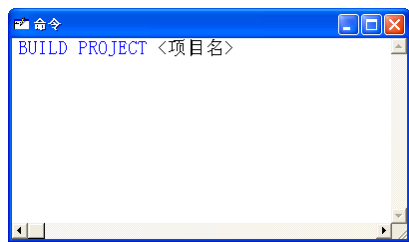


图 13-13 “连编选项”命令



图 13-14 “EXE 版本”对话框

- 3) 重新生成组件 ID

当选中“连编可执行文件”或“连编 COM DLL”单选按钮，同时已经连编了包含 OLE Public 关键字的程序时，该选项才可用。该选项在连编程序时用来指定新的 GUID（全局唯一标识）。

13.1.3 应用程序连编及运行

1. 连编应用程序

在 Visual FoxPro 中，成功进行项目连编之后，首先应该在建立应用程序之前试运行该项目，有以下两种方法：

- (1) 在“项目管理器”中，选中需要运行的主程序文件，单击“运行”按钮，如图 13-15 所示。
- (2) 在命令窗口输入“运行程序”命令，如图 13-16 所示。



图 13-15 在“项目管理器”中运行主程序

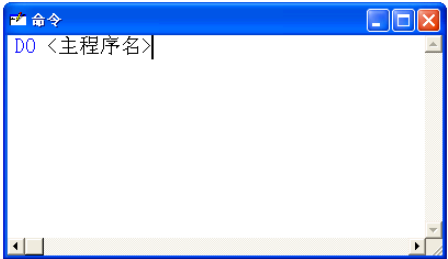


图 13-16 在“命令窗口”中运行主程序

如果程序运行正确，可以最终连编成一个应用程序文件。应用程序连编结果有以下两种文件形式：

- 应用程序文件 (.app)：需要在 Visual FoxPro 环境下运行。
- 可执行文件 (.exe)：可以脱离 Visual FoxPro 环境在 Windows 中运行。

连编应用程序的步骤如下：

- (1) 单击“项目管理器”对话框中的“连编”按钮，弹出“连编选项”对话框。
 - (2) 在“连编选项”对话框中，选中“连编应用程序”或者“连编可执行文件”单选按钮，单击“确定”按钮。
- 另外，也可以通过在“命令窗口”输入“连编应用程序”命令，连编应用程序，如图 13-17 所示。

2. 运行应用程序

当项目建立了一个最终的应用程序文件之后，就可以运行它了。

1) 运行 .app 应用程序

.app 应用程序文件必须在 Visual FoxPro 环境下运行。打开 Visual FoxPro 6.0 环境，在主菜单“程序”菜单中选择“运行”命令，弹出“运行”对话框，选择需要运行的程序文件，如图 13-18、图 13-19 所示。或者在“命令窗口”，输入 DO 命令运行 .app 应用程序文件，如图 13-20 所示。

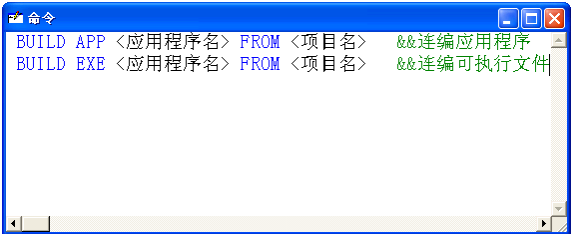


图 13-17 “连编应用程序”命令

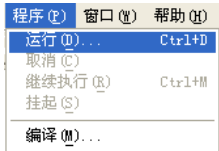


图 13-18 “程序”菜单中“运行”命令



图 13-19 “运行”对话框

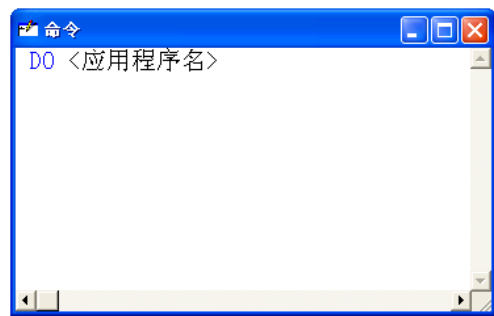


图 13-20 “运行应用程序”命令

2) 运行.exe 可执行文件

对于.exe 可执行文件，可以直接在 Windows 环境下双击该.exe 可执行文件图标运行。或者也可以在 Visual FoxPro 环境下，像运行.app 应用程序文件一样运行。

13.1.4 主程序设计

主程序是整个应用程序的入口点，包括以下功能：

- 初始化环境。
- 显示初始的用户界面。
- 控制事件循环。
- 恢复系统环境。

1. 初始化环境

主程序首先要对应用程序运行的系统环境进行初始化，初始化环境的方法是利用 SET 命令和设置系统变量的值。例如：

```
SET DEFAULT TO D:\产品管理
SET CENTURY ON
SET TALK OFF
SET SAFETY OFF
CLOSE ALL
CLEAR WINDOWS
CLEAR ALL
```

2. 显示初始的用户界面

通常，一个数据库应用系统在打开主窗口之前，首先会出现一个启动界面或者系统登录界面。在主程序中，调用初始用户界面的命令是：DO FORM <初始用户界面表单名>。

3. 控制事件循环

主程序在初始化系统环境、显示初始的用户界面之后，需要建立一个事件循环来处理用户的交互式请求。

在主程序中，设置控制事件循环的命令是：SET EVENTS 命令。该命令后的所有程序将被挂起，即暂时不被执行，直到出现结束当前事件循环的 CLEAR EVENTS 命令后，将控制权返回给 SET EVENTS 命令，SET EVENTS 命令后被挂起的程序才被继续执行。

4. 恢复系统环境

当退出应用程序时，需要恢复原来执行应用程序之前的开发环境。

13.2 使用应用程序生成器

在 Visual FoxPro 中，可以利用应用程序生成器向导生成一个项目和一个应用程序的基本框架，然后在应用程序设计器中，通过添加数据库、表、表单、报表和菜单等组件加以修改和完善。

13.2.1 使用应用程序向导

在 Visual FoxPro 中，创建项目有两种途径：

- (1) 利用应用程序向导。
- (2) 直接创建。

1. 使用应用程序向导创建项目和应用程序框架

启动“应用程序向导”的步骤如下：

- (1) 在主菜单“文件”菜单中选择“新建”命令，或者单击“常用”工具栏中的“新建”按钮，打开“新建”对话框。
- (2) 选中“项目”选项，单击“向导”按钮，弹出“应用程序向导”对话框，如图 13-21 所示。

(3) 在“应用程序向导”对话框中的“项目名称”文本框中输入项目名称，在“项目文件”文本框中指定项目文件所要存放的路径，系统默认与项目名称相同，用户可以加以改变。此外，也可以指定一个已经存在的路径存放该项目文件。另外，“创建项目目录结构”选项，如果选中该选项，系统在创建项目的同时在指定的项目目录中建立一系列子目录，用来分别存放不同类型的文件。

(4) 单击“应用程序向导”对话框中的“确定”按钮，系统便开始创建项目并进行一系列设置，并弹出“应用程序生成器”对话框，如图 13-22 所示。

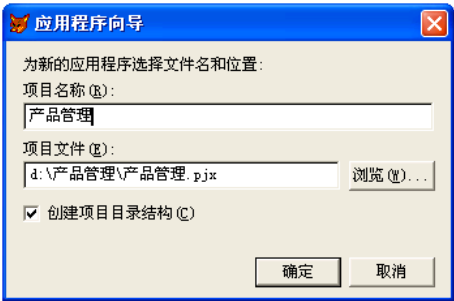


图 13-21 “应用程序向导”对话框

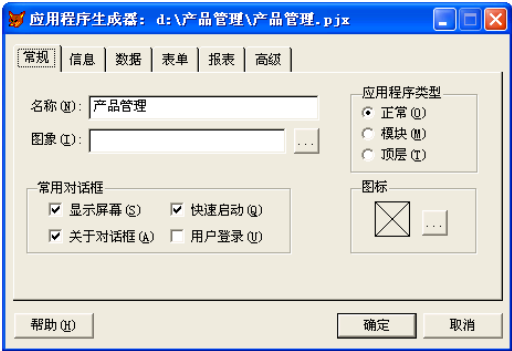


图 13-22 “应用程序生成器”对话框

2. 应用程序框架

单击“应用程序向导”对话框中的“确定”按钮，“应用程序向导”在弹出“应用程序生成器”对话框的同时，形成了一个应用程序框架，如图 13-23 所示。

今后可以利用这个应用程序框架，添加数据库、表、表单、菜单、报表等各类组件，或者直接创建新的组件。应用程序框架可以自动完成以下任务：

- (1) 显示菜单和工具栏。

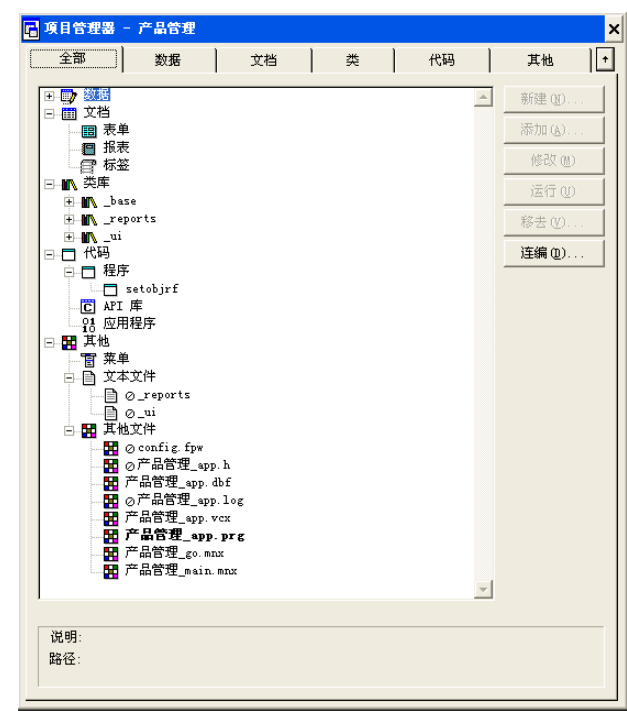


图 13-23 应用程序框架

13.2.2 应用程序生成器

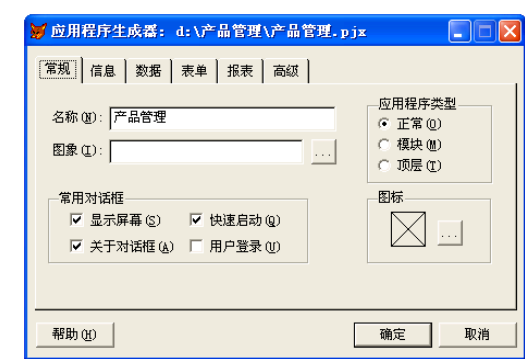


图 13-24 “常规”选项卡

- (2) 提供启动和清理程序，其中包括负责保存和恢复环境状态的程序。
- (3) 帮助开发者确定应用程序的功能、用户输入数据的方式、应用程序外观及其他强大的功能。

应用程序生成器具有以下功能：

- (1) 添加、编辑或删除与应用程序相关的组件。
- (2) 加入常用的应用程序元素，如启动界面、用户登录界面、“关于”对话框和“标准”工具栏等。
- (3) 设定表单和报表的外观、样式。
- (4) 提供应用程序的作者、版本信息。

在“应用程序生成器”对话框中，有 6 个选项卡：常规、信息、数据、表单、报表、高级。

1. “常规”选项卡

“常规”选项卡如图 13-24 所示，其中可以设置的选项和功能如表 13-1 所示。

表 13-1 “常规”选项卡选项及功能

选项	功能
名称	指定应用程序的名称，并且将显示在标题栏、“关于”对话框及整个应用程序中
图像	指定应用程序运行时，显示在启动界面、“关于”对话框中图像文件的文件名
应用程序类型	指定应用程序的运行方式，包括正常、模块和顶层
图标	指定一个在应用程序中使用的图标
常用对话框	选择应用程序中包含的对话框，包括显示屏幕、快速启动、关于对话框、用户登录

其中，应用程序类型中的 3 个类型分别为：

- 正常：表示生成在 Visual FoxPro 主窗口中运行的.app 应用程序。
- 模块：表示该应用程序能够被添加到已有的项目中，或者被其他程序调用。
- 顶层：表示生成一个.exe 可执行文件，脱离 Visual FoxPro 环境，在 Windows 中运行。

2. “信息”选项卡

“信息”选项卡可以规定应用程序的生产信息，如图 13-25 所示，其中可以设置的选项和功能如表 13-2 所示。

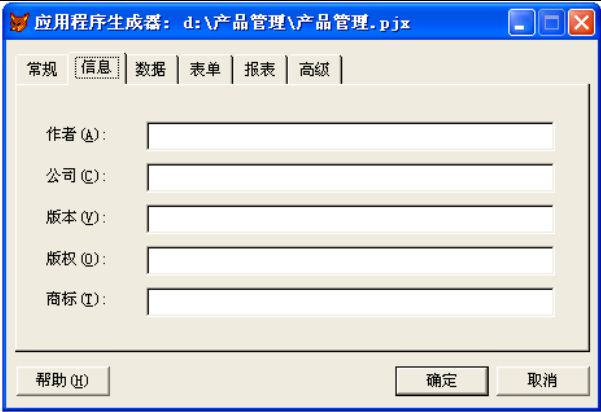


图 13-25 “信息”选项卡

表 13-2 “信息”选项卡选项及功能

选项	功能
作者	指定作者的名字
公司	指定编写或使用应用程序公司的名称
版本	指定应用程序的版本
版权	指定应用程序的版权信息
商标	指定商业或服务标志

3. “数据”选项卡

“数据”选项卡用于指定应用程序的数据源及表单、报表的样式，如图 13-26 所示，其中可以设置的选项和功能如表 13-3 所示。



图 13-26 “数据”选项卡

表 13-3 “数据”选项卡选项及功能

选项	功能
数据库向导	帮助创建应用程序所需要的数据库
数据表向导	帮助创建应用程序所需要的数据表
“选择”按钮	指定在应用程序中使用的已经存在的数据库或数据表
“清除”按钮	清除“数据源”表格中的数据库或数据表
“生成”按钮	根据“数据源”表格中的数据库或数据表按照指定样式生成表单或报表
表单样式	指定生成表单的样式
报表样式	指定生成报表的样式

4. “表单”选项卡

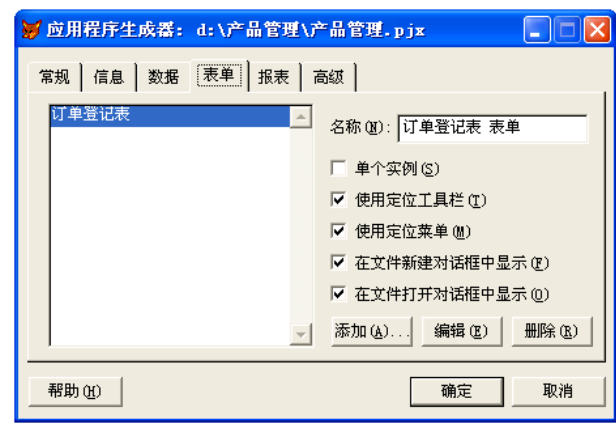


图 13-27 “表单”选项卡

“表单”选项卡用于指定菜单类型、启动表单的菜单、工具栏及是否可以具有多个实例，如图 13-27 所示，其中可以设置的选项和功能如表 13-4 所示。

表 13-4 “表单”选项卡选项及功能

选项	功能
名称	指定表单的名称
单个实例	指定在应用程序中是否只允许打开表单的一个实例
使用定位工具栏	指定应用程序生成器是否为表单附加定位工具栏
使用定位菜单	指定应用程序生成器是否为表单附加定位菜单
在文件新建对话框中显示	指定表单名称是否在应用程序生成器“新建”对话框中出现
在文件打开对话框中显示	指定表单名称是否在应用程序生成器“打开”对话框中出现
“添加”按钮	用于将已有表单添加到应用程序中
“编辑”按钮	用于调用“表单设计器”修改表单
“删除”按钮	用于从应用程序中删除表单

5. “报表”选项卡

“报表”选项卡用于指定在应用程序中使用报表的名称，如图 13-28 所示，其中可以设置的选项和功能如表 13-5 所示。

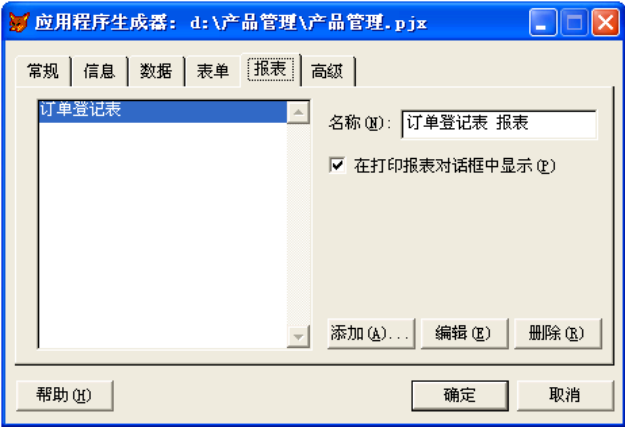


图 13-28 “报表”选项卡

表 13-5 “报表”选项卡选项及功能

选项	功能
名称	指定报表的名称
在打印报表对话框中显示	指定报表名称是否在应用程序生成器的“打印报表”对话框中出现
“添加”按钮	用于将已有报表添加到应用程序中
“编辑”按钮	用于调用“报表设计器”修改报表
“删除”按钮	用于从应用程序中删除报表

6. “高级”选项卡

“高级”选项卡用于指定帮助文件和应用程序的默认目录，以及应用程序是否包含“常用”工具栏和“收藏夹”菜单，如图 13-29 所示，其中可以设置的选项和功能如表 13-6 所示。

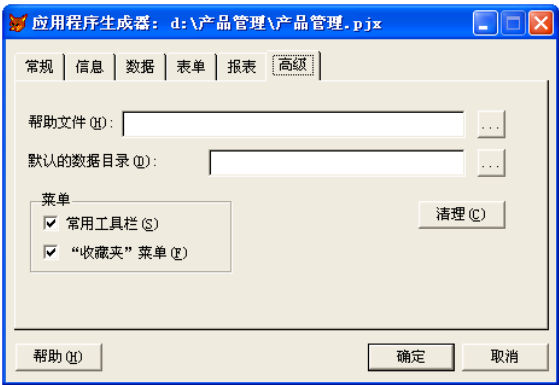


图 13-29 “高级”选项卡

表 13-6 “高级”选项卡选项及功能

选项	功能
帮助文件	指定应用程序帮助文件的名称和路径
默认的数据目录	指定应用程序数据文件的默认目录
常用工具栏	指定应用程序中是否具有常用工具栏
“收藏夹”菜单	指定应用程序中是否具有“收藏夹”菜单
“清理”按钮	使“应用程序生成器”中所做的修改与当前活动项目保持一致

13.3 本章习题

13.3.1 选择题

1. 如果添加到项目中的文件标志为“排除”，则表示（ ）。
- A. 此类文件不是应用程序的一部分
- B. 生成应用程序时不包括此类文件
- C. 生成应用程序时包括此类文件，用户可以修改
- D. 生成应用程序时包括此类文件，用户不能修改
2. 连编应用程序不能生成的文件是（ ）。
- A. APP 文件
- B. EXE 文件
- C. DLL 文件
- D. PRG 文件
3. 有关连编应用程序，下面描述正确的是（ ）。

- A. 项目连编以后将主文件视为只读文件
 - B. 一个项目中可以有多个主文件
 - C. 数据库文件可以被指定为主文件
 - D. 在项目管理器中文件名左侧带有符号Φ的文件在项目连编以后是只读文件
4. 向一个项目中添加一个数据库，应该使用项目管理器的（ ）。
- A. “代码”选项卡
 - B. “类”选项卡
 - C. “文档”选项卡
 - D. “数据”选项卡
5. 把一个项目编译成一个应用程序时，下面的叙述正确的是（ ）。
- A. 所有项目的文件将组合为一个单一的应用程序文件
 - B. 所有项目包含的文件将组合为一个单一的应用程序文件
 - C. 所有项目排除的文件将组合为一个单一的应用程序文件
 - D. 由用户选定的项目文件将组合为一个单一的应用程序文件
6. 下面关于运行应用程序的说法正确的是（ ）。
- A. .app 应用程序可以在 Visual FoxPro 和 Windows 环境下运行
 - B. .exe 只能在 Windows 环境下运行
 - C. .exe 应用程序可以在 Visual FoxPro 和 Windows 环境下运行
 - D. .app 应用程序可只能在 Windows 环境下运行
7. 作为整个应用程序入口点的主程序至少应具有以下功能（ ）。
- A. 初始化环境
 - B. 初始化环境、显示初始用户界面
 - C. 初始化环境、显示初始用户界面、控制事件循环
 - D. 初始化环境、显示初始用户界面、控制事件循环、退出时恢复环境
8. 在应用程序生成器的“数据”选项卡中可以（ ）。
- A. 为表生成一个表单和报表，并可以选择样式
 - B. 为多个表生成的表单必须有相同的样式
 - C. 为多个表生成的报表必须有相同的样式
 - D. 只能选择数据源，但不能创建它

13.3.2 填空题

1. 在项目管理器中，文件的_____和包含是相对的，将一个文件编译成一个应用程序时，该项目中标记为包含的文件将成为只读文件。
2. _____是整个应用程序的入口点，且在一个项目管理器中，只能设置一个，并在项目管理器中以黑体显示。其首要任务是对应用程序的环境进行初始化。
3. 连编项目可直接在命令窗口中输入_____命令进行连编，连编项目获得成功后，直接使用命令 DO <主程序名>，运行该项目。
4. 当连编应用程序时，如果选择连编生成可执行程序，则生成的文件的扩展名是_____。
5. 将一个项目编译成一个应用程序时，如果应用程序中包含需要用户修改的文件，必须将该文件标为_____。
6. 如果项目不是用“应用程序向导”创建的，启动应用程序生成器后可用的选项卡只有_____、“表单”和“报表”3个选项卡。
7. 在生成应用程序时，当把表设计为排除时，_____修改表中的数据。
8. 使用“应用程序向导”创建的项目，除项目外还自动生成一个_____。
9. 在应用程序生成器的“常规”选项卡中，选择程序类型时选中“顶层”，将生成一个_____。

第 14 章

上机模拟试题与解析

为了更好地让考生在应考之前全面了解和掌握上机考试系统环境及模式，熟练操作上机考试系统，提供应试能力，本章将主要介绍如何使用上机考试系统，以及二级 Visual FoxPro 数据库程序设计上机考试的技巧、上机模拟试题及试题解析。

14.1 上机应试技巧

全国计算机等级考试上机考试系统专用软件是在中文版 Windows XP 平台下开发的应用软件，它提供了开放式的考试环境，具有自动计时、断点保护、自动阅卷和自动回收等功能。

14.1.1 上机考试纪律

(1) 上机考试时，考生应在规定的考试时间提前 30 分钟报到，交验准考证和身份证（军人身份证或户口本），同时抽签决定上机考试的工作站号（或微机号）。

(2) 考生提前 5 分钟进入机房，坐在由抽签决定的工作站号（或微机号）上，不允许乱坐位置。

(3) 不得擅自登录与自己无关的考号。

(4) 不得擅自复制或删除与自己无关的目录和文件。

(5) 考生不得在考场中交头接耳、大声喧哗。

(6) 未到 10 分钟不得离开考场。

(7) 迟到 10 分钟者取消考试资格。

(8) 考试中计算机出现故障、死机、死循环、电源故障等异常情况（即无法进行正常考试时），应举手示意与监考人员联系，不得擅自关机。

(9) 考生答题完毕后应立即离开考场，不得干扰其他考生答题。

14.1.2 上机考试系统说明

1. 上机考试环境

1) 硬件环境

主机：主频 1GHz 以上各种兼容机。

内存：512MB 以上基本内存。

硬盘空间：500MB 以上剩余空间。

2) 软件环境

操作系统：中文版 Windows XP。

语言环境：中文版 Microsoft Visual FoxPro 6.0 和 MSDN 6.0。

2. 上机考试时间

全国计算机等级考试二级 Visual FoxPro 数据库程序设计上机考试时间为 90 分钟。考试时间由上机考试系统自动进行计时，提前 5 分钟自动报警来提醒考生应及时存盘，考试时间用完，上机考试系统将自动锁定计算机，考生将不能继续进行考试。

当考生登录成功后，系统将自动抽取考题并在屏幕上显示上机须知，提示考生按【S】键开始考试，系统开始计时；如果是二次登录，则系统将累计计时，考生必须在规定的时间内完成考试内容。当考生超出考试所用时间时机器将自动关闭；当考试只剩下指定时间时，屏幕上会自动报告所剩考试时间，此时考生只需按任意键继续答题，不会影响考生成绩。

3. 上机考试题型及分值

全国计算机等级考试二级 Visual FoxPro 数据库程序设计上机考试满分为 100 分，考试题目类型目前有 3 种：基本操作题（30 分，包括 4 小题）、简单应用题（40 分，包括 2 小题）和综合应用题（30 分，包括 1 小题）。

14.1.3 上机考试步骤

考试过程分为登录、看题、答题、交卷等几个阶段。

1. 上机考试登录

(1) 根据考试要求打开考试系统软件，启动考试程序，出现如图 14-1 所示的登录界面。

(2) 在实际答题之前，需要先进行考试系统登录。一方面，这是考生姓名的记录凭据，系统需要验证考生的姓名和准考证号是否与本人相符；另一方面，考试系统也需要对每一位考生随机选择一套试题，生成试卷。单击图 14-1 中的“开始登录”按钮或按回车键，出现准考证号登录界面，如图 14-2 所示。



图 14-1 登录界面

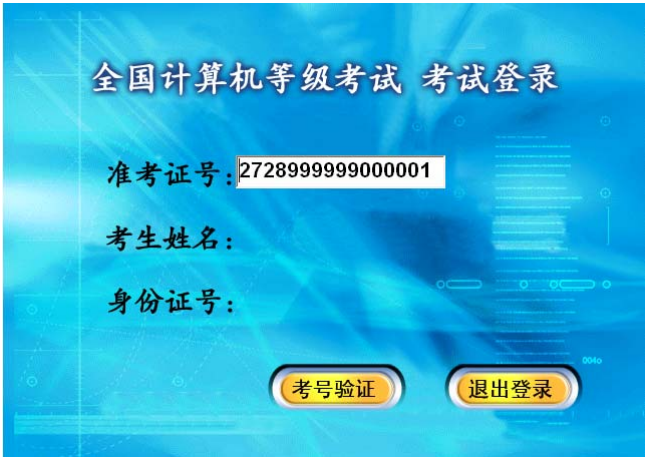


图 14-2 准考证号登录界面

(3) 在图 14-2 中输入正确的准考证号（必须是 16 位数字），再单击“考号验证”按钮或按回车键对输入的准考证号与考生姓名、身份证号进行验证。

(4) 当输入的准考证号不存在时，上机考试系统会显示相应的提示信息，并单击“否”按钮要求考生重新输入准考证号，直至输入正确或单击“是”按钮退出上机考试登录系统为止。

(5) 如果输入的准考证号存在，则屏幕显示准考证号所对应的身份证号和考生姓名并显示相应的应答提示信息，如图 14-3 所示。

(6) 由考生核对自己的姓名和身份证号，如果发现准考证号不正确，则单

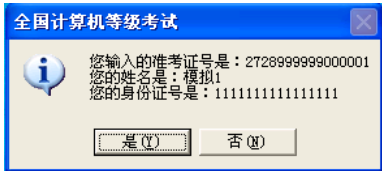


图 14-3 准考证号验证界面

击“否”按钮，重新输入准考证号。上机考试系统最多允许考生输入准考证号 3 次。如果均不正确，则请主考或监考老师帮助查找原因，给予更正。如果输入的准考证号核对后正确，则请考生单击“是”按钮。

(7) 接着上机考试系统进行一系列处理后将随机生成一份二级 Visual FoxPro 上机考试的试卷，并显示提示信息。如果上机考试系统在抽取试题过程中产生错误并显示相应的错误提示信息时，考生应重新进行登录直至试题抽取成功为止。如果准考证号正确，系统将出现“考试须知”窗口，如图 14-4 所示。

(8) 单击“考试须知”窗口上的“开始答题并计时”按钮，进入考试界面，就可以看题、做题，并开始计时。

2. 查看题目与寻求系统帮助

当考生登录成功后，上机考试系统将自动在屏幕中间生成装载试题内容查阅工具的考试窗口，如图 14-5 所示。同时在屏幕顶部始终显示考生的准考证号、姓名、考试剩余时间，以及可以随时显示或隐藏试题内容查阅工具和退出考试系统进行交卷按钮的窗口，如图 14-6 所示。



图 14-4 考试须知



图 14-5 考试窗口



图 14-6 考试工具窗口

1) 查看题目要求

对于二级 Visual FoxPro 考试，在考试窗口中选择工具栏中的题目，单击“基本操作题”、“简单应用题”、“综合应用题”按钮可以查看相应题型的题目要求，如图 14-7、图 14-8、图 14-9 所示。

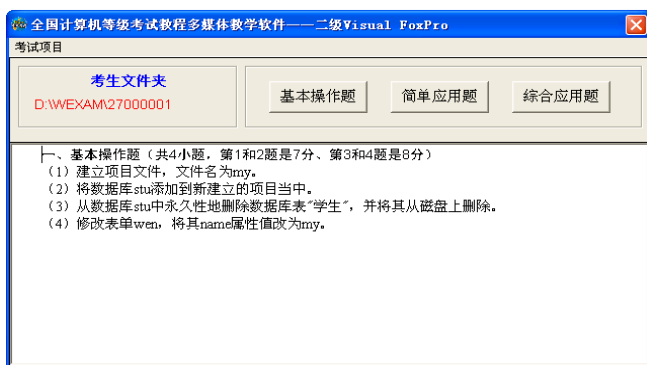


图 14-7 基本操作题

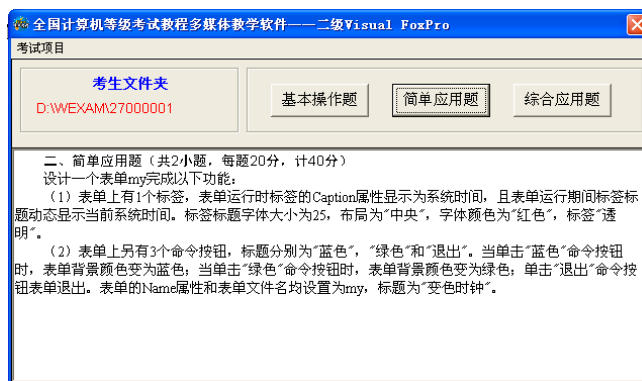


图 14-8 简单应用题

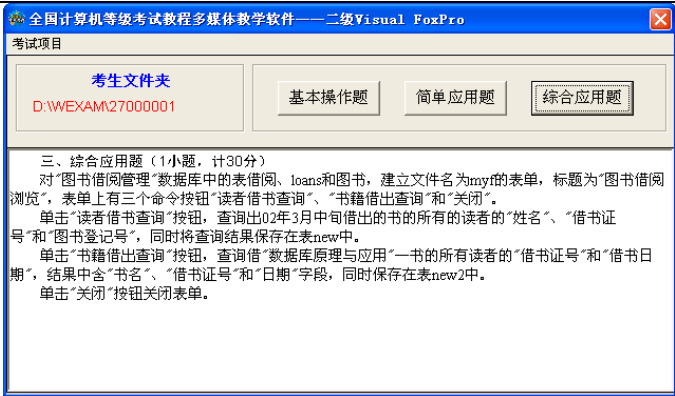


图 14-9 综合应用题

- 2) 寻求系统帮助
- 在“帮助”菜单栏中选择“等级考试系统帮助”命令可以启动考试帮助系统，并显示考试系统的使用说明及注意事项。
3. 答题

当考生登录成功后，上机考试系统将会自动产生一个考生文件夹，该文件夹将存放该考生所有上机考试的考试内容。考生不能随意删除该文件夹，以及该文件夹下与考试题目要求有关的文件及子文件夹。假设考生登录的准考证号为280199990001，则上机考试系统生成的考生文件夹（由准考证号的前4位数字和最后4位数字组成）将存放至上机考试系统所安装盘符的根目录下（假设上机考试系统软件安装在E盘）的WEXAM文件夹下，即考生文件夹为E:\WEXAM\28010001。在考试界面的菜单栏下，左边的区域就显示出考生文件夹路径。

