
华中科技大学计算机科学与技术学院
SCHOOL OF COMPUTER SCIENCE & TECHNOLOGY, HUST

第七章 数据库设计

• 学习内容

- 7.1 数据库设计概述
- 7.2 需求分析
- 7.3 概念数据库设计
- 7.4 逻辑数据库设计
- 7.5 物理数据库设计
- 7.6 管理和维护

• 学习目标

- 掌握数据库设计的内容和特点
- 掌握数据库概念设计的方法
- 掌握数据库逻辑设计的方法
- 了解数据库物理设计的方法

7.1 数据库设计概述

- 7.1.1 数据库应用
- 7.1.2 数据库设计方法和特点
- 7.1.3 数据库设计基本步骤

7.1.1 数据库与MIS

- 信息系统

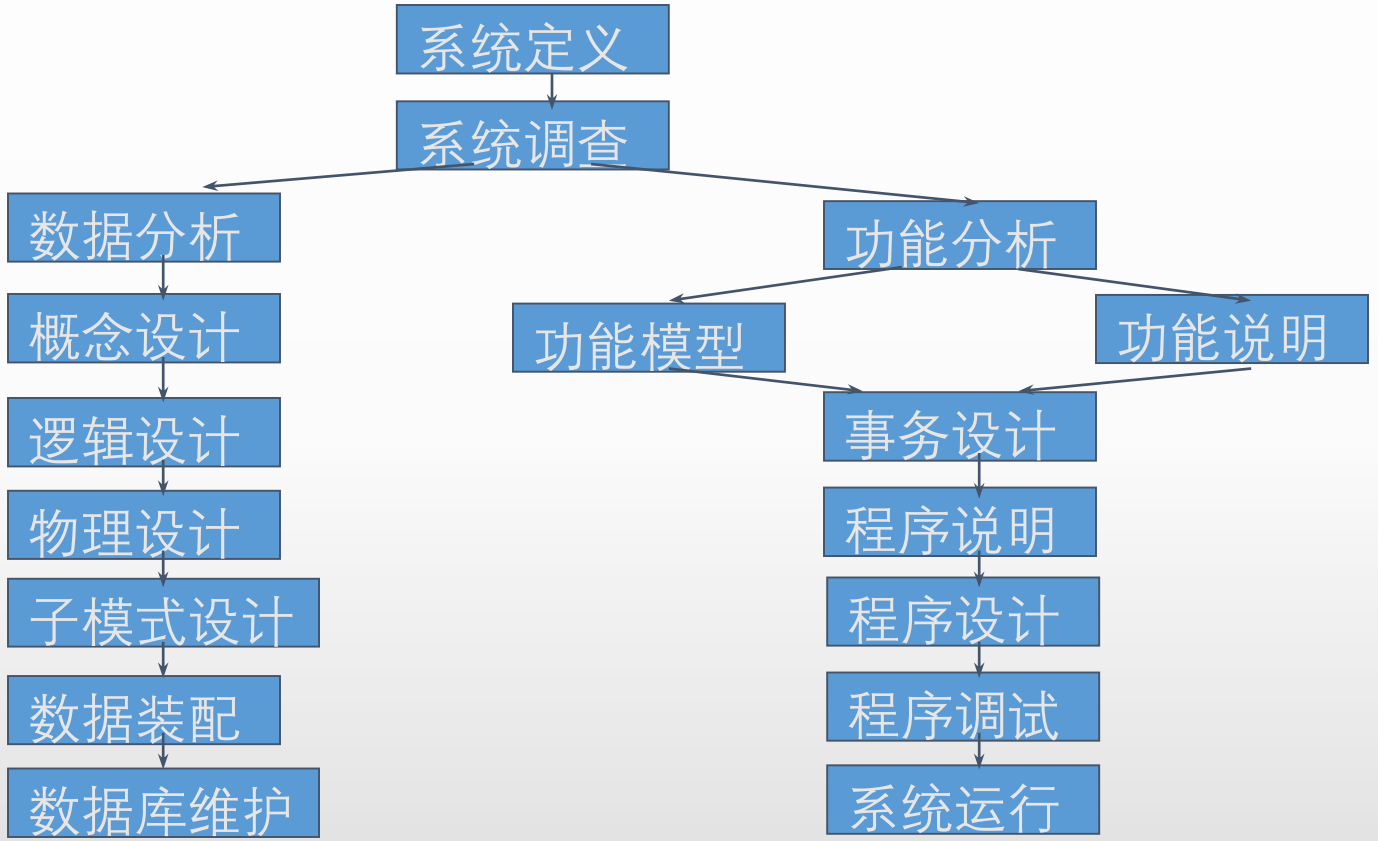
- 信息系统:MIS，数据仓库，ERP,GIS,文档数据库
  信息系统是提供信息、辅助人们对环境进行控制和进行决策的系统
- 数据库设计是指对于一个给定的应用环境，构造最优的数据库模式，建立数据库及其应用系统，有效存储数据，满足用户信息要求和处理要求。

- 一个好的数据库产品不等于就有一个好的应用系统

7.1.2数据库设计设计方法与特点

- 设计特点
  - 静态数据设计与动态行为设计相分离性
  - 反复性、试探性、分布进行
  - 数据库设计多解性

静态数据设计与动态行为设计相分离



数据库设计方法：

- 数据库设计方法目前可分为四类：直观设计法、规范设计法、计算机辅助设计法和自动化设计法。
- 直观设计法也叫手工试凑法。
- 1978年10月，来自三十多个国家的数据库专家在美国新奥尔良（New Orleans）市专门讨论了数据库设计问题，他们运用软件工程的思想和方法，提出了数据库设计的规范，这就是著名的新奥尔良法，它是目前公认的比较完整和权威的一种规范设计法。新奥尔良法将数据库设计分成需求分析（分析用户需求）、概念设计（信息分析和定义）、逻辑设计（设计实现）和物理设计（物理数据库设计）。目前，常用的规范设计方法大多起源于新奥尔良法，并在设计的每一阶段采用一些辅助方法来具体实现。

- 基于E-R模型的数据库设计方法是由P.P.S.chen于1976年提出的数据库设计方法，其基本思想是在需求分析的基础上，用E-R（实体—联系）图构造一个反映现实世界实体之间联系的企业模式，然后再将此企业模式转换成基于某一特定的DBMS的概念模式。
- 基于3NF的数据库设计方法是由S·Atre提出的结构化设计方法，其基本思想是在需求分析的基础上，确定数据库模式中的全部属性和属性间的依赖关系，将它们组织在一个单一的关系模式中，然后再分析模式中不符合3NF的约束条件，将其进行投影分解，规范成若干个3NF关系模式的集合。

- 基于视图的数据库设计方法:此方法先从分析各个应用的数据着手，其基本思想是为每个应用建立自己的视图，然后再把这些视图汇总起来合并成整个数据库的概念模式。
- 除了以上三种方法外，规范化设计方法还有实体分析法、属性分析法和基于抽象语义的设计方法等。
- 规范设计法从本质上来说仍然是手工设计方法，其基本思想是过程迭代和逐步求精。

7.1.3数据库设计基本步骤

数据库设计的步骤

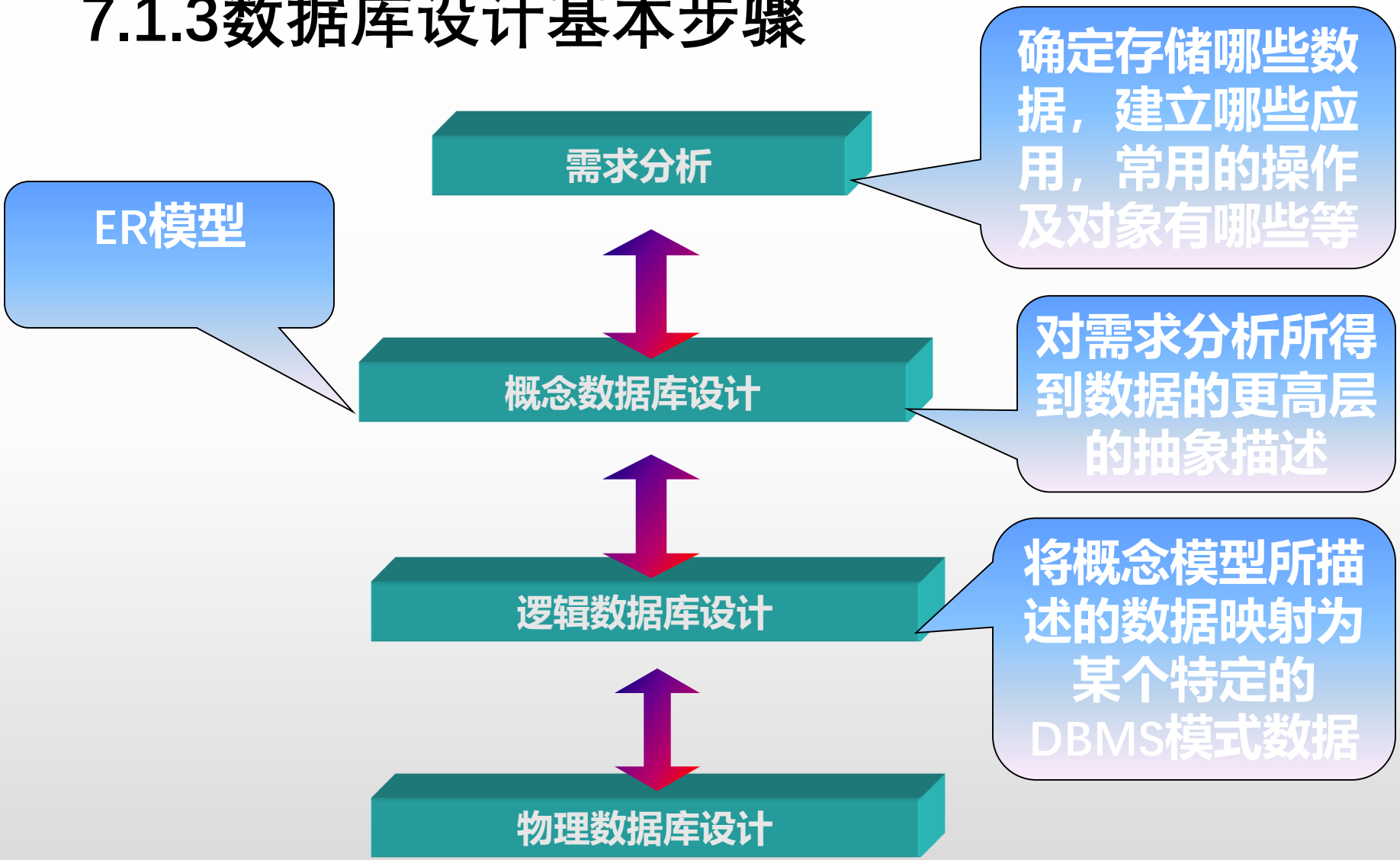
- “数据库设计的生存期”
- 按规范设计法可将数据库设计分为六个阶段:
  - 系统需求分析阶段
  - 概念结构设计阶段
  - 逻辑结构设计阶段
  - 物理设计阶段
  - 数据库实施阶段
  - 数据库运行与维护阶段

- 需求分析是整个数据库设计过程的基础，要收集数据库所有用户的信息内容和处理要求，并加以规格化和分析。这是最费时、最复杂的一步，但也是最重要的一步，相当于待构建的数据库大厦的地基，它决定了以后各步设计的速度与质量。需求分析做得不好，可能会导致整个数据库设计返工重做。在分析用户需求时，要确保用户目标的一致性。
- 概念设计是把用户的信息要求统一到一个整体逻辑结构中，此结构能够表达用户的要求，是一个独立于任何DBMS软件和硬件的概念模型。
- 逻辑设计是将上一步所得到的概念模型转换为某个DBMS所支持的数据模型，并对其进行优化。

- 物理设计是为逻辑数据模型建立一个完整的能实现的数据库结构，包括存储结构和存取方法。
- 在实施阶段，根据物理设计的结果把原始数据装入数据库，建立一个具体的数据库并编写和调试相应的应用程序。应用程序的开发目标是开发一个可依赖的有效的数据库存取程序，来满足用户的处理要求。
- 运行和维护阶段：这一阶段主要是收集和记录实际系统运行的数据，数据库运行的记录用来提高用户要求的有效信息，用来评价数据库系统的性能，进一步调整和修改数据库。在运行中，必须保持数据库的完整性，并能有效地处理数据库故障和进行数据库恢复。在运行和维护阶段，可能要对数据库结构进行修改或扩充。

设计阶段	设计描述	
	数据	处理
需求分析	数据字典、全系统中数据项、 数据流、数据存储的描述	数据流图和定表（判定树） 数据字典中处理过程的描述
概念结构设计	概念模型（E-R图） 数据字典	系统说明书。包括: (1) 新系统要求、方案和概图 (2) 反映新系统信息的数据流图
逻辑结构设计	某种数据模型 关系模型	系统结构图 非关系模型（模块结构图）
物理设计	存储安排 存取方法选择 存取路径建立	模块设计 IPO表
实施阶段	编写模式 装入数据 数据库试运行	程序编码 编译联结 测试
运行维护	性能测试，转储/恢复数据库 重组和重构	新旧系统转换、运行、维护（修正性、 适应性、改善性维护）

7.1.3数据库设计基本步骤



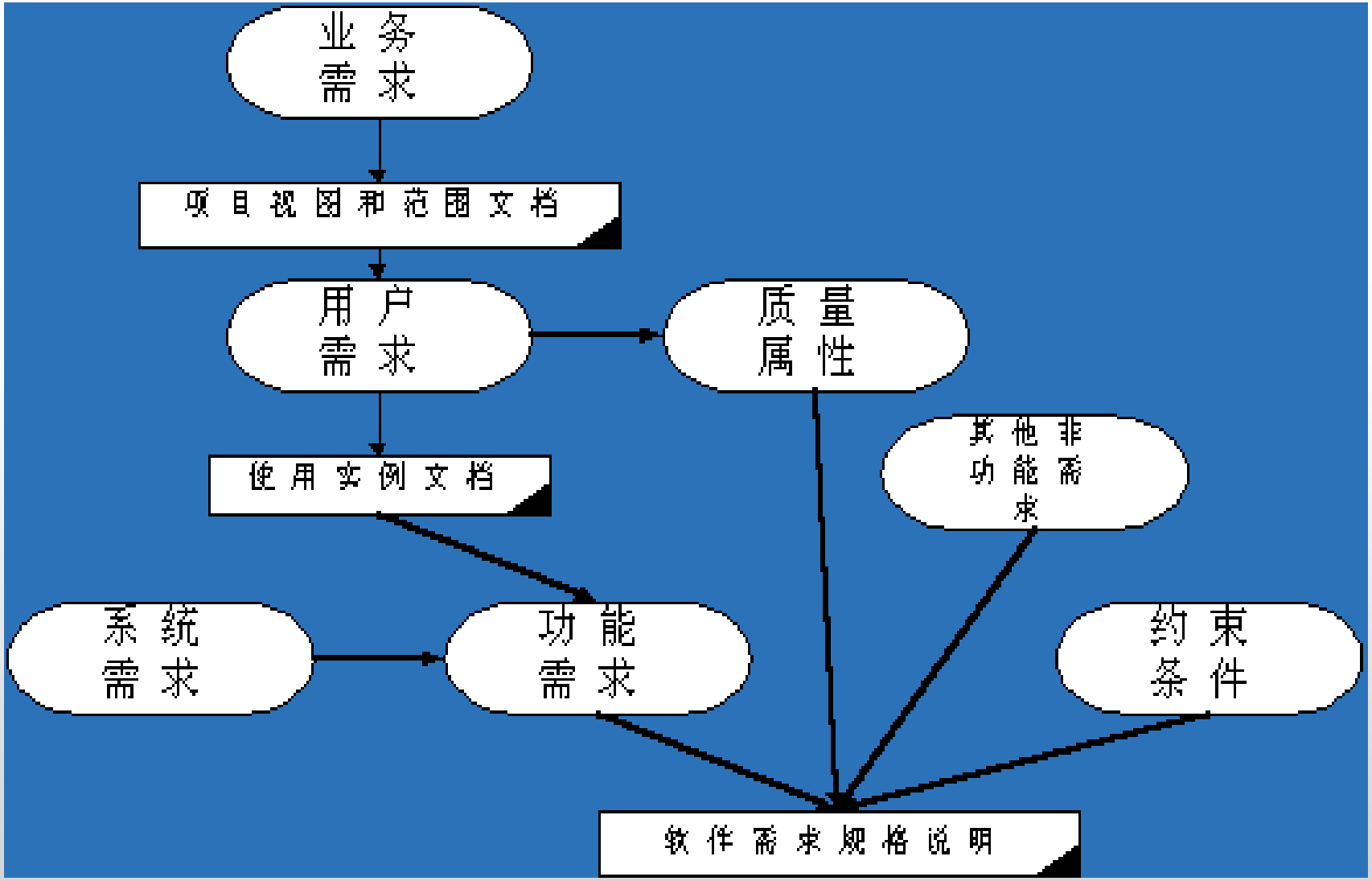
7.2 需求分析

- 7.2.1 需求分析的任务
- 7.2.2 需求分析的方法
- 7.2.3 需求分析的步骤
- 7.2.4 需求分析阶段文档

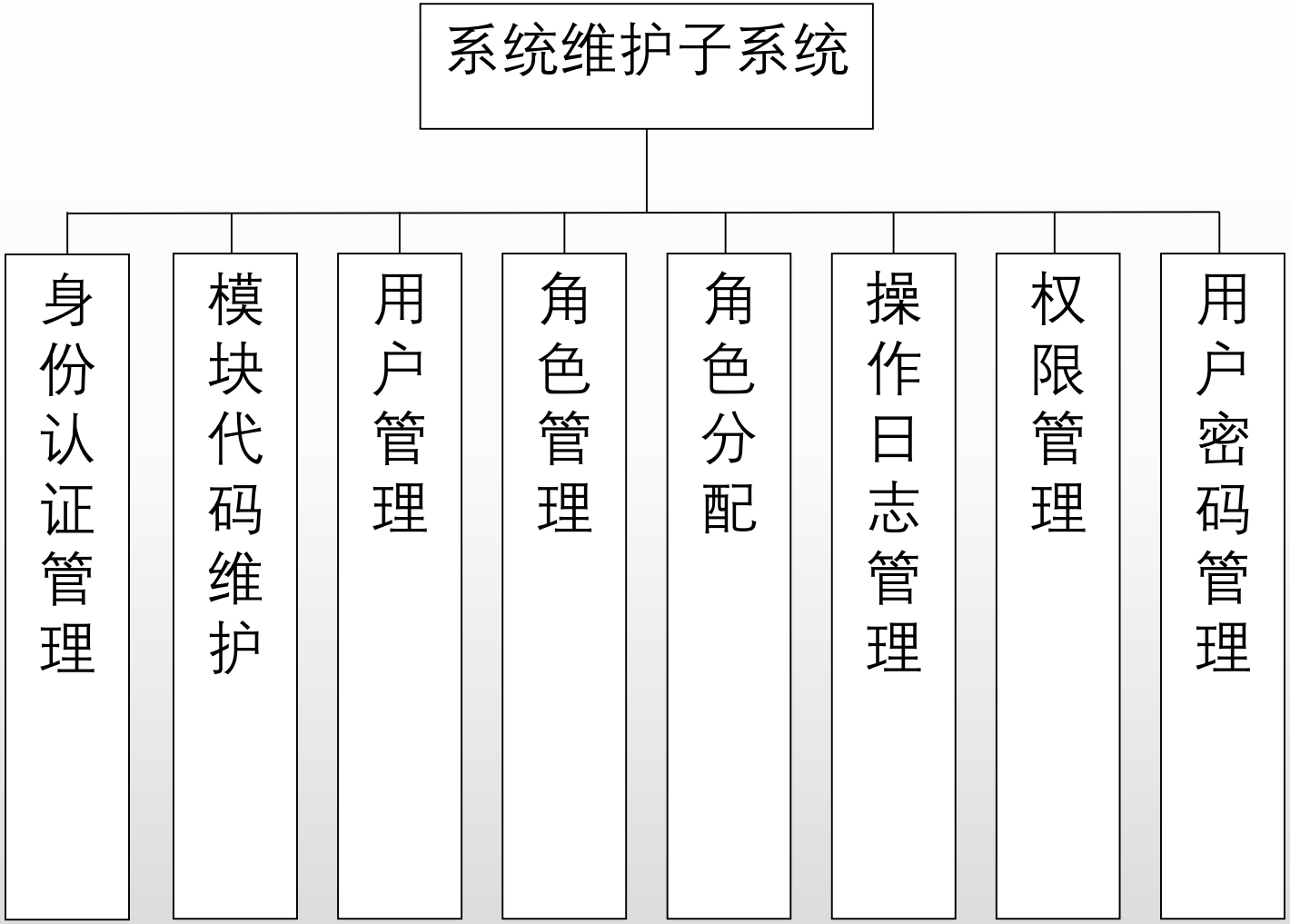
7.2 需求分析

- 7.2.1 需求分析的任务
  - 需求分析就是从系统数据处理加工的过程中抽象并描述出系统的概念模型，为系统的实现做准备。了解系统数据来源、流向、处理过程、处理结果均是需求分析阶段必须完成的工作。软件工程的需求分析分为需求获取、需求分析、编写规格说明书和需求验证。
    - 功能需求
    - 信息（数据）需求
    - 性能、运行需求
    - 完整性、安全性需求
    - 其他需求
    - 提交需求说明文档

