

电 子 计 算 机 软 件

# 数 据 库 原 理 与 方 法

国 防 科 学 技 术 大 学

郑 若 忠    王 鸿 武 编 著

湖 南 科 学 技 术 出 版 社

电子计算机软件  
数据库原理与方法

国防科学技术大学  
郑若忠 王鸿武 编著  
责任编辑：周翰宗

\*

湖南科学技术出版社出版  
(长沙市展览馆路14号)

湖南省新华书店发行 湖南省新华印刷二厂印刷

\*

1983年7月第1版第1次印刷

开本：787×1092毫米 1/16 印张：19.5 字数：485,000  
印数：1—15,200

统一书号：15204·105 定价：3.00元

# 目 录

## 第一章 数据库导论·····( 1 )

### § 1.1 什么是数据库·····( 1 )

### § 1.2 数据管理技术的发展·····( 3 )

#### § 1.2.1 数据组织管理方法的演变过程 ·····( 4 )

#### § 1.2.2 数据库的设计要求·····( 6 )

### § 1.3 现实世界的描述·····( 7 )

#### § 1.3.1 信息的三个领域·····( 7 )

#### § 1.3.2 实体模型·····( 8 )

#### § 1.3.3 数据模型·····( 10 )

#### § 1.3.4 数据模型的相互转换·····( 16 )

### § 1.4 中文信息处理·····( 17 )

### § 1.5 数据库的逻辑结构·····( 20 )

#### § 1.5.1 数据库的分级结构·····( 20 )

#### § 1.5.2 模式·····( 21 )

#### § 1.5.3 子模式·····( 21 )

### § 1.6 数据语言·····( 23 )

#### § 1.6.1 数据描述语言·····( 23 )

#### § 1.6.2 数据操作语言·····( 24 )

#### § 1.6.3 数据操作语言的非过程化·····( 25 )

#### § 1.6.4 自含系统与宿主系统·····( 25 )

#### § 1.6.5 数据独立性·····( 26 )

### § 1.7 数据库管理系统·····( 27 )

#### § 1.7.1 DBMS的程序组成·····( 27 )

#### § 1.7.2 DBMS与OS的关系·····( 28 )

#### § 1.7.3 数据字典·····( 29 )

#### § 1.7.4 数据库管理员·····( 30 )

#### § 1.7.5 用户访问数据的过程·····( 30 )

### § 1.8 数据库历史与发展趋势·····( 31 )

#### 习题·····( 34 )

## 第二章 数据的物理组织·····( 35 )

### § 2.1 外存基本知识·····( 35 )

#### § 2.1.1 磁盘·····( 35 )

#### § 2.1.2 一个样板系统的外存·····( 37 )

#### § 2.1.3 设备格式化·····( 38 )

### § 2.2 文件组织的基本概念·····( 38 )

#### § 2.2.1 OS的文件管理·····( 39 )

#### § 2.2.2 逻辑记录与物理记录·····( 39 )

#### § 2.2.3 地址与指针·····( 41 )

#### § 2.2.4 分页与系统缓冲区·····( 42 )

#### § 2.2.5 文件组织·····( 43 )

### § 2.3 流水文件·····( 44 )

### § 2.4 顺序文件·····( 45 )

#### § 2.4.1 如何确定关键字值的顺序·····( 46 )

#### § 2.4.2 顺序文件的存储组织·····( 46 )

#### § 2.4.3 顺序文件的查找·····( 47 )

#### \* § 2.4.4 查找次数分析·····( 49 )

#### § 2.4.5 顺序文件的维护·····( 50 )

### § 2.5 索引文件·····( 51 )

#### § 2.5.1 索引顺序文件·····( 51 )

#### § 2.5.2 索引无序文件·····( 52 )

#### § 2.5.3 索引的组织·····( 53 )

#### § 2.5.4 索引文件的查找·····( 54 )

#### \* § 2.5.5 索引块的最佳体积·····( 55 )

### § 2.6 IBM的ISAM和VSAM·····( 57 )

#### § 2.6.1 ISAM文件·····( 58 )

#### § 2.6.2 VSAM文件·····( 61 )

### \* § 2.7 B-树·····( 63 )

#### § 2.7.1 二叉树·····( 63 )

#### § 2.7.2 B-树·····( 63 )

#### § 2.7.3 B<sup>+</sup>树·····( 65 )

#### § 2.7.4 一个B<sup>+</sup>树实例·····( 66 )

#### § 2.7.5 B<sup>+</sup>树文件的性能分析·····( 68 )

#### § 2.7.6 B-树的变种·····( 69 )

### § 2.8 HASH文件·····( 70 )

#### § 2.8.1 处理溢出的方法·····( 71 )

#### § 2.8.2 溢出分析·····( 72 )

#### § 2.8.3 若干Hash算法·····( 74 )

#### § 2.8.4 KAT方法综述·····( 76 )

### § 2.9 部分匹配检索·····( 77 )

|                                        |        |                                        |       |
|----------------------------------------|--------|----------------------------------------|-------|
| § 2.9.1 倒排文件 .....                     | ( 77 ) | § 3.7.2 关系代数表达式的等价代换规则 .....           | (141) |
| § 2.9.2 多重表 .....                      | ( 79 ) | * § 3.7.3 关系代数表达式的优化算法 .....           | (143) |
| § 2.9.3 分部Hash函数 .....                 | ( 81 ) | * § 3.7.4 SEQUEL的一种优化器 .....           | (145) |
| * § 2.9.4 桶号位数的分配 .....                | ( 82 ) | § 3.8 数据的相关性 .....                     | (149) |
| § 2.9.5 多维树 .....                      | ( 84 ) | § 3.8.1 函数相关性 .....                    | (150) |
| * § 2.9.6 多维树的操作分析 .....               | ( 85 ) | § 3.8.2 相关性的逻辑蕴含 .....                 | (151) |
| § 2.9.7 联想处理 .....                     | ( 86 ) | * § 3.8.3 多值相关性 .....                  | (152) |
| § 2.10 其他数据物理组织方法 .....                | ( 89 ) | * § 3.8.4 Armstrong公理体系 .....          | (152) |
| § 2.10.1 数据紧缩技术 .....                  | ( 89 ) | * § 3.8.5 Armstrong 公理体系的完<br>备性 ..... | (155) |
| § 2.10.2 快速响应技术 .....                  | ( 91 ) | * § 3.8.6 函数相关性集合的等价问题 .....           | (157) |
| § 2.10.3 多级存储系统 .....                  | ( 92 ) | * § 3.8.7 函数相关性集合的最小集合 .....           | (159) |
| § 2.10.4 网络 .....                      | ( 93 ) | § 3.9 关系的规范理论 .....                    | (161) |
| 习题 .....                               | ( 93 ) | § 3.9.1 关系模型评价 .....                   | (161) |
| <b>第三章 关系方法</b> .....                  | ( 96 ) | § 3.9.2 关系框架分解 .....                   | (162) |
| § 3.1 关系及其基本术语 .....                   | ( 96 ) | § 3.9.3 关系的1NF和2NF .....               | (167) |
| § 3.2 关系运算 .....                       | ( 98 ) | § 3.9.4 关系的3NF和BCNF .....              | (168) |
| § 3.2.1 关系代数 .....                     | ( 98 ) | * § 3.9.5 关系的4NF和W4NF .....            | (172) |
| § 3.2.2 元组关系演算 .....                   | (103)  | * § 3.9.6 连接相关性和关系的5NF .....           | (173) |
| § 3.2.3 域关系演算 .....                    | (105)  | * § 3.9.7 规范理论综述 .....                 | (175) |
| * § 3.3 关系运算的安全约束 .....                | (106)  | § 3.10 关系方法总结 .....                    | (176) |
| * § 3.4 三类关系运算表达能力的等<br>价性 .....       | (109)  | 习题 .....                               | (179) |
| § 3.5 关系数据库的数据操作语言 .....               | (115)  | <b>第四章 层次方法</b> .....                  | (183) |
| § 3.5.1 基于关系代数的语言ISBL .....            | (115)  | § 4.1 一般概念 .....                       | (183) |
