

数据库系统原理

教程：数据库系统概论（第6版）

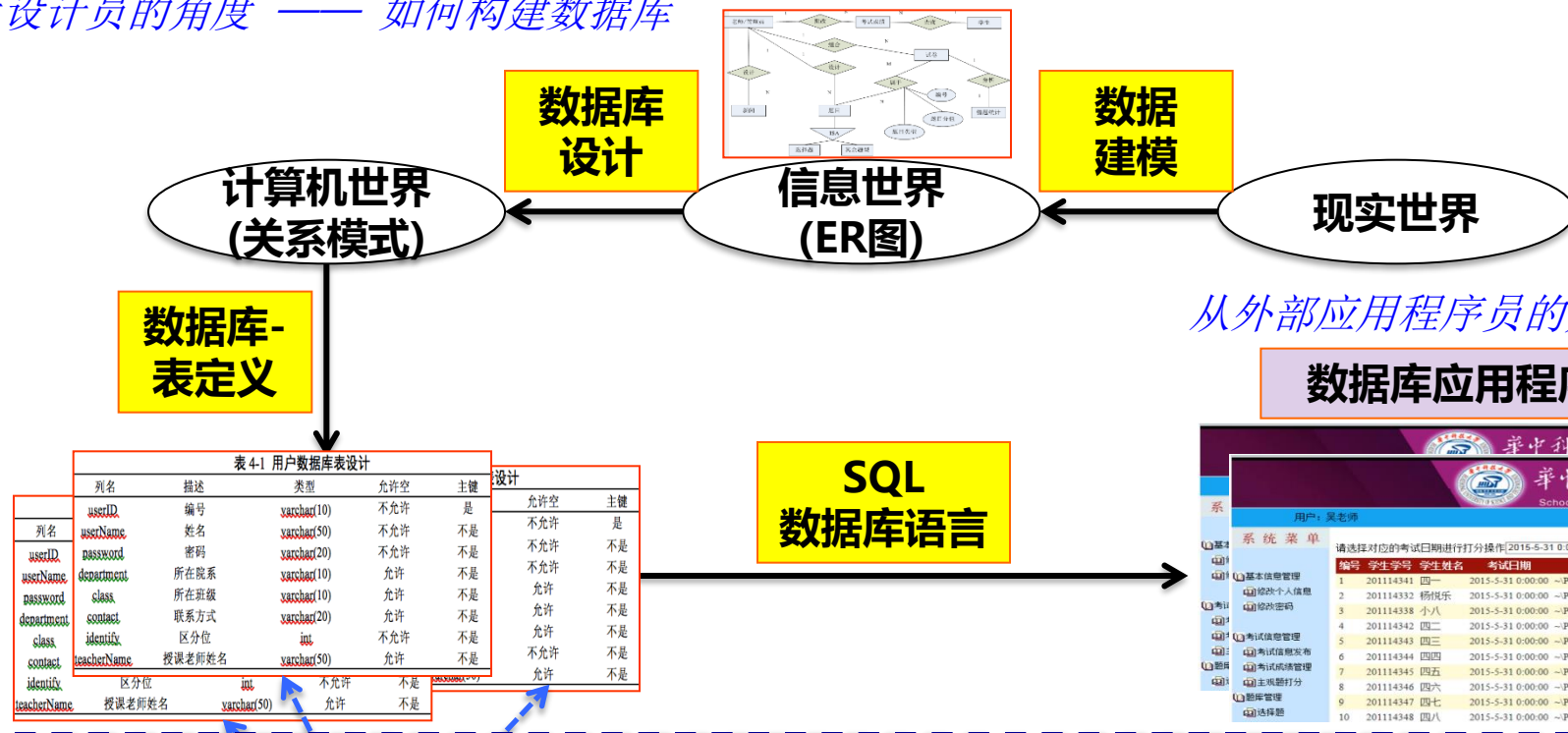
结合：CMU 15-445/645 INTRO TO DATABASE SYSTEMS

华中科技大学 计算机学院

左琼

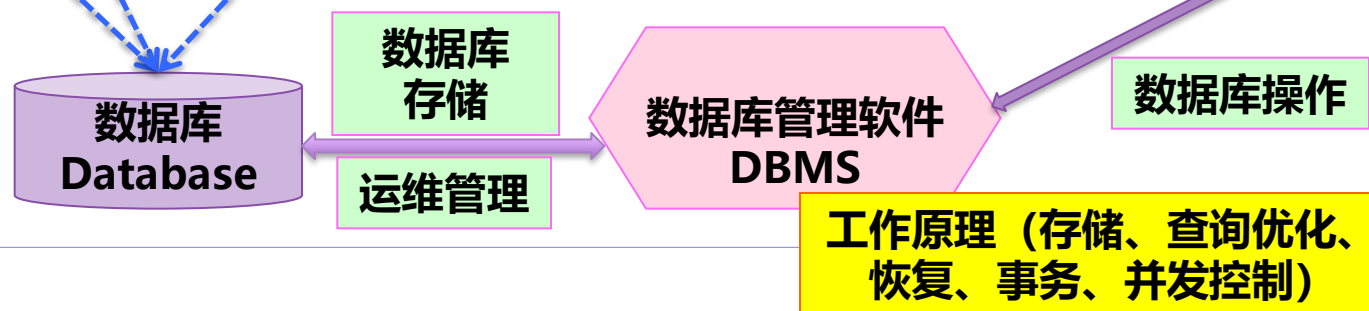
数据库系统原理课程规划

从数据库设计员的角度 —— 如何构建数据库



从外部应用程序员的角度 —— 如何使用数据库

从内部系统组成的角度 —— 如何实现数据库系统软件



课程目标

- 了解并结合关系型数据库系统，深入理解数据库系统的基本概念，结构、工作原理和运行机制。
- 掌握关系数据模型及关系数据语言，能熟练应用SQL语言表达各种数据操作。
- 掌握E-R模型的概念和方法，关系数据库规范化理论和数据库设计方法，通过上机实践的训练，初步具备进行数据库应用系统开发的能力。
- 了解数据库系统的存储机制、查询优化、恢复、并发控制等技术以及最新的数据库技术。

教材及参考书

□ 教材：

- 王珊，萨师煊：数据库系统概论(第6版)，高教出版社，2023.3

□ 参考资料：

- CMU 15-445/645 — INTRO TO DATABASE SYSTEMS
- 数据库系统概念Database System Concepts. Abraham, Silberschatz等著，杨冬青，唐世渭等译，机械工业出版社，任一版
- 杨冬青，唐世渭等译。数据库系统实现（Database System Implementation），原书第二版。机械工业出版社，2010
- 岳丽华，金培权等译。数据库系统基础教程（A First Course in Database Systems），原书第三版。机械工业出版社，2011
- 其它 数据库系统原理教程等

数据库系统原理实践

- 时间：10-16周（4次线上，4次线下）
- 上机：
 - Educoder（MySQL 8.0）
 - 内容包括：SQL练习、数据库设计、并发控制、B+树实现、部分数据库应用Java开发函数等
- 实践课计分：
 - Educoder成绩
 - 平时成绩（平时记录、平时检查）
- /* 请同学们提前(第三章时)下载mysql 8.0，方便SQL练习 */

线上学习资源

□ 华中科技大学计算机学院 “数据库系统原理” 慕课（第9次开课）

- <https://www.icourse163.org/course/HUST-1449788170>

- 用于课前预习、学习互动、课后复习，内含丰富的客观题题库，欢迎同学们按章节刷题，加深对知识点的理解。（不算到平时成绩中）。

□ 华为在线课程系列

- <https://edu.huaweicloud.com/roadmap/colleges.html>

- 华为在线课程数据库系列有两类入口：

 - 一类通过在线课程的技术领域选择“数据库”进入；

 - 一类通过技术领域选择“鲲鹏”后，查找openGauss课程进入。

有兴趣的同学可以选看！

教学组织及成绩评定

- **学在华科大数据库系统原理课堂**（平时作业、课件）；
- **微助教**（签到）课堂编号：**OZ956**；
- 请推荐一位班干部协助管理课堂和班级，建立**QQ群**，方便课程日常交流及通知发布。
- **成绩评定**
 - 总评 = 平时成绩15% + 随堂测2次 15% + 期末考试卷面成绩*70%

学在华科大课堂

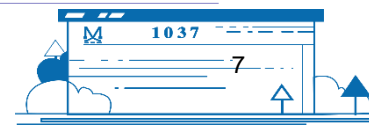
邀请码：**#787814**

企业微信APP课程平台应用首页右上角输入



该邀请码2025年09月05日前有效

计算机2304-2306





第一章 绪论

Principles of Database Systems

第一章 绪论

- 1.1 数据库系统概述
- 1.2 数据模型
- 1.3 数据库系统的三级模式结构
- 1.4 数据库系统的组成
- 1.5 数据库系统的体系结构

本章思考：

- 什么是数据库系统？
- 产生DBMS的动机是什么？
- 如何设计数据库？
- DBMS的结构？
- 数据库系统的组成？

1.1 数据库系统概述

1.1.1 四个基本概念

- 数据(Data)
 - 数据库(Database)
 - 数据库管理系统(DBMS)
 - 数据库系统(DBS)
-
- 为什么要学习四个基本概念?
 - 数据库={数据集, 操作集, 限制集}, 理解其中每个元素。
 - 不同人员, 不同视角。

1. 数据 (Data)

□ **数据**的定义: 描述事物的符号记录。是数据库中存储的基本对象。

□ 数据的种类: 数字、文本、图形、图像、音频、视频等。

例1: “2025年” , “两千零二十五年” 。

□ 数据的特点: **数据与其语义不可分**。

例2: (0005794, 601, 周济, 1, 19460826, 01)

■ 语义1: (工号, 部门编号, 姓名, 性别, 生日, 民族)

■ 语义2: (... , 办公室门牌 / 内部电话号,)

□ **数据是信息的符号表示或载体, 信息则是数据的内涵, 是对数据的语义解释。**

数据与信息的区别:

- 数据可以表示信息, 同一数据也可以有不同的解释;
- 信息是抽象的, 不随数据形式而改变; 而同一信息可以有不同的数据表示方式。

2. 数据库 (DB)

□ 数据库 (DB, DataBase) 的定义:

- DB是**长期存储**在计算机内、**有组织的**、**可共享**的**大量数据**的集合。
- 数据是数据库中存储的基本对象。

□ 数据库的基本特征:

- 数据按一定的**数据模型**组织、描述和存储;
- 可为各种用户**共享**;
- **冗余度**较小 (redundancy) ;
- **数据独立性**较高 (data independency) ;
- 可扩展性 (Scalability) 。



3. 数据库管理系统 (DBMS)

- DBMS: DataBase Management System
- DBMS是**对数据库进行管理的****大型系统软件**，它是数据库系统的**核心**组成部分。
 - 位于**用户与操作系统之间**的一层数据管理软件；
 - 是**基础软件**，是一个大型复杂的软件系统。
- DBMS的用途：
 - 科学地组织和存储数据、高效地获取和维护数据。
- 用户在数据库系统中的一切操作，包括数据定义、查询、更新及各种控制，都是通过DBMS进行的。

DBMS的主要功能

□ 数据定义功能

- 提供数据定义语言(DDL)
- 定义数据库中的数据对象的组成和结构 (逻辑结构、存储结构)

□ 数据组织、存储和管理

- 分类组织、存储和管理各种数据 (数据字典、用户数据、数据存取路径等)
- 确定组织数据的文件结构和存取方式
- 实现数据之间的联系
- 提供多种存取方法提高存取效率

□ 数据操纵功能

- 提供数据操纵语言(DML)
- 实现对数据库的基本操作(查询、插入、删除和修改)

□ 数据库的事务管理和运行管理

- 数据库的建立、运行和维护
- 数据的安全性、完整性
- 多用户对数据的并发使用
- 发生故障后的系统恢复

□ 数据库的建立和维护功能(实用程序)

- 数据库初始数据装载
- 数据库转储
- 介质故障恢复
- 数据库的重组织
- 性能监视分析等

□ 其它功能

- DBMS与网络中其它软件系统的通信
- 两个DBMS系统的数据转换
- 异构数据库之间的互访和互操作

什么是数据库管理系统?

□ 从OS角度看, DBMS是:

一个相互关联的数据集合, 一组访问这些数据的程序。

□ DBMS的目标是什么?

- 为应用提供一个方便、有效的数据支撑环境;
- 数据库技术是一种资源抽象技术, 将文件资源抽象成一种能存放结构化、关联性信息的虚拟数据库资源;
- 从资源共享角度考虑, DBMS应考虑数据一致性、操作原子性及异常处理等性能问题。

4.数据库系统 (DBS)

□ DBS, DataBase System

□ 数据库系统是由:

- 数据库
- 数据库管理系统 (及其开发工具)
- 应用系统
- 数据库管理员 (DBA, DataBase Administrator)

组成的

存储、管理、处理和维护数据的系统。

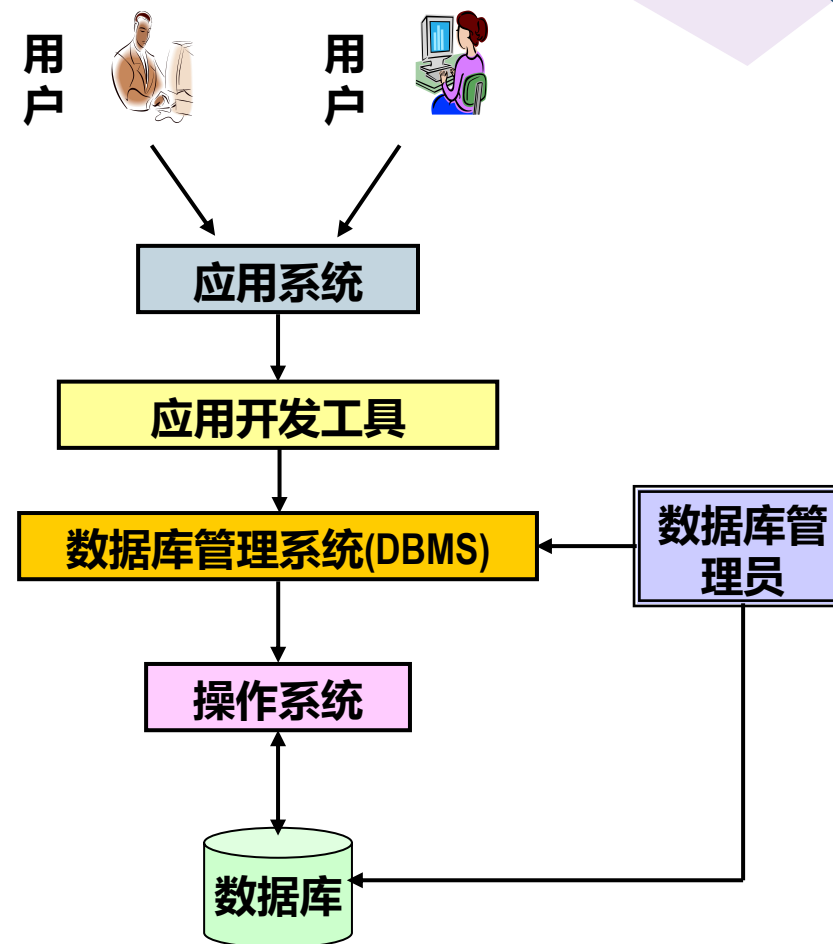


图 数据库系统构成

1.1.2 数据管理技术的产生和发展

□ 数据管理 ≠ 数据处理:

- 数据管理是对数据进行分类、组织、编码、存储、检索和维护
- 数据处理是对各种数据进行收集、加工、储存、传播等一些列活动的总称
- 数据管理是数据处理的中心问题

□ DBMS的产生动机:

- 应用需求的推动
- 计算机硬件的发展
- 计算机软件的发展

推动计算机发展, DBMS出现

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:

手工管理

手工管理 (直接书写机器语言)

```
00101011 11100101 11001001
10101100 11010100 11110000
01010010 10010000 10000000
```