考生在考试过程中所操作的文件和文件夹都不能脱离考生文件夹，否则将会直接影响考生的考试成绩。

若考生在上机考试时遇到死机等意外情况（即无法进行正常考试），考生应向监考老师说明情况，由监考老师确认为非人为造成停机时，方可进行二次登录。考生需由监考老师输入密码方可继续进行上机考试，如图14-10所示。因此考生必须注意在上机考试时不得随意关机，否则考点将有权终止其考试资格。

当系统接受考生的准考证号并显示出姓名和身份证号时，考生确认是否相符，一旦考生确认，则系统给出提示。



图 14-10 二次登录“输入密码”窗口

4. 交卷
- 如果考生要提前结束考试进行交卷处理，则请在屏幕顶部的状态窗口中单击“交卷”按钮，上机考试系统将显示是否要交卷处理的提示信息框，如图14-11所示。
- 此时考生如果单击“确定”按钮，则退出上机考试系统进行交卷处理，由系统管理员进行评分和回收。如果考生还没有做完试题，则单击“取消”按钮继续进行考试。当交卷处理时，系统首先锁住屏幕，并显示“系统正在进行交卷处理，请稍候！”。当系统完成了交卷，在屏幕上显示“交卷正常，请监考老师输入结束密码：”或“交卷异常，请监考老师输入结束密码：”时，这时只要输入结束密码便可结束考试。这个过程不会删除考生文件夹中的任何数据。如果出现“交卷异常”的提示，则说明这个考生有可能得零分或者考生文件夹有问题，要检查这个考生的实际考试情况是否正常。如果在交卷过程中死机，可以重新启动计算机，进行二次登录后再进行“交卷”处理。

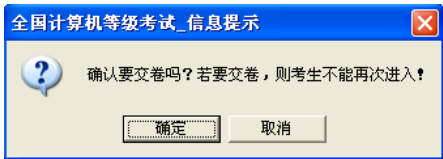


图 14-11 交卷处理提示

在考试过程中，系统会为考生计算剩余考试时间。在剩余5分钟时，系统会显示一个提示信息。

考试时间用完后，系统会锁住计算机并提示输入“延时”密码。这时需要输入延时密码才能解锁计算机并恢复考试

界面，考试系统会自动再运行 5 分钟，这时可以交卷。如果没有进行交卷处理，考试系统运行到 5 分钟后，系统又会锁住计算机并提示输入“延时”密码。只要不进行“交卷”处理，就可以“延时”几次。

当上机考试系统提示“考试时间已到，请停止答卷”后，此时考生要特别注意，所操作的考试内容是否已经存盘。如果考生擅自关机或启动机器，将会直接影响考生自己的考试成绩。

考生完成答题过程，退出考试系统后，监考人员必须进行评分和考生成绩的回收。

14.2 上机模拟试题一

1. 基本操作题（共 4 个小题，第 1 和 2 题是 7 分，第 3 和 4 题是 8 分）

在考生目录下完成如下操作。

(1) 打开项目文件 salary_p，将考生目录下的数据库 salary_db 添加到该项目中。

(2) 在 salary_db 数据库中为 dept 表创建一个主索引（升序），索引名和索引表达式均是“部门号”；为 salary 表创建一个普通索引（升序），索引名和索引表达式均是“部门号”，再创建一个主索引（升序），索引名和索引表达式均是“雇员号”。

(3) 通过“部门号”字段建立 salary 表和 dept 表间的永久联系。

(4) 为以上建立的联系设置参照完整性约束：更新规则为“限制”；删除规则为“级联”；插入规则为“限制”。

2. 简单应用题（2 小题，每题 20 分，计 40 分）

在考生目录下完成如下简单应用。

(1) 创建一个名称为 svview 的视图，该视图的 SELECT 语句查询 salary_db 数据库中 salary 表（雇员工资表）的部门号、雇员号、姓名、工资、补贴、奖励、失业保险、医疗统筹和实发工资，其中实发工资由工资、补贴和奖励 3 项相加，然后再减去失业保险和医疗统筹得出，结果按“部门号”降序排序，最后将定义视图的命令代码存放到命令文件 t1.prg 中并执行该程序。

(2) 设计一个名称为 FORM11 的表单，表单以表格方式（与 BROWSE 窗口方式相似，表格名称为 graSalary）显示 salary_db 数据库中 salary 表的记录，供用户浏览。在该表单的右下方有一个命令按钮，名称为 Command1，标题为“退出浏览”，当单击该按钮时退出表单。

3. 综合应用题（1 小题，计 30 分）

在考生目录下，对数据库 salary_db 完成如下综合应用。

设计一个名称为 FORM12 的表单，在表单上设计一个“选项组”（又称选项按钮组，名称为 Optiongroup1）及两个命令按钮“生成”（名称为 Command1）和“退出”（名称为 Command2）；其中选项按钮组有“雇员工资表”（名称为 Option1）、“部门表”（名称为 Option2）和“部门工资汇总表”（名称为 Option3）3 个选项按钮。然后为表单建立数据环境，并向数据环境添加 dept 表（名称为 Cursor1）和 salary 表（名称为 Cursor2）。

各选项按钮功能如下：

(1) 当用户选择“雇员工资表”选项按钮后，再单击“生成”命令按钮，查询显示在简单应用中创建的 svview 视图中的所有信息，并把结果存入表 gz1.dbf 中。

(2) 当用户选择“部门表”选项按钮后，再单击“生成”命令按钮，查询显示 dept 表中每个部门的部门号和部门名称，并把结果存入表 bm1.dbf 中。

(3) 当用户选择“部门工资汇总表”选项按钮后，再单击“生成”命令按钮，则按部门汇总，将该公司的部门号、部门名、工资、补贴、奖励、失业保险和医疗统筹的支出汇总，合计结果存入表 hz1.dbf 中。请注意：字段名必须与原字段名一致。

(4) 单击“退出”按钮，退出表单。

14.3 上机模拟试题二

1. 基本操作题（共 4 个小题，第 1 和 2 题是 7 分，第 3 和 4 题是 8 分）

在考生目录下的“商品销售”数据库中完成如下操作。

- (1) 将“销售表”中的日期在 2005 年 12 月 31 日前（含 2005 年 12 月 31 日）的记录复制到一个新表“销售表 2005.dbf”中。
- (2) 将“销售表”中的日期（日期型字段）在 2005 年 12 月 31 日前（含 2005 年 12 月 31 日）的记录物理删除。
- (3) 打开“商品表”，使用 BROWSE 命令浏览时，使用“文件”菜单中的命令将“商品表”中的记录生成文件名为“商品表.htm”的 HTML 格式文件。
- (4) 为“商品表”创建一个主索引，索引名和索引表达式均是“商品号”；为“销售表”创建一个普通索引（升序），索引名和索引表达式均是“商品号”。

2. 简单应用题（2 小题，每题 20 分，计 40 分）

在考生目录下完成如下简单应用。

- (1) 在“商品销售”数据库中，根据“销售表”和“商品表”查询每种商品的商品号、商品名、单价、销售数量和销售金额（商品号、商品名取自商品表，单价和销售数量取自销售表，销售金额=单价×销售数量），结果按销售金额降序排序，并将查询结果存储到 see_a 表中。
- (2) 考生目录下有一个 FORM21 表单文件，其中 3 个命令按钮 Click 事件下的语句都是错误的，请按如下要求进行修改（最后保存所做的修改）。
 - ①单击“刷新标题”命令按钮时，将表单的标题改为“商品销售数据输入”。
 - ②单击“商品销售输入”命令按钮时，调用当前文件夹下名称为 sellcomm 的表单文件打开数据输入表单。
 - ③单击“输出销售报表”命令按钮时，调用当前文件夹下名称为 print1 的报表文件对报表进行预览。

3. 综合应用题（1 小题，计 30 分）

在考生目录下，对“商品销售”数据库完成如下综合应用。

- (1) 请编写名称为 change_c 的命令程序并执行，该程序实现下面的功能。
 - ①将“商品表”进行备份，备份文件名为“商品表备份.dbf”。
 - ②将“商品表”中“商品号”第一位编号为“4”的商品的“单价”修改为原单价的 10%
 - ③使用“单价调整表”对商品表的部分商品出厂单价进行修改（按“商品号”相同）。
- (2)设计一个名称为 FORM22 的表单,上面有 3 个命令按钮:“调整”(名称为 Command1)和“退出”(名称为 Command2)两个命令按钮。
 - ①单击“调整”命令按钮时，调用 change_c 命令程序实现商品单价调整。
 - ②单击“退出”命令按钮时，关闭表单。

14.4 上机模拟试题三

1. 基本操作题（共 4 个小题，第 1 和 2 题是 7 分，第 3 和 4 题是 8 分）

在考生目录下的“salary_db”数据库中完成如下操作。

- (1) 在数据库 salary_db 中建立表 dept，表结构如下：

字段名	类型	宽度
部门号	字符型	2
部门名	字符型	20

随后在表中输入 5 条记录，记录内容如下：

部门号	部门名
01	制造部
02	销售部
03	项目部
04	采购部
05	人事部

(2) 为 dept 表创建一个主索引（升序），索引名和索引表达式均是“部门号”。

(3) 通过“部门号”字段建立 salary 表和 dept 表间的永久联系。

(4) 为以上建立的联系设置参照完整性约束：更新规则为“级联”；删除规则为“限制”；插入规则为“限制”。

2. 简单应用题（2 小题，每题 20 分，计 40 分）

在考生目录下完成如下简单应用。

(1) 使用报表向导建立一个简单报表。要求选择 salary 表中所有字段；记录不分组；报表样式为“随意式”；列数为“1”，字段布局为“列”，方向为“纵向”；排序字段为“雇员号”（升序）；报表标题为“雇员工资一览表”；报表文件名为 print2。

(2) 考生目录下有一个 FORM31 表单文件，其中有两个命令按钮。请按如下要求进行设计（最后保存所做的设计）。

①将按钮“刘缆雇员工资”名称修改为“浏览雇员工资”。

②单击“浏览雇员工资”命令按钮时，使用 SELECT 命令查询 salary 表中所有字段信息供用户浏览。

③单击“退出表单”命令按钮时，关闭表单。

3. 综合应用题（1 小题，计 30 分）

(1) 编写名称为 change_c 的命令程序并执行，该程序实现下面的功能。

①将雇员工资表 salary 进行备份，备份文件名为“bak_salary.dbf”。

②利用“人事部”向“财务部”提供的雇员工资调整表 c_salary1 的“工资”，对 salary 表的“工资”进行调整（请注意：按“雇员号”相同进行调整，并且只是部分雇员的工资进行了调整，其他雇员的工资不动）。

(2) 设计一个名称为 FORM32 的表单，上面有“调整”（名称为 Command1）和“退出”（名称为 Command2）两个命令按钮。

①单击“调整”命令按钮时，调用 change_c 命令程序实现工资调整。

②单击“退出”命令按钮时，关闭表单。

14.5 上机模拟试题四

1. 基本操作题（共 4 个小题，第 1 和 2 题是 7 分，第 3 和 4 题是 8 分）

在考生目录下的“订货管理”数据库中完成如下操作。

(1) 将 order_detail、order_list 和 customer 表添加到数据库。

(2) 为 order_list 表创建一个主索引，索引名和索引表达式均是“订单号”。

(3) 建立表 order_list 和表 order_detail 间的永久联系（通过“订单号”字段）。

(4) 为以上建立的联系设置参照完整性约束：更新规则为“限制”，删除规则为“级联”，插入规则为“限制”。

2. 简单应用题（2 小题，每题 20 分，计 40 分）
- 在考生目录下完成如下简单应用。
- (1) 列出客户名为“三川实业有限公司”的订单明细（order_detail）记录（将结果先按“订单号”升序排序，同一订单的再按“单价”降序排列），并将结果存储到 result 表中（表结构与 order_detail 表结构相同）。

(2) 考生目录下有一个 FORM41 表单文件，其中有 3 个命令按钮，请按如下要求进行设计（最后保存所做的设计）。

①单击“刷新标题”命令按钮时，使表单的标题为“简单应用”。

②单击“订单记录”命令按钮时，使表格控件中显示 order_list 表中的记录。

③单击“关闭表单”命令按钮时，关闭表单。
3. 综合应用题（1 小题，计 30 分）
- 首先为 order_detail 表增加一个新字段：新单价（类型与原来的单价字段相同），然后编写满足如下要求的程序。
- 根据 order_list 表中的“订购日期”字段的值确定 order_detail 表的“新单价”字段的值，原则是：订购日期为 2007 年的“新单价”字段的值为原单价的 90%，订购日期为 2006 年的“新单价”字段的值为原单价的 110%（注意：在修改操作过程中不要改变 order_detail 表记录的顺序），最后将程序保存为 prog1.prg，并执行该程序。
- 接着再利用 Visual FoxPro 的“快速报表”功能建立一个简单报表，该报表内容按顺序含有 order_detail 表的订单号、产品号、产品名、新单价和数量字段的值，并将报表文件保存为 report1.frx。

14.6 上机模拟试题五

1. 基本操作题（共 4 个小题，第 1 和 2 题是 7 分，第 3 和 4 题是 8 分）
- 在考生目录下完成如下操作。
- (1) 创建一个新的项目“客户管理”。

(2) 在新建的项目“客户管理”中创建数据库“订货管理”。

(3) 在“订货管理”数据库中建立表 order_list，表结构如下：
字段名	类型	宽度
客户号	字符型	6
订单号	字符型	6
订购日期	日期型	
总金额	浮动型	15, 2
- (4) 为 order_list 表创建一个索引，索引名和索引表达式均是“订单号”。
2. 简单应用题（2 小题，每题 20 分，计 40 分）
- 在考生目录下完成如下简单应用。
- (1) 将 order_list1 表中的全部记录追加到 order_list 表中，然后用 SQL SELECT 语句完成查询，按总金额降序列出所有客户的客户号、客户名及其订单号和总金额，并将结果存储到 result 表中（其中客户号、客户名取自 customer 表，订单号、总金额取自 order_list 表）。

(2) 打开 FORM51 表单，并按如下要求进行设计（注意：最后保存所做的设计）。

①表单中有 5 个随机排列的命令按钮，不要移动或改变“基准按钮”的位置（否则影响成绩），然后使其他命令按钮与“基准按钮”左对齐。

②在这组命令按钮的右边添加一个表格控件，并将它的 RecordSourceType 属性设置为“表”，然后设置另一个相关属性使在表格控件中显示 customer 表的记录。

3. 综合应用题 (1 小题, 计 30 分)

当 order_detail 表中的单价修改后, 应该根据该表中“单价”和“数量”字段修改 order_list 表的总金额字段, 现在编写程序实现此功能。具体要求和注意事项如下:

- (1) 根据 order_detail 表中的记录重新计算 order_list 表的总金额字段的值。
- (2) 一条 order_list 记录可以对应几条 order_detail 记录。
- (3) 在编程前应保证在基础操作中正确地建立了 order_list 表, 在简单应用中为该表追加了记录 (注意只能追加一次)。
- (4) 最后将 order_list 表中的记录按总金额降序排序并存储到 od_new 表中 (表结构与 order_list 表完全相同)。
- (5) 将程序保存为 prog1.prg 文件。

14.7 上机模拟试题一答案与解析

1. 基本操作题

试题 1

- (1) 启动 Visual FoxPro。
- (2) 从“文件”菜单中选择“打开”命令。
- (3) 在“打开”对话框中, 选择考生目录下的项目文件“salary_p.pjx”, 单击“确定”按钮。
- (4) 在“项目管理器”对话框中, 选择“数据”选项卡, 再选择“数据库”选项, 单击“添加”按钮。
- (5) 在“打开”对话框中, 选择考生目录下的“salary_db.dbc”文件, 单击“确定”按钮。

试题 2

- (1) 在“项目管理器”对话框的“数据”选项卡中选择“数据库”选项, 再选择“salary_db”, 单击“修改”按钮。
- (2) 在“数据库设计器”中, 用鼠标右键单击表“dept”, 选择“修改”命令。
- (3) 在“表设计器”中, 选择“索引”选项卡。
- (4) 在“索引名”栏输入“部门号”, “排序”栏选择“升序”, “类型”栏选择“主索引”, “表达式”栏输入“部门号”, 单击“确定”按钮。
- (5) 在“数据库设计器”中, 用鼠标右键单击表“salary”, 选择“修改”命令。
- (6) 在“表设计器”中, 选择“索引”选项卡。
- (7) 在“索引名”栏输入“部门号”, “排序”栏选择“升序”, “类型”栏选择“普通索引”, “表达式”栏输入“部门号”。
- (8) 在“索引名”栏输入“雇员号”, “排序”栏选择“升序”, “类型”栏选择“普通索引”, “表达式”栏输入“雇员号”, 单击“确定”按钮。

试题 3

- (1) 在“数据库设计器”中, 将“dept”表中的“索引”部门号拖放到“salary”表中的“索引”部门号上。
- (2) 用鼠标右键单击表“dept”和表“salary”之间的连线, 选择“编辑关系”命令。
- (3) 在“编辑关系”对话框中, 在“表:dept”框和“相关表:salary”框中都选择“部门号”, 单击“确定”按钮。

试题 4

- (1) 从“数据库”菜单中选择“清理数据库”命令。
- (2) 用鼠标右键单击表“dept”和表“salary”之间的连线, 选择“编辑参照完整性”命令。
- (3) 在“参照完整性生成器”对话框中的“更新规则”选项卡中选择“限制”选项, 在“删除规则”选项卡中选择“级联”选项, 在“插入规则”选项卡中选择“限制”选项, 单击“确定”按钮。

2. 简单应用题

试题 1

- (1) 启动 Visual FoxPro。
- (2) 从“文件”菜单中选择“打开”命令, 在“打开”对话框中, 选择考生目录下的数据库文件“salary_db.dbc”,

单击“确定”按钮。

- (3) 从“文件”菜单中选择“新建”命令。
- (4) 在“新建”对话框中选择“视图”选项，再单击“新建文件”按钮。
- (5) 将“salary”表加入“视图设计器”。
- (6) 依次将“salary. 部门号”、“salary. 雇员号”、“salary. 姓名”、“salary. 工资”、“salary. 补贴”、“salary. 奖励”、“salary. 失业保险”、“salary. 医疗统筹”、“salary. 工资+salary. 补贴+salary. 奖励-salary. 失业保险-salary. 医疗统筹 AS 实发工资”加入“选定字段”框中，如图 14-12 所示。
- (7) 在“排序依据”选项卡中，选择“salary. 部门号”加入“排序条件”框，在“排序选项”栏中选择“降序”选项，如图 14-13 所示。



图 14-12 视图设计器



图 14-13 设置排序依据

- (8) 在“视图设计器”对话框中，单击鼠标右键选择“查看 SQL”命令。
- (9) 将所有 SQL 命令复制到剪贴板，如图 14-14 所示。
- (10) 从“文件”菜单中选择“新建”命令。
- (11) 在“新建”对话框中选择“程序”选项，再单击“新建文件”按钮。
- (12) 将剪贴板中的命令复制到“程序”窗口中，并保存为“t1.prg”，如图 14-15 所示。

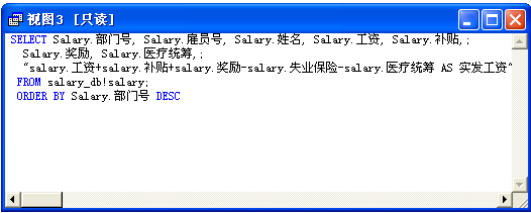


图 14-14 复制 SQL 命令

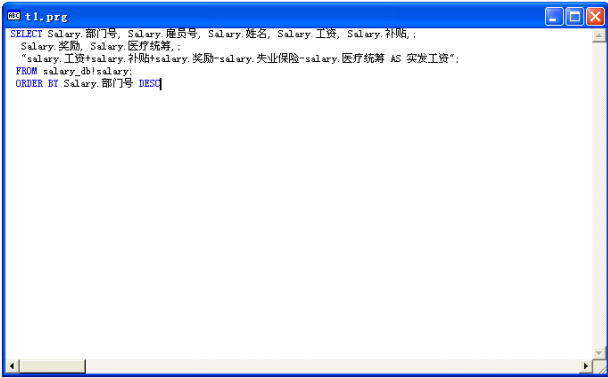


图 14-15 将 SQL 命令复制到程序窗口

- (13) 选择“程序”菜单中的“运行”命令执行程序。

试题 2

- (1) 从“文件”菜单中选择“新建”命令。
- (2) 在“新建”对话框中选择“表单”选项，再单击“新建文件”按钮。

- (3) 从“表单控件”工具条中,选择“表格”控件,放入“FORM1”中。
- (4) 从“表单控件”工具条中,选择“命令按钮”控件,放入“FORM1”中。
- (5) 在“表格”控件上单击鼠标右键,选择“属性”命令。在“属性”对话框的“全部”选项卡中,将“RecordSourceType”项设为“表”,将“RecordSource”项设为“salary”,将“Name”项设为“grdSalary”。
- (6) 在“命令按钮”控件上单击鼠标右键,选择“属性”命令。在“属性”对话框的“全部”选项卡中,将“Caption”项设为“退出浏览”。
- (7) 在“命令按钮”控件上单击鼠标右键,选择“代码”命令,在“Command1.Click”窗口中输入 `thisform.release`。

3. 综合应用题

- (1) 新建一个名为“FORM2”的表单文件,然后在其上添加一个“命令按钮组”控件和两个“命令按钮”控件。
- (2) 修改“命令按钮组”控件,将其命名为“Optiongroup1”,有“雇员工资表”(名称为 Option1)、“部门表”(名称为 Option2)、“部门工资汇总表”(名称为 Option3) 3 个选项按钮。
- (3) 修改两个命令按钮,将其命名为“生成”(名称为 Command1)和“退出”(名称为 Command2)。
- (4) 为表单建立数据环境,并向数据环境添加“dept”表(名称为 Cursor1)和“salary”表(名称为 Cursor2)。
- (5) 在“表单设计器”对话框中,在“雇员工资表”按钮上单击鼠标右键,选择“代码”命令。
- (6) 在“Command1.Click”窗口中,输入 `cur=1`。
- (7) 在“表单设计器”对话框中,在“部门表”按钮上单击鼠标右键,选择“代码”命令。
- (8) 在“Command2.Click”窗口中,输入 `cur=2`。
- (9) 在“表单设计器”对话框中,在“部门工资汇总表”按钮上单击鼠标右键,选择“代码”命令。
- (10) 在“Command3.Click”窗口中,输入 `cur=3`。
- (11) 在“表单设计器”对话框中,在“生成”按钮上单击鼠标右键,选择“代码”命令。
- (12) 在“Command1.Click”窗口中,输入:

```
DO CASE
CASE cur=1
    SELECT salary.部门号, salary.雇员号, salary.姓名, salary.工资,;
           salary.补贴, salary.奖励, salary.失业保险,;
           salary.医疗统筹, salary.工资+salary.补贴+salary.奖励;
           -salary.失业保险-salary.医疗统筹 AS 实发工资;
    FROM salary_db! Salary;
    ORDER BY salary.部门号 DESC;
    INTO TABLE gz1.dbf

CASE cur=2
    SELECT *;
    FROM salary_db! dept;
    INTO TABLE bm1.dbf

OTHERWISE
    SELECT dept.*, SUM(salary.工资) AS 工资, SUM(salary.补贴) AS 补贴,;
           SUM(salary.奖励) AS 奖励, SUM(salary.失业保险) AS 失业保险;
           , SUM(salary.医疗统筹) AS 医疗统筹;
    FROM salary_db! Dept INNER JOIN salary_db! Salary;
    ON dept.部门号 = salary.部门号;
    GROUP BY dept.部门号;
    INTO TABLE hz1.dbf

ENDCASE
```

- (13) 在“表单设计器”对话框中,在“退出”按钮上单击鼠标右键,选择“代码”命令。
- (14) 在“Command2.Click”窗口中,输入 `thisform.release`。

14.8 上机模拟试题二答案与解析

1. 基本操作题

试题 1

- (1) 新建一个表“销售表 2005.dbf”，其结构与“销售表”相同。
- (2) 在“销售表 2005.dbf”中，从“表”菜单中选择“追加记录”命令。
- (3) 在“追加来源”对话框中，在“类型”框中选择“Table”，在“来源于”框中选择“销售表”。
- (4) 单击“选项”按钮，在“追加来源选项”对话框中，单击“字段”按钮，选择所有的字段。
- (5) 在“追加来源选项”对话框中，单击“For”按钮，建立表达式“销售表 2005.日期<={^2005-12-31}”，单击“确定”按钮，如图 14-16 所示。

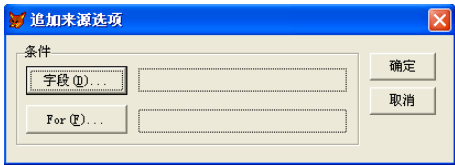


图 14-16 “追加来源选项”对话框

- (6) 在“追加来源”对话框中，单击“确定”按钮。

试题 2

- (1) 打开“销售表”，从“显示”菜单中选择“浏览”命令。
- (2) 从“表”菜单中选择“删除记录”命令，如图 14-17 所示。
- (3) 在“删除”对话框中，在“作用范围”框中选择“ALL”，在“For”文本框中输入“销售表 2005.日期<={^2005-12-31}”，单击“删除”按钮，如图 14-18 所示。



图 14-17 选择“删除记录”命令

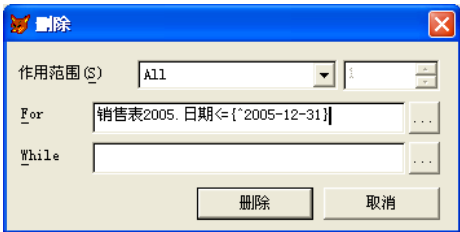


图 14-18 “删除”对话框

- (4) 从“表”菜单中选择“彻底删除”命令，如图 14-19 所示。

试题 3

- (1) 打开“销售表”，从“显示”菜单中选择“浏览”命令。
- (2) 从“文件”菜单中选择“另存为 HTML”命令。
- (3) 在“另存为 HTML”对话框中，单击“确定”按钮。

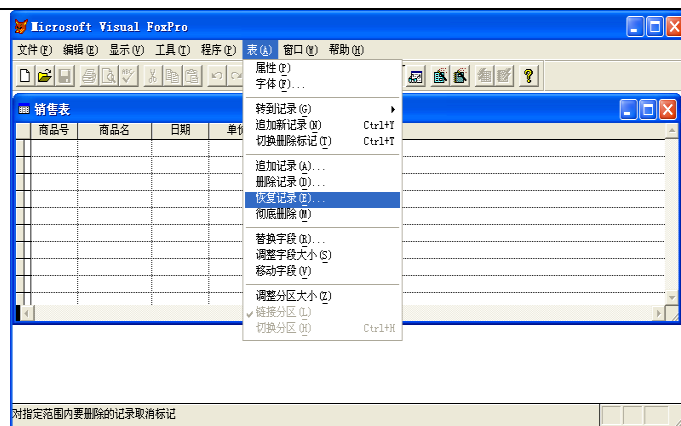


图 14-19 选择“彻底删除”命令

试题 4

- (1) 从考生目录下打开“商品销售”数据库。
- (2) 在“数据库设计器”中，用鼠标右键单击“商品表”，选择“修改”命令。
- (3) 在“表设计器”中，选择“索引”选项卡。
- (4) 在“索引名”栏输入“商品号”，“排序”栏选择“升序”，“类型”栏选择“主索引”，“表达式”栏输入“商品号”，单击“确定”按钮。
- (5) 在“数据库设计器”中，用鼠标右键单击“销售表”，选择“修改”命令。
- (6) 在“表设计器”中，选择“索引”选项卡。
- (7) 在“索引名”栏输入“商品号”，“排序”栏选择“升序”，“类型”栏选择“普通索引”，“表达式”栏输入“商品号”，单击“确定”按钮。

2. 简单应用题

试题 1

- (1) 在“文件”菜单中选择“新建”命令，新建“查询”文件。
- (2) 依次将考生目录下的商品表和销售表添加并保留默认连接。
- (3) 依次将“商品表.商品号”、“商品表.商品名”、“销售表.单价”、“销售表.销售数量”选入“选定字段”框中。
- (4) 在“排序依据”选项卡中，将“销售表.单价×销售表.销售数量”加入“排序条件”框，在“排序选项”栏中选择“升序”选项。
- (5) 在“查询设计器”对话框中单击鼠标右键，选择“输出设置”命令。在“查询去向”中选择“表”，在“表名”框中输入“see_a”，单击“确定”按钮。

试题 2

- (1) 在“表单设计器”对话框中，选择“刷新标题”按钮，单击鼠标右键，选择“代码”命令。
- (2) 在“Command1.Click”窗口中，键入：thisform.Caption="商品销售数据输入"。
- (3) 在“表单设计器”对话框中，选择“商品销售输入”按钮，单击鼠标右键，选择“代码”命令。
- (4) 在“Command2.Click”窗口中，键入：DO FORM sellcomm。
- (5) 在“表单设计器”对话框中，选择“输出销售报表”按钮，单击鼠标右键，选择“代码”命令。
- (6) 在“Command3.Click”窗口中，键入：report form print1 preview。

3. 综合应用题

试题 1

首先备份文件，然后再修改“商品表”中的“单价”，最后利用“单价调整表”对“商品表”中的“单价”进行修改。程序代码如下：

```
USE 商品表.dbf
COPY STRUCTURE TO 商品表备份.dbf
APPEND FROM 商品表.dbf
SELECT a
USE 商品表.dbf
SCAN
    REPLACE a.单价 WITH a.单价*0.1 FOR LEFT(a.商品号,1)='4'
ENDSCAN
SELECT a
USE 单价调整表.dbf
SELECT b
USE 商品表.dbf
UPDATE ON 商品号 FROM a REPLACE 出厂单价 WITH a.出厂单价
```

- 试题 2
- (1)新建一个名为“FORM22”的表单文件，然后在其上添加两个“命令按钮”控件，分别为“调整”(名称为 Command1)和“退出”(名称为 Command2)。
- (2)在“表单设计器”对话框中，选择“调整”按钮，单击鼠标右键，选择“代码”命令。
- (3)在“Command1.Click”窗口中，添加：DO change_c。
- (4)在“表单设计器”对话框中，选择“退出”按钮，单击鼠标右键，选择“代码”命令。
- (5)在“Command2.Click”窗口中，添加：thisform.release。

14.9 上机模拟试题三答案与解析

1. 基本操作题

- 试题 1
- (1)启动 Visual FoxPro，界面如图 14-20 所示。

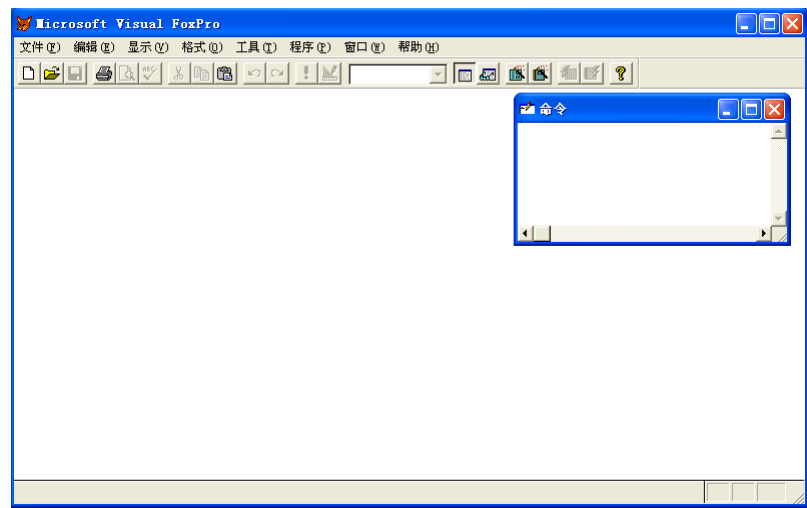


图 14-20 启动 Visual FoxPro 界面

- (2)从“文件”菜单中选择“打开”命令。
- (3)在“打开”对话框中选择考生目录下的“salary_db.dbc”文件，单击“确定”按钮，如图 14-21 所示。

