

Visual FoxPro

程序设计

精讲精练

主 编 党 锋

考点归纳：按新大纲的要求，对考点知识进行了总结归纳性的讲解。

例题精讲：针对新大纲要求，对大量的重点试题进行了详细的分析和解答，以加深考生对考点的理解和掌握考试内容、范围和难度。

考点精练：提供了大量有针对性的模拟试题，这些试题经过精心设计和锤炼，突出了考点、重点和难点。

上机指导：对上机考试的各个环节进行了介绍，以让考生在考前熟悉上机环境，以免由于考试环境的不熟悉影响水平的发挥。



全国计算机等级考试二级精讲精练系列丛书

Visual FoxPro 程序设计精讲精练

主 编 党 锋

丛书编写组 李怀强 党 锋 常家忠

刘敏力 陈 莉 赵海兰

杨 锋 杨朝斌 付 伟

电子科技大学出版社

图书在版编目 (CIP) 数据

Visual FoxPro 程序设计精讲精练 / 党锋主编.

— 成都: 电子科技大学出版社, 2005.2

(全国计算机等级考试二级精讲精练系列丛书)

ISBN 7-81094-760-5

I. V... II. 党... III. 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. TP311.138

中国版本图书馆 CIP 数据核字 (2005) 第 012326 号

内 容 提 要

本书根据 2004 年教育部考试中心最新颁布的全国计算机等级考试 (二级 Visual Foxpro 数据库程序设计) 大纲编写而成。各章按“考点归纳”、“例题精讲”、“考点精练”分为三大部分: “考点归纳”部分按新大纲的要求, 对考点知识进行了总结归纳性的讲解; “例题精讲”部分针对新大纲要求, 对大量的重点试题进行了详细的分析和解答, 以加深考生对考点的理解和掌握考试内容、范围和难度; “考点精练”部分提供了大量有针对性的模拟试题, 这些试题经过精心设计和锤炼, 突出了考点、重点和难点。另外, 本书还安排有“上机指导”内容, 对上机考试的各个环节进行了介绍, 以便让考生在考前熟悉上机环境。

本书力求考点的精讲和针对考点大量典型试题的精练, 使考生能在短时间内掌握考点内容, 顺利通过考试。

④ 全国计算机等级考试二级精讲精练系列丛书

Visual FoxPro 程序设计精讲精练

主编 党 锋

出 版	电子科技大学出版社 (成都市建设北路二段四号, 邮编: 610054)
责任编辑	钟旺志
发 行	电子科技大学出版社
印 刷	成都蜀通印务有限责任公司
开 本	787×1092 1/16 印张 15.75 字数 400 千字
版 次	2005 年 3 月第一版
印 次	2005 年 3 月第一次印刷
书 号	ISBN 7-81094-760-5/TP·410
印 数	1—3000 册
定 价	19.00 元

■ 版权所有 侵权必究 ■

- ◆ 邮购本书请与本社发行科联系。电话: (028) 83201495 邮编: 610054
- ◆ 本书如有缺页、破损、装订错误, 请寄回印刷厂调换。

前 言

国家教育部考试中心顺应社会发展的需要,于1994年推出“全国计算机等级考试”,其目的是以考促学,向社会推广普及计算机知识,从而为各单位和组织录用与考核选拔人才提供了统一、公正、客观和科学的评测手段。由于全国计算机等级考试具有较高的权威性,因而得到了全社会的认可,成为我国规模最大、影响最大的计算机知识与能力的考试。开考以来,截至2004年下半年,已顺利考过20次,千余个考点遍布全国30个省市。考生累计人数达1000多万,累计获得证书人数达300多万。根据我国计算机应用水平的实际情况,教育部考试中心于2004年对计算机等级考试大纲重新进行了修订,并正式颁布新的考试大纲。

为了适应教育部考试中心对计算机等级考试科目和内容的新调整,我们组织了国内部分高校在教学一线工作的教师对新的考试大纲进行了深入研究,并结合实际的教学经验和近年来考生考前复习的情况,精心组织编写了这套全国计算机等级考试二级精讲精练系列丛书。本套丛书共六本:

- ◆ C++语言程序设计精讲精练
- ◆ C语言程序设计精讲精练
- ◆ Java语言程序设计精讲精练
- ◆ Access数据库程序设计精讲精练
- ◆ Visual Basic语言程序设计精讲精练
- ◆ Visual FoxPro程序设计精讲精练

