

数据管理基础

第1章 绪论

智能软件与工程学院



第1章 绪论

- 当今，信息资源已经成为社会的重要财富和资源，计算机信息系统也已成为一个企业或组织生存和发展的重要条件。
- 作为信息系统核心和基础的数据库技术得到越来越广泛的应用。

从 **联机事务处理**(*On-Line Transaction Processing, OLTP*)
到 **联机分析处理**(*On-Line Analysis Processing, OLAP*)

从一般**科学计算**到**企业资源计划(ERP)**、**计算机辅助设计与制造(CAD/CAM)**、**计算机集成制造系统(CIMS)**、**电子政务(e-Government)**、**电子商务(e-Commerce)**、**地理信息系统(GIS)** ...

第1章 绪论

- ❑ 数据库是数据管理的有效技术，是计算机科学的一个重要分支。
- ❑ 数据库课程不仅是计算机类专业、信息管理专业的重要课程，也是许多非计算机专业的选修课程。

❑ 本章主要内容

- 数据库的基本概念
- 数据模型的组成要素 和 常用数据模型
- 数据库的三级模式 和 数据库系统的主要组成部分

1.1 数据库系统概述

1.2 数据模型

1.3 数据库系统的结构

1.4 数据库系统的组成

1.5 小结

1.1 数据库系统概述

1.1.1 数据库的四个基本概念

1.1.2 数据管理技术的产生和发展

1.1.3 数据库系统的特点

1.1.1 数据库的四个基本概念

❑ 数据 (data)

❑ 数据库 (DataBase, DB)

❑ 数据库管理系统 (DataBase Management System, DBMS)

❑ 数据库系统 (DataBase System, DBS)

➤ 数据库用户 (Database User)

➤ 数据库管理员 (DataBase Administrator, DBA)

➤ 数据库应用系统 (DataBase Application System)

1. 数据 (data)

- ‘数据’是指具有一定的语义 (semantic) 含义，并且可以被记录下来的已知信息。
- 什么是‘语义’？
 - 在计算机中，数据被表示为具有一定格式或结构的符号串（描述事物的符号串）。
 - 单纯的符号没有任何意义，需要给予解释，说明其含义。数据的含义即‘语义’。
 - 只有被赋予确定含义后，‘数据’才能变成人们可用的‘信息’。

□ 什么是‘信息’ (information)?

- 信息是用来消除随机不确定性的东西。[Shannon]
- 信息是人们在适应外部世界，并使这种适应反作用于外部世界的过程中，同外部世界进行互相交换的内容和名称。[Norbert Wiener]
- 信息是对客观世界中各种事物的物理状态及其变化的反映，是客观事物之间相互联系和相互作用的表征，表现的是客观事物运动状态和变化的实质内容。

□ 例如，在计算机中有这样一个符号串

(李明, 男, 199505, 江苏省南京市, 计算机系, 2013)

如果没有附加的语义解释和定义，这样的符号串是无法使用的。

数据 & 信息 (2)

- ❑ 现实世界中的信息是多种多样的！
- ❑ 可以从中抽取出人们感兴趣的内容，抽象成为计算机中的‘数据’。

❑ 例：

数据的语义

数据



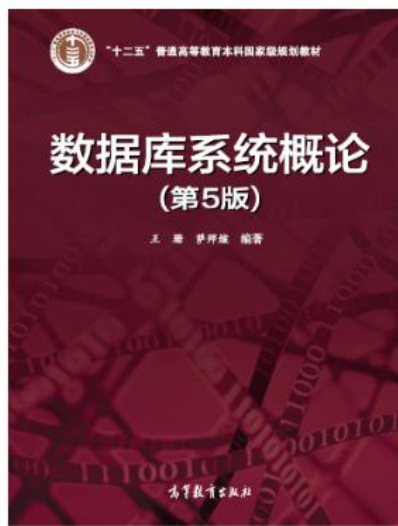
姓名：王东杰
电话：18637161988
0371-65832568
传真：0371-65832578
网址：www.hnhrfs.com
邮箱：hnhrfs@126.com
公司：河南省华瑞防水防腐有限公司

.....

数据 & 信息 (3)

□例:

《数据库系统概论(第5版)》



【作者】	王珊、萨师煊	【书号】	9787040406641
【出版社】	高等教育出版社	【页码】	418
【开本】	16开	【版次】	5
【出版日期】	2014-09-01		

本书第1版于1983年出版，至今已修订至第5版。第5版被列入“十二五”普通高等教育本科国家级规划教材。相应课程于2004年被评为北京市精品课程，2005年被评为国家精品课程，2014年被批准为国家级精品资源共享课

本书系统地全面地阐述了数据库系统的基础理论、基本技术和基本方法。全书分为4篇16章。第一篇基础篇，包括绪论、关系数据库、关系数据库标准语言SQL、数据库安全性和数据库完整性，共5章；第二篇设计与应用开发篇，包括关系数据理论、数据库设计和数据库编程，共3章；第三篇系统篇，包括关系查询处理和查询优化、数据库恢复技术、并发控制和数据库管理系统，共4章；第四篇新技术篇，包括数据库技术发展概述、大数据管理、内存数据库系统和数据仓库与联机分析处理技术，共4章。

本书可以作为高等学校计算机类专业、信息管理与信息系统等相关专业数据库课程的教材。也可供从事数据库系统研究、开发和应用的工程技术人员参考。

书名：数据库系统概论

作者：王珊，萨师煊

出版社：高等教育出版社

书号：9787040406641

开本：16

页码：418

出版日期：2014-09-01

版次：5

内容提要：本书第1版于1983年出版，至今已修订至第5版。第5版被列入“十二五”普通高等教育本科国家级规划教材。.....

数据 & 信息 (4)

计算机世界



蕴涵

· 信息则是数据的内涵

抽象

· 数据是信息的载体



现实世界

1.1.1 数据库的四个基本概念

□数据的特性

1) **数据表现的多样性**：数字、文字、日期/时间、图形/图像、音频/视频、二进制串、文档、抽象数据类型等

2) **数据的可构造性** { 型 type { 数据类型 type
数据结构 structure
数据模式 schema
值 value

3) **数据的挥发性/持久性**

4) **数据的私有性/共享性**

5) **数据的‘量’**：从‘少量’到‘大量’、‘海量’，再到‘大数据’(Big Data)

1.1.1 数据库的四个基本概念

□ 数据特性的变化趋势

1) 数据的量：少量 \Rightarrow 大量 \Rightarrow 海量 \Rightarrow 大数据 (Big Data)

2) 数据的结构：简单 \Rightarrow 复杂 \Rightarrow 无结构

3) 数据的服务范围：私有 \Rightarrow 共享

4) 数据在软件中的地位

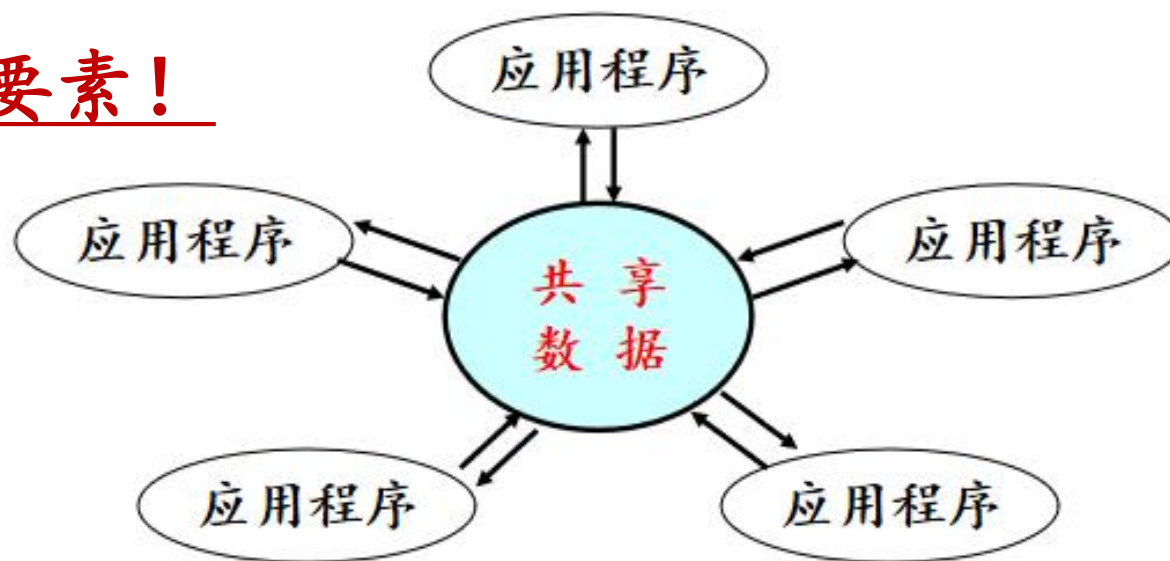
附属地位
(以程序为主体)



主导地位
(以数据为中心)

1.1.1 数据库的四个基本概念

- ❑传统的以程序为中心的数据管理方式已经无法适应上述数据特性的‘变化’情况，需要有新的数据管理技术，以便对数据作集中、统一的管理并使其对应用（程序）共享。
- ❑数据不再是依赖于处理过程的附属品，而是在现实世界中独立存在的对象。
- ❑数据已成为了一种新的生产要素！



以数据为主体的软件系统示意图

2022年新出台的新经济领域政策（摘选）

□2022.1.12: 《“十四五”数字经济发展规划》

□2022.1.26: 《数字乡村发展行动计划（2022-2025年）》

□2022.12.2: 《关于构建数据基础制度更好发挥数据要素作用的意见》

-- 又称“数据二十条”，提出构建数据产权、流通交易、收益分配、安全治理等制度，初步形成我国数据基础制度的“四梁八柱”；还首次提出了探索数据产权结构性分置制度，建立数据资源持有权、数据加工使用权、数据产品经营权“三权分置”的数据产权制度框架，构建中国特色数据产权制度体系。

□2022.12.14: 《扩大内需战略规划纲要（2022—2035年）》-- 把加快培育新型消费作为重点，支持线上线下商品消费融合发展，培育“互联网+社会服务”新模式，促进共享经济等消费新业态发展

□.....

2. 数据库 (DataBase, 简称 DB)

- 是指长期储存在计算机内、有组织的、可共享的大量数据的集合。
- 数据库中的数据按一定的**数据模型**组织、描述和储存，具有较小的**数据冗余**(data redundancy)、较高的**数据独立性**(data independency)和**易扩展性**(scalability)，并可为各种用户**共享**。
- 数据库中的数据具有以下特点
 - **永久存储**：存储在计算机的持久存储设备上
 - **有组织**：
 - ▲ 按一定的数据模型组织、描述和储存
 - ▲ 统一组织、集中存储
 - ▲ 较小的数据冗余度
 - ▲ 较高的数据独立性
 - **可共享**

结构化

集成化

集中管理

数据库的示例

□ Example 1: 高校数据库

- 人员信息: 学生, 教师, 职员, ...
- 机构信息: 院系, 部门, 机构, ...
- 教务信息: 课程, 班级, 选课, 成绩, ...
-



□ Example 2: 企业数据库

- 企业信息: 员工, 部门, 薪水, ...
- 生产信息: 产品, 仓库, 供应链, ...
- 销售信息: 客户, 经销商, 订单, 支付, ...
-

一个关系数据库的例子

students

sid	lname	fname	class	telephone
1	Jones	Allan	2	555-1234
2	Smith	John	3	555-4321
3	Brown	Harry	2	555-1122
5	White	Edward	3	555-3344

courses

cno	cname	croom	time
101	French I	2-104	MW2
102	French II	2-113	MW3
105	Algebra	3-105	MW2
108	Calculus	2-113	MW4

enrollment

sid	cno	major
1	101	No
1	108	Yes
2	105	No
3	101	Yes
3	108	No
5	102	No
5	105	No

1.1.1 数据库的四个基本概念

3. 数据库管理系统 (Database Management System, 简称 DBMS)

- 位于用户（应用程序）与数据库（操作系统）之间的一层数据管理软件，是一种用于管理数据库的大型系统软件。
- DBMS是在文件管理系统的基础上发展起来的，它区别于其它计算机软件系统的特点在于：
 - ① 能对持久性数据进行管理
 - ② 能对大量数据进行有效存取
 - ③ 可实现数据共享（众多用户可使用同一数据）
- DBMS的作用：**科学地组织和存储数据、高效地获取和维护数据**
 - ① 是数据库应用程序与数据库之间的访问接口
 - ② 在保证数据安全、可靠的同时，提高数据库应用时的简明性和方便性

数据库管理系统的功能 (1)

(1) 数据定义功能

- 提供数据定义语言 (Data Definition Language, DDL)
- 定义数据库中的数据对象 (数据模式 data schema)

(2) 数据组织、存储和管理

- 分类组织、存储和管理各种数据
- 确定数据的存储方式 (文件结构、存储组织等), 实现数据之间的联系
- 提供多种存取方法 (顺序扫描、索引扫描、HASH扫描等), 提高存取效率

(3) 数据操纵功能

- 提供数据操纵语言 (Data Manipulation Language, DML)
- 实现对数据库中数据的基本操作 (查询, 插入, 删除, 修改等)

(4) 数据库的事务管理和运行管理

- 数据库在建立、运行和维护时由数据库管理系统统一管理和控制
- 保证数据的安全性、完整性、多用户对数据的并发使用 (事务管理)
- 发生故障后的系统恢复 (数据库恢复)

(5) 数据库的建立和维护功能

- 数据库的建立, 初始数据加载与转换
- 数据库转储, 数据库恢复
- 数据库的重组, 性能监视与分析等

(6) 其他功能

- 与网络中其他软件系统的通信
- 数据库管理系统之间的数据转换
- 异构数据库系统之间的互访/互操作

数据库子语言 (1)

□ 数据子语言 (data sub_language)

➤ 数据定义语言

- Data Definition Language, 简称**DDL**, 负责数据模式的定义及数据存取方法的选择

➤ 数据操纵语言

- Data Manipulation Language, 简称**DML**, 负责数据的操纵, 包括查询及增、删、改等操作

➤ 数据控制语言

- Data Control Language, 简称**DCL**, 负责数据的完整性、安全性的定义与检查以及并发控制、故障恢复等功能

统一的关系数据库子语言 SQL

(Structured Query Language)

□数据子语言的使用方式

➤交互式命令语言

- 能在终端上即席操作，又被称为自含型或自主型语言
- 能够以命令行交互方式或批处理方式独立运行

➤宿主型语言

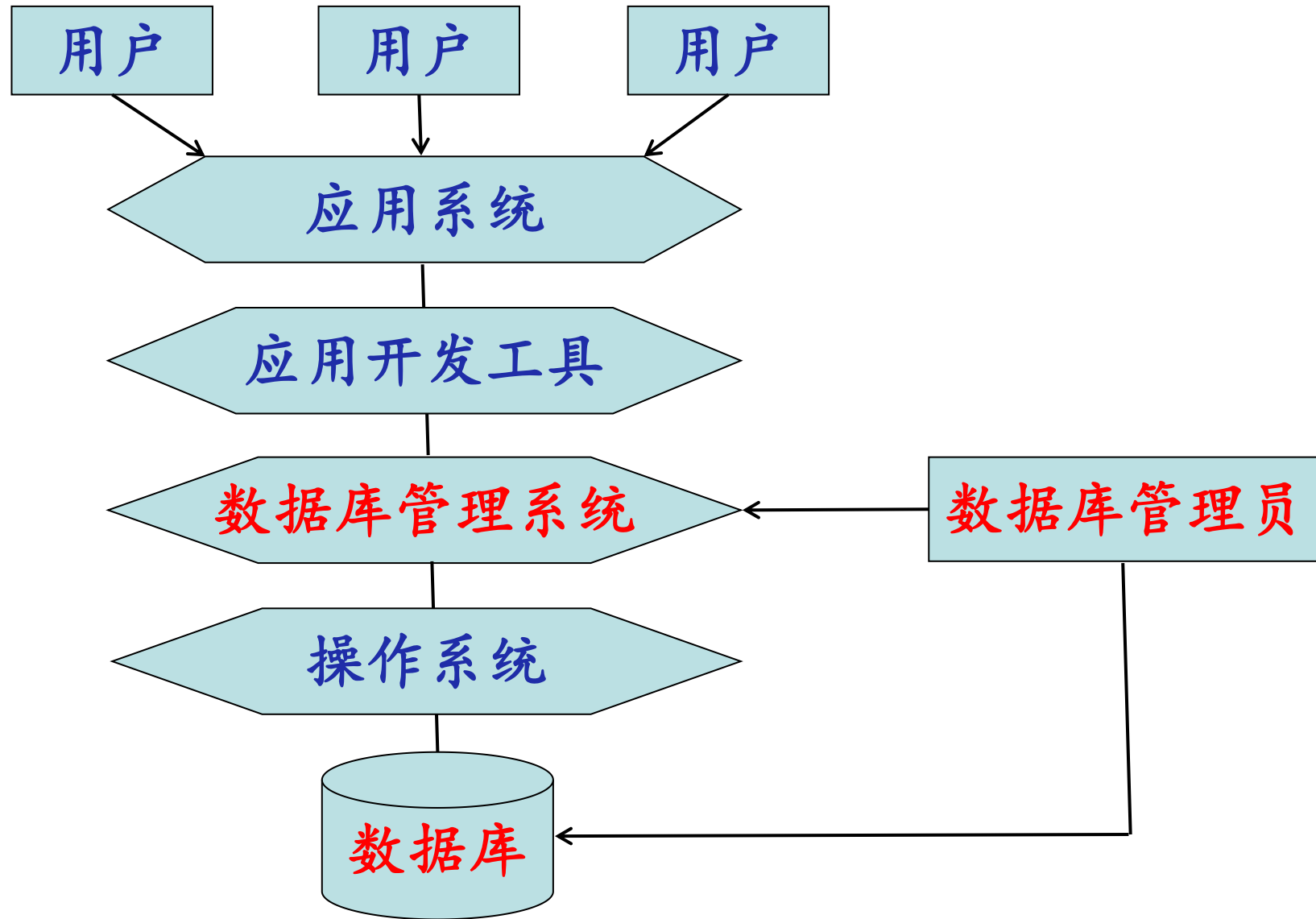
- 需要被嵌入到某种 宿主语言 中
- ‘宿主语言’通常是一种高级过程性程序设计语言，从早期的 Fortran, Cobol, C/C++ 到现在的 JAVA , Python 等，可以在其程序中使用‘数据子语言’来访问数据库。

1.1.1 数据库的四个基本概念

4. 数据库系统 与 数据库应用系统

- **数据库系统** (Database System, 简称 **DBS**) 是由数据库、数据库管理系统、数据库管理员、支撑系统运行所必须的系统软件和硬件平台组成的存储、管理、处理和维持数据的系统。
- **数据库应用系统** (Database Application System) 是由数据库系统、应用开发工具、应用系统、数据库用户组成的计算机系统。
- 在不引起混淆的情况下, 常常把两者都统称为 ‘数据库系统’。
- **数据库系统的构成**
 - 数据库
 - 数据库管理系统
 - 数据库管理员 DBA
 - 系统硬件平台 与 系统软件平台
 - 应用系统 与 应用开发工具
 - 数据库用户