文件管理 VS. 数据库管理

文件管理 (程序管理数据)

```
main () {
    int a, b, c;
    fopen(... ..);
}
```

数据库管理 (自主管理信息)

```
select * from S
insert
delete
```



OS




DBMS




- 各阶段的技术及应用背景
 - 计算机应用范围
 - 外存储设备
 - 数据管理软件
- 各个阶段的差别体现
 - 谁管理数据
 - 数据面向谁
 - 数据与应用的独立性

1.人工管理阶段

- 时期：20世纪40年代中--50年代中
- 产生的背景
 - 应用需求： 科学计算
数据量小、结构简单，如高阶方程、曲线拟和等
 - 硬件水平： 无直接存取存储设备
外存为顺序存储设备，如：磁带、卡片、纸带，没有磁盘等
 - 软件水平： 没有OS，只有汇编语言，没有专门管理数据的软件
 - 处理方式： 数据批处理

1.人工管理阶段

- 特点：程序与数据是一个整体，一个程序中的数据无法被其他程序使用，因此程序与程序之间存在大量的重复数据。
- 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
- 数据的独立性：不独立，完全依赖于程序
- 数据的结构化：无结构
- 数据控制能力：应用程序自己控制

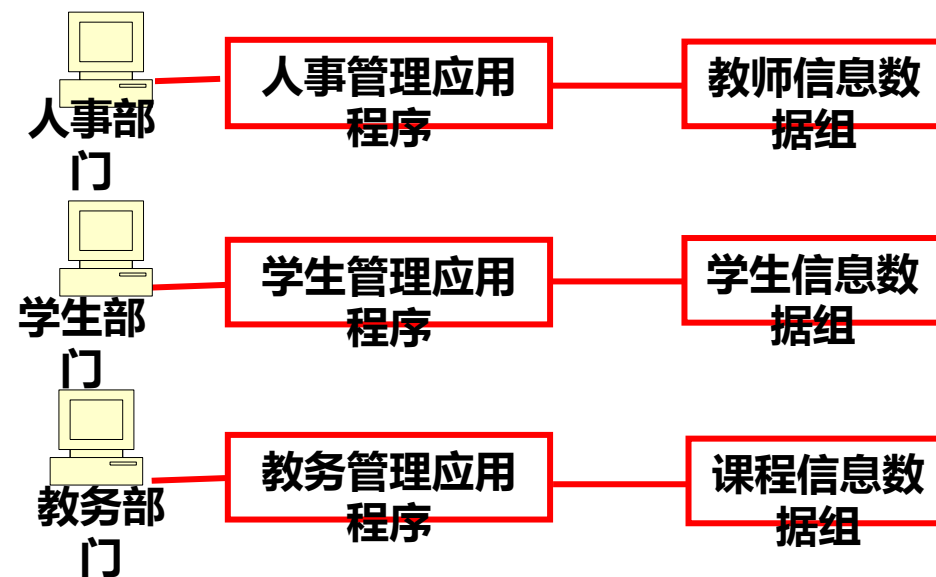


图 应用程序和数据的依赖关系

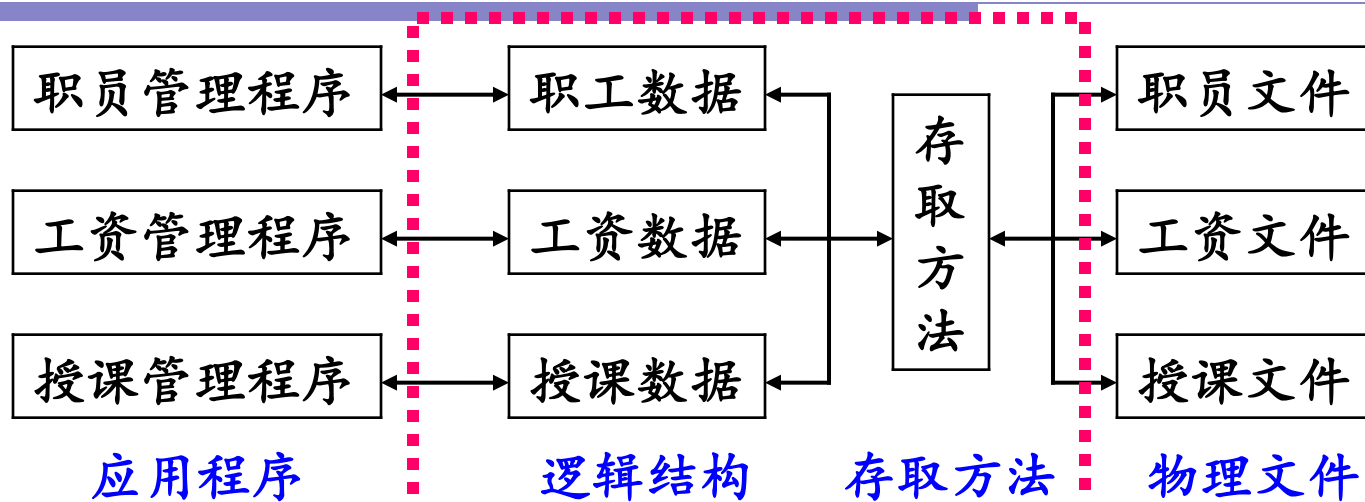
2. 文件系统阶段

- 时期: 20世纪50年代末--60年代中
- 产生的背景:
 - 应用需求: 科学计算、管理
 - 硬件水平: 外存有了磁盘、磁鼓等直接存储设备
 - 软件水平: 有OS和文件系统, 数据保存在文件中
 - 处理方式: 联机实时处理、文件批处理

2. 文件系统阶段

- 程序通过**数据文件**访问数据。文件系统阶段由操作系统的**文件系统**对数据进行管理，用户无需直接涉及物理设备及数据的物理组织细节。
- **文件系统提供一定的数据管理功能：**
 - 存取方法（索引、链接、直接存取、倒排文件等）；
 - 支持对文件的基本操作（增、删、改、查等），用户程序不必考虑物理细节；
 - 数据的存取基本上以**记录**为单位。
- 数据仍是**面向应用的**，一个数据文件对应一个用户程序。
- **数据与程序有一定的独立性**：实现了用户程序与数据的**物理存储结构**的分离。
- **数据共享性弱、冗余度高**：数据可在文件级为**多用户共享**。

开发一个教职员管理系统，采用文件系统如何做？



数据独立性问题

文件系统fopen

职工数据

职工号	姓名	单位	性别	年龄	工龄	职称	工资
-----	----	----	----	----	----	----	----

工资数据

数据共享问题

数据一致性问题

职工号	单位	姓名	职称	工龄	工资	房租	水电
-----	----	----	----	----	----	----	----

数据结构化问题

教课数据

职工号	姓名	单位	职称	课程名	学时
-----	----	----	----	-----	----

文件系统管理数据的缺陷

□ 存在问题：

- **数据完整性问题**：数据库中数据必须满足一致性约束，如学号唯一，只能通过应用程序来保证。
- **故障恢复**：计算机系统出现异常，数据库中数据应恢复到故障前状态；文件系统不保证这种原子性问题。例如，转账问题。
- **共享数据并发访问异常**：文件系统只提供互斥访问共享数据，不考虑并发中的一致性保证。如，对同一账号的两个并发取款操作。
- **安全性问题**：数据库中共享数据要求更高的共享保护，文件系统中由于应用程序与数据密切结合，难于保证。

如何解决文件系统缺陷？

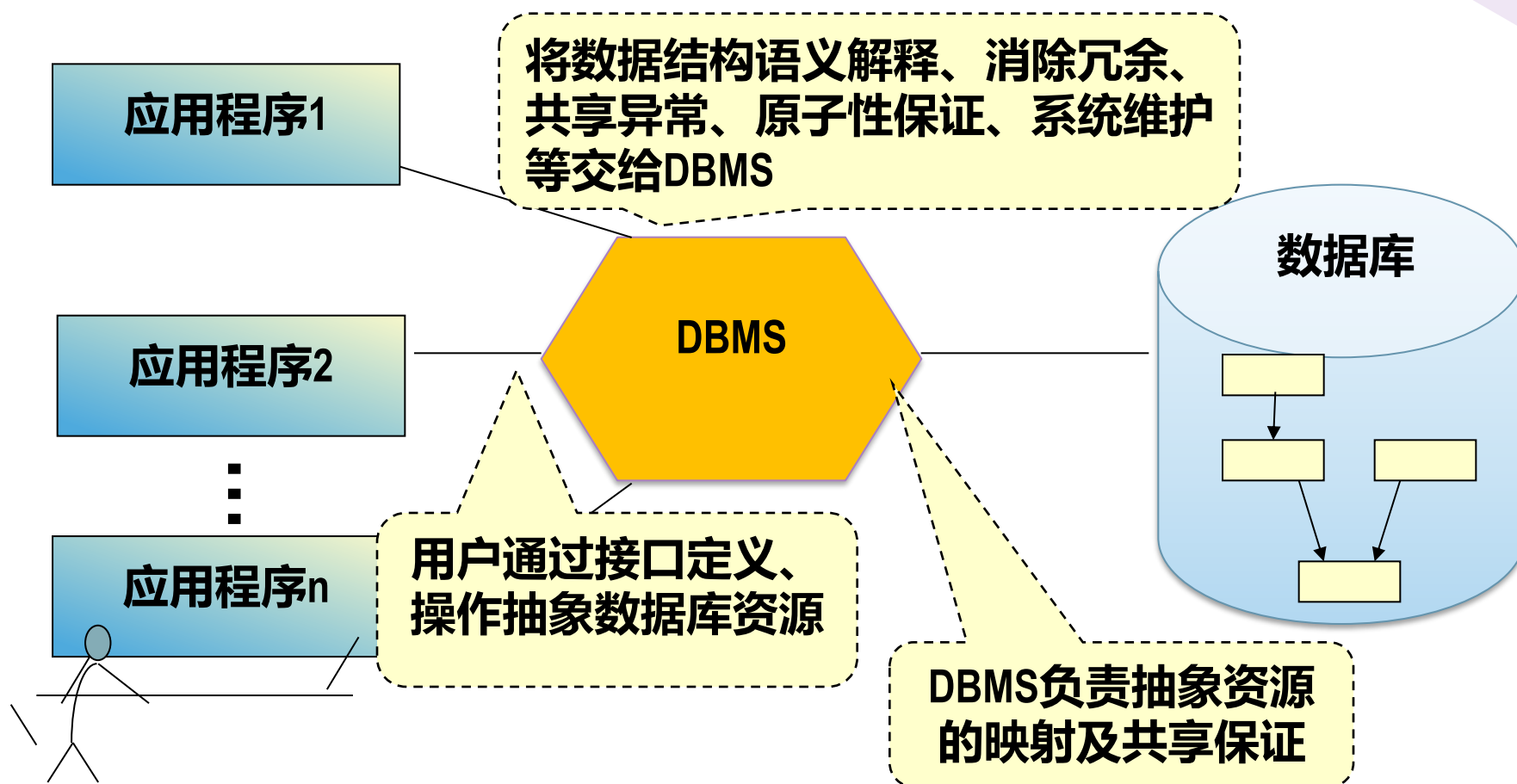
□ 解决方案：

- 开发一种比文件系统更加强大的数据管理系统，将数据的结构语义、关联关系、冗余消除等交给该系统管理，并提供一次一集合的数据访问方式。
- 使用资源抽象、资源共享等虚拟技术，自动保证数据并发共享访问的安全性、隔离性。每个应用对各自的虚拟数据库操作，系统保证虚拟资源到物理资源的映射，保证物理数据库的一致性、完整性。
- 能自动解决共享异常、原子性、完整性问题。

3.数据库系统阶段

- 时期：20世纪60年代末以来
- 产生背景：
 - **应用背景**：大规模管理：计算机管理的数据量大，关系复杂，共享性要求强（多种应用、不同语言共享数据）
 - **硬件背景**：大容量磁盘、磁盘阵列、光盘，硬件价格下降
 - **软件背景**：有**数据库管理系统**；软件价格上升，编制和维护软件及应用程序成本相对增加，求降低
 - **处理方式**：联机实时处理，分布处理，批处理

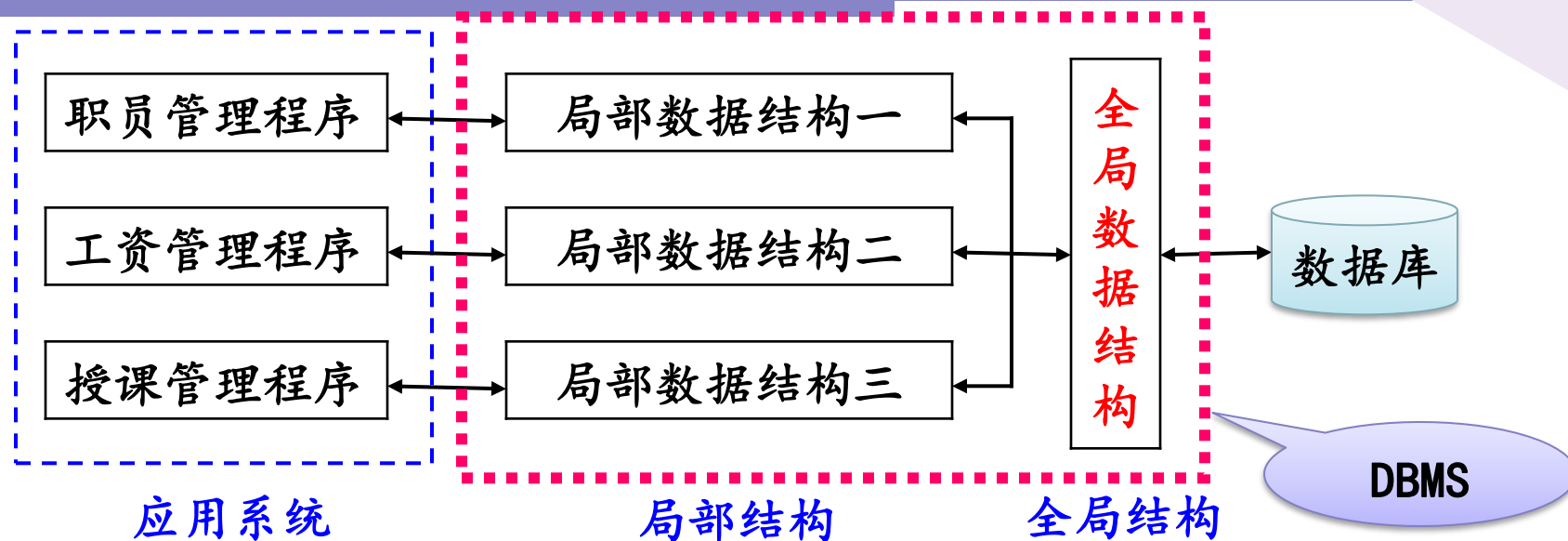
应用程序与数据的对应关系(数据库系统)



第四代编程 1、设计数据库、建立数据库 2、使用数据库

图 数据库系统阶段应用程序与数据之间的对应关系

开发一个教职员管理系统，采用数据库系统如何做？



全局数据结构范例一

工号	姓名	单位	性别	年龄	工龄	职称	工资	房租	水电	课号	学时
----	----	----	----	----	----	----	----	----	----	----	----

全局数据结构范例二

职工号	单位	姓名	性别	年龄	职称
-----	----	----	----	----	----

职工号	工龄	工资	房租	水电
-----	----	----	----	----

职工号	课程名	学时
-----	-----	----

数据库系统的目的

□ 数据库系统的主要目的之一：

给用户提**整体数据的抽象视图**，将磁盘上的所有物理数据集抽象成 **整体结构化的虚拟数据**，隐藏了细节。

□ 例如：类似编译器，用户可以用类型定义语句定义一个存放数据空间，

- `int a;`

- 类似文件系统，用户可以用`fopen()`来创建一个文件；

- DBMS可以用CREATE语句创建数据库；

□ 系统通常采用三层抽象来完成：

视图层、逻辑层、物理层

数据库系统的三层数据抽象

视图层

视图1

视图n

逻辑层 维护定义数据结构语义的数据字典，用户可以定义数据的结构及关系

物理层 维护定义数据物理存储结构逻辑的数据字典，用户可以定义数据的存储方式

- CREATE VIEW AS 可以定义一个视图类型的数据
- CREATE TABLE 可以定义一个表类型的数据
- 模式和实例 定义的数据可类比变量定义，称为**模式**。某时刻的数据称为**实例**，类似值

