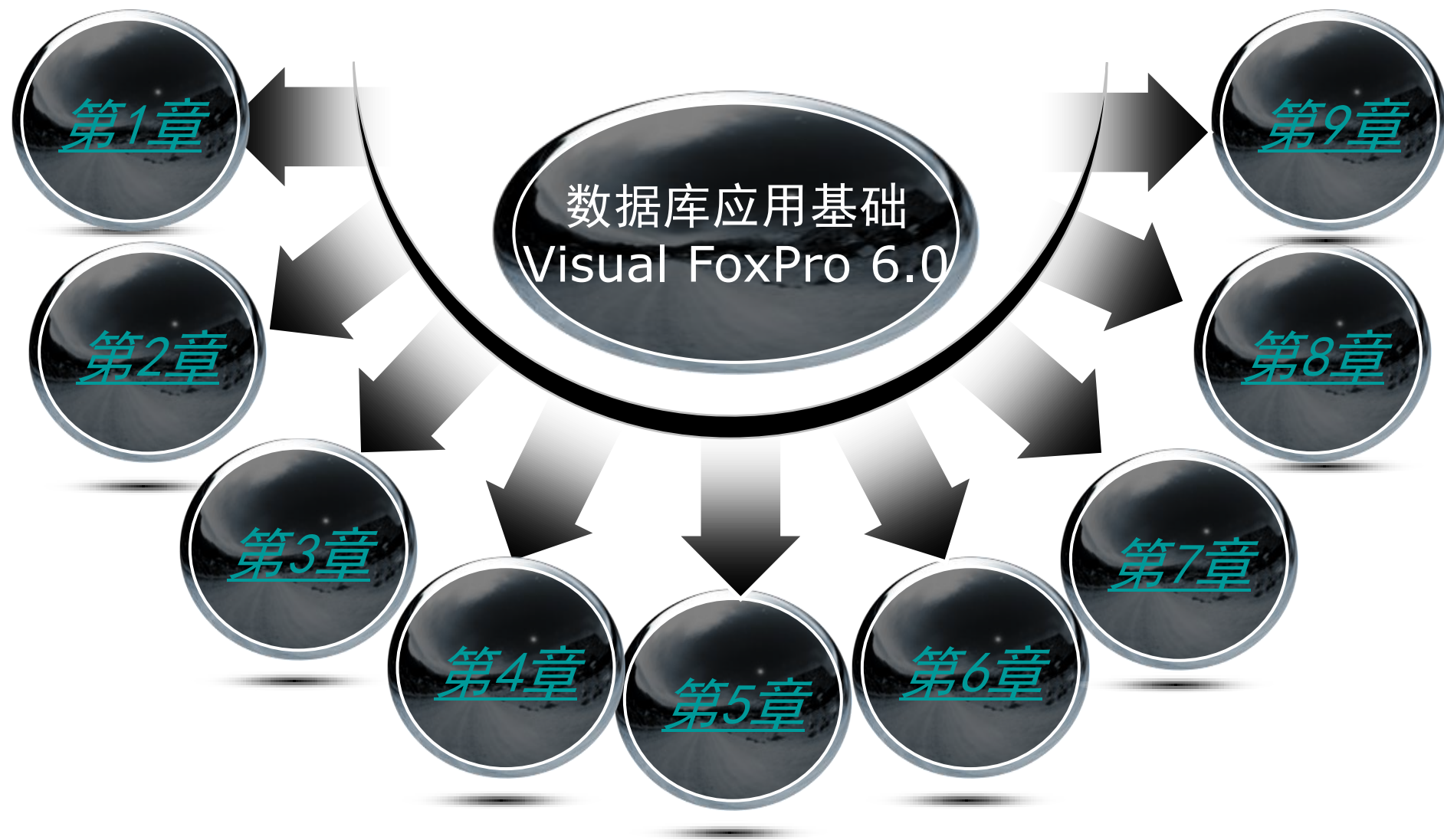


数据库应用基础
Visual FoxPro 6.0

本课程知识结构

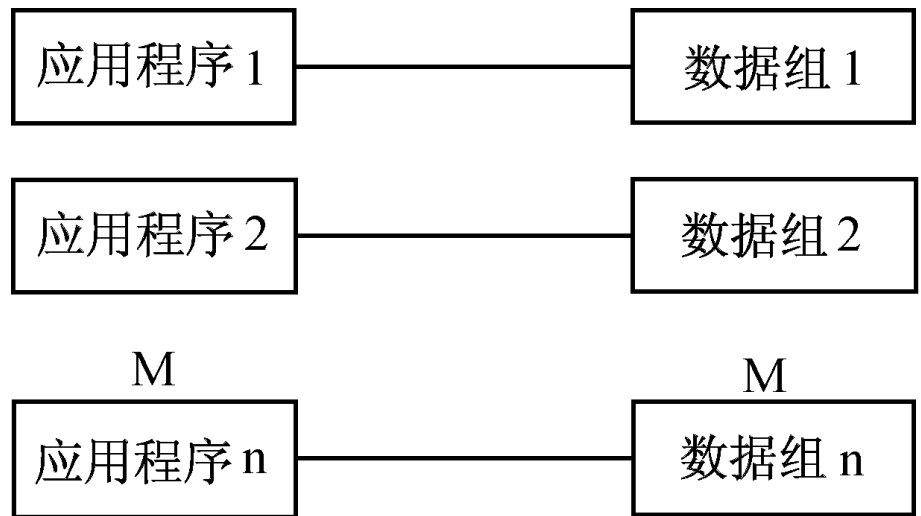


1.1 数据库基础知识

1.1.1 数据管理技术的发展

1. 人工管理阶段

