



21 世纪中等职业教育系列教材
中等职业教育系列教材编委会专家审定

数据库应用基础—ACCESS

主编 侯春瑶



北京邮电大学出版社
<http://www.buptpress.com>

中等职业教育系列教材
中等职业教育系列教材编委会专家审定

数据库应用基础—ACCESS

主 编 侯春瑶
编 者 于增革 赵新秋

北京邮电大学出版社

· 北 京 ·

图书在版编目(CIP)数据

数据库应用基础:ACCESS/侯春瑶主编. -北京:北京邮电大学出版社,2008.4

ISBN 978 - 7 - 5635 - 1736 - 7

I. 数... II. 侯... III. 关系数据库—数据库管理系统, Access—专业学校—教材

IV. TP311.138

中国版本图书馆 CIP 数据核字(2008)第 041751 号

书 名 数据库应用基础—ACCESS

主 编 侯春瑶

责任编辑 周 堃 杨晓敏

出版发行 北京邮电大学出版社

社 址 北京市海淀区西土城路 10 号 邮编 100876

经 销 各地新华书店

印 刷 北京市彩虹印刷有限责任公司

开 本 787mm×960mm 1/16

印 张 13.5

字 数 265 千字

版 次 2008 年 5 月第 1 版 2008 年 5 月第 1 次印刷

书 号 ISBN 978 - 7 - 5635 - 1736 - 7

定 价 18.50 元

如有印刷问题请与北京邮电大学出版社联系

E-mail: publish@bupt.edu.cn

电话: (010) 62283578

Http://www.buptpress.com

版权所有

侵权必究

出版说明

数据库应用基础是中等职业学校计算机及应用专业的一门主干课程,其主要任务是使学生掌握数据库的基础知识和基本技能,培养学生利用数据库系统进行数据处理的能力。使学生能使用所学的数据库知识,根据实际问题进行数据库的创建与维护、检索与统计,能开发简单的数据库应用程序,具有计算机信息管理的初步能力。

本书是中等职业教育计算机应用专业国家规划教材,是根据《中等职业学校计算机应用基础教学大纲》编写,供三、四年制各中等职业学校使用的教材。全书共九章,以 Office 2000 系列的关系数据库管理系统 Access 2000 为蓝本,由浅入深地介绍数据库技术的基础知识、关系数据库的基本操作与使用。

第 1~2 章主要介绍了数据库和 Access 的基础知识。

第 3 章主要讲述数据库和表的创建方法与维护。

第 4 章主要介绍了 Access 的一个重要模块:查询的创建与使用方法。

第 5 章主要讲解窗体所涉及到的常用知识。

第 6 章介绍了报表的创建、维护和打印。

第 7~8 章简单介绍了宏、模块和数据访问页的基础知识。

第 9 章主要介绍了数据库的安全保护方法。

本书自始至终使用了实例,内容由浅入深、循序渐进、图文并茂、理论与实际相结合,遵从学习的认识发展规律,将 Access 的各项基本功能和重要特性展现在读者面前。

本书是北京邮电大学为中等职业学校的学生编写的计算机应用基础课程的使用教材,也可作为参加计算机等级考试人员的培训教材,以及其他学习计算机应用基础知识人员的参考用书。

由于编者编写时间仓促、学识有限,书中难免有构思不当或其他谬误之处,敬请广大师生批评指正。

编 者

目 录

第1章 数据库基础知识	1
1.1 数据库技术的产生和发展	1
1.2 数据模型	2
1.2.1 实体描述	2
1.2.2 实体间联系及种类	3
1.2.3 数据模型简介	3
1.3 关系模型	5
1.3.1 关系操作	5
1.3.2 关系模型的完整性	7
1.4 信息、数据与数据处理	8
1.4.1 信息与数据	9
1.4.2 数据处理与数据管理技术	9
1.5 习题	10
第2章 Access 基本概念	13
2.1 Access 简介	13
2.1.1 Access 的发展过程	13
2.1.2 Access 的主要特点和功能	14
2.2 Access 数据库的系统结构	15
2.2.1 基本对象的功能	16
2.2.2 数据库窗口对象的操作	19
2.3 Access 的启动与退出	20
2.3.1 启动 Access	20
2.3.2 退出 Access	22