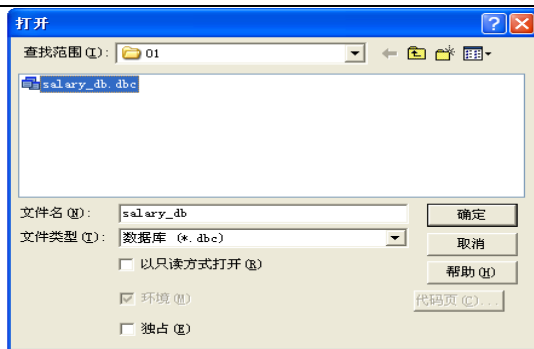


图 14-21 “打开”对话框

(4) 在“数据库设计器”中，单击鼠标右键，选择“新建表”命令，如图 14-22 所示。

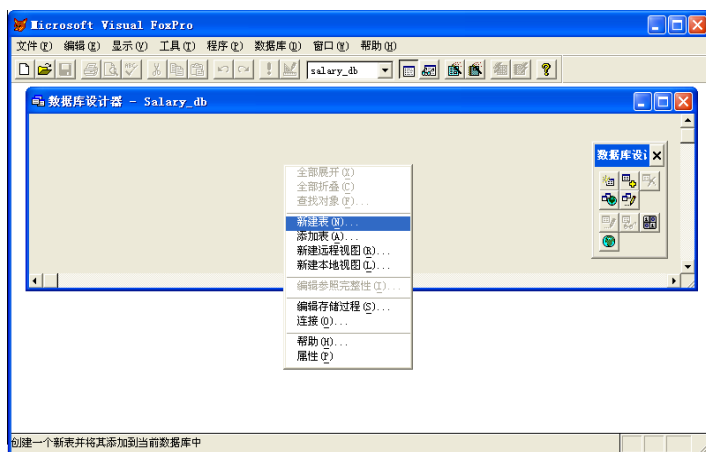


图 14-22 选择“新建表”命令

(5) 在“新建表”对话框中，单击“新建表”按钮。

(6) 在“创建”对话框中输入表名“dept”，单击“保存”按钮，如图 14-23 所示。

(7) 在“表设计器”的“字段”选项卡中，在“字段名”栏输入“部门号”，在“类型”栏选择“字符型”，在“宽度”栏输入“2”；在“字段名”栏输入“部门名”，在“类型”栏选择“字符型”，在“宽度”栏输入“20”；单击“确定”按钮，如图 14-24 所示。

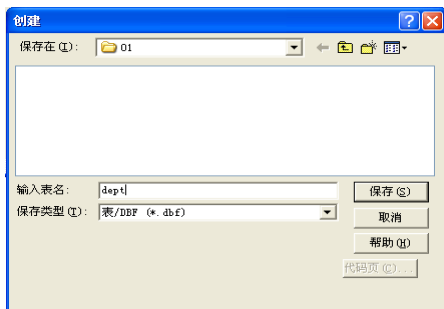


图 14-23 输入表名 dept



图 14-24 设置“字段”选项卡

- (8) 在“数据库设计器”中，用鼠标右键单击表“dept”，选择“浏览”命令，如图 14-25 所示。
- (9) 在表“dept”中依次输入 5 条记录，如图 14-26 所示。

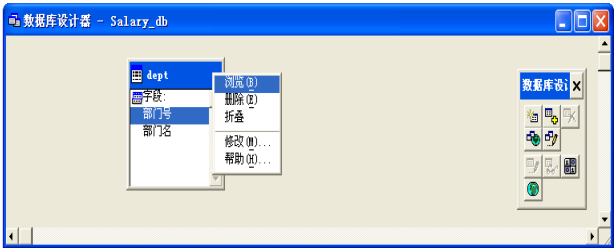


图 14-25 选择“浏览”命令

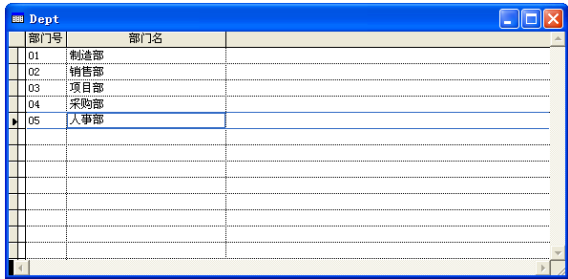


图 14-26 输入 5 条记录

试题 2

- (1) 在“数据库设计器”中，用鼠标右键单击表“dept”，选择“修改”命令，如图 14-27 所示。
- (2) 在“表设计器”中，选择“索引”选项卡。
- (3) 在“索引名”栏输入“部门号”，“排序”栏选择“升序”，“类型”栏选择“主索引”，“表达式”栏输入“部门号”，单击“确定”按钮，如图 14-28 所示。

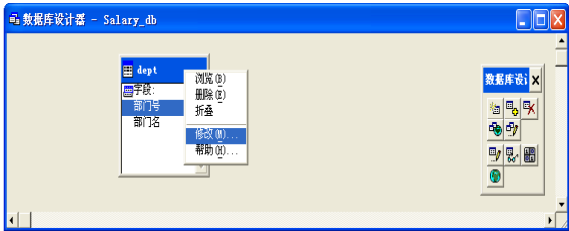


图 14-27 选择“修改”命令

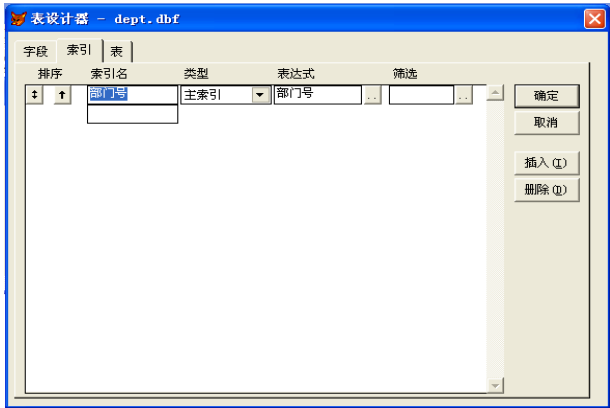


图 14-28 设置“索引”选项卡

试题 3

- (1) 在“数据库设计器”中，将“dept”表中的“索引”部门号拖放到“salary”表中的“索引”部门号上，如图 14-29 所示。
- (2) 用鼠标右键单击表“dept”和表“salary”之间的连线，选择“编辑关系”命令。
- (3) 在“编辑关系”关系对话框中，在“表:dept”框和“相关表:salary”框中都选择“部门号”，单击“确定”按钮，如图 14-30 所示。

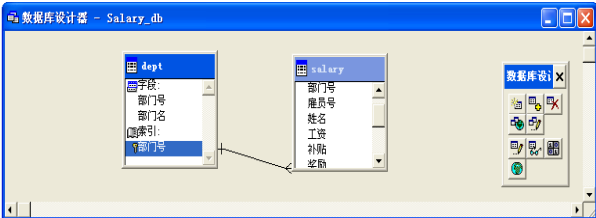


图 14-29 拖放“索引”部门号

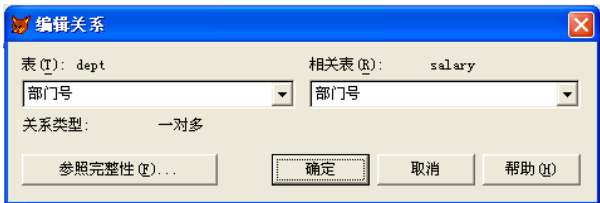


图 14-30 编辑关系

试题 4

(1) 从“数据库”菜单中选择“清理数据库”命令，如图 14-31 所示。

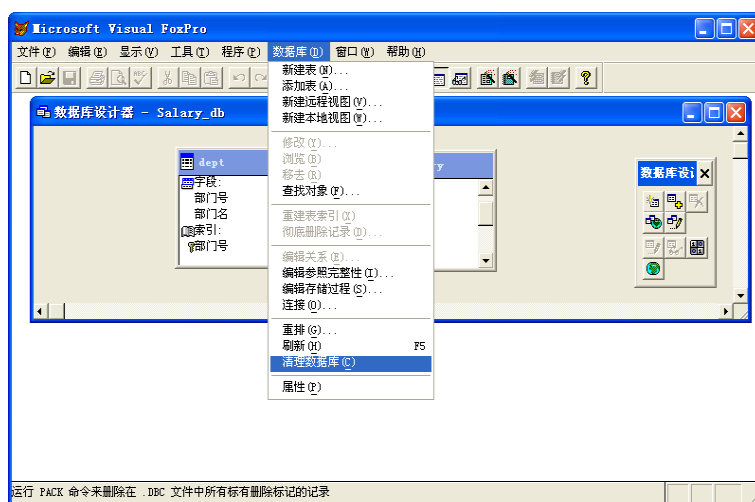


图 14-31 选择“清理数据库”命令

(2) 用鼠标右键单击表“dept”和表“salary”之间的连线，选择“编辑参照完整性”命令，如图 14-32 所示。



图 14-32 选择“编辑参照完整性”命令

(3) 在“参照完整性生成器”对话框的“更新规则”选项卡中选择“级联”选项，在“删除规则”选项卡中选择“限制”选项，在“插入规则”选项卡中选择“限制”选项，如图 14-33 所示。

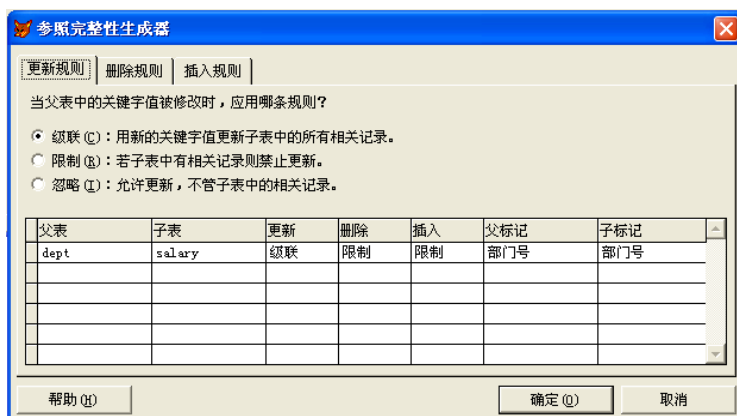


图 14-33 设置参照完整性约束

2. 简单应用题

试题 1

- (1) Visual FoxPro。
- (2) 从“文件”菜单中选择“新建”命令。
- (3) 在“新建”对话框中选择“报表”，单击“向导”按钮，如图 14-34 所示。
- (4) 在“向导选取”对话框中选择“报表向导”，单击“确定”按钮，如图 14-35 所示。
- (5) 在“步骤 1”对话框中，单击“数据库和表”框右侧的“...”按钮，打开考生目录下的“salary”表。将“salary”表中所有字段选入“选定字段”框，单击“下一步”按钮，如图 14-36 所示。
- (6) 在“步骤 2”对话框中，保持默认值，单击“下一步”按钮，如图 14-37 所示。



图 14-34 选择“报表”文件类型

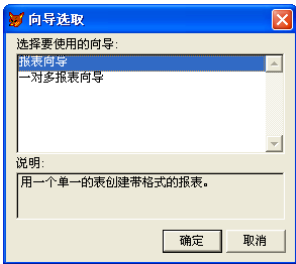


图 14-35 选择“报表向导”

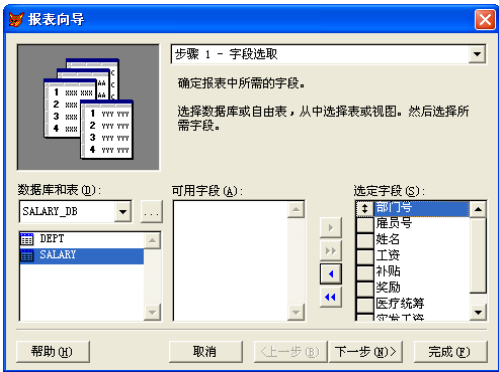


图 14-36 “步骤 1”对话框

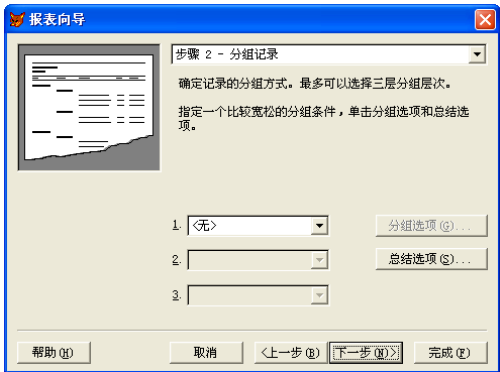


图 14-37 “步骤 2”对话框

- (7) 在“步骤 3”对话框中，“样式”选择“随意式”，单击“下一步”按钮，如图 14-38 所示。
- (8) 在“步骤 4”对话框中，保持默认值，单击“下一步”按钮，如图 14-39 所示。
- (9) 在“步骤 5”对话框中，将“雇员号”选入“选定字段”框，并选中“升序”单选按钮，单击“下一步”按钮，如图 14-40 所示。
- (10) 在“步骤 6”对话框中，在“报表标题”框中输入“雇员工资一览表”，其余保持默认值，单击“完成”按钮，如图 14-41 所示。

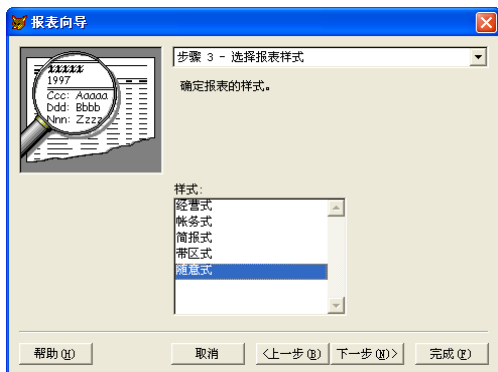


图 14-38 “步骤 3”对话框

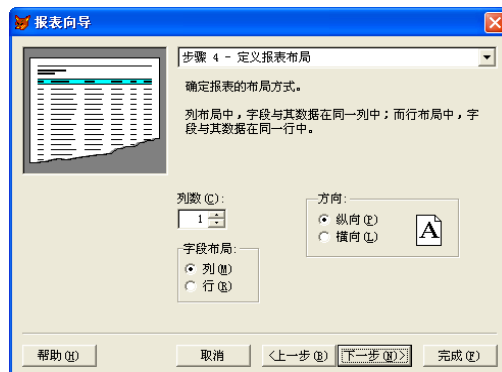


图 14-39 “步骤 4”对话框

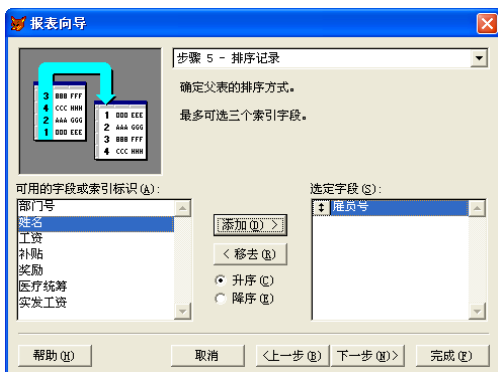


图 14-40 “步骤 5”对话框

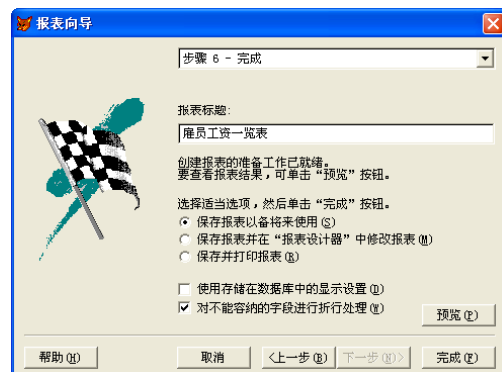


图 14-41 “步骤 6”对话框

(11) 在“另存为”对话框中输入报表文件名“print2”，单击“保存”按钮，如图 14-42 所示。

试题 2

(1) 在“表单设计器”对话框的“刘缆雇员工资”按钮上单击右键，选择“属性”命令，如图 14-43 所示。

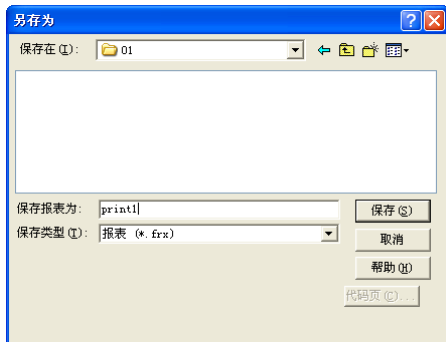


图 14-422 “另存为”对话框

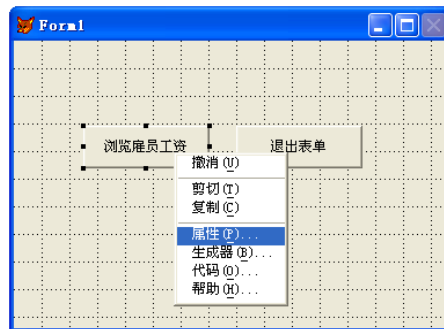


图 14-43 选择“属性”命令

(2) 在“属性”对话框的“全部”选项卡中，选择“Caption”行。

(3) 在文本框中将“刘缆雇员工资”改为“浏览雇员工资”，然后按回车键，如图 14-44 所示。

(4) 在“表单设计器”对话框的“浏览雇员工资”按钮上单击鼠标右键，选择“代码”命令，如图 14-45 所示。

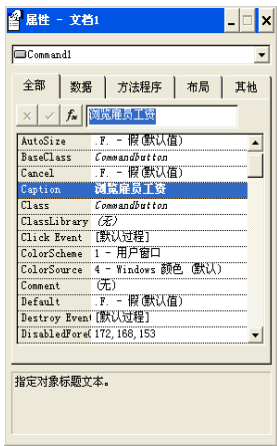


图 14-44 修改 Caption 行

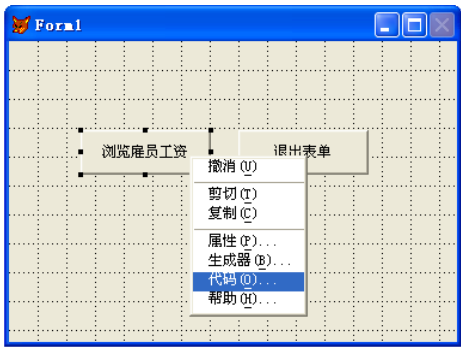


图 14-45 选择“代码”命令

(5) 在“Command1.Click”窗口中，如图 14-46 所示，将原“SELECT”命令改为：

```
SELECT *;  
FROM salary
```

(6) 在“表单设计器”对话框的“退出表单”按钮上单击鼠标右键，选择“代码”命令，如图 14-47 所示。

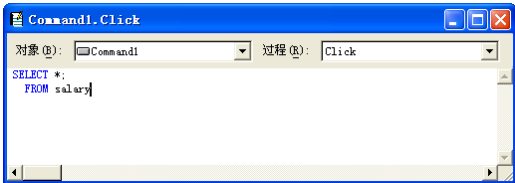


图 14-46 修改 SELECT 命令

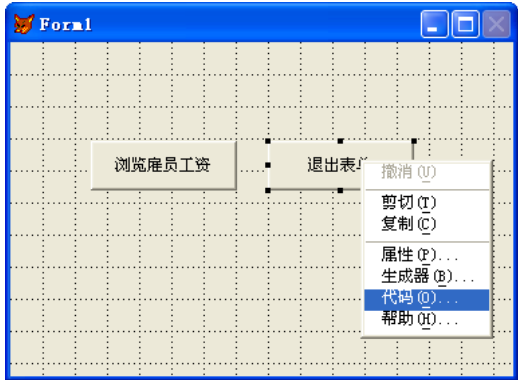


图 14-47 选择“代码”命令

(7) 在“Command2.Click”窗口中，将原代码改为：THISFORM RELEASE，如图 14-48 所示。

3. 综合应用题

试题一
首先建立表，然后再调整“工资”。
程序名称为 change_c，程序代码如下。

```
USE salary  
COPY STRUCTURE TO bak_salary.dbf  
USE bak_salary  
APPEND FROM salary  
SELECT a  
USE c_salary1  
SELECT b  
USE salary
```

试题 2

(1)新建一个名为“FORM2”的表单文件,然后在其上添加两个“Command”控件,分别为“调整”(名称为 Command1)和“退出”(名称为 Command2),如图 14-49 所示。

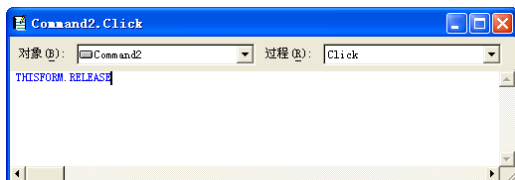


图 14-48 将原代码改为“THISFORM.RELEASE”

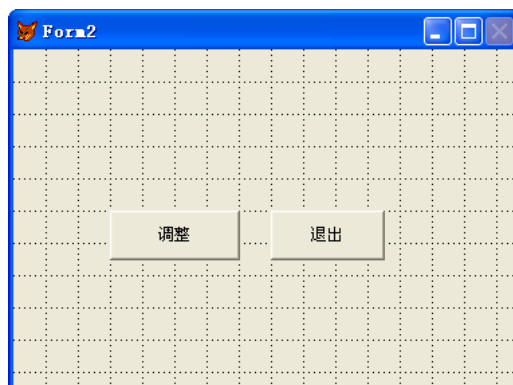


图 14-49 添加两个“Command”控件

- (2) 在“表单设计器”对话框中的“调整”按钮上单击鼠标右键，选择“代码”命令。
- (3) 在“Command1.Click”窗口添加：DO change_c 命令，如图 14-50 所示。
- (4) 在“表单设计器”对话框中的“退出”按钮上单击鼠标右键，选择“代码”命令。
- (5) 在“Command2.Click”窗口添加：thisform.release 命令，如图 14-51 所示。

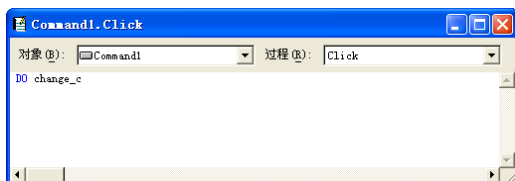


图 14-50 添加 “Do change_c” 命令

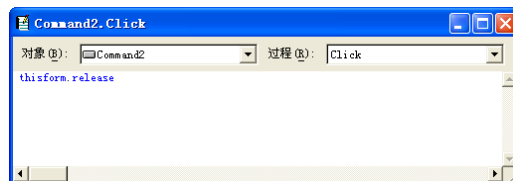


图 14-51 添加“thisform.release”命令

14.10 上机模拟试题四答案与解析

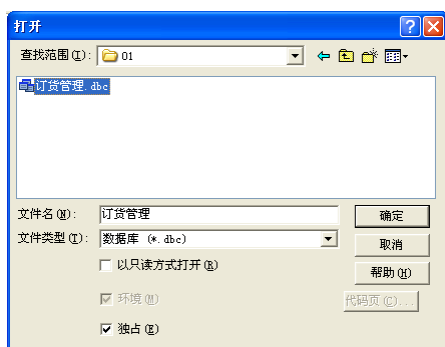


图 14-52 选择“订货管理.dbc”文件

1. 基本操作题

试题 1

- (1) 启动 Visual FoxPro。
- (2) 从“文件”菜单中选择“打开”命令。
- (3) 在“打开”对话框中, 选择考生目录下的“订货管理.dbc”文件, 单击“确定”按钮, 如图 14-52 所示。
- (4) 在“数据库设计器”中, 单击鼠标右键, 选择“添加表”命令, 如图 14-53 所示。

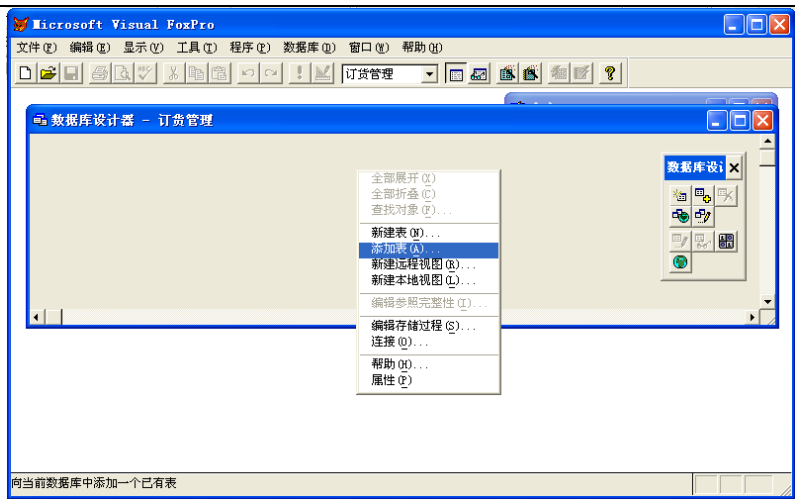


图 14-53 选择“添加表”命令

- (5) 在“打开”对话框中选择考生目录下的“order_detail.dbf”，单击“确定”按钮。
- (6) 按照步骤（4）、（5）依次将“order_list”表和“customer”表添加到数据库中。

试题 2

- (1) 在“数据库设计器”中，用鼠标右键单击“order_list”表，选择“修改”命令。
- (2) 在“表设计器”中，选择“索引”选项卡。
- (3) 在“索引名”栏输入“订单号”，“排序”栏选择“升序”，“类型”栏选择“主索引”，“表达式”栏输入“订单号”，单击“确定”按钮。

试题 3

- (1) 在“数据库设计器”中，用鼠标右键单击“order_detail”表，选择“修改”命令。
- (2) 在“表设计器”中，选择“索引”选项卡。
- (3) 在“索引名”栏输入“订单号”，“排序”栏选择“升序”，“类型”栏选择“普通索引”，“表达式”栏输入“订单号”，单击“确定”按钮。
- (4) 在“数据库设计器”中，将“order_list”表中的“索引”订单号拖放到“order_detail”表中的“索引”订单号上。
- (5) 用鼠标右键单击表“order_list”和表“order_detail”之间的连线，选择“编辑关系”命令。
- (6) 在“编辑关系”对话框中，在“表: order_list”框和“相关表: order_detail”框中都选择“订单号”，单击“确定”按钮。

2. 简单应用题

试题 1

- (1) 在“文件”菜单中选择“新建”命令，新建“查询”文件，并依次添加表“order_detail”和表“order_list”并保留默认连接，如图 14-54 所示。
- (2) 依次将“order_detail. 订单号”、“order_detail. 产品号”、“order_detail. 产品名”、“order_detail. 单价”、“order_detail. 数量”选入“选定字段”框中，如图 14-55 所示。
- (3) 在“排序依据”选项卡中，选择“order_detail. 订单号”放入“排序条件”框，在“排序选项”栏中选择“升序”选项。
- (4) 在“排序依据”选项卡中，选择“order_detail. 单价”放入“排序条件”框，在“排序选项”栏中选择“降序”选项，如图 14-56 所示。



图 14-54 新建“查询”文件

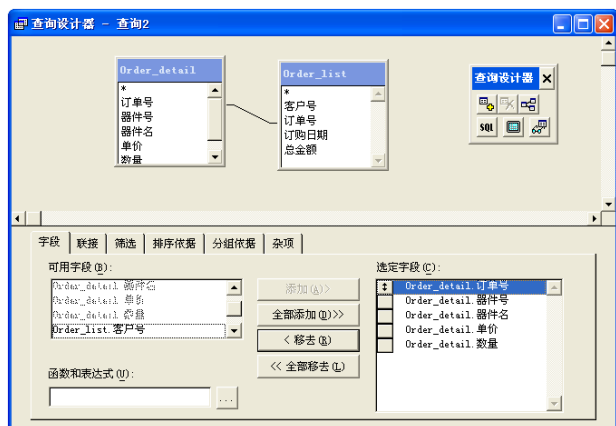


图 14-55 选定字段



图 14-56 设置“排序依据”选项卡

(5) 在“查询设计器”对话框中单击鼠标右键，选择“输出设置”命令，如图 14-57 所示。

(6) 在“查询去向”中选择“表”，在“表名”框中输入“result”，单击“确定”按钮，如图 14-58 所示。



图 14-57 选择“输出设置”命令

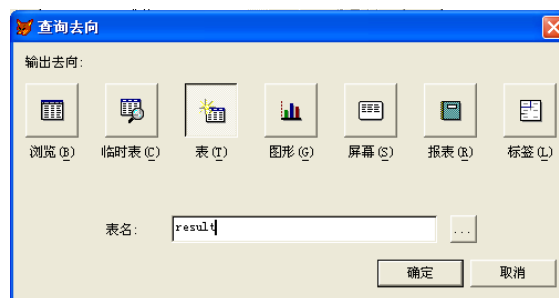


图 14-58 设置查询去向

试题 2

- (1) 在“表单设计器”对话框中，选择“刷新标题”按钮，单击鼠标右键，选择“代码”命令。
- (2) 在“Command1.Click”窗口中，将原代码改为：thisform.Caption="简单应用"。
- (3) 在“表单设计器”对话框中，选择“订单记录”按钮，单击鼠标右键，选择“代码”命令。
- (4) 在“Command2.Click”窗口中，将原代码改为：thisform.Grid1.RecordSource="order_list.dbf"。
- (5) 在“表单设计器”对话框中，选择“关闭表单”按钮，单击鼠标右键，选择“代码”命令。
- (6) 在“Command3.Click”窗口中，将原代码改为：thisform.release。

3. 综合应用题

程序代码如下：

```
SELECT a
USE order_list
SELECT b
USE order_detail
```

- 创建报表步骤：
- (1) 从“文件”菜单中选择“新建”命令，新建“报表”文件。
 - (2) 在“报表”菜单中选择“快速报表”命令，如图 14-59 所示。

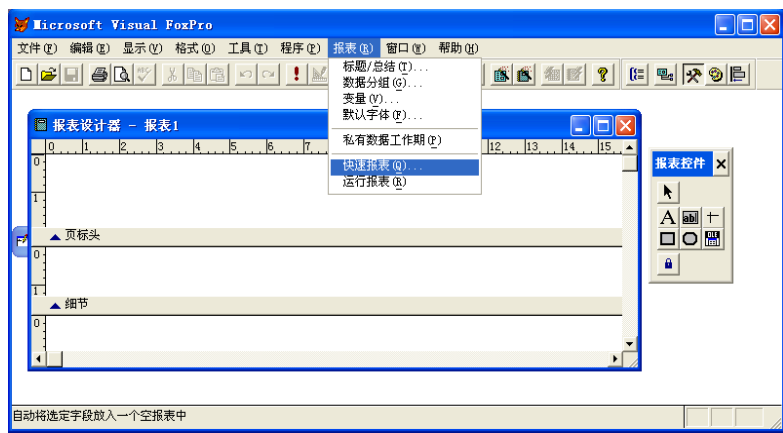


图 14-59 选择“快速报表”命令

- (3) 在“打开”对话框中选择考生目录下的“order_detail.dbf”，单击“确定”按钮。
- (4) 在“快速报表”对话框的“字段布局”栏选择横向布局，单击“字段”按钮，如图 14-60 所示。
- (5) 在“字段选择器”对话框中，依次选择“订单号”、“产品号”、“产品名”、“单价”和“数量”字段，单击“确定”按钮，如图 14-61 所示。

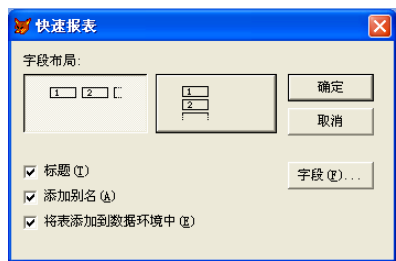


图 14-60 设置字段布局

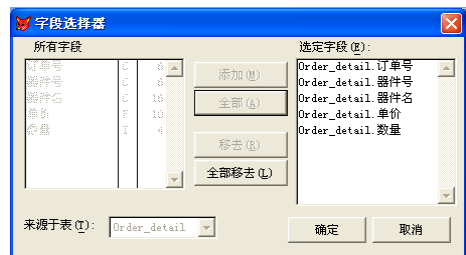


图 14-61 字段选择器

- (6) 在“快速报表”对话框中，单击“确定”按钮。

14.11 上机模拟试题五答案与解析

1. 基本操作题
- 试题 1
- (1) 启动 Visual FoxPro。
 - (2) 从“文件”菜单中选择“新建”命令。
 - (3) 在“新建”对话框中，选择“项目”选项，单击“新建文件”按钮，如图 14-62 所示。
 - (4) 在“创建”对话框中输入项目文件名“客户管理”，单击“保存”按钮，如图 14-63 所示。



图 14-62 选择“项目”文件类型

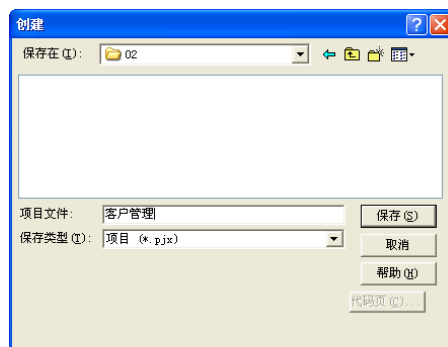


图 14-63 输入项目文件名

试题 2

- (1) 在“项目管理器”中的“数据”选项卡中，选择“数据库”选项，单击“新建”按钮，如图 14-64 所示。
- (2) 在“新建数据库”对话框中，单击“新建数据库”按钮。
- (3) 在“创建”对话框中输入数据库名“订货管理”，单击“保存”按钮，如图 14-65 所示。