3.数据库系统阶段

□ 4大特点:

数据库观点: 数据不是依赖于处理过程的附属品, 而是现实世界中独立存在的对象。

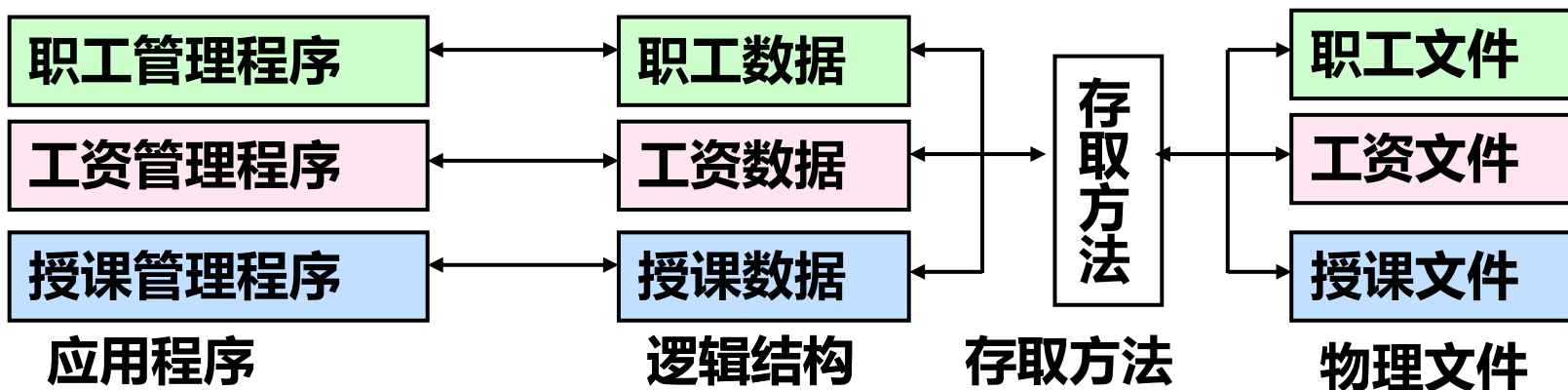
- ① 数据结构化
- ② 数据的共享性高, 冗余度低, 易扩充
- ③ 数据独立性高
- ④ 产生了数据库管理系统 (DBMS) ,数据由DBMS统一管理

1) 数据的结构化

例：设有如下数据：

职工基本记录：	(职工号, 姓名, 单位, 性别, 年龄, 工龄, 职称, 工资)
工资记录：	(职工号, 姓名, 职称, 工龄, 工资, 房租, 水电)
授课记录：	(职工号, 单位, 姓名, 职称, 课程名, 学时)

文件系统中应用程序与数据的使用对应方式：



1) 数据的结构化

例：设有如下数据：(职工号，姓名，单位，性别，年龄，工龄，职称，工资，房租，水电，课程名，学时)

职工基本记录：

(职工号，姓名，单位，性别，年龄，工龄，职称，~~工资~~)

工资记录：

(职工号，~~姓名~~，~~职称~~，工龄，工资，房租，水电)

授课记录：

(职工号，~~单位~~，~~姓名~~，~~职称~~，课程名，学时)

数据库系统中应用程序与数据的使用对应方式：



数据结构化：

按照某种数据模型，将全系统的各种数据组织到一个结构化的数据库中，整个组织的数据不是一盘散沙，可表示出数据之间的有机关联。

1) 数据结构化

- 整体数据的结构化是数据库的主要特征之一
- “整体” 结构化
 - 不再仅仅针对某一个应用，而是面向全组织
 - 不仅数据内部结构化，整体是结构化的，数据之间具有联系
- 数据库中实现的是数据的真正结构化
 - 数据的结构用数据模型描述，无需程序定义和解释
 - 数据可以变长
 - 数据的最小存取单位是数据项

2) 数据的共享性高、冗余度低，易扩充

- **数据面向整个系统**，而不是面向某一应用，数据集中管理，数据共享，因此冗余度低；
- 节省存储空间，减少存取时间，且可避免数据之间的不一致性；
- 每个应用选用数据库的一个子集，只要重新选取不同子集或者加上一小部分数据，就可以满足新的应用要求，这就是易扩充性。

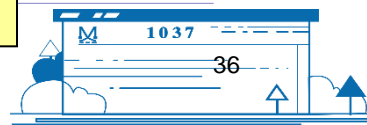
注意：

数据库中并非完全消除冗余！ 有时为了提高数据的存取效率，同一数据可保留多个副本，但是这种冗余是在数据库系统的控制之下的。

3) 数据独立性

- 数据独立性指数据与使用数据的程序分离的特性。分为**逻辑独立性**和**物理独立性**。
- **逻辑独立性**是指当数据的总体逻辑结构改变时，数据的局部逻辑结构不变，由于应用程序是依据数据的局部逻辑结构编写的，所以应用程序不必须修改，从而保证了数据与程序间的逻辑独立性。
 - 例如，在原有的记录类型之间增加新的联系，或在某些记录类型中增加新的数据项，而不需要修改应用程序。
- **数据的物理独立性**是指当数据的存储结构改变时，数据的逻辑结构不变，从而应用程序也不必改变。
 - 例如，改变存储设备和增加新的存储设备，或改变数据的存储组织方式，而不需要修改应用程序。

良好的数据独立性可降低程序开发和维护的代价。



4) 统一的数据管理和控制功能

□ 数据的**安全性控制** (Security)

- 保护数据以防止不合法的使用所造成的数据泄露和破坏
- 措施：用户标识与鉴定，存取控制、数据加密、审计等

□ 数据的**完整性控制** (Integrity)

- 数据的正确性、有效性、相容性
例如：身高 = 17.5米
- 措施：完整性约束条件定义和检查
例如：身高约束 (身高 < 3米)

□ **并发控制** (Concurrency)

- 对多用户的并发操作加以控制、协调，防止其互相干扰而得到错误的结果并使数据库完整性遭到破坏
- 措施：封锁

□ **数据库恢复** (Recovery)

- 把数据库从错误状态恢复到某一已知的正确状态(亦称一致性状态或完整性状态)
- 措施：转储，镜像，日志

文件系统与数据库系统的区别

- 从文件系统管理发展到数据库系统管理是信息处理领域的一个重大变化，表现在：
 - 在文件系统阶段，【以软件为中心】：人们关注的是系统功能的设计，因此程序设计处于主导地位，数据服从于程序设计；
 - 在数据库系统阶段，【以数据为中心】：数据的结构设计成为信息系统首先关心的问题。可以供各种用户共享，具有较小冗余度和较强的数据独立性。

数据管理技术经历了以上三个阶段的发展，已有了比较成熟的数据库技术，但随着计算机软硬件的发展，数据库技术仍不断向前发展。

1.1 回顾

1. 认识了一种新的计算机系统——**数据库系统**。
2. **数据库系统 (DBS)** 与其它系统的主要区别在于其中包含了**数据库 (DB)** 和 **数据库管理系统 (DBMS)** 。
3. **数据库**是特定应用环境中的相关数据按照一定的方式**组织**形成的数据整体，具有**冗余低、数据独立性高、易扩展、可共享**等传统数据文件所不具备的优点。
4. **数据库管理系统**是负责管理数据库的系统软件。它使得应用程序能简单、快速、共享地使用数据库中的数据，并能在此过程中保证数据的**安全性、完整性和一致性**。
5. 数据管理技术发展的3个阶段：**人工、文件系统、数据库系统阶段**。
6. 数据库系统的4大特点。

课堂练习

1. 长期存储在计算机内、有组织、可共享的大量数据的集合是（ ）。
A. 数据 B. DB C. DBMS D. DBS
2. 数据库管理系统是一种系统软件，它建立在()系统之上。
A. 应用系统 B. 编译系统 C. 操作系统 D. 硬件系统
3. 判断题：在文件系统管理阶段，由文件系统提供数据存取方式，所以数据已经达到了很强的独立性，不需要应用程序来管理数据。 （ ）
4. 判断题：引入数据库的计算机系统称为数据库系统。 （ ）

数据库系统原理

教程：数据库系统概论（第5版）

结合：CMU 15-445/645 INTRO TO DATABASE SYSTEMS

华中科技大学 计算机学院

左琼

1.2 数据模型

1.2.1 数据建模

1.2.2 概念模型

1.2.3 数据模型三要素

1.2.4 层次模型

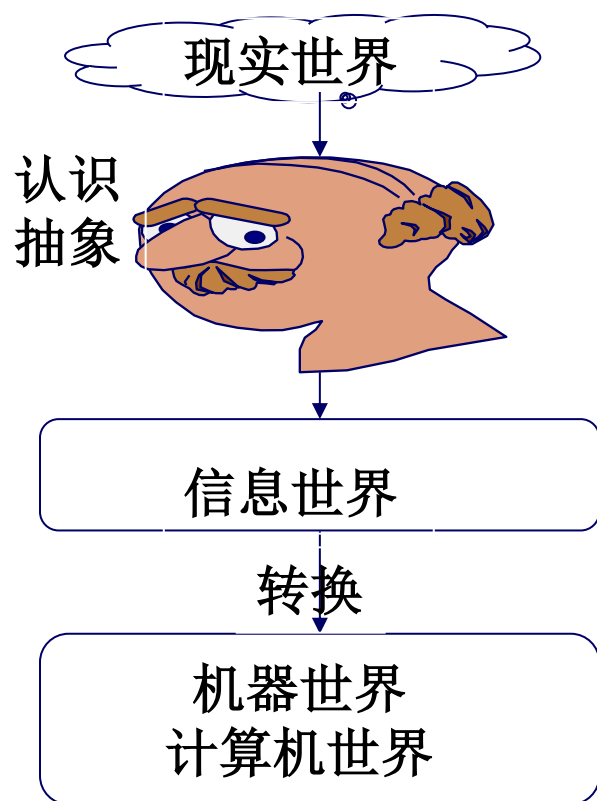
1.2.5 网状模型

1.2.6 关系模型

1.2.7 数据库领域中不断涌现的数据模型

1.2 数据模型 (Data Model)

□ **数据模型**：是一种描述数据、数据联系、数据语义及一致性约束的抽象工具，对现实世界数据特征的描述。

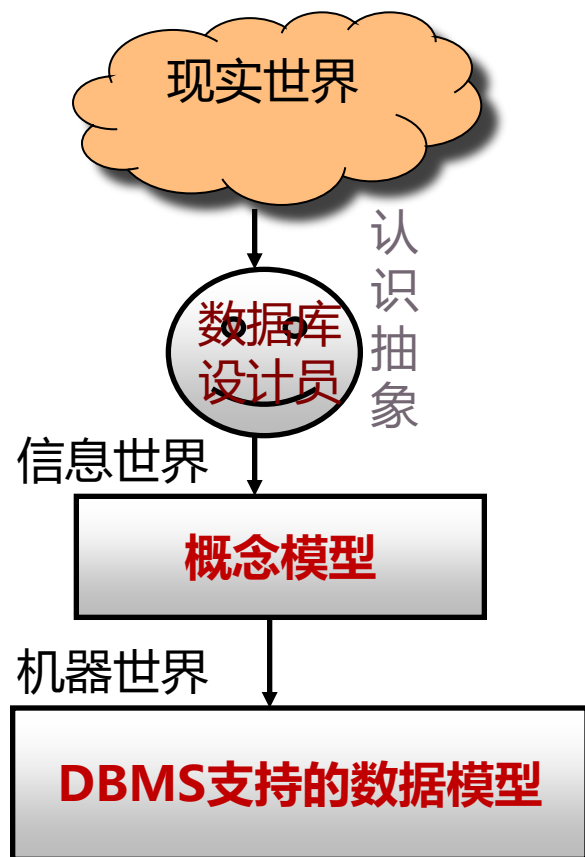


数据模型应满足3方面要求：

- 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中用于提供信息表示和操作手段的形式构架。即：在数据库中用数据模型这个工具来**抽象、表示和处理**现实世界中的数据 and 信息。

1.2.1 数据建模



现实世界 \rightleftharpoons 概念模型
数据库设计人员完成

概念模型 \rightleftharpoons 逻辑模型
数据库设计人员完成

逻辑模型 \rightleftharpoons 物理模型
由DBMS完成

概念模型 也称**信息模型**，它是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 主要包括：网状 / 层次 / 关系 / 面向对象 / 半结构化数据模型等，按计算机系统的观点对数据建模，用于**DBMS实现**。

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，在磁盘或磁带上的存储方式和存取方法。

图 现实世界中客观对象的抽象过程

1.2.2 概念模型

- **概念模型**：按用户的观点对数据建模。
 - **独立于**特定DBMS的现实世界的抽象模型。
 - 数据库设计的有力工具；用户与DB设计人员进行交流的语言。
- **特点**：
 - 1) 较强语义表达能力；
 - 2) 便于直接表示应用语义；
 - 3) 简单、清晰，易于理解。
- **实体-联系(E-R)模型**：
 - P.P.S.Chen于1976年提出：用E-R图来描述现实世界的概念模型：世界由一组称作**实体**的基本对象及这些对象间的**联系**组成。
 - 是一种**语义模型**，语义由**模型力图**来表达数据。

信息世界中的基本概念

1. **实体(entity)**——客观存在可相互区别的事物、事件和概念。(静态、动态、物质、精神、联系等)。注意：不仅可以是具体的人、事、物（如：一个学生 / 书 / 车），或抽象的概念和联系（一堂课、一次比赛）。
2. **属性(Attribute)**——实体具有的特征。
例：Student (学号, 姓名, 性别, 出生日期, ...)
3. **码 / 键(KEY)** ——**唯一标识**一个实体集中任何实体值又**不含多余属性**的属性集。（至少一个属性；至多n个属性）
例：学生 (KEY): 学号
选课 (KEY): (学号, 课程号)
例：学生： (学号, 姓名) **KEY?**

信息世界中的基本概念

4. **实体型 (entity type)** ——具有相同特征和性质的实体与属性命名序列。

实体值 (entity value) ——实体型的具体实例。

例: **Student**

学号	姓名	性别	年龄
----	----	----	----

2001	余得水	男	20
------	-----	---	----

值集/域(domain)——属性的取值范围。

例: 学号的域为6位整数, 姓名的域为字符串集合,

年龄的域为小于40的整数, 性别的域为 (男, 女) 。

简单属性 / 复合属性 / 单值属性 / 多值属性 / NULL属性 / 派生属性

信息世界中的基本概念

5. **实体集 (Entity Set)** —— 具有相同类型及相同性质 (或属性) 的实体集合。
实体集可以相交。



信息世界中的基本概念

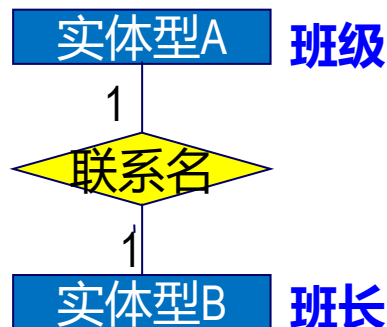
6. **联系 (relationship)** ——现实世界中事物内部以及事物之间的联系在信息世界中被抽象为实体内部的联系和实体之间的联系。实体间的联系可分为三类：

- ① 一对一联系 (1:1)
- ② 一对多联系 (1:m)
- ③ 多对多联系 (m:n)

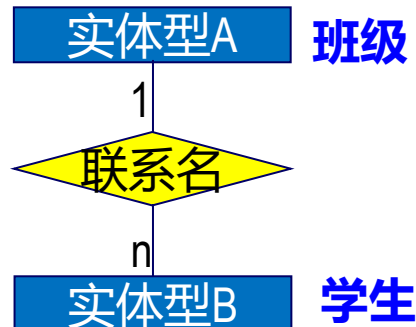
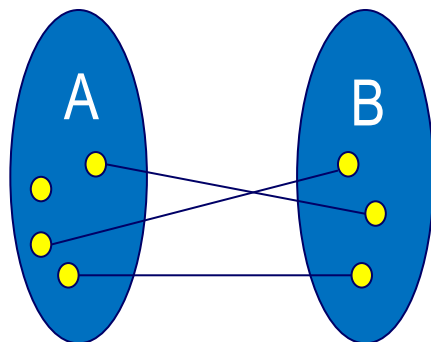
两个实体模型之间的联系——二元联系集的映射基数

