

# 上堂课程主要内容复习

- \* 常用术语

- \* Data、DB、DBMS、DBS、DBA、MetaData、媒体、多媒体、IR、MIRS

- \* 数据库技术 - 数据管理技术

---

- \* 共享程度高、冗余低、易扩充
  - \* 独立性高
  - \* 由DBMS统一管理和控制

- \* MMDB

- \* 所管理数据的特点 - 多媒体数据
  - \* 用户的需求变化

- \* 数据检索、信息检索、基于内容的检索

# 数据模型 & 数据库系统结构

---

孟放

[mengfang@cuc.edu.cn](mailto:mengfang@cuc.edu.cn)

3/17/2020

# 主要内容

---

- \* 数据模型

- \* 基本概念
- \* 数据模型的组成要素
- \* 概念模型 - ER图
- \* 逻辑模型 - 关系模型

- \* 数据库系统结构

- \* 基本概念
- \* 三级模式及两级映像

# 主要内容

---

- \* **数据模型**

- \* 基本概念
- \* 数据模型的组成要素
- \* 概念模型 - ER图
- \* 逻辑模型 - 关系模型

- \* **数据库系统结构**

- \* 基本概念
- \* 三级模式及两级映像

# 数据模型 - 基本概念

- \* 模型：对现实世界中某个研究对象特征的模拟和抽象。如，航模 - 飞机。
- \* 数据模型：对现实世界数据特征的抽象，实现对数据的描述、组织、操纵。
- \* DM: Data Model
- \* DM应满足的三个要求
  1. 比较真实地模拟现实世界
  2. 容易为人所理解
  3. 便于在计算机上实现
- \* 在系统开发的不同阶段，应采用不同的数据模型。

# 数据模型 - 多级数据模型

## \* 不同阶段采用不同DM

**概念模型**：也称“信息模型”，按照用户观点来对数据和信息建模，用于数据库设计。

**逻辑模型**：从计算机观点来对数据建模，用于DBMS实现。

**物理模型**：面向计算机系统，实现数据在系统内部的具体存储。

用户

现实世界

概念化

信息世界

计算机

形式化

计算机世界



# 数据库设计步骤 - 六个阶段

1. 需求分析
  - \* 需求收集和分析 - 了解和分析用户需求
2. 概念设计
  - \* 设计概念结构 - 形成概念模型，独立于具体DBMS
3. 逻辑设计
  - \* 设计逻辑结构；数据模型优化 - 针对某确定DBMS
4. 物理设计
  - \* 设计物理结构；评价设计，性能预测 - 适合应用环境的物理结构（存储结构、存取方法、维护代价等）
5. 数据库实施
  - \* 物理实现；试验性运行 - 建库及试运行
6. 数据库运行和维护
  - \* 使用维护数据库 - 正式运行、不断评估调试和修改

# 各级数据模型的主要特点

- \* **概念模型**：不考虑技术细节；具有较强表达能力（系统的主要功能和逻辑关系）；模型简单、清晰、易理解。
- \* **逻辑模型**：是数据库系统的核心和基础，常简称“**数据模型**”；从计算机系统角度对数据建模；定义了数据库系统的命名。
- \* **物理模型**：其具体实现是DBMS的任务，但DBA和系统设计人员必须熟悉该模型；决定了DBS的最终性能；需考虑“数据存取方式、数据存储结构、系统参数配置”等要素。