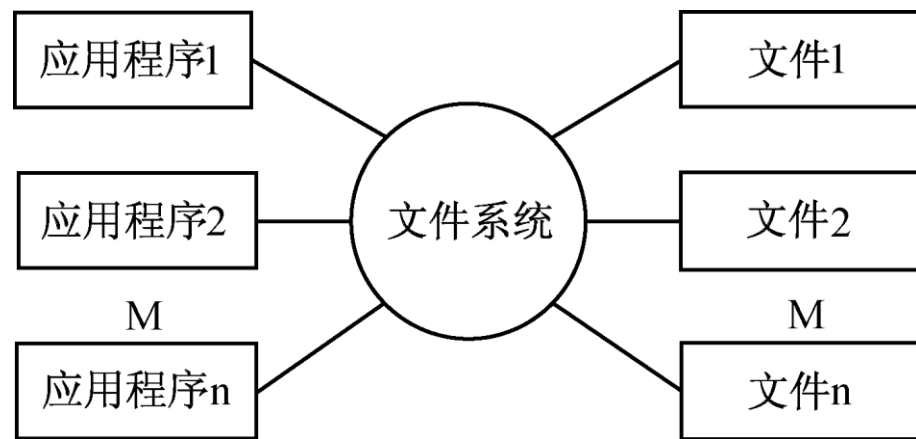
20世纪50年代中期以前，计算机主要用于科学计算，由于无直接用于存取数据的存储设备和专门用于数据处理的软件，数据只能由处理它的程序携带。因此，人工管理数据的特点是：数据不保存、不共享。这一时期数据和程序之间的关系如图所示。



