

计算机专业·任务驱动应用型教材

数据库基础

迟丽娟 余 宁 张继旺 主 编
朱道萌 蒲 陶 副主编

电子工业出版社版权所有
盗版必究

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书采用项目式设计，循序渐进地讲解数据库的基本原理和应用的方法。

全书共 8 个项目，具体内容为数据库系统概述、关系数据库、数据库设计、SQL 基础应用、SQL 高级应用、索引和视图、存储过程和触发器、安全管理。

本书实例丰富、内容翔实、操作方法简单易学，不仅适合作为高等职业院校计算机与大数据相关专业教材，也可供从事数据处理相关工作的专业人士参考。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

图书在版编目 (CIP) 数据

数据库基础 / 迟丽娟, 余宁, 张继旺主编. —北京: 电子工业出版社, 2023.1
ISBN 978-7-121-43875-2

I. ①数… II. ①迟… ②余… ③张… III. ①数据库系统 IV. ①TP311.13

中国版本图书馆 CIP 数据核字 (2022) 第 116938 号

责任编辑: 王艳萍 文字编辑: 张 彬

印 刷:

装 订:

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1 092 1/16 印张: 13.75 字数: 352 千字

版 次: 2023 年 1 月第 1 版

印 次: 2023 年 1 月第 1 次印刷

定 价: 49.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888, 88258888。

质量投诉请发邮件至 zlbs@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式: (010) 88254574, wangyp@phei.com.cn。

数据库技术是目前计算机相关行业中发展最快的技术之一，已经广泛应用于各种类型的数据处理系统之中，了解并掌握数据库知识已经成为对各类技术人员和管理人员的基本要求。“数据库基础”已逐渐成为各个专业本、专科学生的必修课程。

一、本书的特点

1. 实例丰富

本书结合大量的数据库制作与管理实例，详细讲解了数据库原理与应用要点，让读者在学习过程中潜移默化地掌握数据库制作与管理技能。

2. 突出提升技能

本书中的很多实例本身就是数据库开发项目，经过作者精心提炼和改编，不仅保证了读者能够学好知识点，更重要的是能够帮助读者掌握实际的操作技能，同时培养数据库开发实践能力。

3. 技能与思政教育紧密结合

在讲解数据库开发专业知识的同时，从专业知识角度触类旁通引导学生提升相关思政品质。

4. 项目式教学，实操性强

全书采用项目式设计，把数据库理论知识分解并融入一个个实践操作的训练项目中，增强了本书的实用性。

为与数据库语句保持一致，本书中的变量统一用正体表示。

二、本书的基本内容

全书共 8 个项目，具体内容为数据库系统概述、关系数据库、数据库设计、SQL 基础应用、SQL 高级应用、索引和视图、存储过程和触发器、安全管理。

本书由牡丹江大学的迟丽娟、河北科技学院的余宁、哈尔滨科学技术职业学院的张继旺担任主编，由贵州电子信息职业技术学院的朱道萌、四川长江职业学院的蒲陶担任副主编。

编 者

电子工业出版社版权所有
盗版必究

| | |
|----------------|----|
| 项目 1 数据库系统概述 | 1 |
| 任务 1 数据库系统 | 2 |
| 任务引入 | 2 |
| 知识准备 | 2 |
| 一、数据库系统的组成 | 2 |
| 二、数据库系统的体系结构 | 2 |
| 任务 2 数据库管理系统 | 4 |
| 任务引入 | 4 |
| 知识准备 | 4 |
| 一、DBMS 的主要功能 | 4 |
| 二、DBMS 的组成 | 5 |
| 任务 3 数据模型 | 6 |
| 任务引入 | 6 |
| 知识准备 | 7 |
| 一、数据模型的概念 | 7 |
| 二、概念模型 | 8 |
| 三、数据库类型 | 11 |
| 项目总结 | 14 |
| 项目 2 关系数据库 | 15 |
| 任务 1 关系代数 | 16 |
| 任务引入 | 16 |
| 知识准备 | 16 |
| 一、传统的集合运算 | 16 |
| 二、专门的关系运算 | 17 |
| 任务 2 关系数据库设计基础 | 18 |
| 任务引入 | 18 |
| 知识准备 | 19 |
| 一、关系模式中的问题 | 19 |
| 二、函数依赖 | 20 |

| | |
|------------------------------|-----------|
| 项目总结 | 21 |
| 项目 3 数据库设计 | 22 |
| 任务 1 数据库设计基础 | 23 |
| 任务引入 | 23 |
| 知识准备 | 23 |
| 一、需求分析 | 23 |
| 二、概念设计 | 28 |
| 三、逻辑设计 | 30 |
| 四、物理设计 | 31 |
| 任务 2 创建和使用数据库 | 31 |
| 任务引入 | 31 |
| 知识准备 | 31 |
| 一、系统数据库 | 31 |
| 二、创建数据库 | 33 |
| 三、数据库的属性设置 | 35 |
| 四、删除数据库 | 37 |
| 任务 3 创建和使用表 | 38 |
| 任务引入 | 38 |
| 知识准备 | 38 |
| 一、创建表 | 38 |
| 二、修改表的结构 | 40 |
| 三、建立表间关系 | 41 |
| 四、删除表 | 43 |
| 五、新增和修改记录 | 44 |
| 项目总结 | 46 |
| 项目实战 | 47 |
| 实战一 创建一个名为 factory 的数据库 | 47 |
| 实战二 在 factory 数据库中创建表 | 48 |
| 实战三 为 factory 数据库中的 3 个表建立关系 | 48 |
| 实战四 为 factory 数据库中的 3 个表添加记录 | 49 |
| 项目 4 SQL 基础应用 | 51 |
| 任务 1 操作语句 | 52 |
| 任务引入 | 52 |
| 知识准备 | 52 |
| 一、数据库的操作语句 | 52 |
| 二、表的操作语句 | 57 |
| 任务 2 数据查询 | 60 |
| 任务引入 | 60 |
| 知识准备 | 61 |

