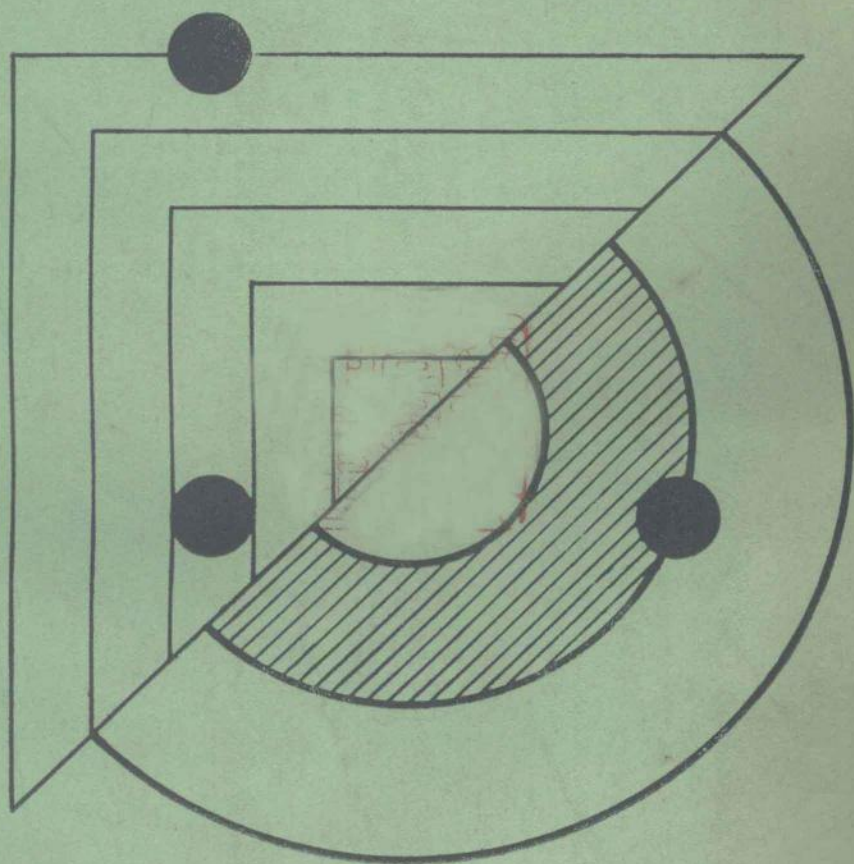


数据库系统基础教程

刘金宏 黄 兴 编译

项源金 审校



中国铁道出版社

数据库系统基础教程

刘金宏 黄 兴 编译

项源金 审校

中 国 铁 道 出 版 社

1 9 9 0 年 · 北 京

内 容 简 介

《数据库系统基础教程》一书由四部分组成。第一篇基本概念；第二篇关系系统与关系理论；第三篇数据库环境；第四篇非关系系统及数据库系统发展方向。全书内容充实全面，理论严谨完整，突出关系模型，既注重反映数据库系统发展的现状和方向，也介绍了数据库系统的发展历史，并具有理论联系实际的特点。

本书可作为高等院校计算机软件专业、管理信息专业的数据库课程的教材和教学参考书，也可供其他计算机软件技术和开发人员参考。

数据库系统基础教程

刘金宏 黄 兴 编译
项源金 审校

*

中国铁道出版社出版、发行
(北京市东单三条14号)

责任编辑 方齐盛 封面设计 裴 忠
各地新华书店经售
河北省深县印刷厂印

开本：787×1092毫米 1/16 印张：21.5 字数：533千

1990年12月 第1版 第1次印刷

印数：1—2000册

ISBN7-113-01042-3/TP·103 定价：10.00元

编 译 者 的 话

数据库是计算机软件专业、计算机应用专业及管理信息专业的大学高年级的重要专业课。学习本课需要计算机软件方面的理论基础、技术基础及各种专业课程的相当广泛的知识。其中重要的是离散数学、计算机语言、数据处理、数据结构、操作系统等课程。因此，完全有理由认为，数据库课程是对各种计算机软件知识和技能的全面综合运用的一门课，是计算机应用及软件开发人员必受的一种专业训练。因此，它在计算机软件专业、计算机应用专业及管理信息等专业中占有相当重要的地位。