1.1.1 数据管理技术的发展

2. 文件管理阶段

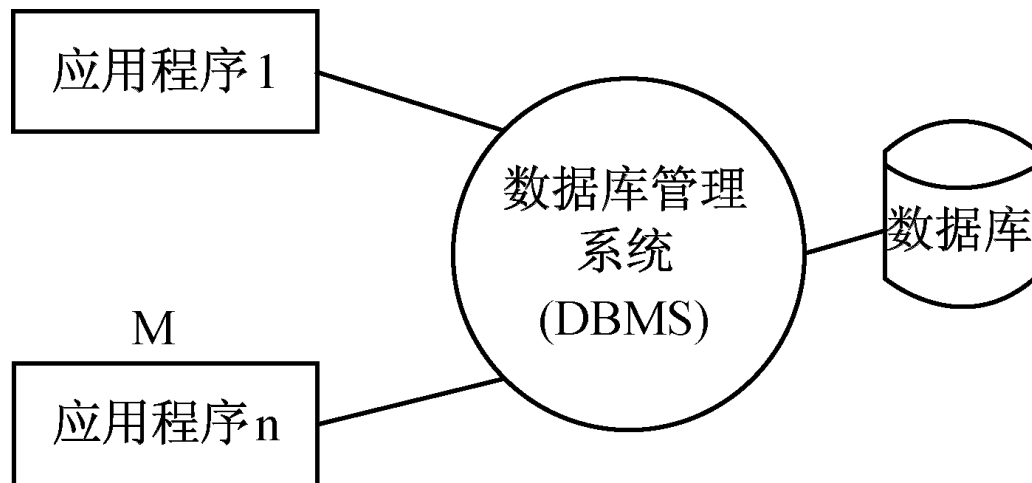
20世纪60年代初期，计算机被大量地用于数据处理、数据存储、检索和维护等工作。操作系统中文件管理模块（即输入/输出控制模块）的重要功能之一就是管理外存储器中的数据，其特点是，数据可以以文件形式长期保存。但由于文件之间缺乏联系，且同样的数据有可能在多个文件中重复存储，因此数据共享性差，冗余度大。这一时期数据和程序之间的关系如图所示。



1.1.1 数据管理技术的发展

3. 数据库系统阶段

20世纪60年代后期，数据管理技术进入数据库系统阶段。数据库系统克服了文件管理阶段的缺陷，提供了对数据更高级、更有效地管理。这个阶段程序和数据的联系通过数据库管理系统（DataBase Management System, DBMS）来实现，如图所示。



1.1.1 数据管理技术的发展

数据库系统阶段具有以下特点。

- (1) 数据结构化。
- (2) 较高的数据独立性。
- (3) 数据共享性高、冗余度小。
- (4) 数据库系统提供了数据控制功能。

1.1.2 数据库的基本概念

1. 信息

信息是有一定含义、已经被加工、有决策价值的**数据**。

2. 数据

数据是信息的符号表示。数据是指存储在某种媒体上能够被计算机处理的物理符号，它的内容是事物特征的反映。

3. 数据处理

数据处理是将数据转换成信息的全部过程，包括数据的收集、整理、存储、加工、分类、维护、排序、检索和传输等一系列活动。

4. 数据管理

根据客观事物的特点，对已知数据进行加工，获得新的数据。

5. 对象 (object)

现实生活中，任何可见的实体都可以视为一个对象。如一只气球是一个对象，一台计算机也是一个对象。

1.1.2 数据库的基本概念

6. 类 (class)

类是已经定义了的关于对象特征和行为的模板，是对象的原型。

7. 属性

每个对象都有一组特征，称为属性。对象的属性用来表示它的状态和特征，以命令按钮为例，其位置、大小、颜色等状态，都可用属性来表示。

8. 事件

事件是由系统预先定义好的、能够被对象识别的动作，如鼠标单击 (Click) 事件、双击 (DoubleClick) 事件等。

9. 方法

对象除拥有自己的属性和事件外，还拥有属于自己的方法。方法是系统提供的一种特殊的子程序，它是与对象相关联的过程，用来完成一定的操作或实现一定的功能。

1.1.3 数据库系统的概念和特点

1. 数据库系统的相关概念

1) 数据库

数据库 (DataBase, DB), 是存储在计算机存储设备上的结构化的相关数据集合。

2) 数据库应用系统

数据库应用系统是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。

3) 数据库管理系统

数据库管理系统 (DataBase Management System, DBMS), 是位于用户和操作系统之间的一个数据管理软件。

1.1.3 数据库系统的概念和特点

4) 数据库系统

数据库系统 (DataBase System, DBS), 是指在计算机系统中引入数据库后的系统构成, 包括计算机硬件系统、数据库、数据库管理系统 (DBMS) 及其工具、数据库应用系统、数据库管理员 (DBA) 和用户。

5) 数据库语言

数据库语言由数据定义语言 (Data Definition Language, DDL) 和数据操纵语言 (Data Manipulation Language, DML) 组成, 为用户提供了交互式使用数据库的方法。