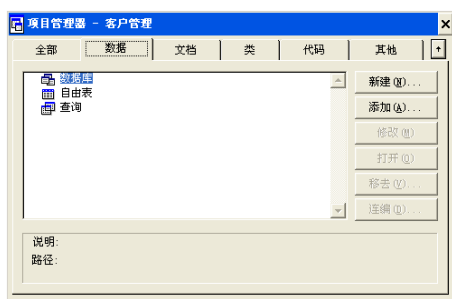


图 14-64 选择“数据库”选项

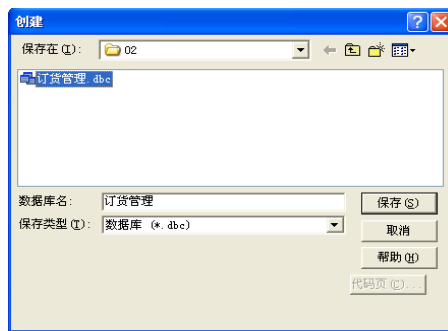


图 14-65 输入数据库名

试题 3

- (1) 在“项目管理器”中的“数据”选项卡中，选择“数据库”中的“订货管理”选项，再选择“表”，单击“新建”按钮，如图 14-66 所示。
- (2) 在“新建表”对话框中，单击“新建表”按钮。
- (3) 在“创建”对话框中输入表名“order_list”，单击“保存”按钮，如图 14-67 所示。

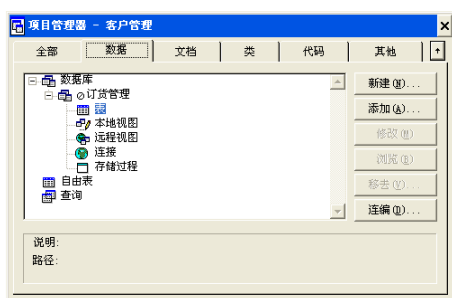


图 14-66 选择“订货管理”下的“表”

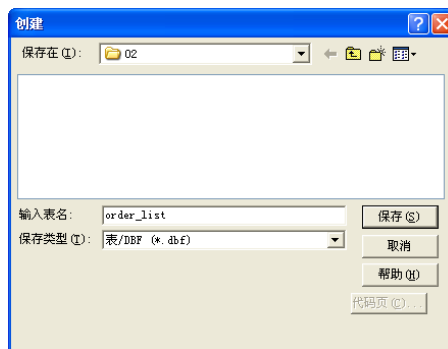


图 14-67 输入表名

(4) 在“表设计器”的“字段”选项卡中，在“字段名”栏中输入“客户号”，在“类型”栏选择“字符型”，在“宽度”栏输入“6”；在“字段名”栏输入“订单号”，在“类型”栏选择“字符型”，在“宽度”栏输入“6”；在“字段名”栏输入“订购日期”，在“类型”栏选择“日期型”；在“字段名”栏输入“总金额”，在“类型”栏选择“浮动型”，在“宽度”栏输入“15”，在“小数位数”栏输入“2”；单击“确定”按钮，如图 14-68 所示。



图 14-68 设置“字段”选项卡

- 试题 4
- (1) 在“表设计器”中，选择“索引”选项卡。
 - (2) 在“索引名”栏输入“订单号”，“排序”栏选择“升序”，“类型”栏选择“主索引”，“表达式”栏输入“订单号”，单击“确定”按钮，如图 14-69 所示。



图 14-69 设置“索引”选项卡

2. 简单应用题
- 试题 1
- (1) 在考生目录下打开“order_list”表。
 - (2) 从“表”菜单中选择“追加记录”命令，如图 14-70 所示。

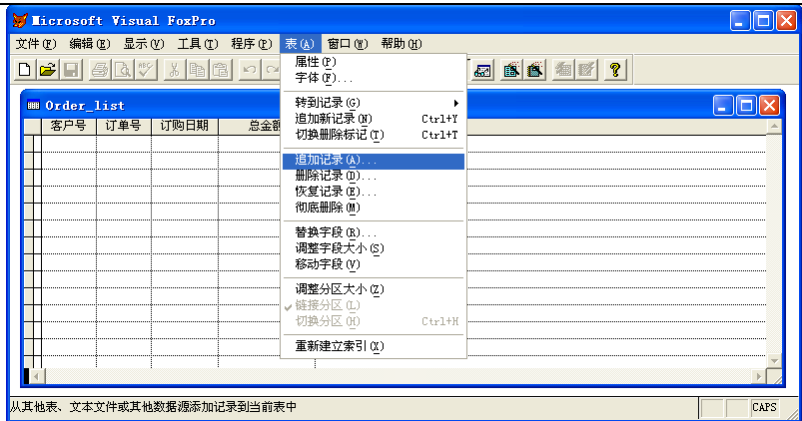


图 14-70 选择“追加记录”命令

(3) 在“追加来源”对话框的“类型”框中选择“Table (DBF)”，单击“来源于”框右侧的“...”按钮，如图 14-71 所示。

(4) 在“打开”对话框中，选择考生目录下的“order_list1”表，单击“确定”按钮。

(5) 在“追加来源”对话框中，单击“确定”按钮。

(6) 在“命令”窗口输入如下 SQL SELECT 语句完成查询。

```
SELECT customer.客户号, customer.客户号, customer.订单号, customer.总金额;  
FROM 订货管理!order_list INNER JOIN customer;  
ON order_list.客户号 = customer.客户号;  
ORDER BY order_list.总金额 DESC;  
INTO TABLE results.dbf
```

试题 2

(1) 在“表单设计器”中，单击“全部”按钮。

(2) 在“布局”工具条中，单击“左边对齐”按钮。

(3) 在“表单控件”工具条中，选择“表格”控件，放在按钮的右边。

(4) 在“表格”控件上单击鼠标右键，选择“属性”命令。

(5) 在“属性”对话框的“数据”选项卡中，将“RecordSourceType”项设为“表”，将“RecordSource”项设为“customer”，如图 14-72 所示。



图 14-71 “追加来源”对话框



图 14-72 设置 RecordSourceType 项和 RecordSource 项

3. 综合应用题

首先利用 SELECT 命令生成“result”表，其包含“订单号”和“总金额”两个字段，然后修改“order_list”表中的“总金额”，并按要求排序。

程序代码如下：

```
SELECT SUM(order_detail.单价*order_detail.数量) AS 总金额,
       order_detail.订单号;
FROM order_detail;
GROUP BY order_detail.订单号;
INTO TABLE result.dbf
CLOSE TABLE
SELECT a
USE result
SELECT b
USE order_list
UPDATE ON 订单号 FROM a REPLACE 总金额 WITH a.总金额
SORT TO od_new ON 总金额 ASCENDING
```

附录

A

习题参考答案

第1章 习题分析与解答

一、选择题

1. 答案: B

解析: 数据的存储结构分顺序存储结构和链式存储结构, 一个数据的逻辑结构可以有多种存储结构。顺序结构中数据元素所占的存储空间是连续的, 而链式存储结构中, 数据元素通过指针联系就在一起了, 所以所占的存储空间不一定是连续的。

2. 答案: D

解析: 队列 (queue) 是指允许在一端进行插入、而在另一端进行删除的线性表。允许插入的一端称为队尾, 允许删除的一端称为队头。在队列这种数据结构中, 最先插入的元素将最先能够被删除; 反之, 最后插入的元素将最后才能被删除。因此, 队列又称“先进先出”或“后进后出”的线性表。

3. 答案: D

解析: 各种排序方法中最坏情况需要比较的次数分别为: 冒泡排序 $N(N-1)/2$ 、快速排序 $N(N-1)/2$ 、简单插入排序 $N(N-1)/2$ 、堆排序 $O(N\log_2 N)$

4. 答案: B

解析: 栈是线性表的一种, 它的特点是先进后出, 并且只能在表的一端进行插入和删除操作的线性表, 入栈和出栈都是在栈顶进行, 因此具有记忆作用, 栈可以采用顺序存储, 也可以采用链式存储。

5. 答案: A

解析: 在任意一棵二叉树中, 度为 0 的结点 (即叶子结点) 总是比度为 2 的结点多一个, 所以该二叉树的叶子结点数等于 $N+1$ 。

6. 答案: C

解析: 二叉树前序遍历的简单描述为: 若二叉树为空, 则结点返回; 否则: 1 访问根结点; 2 前序遍历左子树; 3 前序遍历右子树。可见, 前序遍历二叉树的过程是一个递归的过程。根据题目中给出的二叉树的结构可知道序遍历的结果是 ABDYECFXZ。

7. 答案: C

解析: 对长度为 n 的线性表进行顺序查找时, 从表中的第一个元素开始, 给定的值与表中逐个元素的关键字进行比较, 直到两者相符, 查找完成。在最坏情况下, 要查找的元素是表的最后一个元素或查找失败, 这两种情况都需要将这个元素中的所有元素进行比较, 因此比较次数为 n 。

8. 答案: D

解析: 数据的逻辑结构是指反映数据元素之间逻辑关系的数据结构。数据的逻辑结构在计算机存储空间中的存放形

式称为数据的存储结构（也称数据的物理结构）。一般来说，一种数据的逻辑结构根据需要可以表示成多种存储结构，常用的存储结构有顺序，链接，索引等。

9. 答案：D

解析：算法分析是指对一个算法的运行时间和占用空间做定量的分析，计算相应的数量级，分析算法的目的就是要降低算法的时间复杂度和空间复杂度，提高算法的执行效率。

10. 答案：A

解析：队列是一种特殊的线性表示，只允许在表的一端插入元素，在表的另一端删除元素，插入元素的一端叫“队尾”，删除元素的一端叫“队头”，先插入的元素先被删除，是按先进先出的原则组织数据的。

11. 答案：C

解析：栈和队列都是操作受限的线性表，只允许在端点插入和删除。不同点是栈只允许在表的一端进行插入和删除操作，而队列允许在表的一端进行插入操作，而在另一端进行删除操作。

12. 答案：B

解析：在链式存储结构中，存储数据结构的存储空间可以不连续，各数据结点的存储顺序与数据元素之间的逻辑关系可以不一致，而数据之间的逻辑关系由指针域来确定的。

13. 答案：A

解析：二分法查找只用于顺序存储的有序线性表，而顺序查找用在顺序存储的非有序线性表和线性链表。

14. 答案：A

解析：对线性表进行检索，要求线性表是按顺序方式存储的，并按关键码的大小排好序，而不是按关键码检索频率排序。

二、填空题

1. 答案：有穷性

解析：算法的 4 个基本特性包括有穷性、确定性、可行性和拥有足够的情报，其中，算法的有穷性指的是算法必须在有限的时间完成，即算法必须在执行有限个步骤之后终止。

2. 答案：4

解析：当对表进行二分法查找时，先用 12 和表中间的元素 15 进行比较，接着与数据元素 8 进行比较，再与数据元素 10 进行比较，最后与 12 进行比较，查找成功，所以需要比较的次数为 4。

3. 答案：交换排序

解析：所谓排序是指将一个无序序列整理成按值非递减顺序排列成的有序列，常用的排序方法有交换排序、插入排序和选择排序。其中交换排序包括冒泡排序和快速排序，插入排序包括简单插入排序和希尔排序，选择排序包括直接选择排序和堆排序。

4. 答案：栈顶

解析：栈是限定在表的一端进行插入和删除操作的线性表。在表中，允许插入和删除的一端叫做“栈顶”，不允许插入和删除的一端叫做“栈底”。

第 2 章 习题分析与解答

一、选择题

1. 答案：C

解析：本题考察对面向对象的理解，面向对象的程序设计是对象模拟问题领域中的实体，各对象之间相独立，相互依赖性小，通过消息来实现对象之间的相互联系。

2. 答案：C

解析：对象的封装性是指从外部看只能看到对象的外部特征，即只需知道数据的取值范围和可以对该数据施加的操作，而不需要知道数据的具体结构以及实现操作的算法。对象的内部，即处理能力的实行和内部状态，对外是不可见的，从外面不能直接使用对象的处理能力，也不能直接修改其内部状态，对象的内部状态只能由其自身改变。

3. 答案：B

解析：源程序的文档化主要包括 3 点：①符号名应具有一定含义，便于理解程序功能；②正确的程序注释；③良好的视觉组织：利用空格、空行、缩进等技巧使程序层次清晰。

4. 答案：D

解析：类（class）描述的是具有相似属性与操作的一组对象，类是具体对象的实例。

5. 答案：D

解析：在程序设计中，程序流程图是必须的，程序的正确注释有助于读者理解程序，不是可有可无的。程序的结构化与模块化都是程序设计的基本原则，它们不存在矛盾的地方，程序要求模块化，每个模块要符合结构化原则。

6. 答案：B

解析：程序的 3 种基本控制结构包括顺序、选择和重复（循环），这 3 种结构就足以表达出各种其他形式的结构。

二、填空题

1. 答案：面向对象

解析：面向对象的程序设计方法中涉及的对象是系统中用来描述客观实体，是构成系统的一个基本单位。

2. 答案：封装性

解析：对象具有以下 3 个基本特点：标识唯一性、分类性、多态性、封装性和模块独立性。其中，封装性是指从外面看只能看到对象的外部特征，对象的内部特征即处理能力的实行和内部状态，对外是不可见的，对象的内部状态只能由其自身改变。

3. 答案：对象

解析：将操作相似的对象归为类，也就是说，类是具有共同属性，共同方法的对象的集合。

4. 答案：程序设计风格

解析：（程序设计风格）是指编写程序时所表现出的特点、习惯和逻辑思路。

第 3 章 习题分析与解答

一、选择题

1. 答案：D

解析：需求分析常用的工具有 4 种：数据流图（DFD）、数据字典（DD）、判断树和判定表。PAD（问题分析图）、PFD（程序流程图）、N-S（盒式图）都是详细设计的常用工具，不是需求分析的工具。

2. 答案：A

解析：根据软件生命周期模型，软件设计分为两个阶段，分别是概要设计设计和详细设计。

3. 答案：D

解析：本题考察对软件生命周期的理解，软件生命周期整体上可分为定义阶段、开发阶段、维护阶段。其中定义阶段包括可行性研究与计划制定和需求分析；开发阶段包括概要设计、详细设计、开发和测试；维护阶段是一个单独阶段，不包含在开发阶段内。它是所花费用最多的一个阶段。

4. 答案：B

解析：需求分析的最终结果是生成软件需要规格说明书，可以为用户、分析人员和设计人员之间的交流提供方便，可以直接支持目标软件系统的确认，又可以作为控制软件开发进程的依据。

5. 答案：C

解析：软件工程是建立并使用完善的工程化原则，以较经济的手段获得能在实际机器上有效运行的可靠软件的一系列方法。软件工程主要思想是强调在软件开发过程中需要应用工程化原则。

6. 答案：D

解析：DFD（数据流图 Data-Flow-Diagram）是描述数据处理过程的工具，是需求理解的逻辑模型的图形表示，它直接支持系统功能建模。在软件详细设计的过程阶段，要对每个模块规定的功能以及算法的设计给出适当的算法描述。常见的过程设计工具有：1 图形工具，即程序流程图、N-S、PAD、HTPO；2 表格工具，即判定表；3 语言工具，即 PDL（伪码）。

7. 答案：D

解析：程序调试是由程序开发者完成诊断和改正程序中错误的过程，软件测试是由专门的测试人员完成，是为了发现错误而执行程序的过程。软件维护是指软件系统交付使用以后，为了改正错误或满足新的需要而修改软件的过程，是软件生存周期中非常重要的一个阶段。

8. 答案：D

解析：软件工程包括 3 个要素，既方法、工具和过程。方法是完成软件工程项目的手段；工具支持软件的开发、管理和文档生成；过程支持软件开发的各个环节的控制、管理。

9. 答案：C

解析：软件工程管理主要包括软件管理学、软件工程学、软件心理学等内容。软件经济学是研究软件开发中成本的估算、成本效益分析的方法和技术，用经济学的基本原理来研究软件工程开发中的经济效益问题。

10. 答案：B

解析：需求分析是软件定义时期的最后一个阶段，可以概括为 4 个方面：①需求获取；②需求分析；③编写需求规格说明书；④需求评审。

11. 答案：B

解析：本题是对软件设计结构的考察，设计软件结构是在软件概要设计阶段进行的，而概要设计属于软件开发期。

12. 答案：A

解析：数据流图的主要图符元素有 4 种：①加工，用圆或椭圆表示，输入数据经过加工变换产生输出；②数据流，用箭头表示，沿箭头方向传送数据的通道；③存储文件，用双杠表示，处理过程中存放各种数据文件；④外部实体用方框表示，系统和环境接口。

二、填空题

1. 答案：软件生命周期

解析：软件产品从提出、实现、使用维护到停止使用退役的过程称为软件生命周期，可以将软件生命周期分为软件定义、软件开发及运行维护 3 个阶段。

答案：加工

解析：数据流图是从数据传递和加工的角度，来刻画数据流从输入到输出的移动变换过程。其中的每一个加工对应一个处理模块。

3. 答案：黑盒

解析：等价类划分法是一种典型的黑盒测试方法。它将程序的所有可能输入数据分成若干部分，然后从每个等价类中选取数据作为测试用例。

4. 答案：软件开发

解析：软件生命周期分为软件定义期、软件开发期和软件维护期。软件定义包括问题定义、可行性研究和需求分析 3 个阶段；软件开发期包括系统设计、详细设计、编码和测试 4 个阶段；软件维护即运行和维护阶段。

5. 答案：相关文档

解析：所谓计算机软件是计算机系统中与硬件相互依存的另一部分，是包括程序、数据及相关文档的完整集合。

6. 答案：软件工程环境

解析：本题考查对软件工程的理解，软件工程环境（或软件开发环境），全面支持软件开发过程的软件工具集合。

7. 答案：降低

解析：模块化是指解决一个复杂问题时，自顶向下逐层把软件系统划分成若干模块的过程，由此分解来降低程序设计的复杂性。

第 4 章 习题分析与解答

一、选择题

1. 答案：C

解析：数据定义语言 Data Definition Language（DDL）是 SQL 语言的一部分，用来定义概念模式、内模式和外模式。

2. 答案：D

解析：数据库设计目前一般采用生命周期法，即把整个数据库应用系统的开发分解成目标独立的 4 个阶段。它们分别是需求分析阶段、概念设计阶段、逻辑设计阶段、物理设计阶段。

3. 答案：A

解析：在数据库系统中，物理独立性是指数据的物理结构（包括存储结构、存取方式等）的改变，如存储设备的更换、物理存储的更换、存取方式改变等都不影响数据库的逻辑结构，从而不致引起应用程序的变化。

4. 答案：B

解析：数据库中的数据具有“集成”、“共享”的特点，即数据库集中了各种应用的数据，进行统一地构造与存储，而使它们能被不同的应用程序所使用，因而数据库技术的根本目标是解决数据的共享问题。

二、填空题

1. 答案：数据库系统

解析：在人工管理阶段，数据无法共享冗余度大，完全依赖于程序；在文件系统阶段，数据共享性差；而数据库系统具有三级模式及二级映射的抽象结构系统，从而保证了数据独立性的实现。

2. 答案：逻辑独立性

解析：数据的独立性是指数据和应用程序相互独立，分为物理独立性和逻辑独立性。其中物理独立性是指在数据的物理结构改变时，不需要修改应用程序，而逻辑独立性是指当逻辑结构改变时，不需要修改应用程序。

第 5 章 习题分析与解答

一、选择题

1. 答案：D

解析：在 Visual FoxPro 中，一个“表”就是一个关系；在 Visual FoxPro 中，一个关系存储为一个文件，文件扩展名是“.dbf”，称为“表”；在 Visual FoxPro 中，数据库是一个逻辑上的概念和手段，它通过一组系统文件将相互关联的数据库表及其相关的数据库对象统一管理和组织，它并不直接存储用户数据。

2. 答案：C

解析：计算机数据管理依次经历的几个阶段为：人工管理、文件系统、数据库系统、分布式数据库系统和面向对象数据库系统。

3. 答案：B

解析：在数据库系统中，数据库管理系统（DBMS）是数据库系统的核心。

4. 答案：D

解析：扩展名为.mnx 的文件是菜单文件。

5. 答案：A

解析：在数据库系统中，数据库管理系统（DBMS）是数据库系统的核心，是一个专门对数据库的建立、使用和维护进行管理的软件。

6. 答案：A

解析：从关系中找出满足给定条件的元组的操作称为选择。

7. 答案：B

解析：DBMS 为数据库管理系统。

8. 答案：D

解析：在一个关系中，不能出现相同的属性名；不能有完全相同的元组，即不能存在冗余；元组的次序无关紧要，即可以任意交换两行的次序；属性的次序无关紧要，即可以任意交换两列的次序。

9. 答案：A

解析：扩展名为.dbf 的文件是表文件。

10. 答案：D

解析：数据是指存储在某一种媒体上，能够被识别的物理符号。数据的形式可以是多种多样的，不仅有数字、字母、文字和其他特殊字符组成的文本形式的数据，还可以有图形、图像、声音和其他多媒体数据等非文本形式的数据。数据的概念包括两个方面的内容：描述事物特性的数据内容；存储在某一种媒体上的数据形式。

11. 答案：A

解析：关系模型是指用二维表表示实体以及实体之间联系的模型，因此关系数据库管理系统所管理的关系是若干个二维表。

12. 答案：C

解析：在 Visual FoxPro 中，用二维表来表示实体与实体间联系的数据模型称为关系模型。

13. 答案：C

解析：关系的基本运算有两类：一类是传统的集合运算（并、交、差等），另一类是专门的关系运算（选择、投影、连接）。

14. 答案：C

解析：在数据库系统中，数据库中数据项之间和记录之间都存在联系。

15. 答案：B

解析：在 Visual FoxPro 中，数据完整性包括实体完整性、域完整性和参照完整性。

16. 答案：B

解析：数据库（DB）、数据库系统（DBS）和数据库管理系统（DBMS）之间的关系是：DBS 包括 DB 和 DBMS。

17. 答案：C

解析：在数据库系统中，数据库管理系统（DBMS）是数据库系统的核心。

18. 答案：B

解析：参照完整性是指同一个数据库中的表中数据的完整性和一致性。

19. 答案：A

解析：参照完整性的前提是建立两个表之间的联系。

20. 答案：A

解析：数据库（Database，DB）是存储在计算机存储设备上、结构化的相关数据的集合。

21. 答案：D

解析：数据库技术解决了数据共享问题，因此大大减少了数据冗余。

22. 答案：D

解析：对象可以是具体的实物，也可以是一些概念。一条命令、一个人、一个桌子等都可以看做是一个对象，一个表单也可以看做是一个对象，一次考试也可以看做是一个对象。

23. 答案：C

解析：数据管理是数据处理的中心问题。

24. 答案：B

解析：DBMS 是操作系统支持下的系统软件。

25. 答案：B

解析：Visual FoxPro 是一种数据库管理系统。

26. 答案：B

解析：Visual FoxPro DBMS 基于的数据模型是关系型。

27. 答案：C

解析：Visual FoxPro 是一种关系型数据库管理系统，一个“表”就是一个关系，文件扩展名是“.dbf”。

28. 答案：D

解析：对关系的描述称为关系模式，一个关系模式对应一个关系的结构，其格式为：关系名(属性 1, 属性 2, ……，属性 n)。

29. 答案：B

解析：从关系模式中指定若干个属性组成新的关系称为投影。

30. 答案：C

解析：从表中选择字段形成新关系的操作是投影。

31. 答案：D

解析：进行数据库设计时，需要遵循以下主要原则：关系数据库的设计应遵从概念单一化“一事一地”的原则；避免在表之间出现重复字段；表中的字段必须是原始数据和基本数据元素；用外部关键字保证有关联的表之间的联系。

32. 答案：B

解析：规范化理论是关系数据库进行逻辑设计的理论依据。根据这个理论，关系数据库的关系必须满足其每一个属性都是不可分解的。

二、填空题

1. 答案：主关键字或候选关键字

解析：在关系数据库中，把数据表示成二维表，每一个二维表称为一个关系。

2. 答案：多对多

解析：游泳运动员与游泳比赛项目两个实体之间的联系是多对多联系。

3. 答案：.dbc

解析：Visual FoxPro 中，数据库文件的扩展名是.dbc。

4. 答案：主关键字或候选关键字

解析：在关系模型中，“关系中不允许出现相同元组”的约束是通过主关键字或候选关键字来实现的。

5. 答案：.pjx

解析：在 Visual FoxPro 中，项目文件的扩展名是.pjx。

6. 答案：.sct

解析：对于设计好的表单，系统会生成扩展名分别是.scx 和.sct 两个文件。

7. 答案：命令方式

解析：Visual FoxPro 的 3 种基本操作方式为：命令方式、菜单方式和程序方式。

8. 答案：面向对象

解析：Visual FoxPro 是运行于 Windows 平台的数据库管理系统，它在支持标准的面向过程的程序设计方式的同时还支持面向对象的程序设计方式。

9. 答案：相关事物之间的联系

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

10. 答案：多对多

解析：两个实体间的关系可分为 3 种类型：一对一、一对多和多对多。

11. 答案：外部关键字

解析：如果表中的一个字段不是本表的关键字或候选关键字，而是另一个表的关键字或候选关键字，那么这个字段（属性）就称为外部关键字。

12. 答案：关系运算

解析：通过关系运算可以查询关系数据库中用户需要的数据。

13. 答案：数据

解析：数据是存储在某一种媒体上能够识别的物理符号，其概念包含两个方面：一是描述事物特性的数据内容；二是存储在某一种媒体上的数据形式。

14. 答案：关系

解析：对某个关系进行选择、投影或联结运算后，运算的结果仍然是一个关系。

15. 答案：事物与事物

解析：数据库系统不仅可以表示事物内部各数据项之间的联系，而且还可以表示事物和事物之间的关系。

16. 答案：若干个二维表

解析：关系数据库管理系统所管理的数据库是若干个二维表。

17. 答案：数据模型

解析：任何一个数据库管理系统都是基于数据模型建立起来的。

18. 答案：QUIT

解析：QUIT 命令可以直接退出 Visual FoxPro 系统。

19. 答案：不能

解析：在基本表中，要求字段名不能重复。

20. 答案：数据库管理系统

解析：数据库系统中对数据库进行管理的核心软件是 DBMS，数据库管理系统。

第 6 章 习题分析与解答

一、选择题

1. 答案：C

解析：内存变量的数据类型包括字符型（C）、数值型（N）、货币型（Y）、逻辑型（L）、日期型（D）和日期时间型（T）。

2. 答案：C

解析：在 Visual FoxPro 中，数组在使用之前一般要用 DIMENSION 或 DECLARE 命令显式创建。

3. 答案：B

解析：在 Visual FoxPro 中，一个变量在使用之前并不需要特别的声明或定义，当给变量赋值时，如果该变量并不存在，那么系统会自动建立它。在数组中各元素相当于一个简单变量，数组元素可以分别存放不同类型的数据。数组创建后，系统自动给每个数组元素赋以逻辑假.F。数组大小由下标值的上、下限决定，下限规定为 1。

4. 答案：C

解析：M(2)的值就是 M(1, 2)的值 20。

5. 答案：C

解析：因为日期时间表达式的格式有一定限制，不能任意组合，不能用运算符“+”将两个<日期>连接起来。

6. 答案：D

解析：X\$Y 的结果为真，因此 NOT (X\$Y) 的结果为假。

7. 答案：B

解析：LIKE ('xy?', 'xyz') 的运算结果为逻辑真。

8. 答案：B

解析：“'[信息]'"\$管理信息系统'" 的运算结果为逻辑假。

9. 答案：B

解析：VARTYPE 函数用来测试<表达式>的类型，返回一个大写字母，函数值为字符型。

10. 答案：A

解析：在 Visual FoxPro 中，宏替换可以从变量中替换出字符变量的内容。

11. 答案：D

解析：X\$Y 的结果为真，因此 NOT (X\$Y) 的结果为假。

12. 答案：C

解析：M(2)的值就是 M(1, 2)的值 20。

13. 答案：B

解析：连续执行 SET 命令后的输出结果是：1949.05.01。

14. 答案：A

解析：表达式“'[信息]'"\$管理信息系统'" 返回结果为.F。

15. 答案：B

解析：在 Visual FoxPro 中，数据完整性包括实体完整性、域完整性和参照完整性。

16. 答案：C

解析：表达式 LEN (SPACE (0)) 的运算结果是 0。

17. 答案：A

解析：VARTYPE 函数用来测试<表达式>的类型，返回一个大写字母，函数值为字符型。

18. 答案：D

解析：字符表达式由字符串运算符将字符型数据连接起来形成，其运算结果仍然是字符型数据。“-”示连接前后两个字符串，并将前字符串的尾部空格移到合并后的新字符串尾部。

19. 答案：A

解析：SET CENTURY ON 命令可以将日期型或日期时间型数据中的年份用 4 位数字显示。

20. 答案：D

解析：表达式“b+RIGHT(a,2)”的结果是“微型机”。

21. 答案：B

解析：根据内存变量的书写格式要求，内存变量 X、Y、M、N 和 Z 的数据类型分别是：T、L、Y、N、C。

22. 答案：D

解析：如果当前表中存在一个和内存变量同名的字段变量，则在访问内存变量时，必须在变量名前加上前缀 M. 或 M->，否则系统将访问同名的字段变量。

23. 答案：B

解析：内存变量 s1 的运算结果是 netBIOS。

24. 答案：D

解析：说明数组后，数组元素的初值是逻辑假。

25. 答案：B

解析：表达式 “Right(a,4)” 的结果是 “考试”。

26. 答案：C

解析：该程序段的输出结果是 6。

27. 答案：B

解析：使用 “调试器” 调试程序时，跟踪窗口用于显示正在调试的程序文件。

28. 答案：D

解析：在 Visual FoxPro 中，用来建立程序文件的命令是：MODIFY COMMAND <文件名>。

29. 答案：A

解析：在 Visual FoxPro 中，程序文件的扩展名为 .prg。

30. 答案：C

解析：在 Visual FoxPro 中，代码片段是指一组命令，用于完成某一项操作。

31. 答案：B

解析：INPUT 命令等待用户从键盘输入数据，当用户回车后，系统将表达式的值存入指定的内存变量，程序继续运行。ACCEPT 命令等待用户从键盘输入字符串，当用户回车后，系统将表达式的值存入指定的内存变量，程序继续运行。WAIT 命令显示字符表达式的值作为提示信息，暂停程序的执行，直到用户按任意键或单击鼠标时才继续运行程序。

32. 答案：A

解析：INPUT 命令输入的数据可以是常量、变量，也可以是一般的表达式，但不能不输入任何内容而直接按回车键。ACCEPT 命令只能接收字符串。WAIT 命令也只能接收字符串。

33. 答案：D

解析：结构化程序设计的 3 种基本结构是：顺序结构、选择结构和循环结构。

34. 答案：B

解析：DO CASE 和 ENDCASE 必须成对出现，DO CASE 是本结构的入口，ENDCASE 是本结构的出口。不管有几个 CASE 条件成立，只有最先成立的那个 CASE 条件的对应命令序列被执行。如果所有的 CASE 条件都不成立，且没有 OTHERWISE 子句，则直接跳出本结构。

35. 答案：A

解析：在循环结构程序设计中，在指定范围内扫描数据文件，查找符合条件的记录并执行循环体中的命令，应使用的循环结构为：SCAN...ENDSCAN。

36. 答案：C

解析：过程文件的建立使用 MODIFY COMMAND 命令，过程文件的默认扩展名为 PRG，过程文件只包含过程，可以被其他程序所调用，在调用过程文件中的过程之前必须打开过程文件。