《数据库系统基础教程》一书就是适应上述有关专业的教学需要而编译的。它内容充实全面、理论完整严谨、突出关系模型、反映最新发展并具有理论联系实际的特色。全书共四篇。第一篇基本概念。它有三章内容，主要介绍数据库系统的逻辑层次结构、数据独立性、操作数据、映射、DBMS及DBA等基本概念以及索引、散列、指针链和压缩技术等存贮技术。第二篇介绍关系系统与关系理论。本篇以DB2为背景，以表的概念为基础，较详细地介绍了SQL语言的数据定义功能、数据操作功能以及系统目录、视图等概念，并简要介绍了非SQL系统的INGRES系统的视图技术和QUEL语言。在此基础上又介绍了规范的关系术语、关系完整性法则、关系代数、关系演算以及规范化理论等。这一篇是全书的核心，全篇长达十四章。掌握本篇内容之后，读者可对目前流行的具体关系数据库系统有高度的适应性，并打下了坚实的理论基础。第三篇数据库环境。用三章的篇幅介绍大型多用户系统的并行性、可恢复性、安全保护技术和数据库的配套软件产品。第四篇非关系系统及数据库发展方向。本篇从数据库发展的角度扼要介绍倒排表系统（DATACOM/DB）、层次系统（IMS）、网状系统（IDMS），以及反映了数据库系统发展方向的分布式系统和语义模型设计，全篇共五章。为了配合各章节的学习，帮助读者对所学内容的理解，全书各章都附有适量习题。习题安排由浅入深，少数有较大难度，希望读者根据情况选做。

由于本书内容极为丰富，涉及的设备十分广泛，一般的学校和单位不可能齐备，再加之授课时数一般都较紧张，因此，经常会出现根据设备和教学时数对全书内容进行剪裁的情况。根据我们的经验，本书第一篇为必授内容；第二篇应以DB2为主，其它理论部分可选其以DB2为基础的部分讲授，然而对于具备INGRES系统的学校或单位应当另做别论；第三篇的前两章应该是必授章节；第四篇可根据设备情况在前三章中择一章细讲，其它可作些介绍或留给学生自读。

尽管我们对本书的编译和出版尽了最大的努力，但受经验和水平所限，缺点和错误在所难免，敬请读者不吝赐教。

编 译 者

一九九〇年十一月

目 录

第一篇 基本概念

第一章 数据库管理概述	(1)
第一节 什么是数据库系统.....	(1)
第二节 操作数据.....	(5)
第三节 为什么使用数据库.....	(6)
第四节 数据独立性.....	(8)
习题.....	(10)
第二章 数据库系统结构	(12)
第一节 数据库系统分三层结构.....	(12)
第二节 数据库系统的详细结构.....	(14)
第三节 映射、DBA和DBMS.....	(16)
习题.....	(18)
第三章 内部层	(19)
第一节 概述.....	(19)
第二节 数据库存取过程概述.....	(20)
第三节 页面集和文件.....	(22)
第四节 索引.....	(27)
第五节 散列.....	(32)
第六节 指针链和压缩技术.....	(35)
习题.....	(39)

第二篇 关系系统与关系理论

第四章 DB2概述	(41)
第一节 关系数据库.....	(41)
第二节 SQL语言.....	(43)
第三节 系统的主要结构.....	(46)
第五章 数据定义	(50)
第一节 基本表.....	(50)
第二节 索引.....	(51)
习题.....	(53)
第六章 数据操作	(54)
第一节 简单查询.....	(54)
第二节 联接查询.....	(57)

第三节	内部函数的使用.....	(62)
第四节	SELECT语句的高级功能.....	(64)
第五节	更新操作.....	(71)
习题	(75)
第七章	系统目录	(77)
第一节	系统目录的基本结构.....	(77)
第二节	系统目录的查询.....	(78)
第三节	系统目录的更新.....	(79)
习题	(80)
第八章	视图	(81)
第一节	概述.....	(81)
第二节	视图的定义.....	(82)
第三节	对视图的操作.....	(84)
第四节	视图的作用.....	(87)
习题	(88)
第九章	宿主型SQL	(90)
第一节	基本概念.....	(90)
第二节	无指针的操作.....	(92)
第三节	带指针的操作.....	(93)
第四节	综合举例.....	(95)
第五节	动态SQL.....	(99)
第六节	对宿主型SQL的小结.....	(100)
习题	(100)
第十章	典型的非SQL系统——INGRES简介	(102)
第一节	数据定义.....	(102)
第二节	数据检索操作.....	(105)
第三节	数据更新操作.....	(110)
第四节	INGRES的视图.....	(112)
第五节	宿主型QUEL语言.....	(113)
习题	(115)
第十一章	关系数据结构	(116)
第一节	从一个引例谈起.....	(116)
第二节	域.....	(117)
第三节	关系.....	(120)
第十二章	关系完整性法则	(124)
第一节	主关键字和外关键字.....	(124)
第二节	关系完整性法则.....	(125)
第十三章	关系代数	(129)