图1.1 数据库系统的构成



数据库在计算机系统中的位置



协同软件
办公软件

中间件
应用服务器

数据库系统
操作系统

5. 数据库用户 (Database Users)

➤ 是指规划、设计、建立、访问、管理、维护数据库的人员。

➤ 通常存在以下三类数据库用户

① 数据库管理员 (DataBase Administrator, DBA)

- 负责数据库的规划、设计、建立、管理、维护、监视的专职人员。

② 应用开发人员 (包括系统分析员、数据库设计人员、应用程序员)

③ 最终用户 (End users): 通过应用系统访问使用数据库的人员。

1.1 数据库系统概述

1.1.1 数据库的四个基本概念

1.1.2 数据管理技术的产生和发展

1.1.3 数据库系统的特点

1.1.2 数据管理技术的产生和发展

□ **数据库技术**是应数据管理任务的需要而产生。

□ 什么是数据管理？

➤ **数据管理**是指对数据进行分类、组织、编码、存储、检索和维护，是数据处理的中心问题。

□ 什么是数据处理？

➤ **数据处理**则是指对各种数据进行收集、存储、加工和传播的一系列活动的总和。

1.1.2 数据管理技术的产生和发展

- ❑ 在数据处理中，通常计算比较简单，而且数据处理的需求因业务的不同而不同，需要根据业务的需求编写相应的应用程序来解决。
- ❑ 由于可利用的数据呈爆炸性增长、且数据的种类繁杂，数据管理任务则越来越复杂。
- ❑ 数据管理技术的优劣将对数据处理的效率产生直接影响。
- ❑ 数据管理技术的发展动力：
 - 应用需求的推动
 - 计算机硬件的发展
 - 计算机软件的发展

1.1.2 数据管理技术的产生和发展

❑ 数据管理技术的发展过程

- 人工管理阶段（20世纪50年代中之前）
- 文件系统阶段（20世纪50年代末--60年代中）
- 数据库系统阶段（20世纪60年代末--现在）

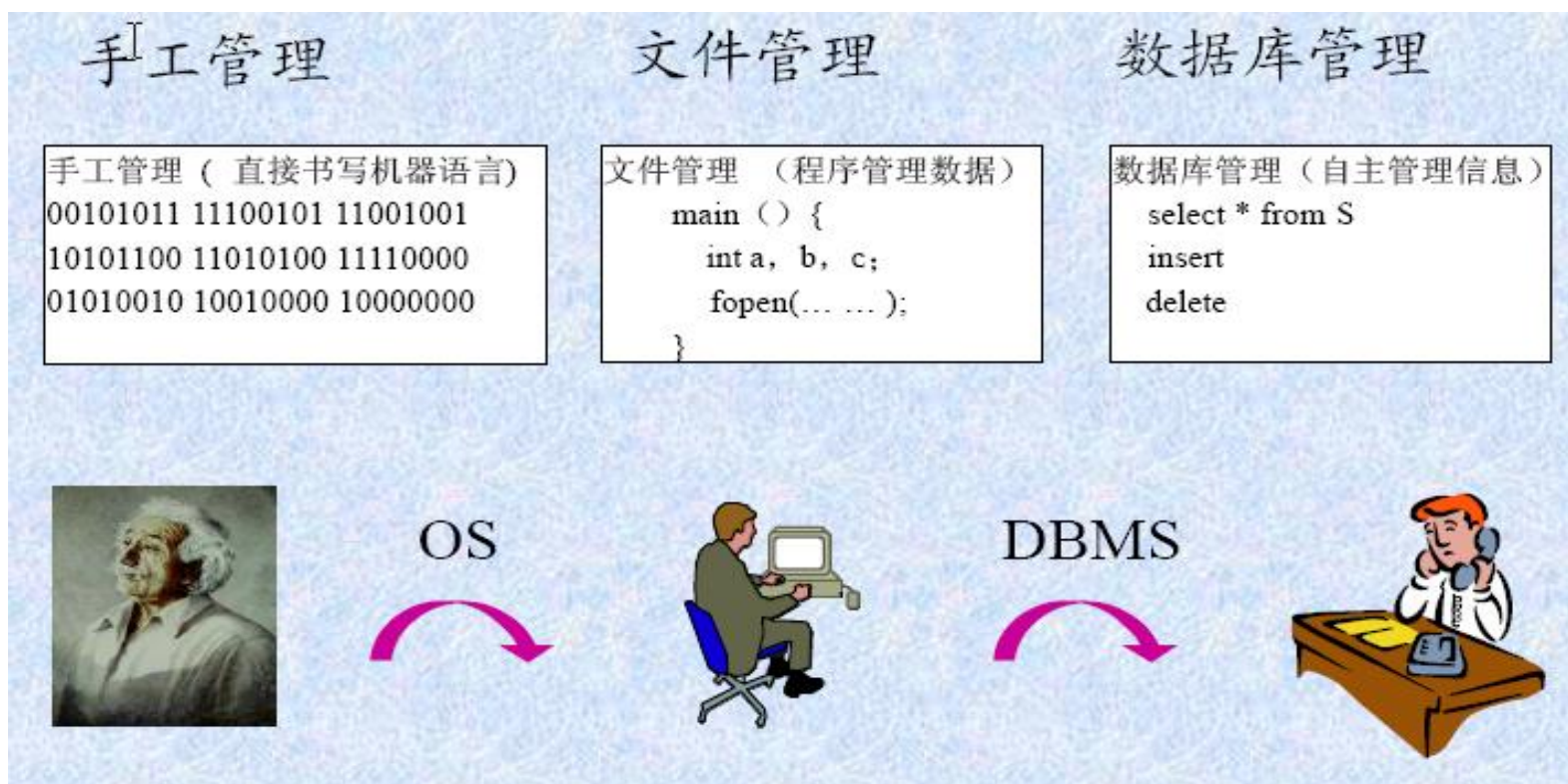


表1.1 数据管理三个阶段的比较

		人工管理阶段	文件系统阶段	数据库系统阶段
背景	应用背景	科学计算	科学计算，数据管理	大规模数据管理
	硬件背景	无直接存取存储设备	磁盘，磁鼓	大容量磁盘，磁盘阵列
	软件背景	没有操作系统	有文件系统	有数据库管理系统
	处理方式	批处理	联机实时处理、批处理	联机实时处理、分布处理、批处理
特点	数据的管理者	用户（程序员）	文件系统	数据库管理系统
	数据面向的对象	某一应用程序	某一应用	现实世界（一个部门、企业、跨国组织等）
	数据的共享程度	无共享，冗余度极大	共享性差，冗余度大	共享性高，冗余度小
	数据独立性	不独立，完全依赖于程序	独立性差	具有较高的物理独立性和一定的逻辑独立性
	数据的结构化	无结构	记录内有结构，整体无结构	整体结构化，用数据模型描述
	数据控制能力	应用程序自己控制	应用程序自己控制	由DBMS提供统一控制

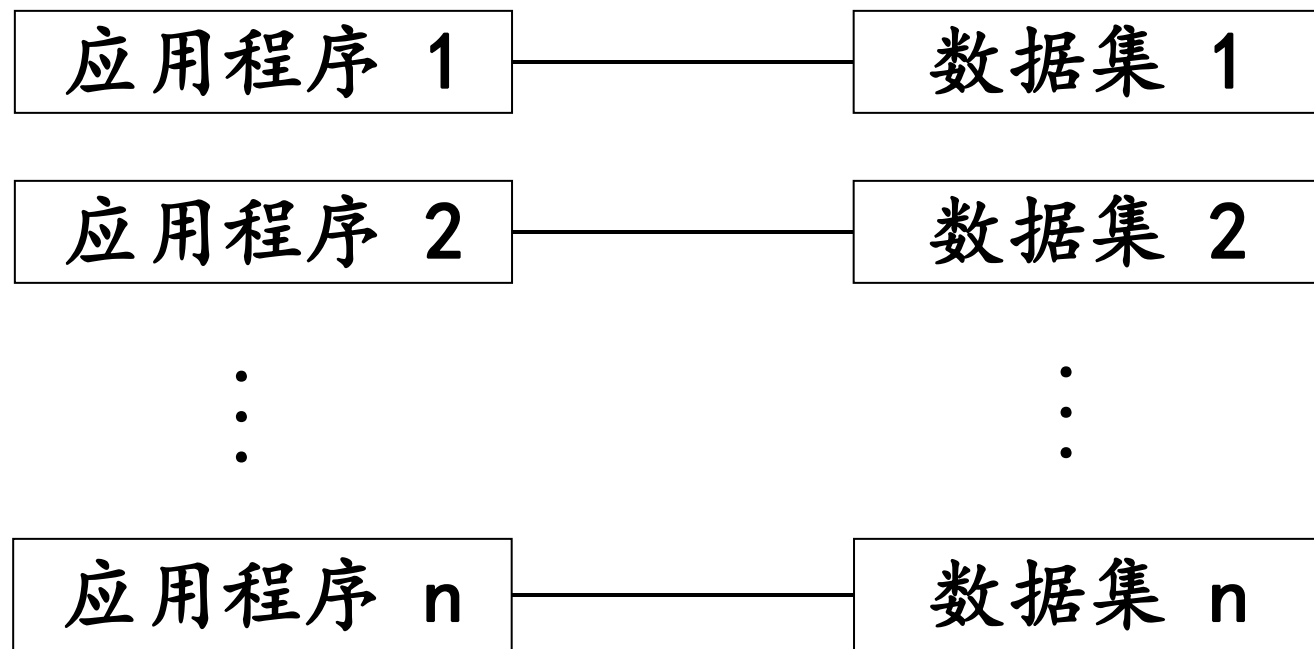


图1.3 人工管理阶段应用程序与数据之间的一一对应关系

□ 特点

- **数据的管理者**：用户（程序员），数据不保存
- **数据面向的对象**：某一应用程序
- **数据的共享程度**：无共享、冗余度极大
- **数据独立性**：不独立，完全依赖于程序
- **数据结构化**：无结构
- **数据控制能力**：应用程序自己控制

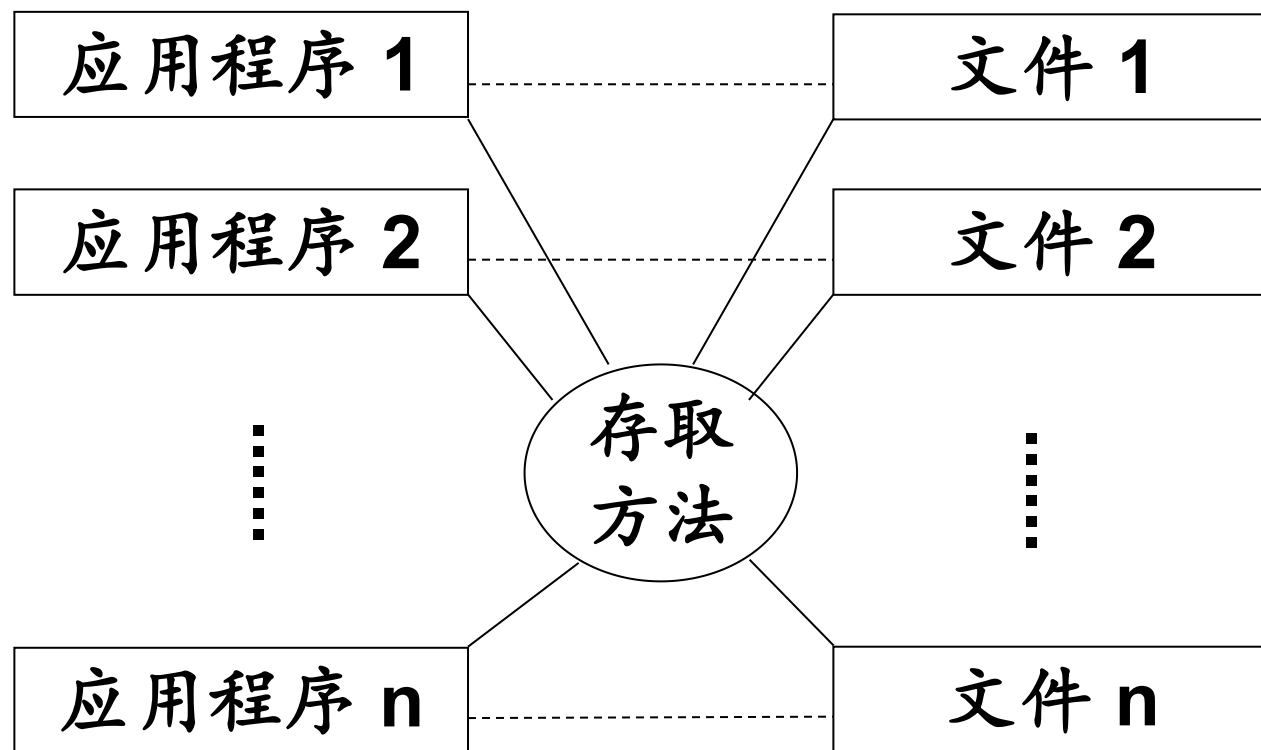


图1.4 文件系统阶段应用程序与数据之间的对应关系

□ 特点

- **数据的管理者**: 文件系统, 数据可长期保存
- **数据面向的对象**: 某一应用
- **数据的共享程度**: 共享性差、冗余度大
- **数据的结构化**: 记录内有结构, 整体无结构
- **数据的独立性**: 独立性差
- **数据控制能力**: 应用程序自己控制

❑ 利用文件系统可以实现简单的数据管理和数据共享，但无法提供完整统一的数据管理功能和并发控制等多用户数据共享功能。

❑ 例如：

➤ 对象持久化

- C++的持久对象存储
- Java对象的序列化/反序列化

➤ 基于文本的数据共享

- XML
- JSON
- K/V

❑ 在前述多个应用共享数据的前提下：

- 谁来负责定义和管理这些数据？
- 如何确保数据结构和存储机制对于所有应用来说都是可以接受的？
- 如何确保数据安全性、完整性？
- 如何实现并发访问下的数据一致性？
- 如何实现数据持久性（故障恢复）？
- 如何提高访问效率？

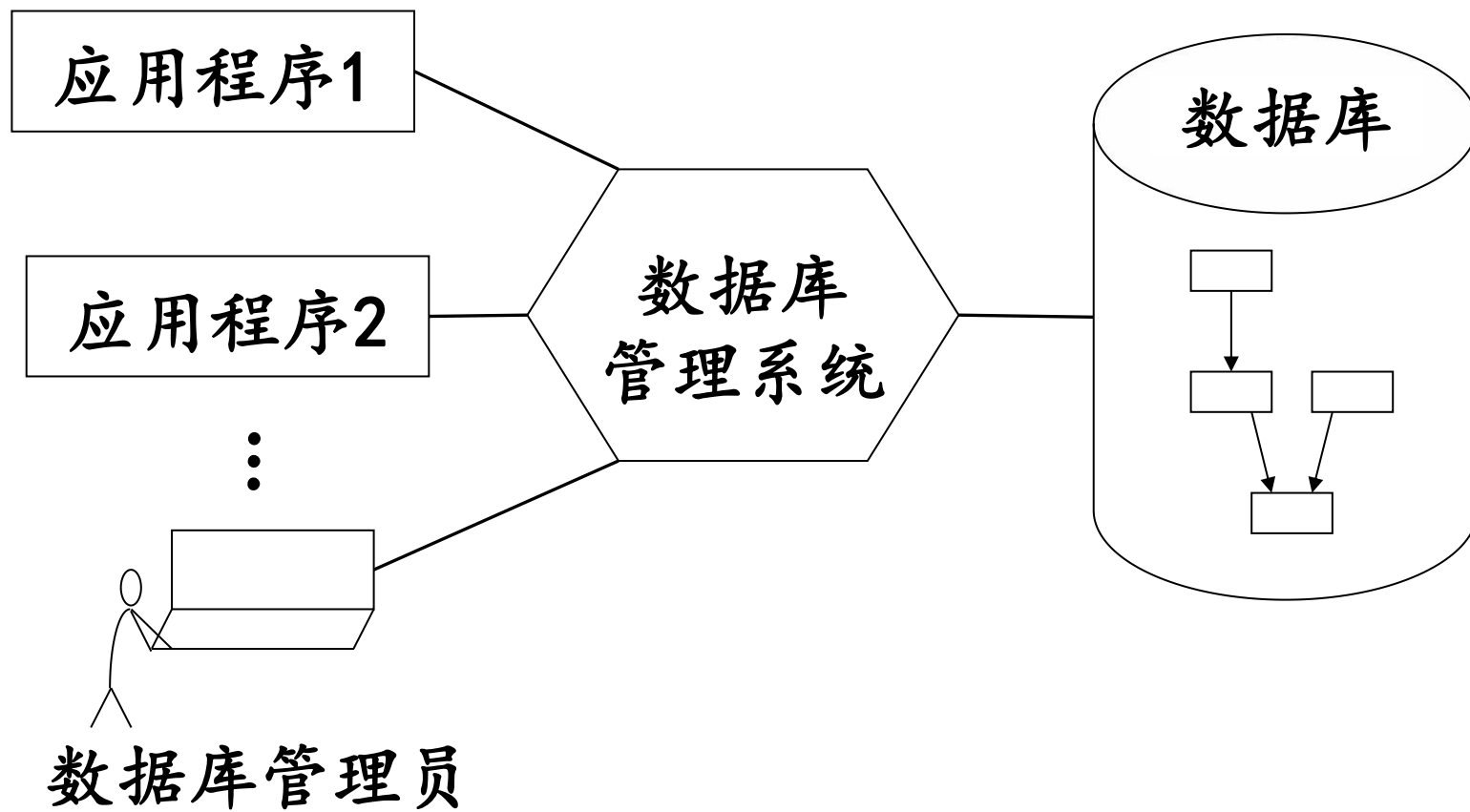


图1.6 数据库系统阶段应用程序与数据之间的对应关系

数据库系统的发展历史 (1)

❑ 第一阶段：层次/网状数据库（20世纪60年代~70年代）

➤ 数据库从无到有，提供简单的共享数据的‘读/写’

❑ 第二阶段：关系数据库（20世纪70年代~）

➤ 数据库从有到 **大规模商业化应用**

➤ 面向事务处理型应用：**OLTP**（联机事务处理）

❑ 第三阶段：关系数据库系统扩展（20世纪90年代~）

➤ 数据仓库 (Data Warehouse)

➤ 面向数据分析型应用：**OLAP**（联机分析处理）

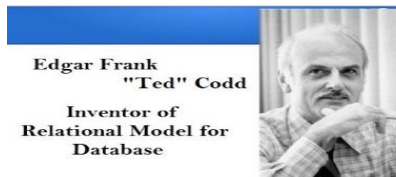
❑ 第四阶段：新型数据管理系统（21世纪~）

➤ NoSQL / NewSQL / 云原生 / AI+DB / 新型数据库系统

图灵奖：数据库概念
Charles Bachman



图灵奖：关系代数
Edgar Codd



图灵奖：事务处理
Jim Gray



图灵奖：数据库系统
Mike Stonebraker



数据库系统的发展历史 (2)

□ 第一阶段：层次数据库与网状数据库

➤ 层次数据库与网状数据库于20世纪60年代末开始发展。它们为统一管理与共享数据提供了有力支撑，**是真正的数据库系统。**

- 1960s早期：网状数据库管理系统 IDS (Integrated Data Storage), 通用电气

- 1960s晚期：层次数据库管理系统 IMS (Information Management System), IBM公司

➤ **优点**：数据库系统为用户提供了统一的数据管理工具，向用户提供了数据共享能力。

➤ **缺点**

- 它们脱胎于文件系统，**受文件的物理影响较大**，对数据库的使用带来不便；

- **数据模式构造烦琐**，不利于推广使用。

数据库系统的发展历史 (3)

□ 第二阶段：关系数据库

➤ 关系数据库于20世纪70年代开始问世，80年代初进入潮流。

- 1970: E. F. Codd 提出 **关系数据模型**
- 1974: 第一个关系数据库管理系统 IBM公司的 **system R**, Ray Boyce和Don Chamberlin 提出数据操纵语言 **SQL**(Structured Query Language)
- 1976:
 - Peter Chen 提出 **实体联系模型(Entity-Relationship Model)**
 - James Gray 提出 **事务处理(Transaction Processing)**技术
- 1979: **oracle** release 1
- 1983: IBM **DB2**
- 1988: **sybase**
- 1989: 微软 **SQL Server**, Berkeley 的 **Postgre SQL**

□ 第三阶段：关系数据库系统扩展

➤ 自上个世纪九十年代以来，数据库技术的发展集中于对传统关系数据库系统的进一步扩充与改造上。

- 对象关系数据库系统
- 数据仓库 与 联机分析处理
- Web数据库
- 安全数据库
- 嵌入式数据库，移动数据库，实时数据库，网格数据库，传感器网络数据库，.....

