



数据库系统概论

An Introduction to Database System

第七章 数据库设计(续1)

第七章 数据库设计



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 数据库的物理设计

7.6 数据库实施和维护

7.7 小结

7.3 概念结构设计



7.3.1 概念结构

7.3.2 概念结构设计的方法与步骤

7.3.3 数据抽象与局部视图设计

7.3.4 视图的集成

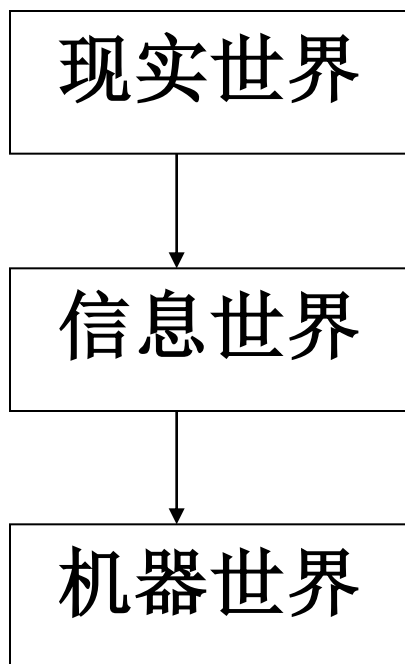
7.3.1 概念结构



❖ 什么是概念结构设计

- 将需求分析得到的用户需求抽象为信息结构即概念模型的过程就是概念结构设计
- 概念结构是各种数据模型的基础，它比数据模型更独立于机器、更抽象，从而更加稳定
- 概念结构设计是整个数据库设计的关键

概念结构（续）



需求分析

概念结构设计

概念结构（续）



❖ 概念结构设计的特点

- (1) 能真实、充分地反映现实世界
- (2) 易于理解
- (3) 易于更改
- (4) 易于向关系、网状、层次等各种数据模型转换

概念结构（续）



❖ 描述概念模型的工具

- E-R模型
- 扩展的E-R模型
 - ISA联系：子类继承父类的属性，也可以有自己的属性
 - 分类属性
 - 不相交约束和可重叠约束
 - 完备性约束
 - 基数约束
 - Part of 联系

7.3 概念结构设计



7.3.1 概念结构

7.3.2 概念结构设计的方法与步骤

7.3.3 数据抽象与局部视图设计

7.3.4 视图的集成

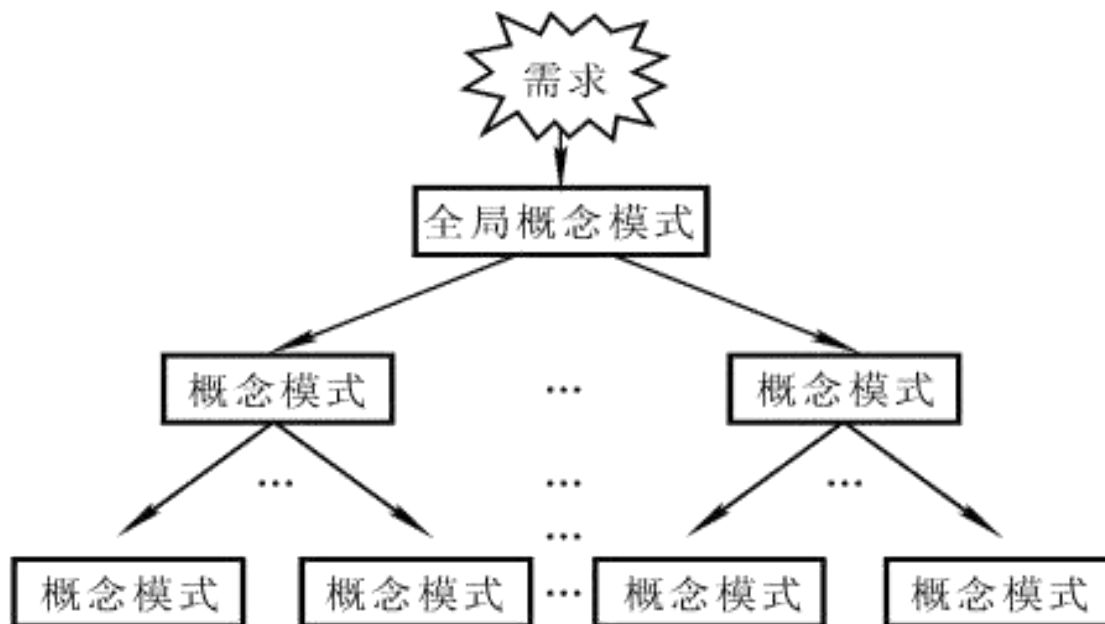
7.3.2 概念结构设计的方法与步骤



❖ 设计概念结构的四类方法

■ 自顶向下

➤ 首先定义全局概念结构的框架，然后逐步细化



自顶向下策略

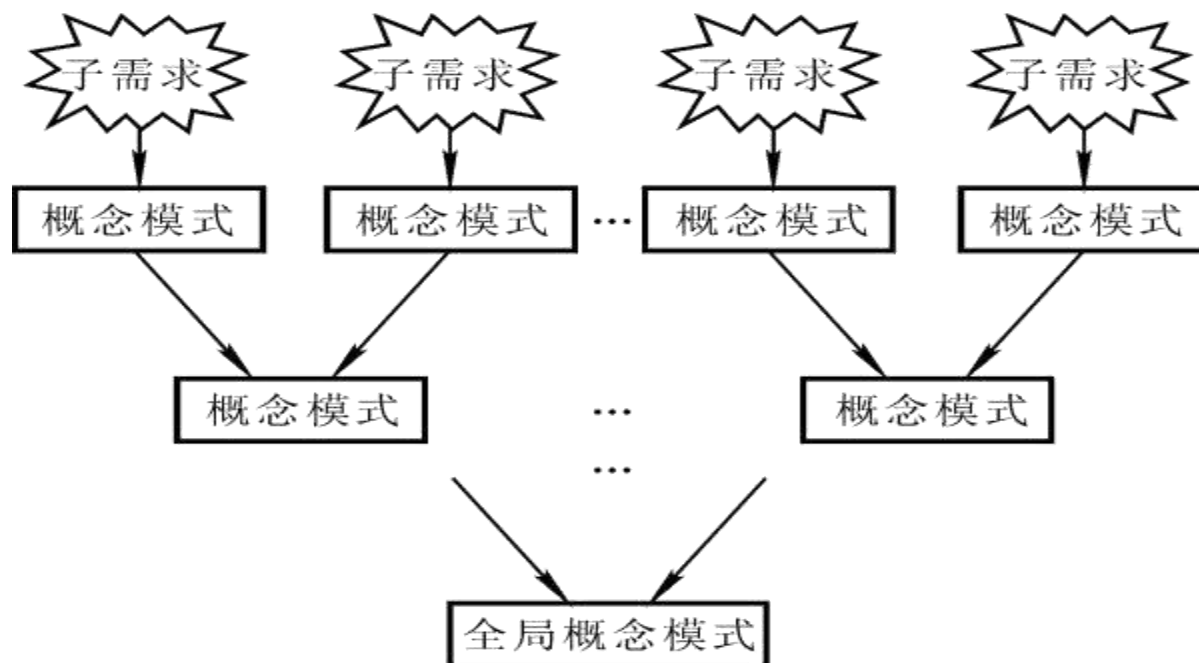
An Introduction to Database System

7.3.2 概念结构设计的方法与步骤



■ 自底向上

- 首先定义各局部应用的概念结构，然后将它们集成起来，得到全局概念结构

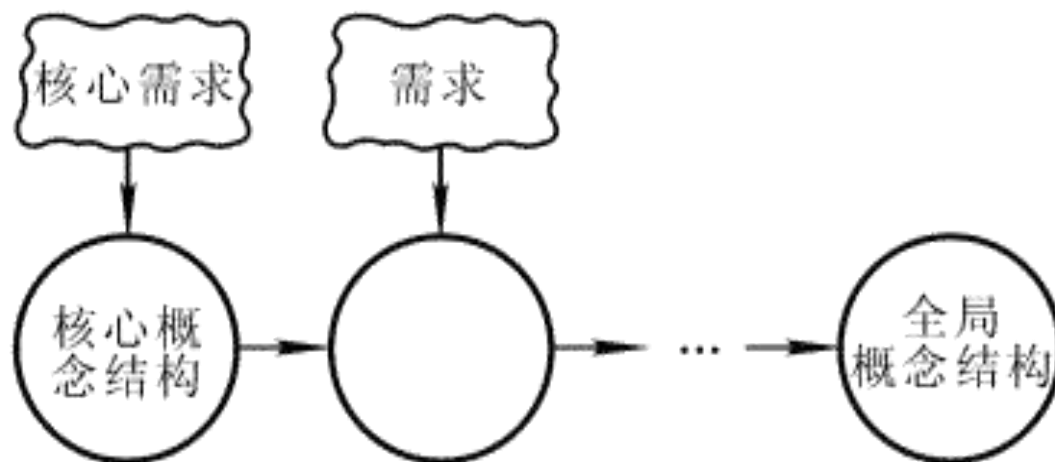


概念结构设计的方法与步骤（续）



■ 逐步扩张

- 首先定义最重要的核心概念结构，然后向外扩充，以滚雪球的方式逐步生成其他概念结构，直至总体概念结构



逐步扩张策略

概念结构设计的方法与步骤（续）



■ 混合策略

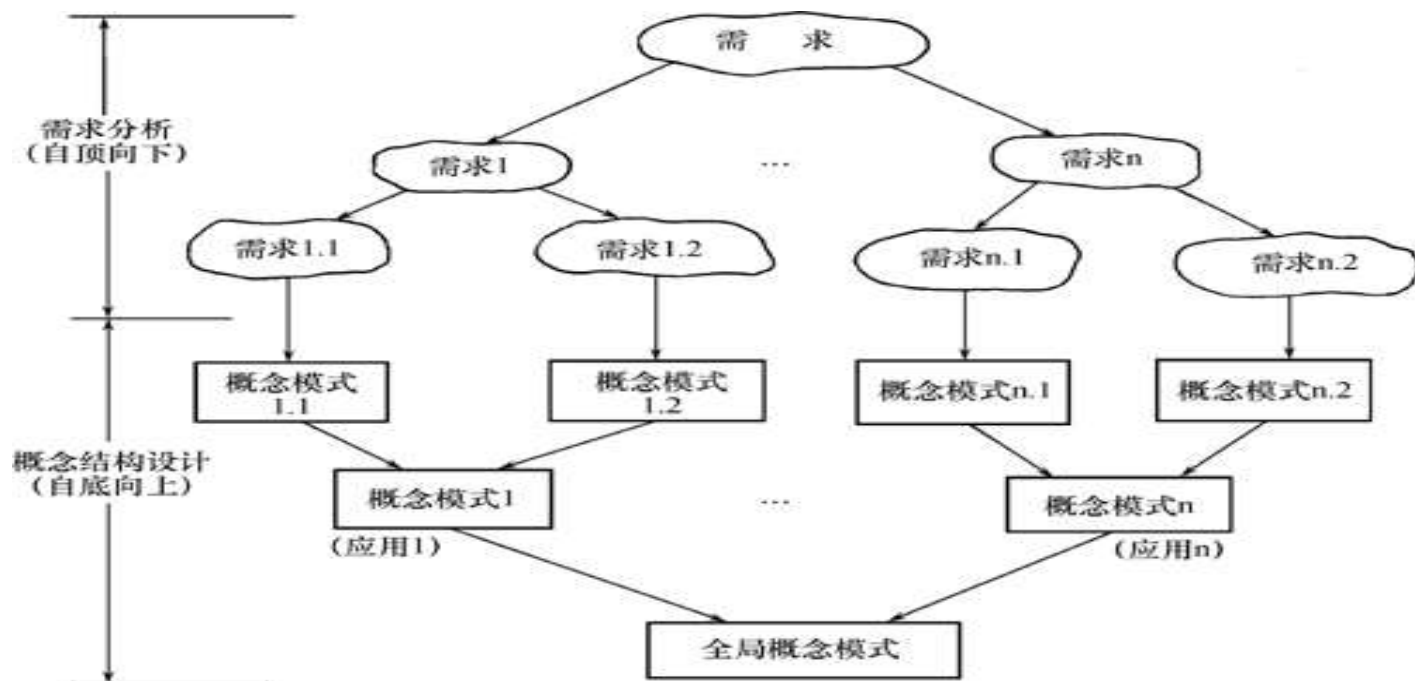
- 将自顶向下和自底向上相结合，用自顶向下策略设计一个全局概念结构的框架，以它为骨架集成由自底向上策略中设计的各局部概念结构。