| § 3.5.2 介于关系代数与演算之间的语言<br>SEQUEL ..... | (118)  | § 4.1.1 树 .....                        | (183) |
| § 3.5.3 基于元组演算的语言QUEL .....            | (124)  | § 4.1.2 层次系统的数据模型 .....                | (185) |
| § 3.5.4 基于域演算的语言QBE .....              | (128)  | § 4.1.3 层次顺序与层次路径 .....                | (186) |
| § 3.5.5 关系系统操作语言的比较 .....              | (132)  | § 4.1.4 层次系统的模式与子模式 .....              | (187) |
| § 3.6 关系数据库的模式和子模式 .....               | (134)  | § 4.2 IMS系统的逻辑结构 .....                 | (188) |
| § 3.6.1 源模式、目标模式及其物理映射 .....           | (134)  | § 4.2.1 IMS的逻辑结构 .....                 | (189) |
| § 3.6.2 子模式、目标子模式及其映射 .....            | (137)  | § 4.2.2 IMS的DBD .....                  | (189) |
| § 3.7 询问的优化 .....                      | (140)  | § 4.2.3 IMS的PSB .....                  | (191) |
| § 3.7.1 优化的一般策略 .....                  | (141)  | § 4.3 IMS的存储结构 .....                   | (193) |
|                                        |        | § 4.3.1 HSAM .....                     | (193) |
|                                        |        | § 4.3.2 HISAM .....                    | (194) |

|            |                     |       |       |
|------------|---------------------|-------|-------|
| § 4.3.3    | HIDAM的HDAM          | ..... | (196) |
| § 4.4      | IMS的数据子语言           | ..... | (200) |
| § 4.4.1    | 子语言DL/1             | ..... | (200) |
| § 4.4.2    | IMS的应用程序            | ..... | (204) |
| § 4.4.3    | 应用程序的运行             | ..... | (207) |
| § 4.5      | IMS 存储结构补充          | ..... | (207) |
| § 4.5.1    | 辅数据集组               | ..... | (207) |
| § 4.5.2    | 数据库的存储映射            | ..... | (209) |
| * § 4.5.3  | IMS的辅索引             | ..... | (211) |
| * § 4.6    | IMS的逻辑数据库           | ..... | (214) |
| § 4.6.1    | 逻辑父与逻辑子             | ..... | (214) |
| § 4.6.2    | LDB的定义              | ..... | (215) |
| § 4.6.3    | 双向逻辑关系              | ..... | (217) |
| § 4.6.4    | 建立LDB的规则            | ..... | (220) |
| 习题         |                     | ..... | (221) |
| <b>第五章</b> | <b>网络方法</b>         | ..... | (223) |
| § 5.1      | CODASYL系统逻辑结构       | ..... | (223) |
| § 5.2      | CODASYL的数据模型        | ..... | (224) |
| § 5.2.1    | 网络                  | ..... | (224) |
| § 5.2.2    | 记录类型                | ..... | (225) |
| § 5.2.3    | 络类型                 | ..... | (226) |
| § 5.2.4    | 络事件                 | ..... | (227) |
| § 5.2.5    | 事物联系的CODASYL表示法     | ..... | (228) |
| § 5.3      | 记录类型描述及其存储映射        | ..... | (230) |
| § 5.3.1    | CODASYL句法使用的符号      | ..... | (230) |
| § 5.3.2    | 记录类型的描述             | ..... | (231) |
| § 5.3.3    | 记录类型的存储映射           | ..... | (232) |
| § 5.3.4    | 记录类型举例              | ..... | (234) |
| § 5.4      | 络类型描述及其存储映射         | ..... | (235) |
| § 5.4.1    | 络模型                 | ..... | (235) |
| § 5.4.2    | 络次序                 | ..... | (236) |
| § 5.4.3    | 从记录类型性质的描述          | ..... | (238) |
| § 5.4.4    | 络选择                 | ..... | (239) |
| § 5.4.5    | 络类型描述举例             | ..... | (240) |
| § 5.4.6    | ON条款                | ..... | (242) |
| § 5.5      | CODASYL数据语言         | ..... | (242) |
| § 5.5.1    | 模式数据描述语言            | ..... | (242) |
| § 5.5.2    | 子模式数据描述语言           | ..... | (247) |
| § 5.5.3    | 数据操作语言              | ..... | (249) |
| § 5.5.4    | 模式、子模式和应用程序举例       | ..... | (253) |
| * § 5.6    | 一个CODASYL系统——IDMS   | ..... | (256) |
| 系统         |                     | ..... | (256) |
| § 5.6.1    | IDMS的数据表示           | ..... | (256) |
| § 5.6.2    | IDMS的存储结构           | ..... | (257) |
| § 5.6.3    | IDMS的存储映射           | ..... | (258) |
| § 5.6.4    | IDMS的处理顺序           | ..... | (259) |
| § 5.6.5    | 批处理与联机工作            | ..... | (260) |
| § 5.6.6    | IDMS系统的维护           | ..... | (261) |
| 习题         |                     | ..... | (262) |
| <b>第六章</b> | <b>数据库保护——完整与安全</b> | ..... | (264) |
| § 6.1      | 保护的基本内容             | ..... | (264) |
| § 6.2      | 完整保护                | ..... | (266) |
| § 6.2.1    | 语义完整                | ..... | (266) |
| * § 6.2.2  | 典型系统的完整约束条件         | ..... | (267) |
| § 6.2.3    | 事务与延迟约束             | ..... | (268) |
| § 6.2.4    | 并发控制                | ..... | (269) |
| * § 6.2.5  | 典型系统的并发控制           | ..... | (271) |
| § 6.2.6    | 数据库的恢复              | ..... | (271) |
| * § 6.2.7  | IMS系统的恢复            | ..... | (273) |
| § 6.2.8    | 恢复的一般原则             | ..... | (274) |
| § 6.3      | 安全保护                | ..... | (274) |
| § 6.3.1    | 安全与完整的关系            | ..... | (274) |
| § 6.3.2    | 安全保护的一般策略           | ..... | (275) |
| § 6.3.3    | 用户鉴别机构              | ..... | (276) |
| § 6.4      | 访问控制                | ..... | (277) |
| § 6.4.1    | 访问规则                | ..... | (277) |
| § 6.4.2    | 控制机构                | ..... | (279) |
| * § 6.4.3  | 访问请求的生效过程           | ..... | (281) |
| * § 6.4.4  | 数据库安全检查纵览           | ..... | (281) |
| * § 6.5    | 几个实际系统的保护           | ..... | (282) |
| § 6.5.1    | QBE系统               | ..... | (282) |
| § 6.5.2    | System R系统          | ..... | (283) |
| § 6.5.3    | IMS系统               | ..... | (285) |