1: 1 (一对一联系)

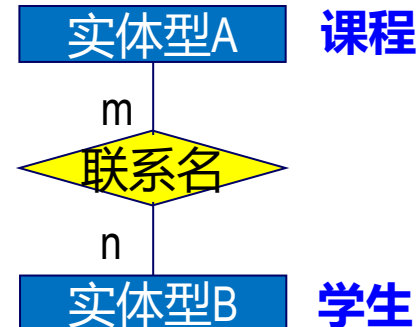
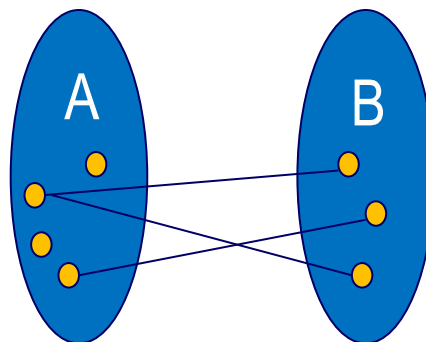
定义：设有实体集A、B，若其中任何一个实体集中每一实体至多与另一实体集中的一个实体有联系，则称A、B间存在一对一联系。



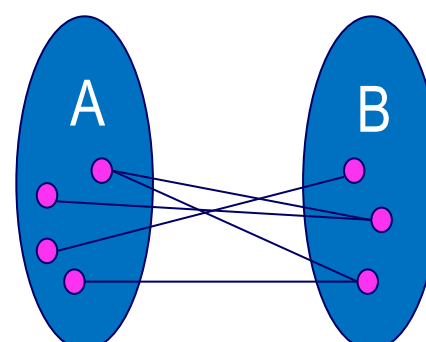
(a) 1: 1的联系



(b) 1: n 的联系



(c) m : n 的联系



m:n联系

定义：设有实体集A、B，若其中任何一个实体集中的每一个实体均与另一个实体集中的n个实体 ($n \geq 0$) 有联系，则称A、B间存在多对多联系。

1: m(一对多联系)

定义：设有实体集A、B，若A中的每一个实体，与B中的n个实体 ($n \geq 0$) 有联系，反之，对于B中的每一个实体，至多与A中的一个实体有联系，则称A、B间存在一对多联系。

信息世界中的基本概念

课堂练习:

顾客可以到多个商场购物，商场可以有很多顾客来购物，则商场与顾客间的联系是_____。

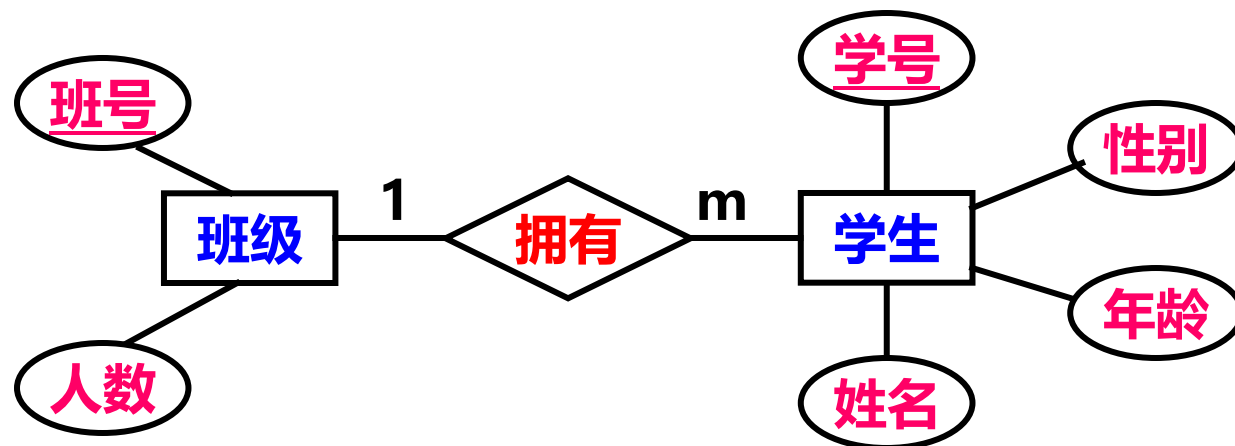
- A. 1: 1
- B. 1: n
- C. ☒ m: n
- D. m: 1

注意：实体间的联系类型反映的是**实体集**间的普遍联系。

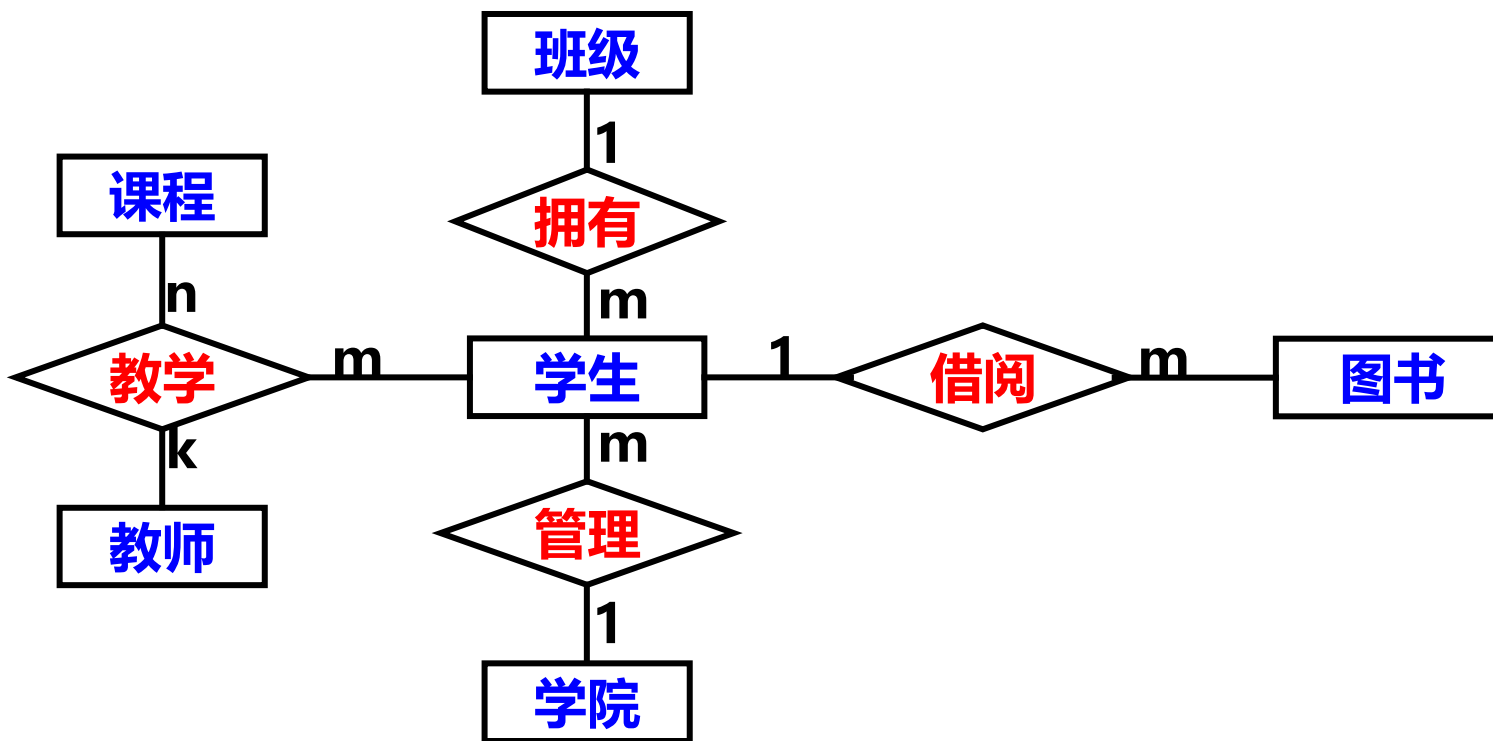
实体-联系 (Entity-Relationship) 模型

采用画E-R图的形式直接表示实体及实体间联系的方法：

- (1) **矩形**表示**实体型**，框内标明实体名；
- (2) **椭圆**表示**属性**，用无向边与其相应实体连接；
- (3) **菱形**表示**联系**，标明联系名，用无向边与相关实体连接；
- (4) **无向边**上标明**联系的类型** ($1:1$, $1:m$, $m:n$)；
- (5) 可据需要任意展开 (E-R图复杂时可略去属性)。



实体-联系 (Entity-Relationship) 模型



E-R图是数据库设计人员与用户进行沟通、交流的工具。
但是，DBMS很难直接支持E-R模型。

1.2.3 数据模型的三要素

- 数据模型是按计算机的观点对现实世界进行抽象、表示事物及事物间联系的模型。
- 数据模型通常由**数据结构**、**数据操作**和**数据的约束条件**三个要素组成。

1. **数据结构**：描述数据库的组成对象及对象之间的关系。

- 数据结构用于描述系统的**静态特性**。
- 数据结构是所研究的对象类型的集合，它是刻画一个数据模型性质最重要的方面。
- 在数据库系统中，人们通常按照其数据结构的类型来命名数据模型。如：**层次模型、网状模型和关系模型的数据结构分别为树型结构、网状结构和关系结构**

1.2.3 数据模型的三要素

2. **数据操作**：是指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。
- 数据操作用于描述系统的**动态特性**。
 - 数据操作是对数据库中各种数据操作的集合，包括操作及相应的操作规则。如数据的检索、插入、删除和修改等。
 - 数据模型必须定义这些操作的确切含义、操作规则以及实现操作的语言。

1.2.3 数据模型的三要素

3. 数据的约束条件

- 数据的约束条件是一组完整性规则的集合。
- 完整性规则是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、相容。
- 数据模型还应该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。

例，在学生数据库中，学生的年龄不得超过40岁。

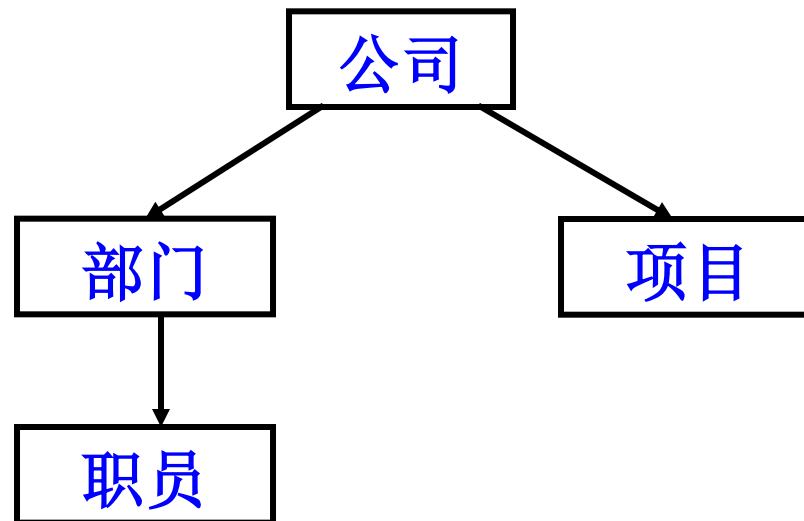
常用的数据模型

- 层次模型(Hierarchical Model)
- 网状模型(Network Model)
- 关系模型(Relational Model)
- 面向对象模型(Object Oriented Model)
- 对象关系模型(Object Relational Model)
- 半结构化模型 (semistructure Model)

- 数据模型主要用于DBMS的实现。
- 一种DBMS通常只支持一种数据模型。数据模型不同，对应的DBMS差别很大，因此，DBMS的类型也通常依据数据模型的不同来划分。

1.2.4 层次模型

- 用**树型结构**表示实体及实体间联系的数据模型。
- 表示方法：
 - ① 一个结点表示一个实体；
 - ② 结点间连线表示实体间联系；
 - ③ 联系中表示1的实体在上层，表示n的结点在下层。



层次数据模型的数据结构(续)

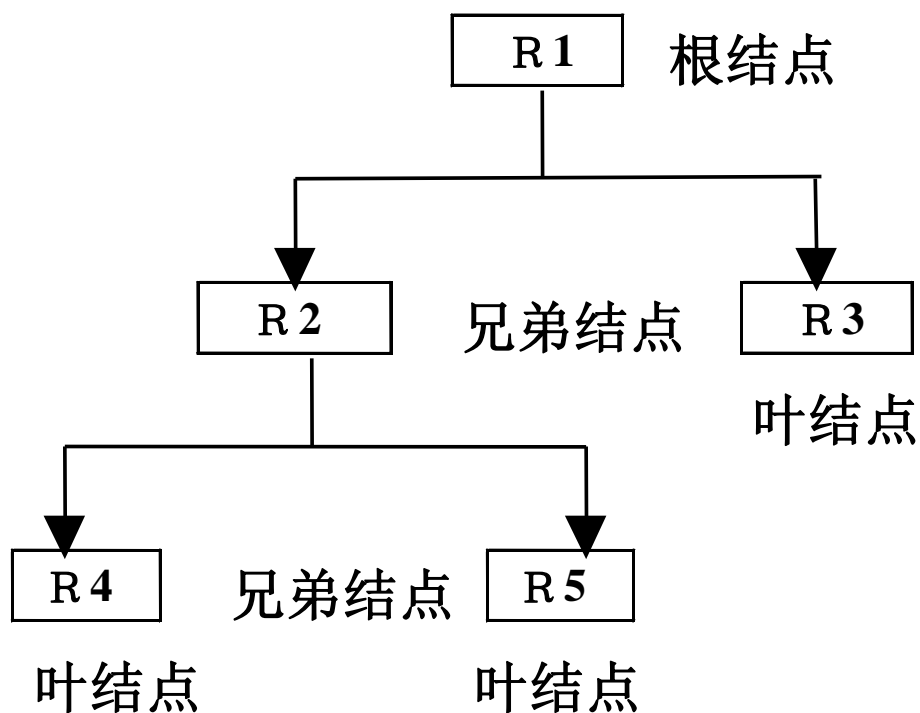


图 一个层次模型的示例

□ 层次模型的特点:

- 结点的双亲是**唯一**的：根节点**没**双亲，其他结点只有一个双亲。
- 只能直接处理一对多的实体联系
- 每个记录类型可以定义一个排序字段，也称为**码**字段
- 任何记录值只有按其**路径**查看时，才能显出它的全部意义
- **没有**一个子女记录值能够脱离双亲记录值而**独立**存在

层次数据模型的数据结构(续)