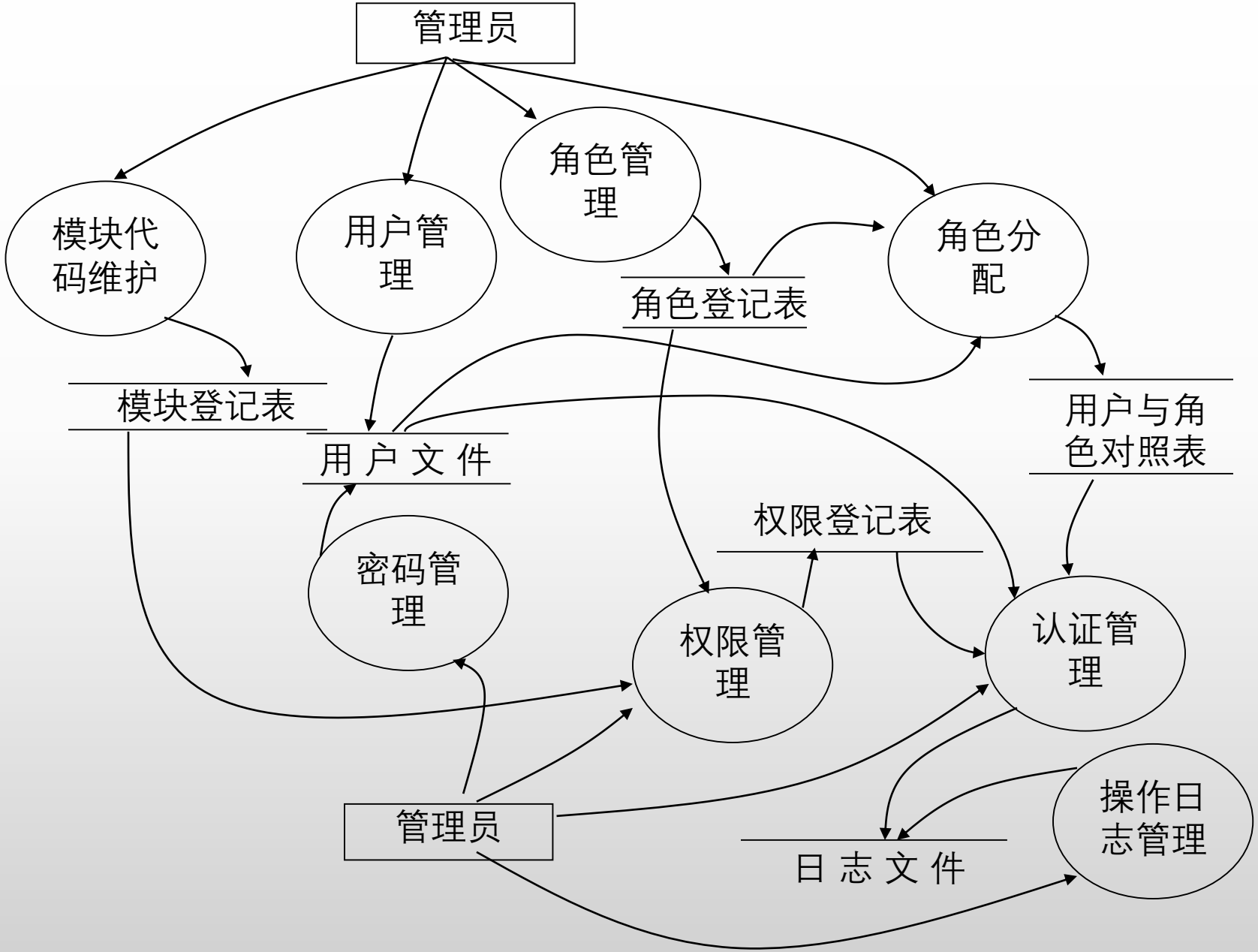
•软件需求各组成部分关系



功能树



功能间的数据关联


管理员
模块代码维护
用户管理
角色管理
角色登记表
角色分配
模块登记表
用 户 文 件
用户与角色对照表
权限登记表
密码管理
权限管理
认证管理
管理员
日 志 文 件
操作日志管理

需求说明文档

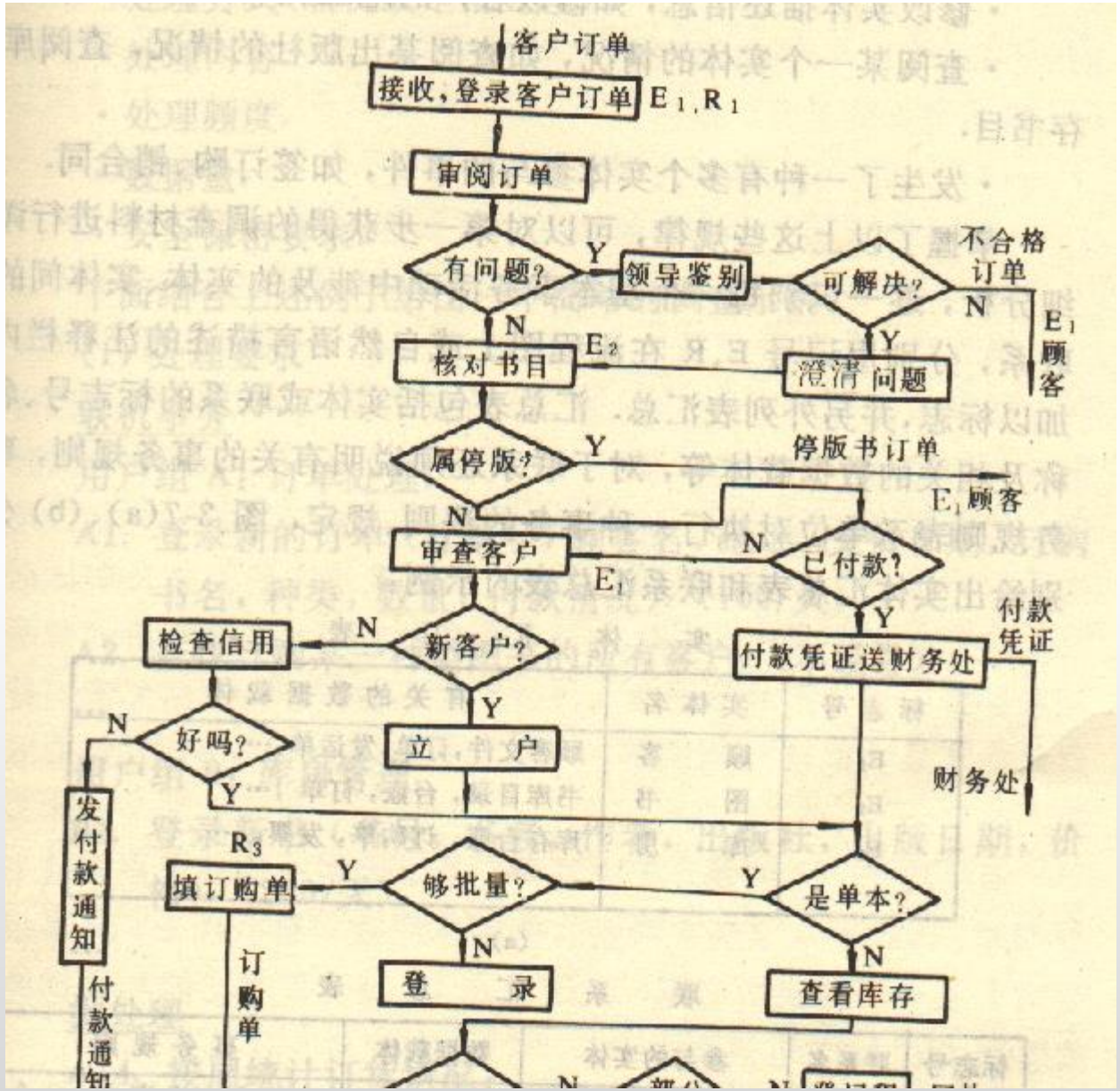
级别编号	模块名称	功能说明	子模块	操作步骤	操纵数据
01##	用户登录子模块	1.通过用户名和用户口令来控制该系统的合法用户，以及这些用户相应的权限 2.用户分成高级用户和普通用户两类，其中高级用户为教务管理科工作人员，普通用户包括管理科工作人员和各院系教务员		1.选择用户名; 2.输入相应口令; 3.系统判断该用户的合法性以及相应权限，并进入相应操作界面	系统用户功能权限表（**）

7.2 需求分析（续）

- 7.2.2 需求分析的方法
  - 1. 方法分类

    - 面向数据的方法
      - 着眼于数据对现实世界的描述作用。

    - 面向过程的方法
      - 着眼于数据在各项功能活动中被加工变换的流程。

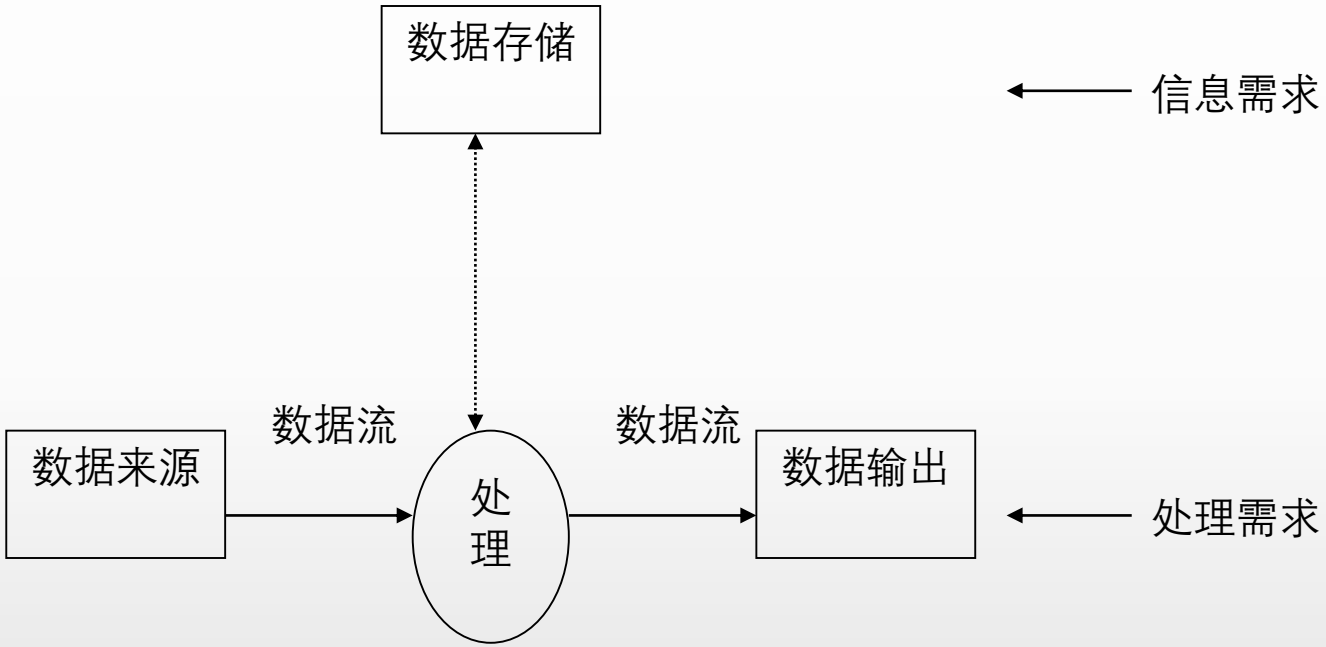

客户订单
接收，登录客户订单
E1，R1
审阅订单
有问题？
Y
领导鉴别
可解决？
不合格订单
N
E1
顾客
N
核对书目
E2
澄清问题
Y
属停版？
Y
停版书订单
E1顾客
N
审查客户
E1
N
已付款？
Y
付款凭证
检查信用
N
新客户？
付款凭证送财务处
财务处
N
好吗？
Y
立　户
Y
发付款通知
R3
填订购单
Y
够批量？
Y
是单本？
N
N
登　录
查看库存
付款通知
订购单

7.2.2 需求分析的方法(续)

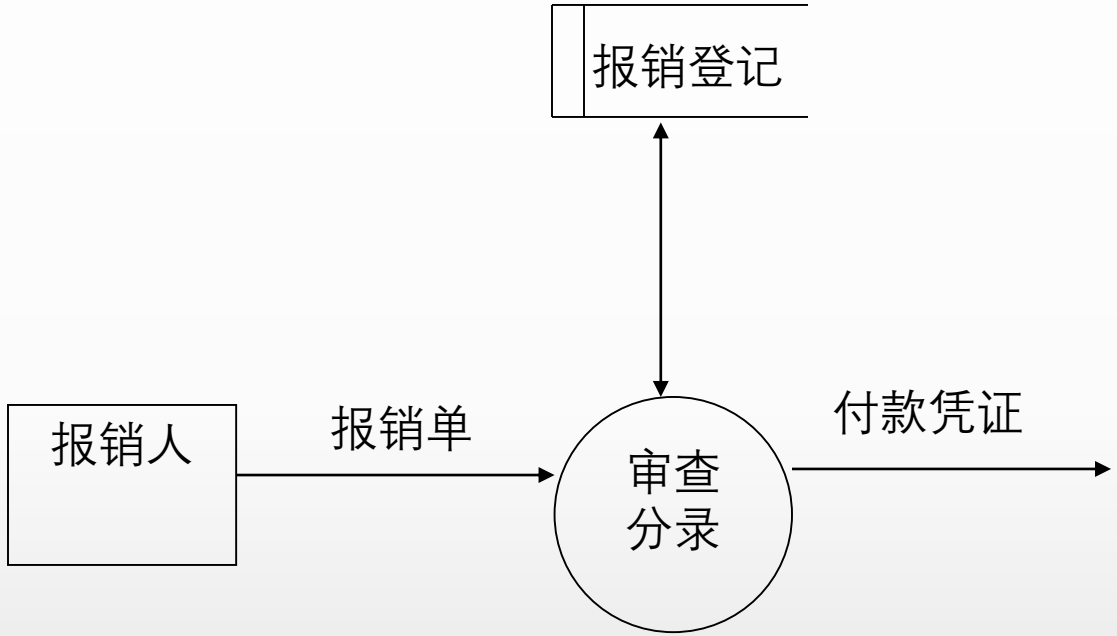
- 2. 结构化分析方法简介
  - 工具：数据流程图、数据字典
  - 数据流图：表达了数据和处理的关系
  - 数据字典：系统中各类数据的集合

- 数据流程图
  - 处理过程
  - 数据流
  - 数据流的终点或源点
  - 存储池

数据流程图示例



数据流程图示例



- 数据字典:是对系统中数据的详细描述，是各类数据结构和属性的清单。它与数据流图互为注释。
- 数据字典贯穿于数据库需求分析直到数据库运行的全过程，在不同的阶段其内容和用途各有区别。
- 在需求分析阶段，它通常包含以下五部分内容。

(1) 数据项

- 数据项是数据的最小单位，其具体内容包括：数据顶名、含义说明、别名、类型、长度、取值范围、与其他数据项的关系。
- 其中，取值范围、与其他数据项的关系这两项内容定义了完整性约束条件，是设计数据检验功能的依据。

(2) 数据结构

- 数据结构是数据项有意义的集合。内容包括：数据结构名、含义说明，组成这些内容的数据项名。

(3) 数据流

- 数据流可以是数据项，也可以是数据结构，它表示某一处理过程中数据在系统内传输的路径。
- 内容包括：数据流名、说明、流出去向、流入去向，组成这些内容的数据项或数据结构。
- 其中，流出过程说明该数据流由什么过程而来；流入过程说明该数据流到什么过程。

(4) 数据存储

- 处理过程中数据的存放场所，也是数据流的来源和去向之一。可以是手工凭证，手工文档或计算机文件。
- 包括｛数据存储名，说明，输入数据流，输出数据流，组成：｛数据项或数据结构｝，数据量，存取频度，存取方式｝。
- 存取方法:批处理/联机处理；检索/更新

(5) 处理过程

- 处理过程的处理逻辑通常用判定表或判定树来描述，数据字典只用来描述处理过程的说明性信息。
- 处理过程包括｛处理过程名，说明，输入：｛数据流｝，输出：｛数据流｝，处理，｛简要说明｝｝。
- 简要说明:说明处理过程的功能及处理要求。
- 功能是指该处理过程用来做什么（不是怎么做），处理要求指该处理频度要求，如单位时间里处理多少事务、多少数据量、响应时间要求等，这些处理要求是后面物理设计的输入及性能评价的标准。

7.2.3 需求分析的步骤

- 1.需求分析的步骤
  - 调查组织结构 ===〉系统的管理模式 ===〉各部门功能
  - 调查各部门业务活动 、职责 ===〉信息流程
  - 收集各种静态信息 ====〉原系统的信息存储
  - 确定系统的功能和系统的边界

7.2.3 需求分析的步骤(续)

- 2. 常用的需求分析调查方法
  - 跟班作业
  - 开调查座谈会
  - 请用户介绍
  - 提问
  - 设计调查问卷请用户填写
  - 查阅历史纪录
  - 在实际操作中往往是若干种方式同时进行

•调查问卷示例

业 务 活 动 询 问 表

部门 物资供应 填表人 李某 职务 库管员

业务项目	具 体 活 动
1. 发 料	接收领料单，审核料单，对合格的料单进行备货
	不合格的料单，…
2. 收 料	接收装运单，审核，质检，入库登记，…
3. 清仓盘点	按类清点库存实物，核对台账，…
…	…

业务名称： 订单处理

本项业务每天处理顾客邮来的订单，在接到客户订单以后首先登记流水账，然后审阅订单，查看每一个重要的数据项是否遗漏或模糊，所订的书日是否属停止出版发行，必要时报请主管领导鉴别确认，在无法确定时退回顾客并指明问题. 如订单附有付款，则将附有付款的订单副本送财务处入账；对没有付款的订单，检查顾客文件，判断客户是否属于已建立信用的个人或团体，如没有建立信用，则填发收到订单的回执及付款通知. 对于一个新的客户，应立户头，即将该客户的有关信息添加到顾客文件中. 对已经付款或信用良好的订单检查库房的库存账，看能否满足所有订货要求，如可满足，填写装运单及发票(对已付款者，加盖收讫字样). 它们将和书一起发运. 如果订单只能满足一部分，则只填写可满足部分的装运单及已付款的发票，附上一份未满足部分的回执. 将暂时缺货的书目填写到拖欠订单账上，一旦从出版社得到这些书立即交付拖欠订货. 对于不存库中的单本书的订单，待积满一定批量后，向出版社购买.

E_1

E_2

E_1

E_1

E_1

E_1

E_3, R_1

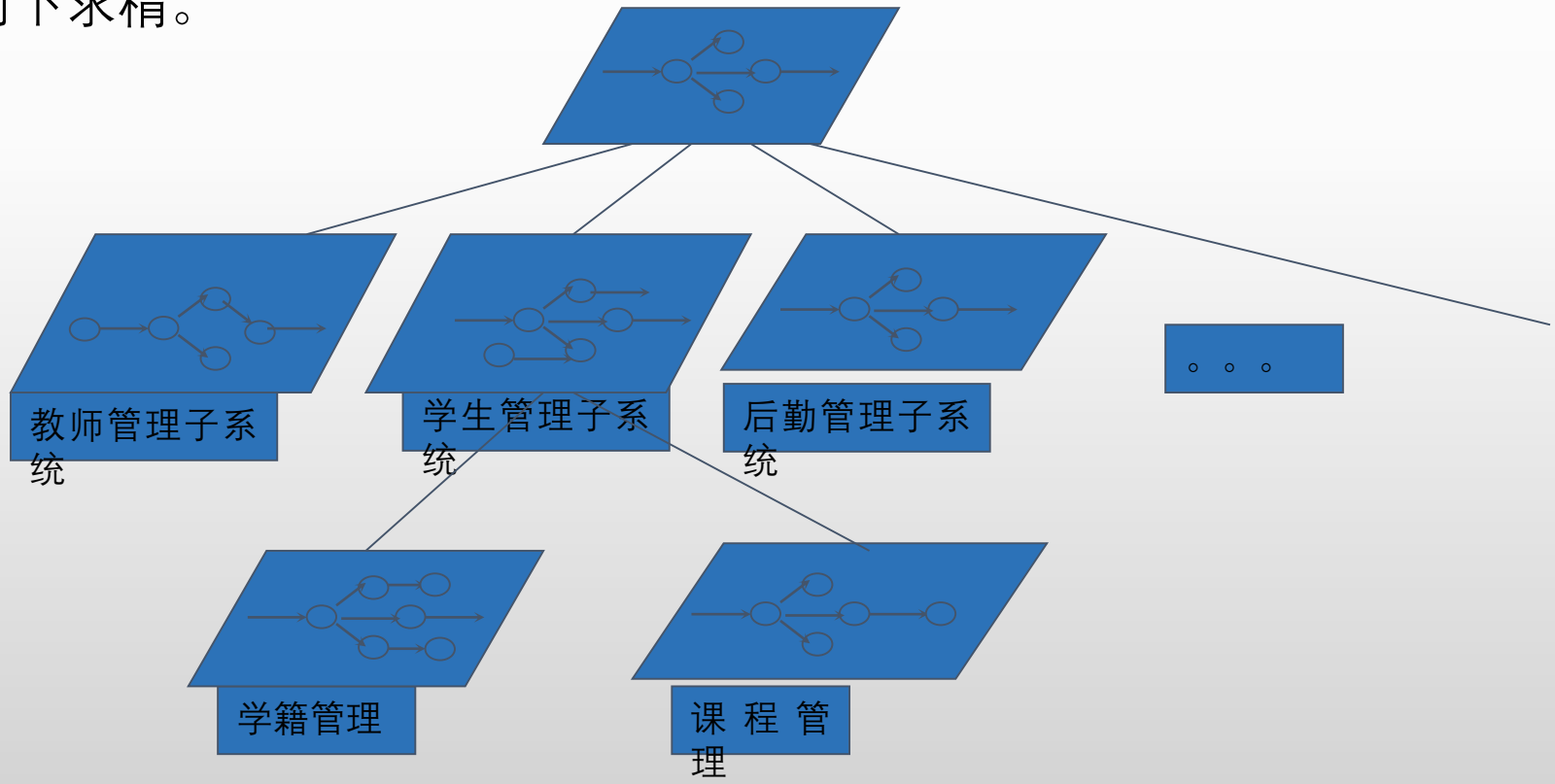
E_2, R_2

E_4, E_2, E_3

E_2, E_4, R_3

需求分析实例

- 假设开发某学校数据库管理系统，经过可行性分析和初步调查，采用自顶向下的方法可以抽象出该系统高层数据流图.逐步往下求精得出：教师管理子系统、学生管理子系统，后勤管理子系统、科研管理子系统、产业管理子系统。限于篇幅我们以学生管理子系统为例向下求精。