概念结构设计的方法与步骤（续）



❖ 常用策略

- 自顶向下地进行需求分析
- 自底向上地设计概念结构



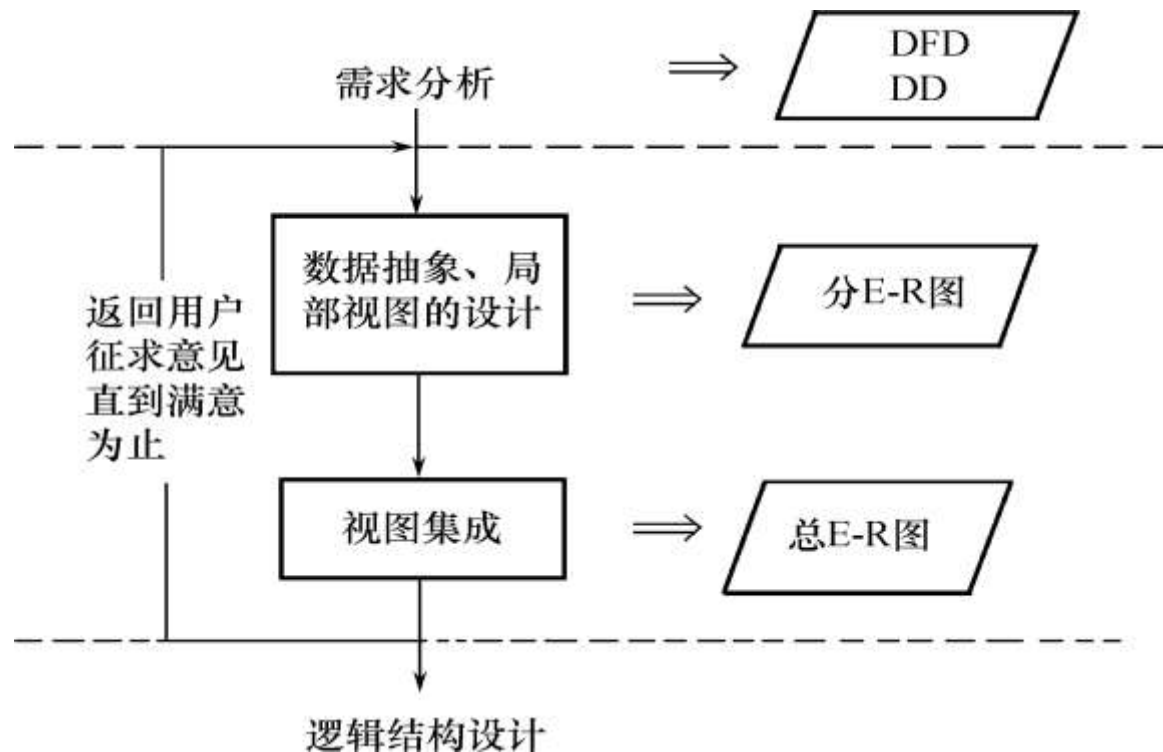
概念结构设计的方法与步骤（续）



❖ 自底向上设计概念结构的步骤

第1步：抽象数据并设计局部视图

第2步：集成局部视图，得到全局概念结构



7.3 概念结构设计



7.3.1 概念结构

7.3.2 概念结构设计的方法与步骤

7.3.3 数据抽象与局部视图设计

7.3.4 视图的集成

7.3.3 数据抽象与局部视图设计



❖ 数据抽象

❖ 局部视图设计

数据抽象



- ❖ 抽象是对实际的人、物、事和概念中抽取所关心的共同特性，忽略非本质的细节，并把这些特性用各种概念精确地加以描述。
 - 概念结构是对现实世界的一种抽象

数据抽象（续）

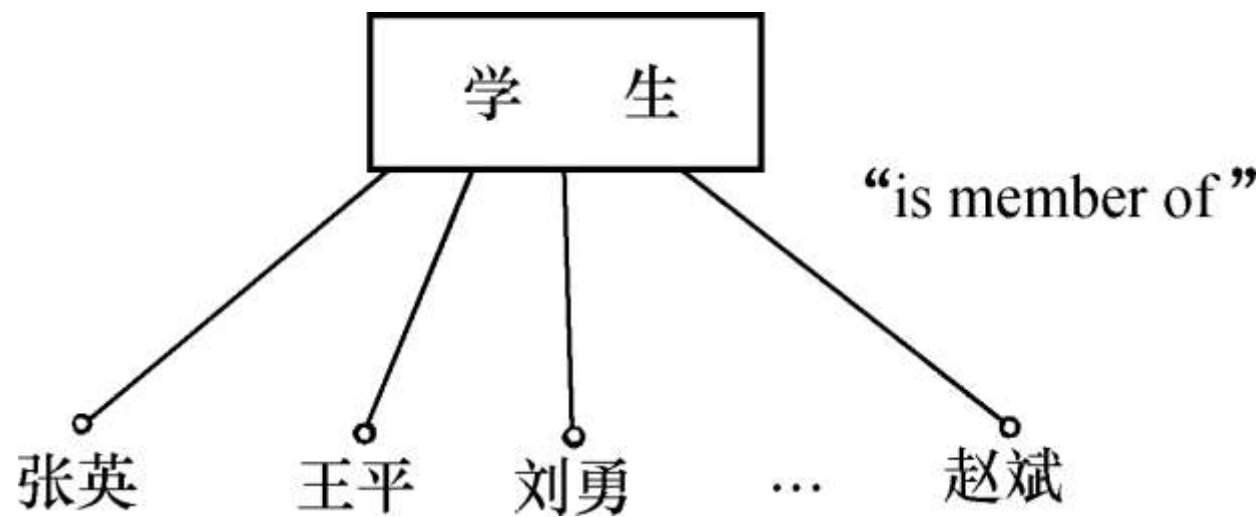


❖ 三种常用抽象

1. 分类（Classification）

- 定义某一类概念作为现实世界中一组对象的类型
- 抽象了对象值和型之间的“is member of”的语义

数据抽象（续）



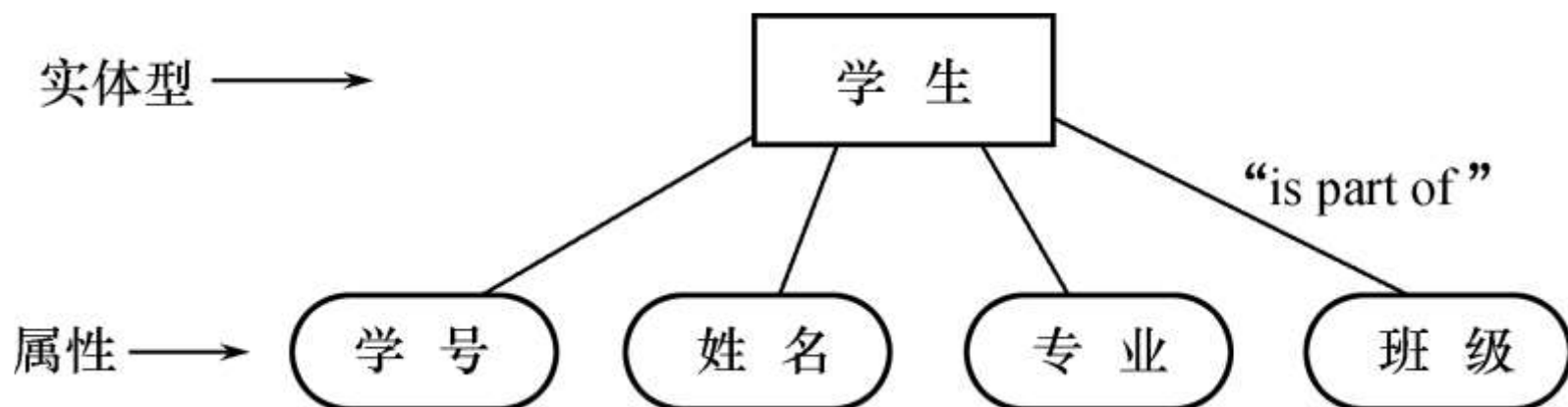
数据抽象（续）



2. 聚集（Aggregation）

- 定义某一类型的组成成分
- 抽象了对象内部类型和成分之间 “is part of”的语义

数据抽象（续）