第一节	概述	(129)
第二节	传统的集合运算	(131)
第三节	特殊的关系运算	(133)
第四节	关系代数的应用	(138)
	习题	(140)
第十四章	关系演算	(141)
第一节	概述	(141)
第二节	面向元组的关系演算	(142)
第三节	元组关系演算举例	(145)
第四节	关系代数与关系演算	(146)
第五节	面向域的关系演算	(150)
第六节	QBE语言简介	(152)
	习题	(157)
第十五章	关系系统	(158)
第一节	关系模型	(158)
第二节	必要数据	(159)
第三节	关系系统定义	(162)
第四节	关系完整性法则的实施	(164)
	习题	(167)
第十六章	查询的优化	(168)
第一节	引例	(168)
第二节	优化过程	(169)
第三节	系统R中的优化	(171)
第四节	INGRES中的优化方法	(173)
第五节	关系运算符的实现	(175)
第十七章	规范化	(180)
第一节	概述	(180)
第二节	函数依赖	(182)
第三节	第一、二和第三范式	(183)
第四节	BCNF范式	(189)
第五节	分解的优劣	(192)
第六节	第四范式	(194)
第七节	第五(PJ/NF)范式	(196)
第八节	小结	(200)
	习题	(201)

第三篇 数据库环境

第十八章	可恢复性和并行性	(203)
第一节	作业处理的恢复.....	(203)
第二节	系统和存贮介质的可恢复性.....	(205)
第三节	三个并行问题.....	(207)
第四节	锁闭.....	(210)
习题	(216)
第十九章	安全性和完整性	(218)
第一节	安全性.....	(218)
第二节	SQL系统的安全性.....	(219)
第三节	安全性其它方面的问题.....	(222)
第四节	完整性.....	(223)
第五节	INGRES中的安全性和完整性.....	(228)
习题	(230)
第二十章	与数据库系统配套的软件产品	(232)
第一节	概述.....	(232)
第二节	数据存取.....	(233)
第三节	数据表示方式.....	(237)
第四节	应用程序生成.....	(242)
习题	(245)

第四篇 非关系系统及数据库系统发展方向

第二十一章	倒排表系统: DATACOM/DB	(247)
第一节	倒排表模型.....	(248)
第二节	DATACOM/DB 概述.....	(249)
第三节	数据定义.....	(250)
第四节	数据操作.....	(254)
第五节	复合布尔选择特性.....	(256)
习题	(258)
第二十二章	层次系统: IMS	(259)
第一节	层次模型.....	(259)
第二节	IMS概述.....	(263)
第三节	数据定义.....	(264)
第四节	数据操作.....	(266)
第五节	存贮结构.....	(271)
第六节	逻辑数据库.....	(274)

第七节 辅助索引.....	(278)
第八节 小结.....	(282)
习题.....	(283)
第二十三章 网状系统: IDMS.....	(285)
第一节 网状模型.....	(285)
第二节 IDMS概述.....	(289)
第三节 数据定义.....	(290)
第四节 数据操作.....	(293)
第五节 存贮结构.....	(300)
第六节 逻辑记录功能.....	(302)
第七节 自动系统功能.....	(306)
第八节 小结.....	(309)
习题.....	(311)
第二十四章 分布式系统.....	(313)
第一节 概述.....	(313)
第二节 目标及其合理性.....	(314)
第三节 分布式系统存在的问题.....	(315)
第四节 几个实例分布式系统.....	(318)
第五节 其它种类的分布式系统.....	(321)
习题.....	(324)
第二十五章 语义模型设计.....	(325)
第一节 一般的实现方法.....	(325)
第二节 实体/联系模型.....	(326)
第三节 扩充的关系模型 RM/T.....	(328)
习题.....	(333)