学籍管理的数据流图

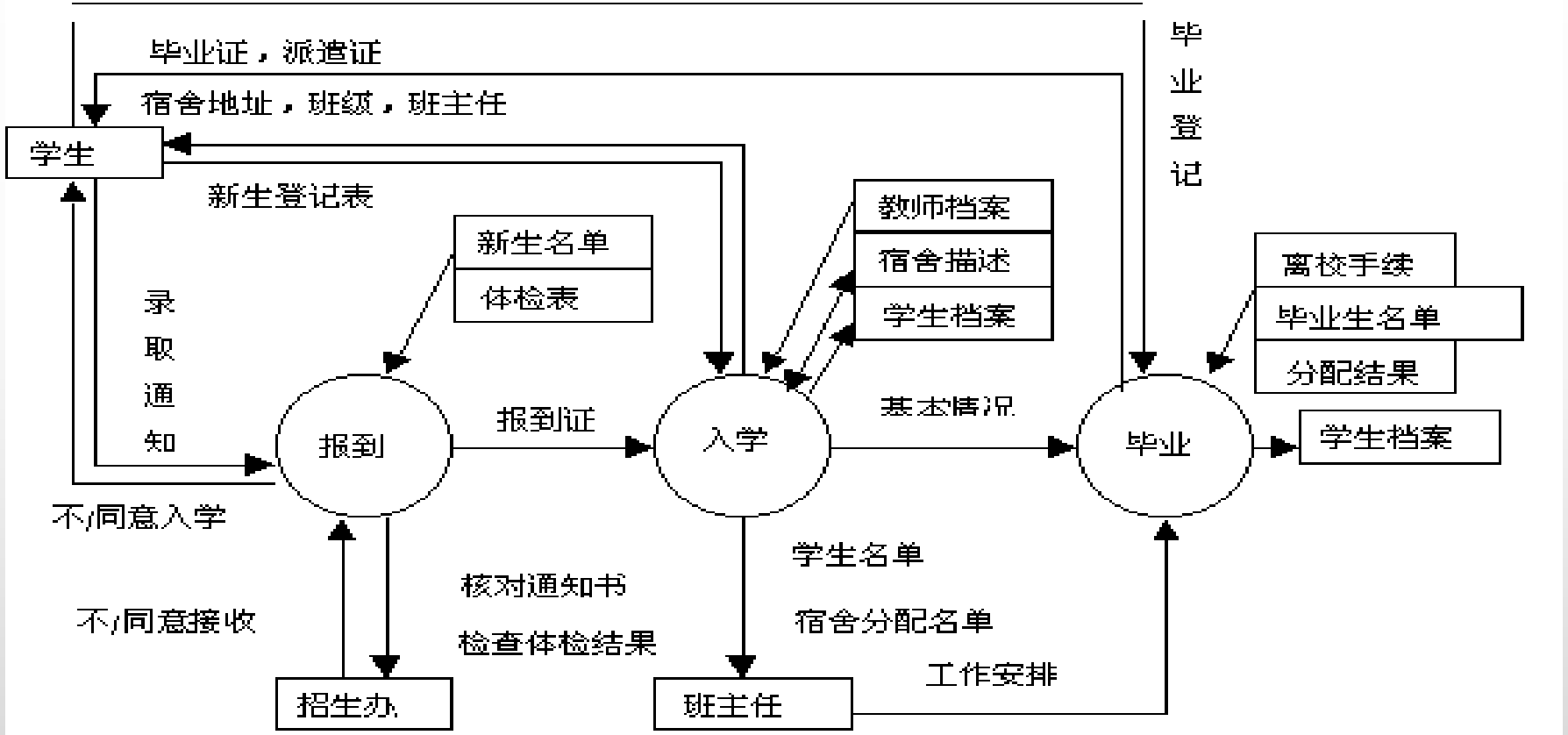
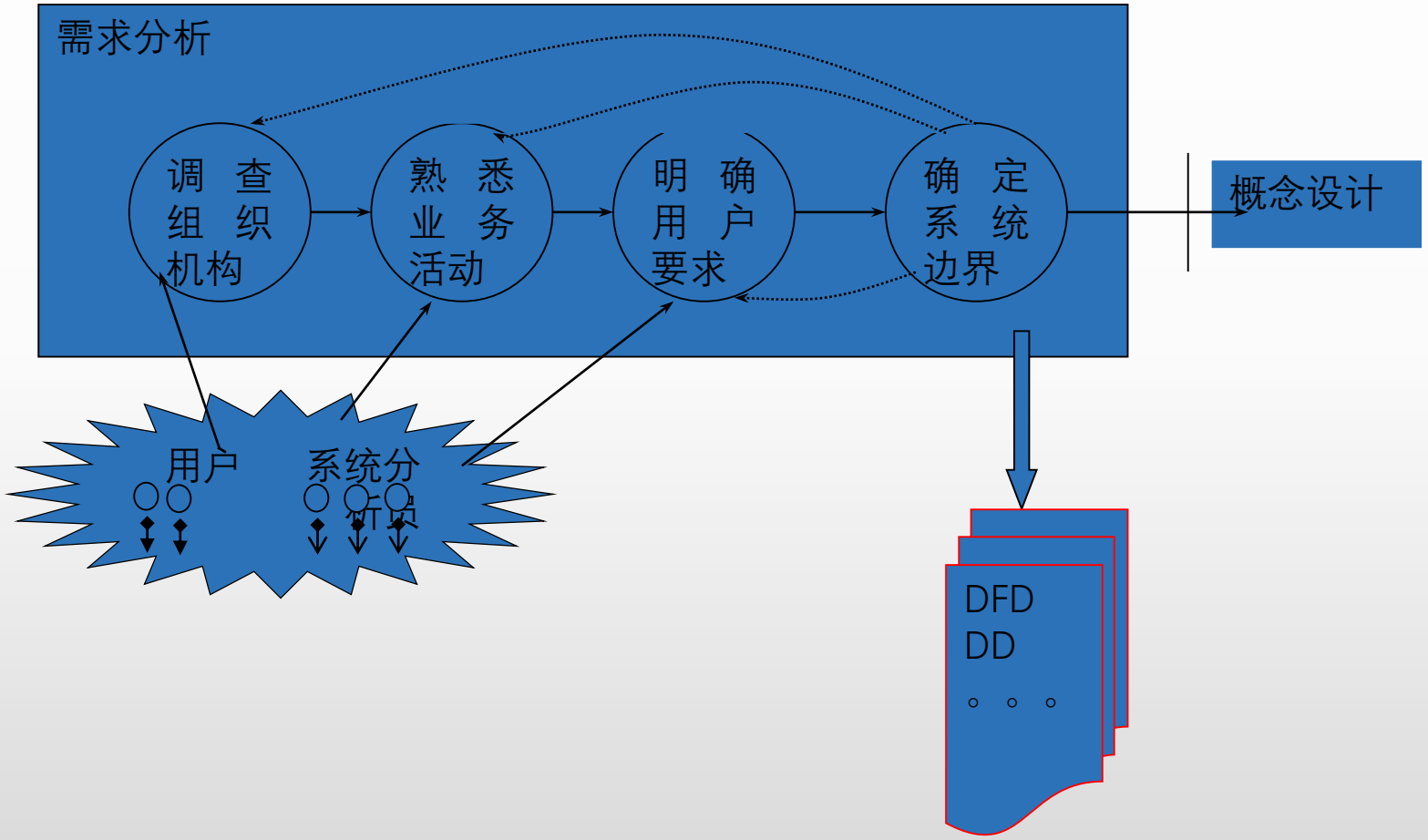


图 6-6　学籍管理的数据流图

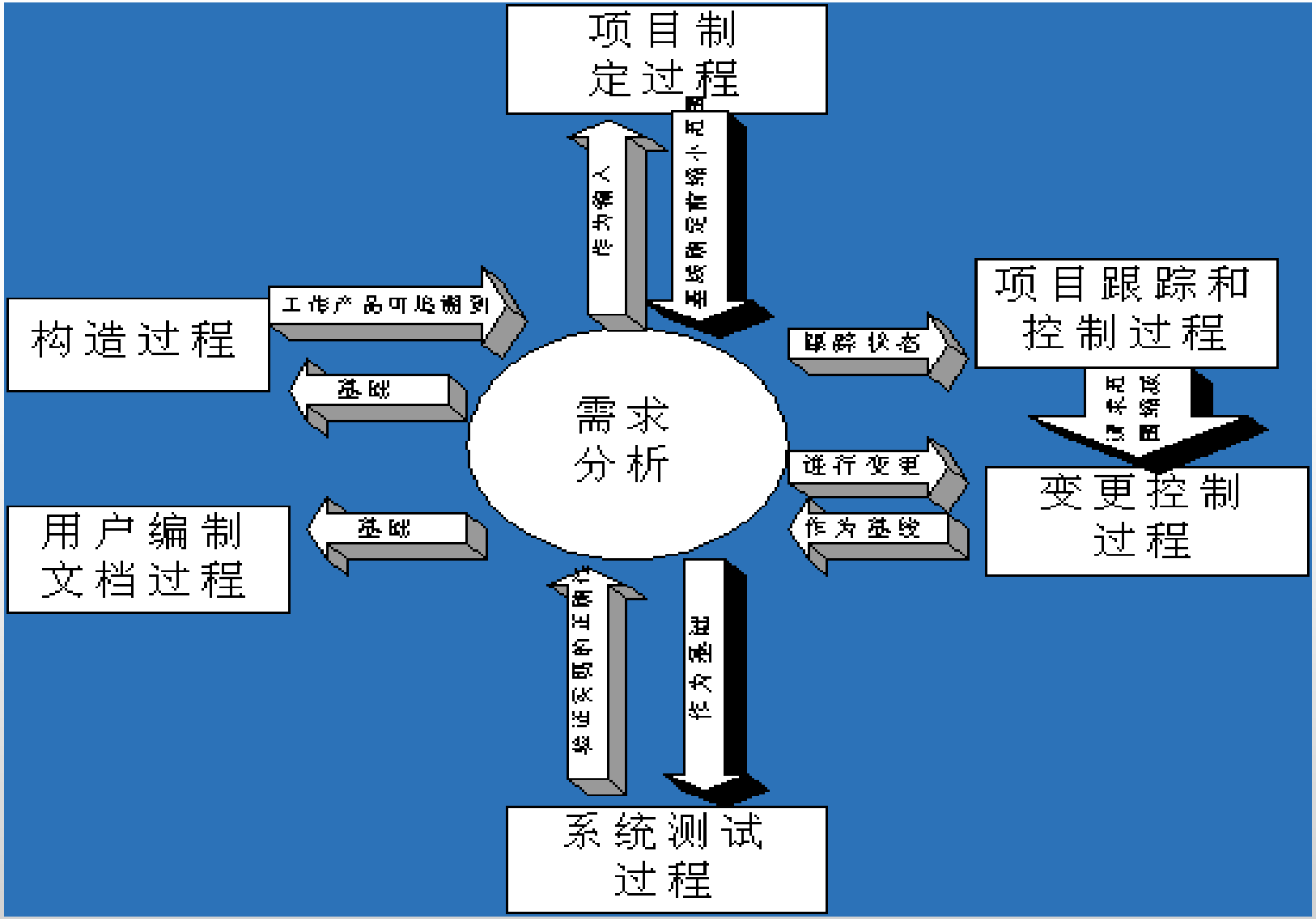
7.2.4 需求分析阶段文档

- 需求分析阶段结束时，应提供的文档包括:
  - 系统组织结构图;
  - 数据流图;
  - 数据字典;
  - 数据处理流程图等

需求分析小结



需求分析和其他项目过程的关系



7.3 概念数据库设计

- 7.3.1 概述
- 7.3.2 概念数据库设计方法与步骤
- 7.3.3 数据抽象与局部ER图设计
- 7.3.4 视图集成

7.3.1 概念数据库设计概述

- 概念设计就是将需求分析得到的用户需求抽象为信息结构，即概念模型。
- 概念模型的特点：

  (1) 语义表达能力丰富。

  (2) 易于交流和理解。概念模型是DBA、应用开发人员和用户之间的主要界面，因此，概念模型要表达自然、直观和容易理解，以便和不熟悉计算机的用户交换意见，用户的积极参与是保证数据库设计和成功的关键。

  (3) 易于修改和扩充。概念模型要能灵活地加以改变，以反映用户需求和现实环境的变化。

  (4) 易于向各种数据模型转换。概念模型独立于特定的DBMS，因而更加稳定，能方便地向关系模型、网状模型或层次模型等各种数据模型转换。

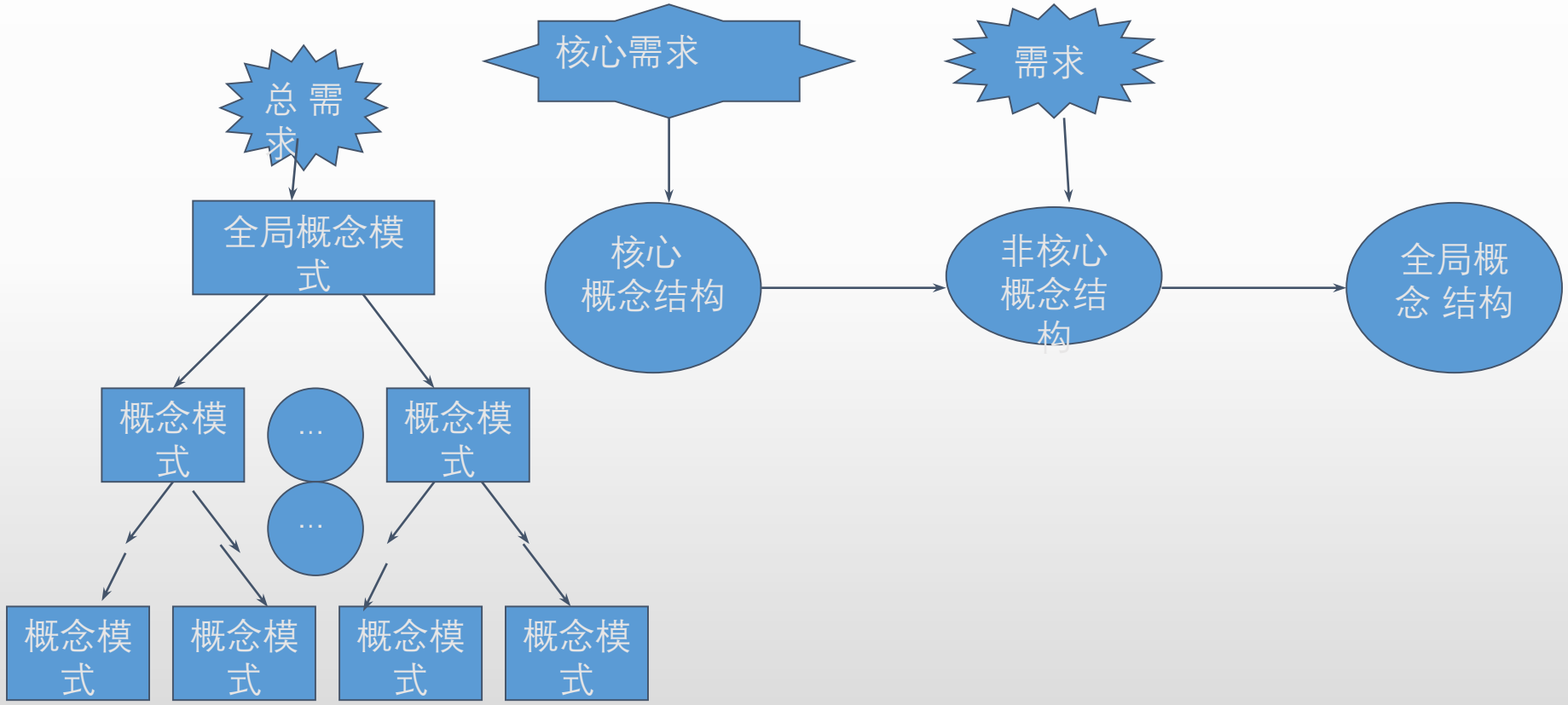
- E-R模型:它将现实世界的信息结构统一用属性、实体以及它们之间的联系来描述。

7.3.2 概念数据库设计方法与步骤

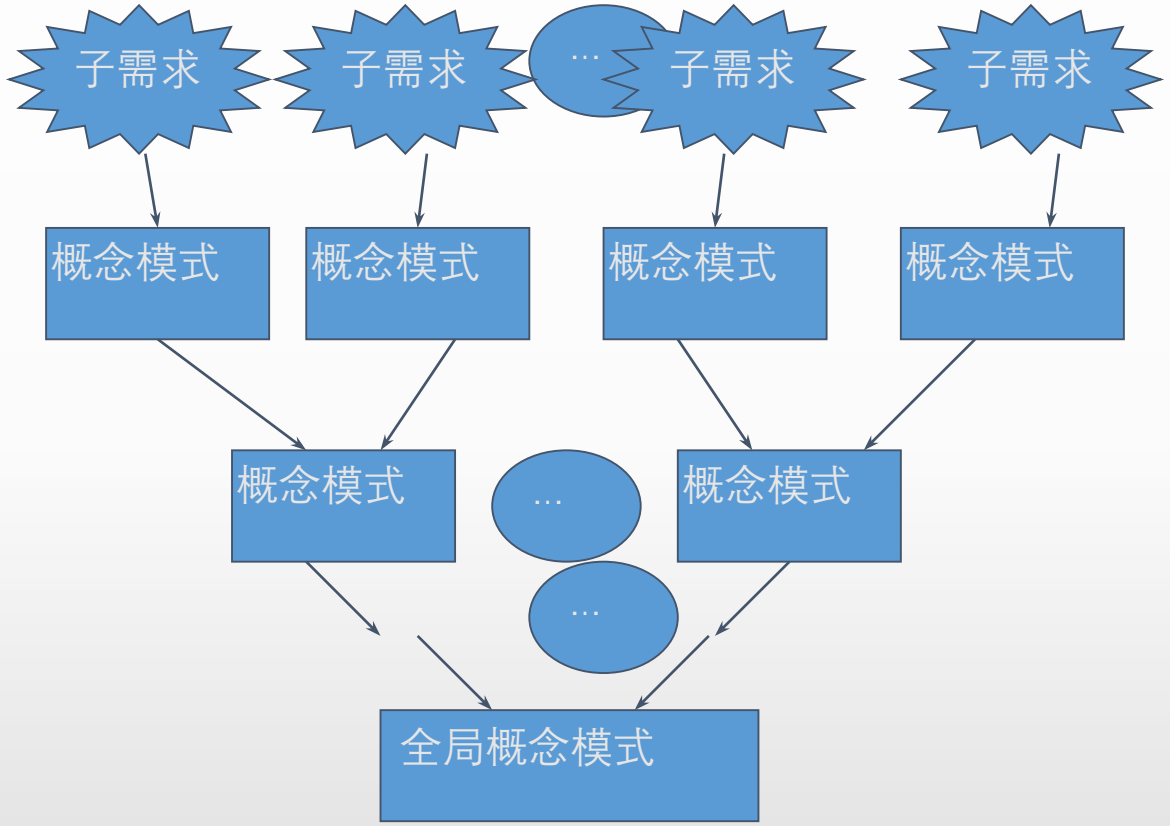
- 概念数据库设计方法与步骤

  - 自顶向下；首先定义全局的概念模型，然后逐步细化得到局部的概念模型.
  - 自底向上：首先定义各局部应用的概念结构，然后将其集成得到全局概念模型.
  - 逐步扩张：首先定义最重要的可信概念结构，然后向外扩充，逐步生成其他的概念结构和总体概念结构.
  - 混合策略

