

数据库原理

Theory of Database

李静

信息科学与技术学院

第8章 数据库设计

- ❖ 8.1 数据库设计概述
- ❖ 8.2 数据库需求分析
- ❖ 8.3 数据库结构设计
- ❖ 8.4 数据库行为设计
- ❖ 8.5 数据库实施

数据库设计概述

❖ 数据库设计

数据库设计是指对于一个给定的应用环境，构造（设计）优化的数据库逻辑模式和物理结构，并据此建立数据库及其应用系统，使之能够有效地存储和管理数据，满足各种用户的应用需求，包括信息管理要求和数据操作要求。

■ 目标：

为用户和各种应用系统提供一个信息基础设施和高效率的运行环境。

8.1 数据库设计概述

- ❖ 8.1.1 数据库设计的特点
- ❖ 8.1.2 数据库设计方法概述
- ❖ 8.1.3 数据库设计的基本步骤

数据库建设的基本规律

- 三分技术，七分管理，十二分基础数据
- 管理
 - 数据库建设项目管理
 - 企业（即应用部门）的业务管理
- 基础数据
 - 收集、入库
 - 更新新的数据

8.1.1 数据库设计的特点

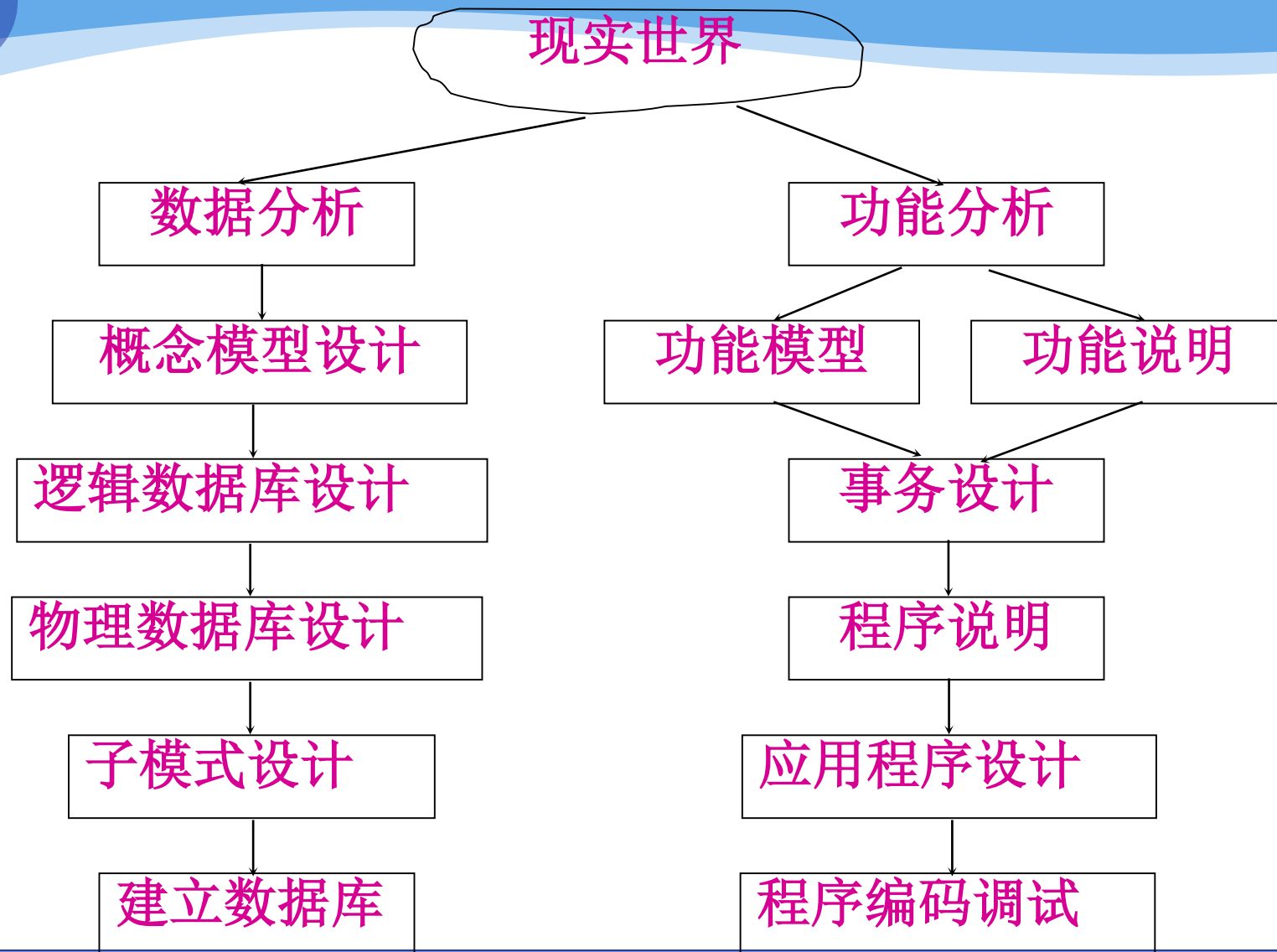
(1) 综合性

- 涉及面广，需包含计算机专业知识及业务系统专业知识；
- 要解决技术及非技术两方面的问题；

(2) 静态结构设计与动态行为设计是分离的

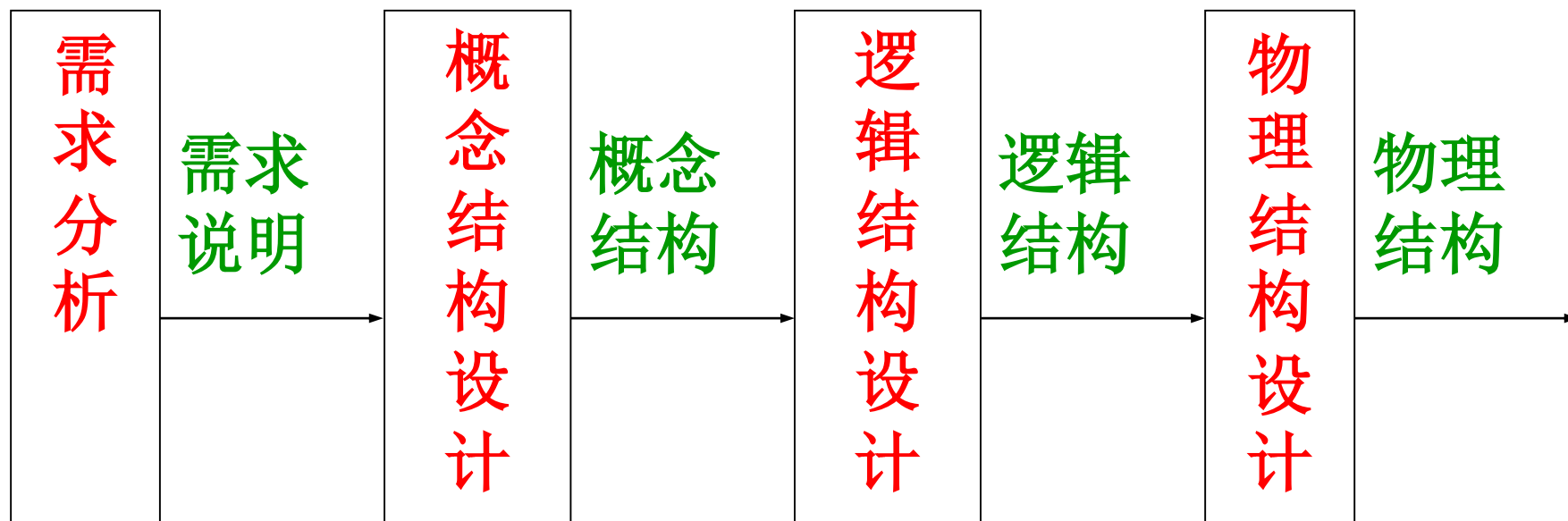
- 静态结构设计是指数据库的模式框架设计（包括语义结构、数据结构、存储结构）；
- 动态行为设计是指应用程序设计（动作操纵：功能组织、流程控制）

结构和行为分离的设计



8.1.2 数据库设计方法概述

❖ 新奥尔良（New Orleans）方法



数据库设计方法概述

- ❖ 基于E-R模型的数据库设计方法
 - 概念设计阶段广泛采用
- ❖ 3NF（第三范式）的设计方法
 - 逻辑阶段可采用的有效方法
- ❖ ODL（Object Definition Language）方法
 - 面向对象的数据库设计方法
- ❖ 计算机辅助设计
 - ORACLE Designer 2000
 - SYBASE PowerDesigner

数据库设计方法概述

❖ 本质:手工与经验相结合方法

- 设计质量与设计人员的经验和水平有直接关系
- 数据库运行一段时间后常常不同程度地发现各种问题，增加了维护代价

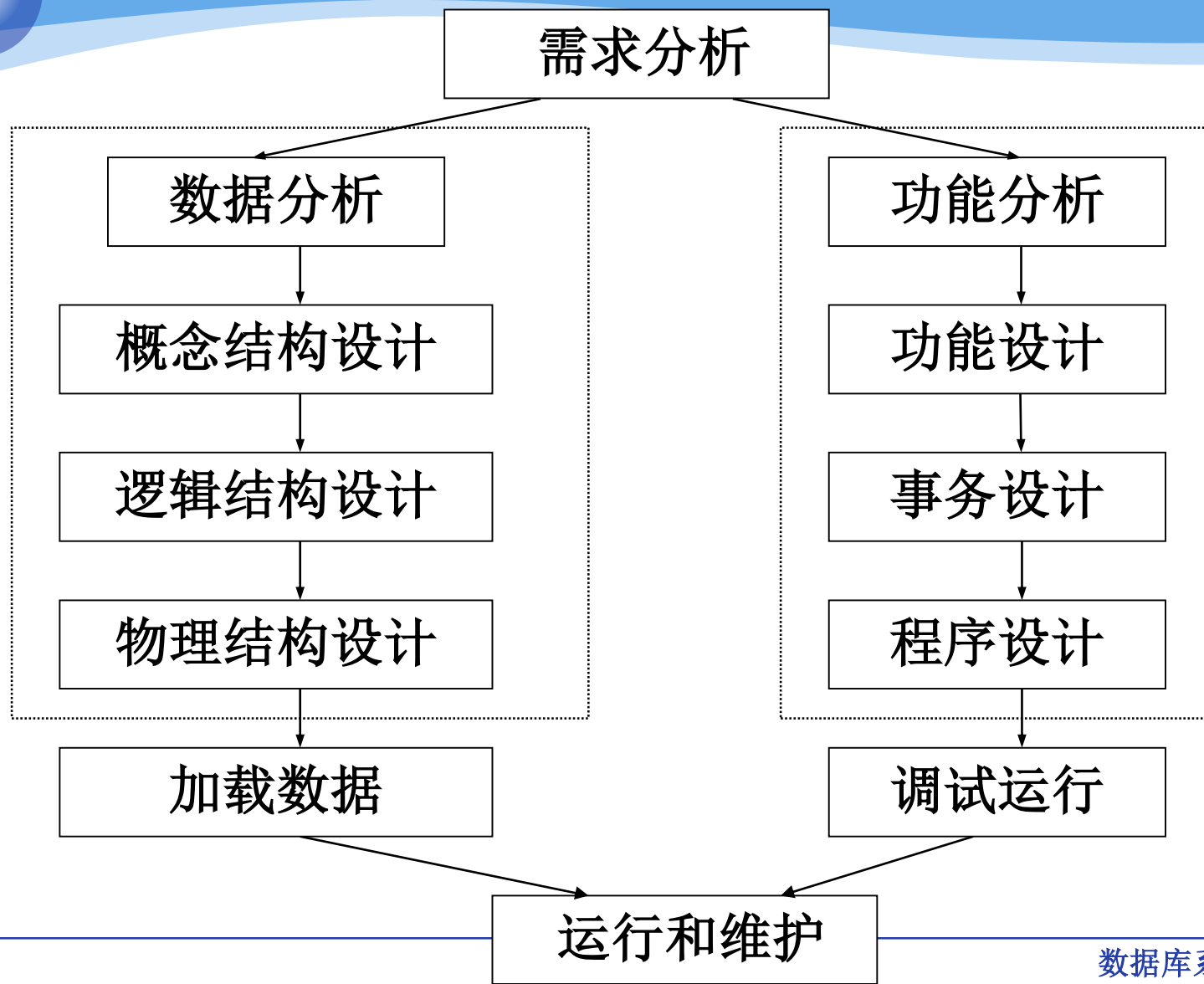
❖ 基本思想

- 过程迭代和逐步求精

8.1.3 数据库设计的基本步骤

- ❖ 需求分析
- ❖ 结构设计
- ❖ 行为设计
- ❖ 数据库实施
- ❖ 数据库运行和维护

数据库设计全过程



8.2 数据库需求分析

❖ 8.2.1 需求分析的任务

❖ 8.2.2 需求调查



8.2.1 需求分析的任务

需求分析阶段的主要任务：

- ❖ 详细调查现实世界要处理的对象（组织、部门、企业等）
- ❖ 充分了解原系统（手工系统或计算机系统）
- ❖ 明确用户的各种需求
- ❖ 确定新系统的功能
- ❖ 充分考虑今后可能的扩充和改变

需求分析的重点

❖ 重点是“数据”和“处理”，获得用户要求

- 信息要求
- 处理要求
- 安全性与完整性要求

需求分析的难点

❖ 信息需求

定义未来数据库系统用到的所有信息，明确用户将向数据库中输入什么样的数据，从数据库中要求获得哪些内容，将要输出哪些信息。同时还要描述数据间的联系等。

❖ 处理需求

定义了系统数据处理的操作功能，描述操作的优先次序，包括操作的执行频率和场合，操作与数据间的联系。处理需求还要明确用户要完成哪些处理功能，每种处理的执行频度，用户需求的响应时间以及处理的方式，比如是联机处理还是批处理，等等。

❖ 安全性与完整性要求

描述了系统中不同用户对数据库的使用和操作情况，完整性要求描述了数据之间的关联关系以及数据的取值范围要求。

需求分析的难点

❖ 确定用户最终需求

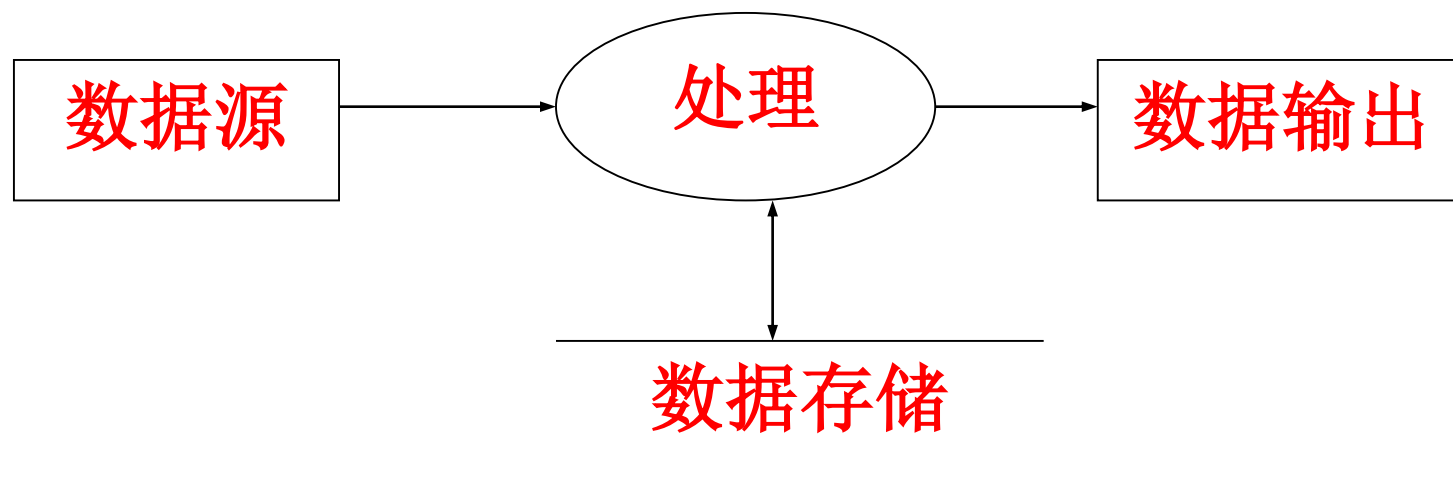
- 用户缺少计算机知识
- 设计人员缺少用户的专业知识

❖ 解决方法

- 设计人员必须不断深入地与用户进行交流

数据处理流图

在需求分析中，通过自顶向下、逐步分解的方法分析系统。任何一个系统都可以抽象为数据流图的形式。



需求分析的方法

- ❖ 调查需求
- ❖ 达成共识
- ❖ 分析表达需求

调查用户需求的具体步骤

- (1) 调查组织机构情况（业务现状）
- (2) 调查各部门的业务活动信息源流情况。
- (3) 明确对新系统的各种要求。（外部要求）

常用调查方法

- (1)跟班作业，观察业务的运转。
- (2)开调查会，面谈交流。
- (3)请专人介绍或通过其他渠道研究分析。
- (4)设计问卷调查表请用户填写。
- (5)查阅历史记录。