37. 答案：D

解析：在某个程序模块中用 **PRIVATE** 语句定义的内存变量，只能在定义该变量的模块及其下属模块中使用。

38. 答案：D

解析：按值传递方式中形参变量值的改变不会影响实参变量的取值，引用传递方式则刚好相反。

39. 答案：C

解析：局部变量只能被本层模块调用，而不能被上层或下层模块调用。

40. 答案：C

解析：在调试器中，可以显示程序、过程和方法程序中的变量、数组和对象的名称，以及当前取值和类型的窗口是局部窗口。

41. 答案：C

解析：在调试器中设置断点的类型有 4 种。

42. 答案：C

解析：选项 C 可以将变量 A，B 的值互换。

43. 答案：D

解析：执行该程序，显示结果为 40。

44. 答案：B

解析：程序运行的结果为：4。

45. 答案：B

解析：程序运行的结果为：及格。

46. 答案：D

解析：S 的显示值为：6。

47. 答案：A

解析：程序运行的结果为：.F.。

48. 答案：A

解析：程序运行的结果为：2 3 4; 3 4 5。

49. 答案：C

解析：在 Visual FoxPro 中，过程的返回语句是 **RETURN**。

50. 答案：A

解析：为了计算一个整数的各位数字之和，应填写的语句是：**x=int (x/10)**。

二、填空题

1. 答案：逻辑

解析：逻辑型常量是用来表示逻辑真和逻辑假的常量。逻辑型只有真和假两个值，逻辑真表示形式为：.T.、.t.、.Y.、.y.。逻辑假表示形式为：.F.、.f.、.N.、.n.。

2. 答案：日期时间型

解析：日期时间型常量是用花括号括起来的日期时间型数据序列，花括号内包括日期和时间两部分内容：{<日期>,<时间>}。

3. 答案：数值

解析：在 Visual FoxPro 中，“/”为除运算符，对于表达式 2005/4/2 来说，如果看做数学表达式则为 2005 除以 4 除以 2，则该表达式的值为 250.63，该值的数据类型为数值型。因为内存变量的类型取决于变量值的类型，所以内存变量 A 的数据类型为数值型。

4. 答案：SCATTER FIELDS 学号，姓名 TO A

解析：将表的当前记录复制到数组的命令为：SCATTER [FIELDS <字段名表>] TO <数组名>。因此，把当前表中当前记录的学号、姓名字段复制到数组 A 的命令是：SCATTER FIELDS 学号, 姓名 TO A。

5. 答案：数值型

解析：日期时间表达式中，<日期时间>-<日期时间> 是数值型，得到两个指定日期时间相差的秒数。

6. 答案：{^1962-10-27}

解析：日期型常量的定界符是一对花括号。花括号内包括年、月、日三部分内容，各部分内容之间用分隔符分隔。分隔符可以是斜杠 (/)、连字号 (-)、句点 (.) 和空格，其中斜杠是系统在显示日期型数据时使用的默认分隔符。因此，“1962 年 10 月 27 日”的日期常量应该写为 {^1962-10-27}。

7. 答案：日期时间型

解析：日期时间型常量是用花括号括起来的日期时间型数据序列，花括号内包括日期和时间两部分内容：{<日期>, <时间>}。

8. 答案：计算机等级二级 Visual FoxPro

解析：略。

9. 答案：13

解析：略。

10. 答案：EXIT

解析：在 Visual FoxPro 中，可以使用 EXIT 语句跳出 SCAN...ENDSCAN 循环体外执行 ENDSCAN 后面的语句。

11. 答案：KROW

解析：略。

第 7 章 习题分析与解答

一、选择题

1. 答案：B

解析：在 Visual FoxPro 中，数据库文件的扩展名为 .dbc。

2. 答案：D

解析：打开数据库的命令是：OPEN DATABASE。

3. 答案：A

解析：建立数据库的命令是：CREATE DATABASE。

4. 答案：A

解析：数据库表和自由表相比，可以为字段指定默认值。

5. 答案：B

解析：修改表结构的命令是：ALTER。

6. 答案：C

解析：打上删除标记的命令是：DELETE。“A*”表示以字母 A 开头的记录。

7. 答案：A

解析：添加字段“平均分数”的命令是：ALTER TABLE student ADD 平均分数 F(6, 2)。

8. 答案：B

解析：将所有性别为“女”的职工增加 100 元工资的命令是：REPLACE 工资 WITH 工资+100 FOR 性别=“女”。

9. 答案：D

解析：设置为当前索引的命令是：SET ORDER TO <索引标识>。

10. 答案: D

解析: 使用 SQL 语句增加字段的有效性规则, 是为了能保证数据的实体完整性。

11. 答案: A

解析: 参照完整性规则的更新规则中“级联”的含义是: 更新父表中连接字段值时, 用新的连接字段自动修改子表中的所有相关记录。

12. 答案: C

解析: 参照完整性规则的删除规则中: 如果删除规则选择的是“级联”, 则当用户删除父表中的记录时, 系统将自动删除子表中的所有相关记录; 如果删除规则选择的是“限制”, 则当用户删除父表中的记录时, 系统将禁止删除与子表相关的父表中的记录; 如果删除规则选择的是“忽略”, 则当用户删除父表中的记录时, 系统将不负责检查子表中是否有相关记录。

13. 答案: C

解析: 数据库表和自由表相比, 可以为字段指定默认值和有效性规则。

14. 答案: B

解析: 可以插入的记录是: ('20080101','cl','90','成绩优秀')。

15. 答案: A

解析: 在 Visual FoxPro 中, SORT 可以实现物理排序。

16. 答案: B

解析: 在 Visual FoxPro 中, 系统默认的工作区编号为 1。

17. 答案: A

解析: 在 Visual FoxPro 中, 每一个工作区中最多能打开数据库表的数量是 1 个。

18. 答案: B

解析: 无论是永久联系还是关联, 子表一定有索引。

19. 答案: B

解析: 执行 USE sc IN 0 命令的结果是: 选择空闲的最小号工作区打开 sc 表。

20. 答案: D

解析: 在表设计器的“字段”选项卡中可以创建的索引是普通索引。

21. 答案: D

解析: 参照完整性规则的删除规则中: 如果删除规则选择的是“级联”, 则当用户删除父表中的记录时, 系统将自动删除子表中的所有相关记录; 如果删除规则选择的是“限制”, 则当用户删除父表中的记录时, 系统将禁止删除与子表相关的父表中的记录; 如果删除规则选择的是“忽略”, 则当用户删除父表中的记录时, 系统将不负责检查子表中是否有相关记录。

22. 答案: B

解析: 使用索引可以加快数据库表的查询速度。

23. 答案: D

解析: 设置“参照完整性规则”, 要求在父表连接字段上建立主索引, 在子表连接字段上建立普通索引。

24. 答案: A

解析: 主索引和候选索引要求在指定字段或表达式中不允许出现重复值。

25. 答案: D

解析: 打开数据库的命令是: OPEN DATABASE。

26. 答案: C

解析: Visual FoxPro 的“参照完整性”中“插入规则”包括: 限制和忽略。

27. 答案：D

解析：在 Visual FoxPro 中，使用 LOCATE FOR <条件表达式>命令按条件查找记录，当查找到满足条件的第一条记录后，如果还需要查找下一条满足条件的记录，应使用 CONTINUE 命令。

28. 答案：B

解析：参照完整性规则的删除规则中：如果删除规则选择的是“限制”，则当用户删除父表中的记录时，系统将禁止删除与子表相关的父表中的记录。

29. 答案：A

解析：主索引和候选索引要求在指定字段或表达式中不允许出现重复值。

30. 答案：B

解析：主索引和候选索引要求在指定字段或表达式中不允许出现重复值。

31. 答案：D

解析：打开表并设置当前有效索引的命令是：USE <表名> IN <工作区号> ORDER <索引关键字>。

32. 答案：B

解析：数据库表的字段有效性规则是一个逻辑表达式。

33. 答案：C

解析：数据库的主要功能是对数据的查询，而索引就是在不改变表中记录物理顺序的前提下，按照索引关键字的顺序（升序或降序）排列记录，从而方便快速查找，但会降低数据更新速度。

34. 答案：A

解析：无论索引是否生效，定位到相同记录上的命令是：GO <记录号>。

35. 答案：B

解析：学号由 7 位数字组成，因此正确输入掩码是：“#####”。

36. 答案：A

解析：所谓自由表，就是不隶属于任何数据库的数据表。可以将自由表添加到数据库中，使之成为数据库表；也可以将数据库表从数据库中移出，使之成为自由表。数据库表和自由表相比，支持主关键字、参照完整性和表之间的联系；可以规定字段级规则和记录级规则；可以为字段指定默认值和输入掩码。

37. 答案：B

解析：在数据库中数据完整性是指保证数据正确的特性，数据完整性一般包括：实体完整性、域完整性和参照完整性。

38. 答案：C

解析：关系数据库中，在表之间建立永久性联系是通过联结两个表的字段来完成和体现的，这种联结是主表中的主关键字与子表中的外部关键字联结。

39. 答案：C

解析：删除规则中选择“级联”的含义是：当删除父表中的记录时，会自动删除子表中所有相关记录。

40. 答案：B

解析：修改规则中选择“限制”的含义是：当修改父表中的记录时，如果父表中所有的记录在子表中都有相关关联的记录，则不允许修改父表中的相关记录。

41. 答案：C

解析：在 Visual FoxPro 中，空值（NULL）是表示字段还没有确定值。

42. 答案：C

解析：在 Visual FoxPro 中，索引就是在不改变表中记录物理顺序的前提下，按照索引关键字的顺序（升序或降序）排列记录，从而方便快速查找。建立索引是创建一个索引文件，该文件包含有指向表记录的指针。索引的数据

将与表的数据存储在一个物理文件中。

43. 答案: C

解析: 在 Visual FoxPro 中, 在数据库中创建表的 CREATE TABLE 命令中定义主索引, 实现实体完整性规则的短语是: PRIMARY KEY。

44. 答案: C

解析: 在 Visual FoxPro 中, 数据库表的字段或记录的有效性规则的设置是在表设计器中进行。

45. 答案: A

解析: 在 Visual FoxPro 的数据库表中只能有一个主索引。

46. 答案: A

解析: 建立一个索引, 要求首先按职称排序, 职称相同时再按性别排序, 正确的命令是: INDEX ON 职称+性别 TO ttt。

二、填空题

1. 答案: 实体

解析: 在 Visual FoxPro 中, 主索引是指在指定字段或表达式中不允许出现重复值的索引, 因此可以保证数据的实体完整性。

2. 答案: 逻辑或条件

解析: 在定义字段有效性规则时, 在规则框中应输入关系表达式或者逻辑表达式。

3. 答案: Remove

解析: 可以使用 REMOVE TABLE 命令一个数据库表从当前数据库中移去, 使之成为自由表。

4. 答案: 级联

解析: 当删除父表记录时, 被关联子表相关记录自动删除。这是参照完整性的删除规则中的级联规则。

5. 答案: 数据库

解析: 数据库表和自由表相比, 可以为字段指定默认值。

6. 答案: 主索引或候选索引

解析: 在 Visual FoxPro 中, 可以通过主索引或候选索引使得数据库表中不允许有重复的记录。

7. 答案: FPT

解析: 在 Visual FoxPro 中, 通用型字段的内容将存储在.fpt 文件中。

8. 答案: .CDX

解析: 在 Visual FoxPro 中, 结构化复合索引文件的扩展名是.cdx。

9. 答案: 数据库

解析: 数据库表和自由表相比, 可以规定字段级规则和记录级规则。

10. 答案: 临时

解析: 取消表之间已经存在的临时联系的命令是: SET RELATION TO。

11. 答案: 主关键字

解析: 在 Visual FoxPro 中, 数据库表的参照完整性是通过表之间的联系建立起来的。

12. 答案: 主关键字

解析: 在 Visual FoxPro 中, 是通过主关键字或候选关键字来保证表中记录(元组)的唯一, 即实体完整性。

13. 答案: .dbc

解析: 在 Visual FoxPro 中, 数据库文件的扩展名是.dbc。

14. 答案: 数据库

解析：所谓自由表，就是不隶属于任何数据库的数据表。

15. 答案：主

解析：一个数据库表只能有一个主索引。

16. 答案：域完整性

解析：在数据库中数据完整性是指保证数据正确的特性，数据完整性一般包括实体完整性、域完整性和参照完整性。

17. 答案：删除规则

解析：参照完整性规则包括更新规则、删除规则和插入规则。

18. 答案：结构复合索引

解析：结构化复合索引文件可以伴随着表的打开而自动打开。

19. 答案：逻辑或条件

解析：在定义字段有效性规则时，在规则框中应输入关系表达式或者逻辑表达式。

20. 答案：实体

解析：在 Visual FoxPro 中，通过建立数据库表的主索引可以实现数据的实体完整性。

第 8 章 习题分析与解答

一、选择题

1. 答案：C

解析：SQL 的修改表结构命令为 ALTER TABLE 命令。

2. 答案：D

解析：SQL 的数据更新命令为 UPDATE 命令，其格式如下：UPDATE <表名> SET <字段名 1>=<表达式 1>[, <字段名 2>=<表达式 2>...] WHERE <条件>。

3. 答案：A

解析：NOT(工资>3000 OR 工资<2000)等价于 BETWEEN 2000 AND 3000。

4. 答案：B

解析：SQL 的 ALTER TABLE 命令可以定义、修改和删除字段有效性规则和默认值定义，其格式如下：ALTER TABLE <表名> ALTER [COLUMN] <字段名> [SET CHECK <逻辑表达式>[ERROR <字符型文本信息>]]。职工号的最左边三位字符是 110 可以表示为：LEFT(职工号,3)="110"。

5. 答案：A

解析：该语句的作用是在教师表中查找所有工资大于等于通信系教师的记录，显示他们的系号。因此，根据语句的内容，该语句的执行结果为系号等于“01”和“02”。

6. 答案：D

解析：该语句的作用是在教师表中按照主讲课程分组计数。根据题干内容，由于一共有 6 门主讲课程，因此，该语句的执行结果为 6。

7. 答案：C

解析：与题干等价的 SQL 语句是：SELECT 学院.系名, COUNT(*) AS 教师人数 FROM 教师 INNER JOIN 学院。

8. 答案：A

解析：与题干等价的 SQL 语句是：SELECT DISTINCT 系号 FROM 教师 WHERE 工资>=(SELECT MAX(工资) FROM 教师 WHERE 系号="02") 教师.系号 = 学院.系号 GROUP BY 学院.系名。

9. 答案：A

解析: SQL 的数据删除命令为 DELETE 命令。在 Visual FoxPro 中, SQL DELETE 命令仅仅是逻辑删除记录, 如果要物理删除记录, 那么还需要继续使用 PACK 命令。同样, 传统的 Visual FoxPro 的 DELETE 命令也只能逻辑删除数据库表中的记录。传统 Visual FoxPro 的 DELETE 命令在删除数据库表中的记录之前, 需要用 USE 命令打开表。而 SQL 的 DELETE 命令在删除数据库表中的记录之前, 不需要用 USE 命令打开表。

10. 答案: B

解析: SQL 的数据删除命令格式如下: DELETE FROM <表名> [WHERE <条件>]。

11. 答案: C

解析: SQL INSERT 命令用于在一个表中添加新记录, 然后给新记录的字段赋值。其格式为: INSERT INTO <表名> [(<字段名 1>[, <字段名 2>, ...])] VALUES (<表达式 1>[, <表达式 2>, ...])。

12. 答案: C

解析: 在 Visual FoxPro 的 SELECT 语句中, 使用 TOP <数值表达式> [PERCENT] 短语可以显示满足条件的前几条记录。其中, 当不使用 PERCENT 时, <数值表达式> 是 1~32767 间的整数, 说明显示前几个记录; 当使用 PERCENT 时, <数值表达式> 是 0.01~0.99 之间的实数, 说明显示结果中前百分之几的记录。TOP 短语要与 ORDER BY 短语同时使用才有效。

13. 答案: B

解析: 利用 GROUP BY 子句, 可以进行分组计算查询。该子句可以按一列或多列分组, 还可以用 HAVING 进一步限定分组的条件。GROUP BY 子句一般跟在 WHERE 子句之后, 没有 WHERE 子句时, 跟在 FROM 子句之后。“订单数在 3 个以上、订单的平均金额在 200 元以上” 可以表示为: COUNT(*)>3 AND AVG(金额)>200。

14. 答案: C

解析: 显示没有签订任何订单的职员信息 (职员号和姓名), 正确的 SQL 语句是: SELECT 职员号, 姓名 FROM 职员 WHERE 职员号 NOT IN (SELECT 职员号 FROM 订单)。

15. 答案: C

解析: SQL 的数据删除命令格式如下: DELETE FROM <表名> [WHERE <条件>]。

16. 答案: A

解析: SQL 的数据更新命令为 UPDATE 命令, 其格式如下: UPDATE <表名> SET <字段名 1>=<表达式 1>[, <字段名 2>=<表达式 2>...] WHERE <条件>。

17. 答案: D

解析: SQL 的数据操作语句包括 INSERT、UPDATE 和 DELETE。

18. 答案: C

解析: SQL 的数据删除命令格式如下: DELETE FROM <表名> [WHERE <条件>]。

19. 答案: C

解析: 在 SQL 的 SELECT 查询结果中, 使用 DISTINCT 子句可以消除重复记录。

20. 答案: D

解析: SELECT 子句中可以包含表中的列和表达式, 可以使用别名, SELECT 子句规定了结果集中的列顺序, 其顺序可以与表中列的顺序不一致。

21. 答案: B

解析: 利用 GROUP BY 子句, 可以进行分组计算查询。该子句可以按一列或多列分组, 还可以用 HAVING 进一步限定分组的条件。HAVING 子句总是跟在 GROUP BY 子句之后, 不可以单独使用。HAVING 子句和 WHERE 子句并不矛盾, 在查询中是首先用 WHERE 子句限定分组, 然后进行分组, 最后再用 HAVING 子句限定分组。

22. 答案: A

解析: 查询选修的每门课程的成绩都高于或等于 85 分的学生的学号和姓名, 正确的 SQL 语句是: SELECT 学号, 姓

名 FROM S WHERE NOT EXISTS (SELECT * FROM SC WHERE SC.学号=S.学号 AND 成绩<85)。

23. 答案：B

解析：利用 GROUP BY 子句，可以进行分组计算查询。该子句可以按一列或多列分组，还可以用 HAVING 进一步限定分组的条件。GROUP BY 子句一般跟在 WHERE 子句之后，没有 WHERE 子句时，跟在 FROM 子句之后。

24. 答案：D

解析：利用 GROUP BY 子句，可以进行分组计算查询。ORDER BY：用来对查询结果进行排序。

25. 答案：B

解析：SQL SELECT 语句的功能是查询。

26. 答案：C

解析：在 SQL 语句中用于限定条件的短语是 WHERE。

27. 答案：A

解析：使用短语 INTO DBF|TABLE <表名>可以将查询结果存放到永久表中（.dbf 文件）。但是 stock 表已经存在而且正处于打开状态，因此系统会提示语句出错。

28. 答案：A

解析：与 BETWEEN 12.76 AND 15.20 等价的语句是：单价<=15.20 AND 单价>=12.76。

29. 答案：A

解析：使用 INTO ARRAY <数组名>短语可以将查询结果存放到数组中，因此 arr[1]内容为 15.20。

30. 答案：C

解析：SQL 的修改表结构命令为 ALTER TABLE 命令。其格式为：ALTER TABLE <表名 1> ALTER [COLUMN] <字段名><类型>[(<字段宽度>[, <小数位数>])]。

31. 答案：B

解析：满足条件 WHERE 交易所="深圳"的记录数为 2，因此建立的视图含有的记录个数是 2。

32. 答案：B

解析：视图是根据对表的查询定义的，其命令格式如下：CREATE VIEW <视图名> AS SELECT 语句。其中：可以是任意的 SELECT 查询语句，它说明和限定了视图中的数据；视图的字段名将与 SELECT 语句中指定的字段名或表中的字段名同名。因此，建立的视图含有的字段名是：名称，单价。

33. 答案：A

解析：满足条件 WHERE 单价=(SELECT MIN (单价) FROM stock)的记录个数为 2，然而 DISTINCT 子句的作用是去掉查询结果中的重复记录，因此执行该语句后，记录个数是 1。

34. 答案：D

解析：利用 GROUP BY 子句，可以进行分组计算查询。ORDER BY：用来对查询结果进行排序。

35. 答案：B

解析：利用 GROUP BY 子句，可以进行分组计算查询。“GROUP BY 交易所”指按照交易所分组，因此分为两组，进行求平均计算，得到结果是：9.99。

36. 答案：A

解析：执行该 SQL 语句的结果是：查询选修课程名称为“税收基础”的学员学号和姓名。

37. 答案：C

解析：查询选修课程编号为“C2”的学员姓名和所属单位，正确的 SQL 命令是：SELECT S.SN,S.SD FROM S,SC WHERE S.[S#]=SC.[S#] AND SC.[C#]='C2'。

38. 答案：B

解析：执行该 SQL 语句的结果是：查询不选修课程编号为“C5”的学员姓名和所属单位。

39. 答案: D

解析: 查询选修全部课程的学员姓名和所属单位, SQL 命令中缺少 GROUP BY 短语。

40. 答案: A

解析: 查询选修课程超过 5 门的学员学号和所属单位, SQL 命令中缺少 HAVING 短语, 用来进一步限定分组的条件。HAVING 子句总是跟在 GROUP BY 子句之后, 不可以单独使用。

41. 答案: B

解析: 查询选修了课程的学员人数, 正确的 SQL 命令是: SELECT 学员人数=COUNT(DISTINCT [S#]) FROM SC。

42. 答案: D

解析: HAVING 短语, 用来进一步限定分组的条件。HAVING 子句总是跟在 GROUP BY 子句之后, 不可以单独使用。

43. 答案: D

解析: INSERT 命令用于在一个表中添加新记录, 然后给新记录的字段赋值。其格式为: INSERT INTO <表名> [(<字段名 1>[, <字段名 2>, ...])] VALUES (<表达式 1>[, <表达式 2>, ...])。其中: 字段名表, 当插入的不是完整的记录时, 用其指定字段。如果省略字段名表, 则按照表结构中字段定义的顺序一次指定每个字段的值。

44. 答案: D

解析: 表达式 “年龄 BETWEEN 12 AND 46” 等价于 “年龄 >=12 AND 年龄 <=46”。

45. 答案: A

解析: SQL 的修改表结构命令为 ALTER TABLE 命令。其格式为: ALTER TABLE <表名 1> [DROP [COLUMN] <字段名 1>], 可以删除字段 (DROP [COLUMN])。

46. 答案: A

解析: ORDER BY 短语, 用来对查询结果进行排序, 格式为: [ORDER BY <排序表达式>[ASC|DESC]。<排序表达式>可以字段名或者字段的序号。因此, 查询所有年龄小于等于 18 岁的女同学, 并按年龄进行降序生成新的表 WS, 正确的 SQL 命令是: SELECT * FROM S WHERE 性别="女" AND 年龄 <=18 ORDER BY 4 DESC INTO TABLE WS。

47. 答案: D

解析: 查询同时选修课程号为 “C1” 和 “C5” 的学生学号, 正确的 SQL 命令是: SELECT 学号 FROM SC WHERE 课程号=

48. 答案: D

解析: 查询学号、姓名和学生所选课程名和成绩, 正确的 SQL 命令是: SELECT S.学号, 姓名, 课程名, 成绩 FROM S JOIN SC JOIN C ON SC.课程号=C.课程号 ON S.学号=SC.学号。

49. 答案: B

解析: 在 SQL SELECT 语句中, 为了将查询结果存储到临时表应该使用短语 INTO CURSOR。

50. 答案: D

解析: 在 SQL 的 ALTER TABLE 语句中, 为了增加一个新的字段应该使用短语 ADD。

51. 答案: A

解析: 查询所有 1982 年 3 月 20 日以后 (含) 出生、性别为男的学生, 正确的 SQL 语句是: SELECT * FROM 学生 WHERE 出生日期 >={ ^ 1982-03-20 } AND 性别="男"。

52. 答案: D

解析: 计算刘明同学选修的所有课程的平均成绩, 正确的 SQL 语句是: SELECT AVG(成绩) FROM 学生, 选课 WHERE 学生.学号=选课.学号 AND 姓名="刘明"。

53. 答案: B

解析: 计算各专业学生选修课程号为 “101” 课程的平均成绩, 正确的 SQL 语句是: SELECT SUBS(学号,3,2) AS 专业,AVG(成绩) AS 平均分 FROM 选课 WHERE 课程号="101" GROUP BY 1。

54. 答案：B

解析：查询选修课程号为“101”课程得分最高的同学，正确的 SQL 语句是：SELECT 学生.学号,姓名 FROM 学生,选课 WHERE 学生.学号=选课.学号 AND 成绩>=ALL(SELECT 成绩 FROM 选课 WHERE 课程号="101")。

55. 答案：A

解析：INSERT 命令用于在一个表中添加新记录，然后给新记录的字段赋值。其格式为：INSERT INTO <表名> [(<字段名 1>[, <字段名 2>, ...])] VALUES (<表达式 1>[, <表达式 2>, ...])。其中：字段名表，当插入的不是完整的记录时，用其指定字段。如果省略字段名表，则按照表结构中字段定义的顺序一次指定每个字段的值。

56. 答案：B

解析：SQL 的数据更新命令为 UPDATE 命令，其格式如下：UPDATE <表名> SET <字段名 1>=<表达式 1>[, <字段名 2>=<表达式 2>...] WHERE <条件>。

57. 答案：B

解析：删除视图的命令格式如下：DROP VIEW <视图名>。

58. 答案：A

解析：查询并降序排序，正确的 SQL 命令是：SELECT * FROM order WHERE YEAR(签订日期)=2007 ORDER BY 金额 DESC。

59. 答案：C

解析：SQL 的数据删除命令格式如下：DELETE FROM <表名> [WHERE <条件>]。

60. 答案：C

解析：ALTER TABLE <表名 1> ADD|ALTER [COLUMN] <字段名><类型>[(<字段宽度>[, <小数位数>])用来向表中添加新的字段或者修改已有的字段。

61. 答案：D

解析：SQL 的数据更新命令为 UPDATE 命令，其格式如下：UPDATE <表名> SET <字段名 1>=<表达式 1>[, <字段名 2>=<表达式 2>...] WHERE <条件>。

62. 答案：A

解析：查询“投中 3 分球”小于等于 5 个的运动员中“得分”最高的运动员的“得分”，正确的 SQL 语句是：SELECT MAX(得分) 得分 FROM 运动员 WHERE 投中 3 分球<=5。

63. 答案：D

解析：在 SQL SELECT 查询中，为了使查询结果排序应该使用短语 ORDER BY。

64. 答案：A

解析：在 SQL SELECT 语句中与 INTO TABLE 等价的短语是 INTO DBF。

65. 答案：B

解析：查询单价在 600 元以上的主机板和硬盘，正确 SQL 命令是：SELECT * FROM 产品 WHERE 单价>600 AND (名称='主板' OR 名称='硬盘')。

66. 答案：D

解析：查询客户名称中有“网络”二字的客户信息，正确 SQL 命令是：SELECT * FROM 客户 WHERE 名称 LIKE “%网络%”。

67. 答案：A

解析：查询订购单的数量和所有订购单平均金额，正确 SQL 命令是：SELECT COUNT(DISTINCT 订单号),AVG(数量*单价)FROM 产品 JOIN 订购单名细 ON 产品.产品号=订购单名细.产品号。

68. 答案：D

解析：由于订购单表参照客户表，因此当插入子表（订购单）中一条新记录时，如果父表（客户）中没有相关的记

录, 则禁止插入。另外, 客户表中有客户号是关键字, 关键字的值是惟一的。因此, 可以正确执行的命令只有: INSERT INTO 订购单 VALUES ('OR9','C5',{^2008/10/10})。

69. 答案: D

解析: SQL 语言的查询语句是 SELECT。

70. 答案: D

解析: SQL 表定义语句 (CREATE TABLE) 可以定义一个新的基本表结构, 可以定义表中的主关键字, 可以定义数据库表的域完整性、字段有效性规则等, 如果建立自由表, 则不能实现其完整性、有效性规则等信息的设置。

71. 答案: A

解析: 条件“研究生导师的女老师”所对应的<逻辑表达式>是: 研究生导师 AND 性别=“女”。

二、填空题

1. 答案: Column

解析: ALTER TABLE <表名> ADD [COLUMN] <字段名>: 向表中添加新的字段。

2. 答案: ADD; CHECK

解析: ALTER TABLE <表名> ADD [COLUMN] <字段名>: 向表中添加新的字段。ALTER TABLE <表名> [SET CHECK <逻辑表达式>[ERROR <字符型文本信息>]]: 定义字段有效性规则。

3. 答案: ON

解析: SELECT ...FROM <表名 1> JOIN <表名 2> ON <连接条件>: 连接查询。

4. 答案: UPDATE; SET

解析: SQL 的数据更新命令为 UPDATE 命令, 其格式如下: UPDATE <表名> SET <字段名 1>=<表达式 1>[, <字段名 2>=<表达式 2>...] WHERE <条件>。

5. 答案: CHECK

解析: 在 Visual FoxPro 中, 使用 SQL 的 CREATE TABLE 语句建立数据库表时, 使用 CHECK 子句说明有效性规则 (域完整性规则或字段取值范围)。

6. 答案: DISTINCT

解析: 在 SQL 的 SELECT 查询结果中, 使用 DISTINCT 子句可以消除重复记录。

7. 答案: DELETE

解析: SQL 的操作语句包括 INSERT、UPDATE 和 DELETE。

8. 答案: COUNT

解析: COUNT 函数, 用于计数计算。

9. 答案: DROP COLUMN

解析: ALTER TABLE <表名> [DROP [COLUMN] <字段名>]用于从指定表中删除指定的字段。

10. 答案: INSERT INTO 学院 VALUES ("04", "计算机")

解析: 插入一条新的记录: INSERT INTO 学院 VALUES ("04", "计算机")。

11. 答案: SUM; 系号

解析: SUM 函数, 用于求和计算。

12. 答案: UPDATE 教师 SET 工资=工资*1.05 WHERE 职称="教授"

解析: SQL 的数据更新命令格式如下: UPDATE <表名> SET <字段名 1>=<表达式 1>[, <字段名 2>=<表达式 2>...] WHERE <条件>。

13. 答案: HAVING

解析: 当 SUM、COUNT 等函数用在 SQL SELECT 语句中作为条件时, 必须使用的短语是 HAVING。

14. 答案: NOT EXISTS

解析: 条件为“没有选修过“李明”老师讲授课程”, 因此, 使用 NOT EXISTS 短语。

15. 答案: SC.SCGRADE

解析: 查询有两门以上(含两门)不及格课程的学生姓名及其平均成绩, 求平均成绩使用 AVG (SC.SCGRADE)。

16. 答案: '1','2'

解析: 查询既学过“1”号课程, 又学过“2”号课程的所有学生姓名, 使用 IN ("1", "2")。

17. 答案: SC1.SCGRADE>SC2.SCGRADE

解析: 查询“1”号课成绩比“2”号同学该门课成绩高的所有学生的学号, 因此使用表达式 SC1.SCGRADE>SC2.SCGRADE。

18. 答案: TOP 10 DESC

解析: 在 Visual FoxPro 的 SELECT 语句中, 使用 TOP <数值表达式> [PERCENT]短语可以显示满足条件的前几条记录。其中, 当不使用 PERCENT 时, <数值表达式>是 1~32767 间的整数, 说明显示前几个记录; 当使用 PERCENT 时, <数值表达式>是 0.01~0.99 之间的实数, 说明显示结果中前百分之几的记录。TOP 短语要与 ORDER BY 短语同时使用才有效。升序为 ASC, 降序为 DESC。

19. 答案: ALTER

解析: 修改表结构的 SQL 命令为 ALTER TABLE 命令, UNIQUE 为字段创建一个候选索引, 索引标识名与字段名相同。

20. 答案: IS NULL

解析: 查询空值的记录通常在 WHERE 子句中采用“字段名 IS NULL”或“字段名 IS NOT NULL”来限定条件。

21. 答案: GROUP BY

解析: 利用 GROUP BY 子句, 可以进行分组计算查询。该子句可以按一列或多列分组, 还可以用 HAVING 进一步限定分组的条件。HAVING 子句总是跟在 GROUP BY 子句之后, 不可以单独使用。

22. 答案: UPDATE

解析: 在 SQL 中, 插入、删除、更新命令依次是 INSERT、DELETE 和 UPDATE。

23. 答案: LIKE

解析: 在 SQL 的 WHERE 子句的条件表达式中, 字符串匹配(模糊查询)的运算符是 LIKE。

24. 答案: PRIMARY KEY

解析: 在使用 SQL 的 CREATE TABLE 语句定义表结构时, 用 PRIMARY KEY 短语说明关键字(主索引)。

25. 答案: AGE IS NULL

解析: 查询空值的记录通常在 WHERE 子句中采用“字段名 IS NULL”或“字段名 IS NOT NULL”来限定条件。

第9章 习题分析与解答

一、选择题

1. 答案: D

解析: 所谓视图, 是一个定制的虚拟表或逻辑表, 它是从一个已经存在或多个相关联的数据表根据需要组织起来的查看数据的一个窗口, 通过它可以查看表中有用的内容。与查询不同的是, 使用视图从表中提取一组记录, 如果改变了记录的值, 变更的结果会被送回到基本表中; 而查询却是“只读”的, 不能更新数据。

