

Application of Database

数据库应用

聂斌玲

binlingnie@hdu.edu.cn

第一章 数据库系统概述

第一节 基本概念

第二节 数据管理技术的发展阶段

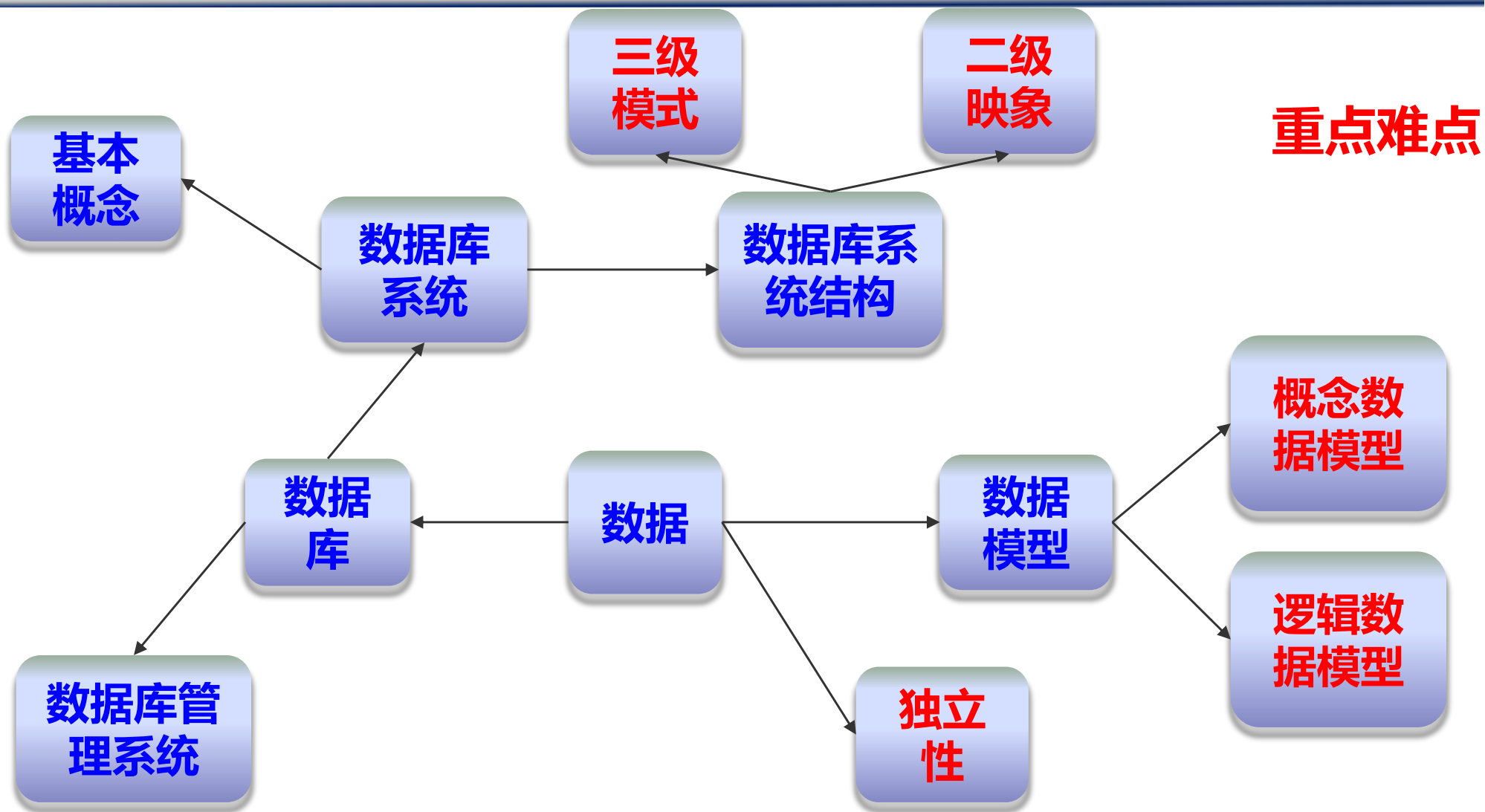
第三节 数据模型

第四节 数据库系统结构

第五节 数据库系统的组成

第六节 数据库管理系统

本章主要教学内容

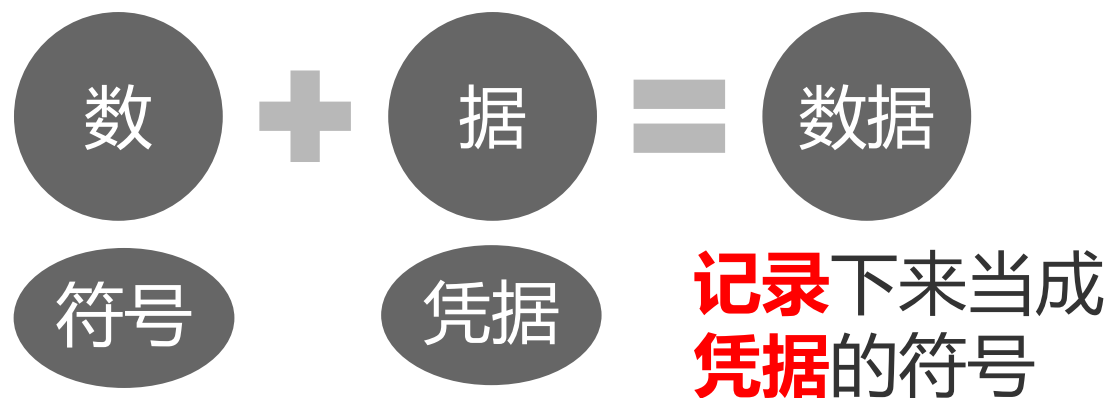


要求掌握这些基本概念

第一节 数据库的四个基本概念

- ▶ **数据: Data**
 - ▶ **数据库: DB, Database**
 - ▶ **数据库管理系统: DBMS**
 - Database Management System
 - ▶ **数据库系统: DBS**
 - Database System
-

什么是数据?



数据—Data, fact

1. 对客观事物的**符号表示**
2. 在计算机科学中是指**所有输入到计算机中**并被计算机程序处理的**符号**的总称
3. 描写事物的符号有**多种形式**，可以是数字、文字、图形、图像和声音，但它们都是经过**数字化**后存入计算机的。
4. 数据是**数据库中存储的基本对象**

什么是数据库？

► **R.W.Engles**: DB是供某个特定机构的应用系统使用的可操作数据的集合。

强调可操作性

► **James Martin**: 把DB比喻作“各种数据用户都可去钓鱼、用水的水库”。

共享性

► **C.J.date**: DB只不过是一个计算机化的记录存储的系统。

数据的组织

► **J.D.Ullman**: DB是存储在计算机系统中的数据。

机器中数据的集合

有需求，有**数据**源，有设施，有空间，实现**数据**控制
有组织，有管理，可共享，有服务，服务于不同的目的

什么是数据库(DB)

数据库是长期存放在**计算机内**、**有组织的**、**可共享的**数据集合。

数据库中的数据按一定的**数据模型**组织、描述和存储，且具有**较小的冗余度**、**较高的数据独立性和易扩展性**，并可为多个用户共享。

A database is an organized collection of **data**.

Formally, a "database" refers to a set of related data and the way it is organized.

数据库管理系统(DBMS)

数据库管理系统是用于**建立、使用和维护**数据库的软件。

“建立”：如何定义、组织和存储数据

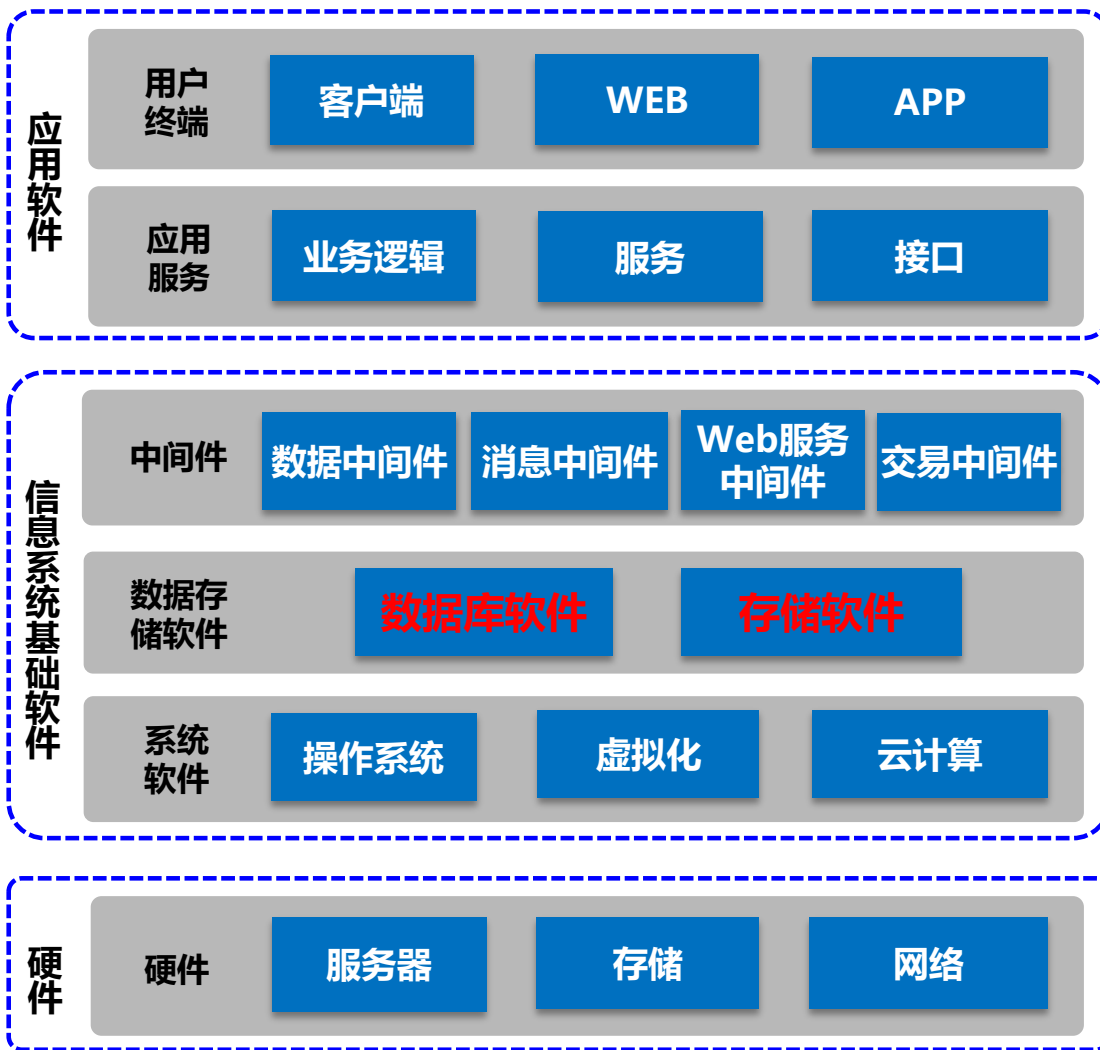
“使用”：查询、插入、删除和修改数据

“维护”：保证数据的安全性、完整性

A database-management system (DBMS) is a **computer software application** that **interacts** with **end-users, other applications**, and the database itself to **capture, provide access to** and **analyze** data.

DBMS的位置

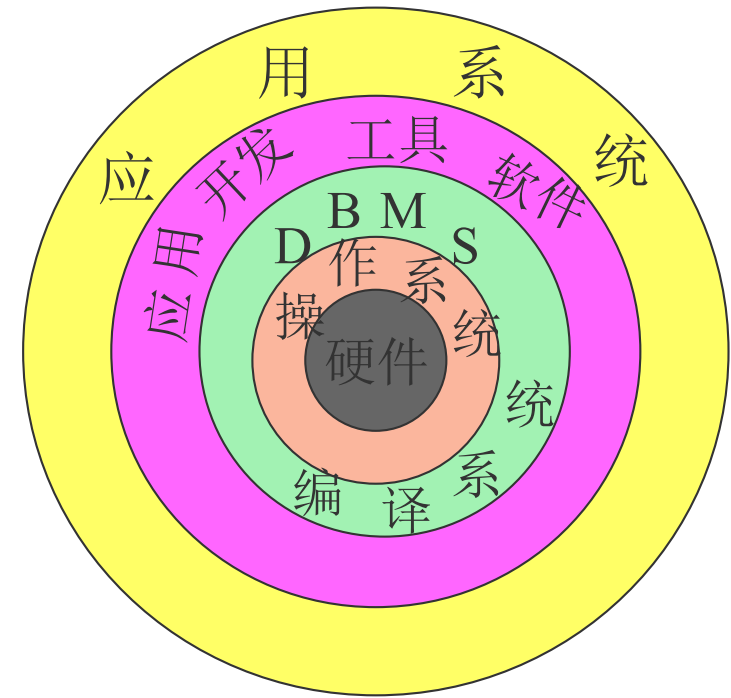
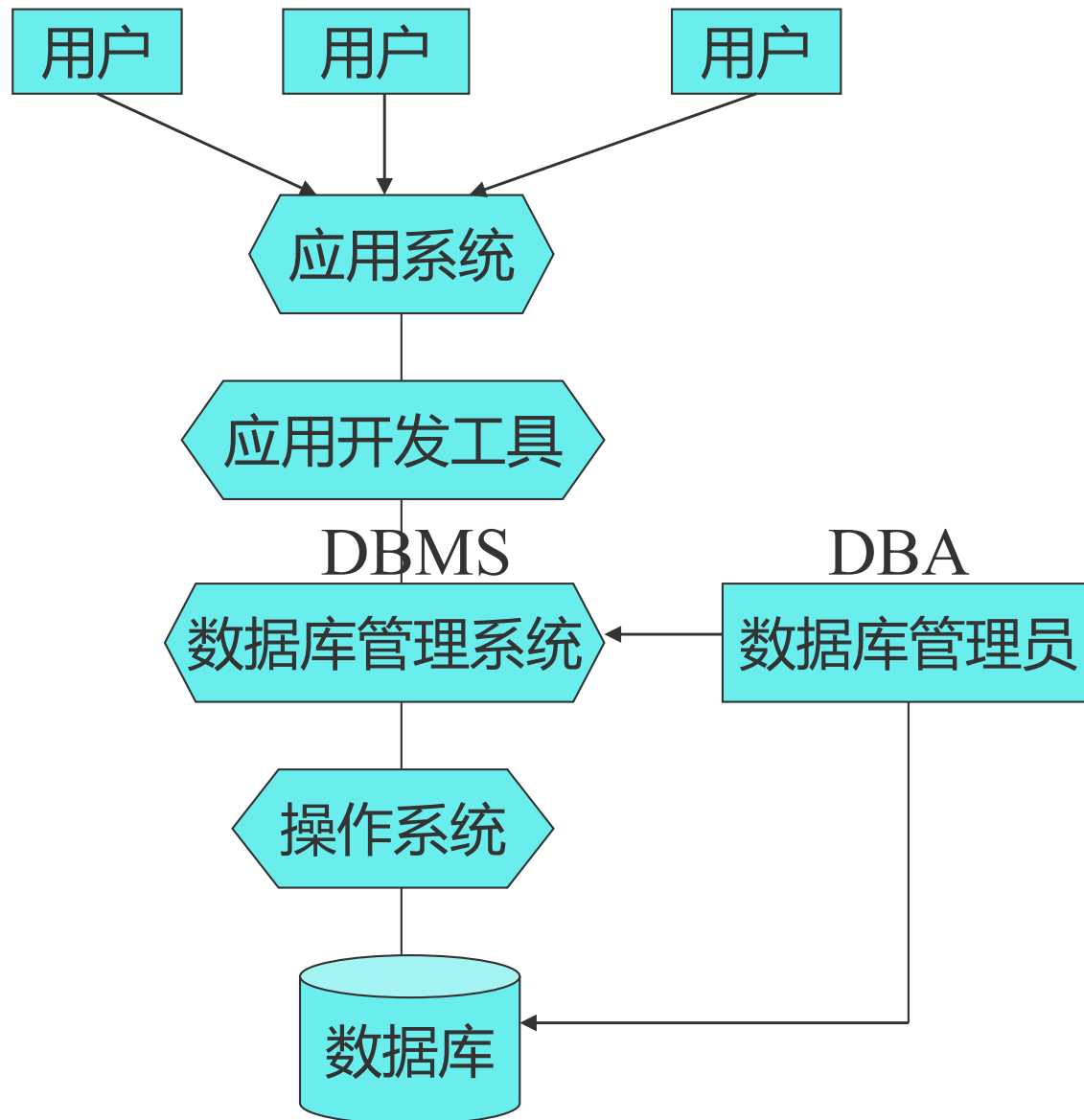
现代信息系统逻辑架构



操作系统之上
离不开操作系统

提供的功能:
数据库的建立
数据定义
数据组织、存储和管理
数据操纵
数据库事务管理和运行管理
数据库维护

数据库系统—Database System(DBS)



采用了数据库技术的计算机系统，包括DB、DBMS、开发工具、应用系统、硬件、软件、用户和DBA

数据处理与数据管理

数据处理:

是对各种形式的数据进行**收集、储存、加工、管理和传播**的一系列活动的总和。

数据管理:

是指数据的**分类、组织、编码、存储、维护、检索**等操作，是数据处理的核心。

数据的管理经历了多个阶段

第二节 数据管理技术的发展阶段

1. 人工管理阶段
 2. 文件系统阶段
 3. 数据库系统阶段
 4. 高级数据库技术阶段
-

人工管理阶段

1950年代中期以前

发展
背景

外存：只有磁带、卡片和纸带

软件：只有汇编语言，无数据管理方面的软件

数据处理方式：批处理

数据存储在**磁带、卡片和纸带**上，以**脱机**方式存放，**人工**标签管理，类似于**纸质图书管理**。需要用的时候**接入**计算机，**只有人知道**哪些数据在哪里，完全**依赖于人**进行数据管理

人工管理阶段



特点： 一组数据对应一个程序，数据是面向应用的。

特点:

(1) 数据**不长期保存**

(2) **应用程序管理**数据

没有专有的软件对数据进行管理, (应用程序要规定数据的逻辑结构和物理结构, 包括存储结构, 存取方法, 输入方式)。

(3) 数据**不具有独立性**

数据的逻辑结构或物理**结构发生变化**后, 必须对应用程序做**相应的修改**。

(4) 数据**不共享**

数据面向程序, 即**一组数据对应一个程序**, 当多个应用涉及相同数据时, 程序之间有大量的冗余数据。

几个重要问题，请认真思考

什么叫**独立**和**耦合**？

如何才能**独立**？

独立与**成长**有什么关系？

独立有什么**利弊**？

独立、共享、合作有什么关系？

独立、分组、组件化、模块化的关系？

2. 文件系统阶段

1950年代后期至60年代中期以前

发展
技术
背景

存储技术发展：

已有磁盘、磁鼓等**直接存取**存储设备

软件技术发展：

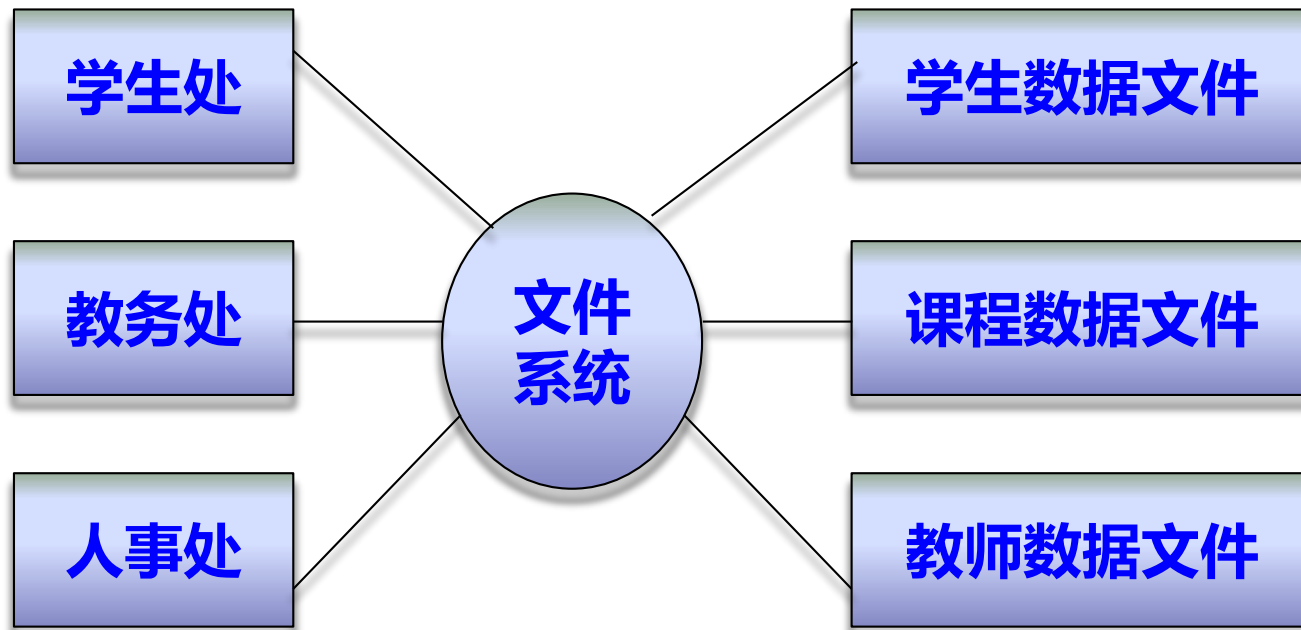
出现了**高级语言**和**操作系统**，OS中的**文件系统**是专门管理外存的**数据管理软件**

数据处理方式发展：

批处理、联机实时处理（OLTP）

你知道哪些处理：**批处理、并行处理、串行处理、分布式处理、并行与分布式处理、联机实时处理、脱机（离线）处理、近实时处理？**

文件系统阶段



数据文件案例

▶ 学生成绩文件

▶ 学号	姓名	成绩
▶ 03601441	马志阳	70
▶ 03601442	杨蕊	76
▶ 03601443	王文超	74
▶ 03601444	周珺	94
▶ 03601445	杨倩	82
▶ 03601446	李喆	85
▶ 03601447	赵静贤	85
▶ 03601448	刘旭	67
▶ 03601449	林霖	75

特点:

(1) 数据可长期保存

在外存的磁盘中

(2) 由文件系统管理数据

实现按文件名访问,按记录存取。

(3) 数据共享性差, 冗余度大

一个文件基本上对应一个应用程序, 不同的应用程序具有相同的数据时,造成数据的**冗余**, 以及由于重复存储, 各自管理造成数据的**不一致性**。

(4) 数据独立性差

一旦数据的逻辑结构改变,必须修改应用程序,修改文件结构的定义。因此**数据与程序之间仍缺乏独立性**。

3. 数据库系统阶段

- ▶ 数据管理技术进入数据库阶段的标志，是1960年代末发生的三件大事：
 - 1968年，美国IBM公司推出**层次模型**的IMS系统
 - 1969年，美国数据系统语言协会(CODASYL)的数据库任务组 (DBTG)发表关于**网状模型**的DBTG报告 (1971年通过)
 - 1970年，美国IBM公司的E.F. Codd提出**关系模型**，奠定了关系数据库的理论基础。
-

技术发展背景

发展 技术 背景

硬件技术发展：

已有大容量磁盘、磁鼓等直接存取存储设备，
硬件价格下降

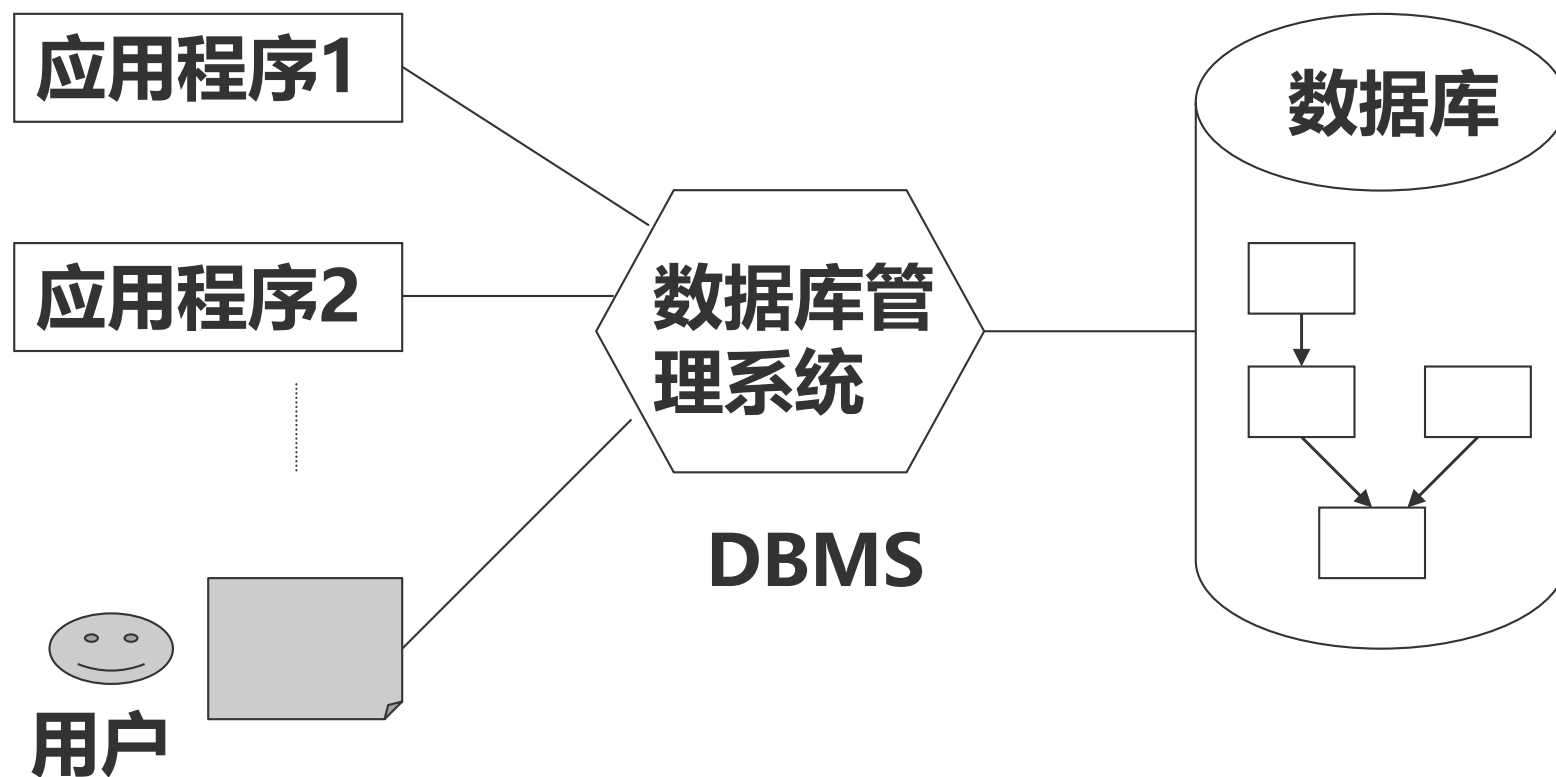
软件行业及技术发展：

软件价格上升，为编制和维护系统软件及应用**软件所需成本**相对增加；

数据处理要求演变：

联机实时处理要求更多，分布式处理需求。

数据库管理系统的出现



- 在这种背景下，为解决多用户，多应用共享数据的需求,使数据为尽可能多的应用服务，数据库技术便应运而生，出现了统一管理数据的**专门软件系统--数据库管理系统**。

数据库系统阶段的管理方式特点

(1)数据结构化

数据模型不仅描述**数据本身**的特点，还描述数据之间的**联系**。整体数据结构化。不仅数据是结构化的，存取数据的方式也灵活。

(2)数据共享性高, 冗余度低, 易扩充

数据是面向整个系统，不是面向应用的，所以多应用多用户**可共享数据**，**减少冗余**，避免数据的不一致性。容易增加新的应用，**易于扩充**。

数据库系统阶段的管理方式特点:

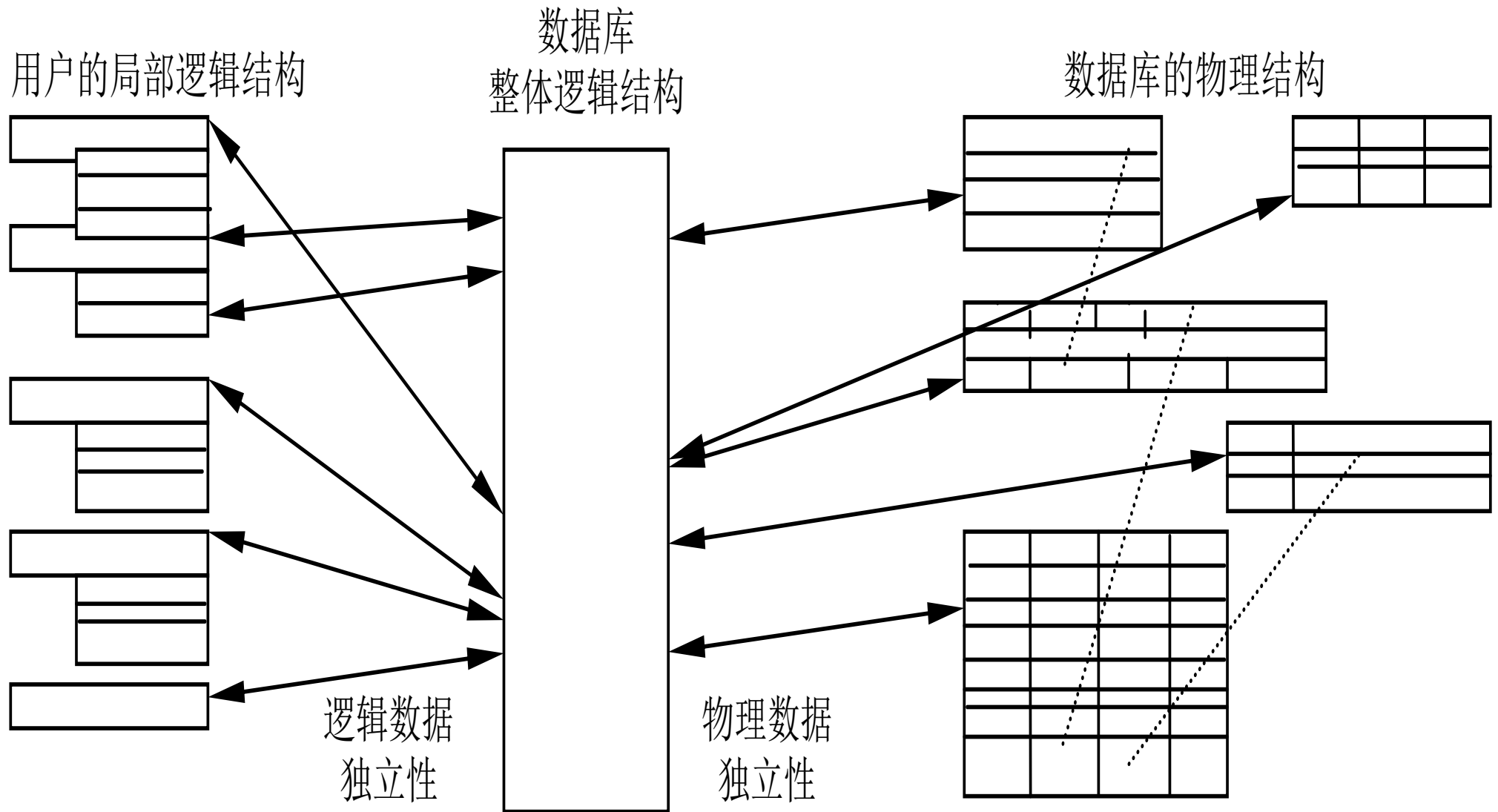
(3)数据独立性高;

包括物理独立性和逻辑独立性。指数据在磁盘上的物理存储改变和逻辑结构改变,应用程序不用变。数据独立性由DBMS的二级映象功能保证。数据与程序独立,把数据的定义从程序中分离,简化程序的编制、维护和修改工作。

(4)数据由DBMS统一管理和控制;

DBMS提供四个方面的数据控制功能:数据的恢复、并发控制、数据完整性和数据安全性。

Mappings between three layers



数据库系统与传统文件系统的根本区别

(1) 在**文件系统**中，记录内部有了某些结构,但记录之间没有联系。**数据库系统实现整体数据结构化。**

“整体结构化”是指数据库中的数据不再仅仅针对某个应用，而是**面向全组织（企业）**；不仅数据内部是结构化的，而且整体是结构化的，**数据之间是具有联系的。**

数据库系统与传统文件系统的根本区别

(2) 在**文件系统**中, 数据的最小存取单位是记录, 粒度不能细到数据项。

数据库系统中可以存取数据库中某一个数据项, 一组数据项, 一个记录, 一组记录。

4. 高级数据库技术阶段

20世纪80年代开始

分布式数据库

并行数据库

面向对象数据库

XML数据库

大数据—集群分布式存储并行处理

第三节 数据模型

什么是模型？

► 用于指导**最终产品制造**的初步作品或结构

- A preliminary work or construction that serves as a plan from which a final product is to be made.

► 用于**测试和完善**最终产品作品或结构

- Such a work or construction used in testing or perfecting a final product

► 一种系统、理论或现象的模式描述，用于**说明其已知或推导出的属性**，并用于研究其进一步的特性

- A schematic description of a system, theory, or phenomenon that accounts for its known or inferred properties and may be used for further study of its characteristics

► 用于**模仿或比较**的案例性东西

- One serving as an example to be imitated or compared.

► **模型**：用于对现实世界**特征的**进行**模拟和抽象的事物**

数据模型

数据模型也是一种模型，是现实世界**数据特征的抽象**。

A data model is an **abstract model** that **organizes elements of data** and **standardizes** how they **relate to one another** and to **properties** of the **real world entities**.

从用途的角度进行定义：

数据模型是用于**描述现实世界**，指导数据的**设计、存储**，并作为数据设计**结果的检验标准的模型**。

(一) 数据模型相关概念

▶ 数据模型应**满足三个要求**:

- 真实地**模拟**现实世界
- 容易**理解**
- 便于计算机上**实现**

▶ 数据模型相关概念

- 数据模型的**层次**
 - 概念、逻辑、物理；高层、中间层、低层
 - OLTP层、数据集成层、ODS或近线层、数据仓库层或离线层
 - 数据模型的**描述工具**
 - 数据模型的**通用性**
 - **行业**数据模型、**企业**级数据模型
 - 主题数据模型、Featuring、自动建模、...
-

数据模型三个组成部分

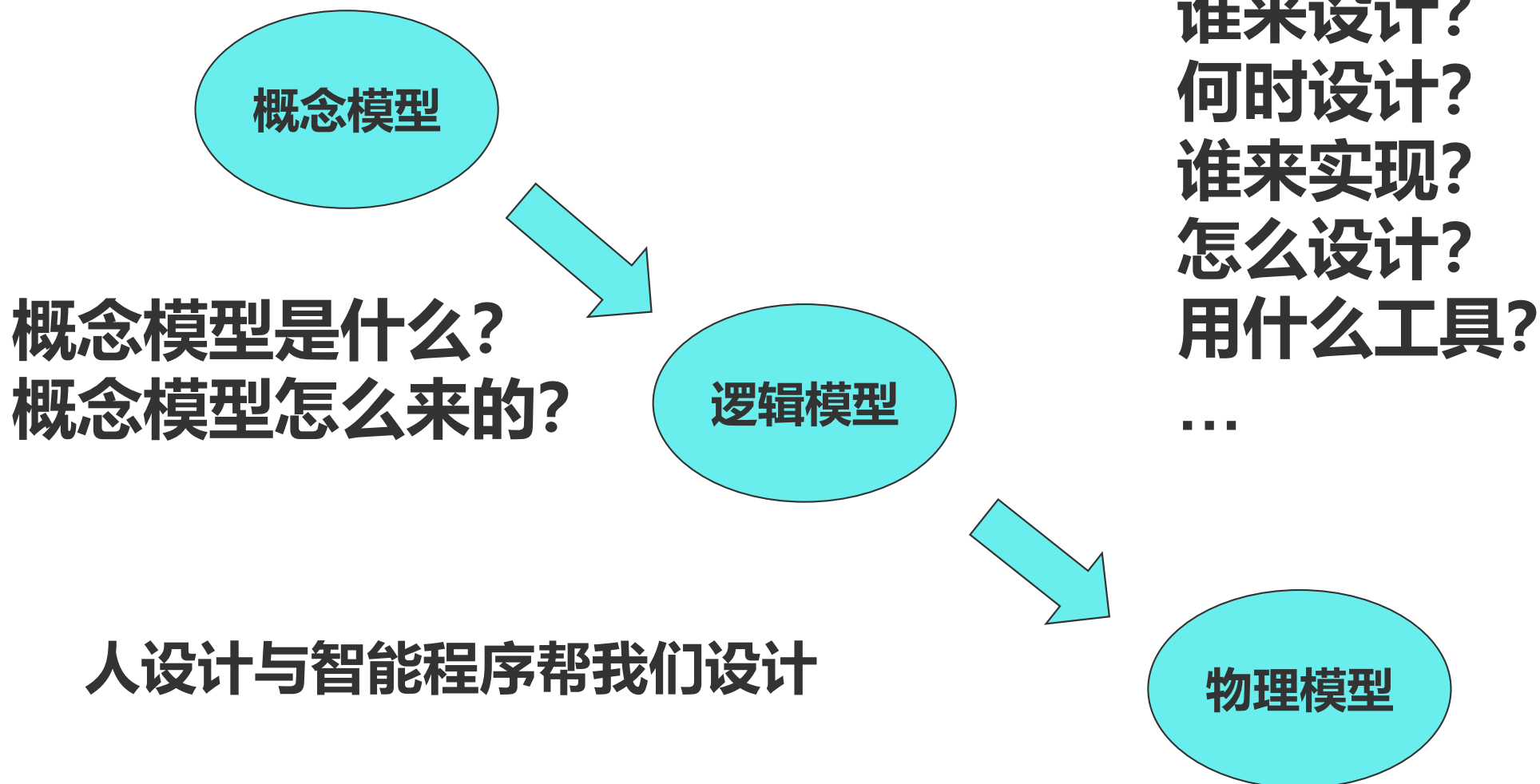
(1)**数据结构**: 是所研究对象类型的集合, 即对**实体类型**和**实体间联系**的表达和实现, 是系统**静态特征**的描述。