| | |
|-------------------------------------|-----|
| 一、投影查询 | 61 |
| 二、选择查询 | 63 |
| 三、排序查询 | 64 |
| 四、聚合函数 | 65 |
| 五、表的连接查询 | 67 |
| 六、子查询 | 71 |
| 七、相关子查询 | 75 |
| 八、带 EXISTS 测试的子查询 | 76 |
| 任务 3 SQL 程序设计基础 | 78 |
| 任务引入 | 78 |
| 知识准备 | 78 |
| 一、标识符 | 78 |
| 二、数据类型 | 81 |
| 三、运算符 | 92 |
| 四、变量 | 98 |
| 五、批处理 | 100 |
| 六、注释 | 101 |
| 七、控制流语句 | 102 |
| 八、函数 | 110 |
| 项目总结 | 113 |
| 项目实战 | 114 |
| 实战一 查询工资为 5000~6000 元的职工的职工号、姓名和部门号 | 114 |
| 实战二 输出 11 号职工的平均工资 | 114 |
| 项目 5 SQL 高级应用 | 116 |
| 任务 1 高级查询 | 117 |
| 任务引入 | 117 |
| 知识准备 | 117 |
| 一、数据汇总 | 117 |
| 二、连接查询 | 120 |
| 三、子查询 | 125 |
| 四、在查询的基础上创建新表 | 130 |
| 任务 2 事务处理 | 131 |
| 任务引入 | 131 |
| 知识准备 | 131 |
| 一、事务分类 | 132 |
| 二、显式事务 | 132 |
| 三、自动提交事务 | 136 |
| 四、隐式事务 | 136 |
| 任务 3 游标 | 138 |

| | |
|---|-----|
| 任务引入 | 138 |
| 知识准备 | 138 |
| 一、使用游标 | 139 |
| 二、游标类型 | 144 |
| 项目总结 | 145 |
| 项目实战 | 145 |
| 实战一 插入一条职工记录, 进行完整外部连接、显示, 然后删除记录 | 145 |
| 实战二 启动一个事务, 向 depart 表中插入一条记录, 然后回滚该事务 | 146 |
| 项目 6 索引和视图 | 148 |
| 任务 1 索引 | 149 |
| 任务引入 | 149 |
| 知识准备 | 149 |
| 一、索引类型 | 149 |
| 二、创建索引 | 153 |
| 三、查看和删除索引 | 157 |
| 任务 2 视图 | 158 |
| 任务引入 | 158 |
| 知识准备 | 158 |
| 一、创建视图 | 159 |
| 二、使用视图 | 162 |
| 三、删除视图 | 166 |
| 项目总结 | 167 |
| 项目实战 | 168 |
| 实战一 为 salary 表的“职工号”和“日期”列创建聚集索引, 并且强制唯一性 | 168 |
| 实战二 创建 View1 视图, 查询所有职工的职工号、姓名、部门号和工资 | 168 |
| 项目 7 存储过程和触发器 | 170 |
| 任务 1 存储过程 | 171 |
| 任务引入 | 171 |
| 知识准备 | 171 |
| 一、创建存储过程 | 171 |
| 二、执行存储过程 | 173 |
| 三、存储过程的参数 | 174 |
| 四、查看和删除存储过程 | 176 |
| 任务 2 触发器 | 177 |
| 任务引入 | 177 |
| 知识准备 | 178 |
| 一、创建触发器 | 178 |
| 二、使用触发器 | 181 |
| 三、修改触发器 | 183 |

| | |
|--|------------|
| 四、删除触发器 | 183 |
| 项目总结 | 184 |
| 项目实战 | 184 |
| 实战一 创建一个为 worker 表添加职工记录的存储过程 Addworker | 184 |
| 实战二 在 worker 表上创建一个触发器 worker_delete，当删除职工记录时同步删除 salary 表中对应职工的工资记录 | 184 |
| 项目 8 安全管理 | 186 |
| 任务 1 账号和角色 | 187 |
| 任务引入 | 187 |
| 知识准备 | 187 |
| 一、服务器的登录账号 | 187 |
| 二、数据库的用户账号 | 190 |
| 三、角色 | 192 |
| 任务 2 备份和还原数据库 | 196 |
| 任务引入 | 196 |
| 知识准备 | 197 |
| 一、备份数据库 | 197 |
| 二、还原数据库 | 202 |
| 项目总结 | 208 |
| 项目实战 | 208 |
| 实战一 创建一个登录账号 ABC，密码为 123456 | 208 |
| 实战二 在 factory 数据库中创建一个用户账号，并将其关联到 ABC 登录账号中 | 208 |

数据库系统概述

思政目标

- 了解数据库系统，对其组成、体系结构、管理等有较清楚的认识
- 从基础入手，培养探索新知识的习惯

技能目标

- 了解数据库系统的组成及体系结构
- 了解数据库管理系统的主要功能和组成
- 能够理解数据间的联系和实体间的联系

项目导读

数据库是一门研究数据管理的技术,始于 20 世纪 60 年代末,经过 50 多年的发展,已形成理论体系,成为计算机软件的一个重要分支。数据库技术主要研究如何存储、使用和管理数据。