2.4	数据类型	23
2.4.1	数据类型	23
2.4.2	数据类型说明	24
2.5	Access 的工作方式	25
2.5.1	交互操作方式	25
2.5.2	程序执行方式	26
2.6	习题	26
第3章	数据库和表	28
3.1	创建数据库	28
3.1.1	创建空数据库	28
3.1.2	使用向导建立数据库	30
3.1.3	打开和关闭数据库	37
3.2	创建表	38
3.2.1	建立表结构	39
3.2.2	输入数据	46
3.3	操作表	48
3.3.1	编辑数据表	48
3.3.2	更改数据表外观	50
3.3.3	整表的复制和删除	53
3.3.4	查找数据	54
3.3.5	替换数据	57
3.3.6	排序记录	59
3.4	创建表之间的关系	62
3.5	表的维护	65
3.5.1	打开表与关闭表	65
3.5.2	修改表结构	67
3.6	习题	69

第4章 查询	71
4.1 查询的作用、分类与准则	71
4.1.1 查询的作用和种类	71
4.1.2 建立查询准则	73
4.2 使用向导创建简单查询	74
4.3 使用查询设计器创建查询	76
4.4 建立复杂的查询	81
4.4.1 参数查询	81
4.4.2 计算查询	83
4.4.3 总计查询	84
4.4.4 交叉表查询	87
4.5 操作查询	90
4.5.1 更新查询	91
4.5.2 追加查询	92
4.5.3 删除查询	94
4.5.4 生成表查询	95
4.6 习题	97
第5章 窗体	99
5.1 窗体的使用 and 分类	99
5.2 创建窗体	101
5.2.1 使用自动创建窗体向导创建窗体	101
5.2.2 使用自动窗体功能创建窗体	102
5.2.3 使用窗体向导创建窗体	103
5.2.4 使用设计视图创建窗体	110
5.3 控件的使用	115
5.3.1 工具箱与窗体中的控件	115
5.3.2 控件使用实例	117
5.4 修饰窗体	136

15	5.5	打印窗体	139
15	5.6	习题	140
	第6章	报表	142
15	6.1	报表的基本概念	142
45	6.2	创建报表	145
105	6.2.1	使用向导创建报表	145
18	6.2.2	使用设计视图创建报表	149
18	6.2.3	自动创建报表	151
18	6.2.4	使用标签向导创建报表	152
48	6.3	在报表中应用控件	156
58	6.3.1	在报表中添加控件	156
100	6.3.2	在报表中修改控件	158
10	6.4	对记录进行排序和分组	159
10	6.4.1	在报表中对记录进行分组与排序	159
40	6.4.2	使报表中的每组都另起一行或另起一列	161
10	6.5	在报表中计算和汇总	162
50	6.5.1	创建计算控件	162
100	6.5.2	在报表中计算总计值或平均值	163
100	6.5.3	在报表中计算运行总和	164
101	6.5.4	对报表中的记录进行计数	165
101	6.6	子报表	166
101	6.7	预览和打印报表	170
101	6.8	习题	171
	第7章	宏和模块	173
111	7.1	宏是什么	173
111	7.1.1	宏的概念	173
111	7.1.2	常用宏操作	174
101	7.2	宏的创建、运行和调试	175

7.2.1	宏组的创建	175
7.2.2	宏的运行	177
7.2.3	宏的调试	177
7.3	过程和模块	179
7.3.1	过程	179
7.3.2	模块	179
7.4	创建过程和模块	180
7.5	习题	182
第 8 章	数据访问页	184
8.1	创建数据访问页	184
8.2	编辑数据访问页	188
8.3	网上发布	191
8.3.1	使用 CuteFTP 软件上传网页	191
8.3.2	使用 FrontPage 发布完整的 Web	192
8.4	习题	193
第 9 章	数据库安全	195
9.1	设置数据库打开权限	195
9.2	数据库的加密和解密	199
9.2.1	数据库加密	200
9.2.2	数据库解密	201
9.3	保护数据库	201
9.4	习题	202
参考答案		203