□ 第四阶段：新型数据管理系统

➤ 新型的应用模式

- OLAP (On-Line Analytical Processing)
- Web应用

➤ 新型应用的特点

- 大规模、海量数据
- 高度实时&并发
- 大数据分析应用

➤ 解决方案：

- NoSQL, NewSQL
- 云数据库
- AI + DB
- 新型数据库系统

数据库系统分类

关系数据库 RDBMS	Oracle, DB2, Microsoft SQL Server, MySQL, PostgreSQL,
键值数据库 Key-value stores	Redis, Memcached, Amazon Dynamo DB, Microsoft Azure Cosmos DB,
文档数据库 Document-oriented stores	MongoDB, Couchbase, CouchDB, Amazon Dynamo DB, Microsoft Azure Cosmos DB,
图数据库 Graph-oriented DBMS	Neo4j, OrientDB, Microsoft Azure Cosmos DB,
列数据库 Wide column store	Cassandra, HBase,
其他	面向对象数据库（object-oriented DBMS），XML数据库，RDF三元组数据库，

} NoSQL 数据库

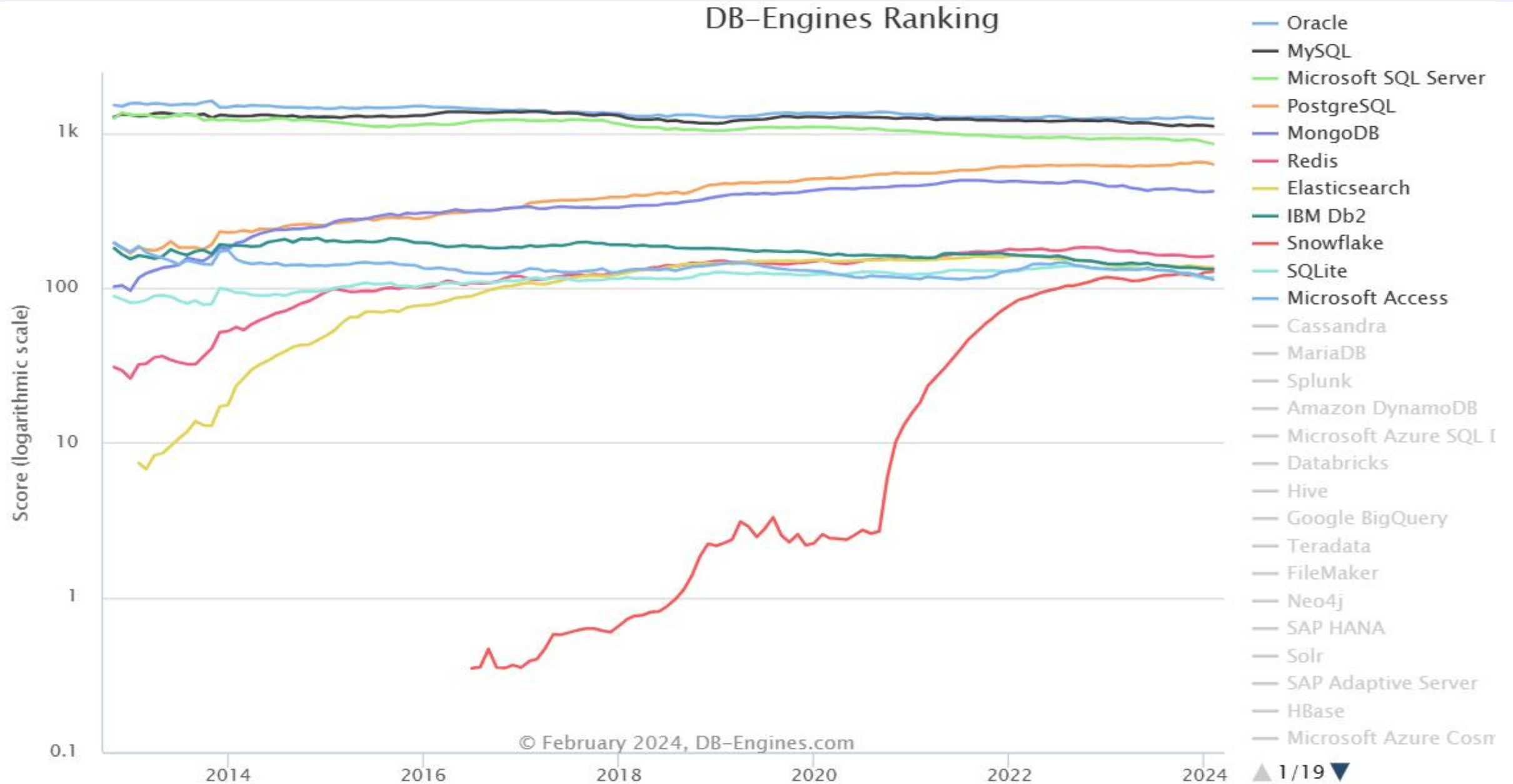
DB-Engines: 2024年2月全球数据库人气排行榜

<https://db-engines.com/en/ranking>

417 systems in ranking, February 2024

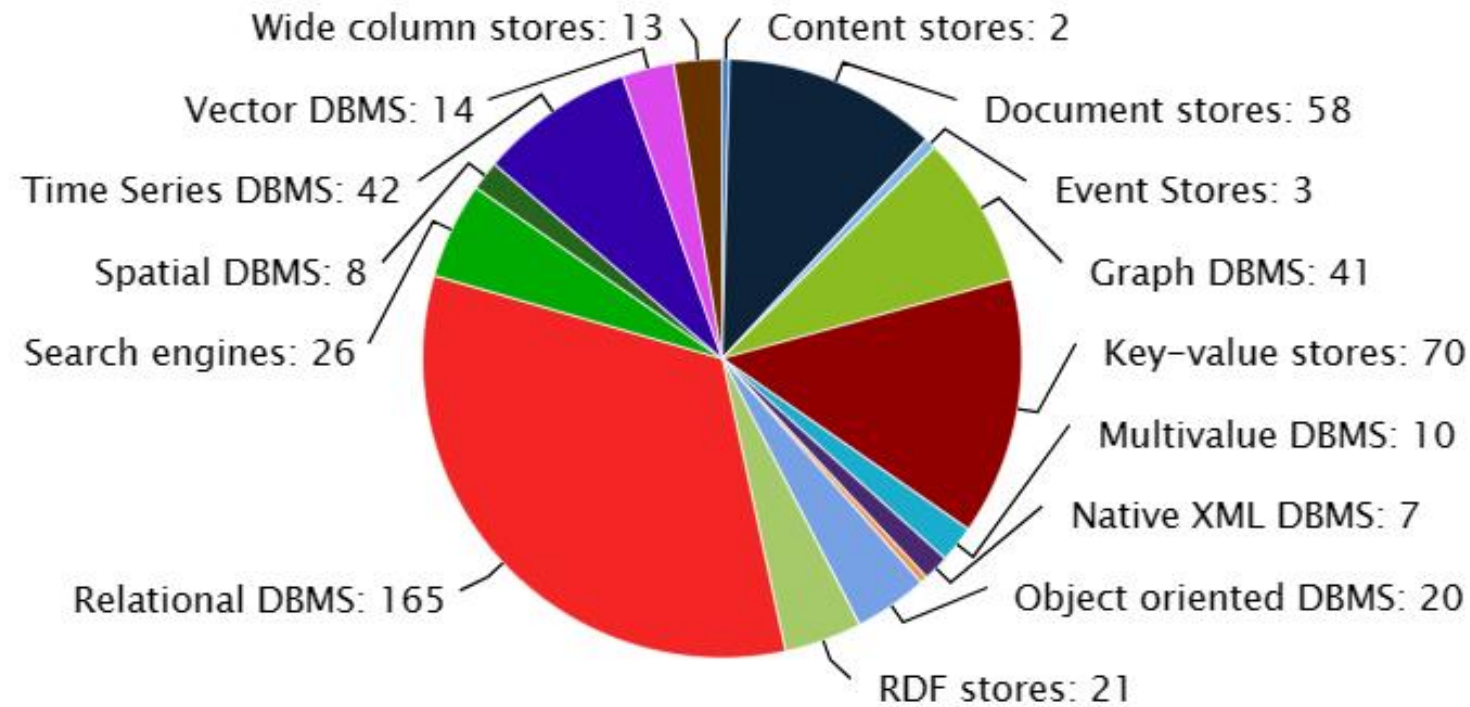
Rank			DBMS	Database Model	Score		
Feb 2024	Jan 2024	Feb 2023			Feb 2024	Jan 2024	Feb 2023
1.	1.	1.	Oracle +	Relational, Multi-model i	1241.45	-6.05	-6.08
2.	2.	2.	MySQL +	Relational, Multi-model i	1106.67	-16.79	-88.78
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model i	853.57	-23.03	-75.52
4.	4.	4.	PostgreSQL +	Relational, Multi-model i	629.41	-19.55	+12.90
5.	5.	5.	MongoDB +	Document, Multi-model i	420.36	+2.88	-32.41
6.	6.	6.	Redis +	Key-value, Multi-model i	160.71	+1.33	-13.12
7.	7.	↑ 8.	Elasticsearch	Search engine, Multi-model i	135.74	-0.33	-2.86
8.	8.	↓ 7.	IBM Db2	Relational, Multi-model i	132.23	-0.18	-10.74
9.	9.	↑ 12.	Snowflake +	Relational	127.45	+1.53	+11.80
10.	↑ 11.	↓ 9.	SQLite +	Relational	117.28	+2.08	-15.38

Popularity changes per category (2013~2024)



DB-Engines: 2024年2月全球数据库产品分布

Number of systems per category, February 2024



© 2024, DB-Engines.com

DB-Engines lists 417 different database management systems, which are classified according to their database model
(<https://db-engines.com/en/ranking>)

国产数据库的发展历史

□ “核高基”

- “核心电子器件、高端通用芯片、基础软件产品”
- 是2006年国务院发布的《国家中长期科学和技术发展规划纲要(2006-2020年)》中的16个重大科技专项之一

□ 国产数据库的研究历史

- 2000年以前：高校、科研院所，研究型
- 2000年后：高校、科研院所 尝试产业化
- 2010年以后：以 企业/公司 为研究主体



“两弹一星”精神

2024年2月中国数据库流行度排行榜（top10）

<https://www.modb.pro/dbrank>

排行	上月	半年前	名称	模型 ∨	数据处理 ∨	部署方式 ∨	商业模式 ∨	专利	论文	案例	资质	书籍	得分
	↑ 2	↑ ↑ ↑ 4	PolarDB	关系型	HP	  	 	592	70	71	14	2	856.07
	↓ 1	↓ 1	OceanBase	关系型	HP	 	 	151	26	43	14	1	727.05
	↑ 4	↓ 2	TiDB	关系型	HP	 	 	40	54	18	8	1	641.97
4	↓ 3	↓ 3	openGauss	关系型	TP			573	11	16	7	5	610.16
5	5	↑ ↑ 7	人大金仓	关系型	TP AP	  		368	0	25	12	3	588.82
6	↑ ↑ 8	↓ 5	GaussDB	关系型	HP	 		630	14	10	9	4	516.64
7	7	↑ 8	GBASE	关系型	AP TP	 		191	1	46	11	0	506.40
8	↓ ↓ 6	↓ ↓ 6	达梦数据库	关系型	TP	  		518	0	10	8	11	500.98
9	↑ 10	↑ ↑ ↑ 13	GoldenDB	关系型	HP	 		581	40	45	11	1	310.58
10	↓ 9	↓ 9	TDSQL	关系型	HP	 	 	136	19	16	12	0	285.37

1.1 数据库系统概述

1.1.1 数据库的四个基本概念

1.1.2 数据管理技术的产生和发展

1.1.3 数据库系统的特点

一个例子：学生学籍管理系统（1）

□学生的信息包括学号、姓名、性别、年龄、专业和奖励

□用文件系统实现学籍管理

➤数据存储

- 学生的基本信息：定长记录，存储在“学生基本信息”文件中
- 学生获得的奖励信息：变长记录，存放在另一个“奖励”文件中
- 在“学生基本信息”记录中，添加‘位置’和‘长度’字段来描述该学生的奖励记录在“奖励”文件中的开始位置和长度

➤数据存取

- 编写应用程序，实现数据的录入和查找

➤缺点：程序员必须关注记录结构和不同文件中记录之间的联系，工作量大，编程复杂，开发速度慢

一个例子：学生学籍管理系统（2）

表1.3 “学生基本信息”文件的结构和内容

学号	姓名	性别	年龄	专业	位置	长度
20100001	史玉明	女	20	计算机	0	30
20100100	李明虎	男	21	机械	30	15
20100234	张翔	男	21	化工	45	0
.....

表1.4 “奖励”文件的结构和内容

奖励
2011校奖学金, 2012国家奖学金
2012校优秀学生

一个例子：学生学籍管理系统（3）

□学生的信息包括学号、姓名、性别、年龄、专业和奖励

□用数据库系统实现学籍管理

➤数据存储

- 建立两张表

- students表：存放学生的基本信息

- awards表：存放学生获得的奖励情况

- 通过外关键字实现“学生基本信息”与“奖励信息”的关联

➤数据存取

- 使用DBMS提供的数据库操纵语言，实现数据的录入和查找

一个例子：学生学籍管理系统（4）

students 学生基本信息表

学号	姓名	性别	年龄	专业
20100001	史玉明	女	20	计算机
20100100	李明虎	男	21	机械
20100234	张翔	男	21	化工
.....

awards 学生奖励信息表

学号	奖励
20100001	2011校奖学金，2012国家奖学金
20100100	2012校优秀学生
.....

文件管理存在的问题

在数据库技术出现之前，应用程序对于（文件中）数据的访问存在以下方面的问题：

❑ 数据存储

- 冗余存储 redundancy
- 数据的不一致性 data inconsistency
- 完整性约束 integrity constraints

❑ 数据访问

- 缺少标准的数据访问接口
- 数据更新操作的原子性 atomicity of updates

❑ 多用户并发

- 多用户访问的隔离、并发控制、故障恢复

❑ 数据安全：用户身份识别，访问控制

1.1.3 数据库系统的特点

- ❑ 数据结构化
- ❑ 数据的共享性高，冗余度低且易扩充
- ❑ 数据独立性高
- ❑ 数据由数据库管理系统统一管理和控制

数据库的特点 (1) - 数据结构化

❑数据的整体结构化是数据库的主要特征之一，也是数据库系统与文件系统的本质区别。

❑整体结构化

- 不再仅仅针对某一个应用，而是面向全组织（**集成性**：采用统一的数据结构，建立一个全局统一的数据模式，根据每个应用的数据需要构造局部模式）
- 不仅数据内部结构化，整体是结构化的，数据之间具有联系
- 数据记录可以变长
- 数据的最小存取单位是数据项

❑数据的组织结构、联系方式、存取方法用‘**数据模型**’描述，无需应用程序定义

数据库的特点 (2) - 数据的共享性高, 冗余度低且易扩充

❑ 数据面向整个系统, 可以被多个用户、多个应用共享使用。

➤ 数据共享

- 可供多个应用程序使用, 并可用于不同的目的
- 可以在已有的数据库系统上开发新的应用程序
- 可向外界提供信息服务功能

➤ 数据冗余

- 同一个数据在系统中的不同地方出现了重复存储

➤ 数据一致性

- 在系统中, 同一数据的不同出现应保持相同的值

❑ 数据共享的好处

- 减少数据冗余, 节约存储空间
- 避免数据之间的不相容性与不一致性
- 使系统易于扩充

数据库的特点 (3) - 数据独立性高

❑ **数据独立性**是指数据库中的数据与使用这些数据的应用程序之间的互不依赖性，即数据或数据结构的改变不会导致对使用这些数据的应用程序的修改，反之亦然。

❑ **物理独立性**

- 指用户的应用程序与数据库中数据的物理存储是相互独立的。
- 当数据的物理存储改变了，应用程序不用改变。

❑ **逻辑独立性**

- 指用户的应用程序与数据库的逻辑结构是相互独立的。
- 数据的逻辑结构改变了，应用程序不用改变。

❑ 在数据库管理系统中，数据独立性是由**三级模式**及它们之间的**二级映像**功能来保证。

数据库的特点（4） - 数据由数据库管理系统统一管理和控制

❑数据库管理系统提供的数据库控制功能包括

➤数据的安全性（Security）保护

- 保护数据以防止不合法的使用造成的数据的泄密和破坏。

➤数据的完整性（Integrity）检查

- 保证数据的正确性、有效性和相容性。

➤并发（Concurrency）控制

- 对多用户的并发操作加以控制和协调，防止相互干扰而得到错误的结果。

➤数据库恢复（Recovery）

- 将数据库从错误状态恢复到某一已知的正确状态。

1.1.3 数据库系统的特点

❑ 数据结构化

❑ 高共享性、低冗余、易扩充

❑ 数据独立性高

❑ 统一管理与控制

• 统一的数据模型

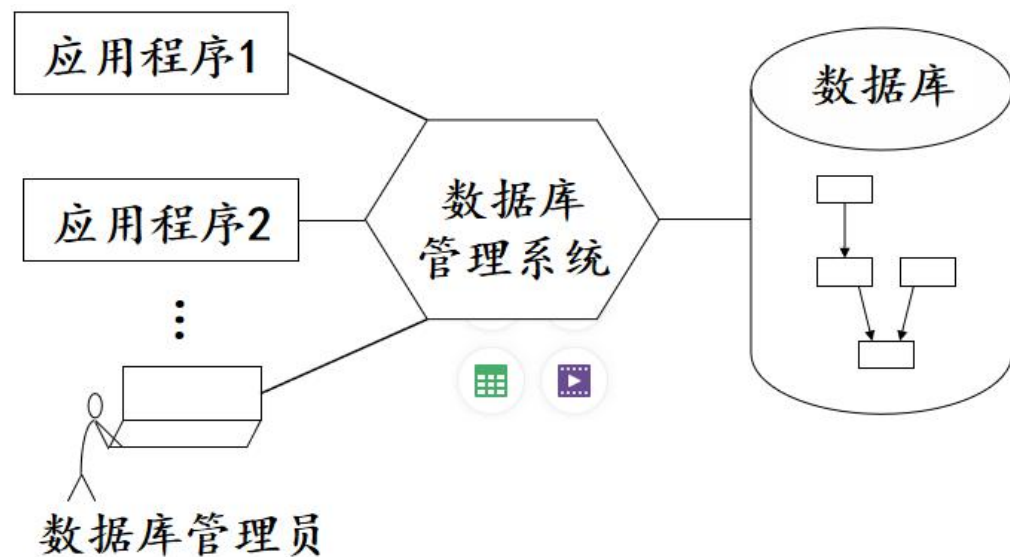
• 标准化数据访问接口
(SQL, ODBC, ...)