7.3.2 概念数据库设计方法与步骤



7.3.2 概念数据库设计方法与步骤

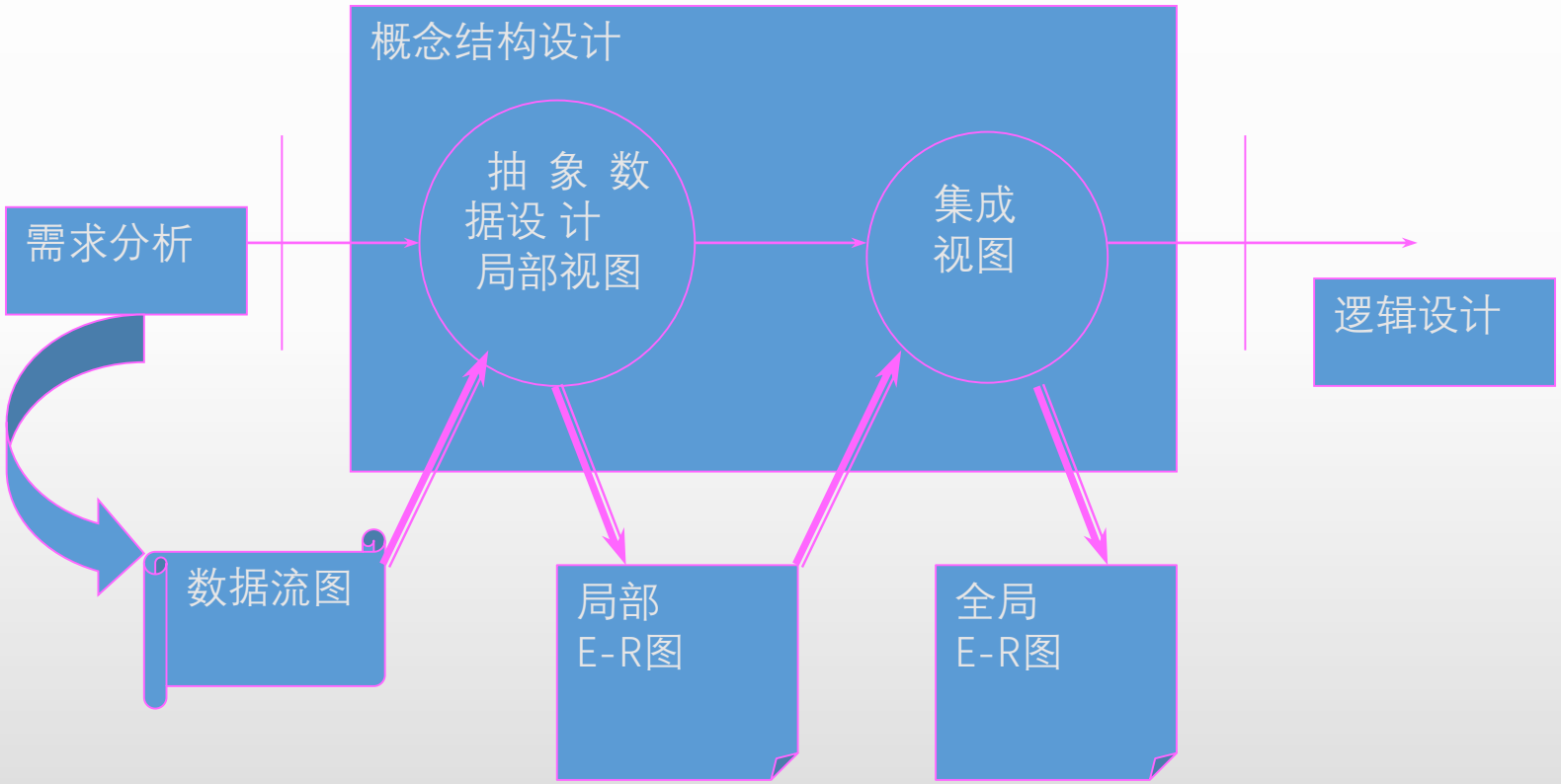


选用哪一种设计策略主要有系统分析员选择，但通常要与需求分析的策略相一致。

7.3.2 概念数据库设计方法与步骤

- 需求分析阶段，已用多层数据流图和数据字典描述了整个系统。
- 设计分E-R图首先需要根据系统的具体情况，在多层的数据流图中选择一个适当层次的数据流图，让这组图中每一部分对应一个局部应用，然后以这一层次的数据流图为出发点，设计分E-R图。
- 通常以中层数据流图作为设计分E-R图的依据。原因：
  - 高层数据流图只能反映系统的概貌
  - 中层数据流图能较好地反映系统中各局部应用的子系统组成
  - 低层数据流图过细

采用自底向上策略的设计过程示意图



7.3.3 数据抽象与局部ER图设计

- 前言
  - 任务
    - 确定局部范围(模块独立性)

      原则： 相对独立；内部联系较紧密；与 外部联系相对较少。

      来源：针对多层的数据流图，选择一个适当层次的数据流图，作为设计分E－R图的出发点。
    - 设计局部E-R图

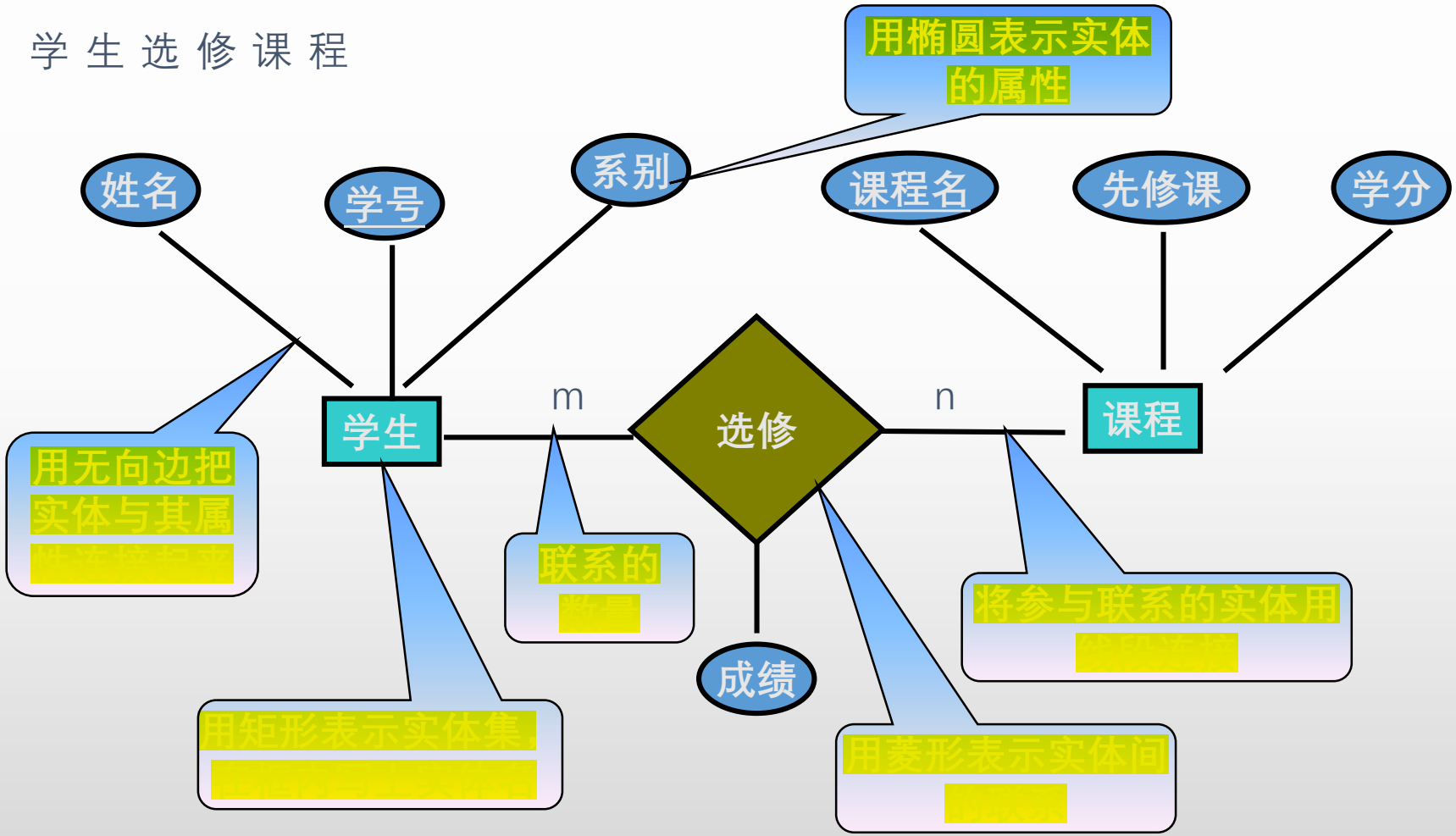
      数据抽象

7.3.3 数据抽象与局部ER图设计

- 1. ER图基本要点复习
- 2. 数据抽象
- 3. ER图设计要点
- 4. 练习

7.3.3-1　ER图基本要点复习

学 生 选 修 课 程



7.3.3-2 ER图基本概念的补充说明

- 1. 属性的类型
  - 简单属性
    - 不可再分的属性
    - 如学号、年龄、性别
  - 复合（Composite）属性
    - 可以划分为更小的属性
    - 可以把相关属性聚集起来，使模型更清晰

7.3.3-2 ER图基本概念的补充说明

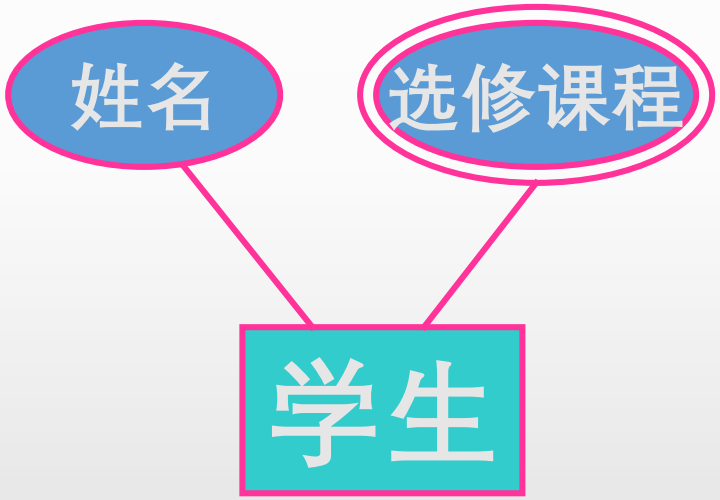
- 1. 属性的类型
  - 单值属性
    - 每一个特定的实体在该属性上的取值唯一
    - 如学生的学号，年龄、性别、系别等
  - 多值属性
    - 某个特定的实体在该属性上的有多于一个的取值
    - 如学生（学号，所选课程，联系电话）

7.3.3-2 ER图基本概念的补充说明

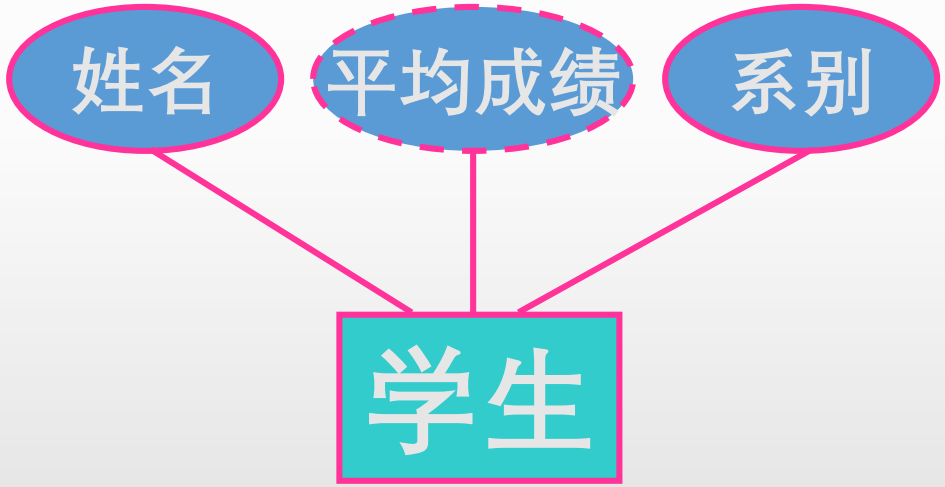
- 1. 属性的类型
  - 派生属性与基属性
    - 派生属性可以从其他相关的属性或实体派生出来的属性值
    - 数据库中，一般只存基属性值，而派生属性只存其定义或依赖关系，用到时再从基属性中计算出来
    - 勿轻易引入派生属性

属性在E-R图中的表示

多值属性用**双椭圆**表示



派生属性用**虚椭圆**表示



7.3.3-2 数据抽象

- 数据抽象

  在系统需求分析阶段，最后得到了多层数据流图、数据字典和系统分析报告。现在就是要根据需求分析，对实际的人、物、事和概念进行人为的处理，抽取所关心的共同特性，忽略非本质的细节，并把各种概念精确地加以描述，利用这些概念形成某种模型。

- 数据抽象的种类
  - 分类(is member of)
  - 聚集(is part of)
  - 概括(is subset of)

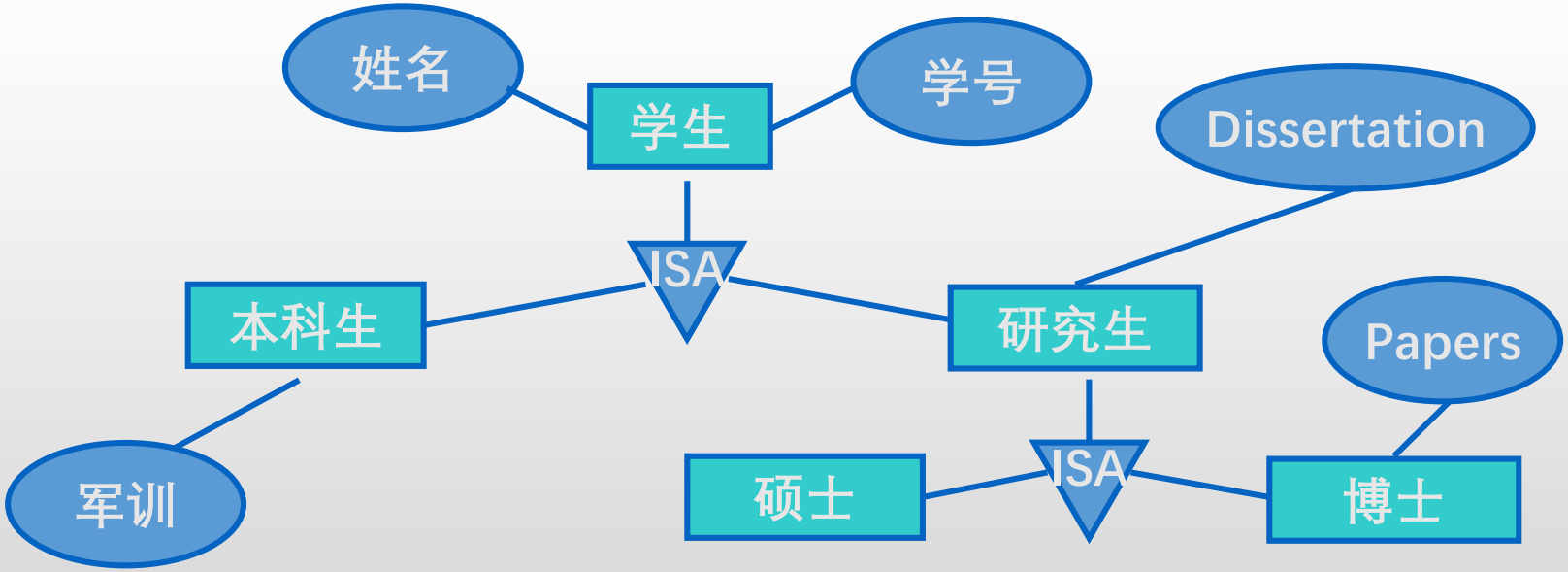
} 确定实体及其属性

- 分类（Classification）
  - 分类定义某一类概念作为现实世界中一组对象的类型，将一组具有某些共同特性和行为的对象抽象为一个实体。对象和实体之间是“is member of”的关系。
  - 例如，在教学管理中，“赵亦”是一名学生，表示“赵亦”是学生中的一员，她具有学生们共同的特性和行为。
  - 又如，对于某销售系统，张三是系统中某用户，他行使采购的功能，李四也是系统中某用户，他行使财务结算的功能。采购和财务结算实际上是一种权限，因此我们抽象出角色，用来表示系统中行使不同权限的用户。

- 聚集（Aggregation）
  - 聚集定义某一类型的组成成份，将对象类型的组成成份抽象为实体的属性。组成成份与对象类型之间是“is part of”的关系。
  - 例如，学号、姓名、性别、年龄、系别等可以抽象为学生实体的属性。

某些扩展ER特性的表示

- 概括
  - 用标记为ISA的三角形来表示，表示高层实体和低层实体之间的“父类－子类”联系（或教材P217方法），它定义了型之间的一种子集联系



7.3.3-3 ER图设计要点

1 确定实体与属性

任务：① 命名 ② 确定实体码 ③ 确定实体内属性

方法

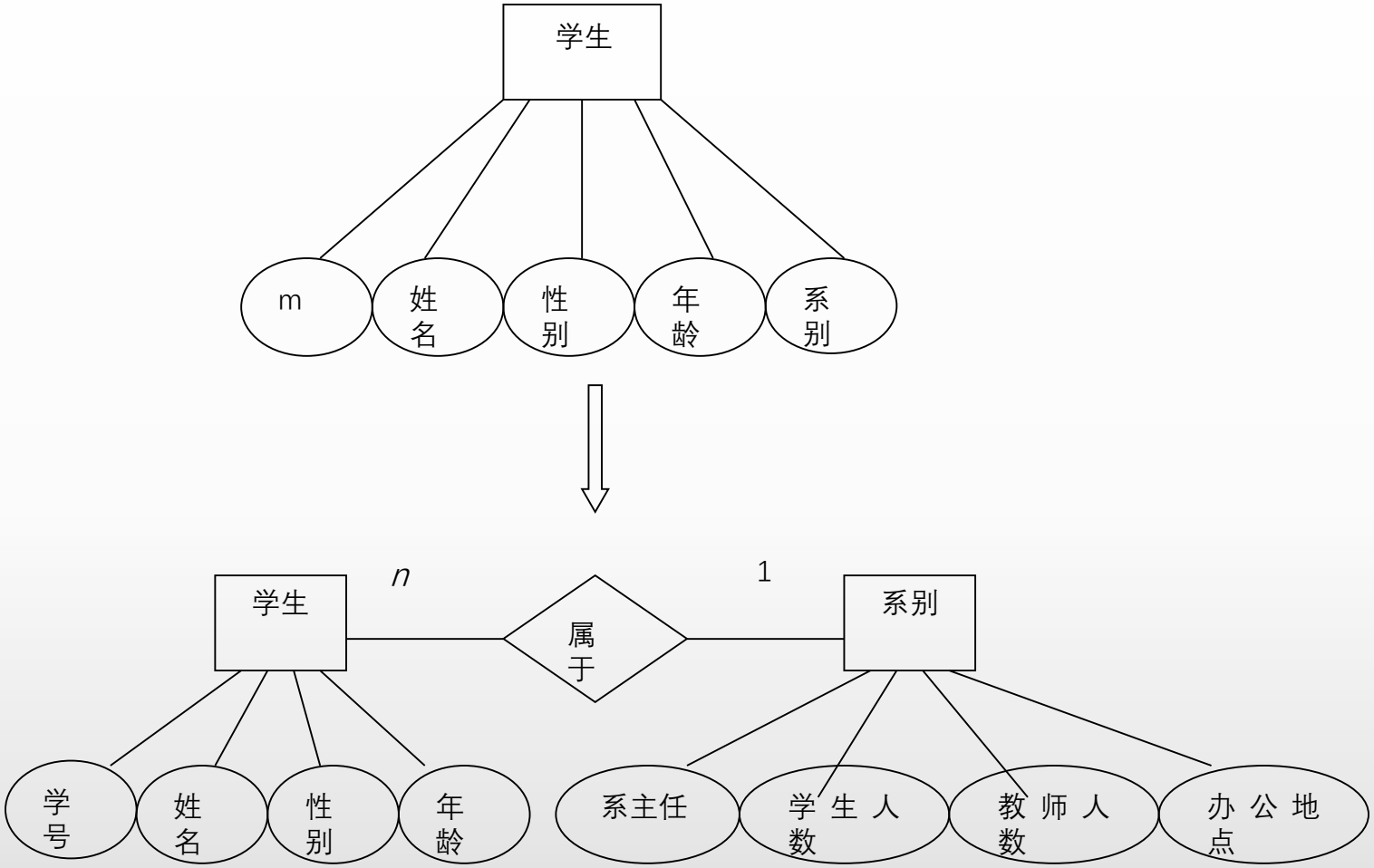
① 以需求分析说明书中DD的数据结构为基础点。

② 可再分者为实体

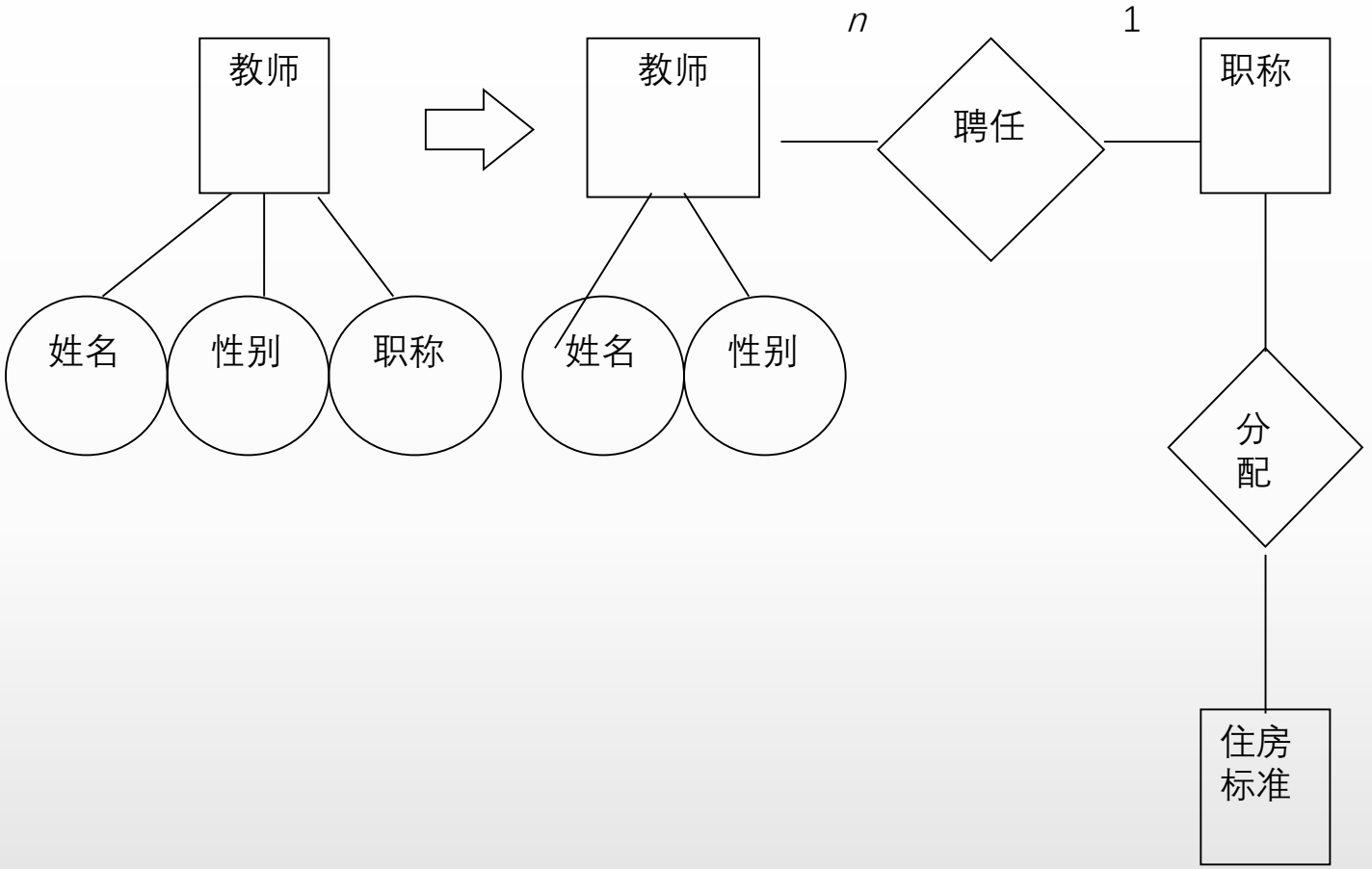
- 年龄显然作属性。
- 单位若还可细分为单位号DH、单位名称DM、单位地址DD，则为实体。

