



# 知识点81：数据库设计概述

# 数据库设计的任务



- **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，使之能够有效地存储数据，满足各种用户的信息要求和处理要求。**
- **也就是把现实世界中的数据，根据各种应用要求，加以合理地组织，满足硬件和操作系统的特性，利用已有的DBMS来建立能够实现系统目标的数据库和应用系统。**

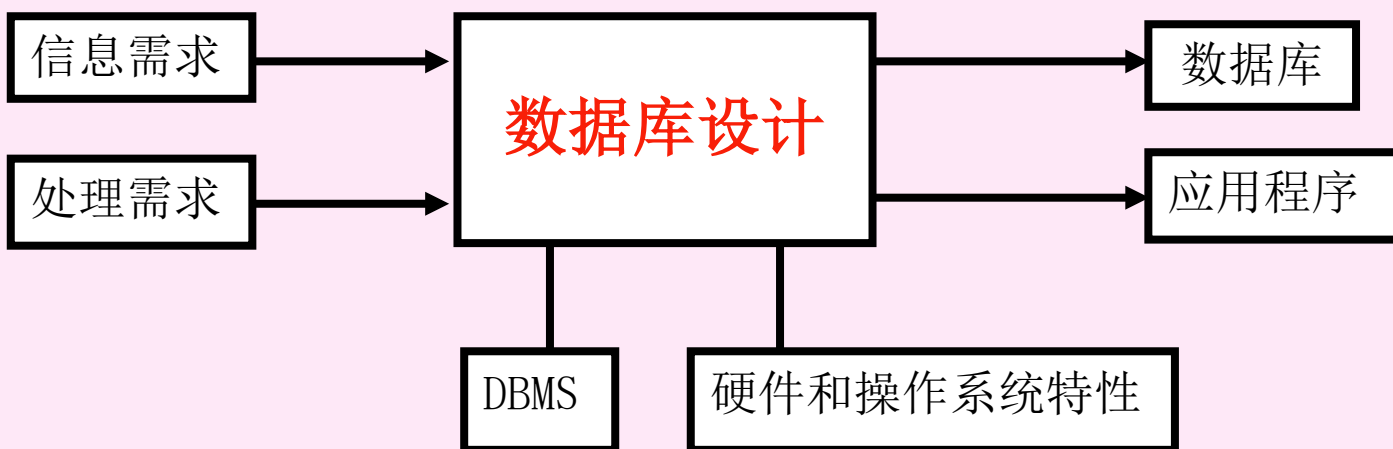


图1 数据库设计的任务

# 数据库设计的特点



## ■ 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

### ■ 结构（数据）设计和行为（处理）设计相结合

### ■ 将数据库结构设计和数据处理设计密切结合

# 数据库设计的特点



## ■ 数据库设计应该与应用系统设计相结合

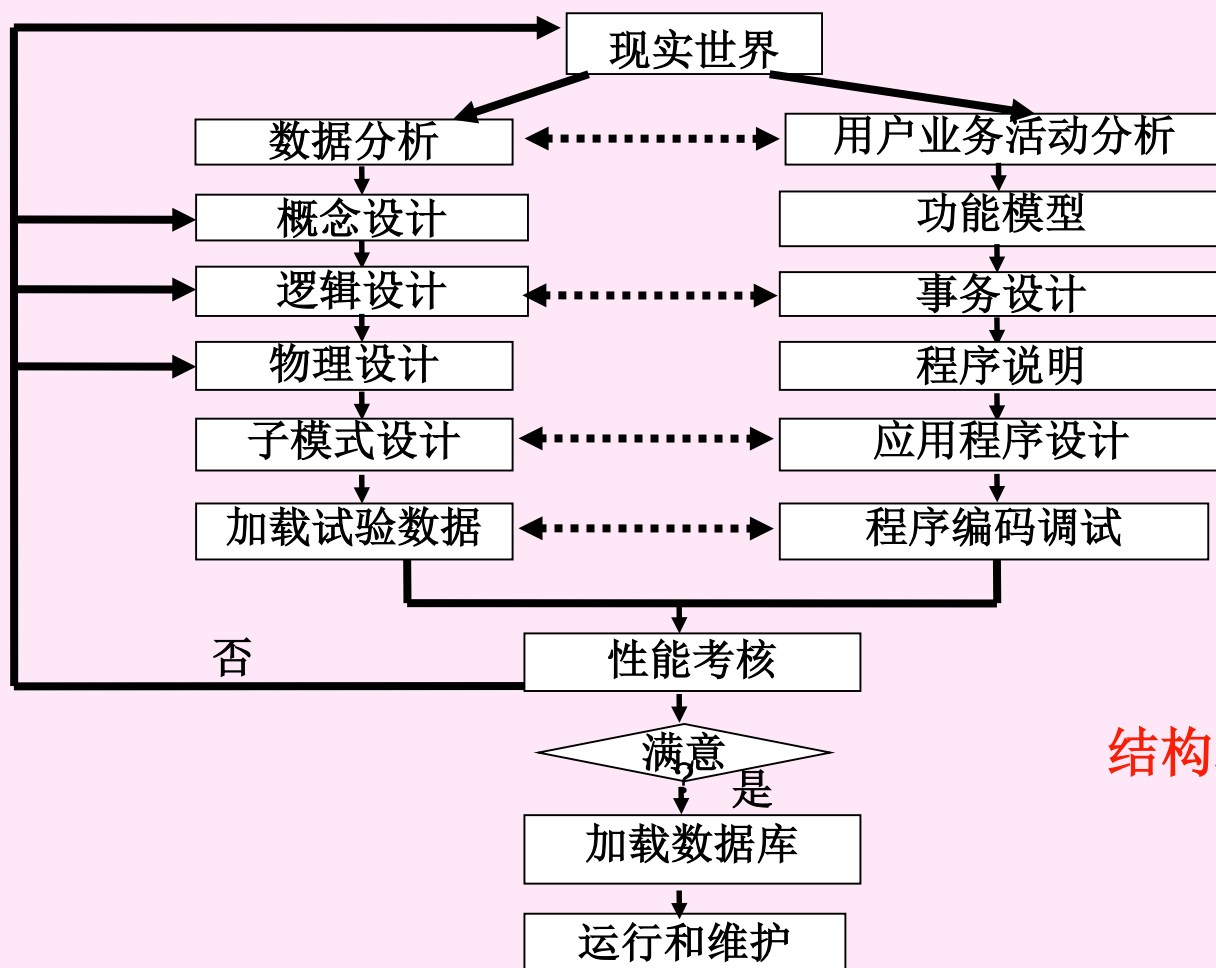
### ■ 结构（数据）设计

- 数据库的结构设计是指根据给定的应用环境，进行数据库的**模式**或**子模式**等的设计
- 它包括数据库的概念结构设计、逻辑结构设计和物理结构设计。

### ■ 行为（处理）设计

- 指用户对数据库的操作，这些要通过应用程序来实现，所以数据库的行为设计就是应用程序的设计。

# 数据库设计的特点（续）



结构和行为分离的设计

# 数据库设计方法



- 数据库设计方法可分为：**手工试凑法**、**规范设计法**、**计算机辅助设计法**。
  - 手工试凑法是最早使用的数据库设计方法。
  - 这种方法依赖于设计者的经验和技巧，缺乏科学理论和工程原则的支持，设计的质量很难保证，常常是数据库运行一段时间后又发现各种问题，这样再重新进行修改，增加了系统维护的代价。
  - 因此这种方法越来越不适应发展的需要。

# 数据库设计方法（续）



- 为了改变这种情况，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。

下面简单介绍几种常用的规范设计方法。



# 数据库设计方法（续）



## 1. 基于E-R模型的数据库设计方法

- 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概念模型
- 然后再将此概念模型转换成基于某一特定的DBMS的数据模型。



## 2. 基于3NF的数据库设计方法

- 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。



- 除了以上两种方法外，规范化设计方法还有基于视图的数据库设计方法、实体分析法、属性分析法和基于抽象语义的设计方法等，这里不再详细介绍。
- 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。

# 数据库设计方法简述（续）



- **计算机辅助设计法是指在数据库设计的某些过程中模拟某一规范化设计的方法，并以人的知识或经验为主导，通过人机交互方式实现设计中的某些部分。**
- **目前许多计算机辅助软件工程（Computer Aided Software Engineering, CASE）工具可以自动或辅助设计人员完成数据库设计过程中的很多任务。比如SYSBASE公司的PowerDesigner。**

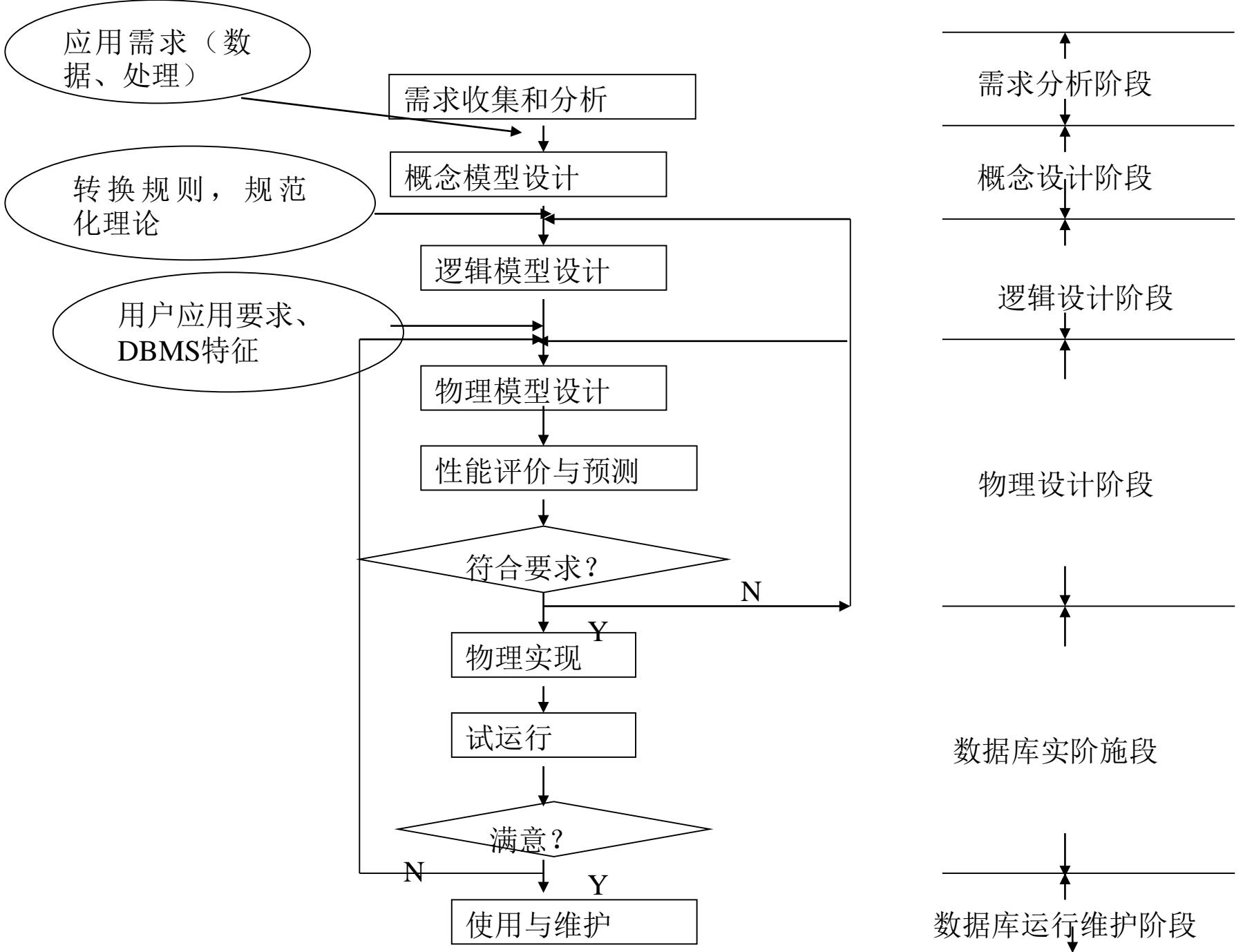


## 知识点82：数据库设计基本步骤

# 数据库设计的基本步骤



- 按照规范设计法，考虑数据库及其应用系统开发全过程，可将数据库设计分为以下六个阶段：
  - 需求分析阶段
  - 概念结构设计阶段
  - 逻辑结构设计阶段
  - 物理结构设计阶段
  - 数据库实施阶段
  - 数据库运行和维护阶段
- 需求分析和概念设计独立于任何数据库管理系统
- 逻辑设计和物理设计与选用的DBMS密切相关





## ■ 一、数据库设计的准备工作：选定参加设计的人

### ■ 1.系统分析人员、数据库设计人员

- 自始至终参与数据库设计

### ■ 2. 用户和数据库管理员

- 主要参加需求分析和数据库的运行维护

### ■ 3.应用开发人员（程序员和操作员）

- 在系统实施阶段参与进来，负责编制程序和准备软硬件环境



# 数据库设计的基本步骤（续）



## ■ 需求分析阶段

分析用户数据需求与处理需求。

## ■ 概念设计阶段

对用户需求进行综合、归纳与抽象，形成独立于具体DBMS的概念模型(E-R图)