任务 1 数据库系统

| 任务引入 |

小明是一名大三的学生，这个学期有 C 语言、数据库、Python 等关于计算机的选修课。小明对数据库比较感兴趣，决定选修数据库课程。那么，数据库系统是由什么组成的？数据库系统的体系结构是什么样的？

| 知识准备 |

通常把引进了数据库技术的计算机系统称为数据库系统，它的作用是存储和产生所需要的有用信息。这些有用的信息可以是对使用该系统的个人或组织有意义的任何信息，换句话说，是对某个人或组织进行决策不可少的辅助信息。

一、数据库系统的组成

数据库系统由数据库、硬件、数据库管理系统、应用程序、人员等组成。

- 数据库：存储在外存上的若干设计合理、能满足应用需要的结构化的数据集合。
- 硬件：数据库赖以存在的物理设备，包括 CPU、存储器、其他外部设备等。数据库系统需要足够大的内存和外存，用来运行操作系统、数据库管理系统和应用程序，以及存储数据库。
- 数据库管理系统（Database Management System, DBMS）：帮助用户创建、维护和使用数据库的软件系统，是数据库系统的核心。常用的 DBMS 有 Access、Visual FoxPro、SQL Server 等。
- 应用程序：数据库是多用户共享的，不同用户的数据视图由设计者组织在数据库中，但是如何使用是用户自己的事，用户可以在远程终端上查询数据，也可以通过编程处理自己的业务。
- 人员：包括数据库管理员（Database Administrator, DBA）和用户。在大型数据库系统中，需要有专人负责数据库系统的建立、维护和管理工作，承担该任务的人员称为数据库管理员。用户分为两类：专业用户和最终用户。专业用户侧重于设计数据库、开发应用系统程序，为最终用户提供友好的用户界面。最终用户侧重于使用数据库，主要通过数据库进行联机查询，或者通过数据库应用系统提供的界面使用数据库。

二、数据库系统的体系结构

数据库系统有着严谨的体系结构。目前，全球有大量的数据库正在运行中，其类型

和规模可能相差很大，但是就其体系结构而言都是大体相同的。

1. 数据库系统的 3 级模式结构

将数据库的组织从内到外分成 3 个层次进行描述，分别称为内模式、概念模式和外模式，如图 1-1 所示。

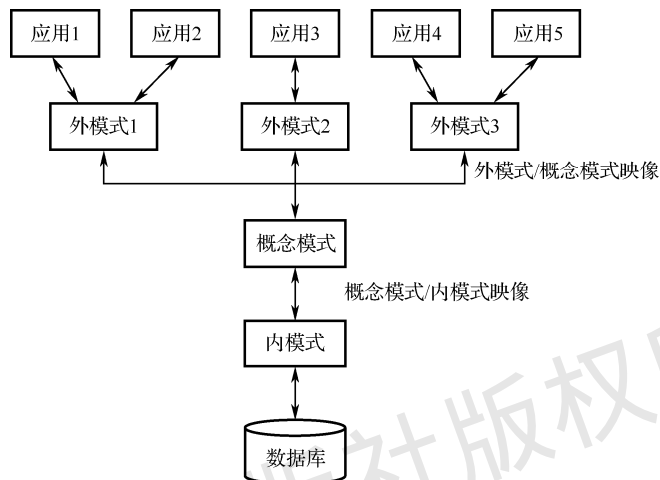


图 1-1 3 级模式结构

- 内模式：又称存储模式，具体描述了数据如何组织存储在存储介质中。内模式是系统程序员用一定的文件形式组织起来的存储文件和联系手段，也是用来实现数据存取的。一个数据库只有一个内模式。
- 概念模式：简称模式，又称逻辑模式，是对数据库的整体逻辑结构和特征的描述，是所有用户的公共数据视图。它是数据库系统模式结构的中间层，并不涉及数据的物理存储细节和硬件环境，与具体的应用程序和应用开发工具无关。
- 外模式：通常是概念模式的一个子集，故又称子模式。外模式面向用户，是数据库用户能够看到和使用局部数据的逻辑结构和特征描述，是与某个应用有关的数据的逻辑表示。

2. 3 级模式之间的映像

前面谈到的 3 级模式，只有内模式才是真正存储数据的，概念模式和外模式仅是一种逻辑上的表示数据的方法，但却可以放心大胆地使用它们，这都是依靠 DBMS 的映像功能实现的。

数据库系统的 3 级模式是对数据的 3 个抽象级别，它把数据的具体组织留给 DBMS 管理，使用户能逻辑地、抽象地处理数据，而不必关心数据在计算机中的具体表示方式与存储方式。为了能够在内部实现这 3 个抽象层次的联系和转换，DBMS 在 3 级模式之间提供了两层映像：

- 外模式/概念模式映像；
- 概念模式/内模式映像。

正是这两层映像保证了数据库系统中的数据具有较高的逻辑独立性和物理独立性。

任务 2 数据库管理系统

| 任务引入 |

小明已经对数据库系统有了大概的了解，知道了数据库系统的组成。但是，怎么能把数据库、用户和应用程序联系到一起呢？

| 知识准备 |

DBMS 是数据库系统的关键组成部分。任何对数据的操作，包括数据库定义、数据查询、数据维护、数据库运行和控制等都是在 DBMS 中进行的。DBMS 是用户与数据库的接口，应用程序只有通过 DBMS 才能和数据库打交道。