|                         |       |                           |       |
|-------------------------|-------|---------------------------|-------|
| § 6.5.4 CODASYL系统 ..... | (285) | § 6.6.4 数据库密码的特点 .....    | (294) |
| § 6.6 密码 .....          | (286) | * § 6.6.5 美国NBS密码算法 ..... | (295) |
| § 6.6.1 制密方法 .....      | (287) | * § 6.6.6 密码的破译 .....     | (299) |
| § 6.6.2 密钥的产生与传递 .....  | (292) | * § 6.6.7 反破译的对策 .....    | (300) |
| § 6.6.3 硬件密码 .....      | (293) | 习题 .....                  | (301) |
| 参考文献 .....              |       |                           | (303) |

# 第一章 数据库导论

## § 1.1 什么是数据库

**数据库 (database)** 是计算机软件的一个重要分支, 是近十几年迅速发展起来的一门新兴学科。目前虽有不少人对数据库的设计原则和方法进行总结与探讨, 使之通用化、标准化和理论化, 但总的来说, 它还处于从工程实践向理论过渡的阶段, 它的概念、原理和方法还在继续发展和变化; 另外数据库是一个很复杂的系统, 涉及面很广, 难以用简练的语言准确地概括其全部特征。因此, 给它下一个确切的定义是比较困难的。尽管如此, 在有关数据库的文献中, 人们还是根据数据库的现状和自己的认识给数据库以定义或解释。如:

(1) DBTG的定义: 数据库由一个特定模式(schema)控制的所有记录、结(set)和区域组成。如果有多个数据库, 则每一数据库必须有自己的模式。并假定不同数据库的内容是彼此无关的。

(2) C. J. Date[3]则从三个方面描述数据库: ①存储在磁鼓、磁盘或其他外存介质上的数据集合——这是指数据库自身; ②存在以这种数据为背景运行的若干个批应用程序, 对其施行通常操作(检索、修改、插入和删除, 一般以检索为最通常和最重要), 另外可能有一些联机用户从远程终端与数据库相互作用; ③数据库是集成的, 即包含许多用户的数据, 每个用户只享用其中的小部分, 且不同用户使用的部分以多种方式重叠——即单独的数据片能够被许多不同用户共享(图1.1)。

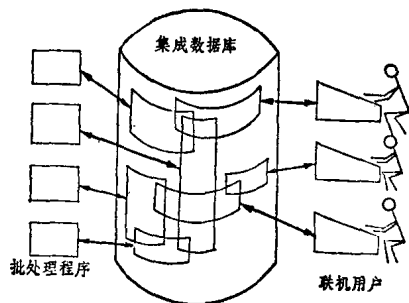


图1.1 数据库系统简化视图

(3) J. Martin[13]的定义: 可以把数据库定义为存储在一起的相关数据的集合, 这些数据无有害的或不必要的冗余, 为多种应用服务; 数据的存储独立于使用它的程序; 对数据库插入新数据, 修改和检索原有数据均能按一种公用的和可控制的方法进行; 数据被结构化, 为今后的应用研究提供基础。当某个系统中存在结构上完全分开的若干数据库时, 则说该系统包含一个数据库集合。

我们认为, 前两种定义的是数据库系统的组成, 很少涉及数据库的特点, 第三种定义比较全面。但这些定义和解释还不能使人们具体而深入地理解数据库。为此, 我们从如下几个方面作进一步阐述。

首先指出, 数据库保存的是属于企业和事业部门、团体和个人的有关数据。例如制造公司的生产管理和产品供销数据、银行帐目数据、医院的患者病例数据、学校的教学管理数据、政府部门的管理、统计和计划数据等。特别应该强调的是, 设计数据库保存这些数据的目的, 不仅仅是为了扩展人们的记忆, 而主要是帮助人们去控制与之相关的事物。因此, 数据库往往不是孤立存在的, 通常是一个更大的信息控制系统的一部分, 两者的关系如图 1.2 所示。

该图表示，人们从观测客观事物中得到大量信息，对这些信息进行记录、整理和归类（总称规范），然后将规范信息数据化并送入数据库中保存起来，其中一部分信息可能直接送入控制决策机构。另一方面，控制决策机构（它既可由一些人组成，也可是一个自动控制系统）向数据库发出询问，并利用数据库响应后提供的信息（以及其他有关信息）作出决策，再行控制客观事物。譬如，就一个教学信息控制系统而言，客观事物可指学生、课程、学习情况等，决策控制机构指教学领导机关，策略是提高教学质量措施。当数据库的管理人员把观测客观事物（学生、课程、学习情况）得到的信息规范化、数据化并送入数据库后，教学领导机关即可通过询问数据库得到学生各门课程的学习情况，根据这些情况，参考其他学校的教学经验，提出改进教学的措施（策略），并在教学中付诸实行（控制客观事物）。

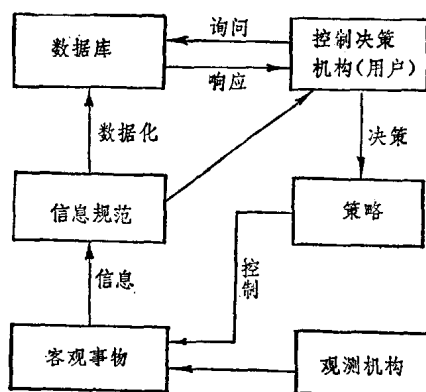


图1.2 信息控制系统图示

对于不熟悉数据库的人来说，把它比作图书馆可能更易于理解。大家知道，图书馆是存储和负责借阅图书的部门；而数据库则是存储数据并负责用户访问数据的机构。正象图书馆不能简单地与书库等同起来一样，我们也不能把数据库仅仅理解为存储数据的集合，而应视为一个系统，即数据库系统。就图书馆而言，如果把书籍胡乱地堆放在书库中，几乎无法从数以百万册计的浩瀚书海中查找出读者要借的一本书来。因此，没有一套完整的书卡作为图书馆藏书的模型，查借书籍就很困难，管理员也很难掌握藏书全貌。如果不知道书卡与书架的对应关系，管理员也难以按借书单找到该书的存放位置。一个图书馆要想很好地为读者服务，必须完成以下工作：