# 数据库设计的基本步骤（续）



## ■ 逻辑设计阶段

- 首先将E-R图转换成具体的DBMS支持的数据模型，如关系模型，形成数据库逻辑模式
- 然后根据用户处理的要求、安全性的考虑，在基本表的基础上再建立必要的视图(View)，形成数据的外模式



## ■ 物理结构设计阶段

**为逻辑数据模型选取一个最适合应用环境的物理结构（包括存储结构和存取方法）**

# 数据库设计的基本步骤（续）



## ■ 数据库实施阶段

**设计人员运用DBMS提供的数据库语言及宿主语言，根据逻辑设计和物理设计的结果**

- **建立数据库**
- **编制与调试应用程序**
- **组织数据入库**
- **并进行试运行**

# 数据库设计的基本步骤（续）



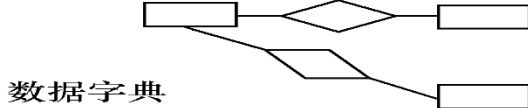
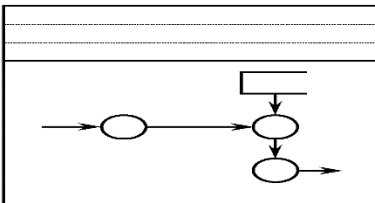
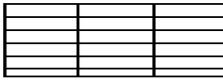
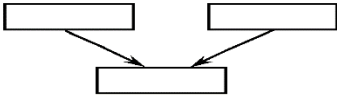
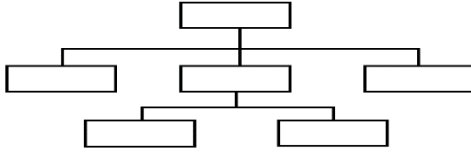
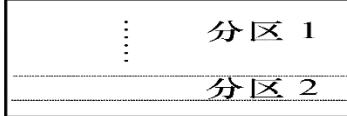
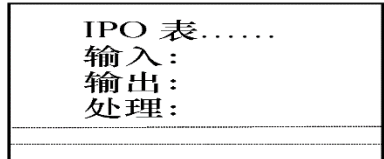
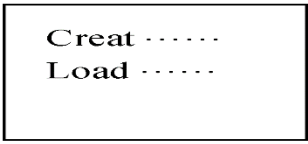
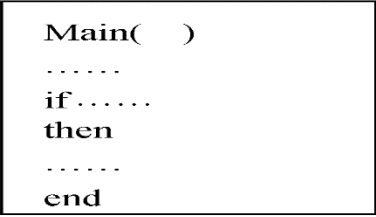
## ■ 数据库运行与维护阶段

**数据库应用系统经过试运行后即可投入正式运行，在数据库系统运行过程中不断进行评价、调整、修改等维护工作。**



- 整个设计步骤既是数据库设计的过程，也包括数据库应用系统的设计过程
- 在设计过程中将**数据设计**与**处理设计**紧密结合起来，将这两个方面的需求分析、抽象、设计、实现在各个阶段同时进行，相互参照，相互补充。

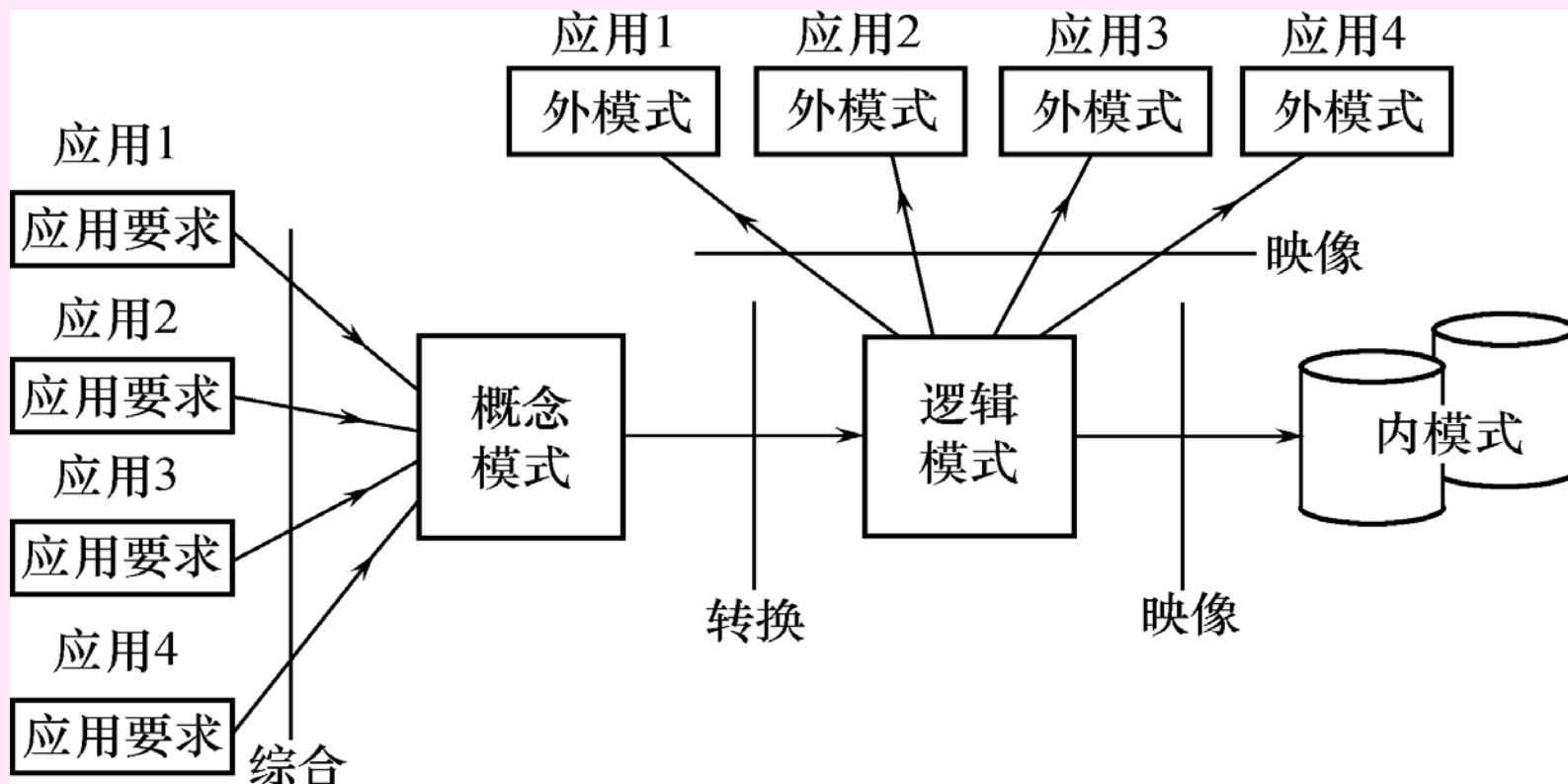
# 数据库设计各个阶段的设计描述

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	数据流图和判定表（判定树）、数据字典中处理过程的描述
概念结构设计	<p>概念模型（E-R图）</p>  <p>数据字典</p>	<p>系统说明书包括：</p> <p>① 新系统要求、方案和概图</p> <p>② 反映新系统信息流的数据流图</p> 
逻辑结构设计	<p>某种数据模型</p> <p>关系</p>  <p>非关系</p> 	<p>系统结构图</p> <p>（模块结构）</p> 
物理设计	<p>存储安排方法选择</p> <p>存取路径建立</p> 	<p>模块设计</p> <p>IPO 表</p> 
数据库实施阶段	<p>编写模式</p> <p>装入数据</p> <p>数据库试运行</p> 	<p>程序编码、编译联结、测试</p> 
数据库运行和维护	性能监测、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）

# 数据库设计过程中的各级模式



## 数据库设计不同阶段形成的数据库各级模式



数据库的各级模式





## 知识点83：需求分析

# 需求分析（续）



- **需求分析就是分析用户的需要与要求**
  - **需求分析是设计数据库的起点**
  - **需求分析的结果是否准确地反映了用户的实际要求，将直接影响到后面各个阶段的设计，并影响到设计结果是否合理和实用**