进一步分析和表达用户需求

❖ 结构化分析方法（**Structured Analysis**，简称**SA方法**）

- 从最上层的系统组织机构入手
- 自顶向下、逐层分解分析系统

进一步分析和表达用户需求（续）

1. 分解处理功能和数据

(1) 分解处理功能

- 将处理功能的具体内容分解为若干子功能

(2) 分解数据

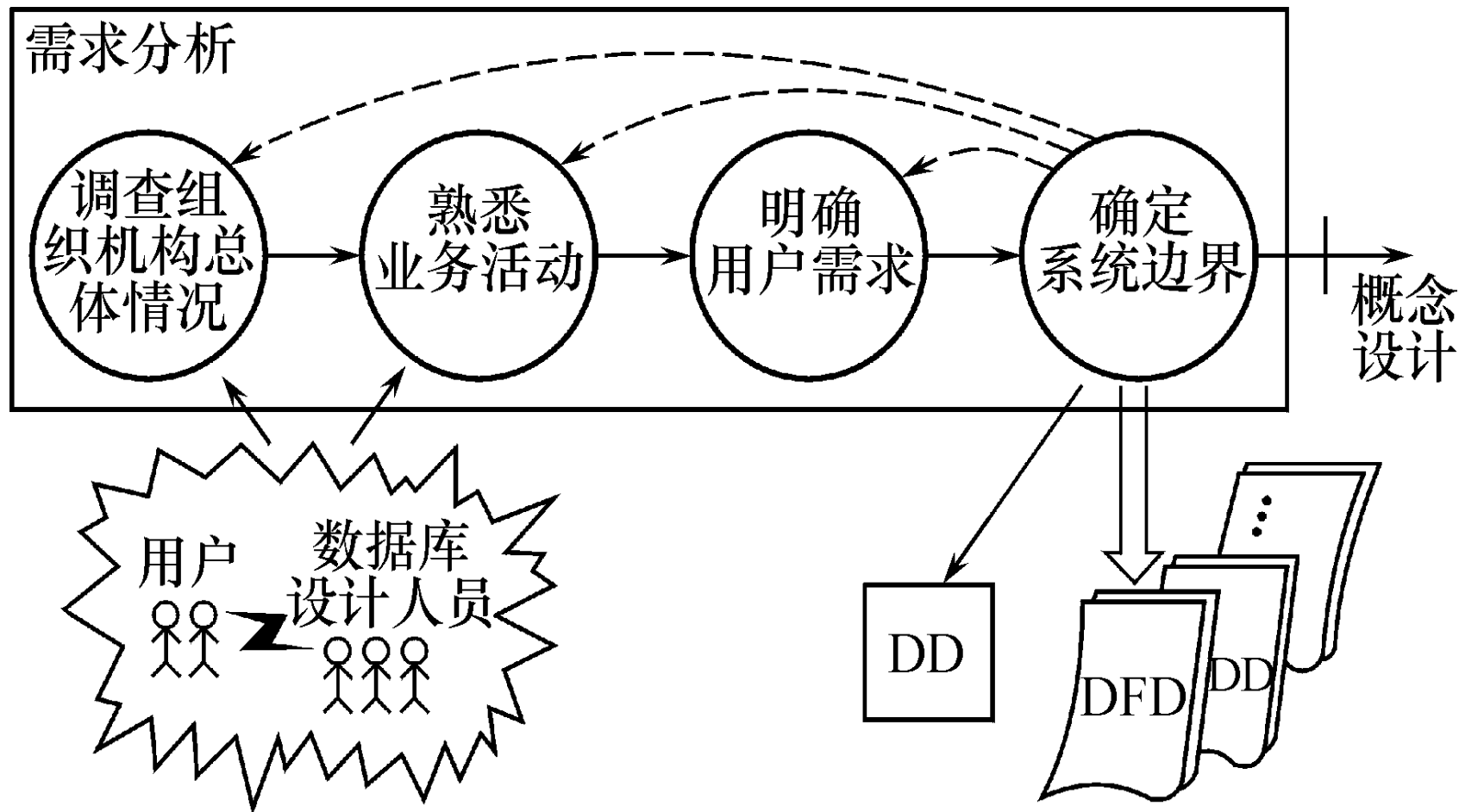
- 处理功能逐步分解同时，逐级分解所用数据，形成若干层次的数据流图

(3) 表达方法

- 处理逻辑：用判定表或判定树来描述
- 数据：用数据字典来描述

2. 将分析结果再次提交给用户，征得用户的认可

需求分析过程



数据字典

❖ 数据字典的用途

- 进行详细的数据收集和数据分析所获得的主要结果

❖ 数据字典的内容

- 数据项
- 数据结构
- 数据流
- 数据存储
- 处理过程

1. 数据项

❖ 数据项是不可再分的数据单位

❖ 对数据项的描述

数据项描述 = { 数据项名, 数据项含义说明, 别名,
数据类型, 长度, 取值范围, 取值含义,
与其他数据项的逻辑关系, 数据项之间的联系 }

定义了数据的完整性约束

2. 数据结构

- ❖ 数据结构反映了数据之间的组合关系。
- ❖ 一个数据结构可以由若干个数据项组成，也可以由若干个数据结构组成，或由若干个数据项和数据结构混合组成。
- ❖ 对数据结构的描述
数据结构描述 = { 数据结构名, 含义说明,
组成: { 数据项或数据结构 } }

3. 数据流

❖ 数据流是数据结构在系统内传输的路径。

❖ 对数据流的描述

数据流描述 = { 数据流名, 说明, 数据流来源,
数据流去向, 组成: { 数据结构 },
平均流量, 高峰期流量 }

4. 数据存储

❖ 数据存储是数据结构停留或保存的地方，也是数据流的来源和去向之一

批处理还是联机处理
检索还是更新

每小时或每天存取几次，
每次存取多少数据

❖ 对数据存储的描述

数据存储描述 = { 数据存储名, 说明, 编号,

输入的数据流, 输出的数据流,

组成: { 数据结构 }, 数据量, 存取频度,

存取方式 }

5. 处理过程

- ❖ 具体处理逻辑一般用判定表或判定树来描述
- ❖ 处理过程说明性信息的描述

处理过程描述 = { 处理过程名, 说明,

输入: { 数据流 },

处理过程的功能及处理要求

输出: { 数据流 },

处理: { 简要说明 } }

数据字典举例

例：学生学籍管理子系统的数据字典。

数据项，以“学号”为例：

数据项： 学号

含义说明：唯一标识每个学生

别名： 学生编号

类型： 字符型

长度： 8

取值范围：00000000至99999999

取值含义：前两位标别该学生所在年级，
后六位按顺序编号

与其他数据项的逻辑关系：

数据结构

数据结构，以“学生”为例

“学生”是该系统中的一个核心数据结构：

数据结构： 学生

含义说明： 学籍管理子系统的主体数据结构，
定义了一个学生的有关信息

组成： 学号，姓名，性别，年龄，所在系，年级

数据流

数据流，“体检结果”可如下描述：

数据流： 体检结果

说明： 学生参加体格检查的最终结果

数据流来源： 体检

数据流去向： 批准

组成：

平均流量：

高峰期流量：

数据存储

数据存储，“学生登记表”可如下描述：

数据存储： 学生登记表

说明： 记录学生的基本情况

流入数据流：

流出数据流：

组成：

数据量： 每年3000张

存取方式： 随机存取

处理过程

处理过程，“用户登录”可描述如下：

处理过程名：用户登录

说明：判断用户登录情况

输入：用户名，密码

输出：登录是否成功信息

处理：检测用户名和密码是否与数据库一致。成功则显示主页面，不成功则提示登录失败信息，提示用户重新输入

数据字典

- ❖ 数据字典是关于数据库中数据的描述，是元数据，而不是数据本身
- ❖ 数据字典在需求分析阶段建立，在数据库设计过程中不断修改、充实、完善

需求分析说明书

软件需求分析说明书是在对用户需求分析的基础上，把用户的需求规范化、形式化。目的是为软件开发提出总体要求，作为用户和开发人员之间相互了解和共同开发的基础。根据我国国家标准**GB856F-88**的规定，软件需求分析说明书的内容如下：

需求分析说明书

❖ 1. 引言

1.1 编写说明

1.2 背景

1.3 定义

1.4 参考资料

2. 任务概述

2.1 目标

2.2 用户的特点

2.3 假定与约束

需求分析说明书

❖ 3. 需求规定

3.1 对功能的规定

3.2 对性能的规定

3.2.1 精度

3.2.2 时间特性要求

3.2.3 灵活性

3.3 输入输出要求

3.4 数据管理能力要求

3.5 故障处理要求

3.6 其它专门要求

需求分析说明书

❖ 4. 运行环境规定

4.1 设备

4.2 支持软件

4.3 接口

4.4 控制

需求分析小结

- ❖ 设计人员应充分考虑到可能的扩充和改变，
使设计易于更改，系统易于扩充
- ❖ 必须强调用户的参与

8.3 数据库结构设计

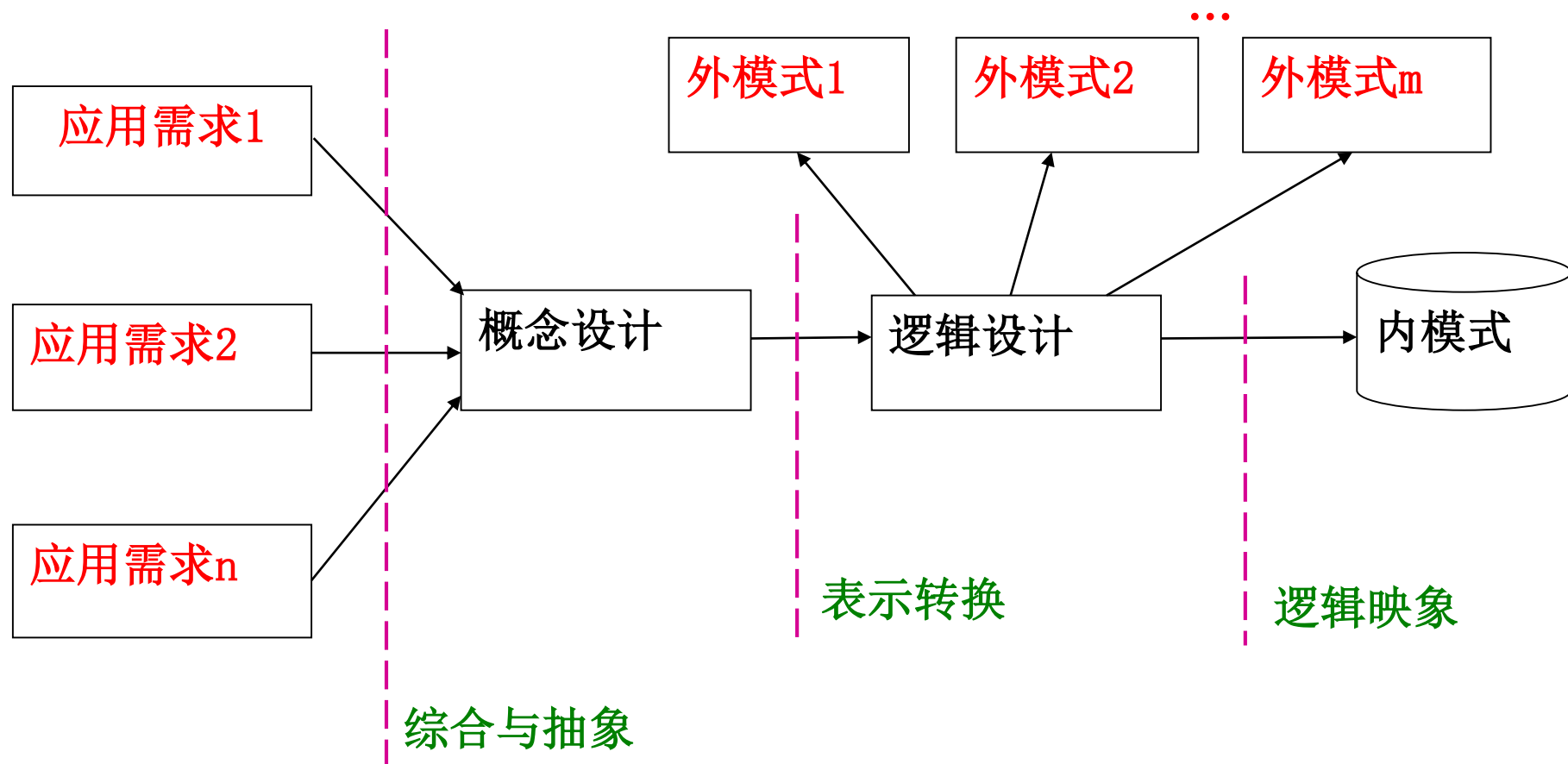
- ❖ 8.3.1 概念结构设计
- ❖ 8.3.2 逻辑结构设计
- ❖ 8.3.3 物理结构设计



数据库设计分类

- ❖ 数据库设计分为：数据库结构设计和数据库行为设计。
- ❖ **结构设计**包括设计数据库的概念结构、逻辑结构和存储结构。
- ❖ **行为设计**包括设计数据库的功能组织和流程控制。

数据库结构设计过程



数据库结构设计包含内容

- ❖ **概念结构设计**：形成DB概念模式，用语义层模型描述，如E—R图。
- ❖ **逻辑结构设计**：形成DB逻辑模式与外模式，用结构层模型描述，例基本表、视图等。
- ❖ **物理结构设计**：形成DB内模式，用文件级术语描述。例DB文件或目录、索引。

8.3.1 概念结构设计

- ❖ 概念结构设计任务是产生反映企业组织信息需求的数据库概念结构，即概念模型。



概念模型的特点

- ❖ 有丰富的语义表达能力。
- ❖ 易于交流和理解。
- ❖ 易于更改。
- ❖ 易于向各种数据模型转换，易于导出与DBMS有关的逻辑模型。

概念结构设计的策略

- ❖ **自底向上**。先定义局部应用的概念结构，然后按一定的规则把它们集成起来，从而得到全局概念模型。
- ❖ **自顶向下**：先定义全局概念模型，然后再逐步细化。
- ❖ **由里向外**：先定义最重要的核心结构，然后再逐步向外扩展。
- ❖ **混合策略**。将自顶向下和自底向上结合起来使用。