(1) 建立完善的书卡。书卡的内容和格式常包括：书号(分类号)、书名、作者名、出版社名、出版时间、内容摘要和其他细节。有时为了方便读者，也按不同分类编排书卡，如以书名、作者名或其它为索引进行编排。

(2) 图书应有组织地存放在书库中。图书馆藏书数量很大，书库中房间、书架很多，需要按照一定的顺序和规则（物理结构）分放图书，并列出具类书籍存放的对应关系表，使管理人员能按此表快速查找。

(3) 规定借阅权限。不同类型读者的借阅权限不同。如善本书只供特定的人借阅；机密图书只供有特权的人借阅；某些书只供读者在馆内翻阅等。

(4) 建立周密的借阅管理制度。规定图书的借还手续。读者借书要先出示借书证，图书管理员验明读者身份和借阅权限后，根据读者填写的借书单(访问请求)，按照书籍与书架的对应关系表，到书库中查找所借图书交与读者(响应)，并作某些登记(日志)；还书时管理员要按书号把交还的图书送回原来存放的书架位置上；如果还书较多，管理员又要立即“响应”其他读者借书，那么，他也可以把还书按归还次序放到一个“当日暂存书架”上，待一天末了有空闲时，再清理“当日暂存书架”上的还书并送回到书库中存放它们的原来位置上。

对数据库来说，也要完成类似于图书馆的上述工作。(1) 数据库要建立数据模型，使用户可以根据数据模型访问数据库中的数据(如检索、插入、删除和修改)，而不必关心数据在数据库中的物理存储位置，就象读者可以按书卡填写借书单，而不用顾及书籍存放在书库的



什么位置一样。当然数据模型也要象书卡那样能反映各种数据之间的内在联系。(2) 数据也应有组织地存放在存储设备上, 并建立**数据模型到物理存储位置**的对应表(这种对应称为**映射**), 使数据库管理系统能够按照用户的访问请求, 找到被访问数据的存储位置。建立数据模型和设计数据的物理存储(组织)方法, 其目的是使用户对数据的应用与数据的存放位置和存储结构无关, 后者的变动不影响前者(正象改变图书的存放位置不影响读者按书卡借书一样), 这一点也称为**数据独立性**, 它是数据库的重要特征之一。现用表 1.1 来归纳数据库与图书馆两者的类似。

至此, 我们不难理解: **数据库系统是**实现有组织地、动态地存储大量关联数据, 方便多用户访问的计算机软、硬资源组成的系统; 它与文件系统的重要区别是数据的充分共享、交叉访问、与应用(程序)的高度独立性。当数据库一词仅表示联机存储的关联数据时, 数据库系统则包括数据库和数据库管理软件两部分。

表1.1 数据库与图书馆类比

| 序号 | 数 据 库                                 | 图 书 馆                           |
|----|---------------------------------------|---------------------------------|
| 1  | 数 据                                   | 图 书                             |
| 2  | 外 存                                   | 书 库                             |
| 3  | 用 户                                   | 读 者                             |
| 4  | 用户标识                                  | 借 书 证                           |
| 5  | 数据模型                                  | 书卡格式                            |
| 6  | 数据库管理系统                               | 图书馆管理员                          |
| 7  | 数据的物理组织方法                             | 图书的物理存放办法                       |
| 8  | 用户对数据库的操作<br>(使用计算机语言<br>检索、插入、删除、修改) | 读者对图书馆的访问<br>(用普通语言)<br>(借书、还书) |
| 9  | 第8项独立于第7项                             | 第8项独立于第7项                       |

## § 1.2 数据管理技术的发展

人类活动的整个历史, 离不开对信息和数据的收集、保存、利用和处理。初始, 人类仅能借用语言、绘画和火光传递信息, 通过结绳和刻画记录信息, 以供人们利用和处理。当发明了文字、纸张和印刷术后, 就以纸张为介质, 通过汇编字典、登记帐目与构造各种类型的表、簿、册等方法来收集和保存信息。待电子技术出现后, 人们才利用磁性材料作介质来保存和收集信息, 使得利用和加工信息进入到更高级的阶段。

自本世纪六十年代以来, 社会生产力高速发展, 新技术层出不穷, 信息量急剧膨胀, 使整个人类社会成为信息化的社会, 人们对信息和数据的利用和处理已进入自动化、网络化和社会化的阶段。例如, 查找情报资料、档案; 处理银行帐目、资金往来; 进行人口统计分析; 办理航空订票、旅馆房间登记; 管理交通运输、生产科研; 订计划、作决策等。这些任务既需利用大量数据, 又要求快速处理及时得出结果, 如果仅由人工靠翻阅文件和查找记录是难以完成的, 甚至是不可能的, 所以迫切需要借助于电子计算机的高速度和大容量。

电子计算机问世以后, 人们便将十进制数变为二进制数以适应计算机的存储、运算要求。随后又将文字(字母、符号、文字编码)表示成位串形式, 因而人们可以把文字、数字、符号组成的文件, 以及曲线、几何图形、图片和声音等都存贮在计算机中。时至今日, 几乎除暗示外, 所有信息均可表成计算机能识别的字符串或位串, 予以存储、传送和运算, 为迅速处理大量数据提供了可能。而计算机硬件技术的发展又使这种可能变成现实。1956年的第一台磁盘(RAMAC磁盘)容量仅为5M字节, 而1965年磁盘最大存储容量为 $3 \times 10^7$ 字节, 1978年扩大到 $6 \times 10^8$ 字节, 磁盘是一种直接访问存储设备, 为数据库提供了良好的物质基础, 大大促进了数据库技术的进步。表1.2是IBM磁盘技术的进展情况[34]。

表1.2

IBM 磁 盘 进 展

| 型 号                                 | 2314  | 3330  | 3330-11 | 3340  | 3350  | 3370  | 3375   | 3380   |
|-------------------------------------|-------|-------|---------|-------|-------|-------|--------|--------|
| 生 产 时 间 (年)                         | 1966  | 1971  | 1974    | 1973  | 1976  | 1979  | 1981   | 1981   |
| 容 量 (MB/轴)                          | 29.2  | 100   | 200     | 35/70 | 317.5 | 571   | 819.7  | 1260   |
| 平均访问时间 (ms)                         | 60    | 30    | 30      | 25    | 25    | 20    | 19     | 16     |
| 转 速 (rpm)                           | 2400  | 3600  | 3600    | 3000  | 3600  | 2970  | 2970   | 3600   |
| 数据传送速度 (MB/秒)                       | 0.312 | 0.806 | 0.806   | 0.885 | 1.198 | 1.859 | 1.859  | 3.0    |
| 位 密 度 (BPI)                         | 2200  | 4040  | 4040    | 5636  | 6425  | 12100 | 12100* | 15240* |
| 道 密 度 (TPI)                         | 100   | 192   | 370     | 300   | 478   | 635   | 800*   | 760*   |
| 面 记 录 密 度 (千位/6.25cm <sup>2</sup> ) | 220   | 776   | 1495    | 1690  | 3071  | 7684  | 9680*  | 11582* |
| 磁头缝隙长度 (μm)                         | 2.54  | 2.5   | 2.5     | 1.4   | 1.4   | 0.8   | —      | —      |
| 磁头浮动高度 (μm)                         | 2.29  | 1.27  | 0.9     | 0.51  | 0.48  | 0.33  | —      | 0.3*   |
| 磁 道 宽 度 (μm)                        | 178   | 109   | 52      | 70    | 37    | 33    | —      | 23*    |
| 每单位字节的相对价格 (1981年)                  | —     | 7.5   | 5.2     | 10.5  | 2.67  | 1.26  | 1.2    | 1      |