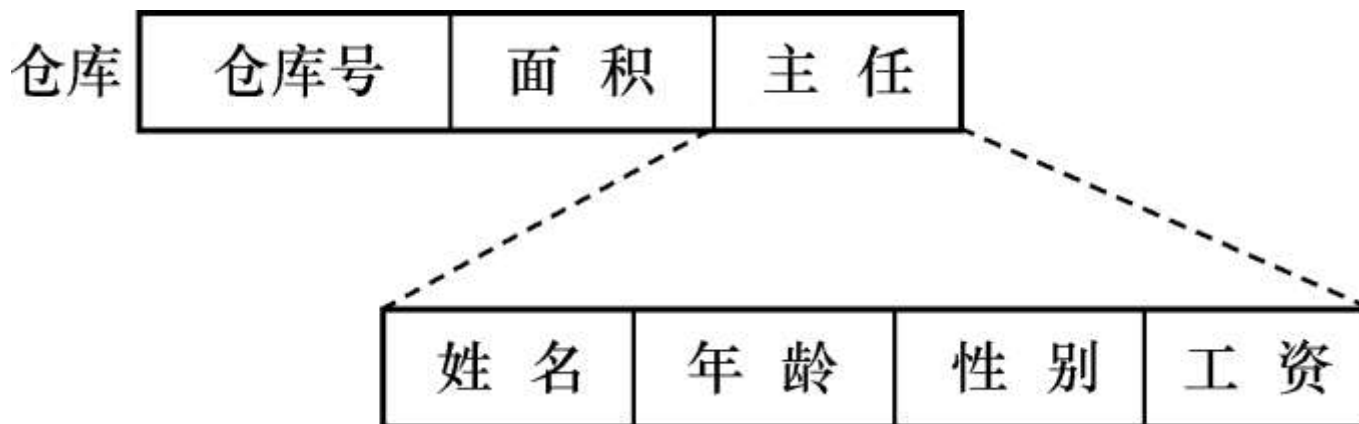


聚集

数据抽象（续）



- 复杂的聚集，某一类型的成分仍是一个聚集



更复杂的聚集

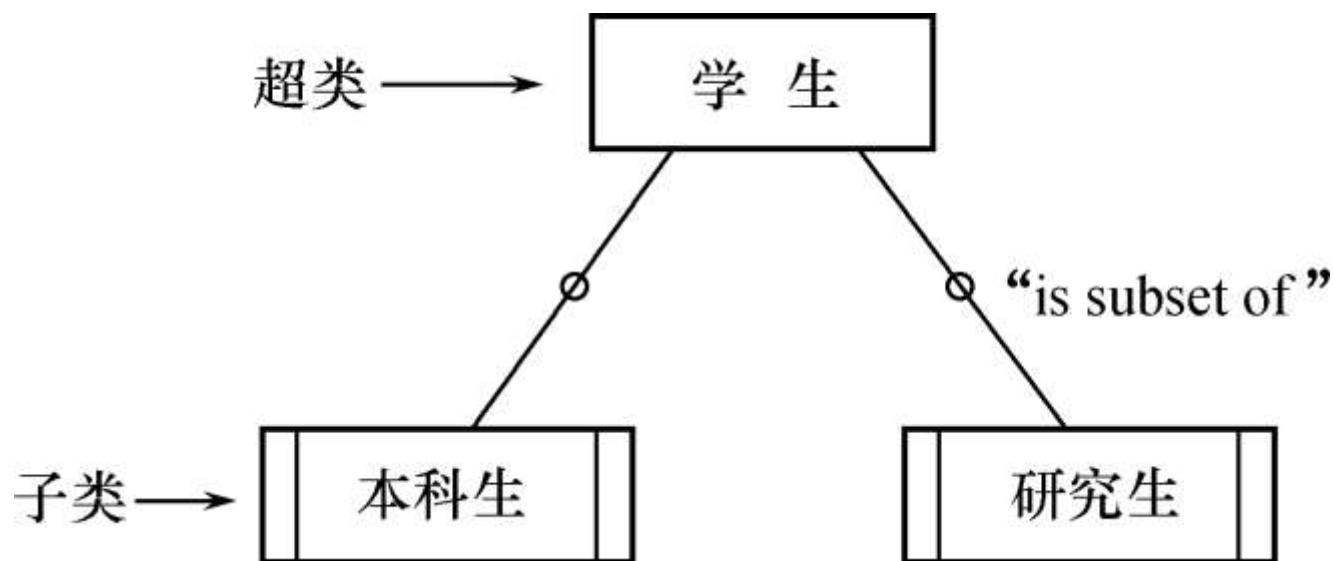
数据抽象（续）



3. 概括（Generalization）

- 定义类型之间的一种子集联系
- 抽象了类型之间的“is subset of”的语义
- 继承性

数据抽象（续）



概括

局部视图设计



设计分E-R图的步骤:

1.选择局部应用

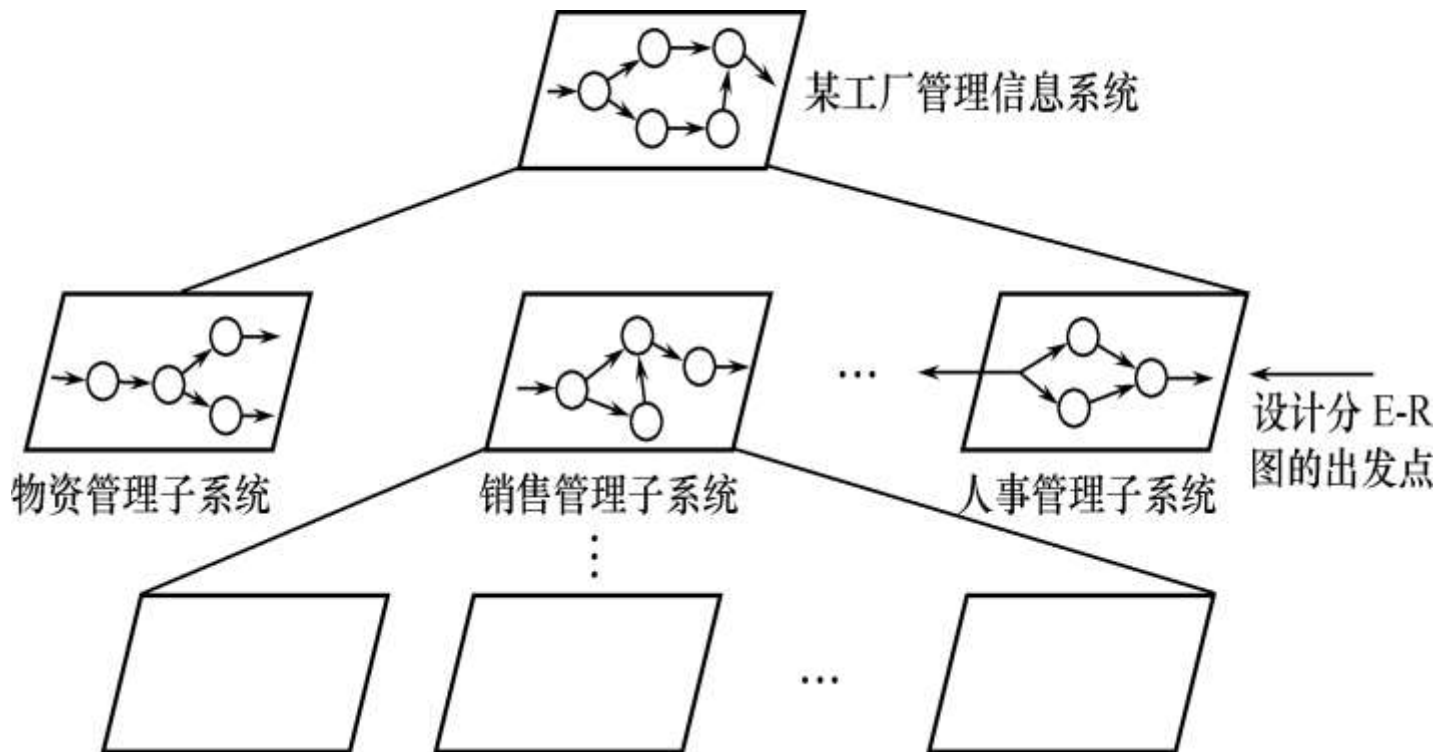
2.逐一设计分E-R图

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）



设计分E-R图的出发点

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
- 确定实体之间的联系及其类型（1:1，1:n，m:n）

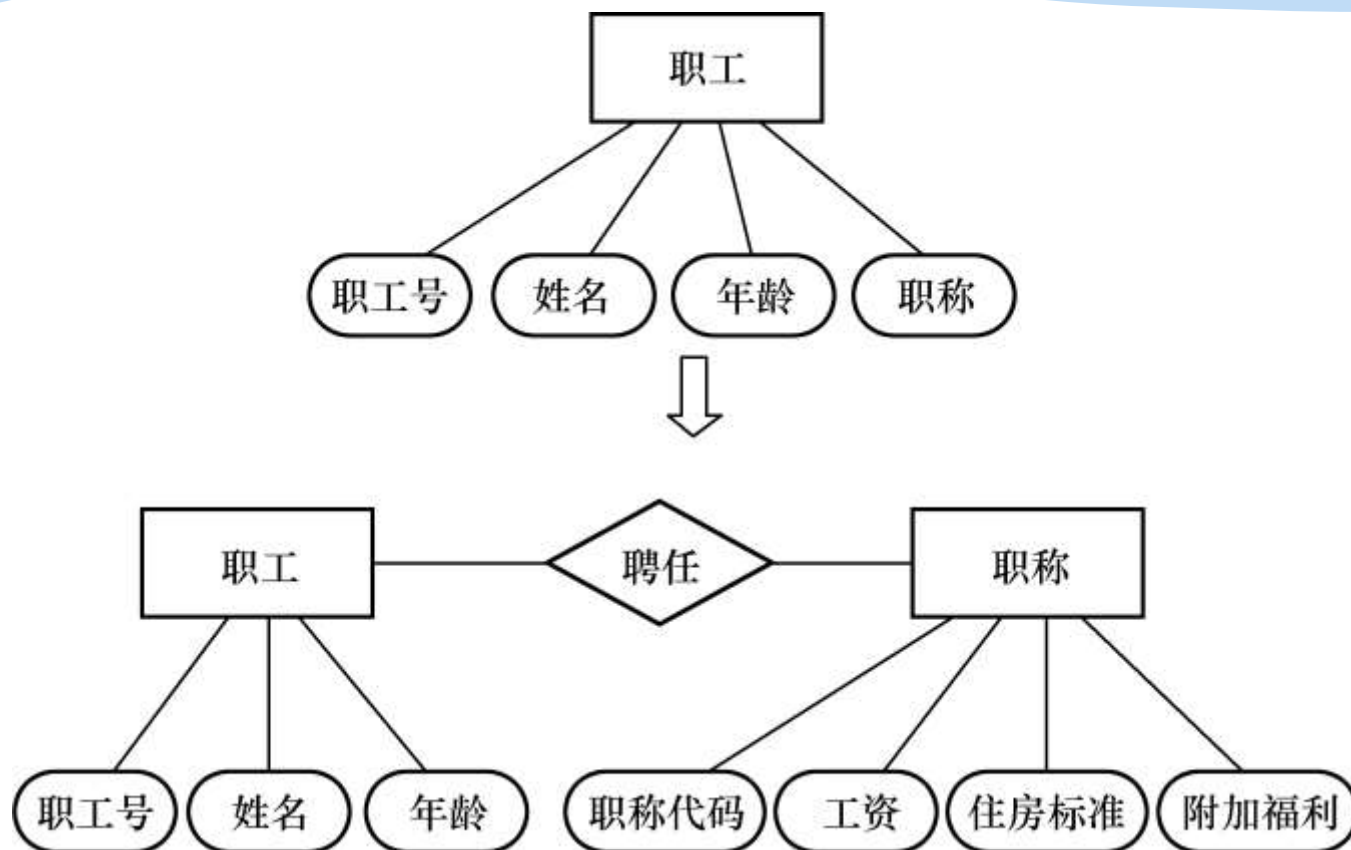
逐一设计分E-R图（续）



❖ 两条准则：

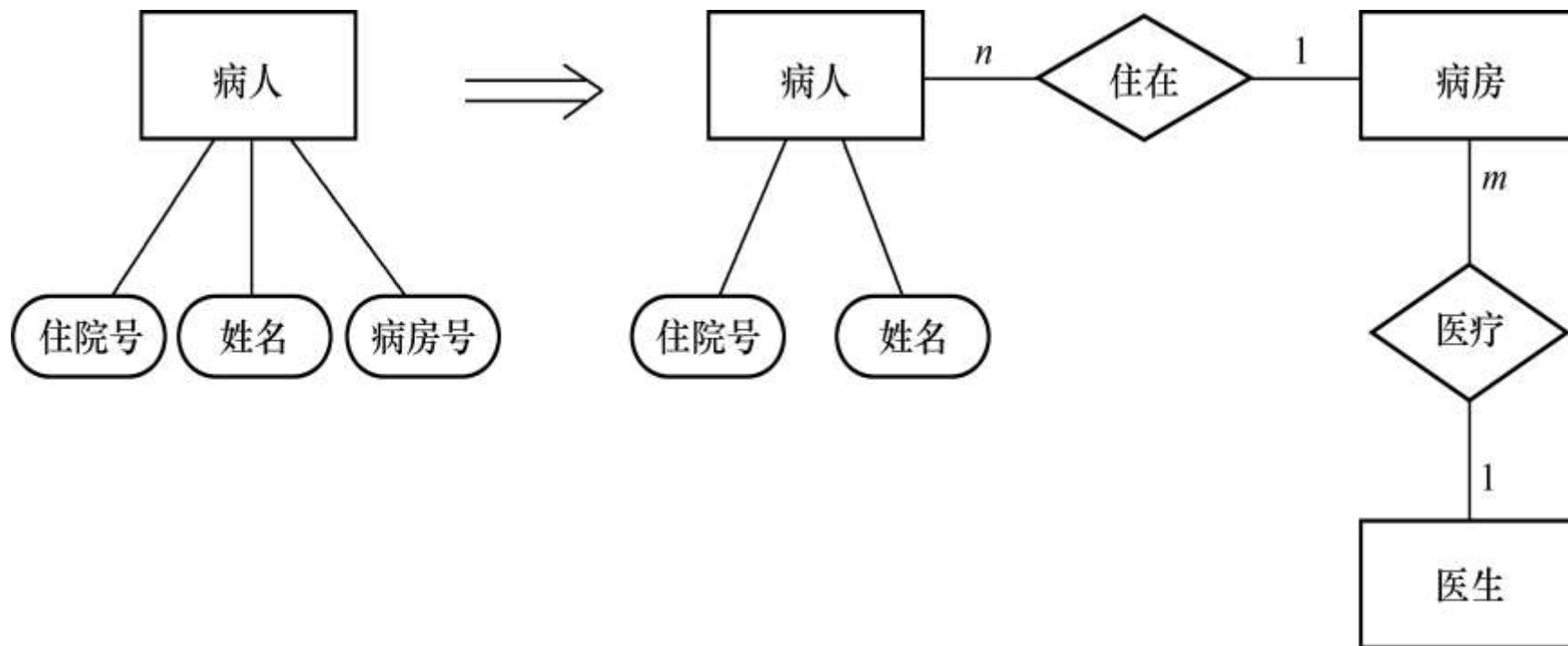
- （1）属性不能再具有需要描述的性质。即属性必须是不可分的数据项，不能再由另一些属性组成
- （2）属性不能与其他实体具有联系。联系只发生在实体之间