第一篇 基本概念

第一篇包括三章。第一章论述什么是数据库和为什么要使用数据库。第二章给出数据库系统的ANSI/SPARC结构以便为后继内容奠定基础。第三章介绍在数据库中物理地存放数据的一些技术。

第一章 数据库管理概述

第一节 什么是数据库系统

一、引例

从根本上说，数据库系统就是一个计算机化的数据文件保管系统。这个系统的整个目标就是维护信息并使之满足个人或组织的需要。数据库系统用户可以方便地对数据库的数据进行下列各种操作：

- 往数据库中添加新文件；
- 在已有文件中插入新数据；
- 访问已有文件中的数据；
- 修改已有文件中的数据；
- 从现有文件中删除数据；
- 从数据库中永久地删除现存文件；

图1.1 给出了一个仅含有一个名为CELLAR文件的很小的数据库。它保存的是个酒库的信息。图1.2是对CELLAR的一次访问及其结果。图1.3是对该数据库的插入、修改和删除数据的操作。

CELLAR文件:

BIN	WINE	PRODUCER	YEAR	BOTTLES	READY	COMMENTS
2	Chardonnay	Buena Vista	83	1	85	
3	Chardonnay	Louis Martini	81	5	84	
6	Chardonnay	Chappellet	82	4	85	Thanksgiving
11	Jo,Riesling	Jekel	84	10	86	
12	Jo,Riesling	Buena Vista	82	1	83	Late Harvest
16	Jo,Riesling	Sattui	82	1	83	very dry
21	Fume Blanc	Ch.St.Jean	79	4	83	Napa Valley
22	Fume Blanc	Robt.Mondavi	78	2	82	

25	Wh. Burgundy	Mirassou	80	6	82
30	Gewurztraminer	Buena Vista	80	3	82
43	Cab. Sauvignon	Robt. Mondavi	77	12	87
50	Pinot Noir	Mirassou	77	3	85 Harvest
51	Pinot Noir	Ch. St. Jean	78	2	86
64	Zinfandel	Mirassou	77	9	86 Anniversary
72	Gamay	Robt. Mondavi	78	2	83

图1.1 酒库数据库

```
SELECT WINE, BIN, PRODUCER
FROM CELLAR
WHERE READY=85;
```

在屏幕或打印机输出结果:

WINE	BIN	PRODUCER
Chardonnay	2	Buena Vista
Chardonnay	6	Chappellet
Pinot Noir	50	Mirassou

图1.2 对酒库数据库进行访问的样例

插入新数据?

```
INSERT INTO CELLAR
VALUES (53, 'pinot Noir', 'Franciscan', 73, 1, 85, 'for Joan');
```

修改已有数据:

```
UPDATE CELLAR
SET BOTTLES=4
WHERE BIN=3;
```

删除已有数据:

```
DELETE FROM CELLAR
WHERE BIN=2;
```

图1.3 插入, 修改和删除实例

图1.2, 1.3中的SELECT、INSERT、UPDATE和DELETE操作都是后面要介绍的数据库语言SQL (Structured Query Language) 的语句。

在后面的章节中, 像CELLAR这样的计算机文件一般就当成表格而不使用文件这个术语。其次, 表的每一行称为记录, 说得更准确一点应叫逻辑记录; 列称为字段。近几年来实现的数据库系统几乎都是关系系统, 当前对数据库的研制都是以关系思想为基础的。“关系方法”无论从实际需要还是从教学需要都代表着当今数据库系统的发展方向, 因此本书特别侧重关系方法, 而非关系系统在本书中显然居于次要地位, 简直可以把它当成数据库系统的发展历史来向大家介绍了。那么什么是关系系统呢? 目前要完整准确地回答这个问题是不可能的。然而出于急需关系的概念, 在这里只能认为, 关系