数据库管理系统

DBMS

数据库概念小结

- ❑ 数据库是长期存储在计算机内、有组织的、大量的、共享的数据集合。
- ❑ 可以供各种用户共享，具有最小数据冗余度和较高的数据独立性。
- ❑ 数据库管理系统在数据库建立、运用和维护时对数据库进行统一控制，以保证数据的完整性、安全性，并在多用户同时使用数据库时进行并发控制，在发生故障后对数据库进行恢复。



1.1 数据库系统概述

1.2 数据模型

1.3 数据库系统的结构

1.4 数据库系统的组成

1.5 小结

1.2 数据模型

❑ 数据模型是对现实世界数据特征的抽象，用以 抽象、表示和处理现实世界中的数据和信息

➤ 用‘**数据模型**’来描述现实世界中的数据及其相互关系，并定义在这些数据上可以执行的操作。

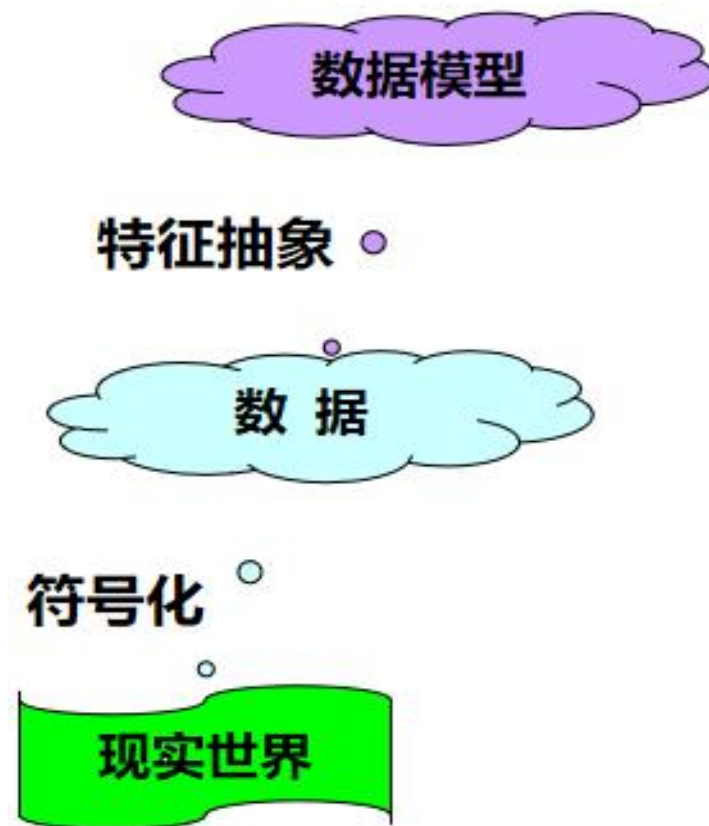
❑ 数据模型应满足三方面要求

➤ 能比较真实地**模拟**现实世界

➤ **容易**为人所**理解**

➤ **便于**在计算机上**实现**

❑ 数据模型是数据库系统的**核心和基础**



1.2.1 两类数据模型

- ❑ 数据模型是数据库系统的核心和基础，如何将现实世界中我们需要的数据及其复杂关系最终反映到数据库中去，这需要有一个逐步转化的过程。
- ❑ 我们用建立在不同抽象层次上的‘**数据模型**’来表示每一步转化的结果（两大类、共三种类型的数据模型）
 - **概念数据模型** (conceptual data model)
 - 又简称为‘**概念模型**’
 - **逻辑数据模型** (logic data model)
 - 又简称为‘**逻辑模型**’
 - **物理数据模型** (physical data model)
 - 又简称为‘**物理模型**’

□概念模型，也称信息模型

- 按用户的观点来对数据和信息建模，用于数据库设计

□逻辑模型

- 按计算机系统的观点对数据建模，用于DBMS实现
- 主要包括网状模型、层次模型、关系模型、面向对象数据模型、对象关系数据模型、半结构化数据模型等。

□物理模型

- 是对数据最底层的抽象，描述数据在系统内部的表示方式和存取方法。

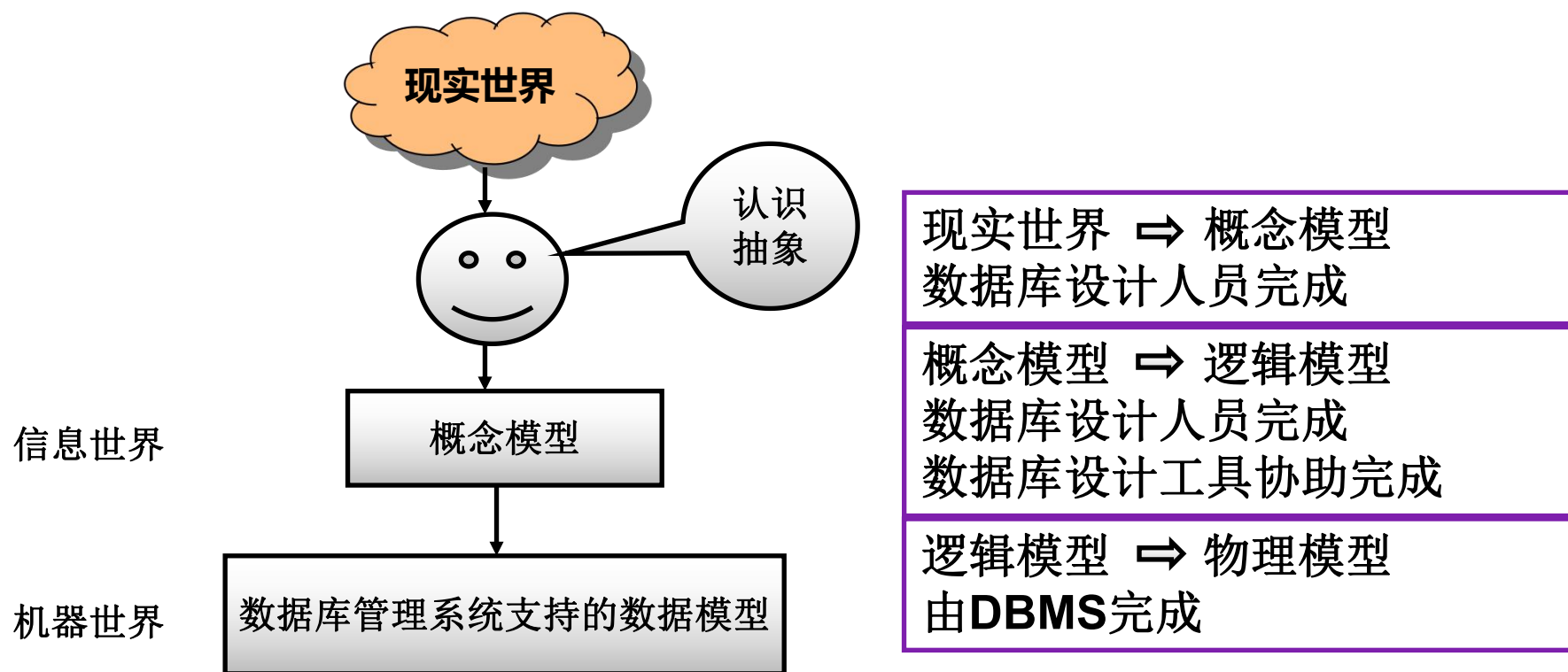
图1.7 客观对象的抽象过程---两步抽象

□将现实世界中的客观对象抽象为概念模型

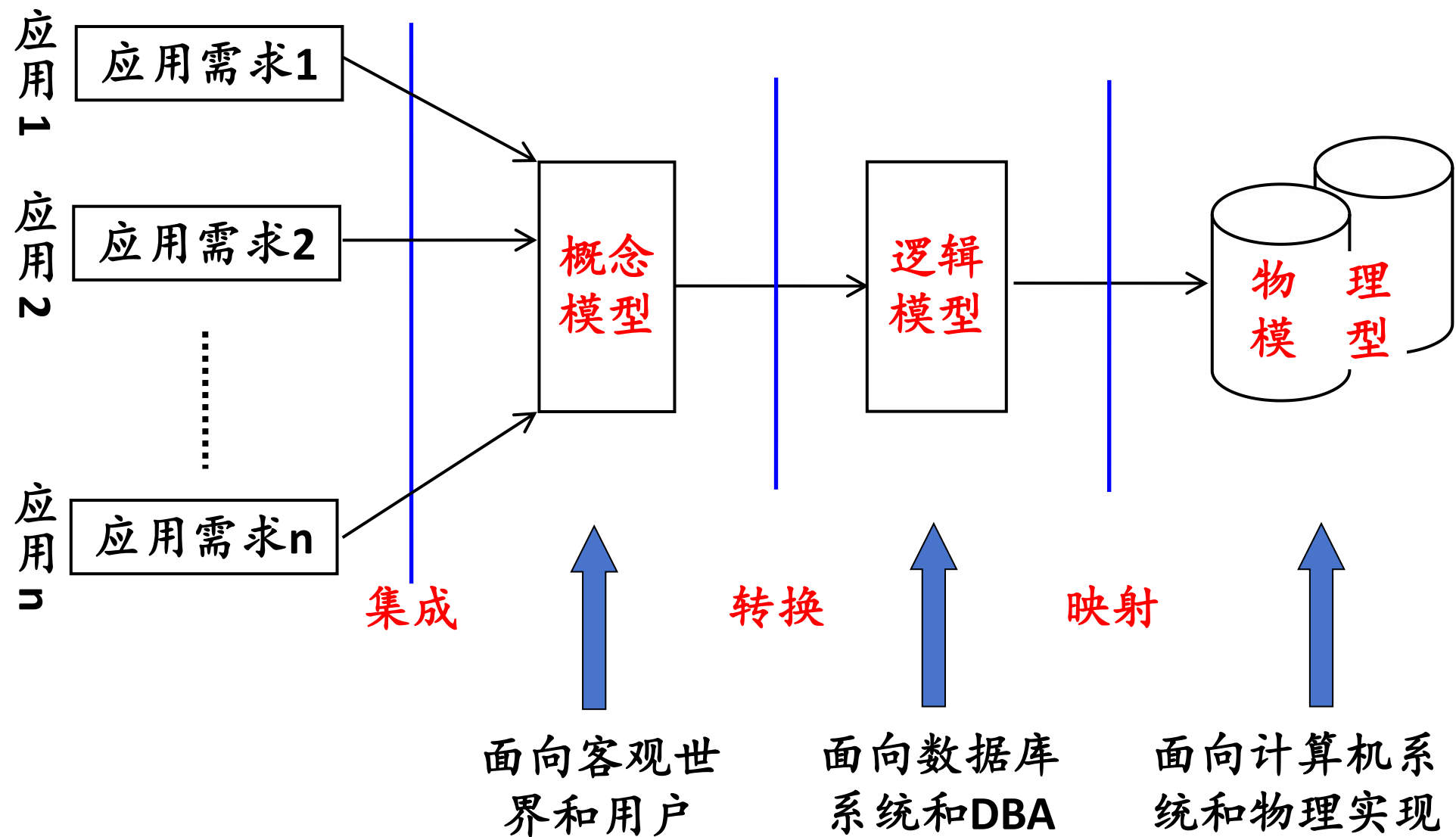
➤将现实世界抽象为信息世界

□把概念模型转换为特定DBMS支持的数据模型(逻辑模型&物理模型)

➤将信息世界转换为机器世界



三种数据模型之间的关系



□概念模型的用途

- 概念模型用于信息世界的建模
- 是现实世界到机器世界的一个中间层次
- 是数据库设计的有力工具
- 是数据库设计人员和用户之间进行交流的语言

□对概念模型的基本要求

- 较强的语义表达能力
- 简单、清晰、易于用户理解

□概念模型的特点

- 侧重于对客观世界中复杂事物的结构描述及它们之间的内在联系的刻画，不涉及具体的描述细节和物理实现因素。
- 是一种面向客观世界和用户的模型，与DBMS及计算机实现无关。

信息世界中的基本概念 (1)

□ 实体 (Entity)

- 客观存在并可相互区别的事物称为实体。可以是具体的人、事、物或抽象的概念。
- 每一个实体有一个‘**实体名**’，同类实体具有相同的实体名。

□ 属性 (Attribute)

- 实体具有的某一特性称为属性。一个实体可以由若干个属性来刻画。
- 每一个属性有一个‘**属性名**’，在同一个实体内属性名互不相同。

□ 联系 (Relationship)

- 现实世界中事物内部以及事物之间的联系，在信息世界中反映为**实体（型）内部的联系**和**实体（型）之间的联系**。
- 每一个联系有一个‘**联系名**’

信息世界中的基本概念 (2)

□ 实体型 (Entity Type)

- 用实体名及其所有属性名的集合来抽象和刻画同类实体
- 实体的实体名及其所有属性名的集合，称为该实体的‘**实体型**’

□ 实体值 (Entity Value)

- 实体中的属性可以有值，一个实体的所有属性值的集合被称为该实体的‘**实体值**’。（也被称为‘**实体实例**’， **Entity Instance**）

□ 实体集 (Entity Set)

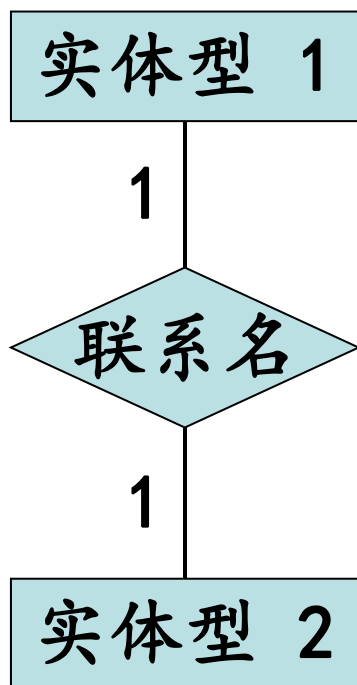
- 同一类型实体的集合，称为‘**实体集**’
- 同一个实体集中的所有实体，具有相同的实体型但实体值互不相同。

□ 码 (Key)

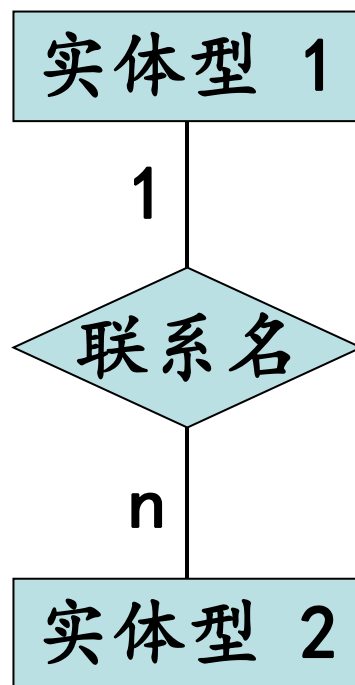
- 唯一标识实体的属性集称为码。
- ‘**码**’也被称为**关键字**、**键**、**标识符** (Identifier)
- 在一个实体集中，可以通过‘**码**’的取值来区分不同的实体。

实体之间的联系

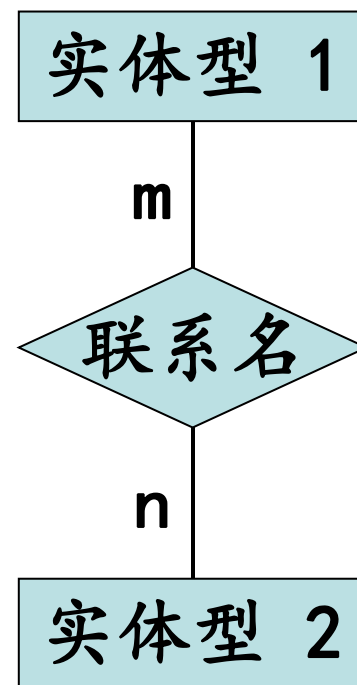
- 实体之间的联系通常是指不同实体集之间的联系
- 实体之间的联系有‘一对一、一对多、多对多’等多种类型，在ER模型中被称为‘**联系的函数关系**’



1:1 联系



1:n 联系



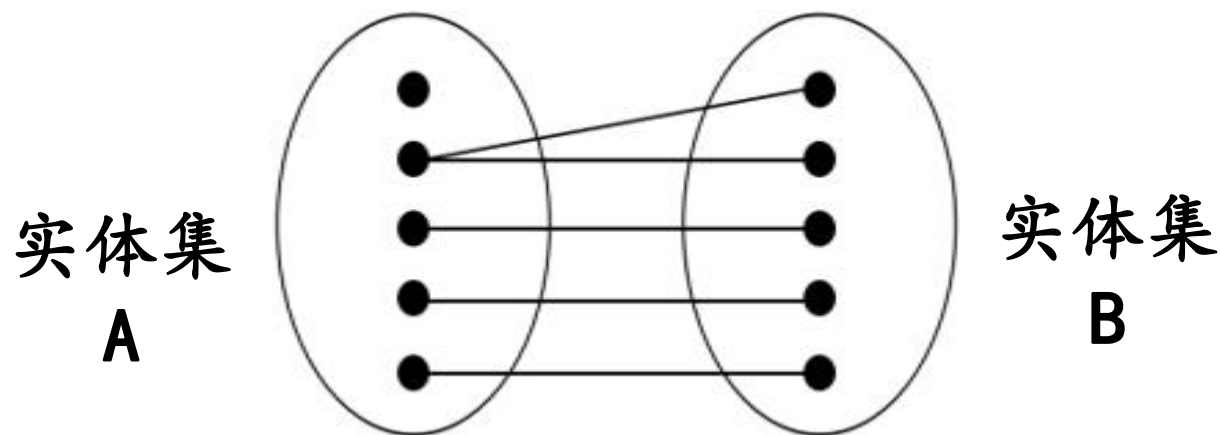
m:n 联系

(说明：用数字 1 表示‘一’，用字母表示‘多’)

联系的函数关系（1）

❑ 为了描述清楚函数关系，我们用示意图来解释三种不同的函数关系。如下图所示，其中：

- 椭圆形代表‘实体集’
- 点 (dot) 代表‘**实体**’（是现实世界中的某个客观对象在ER模型中的抽象表示）
- 点与点之间的**线**(line)代表在现实世界中的‘**实体之间的联系**’情况



