

微机汉字数据库

dBASE

设计和应用

李锦峰编著

《陕西电子》编辑部



前 言

数据库技术是开发各种信息系统、预测系统和决策系统的不可缺少的重要工具，是现代信息化社会的重要技术基石之一。

正因为如此，数据库技术在科学技术比较先进的资本主义国家早已得到普遍的重视和广泛的应用，成为近一、二十年来在计算机科学领域中发展最迅速、影响最深远的重要的学科分枝之一，以致人们把70年代称作数据库时代。

近几年由于微型计算机在我国的广泛普及，特别是微型计算机在企业事务管理中的广泛应用，人们对数据库技术的重要性的认识越来越明确，对数据库技术的学习要求也越来越迫切。本书正是为了满足这种广泛的社会需要而编著的。

本书的主要用途是用来作为大专院校、各种类型学习班、培训班、电大、夜大、函大等的数据库课程教材、参考书和自学教材，读者对象主要是大专院校的学生、从事经济信息、企业管理的工程技术人员和各级行政管理干部。他们之中有些同志可能早已具备了相当水平的计算机知识和丰富的管理经验，而另外一些同志则可能仅是计算机刚刚入门者。

如何编写一本能够满足这样广泛需要的好的数据库教材呢？我是从如下几个方面去做的：

1. 面向应用，突出微机：

设想一个单位买了一台、几台、甚至几十台微机，干什么用？在大多数情况下80%左右是用于管理。

那么这里就会提出一系列问题：什么是数据库？怎样设计数据库？设计数据库系统需要解决哪些理论问题、哪些实际问题？以及怎样使用dBASE？

本书就是紧紧地围绕着这些实际问题展开讨论的，集中回答这样一个问题：即如何在dBASE支持下开发设计各种中小型微机事务管理系统？这样就避免了现在某些数据库教材过份侧重于数据库管理系统本身设计，而脱离具体实际应用的缺点。

2. 理论联系实际：

数据库理论设计必须要和数据库系统的具体实现结合起来，这样才能进一步加深对数据库理论的理解。过去一些数据库方面的教材通常是以大型机数据库管理系统为实例，对于许多单位来说，因为缺少这些大型计算机，同学学了数据库理论而得不到实践的机会，因而对数据库理论很难有深切的理解。

3. 既要立足于当前的普及，也要着眼于未来的提高。

可以说当前全国已经出现了一股学习dBASE的热潮，许多人都在学习怎样使用dBASE？学会使用dBASE之后，下一步必然要提出这样一个问题：怎样设计数据库？因为会不会使用dBASE和设计数据库不完全是一回事。

dBASE只是一种系统软件，数据库设计问题。掌握了使用

dBASE, 并不一定能够设计好一个数据库系统, 为了设计数据库系统还必须要进一步学习设计数据库系统的理论和方法, 这里就存在着进一步的提高问题。这是本书将数据库理论和dBASE结合在一起的主要原因。

4. 模块化内容结构。

考虑到大专院校、各种类型学习班、培训班以及其他各种工程技术人员和管理干部的不同应用要求, 本书在内容选材和组织安排上采用了模块式结构。各篇间既具有一定的连贯性、系统性, 同时也各自保持了一定的相对独立性。读者可以根据学习时间的长短以及学习目的要求的不同, 对全书内容进行适合自己所需要的裁剪, 既可以只学其中一篇, 也可以各篇都学。

虽然朝着以上几个方面作了不少的努力, 但是由于时间仓促和水平的限制, 缺点和错误一定是难免的, 希望国内同行专家和广大读者来信批评指正。

由衷地感谢西北电讯工程学院计算机系蔡希尧教授、系主任王厚生付教授、西安交通大学计算机中心付主任王以和教授的热情帮助和支持。

感谢《陕西电子》编辑部张忠智同志的密切合作; 感谢西安市市委党校印刷厂全体师傅们的大力协助和辛勤劳动; 感谢王洁平等同志为本书设计封面、校对; 感谢吕钧林同志为本书打印许多dBASE汉字命令实例而付出的辛勤劳动。