系统是这样—个系统，在关系系统中：

用户看到的数据只不过是个表；

用户使用的操作符结果是从旧表生成新表；例如图1.1就是一个表，图1.2的结果就是从图1.1的旧表生成的新表。图1.2是图1.1的一个子集。

二、数据库系统的简单结构

图1.4表示一个数据库系统由四个主要部分构成：数据、硬件、软件和用户。下面就简单介绍其每一部分，以后的章节再来进一步研究它们。

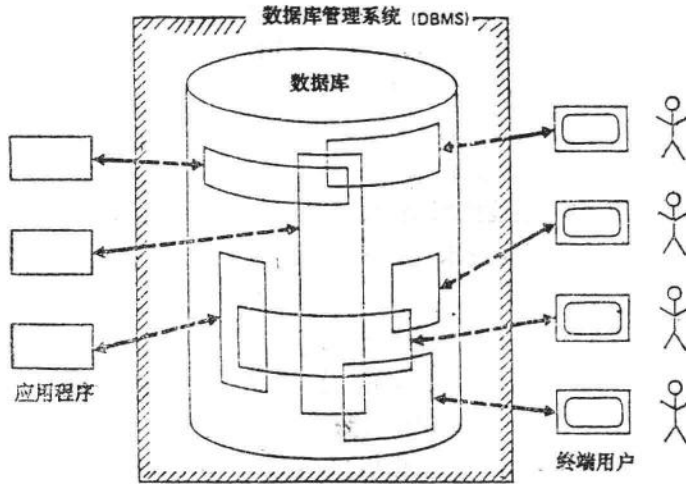


图 1.4 数据库系统简图

1、数据 从巨型机到微机各种类型的计算机都可配置数据库系统，然而其功能在很大程度上与机型及其能力有关。一般来说，大系统多为多用户系统，而较小的系统倾向于单用户系统。图1.4的简图显然表示一个多用户系统。多用户系统和单用户系统的区别在系统内部，大多数应用程序用户是看不到的，在这种用户看来单用户系统和多用户系统并无区别。

为了叙述和分析上的方便，我们做一点假设，即假设存贮在一个系统里的所有数据都存放在—个数据库里。这是真正的假设，因为事实上既便是个小系统也有充分理由将数据分存在几个不同的数据库里。

一般来说，大系统环境下的数据库系统必须具备完整性和共享性两个特点；而小系统环境下的数据库系统至少必须具备完整性。因而完整性和共享性是数据库系统的显著特点。我们来解释这两个术语。

(1) 完整性：完整性的意义是指数据库可以被看成是几个不同数据文件的—体，同时全部或部分地消除诸文件中数据的冗余度。例如，假设一个数据库由两个文件组成，一个文件名为EMPLOYEE，其记录结构为：姓名、住址、部门、工资共四个字段；另一个文件名为ENROLLMENT，它用来为在培训中的雇员注册。假如现在要

进行培训课程管理方面的处理工作，需要知道每一位注册学员的部门，这就不必在 ENROLLMENT 文件中冗余地包括某些字段而通过访问 EMPLOYEE 文件访问到这些信息。

(2) 共享性：共享性的意义是指数据库中的每项数据均可以在同一时刻或不同时刻被若干个不同用户所共同访问来用于不同的目的。不同用户同时访问同一数据项称并行访问，不同时刻访问同一数据项称为非并行访问。例如上述的 EMPLOYEE/ENROLLMENT 数据库中的 EMPLOYEE 文件中的部门字段将是人事部门和教育部门的用户共享的数据。

从上面的分析我们可推论：共享性是完整性所导致的必然结果。我们还可以推断：某一用户一般情况下仅与整个数据库的某一子集相关；不同用户的子集可以通过很多不同的途径发生重叠，换句话说，不同用户将会通过各种不同的途径看待某一给定的数据库。这些内涵可以从图 1.4 分析出来，也是在后面的深入学习中反复体会的内容。

2、硬件 硬件包括存贮数据用的辅助存贮器、相应的 I/O 装置、控制装置及 I/O 通道等。这些设备均不是数据库系统范围内详细讨论的内容。本教程仅在后面介绍一些在辅存上物理组织数据库的技术并在最后简单介绍一下数据库机的概念之外不涉及其它硬件知识。

3、软件 (DBMS) 在物理数据库和系统用户之间是一软件层即数据库管理程序，通称数据库管理系统——DBMS。它的全名是 Data Base Management Systems。我们可以用一句话来概括 DBMS 的功能，DBMS 为用户提供了一个在硬件层之上的看待数据库的视图，它支持着用户使用像 SQL 等这样的高级视图语言对数据库的操作。本书通篇将相当详尽地讨论 DBMS 的功能。

4、用户 数据库系统用户分为三类。

(1) 应用程序员：他们使用象 COBOL、PL/1、APL 或 PASCAL 等这样的高级语言编写应用程序对数据库进行一般范围内的操作。例如创建新文件、检索、删除、修改已有数据等操作。应用程序通常采用批处理或联机处理两种方式。