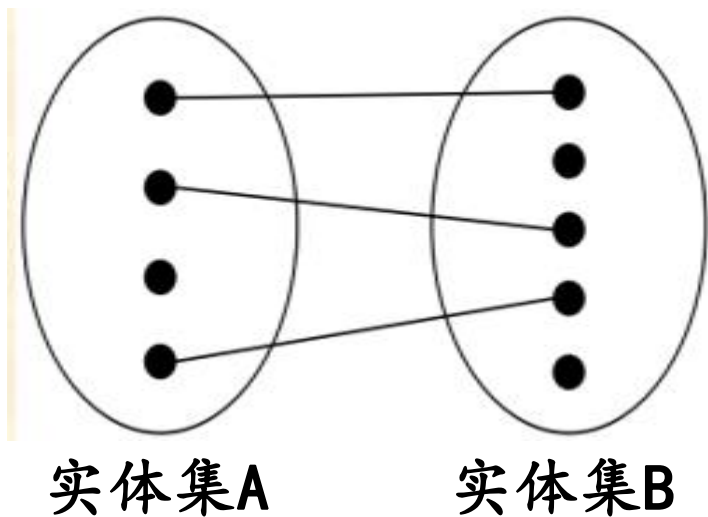
注：这里是解释‘函数关系’的示意图，不是ER图中用来表示属性的‘椭圆’符号！

联系的函数关系 (2)

□ 一对一

一对一的联系应该具备以下两个性质 (**a**和**b**分别是实体集**A**和实体集**B**中的一个实体)：

- ① 一个实体**a**最多只能对应唯一的一个实体**b**
- ② 一个实体**b**最多只能对应唯一的一个实体**a**



➤ ‘一对一’ 联系的例子

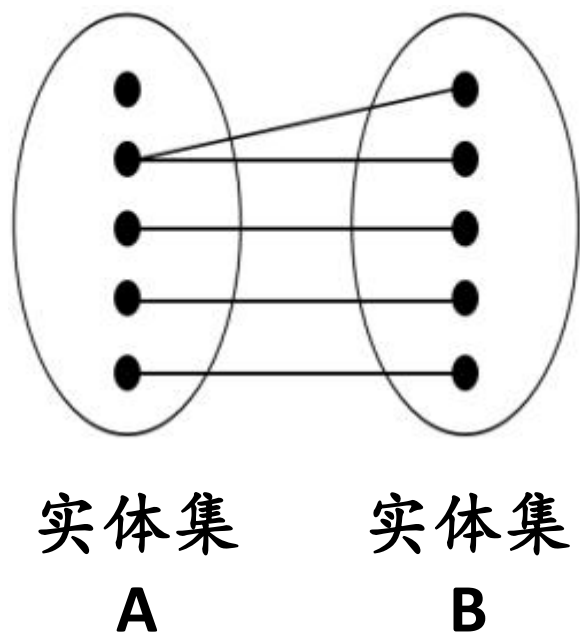
- ‘居民’ vs. ‘身份证’
- ‘学生’ vs. ‘学生证’
- ‘居民’ vs. ‘驾驶证’

【思考】在‘一对一’联系中，允许一个实体集中的某些实体，在另一个实体集中不存在与之对应的任何实体。请列举一个这样的二元联系例子，在两个实体集中，都存在着某些实体，与另一个实体集中的任何实体没有联系。

□ 一对多

假设在实体集A与实体集B之间存在一个‘一对多’的联系，则该联系具备以下性质(实体 $a \in$ 实体集A, 实体 $b \in$ 实体集B)

- ① 允许一个实体a对应多个实体b，但是
- ② 一个实体b最多只能对应唯一的一个实体a



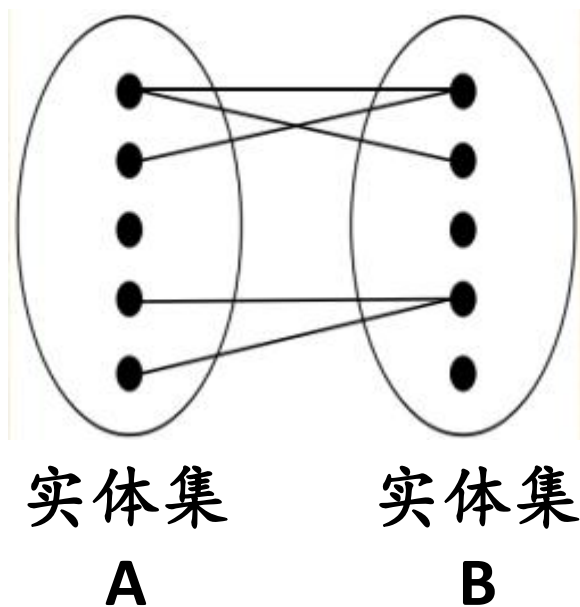
➤ 例如 (从左到右的‘一对多’)

- 住宿: ‘宿舍’ vs. ‘学生’
- 拥有: ‘人’ vs. ‘银行卡’

□ 多对多

‘多对多’的联系具备以下性质 ($a \in$ 实体集 $A, b \in$ 实体集 B)

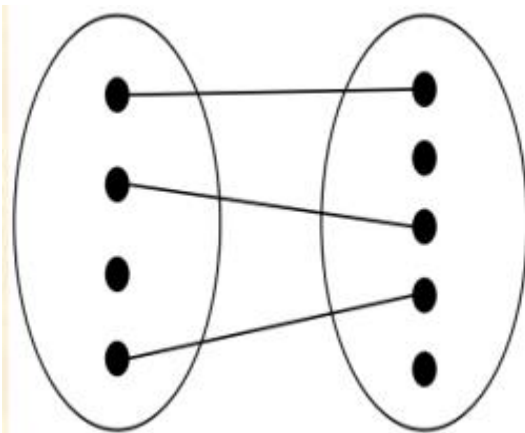
- ① 一个实体 a 可以对应多个实体 b
- ② 一个实体 b 可以对应多个实体 a



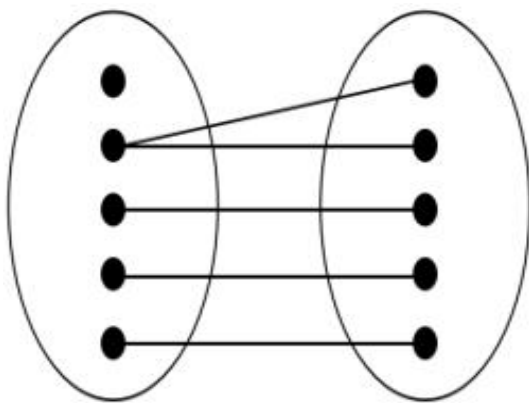
➤ ‘多对多’联系的例子

- 选修: ‘学生’ vs. ‘课程’
- 乘坐: ‘旅客’ vs. ‘航班’

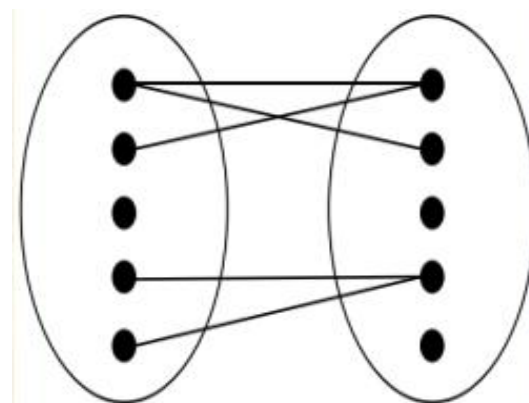
联系的函数关系 (5)



1:1
函数关系



1:m
函数关系

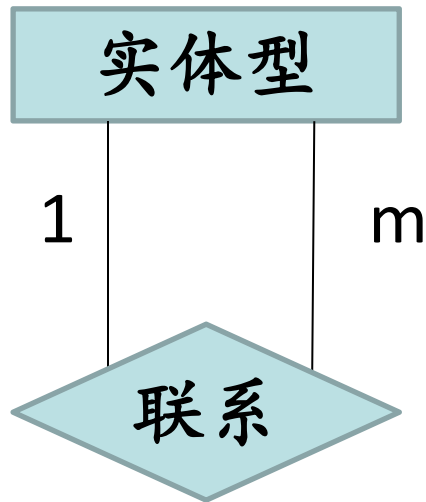


m:n
函数关系

- 不论是哪一种函数关系，也允许一个实体集中的某些实体没有参与到该联系中去（即没有与之相关的连线，或者说在另一个实体集中不存在与之有联系的实体）
- 在“一多对应”的联系中，也允许某些实体之间是“一一对应”的关系；同理，在“多多对应”中，也允许某些实体之间是“一多对应”或“一一对应”的关系。

实体内部的联系

- ❑ 实体内部的联系通常是指：同一个实体集中不同实体之间的联系
- ❑ 实体内部的联系也有‘一对一、一对多、多对多’等多种类型



□ 实体-联系方法 (Entity-Relationship Approach)

➤ 用E-R图来描述现实世界的概念模型

➤ E-R方法也称为E-R模型

➤ E-R模型中的三个基本概念及其图形化表示

● 实体



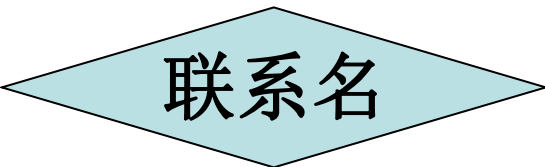
实体名

● 属性



属性名

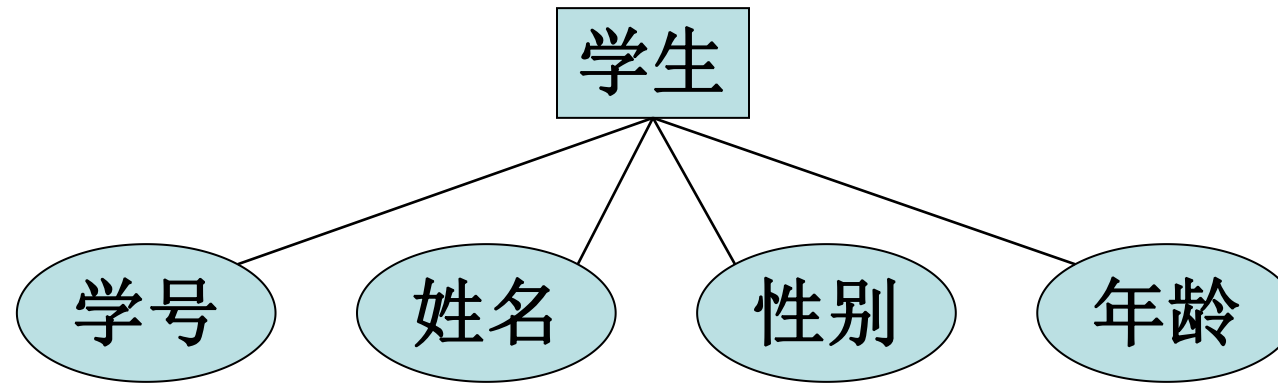
● 联系



联系名

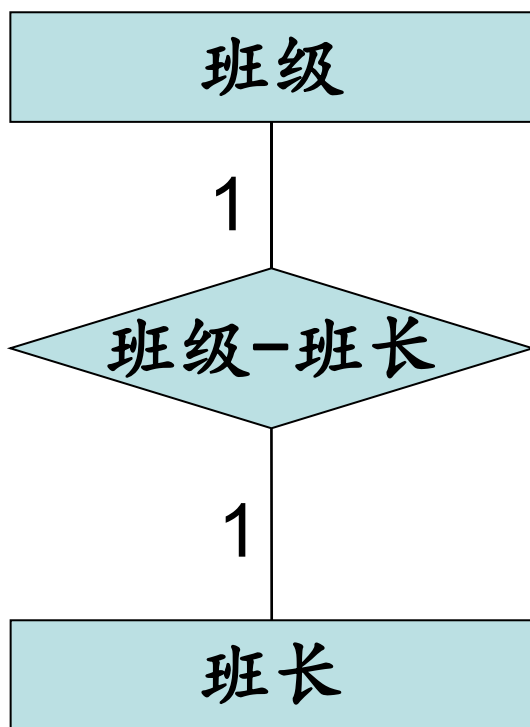
一些例子 (1)

□ 实体型的表示

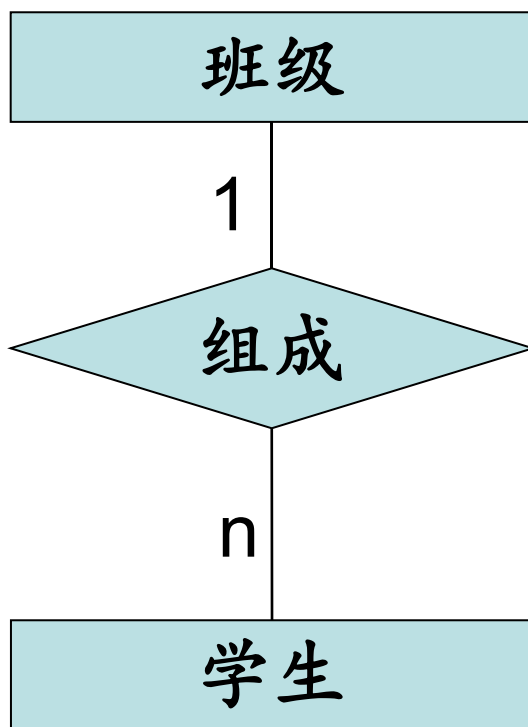


一些例子 (2)

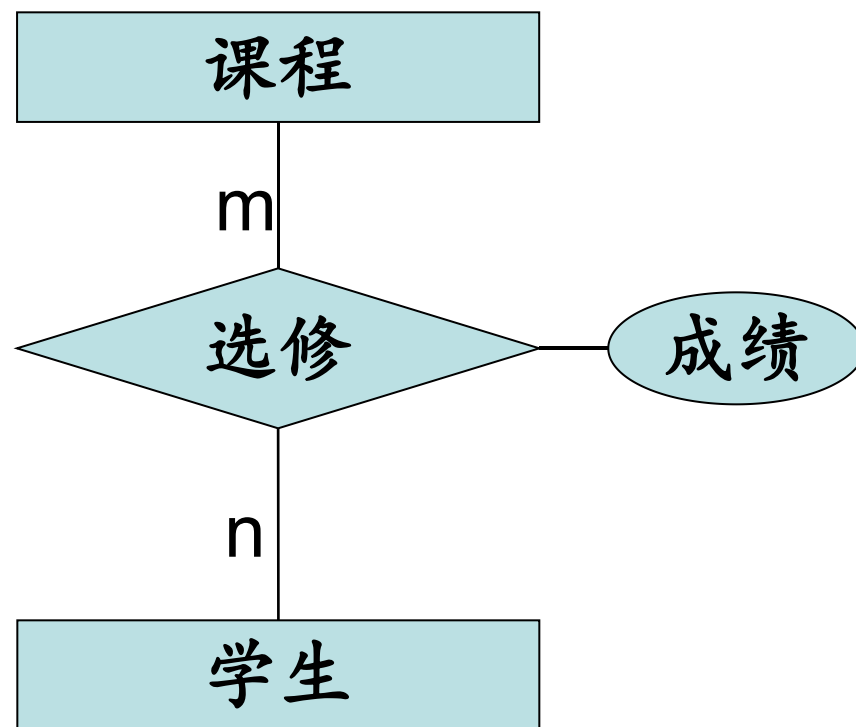
□ 二元联系及联系上的属性



1:1联系



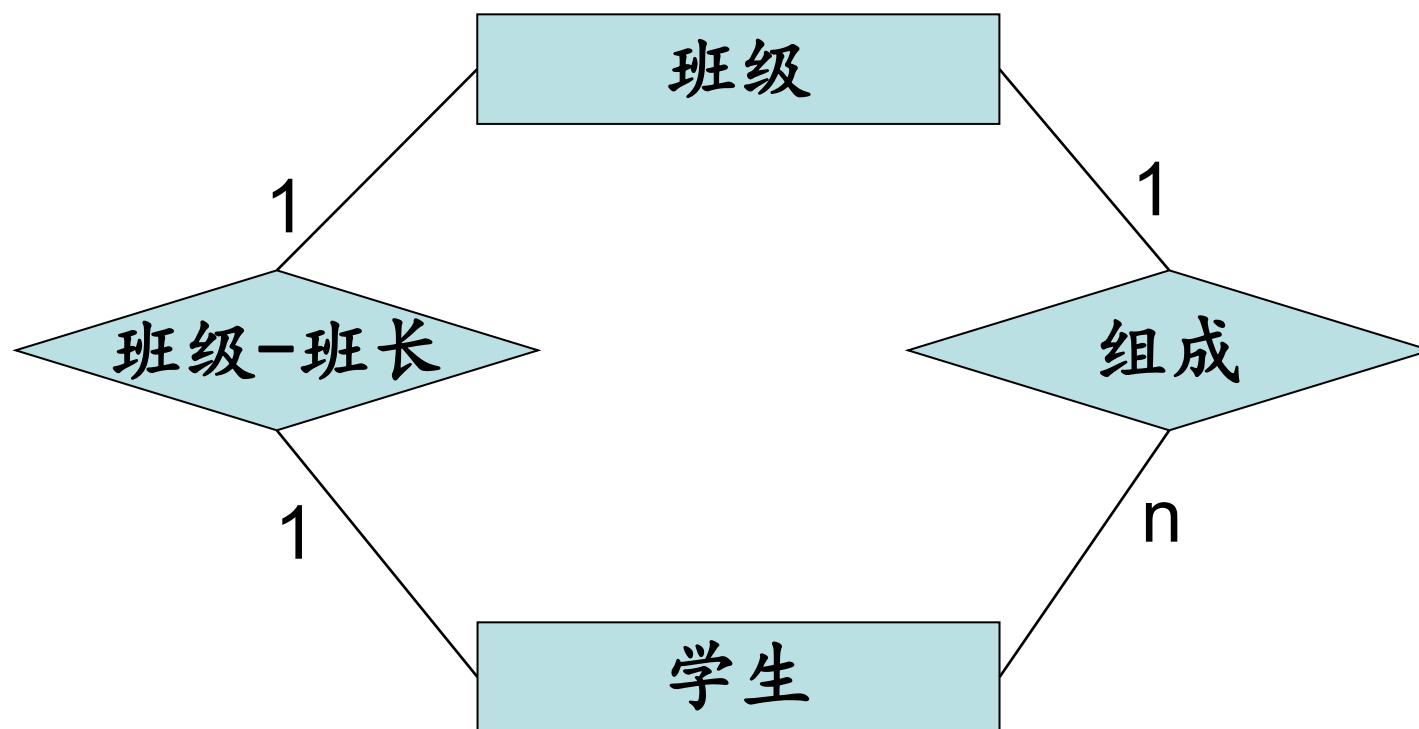
1:n联系



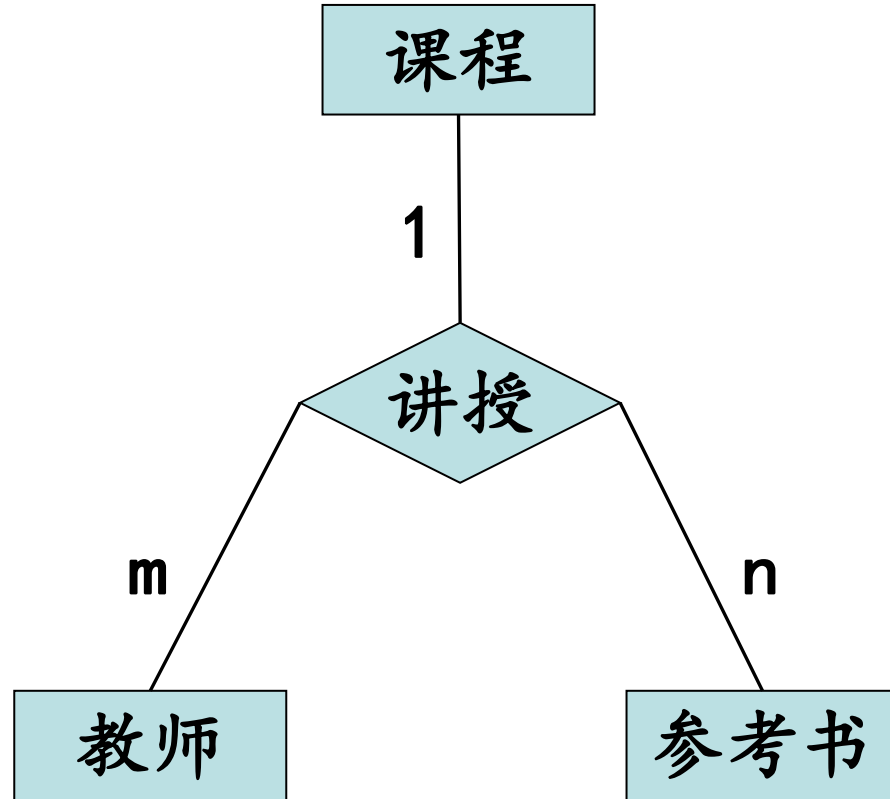
m:n联系

一些例子 (3)