李锦峰

1984年12月

于西安市西北电讯工程学院
计算机系

目 录

第一篇 数据库一般理论

第一章 基本概念

1.1 数据库应用的简单实例	(1)
1.2 实体	(3)
1.2.1 实体集和实体	(3)
1.2.2 记录类型和记录	(4)
1.2.3 实体集—信息—数据	(5)
1.2.4 客观世界的模型化过程	(6)
1.2.5 平面文件	(7)
1.2.6 关键字	(9)
1.3 名词术语	(9)
1.3.1 逻辑术语	(9)
1.3.2 物理术语	(10)
1.4 文件系统	(11)
1.5 数据库	(12)
1.6 数据库对文件系统	(13)

第二章 数据库系统的总体结构

2.1 数据库系统的组成要素	(16)
2.2 数据库系统的简单组成框图	(17)
2.2.1 用户应用程序	(17)
2.2.2 数据库管理系统	(18)
2.2.3 数据	(19)
2.3 数据结构的说明	(19)
2.3.1 数据结构类型	(19)
2.3.2 数据子模式	(20)
2.3.3 数据模式	(21)
2.3.4 数据物理结构	(22)
2.4 数据结构说明语言	(22)
2.5 数据库系统的分层结构	(26)

2.5.1 数据库系统外层	(26)
2.5.2 数据库系统概念层	(27)
2.5.3 数据库系统内层	(28)

第三章 数据模式结构图

3.1 企业数据模式	(29)
3.2 数据模式结构图	(31)
3.3 映射关系	(32)
3.3.1 单值映射和多值映射	(32)
3.3.2 映射关系的表示方法	(33)
3.3.3 其它几种表示方法	(34)
3.3.4 映射关系的几种可能组合	(35)
3.4 如何设计数据模式结构图	(36)
3.4.1 设计数据模式结构图的准则	(36)
3.4.2 数据模式结构图设计举例	(36)
3.5 E-R数据模式结构图	(40)
3.5.1 E-R数据模式结构图的设计	(40)
3.5.2 E-R数据模式结构图的修改	(41)

第四章 数据库的基本功能

4.1 信息查询	(43)
4.1.1 六种基本查询方式	(43)
4.1.2 倒装文件	(45)
4.1.3 组合查询条件	(46)
4.2 信息查询的可预见性	(46)
4.2.1 可预见性的等级划分	(46)
4.2.2 三类数据库用户	(48)
4.2.3 二类数据库系统	(48)
4.3 dBASE 数据操作	(50)

第五章 数据独立性

5.1 独立性概念	(53)
5.2 为什么要数据独立性	(55)
5.3 物理数据独立性和逻辑数据独立性	(56)
5.4 数据独立性的评价	(58)
5.5 提高数据独立性策略手段	(60)

第六章 数据的安全保密性

6.1 安全保密概念	(63)
6.2 用户身份的标识和核实	(64)
6.3 授权矩阵	(65)
6.3.1 用户、数据和操作的密级划分	(65)
6.3.2 授权矩阵	(66)
6.4 安全保密技术	(67)

第七章 并行操作和数据的一致性

7.1 数据的不一致性	(69)
7.2 数据不一致性的类型	(70)
7.3 封锁技术	(71)
7.4 事务处理	(72)
7.5 排它锁	(74)
7.5.1 排它锁定义	(74)
7.5.2 排它锁工作协议PX	(75)
7.5.3 修改的排它锁工作协议PXC	(76)
7.6 共享锁	(77)
7.6.1 共享锁定义	(77)
7.6.2 共享锁工作协议PS	(79)
7.7 修改锁	(80)
7.8 死锁的解除策略	(81)
7.8.1 死锁	(81)
7.8.2 死锁的诊断	(82)
7.8.3 诊断时刻的选择	(82)
7.8.4 死锁的解除	(82)
7.9 死锁的避免策略	(83)

第八章 故障恢复和后援

8.1 故障类型	(85)
8.2 故障恢复的一般策略	(86)
8.3 ‘事务处理’故障	(87)
8.4 系统故障	(88)
8.4.1 撤消和重做	(89)
8.4.2 运行纪录	(89)
8.4.3 检查点	(89)
8.4.4 ‘事务处理’的五种状态	(91)