采用E-R模型方法的概念结构设计

❖ 设计局部E-R模型

E-R模型的设计内容包括确定局部E-R模型的范围、定义实体、联系以及它们的属性。

❖ 设计全局E-R模型

将所有局部E-R图集成为一个全局E-R图，即全局E-R模型。

❖ 优化全局E-R模型

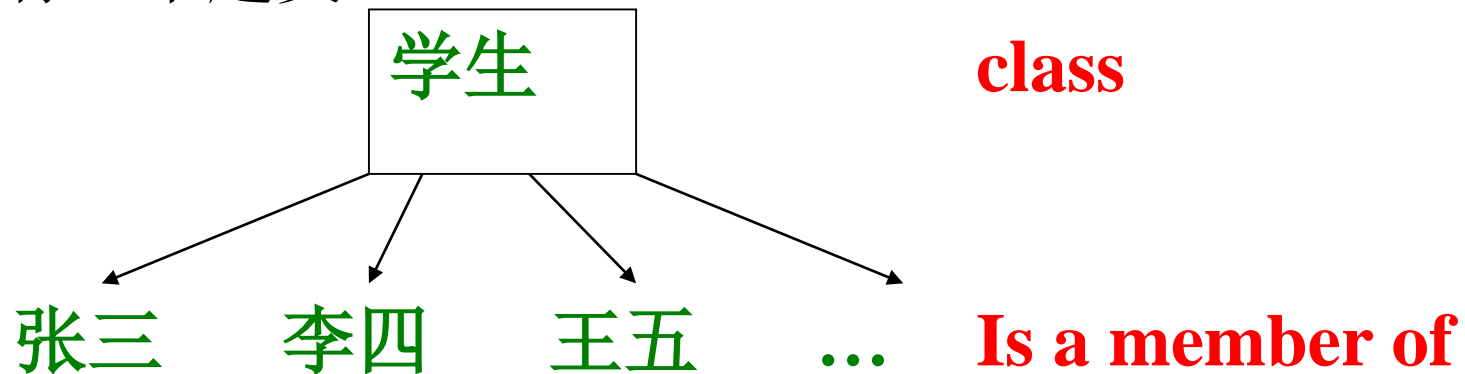
设计局部E-R模型

- ❖ 概念结构是对现实世界的一种抽象。
- ❖ 所谓抽象是对实际的人、物、事和概念进行人为处理，抽取所关心的共同特性，忽略非本质细节，并把这些特性用各种概念准确的加以描述。
- ❖ 一般有三种抽象方法：
 - 分类
 - 概括
 - 聚集

分类

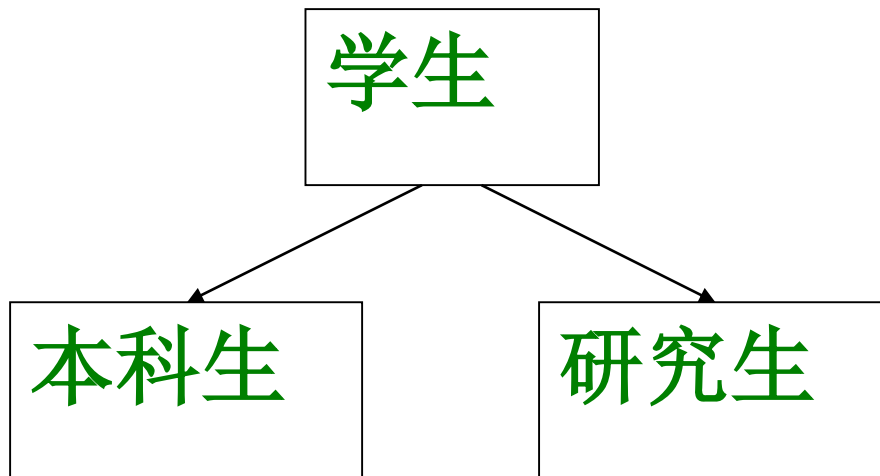
(1) 在相似的个体之间提取共性，建立“类”的概念（集合）。

- **个体与个体之间**：具有相似的状态与行为，有相同的描述结构，相互用主码值区分。
- **个体与类之间**：个体 **Is a member of** 类（子类有且仅有一个超类）。



概括

(2) 定义实体之间的一种子集联系。



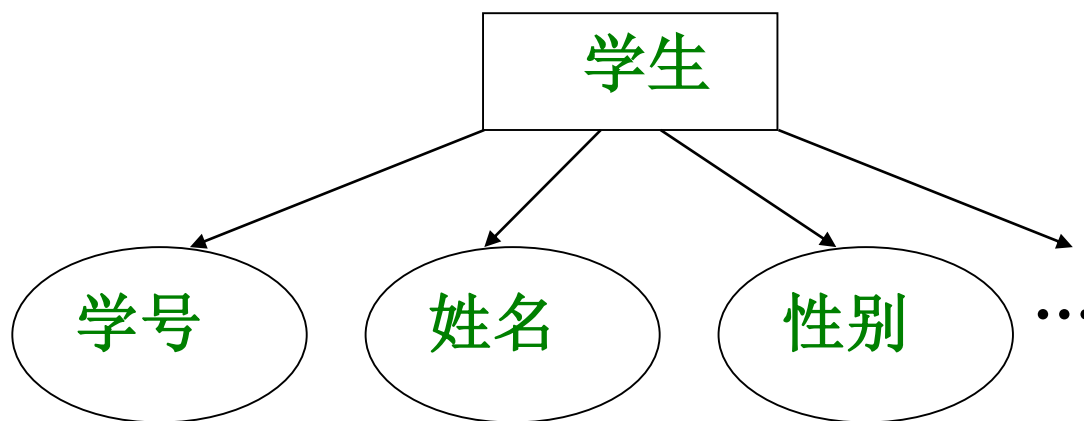
超类

Is a subset of

子类

聚集

❖ (3) 定义了某一类型的组成成分。



实体型

Is a part of

属性

局部视图设计

设计分E-R图的步骤：

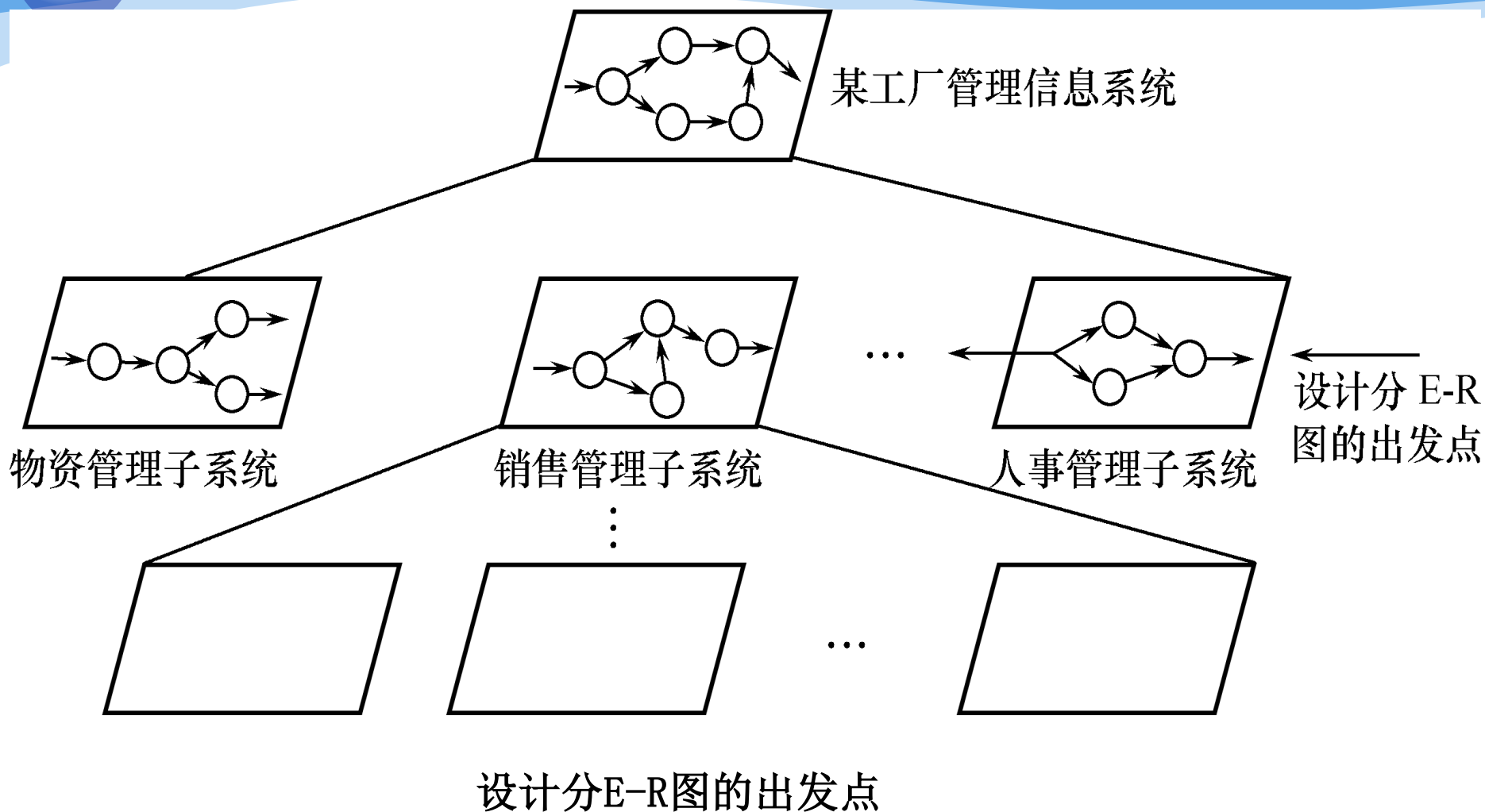
1.选择局部应用

2.逐一设计分E-R图

1. 选择局部应用

- 在多层的数据流图中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
- 通常以中层数据流图作为设计分E-R图的依据

选择局部应用（续）



2. 逐一设计分E-R图

任务

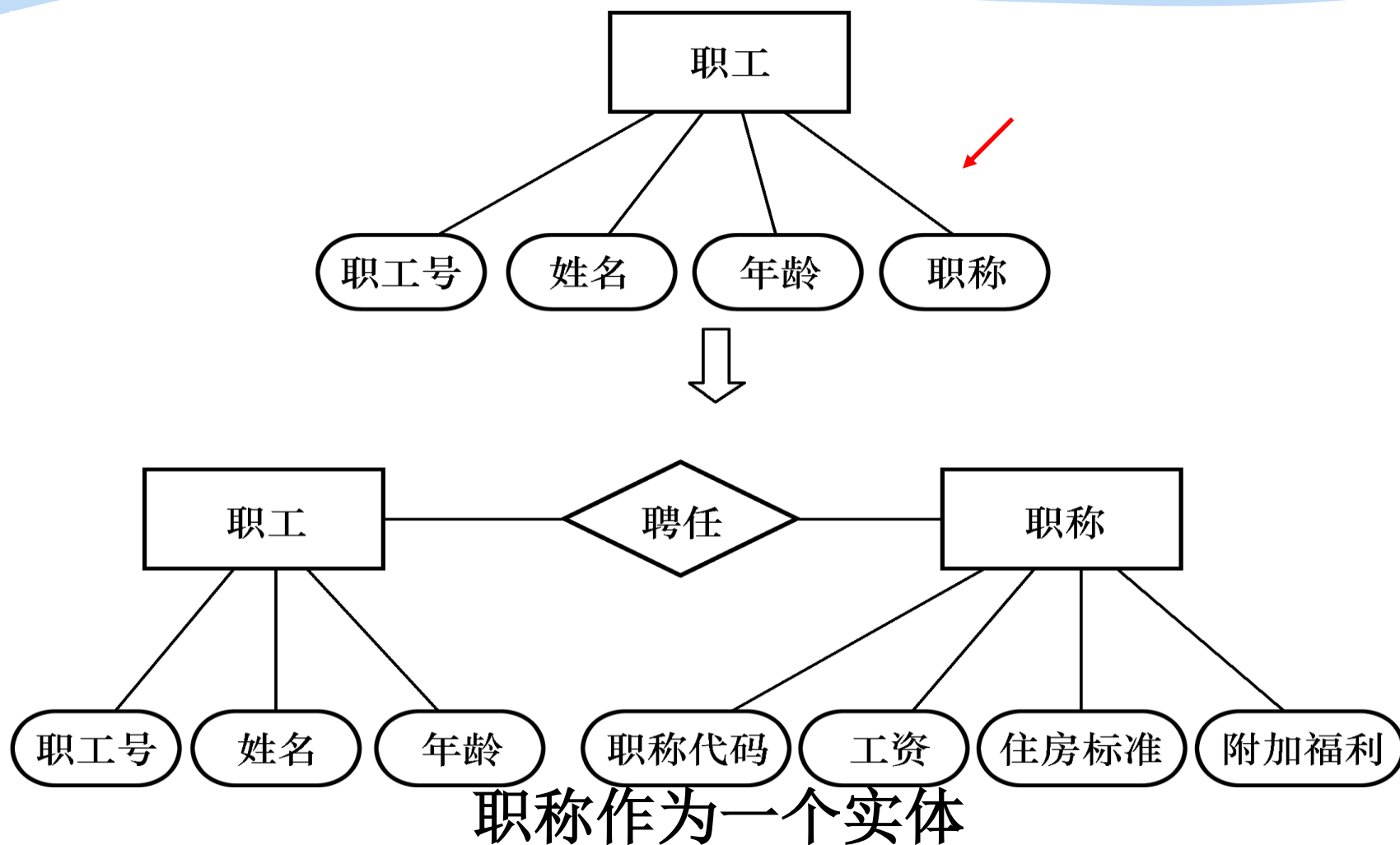
- 将各局部应用涉及的数据分别从数据字典中抽取出来
- 参照数据流图，标定各局部应用中的实体、实体的属性、标识实体的码(根据数据字典中数据流来画数据流图)
- 确定实体之间的联系及其类型（1:1，1:n，m:n）