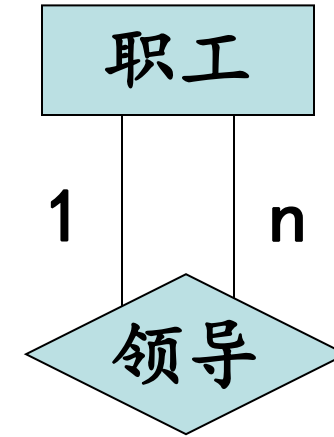
□ 允许在同一组实体型之间建立多个不同的联系



一些例子 (4)

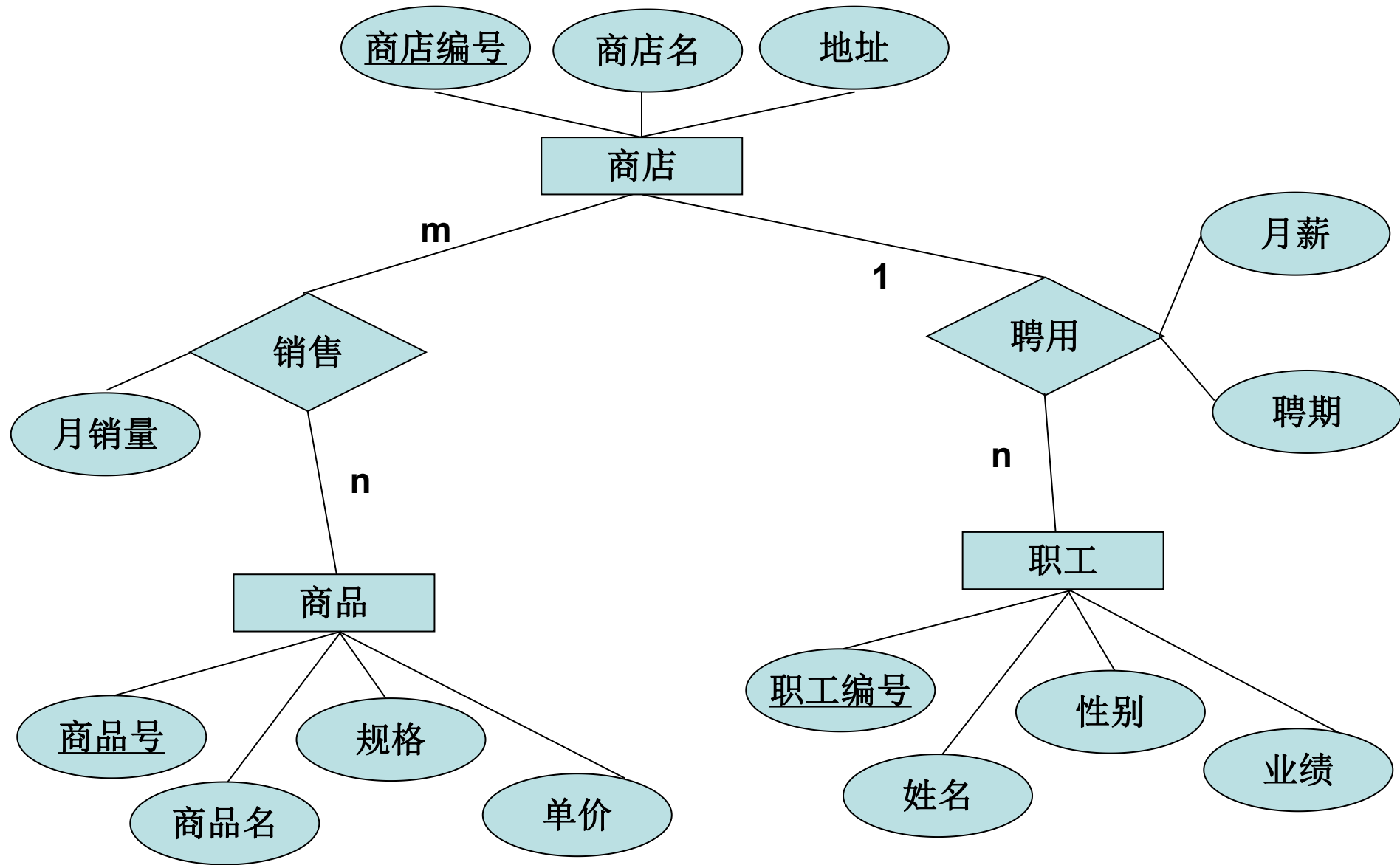


多个实体型间的联系



同一实体型内部的联系

一些例子 (5)



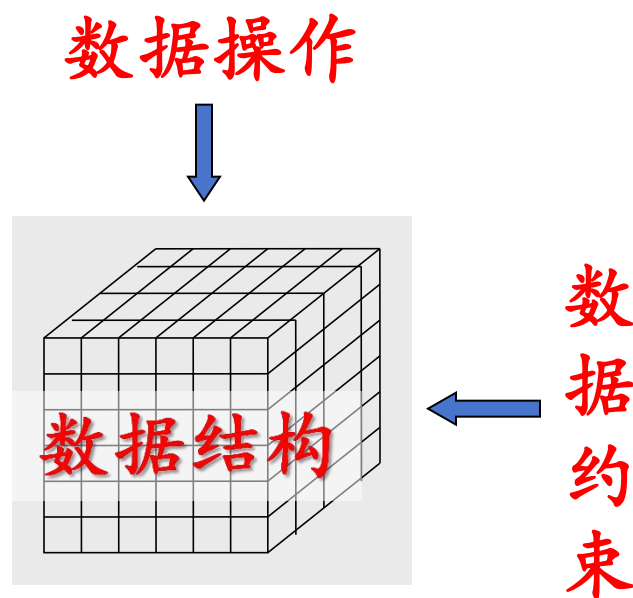
1.2.3 数据模型的组成要素

□ 数据模型 (data model)

➤ 用于描述数据的组织结构，定义在该数据结构上可以执行的操作以及数据之间必须满足的约束条件。

➤ 数据模型的组成要素

- 数据结构
- 数据操作
- 数据约束



❑ 数据模型的数据结构

➤ 描述数据库的组成对象，以及对象之间的联系

❑ 描述的内容

➤ 与对象的类型、内容、性质有关

➤ 与数据之间联系有关

❑ 数据结构是对系统静态特性的描述

❑ 数据结构是一个数据模型的基础，数据操作与数据约束均是建立在相应的数据结构上的。

❑ 为了便于区别，数据模型中的数据结构被称为‘**数据模式**’ (data schema)

❑ ‘**数据模式**’是划分不同类型数据模型的依据。

❑ 数据操作

- 对数据库中各种对象（型）的实例（值）允许执行的操作的集合，包括操作及有关的操作规则

❑ 数据操作的类型

- 查询
- 更新（包括插入、删除、修改）

❑ 数据模型对操作的定义

- 操作的确切含义、操作符号、操作规则（如优先级）
- 实现操作的语言

❑ 数据操作是对系统动态特性的描述

□数据的完整性约束条件

- 一组完整性规则的集合
- 完整性规则：给定的数据模型中数据及其联系所具有的制约和依存规则
- 用以限定符合数据模型的数据库状态以及状态的变化，以保证数据的正确、有效和相容

□数据模型对完整性约束条件的定义

- 反映和规定必须遵守的基本的通用的完整性约束条件。
- 提供定义完整性约束条件的机制，以反映具体应用所涉及的数据必须遵守的特定的语义约束条件。

□ 如何“多快好省”地将信息世界转换为机器世界？

➤ 基本问题，如何在机器世界中表达“低层”数据结构和“高层”数据结构？

➤ 方案1：尽量独立于应用层，采用“中立”的方式表达概念模型

➤ 方案2：在应用层中，使用特定数据结构，并在逻辑模型中高效支持这一数据结构

□ 方案*：通用数据结构采用方案1，关键性数据结构采用方案2

□ 格式化模型

- 层次模型 (Hierarchical Model)
- 网状模型 (Network Model)

□ 关系模型 (Relational Model)

□ 对象模型

- 面向对象数据模型 (Object Oriented Data Model)
- 对象关系数据模型 (Object Relational Data Model)

□ The '60s: 数据库技术的萌芽阶段

1961: IDS (Integrated Data Store)

- Designed by Bachman (73年图灵奖)
- 奠定了网络数据模型(Network Data Model)的基础
 - Conference on Data Systems Languages Database Task Group (CODASYL DBTG)



□ 主要贡献:

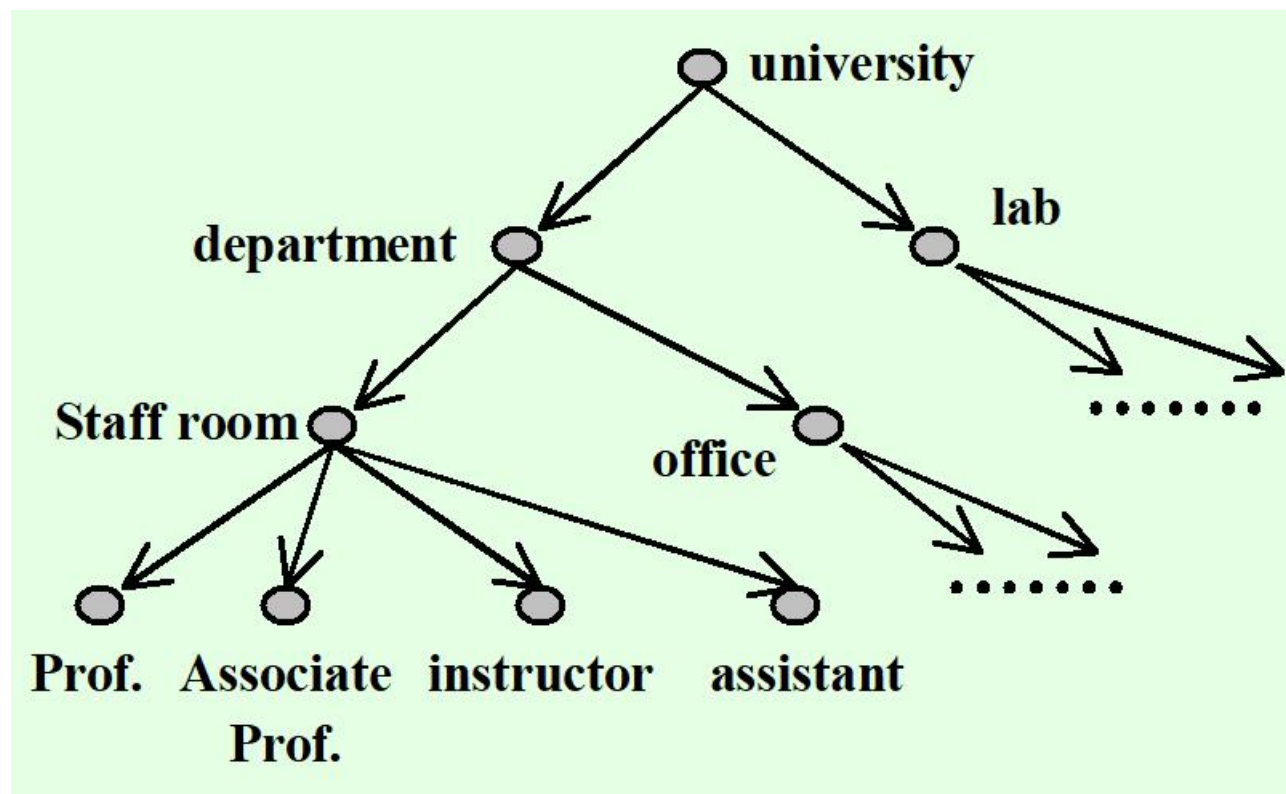
- 在通用电气公司任职期间, 主持设计与开发了最早的网状数据库系统 IDS
- 积极推动与促成数据库标准的制定 (DBTG)

2.4 信息世界与逻辑模型

❑ 1965-1970: **IMS** (Information Management System)

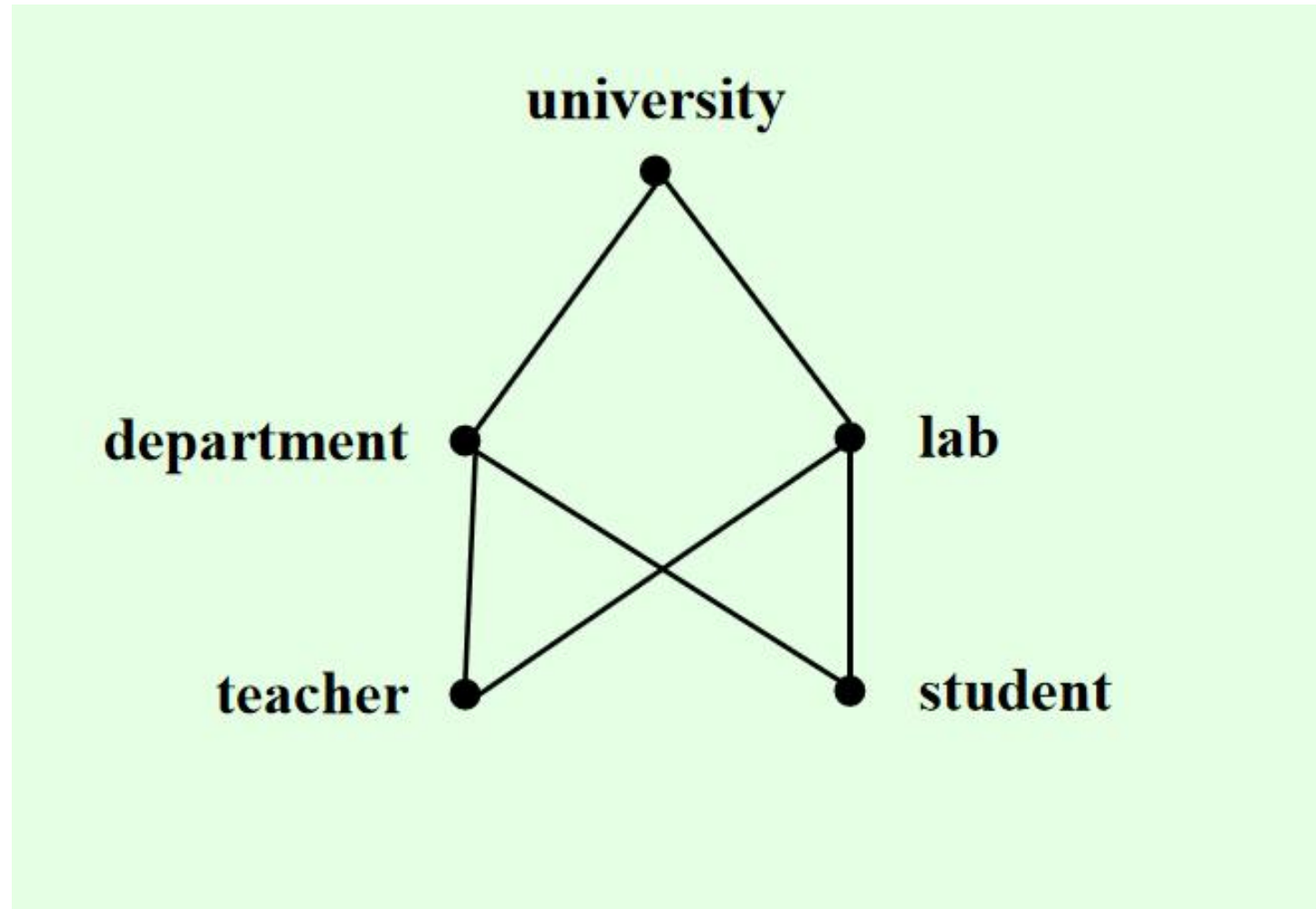
- **Developed by IBM, 奠定了层次数据模型(Hierarchical Data Model)的基础, 允许对数据的多用户存取**

❑ Example of hierarchical data model



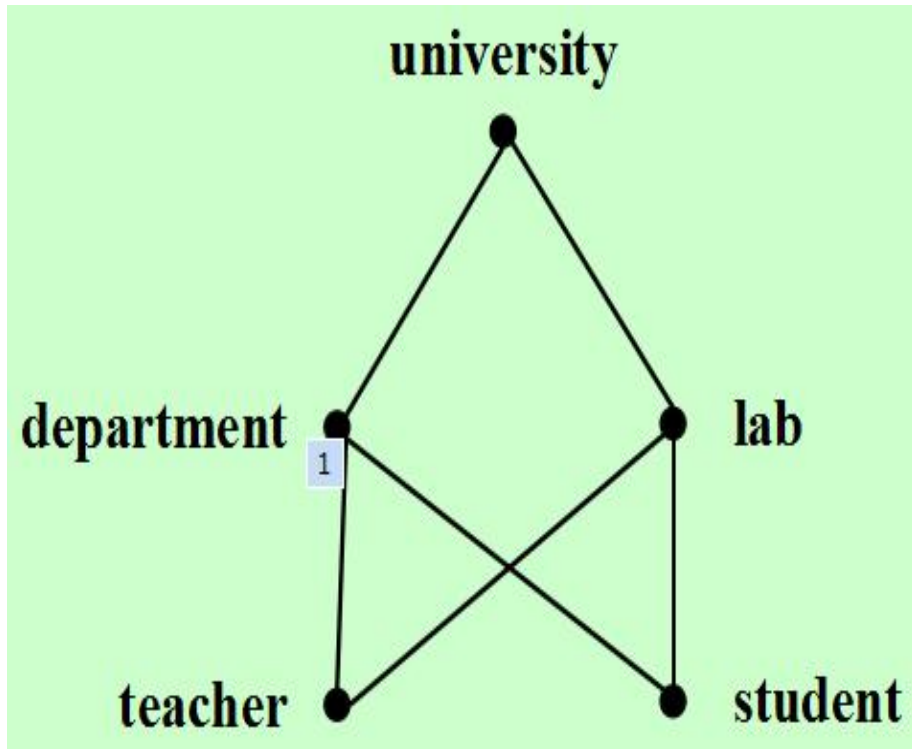
(A directed tree)