(2)**数据操作**: 是指对数据库中各对象**允许执行的操作的集合**, 包括**检索**和**更新**两类, 是系统**动态特征**的描述。(操作是否归入数据模型有争议)

(3)**数据完整性约束**: 给出**数据**及其**联系**应具有的制约和依存规则, 保证数据的正确、有效、相容。有**实体完整性**、**参照完整性**和**用户定义**的完整性。

数据模型及设计问题

有什么样的数据模型？ 设计先后次序？



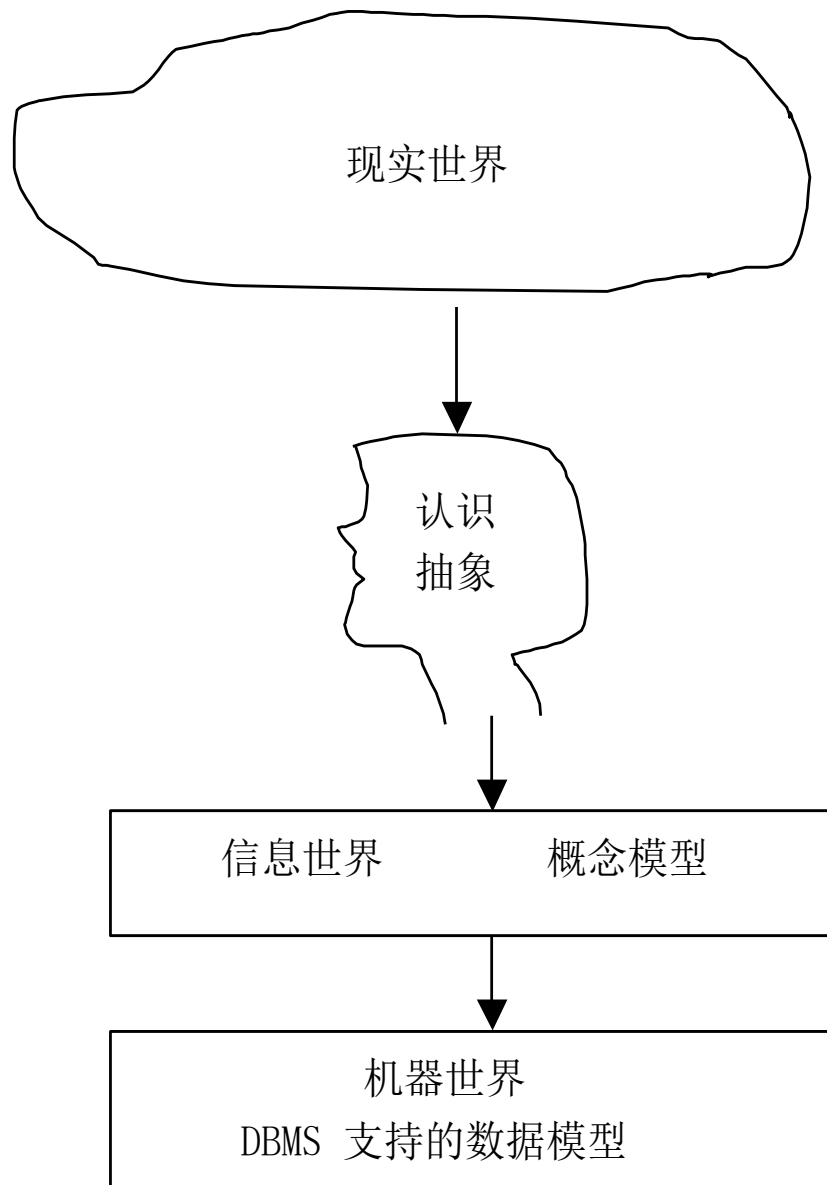
数据模型相关的三个世界或领域

(1)**现实世界**：存在于人脑之外的**客观世界**

(2)**信息世界**：人们通过**头脑**对现实世界进行**抽象**，并通过**描述工具**以实体、实体集、属性、实体标识符等形式**尽可能真实准确地**描述现实世界。

(3)**机器世界**：信息世界的信息在机器世界中的表示。以**数据形式**存储字段、记录、文件、关键码

现实世界中客观对象的抽象过程

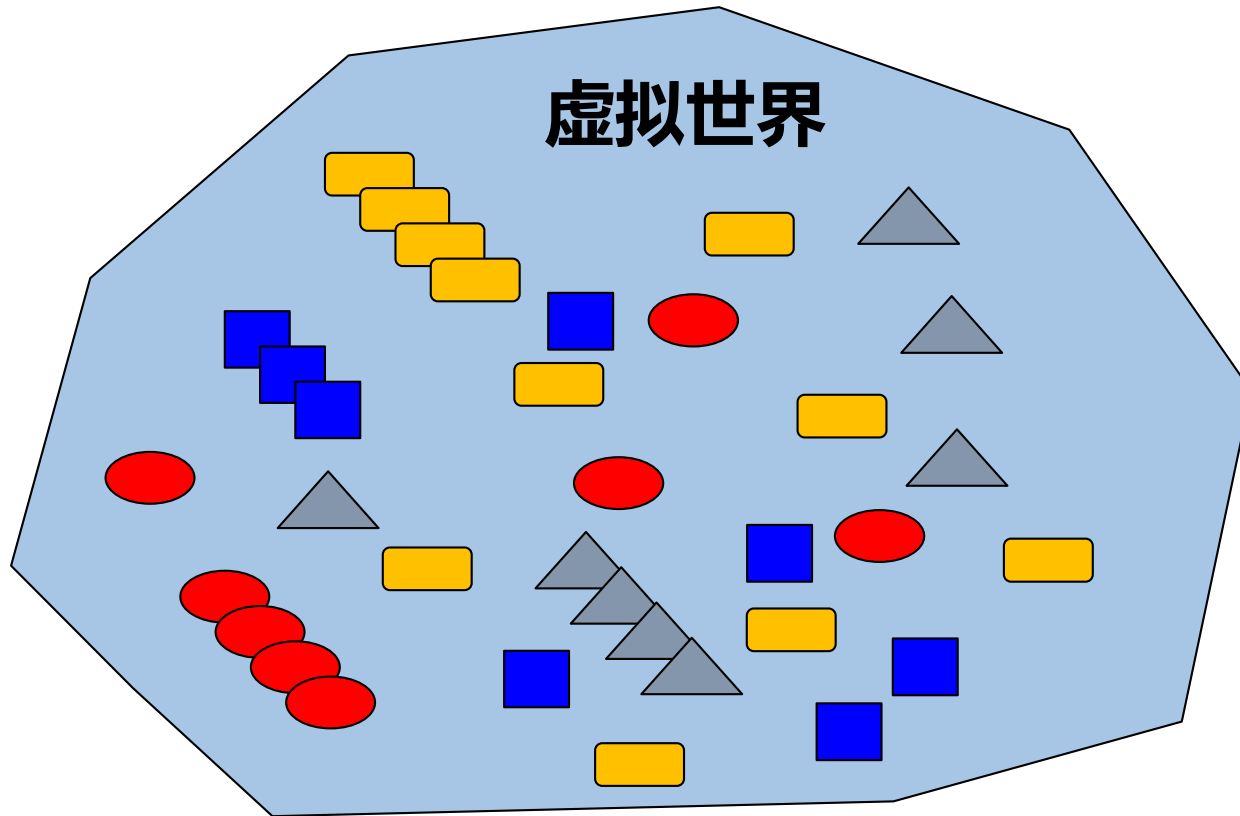


**平行世界
信息物理系统CPS**

思考：此图在人工智能场景下，对应于类似的什么问题？

面向对象程序设计中现实世界表示

► 在机器的软件世界里表示现实世界



程序员写类：

程序员用外部语言去刻画一个虚拟世界中的一类对象的结构、行为，即数据模型与行为模型。

对类进行编译或解释：

最终形成某个机器世界所支撑的一个虚拟世界中的一类对象的内部模型

创建一个世界，开始运作，新的对象不断的出现，对象在持续生命，不断地活动，消耗资源，产生输出，对象不断的消失，直至整个世界消亡

(二) 概念模型

► 概念模型的功能

- 在**信息世界**里对**现实世界**进行建模
- **表达**现实世界中的**概念**
- **表达**概念之间的**关系**
- 表达业务领域的**语义知识**

► 对概念模型的要求

- 具有很强的**表达能力**--能力强
 - 易于理解—容易懂
 - 简单—容易画
-

概念模型的定义及特点

为正确直观地反映客观事物及其联系，对所研究的信息世界，按用户观点对数据和信息建模型，称之为概念模型。(用于数据库设计)

特点：

(1) 是独立于计算机系统的模型，完全不涉及信息在系统中的表示

(2) 用于建立信息世界的数据库模型，是现实世界的第一层抽象（是现实到机器的中间层）

(3) 强调语义表达功能，概念简单、清晰，易于用户理解，是用户和DB设计人员之间交流的语言，是DB设计人员进行数据库设计的工具

1、信息世界的术语

▶ 实体—entity

- 可以相互区别的客观事物和概念的抽象，如人，学生，教师， ...

▶ 属性—attribute

- 对实体某一特征的描述，如学号，姓名，年龄， ...

▶ 域—domain， 属性定义域

- 属性取值范围，如年龄：[0, 200]

▶ 实体的码、实体标识符—entity key or identifier

- 能唯一标识每个实体的属性集，如：学号

▶ 实体型—entity type or entity schema， 实体模式， 实体类型

- 用实体名及其属性名的集合来抽象和刻画同类实体，称为实体型。
 - 学生（学号， 姓名）
-

1、信息世界的术语—续

▶ 实体集—entity set

- 性质相同的同类实体的集合

▶ 联系—relationship, 关系

- 分成两类：实体内部的联系、实体之间的联系

▶ 实体内部的联系

- 实体各属性之间的联系

▶ 实体之间的联系

- 通常指不同实体集之间的联系

本课程需要重点关注的联系：

实体与属性间联系、实体内属性间联系、实体之间的联系

2、机器世界的术语

▶ 记录—record

- 字段的有序集合(实体)

▶ 文件—file

- 同一类记录或不同类记录集合(实体集)

▶ 关键码，主键，主码—key

- 能唯一标识文件中每条记录的字段或字段集（实体标识符）

▶ 字段—field

- 标记实体属性的命名单位，亦称数据项(属性)
-

记录↓

项↓

机器世界

信息世界

实体↓

实体内部
联系

属性↓

实体型→

学 生

学 号

姓 名

年龄

性别

籍贯

←记录型

实体 1↔

学生甲

000001

胡 伟

20

男

湘

←记录 1

实体 2↔

学生乙

000002

张春明

18

男

黑

←记录 2

实体 3↔

学生丙

000015

程会军

21

男

晋

←记录 3

实体 4↔

学生丁

000002

王 翌

18

女

浙

←记录 4

实体 5↔

学生戊

000002

贺正生

19

男

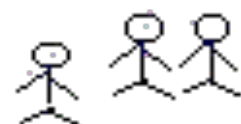
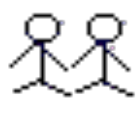
沪

←记录 5

实体之间的
联系

事物↑

特征↑



现实世界

具体实例

三个世界中的术语

3、实体集之间的联系

1:1 (一对一联系)


1:n (一对多联系)

m:n (多对多联系)

1:1 一对一联系

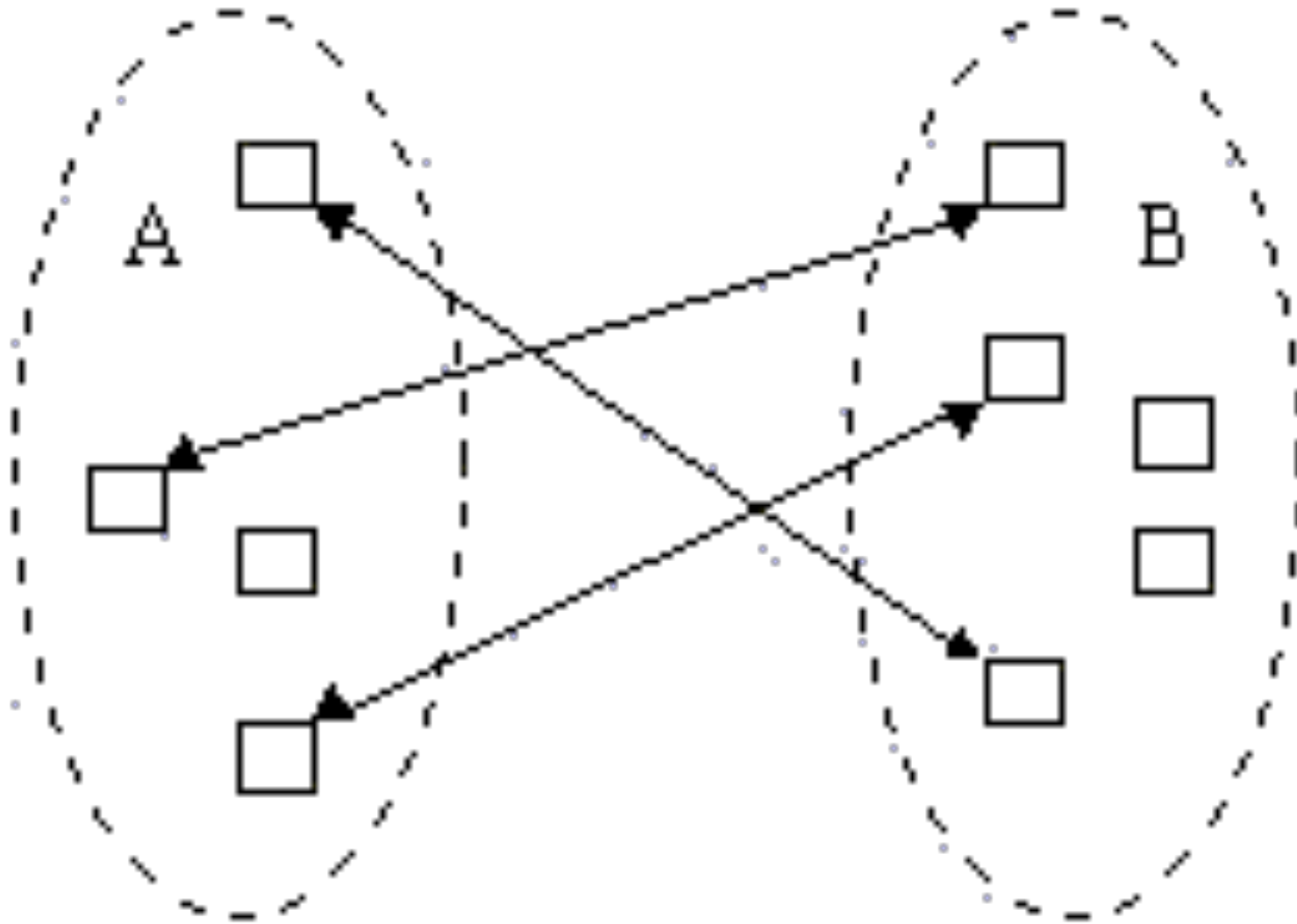
A中任意实体至多对应B中的一个实体，反之B中的任意实体至多对应A中的一个实体

例如:



观众与座位
乘客与车票
病人与病床
灯泡与灯座

1 : 1 一对一联系



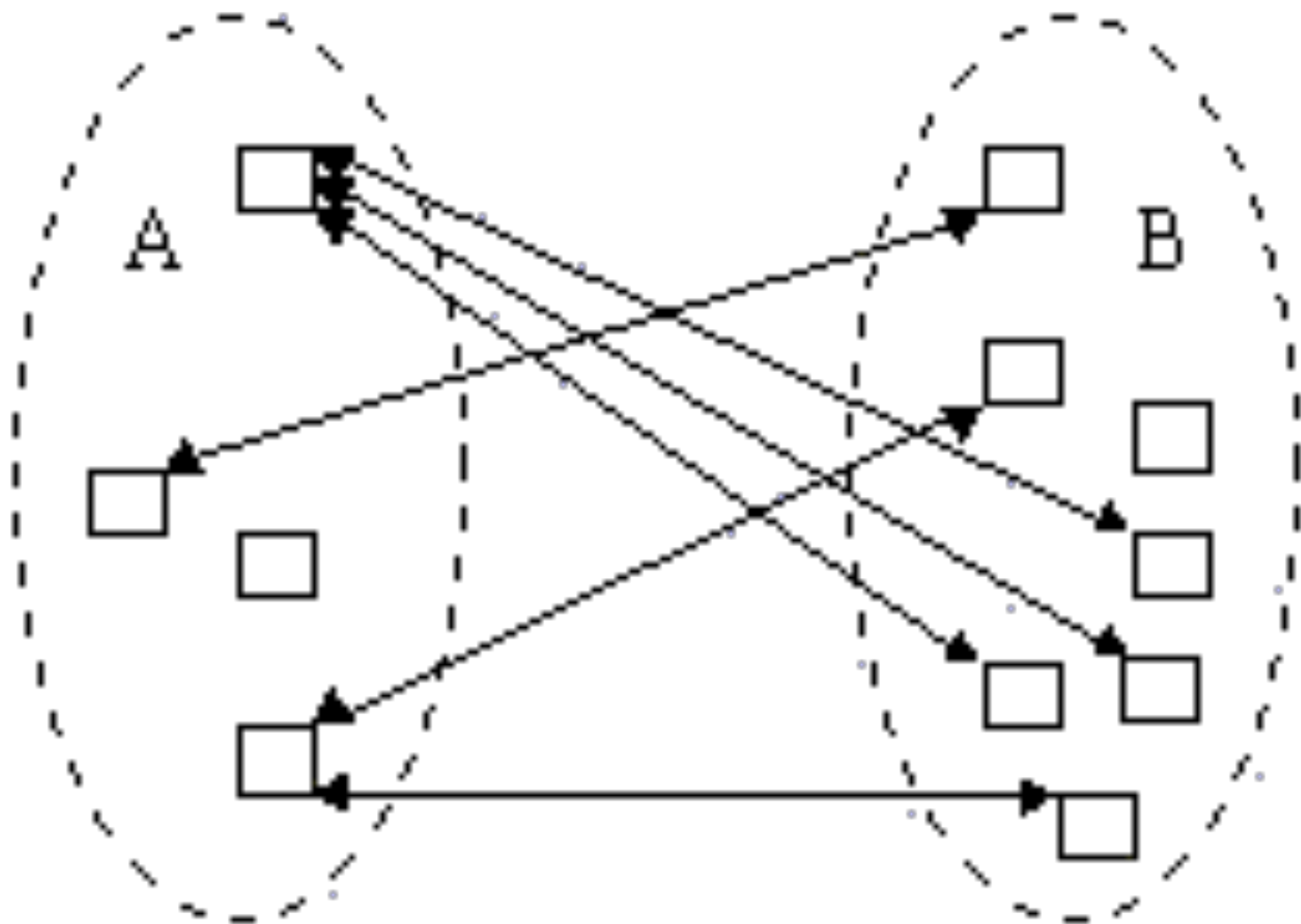
1:n 一对多联系

对于A中的每一个实体，B中有
 $n(n \geq 0)$ 个实体与之对应，反之B中的
任意实体**至多**对应A中的一个实体

例如:

城市与街道
宿舍与学生
父亲与子女
班级与学生

1:n 一对多联系图示



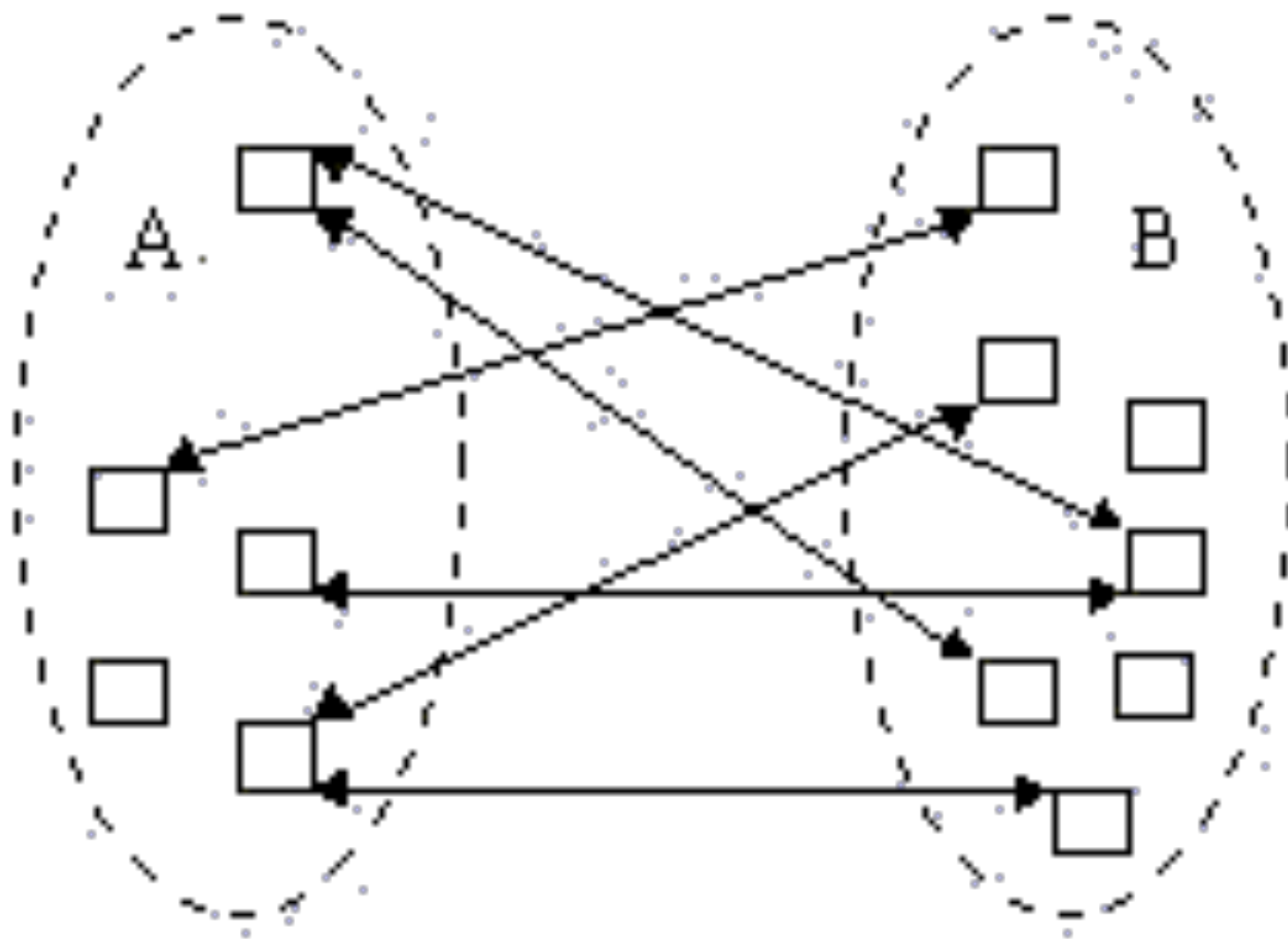
m:n 多对多联系

如果A中每一个实体，实体集B中有 $n(n \geq 0)$ 个实体与之联系，反之，B中的任一个实体，A中也有 $m(m \geq 0)$ 个实体与之联系

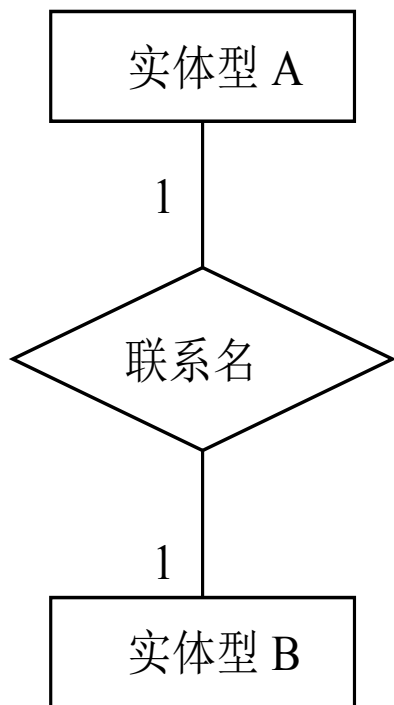
例如：

**学生与课程
工厂与产品
商店与顾客
影片与观众**

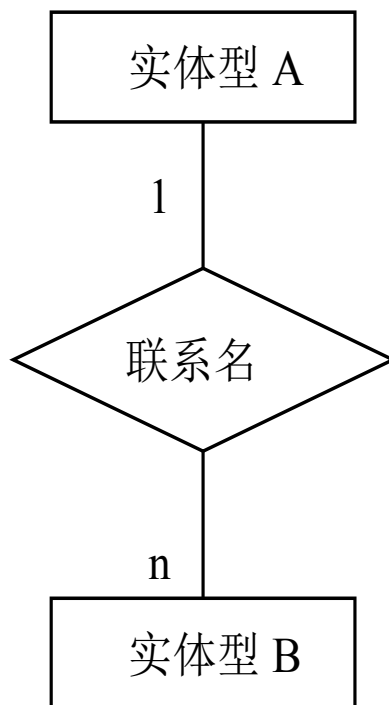
m:n 多对多联系图示



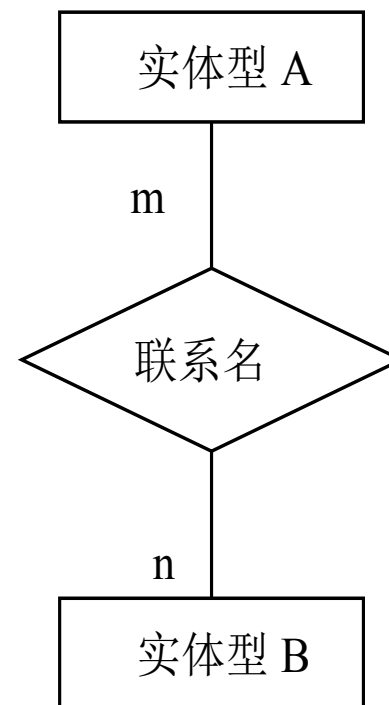
两个实体集之间的三类联系



1: 1 联系



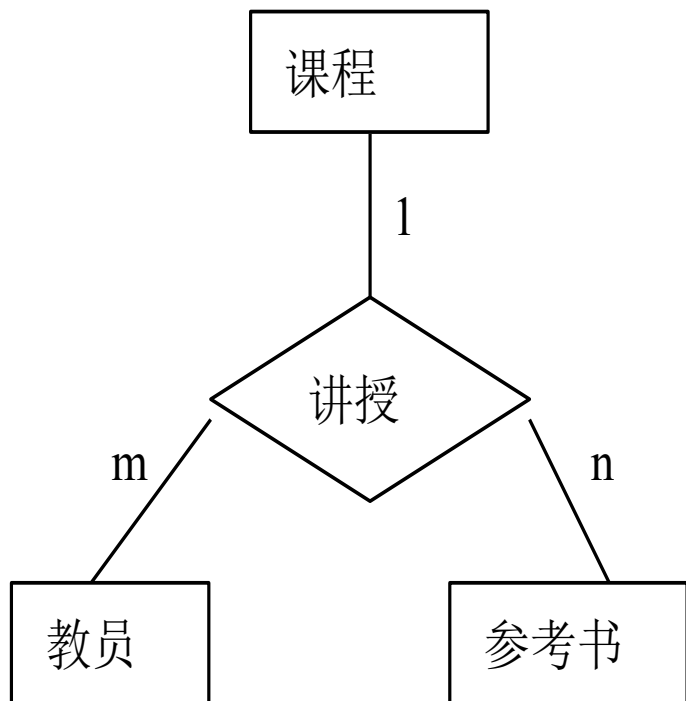
1: n 联系



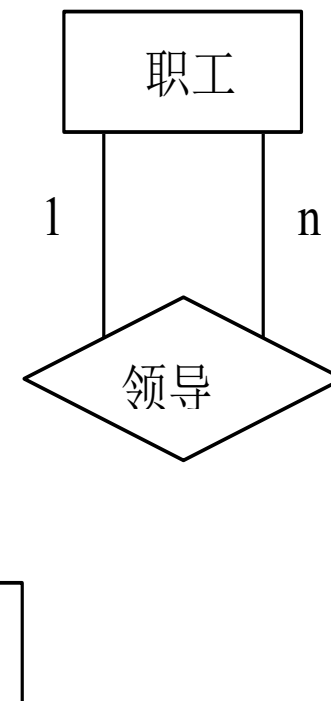
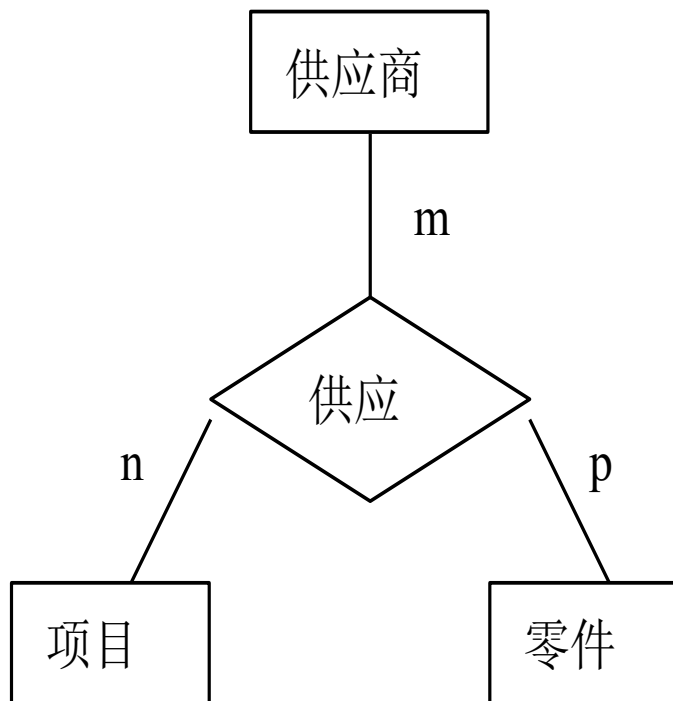
m: n 联系

两个实体型之间的三类联系

其他示例



三个实体型之间的联系示例



一个实体型之间
一对多联系示例

4. 常用的概念模型—实体联系模型

Entity relationship model, 简称**ER模型**, 是由美籍华人陈平山于1976年提出的。
ER图提供了表示**实体、属性和联系**的方法。

(1) ER模型的三要素 (三个基本语义)

A. 实体 (Entity): 表示客观事物。

B. 属性 (Attributes):

表示客观事物的特征 (属性)

C. 联系 (Relationships):

客观事物之间的联系

(2) 刻划工具 — 实体联系图 (ER图)

ER图表示方法:

用**矩形**表示**实体**

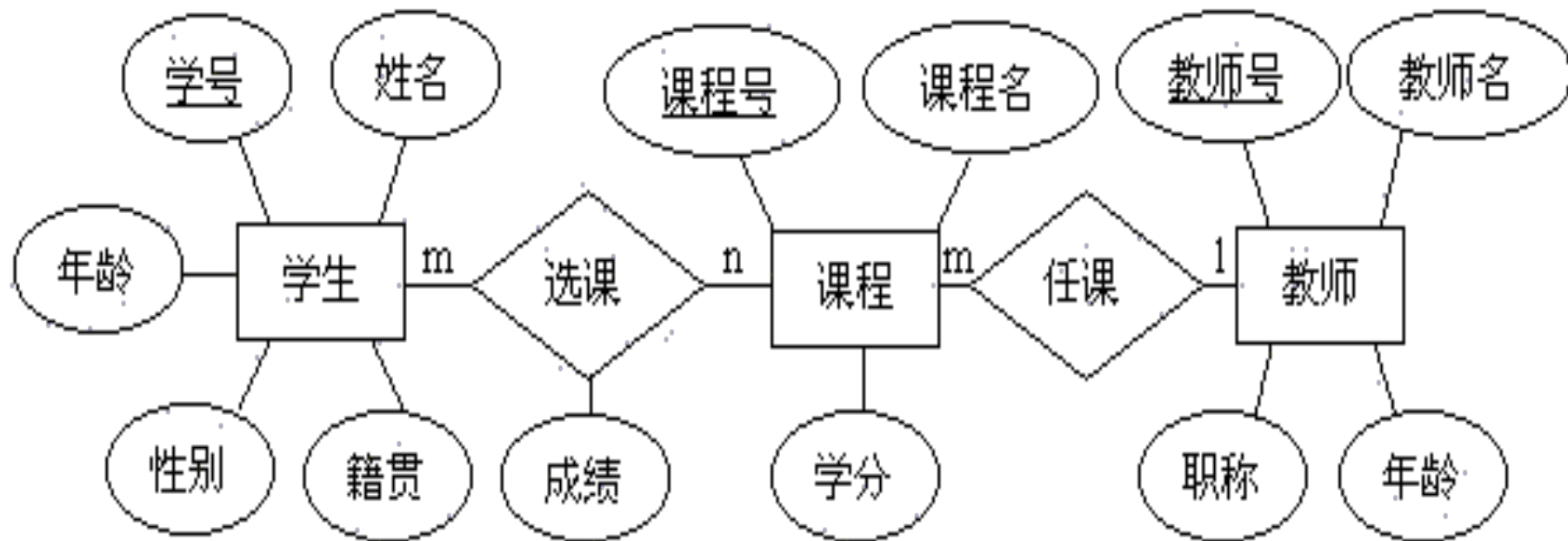
用**椭圆**表示**属性**

用**菱形**表示**实体间的联系**

属性和实体间、实体和联系间用**线段**连接(同时在线段边上标上联系的类型)

例1.2.1

业务场景 (business scenario)： 假设一个学生可选多门课程，而一门课程又有多个学生选修，一个教师可讲多门课程，一门课程至多只有一个教师讲授。



要求自学的扩展E-R模型（一定要学）

► **ISA** 联系—表达**子类父类**关系

- 分类属性
- 不相关约束
- 可重叠约束
- 完备性约束

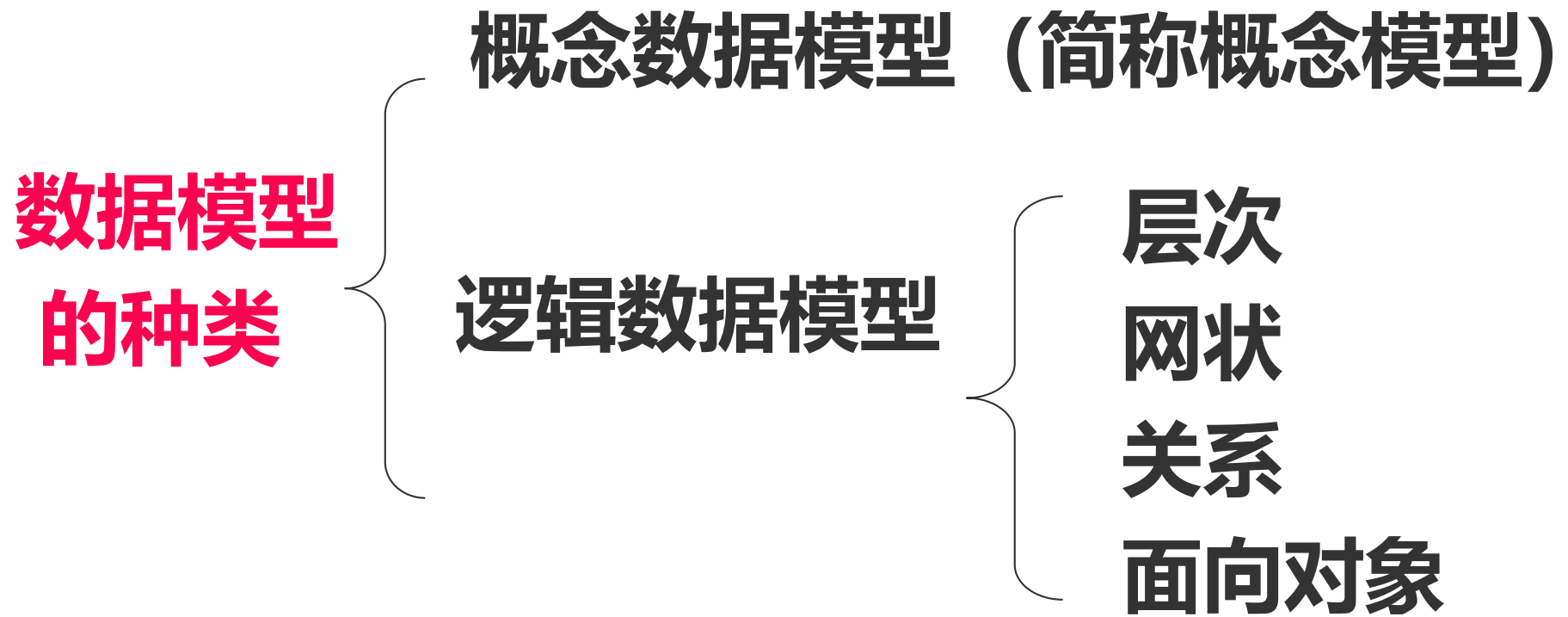
► **基数**约束

► **Part-of** 联系—表达**组成**关系

实体-联系模型的局限性

实体-联系模型是一种静态信息模型，只能反映当前状态，不能反映实体的变化过程。

(三) 逻辑数据模型



1. Logic Model 逻辑模型定义

► What's Logic?