第1章 数据库基础知识

【教学内容与要求】

1. 了解数据库技术的产生与发展；
2. 掌握数据库模型的分类；
3. 理解几种常用的关系运算；
4. 掌握关系模型的完整性规则；
5. 了解信息、数据等概念。

1.1 数据库技术的产生和发展

数据库技术是在 20 世纪 60 年代末兴起的一种数据管理技术,是公认的“计算机和信息科学增长最迅速的领域之一”。在数据库出现之前,计算机用户使用数据文件来存放数据。常见的数据文件格式是一个文件包含若干个记录(record),一个记录包含若干个数据项(Data Item),用户通过对文件的访问实现对记录的存取,通常称支持这种数据管理方式的软件为文件管理系统。

随着社会信息量的迅速增长,计算机处理的数据量不断增加。文件管理系统所采用的一次最多存取一个记录的访问方式以及在不同文件之间缺乏相互联系的结构,越来越不能适应管理大量数据的需要。于是,数据库系统便应运而生。第一个商品化的数据库系统是美国 IBM 公司的 IMS 系统(Information Management System)。与文件系统相比,数据库系统在数据的结构化、数据共享和数据独立性等方面都较前者有很大改进。

经过近 40 年的发展,数据库系统已经历了 3 代:非关系型数据库系统、关系型数据库系统、对象—关系数据库系统。

非关系型数据库系统是第一代数据库系统的总称,包括层次型数据库系统与网状型数据库系统,现在已经逐渐被关系型数据库系统所取代,仅在一些大中型计算机系统中继续使用。

目前,在微型机上广泛使用的是关系型数据库系统(Relational Database Management System, RDBMS)。它采用人们惯常使用的表格作为基本数据结构,通过公共的关键字段来实现不同二维表之间的数据联系。一次查询仅用一条命令或语句就可访问整个“关系”(或二维表),因而查询效率较高。通过多表联合操作,还能对有联系的若干二维表实现“关联”查询。

对象—关系数据库系统(Object-Relational Database System, ORDBS)是随着多媒体应用的扩大而产生的。此种数据库能够存储图形、声音等复杂的对象,并能实现复杂对象的复杂操作。将数据库技术与面向对象技术相结合是数据库技术的研究方向,是构成第三代数

数据库系统的基础。

如今,数据库管理已经成为计算机信息管理的主要方式。

1.2 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。数据模型就是从现实世界到机器世界的一个中间层次。现实世界的事物反映到人的大脑中,人们把这些事物抽象为一种既不依赖于具体的计算机系统又不为某一 DBMS 支持的概念模型,然后再把概念模型转换为计算机上某一 DBMS 支持的数据模型。

1.2.1 实体描述

现实世界中存在各种事物,事物与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。例如,在教学管理系统中有教师、学生和课程,教师为学生授课,学生选修课程取得成绩;在图书馆中有图书和读者,读者借阅图书;在体育竞赛中有参赛队、竞赛项目,代表队中的运动员参加特定项目的比赛等。如果管理的对象较多或者比较特殊,事物之间的联系就可能较为复杂。

1. 实体

客观存在并相互区别的事物称为实体。实体可以是实际的事物,也可以是抽象的事物。例如,学生、课程、读者等都是属于实际的事物;学生选课、借阅图书等都是比较抽象的事物。

2. 实体的属性

描述实体的特性称为属性。例如,学生实体用学号、姓名、性别、出生年份、系、入学时间等属性来描述;图书实体用总编号、分类号、书名、作者、单价等多个属性来描述。

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。同类型的实体的集合,称为实体集。

例如,学生(学号,姓名,性别,出生年份,系,入学时间)就是一个实体型。对于学生来说,全体学生就是一个实体集;(980102,刘力,男,1980,自动控制,1997)就是代表学生名单中的一个具体的学生。在图书实体集中,(098765,TP298,Access 教程,张三,30.50)则代表具体一本书。

在 Access 中,用“表”来存放同一类实体,即实体集。例如,学生表、教师表、成绩表等。Access 的一个“表”包含若干个字段,“表”中的字段就是实体的属性。字段值的集合组成表中的一条记录,代表一个具体的实体,即每一条记录表示一个实体。

1.2.2 实体间联系及种类

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。例如,一个学生可以选修多门课程,同一门课程可以由多名教师讲授。

实体间联系的种类是指一个实体型中可能出现的每一个实体与另一个实体型中多少个实体存在联系。两个实体间的联系可以归结为3种类型:

1. 一对一联系(One-To-One Relationship)