• 原杂志推定

### § 1.2.1 数据组织管理方法的演变过程

为深入理解数据库的内部结构,讨论一下数据组织管理方法的几个演变阶段是有帮助的。在讨论中我们沿用操作系统中讲过的文件、记录、数据项,以及文件物理组织的顺序式、链接式和索引式等概念。众所周知,数据库系统是为满足日益发展的数据处理的需要,在文件系统的基础上发展起来的一种理想的数据管理技术。六十年代早期的文件系统是数据管理方法的雏形,其文件在外存的物理结构与用户观点的逻辑结构完全一致,用户的数据文件主要存储在磁带上,它的组织方式是顺序式的,这时的数据管理软件属于操作系统的一部分,其主要功能是完成I/O设备的输入/输出操作,如图1.3所示。显然,这种组织形式只能应付批处理,不适应于实时访问。由于让用户各自建立文件,用户要花费很大精力为文件设计数据的物理安排细节与编制应用程序,因而文件不宜共享,数据冗余度很高,当文件的物理结构发生变化或更换外存设备时,就得修改或重编应用程序,用户感到负担沉重,促使人们探索新的数据管理方法。

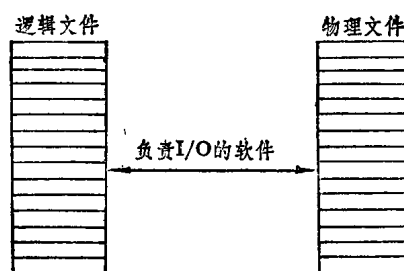


图1.3 初期文件系统特征

大约到六十年代中后期,直接访问设备——磁鼓、磁盘的性能改善使数据组织发生了如下变化:除磁带外,磁鼓、尤其是磁盘成为联机的主要外存设备;文件的物理结构与逻辑结构之间已有所区别(其关系如图1.4所示),在文件的物理结构中增加了链接和索引等形式,因而对文件中的记录可顺序地和随机地访问;数据管理软件(仍属操作系统的一部分)提供从逻辑文件到物理文件的“访问方法”是这一时期数据管理的主要特征;对系统而言,组成文件的最小单位是记录,记录之间仅存在顺序关系而无其他结构;在管理软件中还增加了安全、保密检查机构;部分系统允许用户之间以文件为单位共享数据,但未能实现以记录和数

据项为单位的数据共享；用户仍以文件标识（文件名）与系统交往，也允许以文件中的记录标识访问数据。显然，它不但适应于批处理，也可用于实时联机任务；系统更换外设也无需用户修改应用程序；可以实现以文件为单位的数据共享等。但这种文件系统仍不够理想，未

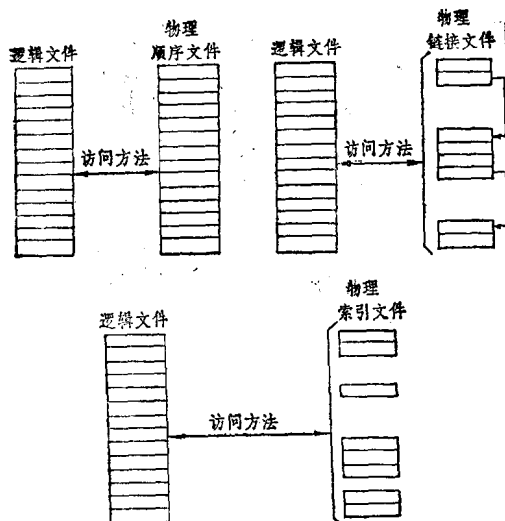


图1.4 带有访问方法的数据组织

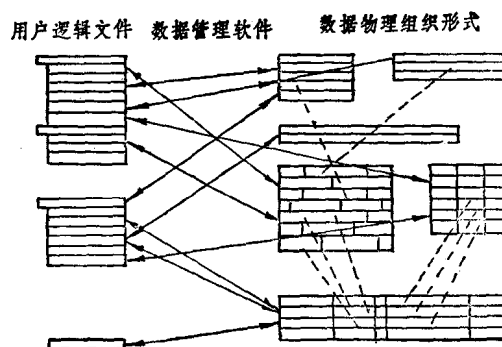


图1.5 初期的数据库系统

能体现用户观点下的数据逻辑结构较大地独立于数据在外存的物理结构。因此，数据物理存储的改变，仍然需要修改用户的应用程序。再者以文件而不以记录或数据项为单位共享数据，必然导致数据的大量冗余，用户也不能以记录或数据项为单位访问数据。同时，文件系统亦难于增删新旧数据以适应新的应用要求。这些亟待解决的问题，促使人们研究一种新的数据管理技术。六十年代末终于出现了数据库系统，初期的数据库系统通常具有如下的特点：①对用户观点的数据进行严格细致的描述，使得文件、记录、数据项等数据单位之间的联系清晰，结构简单；②允许用户以记录或数据项作单位进行访问，也允许多关键字检索和文件之间的交叉访问；③数据的物理存储可以很复杂，同样的物理数据可以导出多个不同的逻辑文件，用户以简单的逻辑结构操作他的数据而无需考虑数据的存储情况，改动数据的物理位置和存储结构不必修改或重写应用程序，用户逻辑数据与它们的物理存储之间的转换由数据管理软件完成(如图1.5所示)。从而解决了数据的应用独立于数据的存储问题。

初期数据库系统数据的整体逻辑结构仅是用户逻辑文件的简单并集，在用户越来越多，系统为每个用户提供的逻辑文件日渐庞杂的情况下，数据库的组织也就越来越乱。为了提高效率，减少冗余，增加新的数据，常常需要改变数据的整体逻辑结构，这就必然导致用户逻辑文件，进而导致用户应用程序的修改。特别是对某些系统来说，改变整体逻辑结构已成为系统活动的方式，这样就提出了用户的数据逻辑结构尽量不受整体逻辑结构变化的影响问题，促使人们把用户观点的逻辑结构从整体逻辑结构中独立出来，形成今天的数据库系统。