- The relationship **between elements** and **between an element and the whole** in a set of objects, individuals, principles, or events

► Logical Data Model

- A logical design of a data

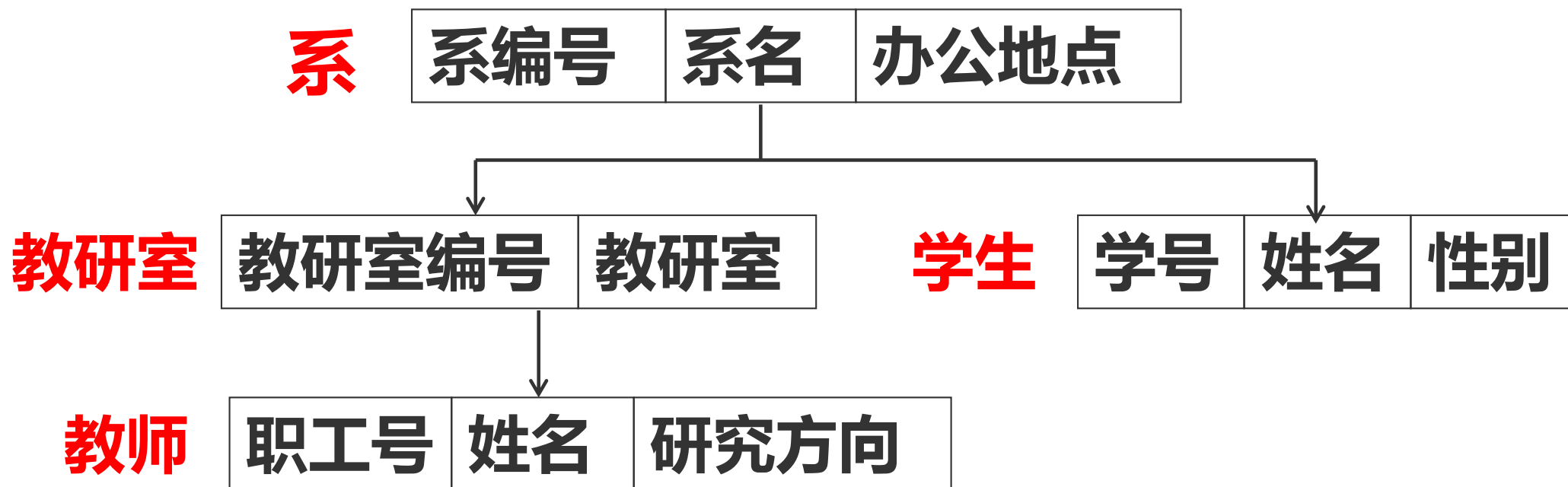
直接面向数据库的逻辑结构，它是现实世界的第二层抽象，这类模型不涉及具体物理平台，仅描述结构，涉及到计算机系统和数据库管理系统的类型。

Logical data model

- ▶ A **logical data model** or **logical schema** is a data model of a specific problem domain expressed **independent of** a particular database management product or storage technology (**physical data model**) **but in terms of data structures** such as relational tables and columns, object-oriented classes, or XML tags.
 - ▶ This is as opposed to a conceptual data model, which describes the semantics of an organization without reference to technology.
 - ▶ 概念模型表达语义，与**实现技术无关**
 - ▶ 逻辑跟**实现技术**有关，但与**物理平台**无关
-

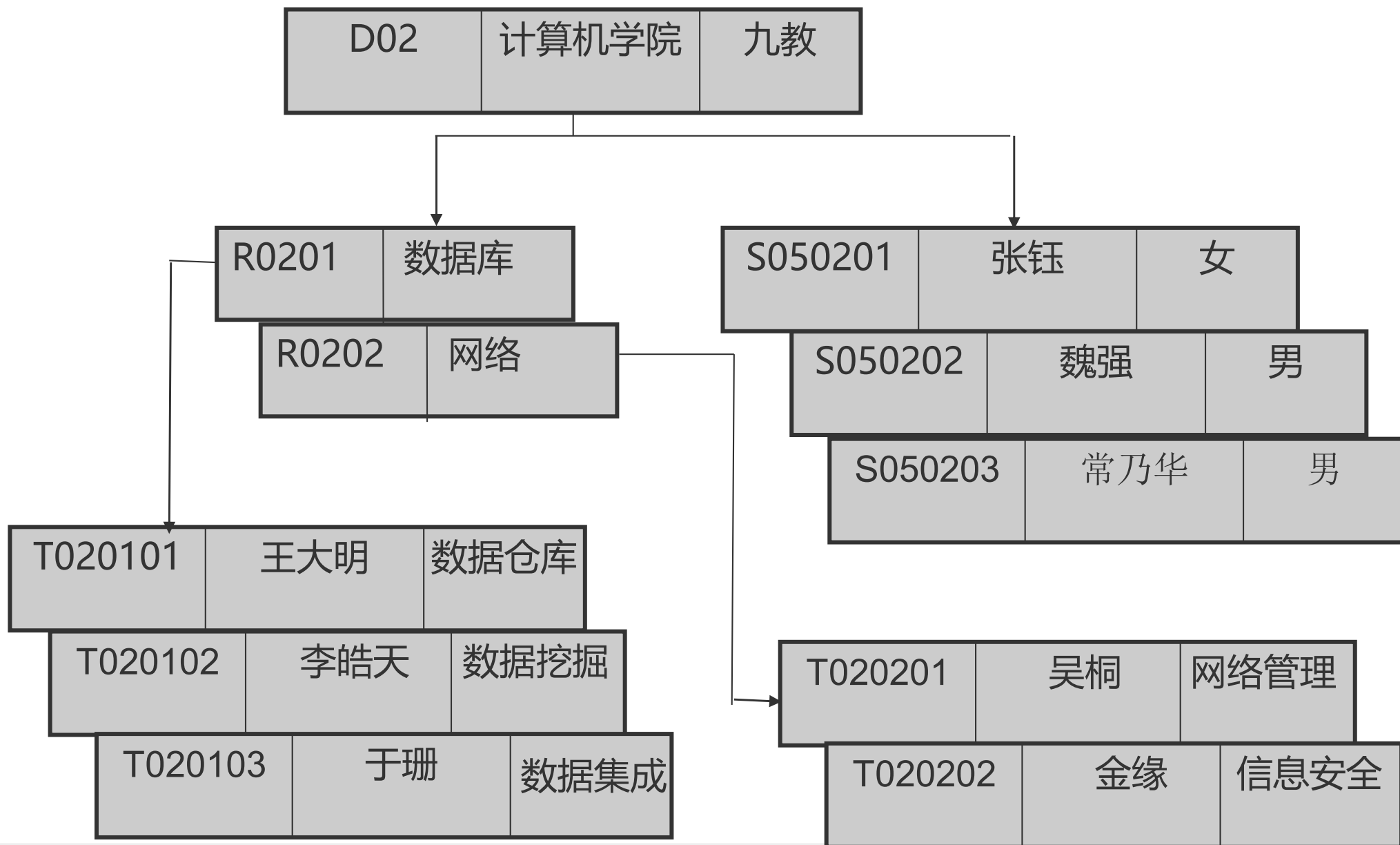
2. 层次数据模型

层次数据模型按树型结构组织数据，它是以记录类型为结点，以结点间联系为边的有序树



有且只有一个结点没有双亲结点,即**根结点**, 根以外的其他结点有且只有一个双亲结点

案例



层次模型的基本特点：

- 层次模型适合行政机构，家族关系等**一对多**关系的描述。如果要表示**多对多**的关系，必须**分解**（冗余结点法和虚拟结点法）成两个**一对多**的关系。
 - 任何一个给定的记录值只有**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脱离双亲记录值而独立存在。
-

冗余结点法

型

学生



课程

课程



学生

王梅



人工智能

数据库



张海

实例

张海



人工智能



数据库

人工智能

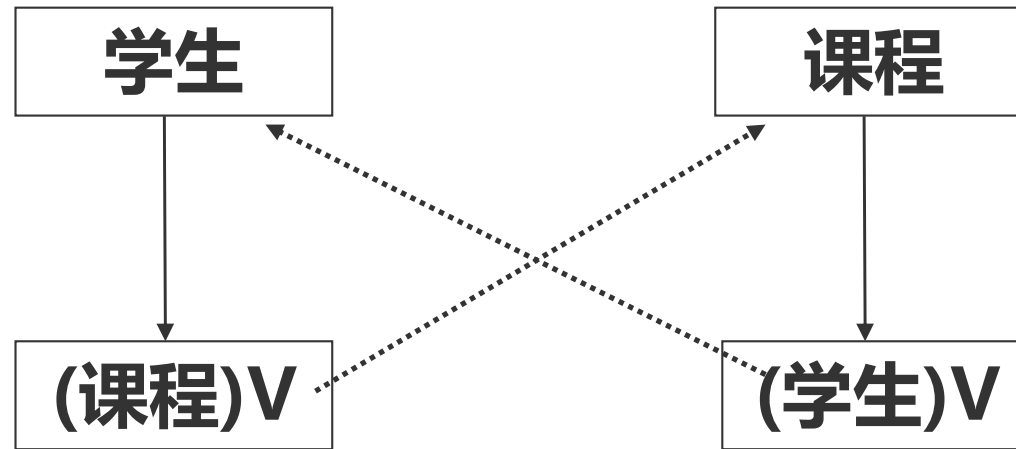


王梅



张海

虚拟结点法



层次模型的数据结构为 有序树或森林

▶ 层次模型的数据操纵

- 数据操作要**满足**模型的**完整性约束**条件

➤ 插入操作:

如果没有相应的双亲结点,不能插入子结点

➤ 删除操作:

如果删除双亲结点,则相应的子女结点也被同时删除.

➤ 修改操作:

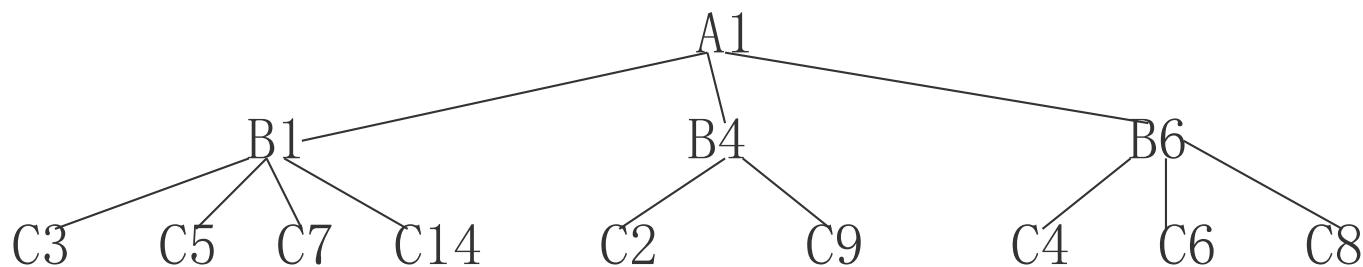
应修改所有相应的记录,以保证数据的一致性.

层次模型的存储结构

- ▶ 层次数据库需要存储两类内容：
 - 既要存储数据本身
 - 还要存储数据之间的层次联系.
 - ▶ 将数据和数据之间联系结合在一起存储
 - ▶ 有两种方法存储
 - 邻接法
 - 链接法
-

①邻接法

按层次树**前序遍历**顺序把所有记录依次邻接存放.

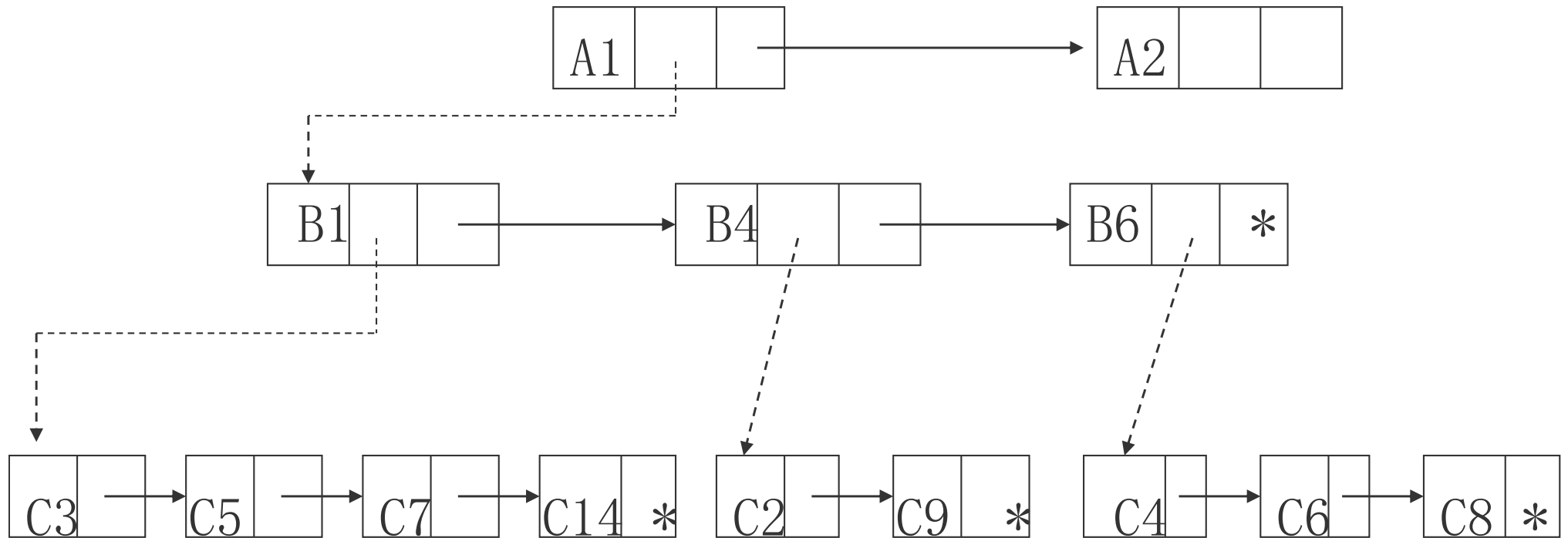


A1	B1	C3	C5	C7	C14	B4	C2	C9	B6	C4	C6	C8	A2	...
----	----	----	----	----	-----	----	----	----	----	----	----	----	----	-----

通过物理空间的位置相邻体现层次顺序

②链接法:用指针反映数据之间的层次联系

每个结点设**两个指针**,分别指向**最左边的子女**和**最近的兄弟**.



虚线表示父子关系,**实线**表示兄弟关系.

层次模型的优缺点

优点:

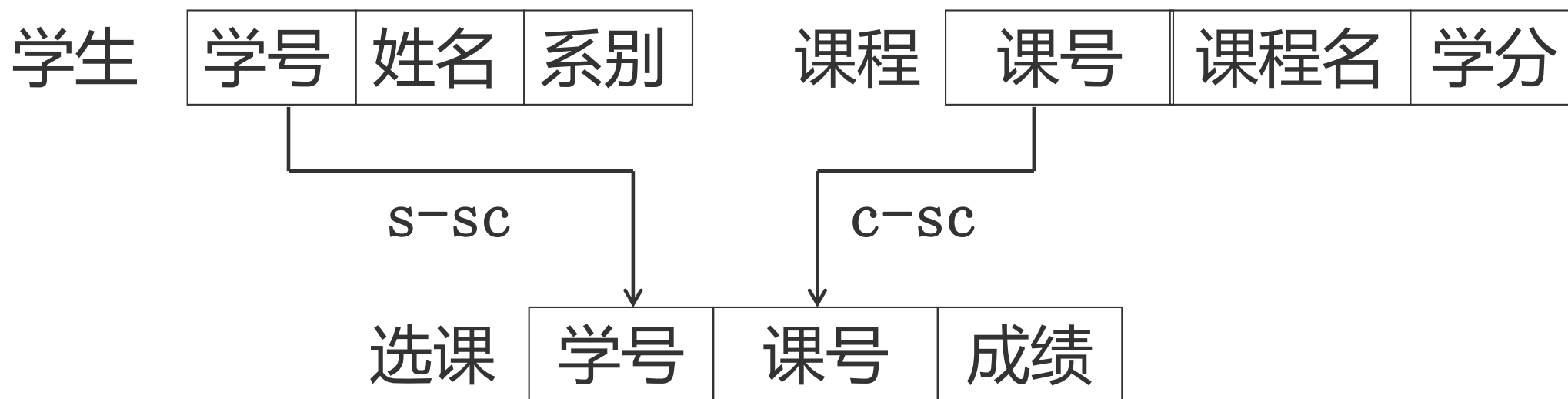
- (1)记录之间的联系通过指针实现，查询效率较高。
- (2)提供了良好的完整性支持

缺点:

- (1)只能表示记录间的一对多联系，而多对多的联系会出现数据冗余。
 - (2)结构严格、复杂，因此编程复杂
 - (3)对插入和删除操作限制比较多
 - (4)查询子女结点必须通过双亲结点.
-

3. 网状数据模型

用**有向图结构**表示实体及实体间联系的数据模型，它是以记录类型为结点，以结点间联系为边（适合于多对多关系）



网状数据模型必须满足两个条件:

- (1) 允许一个以上的结点无双亲结点
- (2) 一个结点可以有多于一个双亲结点

网状数据库之父：C.W.Bachman

1973年图灵奖获得者

The programmer as Navigator

主要贡献：

1.主持与开发了最早的网状数据库系统
IDS(1964)

2. 积极推动与促成了数据库标准的制定。1971年推出的**DBTG报告**首次确定了数据库的**三层体系结构**，是数据库历史上具有里程碑意义的文献



He received the [Turing Award](#) from the [Association for Computing Machinery \(ACM\)](#) in 1973 for "*his outstanding contributions to database technology*".