逐一设计分E-R图（续）



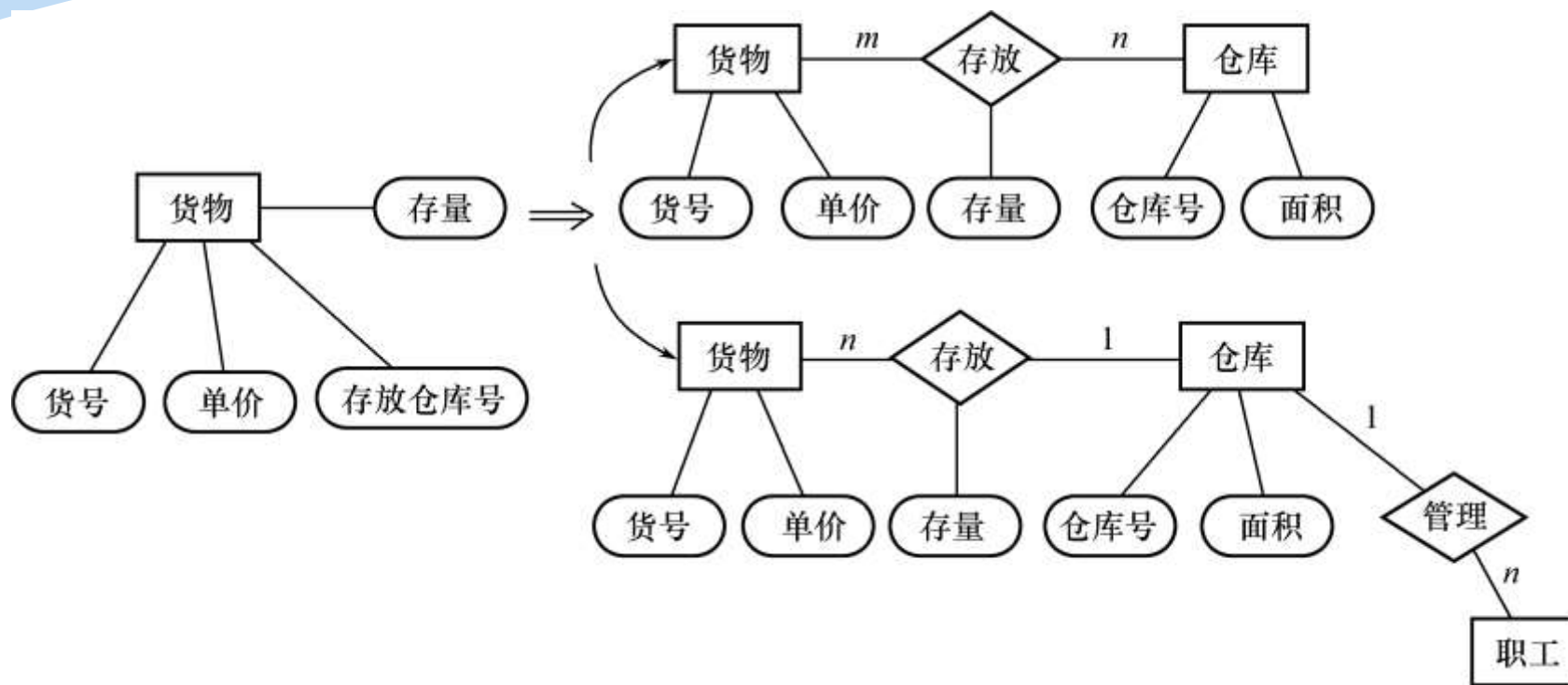
职称作为一个实体

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（续）



❖ 下图是第一层数据流图，虚线部分划出了系统边界

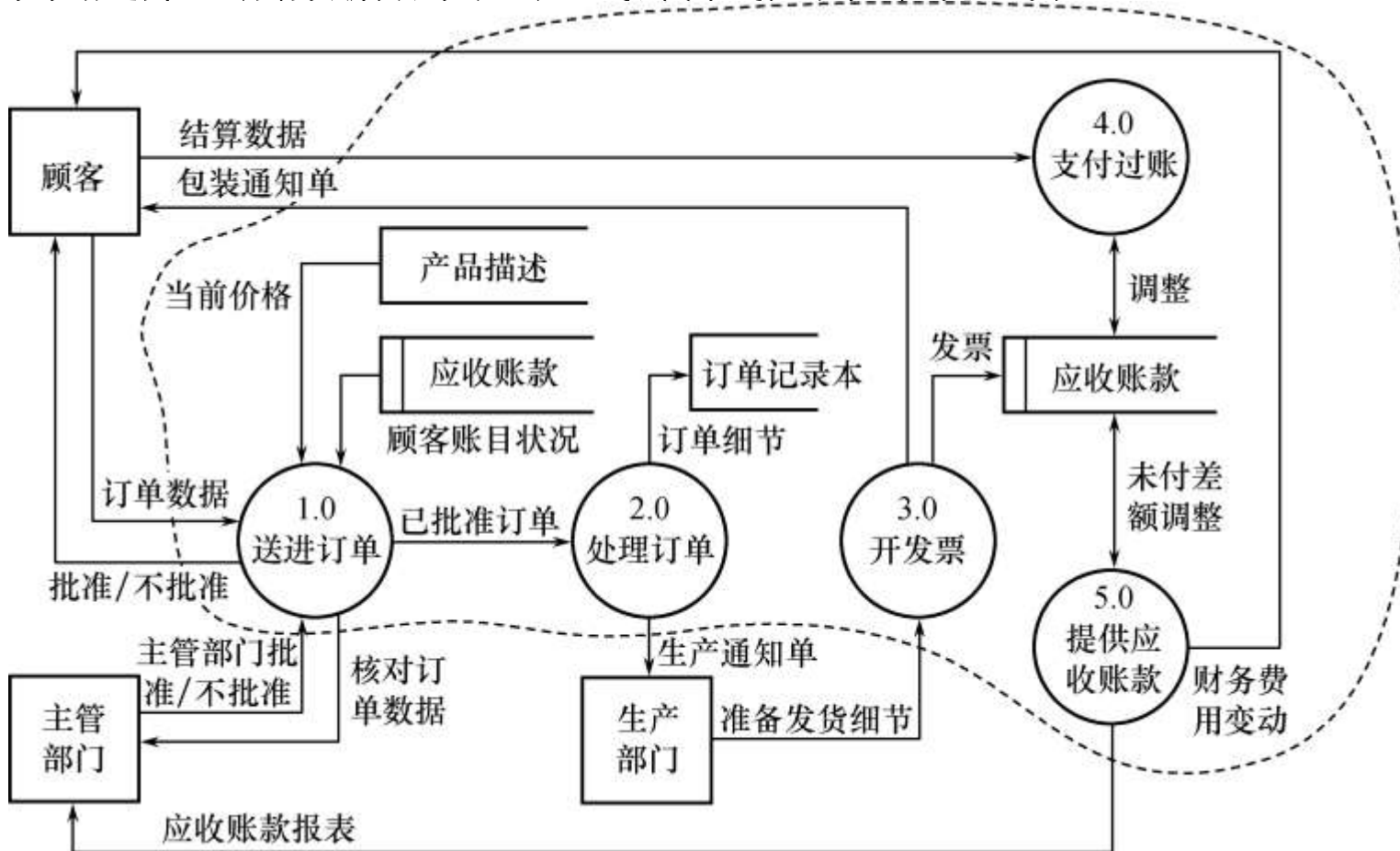


图7.18 销售管理子系统第一层数据流图

逐一设计分E-R图（续）



❖ 上图中把系统功能又分为4个子系统，下面四个图是第二层数据流图