③ 实体内部属性不能再与其它实体有联系。

- 同一实体内两属性间可有联系，但一般不应与其它实体发生联系，这种联系应该是实体间。



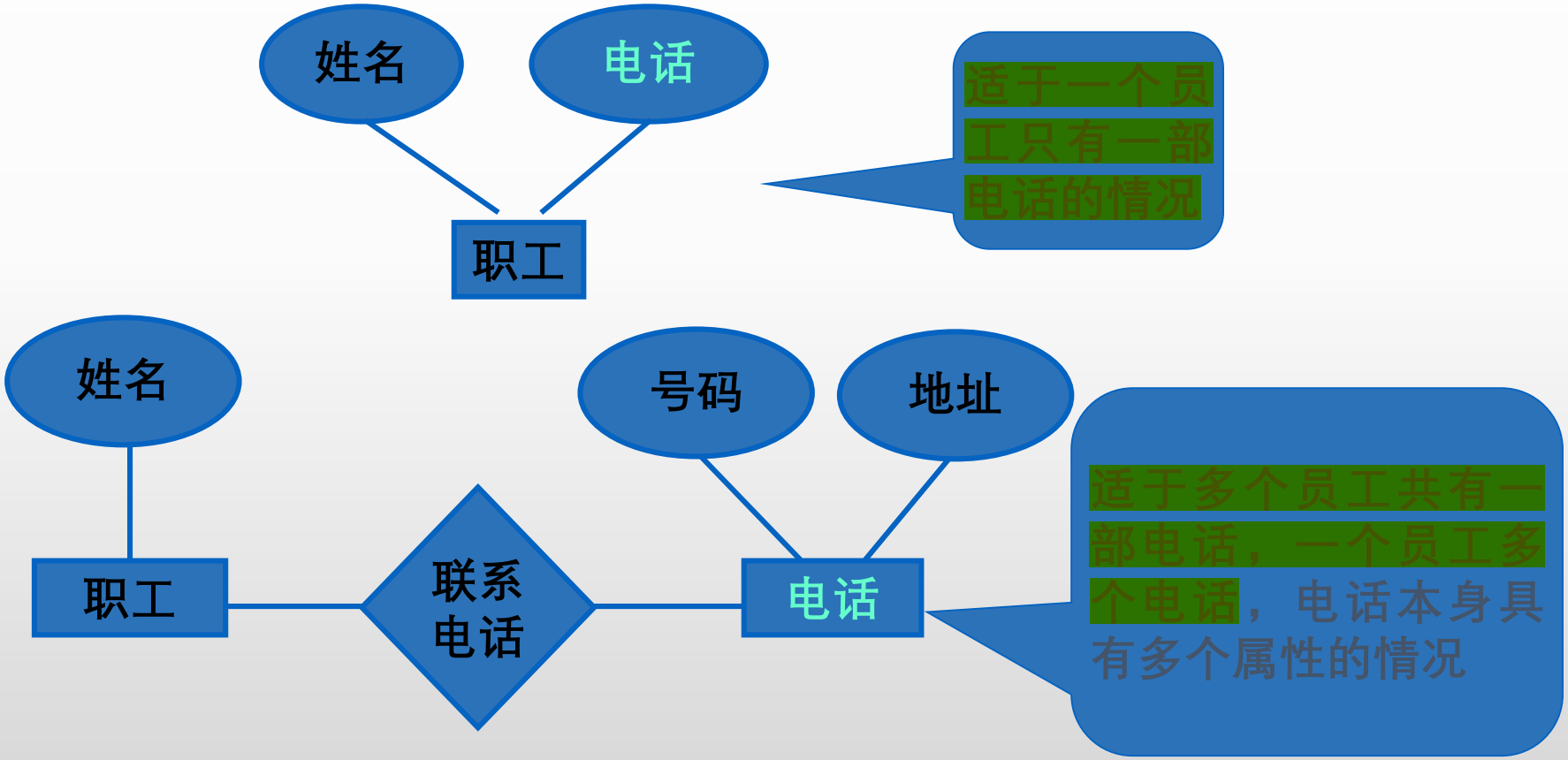
系别作为一个属性或实体



职称作为一个属性或实体

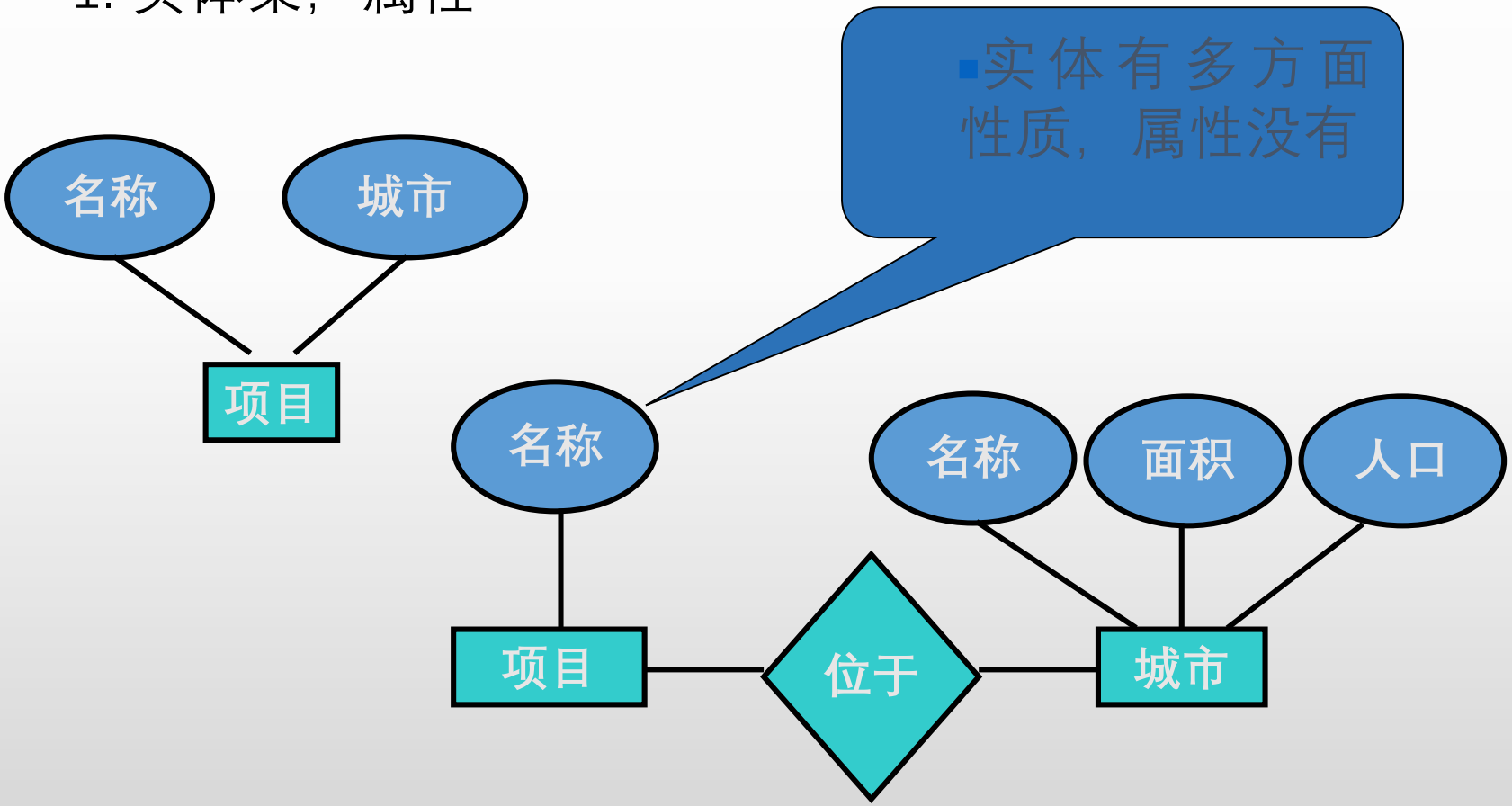
7.3.3-3 ER图设计要点

- 1. 实体集，属性



7.3.3-3 ER图设计要点(续)

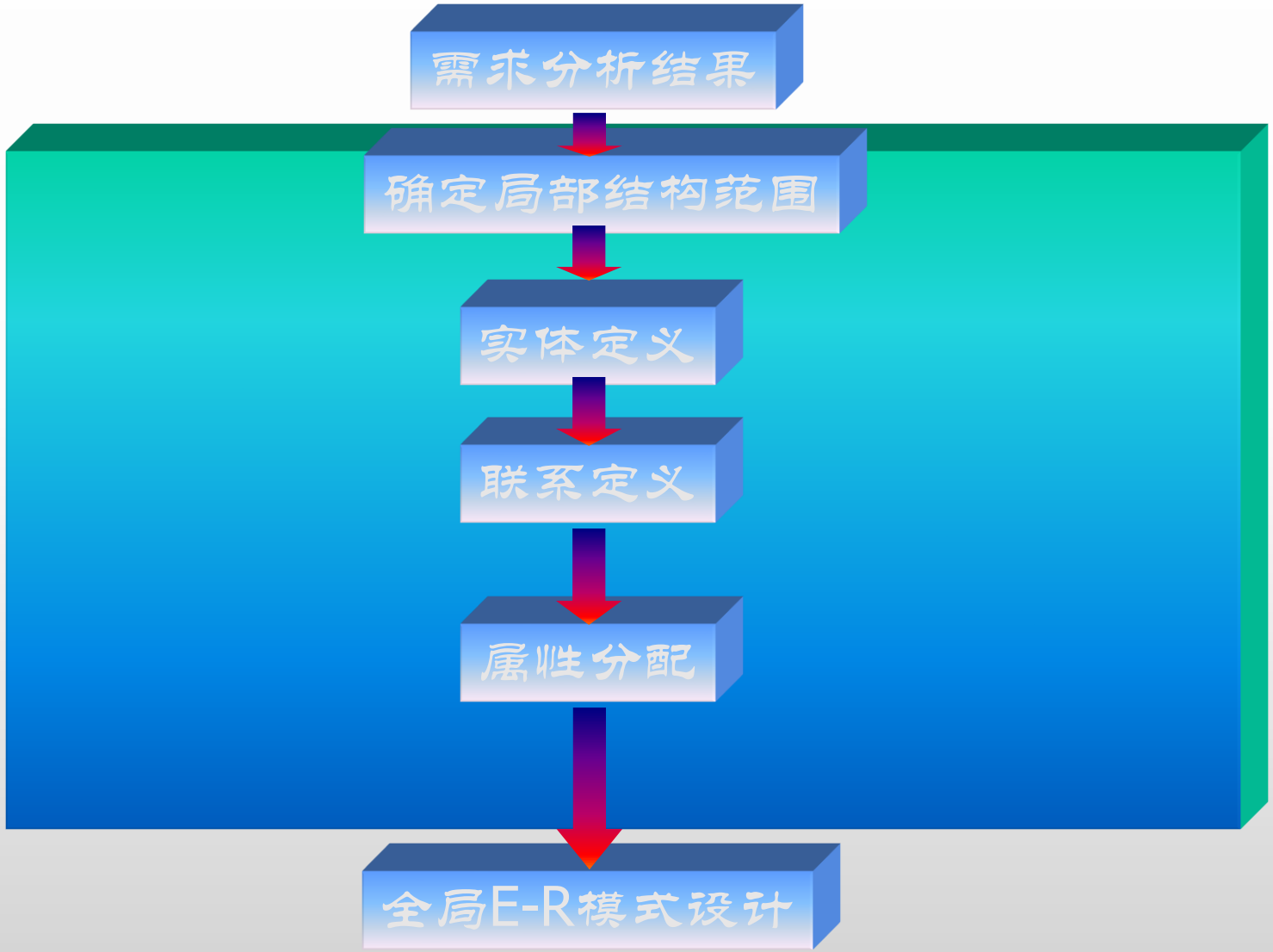
- 1. 实体集，属性



7.3.3-3 ER图设计要点(续)

- 2. 确定联系
  - 存在性联系
    - 系有学生，课程有成绩
  - 功能性联系
    - 教师教学生，工程师参与工程
  - 事件联系
    - 顾客发出订单，学生借书

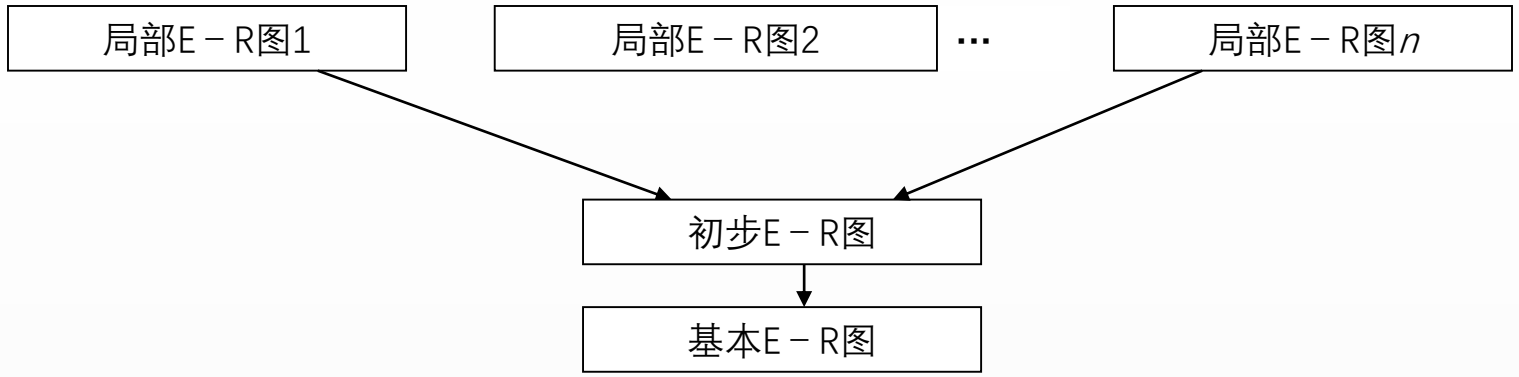
局部ER图设计总结



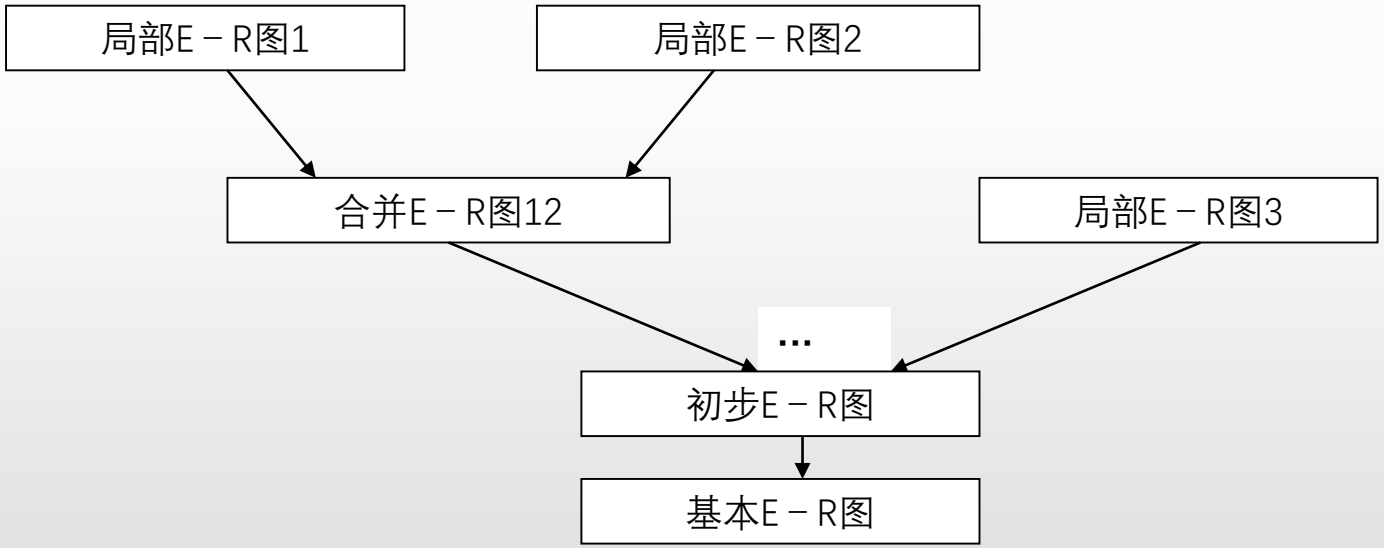
7.3.4 视图集成

①多元集成法，一次性将多个局部E-R图合并为一个全局E-R图。
②二元集成法，首先集成两个重要的局部视图，以后用累加的方法逐步将一个新的视图集成进来。在实际应用中，可以根据系统复杂性选择这两种方案。一般采用逐步集成的方法，如果局部视图比较简单，可以采用多元集成法。一般情况下，采用二元集成法，即每次只综合两个视图，这样可降低难度。无论使用哪一种方法，视图集成均分成两个步骤：

- 合并.消除各局部E-R图之间的冲突，生成初步E-R图。
- 优化.消除不必要的冗余，生成基本E-R图。



(a)　多元集成法



(b)　二元集成法

(1) 合并局部E-R图，生成初步E-R图

- 将所有的局部E-R图综合成全局概念结构。
- 全局概念结构它不仅要支持所有的局部E-R模型，而且必须合理地表示一个完整、一致的数据库概念结构。
- 由于各个局部应用不同，通常由不同的设计人员进行局部E-R图设计，因此，各局部E-R图不可避免地会有许多不一致的的地方，我们称之为冲突。
- 合并局部E-R图时并不能简单地将各个E-R图画到一起，而必须消除各个局部E-R图中的不一致，使合并后的全局概念结构不仅支持所有的局部E-R模型，而且必须是一个能为全系统中所有用户共同理解和接受的完整的概念模型。
- 合并局部E-R图的关键就是合理消除各局部E-R图中的冲突。

E-R图中的冲突有三种：属性冲突、命名冲突和结构冲突。

①属性冲突

- 属性冲突又分为属性值域冲突和属性的取值单位冲突。

  a.属性值域冲突，即属性值的类型、取值范围或取值集合不同。比如学号，有些部门将其定义为数值型，而有些部门将其定义为字符型。又如年龄，有的可能用出生年月表示，有的则用整数表示。

  b.属性的取值单位冲突。比如零件的重量，有的以公斤为单位，有的以斤为单位，有的则以克为单位。

- 属性冲突属于用户业务上的约定，必须与用户协商后解决。

②命名冲突

- 命名不一致可能发生在实体名、属性名或联系名之间，其中属性的命名冲突更为常见。
- 一般表现为同名异义或异名同义（实体、属性、联系名）。

  a.同名异义，即同一名字的对象在不同的部门中具有不同的意义。

  例，局部应用A中将教室称为房间

  局部应用B中将学生宿舍称为房间

  b.异名同义，即同一意义的对象在不同的部门中具有不同的名称。

  比如，对于“科研项目”这个名称，在财务科称为项目，在科研处称为课题，在生产管理处称为工程。
- 命名冲突的解决方法同属性冲突，需要与各部门协商、讨论后加以解决。

③有三类结构冲突

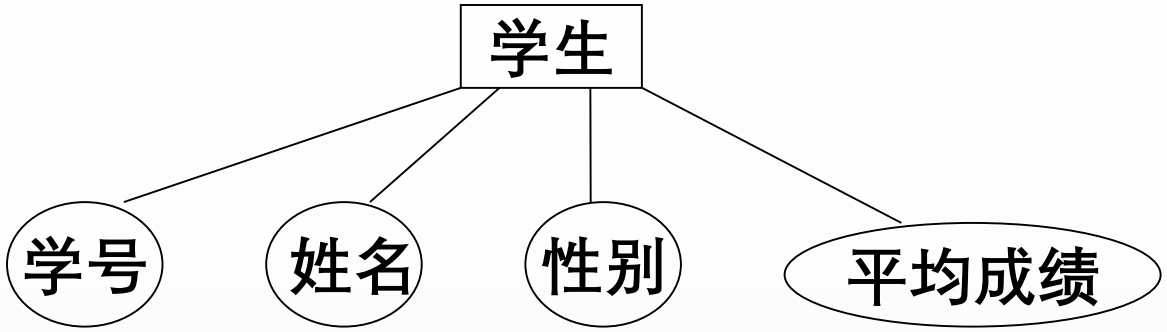
- 同一对象在不同应用中具有不同的抽象

例，“课程”在某一局部应用中被当作实体　　在另一局部应用中则被当作属性

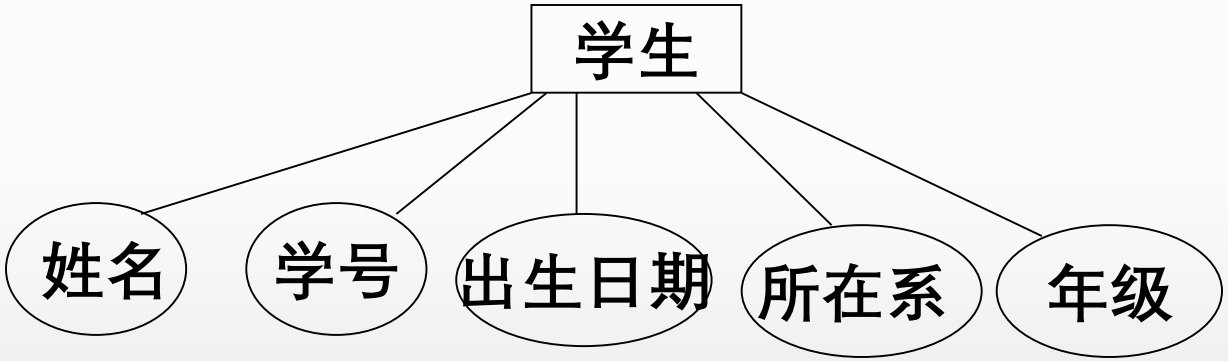
- 解决方法：通常是把属性变换为实体或把实体变换为属性，使同一对象具有相同的抽象。但是要遵照局部视图设计中提到的两个原则。