考察学校和校长这两个实体型,如果一个学校只能有一个正校长,一个校长不能同时在其他学校或单位兼任校长。在这种情况下,学校与校长之间存在一对一联系。

在 Access 中,一对一联系表现为主表中的每一条记录只与相关表中的一条记录相关联。例如,人事部门的教师表和财务部门的工资表之间就存在一对一联系。

2. 一对多联系(One-To-Many Relationship)

考察学校中系和学生两个实体型,一个系中可以有多名学生,而一个学生只能在一个系注册学习。系和学生之间存在一对多联系。考察部门和教师之间的联系,一个教师只能在学校的一个部门任职,占用该部门的一个编制,而一个部门可以有多名在编教师。部门与教师之间也是一对多联系。

在 Access 中,一对多联系表现为主表中的每条记录与相关表中的多条记录相关联。即表 A 中的一条记录在表 B 中可以有多条记录与之对应,但表 B 中的一条记录最多只能与表 A 中的一条记录对应。

一对多联系是最普遍的联系,也可以将一对一联系看做是一对多联系的特殊情况。

3. 多对多联系(Many-To-Many Relationship)

考察学校中学生和课程两个实体型,一个学生可以选修多门课程,一门课程有多名学生选修。因此,学生和课程之间存在多对多联系。图书与读者之间也是多对多联系,因为一位读者可以借阅若干本图书,同一本书可以相继被几个读者借阅。

在 Access 中,多对多的联系表现为一个表中的多条记录在相关表中同样可以有多条记录与之对应。即表 A 中的一条记录在表 B 中可以对应多条记录,而表 B 中的一条记录在表 A 中也可对应多条记录。

1.2.3 数据模型简介

为了反映事物本身及事物之间的各种联系,数据库中的数据必须有一定的结构,这种结构用数据模型来表示。数据库不仅管理数据本身,而且要使用数据模型表示出数据之间的联系。可见,数据模型是数据库管理系统用来表示实体及实体间联系的方法。一个具体的数据模型应当正确地反映出数据之间存在的整体逻辑关系。

任何一个数据库管理系统都是基于某种数据模型的。数据库管理系统所支持的传统数据模型分3种:层次模型、网状模型和关系模型。因此,使用支持某种特定数据模型的数据库管理系统开发出来的应用系统,相应地称为层次数据库系统、网状数据库系统和关系数据库系统。

1. 层次数据模型

层次数据模型是数据库系统中最早出现的数据模型,它用树形结构表示各类实体以及实体之间的联系。层次数据模型数据库系统的典型代表是 IBM 公司的 IMS (Information Management Systems) 数据库管理系统,它是一个曾经广泛使用的数据库管理系统。

在数据库中,对满足以下两个条件的数据模型称为层次模型:

(1) 有且仅有一个节点无双亲,这个节点称为“根节点”。

(2) 其他节点有且仅有一个双亲。

若用图来表示,层次模型是一棵倒立的树。节点层次(Level)从根开始定义,根为第一层,根的孩子称为第二层,根称为其孩子的双亲,同一双亲的孩子称为兄弟。如图 1-1 所示,给出了一个系的层次模型。

层次模型对具有一对多的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。

支持层次数据模型的 DBMS 称为层次数据库管理系统,在这种系统中建立的数据库使用层次数据库。层次数据模型不能直接表示出多对多的联系。

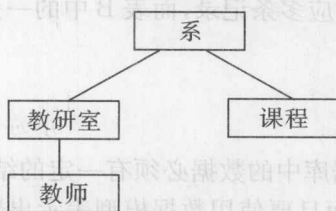
2. 网状数据模型

在数据库中,对满足以下两个条件的数据模型称为网状模型:

(1) 允许一个以上的节点无双亲。

(2) 一个节点可以有多于一个的双亲。

网状数据模型的典型代表是 DBTG 系统,也称 CODASYL 系统,它是 20 世纪 70 年代数据系统语言协会 CODASYL 下属的数据库任务组(Data Base Task Group,简称 DBTG)提出的一个系统方案。若用图表示,网状模型是一个网络。如图 1-2 所示给出了一个抽象的简单的网状模型。