8.4.5 故障恢复步骤	(92)
8.5 存贮媒质故障	(92)

第九章 数据的完整性

9.1 完整性概念	(94)
9.2 完整性控制规则	(95)
9.3 域的数据完整性控制规则	(96)
9.4 关系数据完整性控制规则	(99)

第十章 数据库管理系统

10.1 数据库管理系统的作用和地位	(102)
10.2 数据库管理系统的软件组成	(103)
10.3 定址和检索技术	(104)
10.4 数据库系统的安全保密子系统	(105)
10.5 数据库语言	(107)
10.5.1 程序语言	(107)
10.5.2 数据操作语言	(108)
10.5.3 数据子模式描述语言	(108)
10.5.4 数据模式描述语言	(108)
10.5.5 数据物理结构说明语言	(109)

第二篇 关系式数据库设计

第十一章 层次式和网络式数据库

11.1 数据的层次结构	(111)
11.1.1 什么是层次结构	(111)
11.1.2 平衡二枝树	(113)
11.1.3 层次文件	(114)
11.1.4 同型结构和异型结构	(115)
11.2 层次式数据结构的语言描述	(116)
11.2.1 CODASYL的语言描述	(116)
11.2.2 IBM的DL/1语言描述	(119)
11.3 数据的网络结构	(127)
11.3.1 什么是网络结构	(127)
11.3.2 几种典型的网络式数据模式结构图	(128)
11.3.3 层次式和网络式数据结构分类	(130)
11.4 网络式数据模式向层次式数据模式的转换	(131)

第十二章 关系式数据库

12.1 关系式数据模式的物理概念	(134)
12.2 关系式数据模式的数学定义	(135)
12.2.1 关系定义	(135)
12.2.2 域和属性	(136)
12.2.3 关键字的补充说明	(137)
12.3 层次式和网络式数据结构的关系式描述	(138)
12.3.1 层次结构的关系式描述	(138)
12.3.2 网络式数据结构的关系式描述	(139)
12.4 关系代数	(139)
12.4.1 传统的关系运算	(139)
12.4.2 传统的集合运算	(144)
12.5 关系式数据模式的优点	(146)

第十三章 规范化技术

13.1 为什么要规范化	(147)
13.2 函数依赖FD	(148)
13.3 向第一范式的转换	(149)
13.3.1 第一范式 (1NF)	(149)
13.3.2 向第一范式的转换	(150)
13.4 完全函数依赖	(151)
13.5 向第二范式的转换	(152)
13.5.1 第二范式 (2NF)	(152)
13.5.2 为什么要向第二范式转换	(152)
13.5.3 向第二范式的转换	(153)
13.6 传递依赖	(155)
13.7 向第三范式的转换	(157)
13.7.1 第三范式 (3NF)	(157)
13.7.2 为什么要向第三范式转换	(157)
13.7.3 向第三范式转换	(157)
13.8 向BCNF范式转换	(158)
13.8.1 BCNF 范式	(158)
13.8.2 BCNF范式 和第三范式关系	(158)
13.8.3 向BCNF 范式转换	(161)
13.9 多值依赖 MVD	(162)
13.10 向第四范式转换	(163)
13.10.1 第四范式 (4NF)	(163)

13.10.2 向第四范式转换	(164)
13.11 联结依赖 (JD)	(164)
13.12 向第五范式转换	(166)
13.12.1 第五范式 (5NF)	(166)
13.12.2 向第五范式转换	(166)
13.13 小结	(167)

第十四章 关系式数据库设计

14.1 数据库设计的一般目的要求	(169)
14.2 设计内容	(171)
14.3 设计方法与步骤	(172)

第三篇 dBASE数据库管理系统

第十五章 dBASE 系统总述

dBASE命令索引

15.1 dBASE	(175)
15.2 dBASE的软件系统组成	(175)
15.3 dBASE的运行环境	(176)
15.4 dBASE的主要技术性能指标	(177)
15.5 dBASE命令系统的功能分类	(177)
15.5.1 功能分类	(177)
15.5.2 dBASE命令系统一览表	(179)
15.6 用户与dBASE打交道方式	(182)
15.6.1 如何进入和退出 dBASE	(182)
15.6.2 命令文件	(183)
15.6.3 命令文件的编辑和运行	(184)
15.7 合法字符集	(185)
15.8 数据	(185)
15.8.1 数据类型	(185)
15.8.2 常数	(185)
15.8.3 变量	(186)
15.9 表达式	(186)
15.9.1 运算符	(186)
15.9.2 表达式	(187)
15.10 文件	(188)
15.11 标识符命名规则	(189)