③有三类结构冲突

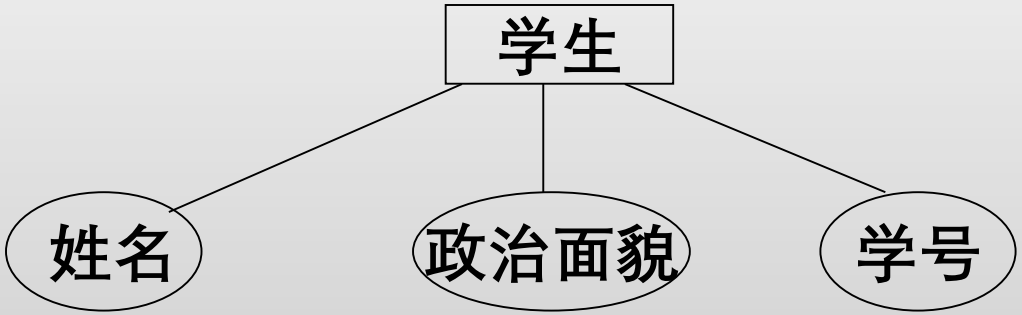
- 同一实体在不同局部视图中所包含的属性不完全相同，或者属性的排列次序不完全相同。
  - 产生原因：不同的局部应用关心的是该实体的不同侧面。
  - 解决方法：使该实体的属性取各分E-R图中属性的并集，再适当设计属性的次序