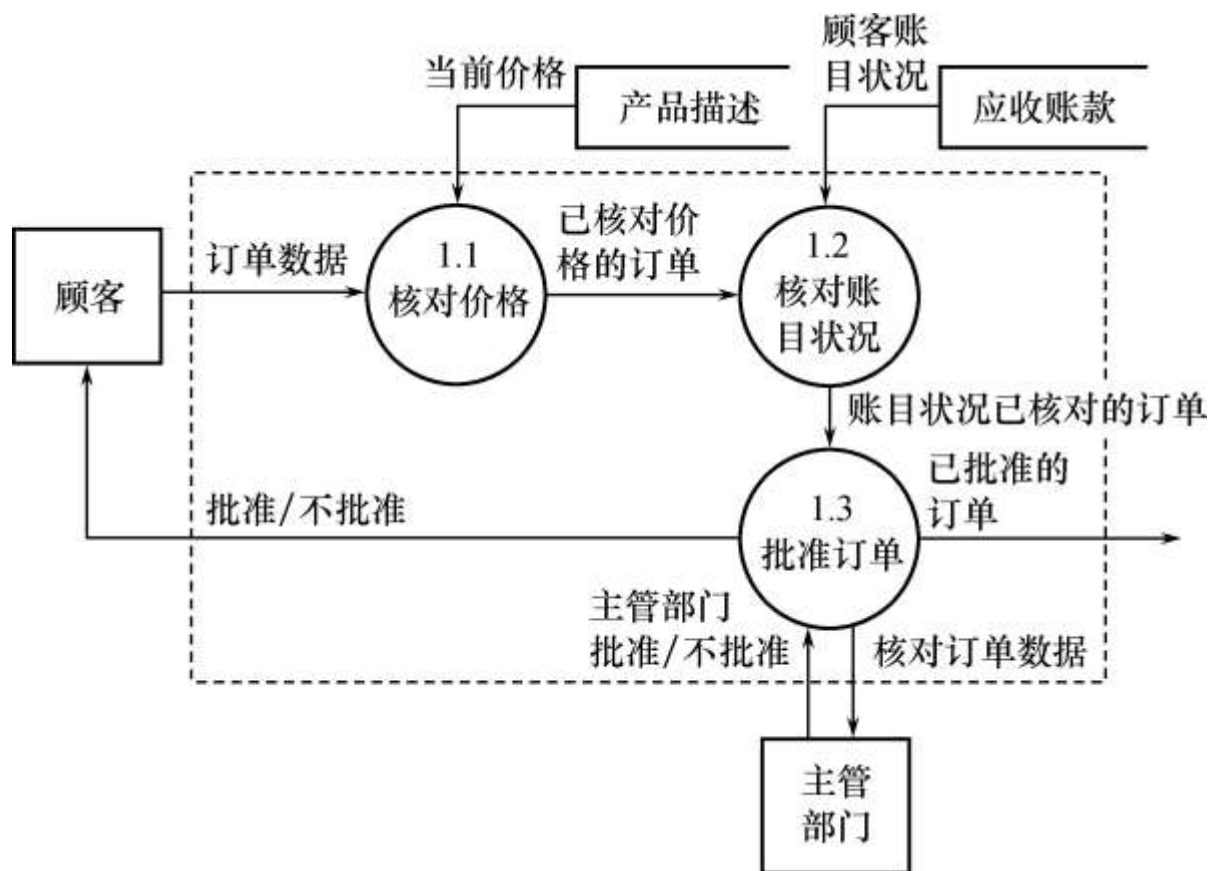


图7.19 接收订单

逐一设计分E-R图（续）

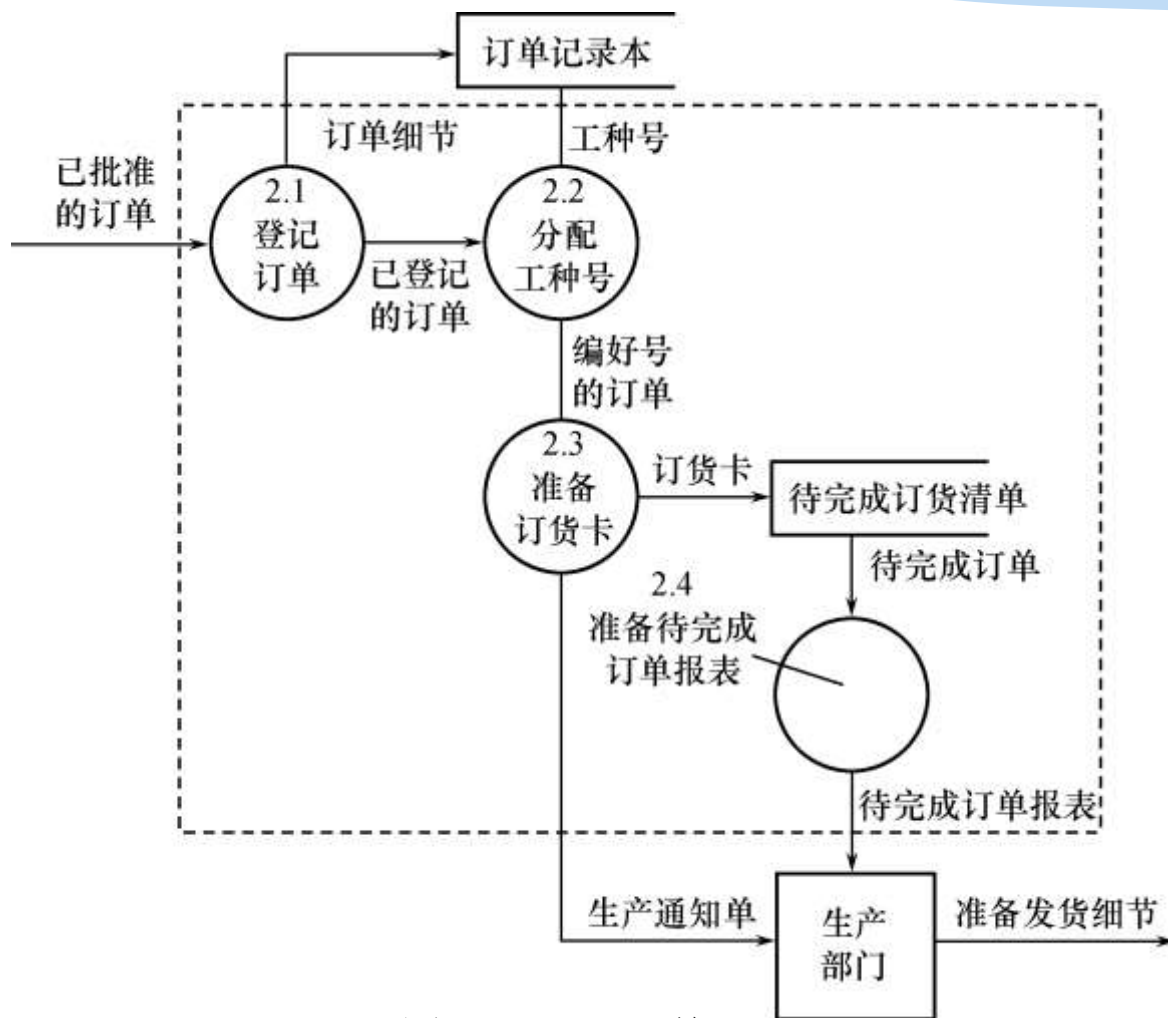


图7.20 处理订单

逐一设计分E-R图（续）

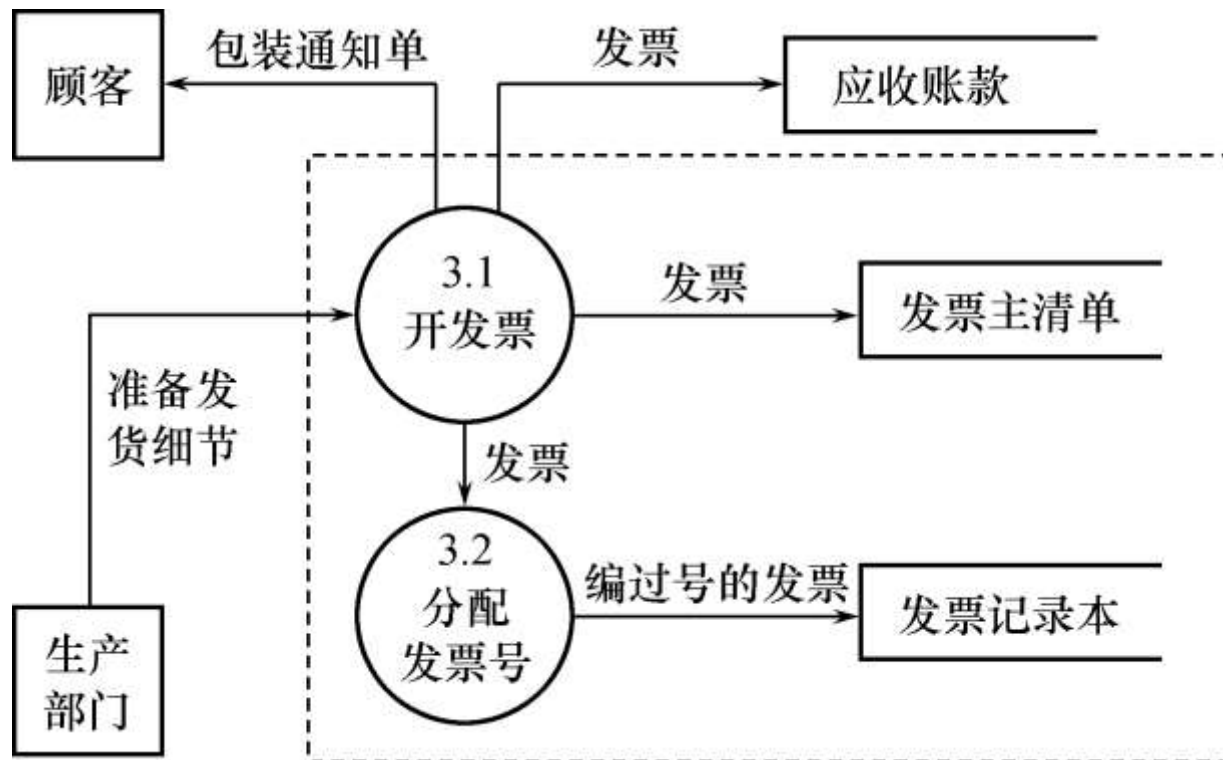


图7.21 开发票

逐一设计分E-R图（续）

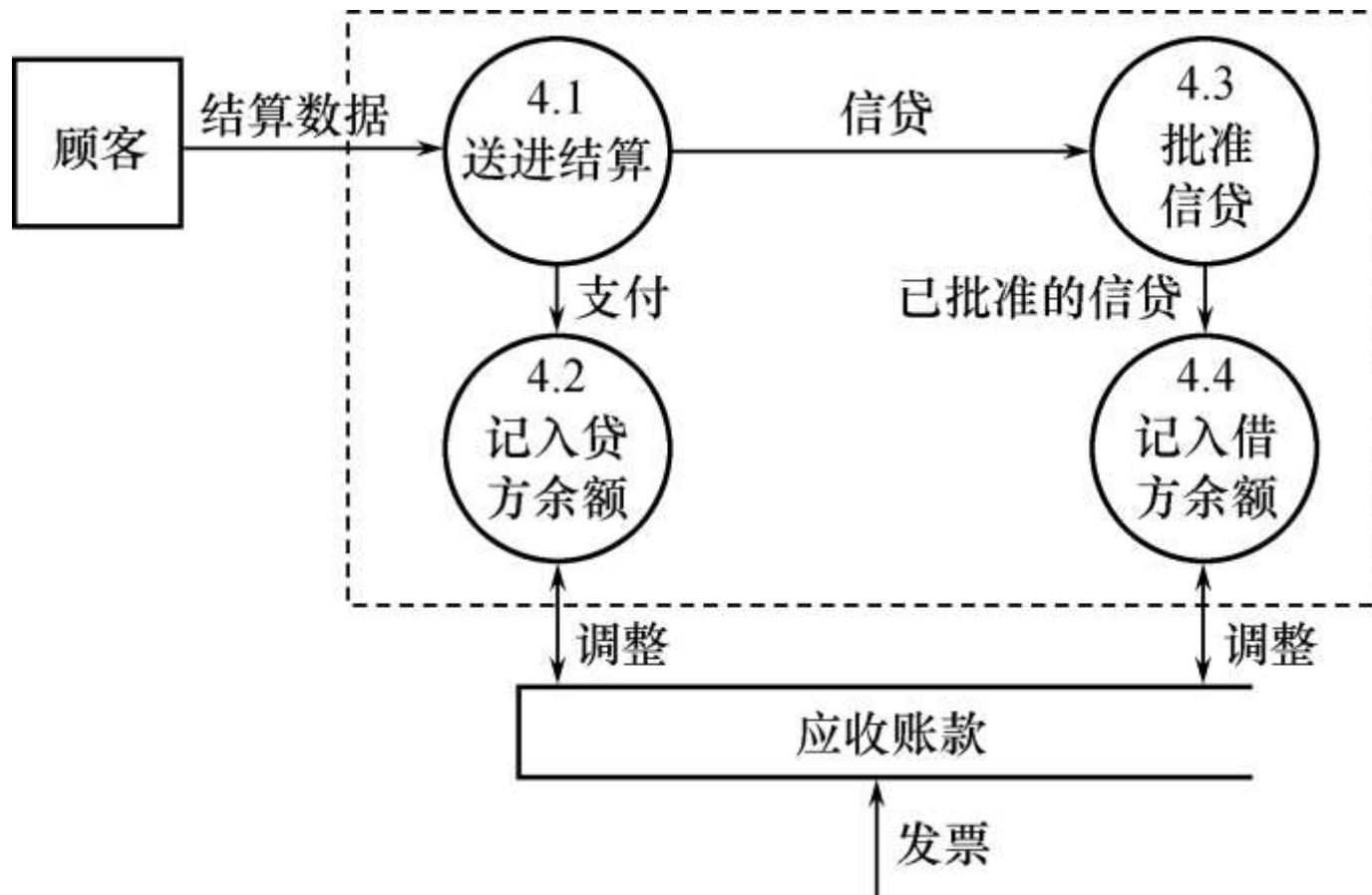
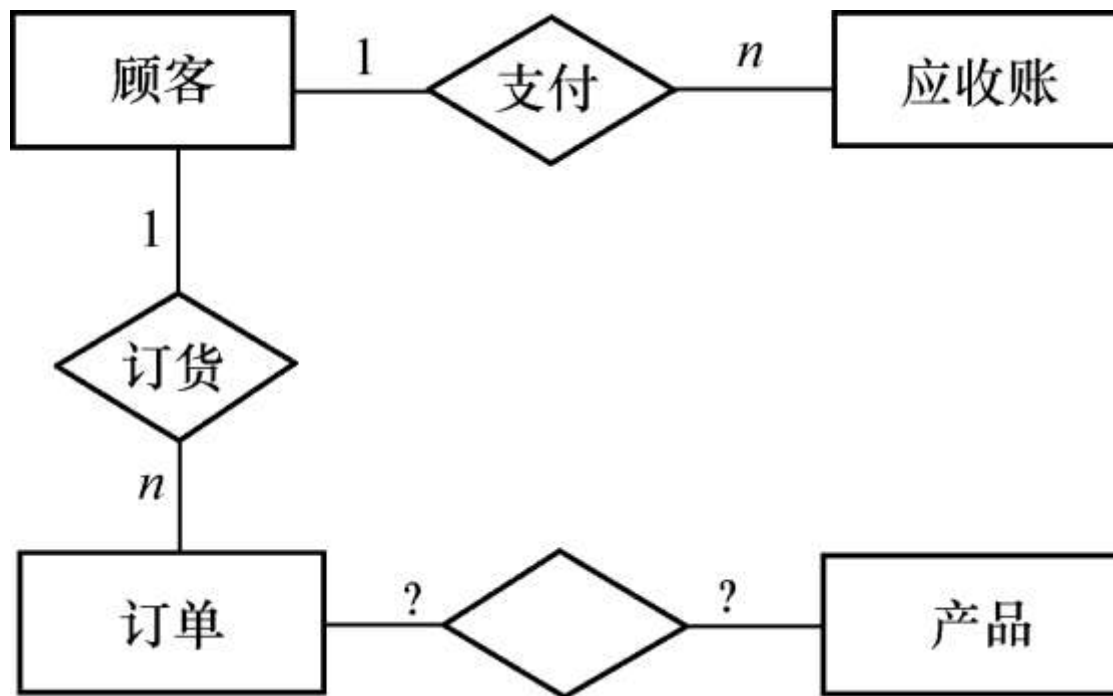