一、DBMS 的主要功能

通常，DBMS 的主要功能包括以下几个方面。

1. 数据库定义

DBMS 提供数据定义语言 (Data Definition Language, DDL) 来定义数据库的 3 级模式。用概念 DDL 编写的概念模式称为源概念模式，用外 DDL 编写的外模式称为源外模式，用内 DDL 编写的内模式称为源内模式。各种源模式可以通过相应的模式翻译程序转换为机器内部代码形式，分别称为目标概念模式、目标外模式和目标内模式。这些目标模式是对数据库结构信息的描述，而不是对数据本身的描述，它们刻画了数据库的框架（结构），被保存在数据字典（又称系统目录）中。数据字典是 DBMS 存取和管理数据的基本依据。例如，DBMS 根据数据字典中的定义，从存储记录中导出全局逻辑记录，又从全局逻辑记录中导出用户所要存取的记录。

2. 数据操纵

DBMS 提供数据操纵语言 (Data Manipulation Language, DML) 实现对数据库数据的操纵功能，包括检索、插入、修改和删除。检索就是查询，是重要的、经常使用的一种操作，所以有些系统把 DML 称为查询语言。插入、修改和删除有时又称更新操作。

DML 有两类：一类是交互式命令语言，语法简单，可独立使用，所以称为自主型或自含型语言；另一类是把数据库存取语句嵌入主语言 (Host Language) 中，如嵌入 Fortran、Pascal 或 C 等高级语言中使用，DML 本身不能独立使用，因此称为宿主型语言。现代的 DBMS 多具有自含型与宿主型两类 DML，允许用户选择使用。著名的结构化查询语言 (Structured Query Language, SQL) 就是一种 DML。

DBMS 控制并执行 DML 语句，完成对数据库的操作。对于自含型语言，DBMS 通常用解释执行的方式；对于宿主型语言，DBMS 一般提供两种方法。

- 预编译：DBMS 提供预编译程序功能，可以对源程序进行扫描，识别嵌入的 DML 语句并把它们转换成主语言代码（或主语言调用语句），以便被主语言编译程序接受和执行。
- 修改、扩充主语言编译程序（又称增强编译）。

DDL 和 DML 是数据库的用户在设计应用程序时必须用的程序设计语言的一个子集，称为数据子语言。

3. 数据库运行控制和管理

DBMS 提供数据库的运行控制和管理功能，即对数据的安全性、完整性和并发控制等进行有效的控制和管理，以确保数据库数据的正确有效和数据库系统的有效运行。

- 数据的安全性（Security）控制：采取一定的安全保密措施以确保数据库中的数据不被非法用户存取。DBMS 提供口令或其他手段来检查用户身份，合格用户才能进入数据库系统；提供用户密级和数据存取权限的定义机制，系统自动检查用户能否执行这些操作，只有检查通过后才能执行被允许的操作。
- 数据的完整性（Integrity）控制：DBMS 提供必要的功能以确保数据库中的数据正确性、有效性和相容性。
- 数据的并发（Concurrency）控制：DBMS 必须对多用户并发进程同时存取、修改数据的操作进行控制和协调，以防止互相干扰，导致结果错误。

4. 数据库建立和维护

DBMS 提供数据库初始数据的装入，数据库的转储、恢复、重组织，系统性能监视、分析等功能。

5. 数据库传输

DBMS 处理数据的传输，可以实现用户程序与 DBMS 之间的通信，通常与操作系统协调完成。

二、DBMS 的组成

DBMS 大多是由许多系统程序所组成的一个集合。每个程序都有自己的功能，一个或几个程序一起完成 DBMS 的一件或几件工作。各种 DBMS 的组成因系统而异，一般来说，由以下几个部分组成。

1. 语言编译处理程序

语言编译处理程序主要包括以下几种。

- 数据定义语言（外 DDL、概念 DDL、内 DDL）翻译程序：把各级源模式翻译成各级目标模式。
- 数据操纵语言处理程序：将应用程序中的 DML 语句转换成可执行程序。
- 终端命令解释程序：解释执行每个终端命令。例如，用户在 Visual Foxpro 的命令窗口中输入的命令都需通过 Visual Foxpro 系统的终端命令解释程序来执行，并将

执行结果输出到主窗口中。

- 数据库控制命令解释程序：解释执行每个控制命令。

2. 系统运行控制程序

系统运行控制程序主要包括以下几种。

- 系统总控程序：DBMS 运行程序的核心，控制和协调各程序的活动。
- 存取控制程序：检查用户（或应用程序）是否合法。
- 并发控制程序：协调各应用程序对数据库的操作，保证数据的一致性。
- 完整性控制程序：检查完整性约束条件，决定是否执行对数据库的操作。
- 保密性控制程序：实现对数据库数据安全的保密控制。
- 数据存取和更新程序：实施对数据库数据的检索，执行插入、修改、删除等操作。

3. 系统建立和维护程序

系统建立和维护程序主要包括以下几种。

- 数据装入程序：完成初始数据库的数据装入。
- 数据库重组程序：当数据库性能变坏（如查询速度变慢、时间超过规定值）时，需要重新组织数据库，可按原组织方法重新装入数据（或采用新方法和新结构）。一般来说，重组是数据库系统的一项周期性活动。
- 数据库系统恢复程序：当数据库系统受到破坏时，用来将数据库系统恢复到可用状态。
- 性能监督程序：监督用户的操作执行时间与数据存储空间占用情况，做出系统性能估算，从而决定数据库是否需要重组。
- 工作日志程序：记载数据库的所有存取过程，包括用户名、进入时间、操作方式、数据对象、修改前数据、修改后数据等，使每个存取过程都留下痕迹。

4. 数据字典

数据字典又称数据目录或系统目录，通常是一系列表（对于关系数据库来讲，是一系列二维表），存储着数据库中有关信息的当前描述，包括数据库 3 级模式、数据类型、用户名表、用户权限、程序与其用户联系等有关数据库系统的信息，起着系统状态目录表的作用，能帮助用户、数据库管理员和数据库管理系统本身使用和管理数据库。