# 需求分析的任务



- ❖ 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



❖ 调查的重点是“数据”和“处理”，获得用户对数据库要求

- 信息要求
- 处理要求
- 安全性与完整性要求

# 需求分析的难点



## ❖ 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

# 需求分析的小结



- 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



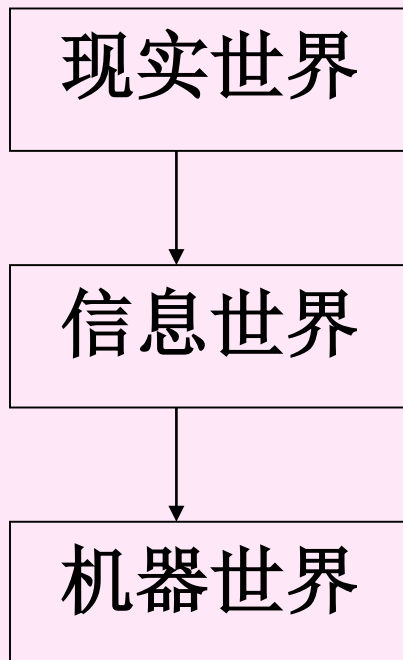
## 知识点84：概念结构设计

## ❖ 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

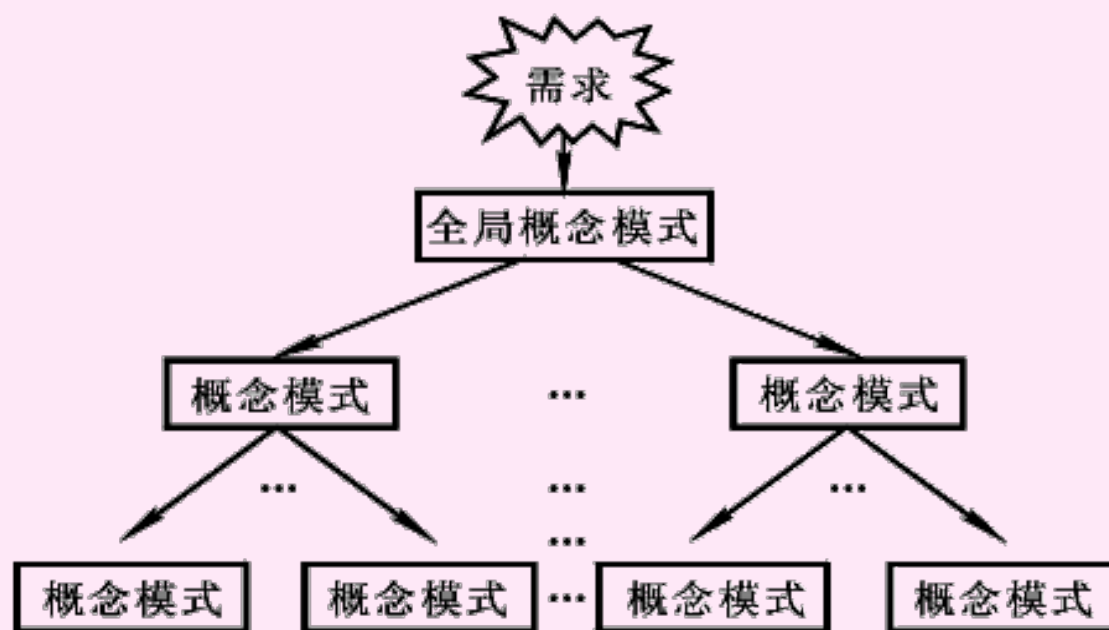
# 概念结构设计的方法与步骤



## ❖ 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



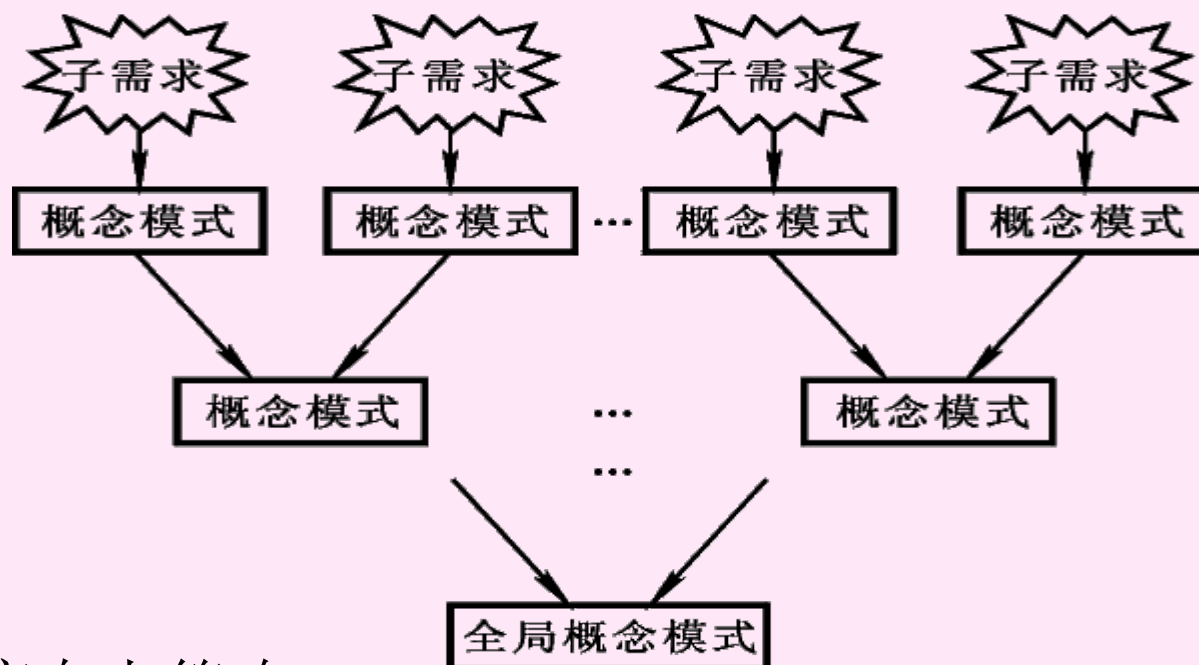
自顶向下策略

# 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



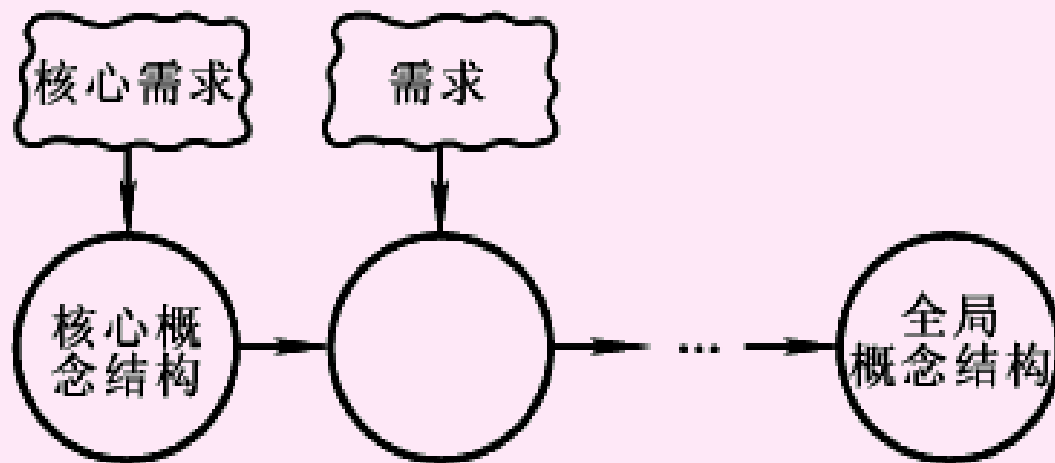
自底向上策略

# 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）



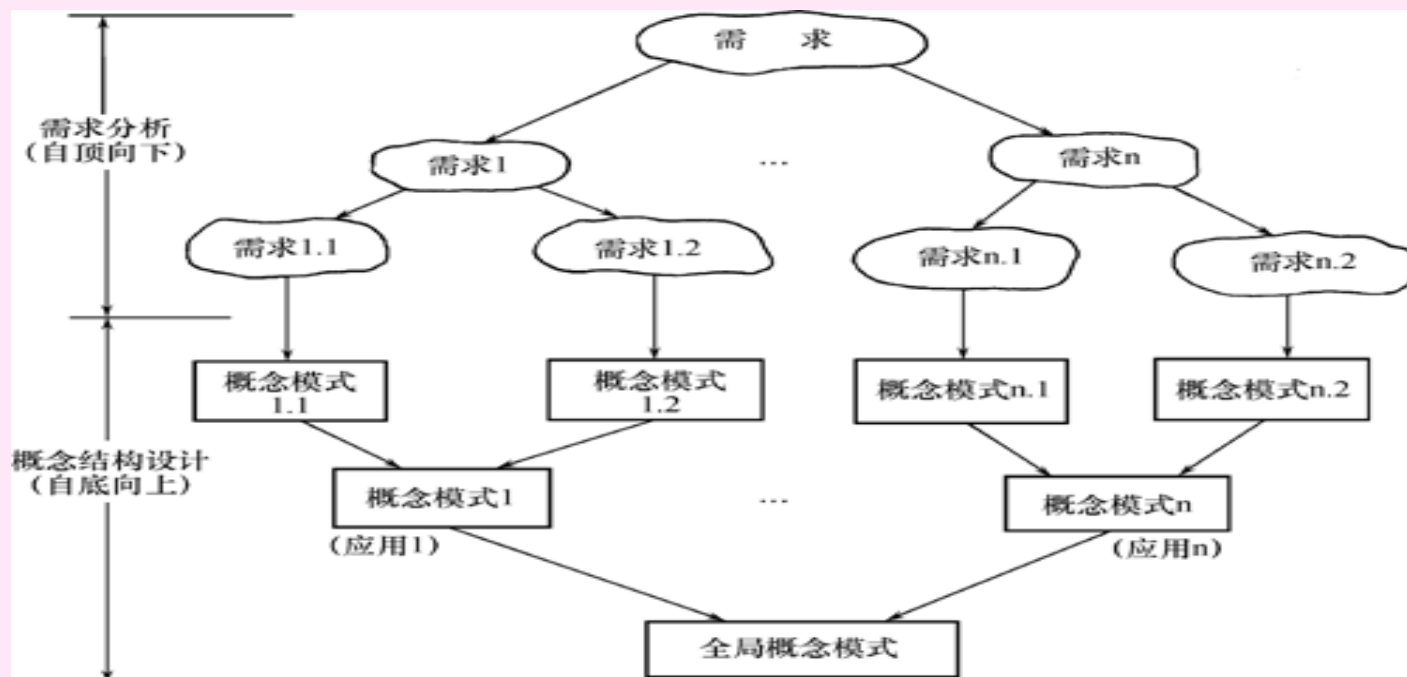
## ■ 混合策略

- 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





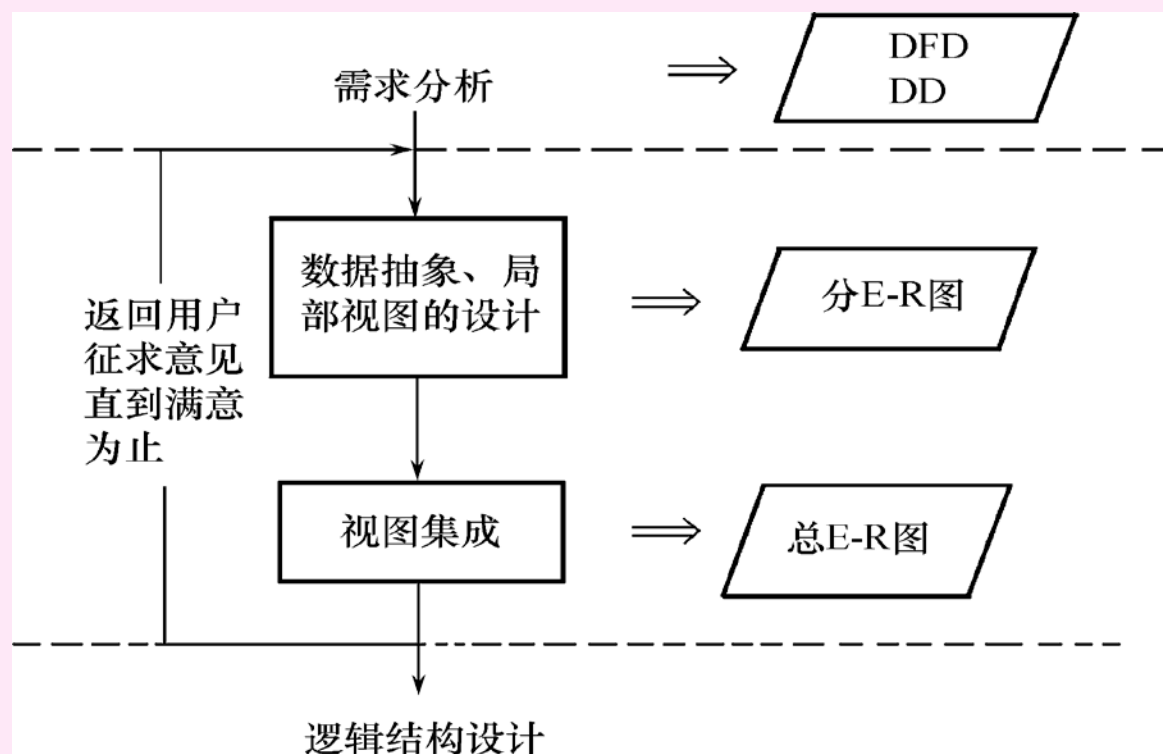
# 概念结构设计的方法与步骤（续）



## ❖ 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





## 知识点85：分E-R图

# 分E-R图



## ■ 设计分E-R图的步骤:

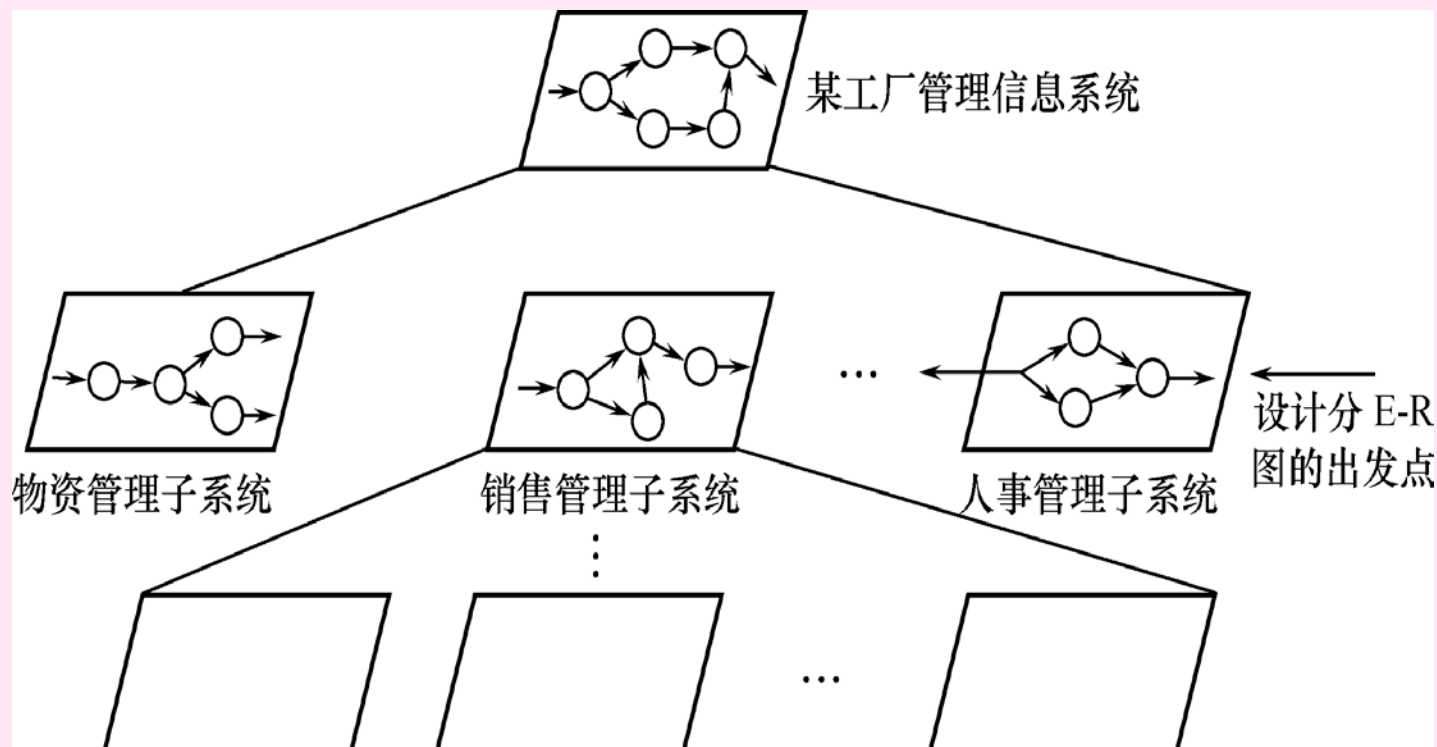