逐一设计分E-R图（续）

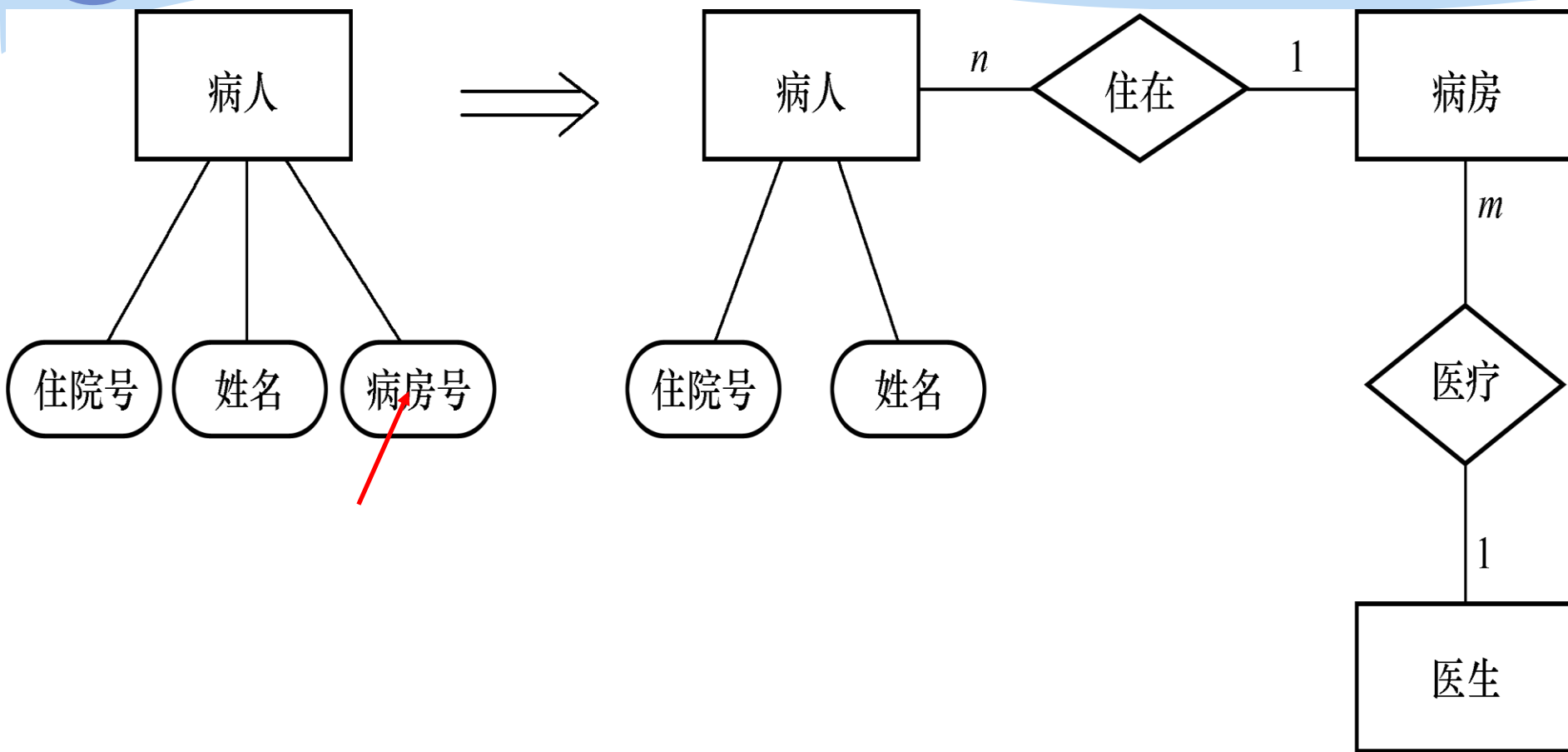
两条准则：

- （1）属性不能再具有需要描述的性质。即属性必须是不可分的数据项，不能再由另一些属性组成
- （2）属性不能与其他实体具有联系。联系只发生在实体之间

逐一设计分E-R图（续）

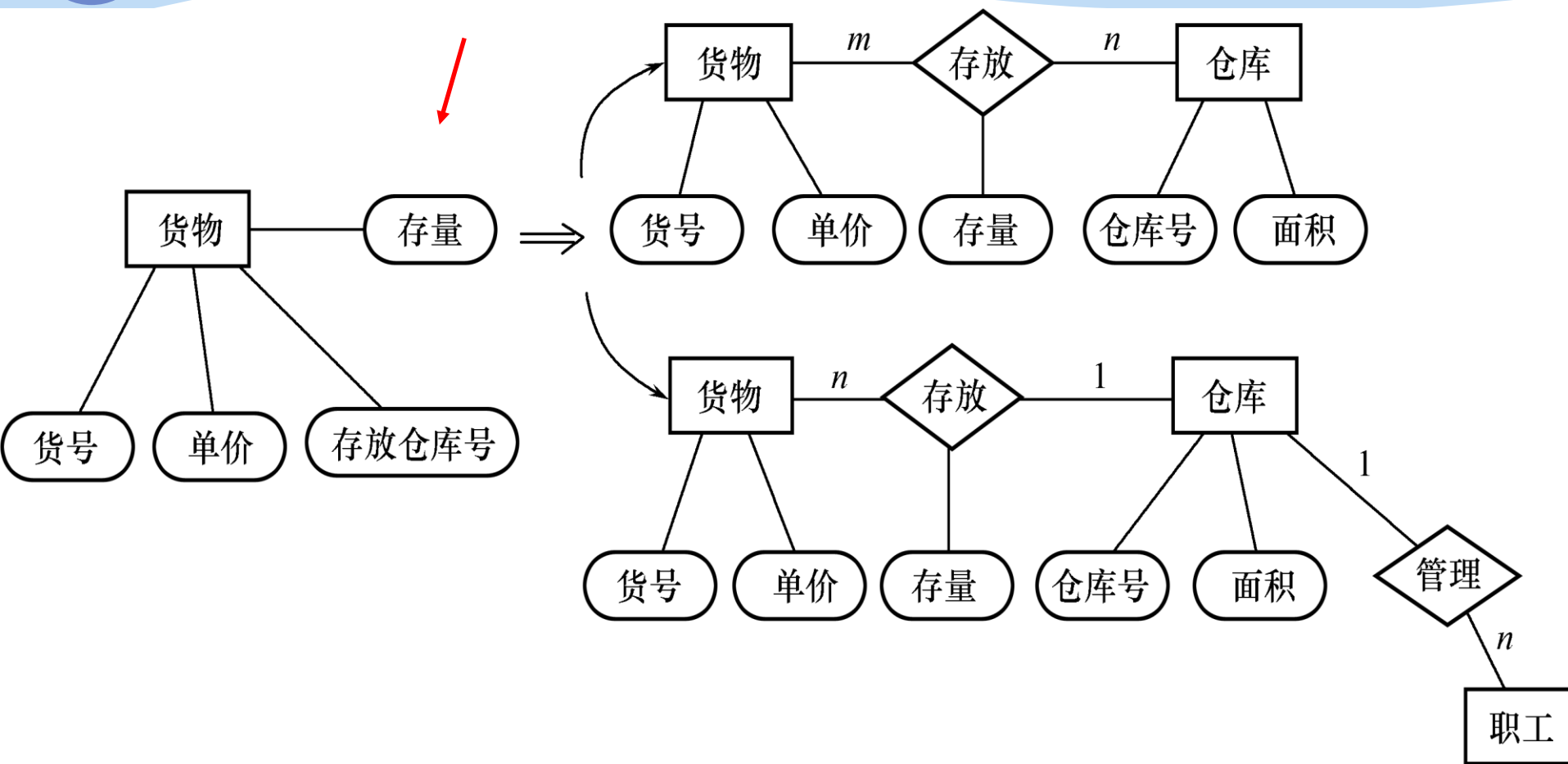


逐一设计分E-R图（续）



病房作为一个实体

逐一设计分E-R图（续）



仓库作为一个实体

逐一设计分E-R图（续）

〔实例〕销售管理子系统分E-R图的设计

销售管理子系统的主要功能：

- 处理顾客和销售员送来的订单
- 工厂是根据订货安排生产的
- 交出货物同时开出发票
- 收到顾客付款后，根据发票存根和信贷情况进行应收款处理

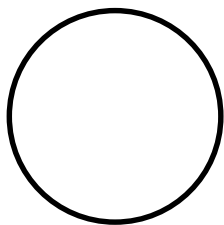
逐一设计分E-R图——数据流图



流动数据



数据的源点或终点

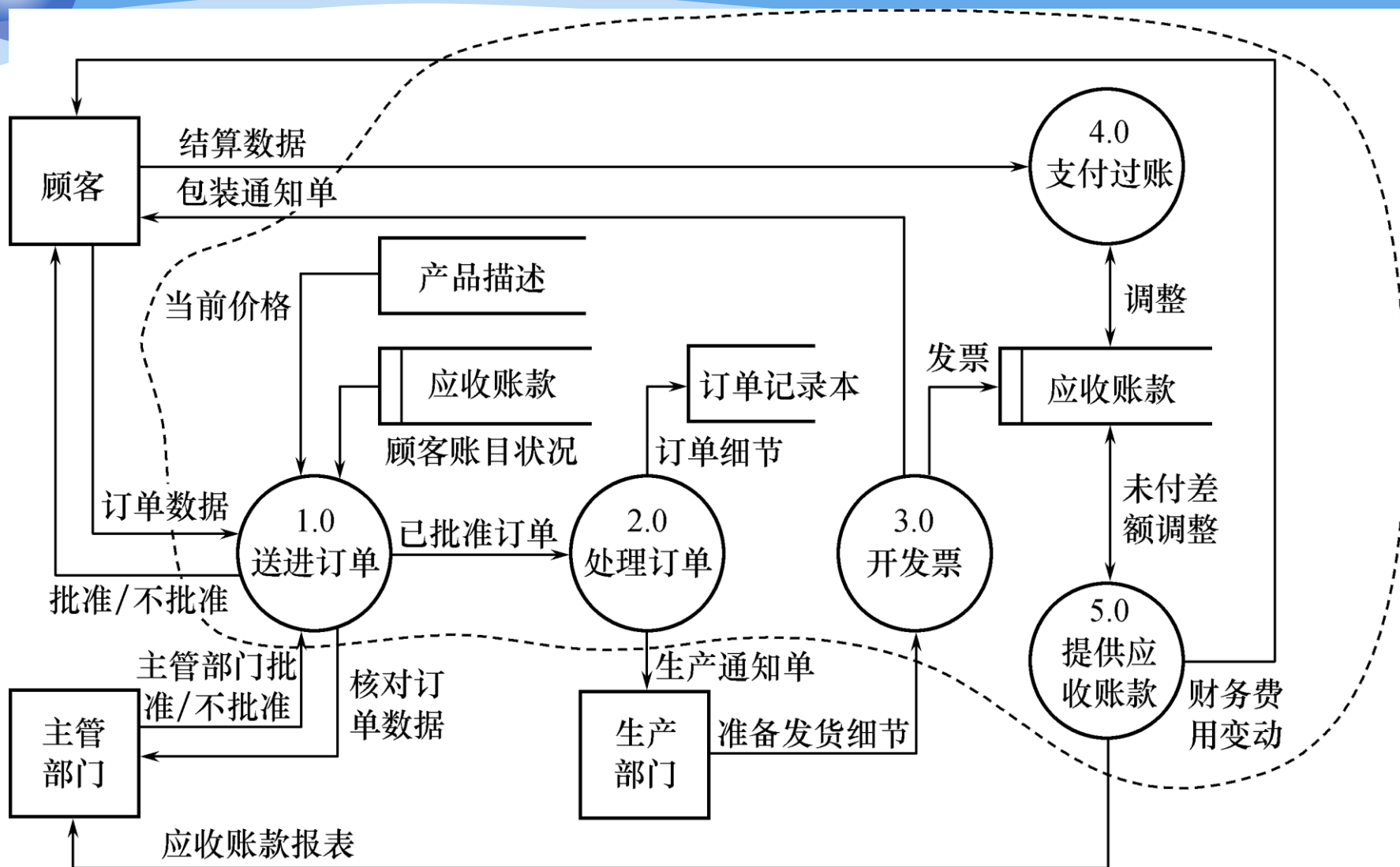


数据处理应用