图7.22 支付过账

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）

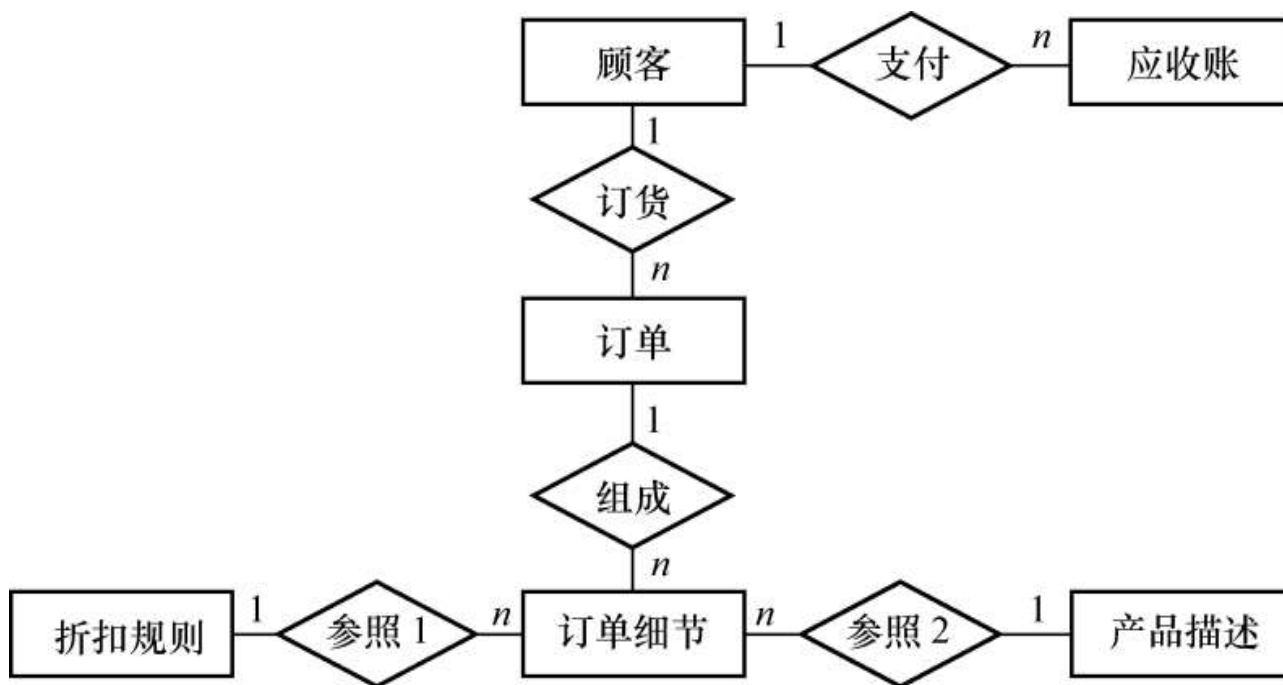


- ❖ 参照第二层数据流图和数据字典，遵循两个准则，进行如下调整：
- (1) 订单与订单细节是1 : n 的联系
 - (2) 原订单和产品的联系实际上是订单细节和产品的联系。
 - (3) 图7.21中“发票主清单”是一个数据存储，不必作为实体加入分E-R图
 - (4) 工厂对大宗订货给予优惠

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，货款限额}
- 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}

7.3 概念结构设计



7.3.1 概念结构

7.3.2 概念结构设计的方法与步骤

7.3.3 数据抽象与局部视图设计

7.3.4 视图的集成

7.3.4 视图的集成



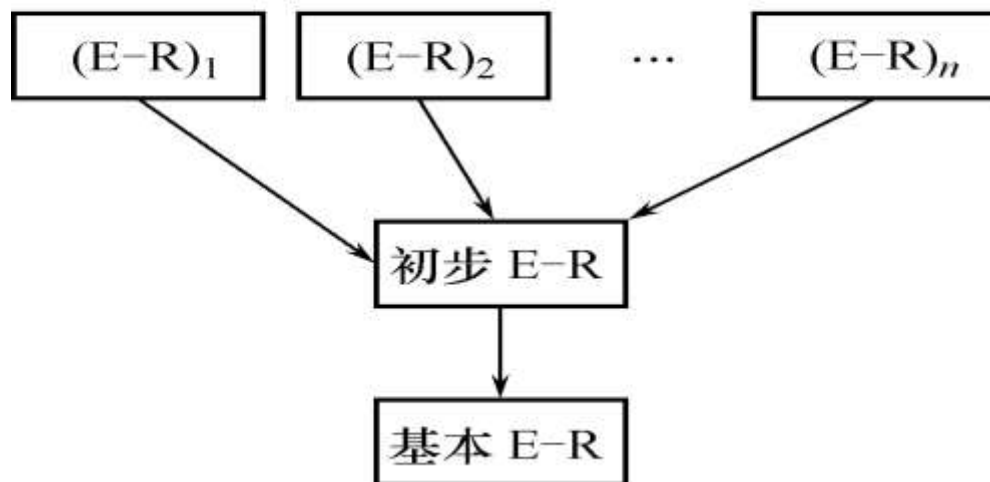
- ❖ 各个局部视图即分**E-R**图建立好后，还需要对它们进行合并，集成为一个整体的数据概念结构即总**E-R**图。