- 选择局部应用
- 逐一设计分E-R图

# 选择局部应用



- 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。

# 选择局部应用（续）



设计分E-R图的出发点

# 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）

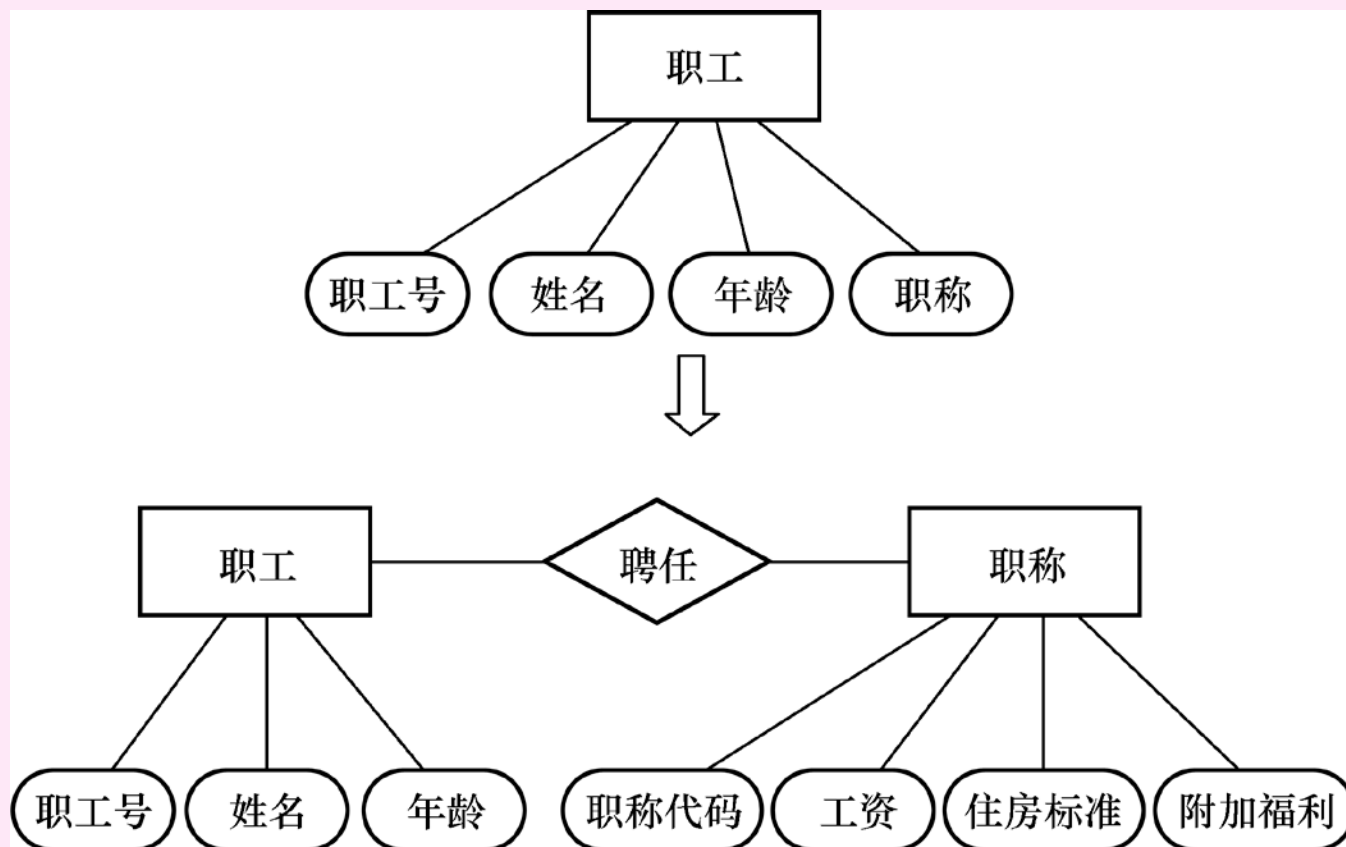
## 逐一设计分E-R图（续）



### ❖ 两条准则：

- 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
- 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）



## [实例] 销售管理子系统分E-R图的设计

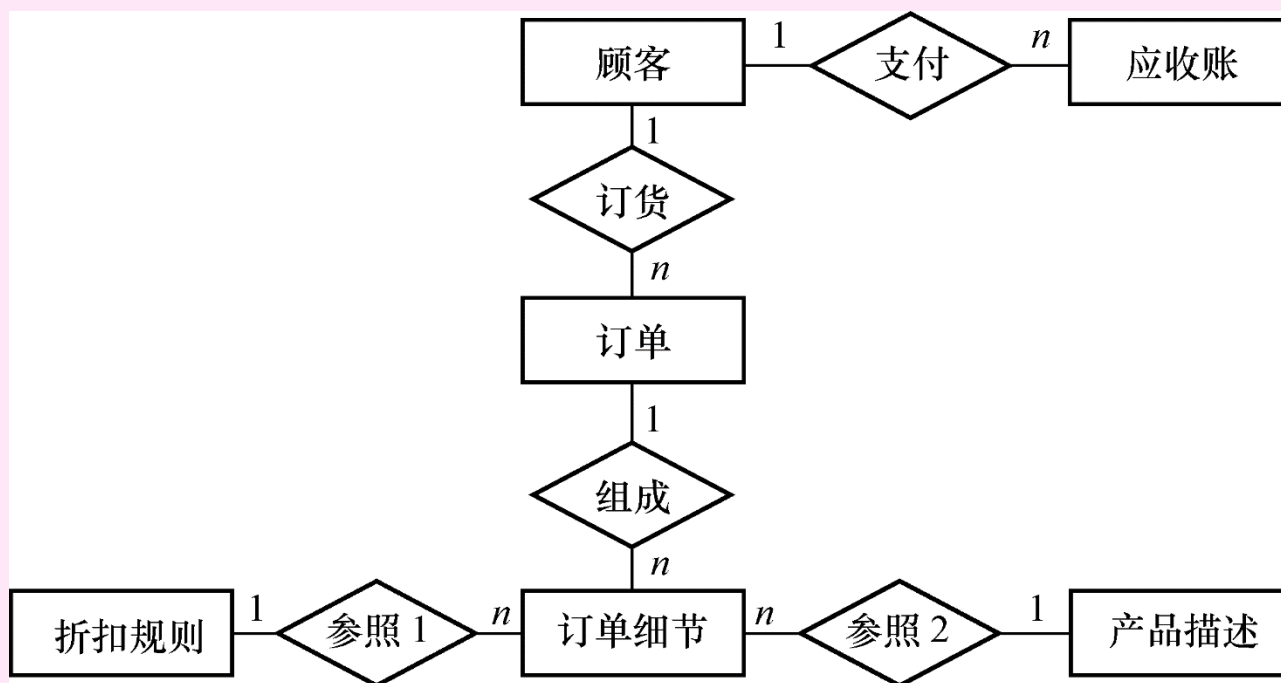
### ❖ 销售管理子系统的主要功能：

1. 处理顾客和销售员送来的订单
2. 工厂是根据订货安排生产的
3. 交出货物同时开出发票
4. 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）

❖ 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}



## 知识点86：合并分E-R图

# 视图的集成



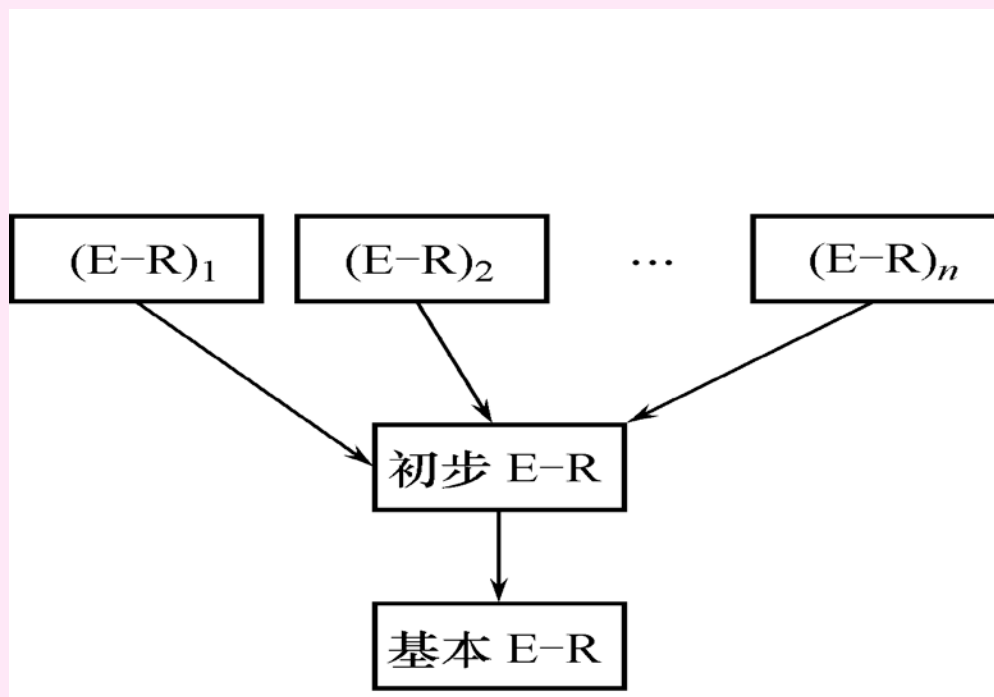
- ❖ 各个局部视图即分**E-R**图建立好后，还需要对它们进行合并，集成为一个整体的数据概念结构即总**E-R**图。



# 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

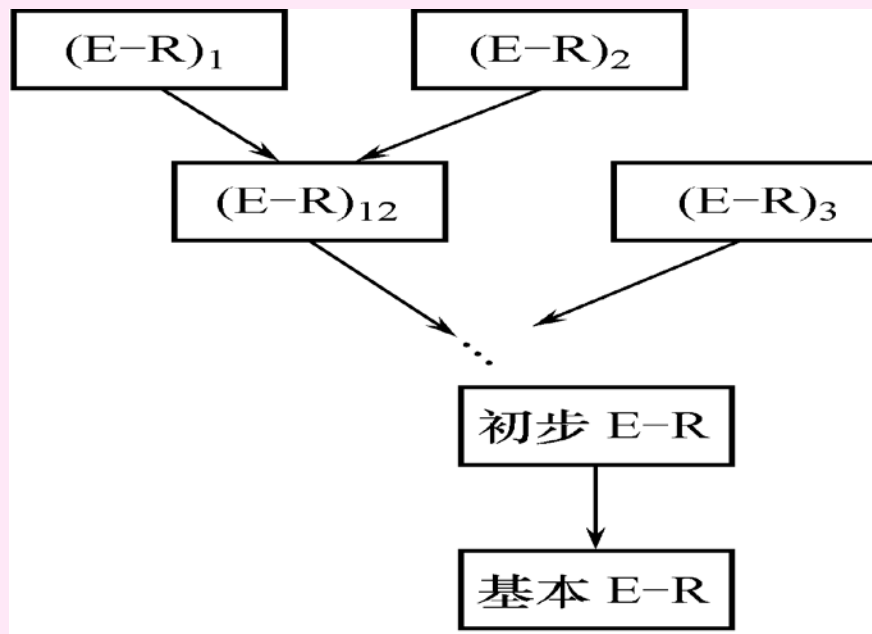




# 视图的集成（续）

## ❖ 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

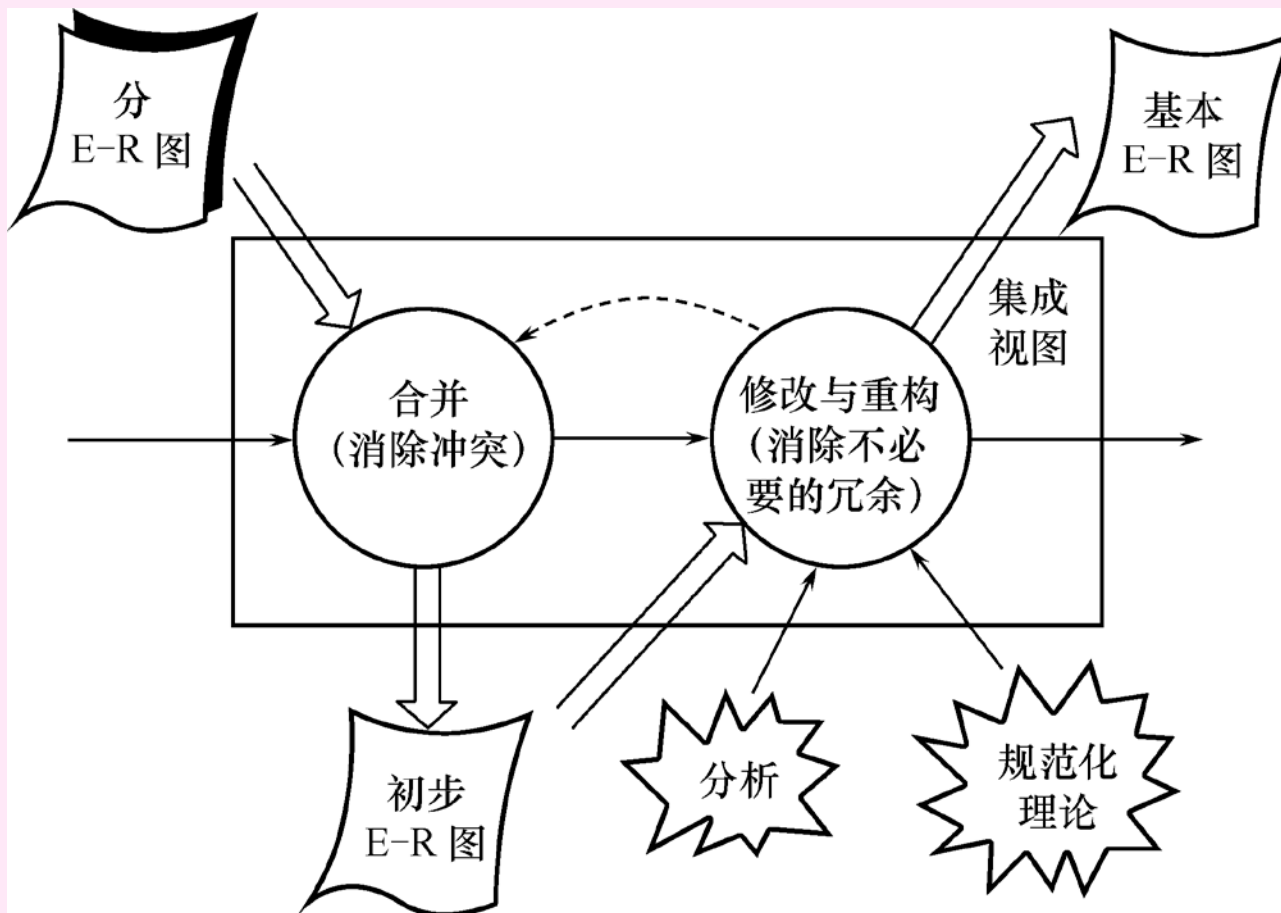
### 1. 合并

1.