15.11.1	变量命名规则	(189)
15.11.2	文件命名规则	(190)
15.12	dBASE 命令	(190)
15.12.1	命令结构	(190)
15.12.2	命令书写规则	(191)
15.12.3	符号定义	(191)
15.13	关系操作命令	(192)
15.14	汉字输入方法	(193)

第十六章 文件操作命令

16.1	数据属性说明命令 CREATE	(196)
16.2	文件建立命令	(198)
16.3	文件操作命令	(203)

第十七章 库文件修改命令

17.1	纪录操作命令	(230)
17.2	数据项操作命令	(252)
17.3	文件结构修改命令MODIFY	(267)

第十八章 信息查询和统计报表命令

18.1	信息查询命令	(272)
18.2	信息显示命令	(280)
18.3	统计命令	(284)
18.4	报表格式文件建立命令 REPORT	(292)

第十九章 dBASE 工作模式设置命令

19.1	系统工作模式设置命令	(299)
19.2	全屏幕编辑控制键功能	(306)
19.3	汉字打印字型设置方法	(307)

第二十章 程序控制语言

20.1	赋值命令	(314)
20.2	程序注解命令	(316)
20.3	人机会话命令	(317)
20.3.1	输出命令	(317)
20.3.2	输入命令	(325)
20.4	条件命令	(330)
20.4.1	IF命令	(330)

20.4.2 DO CASE 命令	(332)
20.5 循环命令	(333)
20.6 程序走向控制命令	(335)
20.7 宏替换命令 &	(337)
20.8 函数	(338)
20.8.1 数字型函数	(338)
20.8.2 字符型函数	(340)
20.8.3 逻辑型函数	(343)

第二十一章 dBASE 应用程序举例

参考文献	(390)
------------	-------

第一章 基本概念

〔提要〕 我们知道，存贮在数据库系统中的各种信息数据实际上是对客观世界的一种符合实际需要的抽象的逻辑描述。

如何从实际生活中抽象出描述客观世界的数据库模型？以及在数据库中如何表达这种数据库模型？这是我们学习研究数据库系统所遇到的首要问题。

本书就是从研究如何对数据库系统管理的最基本对象——‘实体集’、‘关系’进行抽象模型化展开讨论的。

本章围绕这一目的着重研究讨论了二个基本问题：

1. ‘实体集’、‘关系’的模型化过程。

较详细地讨论了实体集和实体、记录类型和记录、实体集—信息—数据之间的相互关系。

2. 什么是数据库？和文件系统相比，数据库系统有哪些优点？

另外为了以后的学习方便起见，对本书将要采用的有关数据库名词术语进行了统一的定义说明。

1.1 数据库应用的简单实例

随着微型计算机的日益普及和广泛应用，微型机已经渗透到工农业生产、商业、行政管理、科学研究和工程技术等国民经济的各个领域，成为发展国民经济、进行社会主义四化建设的不可缺少的有力工具。

而在整个计算机应用领域里，数据处理占着很大的比重。据有关专家估计，计算机在各种事务管理方面的应用约占整个计算机应用的80%。

在当今世界，对于许多应用环境来说，数据已经成为一种十分重要的信息资源。

例如医院中对病人情况的管理，铁路车次的调度，旅馆房间和飞机票等的预定系统，仓库货物的管理，工厂中生产的计划管理，财会帐目管理，人事档案的管理，银行户头管理等等，无不需利用大量的数据资源。因此对数据进行有效地管理和处理，已经成为人们普遍关心的研究课题。数据库系统就是人们迄今为止在这方面研究所达到的最新成就。

数据库系统相对来说，是一个比较复杂的数据管理系统。为了对数据库系统展开较深入地分析讨论，下面我们举一、两个实际例子，并从中引出实体、属性和数据等有用的基本物理概念。