2. 答案: C

解析: SQL 的数据更新命令为 UPDATE 命令, 其格式如下: UPDATE <表名> SET <字段名 1>=<表达式 1>[, <字段名 2>=<表达式 2>...] WHERE <条件>。

3. 答案: D

解析: 利用 SQL 命令 CREATE VIEW <视图名> AS SELECT 查询语句。GROUP BY 后面应该是系号, 而不是系名。

4. 答案: B

解析: 删除视图的命令是: DROP VIEW <视图名>。

5. 答案: C

解析: 在“查询设计器”对话框中, 有 6 个选项卡: “字段”、“联接”、“筛选”、“排序依据”、“分组依据”、“杂项”。这些选项卡与 SQL SELECT 语句的各个短语相互对应。“联接”选项卡与 SQL 语句的 JOIN ON 短语对应; “筛选”选项卡与 SQL 语句的 WHERE 短语对应; “排序依据”选项卡与 SQL 语句的 ORDER BY 短语对应; “分组依据”选项卡与 SQL 语句的 GROUP BY 短语对应。

6. 答案: B

解析: 视图是数据库中的一个特有功能, 只有在包含视图的数据库打开时, 才能使用视图。

7. 答案: B

解析: 运行查询的命令是: DO <查询文件名>。

8. 答案: D

解析: 查询以扩展名为.qpr 的文件保存在磁盘中, 可以根据需要直接或反复使用, 从而可以提高数据使用的效率。

9. 答案: B

解析: 所谓查询, 就是一个预先定义好的, 可以从指定的表或视图中查找满足条件的记录

10. 答案: B

解析: 在查询设计器中, 选择查询去向是“表”, 则原有的 SQL SELECT 语句后面增加的短语是 INTO TABLE。

11. 答案: C

解析: 与查询不同的是, 使用视图从表中提取一组记录, 如果改变了记录的值, 变更的结果会被送回到基本表中; 而查询却是“只读”的, 不能更新数据。因此, 在查询设计器中没有, 但在视图设计器中含有的选项卡是“更新条件”。

12. 答案: A

解析: 视图设计器中包括 7 个选项卡: “字段”、“联接”、“筛选”、“排序依据”、“分组依据”、“杂项”和“更新条件”。

13. 答案: D

解析: 所谓视图, 是一个定制的虚拟表或逻辑表, 它是从一个已经存在或多个相关联的数据表根据需要组织起来的查看数据的一个窗口, 通过它可以查看表中有用的内容。与查询不同的是, 使用视图从表中提取一组记录, 如果改变了记录的值, 变更的结果会被送回到基本表中; 而查询却是“只读”的, 不能更新数据。

14. 答案: D

解析: 在 Visual FoxPro 中, 可以通过视图设计器建立视图; 也可以 SQL 命令 CREATE VIEW ... AS ... 建立视图。

15. 答案: B

解析: 在 Visual FoxPro 中, 可以用“视图设计器”建立视图, 但是只有真正了解了 SQL SELECT 语句才能设计出好视图。

16. 答案: D

解析: 所谓查询, 就是一个预先定义好的, 可以从指定的表或视图中查找满足条件的记录。

17. 答案: B

解析: 与查询不同的是, 使用视图从表中提取一组记录, 如果改变了记录的值, 变更的结果会被送回到基本表中; 而查询却是“只读”的, 不能更新数据。因此, 在查询设计器中没有, 但在视图设计器中含有的选项卡是“更新条件”。

18. 答案: D

解析: 在使用查询设计器创建查询时, “杂项”选项卡可以用来指定是否要重复记录, 对应于 SQL SELECT 语句的

DISTINCT 短语。

19. 答案：A

解析：运行查询的命令是：DO <查询文件名>。

20. 答案：A

解析：所谓视图，是一个定制的虚拟表或逻辑表，它是从一个已经存在或多个相关联的数据表根据需要组织起来的查看数据的一个窗口，通过它可以查看表中有用的内容。与查询不同的是，使用视图从表中提取一组记录，如果改变了记录的值，变更的结果会被送回到基本表中；而查询却是“只读”的，不能更新数据。查询是从指定的表或视图中提取满足条件的记录，然后按照想得到的输出类型定向输出查询结果，诸如浏览器、报表，表、标签等。

21. 答案：D

解析：所谓查询，就是一个预先定义好的，可以从指定的表或视图中查找满足条件的记录，然后按照希望得到的输出类型定向输出结果的 SQL SELECT 语句。视图是基于数据库的一张虚拟表，所以创建视图只能在数据库中建立。

22. 答案：A

解析：在 Visual FoxPro 的查询设计器中“筛选”选项卡对应的 SQL 语句是“WHERE”短语。

23. 答案：C

解析：所谓视图，是一个定制的虚拟表或逻辑表，它是从一个已经存在或多个相关联的数据表根据需要组织起来的查看数据的一个窗口，通过它可以查看表中有用的内容。视图是基于数据库的一张虚拟表，所以创建视图只能在数据库中建立。

24. 答案：A

解析：在“查询设计器”对话框中，有 6 个选项卡：“字段”、“联接”、“筛选”、“排序依据”、“分组依据”、“杂项”。

25. 答案：D

解析：删除视图的命令是：DROP VIEW <视图名>。

二、填空题

1. 答案：条件

解析：查询设计器的筛选选项卡是用来指定查询的条件。

2. 答案：远程

解析：按照建立视图的数据源的不同，可以分为本地视图和远程视图。如果视图中有取自远程数据源（如网络服务器）的数据，则称为远程视图，否则为本地视图。

3. 答案：发送 SQL 更新

解析：如果需要通过视图向基本表发送更新，则选择“视图设计器”对话框中“发送 SQL 更新”选项。

4. 答案：ORDER BY

解析：查询设计器的“排序依据”选项卡对应 SQL SELECT 语句的 ORDER BY 短语。

5. 答案：JOIN ON

解析：查询设计器的“连接”选项卡对应于 SQL SELECT 语句的 JOIN ON 短语。

6. 答案：WHERE

解析：查询设计器的“筛选”选项卡对应于 SQL SELECT 语句的 WHERE 短语。

7. 答案：GROUP BY

解析：查询设计器的“分组”选项卡对应于 SQL SELECT 语句的 GROUP BY 短语和 HAVING 短语，用于分组。

8. 答案：QPR

解析：查询以扩展名为.qpr 的文件保存在磁盘中，可以根据需要直接或反复使用，从而可以提高数据使用的效率。

9. 答案：视图

解析：所谓视图，是一个定制的虚拟表或逻辑表，它是从一个已经存在或多个相关联的数据表根据需要组织起来的查看数据的一个窗口，通过它可以查看表中有用的内容。之所以称之为虚拟表，是因为只能在数据库中看到而在磁盘上看不到它。

第 10 章 习题分析与解答

一、选择题

1. 答案：D

解析：所谓类，是指一组具有相同属性和方法的对象的集合。属性用于描述对象的状态，方法用于表示对象的行为。基于类就可以生成这类对象中的任何一个对象，称为这个类的“实例”。这些“实例”具有相同的属性和方法，但是它们的属性取值却并不相同，同一个方法由不同的对象去执行，也会产生不同的结果。

2. 答案：C

解析：SetFocus 可以让控件获得焦点，使其成为活动对象。

3. 答案：B

解析：Release 是表单方法名；Init、Destroy 是表单事件名；Caption 是表单属性名。

4. 答案：C

解析：ThisForm.Release 可以实现关闭该表单。

5. 答案：A

解析：ThisForm.Caption="登录窗口"可以使表单的标题显示“登录窗口”。

6. 答案：B

解析：“选项组”的 Value 属性，用来指定选项按钮当前状态，即用户选择了哪个按钮。该属性的类型可以是数值型，也可以为字符型。默认是数值型，表示第几个选项按钮被选中。若表单上有一选项组：☐男 ☐女，该选项组的 Value 属性值赋为 0。当其中的第二个选项按钮“女”被选中时，则该选项组的 Value 属性值为 2。

7. 答案：C

解析：当表单运行时，自动打开数据环境中的表；当表单关闭时，自动关闭数据环境中的表。

8. 答案：B

解析：如果要在 Command1 命令按钮的某个方法中访问文本框的 Value 属性值，正确的表达式是：This.Parent.Parent.Text1.Value。

9. 答案：C

解析：在表单中为表格控件指定数据源的属性是 DataSource。

10. 答案：D

解析：可以向表单数据环境设计器中添加表或视图，移出表或视图，设置表之间的联系。

11. 答案：D

解析：在运行表单时，文本框中显示的是“*”号，需要属性是 Passwordchar。因此，在 Form1 的 Init 事件中加入语句：ThisForm.Text2.Passwordchar="*"。

12. 答案：A

解析：如果文本框的 InputMask 属性值是#99999，允许在文本框中输入的是：+12345。

13. 答案：A

解析：AutoCenter 属性设置为真时，表单运行时将自动居中。

14. 答案：C

解析：执行命令 DO FORM XX NAME YY LINKED，产生表单对象引用变量 YY，在释放变量 YY 时自动关闭表单。

15. 答案：D

解析：表格控件的数据源可以是：表、视图或 SQL SELECT 语句。

16. 答案：D

解析：Click 属于命令按钮事件。

17. 答案：D

解析：运行表单时，LOAD 事件在 INIT 事件之前引发；释放表单时，DESTROY 事件在 UNLOAD 事件之前引发；Click 属于命令按钮事件。

18. 答案：A

解析：在面向对象的程序设计中，对象是程序运行的最基本实体。

19. 答案：A

解析：在 Visual FoxPro 中，表单是一个窗口界面。

20. 答案：B

解析：容器是一种能够包容其他的控件或容器的特殊控件；控件是指能以图形化的方式显示出来并能与用户进行交互的对象，控件通常被放置在一个容器里。

21. 答案：C

解析：继承表达了从一般到特殊的进化过程，是指基于现有类创建子类时，子类自动继承其父类的所有属性和方法。除了继承的属性和方法外，还可以为子类添加新的属性和方法，如果有必要，继承的方法也可以改变。可使用〈父类名〉::〈方法〉的命令格式继承父类的方法和事件。

22. 答案：B

解析：Visual FoxPro 中的基类就是系统本身内含的、并不存放在类库中的类。用户可以基于 Visual FoxPro 基类生成所需要的对象，如果有特殊要求，也可以扩展基类创建自己的类。

23. 答案：D

解析：所谓类，是指一组具有相同属性和方法的对象的集合。基于类就可以生成这类对象中的任何一个对象，称为这个类的“实例”。构成面向对象程序设计的基本单位和运行实体是对象，不能对类按其定义的属性、事件和方法进行实际的行为操作，类并不实行任何行为操作，它仅仅表明该怎样做。

24. 答案：B

解析：Parent 是指当前对象的直接容器对象。

25. 答案：A

解析：在 Visual FoxPro 中，基类的最小事件集包含的事件是：init，error，desorty。

26. 答案：B

解析：一般情况下，当运行表单时，如果要重新绘制表单或控件，将调用表单对象的 Refresh 方法。

27. 答案：A

解析：在 Visual FoxPro 中，当创建一个对象时，将产生 Init 事件。

28. 答案：B

解析：“表单”窗口包含在“表单设计器”窗口中，可以在“表单”窗口中可视化地添加和修改控件，“表单”窗口只能在表单设计器中移动。

29. 答案：B

解析：DO FORM <表单名>命令可以运行表单文件。

30. 答案：C

解析：数据环境泛指定义表单或表单集时使用的数据源。

31. 答案：B

解析：如果要为控件设置焦点，必须把 Enabled 和 Visible 属性值置为.T。

32. 答案：C

解析：DO FORM <表单名>命令可以运行表单文件。

33. 答案：A

解析：在 Visual FoxPro 中，释放表单时会引发的事件是 Unload 事件。

34. 答案：A

解析：在 Visual FoxPro 中，释放表单时会引发的事件是 Unload 事件。

35. 答案：B

解析：日期表达式中，<日期>-<日期> 是数值型，得到两个指定日期相差的天数。因此，文本框 Value 属性值的数据类型为数值型。

36. 答案：C

解析：Caption 属于表单的属性；Click 属于表单的事件；This 和 ThisForm 是保留字。

37. 答案：A

解析：AutoCenter 属性设置为真时，表单运行时将自动居中。

38. 答案：B

解析：Option1 及选项按钮和表单都设置了 Click 事件代码，而 Option2 没有设置 Click 事件代码，当表单运行时，如果用户单击 Option2，系统将执行选项按钮组的 Click 事件代码。

二、填空题

1. 答案：前

解析：运行表单时，先触发 Load 事件，然后再触发 Init 事件。

2. 答案：ColumnCount

解析：在 Visual FoxPro 中，“表格”控件的“ColumnCount”属性，可以改变表单上表格对象中当前所显示的列数。

3. 答案：Release 方法

解析：Release 方法，释放表单，将表单从内存中释放（清除）。

4. 答案：RecordSource

解析：在 Visual FoxPro 中，“表格”控件的“RecordSource”属性，用来指定表格的数据源。

5. 答案：Caption

解析：在 Visual FoxPro 中，表单的“Caption”属性，用来指定表单标题栏上的文本。

6. 答案：“真”或“.T.”

解析：为使表单运行时在主窗口中居中显示，应设置表单的 AutoCenter 属性值为.T。

7. 答案：LOAD

解析：Load 事件在表单对象建立之前引发，即运行表单时，最先引发 Load 事件。

8. 答案：CLICK

解析：在 Visual FoxPro 表单中，当用户使用鼠标单击命令按钮时，会触发命令按钮的 Click 事件。

9. 答案：1

解析：“选项组”的 Value 属性，用来指定选项按钮当前状态，即用户选择了哪个按钮。该属性的类型可以是数值型，也可以为字符型。默认是数值型，表示第几个选项按钮被选中。若表单上有一选项组：☐男 ☐女，该选项组的 Value 属性值赋为 0。当其中的第一个选项按钮“男”被选中时，则该选项组的 Value 属性值为 1。

第 11 章 习题分析与解答

一、选择题

1. 答案：A

解析：设置系统菜单的命令是：

```
SET SYSMENU ON|OFF|AUTOMATIC  
|TO [<弹出式菜单名表>]  
|TO [<条形菜单项名表>]  
|TO [DEFAULT] |SAVE|NOSAVE
```

2. 答案：C

解析：指定菜单的名称（标题），同时还可以设置菜单项的访问键，方法是在要作为访问键的字符前加“\<”。

3. 答案：B

解析：在命令窗口中键入 DO <文件名>命令运行菜单程序文件，其中菜单程序文件的扩展名.mpr 不可省略。

4. 答案：D

解析：在 Visual FoxPro 中，扩展名为.mnx 的文件是“菜单”文件。

5. 答案：A

解析：指定菜单的名称（标题），同时还可以设置菜单项的访问键，方法是在要作为访问键的字符前加“\<”。

6. 答案：D

解析：在表单的 RightClick 事件代码中添加调用快捷菜单程序的命令：DO mymenu.mpr WITH THIS。

7. 答案：B

解析：Visual FoxPro 中的快捷菜单一般是一个弹出式菜单，是由一个或一组具有上下级关系的弹出式菜单组成。

8. 答案：A

解析：在 Visual FoxPro 中，“菜单”文件的扩展名为.mnx。

9. 答案：A

解析：在 Visual FoxPro 中，系统提供了两种类型的菜单：条形菜单和弹出式菜单。

10. 答案：B

解析：Visual FoxPro 的主菜单就是一个下拉式菜单，由一个条形菜单和一组弹出式菜单组成，当选择一个条形菜单的选项时，就会激活相应的弹出式菜单。

11. 答案：D

解析：Visual FoxPro 的条形菜单的内部名字为_MSYSMENU。

12. 答案：C

解析：不带任何参数的 SET SYSMENU TO 命令将屏蔽系统菜单，使系统菜单不可用。

13. 答案：D

解析：在 Visual FoxPro 中，可以通过 SET SYSMENU 命令允许或者禁止在程序执行时访问系统菜单，同时也可以重新配置系统菜单。

14. 答案：A

解析：“编辑”菜单中，“清除”的内部名称为_MED_CLEAR。

15. 答案：B

解析：在“菜单设计器”窗口中，选择“菜单级”下拉列表框内的选项，可以返回到上级子菜单或最上层的条形菜单的定义页面。

16. 答案：D

解析：在菜单中选择任何一个选项时都执行一定的动作，这个动作可以是一条命令、一个过程或者激活另一个菜单。

二、填空题

1. 答案：SYSMENU

解析：在 Visual FoxPro 中，可以通过 SET SYSMENU 命令允许或者禁止在程序执行时访问系统菜单，同时也可以重新配置系统菜单。其中，SET SYSMENU TO DEFAULT 将系统菜单恢复为缺省设置。

2. 答案：弹出式菜单

解析：Visual FoxPro 中的快捷菜单一般是一个弹出式菜单，是由一个或一组具有上下级关系的弹出式菜单组成。

3. 答案：ACTIVATE

解析：激活条形菜单和弹出式菜单的命令都包含关键字 ACTIVATE。

4. 答案：条形菜单

解析：Visual FoxPro 中的快捷菜单一般是一个弹出式菜单，是由一个或一组具有上下级关系的弹出式菜单组成。

5. 答案：SET SYSMENU TO DEFAULT

解析：SET SYSMENU NOSAVE 将当前系统菜单缺省设置恢复为标准设置；SET SYSMENU TO DEFAULT 将系统菜单恢复为缺省设置。

6. 答案：\<

解析：指定菜单的名称（标题），同时还可以设置菜单项的访问键，方法是在要作为访问键的字符前加“\<”。

7. 答案：常规选项

解析：在顶层表单中添加下拉式菜单，首先应该在“常规选项”对话框中选中顶层表单复选框。

第 12 章 习题分析与解答

一、选择题

1. 答案：C

解析：在 Visual FoxPro 中，报表的数据源可以是数据库表、自由表、视图、查询或者临时表。

2. 答案：B

解析：报表完成后生成两个文件，一个为完整报表的.frx 文件，另一个为报表备注.ftt 文件。

3. 答案：D

解析：在 Visual FoxPro 中，报表的数据源可以是数据库表、自由表、视图、查询或者临时表，不包括文本文件。

4. 答案：B

解析：域控件和标签控件是报表控件，域控件用于打印表或视图中的字段、变量和表达式的计算结果；标签控件可用于显示说明性文字或标题文本的内容。在报表中打印当前时间应该插入一个域控件。

5. 答案：C

解析：在“报表设计器”窗口中，若要数据分组，则依据为“分组表达式”。可以利用“表达式生成器”对话框创建一个分组表达式。

6. 答案：A

解析：启动报表向导有以下四种途径：在“项目管理器”中，选“文档”选项卡，选择“报表”，单击“新建”按钮，弹“新建报表”对话框，从中选取“报表向导”，即可启动报表向导；在“文件”菜单中选择“新建”，或者单击工具栏上“新建”按钮，打开“新建”对话框，并在“文件类型”中选择“报表”，然后单击“向导”按钮；在“工具”菜单中选择“向导”子菜单，选择“报表”子菜单项；直接单击“常用”工具栏上的“报表向导”图标按钮启动报表向导。

7. 答案：D

解析：在 Visual FoxPro 中，报表的基本组成部分是数据源和布局。

8. 答案：B

解析：在 Visual FoxPro 中，在屏幕上预览报表的命令是 REPORT FORM <报表文件名> PREVIEW。

9. 答案：C

解析：组标头的作用是如果有数据分组，每组打印一次。

10. 答案：B

解析：Visual FoxPro 的主菜单就是一个下拉式菜单，由一个条形菜单和一组弹出式菜单组成，当选择一个条形菜单的选项时，就会激活相应的弹出式菜单。

11. 答案：B

解析：域控件和标签控件是报表控件，域控件用于打印表或视图中的字段、变量和表达式的计算结果；标签控件可用于显示说明性文字或标题文本的内容。在“报表设计器”中，要添加标题或其他说明文字，应使用标签控件。

二、填空题

1. 答案：MODIFY

解析：打开报表设计器的命令是 MODIFY REPORT <报表文件名>。

2. 答案：数据源、布局

解析：在 Visual FoxPro 中，报表的基本组成部分是数据源和布局。

3. 答案：组标头，组注脚

解析：如果已对报表进行了数据分组，则报表中会自动包含“组标头”或“组注脚”带区。

4. 答案：“页面设置”对话框

解析：在主菜单“文件”菜单中选择“页面设置”菜单项，弹出“页面设置”对话框，在“列”选项中，利用“列数”微调器调整栏目数。

5. 答案：变量，表达式结果

解析：域控件用于打印表或视图中的字段、变量和表达式的计算结果。

6. 答案：输入数据记录之外的信息

解析：标签控件可用于显示说明性文字或标题文本的内容。

第 13 章 习题分析与解答

一、选择题

1. 答案：C

解析：标识为“排除”的文件是项目文件一部分，项目编译成一个应用程序后，排除文件没有在应用程序中编译，但排除文件是应用程序的一部分，应用程序包括此类文件。因为这些文件没有被编译，所以用户可以更新它们。

2. 答案：D

解析：PRG 文件是 Visual FoxPro 的命令文件，连编后不能生成。

3. 答案：A

解析：项目连编以后，主文件作为只读文件处理；在 Visual FoxPro 中，任何应用程序都必须包含一个主文件，即系统的主文件是惟一的；只有程序文件、菜单、表单或查询可以设置为主文件；在项目管理器中文件名左侧带有符号排除符号 Φ 的文件，表示此文件从项目中排除，因此用户可以更新它们。

4. 答案：D

解析：项目管理器窗口中共有 6 个选项卡，其中“数据”选项卡用来集中显示该项目中的所有数据——数据库、自由表、查询和视图。

5. 答案: B

解析: 将一个项目编译成一个应用程序时, 项目包含的所有文件将组合成为一个单一的应用程序文件。

6. 答案: C

解析: 应用程序文件(.app), 需要在 Visual FoxPro 中运行; 可执行文件(.exe), 可以在 Visual FoxPro 和 Windows 环境下运行。

7. 答案: D

解析: 主程序是整个应用程序的入口点, 它的任务是设置应用程序的起始点、初始化环境、显示初始的用户界面、控制事件循环, 当退出应用程序时, 恢复原始的开发环境。

8. 答案: A

解析: 应用程序生成器的“数据”选项卡用于指定应用程序的数据源以及表单、报表的样式, 不同的表单或报表可以应用不同的样式。

9. 答案: B

二、填空题

1. 答案: 排除

解析: 在项目管理器中, 文件的排除和包含是相对的, 将一个文件编译成一个应用程序时, 该项目中标记为包含的文件将成为只读文件。

2. 答案: 主程序

解析: 主程序是整个应用程序的入口点, 且在一个项目管理器中, 只能设置一个, 并在项目管理器中以黑体显示。其首要任务是对应用程序的环境进行初始化。

3. 答案: BUILD PROJECT <项目名>

解析: 连编项目可直接在命令窗口中输入命令 BUILD PROJECT <项目名>进行连编。

4. 答案: EXE

解析: 连编应用程序时, 如果选择连编生成可执行文件, 其扩展名是.exe。

5. 答案: 排除

解析: 当对一个项目进行“连编”时, 在项目中的所有标记为“包含”的文件将组合成为一个单一的应用程序文件, 其中所有标记为“包含”的文件都变为只读文件, 不能再做任何修改。

6. 答案: 数据

解析: 如果项目不是用“应用程序向导”创建的, 由于没有事先建立完整的应用程序框架, 启动应用程序生成器只为应用程序创建元表, 即只有“数据”、“表单”和“报表”三个选项卡可用。

7. 答案: 允许

解析: 当对一个项目进行“连编”时, 如果应用程序文件中包含需要修改的文件, 就必须将这些文件标记为“排除”。

8. 答案: 应用程序框架

解析: 在 Visual FoxPro 中, 可以利用应用程序生成器向导生成一个项目和一个应用程序的基本框架。

9. 答案: .exe 可执行文件

解析: 在应用程序生成器的“常规”选项卡中, 选择程序类型时选中“顶层”, 表示生成一个.exe 可执行文件, 脱离 Visual FoxPro 环境, 在 Windows 中运行。

2010 年 3 月笔试试卷

2010 年 3 月全国计算机等级考试二级笔试试卷

Visual FoxPro 数据库程序设计

(考试时间 90 分钟, 满分 100 分)

一、选择题(每小题 2 分, 共 70 分)

下列各题 A、B、C、D 4 个选项中, 只有一个选项是正确的。请将正确选项填涂在答题卡相应位置上, 答在试卷上不得分。

- (1) 下列叙述中正确的是 ()。
 - A. 对长度为 n 的有序链表进行查找, 最坏情况下需要的比较次数为 n
 - B. 对长度为 n 的有序链表进行对分查找, 最坏情况下需要的比较次数为 $(n/2)$
 - C. 对长度为 n 的有序链表进行对分查找, 最坏情况下需要的比较次数为 $(\log_2 n)$
 - D. 对长度为 n 的有序链表进行对分查找, 最坏情况下需要的比较次数为 $(n\log_2 n)$
- (2) 算法的时间复杂度是指 ()。
 - A. 算法的执行时间
 - B. 算法所处理的数据量
 - C. 算法程序中的语句或指令条数
 - D. 算法在执行过程中所需要的基本运算次数
- (3) 软件按功能可以分为: 应用软件、系统软件和支撑软件(或工具软件)。下面属于系统软件的是 ()。
 - A. 编辑软件
 - B. 操作系统
 - C. 教务管理系统
 - D. 浏览器
- (4) 软件(程序)调试的任务是 ()。
 - A. 诊断和改正程序中的错误
 - B. 尽可能多地发现程序中的错误
 - C. 发现并改正程序中的所有错误
 - D. 确定程序中错误的性质
- (5) 数据流程图(DFD 图)是 ()。
 - A. 软件概要设计的工具
 - B. 软件详细设计的工具
 - C. 结构化方法的需求分析工具
 - D. 面向对象方法的需求分析工具
- (6) 软件生命周期可分为定义阶段, 开发阶段和维护阶段。详细设计属于 ()。
 - A. 定义阶段
 - B. 开发阶段
 - C. 维护阶段
 - D. 上述 3 个阶段
- (7) 数据库管理系统中负责数据模式定义的语言是 ()。
 - A. 数据定义语言
 - B. 数据管理语言
 - C. 数据操纵语言
 - D. 数据控制语言
- (8) 在学生管理的关系数据库中, 存取一个学生信息的数据单位是 ()。
 - A. 文件
 - B. 数据库
 - C. 字段
 - D. 记录
- (9) 数据库设计中, 用 E-R 图来描述信息结构但不涉及信息在计算机中的表示, 它属于数据库设计的 ()。

- A. 需求分析阶段
B. 逻辑设计阶段
C. 概念设计阶段
D. 物理设计阶段

(10) 有两个关系 R 和 T 如下:

R			T		
A	B	C	A	B	C
a	1	2	c	3	3
b	2	2	d	2	2
c	3	2			
d	3	2			

则由关系 R 得到关系 T 的操作是 ()。

- A. 选择
B. 投影
C. 交
D. 并
- (11) 在 Visual FoxPro 中, 编译后的程序文件的扩展名为 ()。
- A. PRG
B. EXE
C. DBC
D. FXP
- (12) 假设表文件 TEST.DBF 已经在当前工作区打开, 要修改其结构, 可以使用命令 ()。
- A. MODI STRU
B. MODI COMM TEST
C. MODI DBF
D. MODI TYPE TEST
- (13) 为当前表中所有学生的总分增加 10 分, 可以使用的命令是 ()。
- A. CHANGE 总分 WITH 总分+10
B. PEPLACE 总分 WITH 总分+10
C. CHANGE ALL 总分 WITH 总分+10
D. PEPLACE ALL 总分 WITH 总分+10
- (14) 在 Visual FoxPro 中, 下面关于属性、事件、方法叙述错误的是 ()。
- A. 属性用于描述对象的状态
B. 方法用于表示对象的行为
C. 事件代码也可以象方法一样被显式调用
D. 基于同一个类产生的两个对象的属性不能分别设置自己的属性值
- (15) 有如下赋值语句, 结果为“大家好”的表达式是 ()。
- a="你好"
b="大家"
- A. b+AT(a,1)
B. b+RIGHT(a,1)
C. b+LEFT(a,3,4)
D. b+RIGHT(a,2)
- (16) 在 Visual FoxPro 中, “表”是指 ()。
- A. 报表
B. 关系
C. 表格控件
D. 表单
- (17) 在下面的 Visual FoxPro 表达式中, 运算结果为逻辑真的是 ()。
- A. EMPTY(NULL.)
B. LIKE(' xy? ', ' xyz')
C. AT(' xy', ' abcxyz')
D. LSNUL(SPACE(0))
- (18) 以下关于视图的描述正确的是 ()。
- A. 视图和表一样包含数据
B. 视图物理上不包含数据
C. 视图定义保存在命令文件中
D. 视图定义保存在视图文件中
- (19) 以下关于关系的说法正确的是 ()。
- A. 列的次序非常重要
B. 行的次序非常重要
C. 列的次序无关紧要
D. 关键字必须指定为第一列
- (20) 报表的数据源可以是 ()。
- A. 表或视图
B. 表或查询
C. 表、查询或视图
D. 表或其他报表
- (21) 在表单中为表格控件指定数据源的属性是 ()。

- A. DataSource B. RecordSource
C. DataFrom D. RecordFrom

(22) 如果指定参照完整性的删除规则为“级联”，则当删除父表中的记录时（ ）。

- A. 系统自动备份父表中被删除记录到一个新表中
B. 若子表中有相关记录，则禁止删除父表中记录
C. 会自动删除子表中所有相关记录
D. 不作参照完整性检查，删除父表记录与子表无关

(23) 为了在报表中打印当前时间，这时应该插入一个（ ）。

- A. 表达式控件 B. 域控件
C. 标签控件 D. 文本控件

(24) 以下关于查询的描述正确的是（ ）。

- A. 不能根据自由表建立查询 B. 只能根据自由表建立查询
C. 只能根据数据库表建立查询 D. 可以根据数据库表和自由表建立查询

(25) SQL 语言的更新命令的关键词是（ ）。

- A. INSERT B. UPDATE C. CREATE D. SELECT

(26) 将当前表单从内存中释放的正确语句是（ ）。

- A. ThisForm.Close B. ThisForm.Clear C. ThisForm.Release D. ThisForm.Refresh

(27) 假设职员表已在当前工作区打开，其当前记录的“姓名”字段值为“李彤”。(C 型字段)。在命令窗口输入并执行如下命令：

姓名=姓名-"出勤"

?姓名

屏幕上会显示

- A. 李彤 B. 李彤出勤 C. 李彤出勤 D. 李彤-出勤

(28) 假设“图书”表中有 C 型字段“图书编号”，要求将图书编号以字母 A 开头的图书记录全部打上删除标记，可以使用 SQL 命令（ ）。

- A. DELETE FROM 图书 FOR 图书编号="A"
B. DELETE FROM 图书 WHERE 图书编号="A%"
C. DELETE FROM 图书 FOR 图书编号="A*"
D. DELETE FROM 图书 WHERE 图书编号 LIKE "A%"

(29) 下列程序段的输出结果是（ ）。

ACCEPT TO A

IF A=[123]

S=0

ENDIF

S=1

?S

- A. 0 B. 1 C. 123 D. 由 A 的值决定

第 (30) 到第 (35) 题基于图书表、读者表和借阅表 3 个数据库表，它们的结构如下：

图书 (图书编号,书名,第一作者,出版社)：图书编号、书名、第一作者和出版社为 C 型字段，图书编号为主关键字；

读者 (借书证号,单位,姓名,职称)：借书证号、单位、姓名、职称为 C 型字段，借书证号为主关键字；

借阅 (借书证号,图书编号,借书日期,还书日期)：借书证号和图书编号为 C 型字段，借书日期和还书日期为 D 型字段，还书日期默认值为 NULL，借书证号和图书编号共同构成主关键字。