### 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



# 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

# 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

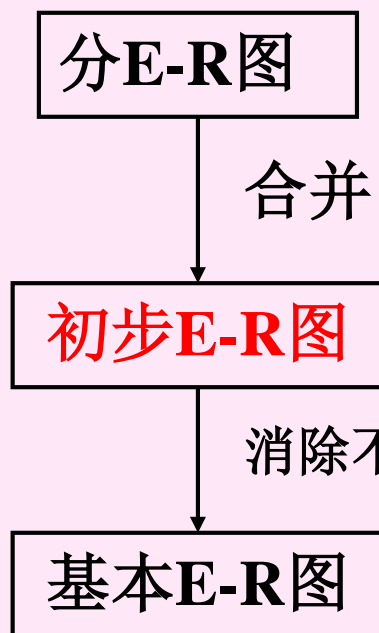
- 同一对象在不同应用中具有不同的抽象
- 同一实体在不同分E-R图所包含的属性个数和属性排列次序不完全相同
- 实体之间的联系在不同局部视图中呈现不同的类型

# 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

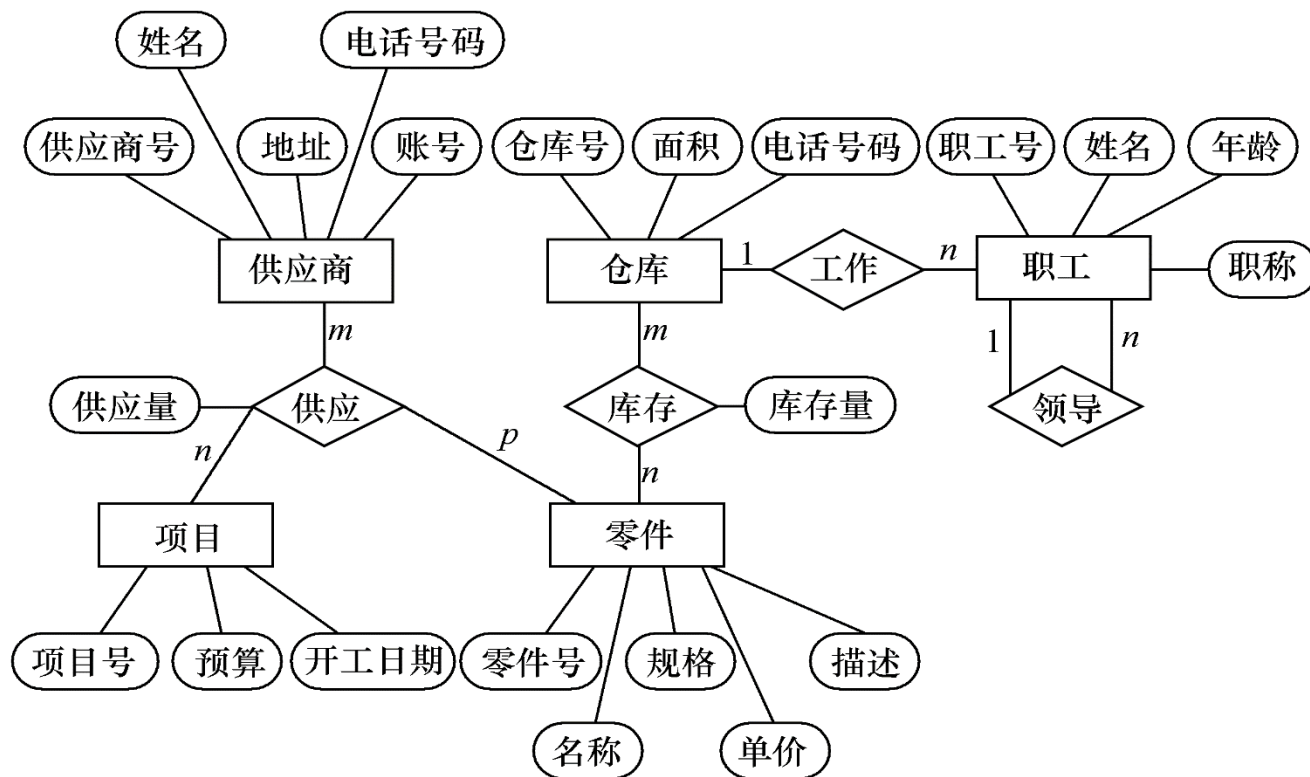
- 函数依赖的概念提供了消除冗余联系的形式化工具



# 物资管理分E-R图



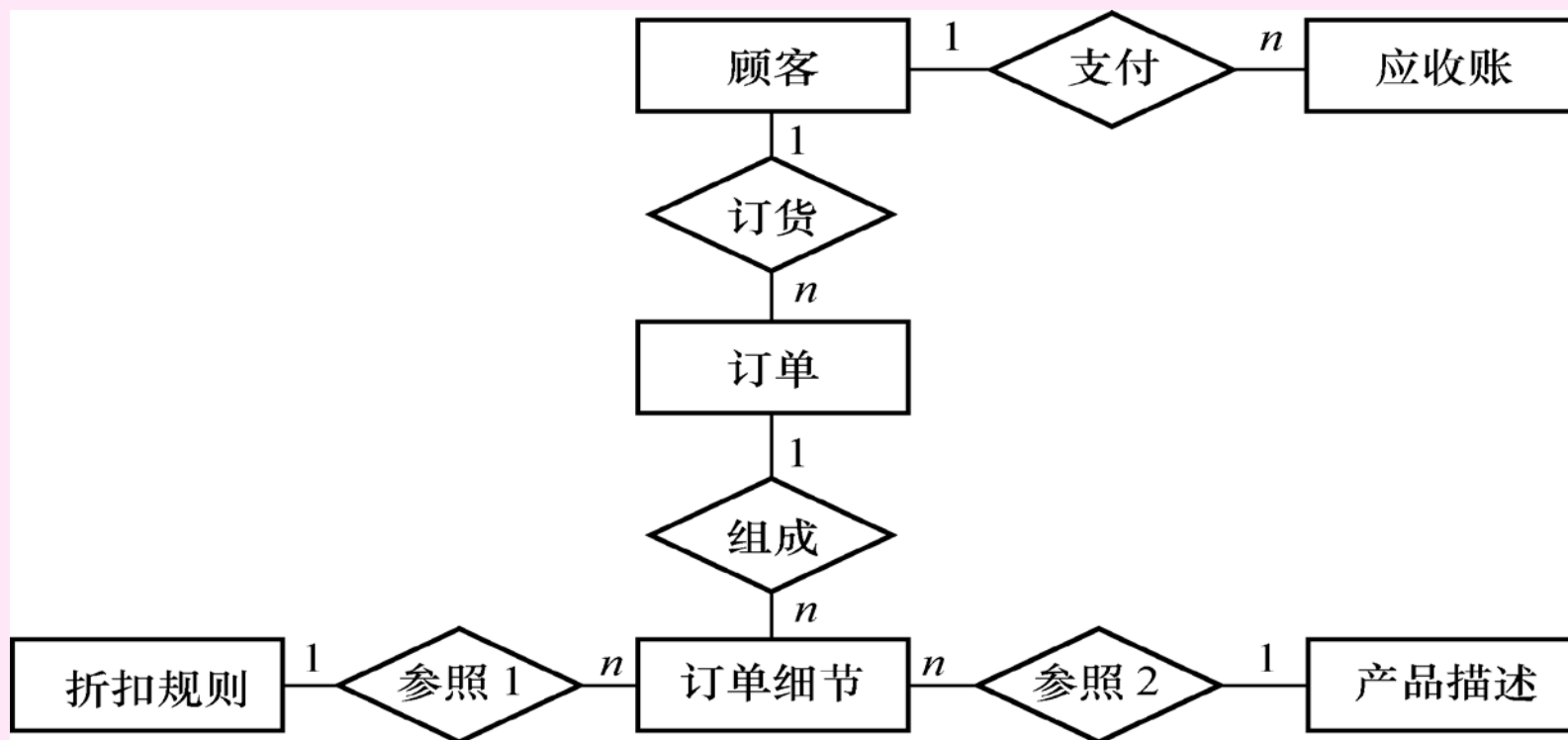
该厂物资管理分E-R图



# 销售管理分E-R图



该厂销售管理分E-R图



# 劳动人事管理分E-R图



该厂劳动人事管理分E-R图

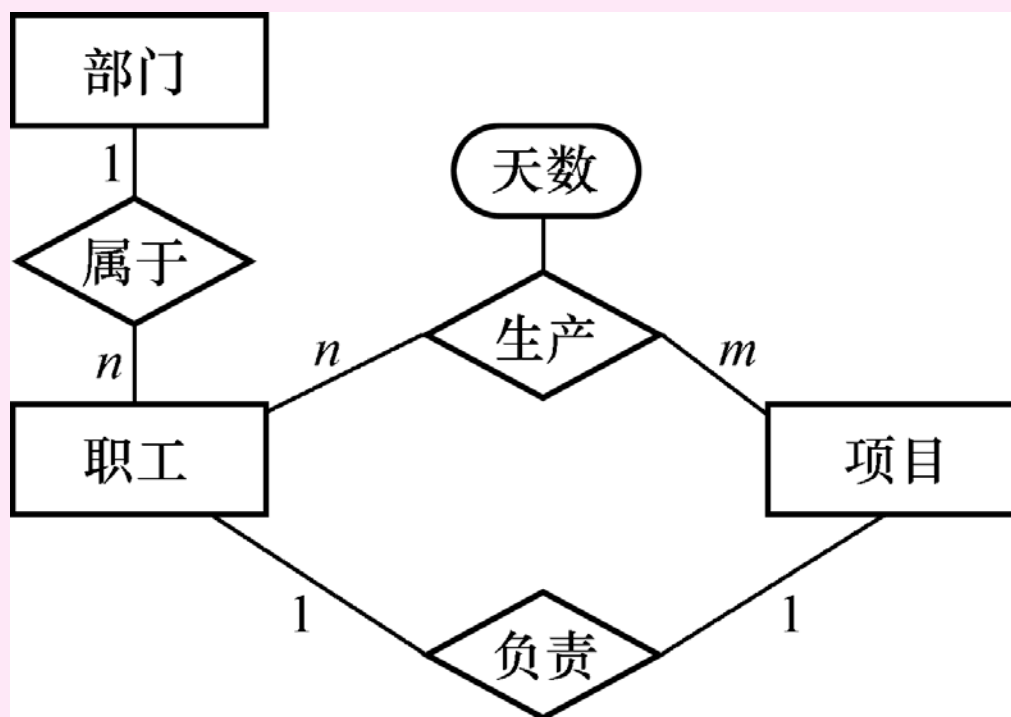
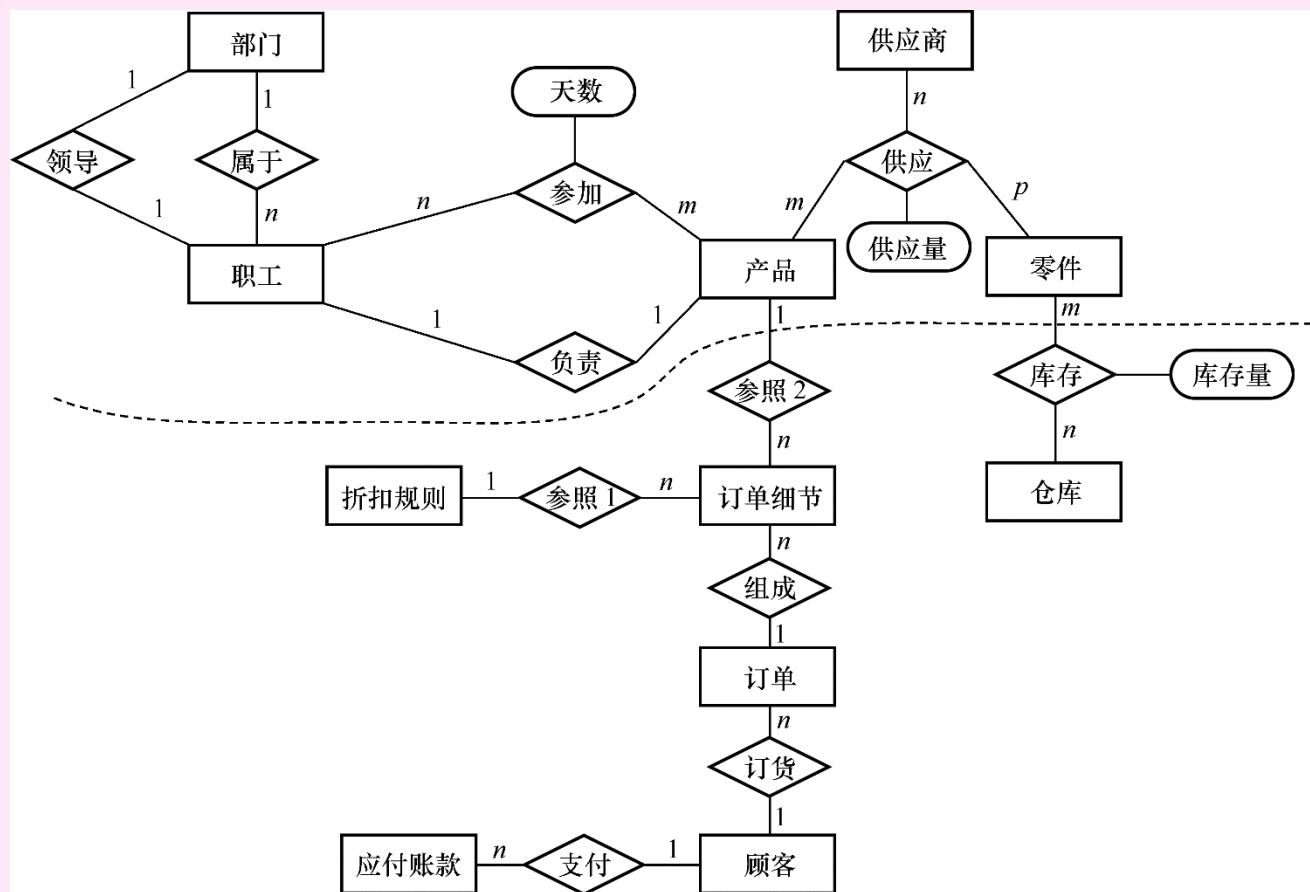