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

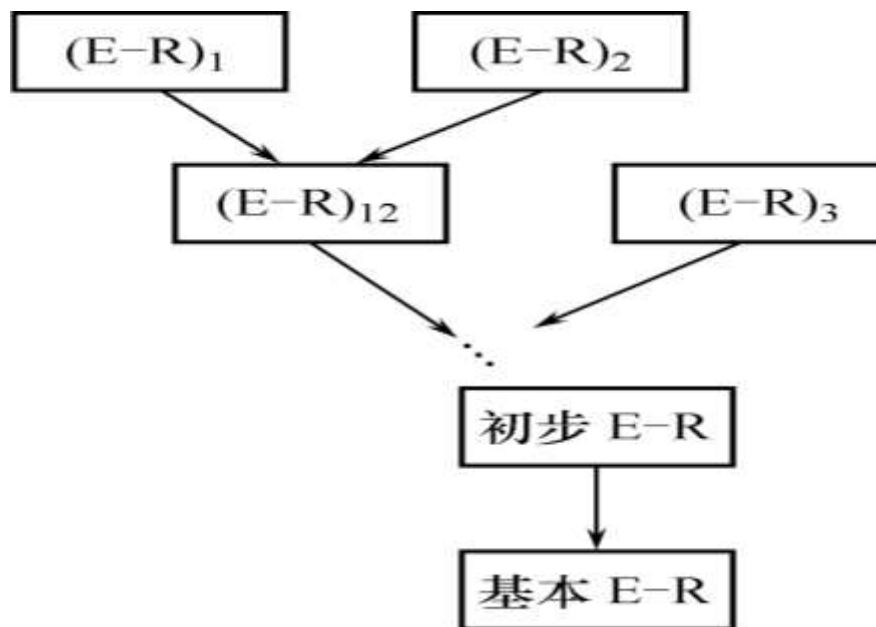


视图的集成（续）



❖ 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）

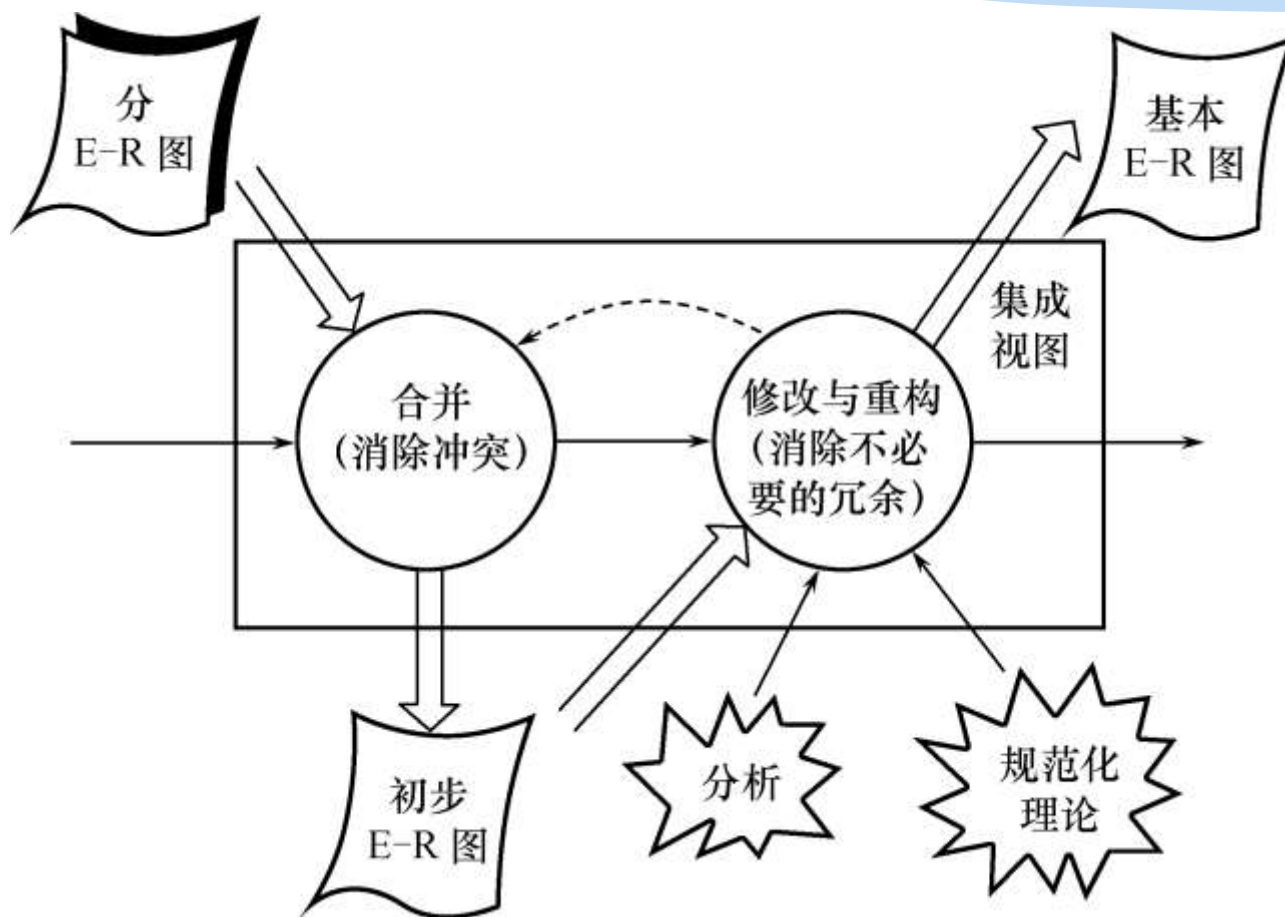


❖ 集成局部E-R图的步骤

1. 合并

2. 修改与重构

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

- 属性冲突
- 命名冲突
- 结构冲突

1. 属性冲突



❖ 两类属性冲突

- 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

- **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

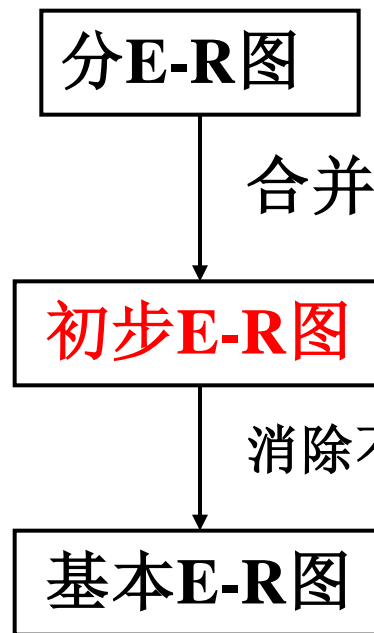
- 同一对象在不同应用中具有不同的抽象
- 同一实体在不同分E-R图所包含的属性个数和属性排列次序不完全相同
- 实体之间的联系在不同局部视图中呈现不同的类型

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

消除不必要的冗余，设计基本E-R图（续）



- ❖ 冗余
- ❖ 消除冗余的方法

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

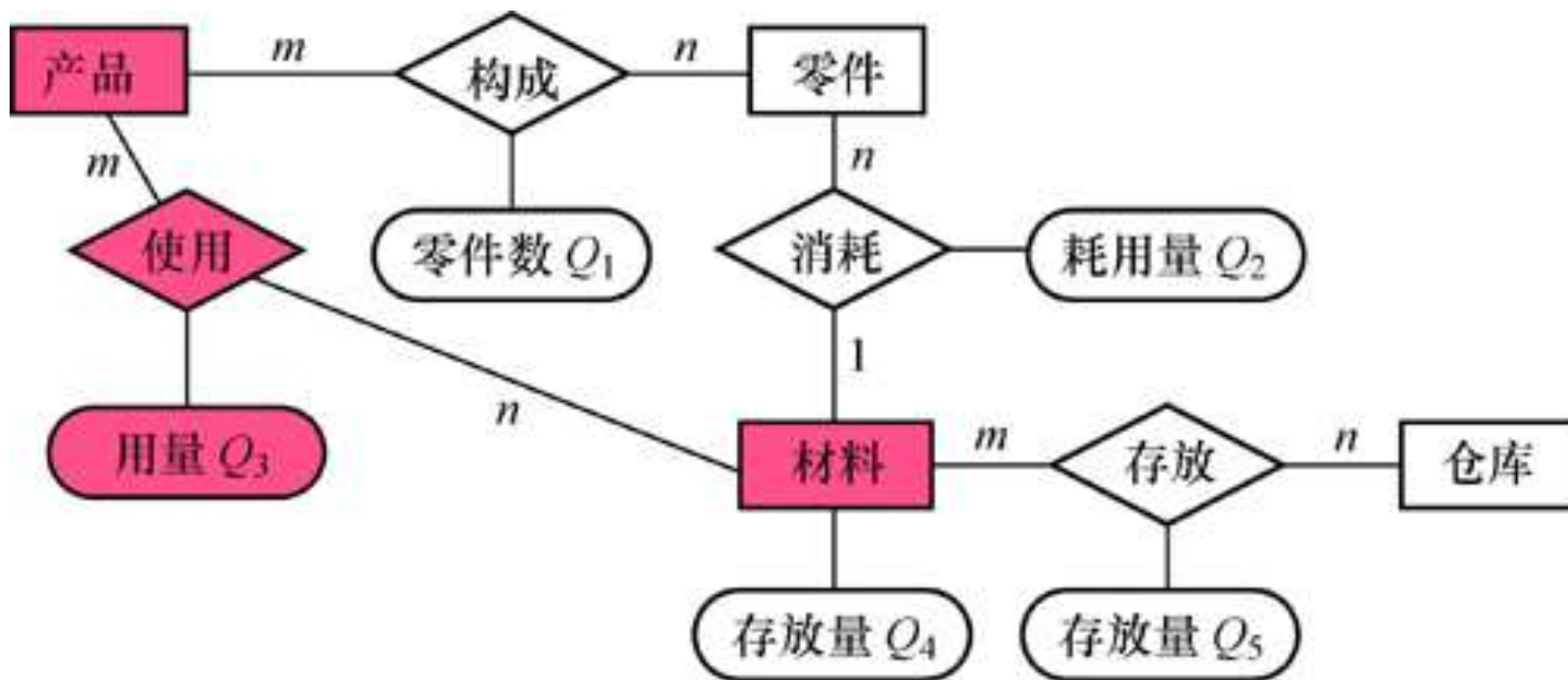
消除冗余的方法



❖ 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

消除冗余的方法（续）



消除冗余

消除冗余的方法（续）



- 效率VS冗余信息
 - 需要根据用户的整体需求来确定
- 若人为地保留了一些冗余数据，则应把数据字典中数据关联的说明作为完整性约束条件
 - $Q4 = \sum Q5$
 - 一旦Q5修改后就应当触发完整性检查，对Q4进行修改

消除冗余的方法（续）



❖ 规范化理论

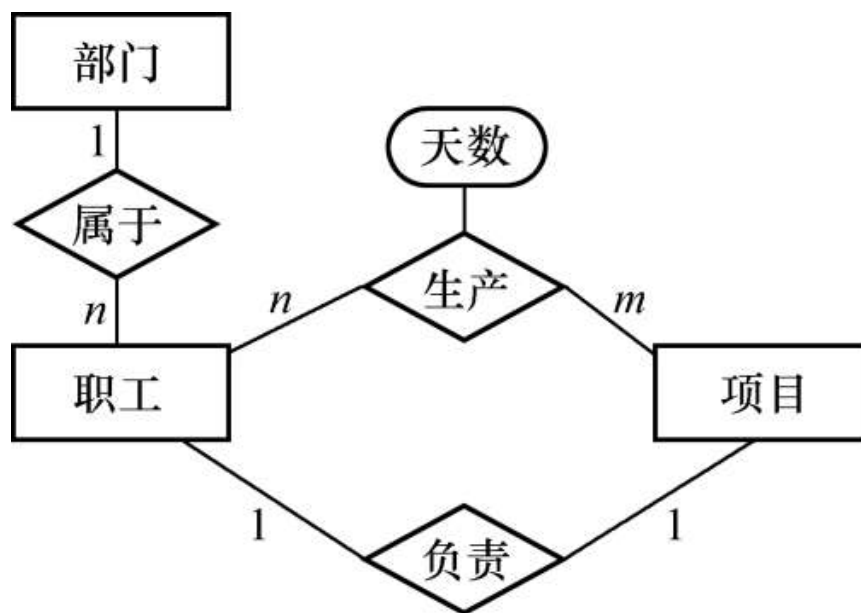
- 函数依赖的概念提供了消除冗余联系的形式化工具

消除冗余的方法（续）



■ 方法

1. 确定分E-R图实体之间的数据依赖，并用实体码之间的函数依赖表示。



劳动人事管理的分E-R图

消除冗余的方法（续）



上图中，

- 部门和职工之间一对多的联系可表示为：
职工号 \rightarrow 部门号
- 职工和产品之间多对多的联系可表示为：
(职工号, 产品号) \rightarrow 工作天数
- 得到函数依赖集 F_L