任务 3 数据模型

| 任务引入 |

小明通过学习，知道了 DBMS 是用户与数据库的接口，应用程序只有通过 DBMS 才能和数据库打交道。但是，知道了 DBMS 还不能建立数据库，还得了解数据库的组织

形式，即数据库的框架。那么，数据库的框架是什么？数据间如何联系？实体间的联系方式和表示方法是什么？

| 知识准备 |

数据模型是数据库的框架，这个框架描述了数据库的数据组织形式。数据模型是定义数据库的依据，现有的数据库系统均是基于某种数据模型的。因此，了解数据模型是学习数据库的基础。

一、数据模型的概念

数据模型是客观事物及其联系的数据描述，应具有描述数据和数据联系两方面的功能。组成数据模型的三要素是数据结构、数据操作和数据的约束条件。其中，数据结构是研究的记录类型的集合，是对系统静态特性的描述；数据操作是对数据库中各种对象（型）的实例（值）允许执行的操作的集合；数据的约束条件是一组完整性规则的集合。所谓完整性规则是指数据模型中的数据及其联系所具有的制约和依存规则，用于限定符合数据模型的数据库状态和状态变化，以保证数据的正确、有效、相容。

数据模型可以表示为如下形式：

$$DM=\{R,L\}$$

其中，DM（Data Model）是数据模型的英文简称；R 代表记录类型的集合；L 代表不同记录类型的联系的集合。

例如，在学生选课问题中，R 是学生和课程两个记录类型的集合，L 是它们之间的联系，即“选修”联系。

不同的数据模型实际上是提供模型化数据和信息的不同工具。根据模型应用的不同目的，可以将这些模型划分为两类，分属于两个不同的层次。

第一类模型是概念模型，又称信息模型，其按用户的观点来对数据和信息进行建模，主要用于数据库设计。另一类模型是数据模型，主要包括网状模型、层次模型、关系模型等，其按计算机系统的观点对数据进行建模，主要用于 DBMS 的实现。

数据模型是数据库系统的核心和基础。各种机器上的 DBMS 软件都是基于某种数据模型实现的。

1. 对数据的描述

对数据的描述应指出模型中包含哪些记录类型，并对记录类型进行命名；指明各个记录类型由哪些数据项构成，并对数据项进行命名，每个数据项均需指明其数据类型和取值范围，这是数据完整性约束所必需的。

例如，在学生选课问题中，学生记录类型是由学号、姓名、性别、班号等数据项组成的，课程记录类型是由课程号、课程名、任课教师等数据项组成的。例如，学号由长度为 10 的字符型数据构成，性别只能为“男”或“女”。

2. 对数据间联系的描述

对数据间联系的描述要指明各个不同记录类型间所存在的联系和联系方式。数据模型中的“联系”是一种特殊类型记录，通常还要对这种“联系”进行命名。数据库系统与文件系统本质上的不同就表现在数据库中的各条记录是互相联系的，正是通过这种联系，数据库才能支持访问不同记录类型的数据。

例如，在学生选课问题中，“选修”联系将多条学生记录与多条课程记录关联起来，即多名学生可以选修同一门课程，一名学生也可以选修多门课程。

数据模型仅是一种形式化描述记录类型及其联系的方法，与具体的 DBMS 无关。当选定某种 DBMS 时，根据数据模型，用它的 DDL 加以描述，就定义了数据库，即确定了 3 级模式。

二、概念模型

计算机只能处理数据，所以首先要解决的问题是按用户的观点对数据和信息进行建模，然后按计算机系统的观点对数据进行建模。换句话说，就是现实世界的问题如何表达为信息世界的问题，以及信息世界的问题如何在具体的机器世界表达出来。如图 1-2 所示为现实世界中客观对象的抽象过程。

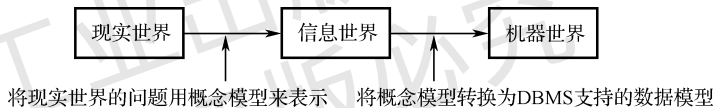


图 1-2 现实世界中客观对象的抽象过程

1. 信息世界中的基本概念

信息世界涉及的主要概念如下。

(1) 实体 (Entity)

实体是指客观存在并可相互区别的事物，可以是具体的人、事、物，也可以是抽象的概念或联系。

例如，一位教师、一名学生、一个部门、一门课、学生的一次选课、教师与系的工作关系（某位教师在某系工作）等都是实体。

(2) 属性 (Attribute)

属性是指实体所具有的某个特性。一个实体可以由若干属性来刻画。属性是相对实体而言的，是实体所具有的特征。

例如，学生实体可以由学号、姓名、性别、出生日期、系、入学时间等属性组成，将这些属性组合起来，如(202131500180,刘凯,男,05/02/2000,计算机系,2021)，即表示一名学生。

(3) 码 (Key)

码又称关键字，是指在实体属性中，可用于区别实体中不同个体的一个或几个属性的组合，称为该实体的码。

例如，在“学生”实体中，能作为码的属性可以是“学号”，因为一旦码有一个取值，

便唯一地标识了实体中的某个个体；当然“姓名”也可作为码，但如果出现重名现象，“姓名”这个属性就不能作为码了。当有多个属性可作为码而选定其中一个时，则称它为该实体的主码。若在实体诸属性中，某属性虽非该实体的主码，却是另一个实体的主码，则称此属性为外部码，简称外码。

(4) 域 (Domain)

域是指属性的取值范围。

例如，学号的域为12位整数，姓名的域为长度不大于10个字符的字符串集合，性别的域为(男,女)。

(5) 实体型 (Entity Type)