1.1.3 数据库系统的概念和特点

2. 数据库系统的特点

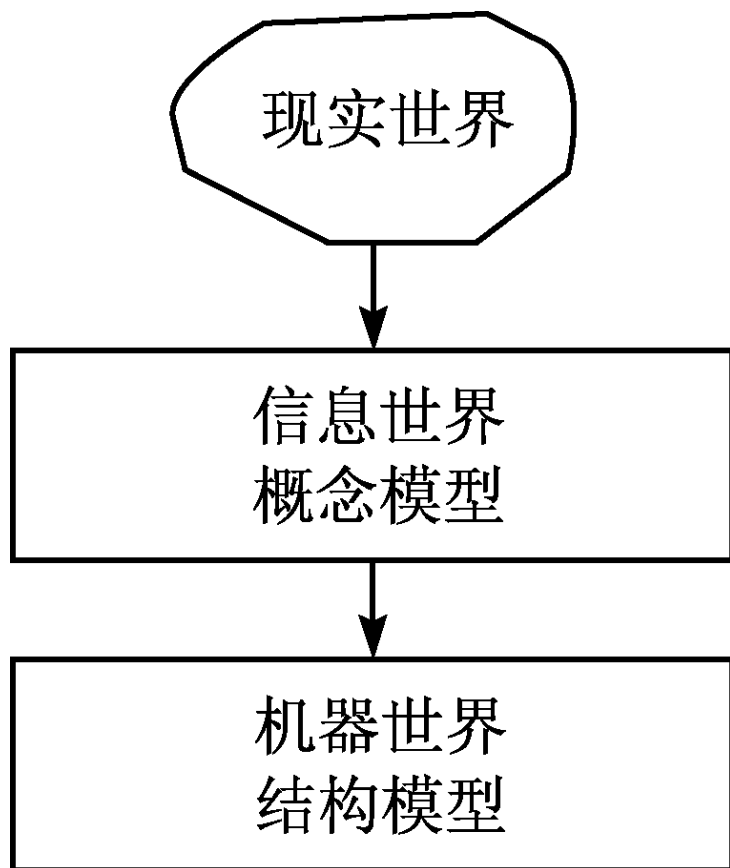
数据库系统的主要特点如下：

- 1) 实现数据共享，减少数据冗余度
- 2) 采用特定的数据模型
- 3) 具有较高的数据独立性
- 4) 具有统一的数据控制功能

1.1.4 数据模型

数据模型是现实世界数据特征的抽象。现实世界中大量信息需要经过抽象，提取其中有意义、有价值的信息，并以数据形式表示。数据模型是描述这些数据及其联系的工具

如图所示，数据抽象过程通常经过两步：现实世界→信息世界；信息世界→机器世界。根据模型应用的不同目的，数据模型分为两个层次：概念模型和结构模型。



1.1.4 数据模型

2. 数据模型的要素

1) 数据结构

数据结构用于描述系统的静态特性,是刻画一个数据模型最重要的方面。因此,在数据库模型系统中,通常按照其数据结构的类型来命名数据模型。

2) 数据操作

数据操作用于描述系统的动态特性。

数据操作是指对数据库中各种对象的实例允许执行的操作集合,包括操作及有关的操作规则。

3) 数据完整性约束

数据完整性约束是一组完整性规则的集合。完整性规则是给定的数据模型中数据及其联系所具有的制约和储存规则,用以限定符合数据模型的数据库以及状态的变化,以保证数据的正确、有效和相容。

1.1.4 数据模型

3. 概念模型 (E-R模型)

1) 信息世界中的基本概念

- (1) 实体。
- (2) 实体的属性。
- (3) 域。
- (4) 实体型。
- (5) 实体集。
- (6) 主码。
- (7) 联系。

1.1.4 数据模型

2) 概念模型的表示方法

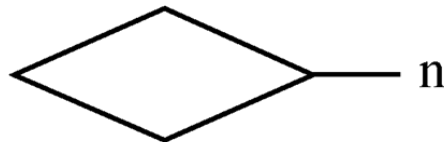
概念模型是对信息世界建模，所以概念模型应该能够方便、准确地表示信息世界中的常用概念。