目前数据库系统的最大特点就是三级结构和两级数据独立性，即在用户数据逻辑结构与数据的物理存储结构之间加入数据的整体逻辑结构，使得数据的物理存储结构的变化尽量不

影响数据的整体逻辑结构或用户的应用程序；数据整体逻辑结构的改变也尽量不影响用户应用程序。这种系统的结构如图1.6所示。

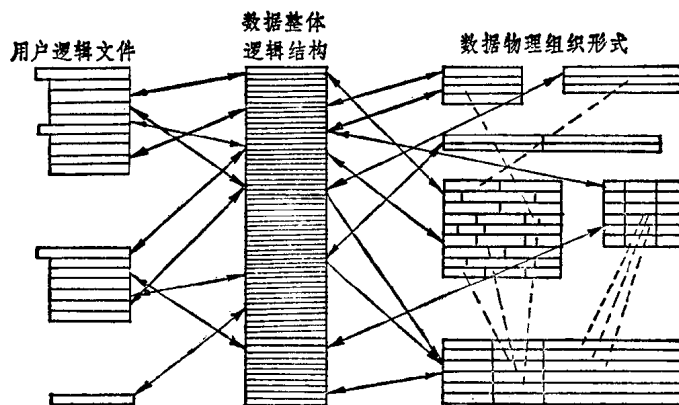


图1.6 今天的数据库系统

### §1.2.2 数据库的设计要求

在讨论数据库的结构与设计方法之前，先介绍一下数据库设计的原则要求，以便在设计时心中有数，能根据用户要求、当前的经济技术条件和已有的软、硬件实践经验，来选择有效的、与之适应的设计方法与技术。

(1) 数据独立性 (data independence)。设计数据库时，首先要求保证数据独立性，作到系统数据存储结构与数据逻辑结构的变化，尽量不影响应用程序和用户原有的应用。

(2) 减少数据冗余，提高共享程度。同一系统包含大量重复数据，不但浪费大量存储空间，还潜在有不一致的危险，即同一记录在不同文件中可能不一样（如修改某个文件中某个数据而没有在另外的文件中作相应的修改）。因此，设计数据库时要消灭有害的数据冗余，提高数据的共享程度。但是，有时为了缩短访问时间或简化寻址方法，也人为地使用数据冗余技术；为了保证数据库的快速恢复，也需要不断地建立数据库的副本。所以，在设计数据库时，只能要求消除有害冗余，而不能要求去掉一切冗余数据。

(3) 用户与系统的接口要尽量简单，系统应具有很强的数据管理能力，能满足用户容易掌握、使用方便的要求。例如：使用高级的非过程化的询问语言或简单的终端操作命令，为用户提供简单的逻辑数据结构；能适应批处理应用程序要求数据流量大、终端用户需要“响应时间”满足人机对话的要求、实时系统要求快速响应等的操作环境；具有处理非预期询问的功能等。

(4) 确保数据库系统的可靠、安全与完整。一个数据库系统的可靠性体现在它的软、硬件故障率小，运行可靠，出了故障时可以快速地恢复到可用状态；数据的安全性是指系统对数据的保护能力，即防止数据有意或无意的泄露，控制数据的授权访问。故在设计系统时必须增加各种安全措施，这已成为当前计算机系统专家们专门研究的课题；完整性是保证数据库仅仅包含正确数据的问题，不正确的数据可能由有意或无意的错误操作产生，也可能由某些不符合实际情况的错误推导产生。总之，设计数据库时要求系统尽可能作到维护数据的完整性，目前的系统通常设置各种完整约束条件来解决这一问题。

(5) 一个设计优良的数据库系统应具有重新组织数据的能力。数据库系统通常把用户频繁访问的数据放在快速访问设备上(如磁鼓或磁盘), 而把很少访问的数据保存在慢速访问设备中(如磁带), 但数据访问的频繁程度并不是一成不变的; 另外, 数据库经过一段时间运行后, 由于频繁的插入、删除操作, 使原有的物理文件变得很乱, 时空性能很差。为了适应数据访问频率的变化, 提高系统性能, 改善数据组织的零乱和时空性能差的状况, 都要及时有效地改变文件的结构或物理布局, 即改变数据的存储结构或移动它们在数据库中的存储位置, 这种改变称为数据的重新组织。现今设计的数据库系统总是周期地由系统自动来完成这个任务。

(6) 设计数据库时应充分注意系统的可修改与可扩充性, 整个系统在结构和组织技术上应该是容易修改和扩充的。因为一个数据库通常不是一次而是逐步建立起来的, 企业的操作数据常在不断地增加和扩充; 另外, 数据库的用户和应用也会不断地变化。所以在设计数据库时要考虑与未来应用接口的问题, 不致于因为以后情况的变化而使整个数据库设计推倒重来或使已经建成的数据库系统不能正常工作。并且在修改和扩充系统后, 不应影响原有用户的使用方式, 如不必修改和重写原有的应用程序。

(7) 数据库应能充分描述数据间的内在联系。人们建立数据库, 是想用数据反映客观事物及其间的联系。于是数据库系统必须有能力描述反映客观事物及其联系的复杂的数据逻辑结构, 而不应使用那些不能充分反映事物内在联系的简单的数据结构。如学生与所学课程之间是一种多对多的联系, 不适合用树形结构表示; 但系、教研室、教员、课程、班级、学生之间的联系用图 1.7 的树形结构来表示却是可以的。

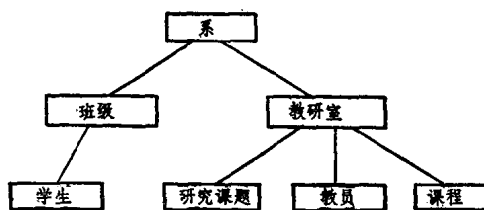


图1.7 树形数据结构

## § 1.3 现实世界的的数据描述

### § 1.3.1 信息的三个领域

从图1.2看出, 在信息控制系统中, 信息从客观事物出发, 流经数据库, 通过控制决策机构, 最后又回到客观事物。信息的这一循环经历了三个领域: 现实世界、观念世界和数据世界。

(1) 现实世界 是存在于人们头脑之外的客观世界, 事物及其相互联系就处在这个世界之中。事物可分成“对象”与“性质”两大类, 又可分为“特殊事物”与“共事物”两个重要级别。

(2) 观念世界 是现实世界在人们头脑中的反映。客观事物在观念世界中称为实体, 反映事物联系的是实体模型。

(3) 数据世界 数据是观念世界中信息的数据化, 现实世界中的事物及联系在这里用数据模型描述。

三个领域的内容及其联系可用图1.8表示。

因此, 客观事物系信息之源, 是设计数据库的出发点, 也是使用数据库的最终归宿。实体模型与数据模型是对客观事物及其联系的两级抽象描述。数据库的核心问题是数据模型, 为了得到正确的模型, 首先要充分了解客观事物。