(2) 终端用户：终端用户从联机终端使用系统提供的某种用户/系统接口采用和系统对话的方式使用数据库。用户/系统接口一般有两类，第一种是交互式查询语言处理系统，SQL 就是典型一例，用户使用这种查询语言向 DBMS 发送查询命令来使用数据库，因此类似于 SQL 这样的查询语言也称命令驱动接口。第二种用户/系统接口称表格驱动接口，QBE 就是典型一例，使用这种接口的用户采用填表的方式和 DBMS 对话使用数据库。两种驱动接口比较起来各有千秋，命令驱动接口可以嵌入高级程序设计语言里一起使用，因而它具有较大的灵活性，但使用起来需要某些数据处理方面的专门知识；表格驱动接口使用起来比较简单直观无需数据处理方面的专门知识，但其灵活性显得不足。本教程对命令驱动的 SQL 查询语言介绍得较多，对表格驱动的 QBE 也有一定的介绍。

(3) 数据库管理员 (DBA)：DBA 是对数据库进行全面控制和管理的一个人或一组人员。这是一类非常特殊的用户，本教程将在第二章专节介绍 DBA。

第二节 操作数据

我们先来区分操作数据和输入输出数据，而后举例来进一步强化操作数据的概念。

一、操作数据和输入输出数据

1、操作数据 早些时候人们曾对数据库定义如下，数据库是由某特定企事业应用系统使用的存贮起来的操作数据的集合。这里的企事业可以是一个制造公司、一个银行、一个医院、一个大学或一个政府部门等等。任何这样的企事业必须维护与它的业务有关的大量数据，这些数据就是“操作数据”。相对于上述各企事业的操作数据分别是生产数据、帐目数据、病人数据、学生数据及计划数据。输入输出数据，中间结果数据或其它暂时性信息都不是操作数据。

2、输入输出数据 输入数据是指那些第一次从终端键盘、读卡机或类似的装置进入系统的信息，这种信息要经过变换才能部分或全部成为操作数据，但它们并不是数据库的初始部分。输出数据是指由系统送出的打印结果或终端屏幕显示信息，这种信息可能从操作数据转化而来，但它们本身不能被看成是数据库的一部分。

二、一个操作数据实例

图1.5是一个制造公司的操作数据的实例。通过对它的分析我们可以深入了解操作数据的含义。

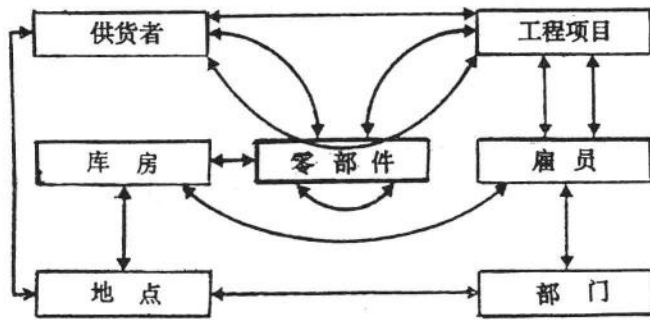


图 1.5 操作数据实例

1、实体与关系 对于图1.5这样一个企业，要维护它现有的工程项目相关的信息，就要记录下工程项目、所使用的部件、部件的供应者、部件存放的库房、项目所雇佣的雇员等等基本实体的信息。“实体”这个概念在数据库领域里广泛使用，它用来在数据库中表示性质不同的事物。

图中除基本实体之外，还存在着将这些基本实体联系在一起的关系，图中用带箭头的连线表示。例如在部件及其供应者之间存在着关系，即某供应者供应某些部件，反之每个部件都由某供应者供应，类似地，部件用于项目，项目使用部件；部件存放在库房里，库房存放部件等等。要注意这种关系都是双重的，这意味着它们可以从任一方向上

访问。例如雇员和部门之间的关系可以用来回答下面任一问题：1) 给定某雇员查找其相应部门；2) 给定某部门查找相应雇员。

从上面的分析推断，关系就像基本实体一样是操作数据的不可缺少的一部分，因此它们必须在数据库中以说明和记录。我们可以认为关系是一种特殊的实体。这样从更广泛的意义上讲数据库中的实体指的是“我们希望记录信息的任何客体”。操作数据指向就是在数据库中对所有实体性质的记录信息。