概念模型的表示方法一般采用E-R图描述现实世界的概念模型，称为实体-联系模型（entity relationship model, E-R模型）。

E-R图提供了表示实体型、属性和联系的方法。E-R图中的符号及其意义如图所示。



实体型



实体间的联系

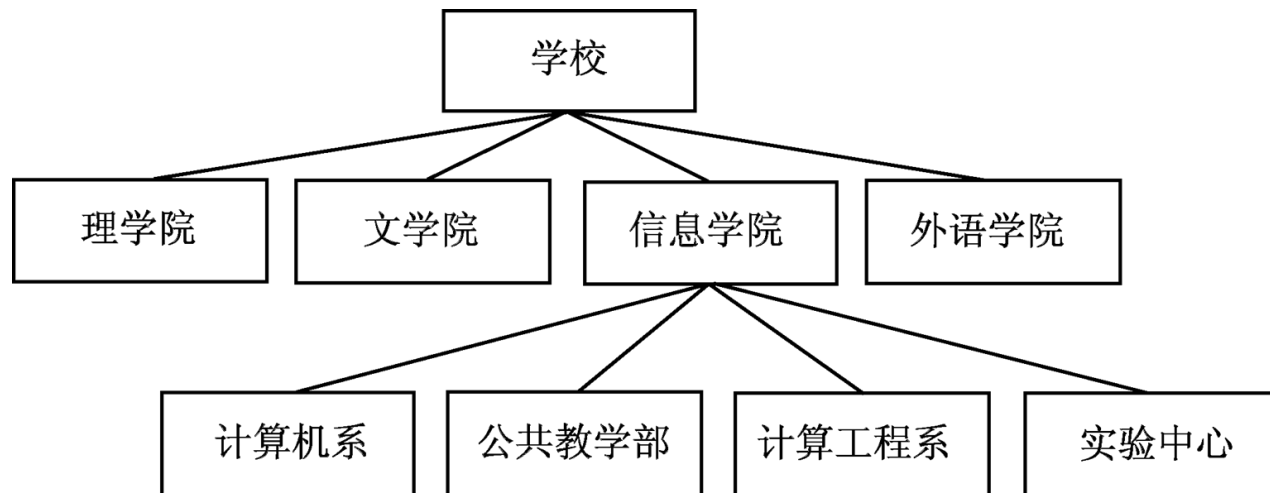


实体的属性

1.1.4 数据模型

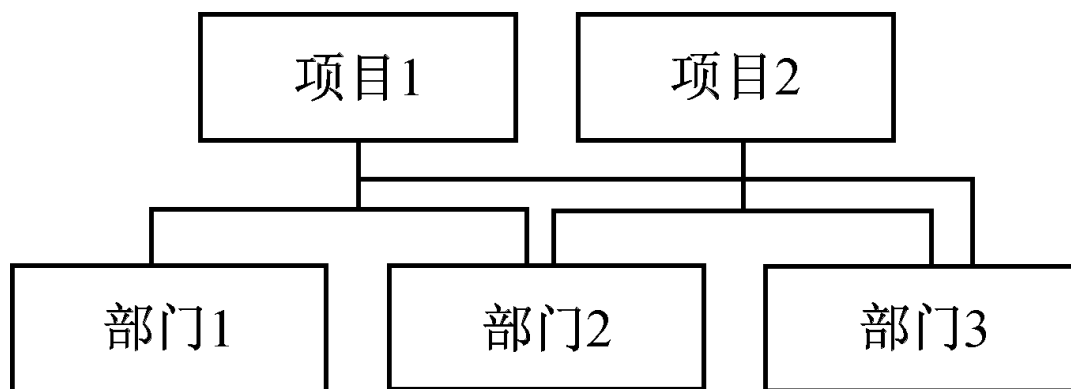
4. 结构数据模型

(1) 层次模型 (hierarchical model)：用树形结构表示实体间联系的数据模型。该模型的实际存储数据由链接指针来体现联系。其特点是：有且仅有一个结点，无父结点，此结点即为根结点；其他结点有且仅有一个父结点；适合于表示一对多的联系, 如图所示。