每本书的每章内容分为“考点归纳”、“例题精讲”、“考点精练”三大部分。

考点归纳:按新大纲的要求,对考点知识进行了总结归纳性的讲解。

例题精讲:针对新大纲要求,对大量的重点试题进行了详细的分析和解答,以加深考生对考点的理解和掌握考试内容、范围和难度。

考点精练:提供了大量有针对性的模拟试题,这些试题经过精心设计和锤炼,突出了考点、重点和难点。

另外,本书还安排有“上机指导”内容,对上机考试的各个环节进行了介绍,以让考生在考前熟悉上机环境,以免由于考试环境的不熟悉影响水平的发挥。

本书在编写过程中充分考虑了等级考试的性质和考生考前复习及应试的特点,力求考点的精讲和针对考点大量典型试题的精练,以帮助考生在短期的复习中获得好的效果,从而加深对基本概念的理解,熟悉等级考试的形式和试题,掌握考点知识,突破难点,熟练掌握答题方法及技巧,以便在考试中发挥出水平并顺利通过考试。在这里,我们预祝考生朋友在考试中取得好成绩。

编写组
2005年1月

目 录

第1章 数据库系统基础	1
考点归纳	1
例题精讲	10
考点精练	14
参考答案	24
第2章 Visual FoxPro 系统初步	28
考点归纳	28
例题精讲	32
考点精练	36
参考答案	38
第3章 数据与数据运算	39
考点归纳	39
例题精讲	47
考点精练	52
参考答案	63
第4章 Visual FoxPro 数据库及其操作	66
考点归纳	66
例题精讲	72
考点精练	77
参考答案	88
第5章 关系数据库标准语言 SQL	91
考点归纳	91
例题精讲	95
考点精练	102
参考答案	107
第6章 查询与视图	109
考点归纳	109
例题精讲	111
考点精练	113
参考答案	117
第7章 程序设计基础	119
考点归纳	119
例题精讲	124
考点精练	136

参考答案.....	157
第8章 表单设计与应用.....	160
考点归纳.....	160
例题精讲.....	164
考点精练.....	173
参考答案.....	185
第9章 菜单设计与应用.....	189
考点归纳.....	189
例题精讲.....	192
考点精练.....	194
参考答案.....	196
第10章 报表设计	197
考点归纳.....	197
例题精讲.....	201
考点精练.....	204
参考答案.....	206
第11章 开发应用程序	207
考点归纳.....	207
例题精讲.....	209
考点精练.....	214
参考答案.....	217
第12章 上机考试指导	218
考试内容.....	218
考试步骤.....	218
考试方法精讲.....	219
上机考试模拟实训.....	223
参考答案.....	225
附录A 2004年9月全国计算机等级考试	231
二级笔试 Visual FoxPro 程序设计试卷.....	231
附录B 2004年9月全国计算机等级考试	239
二级笔试 Visual FoxPro 程序设计试卷参考答案及评分标准.....	239
附录C 二级笔试 Visual FoxPro 考试大纳	243

第 1 章 数据库系统基础

考 点 归 纳

1.1 数据库基本概念

一、数据、数据库、数据库管理系统和数据库系统

1. 数据

数据是指存储在某一媒体上能被识别的物理符号。数据的概念包括两个方面:其一是描述事物特性的数据内容;其二是存储在某一媒体上的数据形式。由于描述事物特性必须借助于一定的符号,这些符号就是数据形式。

数据的概念在数据处理领域已经大大地拓宽了。数据不仅仅指数字、字母、文字和其他特殊字符组成的文本形式的数据,而且还包括图形、图像、动画、影像、声音等多媒体数据。

2. 数据库

数据库可以看成是长期存储在计算机内的、有组织的、可共享的数据集合。

数据库可以直观地理解为存放数据的仓库,只不过这个仓库是在计算机的大容量存储器上,例如硬盘就是一种最常见的计算机大容量存储设备,而且数据必须按一定的格式存放,因为它不仅需要存放,而且还要便于查找。

3. 数据库管理系统

数据库管理系统(DataBase Management System,简称 DBMS)是位于用户与操作系统之间的完成数据管理的系统软件。它的主要功能包括以下几个方面:

(1) 数据定义功能

提供“数据定义语言”(DDL),用户通过它可以方便地对数据库中的相关内容进行定义。例如,对数据库、表、索引进行定义。

(2) 数据操作功能

提供“数据操作语言”(DML),支持用户对数据库中的数据进行查询、更新(包括增、删、改)等操作。

(3) 数据库运行控制功能

包括并发控制(即处理多个用户同时使用某些数据时可能产生的问题)、安全性检查、完整性约束条件的检查和执行、数据库恢复以及数据库的内部维护(例如,索引的自动维护),这是 DBMS 的核心部分,所有数据库的操作都要在这些控制程序的统一管理下进行。

(4) 数据库的建立和维护功能

数据库的建立和维护功能包括数据库初始数据的输入、转换功能,数据库的转储、恢复功能,数据库的重新组织功能和性能监视、分析功能等。

4. 数据库系统

数据库系统是指在计算机系统中引入数据库后的系统,一般由数据库、数据库管理系统(及其开发工具)、应用系统、数据库管理员和用户构成。

数据库的建立、使用、维护等工作只靠一个数据库管理系统(DBMS)远远不够,还要有专门的人员来完成,这些人被称为数据库管理员(DataBase Administrator,简称DBA)。