(30) 查询第一作者为“张三”的所有书名及出版社，正确的 SQL 语句是（ ）。

- A. SELECT 书名, 出版社 FROM 图书 WHERE 第一作者=张三

- B. SELECT 书名, 出版社 FROM 图书 WHERE 第一作者="张三"
- C. SELECT 书名, 出版社 FROM 图书 WHERE "第一作者"="张三"
- D. SELECT 书名, 出版社 FROM 图书 WHERE "第一作者"="张三"
- (31) 查询尚未归还书的图书编号和借书日期, 正确的 SQL 语句是 ()。
- A. SELECT 图书编号, 借书日期 FROM 借阅 WHERE 还书日期=" "
- B. SELECT 图书编号, 借书日期 FROM 借阅 WHERE 还书日期=NULL
- C. SELECT 图书编号, 借书日期 FROM 借阅 WHERE 还书日期 IS NULL
- D. SELECT 图书编号, 借书日期 FROM 借阅 WHERE 还书日期
- (32) 查询“读者”表的所有记录并存储于临时表文件 one 中的 SQL 语句是 ()。
- A. SELECT* FROM 读者 INTO CURSOR one
- B. SELECT* FROM 读者 TO CURSOR one
- C. SELECT* FROM 读者 INTO CURSOR DBF one
- D. SELECT* FROM 读者 TO CURSOR DBF one
- (33) 查询单位名称中含“北京”字样的所有读者的借书证号和姓名, 正确的 SQL 语句是 ()。
- A. SELECT 借书证号, 姓名 FROM 读者 WHERE 单位="北京%"
- B. SELECT 借书证号, 姓名 FROM 读者 WHERE 单位="北京*"
- C. SELECT 借书证号, 姓名 FROM 读者 WHERE 单位 LIKE "北京*"
- D. SELECT 借书证号, 姓名 FROM 读者 WHERE 单位 LIKE "%北京%"
- (34) 查询 2009 年被借过书的图书编号和借书日期, 正确的 SQL 语句是 ()。
- A. SELECT 图书编号, 借书日期 FROM 借阅 WHERE 借书日期=2009
- B. SELECT 图书编号, 借书日期 FROM 借阅 WHERE year(借书日期)=2009
- C. SELECT 图书编号, 借书日期 FROM 借阅 WHERE 借书日期=year(2009)
- D. SELECT 图书编号, 借书日期 FROM 借阅 WHERE year(借书日期)=year(2009)
- (35) 查询所有“工程师”读者借阅过的图书编号, 正确的 SQL 语句是 ()。
- A. SELECT 图书编号 FROM 读者, 借阅 WHERE 职称="工程师"
- B. SELECT 图书编号 FROM 读者, 图书 WHERE 职称="工程师"
- C. SELECT 图书编号 FROM 借阅 WHERE 图书编号=(SELECT 图书编号 FROM 借阅 WHERE 职称="工程师")
- D. SELECT 图书编号 FROM 借阅 WHERE 借书证号 IN (SELECT 借书证号 FROM 读者 WHERE 职称="工程师")

二、填空题(每空 2 分, 共 30 分)

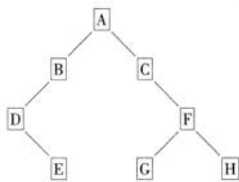
请将每一个空的正确答案写在答题卡【1】~【15】序号的横线上, 答在试卷上不得分。

注意: 以命令关键字填空的必须拼写完整。

(1) 一个队列的初始状态为空。现将元素 A, B, C, D, E, F, 5, 4, 3, 2, 1 依次入队, 然后再依次退队, 则元素退队的顺序为【1】。

(2) 设某循环队列的容量为 50, 如果头指针 front=45(指向队头元素的前一位置), 尾指针 rear=10(指向队尾元素), 则该循环队列中共有【2】个元素。

(3) 设二叉树如下:



对该二叉树进行后序遍历的结果为【3】。

- (4) 软件是【4】、数据和文档的集合。
- (5) 有一个学生选课的关系，其中学生的关系模式为：学生(学号，姓名，班级，年龄)，课程的关系模式为：课程(课号，课程名，学时)，其中两个关系模式的键分别是学号和课号。则关系模式选课可定义为：选课(学号，【5】，成绩)。
- (6) 为表建立主索引或候选索引可以保证数据的【6】完整性。
- (7) 已有查询文件 queryone. qpr，要执行该查询文件可使用命令【7】。
- (8) 在 Visual FoxPro 中，职工表 EMP 中包含有通用型字段，表中通用型字段中的数据均存储到另一个文件中，该文件名为【8】。
- (9) 在 Visual FoxPro 中，建立数据库表时，将年龄字段值限制在 18~45 岁之间的这种约束属于【9】完整性约束。
- (10) 设有学生和班级两个实体，每个学生只能属于一个班级，一个班级可以有 multiple 学生，则学生和班级实体之间的联系类型是【10】。
- (11) Visual FoxPro 数据库系统所使用的数据的逻辑结构是【11】。
- (12) 在 SQL 语言中，用于对查询结果计数的函数是【12】。
- (13) 在 SQL 的 SELECT 查询中，使用【13】关键词消除查询结果中的重复记录。
- (14) 为“学生”表的“年龄”字段增加有效性规则“年龄必须在 18~45 岁之间”的 SQL 语句是：ALTER TABLE 学生 ALTER 年龄【14】年龄<=45 AND 年龄>=18
- (15) 使用 SQL Select 语句进行分组查询时，有时要求分组满足某个条件时才查询，这时可以用【15】子句来限定分组。

2010 年 3 月笔试试卷答案及详解

一、选择题

(1) A

【解析】对长度为 n 的有序链表进行查找，最坏情况是从最小值开始查找最大值（或从最大值开始查找最小值），这个过程需要比较的次数为 n ，故选项 A 正确。对分查找只能针对随机存取的有序表进行，而有序链表只能进行顺序存取，不能进行随机存取，在有序链表上不能进行对分查找，故 B、C、D 选项都错误。

(2) D

【解析】算法的时间复杂度是指算法需要消耗的时间资源，是独立于机器的，选项 D 正确；相同的算法在不同的机器上运行时间可能不同，选项 A 错误；算法所处理的数据量与算法无关，选项 B 错误；程序的长短与程序执行时需要消耗的时间资源没有必然的关系，比如一个条件语句可以很长，但只需判断一次，而一个循环语句可能只有几行，但可以执行很多次，故选项 C 错误。

(3) B

【解析】编辑软件、教务管理软件属于应用软件，操作系统属于系统软件，浏览器属于支撑软件，故本题答案选 B。

(4) A

【解析】软件调试的任务是诊断和改正程序中的错误，本题正确答案为 A。其他 3 项描述不正确，选项 B 描述不全面，选项 C 描述太绝对，选项 D 描述不符合定义。

(5) C

【解析】数据流图是结构化分析方法中使用的工具，它以图形的方式描绘数据在系统中流动和处理的过程。软件结构图是软件概要设计的工具，而详细设计是对概要设计的每个模块进行细化，UML 是面向对象的需求分析的工具。

(6) B

【解析】软件定义阶段的基本任务是确定软件系统的工程需要，分为软件系统的可行性研究和项目需求分析。软件开发阶段包括概要设计、详细设计、实现、组装测试和确认测试 5 个阶段。维护阶段主要是软件的使用、系统维护和系统更新换代。

(7) A

【解析】数据定义语言用于定义数据库的所有特性和属性，如行布局、列定义、文件位置和存储策略等。

(8) D

【解析】在关系数据表中，行称为元组，对应存储文件中的记录，列称为属性，对应存储文件中的字段，所以本题答案为 D。

(9) C

【解析】E-R 图即实体-联系图，用来建立数据模型，在数据库系统概论中属于概念设计阶段。

(10) A

【解析】关系 T 中的两行和关系 R 中的最后两行一致，通过选择运算即可由 R 得到 T。

(11) D

【解析】编译过的程序文件的扩展名是.fxp，可执行文件的扩展名是.exe，数据库文件的扩展名是.dbc，程序文件的扩展名是.prg。

(12) A

【解析】“MODIFY COMMAND <文件名>”用来建立和编辑程序，MODIFY STRU 命令可修改表的结构，C、D 选项均为错误命令。

(13) D

【解析】修改表记录用 REPLACE 命令，其常用格式为：REPLACE FieldName1 WITH eExpression1[, FieldName2 WITH eExpression2][FOR lExpression1]，由题意可知为所有学生加 10 分的命令为：REPLACE ALL 总分 WITH 总分+10。

(14) D

【解析】属性用来表示对象的状态，方法用来描述对象的行为，事件代码既能在事件引发时执行，也可以像方法一样被显式调用。而基于同一个类产生的两个对象的属性（如同样是两个命令按钮），可以设置不同的属性（如“确定”和“取消”两个名称）。

(15) D

【解析】从 a 的右边取一个字符要用函数 RIGHT，由于一个汉字占两个字符，因此用 RIGHT(a,2)。

(16) B

【解析】一个关系就是一张二维表，每个关系有一个关系名。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”。

(17) B

【解析】EMPTY()是根据指定表达式的运算结果是否为“空”值，返回逻辑真 (.T.) 或逻辑假 (.F.)，而这里所指的“空”值与 NULL 值是两个不同的概念，A 选项的运算结果是逻辑假；LIKE()是比较两个字符串对应位置上的字符，若所有对应字符都相匹配，则返回逻辑真 (.T.)，否则返回逻辑假 (.F.)，B 选项中“?”号可以与任何单个字符相匹配；AT(<表达式 1>, <表达式 2>)中，如果<表达式 1>是<表达式 2>的子串，则返回<表达式 1>的首字符在<表达式 2>值中的位置，否则返回 0；ISNULL()是判断一个表达式的运算结果是否为 NULL 值，若是则返回逻辑真 (.T.)，否则返回逻辑假 (.F.)。故答案为 B。

(18) B

【解析】视图兼有“表”和“查询”的特点，与查询类似的地方是：可以用来从一个或多个相关联的表中提取有用信息；与表类似的地方是：可以用来更新其中的信息，并将更新结果永久保存在磁盘上。视图是根据表定义的，因此视图基于表。视图是数据库的一个特有功能，只有在包含视图的数据库打开时才能使用视图。视图物理上不包含数据。

(19) C

【解析】在一个关系中列的次序无关紧要，任意交换两列的位置也不影响数据的实际含义。同样任意交换两行的位置也不影响数据的实际含义；而对于关键字而言，无须指定列数，关键字可以任意存在于一列中。

(20) C

【解析】数据源是报表的数据来源，通常是数据库中的表或自由表，也可以是视图、查询或临时表。

(21) B

【解析】在表单的表格控件中，RecordSource 属性用于指定表格的数据源。其他几个选项不属于表格控件的属性。

(22) C

【解析】删除规则规定了当删除父表中的记录时：如果选择了“级联”，则自动删除子表中的所有相关记录；如果选择“限制”，若子表中有相关的记录，则禁止修改父表中的连接字段值；如果选择“忽略”，则不进行参照完整性检查，即删除父表的记录时与子表无关。

(23) B

【解析】在报表中插入“域控件”之后，能够更改该控件的数据类型和打印格式。其中数据类型可以是字符型、数值型或日期型。

(24) D

【解析】查询是从指定的表或视图中提取满足条件的记录，然后按照要得到的输出类型定向输出查询结果。其中指定的表可以是数据库中存在的表，也可以是数据库之外的自由表。

(25) B

【解析】在 SQL 命令中，INSERT 是插入语句的关键词，CREATE 是创建语句的关键词，SELECT 是查询语句的关键词，UPDATE 是更新语句的关键词。

(26) C

【解析】表单的释放语句是 ThisForm.Release (或 Release ThisForm)。其中 A 和 B 选项中的语句不存在，而 D 选项中的含义是刷新表单，并不是题目要求的释放表单。

(27) A

【解析】C 型字段即字符型字段，其表示方法是用半角单引号、双引号或方括号把字符串括起来，字符型变量的定界符必须成对匹配，如果某种定界符本身也是字符串的内容，则需要用另一种定界符为该字符串定界。本题输出结果为“李彤”。

(28) D

【解析】删除数据用命令：DELETE FROM TableName [WHERE Condition]，因此首先排除 A 和 C 选项。而“%”在这里表示通配符，A%表示以 A 开头。因此删除图书编号中以 A 开头的记录的命令为：DELETE FROM 图书 WHERE 图书编号 LIKE “A%。”，即 D 选项正确。

(29) B

【解析】本题考察的是条件语句的知识，条件语句格式为：IF <条件><语句序列 1>[ELSE <语句序列 2>] ENDIF，其功能是根据<条件>是否成立从两组子句中选择一组执行，当无 ELSE 子句时，可看做第二组子句不包含任何命令。若<条件>成立，则执行<语句序列 1>，然后转向 ENDIF 的下一条语句，因此本题不论 A 为多少都要执行 ENDIF 后的语句，输出结果始终为 S=1。

(30) B

【解析】查询的基本形式为“SELECT-FROM-WHERE 查询块”，多个查询块可嵌套执行。SELECT 说明要查询的数据，FROM 说明要查询的数据来自哪个或哪些表，WHERE 说明查询条件，即选择元组的条件。此处需要注意的是字段值的表示方式，当需要指定某个特定的字段值时，应当用双引号(" ")进行表示，而对于特定的字段，不需加双引号。

(31) C

【解析】SQL 支持空值，同样可以利用空值进行查询。查询空值时要使用 ISNULL，而“=NULL”是无效的，因为空值不是一个确定的值，所以不能用“=”这样的运算符进行比较。

(32) A

【解析】Visual FoxPro 中将 SQL SELECT 的查询结果存放到临时文件中，用命令：INTO CURSOR CursorName。

(33) D

【解析】只知道部分字段值时，通常采用 LIKE 进行模糊查找，查找全部符合要求的字段值。“%”和“*”都可以表示多个任意字符，而题目中要求查找名称中含有“北京”字样的字段值，所以需要在“北京”字样的前后都加上“%”或“*”。

(34) B

【解析】此处需要注意的是 year()函数的使用，“year(借书日期)=2009”表示借书日期的年份为 2009。

(35) D

【解析】此处涉及嵌套查询的知识，根据题目分析可知需要采用嵌套的 SQL 语句进行查询。“工程师”来自读者表中的职称字段，“图书编号”来自读者表，而读者表和借阅表共有字段为借书证号，所以此查询 WHERE 条件应该为借书证号，所以选 D。

二、填空题

(1) A, B, C, D, E, F, 5, 4, 3, 2, 1

【解析】队列是先进先出的数据结构，所以出队列的顺序与进队列的顺序一致。

(2) 15

【解析】计算队列中元素个数的公式为： $(\text{rear}-\text{front}+\text{MAXQSIZE})\%\text{MAXQSIZE}$ ，其中 MAXQSIZE 为队列的容量，所以该队列中的元素为 $(10-45+50)\%50=15$ 。

(3) EDBGHFCA

【解析】后序遍历二叉树的定义为：若二叉树为空，则空操作；否则，后序遍历左子树，后序遍历右子树，访问根结点。根据该规则，遍历结果应为 EDBGHFCA。

(4) 程序

【解析】本题考查软件的定义，软件是程序、数据和文档的集合。

(5) 课号

【解析】选课关系是学生和课程之间的关系，应该由学号、课号和成绩组成。

(6) 实体

【解析】实体完整性是保证表中记录唯一的特性，即在一个表中不允许有重复的记录。在 Visual FoxPro 中利用主关键字或候选关键字来保证表中的记录唯一，即保证实体唯一性。而 Visual FoxPro 中将主关键字称为主索引，将候选关键字称为候选索引。

(7) do queryone.qpr

【解析】执行程序文件有两种方式：①菜单方式。从“程序”菜单中选择“运行”命令，打开“运行”对话框，从文件列表框中选择要运行的程序文件，并单击“运行”命令按钮。②命令方式。DO <文件名>。

(8) EMP(或 EMP.fpt)

【解析】表中所有的备注型和通用型字段的内容都是统一存放在表的一个备注文件中，并以该文件命名，即 EMP 或 EMP.fpt，无论有几个，该类型字段都一样。

(9) 域

【解析】域完整性是通过指定不同的宽度说明不同范围的数值数据类型，从而可以限定字段的取值类型和取值范围。域约束规则也称作字段有效性规则，在插入或修改字段值时被激活，主要用于数据输入正确性的检验。

(10) 多对一

【解析】两个实体间的联系主要归结为 3 种类型：一对一、一对多和多对多。在 Visual FoxPro 中从未涉及多对一的关系，但此外每个学生只能属于一个班级，而一个班级可以有多名学生，这种学生和班级实体的关系确实属于多对一关系，而班级和学生实体之间的关系属于一对多关系。

(11) 关系（或二维表）

【解析】在关系数据库中，用关系（二维表）表示实体及其之间联系的模型称为关系数据模型，一个关系的逻辑结构就是一张二维表，每个关系有一个关系名。

(12) COUNT()

【解析】SQL 不仅具有一般的检索能力，而且还有计算方式的检索，用于计算检索的函数有：COUNT（计数）、SUM（求和）、AVG（计算平均值）、MAX（求最大值）、MIN（求最小值）。

(13) DISTINCT

【解析】SQL 语句中，为了避免查询到重复纪录，可使用 DISTINCT 短语，但是每一个子句中只能使用一次 DISTINCT。

(14) CHECK

【解析】由题意可知此题是修改表结构，且修改的是表的有效性规则结构。在 SQL 语句中修改表的有效性规则的命令语句通常是 ALTER 后跟 CHECK 关键词，该关键词主要是说明需要修改约有效性规则的具体内容。

(15) HAVING

【解析】HAVING 子句总是跟在 GROUP BY 子句之后，而不能单独使用。在查询中是先用 WHERE 子句限定元组，然后进行分组，最后再用 HAVING 子句来限定分组。

2010 年 9 月笔试试卷及答案

2010 年 9 月全国计算机等级考试二级笔试试卷

Visual FoxPro

(考试时间 90 分钟, 满分 100 分)

一、选择题 (每小题 2 分, 共 70 分)

下列各题中 A.、B.、C.、D. 四个选项中, 只有一个选项是正确的, 请将正确选项涂写在答题卡相应位置上, 答在试卷上不得分。

(1) 下列叙述中正确的是

- A. 线性表的链式存储结构与顺序存储结构所需要的存储空间是相同的
- B. 线性表的链式存储结构所需要的存储空间一般要多于顺序存储结构
- C. 线性表的链式存储结构所需要的存储空间一般要少于顺序存储结构
- D. 上述 3 种说法都不对

(2) 下列叙述中正确的是

- A. 在栈中, 栈中元素随栈底指针与栈顶指针的变化而动态变化
- B. 在栈中, 栈顶指针不变, 栈中元素随栈底指针的变化而动态变化
- C. 在栈中, 栈底指针不变, 栈中元素随栈顶指针的变化而动态变化
- D. 上述 3 种说法都不对

(3) 软件测试的目的是

- A. 评估软件可靠性
- B. 发现并改正程序中的错误
- C. 改正程序中的错误
- D. 发现程序中的错误

(4) 下面描述中, 不属于软件危机表现的是

- A. 软件过程不规范
- B. 软件开发生产率低
- C. 软件质量难以控制
- D. 软件成本不断提高

(5) 软件生命周期是指

- A. 软件产品从提出、实现、使用维护到停止使用退役的过程
- B. 软件从需求分析、设计、实现到测试完成的过程
- C. 软件的开发过程
- D. 软件的运行维护过程

(6) 面向对象方法中, 继承是指

- A. 一组对象所具有的相似性质
- B. 一个对象具有另一个对象的性质
- C. 各对象之间的共同性质
- D. 类之间共享属性和操作的机制

(7) 层次型、网状型和关系型数据库划分原则是

- A. 记录长度一 B. 文件的大小
C. 联系的复杂程度 D. 数据之间的联系方式

(8) 一个工作人员可以使用多台计算机，而一台计算机可被多个人使用，则实体工作人员与实体计算机之间的联系是

- A. 一对一 B. 一对多 C. 多对多 D. 多对一

(9) 数据库设计中反映用户对数据要求的模式是

- A. 内模式 B. 概念模式 C. 外模式 D. 设计模式

(10) 有三个关系 R、S 和 T 如下：

R			S		T			
A	B	C	A	D	A	B	C	D
a	1	2	e	4	e	3	1	4
b	2	1						
c	3	1						

则由关系 R 和 S 得到关系 T 的操作是

- A. 自然连接 B. 交 C. 投影 D. 并

(11) 在 Visual FoxPro 中，要想将日期型或日期时间型数据中的年份用 4 位数字显示，应当使用设置命令

- A. SET CENTURY ON B. SET CENTURY TO 4
C. SET YEAR TO 4 D. SET YAER TO yyyy

(12) 设 $A=[6*8-2]$ 、 $B=6*8-2$ 、 $C="6*8-2"$ ，属于合法表达式的是

- A. A+B B. B+C C. A-C D. C-B

(13) 假设在数据库表的表设计器中，字符型字段“性别”已被选中，正确的有效性规则设置是

- A. ="男". OR. "女" B. 性别二"男". OR. "女"
C. \$"男女" D. 性别夕男女 "

(14) 在当前打开的表中，显示“书名”以“计算机”打头的所有图书，正确的命令是：

- A. list for 书名="计算*" B. list for 书名="计算机"
C. list for 书名="计算%" D. list where 书名="计算机"

(15) 连续执行以下命令，最后一条命令的输出结果是

```
SET EXACT OFF
a="北京"
b=(a="北京交通"). '
? b
```

- A. 北京 B. 北京交通 C. .F. D. 出错

(16) 设 $x="123"$ ， $y=123$ ， $k="y"$ ，表达式 $x+\&k$ 的值是

- A. 123123 B. 246 C. 123y D. 数据类型不匹配

(17) 运算结果不是 2010 的表达式是

- A. int(2010.9) B. round(2010.1,0) C. ceiling(2010.1) D. floor(2010.9)

(18) 在建立表间一对多的永久联系时，主表的索引类型必须是

- A. 主索引或候选索引 B. 主索引、候选索引或唯一索引
C. 主索引、候选索引、唯一索引或普通索引 D. 可以不建立索引

(19) 在表设计器中设置的索引包含在

- A. 独立索引文件中 B. 唯一索引文件中
C. 结构复合索引文件中 D. 非结构复合索引文件中

(20) 假设表“学生.dbf”已在某个工作区打开，且取别名为 student。选择“学生”表所在工作区为当前工作区的命

令是

- A. SELECT 0 B. USE 学生 C. SELECT 学生 D. SELECT student

(21) 删除视图 myview 的命令是

- A. DELETE myview. B. DELETE VIEW myview
C. DROP VIEW myview D. REMOVE VIEW myview

(22) 下面关于列表框和组合框的陈述中, 正确的是

- A. 列表框可以设置成多重选择, 而组合框不能
B. 组合框可以设置成多重选择, 而列表框不能
C. 列表框和组合框都可以设置成多重选择
D. 列表框和组合框都不能设置成多重选择

(23) 在表单设计器环境中, 为表单添加一选项按钮组: ☒ 男 ☐ 女。在默认情况下, 第一个选项按钮“男”为选中状态, 此时该选项按钮组的 Value 属性值为

- A. 0 B. 1 C. "男" D. .T.

(24) 在 Visual Foxpro 中, 属于命令按钮属性的是

- A. Parent B. This C. ThisForm D. Click

(25) 在 Visual Foxpro 中, 可视类库文件的扩展名是

- A. .dbf B. .scx C. .vcx D. .dbc

(26) 为了在报表中打印当前时间, 应该在适当区域插入一个

- A. 标签控件 B. 文本框 C. 表达式 D. 域控件

(27) 在菜单设计中, 可以在定义菜单名称时为菜单项指定一个访问键。指定访问键为"x"的菜单项名称定义是

- A. 综合查询(>x) B. 综合查询(/>x)
C. 综合查询(0 众) D. 综合查询(/<x)

(28) 假设新建了一个程序文件 myProc.prg(不存在同名的.exe, .app 和.fxp 文件), 然后在命令窗口输入命令 DO myProc, 执行该程序并获得正常的结果。现在用命令 ERASE myProc.prg 删除该程序文件, 然后再次执行命令 DO myProc, 产生的结果是

- A. 出错(找不到文件) B. 与第一次执行的结果相同
C. 系统打开“运行”对话框, 要求指定文件 D. 以上都不对

(29) 以下关于视图描述错误的是:

- A. 只有在数据库中可以建立视图 B. 视图定义保存在视图文件中
C. 从用户查询的角度视图和表一样 D. 视图物理上不包括数据

(30) 关闭释放表单的方法是:

- A. 'shut B. closeForm C. release D. close

(31) ~ (35) 题使用如下数据表:

学生. DBF: 学号 (C, 8), 姓名 (C, 6), 性别 (C, 2)

选课. DBF: 学号 (C, 8), 课程号 (C, 3), 成绩 (N, 3)

(31) 从“选课”表中检索成绩大于等于 60 并且小于 90 的记录信息, 正确的 SQL 命令是

- A. SELECT * FROM 选课 WHERE 成绩 BETWEEN 60 AND 89
B. SELECT * FROM 选课 WHERE 成绩 BETWEEN 60 TO 89
C. SELECT * FROM 选课 WHERE 成绩 BETWEEN 60 AND 90
D. SELECT * FROM 选课 WHERE 成绩 BETWEEN 60 TO 90

(32) 检索还未确定成绩的学生选课信息, 正确的 SQL 命令是

- A. SELECT 学生. 学号, 姓名, 选课. 课程号 FROM 学生 JOIN 选课
WHERE 学生. 学号=选课. 学号 AND 选课. 成绩 IS NULL
B. SELECT 学生. 学号, 姓名, 选课. 课程号 FROM 学生 JOIN 选课

- WHERE 学生. 学号=选课. 学号 AND 选课. 成绩=NULL
 C. SELECT 学生. 学号, 姓名, 选课. 课程号 FROM 学生 JOIN 选课
 ON 学生. 学号=选课. 学号 WHERE 选课. 成绩 IS NULL
 D. SELECT 学生. 学号, 姓名, 选课. 课程号 FROM 学生 JOIN 选课
 ON 学生. 学号=选课. 学号 WHERE 选课. 成绩=NULL

(33) 假设所有的选课成绩都已确定。显示“101”号课程成绩中最高的 10%记录信息，正确的 SQL 命令是

- A. SELECT * TOP 10 FROM 选课 ORDER BY 成绩 WHERE 课程号="101"
 B. SELECT * PERCENT 10 FROM 选课 ORDER BY 成绩 DESC
 WHERE 课程号="101"
 C. SELECT * TOP 10 PERCENT FROM 选课 ORDER BY 成绩
 WHERE 课程号="101"
 D. SELECT * TOP 10 PERCENT FROM 选课 ORDER BY 成绩 DESC
 WHERE 课程号="101"

(34) 假设所有学生都已选课，所有的选课成绩都已确定。检索所有选课成绩都在 90 分以上（含）的学生信息，正确的 SQL 命令是（ ）

- A. SELECT * FROM 学生 WHERE 学号 IN (SELECT 学号 FROM 选课 WHERE 成绩>=90)
 B. SELECT * FROM 学生 WHERE 学号 NOT IN (SELECT 学号 FROM 选课
 C. WHERE 成绩<90)C) SELECT * FROM 学生 WHERE 学号!=ANY (SELECT 学号 FROM 选课 WHERE 成绩<90)
 D. SELECT * FROM 学生 WHERE 学号=ANY (SELECT 学号 FROM 选课 WHERE 成绩>=90)

(35) 为“选课”表增加一个“等级”字段，其类型为 C、宽度为 2，正确的 SQL 命令是（ ）

- A. ALTER TABLE 选课 ADD FIELD 等级 C (2)
 B. ALTER TABLE 选课 ALTER FIELD 等级 C(2)
 C. ALTER TABLE 选课 ADD 等级 C(2)
 D. ALTER TABLE 选课 ALTER 等级 C(2)

二、填空题（每空 2 分，共 30 分）

请将每一个空的正确答案写在答题卡【1】～【15】序号的横线上，答在试卷上不得分。

注意：以命令关键字填空的必须拼写完整。

1. 一个栈的初始状态为空。首先将元素 5, 4, 3, 2, 1 依次入栈，然后退栈一次，再将元素 A,B,C,D 依次入栈，之后将所有元素全部退栈，则所有元素退栈（包括中间退栈的元素）的顺序为【1】。

2. 在长度为 n 的线性表中，寻找最大项至少需要比较【2】次。

3. 一棵二叉树有 10 个度为 1 的结点，7 个度为 2 的结点，则该二叉树共有【3】个结点。

4. 仅由顺序、选择（分支）和重复（循环）结构构成的程序是【4】程序。

5. 数据库设计的四个阶段是：需求分析，概念设计，逻辑设计和【5】。

6. Visual Foxpro 索引文件不改变表中记录的【6】顺序。

7. 表达式 score<=100 AND score>=0 的数据类型是【7】。

8. A=10

B=20

?IIF(A>B,"A 大于 B",A 不大于 B")

执行上述程序段，显示的结果是【8】

9. 参照完整性规则包括更新规则、删除规则和【9】规则。

10. 如果文本框中只能输入数字和正负号，需要设置文本框的【10】属性。

11. 在 SQL Select 语句中使用 Group By 进行分组查询时，如果要求分组满足指定条件，则需要使用【11】子句来限定分组。

12. 预览报表 myreport 的命令是 REPORT FORM myreport 【12】。
13. 将“学生”表中学号左 4 位为“2010”的记录存储到新表 new 中的命令是 SELECT*FROM 学生 WHERE 【13】="2010" 【14】 DBF new
14. 将“学生”表中的学号字段的宽度由原来的 10 改为 12（字符型），应使用的命令是：ALTER TABLE 学生 【15】。

试题答案及解析

一. 选择题（1~35 每小题 2 分，共 70 分）

1. **【解析】**顺序存储结构是存储结构类型中的一种，该结构是把逻辑上相邻的节点存储在物理位置上相邻的存储单元中，结点之间的逻辑关系由存储单元的邻接关系来体现。由此得到的存储结构为顺序存储结构，通常顺序存储结构式借助于计算机程序设计语言的数组来描述的。顺序存储结构的主要优点是节省存储空间，因为分配给数据的存储单元全用存放结点的数据，结点之间的逻辑关系没有占用额外的存储空间。

线性表的链式存储表示的特点是用一组任意的存储单元存储线性表的数据元素（这组存储单元可以是连续的，也可以是不连续的）。因此，为了表示每个数据元素与其直接后继数据元素之间的逻辑关系，对数据元素来说，除了存储数据外，还需要存储一个或多个指针，而顺序存储不需要存储地址，所以从存储大小来看，自然是链表占空间大，不过访问灵活链表有很大优势。

【答案】B

2. **【解析】**栈（stack）在计算机科学中是限定仅在表尾进行插入或删除操作的线性表。栈是一种数据结构，是只能在某一端插入和删除的特殊线性表。它按照后进先出的原则存储数据，先进入的数据被压入栈底，最后的数据在栈顶，需要读数据的时候从栈顶开始弹出数据（最后一个数据被第一个读出来）。栈是允许在同一端进行插入和删除操作的特殊线性表。允许进行插入和删除操作的一端称为栈顶（top），另一端为栈底（bottom）；栈底固定，而栈顶浮动；栈中元素个数为零时称为空栈。插入一般称为进栈（PUSH），删除则称为退栈（POP）。栈也称为后进先出表（LIFO——Last IN First Out 表）。

因此，在栈中，栈底指针不变，栈中元素随栈顶指针的变化而动态变化。

【答案】C

3. **【解析】**软件测试是使用人工或者自动手段来运行或测试某个系统的过程，其目的在于检验它是否满足规定的要求或弄清预期结果与实际结果之间的差别。软件测试应严格执行测试计划，排除测试的随意性。另外，还必须注意：

（1）测试并不仅仅是为了找出错误，通过分析错误产生的原因和错误的发生趋势，可以帮助项目管理者发现当前软件开发过程中的缺陷，以便及时改进。

（2）这种分析也能帮助测试人员设计出有针对性的测试方法，改善测试的效率和有效性。

（3）没有发现错误的测试也是有价值的，完整的测试是评定软件质量的一种方法。

因此，软件测试的目的不仅仅是发现并改正程序中的错误，而是评估软件可靠性。

【答案】A

4. **【解析】**软件危机（software crisis），用来泛指在计算机软件的开发和维护过程中所遇到的一系列严重问题。我们可以将软件危机归结为成本、质量、生产率等问题。

早期出现的软件危机主要表现在：

（1）软件开发费用和进度失控。费用超支、进度拖延的情况屡屡发生。有时为了赶进度或压成本不得不采取一些权宜之计，这样又往往严重损害了软件产品的质量。

（2）软件的可靠性差。尽管耗费了大量的人力物力，而系统的正确性却越来越难以保证，出错率大大增加，由于软件错误而造成的损失十分惊人。

（3）生产出来的软件难以维护。很多程序缺乏相应的文档资料，程序中的错误难以定位，难以改正，有时改正了已有的错误又引入新的错误。随着软件的社会拥有量越来越大，维护占用了大量人力、物力和财力。进入 80 年代以来，尽