图7.29 劳动人事管理的分E-R图

# 基本E-R图实例



系统的基本E-R(图7.30)



某工厂管理信息系统的基本E-R图

## 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



## ❖ 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



## 知识点87：逻辑结构设计

# 逻辑结构设计



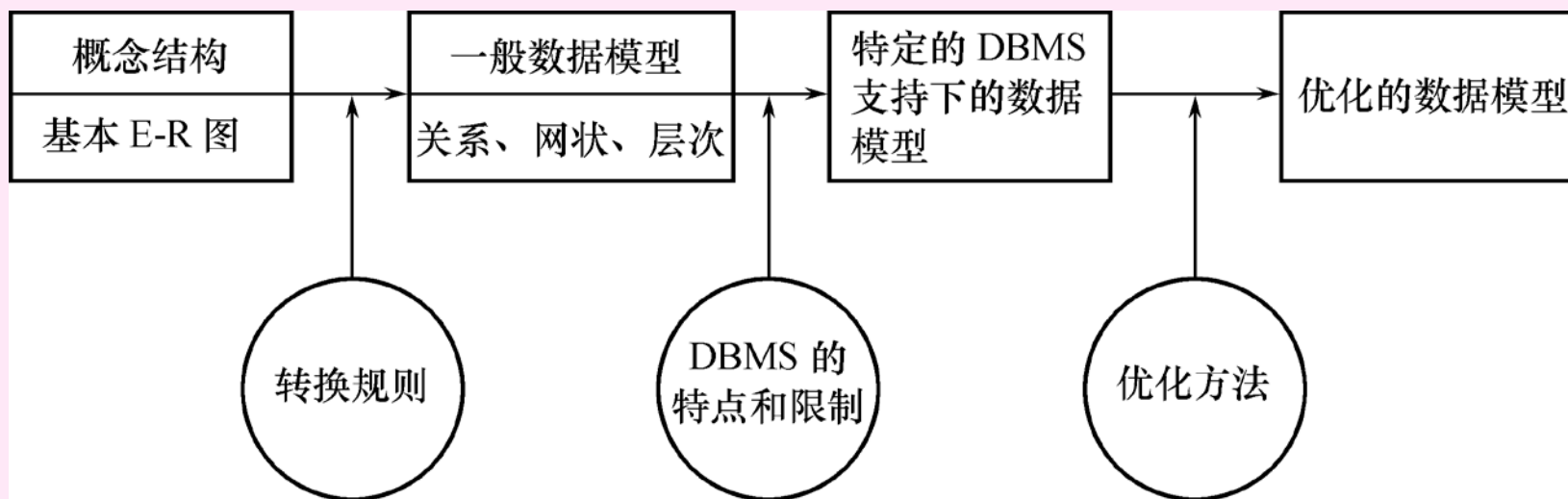
## ❖ 逻辑结构设计的任务

- 把概念结构设计阶段设计好的基本E-R图转换为与选用DBMS产品所支持的数据模型相符合的逻辑结构

## ❖ 逻辑结构设计的步骤

- 将概念结构转化为一般的关系、网状、层次模型
- 将转换来的关系、网状、层次模型向特定DBMS支持下的数据模型转换
- 对数据模型进行优化

# 逻辑结构设计(续)



逻辑结构设计时的3个步骤



## 知识点88: E-R图向关系模型的转换

# E-R图向关系模型的转换



## ❖ E-R图向关系模型的转换要解决的问题

- 如何将实体型和实体间的联系转换为关系模式
- 如何确定这些关系模式的属性和码

## ❖ 转换内容

- 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换为关系模式。

# 从ER模型到关系模型的转换



- 属于逻辑设计
- 基本原则
  - ER图中的每个实体转换为一个关系模式
    - 实体的属性即关系模式的属性
    - 实体标识符即关系模式的键



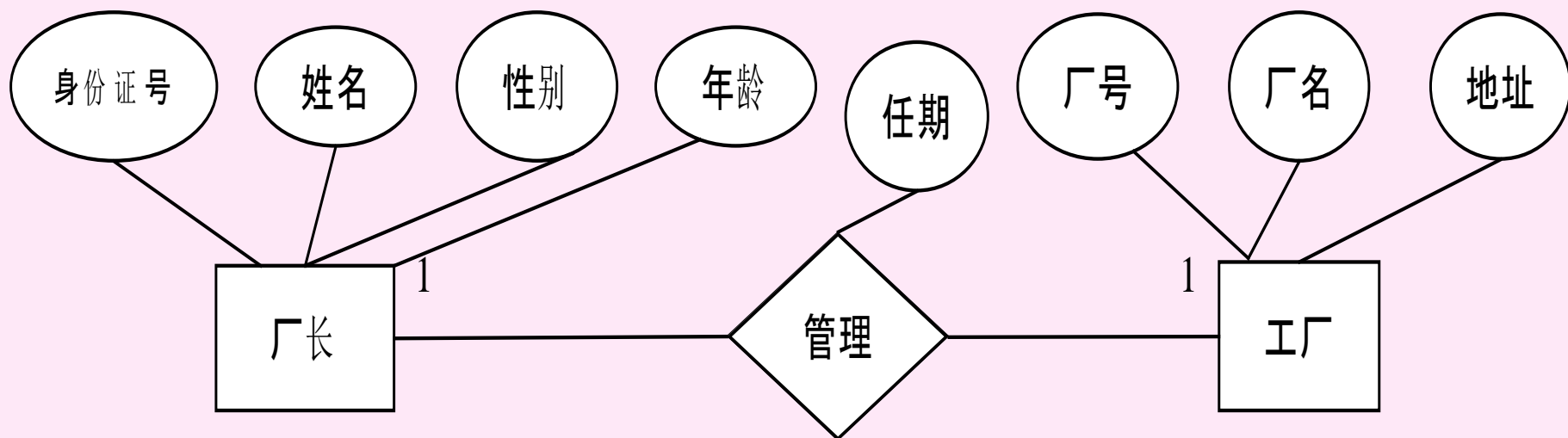
# 从ER模型到关系模型的转换（续）



- 联系的转换根据不同情况做不同的处理
  - 联系是1:1的，在任意一个关系模式的属性中加入另一个关系模式的键和联系的属性；
  - 联系是1:n的，在N端实体类型转换的关系模式中加入1端的关系模式的键和联系类型的属性；
  - 联系是m:n的，将联系类型也转换为关系模型，其属性为两端实体类型的键加上联系类型，键为两端实体键的组合。



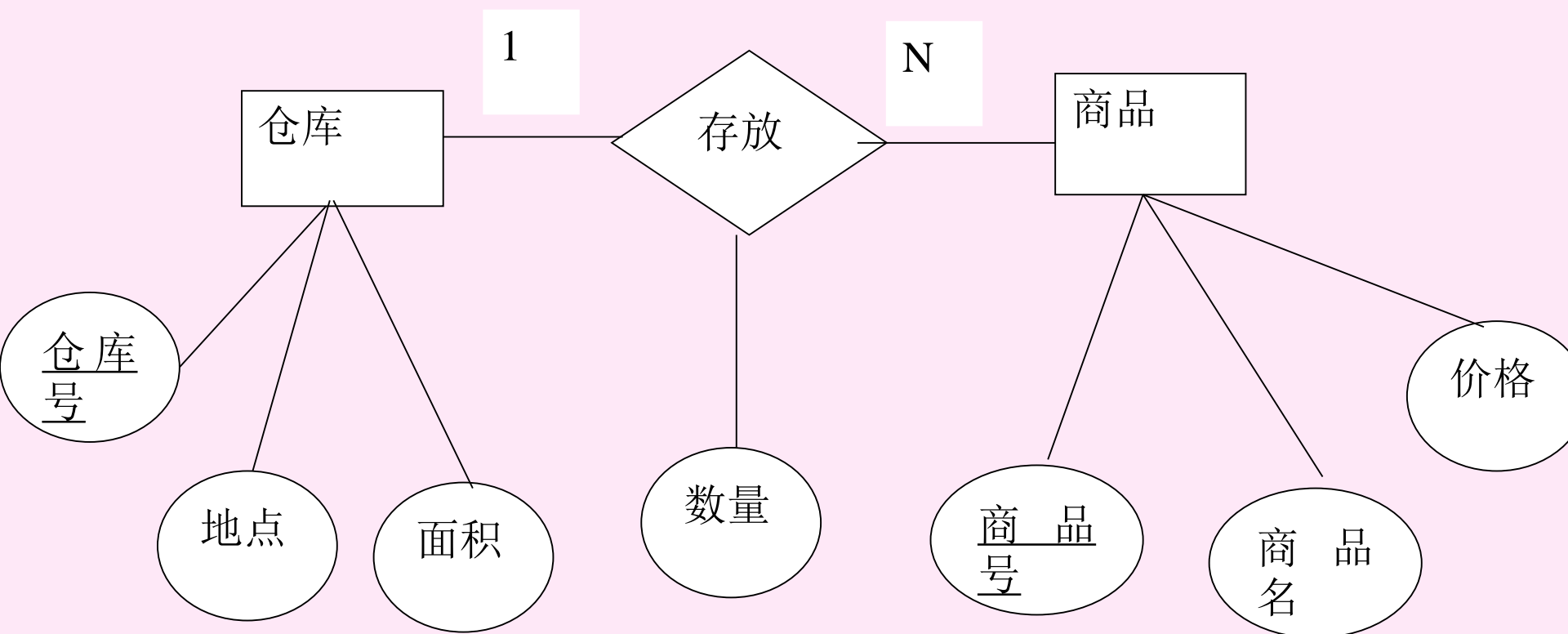
- 厂长实体集有身份证号、姓名、性别和年龄等属性；而工厂实体集有厂号、厂名和地址等属性；厂长与工厂之间联系是一对一联系，联系属性为任期。
- 用ER工具画出”工厂管理”的概念模型，并指出其中的实体标示符；将得到的ER模型转换为关系模型，指出其中的关键字和外键。



厂长（身份证号，姓名，性别，年龄，**厂号**，任期）  
工厂（厂号，厂名，地址）



- 仓库实体集有仓库号、地点和面积等属性；商品实体集有商品号、商品名和价格；商品实体集与仓库实体集是多对一联系，联系名为存放，联系属性为数量。
- 用ER工具画出商品仓储存放的概念模型，并标出实体标示符；将该ER模型转换为关系模型，并指出关键字、外键。





仓库 (仓库号 地点 面积)

商品 (商品号 商品名 价格 **仓库号** 数量)