1. 物资供应计划管理系统

作为第一个例子是物资供应计划管理系统。这样一个管理系统可能应用于工厂、矿山、商业、学校、研究机关等企事业单位。

在这个管理系统中，我们所关心的可能是下列一些研究对象：

任务课题：任务课题对于不同的单位部门来说可能有不同的含义。任务课题可能是一个建设工程、生产产品、科研项目等。

研究人员：从事某个任务课题研究的工作人员。

单位部门：参加各个任务课题的工作人员所属的单位或部门，以及各个任务课题的归属单位。对一个学校来说，这种单位部门可能是系、教研室、实验工厂等。

物资设备：各个任务课题所需要的零件、物资、设备等。

供应厂家：供应这些物资设备的各个生产厂家。

仓库：用来存放各种物资设备场所。

地理位置：用来说明各个单位部门、供应厂家、仓库等具体地理位置所在。

人们关心的不仅仅是这些被研究的事物对象，更感兴趣的是这些事物对象之间存在着相互关系。因为我们只有知道了它们之间的相互关系，才能从数据库中获得更多的有用信息。譬如说，假若我们知道了任务课题和研究人员之间的‘关系’，数据库就可以回答用户这样一些询问：

某个任务课题有哪些人员参加？

或者

某个同志参加了哪些任务课题？

同样，如果我们知道任务课题和物资设备之间的关系，数据库就可以用来回答这样一些查询：

某个任务课题需要哪些物资设备？

或者

某种物资设备各个任务课题加起来总共需要多少？

从上面这几个查询例子，我们可以很清楚地看到有些查询只涉及到一个研究对象，有些查询却要涉及到二个甚至多个研究对象。例如‘某个任务课题有哪些研究人员参加？’的查询就涉及到二个研究对象：‘任务课题’和‘研究人员’。因此根据实际的应用需要，我们必须在这些研究对象之间建立起某种内在联系。

根据各种实际应用的不同要求，各个研究对象之间的这种相互联系可能是多种多样的。图1.1给出了其中的一种可能联系示意图。

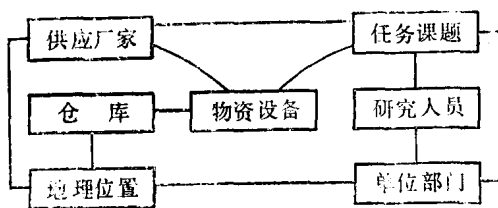


图1.1 物资供应计划管理系统

在这个图中，我们用一条直线来表示某二个研究对象之间存在着的某种内在联系。至于这是一种什么样性质的联系，以及用什么方法来进一步描述各种不同性质的联系，我们将在以后再作更详细的讨论。

图1.1表示的各个研究对象之间的关系，我们可以这样简单地来理解：

任务课题——供应厂家：从任务课题角度来看，它从供应厂家得到各种物资设备供应；而从供应厂家的角度来看，它给各个任务课题供应各种所需物资设备。

当然，如果理解细致一点，我们可以很容易想到，一个任务课题可能从多个厂家得到他们所需要的物资供应；而同一个供应厂家可以给多个任务课题供应物资设备。

其它研究对象之间的相互关系，也可以作相类似的理解。例如：

任务课题——物资设备：从任务课题角度来看，一个任务课题可能需要多种物资设备；而从物资设备角度来看，同一类型物资设备可能要供应给多个任务课题。

任务课题——单位部门：从任务课题角度来看，某个任务课题可能由一个部门负责，也可能由多个部门共同协作；而从单位部门角度来看，某个部门可能拥有多个任务课题。

2: 宾馆管理系统

对于一个宾馆管理系统来说，它所关心的最基本研究对象是旅客、客房以及宾馆可能向旅客提供的各种服务。这些服务可能包括电话服务、汽车接送、餐食饮料、娱乐、以及换洗衣服卫生等。它们之间存在的一种可能关系如图1.2所示：