# (逻辑)数据模型的组成要素

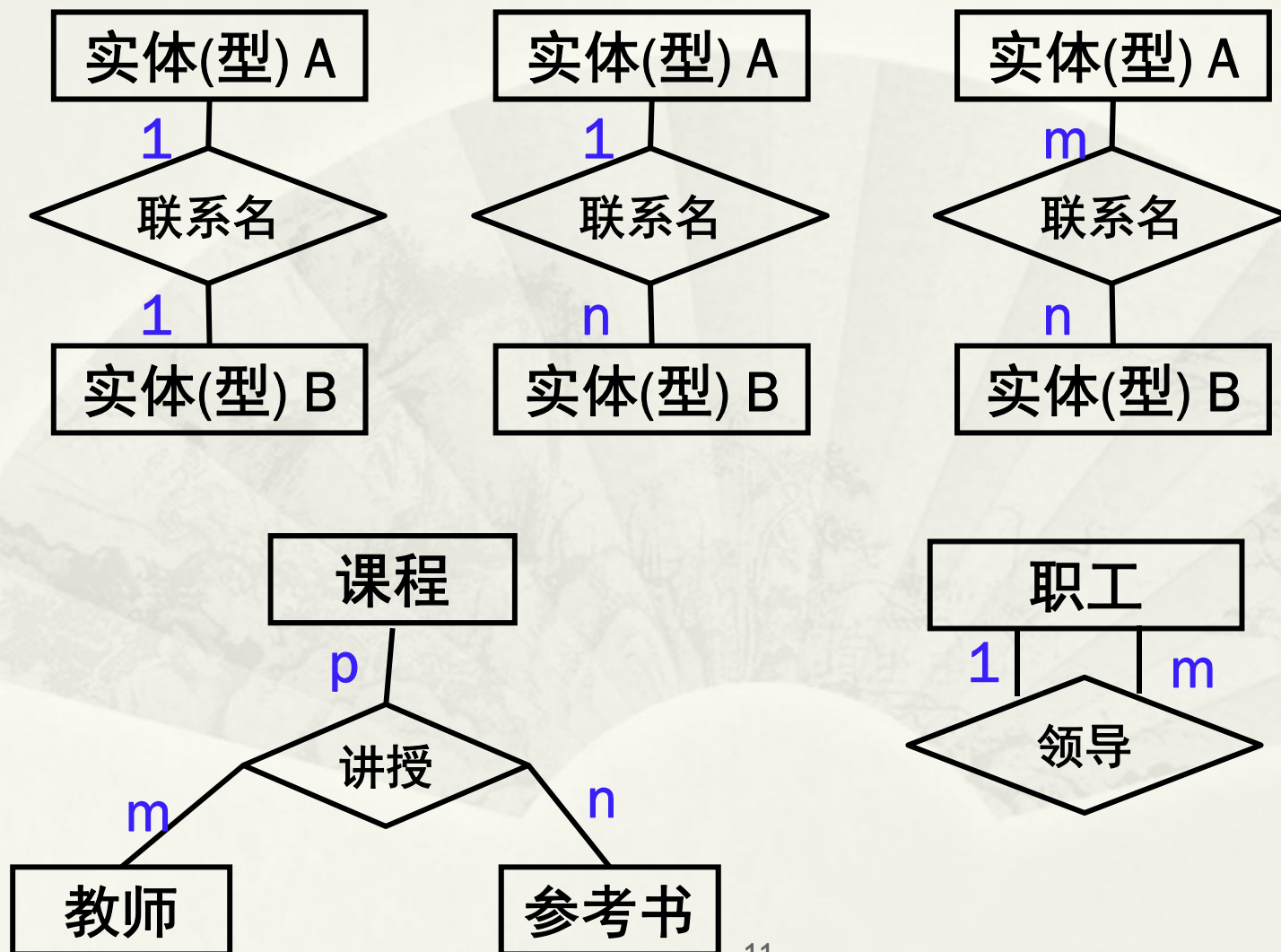
- \* **数据模型**：严格定义的一组概念集合，精确描述了系统的静态特性、动态特性和完整约束性条件。
  - \* 数据结构 – 静态特性
    - \* 数据库内组成对象的描述、对象之间联系的定义
  - \* 数据操作 – 动态特性
    - \* 对库内各对象允许执行的操作集合，包括操作规则
  - \* 完整性约束条件
    - \* 一组完整性规则，给定了数据及其联系所具有的制约和依存条件

# 数据模型 – 概念模型 – E-R模型

## \* 信息世界中的基本概念

- \* 实体 (Entity) : 客观存在、可以相互区别的事物称为实体。实体可以是具体的人/事/物, 也可以是抽象的概念。 学生、课程
- \* 实体集 (Entity set) : 同一类型实体的集合。 全部学生
- \* 属性 (Attribute) : 实体的某一方面的特征
- \* 属性域 (Domain) : 属性的取值范围; 含值的类型 学生姓名的域: 字符串集合、学生性别的域: (男、女)
- \* 码(Key) : 唯一标识每个实体的属性或属性集 学号
- \* 实体型 (Entity type) : 实体名及其属性名集合来抽象和刻画同类实体。 学生 (学号, 姓名, 性别, ...)
- \* 联系 (Relationship) : 实体内部、实体间的联系