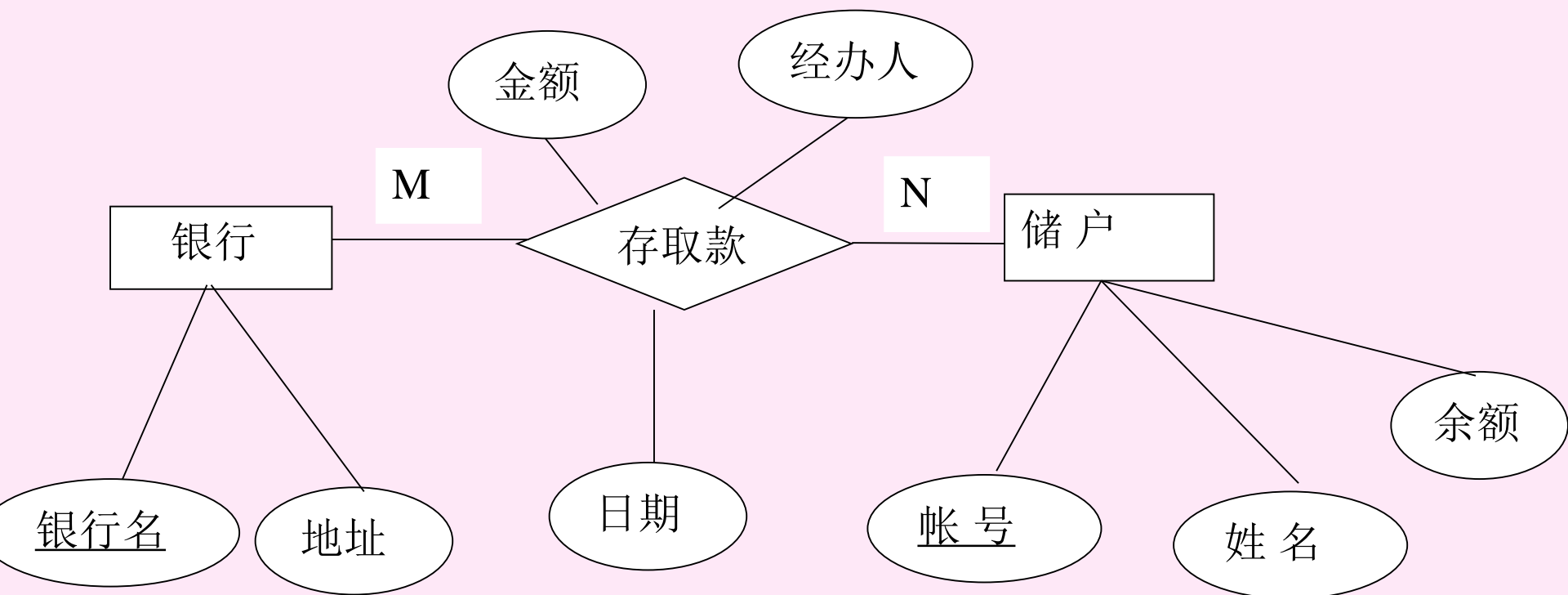
关键字 仓库号

关键字 商品号

外键 仓库号



- 银行实体集包含属性有银行名、地址等，储户实体集包含的属性有帐号、姓名和余额，银行实体集与储户实体集存在多对多联系，联系名为存取款，联系属性为日期、金额和经办人。
- 用ER工具画出该概念模型，并标示出实体标示符；将该ER模型转化为关系模型，并指出主键和外键。







银行（银行名、地址）

主键 银行名

储户（帐号、姓名、余额）

关键字 帐号

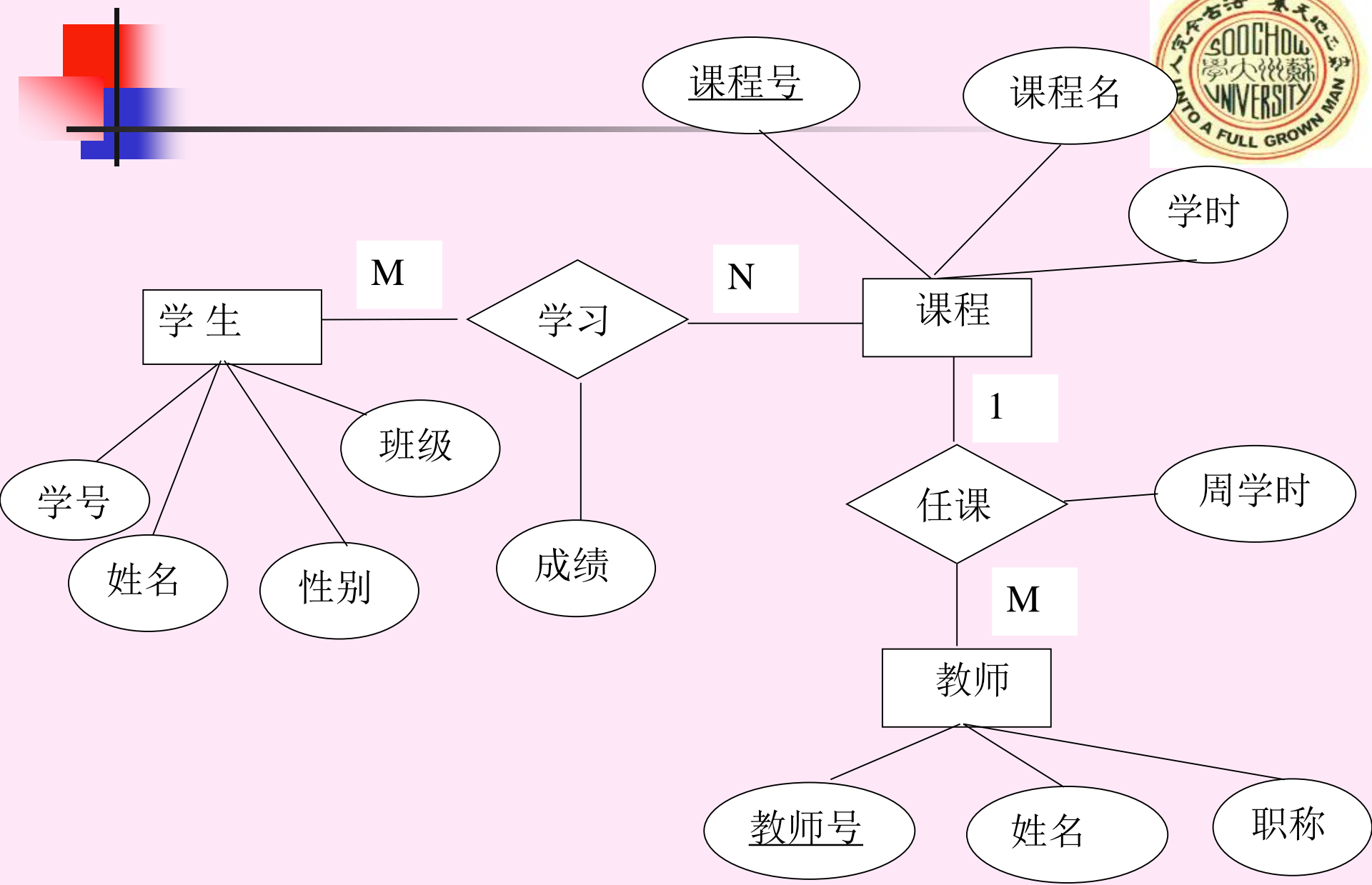
存取款（银行名、帐号、日期、金额、经办人）

关键字 银行名、帐号

外键 银行名、帐号



- 学生实体集中有学号、姓名、性别和班级等属性，课程实体集中有课程号、课程名和学时数等属性，教师实体集中有教师编号、姓名和职称等属性；学生与课程之间联系是多对多联系，联系名为学习，联系属性为成绩，课程与教师之间联系是一对多联系，联系名为任课，联系属性为周学时。
- 用ER工具画出教学管理的概念模型，标示出实体标示符；将获得概念模型转换为关系模型，并指出关键字和外键。





学生 (学号、姓名、性别、班级)

课程 (课程号、课程名、学时数)

教师 (教师号、姓名、职称, 教师号、周学时)

学习 (学号、课程号、成绩)

关键字 学号

关键字 课程号

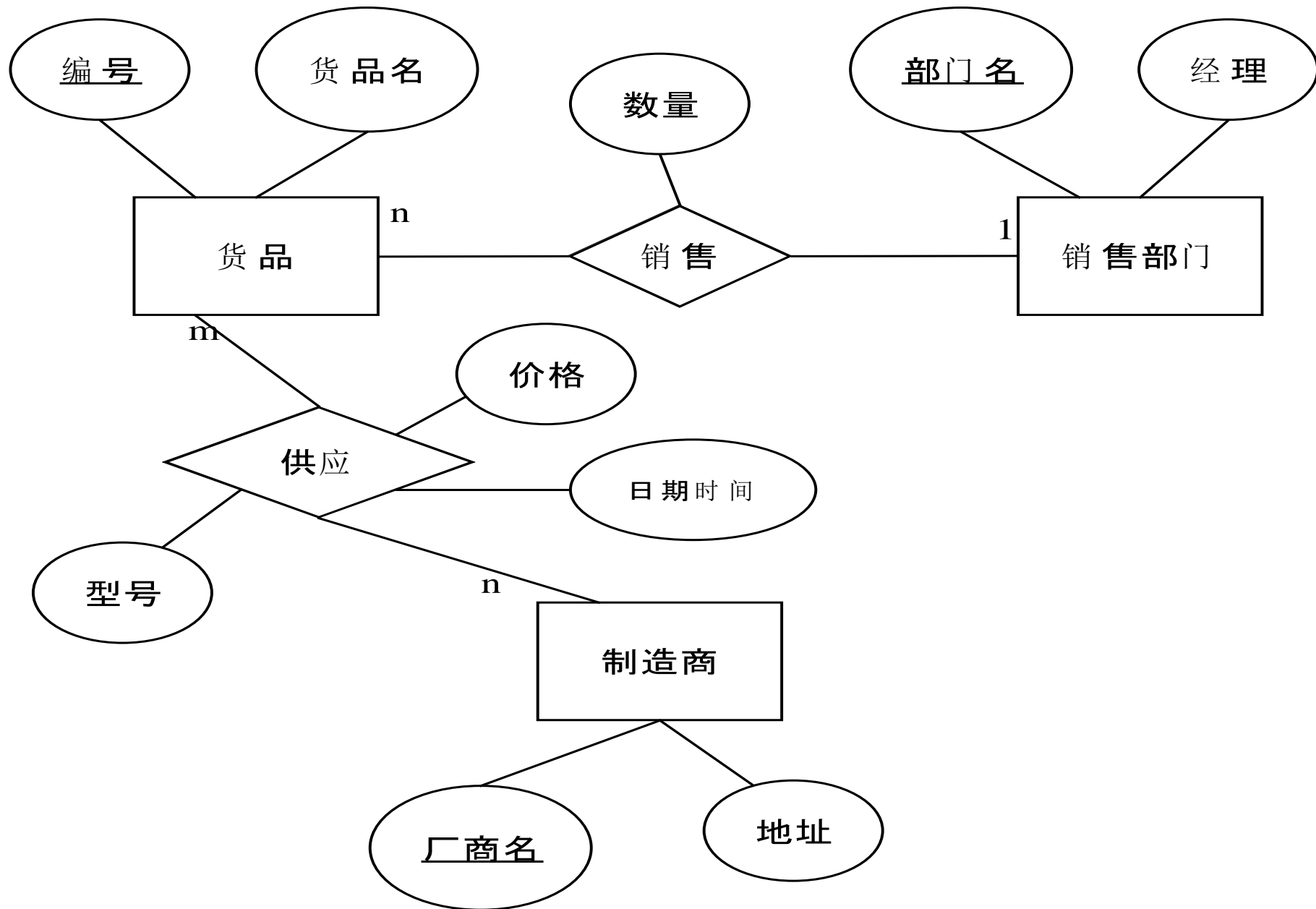
关键字 教师号

关键字 学号、课程号

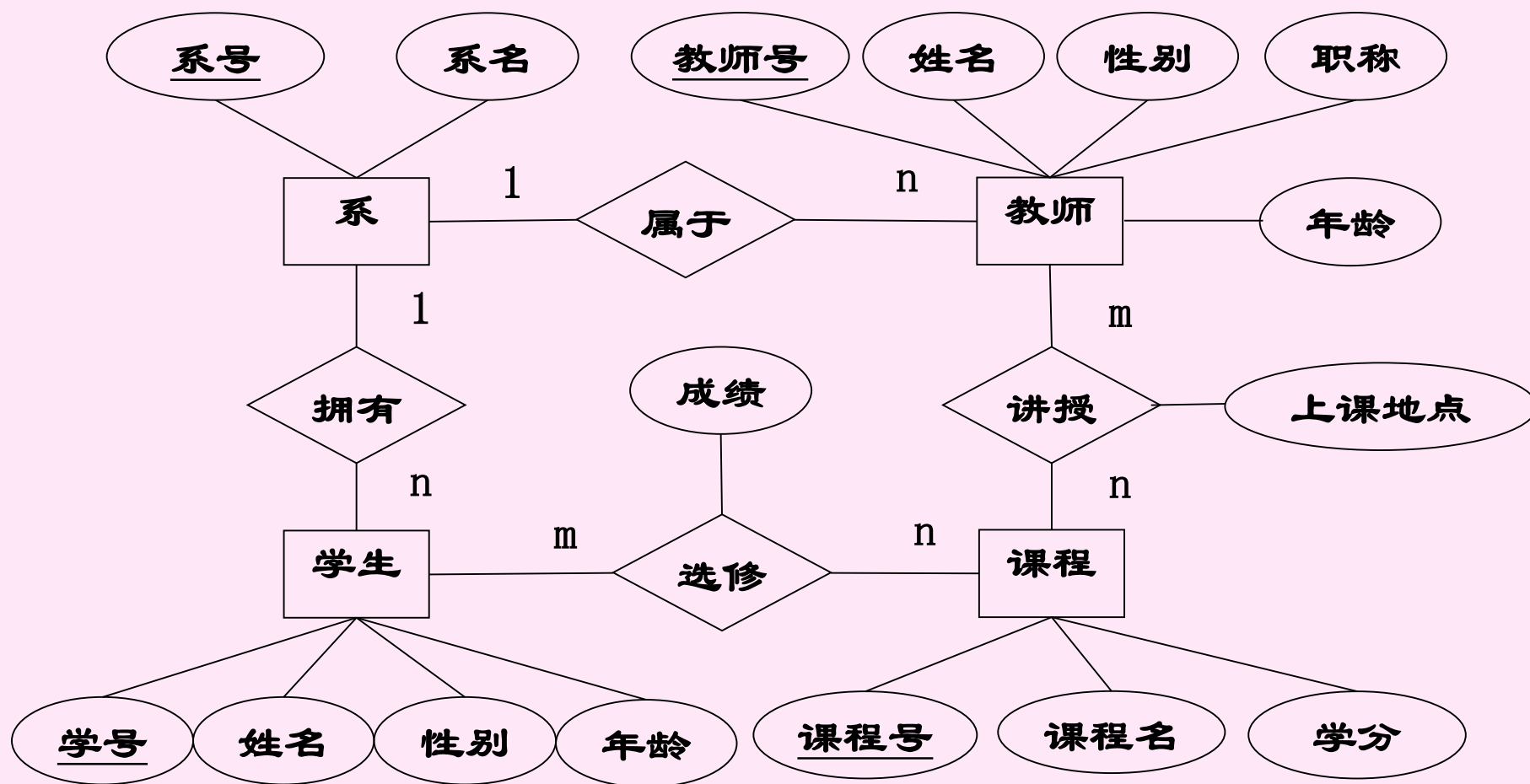
外键 学号、课程号、教师号

一个超市货品销售管理系统有如下信息：货品：编号、货品名；销售部门：部门名、经理；制造商：厂商名、地址；其中约定：超市中的货品用编号唯一标识。一个制造商可以提供多种货品，一种货品也可以由多个制造商提供，制造商具有自己专有的厂商名，制造商提供货品的型号、价格和提供货品日期时间；超市的一个部门可以销售多种货品，某一种货品仅能在一个部门销售，超市内的各部门名具有唯一性。试完成如下设计：