管软件工程研究与实践取得了可喜的成就，软件技术水平有了长足的进展，但是软件生产水平依然远远落后于硬件生产水平的发展速度。

软件危机不仅没有消失，还有加剧之势。主要表现在：

(1) 软件成本在计算机系统总成本中所占的比例居高不下，且逐年上升。由于微电子学技术的进步和硬件生产自动化程度不断提高，硬件成本逐年下降，性能和产量迅速提高。然而软件开发需要大量人力，软件成本随着软件规模和数量的剧增而持续上升。从美、日两国的统计数字表明，1985 年度软件成本大约占总成本的 90%。

(2) 软件开发生产率提高的速度远远跟不上计算机应用迅速普及深入的需要，软件产品供不应求的状况使得人类不能充分利用现代计算机硬件所能提供的巨大潜力。

由上述知识点可知，软件危机的表现不包括软件过程不规范。

【答案】A

5. 【解析】同任何事物一样，一个软件产品或软件系统也要经历孕育、诞生、成长、成熟、衰亡等阶段，一般称为软件生存周期（软件生命周期）。

把整个软件生存周期划分为若干阶段，使得每个阶段有明确的任务，使规模大，结构复杂和管理复杂的软件开发变的容易控制和管理。通常，软件生存周期包括可行性分析与开发项计划、需求分析、设计（概要设计和详细设计）、编码、测试、维护等活动，可以将这些活动以适当的方式分配到不同的阶段去完成。

软件生命周期（SDLC，软件生存周期）是软件的产生直到报废的生命周期，周期内有问题定义、可行性分析、总体描述、系统设计、编码、调试和测试、验收与运行、维护升级到废弃等阶段，这种按时间分程的思想方法是软件工程中的一种思想原则，即按部就班、逐步推进，每个阶段都要有定义、工作、审查、形成文档以供交流或备查，以提高软件的质量。但随着新的面向对象的设计方法和技术的成熟，软件生命周期设计方法的指导意义正在逐步减少。

因此软件生命周期是指软件产品从提出、实现、使用维护到停止使用退役的过程。

【答案】A

6. 【解析】面向对象方法（Object-Oriented Method）是一种把面向对象的思想应用于软件开发过程中，指导开发活动的系统方法，简称 OO（Object-Oriented）方法，是建立在“对象”概念基础上的方法学。对象是由数据和容许的操作组成的封装体，与客观实体有直接对应关系，一个对象类定义了具有相似性质的一组对象。而继承性是对具有层次关系的类的属性和操作进行共享的一种方式。所谓面向对象就是基于对象概念，以对象为中心，以类和继承为构造机制，来认识、理解、刻画客观世界和设计、构建相应的软件系统。

因此面向对象方法中，继承是指类之间共享属性和操作的机制。

【答案】D

7. 【解析】数据库是依照某种数据模型组织起来并存放二级存储器中的数据集合。数据库通常分为层次式数据库、网络式数据库和关系式数据库 3 种。而不同的数据库是按不同的数据结构来联系和组织的。

数据结构是目标类型的集合。目标类型是数据库的组成成分，一般可分为两类：数据类型、数据类型之间的联系。层次模型、网状模型和关系模型是 3 种重要的数据模型。这 3 种模型是按其数据结构而命名的。

(1) 层次结构模型

按照层次模型建立的数据库系统称为层次模型数据库系统。IMS（Information Management System）是其典型代表。层次结构采用关键字来访问其中每一层次的每一部分。层次模型的特点是将数据组织成一对多关系的结构。

其优点是：

- 存取方便且速度快。
- 结构清晰，容易理解。
- 数据修改和数据库扩展容易实现。
- 检索关键属性十分方便。

其缺陷是：

- 结构呆板，缺乏灵活性。
- 同一属性数据要存储多次，数据冗余大（如公共边）。
- 不适合于拓扑空间数据的组织。

(2) 网状结构模型

按照网状数据结构建立的数据库系统称为网状数据库系统，其典型代表是 DBTG (Data Base Task Group)。用数学方法可将网状数据结构转化为层次数据结构。网状模型用连接指令或指针来确定数据间的显式连接关系，是具有多对多类型的数据组织方式。

其优点是：

- 能明确而方便地表示数据间的复杂关系。
- 数据冗余小。

其缺陷是：

- 网状结构的复杂，增加了用户查询和定位的困难。
- 需要存储数据间联系的指针，使得数据量增大。
- 数据的修改不方便（指针必须修改）。

(3) 关系结构模型

关系式数据结构把一些复杂的数据结构归结为简单的二元关系（即二维表格形式）。例如某单位的职工关系就是一个二元关系。关系数据库模型是以记录组或数据表的形式组织数据，以便于利用各种地理实体与属性之间的关系进行存储和变换，不分层也无指针，是建立空间数据和属性数据之间关系的一种非常有效的数据组织方法。

其优点是：

- 结构特别灵活，满足所有布尔逻辑运算和数学运算规则形成的查询要求。
- 能搜索、组合和比较不同类型的数据。
- 增加和删除数据非常方便。

其缺陷是：

- 数据库大时，查找满足特定关系的数据费时。
- 对空间关系无法满足。

因此层次型、网状型和关系型数据库划分原则是数据之间的联系方式。

【答案】D

8. 【解析】实体就是客观存在并可相互区别的事物，实体可以是具体的人、事、物，也可以是抽象的概念或联系。

实体之间的联系实际上反映了实体之间的语义关系。联系也是实体，因此它也可以有属性。实体间联系情况比较复杂，就其联系方式来说，我们可把它分为以下 3 种：

- (1) 一对一联系（简记为 1:1）
- (2) 一对多联系（简记为 1:n）
- (3) 多对多联系（简记为 m:n）

本题中的工作人员和计算机两个实体中，一个工作人员可以使用多台计算机，而一台计算机可被多个人使用，则实体工作人员、与实体计算机之间的联系类型是“多对多”。

【答案】C

9. 【解析】数据库设计 (Database Design) 是指根据用户的需求，在某一具体的数据库管理系统上，设计数据库的结构和建立数据库的过程。就是规划和结构化数据库中的数据对象及这些数据对象之间关系的过程。

(1) 外模式是用户与数据库系统的接口，是用户用到的那部分数据的描述。它由若干个外部记录类型组成。

外模式又称子模式，对应于用户级。它是某个或某几个用户所看到的数据库的数据视图，是与某一应用有关的数据的逻辑表示。外模式是从模式导出的一个子集，包含模式中允许特定用户使用的那部分数据。用户可以通过外模式描述语言来描述、定义对应于用户的数据记录（外模式），也可以利用数据操纵语言 (DML) 对这些数据记录进行。外模式反映了数据库的用户观。

(2) 概念模式 (Conceptual Schema) 是数据库中全部数据的整体逻辑结构的描述。它由若干个概念记录类型组成，还包含记录间联系、数据的完整性、安全性等要求。

描述概念模式的数据定义语言称为“模式 DDL”。

(3) 内模式（也称存储模式）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；记录的存储方

式（顺序存储，按照 B 树结构存储，按 hash 方法存储）；索引的组织方式；数据是否压缩存储 数据是否加密；数据存储记录结构的规定。

一个数据库只有一个内模式，能够给出数据库物理存储结构与物理存取方法。

内模式对应于物理级，它是数据库中全体数据的内部表示或底层描述，是数据库最低一级的逻辑描述，它描述了数据在存储介质上的存储方式的物理结构，对应着实际存储在外存储介质上的数据库。内模式由内模式描述语言来描述、定义，它是数据库的存储观。

因此数据库设计中反映用户对数据要求的模式是外模式。

【答案】C

10. 【解析】关系的基本运算有两类：一类是传统的集合运算（并、差、交等），另一类是专门的关系运算（选择、投影、连接等），有些查询需要几个基本运算的组合，要经过若干步骤才能完成。

（1）传统的集合运算

- 并（UNION）：设有两个关系 R 和 S，它们具有相同的结构。R 和 S 的并是由属于 R 或属于 S 的元组组成的集合，运算符为 $\cup$ 。记为 $T=R \cup S$ 。
- 差（DIFFERENCE）：R 和 S 的差是由属于 R 但不属于 S 的元组组成的集合，运算符为 $-$ 。记为 $T=R-S$ 。
- 交（INTERSECTION）：R 和 S 的交是由既属于 R 又属于 S 的元组组成的集合，运算符为 $\cap$ 。记为 $T=R \cap S$ 。 $R \cap S = R - (R - S)$ 。

（2）专门的关系运算

- 选择运算：从关系找出满足给定条件的那些元组称为选择。其中的条件是以逻辑表达式给出的，值为真的元组将被选取。这种运算是从水平方向抽取元组。在 Visual FoxPro 中的短语 FOR<条件>和 WHILE<条件>均相当于选择运算。
- 投影运算：从关系模式中挑选若干属性组成新的关系称为投影。这是从列的角度进行的运算，相当于对关系进行垂直分解。在 FOXPRO 中短语 FIELDS<字段 1, 字段 2, ...>相当于投影运算。
- 联接运算：选择和投影运算都是属于一目运算，它们的操作对象只是一个关系。连接运算是二目运算，需要两个关系作为操作对象。连接是将两个关系模式通过公共的属性名拼接成一个更宽的关系模式，生成的新关系中包含满足联接条件的元组。运算过程是通过连接条件来控制的，连接条件中将出现两个关系中的公共属性名，或者具有相同语义、可比的属性。连接是对关系的结合。专门的关系运算有，选择、投影、连接等。

（3）自然连接

自然连接（Natural join）是一种特殊的等值连接，它要求两个关系中进行比较的分量必须是相同的属性组，并且在结果中把重复的属性列去掉。而等值连接并不去掉重复的属性列。在连接条件中使用等于（=）运算符比较被连接列的列值，但它使用选择列表指出查询结果集合中所包括的列，并删除连接表中的重复列。

【答案】A

11. 【解析】命令格式：SET CENTURY ON/OFF/TO [世纪值] ROLLOVER [年份参照值]。

命令功能：用于设置显示日期型数据时是否显示世纪。“To”选项确定用 2 位数字表示年份所处的世纪，具体地说，如果该日期的 2 位数字年份大于等于[年份参照值]，则它所处的世纪即为[世纪值]；否则为[世纪值]+1。SET CENTURY 命令用于设置年份的位数，当取 ON 时为 4 位年份，取 OFF 时为 2 位年份。

【答案】A

12. 【解析】由题可知， $A=[6*8-2]$ 、 $B=6*8-2$ 、 $C="6*8-2"$ ，其中变量 A 和 C 中存放的是字符型数据，而变量 B 中存放的是数值型数据，在 Visual FoxPro 中，字符型数据与数值型数据是不能进行运算的，因此 A、C 均不能与 B 进行运算，但是 A 与 C 可以进行运算。

字符表达式由字符串运算符将字符型数据连接起来形成，其运算结果仍然是字符型数据。字符串运算符有以下两个，它们的优先级相同。

- +：前后两个字符串首尾连接形成一个新的字符串。
- -：连接前后两个字符串，并将前字符串的尾部空格移到合并后的新字符串尾部。

【答案】C

13. 【解析】本题考查的是考生对于字段有效性规则的书写，设置字段“性别”的有效性规则正确的写法有很多，常用的是两种：

- 性别="男" .or. 性别="女"
- 性别\$"男女"

【答案】D

14. 【解析】List 命令中[FOR 条件]用于在指定的范围内进行选择操作，仅显示满足条件的记录。[WHELE 条件]是从当前记录开始显示满足条件的记录，一旦遇到不满足条件的记录即终止命令的执行，即使后面有满足条件的记录也不再显示。没有用 where 表示条件，因此答案 D 排除。

在 Visual Foxpro 中用单等号运算符“=”比较两个字符串时，运算结果与 SET EXACT ON|OFF 设置有关，该命令是设置精确匹配与否的开关。

(1) 系统默认 OFF 状态。当处于 OFF 状态时，只要右边的字符串与左边的字符串的前部分内容相匹配，即可得到逻辑真.T.的结果。换言之，字符串的比较是以“=”右面的字符串为目标，右字符串结束即终止比较。

(2) 当处于 ON 状态时，将会比较到两个字符串全部结束，系统现在较短字符串的尾部加上若干个空格，使两字符串长度相等，然后再进行比较。

【答案】B

15. 【解析】本题考查的是字符串的比较，理论知识同上。

SET EXACT OFF 确定匹配与否的开关是 OFF 状态
a="北京" 变量 a 的值为“北京”
b=(a="北京交通") a="北京交通"的结果为.F.，因此变量 b 的值是.F.
?b

【答案】C

16. 【解析】宏替换函数&<字符型变量>[.]: 替换出字符型变量的内容，即&的值是变量中的字符串。如果该函数与后续的字符无明确分界，则要用“.”作为函数结束标识。宏替换可以嵌套使用。

本题中，x="123"，y=123，k="y"，则&k=y=123，因此x+&k的结果是数据类型不匹配。

【答案】D

17. 【解析】INT(<数值表达式>): 返回指定数值表达式的整数部分。

CEILING(<数值表达式>): 返回大于或者等于指定数值表达式的最小整数。

FLLOR(<数值表达式>): 返回小于或者等于指定数值表达式的最大整数。

ROUND(<数值表达式 1>, <数值表达式 2>): 返回指定表达式在指定位置四舍五入后的结果。<数值表达式 2>指明四舍五入的位置。若<数值表达式 2>大于或者等于 0，那么它表示的是要保留的小数位数；若<数值表达式 2>小于 0，那么它表示的是整数部分的舍入位数。

本题中只有 C 答案中的 ceiling(2010.1)=2011，其它选项结果均为 2010。

【答案】C

18. 【解析】在 Visual FoxPro 中有 4 种类型的索引。

(1) 主索引，主索引只能在数据库表中建立，不能在自由表中建立，主索引是在指关键字段中不允许重复的索引。如果在任何含有重复数据的字段中指定主索引，VF 就会提示出错。主索引经常用于建立表与表之间的关联。一个数据库表只能建立一个主索引。

(2) 候选索引，同主索引类似，要求关键字的值是唯一的不能重复。一个表中可以建立多个候选索引（这就是候选索引和主索引的区别）。候选索引在需要的时候可以变成主索引。

(3) 普通索引，这种索引用的比较多，它不要求关键字中的数据具有唯一性。也就是说允许字段中有重复的值。另外，在一个表中可以有加入多个普通索引。

(4) 唯一索引，唯一索引对于有重复记录只选择首次出现的一个作为关键字。保留这一索引类型的目的是为了与早期 FoxPro 版本的兼容性，微软公司不推荐用户使用唯一索引。

在 Visual FoxPro 中建立永久联系时，主表的索引类型必须是主索引或候选索引。

【答案】A

19.【解析】Visual FoxPro 建立索引文件可以根据特定顺序处理表的记录，并且加快对表的查询操作。索引是由指针构成的文件，这些指针逻辑上按照索引关键字值进行的排序。索引文件和表文件分别存储，并且不改变表文件中记录的物理顺序。实际上索引文件是由指向表文件记录指针构成的文件。

在 Visual FoxPro 中按组织方式分，可分为：

- (1) 单索引：扩展名.idx 索引文件（为了兼容以前版本），一个索引文件只能包含一个索引。
- (2) 结构复合索引：扩展名.cdx 索引文件，一个索引文件可包含多个索引。
- (3) 非结构化复合索引：扩展名.cdx 索引文件，须用 set index to 索引文件列表打开索引文件。

结构复合索引文件是 Visual FoxPro 数据库中最常用的，也是最重要的一种索引文件，它与表文件名同名，它会随着表的打开而自动打开，在表设计器中设置的索引包含在其中。

【答案】C

20.【解析】在 Visual FoxPro 中，工作区是用来保存表及其相关信息的一片内存空间。平时讲的打开表实际上就是将它从磁盘调入到内存的某一个工作区。在每个工作区中只能打开一个表文件，但可以同时打开与表相关的其他文件，如索引文件、查询文件等。若在一个工作区中打开一个新的表，则该工作区中原来的表将被关闭。有了工作区的概念，就可以同时打开多个表，但在任何一个时刻用户只能选中一个工作区进行操作。当前正在操作的工作区称为当前工作区。不同工作区可以用其编号或别名来加以区分。

Visual FoxPro 提供了 32767 个工作区，系统以 1~32767 作为各工作区的编号。

工作区的别名有两种，一种是系统定义的别名：1~10 号工作区的别名分别为字母 A~J。另一种是用户定义的别名，用命令“USE(表文件名)ALIAS(别名)”指定。由于一个工作区只能打开一个表，因此可以把表的别名作为工作区的别名。若未用 ALIAS 子句对表指定别名，则以表的主名作为别名。

指定工作区的命令是：SELECT 工作区名|表别名，由于本题将学生.dbf 指定别名为 student，因此应使用命令：SELECT student 打开该工作区。

【答案】D

21.【解析】删除视图的命令格式：DROP VIEW<视图名>。

【答案】C

22.【解析】列表框提供一组条目（数据项），用户可以从选择一个或多个条目。在一般情况下，列表框显示其中的若干条目，用户可以通过滚动条浏览其他条目。

组合框与列表框类似。这两个控件具有相同的属性（组合框没有 MultiSelect 属性）。

特别注意：

- 组合框只有一个条目是可见的。
- 组合框不提供多重选择的功能。
- 组合框有两种形式：下拉组合框和下拉列表框，通过设置 Style 属性可选择想要的形式。

【答案】C

23.【解析】选项组是包含选项按钮的容器。通常，选项按钮允许用户指定对话框中几个操作选项中的一个，而不是输入数据。

选项组的 Value 属性表明用户选定了哪一个按钮。本题中“男”按钮为第一个按钮，因此 Value 应为 1。

【答案】B

24.【解析】命令按钮的常用属性如下：

- Caption 属性：设置命令按钮标题。若在该属性的某字符前加“\<”，该字符就成为热键。
- Enabled 属性：用于确定命令按钮对象是否响应用户事件。
- Visible 属性：用于确定命令按钮对象是可见还是隐藏。
- Default 属性：将命令按钮设为默认命令按钮，当所有命令按钮都未获得焦点时，用户按回车键，则执行默认命令按钮的 Click 事件；表单中只能有一个默认命令按钮。
- Cancel 属性：属性值为.T.时，用户按【Esc】键执行该命令按钮的 Click 事件。

- Picture 属性：设置命令按钮上显示的图形。
- ToolTipText 属性：设置命令按钮的提示文本；设置该属性值之前，必须设置表单的 ShowTip 属性值为.T。
- Click 事件：命令按钮的单击鼠标左键事件。
- RightClick 事件：命令按钮的单击鼠标右键事件。

【答案】D

25. 【解析】在 Visual Foxpro 中，.dbf 是表文件扩展名；

.scx 是表单文件扩展名。

.vcx 是可视类库文件扩展名。

.dbc 是数据库文件扩展名。

【答案】C

26. 【解析】本题考查的是报表的控件。

(1) 表达式控件、文本控件：没有。

(2) 域控件：用于打印报表数据源中的字段、变量和表达式的计算结果。

(3) 标签控件：用于打印报表中说明性的文字，如标题等。

而当前的时间是不断变化的，属于变量。

【答案】D

27. 【解析】本题考查菜单快捷键的设计方法。添加访问键方法是在文件名后的括号内添加“/<”两个字符，再加上访问键的字母。

【答案】C

28. 【解析】在 Visual FoxPro 中 ERASE 命令可以用来删除磁盘上的指定文件，但是必须规定该文件的路径。

因此本题中执行 ERASE myProc.prg 命令不能将该文件删除，当再次执行命令 DO myProc，产生的结果是与第一次执行的结果相同。

【答案】B

29. 【解析】视图是存在于数据库中的一个虚表，不以独立的文件形式保存；视图中的数据是可以更改的，它不仅具有查询的功能，且可以把更新结果反映到源数据表中；当视图打开时，其基表自动打开，当视图关闭时，其基表并不随之自动关闭；视图的数据源可以是自由表、数据库表或另一个视图；视图是可以删除的。

因此答案 B 视图定义保存在视图文件中是错误的。

【答案】B

30. 【解析】关闭释放表单的方法是：release。

【答案】B

31. 【解析】在 SQL 的查询语句中，WHERE 子句可包括各种条件运算符：

(1) 比较运算符(大小比较)：>、>=、=、<、<=、<>、!>、!<

(2) 范围运算符(表达式值是否在指定的范围)：BETWEEN...AND...；NOT BETWEEN...AND...。

BETWEEN ...AND...运算符指定了要搜索的一个闭区间。

因此，本题答案是 A，如果本题要求检索成绩大于等于 60 并且小于或等于 90 的记录信息，则应选择 C。

【答案】A

32. 【解析】在 SQL 超链接查询中，利用 JOIN 短语进行多表间的联系，其两表之间的联接字段用 ON 短语指明。

本题条件“还未确定成绩的学生选课信息”的含义是有记录，但是成绩的值是 NULL 值，在定义中说，NULL 值不与任何值相等。因此不能使用=null 做判断。在 SQL 语句中，唯一可以使用=null 的就是 UPDATE 语句中的赋值语句，其他地方都不可以使用。IS NULL 短语用于判断是 NULL 值。

【答案】C

33. 【解析】SELECT 语句的完整语法为：

SELECT[ALL|DISTINCT|DISTINCTROW|TOP]

{*[table.*|[table.]field1[AS alias1],[table.]field2[AS alias2],[...]]}

FROM tableexpression[...][IN externaldatabase]
[WHERE...]
[GROUP BY...]
[HAVING...]
[ORDER BY...]
[WITH OWNERACCESS OPTION]

ALL、DISTINCT、DISTINCTROW、TOP 谓词的用法如下：

- (1) ALL 返回满足 SQL 语句条件的所有记录。如果没有指明这个谓词，默认为 ALL。
- (2) DISTINCT 如果有多个记录的选择字段的数据相同，只返回一个。
- (3) DISTINCTROW 如果有重复的记录，只返回一个。
- (4) TOP 显示查询头尾若干记录。也可返回记录的百分比，这是要用 TOP N PERCENT 子句（其中 N 表示百分比），该短语必须和 ORDER BY 短语一起使用。

由于本题要求是成绩最高的 10%，因此应该按照成绩的降序排序。

【答案】D

34. 【解析】在 SQL 中使用嵌套查询，当子查询返回一组值时，可使用 IN 和 NOT IN 运算符。

本题考点是多表查询。在这里我们注意到两点：

- (1) 所有选课成绩都在 90 分以上（含）的学生信息。
- (2) 内层查询的结果不是一个值而是多个值，所以不能使用“=”，而应该使用“IN”或“NOT IN”。

【答案】B

35. 【解析】在 SQL 中修改或增加一个字段的命令格式是：

ALTER TABLE TableName1 ADD | ALTER [COLUMN] FieldName1
FieldType [(nFieldWidth [, nPrecision])] [NULL | NOT NULL]
[CHECK IExpression1 [ERROR cMessageText1]]
[AUTOINC [NEXTVALUE NextValue [STEP StepValue]]] [DEFAULT eExpression1]
[PRIMARY KEY | UNIQUE [COLLATE cCollateSequence]]
[REFERENCES TableName2 [TAG TagName1]] [NOCPTRANS] [NOVALIDATE]

参数说明：

- ALTER TABLE TableName1：要修改结构的表。TableName1 参数指定了要修改的表的名称。
- ADD | ALTER [COLUMN] FieldName1：指定要添加或修改的字段名。单个表中最多包含 255 个字段。如果一个或多个字段允许空值（NULL），最高限制减少一个，变为 254 个字段。使用 ALTER COLUMN 重排默认排序。执行 ALTER COLUMN 之后需要为想要的排序顺序重新索引。也可以使用 ALTER TABLE 命令中的 COLLATE 子句确定适当的（主关键字或外部关键字）。

【答案】C

二. 填空题（每小题 2 分，共 30 分）

1. 【解析】栈（stack）在计算机科学中是限定仅在表尾进行插入或删除操作的线性表。栈是一种数据结构，是只能在某一端插入和删除的特殊线性表。它按照后进先出的原则存储数据，先进入的数据被压入栈底，最后的数据在栈顶，需要读数据的时候从栈顶开始弹出数据（最后一个数据被第一个读出来）。栈是允许在同一端进行插入和删除操作的特殊线性表。允许进行插入和删除操作的一端称为栈顶（top），另一端为栈底（bottom）；栈底固定，而栈顶浮动；栈中元素个数为零时称为空栈。插入一般称为进栈（PUSH），删除则称为退栈（POP）。栈也称为后进先出表（LIFO--Last IN First Out 表）。

本题首先将元素 5，4，3，2，1 依次入栈，然后退栈一次，再将元素 A，B，C，D 依次入栈，之后将所有元素全部退栈，则根据栈先进后出的原理所有元素退栈（包括中间退栈的元素）的顺序为 1，D，C，B，A，2，3，4，5。

【答案】1，D，C，B，A，2，3，4，5。

2. 【解析】在长度为 n 的线性表中，寻找任意项至少需要比较 1 次。

【答案】1。

3. 【解析】二叉树是每个结点最多有两个子树的有序树。通常子树的根被称作“左子树”(left subtree)和“右子树”(right subtree)。二叉树常被用作二叉查找树和二叉堆。二叉树的每个结点至多只有二棵子树(不存在度大于 2 的结点)，二叉树的子树有左右之分，次序不能颠倒。二叉树的第 i 层至多有 2^{i-1} 个结点；深度为 k 的二叉树至多有 $2^k - 1$ 个结点；对任何一棵二叉树 T ，如果其终端结点数(即叶子结点数)为 n_0 ，度为 2 的结点数为 n_2 ，则 $n_0 = n_2 + 1$ 。

由此可知，一棵二叉树有 10 个度为 1 的结点，7 个度为 2 的结点，则该二叉树共有度为 0 的结点个数 $n_0 = 7 + 1 = 8$ + 度为 1 的结点个数 10 + 度为 2 的结点个数 7 即总结点个数 $= 8 + 10 + 7 = 25$ 。

【答案】25。

4. 【解析】符合结构化原则的 3 种基本控制结构是：选择结构、循环结构和顺序结构。因此仅由顺序、选择(分支)和重复(循环)结构构成的程序是结构化程序。

【答案】结构化。

5. 【解析】数据库设计(Database Design)是指根据用户的需求，在某一具体的数据库管理系统上，设计数据库的结构和建立数据库的过程。

其设计通常有 4 个阶段：(1)需求分析。调查和分析用户的业务活动和数据的使用情况，弄清所用数据的种类、范围、数量，以及它们在业务活动中交流的情况，确定用户对数据库系统的使用要求和各种约束条件等，形成用户需求规约。

(2)概念设计。对用户要求描述的现实世界(可能是一个工厂、一个商场或者一个学校等)，通过对其中住处的分类、聚集和概括，建立抽象的概念数据模型。这个概念模型应反映现实世界各部门的信息结构、信息流动情况、信息间的互相制约关系以及各部门对信息储存、查询和加工的要求等。所建立的模型应避免数据库在计算机上的具体实现细节，用一种抽象的形式表示出来。以扩充的实体-联系模型(E-R 模型)方法为例，第一步先明确现实世界各部门所含的各种实体及其属性、实体间的联系以及对信息的制约条件等，从而给出各部门内所用信息的局部描述(在数据库中称为用户的局部视图)。第二步再将前面得到的多个用户的局部视图集成为一个全局视图，即用户要描述的现实世界的概念数据模型。

(3)逻辑设计。主要工作是将现实世界的概念数据模型设计成数据库的一种逻辑模式，即适应于某种特定数据库管理系统所支持的逻辑数据模式。与此同时，可能还需为各种数据处理应用领域产生相应的逻辑子模式。这一步设计的结果就是所谓“逻辑数据库”。

(4)物理设计。根据特定数据库管理系统所提供的多种存储结构和存取方法等依赖于具体计算机结构的各项物理设计措施，对具体的应用任务选定最合适的物理存储结构(包括文件类型、索引结构和数据的存放次序与位逻辑等)、存取方法和存取路径等。这一步设计的结果就是所谓的“物理数据库”。

【答案】物理设计。

6. 【解析】Visual FoxPro 建立索引文件可以根据特定顺序处理表的记录，并且加快对表的查询操作。索引是由指针构成的文件，这些指针逻辑上按照索引关键字值进行的排序。索引文件和表文件分别存储，并且不改变表文件中记录的物理顺序。实际上索引文件是由指向表文件记录指针构成的文件。

因此 Visual Foxpro 索引文件不改变表中记录的物理顺序。

【答案】物理。

7. 【解析】逻辑表达式由逻辑运算符将逻辑型数据连接起来而形成，其运算结果仍然是逻辑型数据。逻辑运算符有 3 个：.NOT. 或者 ! (逻辑非)、.AND. (逻辑与) 及 .OR. (逻辑或)。也可以省略两端的点，写成 NOT、AND、OR。其优先级顺序依次为 NOT、AND、OR。

本题中的表达式 $\text{score} \leq 100 \text{ AND } \text{score} \geq 0$ 是逻辑型表达式，其结果是逻辑性数据。

【答案】逻辑型。

8. 【解析】IIF(<逻辑表达式>, <表达式 1>, <表达式 2>): 测试<逻辑表达式>的值，如果为真函数则返回<表达式 1>的值；若为逻辑假，则函数返回<表达式 2>的值。

本题中 $A=10, B=20, ?\text{IIF}(A>B, "A \text{ 大于 } B", "A \text{ 不大于 } B")$ ，执行上述程序段，显示的结果是 A 不大于 B。

【答案】A 不大于 B。

9. 【解析】Visual FoxPro 的数据库完整性包括了 3 个完整性。

(1) 实体完整性，指得是建立主索引和候选索引保证记录的唯一性。

(2) 域完整性，选定字段的取值类型和取值范围保证域完整性，在表设计器的字段选项卡中定义字段有效性规则的项目有规则、信息和默认值 3 项。

(3) 参照完整性，当更新、删除、插入一个表中的数据时，通过参照引用相互关联的另一个表中的数据，来检查对表的数据操作是否正确。建立关联的两个表，可以利用“参照完整性生成器”分别对更新规则、删除规则和插入规则进行设置。

【答案】插入。

10. 【解析】文本框的常用属性有：

- Value：指定文本框对象的值，并在框中显示。如果遇到长数据能自动换行，若按下回车键，则终止数据输入。
- ControlSource：设置文本框对象数据源。文本框控件的数据源可以是字段和变量两种，若是字段，则必须是来自数据环境中的表。当使用字段映像功能创建对象时，系统将自动设置该对象的数据源属性。
- Format：指定 Value 属性输入和显示时的格式。
- PasswordChar：指定文本框控件内显示用户输入的是字符还是占位符；指定用作占位符的字符。
- InputMask：用于确定控件中如何输入和显示数据。这与 Format 属性不同，Format 属性规定了整个文本框的格式，而 InputMask 属性规定了同一位置字符的格式。

因此如果文本框中只能输入数字和正负号，需要设置文本框的 InputMask 属性。

【答案】InputMask。

11. 【解析】在 SQL 的语法里，GROUP BY 和 HAVING 子句用来对数据进行汇总。GROUP BY 子句指明了按照哪几个字段来分组，而将记录分组后，用 HAVING 子句过滤这些记录。

【答案】Having。

12. 【解析】在 Visual FoxPro 中预览报表的命令是 REPORT FORM 报表名 Preview。

【答案】Preview。

13. 【解析】SELECT 语句的完整语法为：

```
SELECT[ALL|DISTINCT|DISTINCTROW|TOP]
{*[table.*|[table.]field1[AS alias1],[table.]field2[AS alias2],[...]]}
FROM tableexpression[,...][IN externaldatabase]
[WHERE...]
[GROUP BY...]
[HAVING...]
[ORDER BY...]
[WITH OWNERACCESS OPTION]
```

其中 WHERE 子句指定查询条件。本题中的条件是学号左 4 位为"2010"的记录，可以使用 LEFT(<字符表达式>,<长度>)：返回字符表达式左侧起提取指定长度的字符串。SUBSTR(<字符表达式>,<起始位置>,<长度>)：返回字符表达式从左侧起始位置开始的指定长度的字符串，如果没有长度参数，则提取字符表达式从左侧起始位置后所有字符作为函数值返回。

在 SQL 语句中 INTO TABLE 或者 INTO DBF 均表示将查询结果存放到一个永久表中。

【答案】left(学号)或 substr(学号,1,4)，into。

14. 【解析】本题知识点同选择题 35 题，修改表中的某个字段的命令是：

ALTER table 表名 alter 字段名 类型(长度)

因此本题中应是 ALTER TABLE 学生 alter 学号 C(12)。

【答案】alter 学号 C(12)。