二、数据管理技术的发展

计算机对数据的管理是指为数据的组织、分类、编码、存储、检索和维护提供操作手段。

与其他技术的发展一样,计算机数据管理也经历了由低级到高级的发展过程。计算机数据管理随着计算机硬件、软件技术和计算机应用范围的发展而不断发展,大致经历了如下三个阶段:

1. 人工管理阶段

20世纪50年代以前,计算机主要用于数值计算。从当时的硬件看,外存只有纸带、卡片、磁带,没有直接存取设备;从软件看(实际上,当时还未形成软件的整体概念),没有操作系统及管理数据的软件;从数据看,数据量小,数据无结构,由用户直接管理,且数据间缺乏逻辑组织,数据依赖于特定的应用程序,缺乏独立性。

2. 文件系统阶段

20世纪50年代后期到60年代中期,这时硬件方面已有了硬盘、磁鼓等直接存取存储设备;软件方面,操作系统中已经有了专门的数据管理软件,一般称为文件系统。

文件系统把数据组织成相互独立的数据文件,用户可以对文件进行增、删、改的操作。文件系统实现了记录内的结构性,但整体无结构。程序和数据之间由文件系统提供存取方法来转换,从而使应用程序与数据之间有了一定的独立性。其数据面向特定的应用程序,因此数据共享性、独立性差,且冗余度大,管理和维护的代价也很大。

3. 数据库系统阶段

20世纪60年代后期,计算机性能得到提高,出现了大容量磁盘。在此基础上,出现了数据库这样的数据管理技术。数据库的特点是数据不再只针对某一特定应用,而是面向全组织,具有整体的结构性,共享性高,因此冗余度小,具有一定的程序与数据间的独立性,并且实现了对数据进行统一的控制。

从文件系统到数据库系统,标志着数据管理技术质的飞跃。20世纪80年代后,不仅在大、中型机上实现并应用了数据库管理系统,在微型计算机上也配置了经过简化的数据库管理系统(例如,Visual FoxPro等),从而使数据库技术得到广泛的应用和普及。

三、数据库系统的发展

第一代数据库系统又称非关系型数据库系统,其中包括“层次”与“网状”两种类型。这一代数据库系统采用“记录”为基本的数据结构。在不同的“记录型”之间,允许存在相互联系。无论层次模型或网状模型,一次查询只能访问数据库中的一个记录,存取效率不高。对于具有复杂联系的系统,还需用户详细描述对数据的访问路径(存取路径)。

第二代数据库系统又称关系型数据库系统(RDBS)。它采用了人们习惯使用的二维表格作为基本的数据结构(“关系模型”),通过公共的关键字段来实现不同二维表之间的数据联系。RDBS允许一次查询仅用一条命令或语句来访问整个“关系”;通过多表联合操作,还能在有联系的多个二维表之间实现“关联”查询。关系模型简单明了,理论严谨,容易理解,使用起来方便,因此受到普遍欢迎。

第三代数据库系统又称对象关系数据库系统(ORDBS)。它将数据库技术与面向对象技术相结合,是第三代数据库系统的基础。第三代数据库系统除应包含第二代数据库系统的功能外,应能支持类、继承、函数/方法等丰富的对象机制,还应支持正文、图像、声音等新的数据

类型,并能提供高度集成的、可支持客户机/服务器应用的用户接口。这种数据库系统目前还处在发展的过程中,在技术上和应用上都还有许多工作要做,但已经显示出光明的发展前景。

四、数据库系统的组成

数据库系统由四部分组成:硬件系统、系统软件(包括操作系统和数据库管理系统)、数据库应用系统和各类人员(包括数据库管理员、系统分析员、数据库设计人员、应用程序员和最终用户)。

1. 硬件系统

整个数据库系统对硬件资源提出了较高的要求,这些要求主要包括:(1)有足够大的内存来存放操作系统、DBMS 的核心模块、数据缓冲区和应用程序;(2)有足够大的直接存取设备来存放数据(例如,磁盘),有足够的其他存储设备来进行数据备份;(3)有较高的数据传输能力,以提高数据传送率。

2. 系统软件

系统软件主要包括操作系统、数据库管理系统、与数据库接口的高级语言及其编译系统和以 DBMS 为核心的应用开发工具。

操作系统是计算机系统必不可少的系统软件,也是支持 DBMS 运行必不可少的系统软件。

数据库管理系统是数据库系统不可或缺的系统软件,它提供数据库的建立、使用和维护功能。

一般来讲,数据库管理系统的数据处理能力较弱,所以需要提提供与数据库接口的高级语言及其编译系统,以便于开发应用程序。

以 DBMS 为核心的应用开发工具是系统为应用开发人员和最终用户提供的高效率、多功能的应用程序生成器、第四代语言等各种软件工具。例如,报表设计器、表单设计器等。它们为数据库系统的开发和应用提供了有力的支持。当前,开发工具已成为数据库软件的有机组成部分。