自然界中实体型间的联系更多地是非层次关系,用层次模型表示非树形结构是很不直接的,网状模型则可以克服这一缺点。

3. 关系数据模型

关系数据模型是目前最重要的一种模型。美国 IBM 公司的研究员 E. F. Codd 于 1970 年发表了题为“大型共享系统的关系数据库的关系模型”的论文,文中首次提出了数据库系统的关系模型。20 世纪 80 年代以来,计算机厂商新推出的数据库管理系统(DBMS)几乎都支持关系模型,非关系系统的产品也大都加上了关系接口。

用二维表结构来表示实体以及实体之间联系的模型称为关系数据模型。关系数据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,这种二维表就是关系。

关系模型与层次型、网状型的本质区别在于数据描述的一致性,模型概念单一。在关系型数据库中,每一个关系都是一个二维表,无论实体本身还是实体间的联系均用称为“关系”的二维表来表示,使得描述实体的数据本身能够自然地反映他们之间的联系。而传统的层次和网状模型数据库是使用链接指针来存储和体现联系的。

关系数据库以其完备的理论基础、简单的模型、说明性的查询语言和使用方便等优点得到了最广泛的应用。

1.3 关系模型

关系模型是最重要的一种数据库模型。关系数据模型应由数据结构、关系操作和完整性约束 3 部分组成。这里主要介绍关系操作和完整性约束。

1.3.1 关系操作

关系数据库所使用的关系语言是高度非过程化的,即用户只需要说明“做什么”而不必说明“怎样做”。用户不必请求数据库管理员为其建立特殊的存取路径,存取路径的选择是由 DBMS 自动完成。这也是关系数据库的主要优点之一。

早期,关系操作有两种表示方式:关系代数与关系演算。理论上,关系代数与关系演算被证明是完全等价的。

关系代数通过对关系的运算来表达查询,其操作对象是关系,操作结果亦为关系。关系代数的运算可分为两类,包括传统的集合操作和专门的关系操作两类。这里只介绍专门的关系操作。专门的关系操作包括选择运算、投影运算、连接运算等,这类操作不仅涉及行,而且也涉及列。

(1) 选择运算

选择运算(calculus of selection)是指在关系中选择某些满足条件的元组。例如,要在如图 1-3 所示的“员工”基本信息表中找出女性员工的所有数据,就可以对“员工”基本信息表作选择操作,条件是:性别=“女”。

员工编号	姓名	性别	联系电话
1	王东	男	65623584
2	于敏	女	62532541
3	张弓	男	68541235
4	李可	女	68534741
5	万乐灵	男	69854235
6	紫薇	男	69854127
7	庄杰	女	65478214
8	赵金	女	65478214
9	尹伟	男	65325987
10	何若如	女	65258974

图 1-3 关系模型——“员工”表

(2) 投影运算

投影运算(Calculus Of Projection)是在关系中选择某些属性列。例如,要求根据图 1-3 “员工”基本信息表的信息,制作出一个所有员工的通讯录,就需要找出员工姓名、联系电话这两列所有的数据,此时就可以对员工基本信息表作投影操作,将表数据投影到一个新的表——“员工通讯录”中。

(3) 连接运算

连接运算(Calculus Of Join)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。例如,图 1-4 中(a)、(b)分别为具有 3 个属性列的关系 R 、 S ,对关系 R 和 S 作连接运算,连接条件是 $S.B=R.B$,即是在图 1-4 中(c)所表示的两个关系的笛儿积中选取 $S.B=R.B$ 的元组得到的结果关系如图 1-4(d)。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是可比的,即是同一类的数据类型,如都是数字型或都是字符型。连接条件中的运算符为算术比较运算符,当此运算符取“=”时,为等值连接。图 1-4(d)则是关系 R 和 S 作等值连接后得到的结果关系。若等值连接中,连接属性为相同属性(或属性组),且在结果关系中去掉重复组,则此等值连接为自然连接。如图 1-4(e)即是关系 R 和 S 作自然连接得到的结果,其连接条件为 $S.B=R.B$ 。自然连接是最常用的连接操作。