2、对图1.5的进一步说明

(1) 虽然图中的多数关系含有两个实体，即它们是二元关系。但这并不意味着所有关系都必须都是二元关系，而图中的供应者、部件和项目关系就是个三元关系。其目的是表示某供应者供应某部件给某项目。仔细考查一下，我们会发现“供应者供应部件给项目”一般来说不等价于“供应者供应部件”、“部件用于项目”和“项目由供应者供应”这样三个二元关系的组合。例如：1) Smith供应活动扳钳给Manhattan项目比；2) Smith供应活动扳钳，3) 活动扳钳用于Manhattan项目以及，4) Manhattan项目由Smith供应提供给我们的信息要多。显然，我们不能只知道2)，3)和4)就断定1)。很明显，如果我们知道2)、3)和4)，那么我们可推断Smith供应活动扳钳给某项目Jz；某供应者Sx供应活动扳钳给Manhattan项目；以及Smith供应某种部件Py给Manhattan项目；但是我们显然不能推断Sx就是Smith，Py就是活动扳钳，Jz就是Manhattan项目。这种错误的推理叫做联接中断。

(2) 图中还有一条箭弧只指向部件这一个实体。这个关系表示某部件由另外的部件构成。例如，螺钉是铰装置的部件，而铰又可以看作是个部件，它可是更高一层的部件的部件。特别要指出，这也是个二元关系，只不过被连接在一起的两个实体恰好是同一实体，称作部件对部件。

(3) 一般情况下，给定实体类型集合可以联接成任意数量个关系。图1.5中项目和雇员间有两条箭弧。一条可表示“工作关系”，即在某项目中工作的雇员；另一条可表示“经理人员关系”，即雇员是该项目的经理。

第三节 为什么使用数据库

为什么使用数据库的问题说到根本上是谈数据库具有哪些优点以及使用数据库会给我们带来哪些益处的问题。我们先分析单用户系统再分析多用户系统。

一、单用户系统的优点

图1.1给出的酒库数据库实例是一个很小的单用户系统。设想它是一个大型餐馆或是一存有千万瓶各种酒类的酒库，其特点是销售、进货变化频繁，这时若使用数据库保管信息比起传统的用纸张帐目的方法来可以获得下列明显益处。

- 1、紧缩：无需很多卷纸页文件。
- 2、迅速：机器查询和修改数据远比人工快。
- 3、不乏味：消除了非常简单单调的手工维护文件的工作，技术工作由机器完成更有效。

4、及时：可按要求及时提供准确的最新信息。

二、多用户系统的优点

多用户系统除了兼有上述各种优点之外，其最有价值的特点是多用户系统实现了对企事业单位数据的集中控制。这种集中控制起来的数据较之没有集中控制的数据的益处是明显的。

1、减少冗余度 在非数据库系统里，每一种应用都有自己专用文件，这就会导致存贮数据上的大量冗余，不但造成存贮空间上的浪费，也给修改数据带来很大困难。例如人事部门和教育部门都可以有自己的包含有雇员信息的文件，如果集中控制就可消除冗余。

在这里需要指出的一点是，我们并不强调务必消除所有冗余度。有时出于业务或技术上的理由还要维护某些存贮数据的多重副本。我们只强调对任何冗余应严加控制并且强调DBMS应有相应的技术对冗余数据负起“延续”修改的责任，这些技术在后面的章节都有论述。

2、可在某种程度上避免数据不一致性 避免不一致是消除冗余的必然结果。例如雇员E3在部门D8工作这一信息存在数据库中的两个不同的记录中，如果DBMS没有对这种冗余的控制，若其中之一被修改后就造成了同样一个E3信息的不一致性，显然处于数据不一致中的数据库会给用户提供错误甚至是互相矛盾的信息。如果消除了这种冗余就不会出现不一致性。不消除冗余，而每次修改上述两者之一时DBMS都能确保自动地修改另一个也能确保数据库无任何不一致性。这个自动修改过程就是上面提到的“延续修改”。需要注意的是，目前支持这种冗余控制技术的产品不多。