3. 数据库应用系统

数据库应用系统是为特定应用开发的数据库应用软件。数据库应用系统是对数据库中的数据进行处理和加工的软件,它面向特定应用。例如,基于数据库的各种管理软件:管理信息系统、决策支持系统、办公自动化等都属于数据库应用系统。

4. 各类人员

参与分析、设计、管理、维护和使用数据库的人员均是数据库系统的组成部分,这些人员分别对应的是数据库管理员、系统分析员、应用程序员和最终用户。

五、数据库系统的特点

1. 数据结构化

数据库系统实现整个数据的结构化,是数据库的主要特征之一,也是数据库系统与文件系统的本质区别。

2. 数据的共享性高,冗余度低,易扩充

数据库系统中,数据不再面向某个应用,而是面向整个系统,因此数据可以被多个用户、多个应用共享使用。数据共享可以减少数据冗余,节约存储空间,数据共享还能够避免数据之间的不相容性与不一致性,同时还易于扩充。

3. 数据独立性高

数据独立性包括数据的物理独立性和逻辑独立性。

物理独立性是指用户的应用程序与存储在磁盘上的数据库中的数据是相互独立的。数据在磁盘上的数据库中怎样存储是由 DBMS 管理的,用户程序不需要了解,应用程序要处理的只是数据的逻辑结构,这样当数据的物理存储结构改变时,应用程序不用改变。

逻辑独立性是指用户的应用程序与数据库的逻辑结构是相互独立的,也就是说,数据的逻辑结构改变了,用户程序也可以不变。

数据独立性是由 DBMS 的二级映像功能来保证的。

4. 数据由 DBMS 统一管理和控制

数据库的共享是并发的共享,即多个用户可以同时存取数据库中的数据,甚至可以同时存取数据库中同一数据。为此,DBMS 必须提供以下几方面的数据控制功能:

(1) 数据的安全性保护

数据的安全性是指保护数据以防止不合法的使用所造成的数据泄密和损坏。每个用户只能按被授权方式对数据进行使用和处理。

(2) 数据的完整性检查

数据的完整性检查,是对数据的正确性、有效性和相容性进行检查。

(3) 并发控制

当多个并发用户进程同时存取或修改数据库时,可能会发生相互干扰而得到错误的结果或使得数据库的完整性遭到破坏,因此必须对多用户的并发操作加以控制和协调。

六、数据库系统的结构

考察数据库系统的结构可以有多种不同的层次或不同的角度。从数据库管理系统角度看,数据库系统通常采用三级模式结构:模式、外模式和内模式,这是数据库管理系统内部的系统结构。

从数据库最终用户角度看,数据库系统的结构分为集中式结构、分布式结构、客户机/服务器结构和并行结构。这是数据库系统外部的体系结构。

1. 数据库系统模式的概念

数据模型中有“型”和“值”的概念。型是指对某一类数据的结构和属性的说明,值是型的一个具体赋值。模式是数据库中全体数据的逻辑结构和特征的描述,它仅仅涉及到型的描述,不涉及到具体的值。模式的一个具体值称为模式的一个实例。

数据库系统的三级模式结构是指数据库系统是由模式、外模式和内模式三级构成。

(1) 模式

模式也称逻辑模式或概念模式,是数据库中全体数据的逻辑结构和特征的描述,是所有用户的公共数据视图。

模式实际上是数据库数据在逻辑级上的视图。一个数据库只有一个模式。定义模式时不仅要定义数据的逻辑结构,而且要定义数据之间的联系,定义与数据有关的安全性、完整性要求。

(2) 外模式

外模式也称用户模式,它是数据库用户能够看见和使用的局部数据的逻辑结构和特征的描述,是数据库用户的数据视图,是与某一应用有关的数据的逻辑表示。

外模式通常是模式的子集。一个数据库可以有多个外模式。应用程序都是和外模式打交

道。外模式是保证数据库安全性的一个有力措施。每个用户只能看见和访问所对应的外模式中的数据,数据库中的其余数据对他们是不可见的。

(3)内模式

内模式也称存储模式,一个数据库只有一个内模式。它是数据物理结构和存储方式的描述,是数据在数据库内部的表示方式。例如,记录的存储方式是顺序结构存储还是B树结构存储;索引按什么方式组织;数据是否压缩,是否加密;数据的存储记录结构有何规定等。

2. 数据库的二级映像功能与数据独立性

为了能够在内部实现数据库的三个抽象层次的联系和转换,数据库管理系统在这三级模式之间提供了两层映像。

(1)外模式/模式映像

应用于同一个模式可以有任意多个外模式。对于每一个外模式,数据库系统都有一个外模式/模式映像,它定义了该外模式与模式之间的对应关系。

当模式改变时,由数据库管理员对各个外模式/模式映像做相应的改变,可以使外模式保持不变。应用程序是依据数据的外模式编写的,从而应用程序可以不必修改,保证了数据与程序的逻辑独立性。