旅客——客房：每个旅客在某一时刻只能住一个房间；而一个房间可能住一个或多个旅客。

旅客——服务：每个旅客在住旅馆期间可能得到多种服务，而每一种服务可以提供给多个旅客。

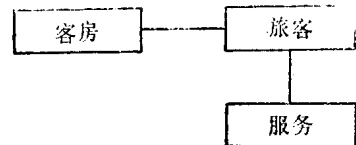


图1.2 宾馆管理系统

上面列举的这二个例子所考虑的研究对象以及各个研究对象之间的相互关系是十分简单粗略的，目的是通过这些例子引出一些基本概念，而不是研究一个具有实际应用价值的数据库设计。事实上，在设计满足某类特定用户应用要求的数据库时，应该选择哪些研究对象？如何确定这些研究对象之间的相互关系？是数据库设计的一个基本问题，也是一个很复杂的问题。

1.2 实体

我们知道，数据库系统的组成结构、工作原理是十分复杂的。为了对数据库系统展开深入细致地分析研究，首先我们应该对数据库研究的最基本对象——‘实体’进行某些必要的说明和讨论。

1.2.1 实体集和实体

我们在上面列举的二个数据库应用的简单实例中，每个数据库所关心的具体研究对

象一般被称作**实体集** (Entity Set)，有时也被称作**实体类型** (Entity type)。

例如在物资供应计划管理系统中的‘任务课题’、‘研究人员’、‘单位部门’、‘物资设备’、‘供应厂家’、‘仓库’、‘地理位置’等，以及在宾馆管理系统中的‘旅客’、‘客房’和‘服务’等都是**实体集**。

实体集是一个具有广泛含义的抽象概念。它既可以代表那些看得见摸得着的有形客观事物，如上面例子中的旅客、客房、物资设备等；它也可以代表那些看不见摸不着的抽象事件和思想概念，如上面例子中的服务、地理位置等。

‘**实体集**’是对某类具有共同特征的客观事物的总称，而作为某类事物中的各个具体个体则被称作‘**实体**’。

‘**实体集**’是一个抽象的笼统的物理概念，而‘**实体**’则是一个具体的物理概念。

例如当我们谈到旅客，我们并没有指明是哪个旅客；而当我们说到客房，我们也没有指明是哪间房客。只有当我们说旅客，苏之明，客房，3楼302号房间，人们才知道旅客是苏之明，客房是指3楼302号房间。

‘**实体集**’和‘**实体**’是数据库系统中的二个基本概念，我们应该很好地理解这两个物理概念的确切含义以及它们之间的区别。

1.2.2 记录类型和记录

每类**实体集**都具有自己的一定特征，为了具体地刻划**实体集**的这种固有特征，人们总是要赋予**实体集**某些属性 (Attributes)。

为了能够细致地反映出**实体集**的各个侧面的物理特征，并且能够区别出各个**实体**的差异性，人们赋予某个**实体集**的往往是一组属性。

例如为了刻划描述**实体集**‘任务课题’的特征，我们可以采用下述一组属性：

任务编号，

任务名，

任务负责人，

任务经费，

任务下达日期，

任务完成日期。

又如为了描述说明参加课题的‘研究人员’的特征情况，我们可以采用下列一组属性：

姓名，

年龄，

性别，

职称，

技术专长，

工作部门。

用来描述某类**实体**的特征的所有属性的有序排列，我们称之为**记录类型** (Record

type)。一个实体集对应着一个记录类型，或者换句话说，一类实体可以用一个记录类型来刻划描述它的基本属性特征。实体集和记录类型之间的这种一一对应关系如图1.3所示：

实 体 集	研究人员					
记录类型						
	姓 名	年 龄	性 别	职 称	技术专长	工作部门

图1.3 实体集和记录类型

一个实体集对应着一个记录类型，那么一个实体也就对应着一个记录实现(Record Occurrence)。或者换句话说，一个实体可以用一个记录实现来描述刻划它的基本属性特征。为了方便起见，记录实现常常被简称为记录。

实体和记录之间的这种一一对应关系如图1.4所示：

实 体	(江 辉)					
记 录	江 辉	45	男	讲 师	数据库	计算机系

图1.4 实体和记录

需要指出的一点是，用来刻划描述实体集特征的所谓‘属性’本身也可能是另一类实体集。例如，在上面这个例子中，作为实体集‘研究人员’的一个属性的‘工作部门’本身就是一个实体集，也就是说‘工作部门’本身也可以用一组属性来描述其物理特征。