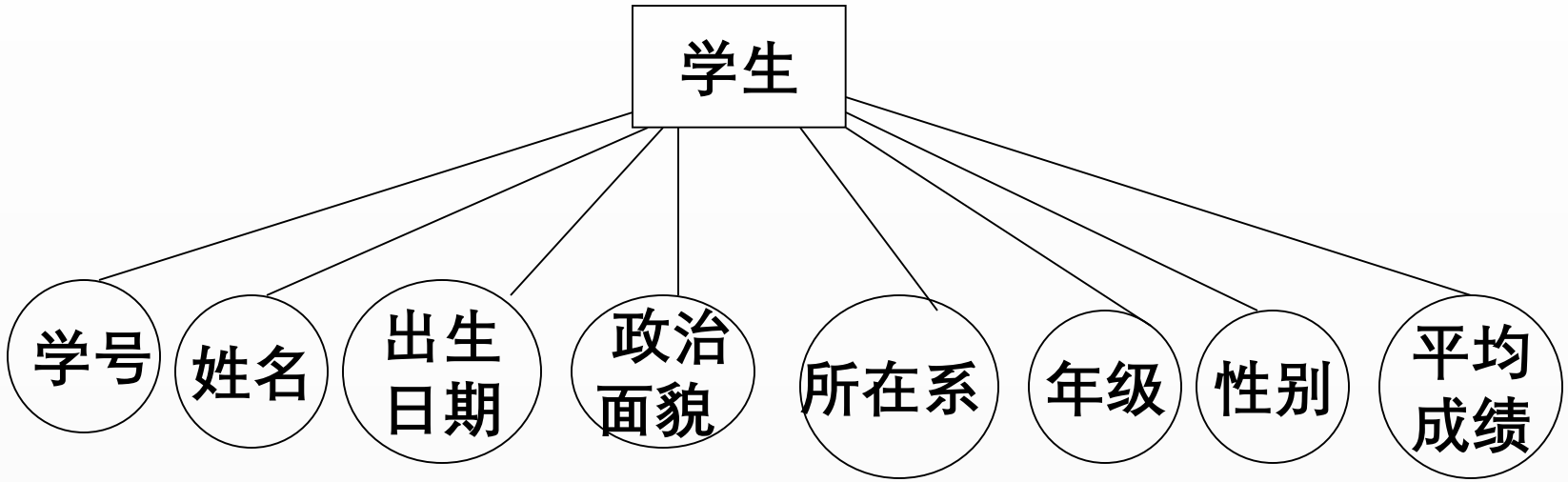
(a)在局部应用A中



(b)在局部应用B中



(c)在局部应用C中



(d)合并后

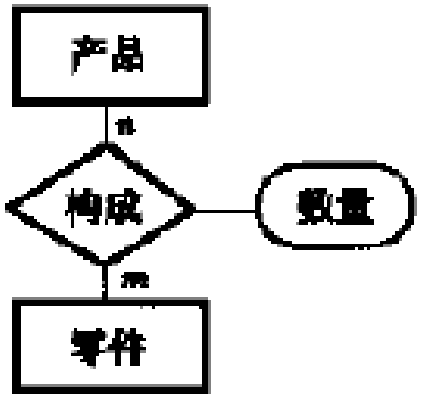
③有三类结构冲突

- 实体之间的联系在不同局部视图中呈现不同的类型

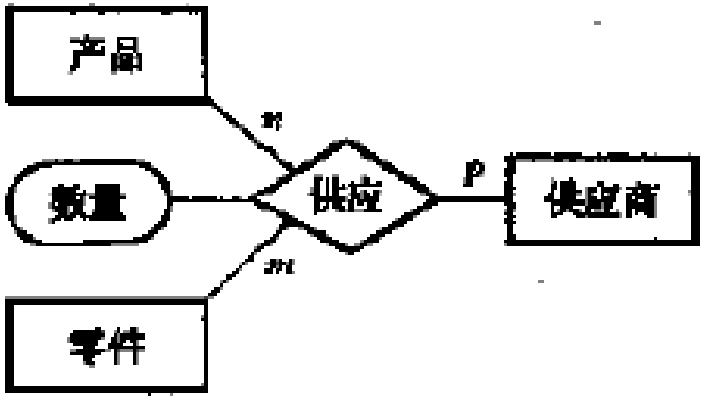
例1， 实体E1与E2在局部应用A中是多对多联系，而在局部应用B中是一对多联系

例2， 在局部应用X中E1与E2发生联系，而在局部应用Y中E1、E2、E3三者之间有联系。

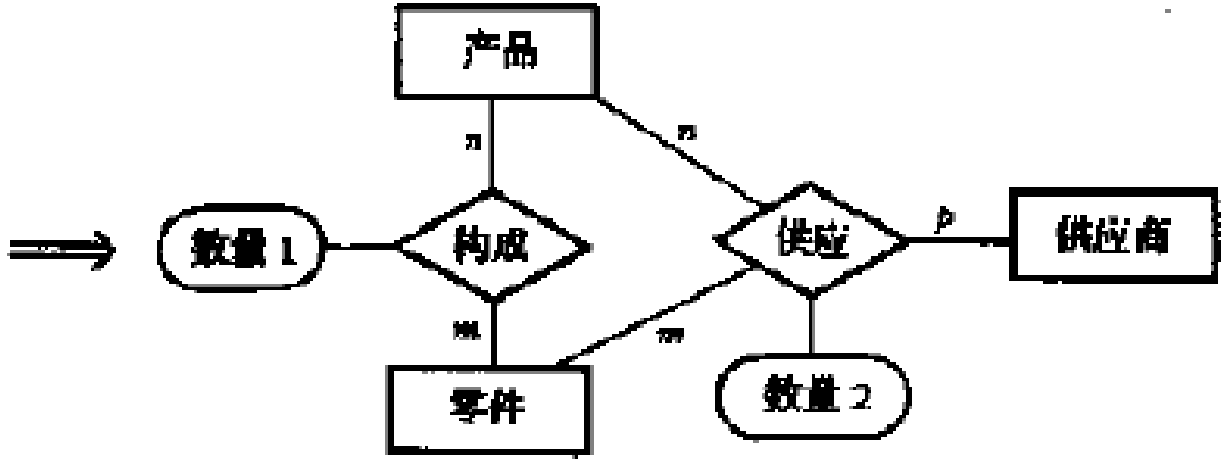
- 解决方法：根据应用语义对实体联系的类型进行综合或调整。



(E-R)1



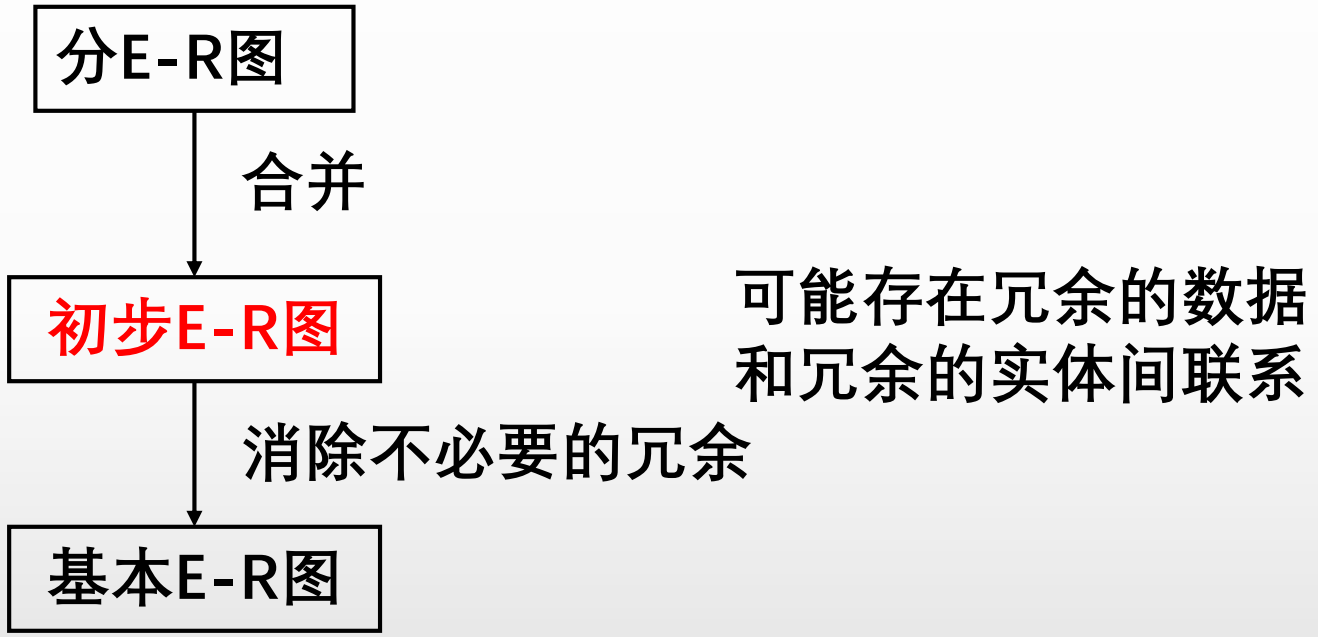
(E-R)2



(E-R)12

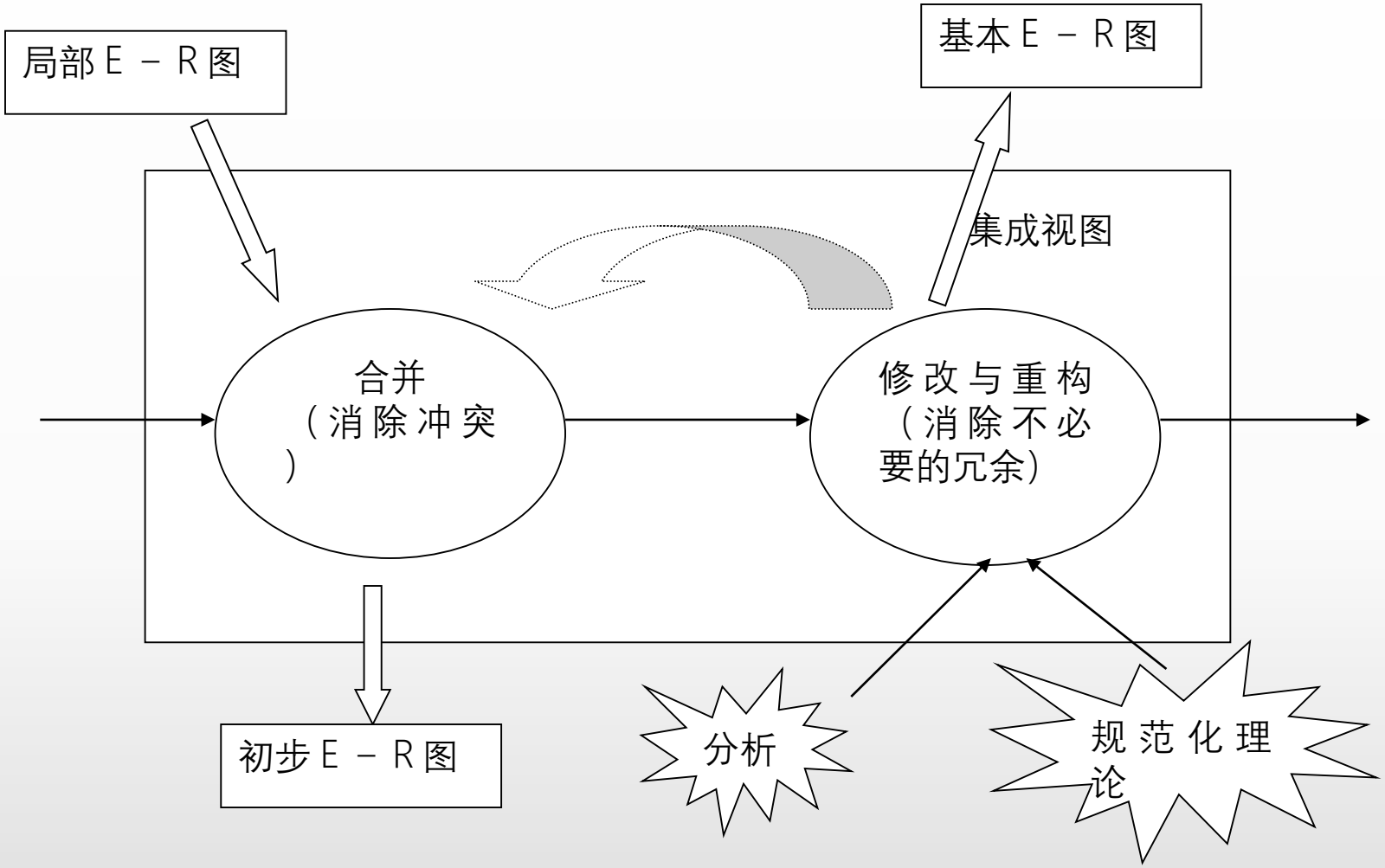
（2）修改与重构

- 基本任务：消除不必要的冗余，设计生成基本E-R图

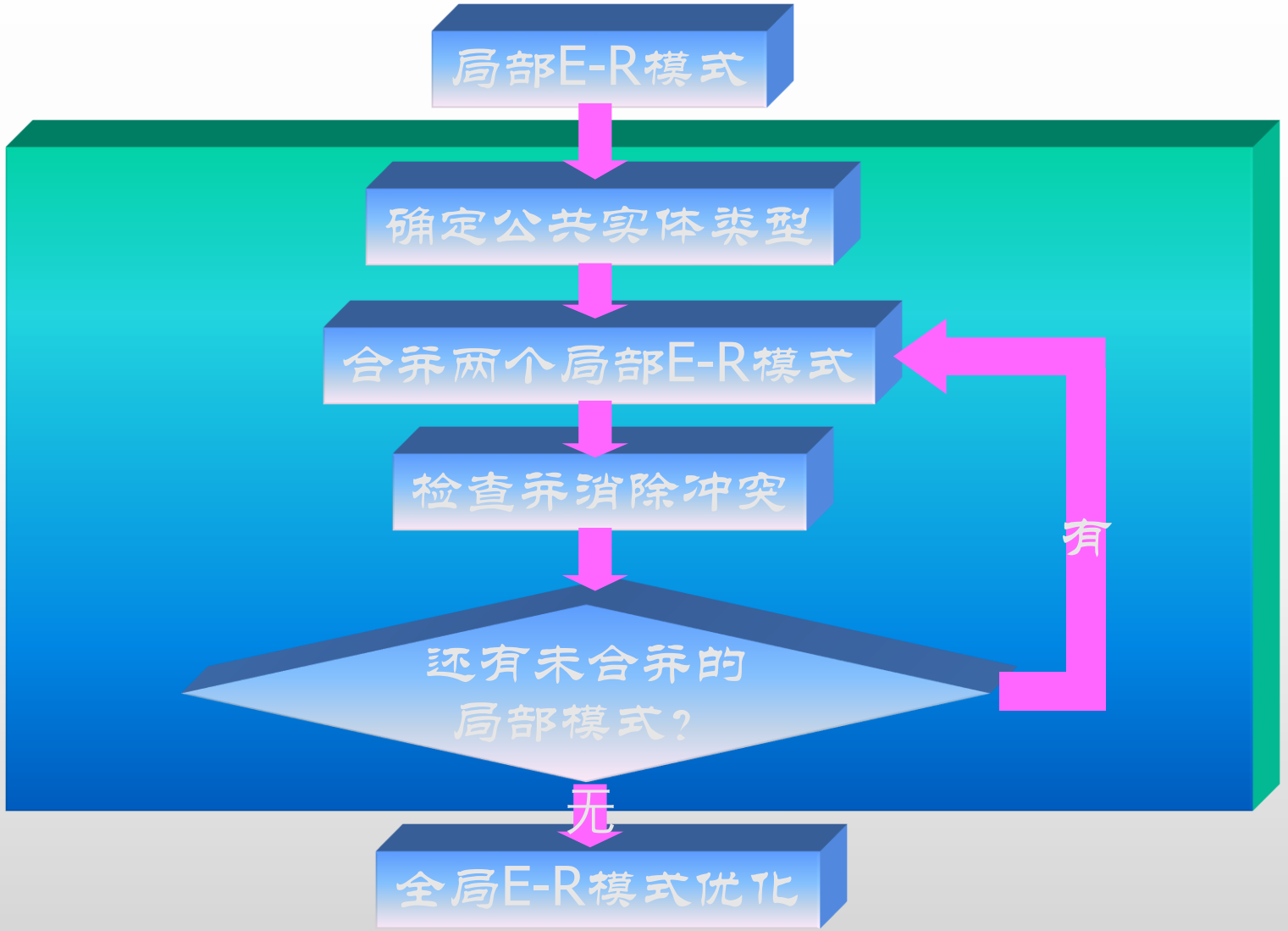


（2）修改与重构

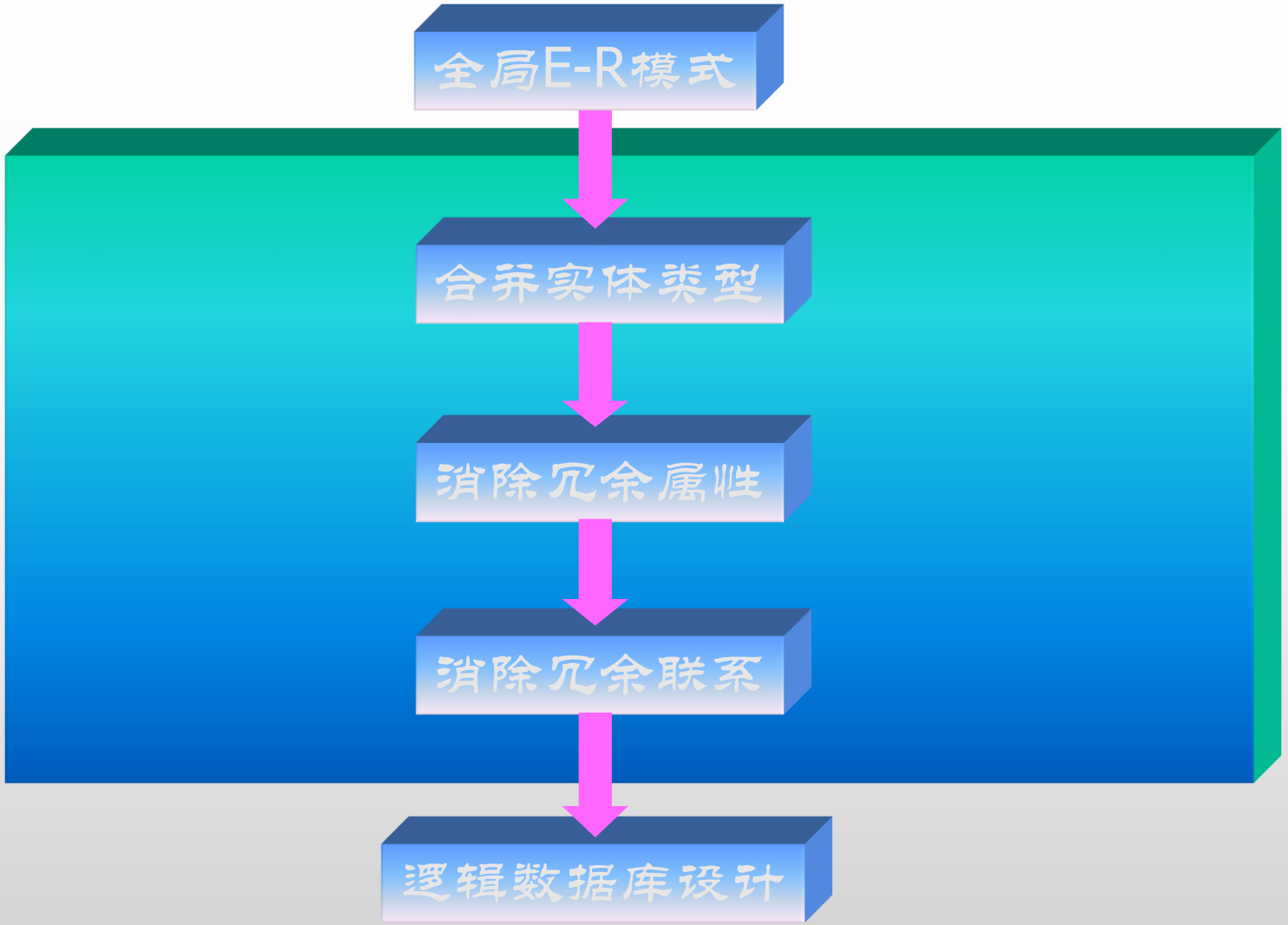
- 函数依赖的概念提供了消除冗余的形式化工具。


局部E－R图
基本E－R图
集成视图
合并
（消除冲突）
修改与重构
（消除不必要的冗余）
初步E－R图
分析
规范化理论

全局ER图设计 1


局部E-R模式
确定公共实体类型
合并两个局部E-R模式
检查并消除冲突
还有未合并的
局部模式？
有
无
全局E-R模式优化

7.3.4 – 全局ER图模式优化

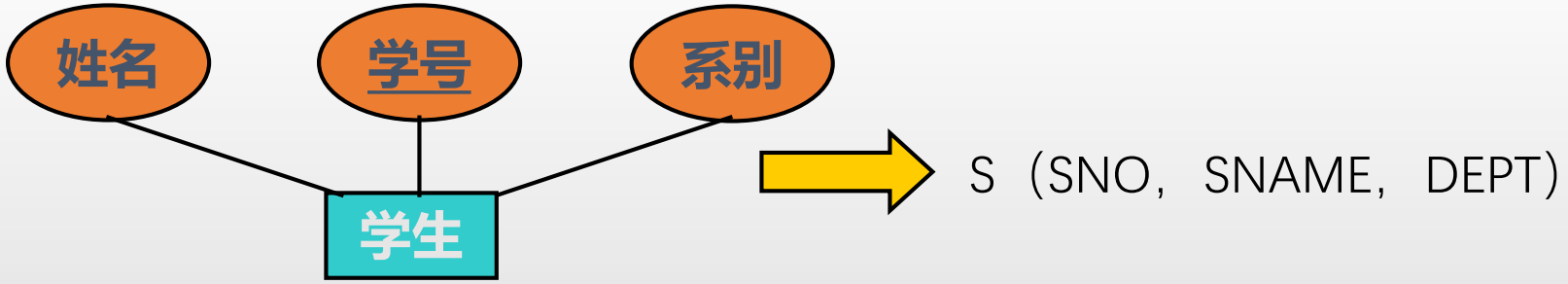


7.4 逻辑数据库设计

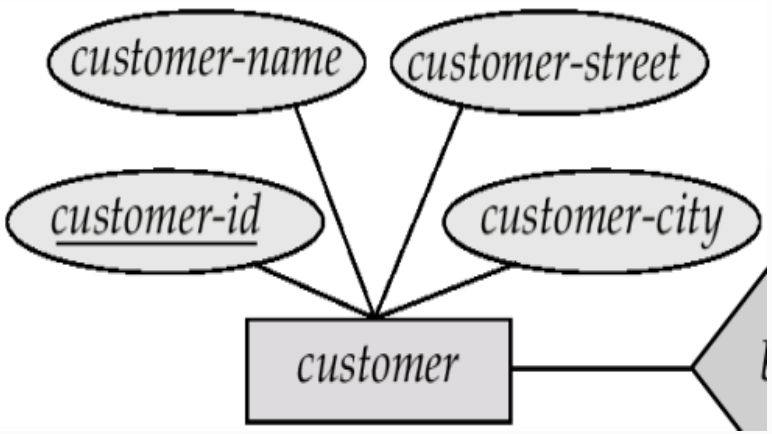
- 逻辑结构设计的任务
  - 概念结构是各种数据模型的共同基础
  - 为了能够用某一DBMS实现用户需求，还必须将概念结构进一步转化为相应的数据模型，这正是数据库逻辑结构设计所要完成的任务。
- 逻辑结构设计的步骤
  - 将概念结构转化为一般的关系模型
  - 对数据模型进行优化
- 转换内容
  - E-R图由实体、实体的属性和实体之间的联系三个要素组成
  - 关系模型的逻辑结构是一组关系模式的集合
  - 将E-R图转换为关系模型：将实体、实体的属性和实体之间的联系转化为关系模式。

7.4.3 ER图向关系模型的转换

- 1. 实体和属性
  - 实体 → 关系
  - 属性 → 关系的属性



7.4.3 -1 实体和属性



customer-id	customer-name	customer-street	customer-city
019-28-3746	Smith	North	Rye
182-73-6091	Turner	Putnam	Stamford
192-83-7465	Johnson	Alma	Palo Alto
244-66-8800	Curry	North	Rye
321-12-3123	Jones	Main	Harrison
335-57-7991	Adams	Spring	Pittsfield
336-66-9999	Lindsay	Park	Pittsfield
677-89-9011	Hayes	Main	Harrison
963-96-3963	Williams	Nassau	Princeton

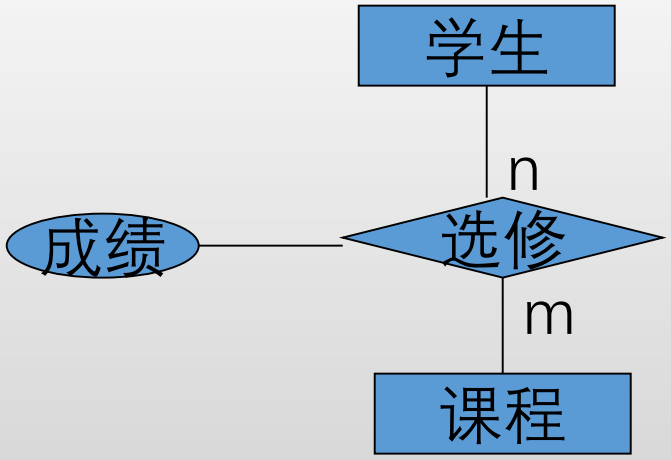
7.4.3 ER图向关系模型的转换(续)

2. 一个m:n联系转换为一个关系模式。

- *关系的属性*：与该联系相连的各实体的码以及联系本身的属性
- *关系的码*：各实体码的组合

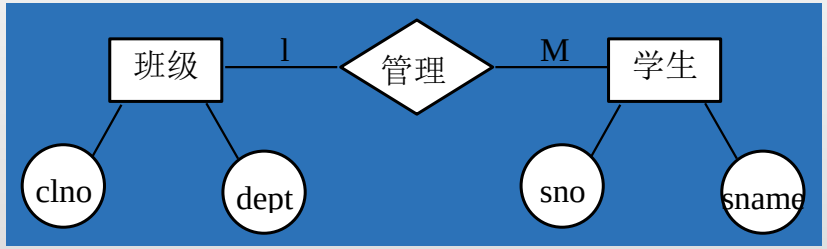
例，“选修”联系是一个m:n联系，可以将它转换为如下关系模式，其中学号与课程号为关系的组合码：

选修（学号，课程号，成绩）



7.4.3 ER图向关系模型的转换(续)

- 3. 一个1:n联系可以转换为一个独立的关系模式，也可以与n端对应的关系模式合并。
  - 1) 转换为一个独立的关系模式
    - *关系的属性*：与该联系相连的各实体的码以及联系本身的属性
    - *关系的码*：n端实体的码



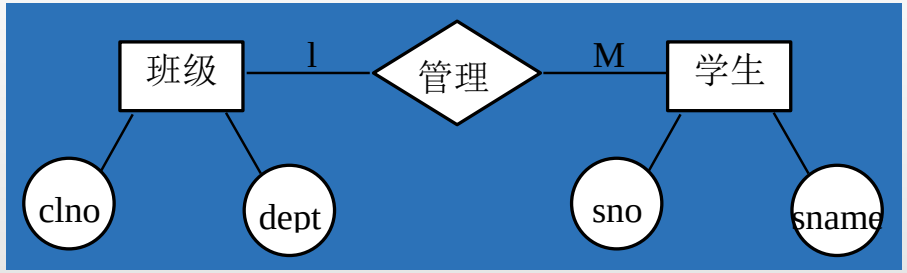
管理（<u>sno</u>,clno）

7.4.3 ER图向关系模型的转换(续)

2) 与n端对应的关系模式合并

- *合并后关系的属性*：在n端关系中加入1端关系的码和联系本身的属性
- *合并后关系的码*：不变

- 可以减少系统中的关系个数，一般情况下更倾向于采用这种方法



学生（sno，sname，clno）

7.4.3 ER图向关系模型的转换(续)

4. 一个1:1联系可以转换为一个独立的关系模式，也可以与任意一端对应的关系模式合并。

- 1) 转换为一个独立的关系模式
  - *关系的属性*：与该联系相连的各实体的码以及联系本身的属性
  - *关系的候选码*：每个实体的码均是该关系的候选码

7.4.3 ER图向关系模型的转换(续)

- 2) 与某一端对应的关系模式合并
  - *合并后关系的属性*：加入对应关系的码和联系本身的属性
  - *合并后关系的码*：不变

7.4.3 ER图向关系模型的转换(续)

注意：

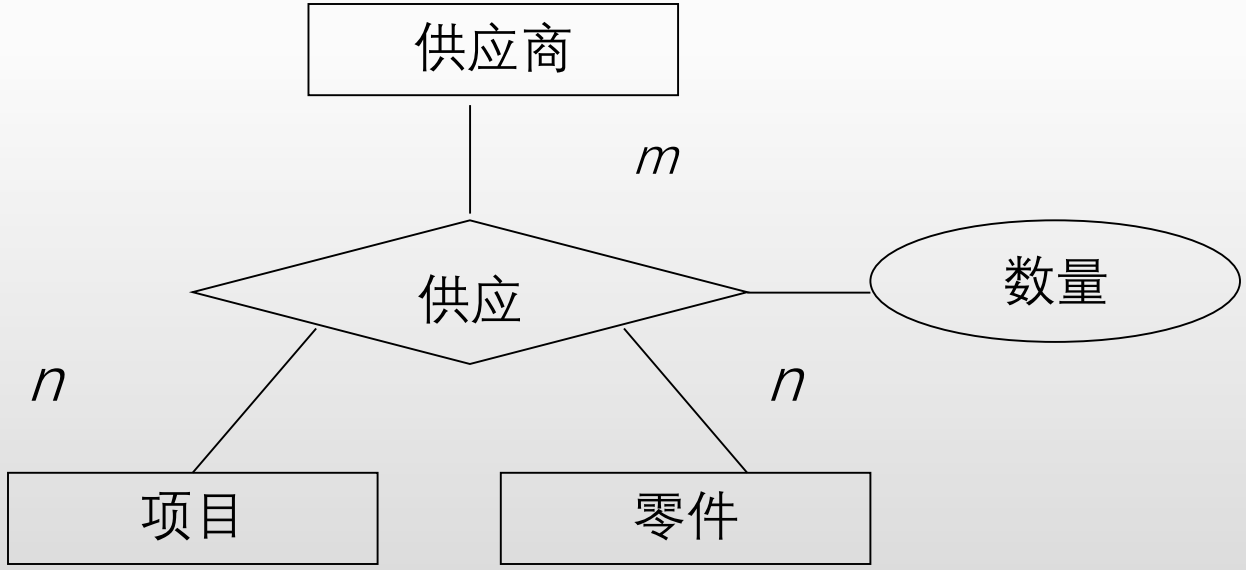
从理论上讲，1:1联系可以与任意一端对应的关系模式合并。

但在一些情况下，与不同的关系模式合并效率会大不一样。因此究竟应该与哪端的关系模式合并需要依应用的具体情况而定。

由于连接操作是最费时的操作，所以一般应以尽量减少连接操作为目标。

7.4.3 ER图向关系模型的转换(续)

5. 三个或三个以上实体间的一个多元联系转换为一个关系模式。
   - *关系的属性*：与该多元联系相连的各实体的码以及联系本身的属性
   - *关系的码*：各实体码的组合



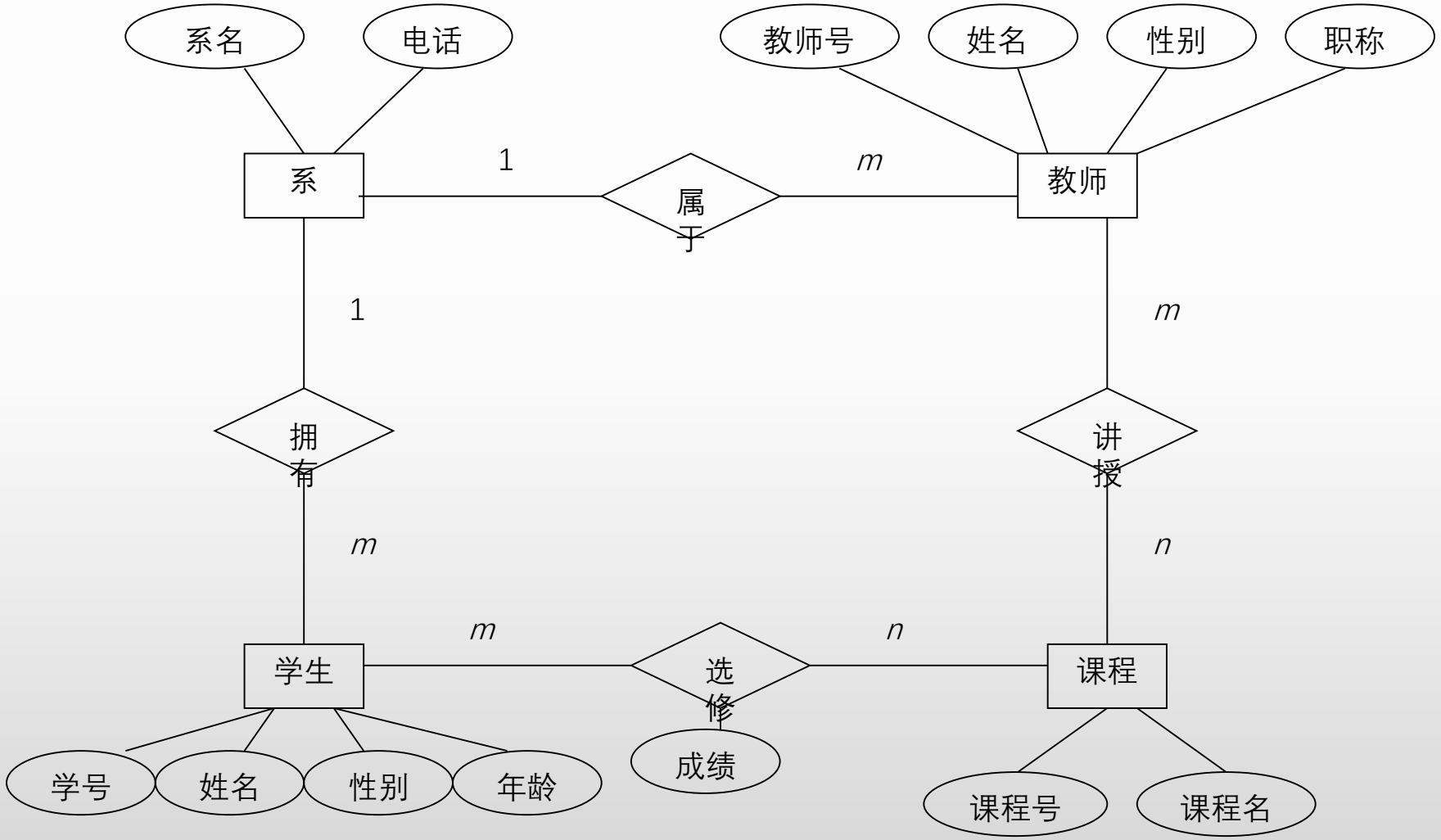
供应（<u>供应商号，项目号，零件号</u>，数量）

7.4.3 ER图向关系模型的转换(续)

6 具有相同码的关系模式可合并。

- 目的：减少系统中的关系个数。
- 合并方法：将其中一个关系模式的全部属性加入到另一个关系模式中，然后去掉其中的同义属性（可能同名也可能不同名），并适当调整属性的次序。

例子


系名
电话
教师号
姓名
性别
职称
系
1
属于
m
教师
1
m
拥有
讲授
m
n
m
n
学生
选修
课程
学号
姓名
性别
年龄
成绩
课程号
课程名

2．具体做法

(1) 把每一个实体转换为一个关系

- 首先分析各实体的属性，从中确定其主键，然后分别用关系模式表示。例如，前页E－R图中的四个实体分别转换成四个关系模式:
  - 学生（学号，姓名，性别，年龄）
  - 课程（课程号，课程名）
  - 教师（教师号，姓名，性别，职称）
  - 系（系名，电话）
- 其中，有下划线者表示是主键。

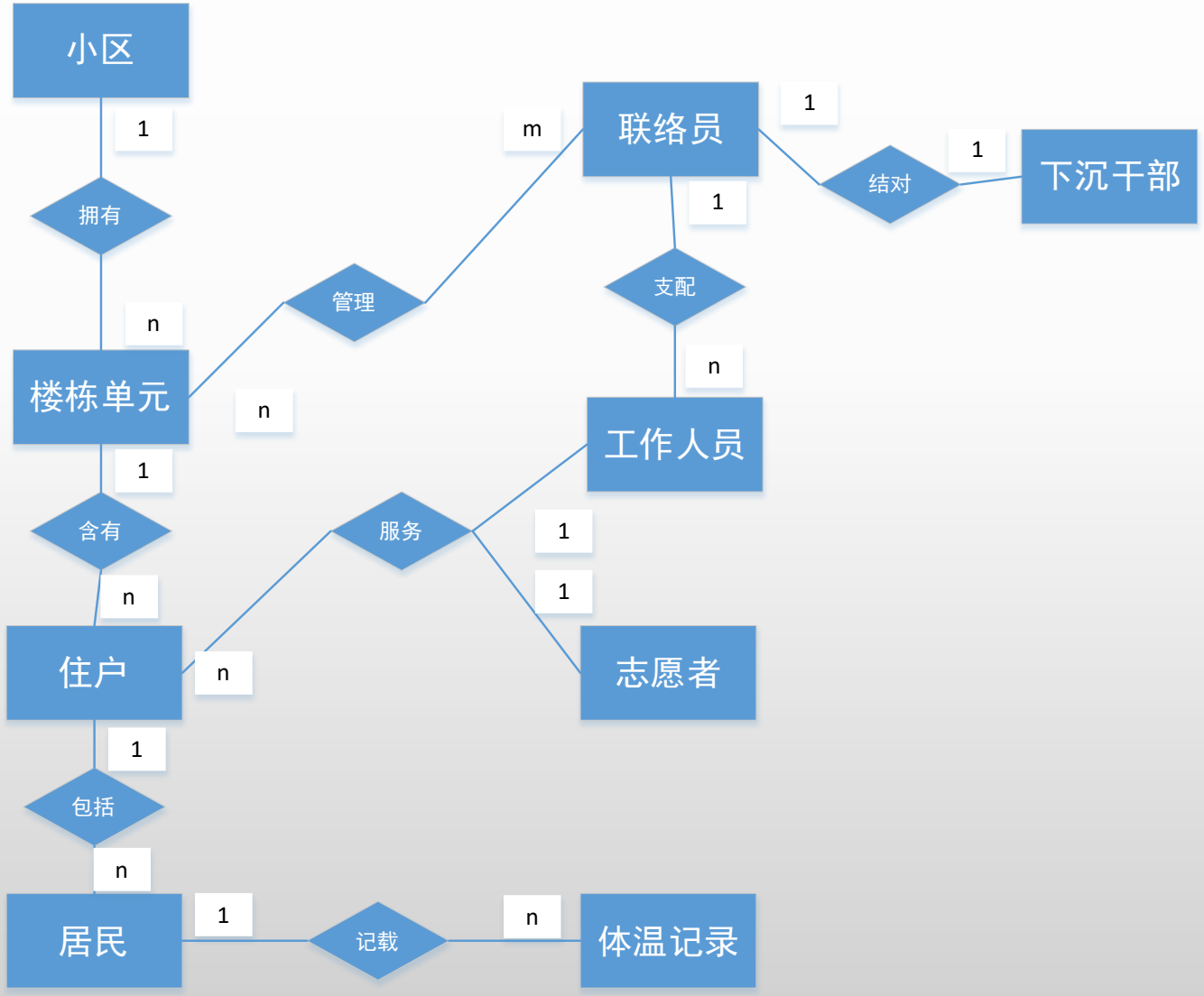
(2) 把每一个联系转换为关系模式

- 由联系转换得到的关系模式的属性集中，包含两个发生联系的实体中的主键以及联系本身的属性，其关系键的确定与联系的类型有关。
- 例如，前图四个联系也分别转换成四个关系模式:
  - 属于（教师号，系名）
  - 讲授（教师号，课程号）
  - 选修（学号，课程号，成绩）
  - 拥有（系名，学号）
- 1:1的联系也可以与任意一端对应的关系模式合并，并在该关系模式的属性中加入另一个关系模式的码和联系本身的属性。
- 1:n的联系也可以与n端对应的关系模式合并，并在这一端加上1端关系模式的码和联系本身的属性。