网状数据模型的优缺点

优点:

- 1、在两个结点之间可以有两种或多种联系。
- 2、记录之间的联系通过指针实现，M:N联系也容易实现，存取(查询)效率较高。

缺点:

结构比较复杂，编程复杂，程序员必须熟悉数据库的逻辑结构，不利于用户掌握。

层次与网状数据模型的最大缺点

- ▶ 缺点：联系通过指针实现
 - ▶ 带来的问题
 - 数据分布分散
 - 数据插入、修改操作麻烦
 - 要求程序员熟悉底层结构
 - 应用程序编写负担重
 - ▶ 需要有更好的解决策略：
 - 抛弃指针策略
 - 联系或关系用数据来体现
 - 数据存在，关系就得到体现
-

5. 关系数据模型

用**二维表**来表示实体集，用**外键**表示实体间的联系，这样的数据模型称为**关系数据模型**

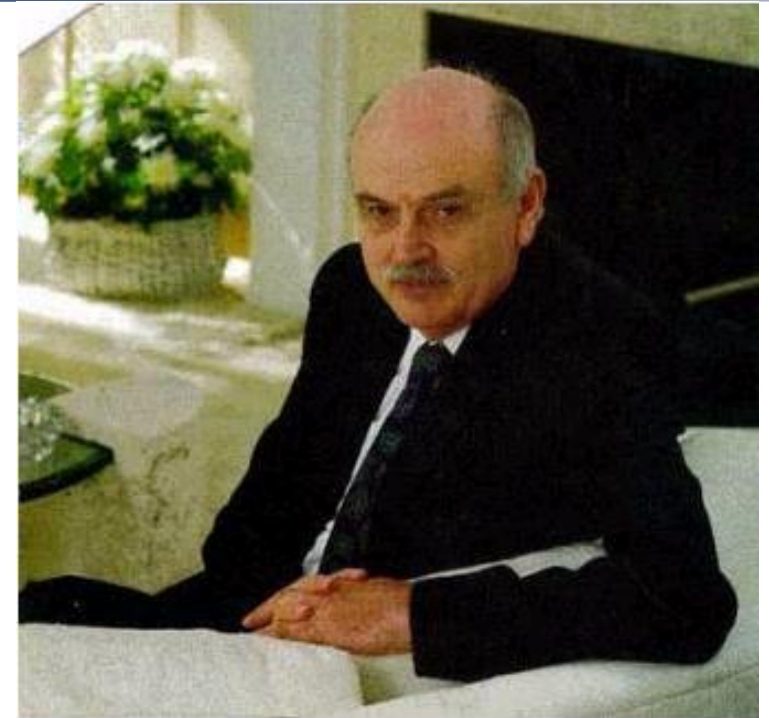
关系模型中的一些术语:

- **关系**: 一个关系对应通常说的一张表
 - **元组**: 表中的一行即为一个元组
 - **属性**: 表中的一列即为一个属性
 - **主码**: 表中某个属性组,可以唯一确定一个元组
 - **域**: 属性的取值范围
 - **分量**: 元组中的一个属性值
 - **关系模式**: 二维表的表头那一行
-

关系数据库之父：E.F.Codd

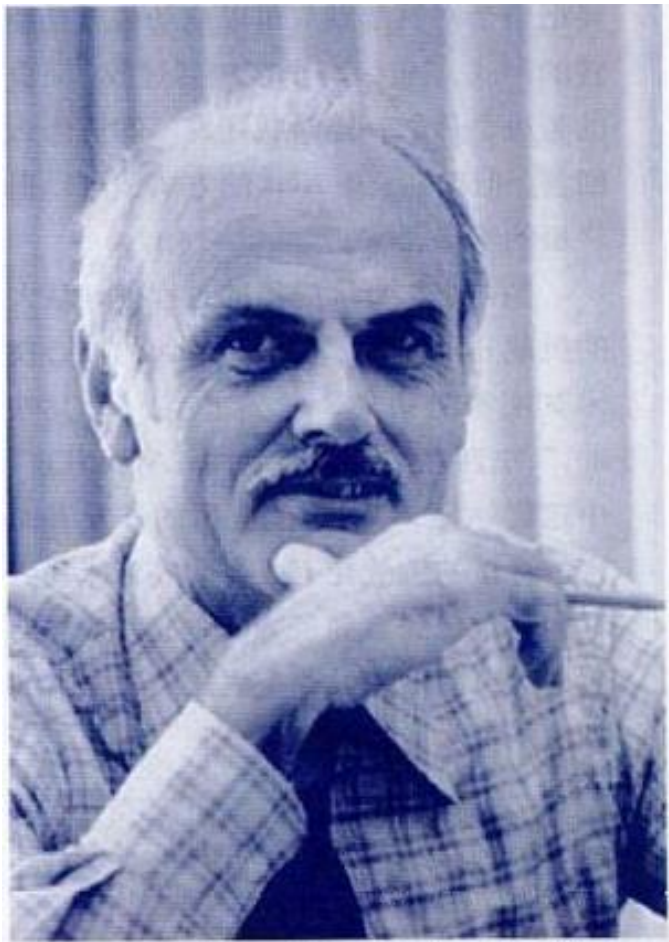
E. F. Codd 于1969年创立了著名的关系数据库模型理论：

《大型共享数据库数据的关系模型》 “A Relational Model of Data for Large Shared Data Banks,” E.F. Codd, Comm. ACM 13 (6), June 1970, pp. 377-387。



E.F.Codd从前的同事，IBM公司数据技术专家chris Date说：“在他找到Relational Model这个最佳的商用数据库技术前，整个市场一片混乱。”

关系数据库之父：E.F. Codd



► 考特勾画了关系数据库的蓝图，
同时也奠定了 SQL 的基础

1923-2003，英国人

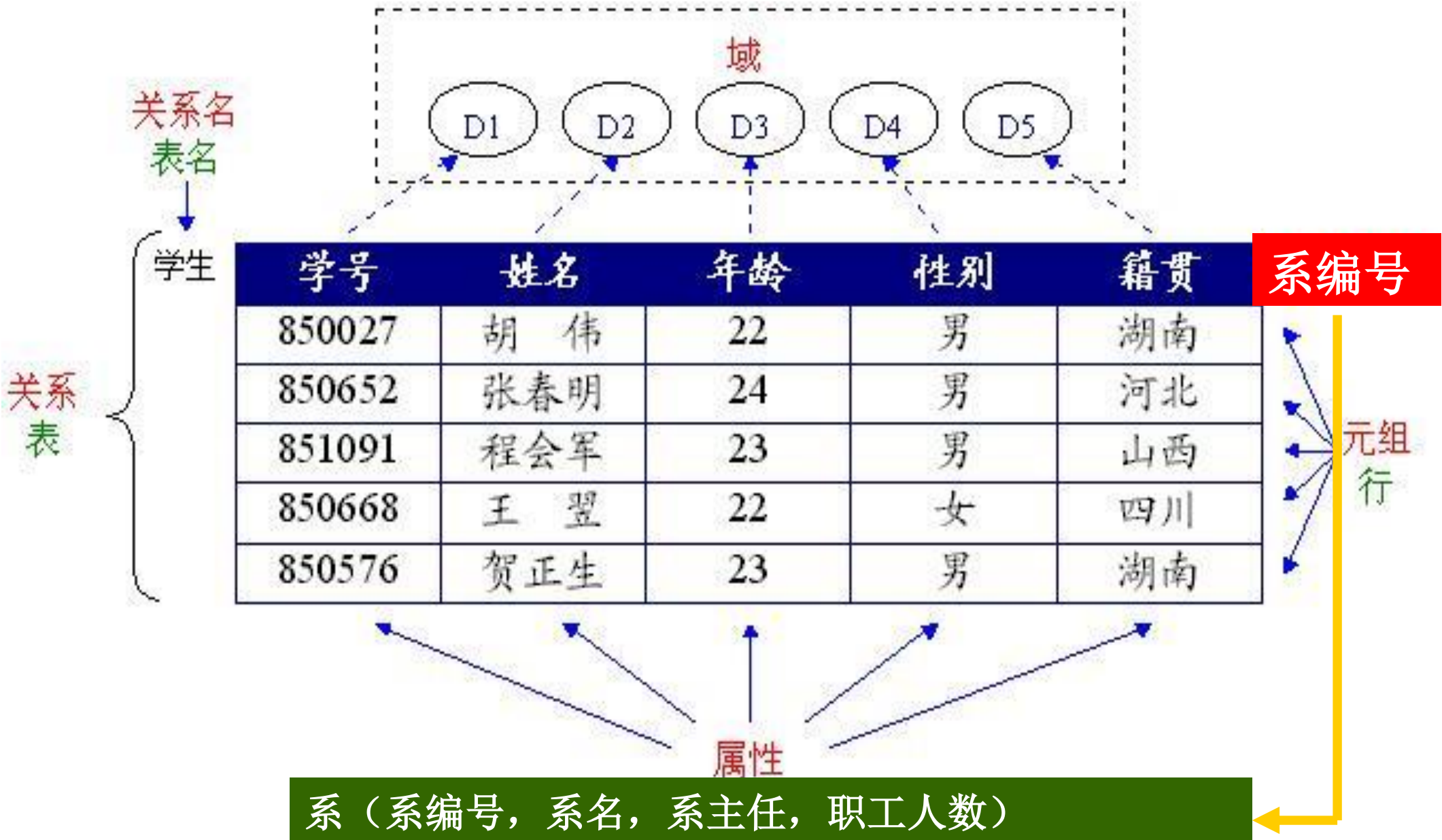
1981年图灵奖获得者

Rational Database: A **Practical**
Foundation for **Productivity**

美国计算机协会将他提出的关系型理论
评为半个世纪以来最具里程碑意义的25
篇论文之一

《今日美国》在他去世的第二天写道：
“考特，一位数学家、计算机科学家，
为我们开创了关系型数据库的全新技术，
为计算机信息化的发展铺平了道路，**开
创了一个伟大的产业。**

关系数据模型示例



关系数据模型

- △ 关系数据模型中数据的**逻辑结构**
是一张**二维表**
 - △ 关系模型的**数据操纵**
查询、插入、删除、修改
 - △ 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
 - △ 关系的**存储结构**
文件
-

关系型数据库的特点

△ **关系数据库**采用关系数据模型作为数据的组织方式

△ **关系的特点**

- (1) 一个关系(表)中不存在两个元组(行)在各个分量 (或属性列) 上完全相同
 - (2) 行的次序无关;
 - (3) 列的次序无关;
 - (4) 每个分量必须是不可分的量
-

不符合分量不可分的案例

职工号	姓名	工资		扣款		实发
		基本	职务	公积金	税	
00001	王一	1000	1000	200	100	1700

Oracle：第一个基于关系型技术的数据库产品

► 《时代周刊》

- 考特在70年代初的技术理论《关于大型共享数据库的关系数据模型》为整个信息技术产业提供了一个全新的发展模式
- 幸运的甲骨文率先选择了E.F. CODD开创的关系型数据管理 (Relational Model) 技术，从而在数据库市场中遥遥领先。

► 甲骨文

- 抢先采用了**关系型数据库技术**作为其产品的核心
- 并于1977年首次推出了基于关系型技术的数据库产品。
- 几年后，甲骨文的产品在企业软件市场中大获成功，把包括IBM在内的众多公司甩在了后面。

► Codd coined the term OLAP: Online Analytical Processing

关系数据模型的优缺点

优点:

关系模型建立在**严格的数学概念**基础上。

关系模型概念单一，无论**实体**还是实体之间的**联系**都用**关系表示**，**检索结果也是关系**。结构简单、清晰，用户易懂易用。

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**。

缺点:

由于存取路径对用户透明，**查询效率**往往不如非关系数据模型高。

关系数据模型和层次、网状数据模型的最大差别：

用关键码**而不是用**指针**导航数据**

各有什么优缺点？

第四节 数据库系统结构

数据库系统**内部**的架构：

实现三级模式结构

提供两级映象功能

数据库系统模式的概念

型(Type): 是指对某一类数据的**结构**和**属性**的说明

值(Value): 是型的一个具体赋值

模式(Schema): 是数据库中全体数据的**逻辑结构**和**特征**的描述。模式反映的是**数据的结构及其联系**。

实例(Instance): 是模式的一个具体值。反映的是数据库某一时刻的状态。同一模式有很多实例。

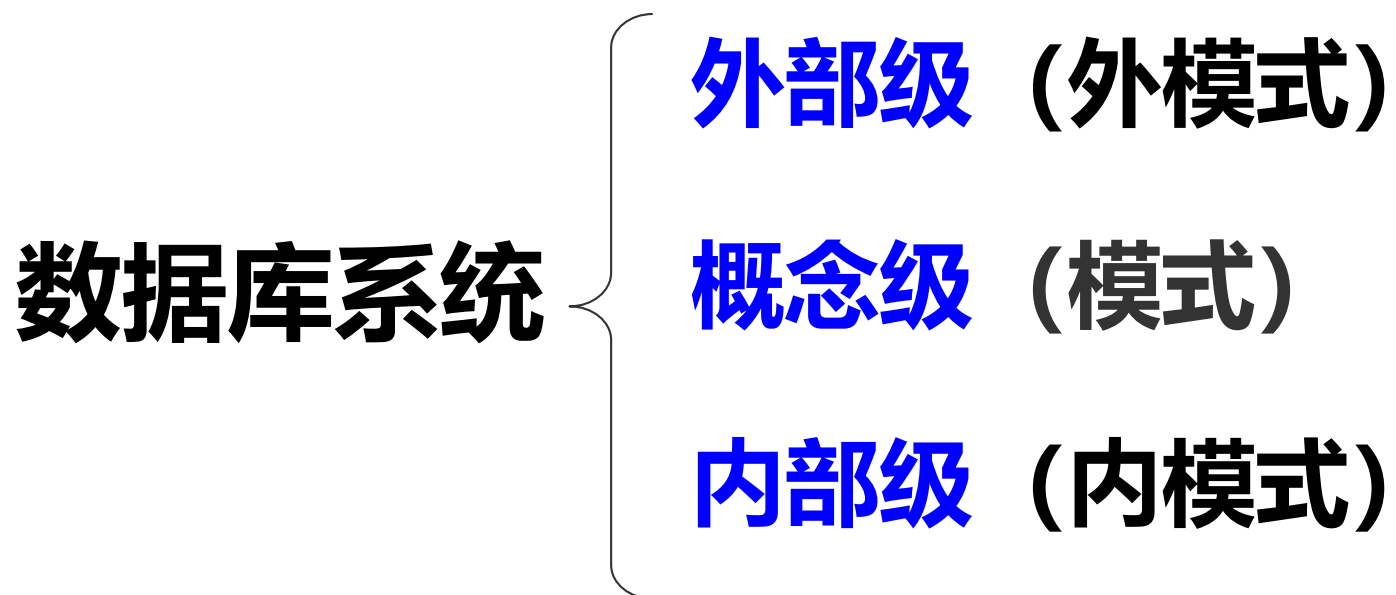
数据库系统模式的概念

数据库系统的种类很多，但架构上有共同特征

在 { 数据模型
数据库语言
操作系统
存储结构 } 不同

采用三级模式结构，
提供两级映象功能

1.数据库系统的三级模式结构



外模式（用户模式）

最接近于用户的一级，是用户与DBS的接口，是用户看到和使用的局部数据的逻辑结构和特征的描述，又称用户视图。与某一应用有关的数据的逻辑表示。

外模式（用户模式）

- △ 用户的所有操作都是针对用户视图进行。
 - △ 外模式又称为“子模式”或“用户模式”，是模式的子集。
 - △ 一个数据库可以有多个外模式
 - △ 每个用户必须使用一个外模式，一个外模式可以有数个不同的用户使用，一个应用程序只能使用一个外模式。
-

外模式（用户模式）

优点：

- (1) 接口简单、使用方便**
 - (2) 保证数据的独立性**
 - (3) 增强了数据的安全保密性（每个用户只能看到对应外模式中的数据，其余数据不可见）**
-

模式（逻辑模式）

介于用户级和物理级之间，是所有用户的公共数据视图（**全局的数据视图**），是数据库管理员看到和使用的数据库。

模式**不涉及**到存储结构、硬件环境、访问技术等细节，也**不涉及**应用程序、开发工具、程序设计语言等。

模式（逻辑模式）

△ 模式 是数据库中全部数据的**逻辑结构**和**特征**的描述，不仅要**描述**概念**记录类型**，还要**描述**记录间的**联系**、**操作**、数据的**完整性和安全性**等。

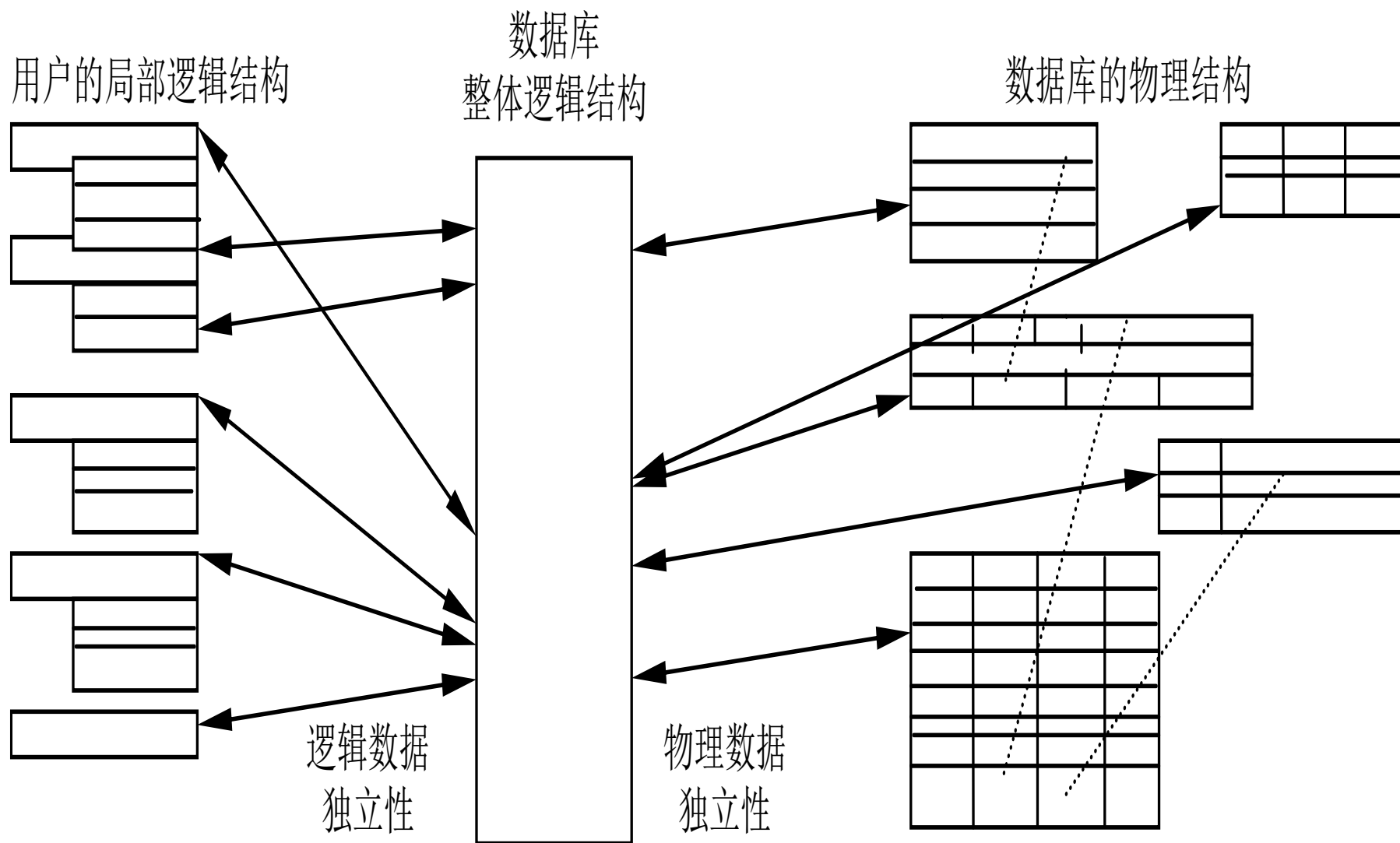
△ 一个数据库可有多个不同的用户视图，每个用户视图由数据库某一部分数据的抽象表示所组成。