- ①试画出该超市信息完整的E-R图；
- ②将E-R图转换为关系模型结构，并指出关键字和外键。



# 将 ER 图映射到表





## 知识点89：数据模型的优化



# 数据模型的优化



- ❖ 得到初步数据模型后，还应该适当地修改、调整数据模型的结构，以进一步提高数据库应用系统的性能，这就是数据模型的优化
- ❖ 关系数据模型的优化通常以规范化理论为指导

# 数据模型的优化（续）



## ❖ 优化数据模型的方法

### 1. 确定数据依赖

按需求分析阶段所得到的语义，分别写出每个关系模式内部各属性之间的数据依赖以及不同关系模式属性之间数据依赖

### 2. 消除 冗余的联系

对于各个关系模式之间的数据依赖进行极小化处理，消除冗余的联系。

# 数据模型的优化（续）



## 3. 确定所属范式

- 按照数据依赖的理论对关系模式逐一进行分析
- 考查是否存在部分函数依赖、传递函数依赖、多值依赖等
- 确定各关系模式分别属于第几范式

- ## 4. 按照需求分析阶段得到的各种应用对数据处理的要求，分析对于这样的应用环境这些模式是否合适， 确定是否要对它们进行合并或分解。

**注意：**并不是规范化程度越高的关系就越优，一般说来，  
第三范式就足够了

# 数据模型的优化（续）



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

# 数据模型的优化（续）



显然有：

学号 $\rightarrow$ (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，仍然可保留该冗余数据，对关系模式不再做进一步分解



## 知识点90：设计用户子模式

# 设计用户子模式



## ❖ 定义用户外模式时应该注重的問題

包括三个方面：

(1) 使用更符合用户习惯的别名

(2) 针对不同级别的用户定义不同的**View**，以满足系统对安全性的要求。

(3) 简化用户对系统的使用



## 设计用户子模式（续）

[例] 关系模式产品（产品号，产品名，规格，单价，生产车间，生产负责人，产品成本，产品合格率，质量等级），可以在产品关系上建立两个视图：

为一般顾客建立视图：

产品1（产品号，产品名，规格，单价）

为产品销售部门建立视图：

产品2（产品号，产品名，规格，单价，车间，生产负责人）

- 顾客视图中只包含允许顾客查询的属性
- 销售部门视图中只包含允许销售部门查询的属性
- 生产领导部门则可以查询全部产品数据
- 可以防止用户非法访问不允许他们查询的数据，保证系统的安全性



# 逻辑结构设计小结



## ❖ 任务

- 将概念结构转化为具体的数据模型

## ❖ 逻辑结构设计的步骤

- 将概念结构转化为一般的关系、网状、层次模型
- 将转化来的关系、网状、层次模型向特定**DBMS**支持下的数据模型转换
- 对数据模型进行优化
- 设计用户子模式

# 逻辑结构设计小结



- ❖ E-R图向关系模型的转换内容
- ❖ E-R图向关系模型的转换原则

# 逻辑结构设计小结



## ❖ 优化数据模型的方法

1. 确定数据依赖
2. 对于各个关系模式之间的数据依赖进行极小化处理，消除冗余的联系。
3. 确定各关系模式分别属于第几范式。
4. 分析对于应用环境这些模式是否合适，确定是否要对它们进行合并或分解。
5. 对关系模式进行必要的分解或合并

# 逻辑结构设计小结



## ❖ 设计用户子模式

1. 使用更符合用户习惯的别名
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。
3. 简化用户对系统的使用



# 知识点91：数据库物理设计

# 数据库的物理设计



## ❖ 数据库的物理设计

- 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，它依赖于选定的数据库管理系统
- 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

# 数据库的物理设计(续)

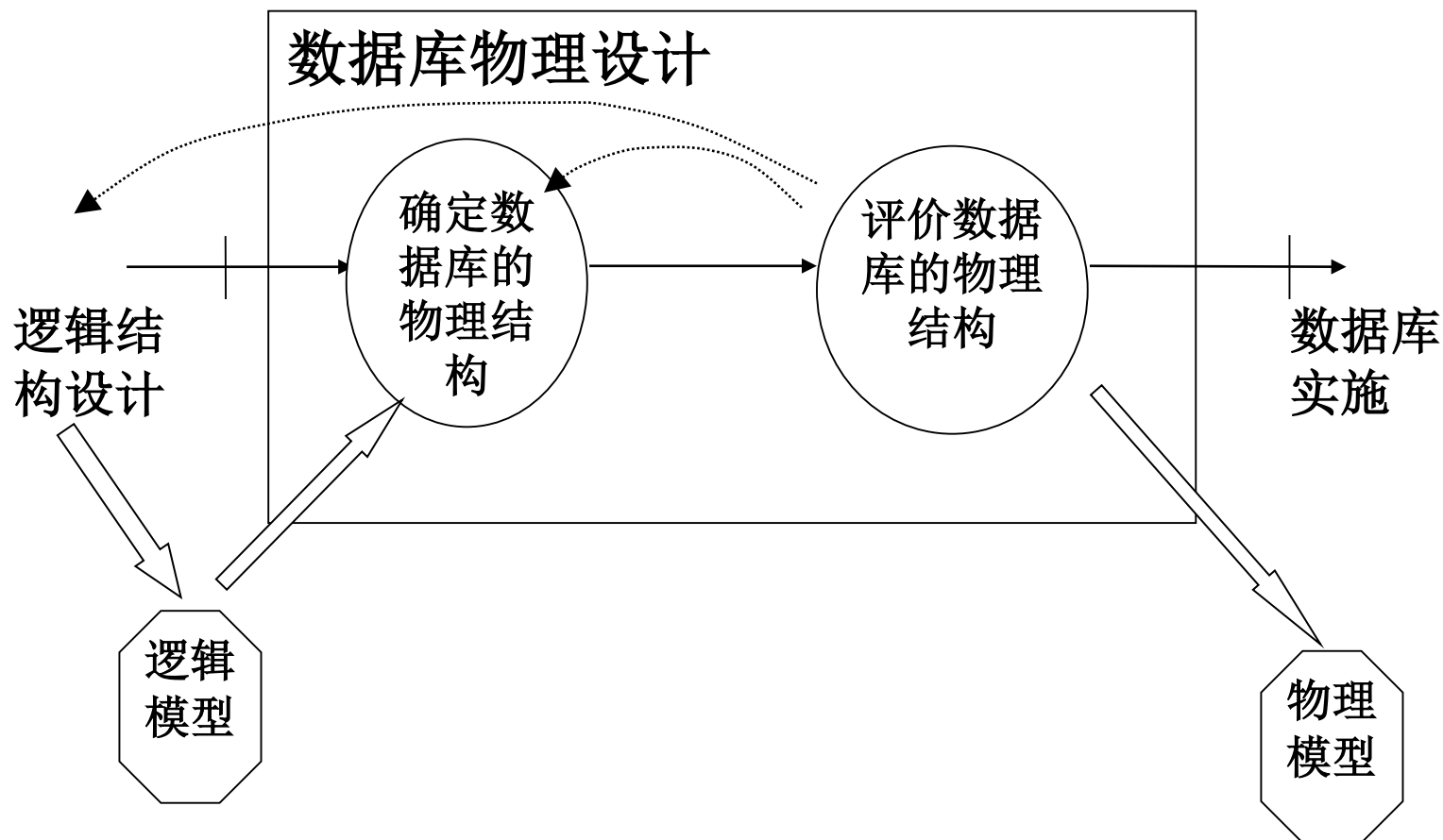


## ❖ 数据库物理设计的步骤

- 确定数据库的物理结构，在关系数据库中主要指存取方法和存储结构
- 对物理结构进行评价，评价的重点是时间和空间效率

如果评价结果满足原设计要求，则可进入到物理实施阶段，否则，就需要重新设计或修改物理结构，有时甚至要返回逻辑设计阶段修改数据模型

# 数据库的物理设计(续)





# 数据库物理设计的内容和方法



## ❖ 设计物理数据库结构的准备工作

- 对要运行的事务进行详细分析，获得选择物理数据库设计所需参数
- 充分了解所用RDBMS的内部特征，特别是系统提供的存取方法和存储结构

# 数据库的物理设计的内容和方法（续）



## ❖ 选择物理数据库设计所需参数

### ■ 数据库查询事务

- 查询的关系
- 查询条件所涉及的属性
- 连接条件所涉及的属性
- 查询的投影属性

# 数据库的物理设计的内容和方法（续）



## ❖ 选择物理数据库设计所需参数(续)

### ■ 数据更新事务

- 被更新的关系
- 每个关系上的更新操作条件所涉及的属性
- 修改操作要改变的属性值

### ■ 每个事务在各关系上运行的频率和性能要求

# 数据库的物理设计的内容和方法（续）



## ❖ 关系数据库物理设计的内容

- 为关系模式选择存取方法(建立存取路径)
- 设计关系、索引等数据库文件的物理存储结构



## 知识点92：数据库实施和维护

# 数据的载入



- ❖ 数据库结构建立好后，就可以向数据库中装载数据了。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

# 应用程序的编码和调试



- ❖ 数据库应用程序的设计应该与数据设计并行进行
- ❖ 在组织数据入库的同时还要调试应用程序

# 数据库的试运行



- ❖ 在原有系统的数据有一小部分已输入数据库后，就可以开始对数据库系统进行联合调试，称为数据库的试运行
- ❖ 数据库试运行主要工作包括：
  - 1) 功能测试
    - 实际运行数据库应用程序，执行对数据库的各种操作，测试应用程序的功能是否满足设计要求
    - 如果不满足，对应用程序部分则要修改、调整，直到达到设计要求
  - 2) 性能测试
    - 测量系统的性能指标，分析是否达到设计目标
    - 如果测试的结果与设计目标不符，则要返回物理设计阶段，重新调整物理结构，修改系统参数，某些情况下甚至要返回逻辑设计阶段，修改逻辑结构



# 数据库的试运行（续）



强调两点：

## ❖ 分期分批组织数据入库

- 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太大，费时、费力，所以应分期分批地组织数据入库
- 先输入小批量数据供调试用
  - 待试运行基本合格后再大批量输入数据
  - 逐步增加数据量，逐步完成运行评价

# 数据库的试运行（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。

# 数据库的运行与维护



- ❖ 数据库试运行合格后，数据库即可投入正式运行。
- ❖ 数据库投入运行标志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
  - 数据库运行过程中物理存储会不断变化

# 数据库的运行与维护（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
  2. 数据库的安全性、完整性控制
  3. 数据库性能的监督、分析和改进
  4. 数据库的重组和重构造

# 小结



## ❖ 数据库的设计过程

- 需求分析
- 概念结构设计
- 逻辑结构设计
- 物理设计
- 实施和维护

## 小结（续）



### ❖ 数据库各级模式的形成

- 数据库的各级模式是在设计过程中逐步形成的
- 需求分析阶段综合各个用户的应用需求（现实世界的需求）
- 概念设计阶段形成独立于机器特点、独立于各个**DBMS**产品的**概念模式**（信息世界模型），用**E-R**图来描述

## 小结（续）



- 在逻辑设计阶段将**E-R**图转换成具体的数据库产品支持的数据模型如关系模型，形成数据库**逻辑模式**。然后根据用户处理的要求，安全性的考虑，在基本表的基础上再建立必要的视图（**VIEW**）形成数据的**外模式**
- 在物理设计阶段根据**DBMS**特点和处理的需要，进行物理存储安排，设计索引，形成数据库**内模式**