Example of network data model

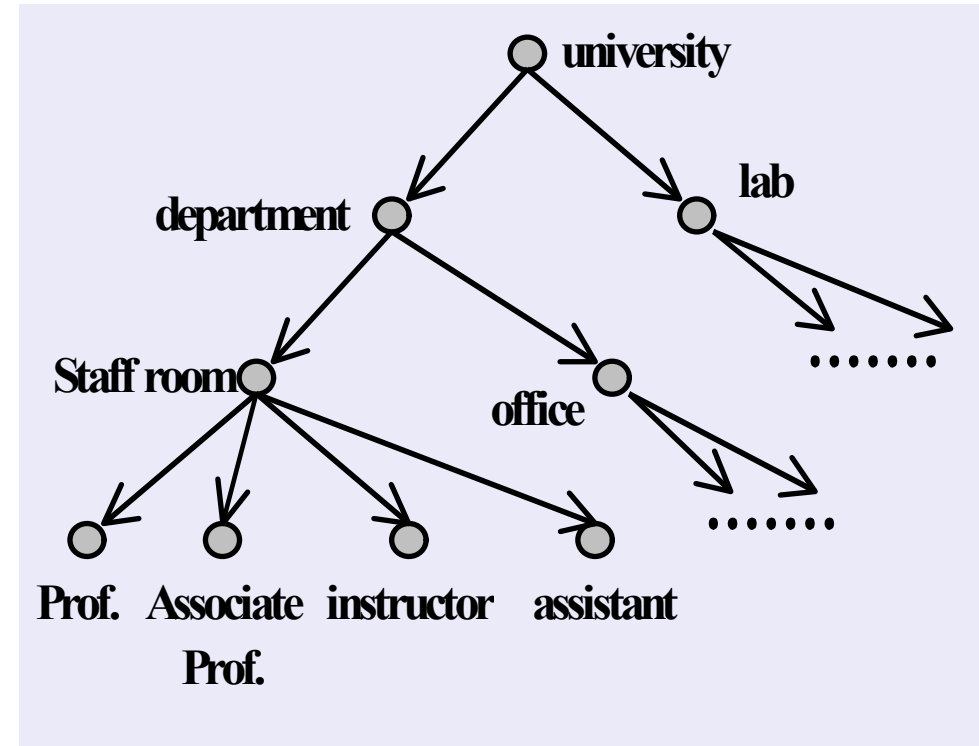


(A directed graph without circuits)

Hierarchical Data Model vs. Network Data Model



(A directed graph without circuits)



(A directed tree)

第一代数据库系统

1.2.7 关系模型 - 数据结构

❑ 在用户观点下，关系模型中数据的逻辑结构是一张二维表，它由行和列组成。



学 号	姓 名	年 龄	性 别	系 名	年 级
2013004	王小明	19	女	社会学	2013
2013006	黄大鹏	20	男	商品学	2013
2013008	张文斌	18	女	法律	2013
...

1.2.7 关系模型 - 基本概念

- **关系 (Relation)** : 一个关系对应通常说的一张表
- **元组 (Tuple)** : 表中的一行即为一个元组
- **属性 (Attribute)** : 表中的一列即为一个属性, 给每一个属性起一个名称即属性名
- **主码 (Key)** : 也称码键。表中的某个属性组, 它可以唯一确定一个元组
- **域 (Domain)** : 是一组具有相同数据类型的值的集合。属性的取值范围来自某个域。
- **分量** : 元组中的一个属性值。
- **关系模式**
 - 对关系的描述: 关系名 (属性1, 属性2, ..., 属性n)
 - 例: 学生 (学号, 姓名, 年龄, 性别, 系名, 年级)

1.2.7 关系模型 - 模型的规范化

- ❑ 关系必须是规范化的，满足一定的规范条件
 - 最基本的规范条件：关系的每一个分量必须是一个不可分的数据项，不允许表中还有表

❑ 不符合规范化要求的例子

职工号	姓名	职称	工资			扣除		实发
			基本工资	岗位津贴	业绩津贴	三险	个人所得税	
86051	陈平	讲师	1305	1200	1850	160	112	4083
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

关系模型的操作与完整性约束

- ❑ 数据操作（查询、插入、删除、更新）是集合操作，操作对象和操作结果都是关系
- ❑ 存取路径对用户隐蔽，用户只要指出“干什么”，不必详细说明“怎么干”
- ❑ 关系的完整性约束条件
 - 实体完整性
 - 参照完整性
 - 用户定义的完整性

关系模型的优缺点

优点

- ❑ 建立在严格的数学概念的基础上
- ❑ 概念单一
 - 实体和各类联系都用关系来表示
 - 对数据的检索结果也是关系
- ❑ 关系模型的存取路径对用户透明
 - 具有更高的数据独立性，更好的安全保密性
 - 简化了程序员的工作和数据库开发建立的工作

缺点

- ❑ 存取路径对用户透明，查询效率往往不如格式化数据模型
- ❑ 为提高性能，必须对用户的查询请求进行优化，增加了开发DBMS的难度

1.1 数据库系统概述

1.2 数据模型

1.3 数据库系统的结构

1.4 数据库系统的组成

1.5 小结

1.3 数据库系统的结构

❑ 从数据库应用开发人员角度看

- 数据库系统通常采用三级模式结构，是数据库系统内部的系统结构

❑ 从数据库最终用户角度看，数据库系统的结构分为：

- 单用户结构
- 主从式结构
- 分布式结构
- 客户-服务器
- 浏览器-应用服务器 / 数据库服务器多层结构等

□ 模式 (Schema)

- 数据库逻辑结构和特征的描述
- 是‘**型**’的描述，不涉及具体‘**值**’，反映的是数据的结构及其联系
- 模式是相对稳定的

□ 例如：

- 学生记录的‘**型**’：（学号，姓名，性别，系别，年龄，籍贯）
- 一个学生记录的‘**值**’：
(201315130, 李明, 男, 计算机系, 19, 江苏省南京市)

□ 实例 (Instance)

- 反映数据库某一时刻的状态，是模式的一个具体值
- 同一个模式可以有很多实例
- 实例随数据库中的数据更新而变动

模式和实例 (2)

❑ 例如：在学生选课数据库模式中，包含学生记录、课程记录和学生选课记录

➤ 2013年的一个学生数据库实例，包含：

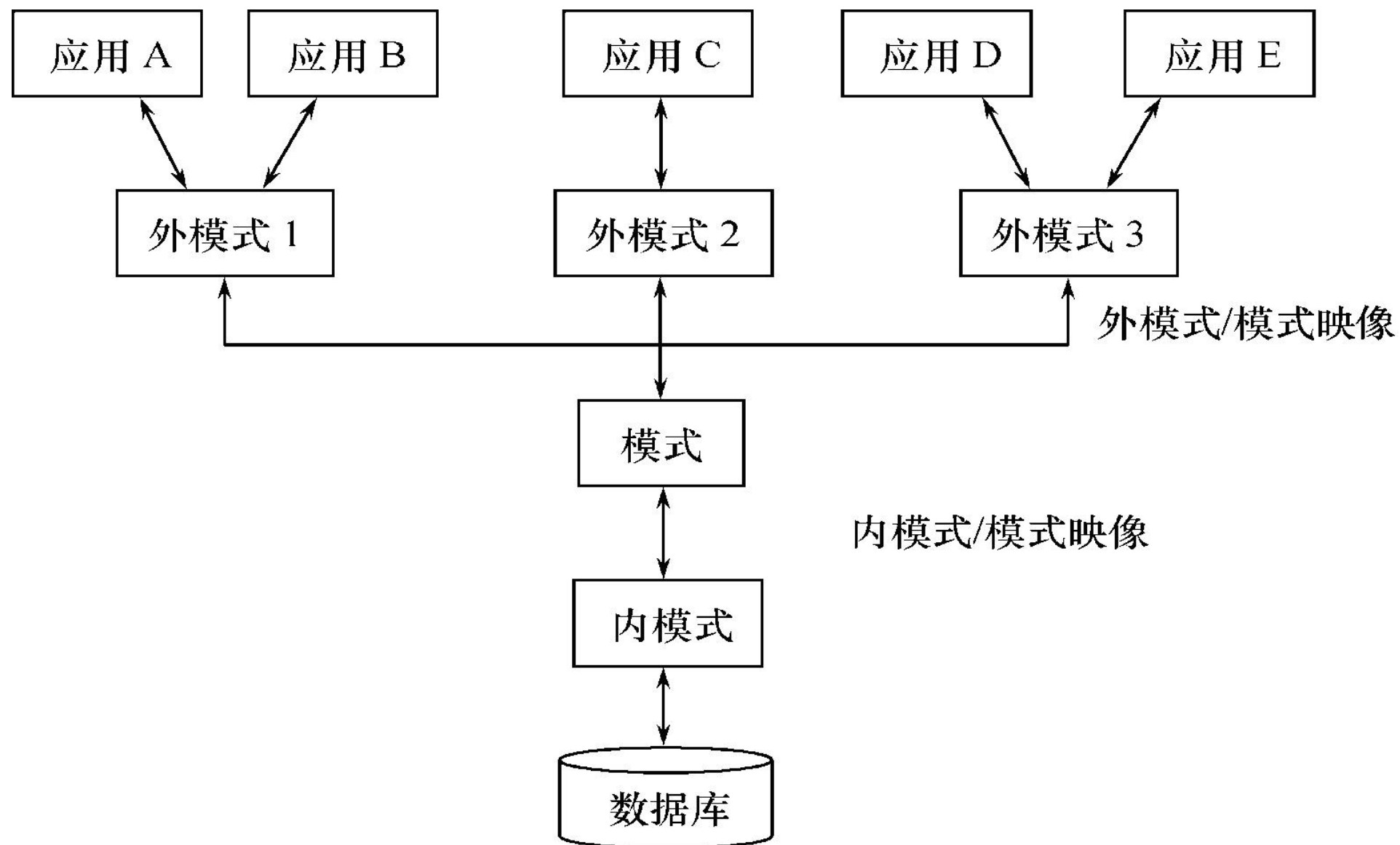
- 2013年学校中所有学生的记录
- 2013年学校开设的所有课程的记录
- 2013年所有学生选课的记录

➤ 2012年的一个学生数据库实例，包含：

- 2012年学校中所有学生的记录
- 2012年学校开设的所有课程的记录
- 2012年所有学生选课的记录

➤ 2012年度学生数据库模式对应的实例与2013年度学生数据库模式对应的实例是不同的

图1.16 数据库系统的三级模式结构



模式 (Schema)

□模式（也称‘逻辑模式’）

- 数据库中全体数据的逻辑结构和特征的描述
- 是所有用户的公共数据视图

□一个数据库只有一个模式

□模式的地位：是数据库系统模式结构的中间层

- 与数据的物理存储细节和硬件环境无关
- 与具体的应用程序、开发工具及高级程序设计语言无关

□模式的定义

- 数据的逻辑结构（数据项的名字、类型、取值范围等）
- 数据之间的联系
- 数据有关的安全性、完整性要求

□ 外模式 (External Schema)

- 也称‘子模式’或‘用户模式’
- 是对数据库用户（包括应用程序员和最终用户）使用的局部数据的逻辑结构和特征的描述
- 属于数据库用户的数据视图，是与某一应用有关的数据的逻辑表示

外模式 (2)

□外模式的地位：介于模式与应用之间

➤模式与外模式的关系：一对多

- 外模式通常是模式的子集
- 一个数据库可以有多个外模式。反映了不同的用户的应用需求、看待数据的方式、对数据保密的要求
- 对模式中同一数据，在外模式中的结构、类型、长度、保密级别等都可以不同

➤外模式与应用的关系：一对多

- 同一外模式也可以为某一用户的多个应用系统所使用
- 但一个应用程序只能使用一个外模式

□外模式的用途

- 保证数据库安全性的一个有力措施
- 每个用户只能看见和访问所对应的外模式中的数据

内模式 (Internal Schema)

□ 内模式 (也称‘存储模式’或‘物理模式’)

- 是数据物理结构和存储方式的描述
- 是数据在数据库内部的表示方式
 - 记录的存储方式 (例如, 顺序存储、堆存储、hash存储等)
 - 索引的组织方式
 - 数据是否压缩存储
 - 数据是否加密
 - 数据存储记录结构的规定

□ 一个数据库只有一个内模式

- ❑ 三级模式是对数据的三个抽象级别
- ❑ 二级映象在数据库管理系统内部实现这三个抽象层次的联系和转换
 - 外模式 / 模式映像
 - 模式 / 内模式映像

□**模式**：描述的是数据的全局逻辑结构

□**外模式**：描述的是数据的局部逻辑结构

- 同一个模式可以有任意多个外模式
- 每一个外模式，数据库系统都有一个外模式 / 模式映象，定义外模式与模式之间的对应关系
- 映象定义通常包含在各自外模式的描述中

□**保证数据的逻辑独立性**

- 当模式改变时，数据库管理员对外模式 / 模式映象作相应改变，使外模式保持不变
- 应用程序是依据数据的外模式编写的，应用程序不必修改，保证了数据与程序的逻辑独立性，简称数据的逻辑独立性

❑ 模式 / 内模式映象定义了数据全局逻辑结构与存储结构之间的对应关系。

➤ 例如，说明逻辑记录和字段在计算机内部是如何表示的

❑ 数据库中模式 / 内模式映象是唯一的

❑ 该映象定义通常包含在模式描述中

❑ 保证数据的物理独立性

➤ 当数据库的存储结构改变了（例如选用了另一种存储结构），数据库管理员修改模式 / 内模式映象，使模式保持不变。

➤ 应用程序不受影响。保证了数据与程序的物理独立性，简称数据的物理独立性。

□数据库模式

- 即全局逻辑结构是数据库的中心与关键
- 独立于数据库的其他层次
- 设计数据库模式结构时应首先确定数据库的逻辑模式

□数据库的内模式

- 依赖于它的全局逻辑结构
- 独立于数据库的用户视图，即外模式
- 独立于具体的存储设备
- 将全局逻辑结构中所定义的数据结构及其联系按照一定的物理存储策略进行组织，以达到较好的时间与空间效率

□数据库的外模式

- 面向具体的应用程序
- 定义在逻辑模式之上
- 独立于存储模式和存储设备
- 当应用需求发生较大变化，相应外模式不能满足其视图要求时，该外模式就得做相应改动
- 设计外模式时应充分考虑到应用的扩充性

□特定的应用程序

- 在外模式描述的数据结构上编制的
- 依赖于特定的外模式
- 与数据库的模式和存储结构独立
- 不同的应用程序有时可以共用同一个外模式

□数据库的二级映像

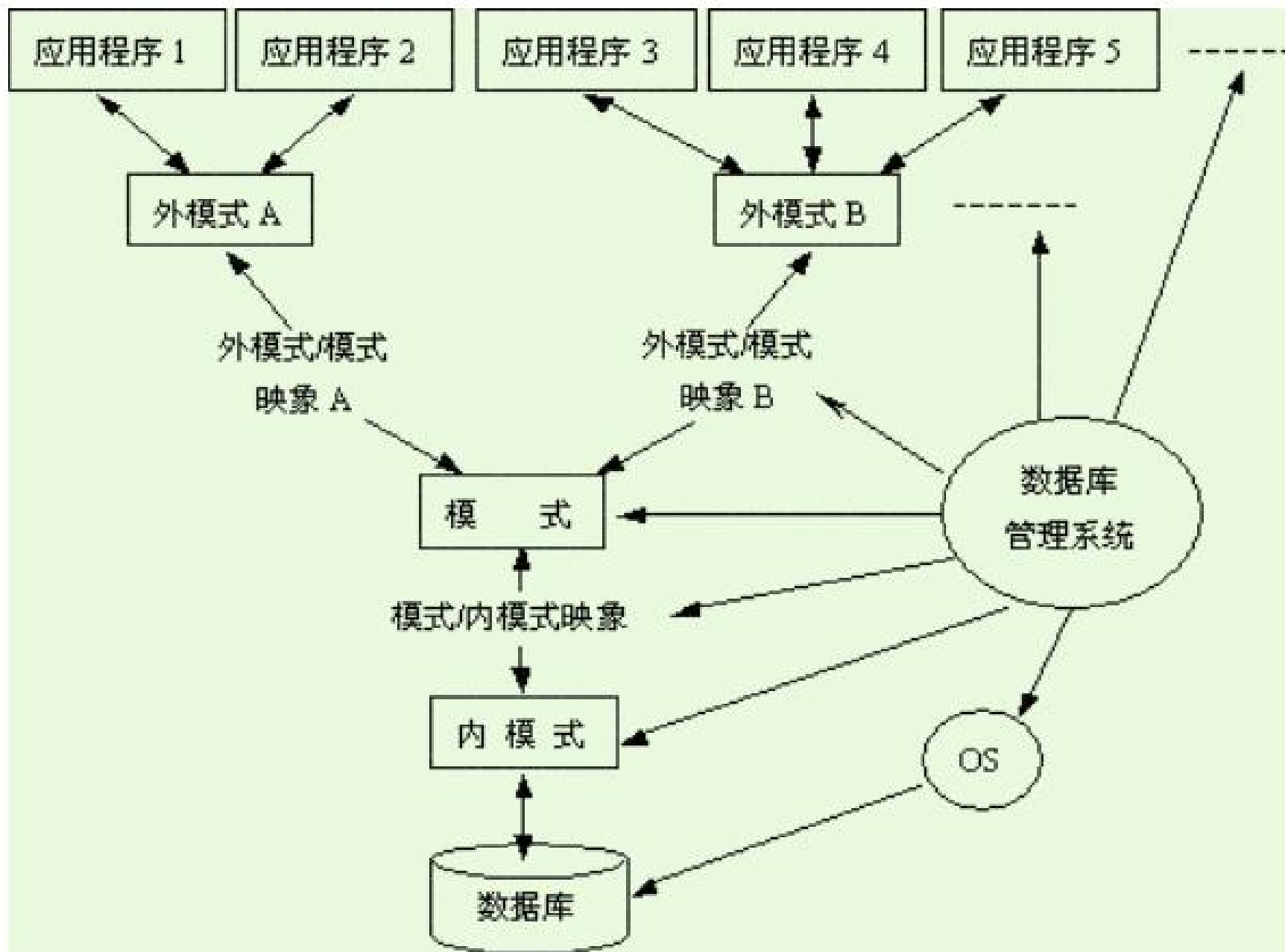
- 保证了数据库外模式的稳定性
- 从底层保证了应用程序的稳定性，除非应用需求本身发生变化，否则应用程序一般不需要修改