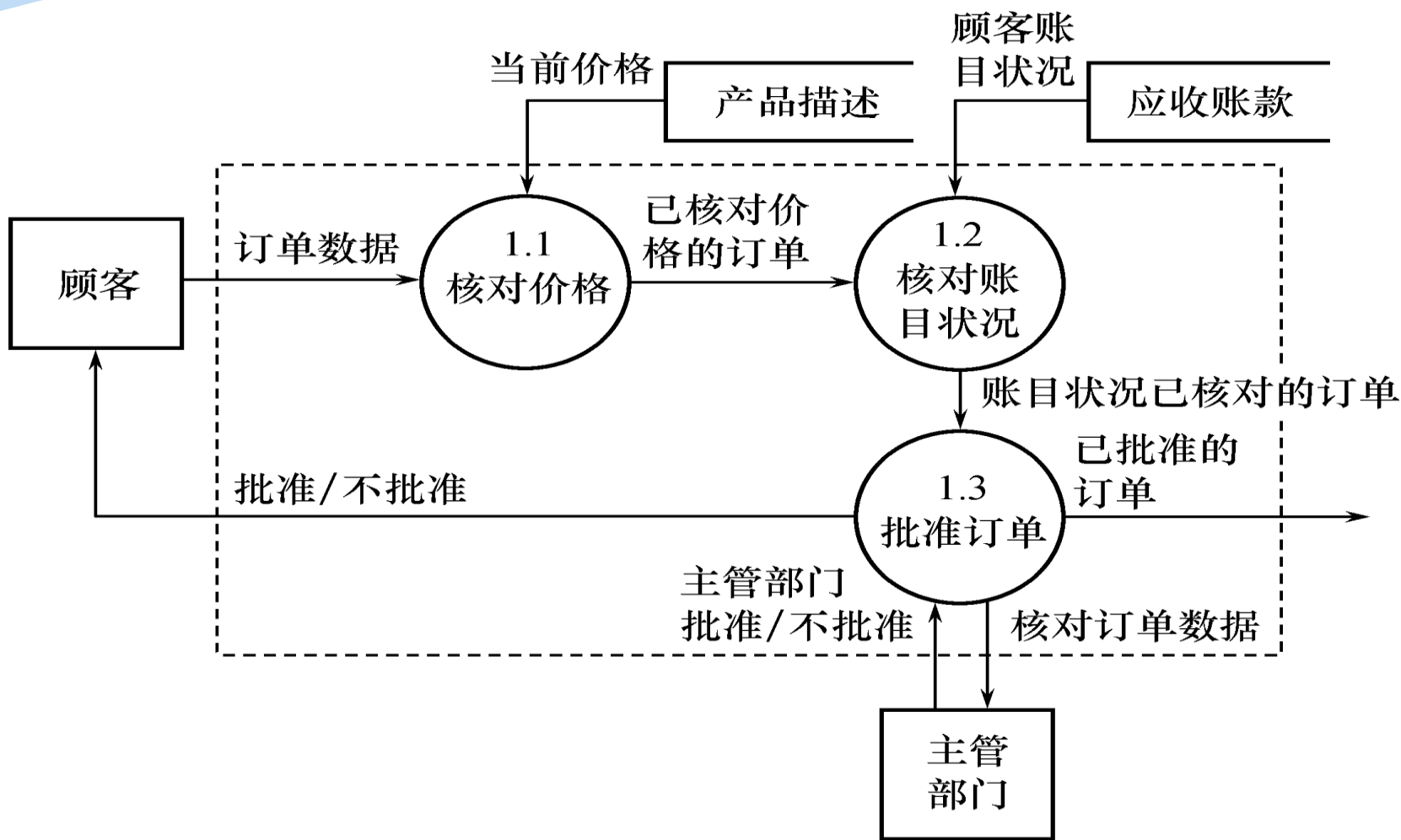


需要存储的数据

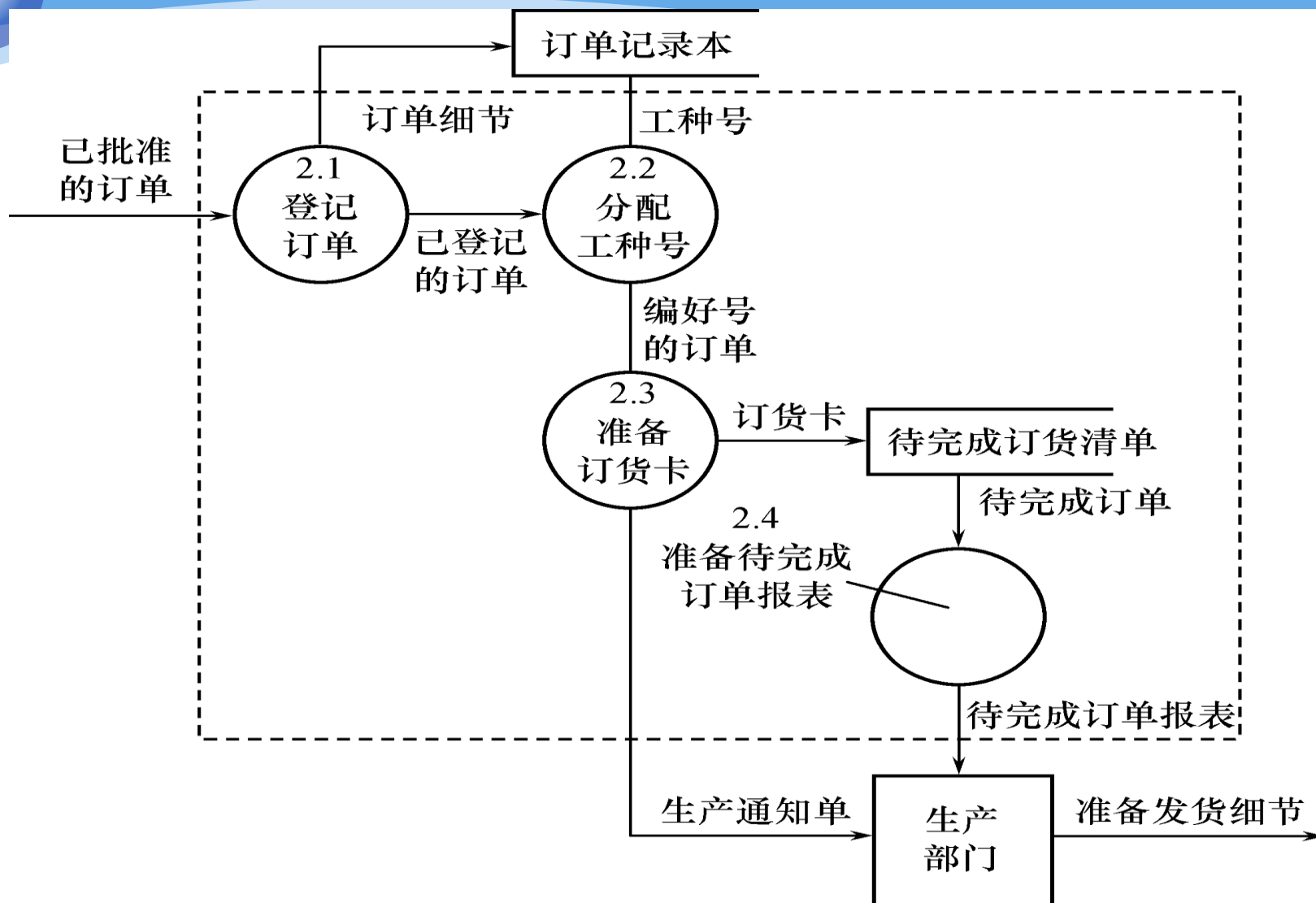
销售管理子系统第一层数据流图



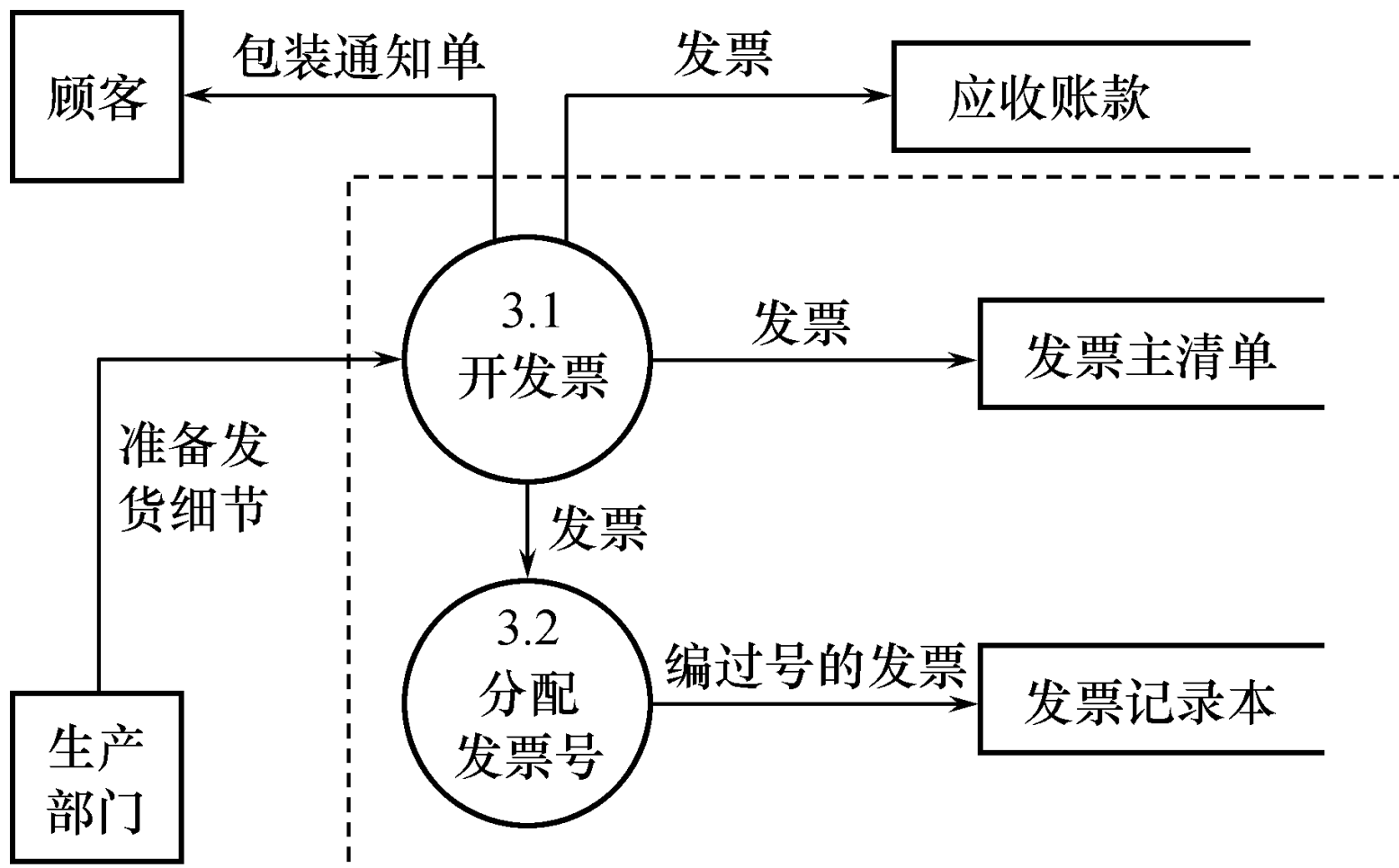
第二层数据流图——接受订单



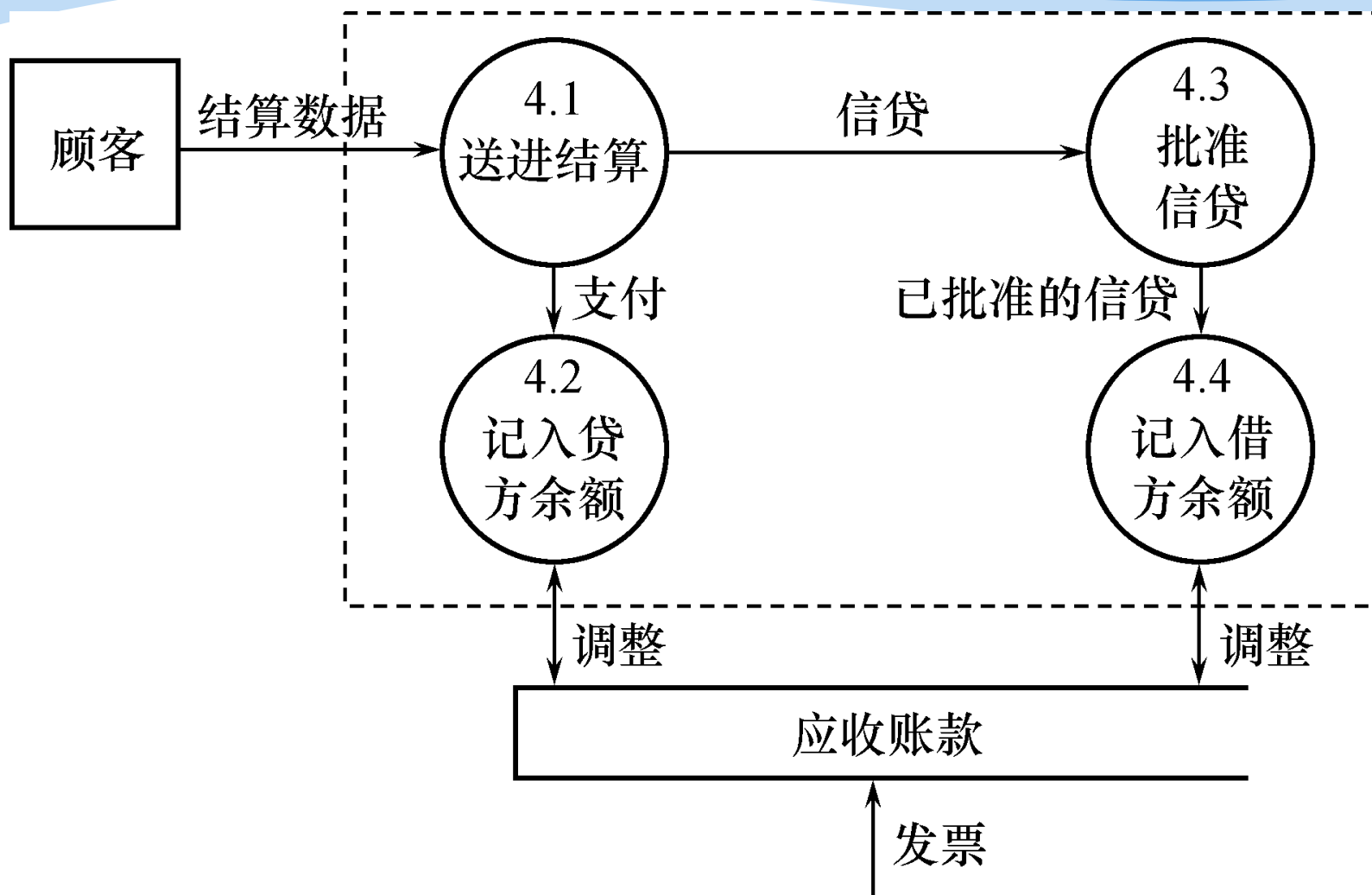
第二层数据流图——处理订单



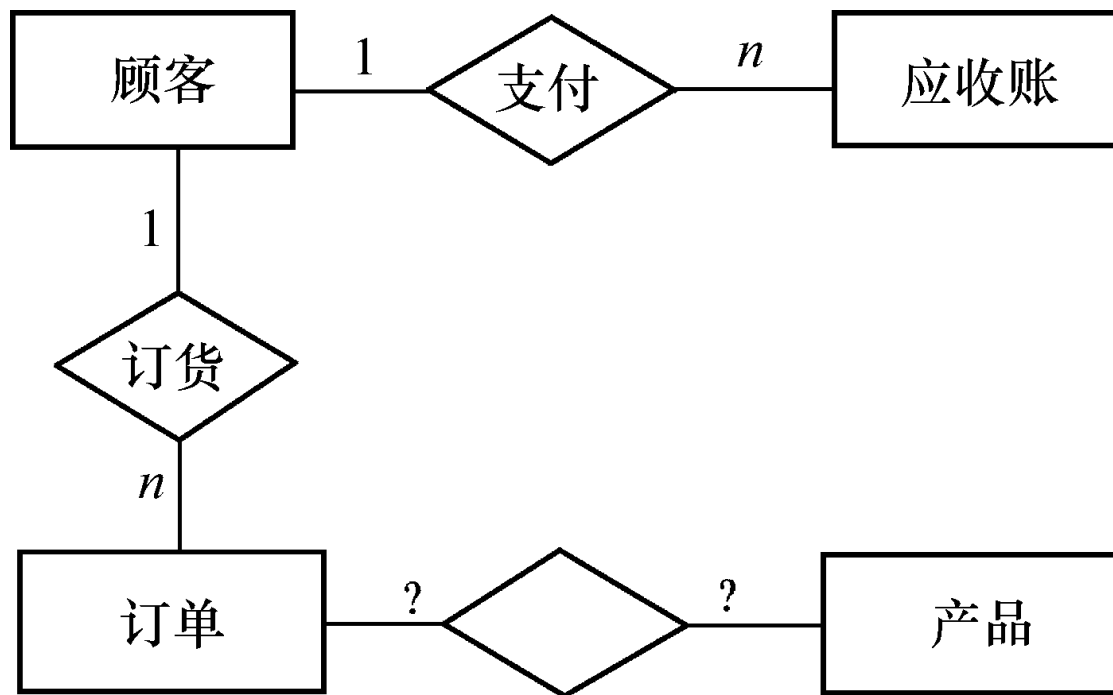
第二层数据流图——开发票



第二层数据流图——支付过账



逐一设计分E-R图（续）



分E-R图的框架

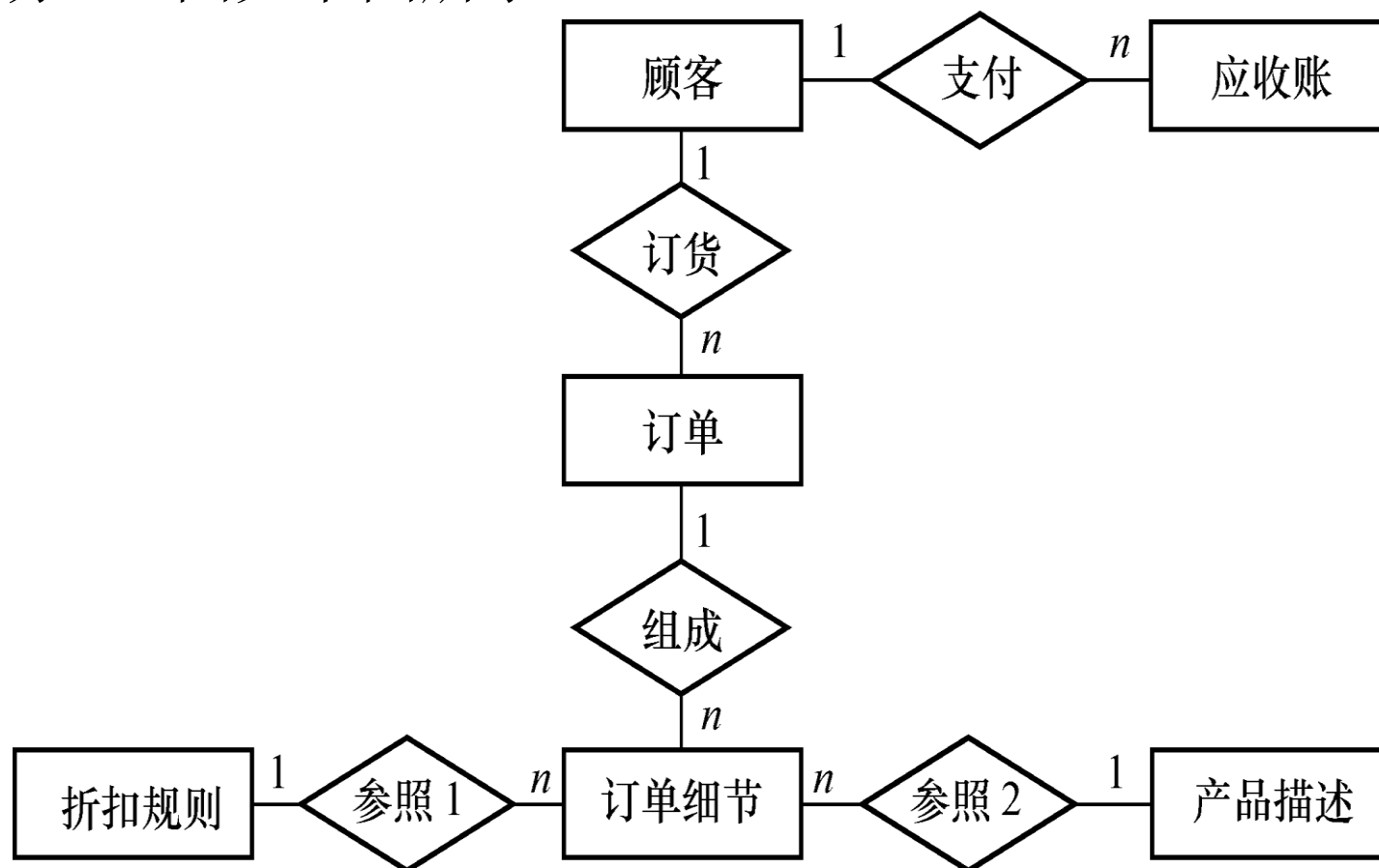
逐一设计分E-R图（续）

参照第二层数据流图和数据字典，遵循两个准则，进行如下调整：

- (1) 订单与订单细节是1： n 的联系。
- (2) 原订单和产品的联系实际上是订单细节和产品的联系。
- (3) “发票主清单”是一个数据存储，不作为实体加入分E-R图
- (4) 工厂对大宗订货给予优惠。