1.2.3 实体集——信息——数据

从另一个角度来说，经过人们抽象思维而赋予某类实体集的所谓‘属性’，实际上便是人们所关心的关于该实体集的所有有关信息。

一个实体集中的各个具体实体的所有属性包含的内容便是数据。

实体集、信息和数据之间的相互关系如图1.5所示：

实 体 集	研 究 人 员					
属性(信息)	姓 名	年 龄	性 别	职 称	技术专长	工作部门
数据记录	江 辉	45	男	讲 师	数据库	计算机系

图1.5 实体集、信息和数据

实体集、信息和数据这三者是相互联系的一个有机整体。一组孤立的数据记录往往是没有物理意义的。人们只有把数据记录和其相对应的属性联系起来考虑，这些数据记录才具有一定的明确的物理意义。同样，属性也是如此。人们只有把属性和其描述的实体集联系起来考虑，这些属性也才具有一种明确的物理概念。换句话说，只有当人们把这三者联系起来考虑时才能获得关于某个实体的完整的信息。例如：

实 体 集 零件

属性 (信息)

数 据 记 录

代 号	单 价	数 量	总 值
100	1	10	10

图1.6 实体集、信息和数据之间关系

在数据库设计过程中,选择哪些属性来描述一个实体集的特征,这是每个数据库设计者所面临的一个十分重要的设计课题。但是即使对于同一个实体集,不同的用户从不同的角度去分析考察它时,也可能赋予其完全不同的属性范围和内容。例如同一个‘人’,技术部门关心的主要是其学历和技术专长,人事部门关心的主要是其个人经历和政治表现,而公安部门关心的却是有无犯罪历史纪录、以及外形特征等。

所以说,对于某类实体集的属性选择,主要要根据用户对数据库的用途要求而定。但是就从数据库设计者的角度来说,他所选择的一组属性应该能够唯一地确定该实体集中的每个实体,保证在同一类实体集中不会有二个具有完全相同属性的实体出现,这是设计选择属性项目的一条基本原则。

1.2.4 客观世界的模型化过程

人们用存贮于数据库中的数据记录来描述存在于客观世界的实体,这实际上就是对客观世界的一种抽象的模型化过程。这个模型化过程涉及到客观事物存在的三个不同领域:客观世界、人们的精神世界和计算机世界。同一个客观事物在这三个领域中有三种不同的表示方法。存在于客观世界的是实体集,存在于人们思想脑海中是关于该实体集的有关信息,即由多个属性构成的记录类型,而存在于数据库中的便是对应某个实体的数据记录。客观世界的这种模型化过程如图

1.7所示:

数据库系统所关心的事物是具体存在于客观世界中的客观实体集,正像我们所指出的那样,每个实体集都具有一定的物理特征。

人们对于存在于客观世界的各种客观实体集的抽象思维,便获得能够反映实体集本质的各种有用的信息。这些有价格的信息是用描述实体集的各种属性来表示的。

反映到计算机科学领域里,每个属性就相当于一个数据变量—数据项 (Data—item),

在计算机程序执行过程中,每个数据项是以数据项名进行标识的。而由属性构成的记录类型对应着高级计算机语言中的一种数据变量结构—记录类型 (结构)。在程序运行中,每种记录类型是以文件名进行标识的。文件名和实体集名相对应。文件名和实体集名、数据项和属性只是一种对应关系,而不一定完全相同。他们之间的这种对应关系如图1.8所示:

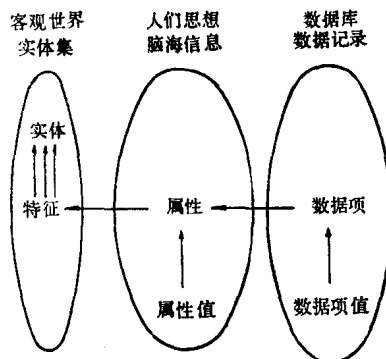


图1.7 客观世界的模型化过程