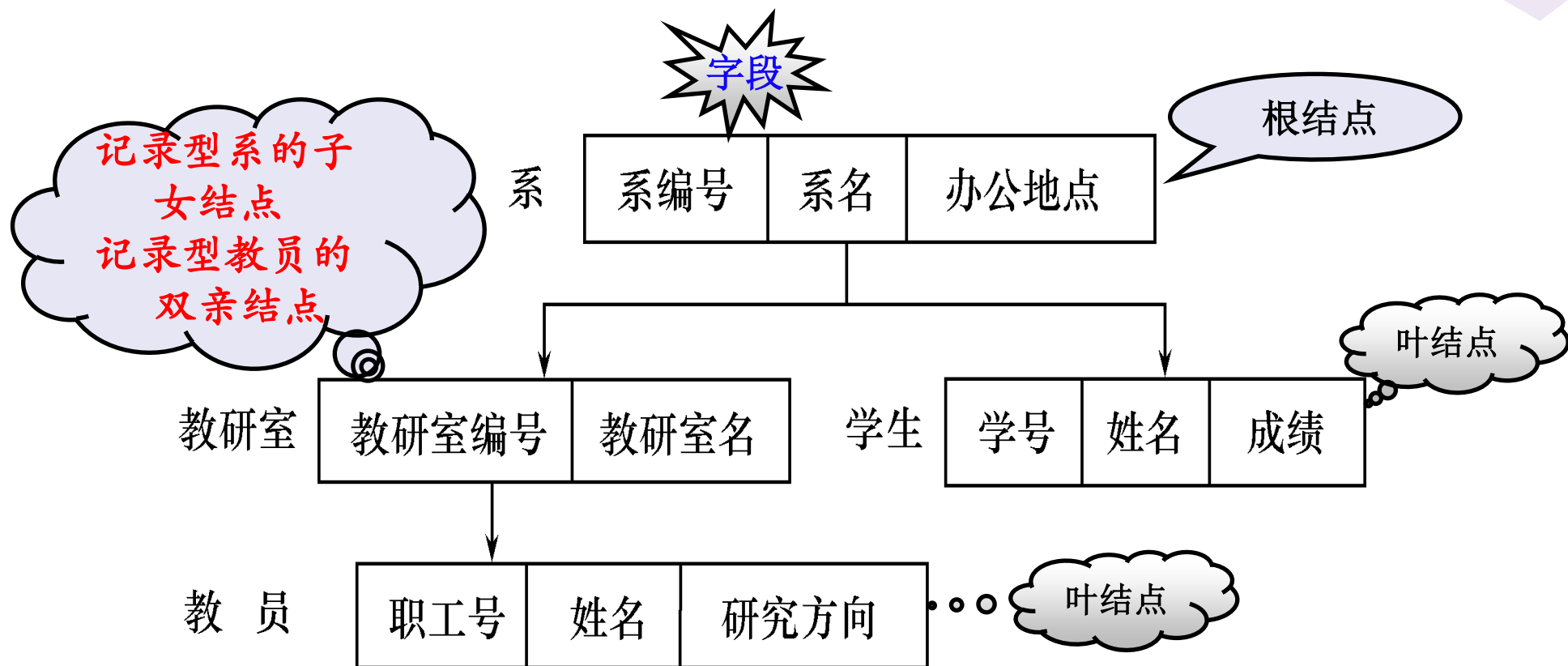


图 教员学生层次数据库模型

层次数据模型的数据结构(续)

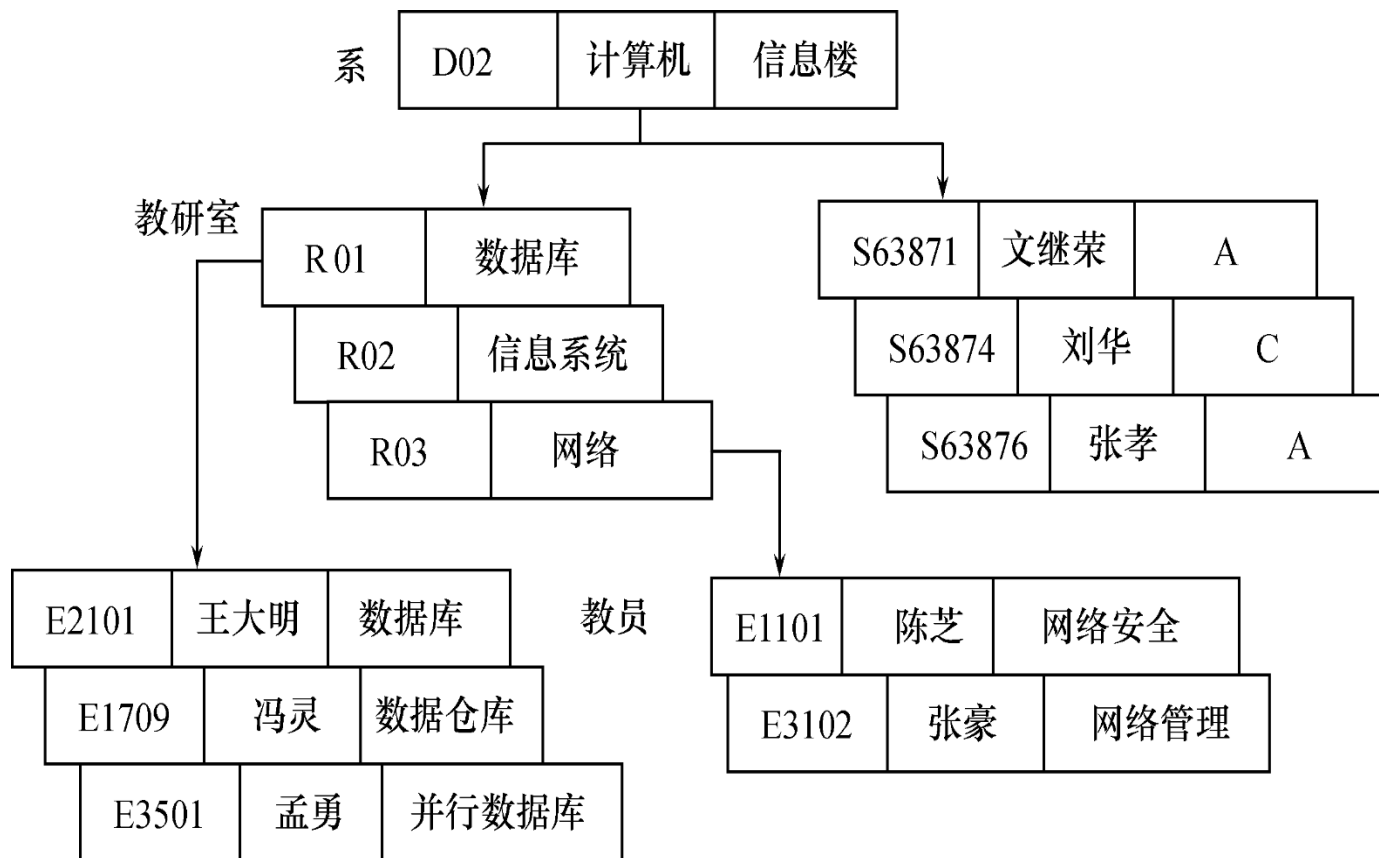


图 教员学生层次数据库的一个值

层次数据模型的存储结构:

1. 邻接法

按照层次树前序遍历的顺序把所有记录值依次邻接存放, 即通过物理空间的位置相邻来实现层次顺序。

2. 子女-兄弟链接法

每个记录设两类指针, 分别指向最左边的子女 (每个记录型对应一个) 和最近的兄弟。

3. 层次序列链接法

按树的前序穿越顺序链接各记录值。

1.2.4 层次模型(续)

2. 操作:增、删、查、改

3. 层次模型的完整性约束条件

- 无相应的双亲结点值就不能插入子女结点值;
- 如果删除双亲结点值, 则相应的子女结点值也被同时删除;
- 更新操作时, 应更新所有相应记录, 以保证数据的一致性。

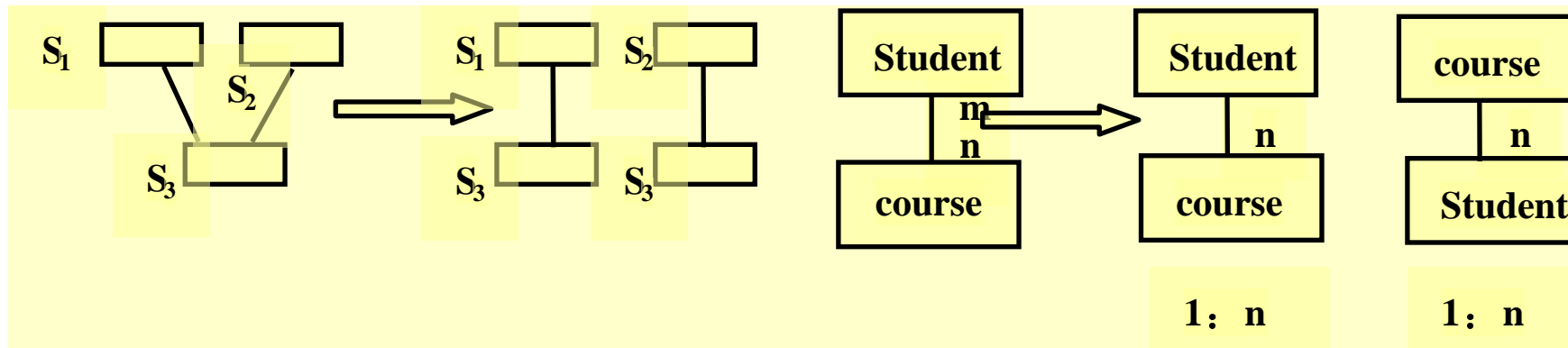
4. 优点

- 简单易用 (几条操作命令) ;
- 自然表示1: M联系;
- 查询效率高。

1.2.4 层次模型(续)

缺点:

1) 不能直接表示 $m:n$ 联系 (须引进冗余结点)



2) 插入, 删除操作限制多

3) 在层次模型中具有一定的存取路径, 查子女须经过双亲 (从上到下, 从左到右: 它仅允许自顶向下的单向查询, 随机存取效率低)

1.2.5 网状模型

- 现实世界中事物之间的联系更多的是非层次关系的，用层次模型表示这种关系很不直观，出现网状模型；
- 20世纪70年代，数据系统语言研究会CODASYL下属的数据库任务组DBTG（Data Base Task Group）提出了一个系统方案，**DBTG系统**，也称CODASYL系统，成为了网状模型的代表。
- 网状模型取消了层次模型的两个限制：两个或两个以上的结点都可以有多个双亲结点，则此时有向树变成了有向图，该有向图描述了网状模型。

1.2.5 网状模型

□ 用网状结构表示实体及实体间联系的数据模型。

□ 表示方法：

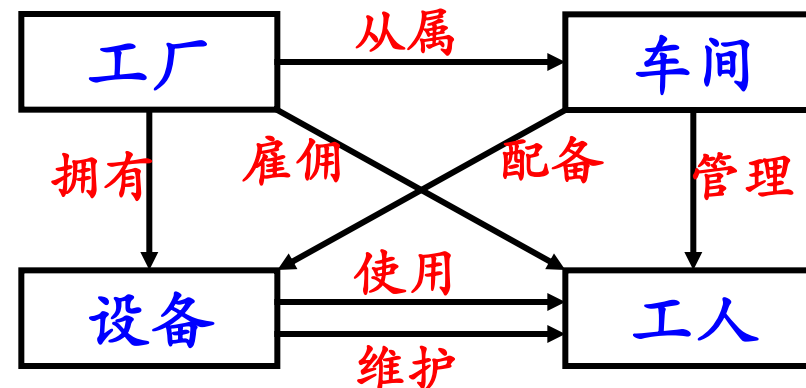
- ① 一个结点表示一个实体；
- ② 有向连线表示实体间联系；
- ③ 两个结点间的联系不唯一，因此联系必须命名。

□ 特征：

- ① 可以有多个结点无父结点；
- ② 至少有一个结点有多个父结点。

□ 缺点：

- ① 模型复杂；
- ② 数据独立性不高。



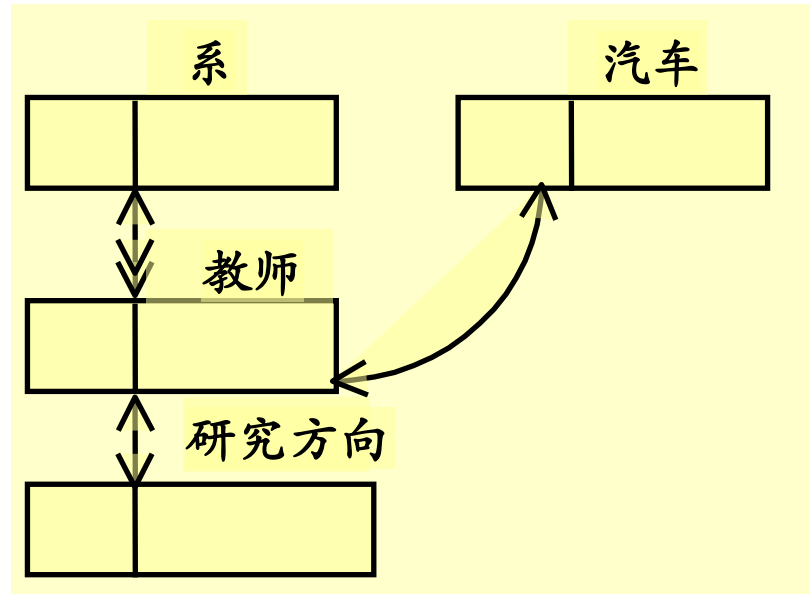
网状数据模型的数据结构

- 网状模型与层次模型的区别：
 - 网状模型允许多个结点**没有双亲**结点
 - 网状模型允许结点有**多个双亲**结点
 - 网状模型允许两个结点之间有**多种联系**（复合联系）
 - 网状模型可以更直接地去描述现实世界
 - 层次模型实际上是网状模型的一个**特例**

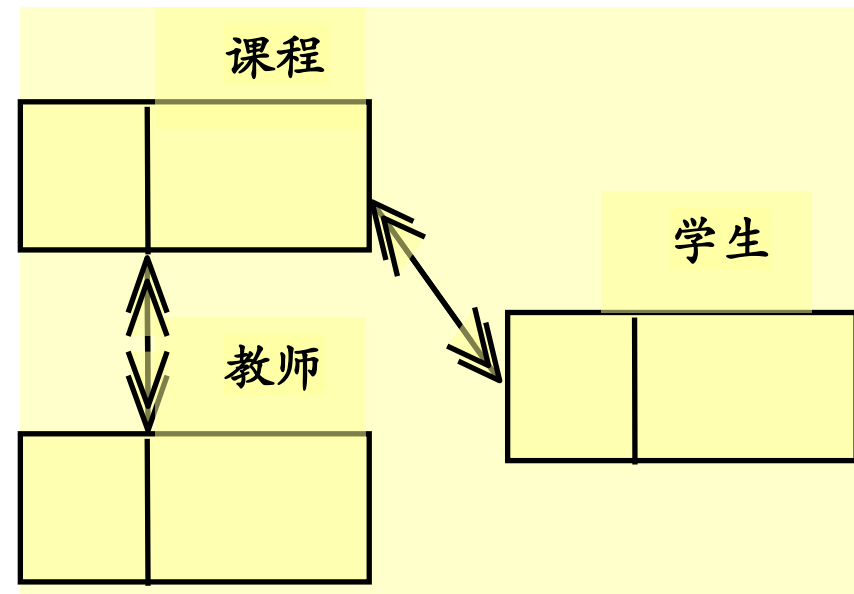
1.2.5 网状模型

1. 数据结构

a: 简单网

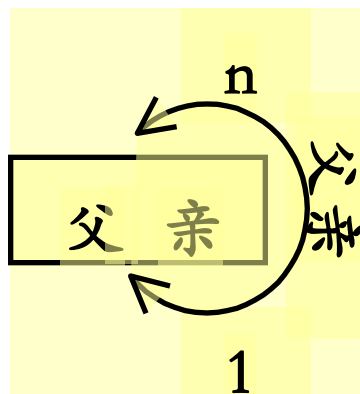


b: 复杂网



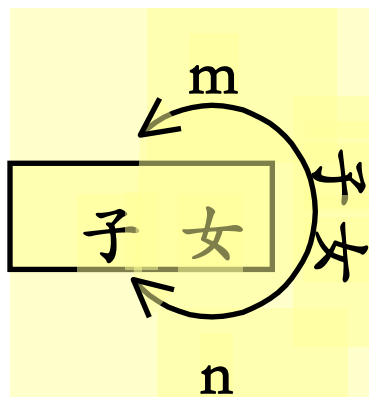
1.2.5 网状模型(续)

c: 简单环网



- 一个父亲有多个已为人父的儿子;
- 已为人父的儿子只有一个父亲。

d: 复杂环网

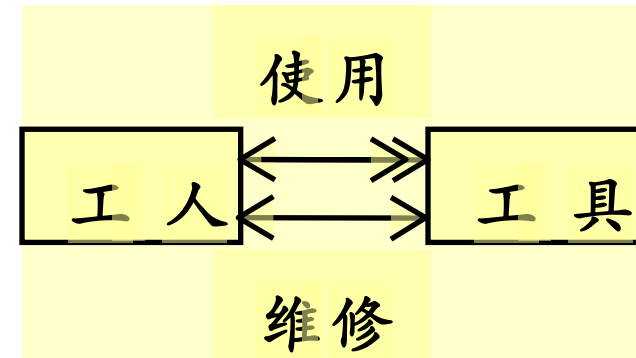


- 每个子女可多个子女;
- 每个为人子女者又可有多多个子女。

12.5 网状模型(续)