△ 一个数据库只存在一个DBA视图，即**一个数据库只有一个模式**。

内模式（存储模式）

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。定义所有的内部记录类型、索引和文件的组织方式，以及数据控制方面的细节。
一个数据库只有一个内模式。

三级模式、二级映象



2.两级映象

数据的组织由DBMS管理，用户**抽象地**处理数据，而不关心数据在计算机中的表示和存储方式。为了实现三层之间的联系和转换，数据库管理系统在三层之间提供了两级映象。

(1)模式/内模式映象

存在于模式和内模式之间

(2)外模式/模式映象

存在于外模式和模式之间

两级映象保证数据库系统的数据具有较高的**逻辑独立性**和**物理独立性**。

(1)模式/内模式映象

在模式和内模式之间存在模式/内模式映象，用于定义模式和内模式间的对应性。定义了数据库全局逻辑结构与存储结构之间的对应关系

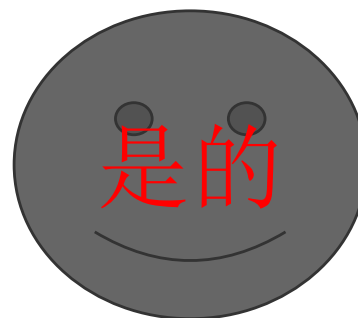
数据库只有一个模式，也只有一个内模式，所以模式/内模式映象是唯一的。

模式/内模式映象一般由系统内部的模式描述体现。

(2)外模式/模式映像

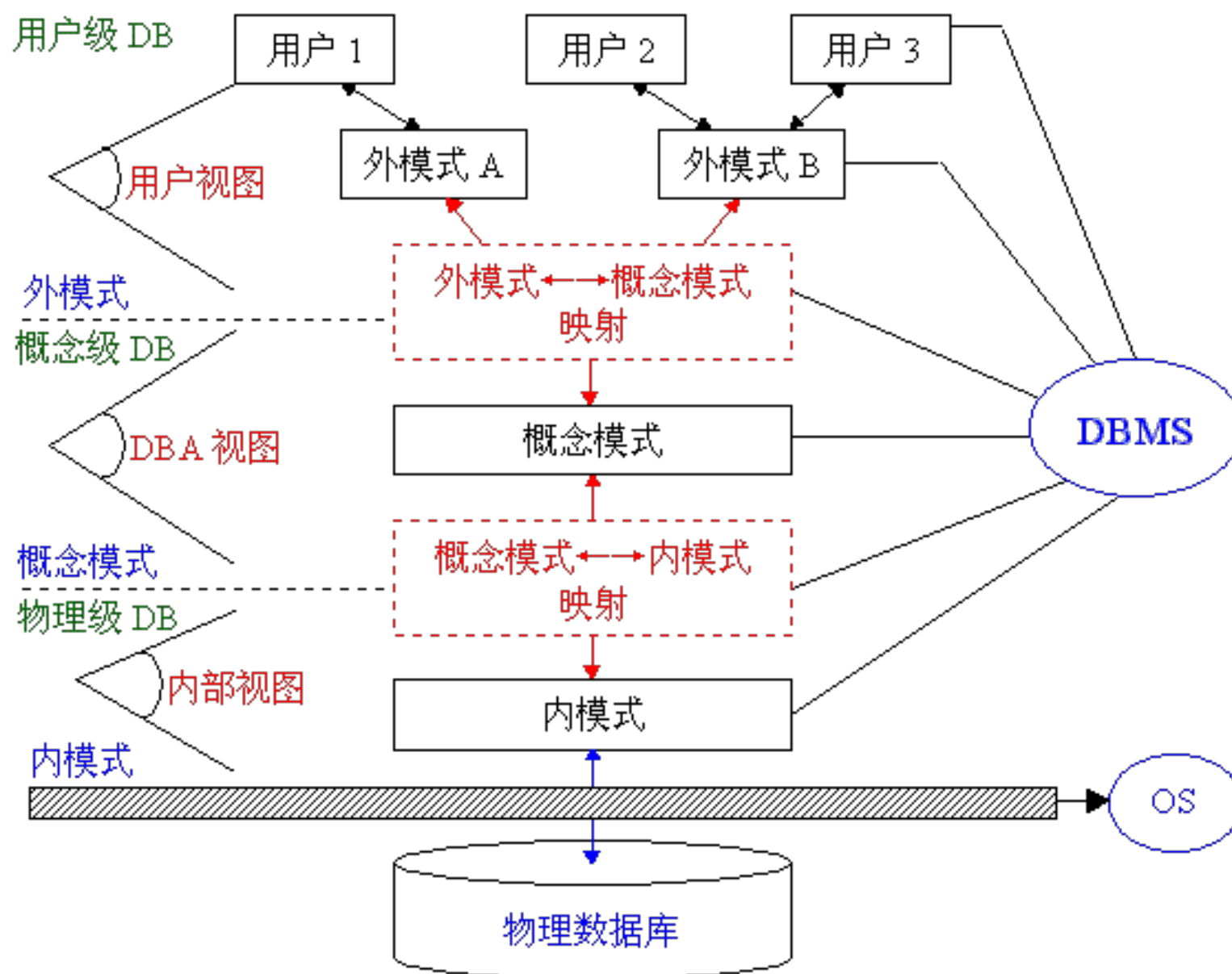
在外模式和模式之间存在外模式/模式**映像**，用于**定义外模式和模式间的对应关系**。

每一个外模式，数据库系统都有一个外模式/
模式映像吗？



外模式/模式映像一般是在**外模式**中描述的。

(2)外模式/模式映射



3. 数据库系统两级独立性

三个抽象级间通过两级映象（根据一定的对应规则）进行相互转换，使得数据库的三级形成一个统一整体，保证了数据的独立性。

什么是数据独立性？

是指与数据间相互独立，不受影响。

分为两级：逻辑独立性、物理独立性

(1) 物理独立性

如果数据库的内模式（**存储结构**）要进行**修改**那么DBA对**模式/内模式映象**也要进行相应的**修改**，从而使**模式**尽可能保持**不变**。

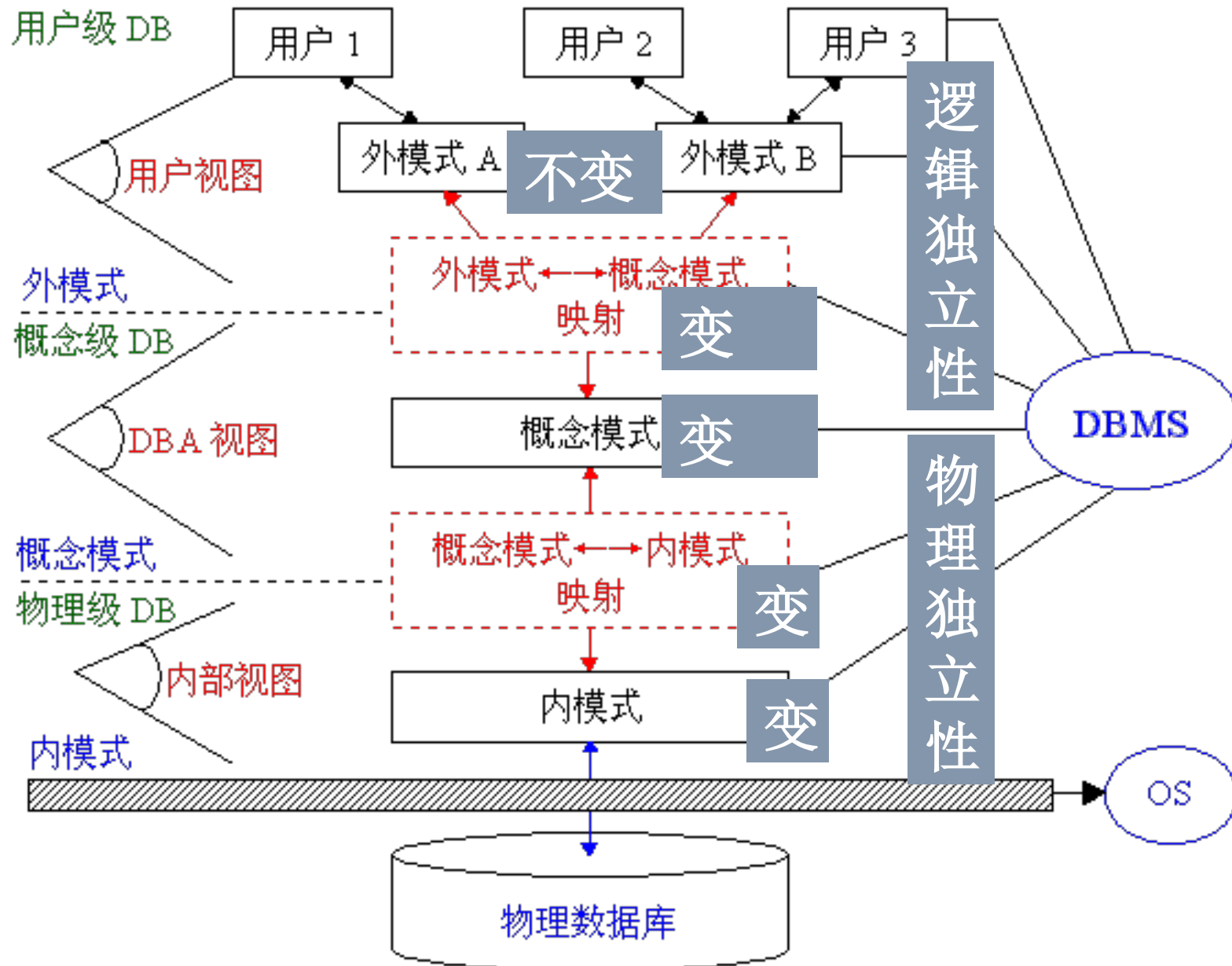
对内模式的修改尽量不影响模式，当然，对于外模式和应用程序的影响更小，**应用程序不必修改**，保证**数据与程序的物理独立性**。称为数据库的物理独立性。

(2) 逻辑独立性

如果数据库的**模式**要进行**修改**，那么DBA对各个**外模式/模式映象**也要进行相应的**修改**，使**外模式**尽可能保持**不变**。

对模式的修改尽量不影响外模式，从而**应用程序不必修改**，这样保证数据与程序的**逻辑独立性**。称为数据库的逻辑独立性。

(3) 逻辑独立性与物理独立性关系



(3) 逻辑独立性与物理独立性关系

数据库**模式**依赖于全局逻辑结构，**独立于外模式**，**独立于存储设备**。

数据库**外模式**面向具体应用程序，**独立于存储模式**和存储设备。**应用程序****依赖于**特定的**外模式**，与数据库的模式和存储结构独立。

二级映象保证数据库**外模式的稳定性**，从而保证**应用程序的稳定性**。

数据与程序之间的独立性，使得数据的定义和描述从程序中分离出去，数据的存取由DBMS管理，**简化了**程序的编制，**减少**程序的维护和修改。

第五节 数据库系统的组成

DBS的组成：数据库、人员、硬件、软件。

(1)数据库

物理数据库：应用数据的集合，是DB的主体。

描述数据库：关于各级数据结构的描述。

数据库系统的组成

(2)硬件：包括CPU、内存、外存、I/O设备、数据通道等设备

(3)软件：DBMS、OS、各种与数据库接口的高级语言及编译系统、应用开发工具、数据库应用系统。

(4)人员：数据库管理员、系统分析员、数据库设计人员、应用程序员、最终用户。

数据库系统的组成

(a) 数据库管理员 (DBA) 的职责为:

- ① **模式定义**, 决定数据库中信息内容和结构
 - ② **内模式定义**, 决定数据库的存储结构和存取策略
 - ③ 根据要求**修改**数据库的**模式和内模式**
 - ④ 对数据库访问权限的**定义**, 保证数据**安全性**
 - ⑤ **完整性约束条件的说明**
 - ⑥ **监控**数据库的使用和**运行** (处理出现的问题)
 - ⑦ 数据库的**改进和重组重构** (改进数据库设计)
-

数据库系统的组成

(b) 系统分析员的责任是

负责应用系统的需求分析和规范说明，确定系统的硬件软件配置，参与数据库系统的概要设计。

(c) 数据库设计人员的责任是

负责数据库中数据的确定，各级模式的设计，参加用户需求调查和系统分析，进行数据库设计

(d) 应用程序员的责任是

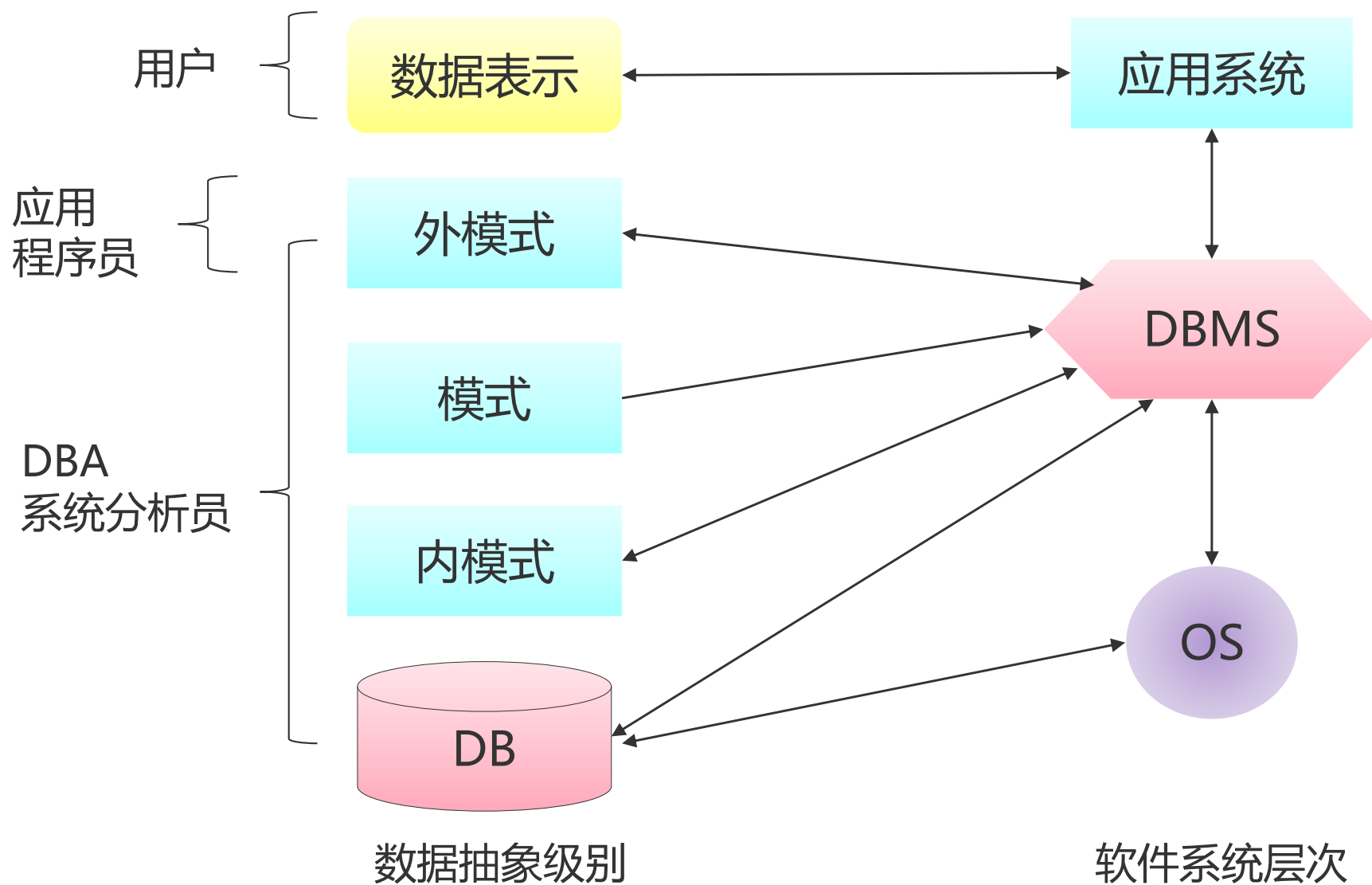
负责设计和编写应用系统的程序模块、调试和安装。

数据库系统的组成

(e) 用户—最终用户

最终用户通过应用系统的用户接口使用数据库。常用的接口方式有浏览器、菜单驱动、表格操作、图形显示等，以简明直观的表示方式显示数据。

各种人员的数据视图



各种人员的数据视图

第六节： 数据库管理系统

数据库管理系统（DBMS）是指数据库系统中管理数据的软件系统，是DBS的核心组成部分。对数据库的一切操作（定义、查询、更新及各种控制），都是通过DBMS进行的。

DBMS总是基于某种数据模型，根据数据模型的不同，DBMS可以分成层次型、网状型、关系型、面向对象型、对象关系型等。

1. DBMS的主要功能

(1)数据库的定义功能

DBMS提供数据定义语言 (DDL)**定义数据库的三级结构**，包括外模式、模式、内模式及相互之间的映象，定义数据的**完整性、安全控制**等约束。

数据字典管理功能

数据字典 (DD)

数据库系统中存放三级结构定义的数据字典称为数据字典。

操纵功能

(2)数据库的操纵功能（数据库存取）

DBMS提供数据操纵语言（DML）实现对数据库中数据的操作，基本的数据操作



保护功能

(3)数据库的保护功能（数据库运行处理）

DBMS对数据库的保护主要通过四个方面实现：

- ①数据库的恢复**
 - ②数据库的并发控制**
 - ③数据库的完整性控制**
 - ④数据库的安全性控制**
-

存储功能

(4)数据库的存储管理

DBMS的一个目标应该是简化和促进数据的访问。DBMS的**存储管理子系统**提供了数据库中数据和应用程序的一个界面，与磁盘中数据打交道的是OS中的文件系统。DBMS**存储管理子系统的职责**是把各种**DML语句转换成低层的文件系统命令**，起到数据的存储、检索和更新的作用。