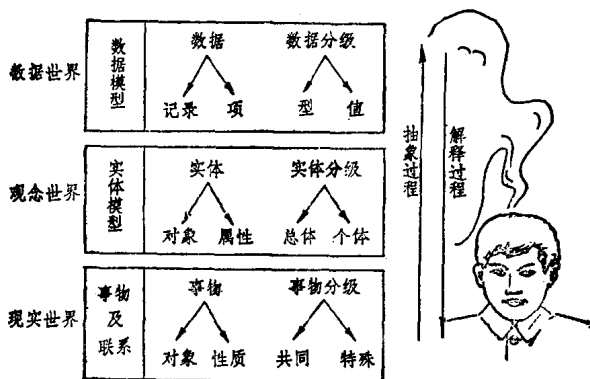


图1.8 信息的三个领域

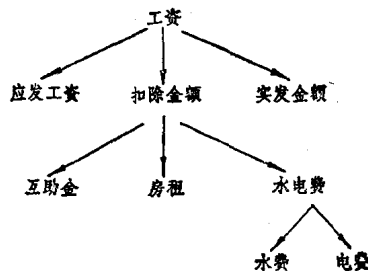


图1.9 可分属性

### § 1.3.2 实体模型

实体模型是设计数据库的先导，因为它是确定数据库包含哪些信息内容的关键。为了使数据库的设计不至由于考虑欠周须大量修改或推倒重来，设计者必须与用户合作，开列用户问题表，而后对现实世界中所要考虑的客观事物及其联系进行模拟，建立一个正确反映客观事实的实体模型。故应周密地考虑如下几个方面：

(1) 对象与属性 在观念世界中，我们用实体描述客观事物。实体可分成“对象”与“属性”两大类：如，人、车、工厂、学校，又如张三、第一汽车制造厂、国防科技大学等属于前者；后者表示对象的某种特征，例如姓名、出生、性别、民族、籍贯，表示了对象“人”的五个方面的特征。对象与属性分别是客观事物中对象与性质的抽象描述，既有区别又有联系。一个对象具有某些属性，若干属性又描述某个对象。但这种区别又是相对的。即一个对象具有的某一属性，又可能是另一些属性描述的对象。例如对象“车”具有属性：型号、颜色、制造厂等，而制造厂又是属性厂名、厂址、建厂时间、规模等所描述的对象。

不能再细分的属性称为原子属性，如性别、颜色等；还可细分的属性为可分属性，如“出生”可分为“年”、“月”、“日”，“健康状况”可分为“身高”、“体重”、“血压”等。当然可分与不可分也具有相对性。例如在描述职工一般情况时，“工资”可作为原子属性，对每个职工只须指出工资为多少即可；但财务部门发放职工工资时，工资单上一般又分为应发工资、扣除金额、实发金额三个方面。因而“工资”又成了可分属性。“扣除金额”还可分为互助金、房租、水电费等；“水电费”又可进一步分成水费、电费（图1.9）。可见，可分属性又由原子属性、可分属性组成。

上述对象与属性、原子属性与可分属性之间的相对性问题，对构造实体模型非常重要。因为，其所以具有相对性是由于描述的事物不同，观察研究问题的角度不同而引起的。所以在构造实体模型时，要辩证地研究客观事物。

(2) 个体与总体 实体又分为两级，一级是“个体”，指单个的能互相区别的特定实体，如“张三”、“国防科技大学”；另一级是“总体”，泛指某一类个体组成的集合。如“人”泛指张三、李四等个体组成的集合；“学校”泛指北大、武大、清华大学等组成的集合。

概括地说，对象与属性的联系是对象内部的联系，而个体与总体的联系是对象的外部联系。但随着考虑问题的范围的变化，内部与外部的概念也在变化。从小范围看是外部的东西，从大范围看就是内部的了。

(3) 总体之间的联系 设有两个均包含有若干个体的总体 A、B, 其间建立了某种联系。可将联系方式分为如下三种:

① **一对一联系** 如果 A 中的任一个体至多对应于 B 中的一个个体; 反之, B 中的任一个体, 至多对应于 A 中的一个个体, 则称 A 对 B 是一对一的联系。用图 1.10 表示。例如, 电影

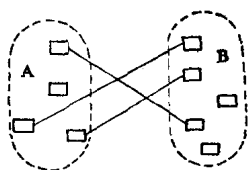


图 1.10 一对一联系

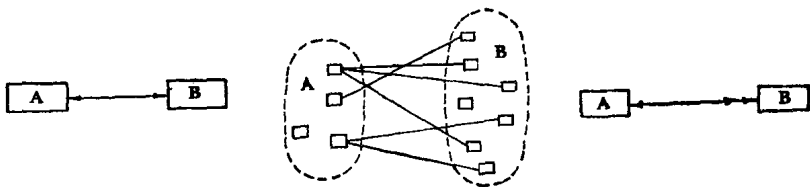


图 1.11 一对多联系

院中观众与座位之间、乘车旅客与车票之间、病人与床位之间、学校与校长(不包括副校长)之间都是一对一联系。

② **一对多联系** 如果 A 中至少有一个个体对应于 B 中一个以上个体; 反之, B 中任一个体至多对应于 A 中一个个体, 则称 A 对 B 是一对多联系。用图 1.11 表示。例如, 父亲对子女、省对县、城市对街道、学校对系、班级对学生等都是—对多联系。

③ **多对多联系** 如果 A 中至少有一个个体对应于 B 中一个以上个体; 反之, B 中也至少有一个个体对应于 A 中一个以上个体, 则称 A 对 B 是多对多联系。用图 1.12 表示。例如学生与课程、工厂与产品、商店与顾客等都是多对多联系。

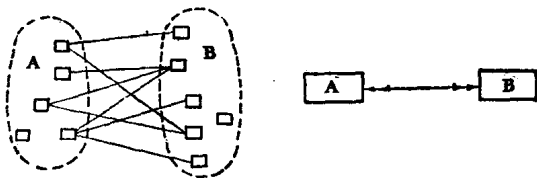


图 1.12 多对多联系

有人还提到“多对一”联系, 但从本质上说, “多对一”不过是“一对多”的逆转。因为 A 对 B 的多对一联系等价于 B 对 A 的一对多联系。原则上, 许多总体之间的复杂联系, 都可用若干组上述联系等价地表示。

(4) 实体模型 考察和研究了客观事物及其联系后, 即可着手建立实体模型进行描述。在模型中, 实体要逐一命名以资区别, 并描述其间的各种联系。现以教学情况为例来建立实体模型。

教学情况可由学生、课程、教员、学习、任课等方面的情况组成。其中, 学生具有属性: 学号、姓名、年龄、性别(当然还有其他属性, 为使讨论简单故予省略。下同); 课程具有属性: 课程号、课程名、学时数; 教员具有属性: 姓名、年龄、职称; 学习实际上是学生与课程之间的一种联系, 它也可看作一种对象, 并具有属性: 学号、课程号、分数; 任课是教员与课程之间的一种联系, 也可看作对象, 并且有属性: 课程号、任课教员。

学生对课程是多对多联系, 因一个学生可以学习多门课程, 而一门课程又有多个学生学习。教员对课程假设是一对多联系, 即一个教员可以讲授多门课程, 但一门课程至多只能由

一个教员任教（同一门课程分别由几个不同教员开课时，看作几门课程，它们的课程名可以相同，但课程号不能相同就可以区别，这样就可以满足一对多的假设）。我们用矩形表示对象，用椭圆表示属性，把教学情况的实体模型表示在图1.13上。