(2)模式/内模式映像

数据库中只有一个模式,也只有一个内模式,所以模式/内模式映像是惟一的,它定义了数据库的全局逻辑结构与存储结构之间的对应关系。当数据库的存储结构改变时,由数据库管理员对模式/内模式映像做相应改变,可以使模式保持不变,从而应用程序也不必修改。这样保证了数据与程序的物理独立性。

在数据库的三级模式结构中,数据库模式即全局逻辑结构是数据库的中心与关键,它独立于数据库的其他层次。因此,涉及数据库模式结构时应首先确定数据库的逻辑结构。

1.2 数据模型

模型是现实世界特征的模拟和抽象。

计算机不可能直接处理现实世界中的具体事物,人们必须实现把具体事物转换成计算机能够处理的数据。在数据库中,用数据模型这个工具来抽象、表示和处理现实世界中的数据和信息。

数据模型应满足三方面要求:一是能比较真实地模拟现实世界;二是容易为人所理解;三是便于在计算机上实现。数据结构、数据操作和完整性约束是构成数据模型的三要素。数据模型主要包括网状模型、层次模型、关系模型等,它是按计算机系统的观点对数据建模,用于DBMS的实现。

一、层次模型

若用图来表示,层次模型是一棵倒立的树。在数据库中,满足以下条件的数据模型称为层次模型:

- (1)有且仅有一个结点无父结点,这个结点称为根结点;
- (2)其他结点有且仅有一个父结点。

根据层次模型的定义可以看到,这是一个典型的树状结构。结点层次从根开始定义,根为第一层,根的子结点为第二层,根为其子结点的父结点,同一父结点的子结点称为兄弟结点,没有子结点的结点称为叶结点。

层次模型对具有一对多层次关系的描述非常自然、直观、容易理解,这是层次数据库的突

出优点。

1. 层次数据模型的数据结构

在层次模型中,每个结点表示一个记录类型,记录(类型)之间的联系用结点之间的连线(有向边)表示,这种联系是父子之间的一对多的联系。层次数据库系统只能处理一对多的实体联系。

层次模型的一个基本特点是:任何一个给定的记录值只有按其路径查看时,才能显示出它的全部意义,没有一个子女记录值能够脱离双亲记录值而独立存在。

2. 多对多联系在层次模型中的表示

层次数据模型只能直接表示一对多(包括一对一)的联系,需要表示多对多联系时,必须首先将其分解成一对多联系。分解方法有两种:冗余结点法和虚拟结点法。

冗余结点法的优点是结构清晰,允许结点改变存储位置,缺点是需要额外占用存储空间,有潜在的不一致性;虚拟结点法的优点是减少存储空间的浪费,避免产生潜在的 inconsistency,缺点是结点改变存储位置可能引起虚拟结点中指针的修改。

3. 层次模型的数据操纵与完整性约束

层次模型的数据结构将产生如下的约束:

(1)进行插入操作时,如果没有相应的双亲结点值就不能插入子女结点值。例如,如果新调入一名教师,但尚未分配到某个教研室,这时就不能将新教师插入到数据库中。这样对于一些不明确父结点值的结点无法完成插入操作,对现实应用有一定限制。

(2)进行删除操作时,如果删除双亲结点值,则相应的子女结点值也被同时删除。现实应用中,有时我们删除父结点,但并不希望其所有子结点都被删除。例如,撤销了一个系,成立一个新的系,但原来教师的记录并不一定删除。

(3)进行修改操作时,应修改所有相应记录,以保证数据的一致性。

4. 层次数据模型的存储结构

层次数据模型的存储结构可采用邻接法或链接法。

5. 层次模型的优缺点

(1)层次模型的优点

层次数据模型本身比较简单。对于实体间联系是固定的且预先定义好的应用系统,若采用层次模型来实现,其性能优于关系模型,不低于网状模型。另外,层次数据模型提供了良好的完整性支持。

(2)层次模型的缺点

现实世界中很多联系是非层次的,如多对多联系、一个结点具有多个双亲等,层次模型表示这类联系的方法很笨拙,只能通过引入冗余数据(易产生不一致性)或创建非自然的数据组织(引入虚拟结点)来解决;对插入和删除操作的限制比较多;查询子女结点必须通过双亲结点。

二、网状模型

在现实世界中,事物之间的联系更多的是非层次关系的,用层次模型表示非树型结构是很不直接的,网状模型则可以克服这一弊病。在数据库中,满足以下两个条件的数据模型称为网状模型:

- (1)允许一个以上的结点无父结点;
- (2)一个结点可以有多个父结点。

从以上定义看出,网状模型构成了比层次结构复杂的网状结构。

1. 网状数据模型的数据结构

层次模型中,子女结点与双亲结点的联系是惟一的,而在网状模型中这种联系可以不惟一。因此,要为每个联系命名,并指出与该联系有关的双亲记录和子女记录。

下面以学生选课为例,讲一下网状数据模型如何来组织数据。

一个学生可以选修若干门课程,某一课程可以被多个学生选修,因此学生与课程之间是多对多的联系。这样的实体联系图不能直接用网状模型来表示,因为网状模型不能直接表示记录之间的多对多的联系。为此,我们可以把联系也作为一个结点来表示,并引入学生选课记录。它由三个数据项组成,即学号、课程号、成绩,表示某个学生选修某一门课程及其成绩。利用学生到选课的一对多联系和课程到选课的一对多联系表示学生和课程之间的多对多的联系。

2. 网状数据模型的操纵与完整性约束

数据定义语言(DDL)中提供了定义数据库完整性的若干概念和语句,主要有:

(1)支持记录关键字(Key)的概念,关键字是惟一标识记录数据项的集合。

(2)保证一个联系中双亲记录和子女记录之间是一对多的联系。

(3)可以支持双亲记录和子女记录之间的某些约束条件。例如,有些子女记录要求双亲记录存在时才能插入,双亲记录删除时也连同删除。

3. 网状数据模型的优缺点

(1)网状数据模型的优点

能够更直接地描述现实世界,如一个结点可以有多个双亲;具有良好的性能和较高的存取效率。

(2)网状数据模型的缺点

结构比较复杂,而且随着应用环境的扩大,数据库的结构会变得越来越复杂,因而不利于最终用户掌握;其DDL和DML语言复杂,用户不容易使用;由于记录之间的联系是通过存取路径实现的,应用程序在访问数据时必须选择合适的存取路径,因此,用户必须了解系统结构的细节,进而加重了编写应用程序的负担。

三、关系模型

在关系模型中,数据的逻辑结构是一张二维表。在数据库中,满足下列条件的二维表称为关系模型:

(1)每一列中的分量是类型相同的数据;

(2)列的顺序可以是任意的;

(3)行的顺序可以是任意的;

(4)表中的分量是不可再分割的最小数据项,即表中不允许有子表;

(5)表中的任意两行不能完全相同。

关系数据库采用关系模型作为数据的组织方式。

关系数据库因其严格的数学理论、使用简单灵活、数据独立性强等特点,而被公认为是最有前途的一种数据库管理系统。它的发展十分迅速,目前已成为占据主导地位的数据库管理系统。自20世纪80年代以来,作为商品推出的数据库管理系统几乎都是关系型的,例如,Oracle, Sybase, Informix, Visual Foxpro等。

1. 关系数据模型的数据结构

关系模型是建立在严格的数学概念基础上的。关系模型中的数据逻辑结构是一张二维表,它由行和列组成;一个关系对应通常说的一张表,表中一列表示实体的一项属性,称为一个字段;表中一行包含了一个实体的全部属性值,称为一个记录。

2. 关系数据模型的优缺点

(1) 关系数据模型的优点

关系模型是建立在严格的数学概念基础上的;关系模型的概念单一,无论实体还是实体间的联系都用关系表示;关系模型的存取路径对用户透明,从而具有更高的数据独立性、更好的安全保密性,也简化了程序员的工作和数据库开发建立的工作。

(2) 关系数据模型的缺点

由于存取路径对用户透明,查询效率往往不如非关系数据模型高。因此为了提高性能,必须对用户的查询请求进行优化,从而增加了开发数据库管理系统的难度。

1.3 关系数据库

一、关系数据库的基本概念

在关系数据库中,经常会提到关系、属性等关系模型中的一些基本概念。为了进一步了解关系数据库,首先给出关系模型中的一些基本概念。

(1) 关系:一个关系就是一张二维表,每个关系有一个关系名。在计算机中,一个关系可以存储为一个文件。在 Visual FoxPro 中,一个关系就是一个表文件。

关系可以有三种类型:基本关系、查询表和视图表。基本表是实际存在的表,它是实际存储数据的逻辑表示。查询表是查询结果对应的表。视图表是由基本表或其他视图表导出的表,是虚表,不对应实际存储的数据。

(2) 属性:二维表中垂直方向的列称为属性,有时也叫做一个字段。

(3) 域:一个属性的取值范围叫做一个域。

(4) 元组:二维表中水平方向的行称为元组,有时也叫做一条记录。

(5) 分量:元组中的一个属性值叫做元组的一个分量。

(6) 关系模式:是对关系的描述,它包括关系名、组成该关系的属性名、属性到域的映像。关系模式可以形式化地表示为: $R(U, D, \text{dom}, F)$,其中 R 为关系名, U 为组成该关系的属性名集合, D 为属性组 U 中属性所来自的域, dom 为属性向域的映像集合, F 为属性间数据的依赖关系集合。