□ e: 多种联系

- 1) 结点表示实体, 称为**记录类型**;
- 2) 结点内含数据项, 表示属性;
- 3) **有向连线**表示实体间联系;
- 4) **属性可嵌套**。
- 5) 特征
 - (1) 可多个结点无双亲结点;
 - (2) 子女结点可多个双亲结点;
 - (3) 两记录间可多种联系。



网状数据模型的存储结构

- 关键
 - 实现记录之间的联系
- 常用方法
 - 单向链接
 - 双向链接
 - 环状链接
 - 向首链接

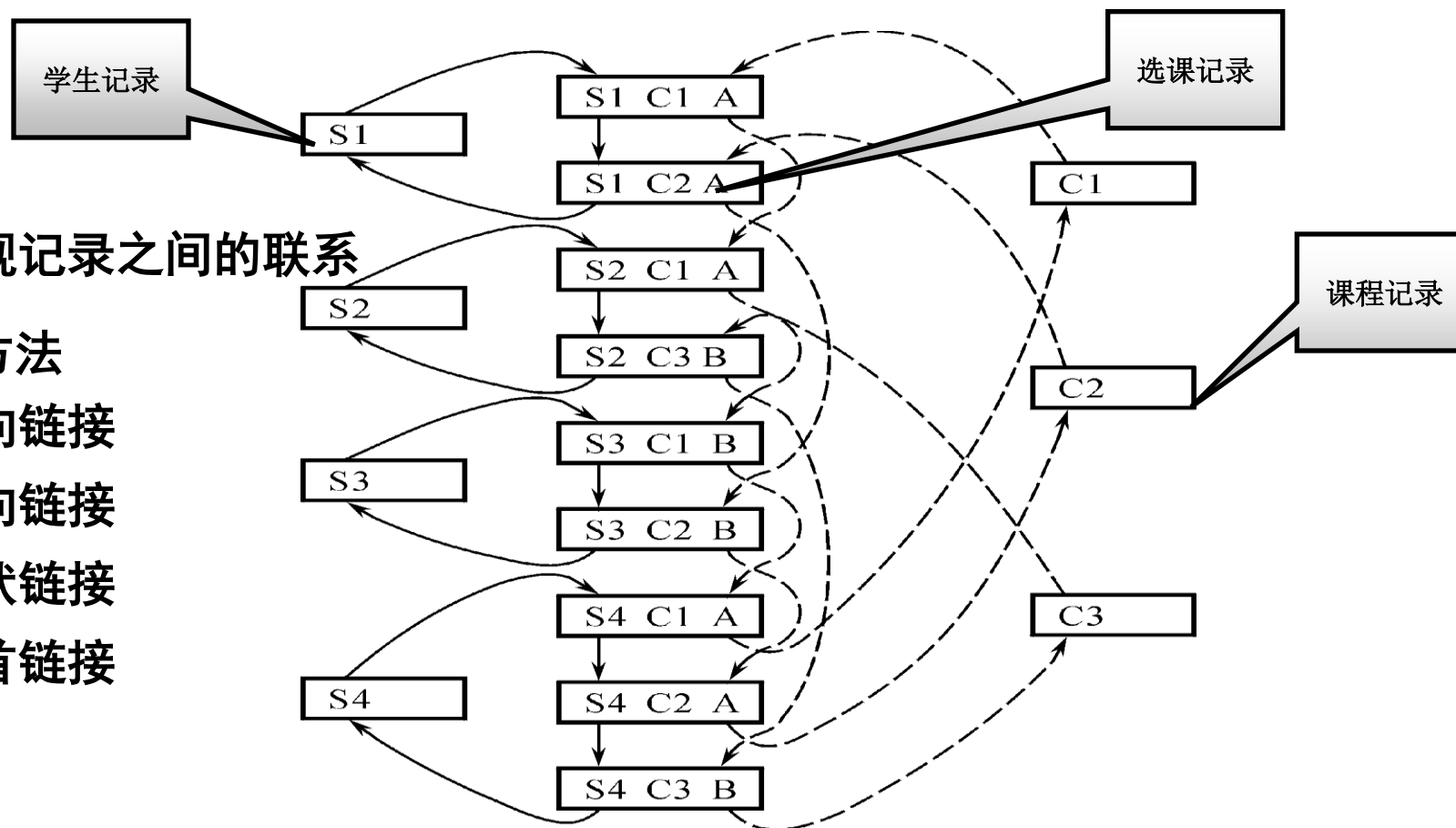


图 学生/选课/课程的网状数据库实例

1.2.5 网状模型(续)

2. 操作：增、删、查、改

3. 约束

1) 插入不受限制； 2) 删去双亲，子女不受影响。

4. 优点

1) 直接表示的m：n联系； 2) 存取效率高。

5. 缺点

1) 结构复杂；

2) DDL、DML复杂；

3) 一次存取一个记录值；

4) 应用程序与数据结构相互依赖，应用程序在访问数据时必须选择适当的存取路径。

1.2.6 关系模型

- 1970年美国IBM公司的研究员E.F.Codd首次提出了数据库系统的关系模型。
 - 他发表论文 “A Relation Model of Data for Large Shared Data Banks”，在文中解释了关系模型，定义了某些关系代数运算，研究了数据的函数相关性，定义了关系的第三范式，从而开创了数据库的关系方法和数据规范化理论的研究。为此他获得了1981年的图灵奖。
 - 此后许多人把研究方向转到关系方法上，陆续出现了关系数据库系统。
 - 1977年IBM公司研制的关系数据库的代表系统System R开始运行，其后又进行了不断的改进和扩充，出现了基于System R的数据库系统SQL/DB。
- 关系数据库已成为目前应用最广泛的数据库系统。

1.2.6 关系模型

- 关系数据模型是以集合论中的关系(relation)概念为基础发展起来的数据模型。
- 在关系模型中，无论是实体还是实体间的联系，均由单一的结构类型即关系(表)来表示。
- 比E-R模型的抽象层次更低。

关系模型的数据结构

- 从用户的角度来看，关系模型是用二维表格表示实体及其间联系的数据模型。
- 一个关系就是没有重复行和重复列的二维表。



关系名：学生登记表

关系模式：对关系的描述

元组中的一个属性值为元组的一个分量

关系数据模型的数据结构（续）

学号	姓名	年龄	性别	所在系	年级
2003010001	张三	19	男	计算机	2
2003010002	李四	20	男	计算机	2
2004050001	王莹	18	女	物理	1
...

表 术语对比

关系术语	一般表格的术语
关系名	表名
关系模式	表头（表格的描述）
关系	（一张）二维表
元组	记录或行
属性	列
属性名	列名
属性值	列值
分量	一条记录中的一个列值
非规范关系	表中有表（大表中嵌有小表）

关系数据模型的数据结构 (续)

- 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**
- 图中工资和扣除是可分的数据项, 不符合关系模型要求

职工号	姓名	职 称	工 资			扣 除		实 发
			基 本	津 贴	职 务	房 租	水 电	
86051	陈 平	讲 师	1305	1200	50	160	112	2283
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

图 一个工资表(表中有表)实例

举例

- 表1、表2、表3 给出三个不同的关系，其中表1 和 表2是基本关系，用于表示现实世界的基本实体。
- 而表3则是“联系”关系，用于表示哪些学生选修哪些课程的联系，这种**联系**是通过在“选课”关系中引入“学生”关系中的“学号”和“课程”关系中的“课程号”属性的数据建立的。

表1：学生

学号	姓名	性别
----	----	----

表2：课程

课程号	课程名	学时
-----	-----	----

表3：选课

学号	课程名	成绩
----	-----	----

关系数据模型结构简单、理论基础严密、数据独立性高、支持非过程化语言、一次操作可存取多个元组，并且可直接表示多对多联系。
主要不足是查询效率低。

关系数据模型的操纵与完整性约束

- 数据操作是**集合**操作，**操作对象和操作结果都是关系**，即若干元组的集合：
 - 查询、插入、删除、更新
- **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**

关系数据模型的优缺点

□ 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导致查询效率往往不如非关系数据模型;
- 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。

1.2.7 数据库领域中不断涌现的数据模型

- 层次模型、网状模型
- 关系模型：严格的理论基础，概念简单、清晰，易于用户理解和使用。
- 面向对象模型、对象关系模型
- 半结构化XML模型

- 键值对数据模型
- 文档数据模型
- 图数据模型
- 时序数据模型
- 时空数据模型
- 流数据模型

1.2 小结

- 使用计算机来管理信息，首要解决如何将现实世界恰当地抽象、表示为计算机中的数据集合的问题。 解决该问题要用到的工具就是——**数据模型**。
- **二级抽象**的过程是先由现实世界到信息世界，再到机器世界。
- 从现实世界映射到信息世界的模型称为**概念模型**，以**实体-联系模型**（E-R模型）为主要代表，它将现实世界抽象为**E-R图**。
- 从信息世界映射到机器世界的模型称为**数据模型**，根据其采用的数据结构的不同，涌现了由层次模型、网状模型到关系模型、面向对象模型、对象关系模型等数据模型。目前主流DBMS支持的数据模型之一为关系模型。

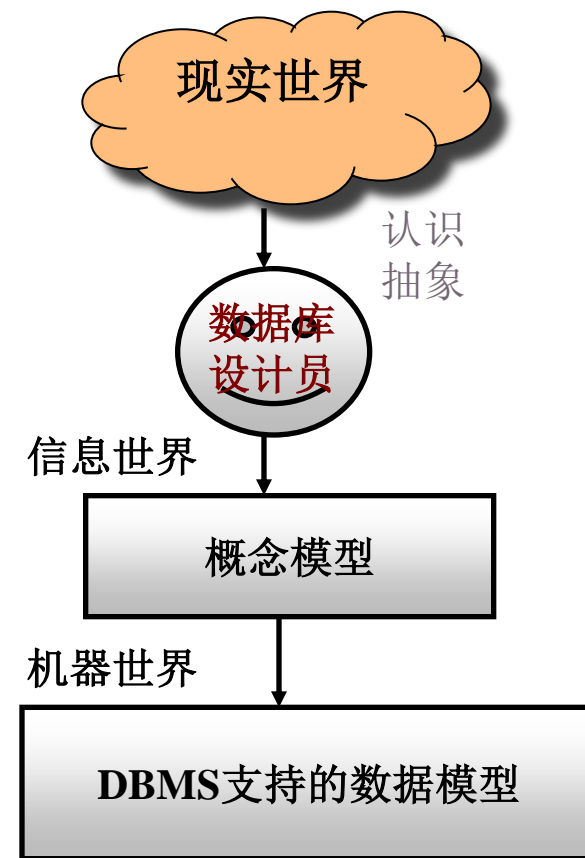
课堂练习

- 层次型、网状型、关系型数据库划分原则是?
 - A. 记录长度
 - B. 文件大小
 - C. 联系的复杂程度
 - D. 数据之间的联系方式

课堂练习

□ 什么模型最适合与用户交流?

- A. 关系模型
- B. 概念模型
- C. 面向对象模型
- D. 逻辑模型



数据库系统原理

教程：数据库系统概论（第5版）

结合：CMU 15-445/645 INTRO TO DATABASE SYSTEMS

华中科技大学 计算机学院

左琼

第一章 绪论

1.1 数据库系统概述

1.2 数据模型

1.3 数据库系统的三级模式结构

1.4 数据库系统的组成

1.5 数据库系统的体系结构

1.3 数据库系统的三级模式结构

□ 数据库中数据如何存储？

- 数据库不仅存放数据的**值**，也需存放数据的语义。
- 数据的**语义**是通过**存储数据的结构**来实现的，库中数据有值与型两部分，分别存放在数据文件和字典中。

□ 数据模型中 “**型**” 和 “**值**” 的区别：

- **型**(Type)：对某一类数据的结构和属性的说明。
- **值**(Value)：是型的一个具体赋值。
- 例：

□ 学生记录型：（学号，姓名，性别，系别，年龄，籍贯）

□ 一个记录值：（900201，李明，男，计算机，22，江苏）

1.3.1 数据库系统模式的概念

- 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。
- 模式 (Schema)，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。
 - 仅仅是型的描述，与具体的值无关。
 - 反映的是数据的结构及其联系。
 - 模式是相对稳定的。
- 实例 (Instance)
 - 模式的一个具体值；反映数据库某一时刻的状态；
 - 同一个模式可以有多个实例；
 - 实例随数据库中数据的更新而变动。

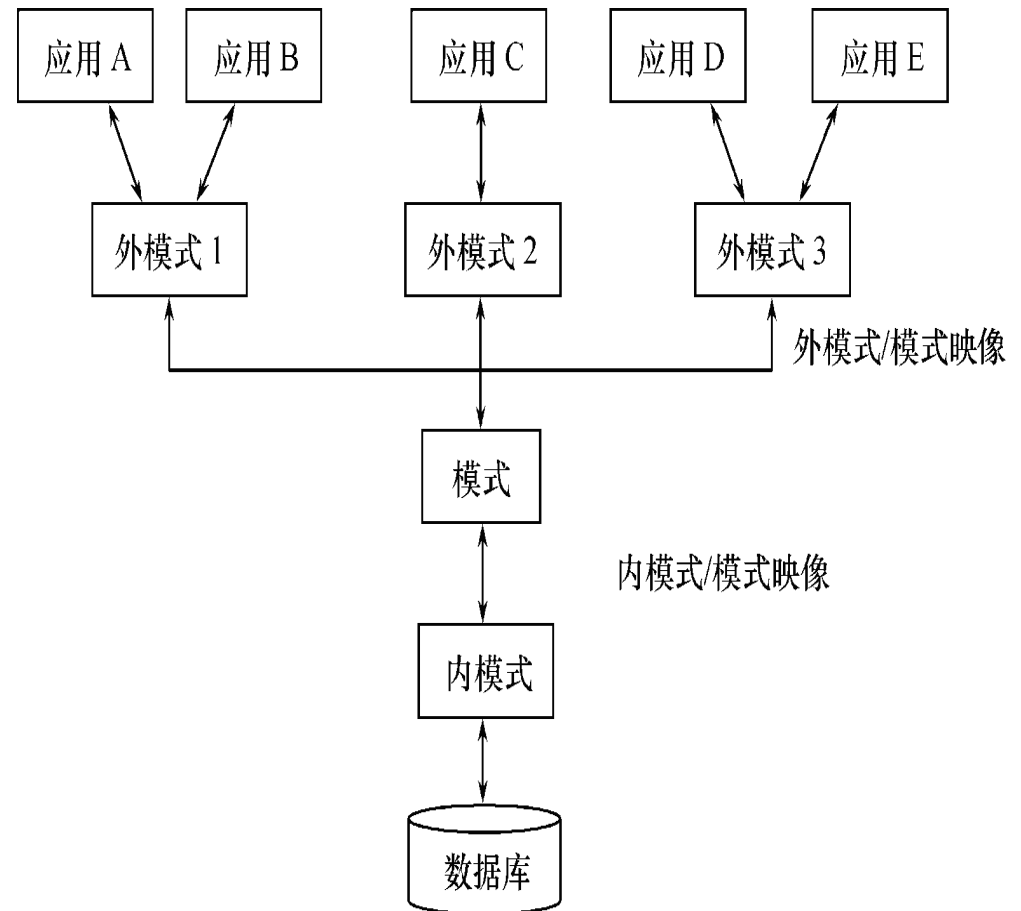
1.3.2数据库系统的三级模式

□ 数据库系统三级模式结构

- CODASYL (Conference On Data System Language) 提出**模式、外模式、内模式三级模式**的概念，三级模式之间有**两级映象**。

□ 模式的分级

- 从数据库用户观点，即用户看到的数据库，与数据库的物理方面，即实际存储的数据库区分开来，**数据库系统的模式是分级的**。
- 提高了**数据的逻辑独立性和物理独立性**。



1. 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;
- 所有用户的公共数据视图——全局逻辑视图, 综合了所有用户的需求。
- 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, 独立于数据库的其他层次。
 - 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。