# 数据模型 - 概念模型 - 实体间联系



# 数据模型 - 概念模型 - ER图

在信息世界中建立的，完全不涉及信息在计算机系统上的表示（独立于计算机系统）的数据模型，称为概念数据模型，简称概念模型。最典型的是“实体联系模型”即E-R模型。

## E-R模型

实体联系模型（Entity Relationship Model）。该模型直接从现实世界中抽象出实体类型及实体间联系，然后用E-R图表示的数据模型。

## E-R图的四个基本成分：

实体名

矩形框表示实体型

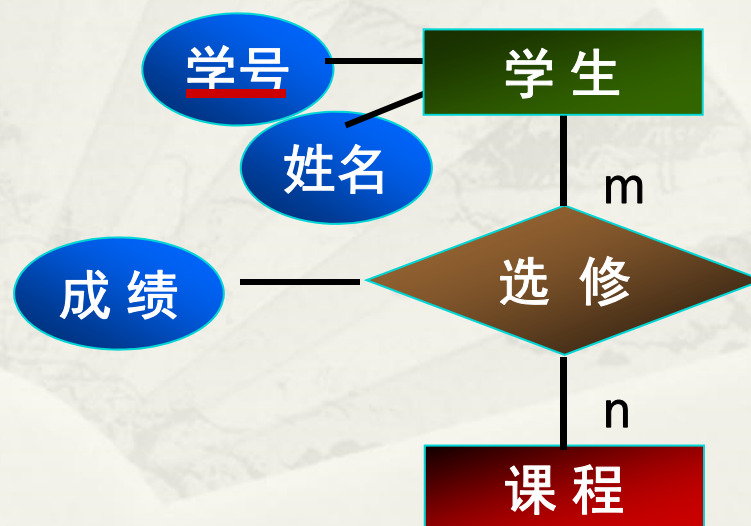
属性名

椭圆形表示属性

联系名

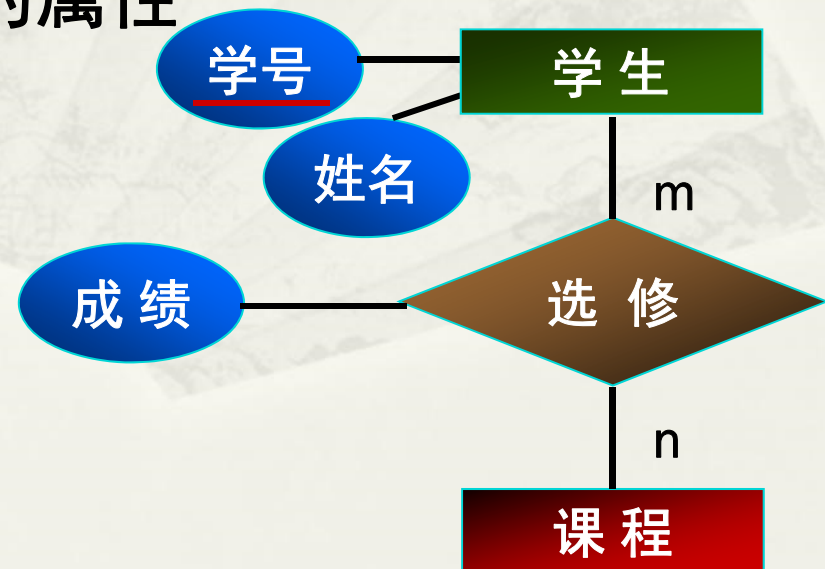
菱形表示联系

连接实体型与联系类型，也可用于表示实体与属性的联系并注明种类；对构成码的属性，在属性名下画一横线表示。



# 数据模型 – 概念模型 – ER图设计流程

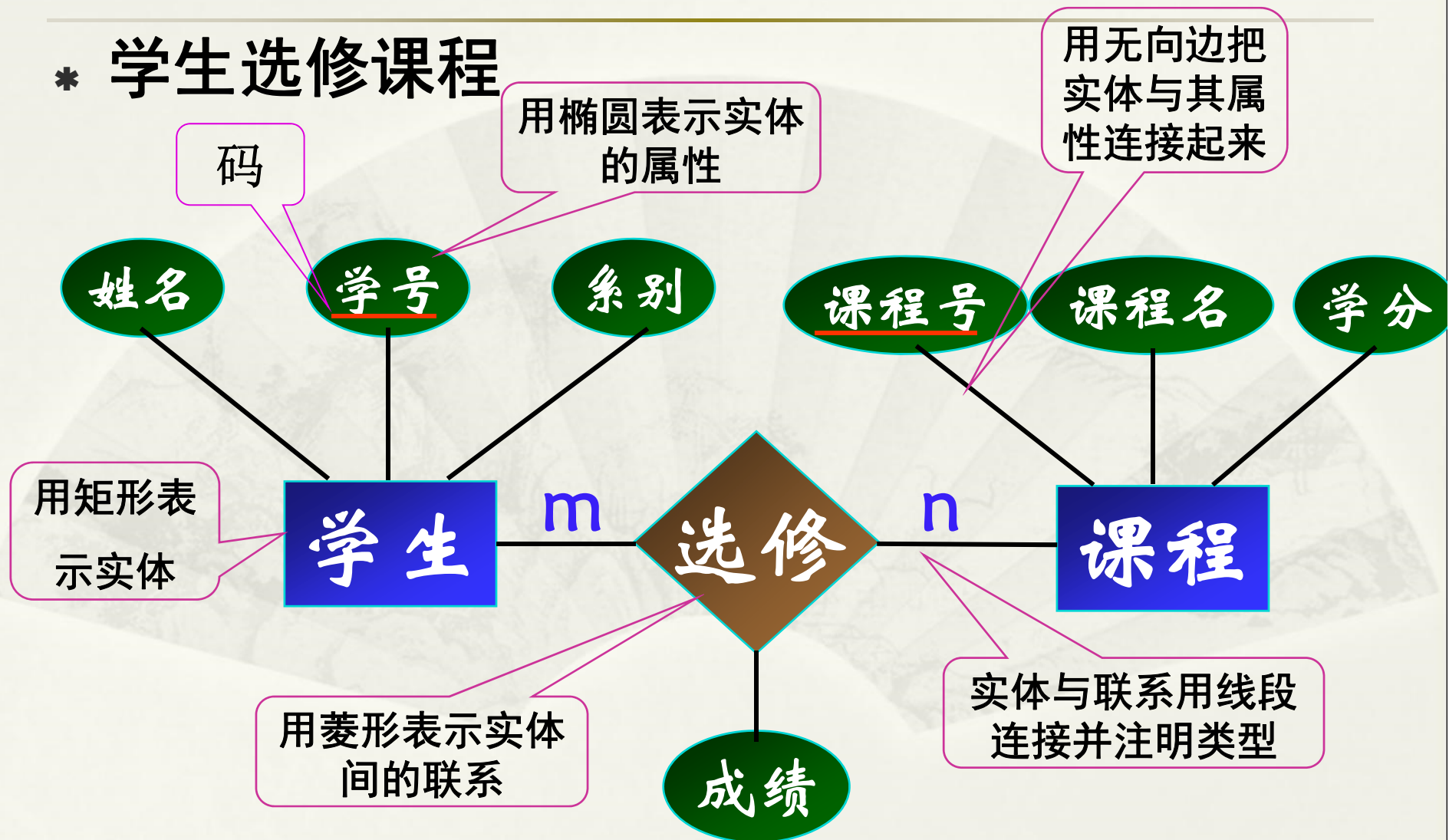
1. 首先确定实体类型：几个实体类型及相应的实体名
2. 确定联系类型：各实体类型之间是否有联系，是何种联系类型及相应的联系名
3. 连接实体类型和联系类型，组合成ER图
4. 确定实体类型和联系类型的属性
5. 确定实体类型的码





# 数据模型 - 概念模型 - ER图 - 例

## \* 学生选修课程





# 数据模型 - 逻辑模型

- \* 格式化模型
  - \* 层次模型、网状模型
- \* 关系模型
- \* 支持面向对象技术
  - \* 面向对象模型、对象关系模型
- \* 采纳/支持上述某种逻辑模型的数据库系统  
则命名为：层次数据库、网状数据库、关系数据库、对象关系数据库等。

# 数据模型 – 逻辑模型 – 关系模型

- \* 1970年IBM公司数据库系统的关系模型

学号	姓名	年龄	性别	系名	年级
2012004	张三	19	男	社会学	2012
2012006	李四	20	男	法律	2012
2012010	王五	18	女	计算机	2012
.....	.....	.....	.....	.....	.....