关系模式通常可以简记为: $R(U)$ 或 $R(A_1, A_2, \dots, A_N)$,其中 R 为关系名, $A_1, A_2, \dots, A_N$ 为属性名。而域名及属性向域的映像常常直接说明为属性的类型、长度等。

一个关系模式在某一时刻的内容(称为相应模式的状态)是元组的集合,称为关系。在不至于引起混淆的情况下,往往将关系模式和关系统称为关系。

(7) 关系数据库:采用关系模式作为数据的组织方式的数据库叫做关系数据库。对关系数据库的描述称为关系数据库的型,它包括若干域的定义以及在这些域上定义的若干关系模式。这些关系模式在某一时刻对应的关系的集合称为关系数据库的值。

(8) 码:又称为关键字。二维表中的某个属性,若它的值唯一地标识了一个元组,则称该属性为候选码。若一个关系有多个候选码,则选定其中一个为主码(或称主关键字),这个属性称为主属性。主码的诸属性称为主属性。不包含在任何候选码中的属性称为非码属性。

外码定义:设 F 是基本关系 R 的一个或一组属性,但不是关系 R 的码,如果 F 与基本关系

S 的主码 Ks 相对应,则称 F 是基本关系 R 的外码(Foreign Key),并称基本关系 R 为参照关系,基本关系 S 为被参照关系或目标关系。关系 R 和 S 不一定是不同的关系。

注意:F 不能是关系 R 的主码,Ks 必须是关系 S 的主码。

显然,目标关系 S 的主码 Ks 和参照关系的外码 F 必须定义在同一个(或一组)域上。

需要强调的是:外码并不一定要与相应的主码同名。不过,在实际应用当中,为了便于识别,当外码与相应的主码属于不同的关系时,往往给它们取相同的名字。

二、关系运算

对关系数据库进行查询时,若要找到用户关心的数据,就需要对关系进行一定的关系运算。关系运算有两种:一种是传统的集合运算(并、差、交、广义笛卡尔积算),它不仅涉及关系的水平方向(即二维表的行),而且涉及关系的垂直方向(即二维表的列);另一种是专门的关系运算(选择、投影、连接等),关系运算的操作对象是关系,运算的结果仍为关系。

1. 选择

选择运算即在关系中选择满足某些条件的元组。也就是说,选择运算是在二维表中选择满足指定条件的行。

2. 投影

投影运算是在关系中选择某些(部分)属性列。

3. 连接

它是从两个关系的笛卡尔积中选取属性间满足一定条件的元组。连接条件中的属性称为连接属性,两个关系中的连接属性应该有相同的数据类型,以保证其是可比的。连接运算中有两种最为重要和常用的连接:一种是等值连接,一种是自然连接。

当连接条件中的关系运算符为“=”时,表示等值连接。它是从两个关系的广义笛卡尔积中选取属性值相等的那些元组。

自然连接是一种特殊的等值连接,它要求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。

以上这些关系运算,在关系数据库管理系统中都有相应的操作命令。对关系数据库的实际操作,往往是以上几种操作的综合应用。

三、关系的完整性

关系模型的完整性(Integrity)规则是对关系的某种约束条件。关系模型中可以有三类完整性约束:实体完整性、参照完整性和用户定义的完整性。其中,实体完整性和参照完整性是关系模型必须满足的完整性约束条件,被称为关系的两个不变性,应该由关系系统自动支持。

1. 实体完整性

实体完整性是指码的值在关系中必须非空且必须是惟一的。

实体完整性规则规定基本关系的所有主属性都不能取空值,而不仅是主码整体不能取空值。

2. 参照完整性

现实世界中的实体之间往往存在某种联系,在联系模型中实体及实体间的联系都是用关系来描述的。这样就自然存在着关系与关系之间的引用。引用的时候,必须取基本表中已经存在的值。参照完整性规则是定义外码与主码之间引用规则的。

参照完整性是指一个关系中外码的码值必须是相应数据库中其他关系的主码值之一,或为空。例如,选课关系的学号属性(字段)的每一个分量必须是学生关系的学号属性(字段)的

分量之一。

3. 用户定义的完整性

不同的关系数据库系统根据其应用环境的不同,往往还需要一些特殊的约束条件,而用户定义的完整性就是针对某一具体关系数据库的约束条件。

例题精讲

例 1. 已知三个关系 R、S 和 T 如下图,则下列各项正确的是()。

- A. $T = R - S$ B. $T = R \cap S$ C. $T = R \cup S$ $T = \sigma_A = 'a' (R)$

R:

A	B	C
a	b	c
d	a	f
c	b	d

S:

A	B	C
b	g	a
d	a	f

T:

A	B	C
a	b	c
c	d	b

分析: 分别计算各项的运算结果,可以得出 $T = R - S$ 。

答案: A

例 2. 下列各公式中存在约束元组变量的是()。

- A. $\{t | (R(t) \vee S(t))\}$
 B. $\{t | (R(t) \wedge S(t))\}$
 C. $\{t | (\exists u)(S(t) \wedge R(u) \wedge t[3] < u[2])\}$
 D. $\{t | S(t) \wedge t[1] > [2]\}$

分析: 自由元组变量是指在一个公式中未用存在量词或全称量词符号定义的元组变量。约束元组变量则是用存在量词或全称量词符号定义的元组变量。观察各备选答案,只有 C 中的元组变量 u 为约束元组变量。

答案: C

例 3. 假设 E_1 和 E_2 为关系代数表达式, F 为连接条件,则下列式子不成立的是()。

- A. $\sigma_F(E_1 \times E_2) \equiv \sigma_{F_1}(E_1) \times E_2$ B. $E_1 \underset{F}{\bowtie} E_2 \equiv E_2 \underset{F}{\bowtie} E_1$
 C. $E_1 \underset{F}{\bowtie} E_2 \equiv E_2 \underset{F}{\bowtie} E_1$ D. $E_1 \times E_2 \equiv E_2 \times E_1$

分析: 对于选择项 A,应有一个限制条件,即此处 F 只涉及到 E_1 中的属性。若 F 形为 $F_1 \wedge F_2$,且 F_1 只涉及到 E_1 的属性, F_2 只涉及到 E_2 的属性,则为 $\sigma_F(E_1 \times E_2) \equiv \sigma_{F_1}(E_1) \times \sigma_{F_2}(E_2)$ 。其他三个选择项没有限制条件。

答案: A

例 4. 下列命题中错误的是()。

- A. 关系中每一个属性对应一个值域 B. 关系中不同的属性可对应同一值域
 C. 对应于同一值域的属性为不同的属性 D. $DOM(A)$ 表示属性 A 的取值范围

分析: 关系中每一个属性都有一个取值范围,称为属性的值域。每一个属性对应一个值域,不同的属性可对应同一个值域。对应同一值域的也可能是同一属性。属性的值域用 $DOM(A)$ 来表示,此处 A 为属性名称。

答案: C

例 5. 已知关系 S 如下表,其中 S#为主键,则下列说法中正确的是()。

- ☒ A. 违反了关系模型的实体完整性规则
☐ B. 违反了关系模型的参照完整性规则
☐ C. 违反了关系模型的用户定义的完整性规则
☐ D. 没有违反关系模型的任何一类完整性规则

S#	SN	AGE	SEX
S1	Wang	23	M
	Liu	20	F
S2	Hou	23	M

分析: 关系模式的实体完整性规则要求关系中元组在组成主键的属性上不能有空值,参照完整性规则要求不引用不存在的实体,用户定义的完整性规则是针对某一具体数据的约束条件,由应用环境决定,它反映某一具体应用涉及的数据必须满足的语义要求。观察关系 S,可发现第二个元组在主键 S#上的值为空值,违反了实体完整性规则。

答案: A

例 6. () 可以看成是现实世界到机器世界的一个过渡的中间层次。

- ☒ A. 概念模型 ☐ B. 逻辑模型 ☐ C. 结构模型 ☐ D. 物理模型

分析: 在进行数据库设计时,通常是将现实世界中的客观对象首先抽象为不依赖于任何具体机器的信息结构,即概念模型,然后再把概念模型转换成具体机器上 DBMS 支持的数据模型。故概念模型可以看成是现实世界到机器世界的一个过渡的中间层次。

答案: A

例 7. 软件生存期的()阶段的最终结果是提供一个可作为设计基础的系统规格说明书。

- ☐ A. 规划 ☒ B. 需求分析 ☐ C. 设计 ☐ D. 程序编制

分析: 在软件生存期的需求分析阶段,应认真细致地了解用户对数据的加工要求,确定系统的功能与边界。最终结果是提供一个可作为设计基础的系统规格说明书,包括软硬件环境的需求和一整套完整的数据流程图。

答案: B

例 8. 整个软件生存期中时间最长的是()。

- ☐ A. 需求分析 ☐ B. 设计 ☐ C. 程序编制 ☒ D. 运行维护

分析: 从软件生存期的调试阶段结束一直到该软件停止使用,这中间的阶段即是运行维护阶段,它是整个生存期中时间最长的阶段。

答案: D

例 9. 要想改变关系中属性的排列顺序,应使用关系运算中的()运算。

- ☐ A. 选择 ☒ B. 投影 ☐ C. 连接 ☐ D. 自然连接

分析: 选择是从行的角度进行的运算,所有的字段都在被选择的记录中;投影是从列的角度进行的运算,所得到的字段个数要比原关系少。连接也是横向结合,生成一个新关系。自然连接包括连接,根据字段值对应相等为条件进行的连接。

答案: B

例 10. 下面不包括在用户需求中的是()。