消除冗余的方法（续）



2. 求 F_L 的最小覆盖 G_L ，差集为 $D = F_L - G_L$ 。

逐一考察 D 中的函数依赖，确定是否是冗余的联系，若是，就把它去掉。

- (1) 冗余的联系一定在 D 中，而 D 中的联系不一定是冗余的；
- (2) 当实体之间存在多种联系时要将实体之间的联系在形式上加以区分。

消除冗余，设计生成基本E-R图实例



[实例] 某工厂管理信息系统的视图集成。

书中图1.14(c)、图7.24、图7.29分别为该厂物资、销售和劳动人事管理的分E-R图

图7.30为该系统的基本E-R图

消除冗余，设计生成基本E-R图实例（续）



该厂物资管理分E-R图

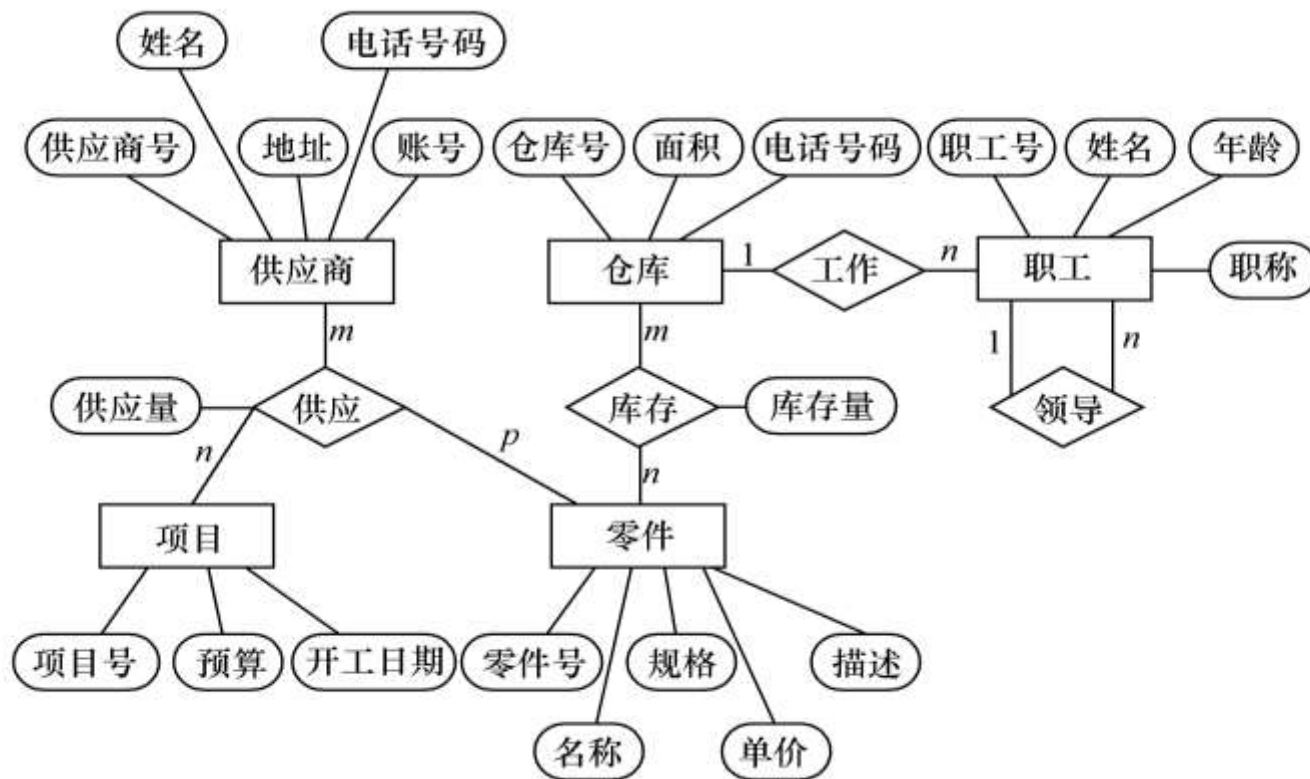


图1.14(c) 工厂物资管理E-R图

消除冗余，设计生成基本E-R图实例（续）



该厂销售管理分E-R图

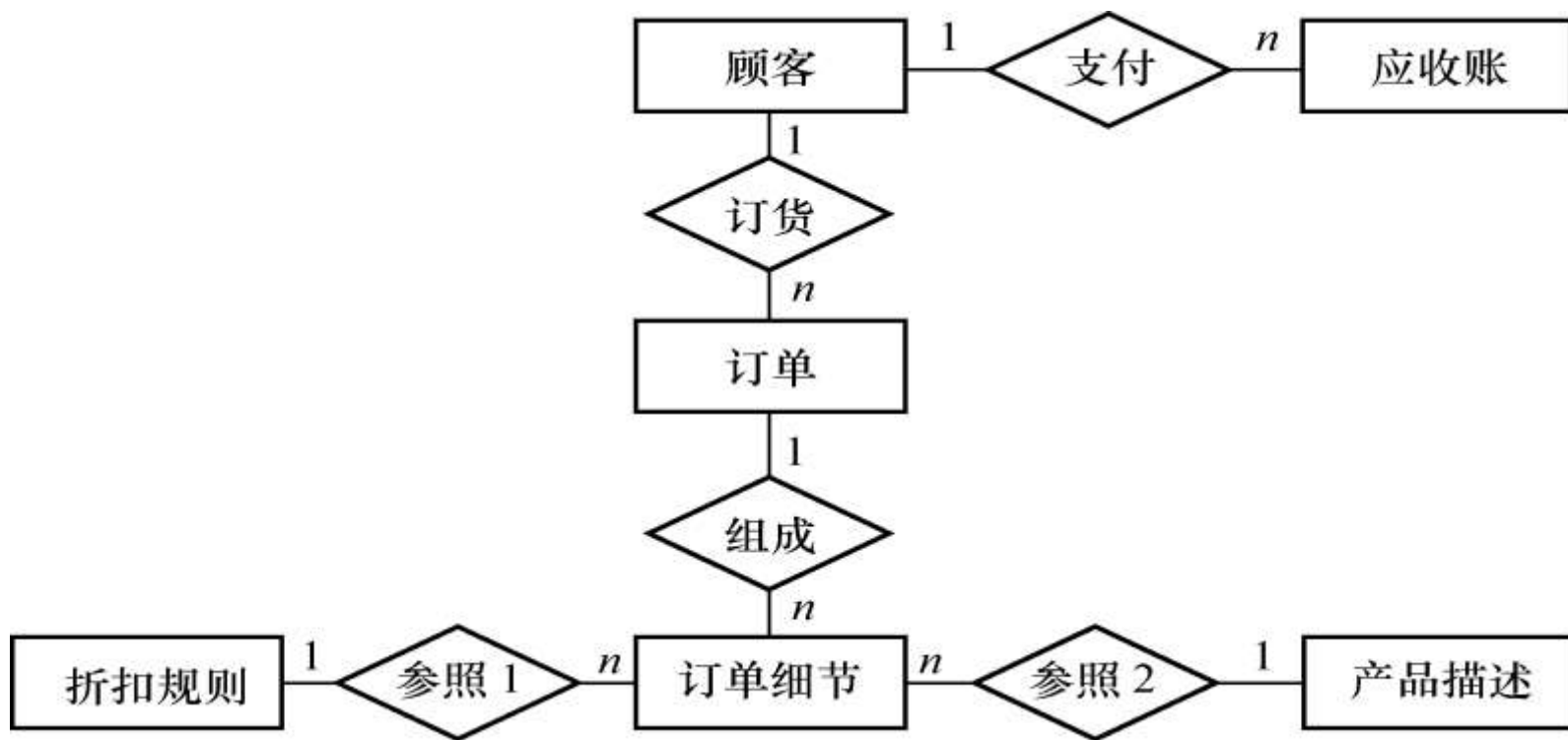


图7.24 销售管理子系统的分E-R图

消除冗余，设计生成基本E-R图实例（续）



该厂劳动人事管理分E-R图

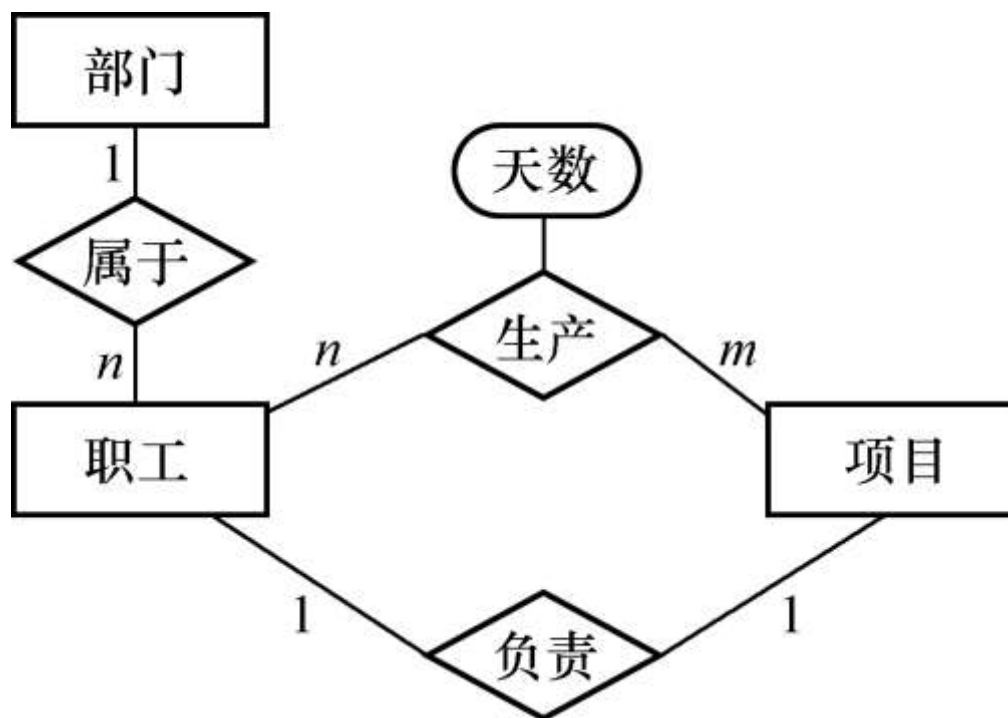
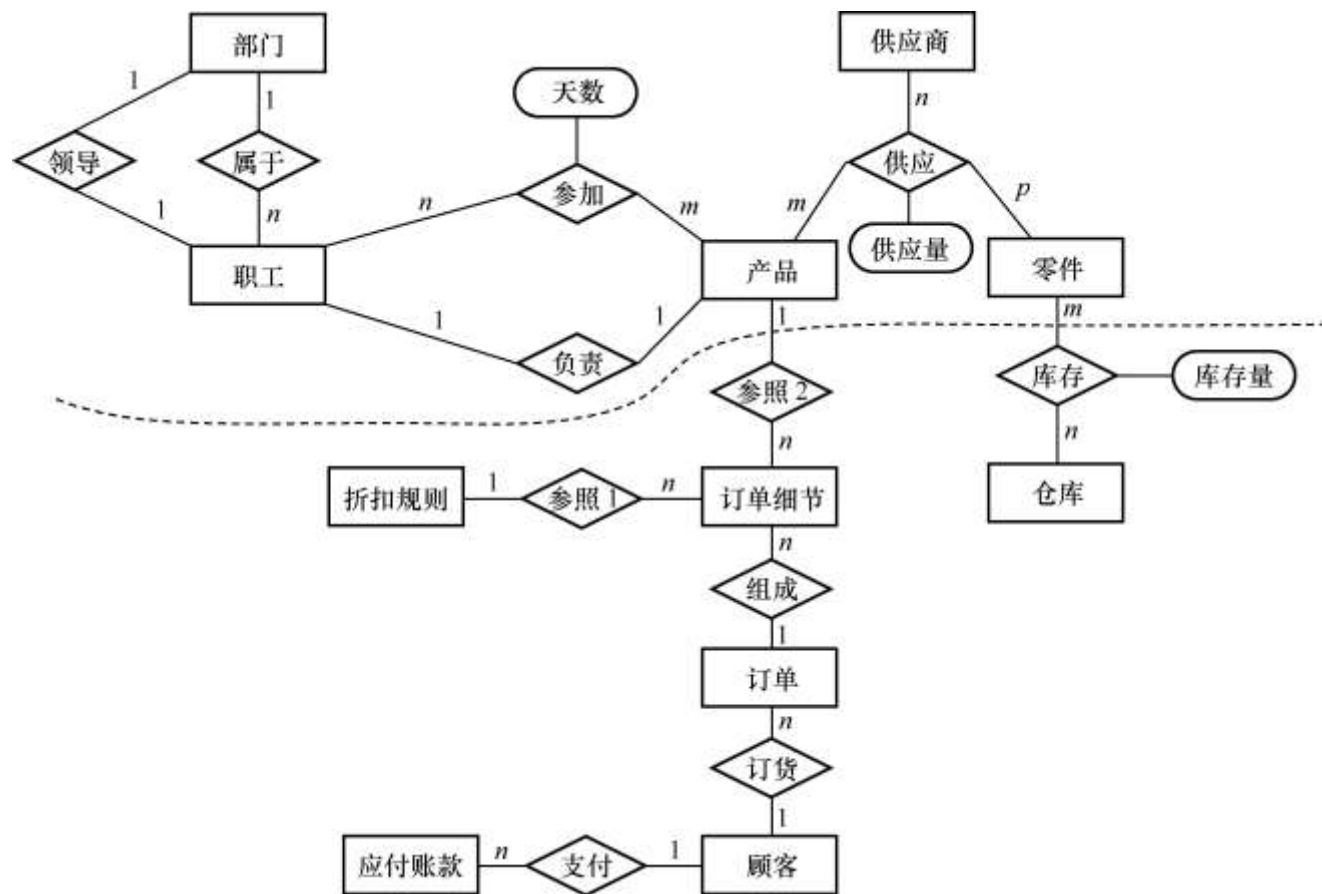


图7.29 劳动人事管理的分E-R图

消除冗余，设计生成基本E-R图实例（续）



系统的基本E-R(图7.30)



消除冗余，设计生成基本E-R图实例（续）



集成过程，解决了以下问题：

- ❖ 异名同义，项目和产品含义相同
- ❖ 库存管理中职工与仓库的工作关系已包含在劳动人事管理的部门与职工之间的联系之中，所以可以取消
- ❖ 职工之间领导与被领导关系可由部门与职工（经理）之间的领导关系、部门与职工之间的从属关系两者导出，所以也可以取消

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，对该整体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

- 抽象数据并设计局部视图
- 集成局部视图，得到全局概念结构
- 验证整体概念结构

概念结构设计小结



❖ 数据抽象

- 分类
- 聚集
- 概括

概念结构设计小结



❖ 设计局部视图

- 1. 选择局部应用
- 2. 逐一设计分E-R图
 - 标定局部应用中的实体、属性、码，实体间的联系
 - 用E-R图描述出来

概念结构设计小结



❖ 集成局部视图

■ 1. 合并分E-R图，生成初步E-R图

➤ 消除冲突

- 属性冲突
- 命名冲突
- 结构冲突

■ 2. 修改与重构

➤ 消除不必要的冗余，设计生成基本E-R图

- 分析方法
- 规范化理论

下课了。。。



攀
登



休息一会儿。。。

*Darling,
We were meant to be!*