- \* 基本术语

- \* 关系(Relation): 一张（二维）表格
- \* 元组(Tuple): 表中一行
- \* 属性(Attribute): 表中一列
- \* 码(Key): 唯一确定一个元组的某个属性组
- \* 域(Domain): 属性的取值范围
- \* 分量: 元组中的一个属性值
- \* 关系模式: 对关系的描述 – 表头
  - \* 如, 学生(学号, 姓名, 年龄, 性别, 系名, 年级)

# 数据模型 - 逻辑模型 - 关系模型

- \* 关系模型要求关系一定要规范
  - \* 不允许表的嵌套；对关系的定义必须满足[关系数据理论](#)。
- \* 关系模型中的数据操作是集合运算，操作对象和操作结果都是关系，即表格。
- \* 关系模型的优点
  - \* 关系模型概念单一，只有表格一种形式；
  - \* 关系模型的存取路径对用户透明，更好满足数据的独立性、安全保密性，也简化系统开发流程。

# 关系数据理论

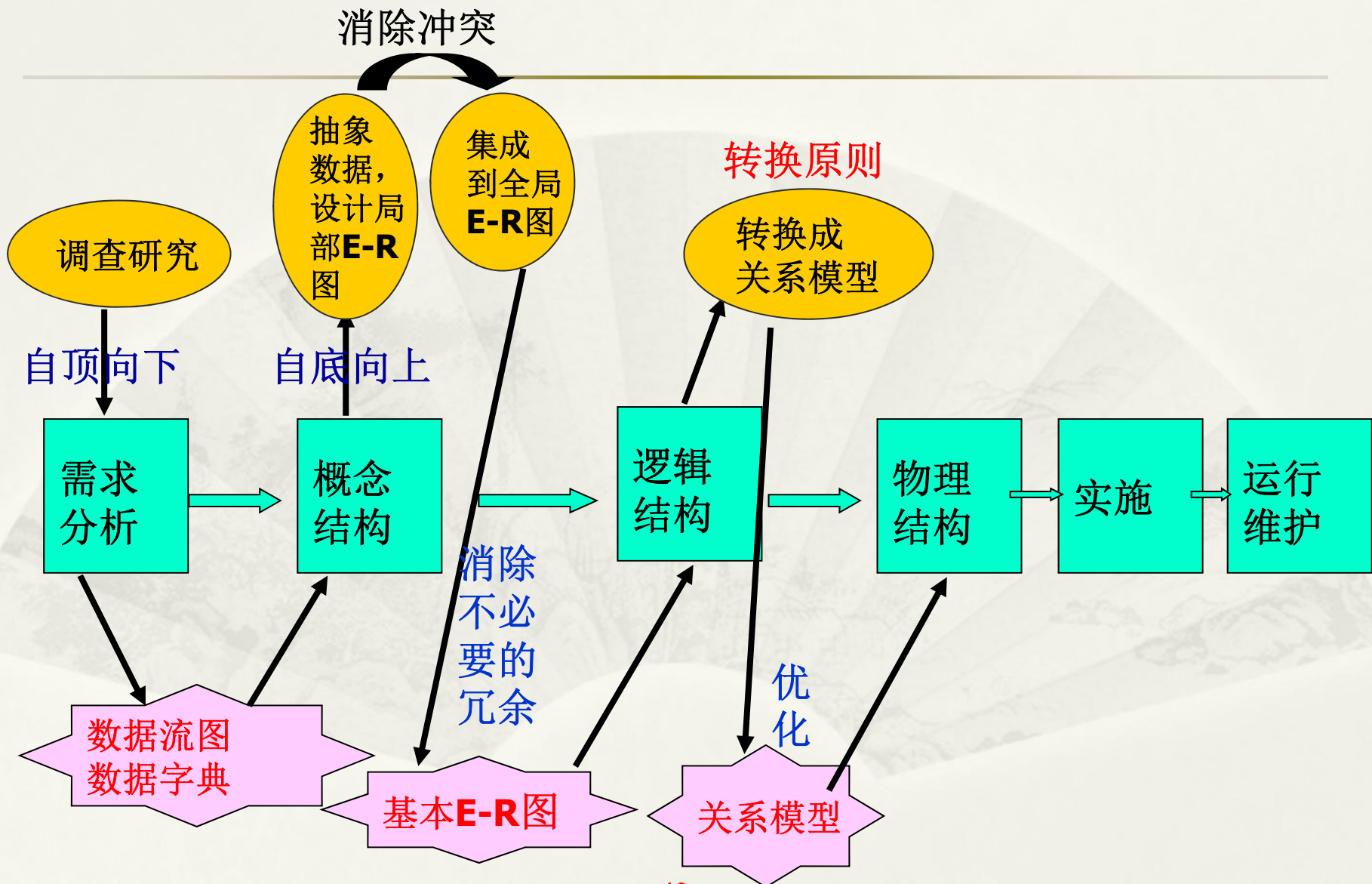
\* 关系数据库的逻辑设计问题：如何构建适合给定应用需求的数据库**模式**？

\* 构建几个关系模式？关系模式的属性组成？

Sno	Sdept	Instructor	Cno	Grade
S1	广电工	李老师	C1	95
S2	广电工	李老师	C1	90
S3	广电工	李老师	C1	88
S4	广电工	李老师	C1	70
S5	广电工	李老师	C1	78
.....	.....	.....	.....	.....

**存在问题：**

1. 数据冗余
2. 更新异常
3. 插入异常
4. 删除异常
5. ....