逐一设计分E-R图（续）

得到分E-R图如下图所示



销售管理子系统的分E-R图

逐一设计分E-R图（续）

对每个实体定义的属性如下：

顾客：{顾客号，顾客名，地址，电话，信贷状况，账目余额}

订单：{订单号，顾客号，订货项数，订货日期，交货日期，
工种号，生产地点}

订单细则：{订单号，细则号，零件号，订货数，金额}

应收账款：{顾客号，订单号，发票号，应收金额，支付日期，
支付金额，当前余额，贷款限额}

产品描述：{产品号，产品名，单价，重量}

折扣规则：{产品号，订货量，折扣}

小结

需求分析

概念结构设计的步骤

- 抽象数据并设计局部视图

数据抽象：分类、
聚集、概括

标定局部的实体、
属性、码、联系，
用E-R图描述出来

8.3.1 概念结构设计

8.3.1 概念结构

8.3.2 概念结构设计的方法与步骤

8.3.3 数据抽象与局部视图设计

8.3.4 视图的集成

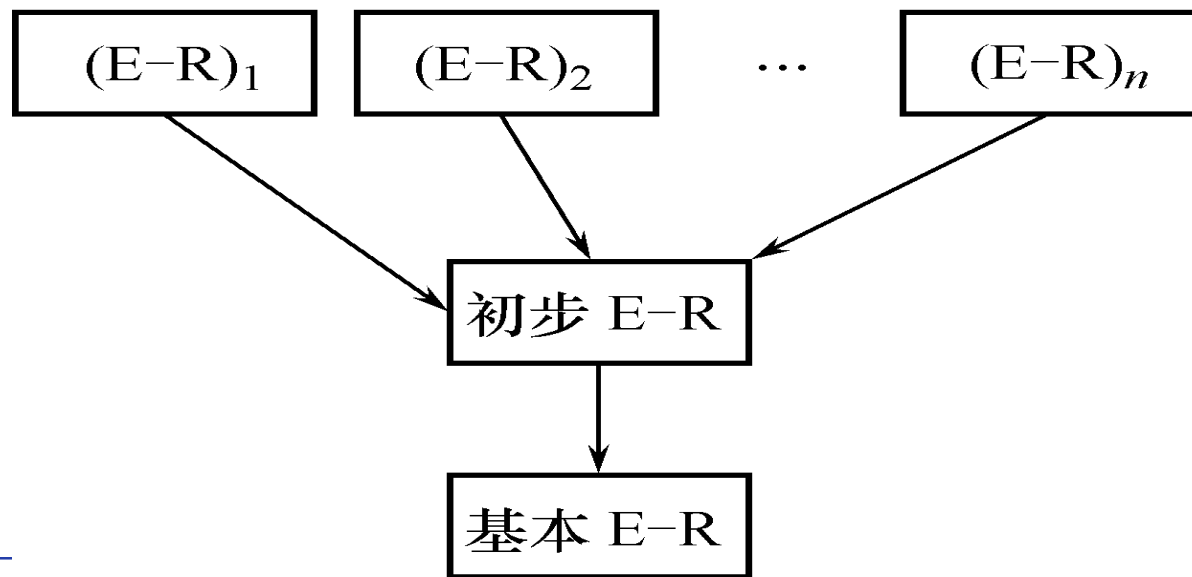
全局视图的集成

- ❖ 各个局部视图即分E-R图建立好后，还需要对它们进行合并，集成为一个整体的数据概念结构即总E-R图。

视图集成的两种方式

多个分E-R图一次集成

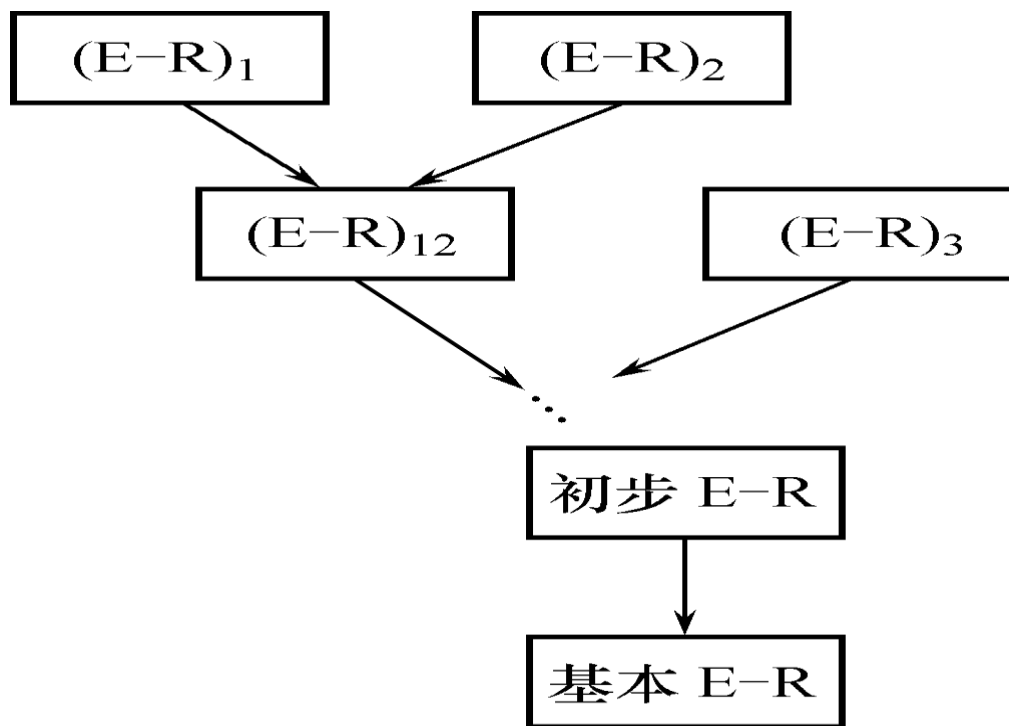
- 一次集成多个分E-R图
- 通常用于局部视图比较简单时



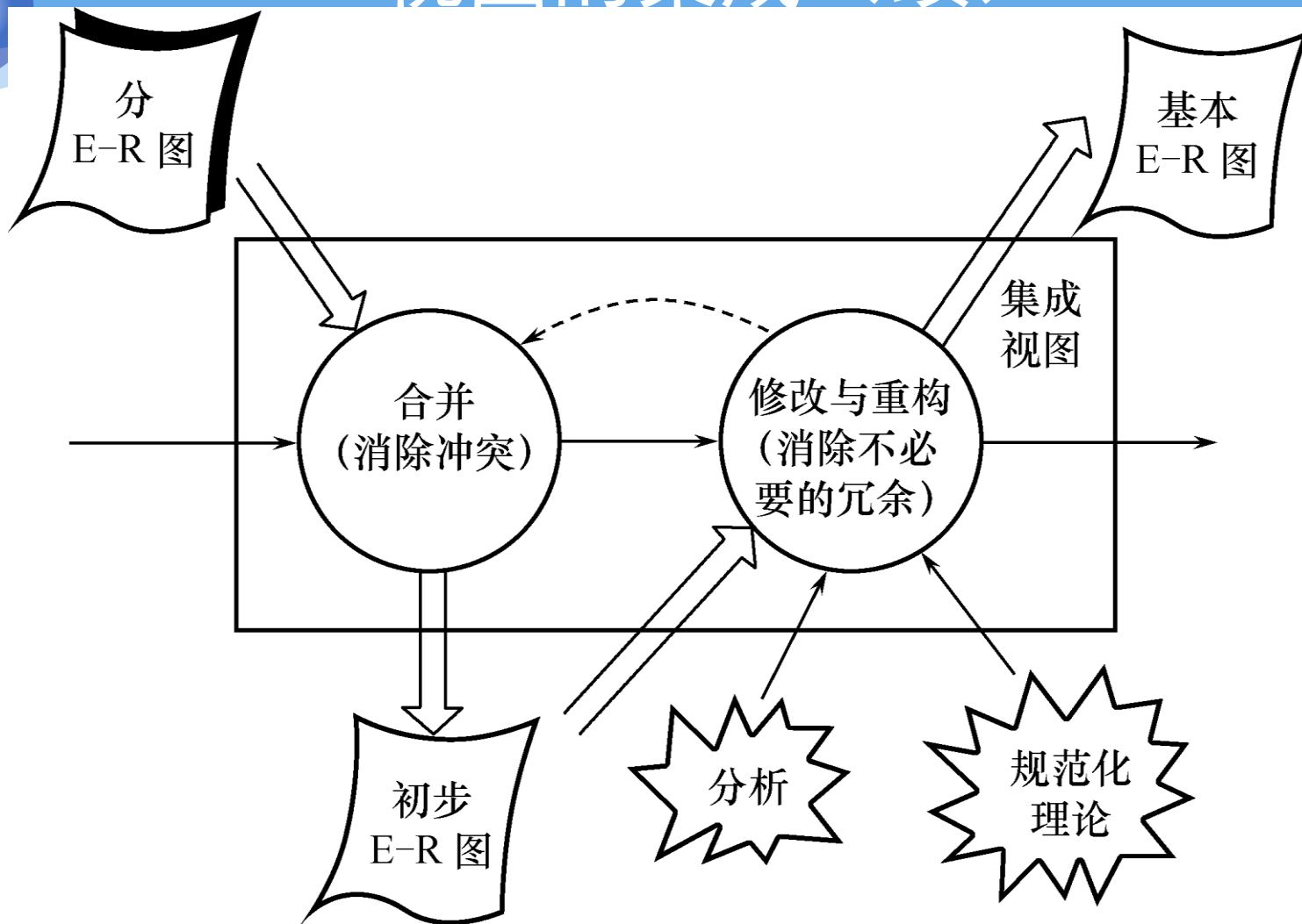
视图的集成（续）

逐步集成

- 用累加的方式一次集成两个分E-R图



视图的集成（续）



视图集成

合并分E-R图，生成初步E-R图

各分E-R图存在冲突

- 各个分E-R图之间必定会存在许多不一致的地方

合并分E-R图的主要工作与关键

- 合理消除各分E-R图的冲突

合并分E-R图，生成初步E-R图（续）

冲突的种类

1. 属性冲突
2. 命名冲突
3. 结构冲突

1. 属性冲突

两类属性冲突

协商解决

- 属性域冲突
 - 属性值的类型
 - 取值范围
 - 取值集合不同
- 属性取值单位冲突

2. 命名冲突

两类命名冲突

协商解决

- **同名异义**：不同意义的对象在不同的局部应用中具有相同的名字
- **异名同义（一义多名）**：同一意义的对象在不同的局部应用中具有不同的名字

3. 结构冲突

三类结构冲突

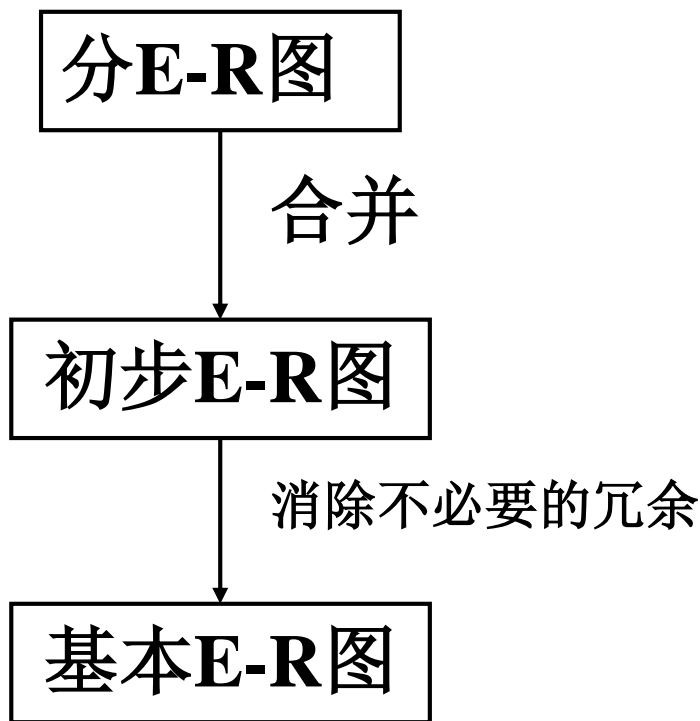
- 同一对象在不同应用中具有不同的抽象
- 同一实体在不同分**E-R**图所包含的属性个数和属性排列次序不完全相同
- 实体之间的联系在不同局部视图中呈现不同的类型

怎么办？

消除不必要的冗余，设计基本E-R图

基本任务

- 消除不必要的冗余，设计生成基本E-R图



可能存在冗余的数据
和冗余的实体间联系

1. 冗余

- ❖ 冗余的数据是指可由基本数据导出的数据
冗余的联系是指可由其他联系导出的联系
- ❖ 冗余数据和冗余联系容易破坏数据库的完整性，
给数据库维护增加困难
- ❖ 消除不必要的冗余后的初步E-R图称为基本E-R图

消除冗余的方法

分析方法

- 以数据字典和数据流图为依据
- 根据数据字典中关于数据项之间的逻辑关系
- 规范化理论

函数依赖的概念提供了消除冗余联系的形式化工具

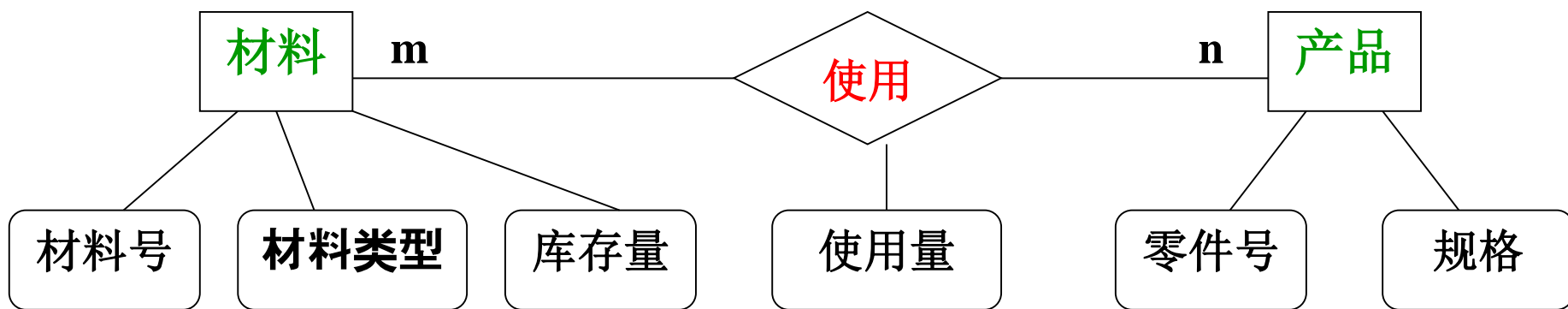
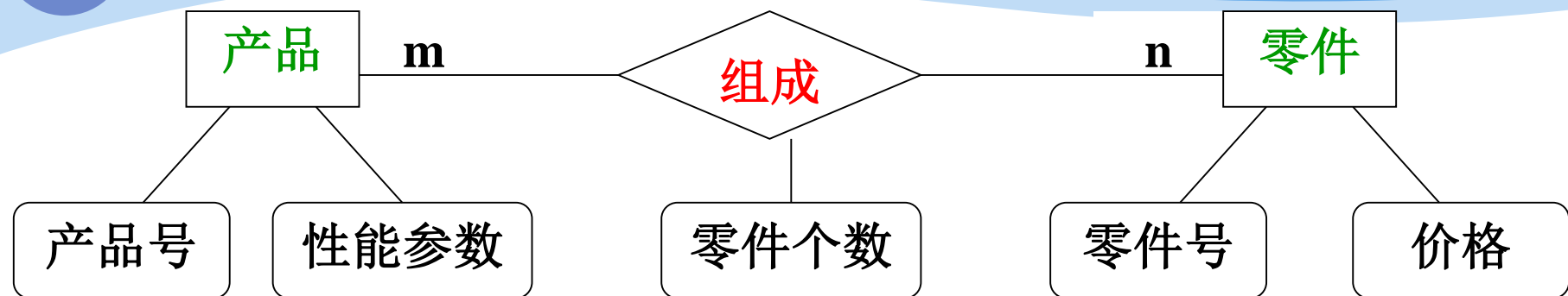
设计全局E-R模型

- ❖ 将局部E-R图集成为全局E-R图；
- ❖ 需消除各分E-R图合并时产生的冲突；
- ❖ 解决冲突是合并E-R图的主要工作和关键所在。
- ❖ 冲突主要有三类：
 - 属性冲突：属性域冲突、属性取值单位冲突
 - 命名冲突：同名异义和异名同义
 - 结构冲突：同一对象在不同应用中具有不同的抽象、同一实体在不同的局部E-R图所包含的属性个数和属性的排列次序不完全相同。