实体型是指用实体名及其属性名集合来抽象和刻画同类实体。具有相同属性的实体必然具有共同的特征和性质。实体型是概念的内涵，而实体值是概念的实例。

例如，学生(学号,姓名,性别,出生日期,系,入学时间)就是一个实体型，它通过学号、姓名、性别、出生日期、系、入学时间等属性来表明学生情况。而每名学生的具体情况，则称为实体值。可见，实体型表达的是个体的共性，而实体值是个体的具体内容。通常，实体型是变量，实体值是该变量的取值。

(6) 实体集 (Entity Set)

实体集是指同型实体的集合。

例如，全体学生就是一个实体集。实体集名是实体型。学生和(学号,姓名,年龄,系,年级)均是实体型，而学生是给实体型(学号,姓名,年龄,系,年级)所起的名称，两者指同一客观对象。本科生和研究生可以是相同的实体型，但实体集不同。

2. 实体间的联系方式

在现实世界中，事物内部以及事物之间是有联系的，这些联系在信息世界中反映为实体内部的联系和实体之间的联系。实体内部的联系通常是指组成实体的各属性之间的联系。实体之间的联系通常是指不同实体集之间的联系。

两个实体集之间的联系可以分为以下3种。

(1) 一对一 (1:1) 联系

若对于实体集A中的每个实体，实体集B中至多有一个（也可以没有）实体与之联系，反之亦然，则称实体集A与实体集B具有一对一联系，记为1:1。

例如，在学校里面，一个班中只有一个班长，而一个班长只在一个班中任职，则班级与班长之间具有一对一联系。

(2) 一对多 (1:n) 联系

若对于实体集A中的每个实体，实体集B中有n个实体 ($n \geq 0$) 与之联系；对于实体集B中的每个实体，实体集A中至多有一个实体与之联系，则称实体集A与实体集B具有一对多联系，记为1:n。

例如，一个班中有若干学生，而每名学生只在一个班中学习，则班级与学生之间具有一对多联系。

(3) 多对多 (m:n) 联系

若对于实体集A中的每个实体，实体集B中有n个实体 ($n \geq 0$) 与之联系；对于实体集B中的每个实体，实体集A中有m个实体 ($m \geq 0$) 与之联系，则称实体集A与实

体集 B 具有多对多联系，记为 $m:n$ 。

例如，一门课程同时有若干学生选修，而一名学生可以同时选修多门课程，则课程与学生之间具有多对多联系。

3. 实体—联系表示法

概念模型中常用的方法称为实体—联系方法，简称 E-R 方法。该方法直接从现实世界中抽象出实体和实体间的联系，然后用 E-R 图来表示数据模型。在 E-R 图中，实体用方框表示；联系用菱形表示，并且用边将其与有关的实体连接起来，并在边上标注联系的类型；属性用椭圆表示，并且用边将其与相应的实体连接起来。对于有些联系，其自身也会有某些属性，和实体与属性的连接方法类似，用边将联系与其属性连接起来。

需要说明的是，E-R 方法是软件工程设计中的一个重要方法，因为它接近人的思维方式，容易理解并且与计算机无关，所以用户容易接受。但是，E-R 方法只能说明实体间的语义联系，还不能进一步说明详细的数据结构。一般遇到实际问题时，应先设计一个 E-R 图，再把它转换成计算机能接受的数据模型。

(1) 两个不同实体集之间联系的画法

两个不同实体集之间存在 $1:1$ 、 $1:n$ 和 $m:n$ 联系，可以用图形来表示两个实体集之间的这 3 种联系，如图 1-3 所示。

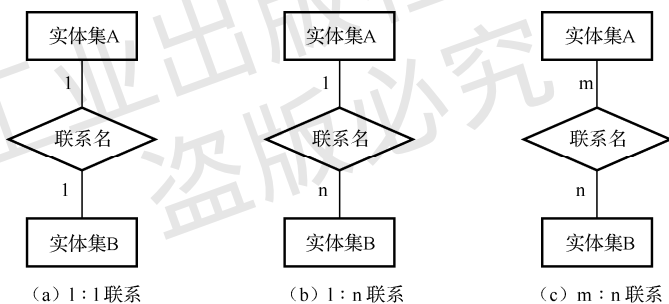


图 1-3 两个不同实体集之间的 3 种联系

(2) 两个以上不同实体集之间联系的画法

两个以上不同实体集之间可能存在各种联系，以 3 个不同实体集 A、B 和 C 为例，它们之间的典型联系有 $1:n:m$ 和 $r:n:m$ 联系。对于 $1:n:m$ 联系，表示 A 和 B 之间是 $1:n$ （一对多）联系，B 和 C 之间是 $n:m$ （多对多）联系，A 和 C 之间是 $1:m$ （一对多）联系。 $r:n:m$ 联系的表示与上述表示类似。这两个典型关系的表示方法如图 1-4 所示。

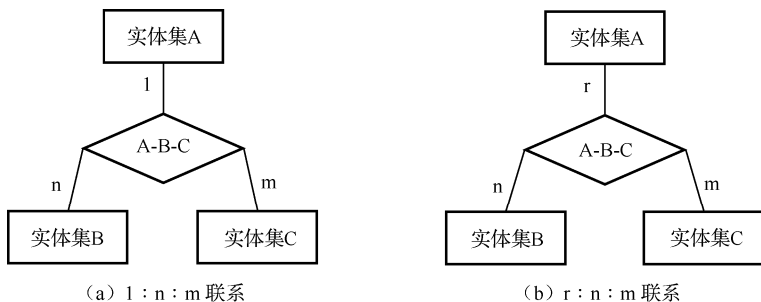


图 1-4 3 个不同实体集之间的 $1:n:m$ 和 $r:n:m$ 联系

(3) 同一实体集内的二元联系的画法

同一实体集内的二元联系表示实体集中实体之间的相互联系，同样有 1:1、1:n 和 n:m 联系。例如，职工实体集内的领导与职工的联系可以是 1:n 的，而职工实体集内的婚姻联系是 1:1 的。同一实体集内的 1:1、1:n 和 n:m 联系如图 1-5 所示。