1.1.4 数据模型

(2) 网状模型 (network model)：用有向图表示实体类型及实体间联系的数据模型。其特点是：允许结点有多个父结点，也可以有多个结点无父结点，适合于表示多对多的联系, 如图所示。



1.1.4 数据模型

(3) 关系模型 (relational model)：在关系模型中，数据被组织成若干张二维表，每张表称为一个“关系”。一张表格中的一列称为一个“属性”，相当于记录中的一个数据项（或称为字段），属性的取值范围称为“域”。表格中的一行称为一个“元组”，相当于记录值。可用一个或若干个属性集合的值标识这些元组，称为“关键字”。每一行对应的属性值称为一个“分量”。

(4) 面向对象模型 (object oriented model)：主要采用对象和类的概念，用于存储彼此没有内在联系的数据对象（而不必把它们安排到数据表中）。面向对象的数据产品在市场上只占相当小的份额。

1.2 关系数据库

1.2.1 基本概念

1. 关系术语

在Visual FoxPro中，一个“表”就是一个关系。

1) 关系

一个关系就是一张二维表格（.dbf文件）。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。格式为：

关系名（属性名1，属性名2，…，属性名n）

在Visual FoxPro中表示为表结构：

表名（字段名1，字段名2，…，字段名n）

2) 元组

二维表中水平方向的行称为元组，每一行为一个元组。在Visual FoxPro中表示为表记录。

1.2.1 基本概念

3) 属性

二维表中垂直方向的列称为属性。在Visual FoxPro中表示为字段。每一列有一个属性名。在Visual FoxPro中表示为字段名。

4) 域

域是指属性的取值范围。例如，姓名的域是文字字符；性别的域为（男，女）；逻辑型属性（是否借完）的域为逻辑真或逻辑假值

5) 关键字

属性或属性的组合（字段或字段的组合），其值能够唯一地标识一个元组（主关键字和候选关键字）。

6) 外部关键字

如果表中的一个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，这个字段（属性）就称为外部关键字（公共字段名在表间起桥梁作用）。

1.2.1 基本概念

2. 关系的特点

- (1) 关系必须规范化(每一列均为不可分割的最小数据项)。
- (2) 同一关系中不能出现相同的属性名(表中字段名)。
- (3) 关系中不允许有完全相同的元组,即冗余。
- (4) 在一个关系中元组的次序无关紧要(不影响数据的实际含义)。
- (5) 在一个关系中列的次序无关紧要(不影响数据的实际含义)。

3. 实际关系模型

在Visual FoxPro中,一个数据库中包含相互之间存在联系多个表。这个数据库文件就代表一个实际的关系模型。即一个数据库(.dbc文件)就是一个实际关系模型,它是一个或多个表(.dbf文件)或视图信息的容器。

1.2.2 关系运算

1. 传统的集合运算

进行并、差、交集运算的两个关系必须是具有相同的模式，即结构相同。

1) 并运算

两个相同结构的并是有属于这两个关系的元组（记录）组成的集合。

2) 差运算

有关系R和关系S，是有属于R而不属于S的元组组成的集合，从R中去掉S中也有的元组。

3) 交运算

有关系R和关系S，既属于R又属于S的元组组成的集合。

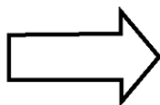
1.2.2 关系运算

2. 专门的关系运算

1) 选择 (selection)

选择是指从关系中找出满足指定条件的元组的操作。选择是从行的角度进行的运算，即选择水平方向的记录。选择的对象是一个表。运算是从关系中查找符合指定条件元组的操作。经过选择运算得到的结果可以形成新的关系，其关系模式不变，但其中的元组是原来的关系的一个子集，如图所示。

学号	姓名	班级
200810279	王艳燕	08机设1班
200810281	高鹏	08机设2班
200810282	杨金柳	08机设1班
200810283	吴军	08机设2班



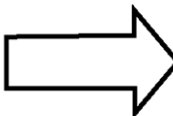
学号	姓名	班级
200810279	王艳燕	08机设1班
200810282	杨金柳	08机设1班