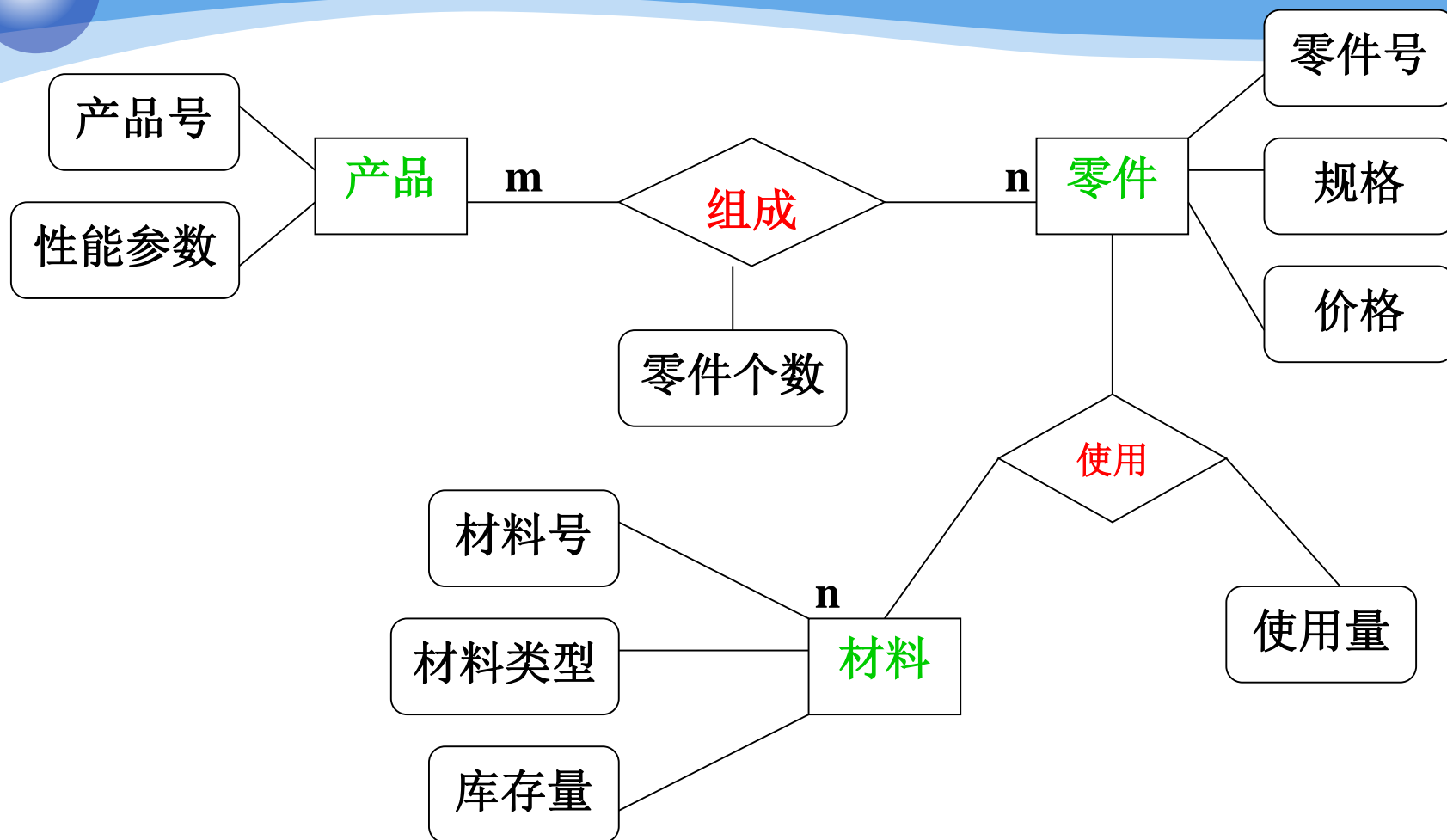
优化全局E-R模型

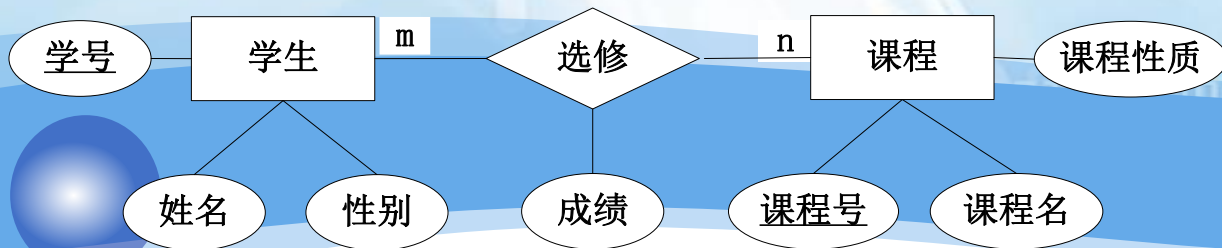
- 实体个数尽可能少；
- 实体所包含的属性尽可能少；
- 实体间联系无冗余。

局部E-R图

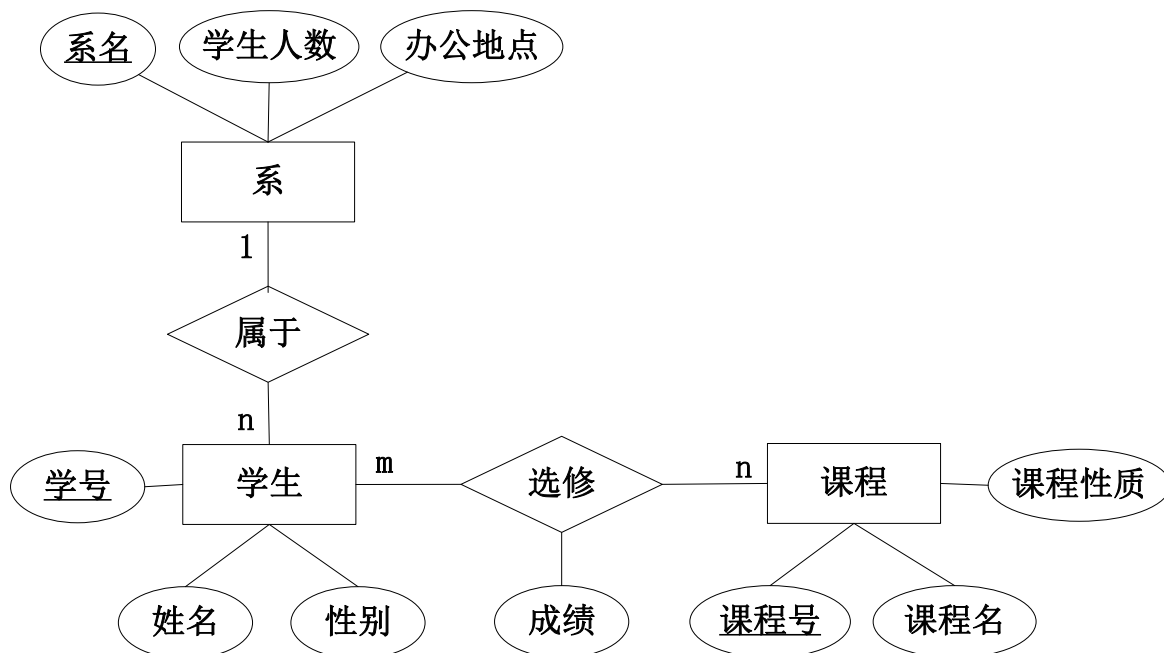
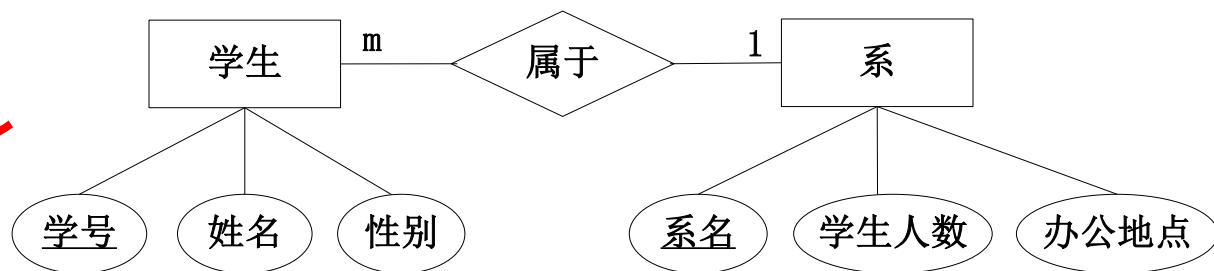


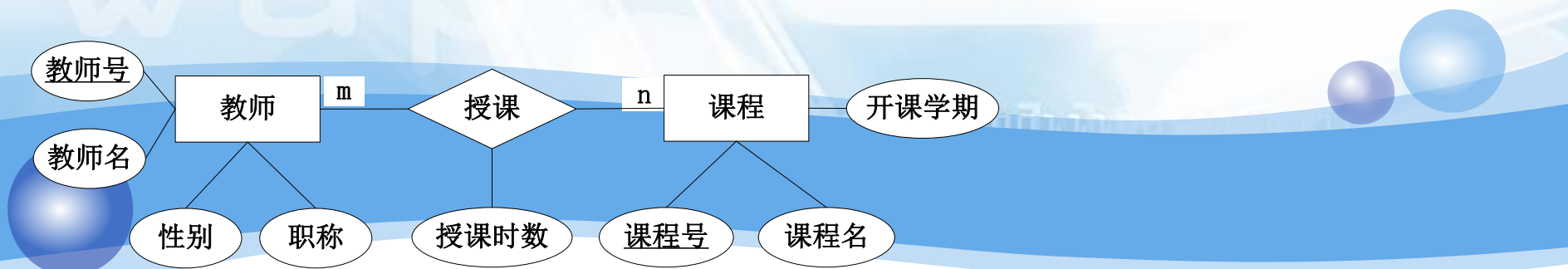
合并示例



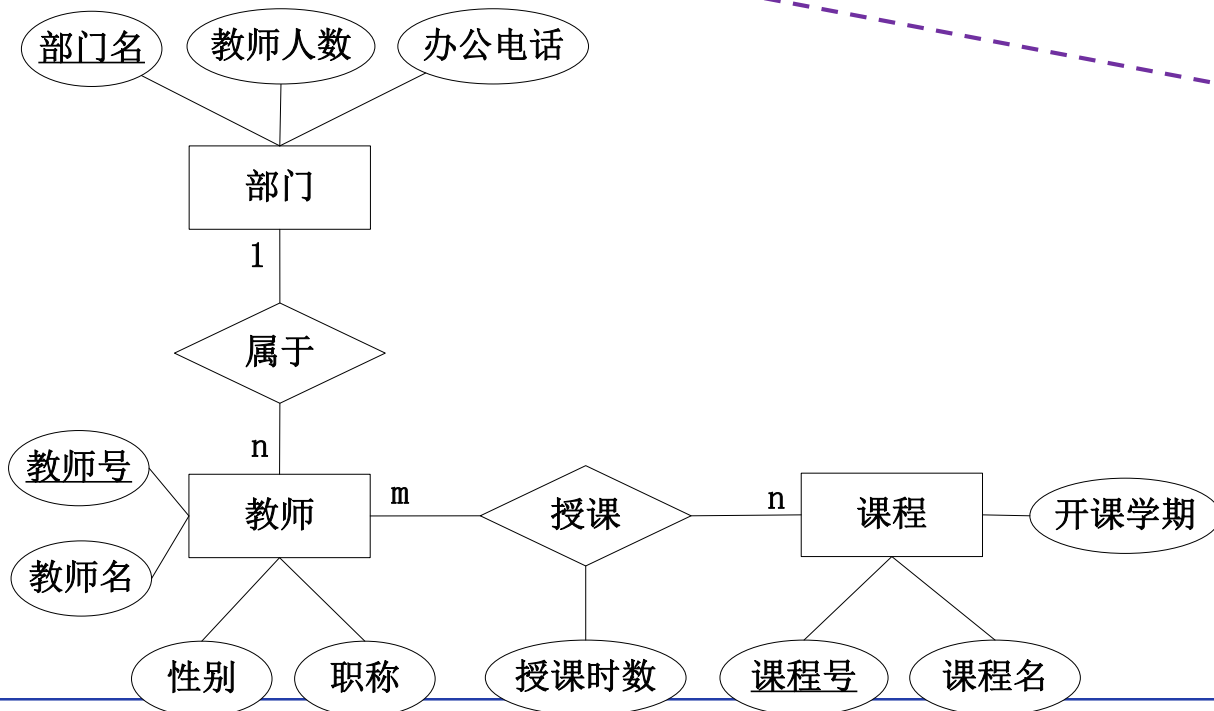
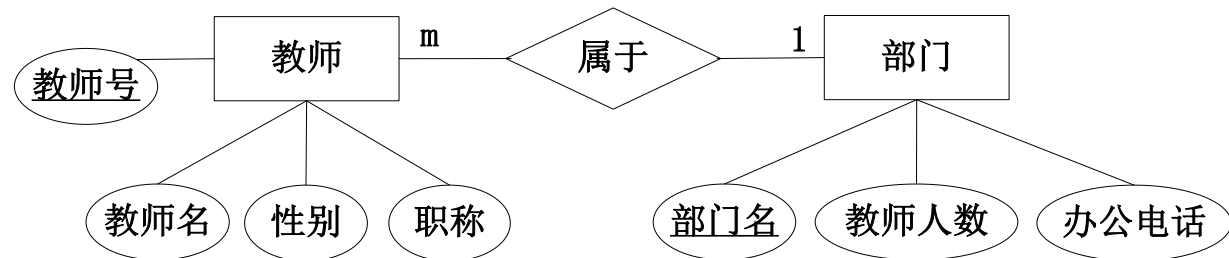