# 主要内容

---

- \* 数据模型

- \* 基本概念
- \* 数据模型的组成要素
- \* 概念模型 - ER图
- \* 逻辑模型 - 关系模型

- \* 数据库系统结构

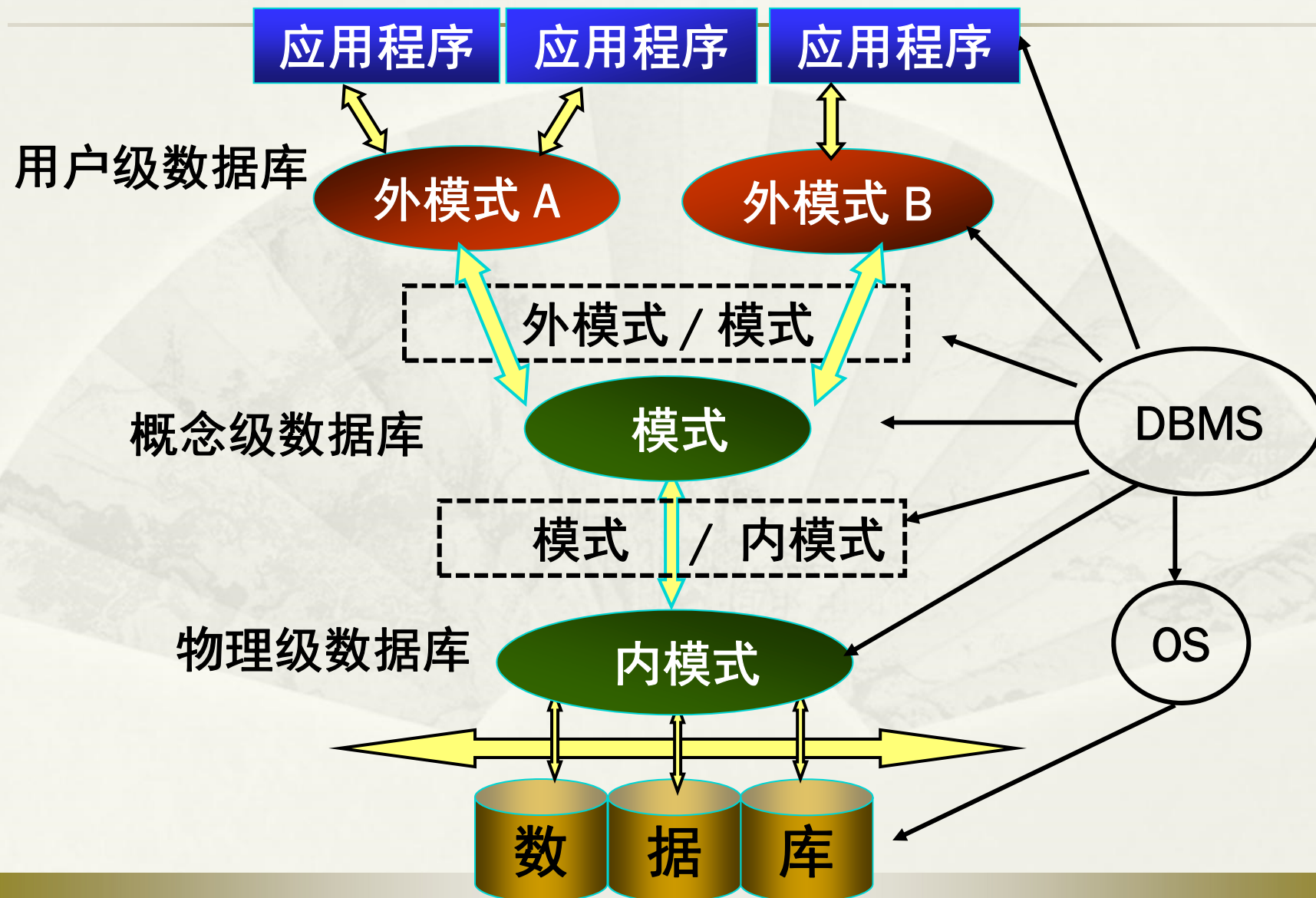
- \* 基本概念
- \* 三级模式及两级映像



# 基本概念

- \* **“型”**：对实体的结构和属性的说明
  - \* （关系数据库）实体型：一张二维表的结构和列属性
- \* **“值”**：型的一个具体取值
  - \* 实体值：表中记录。全部记录可成为实体集。
- \* **模式**：数据库内全体数据的逻辑结构和特征描述。对应“型”的概念。
  - \* 相对稳定
- \* **实例**：模式的一个具体取值即为模式的一个实例，一个模式可以有多个实例。
  - \* 相对变动

# 数据库系统结构 - 三级模式、两级映像



# 数据库系统结构 – 三级模式

- \* 模式Schema：数据逻辑结构和特征的描绘，与具体取值相比，是相对稳定的。
- \* 三级模式
  - \* 模式：逻辑模式。库中全部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。
    - \* 即不涉及数据的物理存储细节、也与具体应用程序等无关
  - \* 外模式：用户模式。用户能看到的、与某个应用程序有关的局部数据的逻辑结构和特征描述。
    - \* 保护数据库安全的有力措施