1. 模式 (Schema)

□ 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**。

□ 模式一般由多个 “**记录**” 组成, 包含数据库的所有信息。

- 如: 某课程选修信息系统的数据库模式包括下列记录:

- 学生 (学号, 姓名, 性别, 系别, 年龄)
- 课程 (课程号, 课程名)
- 选修 (学号, 课程号, 成绩)

□ 设计数据库模式结构时应**首先**确定数据库的逻辑模式。

□ 模式的作用是为了支持数据的**少冗余共享**。

2. 外模式(external schema)

- 外模式（也称子模式subschema或用户模式）：
 - 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；
 - 数据库用户的数据视图，面向具体的应用程序，是与某一应用有关的数据的逻辑表示；
 - 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。

2. 外模式

□ 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。

1.3.2 数据库的三级模式结构

外模式

劳资科

学号	姓名	系别	补贴
----	----	----	----

房产科

学号	姓名	性别	系别	住址
----	----	----	----	----

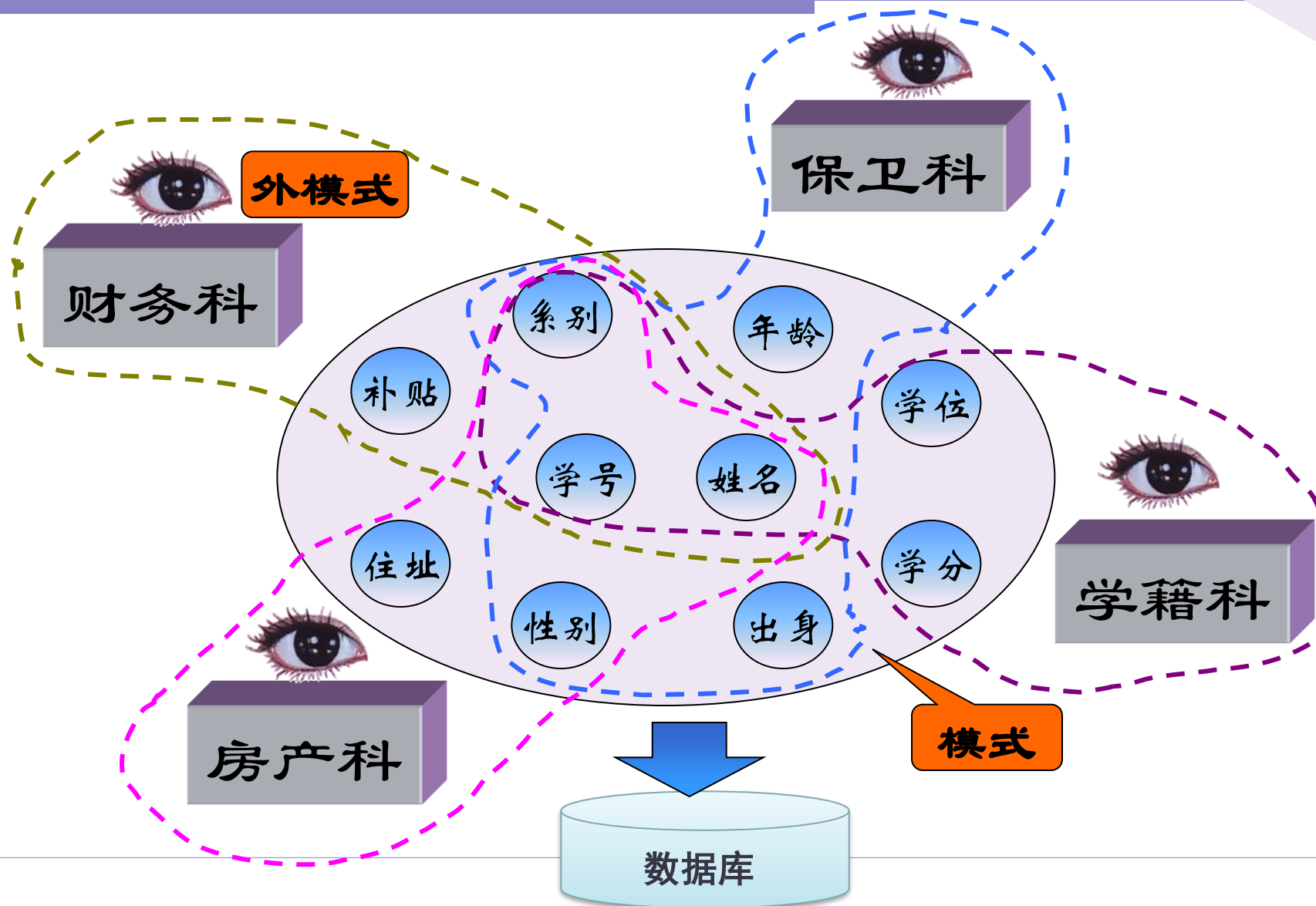
学籍科

学号	姓名	系别	学分	学位
----	----	----	----	----

人事科

学号	姓名	性别	系别	年龄	学位	出身
----	----	----	----	----	----	----

模式 vs. 外模式



模式 vs. 外模式

□ 又如，数据库模式如下：

学生 (学号, 姓名, 性别, 系别, 年龄)

课程 (课程号, 课程名)

选修 (学号, 课程号, 成绩)

■ 可定义子模式如下：

① **单关系子模式**： student_1 (学号, 姓名, 系别)

② **多关系子模式**： SC(学号, 姓名, 课程号, 课程名, 成绩)

□ 外模式的用途：

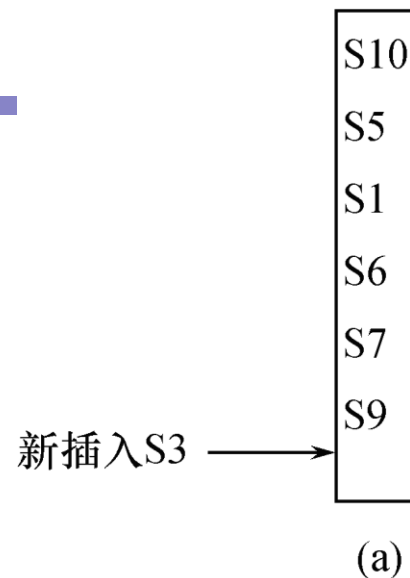
- 1) 支持不同用户建立适应局部应用特征的结构，每个用户只能看见和访问所对应的外模式中的数据；
- 2) 简化应用处理；
- 3) 提高安全性，保证数据库安全性的一个有力措施。

3. 内模式 (Internal schema)

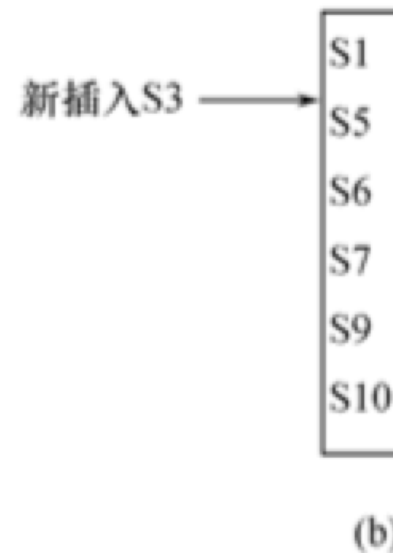
- 内模式 (也称存储模式storage schema或物理模式) :
 - 是数据物理结构和存储方式的描述;
 - 是数据在数据库内部的表示方式, 例:
 - 记录的存储方式 (顺序存储, 按照B树结构存储, 按hash方法存储)
 - 索引的组织方式
 - 数据是否压缩存储
 - 数据是否加密
 - 数据存储记录结构的规定
- 一个数据库只有一个内模式。

3. 内模式

□ 例如：学生记录，如果按**堆**存储，则插入一条新记录总是放在学生记录存储的最后，如图(a)所示。



□ 如果按学号**升序**存储，则插入一条记录就要找到它应在的位置插入，如图(b)所示。



3. 内模式

- 如果按照学生年龄聚簇存放，假如新插入的S3是16岁，则应插入的位置如图(c)所示。

内模式的作用是支持用户建立适应需求（如存取效率、空间效率、数据安全）的物理结构。

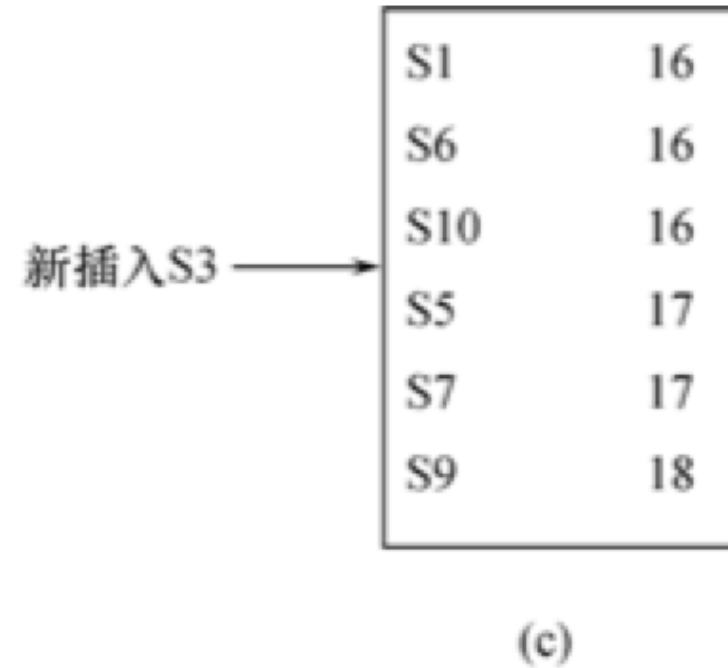


图 记录不同的存储方式示意图

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。

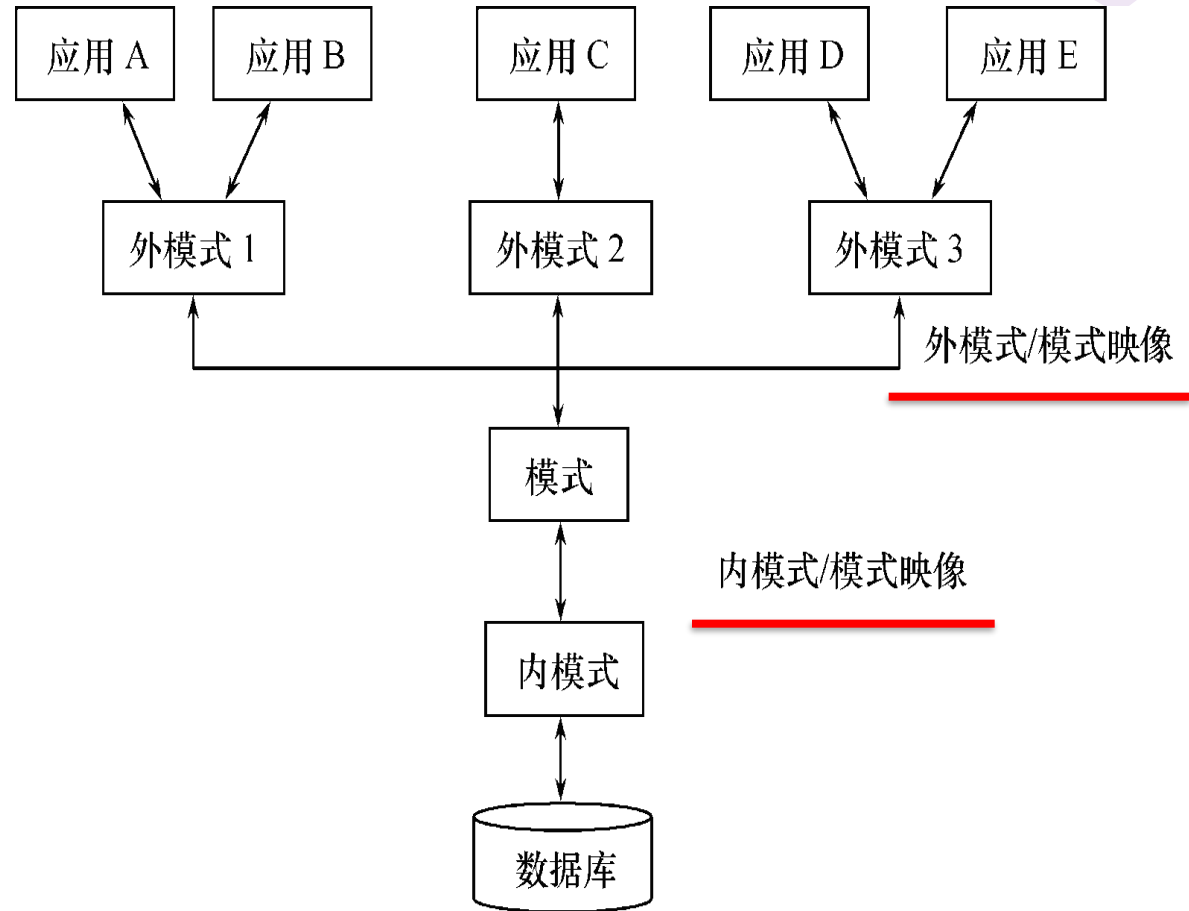
1.3.2 数据库系统的三级模式

□ 三级模式小结：

- 内模式是整个数据库实际存储的表示，反映的是数据的**存储观**；
- 概念模式是整个数据库的抽象表示，反映的是数据的**全局观**；
- 外模式是概念模式的某一部分的抽象表示，反映的是数据的**用户观**。

1.3.3 数据库的二级映像与数据独立性

- 三级模式是对数据的三个抽象级别;
- **二级映象**在DBMS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**



1. 外模式 / 模式映象

- 外模式/模式映像定义某一个外模式和模式之间的对应关系。
- 作用：保证数据的逻辑独立性
 - 当模式改变时，数据库管理员修改有关的外模式 / 模式映象，使外模式保持不变；
 - 应用程序是依据数据的外模式编写的，从而应用程序不必修改，保证了数据与程序的逻辑独立性，简称数据的逻辑独立性。

1. 外模式 / 模式映象

student

old模式

NO	NAME	AGE	SEX
301	赵钱	20	男女
302	钱	21	男女



Create View Stud(学号,姓名,性别,年龄)
As Select NO, NAME, AGE, SEX,
From student

Stud (外模式)

学号	姓名	性别	年龄
301	赵钱	男女	20
302	钱	男女	21

映像变化
而外模式
不变

模式变化

student

new模式

XH	XM	XB	CSRQ
301	赵钱	男女	95.07.01
302	钱	男女	94.03.07



Create View Stud(学号,姓名,性别,年龄)
As Select XH,XM,XB,
datediff(year,CSRQ,getdate())
From student

2. 模式 / 内模式映像

- 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。
 - 如：说明逻辑记录 and 字段在内部是如何表示的。
- 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；
 - 存储结构变化的影响被限制在模式之下，这使数据的存储结构和存储方法独立于应用程序应用程序不受影响。保证了数据与程序的物理独立性，简称数据的物理独立性。

三级模式结构的优点

□ 保证数据的独立性

- 将模式和内模式分开，保证了数据的物理独立性；
- 将外模式和模式分开，保证了数据的逻辑独立性。

□ 简化了用户接口

- 按照外模式编写应用程序或敲入命令，而不需了解数据库内部的存储结构，方便用户使用系统。

□ 有利于数据共享

- 在不同的外模式下可有多个用户共享系统中数据，减少了数据冗余。

□ 利于数据的安全保密

- 在外模式下根据要求进行操作，不能对限定的数据操作，保证了其他数据的安全。

第一章 绪论

- 1.1 数据库系统概述
- 1.2 数据模型
- 1.3 数据库系统的三级模式结构
- 1.4 数据库系统的组成
- 1.5 数据库系统的体系结构

1.4 数据库系统的组成

1. 硬件平台及数据库
2. 软件
3. 人员

1.硬件平台及数据库

■数据库系统对硬件资源的要求:

(1) 足够大的内存

操作系统

DBMS的核心模块

数据缓冲区

应用程序

(2) 足够大的外存

磁盘或磁盘阵列

➤数据库

光盘、磁带

➤数据备份

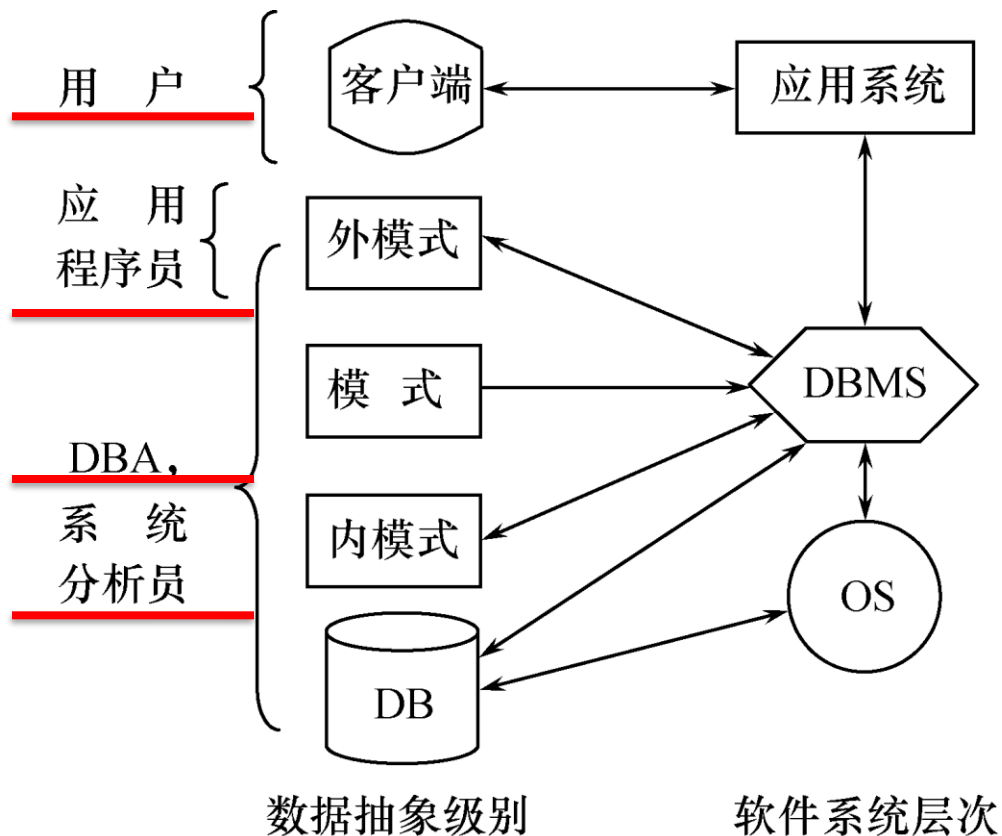
(3) 较高的通道能力，提高数据传送率

2. 软件

- 1) DBMS
- 2) 支持DBMS运行的操作系统
- 3) 与数据库接口的高级语言及其编译系统
- 4) 以DBMS为核心的应用开发工具
- 5) 为特定应用环境开发的数据库应用系统

3. 人员

□ 不同的人员涉及不同的数据抽象级别，具有不同的数据视图，如图。



- **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DBA协商，确定系统的硬软件配置
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

- **用户**是指最终用户 (End User)
最终用户通过应用系统的用户接口使用数据库

3. 人员:数据库管理员(DBA)

- DBA: 负责全面管理和控制数据库系统。
- 具体职责:
 - (1) 决定数据库中的信息内容和结构——DB设计
 - (2) 决定数据库的存储结构和存取策略——内模式设计
 - (3) 定义数据的**安全性要求**和**完整性约束条件**
 - (4) **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
 - 监视审计文件; 系统故障恢复; 介质故障恢复
 - (5) 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织, 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

补：13.2.1 RDBMS层次结构

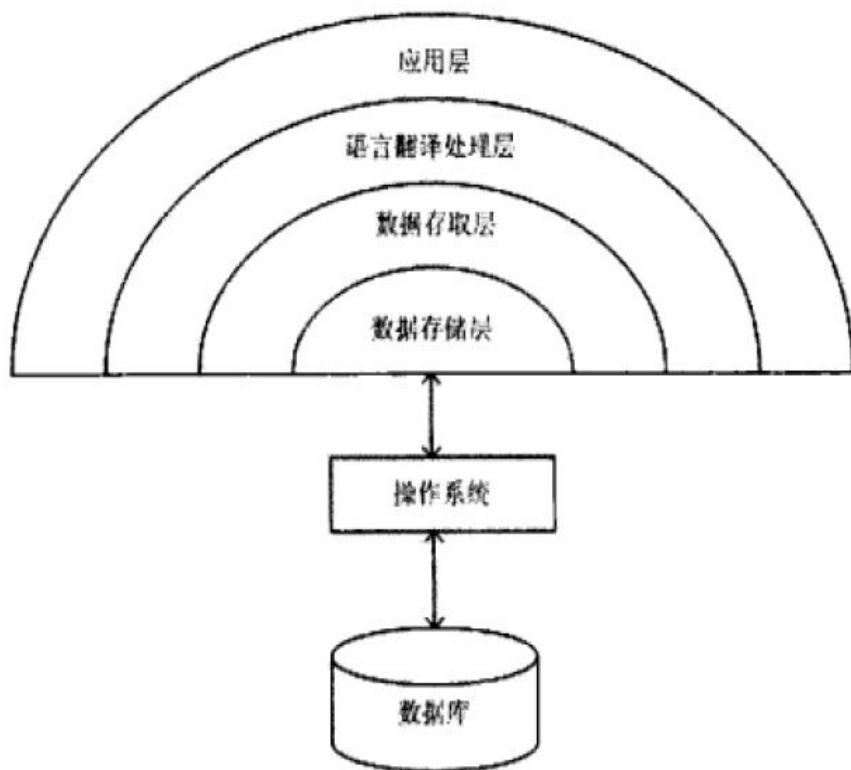


图1 RDBMS层次结构

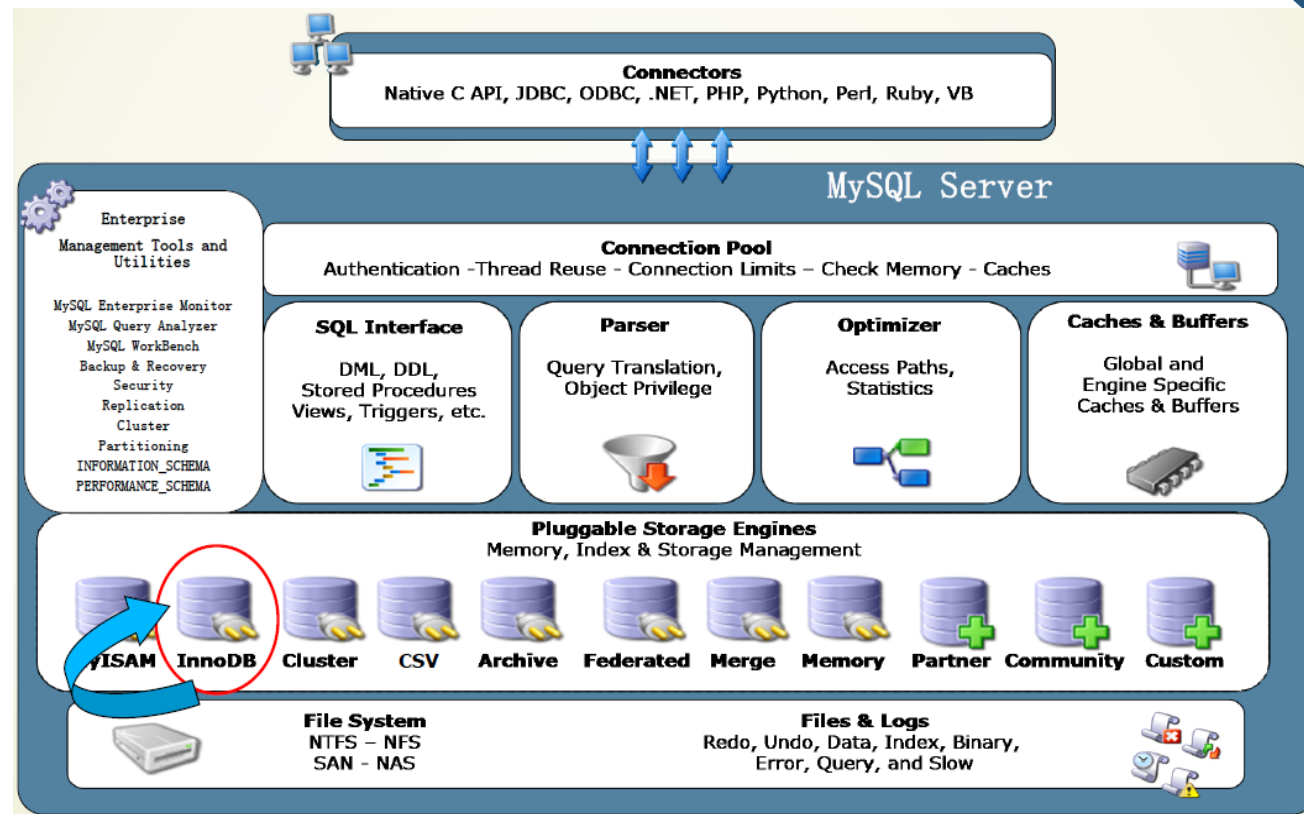
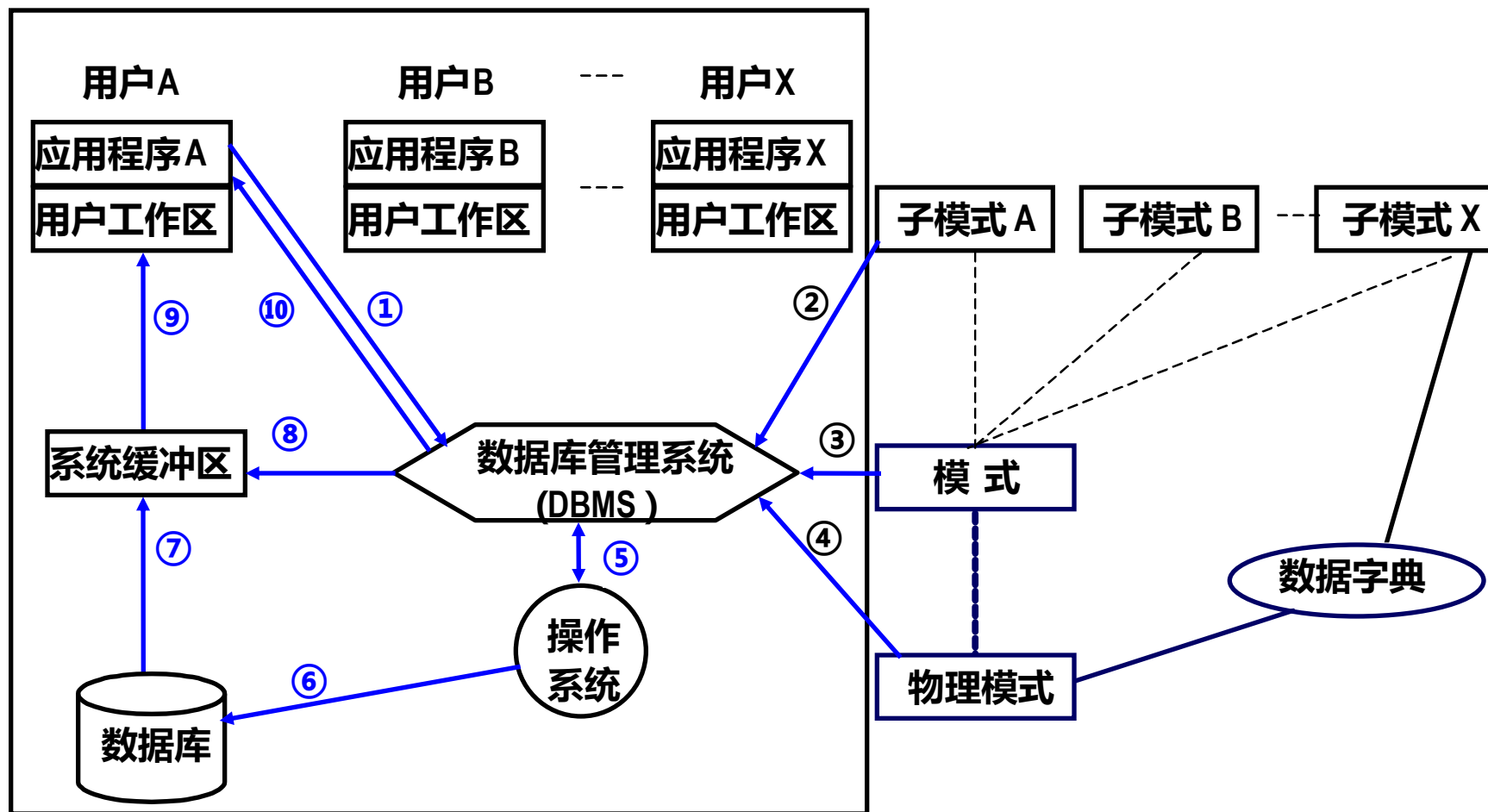


图2 Mysql服务器架构

补：13.2.2 DBMS的工作流程



DBMS工作过程

在数据库系统中，当一个应用程序或用户需要存取数据库中的数据时，应用程序、DBMS、操作系统、硬件等几个方面必须协同工作，共同完成用户的请求。

应用程序从数据库读取一个数据通常需要以下步骤：

1. 应用程序A向DBMS发出从数据库中读数据记录的命令；
2. DBMS对该命令进行语法检查、语义检查，并调用应用程序A对应的子模式，检查A的存取权限，决定是否执行命令，如果拒绝执行，则向用户返回错误信息；
3. 在决定执行该命令后，DBMS调用模式，依据子模式/模式映象的定义，确定应读入模式中的哪些记录；

DBMS工作过程（续）

4. DBMS调用物理模式，依据模式/物理模式映象的定义，决定从哪个文件、用什么存取方式、读入哪个或哪些物理记录；
5. DBMS向操作系统发出执行读取所需物理记录的命令；
6. 操作系统执行读数据的有关操作；
7. 操作系统将数据从数据库的存储区送到系统缓冲区；
8. DBMS依据子模式/模式映象的定义，导出应用程序A所要读取记录的格式；
9. DBMS将数据记录从系统缓冲区传送到应用程序A的用户工作区；
10. DBMS向应用程序返回命令执行情况的状态信息。

1.5* 数据库系统的体系结构

- 集中式数据库系统
- 客户-服务器数据库系统
- 并行数据库系统
- 分布式数据库系统
- 云数据库系统

本章小结

□ 数据库系统概述

- 数据库的基本概念
- 数据管理的发展过程

□ 数据模型

- 数据模型的三要素
- 概念模型, E-R 模型
- 三种主要数据库模型 (了解各种模型的主要特性, 数据库系统的特点)

□ 数据库系统的结构

- 数据库系统三级模式结构
- 数据库系统两层映像系统结构

□ 数据库系统的组成

作业:

P30 3, 15

(提交时间: 布置后一周)

提交方式: 学在华科大)

回顾本章主要内容

- 什么是数据库系统?
- 产生DBMS的动机是什么?
- 如何设计数据库?
- 数据库系统的组成?
- 数据库系统的结构?

数据库系统是由DB、DBMS、应用系统、DBA组成的存储、管理、处理和维护数据的系统。

见P7 表1.1, 要答出数据库系统的特点

数据模型: ER模型, 逻辑模型, 物理模型

硬件平台+DB, 软件, 人员

三级模式, 二级映像

讨论题

□ 数据库、数据库管理系统、数据库应用系统、数据库系统的区别是什么？

存在的问题：

1. “数据库应用系统是一个系统软件？”

错误！

大家学过的系统软件只有：OS，DBMS，语言处理程序（如各种编译器）。应用系统无论大小，只要是面向特定应用的，都属于应用软件。

2. “数据库应用系统属于数据库管理系统？”

错误！12306是一个应用，它能是一个DBMS（如：Mysql）的一部分吗？显然不是。



Chapter 1 ends。 。 。

追求

休息一会儿。 。 。