1.2.2 关系运算

2) 投影 (projection)

投影是指从一个关系模式中选择若干个属性组成新的关系的操作。投影是从列的角度进行运算。投影的操作对象是一个表。运算是从关系模式中选取若干个属性的操作。经过投影运算可以得到一个新的关系，其关系模式所包含的属性个数往往比原来的关系少，或者属性的排列顺序不同，如图所示。

学 号	姓 名	班 级
200810279	王艳燕	08机设1班
200810281	高鹏	08机设2班
200810282	杨金柳	08机设1班
200810283	吴军	08机设2班



学 号	班 级
200810279	08机设1班
200810281	08机设2班
200810282	08机设1班
200810283	08机设2班

1.2.2 关系运算

3) 连接 (join)

连接是从两个关系模式选择符合条件的元组或属性组成一个新的关系。连接结果是满足指定条件的所有记录。连接的操作对象是两个表。运算是将两个关系模式的若干属性拼接成一个新的关系模式的操作，对应的新关系中，包含满足连接条件的所有元组。

1.2.3 关系完整性

1. 实体完整性
2. 值域完整性
3. 用户自定义完整性
4. 参照完整性

1.2.4 数据库设计原则

数据库设计遵循如下原则：

- (1) 遵从概念单一化、“一事一地”的原则（一个表描述一个实体或实体间的一种联系）。
- (2) 避免表间字段重复（减少数据冗余）。
- (3) 表中字段必须是原始数据和基本数据元素（表中不应包括计算可以得到的“二次数据”或多项数据的组合）。
- (4) 用外部关键字保证有关联的表间的联系。

1.2.5 数据库系统设计步骤

设计数据库系统一般要经过以下步骤。

- 1) 需求分析
- 2) 概念结构设计
- 3) 逻辑结构设计
- 4) 物理结构设计
- 5) 数据库系统实施
- 6) 数据库系统运行和维护

1.3 Visual FoxPro 6.0概述

1.3.1 Visual FoxPro 6.0的发展历史

Visual FoxPro原名FoxBase，最初是由美国Fox Software公司于1988年推出的数据库产品，在DOS操作系统上运行。FoxPro是FoxBase的加强版，最高版本为2.6。

1992年，微软公司收购Fox Software公司，继续发展FoxBase，使其可以在Windows操作系统上运行，并且更名为Visual FoxPro。1995年，微软公司推出的Visual FoxPro 3.0很快占据市场主流。1997年，微软公司推出了Visual FoxPro 5.0。

1998年，微软公司推出Visual FoxPro 6.0，并将Visual FoxPro 6.0加入Visual Studio的系列产品中。

后来，为了适应.NET的发展策略，微软公司又将 Visual FoxPro 移出Visual Studio，并将Visual Studio更名为Visual Studio.NET。目前Visual FoxPro的最新版本是9.0。

1.3.2 Visual FoxPro 6.0的特点

Visual FoxPro 6.0的特点主要体现在以下几方面：

1. 易于开发应用程序
2. 具有强大的查询和视图设计功能
3. 支持面向对象
4. 增强了对SQL的支持
5. 集成了OLE与ActiveX的支持

1.3.3 Visual FoxPro 6.0的文件类型

Visual FoxPro数据库中的各种数据都以文件的形式存储在磁盘中，不同类型的文件保存了不同的数据信息。文件的扩展名用于区分不同类型的文件。

1.4 Visual FoxPro 6.0的集成开发环境

1.4.1 Visual FoxPro 6.0的工作方式

1. 交互方式

1) 命令执行方式

命令执行方式是指在命令窗口中输入命令执行数据库操作。

2) 可视化操作

可视化的操作可通过菜单或工具执行。

(1) 菜单执行方式：大多数系统命令都有相应的菜单命令，通过键盘或鼠标选择菜单命令可执行相应的数据库操作。

(2) 工具执行方式：Visual FoxPro为用户提供了多种工具，如数据库设计器、表设计器和窗体设计器等，用户在设计器的对话框中单击相应的工具即可完成各种数据库操作。