3、数据可以共享 共享问题前面已有论述，在此仅强调一点，这就是不仅已有的用户可共享数据库中的数据，而且新加入的用户也可以共享这些数据。

4、可以实行标准化 有了数据的集中控制，就为DBA在数据的表示中遵循标准格式提供了条件。标准化可以是一个协会的、一个计算站的、一个部门的、全国的或国际间的标准。标准化的数据存贮格式特别有利于数据交换及传输，数据命名和文件编制的标准化也有助于共享和便于理解。

5、可以施加保密限制 对于整个操作数据拥有全部权限的DBA可以为用户存取数据提供适当渠道和手段，当然也必须对用户的存取提供安全检查方法。这种检查方法就是数据库系统集中化本身决定了必须设置可靠的安全保密系统。

6、可以维护数据的完整性 完整性是指保证数据库中的数据准确无误。前面提到的数据库不一致性是破坏了数据库完整性的一种情况。除此之外还有一种错误信息，例如某雇员一周工作小时不是40而是400，或者雇员属于一个不存在的部门等都是另一种不完整性。数据库集中控制有助于避免这类问题。每当对数据库进行修改时都要执行完整性检查。需要指出的是，数据完整性在多用户系统中比在“专用文件”系统中更为重要。这是因为由一个用户错误地修改的数据会无辜地影响到其它使用该数据的用户。目前大多数数据库产品在支持完整性控制方面仍是个弱点。

7、互相冲突的要求可以得到充分平衡 DBA了解整个企事业的所有要求，就可以把系统设计得能为整个企事业提供最适当的服务水平。例如，存贮数据可以选择得能

为最主要的用户提供最快速的存取结构。

在上面对多用户系统优点的分析中多次提到DBA，显见DBA在设计和维护一个多用户数据库系统中的地位的重要。DBA应该具有高深的数据库设计和维护的技术经验也要在高层管理水平上深刻了解一个企事业单位，因而DBA一般是由熟知管理工作和熟知技术工作两部分人员组成的一个小组。

第四节 数据独立性

第三节分析的数据库系统的优点是明显的，也带有某种必然性。除此之外，数据库系统还具备有一点更为重要的、不甚明显的、可以认为是隐含在其它优点之中的、实质上是数据库系统要达到的设计目标的优点，这就是数据独立性。我们逐步论述数据独立性。

一、数据相依性和数据独立性

1、数据相依性 数据在辅助存贮器上的组织方法和对于它们的存取方法都取决于所涉及它们的应用程序的要求叫做数据相依性。这也就是说这些数据的组织和存取技术都包含在应用逻辑和程序之中。例如，我们决定文件EMPLOYEE以“雇员号码”字段为索引存放。因而在使用该文件时一般都要考虑索引的存在、还要考虑由索引所确定的文件顺序，应用程序在设计时要充分考虑这一外部环境。换句话说，应用程序中各种数据存取的准确格式和特殊检查过程在很大程度上将依赖于文件管理系统软件提供的接口的细节。

我们称这种应用为数据相依的。因为在这种情况下，要改变存贮结构或存取策略而不影响其应用程序是断然办不到的。例如不对应用程序做重大修改而用散列编址文件代替上述索引文件是不可能的。更严重的困难在于，在这种情况下应用程序要做的修改部分恰好是和文件管理软件通讯的部分，其之所以困难在于它和应用程序原来设计所要解决的问题无任何关系。

2、数据库系统应避免数据相依 其理由有两条：

(1) 不同的应用程序需要用不同的观点看待同一数据。例如某数据库系统有两个应用程序A和B都使用同一字段“客户余额”，A使用十进制记录该值，B使用二进制记录。为消除冗余而将该字段集成并采用十进制物理地存放，则在B每次存取该字段时DBMS都要进行二——十进制之间的转换。B所看到的数据和它的物理存贮是不一致的。

(2) DBA必须有充分自由改变存贮结构及存取策略以适应变化了的环境而无需修改现有的应用程序。例如随环境发展变化，数据库需要增添新数据、需要采用新的标准化、需要改变应用程序的优先权、准备采用新型存贮器等。如果应用程序是数据相依的，上述改变自然要求应用程序也作相应改变，这要花费大量人力物力资源，是极其难以维护的。因此，数据独立性是数据库系统要实现的一个主要目标。

3、数据独立性的定义 在改变存贮结构和存取策略时不影响到应用程序叫数据独立性。第二章我们将通过对数据库系统结构的进一步说明来进一步完善数据独立性的