数据库设计案例-社区疫情防控

- 疫情刻不容缓，数据库技术支撑社区级防控
- 疫情期间，某社区有若干小区，每个小区有多个楼栋单元，一个单元有多个家庭住户，每户人家有多位居民，每位居民每天必须上报自己的体温数据，居民出入小区也必须测温并记录在案，社区派出多名联络员，每名联络员需要负责多个楼栋单元的疫情管控，一个楼栋单元也需要多名联络员分别处理不同疫情事务，社区接收了若干上级单位的下沉干部，并建立了下沉干部与联络员的一对一工作关系，一名联络员负责某个小区多名工作人员的管理工作，社区还建立了共产党员志愿者团队，志愿者和小区工作人员采用一对一结对方式为若干住户提供日常生活物资服务。

数据库设计案例-社区疫情防控



数据库设计案例-社区疫情防控

- 小区（小区ID，小区名称，地址，物业电话）
- 楼栋单元（单元ID，单元名称，*小区ID*）
- 住户（住户ID，住户地址，户主姓名，联系电话，*单元ID*，*人员ID*，*志愿者ID*）
- 居民（居民身份证号，姓名，联系电话，*住户ID*）
- 体温记录（记录ID，居民身份证号，记录类型（出入小区、日常测温），记录日期时间）
- 联络员（联络员ID，联络员姓名，联系电话）
- 管理（单元ID，联络员ID）（两个字段同时也为外码）
- 下沉干部（干部ID，干部姓名，*联络员ID*）
- 工作人员（人员ID，服务部门，联系电话，*联络员ID*）
- 志愿者（志愿者ID，志愿者姓名，联系电话）

7.4.4 数据模型的优化

并不是规范化程度越高的关系就越优。当一个应用的查询中经常涉及到两个或多个关系模式的属性时，系统必须经常地进行联接运算，而联系运算的代价是相当高的，可以说关系模型低效的主要原因就是做联接运算引起的，因此在这种情况下，第二范式甚至第一范式也许是最好的。

7.4.4 数据模型的优化

- 非BCNF的关系模式虽然从理论上分析会存在不同程度的更新异常，但如果在实际应用中对此关系模式只是查询，并不执行更新操作，则就不会产生实际影响。

- 对于一个具体应用来说，到底规范化进行到什么程度，需要权衡响应时间和潜在问题两者的利弊才能决定。一般说来，第三范式就足够了。

7.4.4 数据模型的优化

例：在关系模式

学生成绩单(学号,英语,数学,语文,平均成绩)

中存在下列函数依赖：

学号→英语

学号→数学

学号→语文

学号→平均成绩

(英语, 数学, 语文)→平均成绩

7.4.4 数据模型的优化

显然有:

学号→(英语,数学,语文)

因此该关系模式中存在传递函数信赖，是2NF关系。

虽然平均成绩可以由其他属性推算出来，但如果应用中需要经常查询学生的平均成绩，为提高效率，我们仍然可保留该冗余数据，对关系模式不再做进一步分解。

7.4.5 设计用户子模式

- 定义数据库模式主要是从系统的时间效率、空间效率、易维护等角度出发。

- 定义用户外模式时应该更注重考虑用户的习惯与方便。包括三个方面:

设计用户子模式（续）

(1) 使用更符合用户习惯的别名

- 合并各分E-R图曾做了消除命名冲突的工作，以使数据库系统中同一关系和属性具有唯一的名字。这在设计数据库整体结构时是非常必要的。
- 但对于某些局部应用，由于改用了不符合用户习惯的属性名，可能会使他们感到不方便，因此在设计用户的子模式时可以重新定义某些属性名，使其与用户习惯一致。

例：负责学籍管理的用户习惯于称教师模式的职工号为教师编号。因此可以定义视图，在视图中职工号重定义为教师编号。

设计用户子模式（续）

(2) 针对不同级别的用户定义不同的外模式，以满足系统对安全性的要求。

设计用户子模式（续）

例:

教师关系模式中包括职工号、姓名、性别、出生日期、婚姻状况、学历、学位、政治面貌、职称、职务、工资、工龄、教学效果等13个属性。

学籍管理应用只能查询教师的职工号、姓名、性别、职称数据;

课程管理应用只能查询教师的职工号、姓名、性别、学历、学位、职称、教学效果数据;

教师管理应用则可以查询教师的全部数据。

设计用户子模式（续）

定义两个外模式：

教师_学籍管理(职工号，姓名，性别，职称)

教师_课程管理(工号，姓名，性别，学历，
学位，职称，教学效果)

授权学籍管理应用只能访问教师_学籍管理视图

授权课程管理应用只能访问教师_课程管理视图

授权教师管理应用能访问整个教师表

这样就可以防止用户非法访问本来不允许他们查询的数据，保证了系统的安全性。

设计用户子模式（续）

(3) 简化用户对系统的使用

- 如果某些局部应用中经常要使用某些很复杂的查询，为了方便用户，可以将这些复杂查询定义为视图。

7.1 数据库设计概述
7.2 需求分析
7.3 概念结构设计
7.4 逻辑结构设计
7.5 物理设计
7.6 数据库实施
7.7 数据库运行和维护

7.5 物理设计

- 对于给定的逻辑数据模型，选取一个最适合应用环境的物理结构的过程，称为数据库物理设计。
- 物理设计的任务是为了有效地实现逻辑模式，确定所采取的存储策略。此阶段是以逻辑设计的结果作为输入，结合具体DBMS的特点与存储设备特性进行设计，选定数据库在物理设备上的存储结构和存取方法。
- 数据库的物理设计可分为两步：

  (1)确定物理结构，在关系数据库中主要指存取方法和存储结构；

  (2)评价物理结构，评价的重点是时间和空间效率。

数据库物理设计

- 数据库在物理设备上的存储结构与存取方法称为数据库的物理结构，它依赖于给定的计算机系统。
- 为一个给定的逻辑数据模型选取一个最适合应用环境的物理结构的过程，就是数据库的物理设计。
- 目标：
  - 提高数据库性能，以满足应用的性能需求；
  - 有效利用存储空间；
  - 在性能和代价之间做出最优平衡。
- 内容：
  - 确定数据库的存储结构；
  - 为数据选择合适的存取路径，即索引的设计；
  - 对物理结构进行评价，重点是评价时间和空间效率。

数据库的物理组织

- 数据库的基础是基于操作系统的文件系统，对数据库的操作都要转化为对文件的操作，如何设计文件结构以及有效利用操作系统提供的文件存取方法是DBMS要考虑的事情。
- 因此，选定DBMS后，数据库物理组织的大概框架也就基本确定了，如一个数据库需要多少个文件，每个文件的作用是什么，等等。
- 关系数据库中要存储的数据主要包括：关系表、数据字典、索引、日志和备份等。DBMS对不同数据的物理组织方式通常是不一样的。

确定数据库存储结构

- 确定数据存放位置： 为了提高系统性能，数据应该根据应用情况将易变部分和稳定部分、经常存取部分和存取频率较低部分分开来存放。
- 确定数据库存储结构： 确定数据库存储结构时要综合考虑存取时间、存储空间利用率和维护代价三个方面的因素。这三个方面常常是相互矛盾的。例如，消除一切冗余数据虽然能够节约存储空间，但往往会导致检索代价的增加，因此必须进行权衡，选择一个折衷方案。

确定数据存取路径

- 在关系数据库中，选择存取路径主要是指确定如何建立索引。例如，应选择哪些属性作为搜索码建立索引，建立多少个索引，建立聚集索引(主索引)还是非聚集索引(辅助索引)，建立单码索引还是组合索引，等等。
- 常用的存储方式有三种：索引方法、聚集方法和Hash方法。目前使用最普遍的是B^+树索引 。
- 关系数据库中存取路径具有下列特点：
  - 存取路径和数据是分离的；
  - 存取路径可以由用户建立、删除，也可以由系统动态建立和删除；
  - 存取路径的物理组织可以采用顺序文件、B^+树文件或Hash文件。

确定系统配置

- 通常情况下，系统配置变量包括：同时使用数据库的用户数，同时打开数据库对象数，使用的缓冲区长度、个数，时间片大小，数据库的大小，装填因子，锁的数目等。这些参数值影响存取时间和存储空间的分配，在数据库物理设计时要根据应用环境确定这些参数值，以使系统性能最优。
- 注意，在数据库物理设计时对系统配置参数的调整只是初步的，在系统运行时还要根据系统实际运行情况做进一步的调整，以期切实改进系统性能。

物理结构评价

- 数据库物理设计过程中，需要对时间效率、空间效率、维护代价和各种用户要求进行权衡，其结果可以产生多种方案。
- 数据库设计人员必须对这些方案进行细致的评价，从中选择一个较优的方案作为数据库的物理结构。
- 评价物理数据库的方法完全依赖于所选用的DBMS，主要是从定量估算各种方案的存储空间、存取时间和维护代价入手，对估算结果进行权衡、比较，选择出一个较优的合理的物理结构。如果该结构不符合用户需求，则需要修改设计。

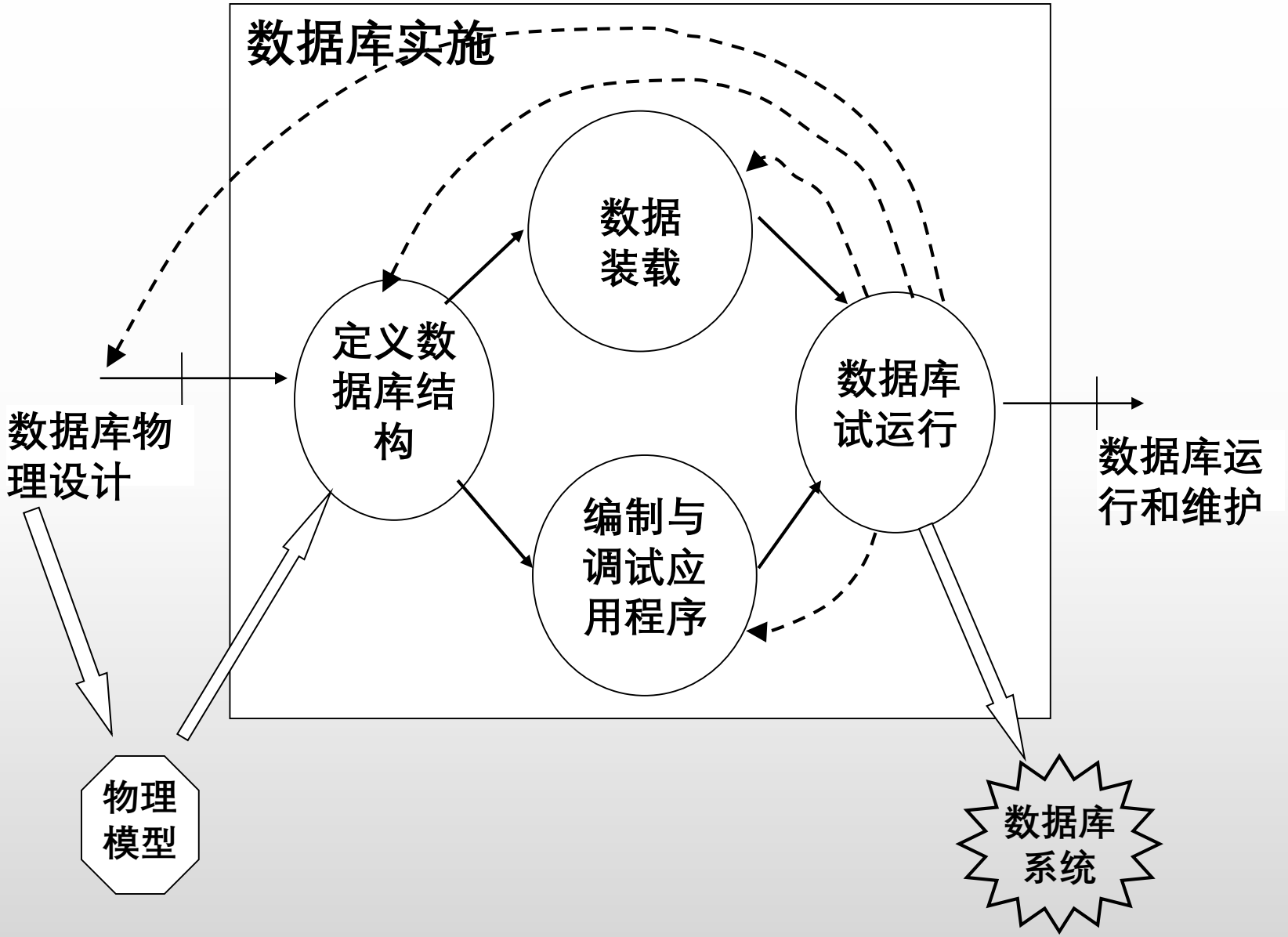
影响物理设计的主要因素

- 应用处理需求。在进行数据库物理设计前，应先弄清应用的处理需求，如吞吐量、平均响应时间、系统负荷、事务类型及发生频率等，这些需求直接影响着设计方案的选择，而且它们还会随应用环境的变化而变化。
- 数据特征。数据本身的特性对数据库物理设计也会有较大影响。如关系中每个属性值的分布、记录的长度与个数等，这些特性都影响到数据库的物理存储结构和存取路径的选择。
- 运行环境。数据库物理设计与运行环境有关，因此在设计时还要充分考虑DBMS、操作系统、网络、计算机硬件等运行环境的特征和限制。
- 物理设计的调整。数据库物理设计是基于数据库当前状况从许多可供选择的方案中选择一个合适的设计方案。但是随着时间的变化，数据库的状态和特性也会发生变化，因此可能导致以前的物理设计不能再满足目前的应用需求，因此，需对物理设计不断调整，甚至有时需要重新设计。

7.1 数据库设计概述
7.2 需求分析
7.3 概念结构设计
7.4 逻辑结构设计
7.5 物理设计
7.6 数据库实施
7.7 数据库运行和维护

7.6 数据库的实施

- 数据库实施的工作内容
  - 用DDL定义数据库结构
  - 组织数据入库
  - 编制与调试应用程序
  - 数据库试运行


数据库实施
数据装载
定义数据库结构
数据库试运行
编制与调试应用程序
数据库物理设计
数据库运行和维护
物理模型
数据库系统

一、定义数据库结构

- 确定了数据库的逻辑结构与物理结构后，就可以用所选用的DBMS提供的数据定义语言（DDL）来严格描述数据库结构。

例，对于前面的例子，可以用SQL语句如下定义表结构:

```
CREATE  TABLE  学生
      (学号  CHAR(8),
        ……
      );
CREATE  TABLE  课程
      (
        ……
      );
……
```

定义数据库结构（续）

接下来是在这些基本表上定义视图：

```
CREATE VIEW ....
(
            ..............
 );
..............
```

如果需要使用聚簇，在建基本表之前，应先用CREATE CLUSTER语句定义聚族。

二、数据装载

- 数据库结构建立好后，就可以向数据库中装载数据。组织数据入库是数据库实施阶段最主要的工作。

- 数据装载方法
  - 人工方法
  - 计算机辅助数据入库

数据装载（续）

- 人工方法：适用于小型系统
- 计算机辅助数据入库：适用于中大型系统
  - 步骤

1) 筛选数据。需要装入数据库中的数据通常都分散在各个部门的数据文件或原始凭证中，所以首先必须把需要入库的数据筛选出来。

2) 转换数据格式。筛选出来的需要入库的数据，其格式往往不符合数据库要求，还需要进行转换。这种转换有时可能很复杂。

3) 输入数据。将转换好的数据输入计算机中。

4) 校验数据。检查输入的数据是否有误。

数据装载（续）

- 如果数据库是在老的文件系统或数据库系统的基础上设计的，则数据输入子系统只需要完成转换数据、综合数据两项工作，直接将老系统中的数据转换成新系统中需要的数据格式。

- 为了保证数据能够及时入库，应在数据库物理设计的同时编制数据输入子系统。

三、编制与调试应用程序

- 在数据库实施阶段，当数据库结构建立好后，就可以开始编制与调试数据库的应用程序。调试应用程序时由于数据入库尚未完成，可先使用模拟数据。

四、数据库试运行

- 应用程序调试完成，并且已有一小部分数据入库后，就可以开始数据库的试运行。

- 数据库试运行也称为联合调试，其主要工作包括：
  1）功能测试：实际运行应用程序，执行对数据库的各种操作，测试应用程序的各种功能。

  2）性能测试：测量系统的性能指标，分析是否符合设计目标。

数据库试运行（续）

- 数据库性能指标的测量
  - 数据库物理设计阶段在评价数据库结构估算时间、空间指标时，作了许多简化和假设，忽略了许多次要因素，因此结果必然很粗糙。
  - 数据库试运行则是要实际测量系统的各种性能指标（不仅是时间、空间指标），如果结果不符合设计目标，则需要返回物理设计阶段，调整物理结构，修改参数；有时甚至需要返回逻辑设计阶段，调整逻辑结构。

数据库试运行（续）

- 数据的分期入库
  - 重新设计物理结构甚至逻辑结构，会导致数据重新入库。
  - 由于数据入库工作量实在太大，所以可以采用分期输入数据的方法
    - 先输入小批量数据供先期联合调试使用
    - 待试运行基本合格后再输入大批量数据
    - 逐步增加数据量，逐步完成运行评价

数据库试运行（续）

- 数据库的转储和恢复
  - 在数据库试运行阶段，系统还不稳定，硬、软件故障随时都可能发生
  - 系统的操作人员对新系统还不熟悉，误操作也不可避免
  - 因此必须做好数据库的转储和恢复工作，尽量减少对数据库的破坏。

第七章 数据库设计

7.7 数据库运行与维护

- 数据库试运行结果符合设计目标后，数据库就可以真正投入运行了。
- 数据库投入运行标着开发任务的基本完成和维护工作的开始
- 对数据库设计进行评价、调整、修改等维护工作是一个长期的任务，也是设计工作的继续和提高。
  - 应用环境在不断变化
  - 物理存储也在变化

数据库运行与维护（续）

- 在数据库运行阶段，对数据库经常性的维护工作主要是由DBA完成的，包括：

1. 数据库的转储和恢复
   - 转储和恢复是系统正式运行后最重要的维护工作之一。
   - DBA要针对不同的应用要求制定不同的转储计划，定期对数据库和日志文件进行备份。
   - 一旦发生介质故障，即利用数据库备份及日志文件备份，尽快将数据库恢复到某种一致性状态。

数据库运行与维护（续）

2.数据库的安全性、完整性控制

- DBA必须根据用户的实际需要授予不同的操作权限。
- 在数据库运行过程中，由于应用环境的变化，对安全性的要求也会发生变化，DBA需要根据实际情况修改原有的安全性控制。
- 由于应用环境的变化，数据库的完整性约束条件也会变化，也需要DBA不断修正，以满足用户要求。

数据库运行与维护（续）

3.数据库性能的监督、分析和改进

- 在数据库运行过程中， DBA必须监督系统运行，对监测数据进行分析，找出改进系统性能的方法。
  - 利用监测工具获取系统运行过程中一系列性能参数的值
  - 通过仔细分析这些数据，判断当前系统是否处于最佳运行状态
  - 如果不是，则需要通过调整某些参数来进一步改进数据库性能

数据库运行与维护（续）

4.数据库的重组织和重构造

1）数据库的重组织

- 为什么要重组织数据库
  - 数据库运行一段时间后，由于记录的不断增、删、改，会使数据库的物理存储变坏，从而降低数据库存储空间的利用率和数据的存取效率，使数据库的性能下降。

数据库运行与维护（续）

- 重组织的形式
  - 全部重组织
  - 部分重组织
    - 只对频繁增、删的表进行重组织
- 重组织的目的
  - 提高系统性能
- 重组织的工作
  - 按原设计要求
    - 重新安排存储位置
    - 回收垃圾
    - 减少指针链

数据库运行与维护（续）

- DBMS一般都提供了供重组织数据库使用的实用程序，帮助DBA重新组织数据库。

数据库运行与维护（续）

2）数据库的重构造

- 为什么要进行数据库的重构造
  - 数据库应用环境发生变化，会导致实体及实体间的联系也发生相应的变化，使原有的数据库设计不能很好地满足新的需求
    - 增加新的应用或新的实体
    - 取消某些已有应用
    - 改变某些已有应用

数据库运行与维护（续）

- 数据库重构造的主要工作
  - 根据新环境调整数据库的模式和内模式
    - 增加新的数据项
    - 改变数据项的类型
    - 改变数据库的容量
    - 增加或删除索引
    - 修改完整性约束条件

数据库运行与维护（续）

- 重构造数据库的程度是有限的
  - 若应用变化太大，已无法通过重构数据库来满足新的需求，或重构数据库的代价太大，则表明现有数据库应用系统的生命周期已经结束，应该重新设计新的数据库系统，开始新数据库应用系统的生命周期了。