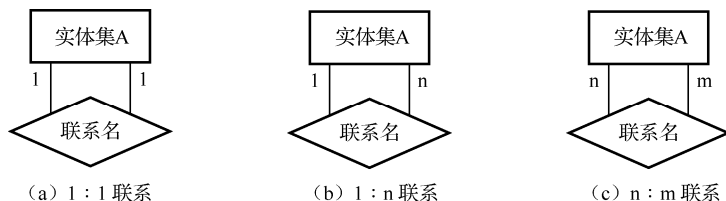


图 1-5 同一实体集内的 1:1、1:n 和 n:m 联系

注意：一个系统的 E-R 图不是唯一的，从不同的角度出发画出的 E-R 图可能很不同。总体 E-R 图所表示的实体—联系模型只能说明实体间的联系，还需要把它转换成数据模型才能被实际的 DBMS 所接受。

三、数据库类型

数据库有类型之分，是根据数据模型划分的，而任何一个 DBMS 也是根据数据模型有针对性地设计出来的。这意味着必须把数据库组织成符合 DBMS 规定的数据库模型。目前成熟地应用于数据库系统中的数据模型有层次模型、网状模型和关系模型。它们之间的根本区别在于数据之间联系的表示方式不同（记录型之间的联系方式不同）。层次模型用“树结构”来表示数据之间的联系。网状模型用“图结构”来表示数据之间的联系。关系模型用“二维表”（或称关系）来表示数据之间的联系。

1. 层次模型

层次模型是数据库系统最早使用的一种模型，它的数据结构是一棵“有向树”。层次模型的特征如下。

- 有且仅有一个节点（根节点），没有父节点。
- 其他节点有且仅有一个父节点。

在层次模型中，每个节点描述一个实体型，称为记录型。一条记录型可有許多记录值，简称记录。节点之间的有向边表示记录之间的联系。如果要存取某个记录型的记录，可以从根节点开始，按照有向树的层次逐层向下查找，查找路径就是存取路径。

例如，图 1-6 所示为一个系教务管理层次模型，图 1-6 (a) 表示实体之间的联系，图 1-6 (b) 表示实体型之间的联系。图 1-7 所示为一个系教务管理层次模型实例。

2. 网状模型

用网状结构表示实体和实体之间联系的模型称为网状模型。网中的每个节点代表一条记录类型，联系用链接指针来实现。广义地讲，任何一个连通的基本层次联系的集合都是网状模型。它取消了层次模型的两点限制。网状模型的特征如下。

- 允许节点有多于一个的父节点。
- 可以有一个以上的节点没有父节点。

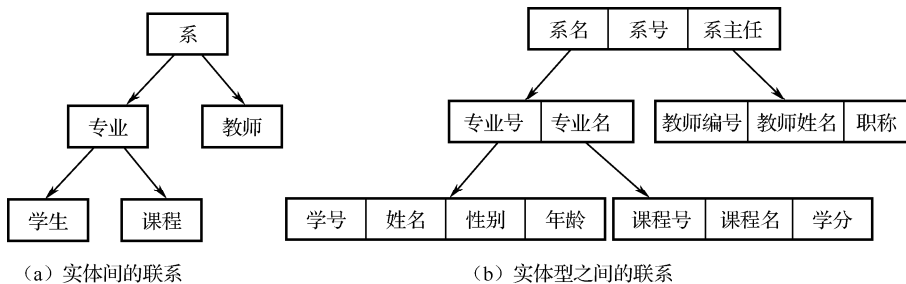


图 1-6 系教务管理层次模型

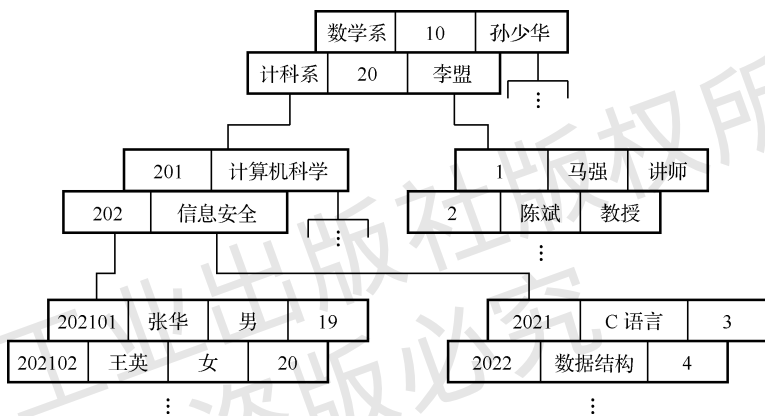


图 1-7 系教务管理层次模型实例

图 1-8 所示为一个简单的网状模型示例，其中图 1-8 (a) 是学生选课 E-R 图；图 1-8 (b) 中，S 表示学生记录型，C 表示课程记录型，联系记录型 L 表示 S 和 C 之间的一个多对多的选修联系。图 1-9 所示为一个具体实例，其中 C 记录有一个指针，指向该课程号的第一个 L 记录。L 记录有两个指针，第一个指针指向下一个同课程号的 L 记录，第二个指针指向下一个学号的 L 记录。S 记录有一个指针，指向该学号的第一个 L 记录。这里构成的单链表均为循环单链接，用这些链表指针可以实现联系。