R 和 S 广义笛卡儿积					
A	B	C	A	B	C
a ₁	b ₁	c ₁	a ₁	b ₂	c ₂
a ₁	b ₁	c ₁	a ₁	b ₃	c ₂
a ₁	b ₁	c ₁	a ₂	b ₂	c ₁
a ₁	b ₂	c ₂	a ₁	b ₂	c ₂
a ₁	b ₂	c ₂	a ₁	b ₃	c ₂
a ₁	b ₂	c ₁	a ₂	b ₂	c ₁
a ₂	b ₂	c ₁	a ₁	b ₂	c ₂
a ₂	b ₂	c ₁	a ₁	b ₃	c ₂
a ₂	b ₂	c ₁	a ₂	b ₂	c ₁

R			S		
A	B	C	A	B	C
a ₁	b ₁	c ₁	a ₁	b ₂	c ₂
a ₁	b ₂	c ₂	a ₁	b ₃	c ₂
a ₂	b ₂	c ₁	a ₂	b ₂	c ₁

R 和 S 等级连接					
A	B	C	A	B	C
a ₁	b ₂	c ₂	a ₁	b ₂	c ₂
a ₁	b ₂	c ₂	a ₂	b ₂	c ₁
a ₂	b ₂	c ₁	a ₁	b ₂	c ₂
a ₂	b ₂	c ₁	a ₂	b ₂	c ₁

R 和 S 的自然连接				
A	B	C	A	C
a ₁	b ₂	c ₂	a ₁	c ₂
a ₁	b ₂	c ₂	a ₂	c ₁
a ₂	b ₂	c ₁	a ₁	c ₂
a ₂	b ₂	c ₁	a ₂	c ₁

图 1-4 连接运算参考图

1.3.2 关系模型的完整性

关系模型中的完整性是指数据库中数据的正确性和一致性。数据的完整性由数据完整性规则来维护,具体包括实体完整性规则和参照完整性规则。当然任何给定数据库中都还要包括与应用有关的完整性规则,这类完整性规则并不是关系模型特有的,这里将其称为用户定义的完整性规则。

1. 实体完整性规则

实体完整性是关系的主属性,即主关键字的组成不能为空,也就是关系的主属性不能是空值(Null)。在关系系统中的一个关系通常对应一个表。在机器上实际存储数据的表称为基本表,除此之外的查询结果表是临时表。视图是虚表,不是实际存储数据的表,而实体完整性是针对基本表的。因此,具体地说,实体完整性是指在实际存储数据的基本表中,主属性不能取空值。

关系对应现实世界中的实体,而现实世界中实体是可以区分的,所以定义实体完整性是

非常必要的。每个实体具有惟一性标识,在关系模型中,由主关键字做惟一标识。若主关键字取空值,则说明这个实体不可识别、不可区分,这显然是错误的,与现实世界应用环境矛盾。因此,不存在这样的无标识实体,从而引入了实体完整性的概念。

前面所定义的实体“员工”表,其主关键字为员工编号,该实体中的任一元组中,员工编号这个属性的值不能为空值,否则就违反了实体完整性要求。一般而言,在 Access 中,主关键字可以是系统自动编号,从而杜绝空值的可能。

2. 参照完整性规则

参照完整性规则是指:如果在基本关系 R_2 的属性中包含与另一基本关系 R_1 的主关键字 K 相对应的属性组 F ,则 F 称为外部关键字。对于 R_2 每个元组在 F 上的取值只允许两种可能:一是空值;二是等于 R_1 中的某个元组的主关键字值。这里 R_1 与 R_2 不一定是不同的关系; K 与 F 是定义在同一域中的。

举一个实例:员工关系(员工号,员工名,工资,部门号)和部门关系(部门号,部门名),其中部门关系中的部门号是主关键字。员工关系中,对每个员工也有部门号一项,表明这个员工是在哪个部门工作的。员工关系中的部门号属性与部门关系中的部门号属性相对应,员工关系中部门号则是外部关键字。

在上面的例子中可以看到:在员工关系中,每个员工的部门号一项,要么取空值,表示这个员工还未分配到任何一个部门;要么取值必须与部门关系中的某个元组的部门号相同,表示这个员工分配到某个部门工作,这就是参照完整性。上例中,如果是员工关系中的某个员工的部门号取值不和部门关系中任何一个元组的部门号值相同,则表示这个职工被分配到不属于所在单位的部门工作,这与实际应用环境是不相符的,显然是错误的。

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

3. 用户定义的完整性规则

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

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

1.4 信息、数据与数据处理

信息是一种资源。从信息论的角度看,任何社会实践活动可以抽象为人流、物流、财流、