□数据与程序之间的独立性，使得数据的定义和描述可以从应用程序中分离出去

□数据的存取由数据库管理系统管理

- 简化了应用程序的编制
- 大大减少了应用程序的维护和修改

三级模式与 DBMS、应用程序之间的关系



□从数据库最终用户角度看，数据库系统的结构分为：

➤单用户结构

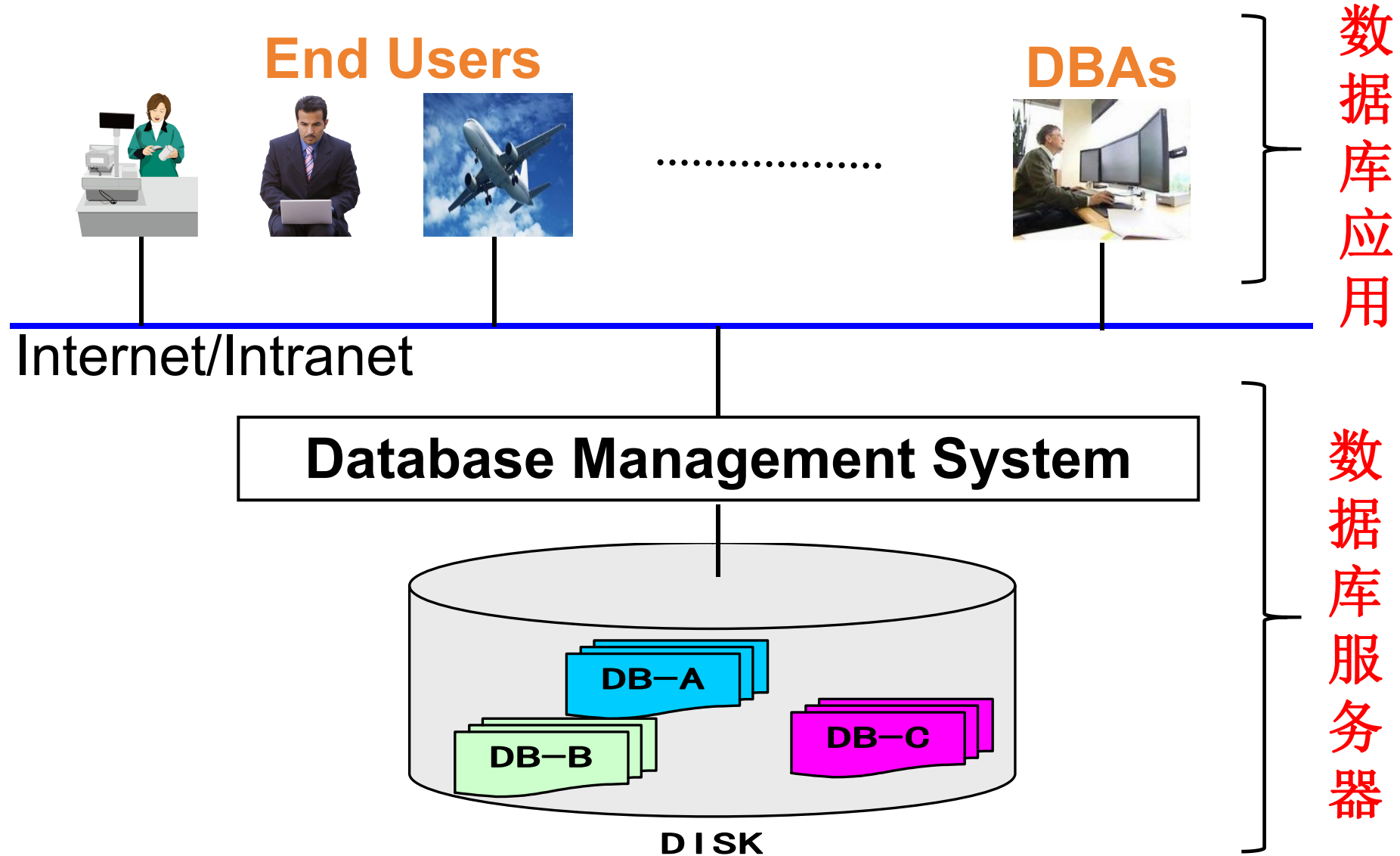
➤主从式结构

➤分布式结构

➤客户-服务器

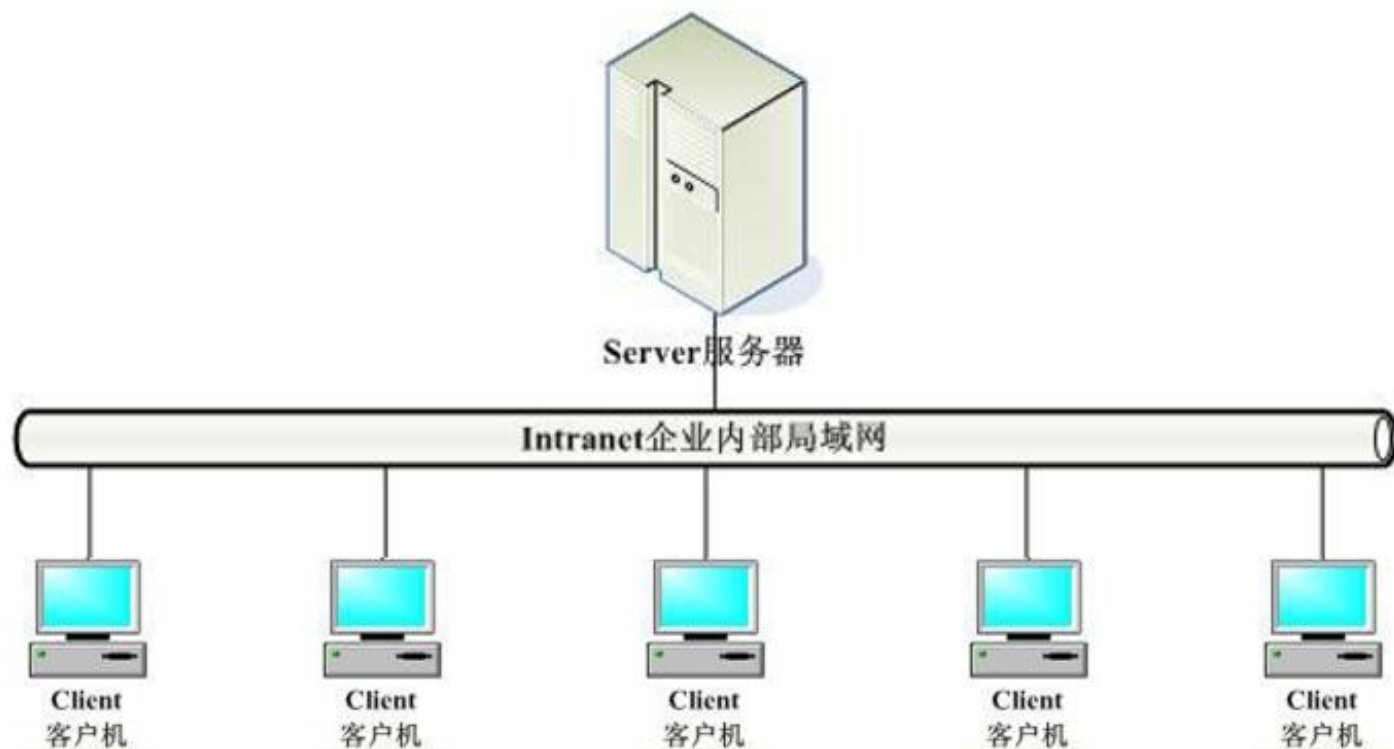
➤浏览器-应用服务器 / 数据库服务器多层结构等

Architecture of Database System



Two-tier Architecture vs. Three-tier Architecture

- 在两层C/S架构中，数据库和DBMS运行在数据库服务器中，数据库应用程序运行在客户机中
- 两者之间通过局域网实现数据访问



(两层客户-服务器架构)

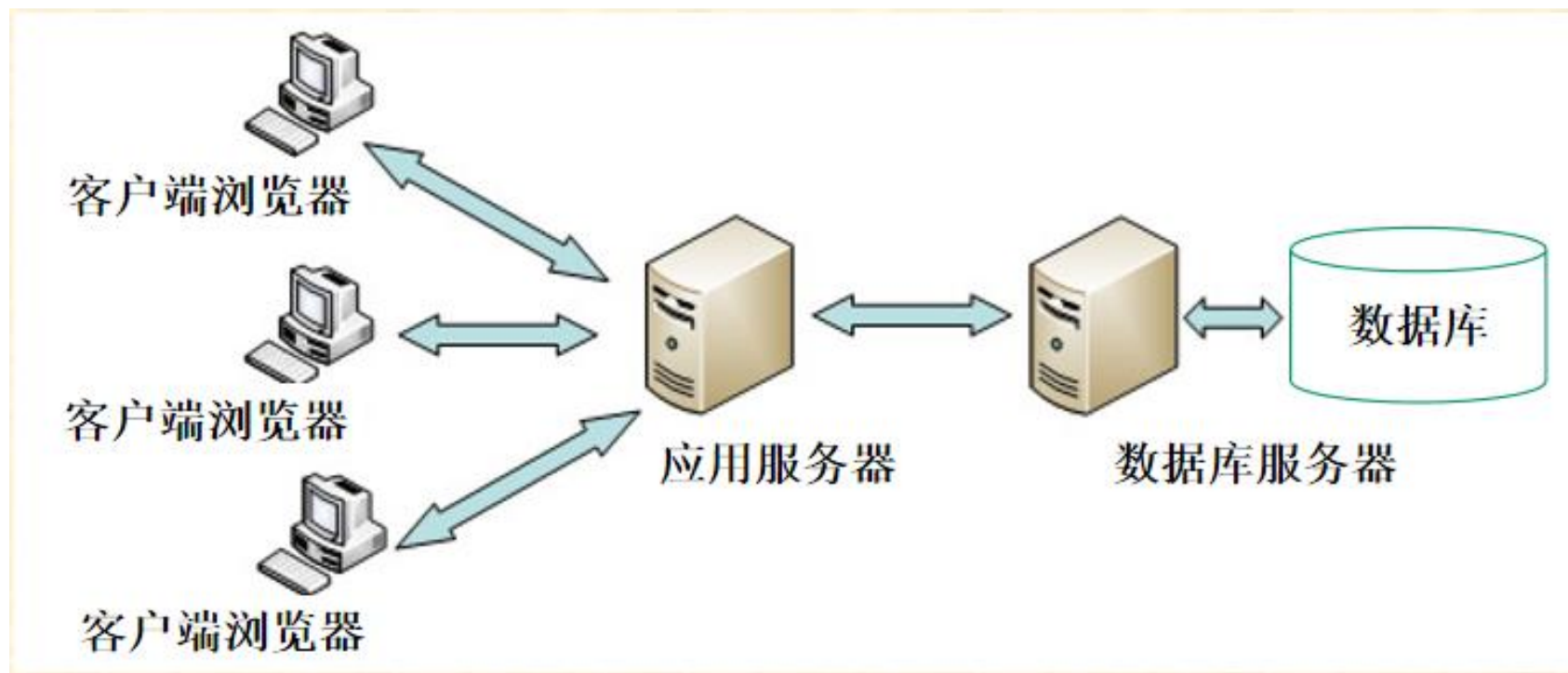
Two-tier Architecture vs. Three-tier Architecture

- 但更多的时候，数据库服务器及其应用程序可能分布在距离遥远的不同地方（如下图所以）
- 它们相互之间无法通过企业内部的局域网相连，只能通过更广阔的互联网来实现数据访问和数据传输。



Two-tier Architecture vs. Three-tier Architecture

- 在三层B/S架构中，数据库和DBMS运行在数据库服务器中，数据库应用程序运行在应用服务器(也称“Web服务器”)中
- 用户客户端只需要安装常用的浏览器，负责接收用户输入和结果展示



(三层浏览器-应用服务器/数据库服务器架构)

1.1 数据库系统概述

1.2 数据模型

1.3 数据库系统的结构

1.4 数据库系统的组成

1.5 小结

数据库系统的组成-软硬件平台

硬件平台

- ❑ 数据库系统对硬件资源的要求
 - 足够大的内存
 - 足够的大的磁盘或磁盘阵列等设备
 - 较高的通道能力，提高数据传送率

软件平台

- ❑ 数据库管理系统
- ❑ 支持数据库管理系统运行的操作系统
- ❑ 与数据库接口的高级语言及其编译系统
- ❑ 以数据库管理系统为核心的应用开发工具
- ❑ 为特定应用环境开发的数据库应用系统

数据库系统的组成-人员

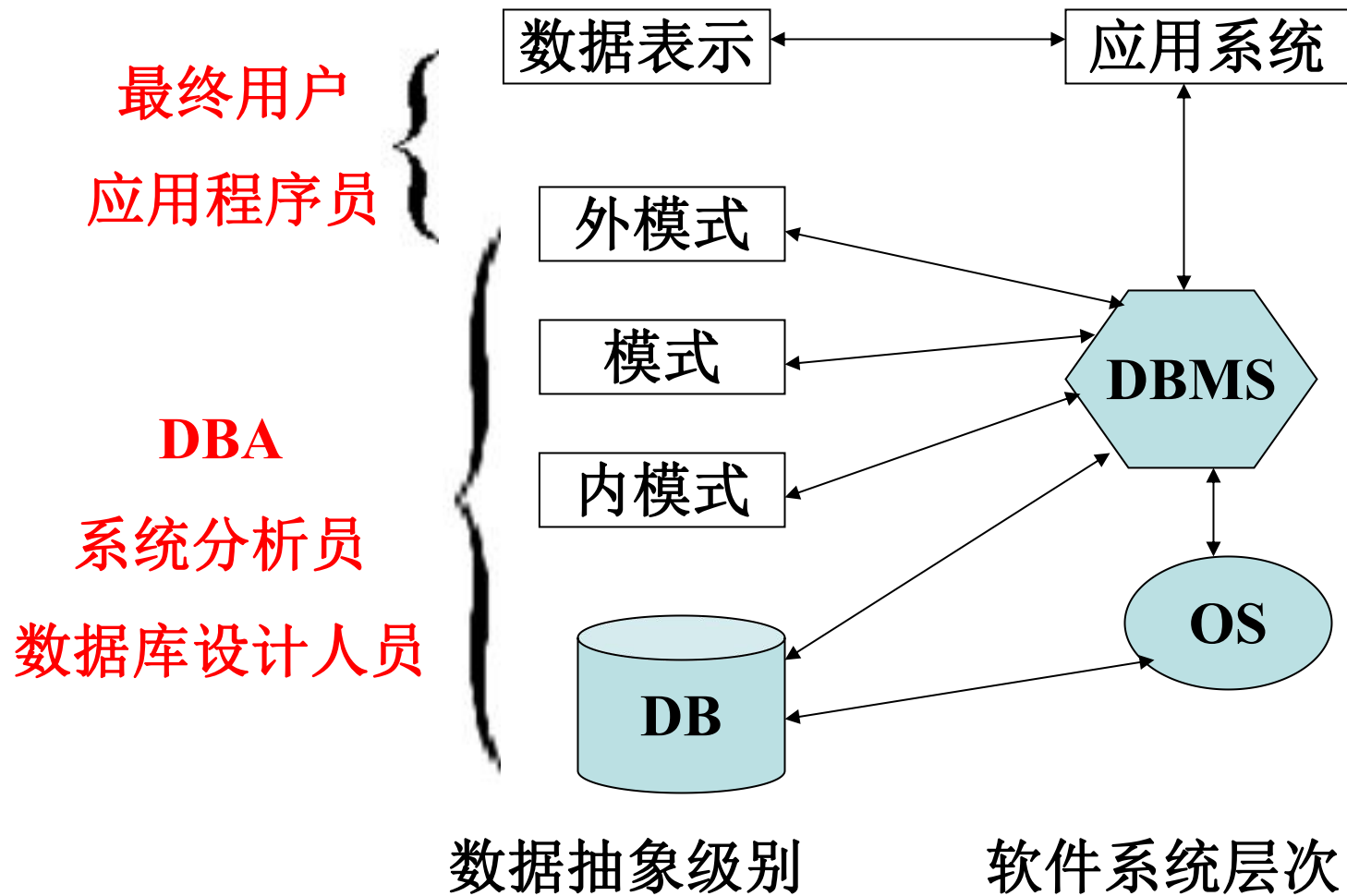


图1.17 各种人员的数据视图

数据库管理员 (DBA)

- ❑ 决定数据库中的信息内容和结构
- ❑ 决定数据库的存储结构和存取策略
- ❑ 定义数据的安全性要求和完整性约束条件
- ❑ 监控数据库的使用和运行
 - 周期性转储数据库：数据文件 & 日志文件
 - 系统故障恢复
 - 介质故障恢复
 - 监视审计文件
- ❑ 数据库的改进和重组
 - 性能监控和调优
 - 定期对数据库进行重组，以提高系统的性能
 - 需求增加和改变时，数据库须需要重构造

□ 系统分析员

- 负责应用系统的需求分析和规范说明
- 与用户及数据库管理员结合，确定系统的软硬件配置
- 参与数据库系统的概要设计

□ 数据库设计人员

- 参加用户需求调查和系统分析
- 确定数据库中的数据
- 设计数据库各级模式

□应用程序员

- 设计和编写应用系统的程序模块
- 进行调试和安装

□最终用户(End User): 最终用户通过应用系统的用户接口使用数据库

➤偶然用户

- 不经常访问数据库，但每次访问时往往需要不同的数据库信息
- 企业或组织机构的高中级管理人员（一般通过软件工具来访问数据库）

➤简单用户

- 通过操作业务系统来使用（查询和更新）数据库中的数据
- 银行的职员、机票预定人员、旅馆总台服务员（不需要直接访问数据库）

➤复杂用户

- 工程师、科学家、经济学家、科技工作者等
- 直接使用数据库语言访问数据库，甚至能够基于数据库管理系统的应用程序接口编制自己的应用程序

1.1 数据库系统概述

1.2 数据模型

1.3 数据库系统的结构

1.4 数据库系统的组成

1.5 小结

❑ 数据库系统概述

- 数据库的基本概念
- 数据管理的发展过程
- 数据库系统的特点

❑ 数据模型

- 数据模型的基本概念
- 数据模型的三要素
- 三种主要的数据模型

❑ 数据库系统内部的系统结构

- 三级模式及其相互关系
- 二级映像
- 三级模式、二级映像、数据独立性的关系

❑ 数据库系统的组成

- 数据库系统的组成成分
- 数据库用户

本章复习思考题

1. 理解下述四个概念及其相互关系：数据库，数据库管理系统，数据库系统，数据库应用系统
2. 数据库管理系统的主要功能有哪些？
3. 数据库用户分哪几种类型？数据库管理员的主要工作是什么？
4. 什么是两层客户-服务器架构？什么是三层浏览器-应用服务器/数据库服务器架构？
5. 什么是数据冗余存储？什么是数据的一致性和不一致性？
6. 数据库系统的特点有哪些？
7. 什么是数据模型？数据模型的三要素是什么？为什么要建立三种不同抽象层次上的数据模型？
8. 什么是数据模式？数据库系统中的三级模式分别用于描述哪个层次上的数据结构？
9. 什么是数据独立性？什么是逻辑独立性和物理独立性？数据库系统是如何实现数据独立性的？
10. 试述数据库系统的组成。