维护功能

(5)数据库的维护功能

DBMS还有许多实用程序提供给DBA运行DBS时使用。这些实用程序起着数据库维护的功能。主要的实用程序有四个：

- ①数据**装载**程序
 - ②**备份**程序
 - ③文件**重组**程序
 - ④性能**监控**程序
-

2.DBMS的系统结构

► 最上层：接口层

DBMS一般都按不同用户的需要提供多种用户接口，如交互式SQL接口、嵌入式SQL接口、自然语言查询接口等。用这些语言书写的应用程序经接口软件处理后，抽出其中数据库语言语句，转换成一种最基本的数据库语言，如SQL，交语言处理层处理。

► 第二层：语言处理层

语言处理层对数据库语言的各类语句进行语法分析、视图转换、安全性检查、完整性检查、查询优化等，通过对下层基本模块的调用，生成可执行代码。

DBMS的系统结构

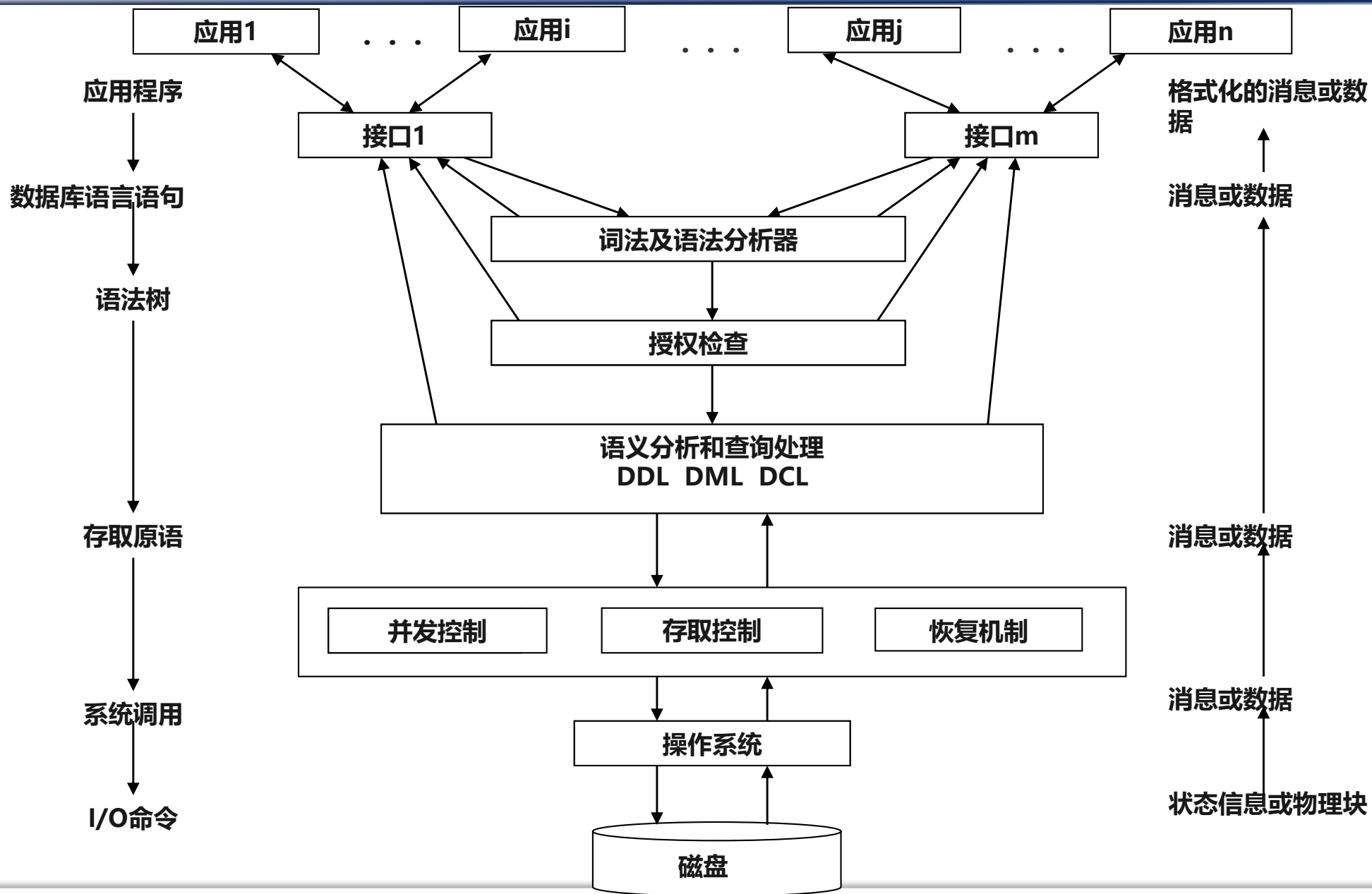
► 第三层：数据存取层

该层处理的对象是单个元组，它将上层的集合操作转换为单记录操作，执行扫描、排序、元组的查找、插入、修改、删除、封锁等基本操作，完成数据记录的存取、存取路径维护、事务管理、并发控制和恢复等工作。

► 第四层：数据存储层

该层处理的对象是数据页和系统缓冲区，执行文件的逻辑打开、关闭、读页、写页、缓冲区读和写、页面淘汰等操作，完成缓冲区管理、内外存交换、外存的数据管理等功能。

RDBMS的系统结构



3. 用户访问数据的过程

- ▶ (1) 用户通过应用程序向RDBMS发出调用数据库数据的命令，如SELECT命令，命令中给出一个关系名和查找条件。
 - ▶ (2) RDBMS先对命令进行语法检查，检查通过后进行语义检查和用户存取权限检查。具体做法是：RDBMS读取数据字典，检查是否存在该关系及相应字段，该用户能否读取它们等。确认语义正确、存取权限合法后便决定执行该命令，否则拒绝执行。
 - ▶ (3) RDBMS执行查询优化。优化器要依据数据字典中的信息进行优化，并把该命令转换成一串单记录的存取操作序列。
-

用户访问数据的过程

- ▶ (4) RDBMS执行存取操作序列（反复执行以下各步，直至结束）。
 - ▶ (5) RDBMS首先在缓冲区中查找记录，若找到满足条件的记录则转(10)，否则转(6)。
 - ▶ (6) RDBMS查看存储模式，决定从哪个文件、用什么方式读取哪个物理记录。
 - ▶ (7) RDBMS根据(6)的结果，向操作系统发出读取记录的命令。
 - ▶ (8) 操作系统执行读数据的有关操作。
-

用户访问数据的过程

- ▶ (9) 操作系统将数据从数据库的存储区送至系统缓冲区。
 - ▶ (10) RDBMS根据查询命令和数据字典的内容导出用户所要读取的记录格式。
 - ▶ (11) RDBMS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。
 - ▶ (12) RDBMS将执行状态信息，如成功读取或不成功的错误指示，例外状态信息等返回给应用程序A。
-

本章小结

重要概念

- ◆ **DB:** 数据库 (Database), DB是统一管理的相关数据的集合。DB能为各种用户共享, 具有最小冗余度, 数据间联系密切, 而又有较高的数据独立性。
 - ◆ **DBMS:** 数据库管理系统 (Database Management System), DBMS是位于用户与操作系统之间的一层数据管理软件, 为用户或应用程序提供访问DB的方法, 包括DB的建立、查询、更新及各种数据控制。DBMS总是基于某种数据模型, 可以分为层次型、网状型、关系型、面向对象型DBMS。
-

本章小结

- ◆ **DBS:** 数据库系统 (Database System), DBS是实现有组织地、动态地存储大量关联数据, 方便多用户访问的计算机软件、硬件和数据资源组成的系统, 即采用了数据库技术的计算机系统。
 - ◆ **1: 1联系:** 如果实体集E1中的每个实体最多只能和实体集E2中的一个实体有联系, 反之亦然, 那么实体集E1对E2的联系称为“一对一联系”, 记为“1: 1”。
-

本章小结

- ◆ **1: N联系**: 如果实体集E1中每个实体与实体集E2中任意个（零个或多个）实体有联系，而E2中每个实体至多和E1中的一个实体有联系，那么E1对E2的联系是“一对多联系”，记为“1: N”。
 - ◆ **M: N联系**: 如果实体集E1中每个实体与实体集E2中任意个（零个或多个）实体有联系，反之亦然，那么E1对E2的联系是“多对多联系”，记为“M: N”。
-

本章小结

- ◆ **数据模型**：表示实体类型及实体类型间联系的模型称为“数据模型”。它可分为两种类型：**概念数据模型**和**结构数据模型**。
 - ◆ **概念数据模型**：它是独门于计算机系统的模型，完全不涉及信息在系统中的表示，只是用来描述某个特定组织所关心的信息结构。
 - ◆ **结构数据模型**：它是直接面向数据库的逻辑结构，是现实世界的第二层抽象。这类模型涉及到计算机系统和数据库管理系统，所以称为“结构数据模型”。结构数据模型应包含：**数据结构、数据操作、数据完整性约束三部分**。它主要有：**层次、网状、关系、面向对象四种模型**。
-

本章小结

- ◆ **层次模型**：用树型结构表示实体间联系的数据模型
 - ◆ **网状模型**：用有向图结构表示实体类型及实体间联系的数据模型。
 - ◆ **关系模型**：是由若干个关系模式组成的集合，其主要特征是用二维表格结构表达实体集，用外键表示实体间联系。
-

本章小结

- ◆ **模式**：是数据库中全部数据的整体逻辑结构的**描述**。它由若干个概念记录类型组成。模式不仅要**描述**概念记录类型，还要描述记录间的联系、操作、数据的完整性、安全性等要求。
 - ◆ **外模式**：是用户与数据库系统的接口，是用户用到的那部分数据的**描述**。
 - ◆ **内模式**：是数据库在物理存储方面的**描述**，定义所有的内部记录类型、索引和文件的组成方式，以及数据控制方面的细节。
-

本章小结

- ◆ **模式/内模式映象**：这个映象存在于概念级和内部级之间，用于定义模式和内模式间的对应性，即**概念记录**和**内部记录**间的对应性。此映象一般在模式中描述。一个数据库只有一个模式/内模式映象。
 - ◆ **外模式/模式映象**：这个映象存在于外部级和概念级之间，用于定义外模式和模式间的对应性，即**外部记录**和**概念记录**间的对应性。此映象都是在外模式中描述。每一个外模式有一个外模式/模式映象。
-

本章小结

- ◆ **数据独立性**：在数据库技术中，数据独立性是指应用程序和数据之间相互独立，不受影响。数据独立性分成**物理数据独立性**和**逻辑数据独立性**两级。
 - ◆ **物理数据独立性**：如果数据库的内模式要进行修改，即数据库的存储设备和存储方法有所变化，那么模式/内模式映象也要进行相应的修改，使概念模式尽可能保持不变。也就是对内模式的修改尽量不影响模式。
 - ◆ **逻辑数据独立性**：如果数据库的模式要进行修改（如增加记录类型或增加数据项），那么外模式/模式映象也要进行相应的修改，使外模式尽可能保持不变。也就是对模式的修改尽量不影响外模式和应用程序。
-

本章小结

- ◆ **宿主语言**：编写应用程序的语言（即高级程序设计语言）在数据库技术中称为宿主语言（host language),简称主语言。
 - ◆ **DDL**：数据定义语言（Data Definition Language),用于定义数据库的**三级结构**，包括外模式、模式、内模式及其相互之间的**映象**，定义数据的**完整性约束**等。
 - ◆ **DML**：数据操纵语言（Data Manipulation Language),用于让用户或程序员使用，实现对数据库中数据的操作。基本的数据操作分成**两类四种**：检索（查询）和更新（插入、删除、修改）。DML分成**交互型DML**和**嵌入型DML**两类。依据语言的级别，DML又可分成**过程性DML**和**非过程性DML**两种。
-

本章小结

- ◆ **交互型DML**: 这类DML自成系统，可在终端上直接对数据库进行操作。
 - ◆ **嵌入型DML**: 这类DML是嵌入在主语言中使用。此时主语言是经过扩充能处理DML语句的语言。
 - ◆ **过程性DML**: 用户编程时，不仅需要指出“做什么”（需要什么样的数据），还需要指出“怎么做”（怎么获得数据）。层状、网状的DML属于过程性语言。
 - ◆ **非过程性DML**: 用户编程时，只需要指出“做什么”，不需要指出“怎么做”。关系型DML属于非过程性语言。
-

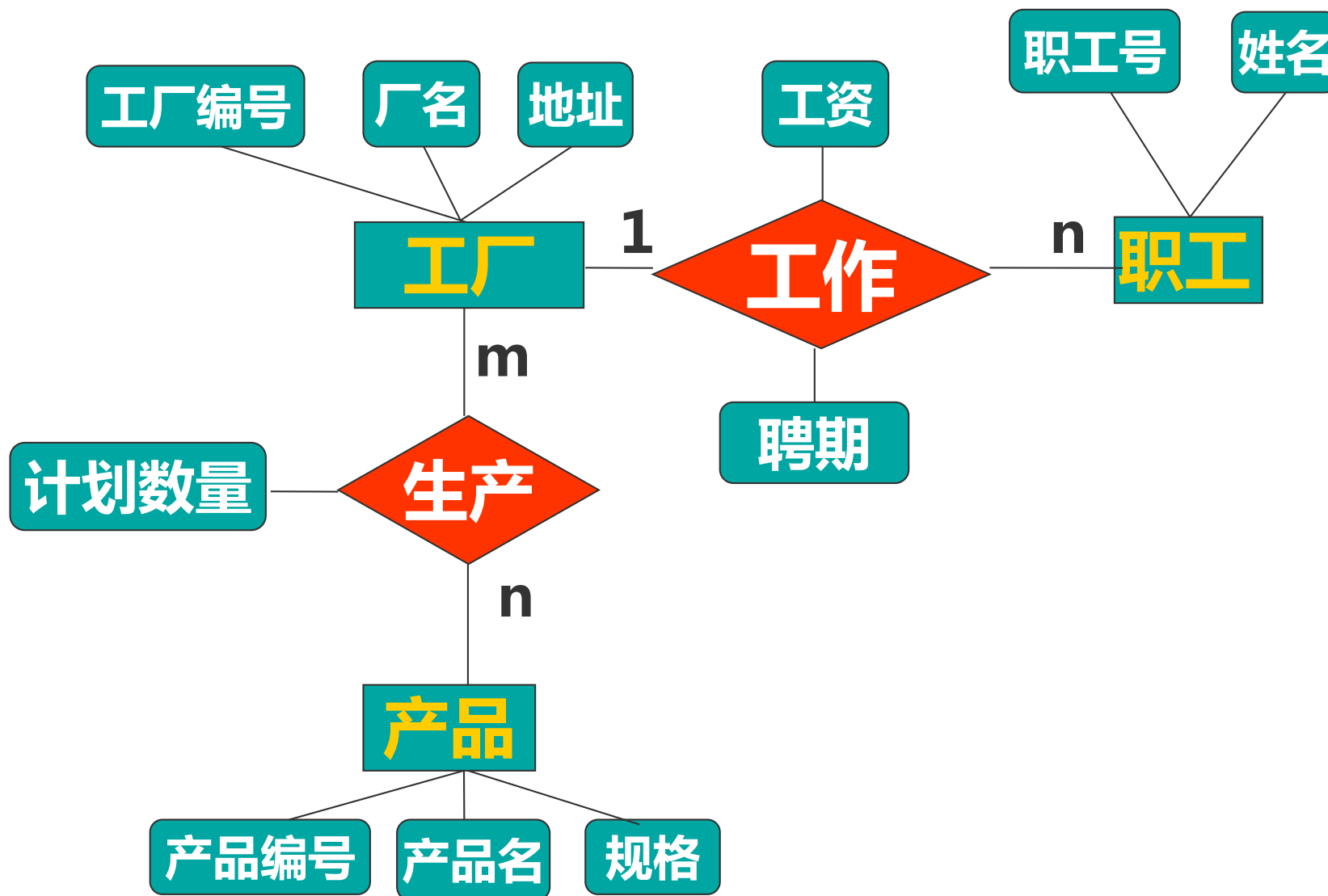
本章小结

- ◆ **DD**: 数据字典 (Data Dictionary), 数据库系统中存放三级结构定义的数据库称为数据字典。 (通常DD还存放数据库运行时的统计信息)
 - ◆ **DD系统**: 管理DD的实用程序称为 “DD系统”。
-

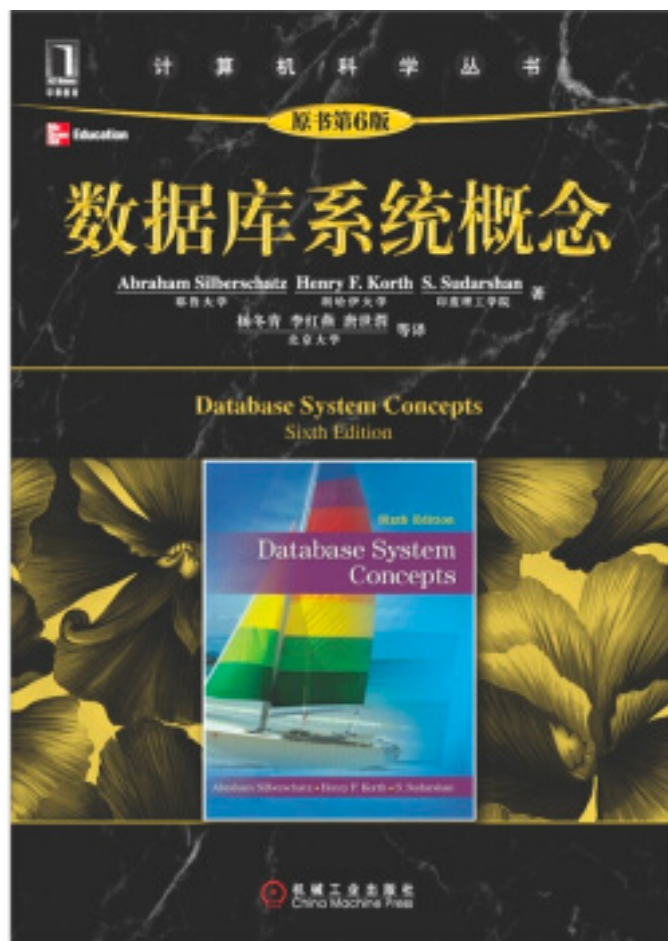
练习——ER图

- ▶ 某企业集团有若干工厂，每个工厂生产多种产品，且每一种产品可以在多个工厂生产，每个工厂按照固定的计划数量生产产品；每个工厂聘用多名职工，且每名职工只能在一个工厂工作，工厂聘用职工有聘期和工资。工厂的属性有工厂编号、厂名、地址，产品的属性有产品编号、产品名、规格，职工的属性有职工号、姓名。
-

练习——ER图



推荐教材



Architecture of a Database System

<https://dsf.berkeley.edu/papers/fntdb07-architecture.pdf>