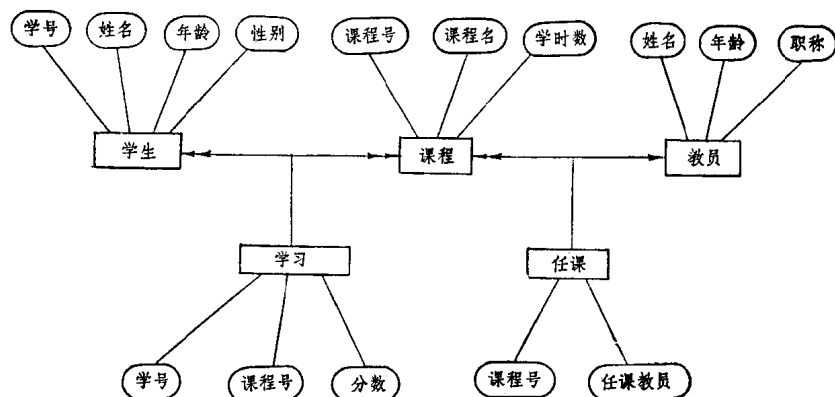


图1.13 教学实体模型

这是一种静态实体模型，只反映实体的当前状态，而不能反映实体状态的变化过程。目前数据库多是根据这种模型设计的，因而只能回答有关实体当前状态的问题。例如回答“某学生的年龄”、“某课程的学时数”、“某学生某课程的分数”等。至于动态实体模型尚待进一步研究。

为使建立的实体模型包含所需信息，以及信息间的联系清晰，在其建立过程中应考虑：

- ① 用户需要数据库回答哪些问题，即用户问题表中要填写哪些项目；
- ② 根据问题表，实体模型应该包含哪些实体才能圆满回答用户的问题；
- ③ 这些实体中哪些是明显的对象；
- ④ 根据考虑问题的范围和角度，确定对象的何种联系与属性也应视为对象；
- ⑤ 根据现有技术条件，实际上能够观测和存储哪些对象和哪些属性。

### § 1.3.3 数据模型

数据模型是对客观事物及其联系的数据描述，即实体模型的数据化。数据库设计的核心问题之一就是设计一个好的数据模型。下面将对数据模型设计的有关问题予以简略介绍。

(1) **记录与数据项**（简称项） 数据模型中，用数据描述的实体有对象与属性之分，我们把描述对象的数据称为记录，而把描述属性的数据称为项。由于一个对象具有若干属性，故记录亦由若干项组成，或者说记录是项的一个序列。又由于有原子属性和可分属性，所以项又分成基本项与组合项。基本项（原子属性）是具有名称的最小逻辑数据单位；组合项（可分属性）则由基本项与组合项构成。

一般就采用属性名作为描述它的数据项名。但用作属性名时表示观念信息；用作项名时表示数据信息，它包含项的特征——数据类型（数字、字母、字符串等）与数据长度（几位数、几个字母、几个字符或字节等）。

(2) **型与值** 由于实体分为总体与个体两级，所以表示它的数据也分为“型”（表示总体，如学生、年龄、性别、课程等）与“值”（表示个体，如张三、20岁、男、数据库等）两级。前述记录与数据项都有型与值之分，数据项“年龄”的型就是“名称为年龄，数据特征为三位的



十进制整数”，而它的值是“1岁、2岁、…、200岁等”。一般都规定有数据项的取值范围（值域），超出值域的值则认为无实际意义。例如“年龄”的值域可规定在1到200之间，超过200岁的人认为是不存在的。

数据的型也称为**元数据**（或关于数据的数据），它在数据模型中占据极其重要的地位。但型与值是相对的。即一个数据项的值可以是另一个数据项的型，反之亦然。例如，数据项“车”的值“汽车”，可以是数据项“汽车”的型，而数据项“汽车”的值又是“公共汽车”、“吉普车”、“小卧车”、“卡车”等。这是实体中总体与个体相对性在数据上的反映。

为了明显地区分型与值，给数据术语附加后缀“类型”，表示该术语用于型级。如“记录类型”；而附加后缀“值”或“实例”表示用于值级。如“记录值”、“记录实例”。但在许多情况下并不附加后缀，同一术语却在型与值不同的两级平行使用。这时就要根据上下文的含义，才能判断该术语是指型还是值。这一特点在数据库文献中是很突出的。

（3）**记录与文件** 记录类型是数据项型的一个有序组。同理，记录值是数据项值的同一有序组（常简称记录）。记录类型是一个框架，只有给它的每个数据项取值后才得到记录。正如一张学生登记表一样，填写前是学生记录类型，填写后就得到一个学生记录。

表示记录类型有图示与使用数据语言描述两种方法。在图示法中，记录类型用方框图表示，它的名称写在方框图左上方，各数据项名按顺序写于方框图的小格中。现分别用图示法和DBTG数据描述语言，描述学生记录类型如下：

| 学 生 |     |     |   |   |     |
|-----|-----|-----|---|---|-----|
| 学 号 | 姓 名 | 出 生 |   |   | 性 别 |
|     |     | 年   | 月 | 日 |     |

```

RECORD NAME IS STUDENT,
03 S-NO, PICTURE IS 9 (5)
03 NAME, TYPE IS CHARACTER 12
03 BIRTH-DATE.
04 YEAR, PICTURE IS 9 (4)
04 MONTH, PICTURE IS 99
04 DAY, PICTURE IS 99
03 SEX, PICTURE IS A
    
```

| 表1.3 学 生 文 件 |       |      |    |    |     |
|--------------|-------|------|----|----|-----|
| 学 号          | 姓 名   | 出 生  |    |    | 性 别 |
|              |       | 年    | 月  | 日  |     |
| 78601        | 王 晓 燕 | 1961 | 6  | 15 | 女   |
| 78602        | 李 波   | 1958 | 11 | 9  | 男   |
| 78603        | 陈 志 坚 | 1960 | 10 | 27 | 男   |
| 78604        | 张 兵   | 1961 | 6  | 15 | 男   |
| 78605        | 张 兵   | 1960 | 1  | 30 | 女   |

我们把**文件**定义为**记录型与值的总和**，一个记录类型和它的一些当前记录组成**同质文件**。不同记录类型和它们的当前记录组成**异质文件**。例如表1.3的学生文件就是一个同质文件，它具有五个当前记录。其中第一个记录“78601王晓燕 1961.6.15 女”，描述了学生王晓燕的有关属性。

在文件中，如果两个记录至少有某个数据项的值不相同，则称为不同记录；如果它们对应数据项的值都相同，则称为相同记录。相同记录中，一个以外的其他记录都称为重复记录。文件中不允许有无法区分的重复记录。通常只要给出记录中某个或某些数据项的值，就能确定所指记录。例如指出学号的值“78604”，就知道所指记录是“78604 张兵 1961.6.15 男”；但若给出数据项性别的值“男”，是无法确定所指记录的。因为有三个记录的性别值都为“男”。因此，我们把其值能够唯一标识记录的一个或多个数据项称为记录类型（或文件）的**关键字(key)**。把用于组织文件的关键字称为主关键字(*primary key*)。有时也把不能唯一标识记录的数据项称为辅关键字(*secondary key*)，或简称关键字（这时它已失去唯一标识记录的涵义了）。