2. 程序执行方式

程序执行方式是指将执行一系列数据库操作命令编写成一个程序，通过运行程序完成一系列操作。

1.4.2 Visual FoxPro 6.0的工作界面

Visual FoxPro 6.0主界面主要由标题栏、菜单栏、工具栏、状态栏、工作区和命令窗口组成，如图所示。



1.4.2 Visual FoxPro 6.0的工作界面

1. 菜单栏

菜单栏显示了Visual FoxPro 6.0的各个主菜单项，包括“文件”、“编辑”、“显示”、“格式”、“工具”、“程序”、“窗口”和“帮助”，其他一些菜单项，如“表”、“表单”和“报表”等根据当前操作状态自动显示。例如，在执行Browse命令浏览表时，“表”菜单项自动出现在菜单栏中；在设计表单时则自动显示“表单”菜单项。

- 1) 键盘选择键
- 2) 快捷键
- 3) 省略号
- 4) 无效命令
- 5) 复选标志

1.4.2 Visual FoxPro 6.0的工作界面

2. 工具栏

使用下列两种方法可打开工具栏

(1) 右击已打开的工具栏，在弹出的快捷菜单中选择需要打开的工具栏。

(2) 单击“显示”→“工具栏”命令，打开“工具栏”对话框，如图所示。在对话框中选中工具栏对应的复选框，然后单击“确定”按钮即可。



1.4.2 Visual FoxPro 6.0的工作界面

3. 状态栏

状态栏位于Visual FoxPro主窗口的最下方，用于显示命令提示或命令执行结果。

4. 命令窗口

命令窗口用于执行各种命令。在命令窗口中输入命令后按Enter键即可执行命令。执行过的命令显示在命令窗口中，要执行相同的命令，只需将光标移动到该命令所在行，再按Enter键即可。

打开或关闭命令窗口有以下三种方法：

- (1) 按Ctrl+F2组合键打开命令窗口，按Ctrl+F4组合键关闭命令窗口
- (2) 单击“常用”工具栏中的“命令窗口”按钮可打开或关闭命令窗口。
- (3) 单击“窗口”→“命令窗口”命令打开命令窗口，单击“窗口”→“隐藏”命令关闭命令窗口。

5. 工作区

工作区是位于工具栏和状态栏之间的空白区域，用于显示命令或程序的运行结果。

1.4.3 Visual FoxPro 6.0项目管理器

项目管理器为用户组织数据提供了一个良好的分层结构视图。如果要处理项目中某一特定类型的文件或对象，可以选择相应的选项卡，如图所示。



1.4.3 Visual FoxPro 6.0项目管理器

1. 创建项目

命令方式:

CREATE PROJECT <文件名>

菜单方式:

- (1) 打开“文件”菜单，选择“新建”菜单项。
- (2) 选择“项目”文件类型，单击“新建文件”按钮。
- (3) 在“创建”对话框中输入项目文件名, 选择项目文件类型(.pjx) 及保存位置。
- (4) 单击“保存”按钮。

1.4.3 Visual FoxPro 6.0项目管理器

2. 查看项目内容

项目管理器中的项目类似于大纲组织结构，可以展开（前面带有“+”号）或折叠（前面带有“-”号），以便查看不同层次中的详细内容。

3. 定制项目管理器

项目管理器是一个独立的窗口，可以移动位置、改变大小，单击项目管理器窗口右上角的上箭头可以将其折叠或展开，也可以将项目管理器窗口“停放”在工具栏的位置。

1.4.3 Visual FoxPro 6.0项目管理器

4. 为文件添加说明

创建或新建文件时，可以为文件添加说明。当添加有说明的文件被选定时，为文件添加的说明将显示在项目管理器底部。

要为文件添加说明的步骤：

- (1) 在“项目管理器”中选定文件。
- (2) 在“项目”菜单中选择“编辑说明”菜单项。
- (3) 在“说明”对话框中输入文件说明信息。
- (4) 单击“确定”按钮。

1.4.4 Visual FoxPro 6.0的设计器

VFP系统提供的设计器，为用户提供了一个友好的图形用户界面。用户可以通过它创建表、数据库、表单、报表等，并定制数据表结构、数据库结构、报表格式和应用程序组件等。