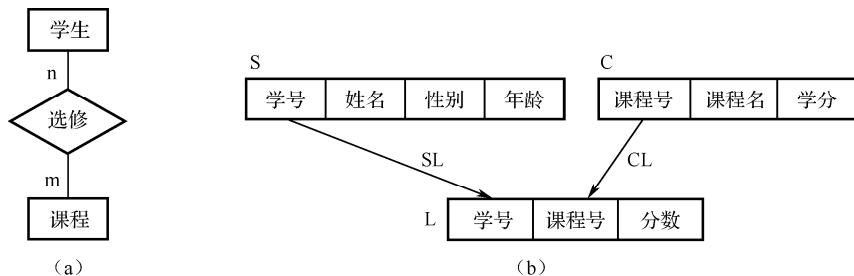


图 1-8 学生选修课网状模型

网状模型和层次模型在本质上是一样的。从逻辑上看，它们都是基本层次联系的集合，用节点表示实体，用有向边（箭头）表示实体间的联系。从物理上看，它们的每个

节点都是一个存储记录，用链接指针来实现记录之间的联系。当存储数据时，这些指针就固定下来了，所以进行数据检索时必须考虑存取路径问题，进行数据更新时，会涉及链接指针的调整，缺乏灵活性，所以系统扩充相当麻烦。网状模型中的指针相对层次模型更多，纵横交错，从而使数据结构更加复杂。

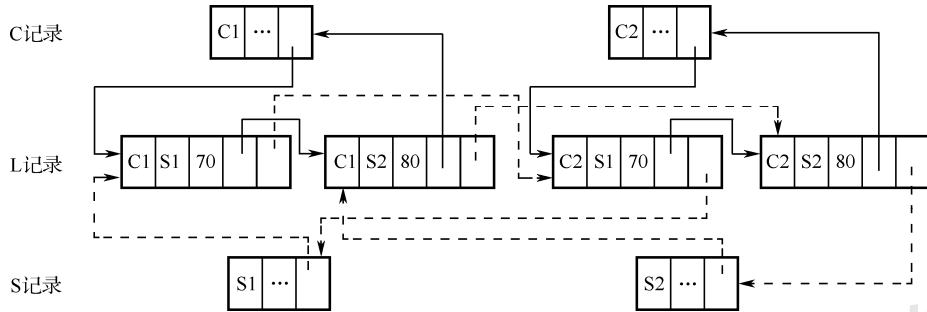


图 1-9 学生选修课网状模型实例

3. 关系模型

关系模型是用二维表结构来表示实体和实体之间联系的数据模型。关系模型的数据结构是一个由“二维表框架”组成的集合，每个二维表被称为关系，因此关系模型是“关系框架”组成的集合。目前大多数 DBMS 都是关系型的。例如，Visual FoxPro 就是一种关系数据库管理系统，支持关系数据模型。

图 1-10 给出了一个简单的关系模型，其中图 1-10 (a) 给出了关系模式：

教师(教师编号,姓名,性别,所在系名)

课程(课程号,课程名,教师编号,上课教室)

图 1-10 (b) 给出了这两个关系模式的关系，关系名分别为教师关系和课程关系，均包含两个元组。教师关系中，教师编号为主码；课程关系中，课程号为主码。

教师关系框架

| 教师编号 | 姓名 | 性别 | 所在系名 |
|------|----|----|------|
|------|----|----|------|

课程关系框架

| 课程号 | 课程名 | 教师编号 | 上课教室 |
|-----|-----|------|------|
|-----|-----|------|------|

(a)

教师关系

| 教师编号 | 姓名 | 性别 | 所在系名 |
|------|-----|----|-------|
| 001 | 王丽华 | 女 | 计算机系 |
| 008 | 孙军 | 男 | 电子工程系 |

课程关系

| 课程号 | 课程名 | 教师编号 | 上课教室 |
|------|------|------|-------|
| 21-1 | 软件工程 | 001 | 5-301 |
| 21-3 | 电子技术 | 008 | 2-205 |

(b)

图 1-10 关系模型实例

关系模型的特征如下。

- 描述的一致性，不仅用关系描述实体本身，还用关系描述实体之间的联系。
- 可直接表示多对多联系。
- 关系必须是规范化的，即每个属性是不可分的数据项，不允许表中有表。
- 关系模型是建立在数学概念基础上的，有较强的理论根据。

在关系模型中，基本数据结构就是二维表，不像层次模型或网状模型那样是链接指针。记录之间的联系是通过不同关系中的同名属性得到的。例如，要查找“王丽华”老师所上课程，首先要在教师关系中根据姓名找到教师编号“001”，然后在课程关系中找到教师编号“001”对应的课程名。在上述查询过程中，同名属性的教师编号是连接两个关系的纽带。由此可见，关系模型中的各个关系模式不应当被孤立，它们不是随意拼凑的一堆二维表，必须满足相应的要求。

关系是一张二维表，关系框架是一个关系的属性名表，可以表示为如下形式：

$R(A_1, A_2, \dots, A_n)$

其中，R 为关系名； A_i ($i=1, 2, \dots, n$) 为关系的属性名。

关系之间通过公共属性实现联系。例如，图 1-10 给出的两个关系，通常情况下需要增加如下任课关系：

任课(教师编号,姓名,课程号,上课教室)

该关系与前两个关系都有公共属性，从而实现关系之间的联系。

关系数据库指对应于一个关系模型的所有关系的集合。例如，在一个学生课程管理关系数据库中，包含教师关系、课程关系、学生关系、任课关系等。

项目总结