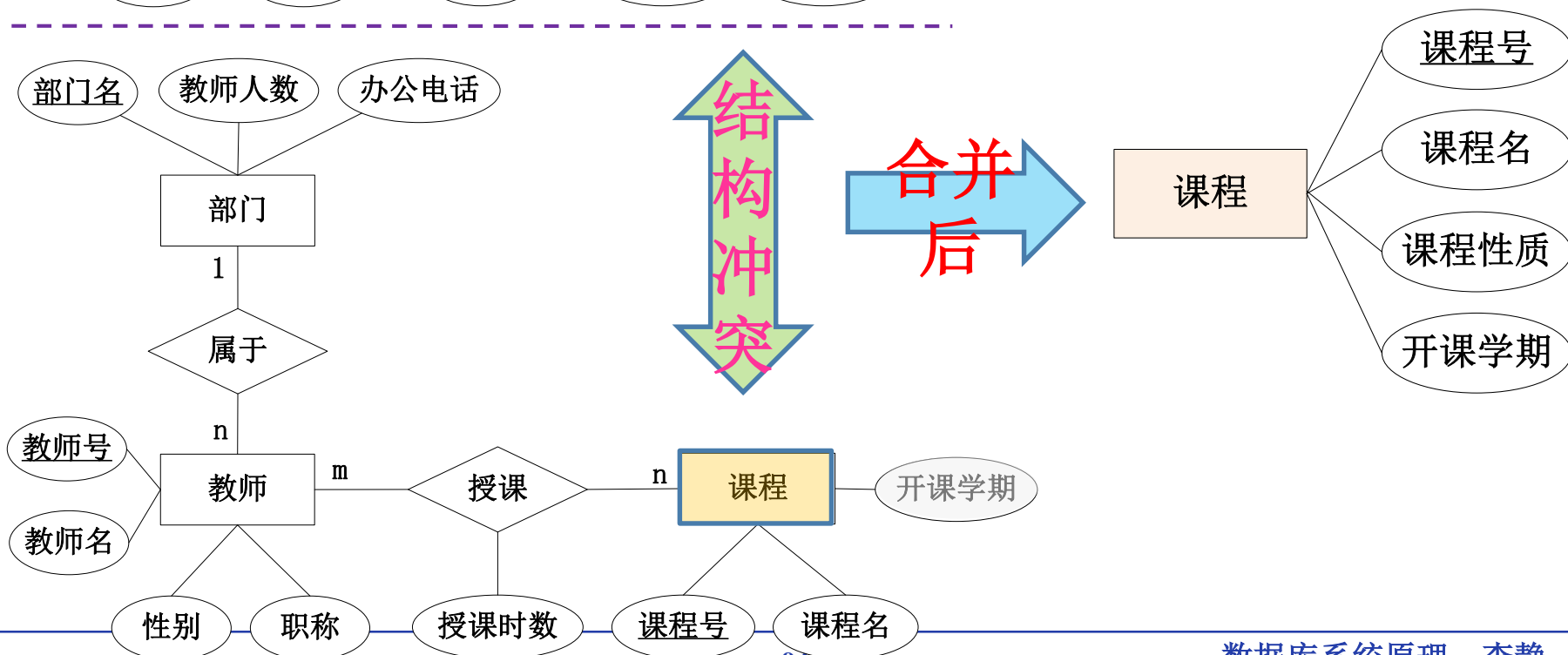
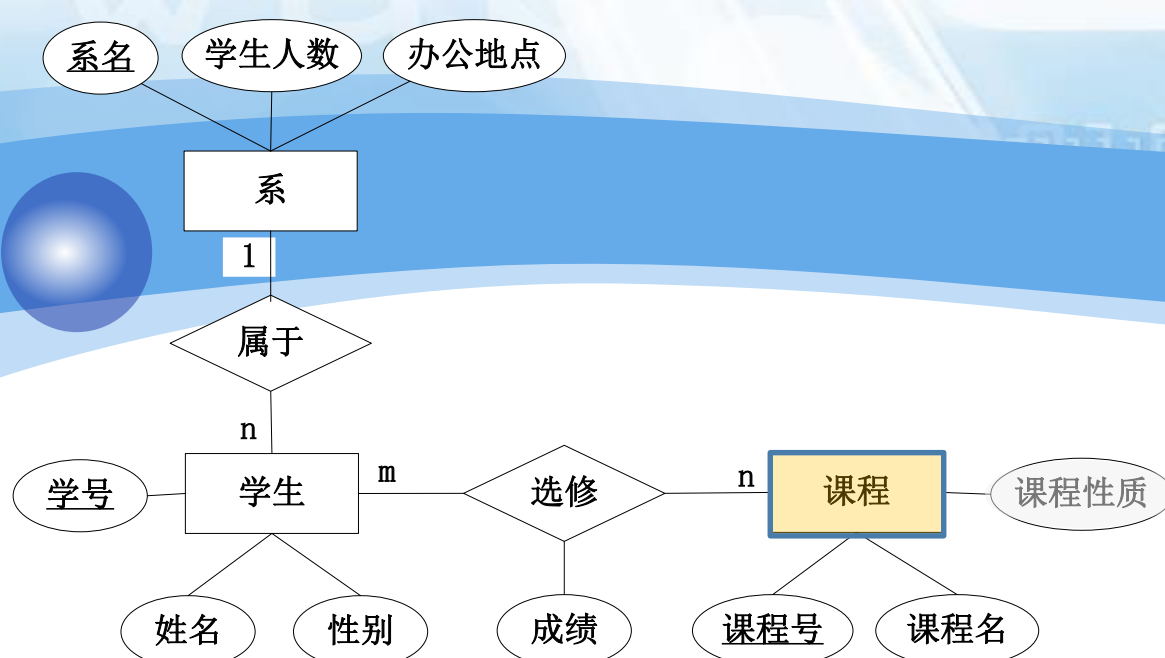
无冲突

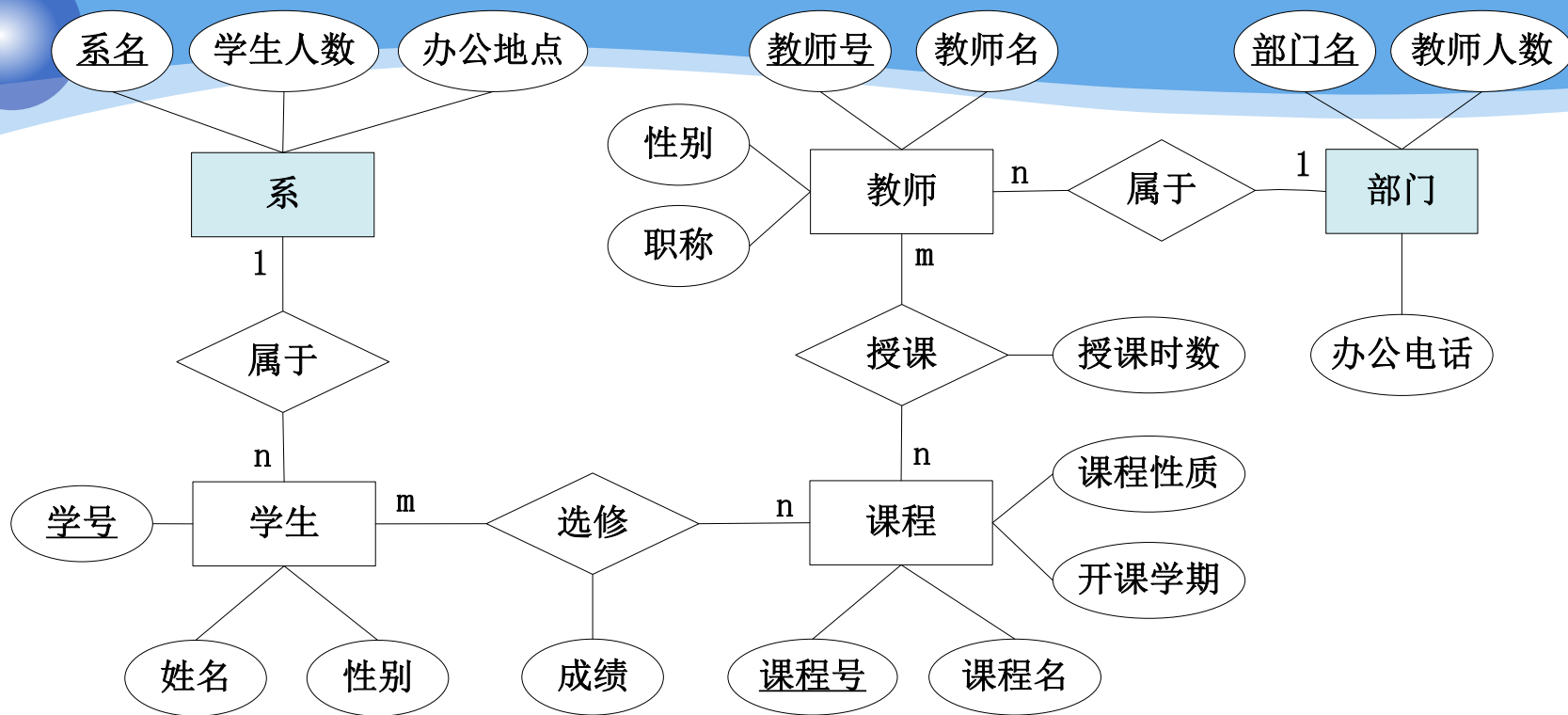




无冲突

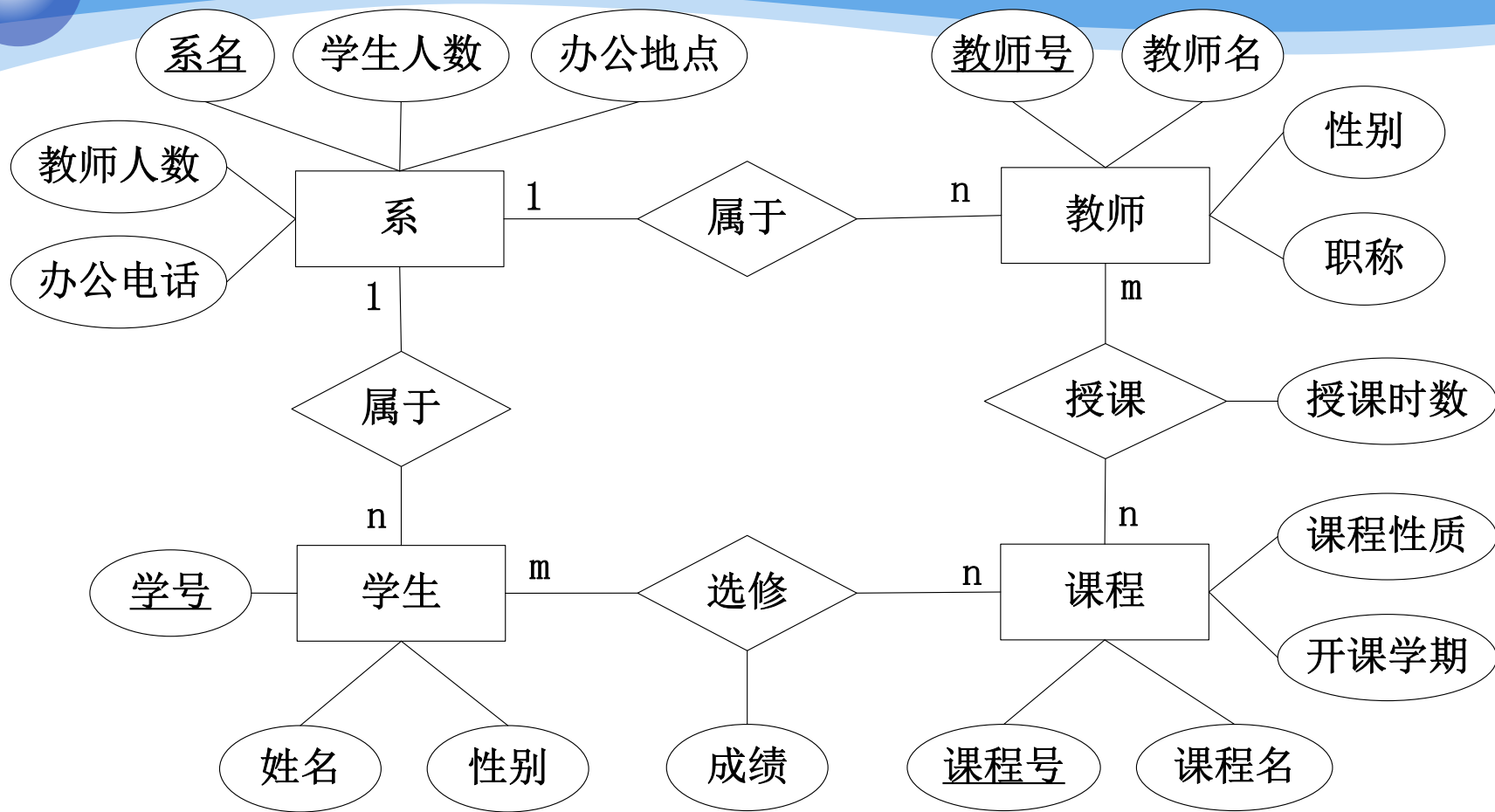




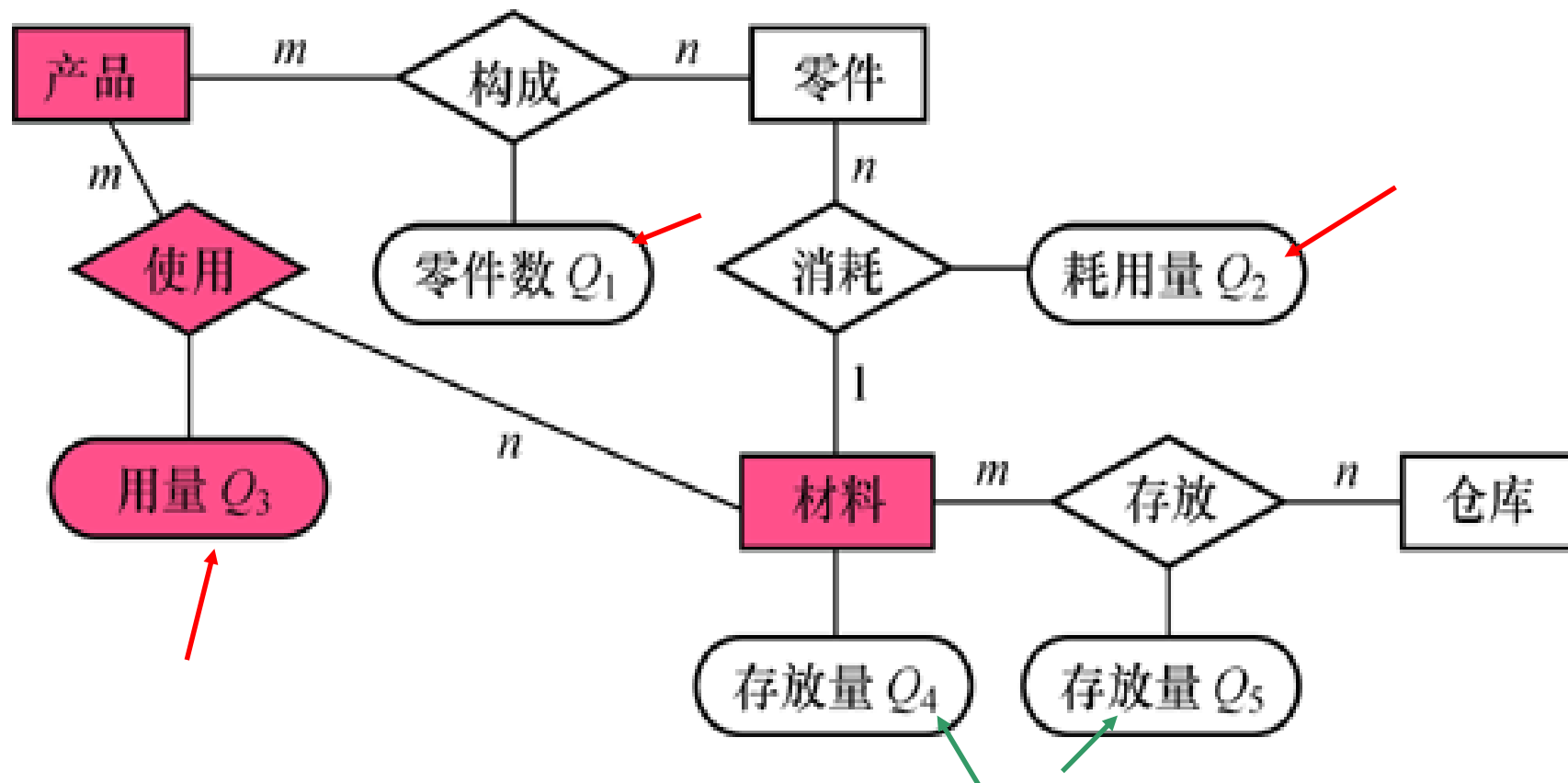


“系”与“部门”：

- ✓ 代表的含义相同，可合并为一个实体。但有：
- ✓ **命名冲突**：“系名”和“部门名”，合并后统一为“系名”。
- ✓ **结构冲突**：合并后为属性并集。



消除冗余的方法（续）



消除冗余

验证整体概念结构

视图集成后形成一个整体的数据库概念结构，还必须进行进一步验证，确保它能够满足下列条件：

- 整体概念结构内部必须具有一致性，不存在互相矛盾的表达
- 整体概念结构能准确地反映原来的每个视图结构，包括属性、实体及实体间的联系
- 整体概念结构能满足需要分析阶段所确定的所有要求

验证整体概念结构（续）

整体概念结构最终还应该提交给用户，征求用户和有关人员的意见，进行评审、修改和优化，然后把它确定下来，作为数据库的概念结构，作为进一步设计数据库的依据。

概念结构设计小结

概念结构设计的步骤

两个准则

- 抽象数据并设计局部视图
- 集成局部视图，消除冲突（属性、命名、结构）
- 验证整体概念结构（修改与重构）
消除不必要的冗余
（规范化理论）

作业



P141 8、9、10