  - \* 内模式：存储模式。是数据在数据库内部的表示，如存储结构、索引方式、是否加密等。

# 数据库系统结构 – 两级映像

- \* 保证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。

- \* 外模式/模式映像 – 逻辑独立性

- \* 对于每个外模式，由该映像定义其与模式之间的对应关系，以确保模式改变时，该映像做相应改变即可。不需修改外模式，也就不需修改应用程序。

- \* 模式/内模式映像 – 物理独立性

- \* 模式/内模式映像是唯一的，定义了数据全局逻辑结构与存储结构之间的对应关系。当数据库存储结构改变时，可通过该映像的相应改变来确保模式不需改变，从而应用程序也就不需改变了。

# 数据库三级模式结构的优点

---

- \* 保证了数据的相对独立性
- \* 简化用户接口（只考虑相关的外模式）
- \* 有利于数据共享
- \* 有利于安全保密
- \* 提高应用系统开发效率



# 小结

---

- \* 数据模型

- \* 基本概念
- \* 数据模型的组成要素
- \* 概念模型-ER图
- \* 逻辑模型-关系模型

- \* 数据库系统结构

- \* 基本概念
- \* 三级模式及两级映像



# 作业

\* P.34 4、6、13、15、17

**[思考题，四周时间]** 如果要开发一个电影资源查询系统（考虑其所应具备的主要功能），请问：

- 需要定义几张表格、各自属性；并画出对应的ER图
- 或，自行设计一个应用系统的后台数据需求。