

第7章 数据库设计

本章目标

■ 完成本章的学习，你应该能够

- 理解数据库设计的生命周期
- 掌握需求分析的方法和步骤
- 熟练掌握概念设计的E-R图建模
- 熟练掌握E-R图的集成方法及过程
- 熟练掌握E-R图转化为关系模型的方法
- 熟练应用关系规范化理论优化关系模型
- 理解不同索引存取方法的异同，掌握索引选择的基本方法
- 了解数据库实施和维护的基本内容

大纲

- **数据库设计概述**
- 需求分析
- 概念结构设计
- 逻辑结构设计
- 物理结构设计
- 数据库的实施和维护
- 本章小结

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。

- 信息管理要求：在数据库中应该存储和管理哪些数据对象。
- 数据操作要求：对数据对象需要进行哪些操作，如查询、增、删、改、统计等操作。

■ 数据库设计的目标是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：

- 数据库数据的存取效率高；
- 数据库存储空间的利用率高；
- 数据库系统运行管理的效率高

数据库设计概述

- **数据库设计的特点**
- 数据库设计方法
- 数据库设计的基本步骤
- 数据库设计过程中的各级模式

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

- 数据的收集、整理、组织和不断更新是数据库建设中的重要环节

■ 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

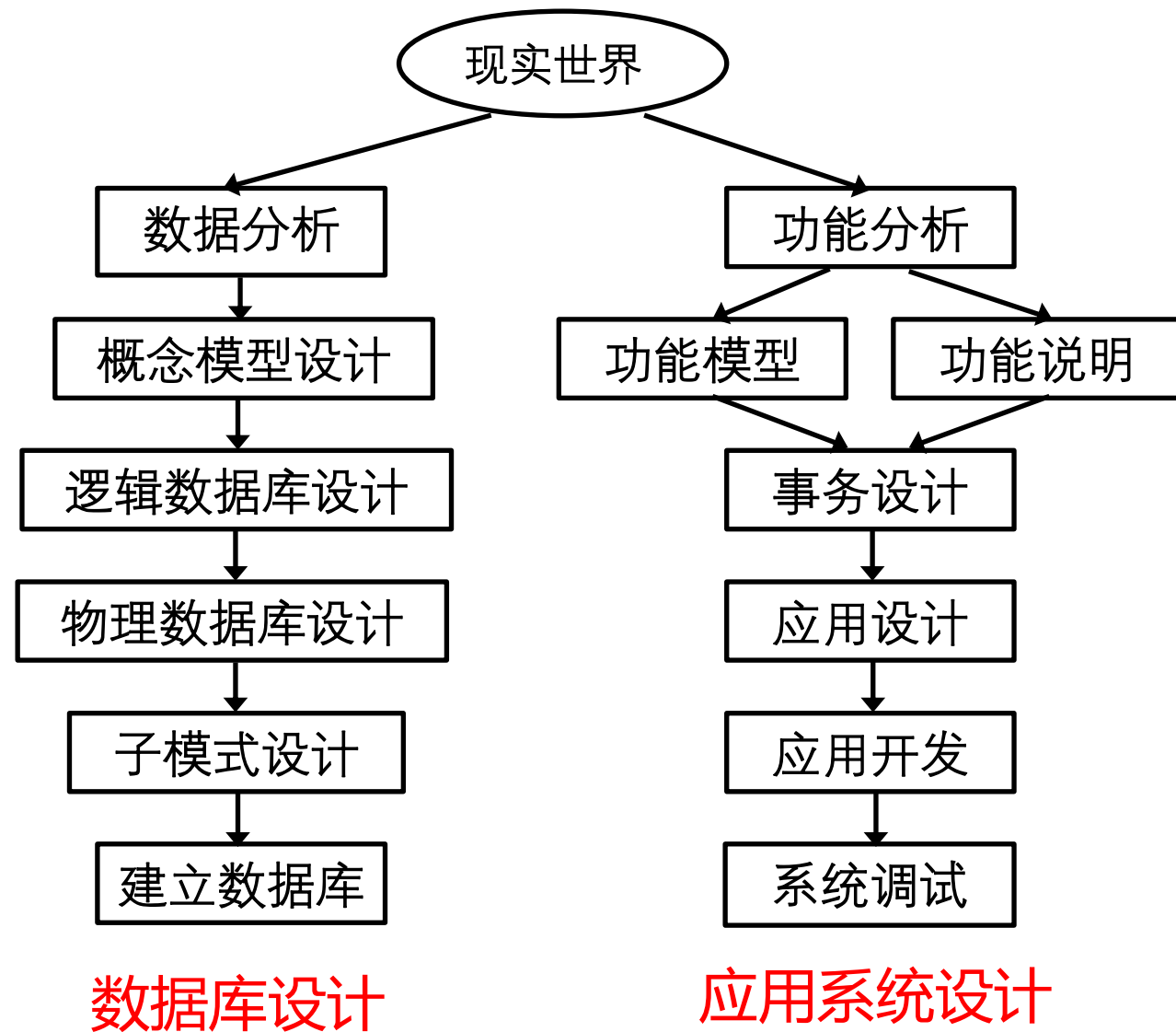


图7.1 结构和行为分离的设计

数据库设计概述

- 数据库设计的特点
- **数据库设计方法**
- 数据库设计的基本步骤
- 数据库设计过程中的各级模式

数据库设计方法

- 大型数据库设计是涉及多学科的综合性的技术，又是一项庞大的工程项目。
- 它要求多方面的知识和技术，主要包括：
 - 计算机的基础知识
 - 软件工程的原理和方法
 - 程序设计的方法和技巧
 - 数据库的基本知识
 - 数据库设计技术
 - 应用领域的知识

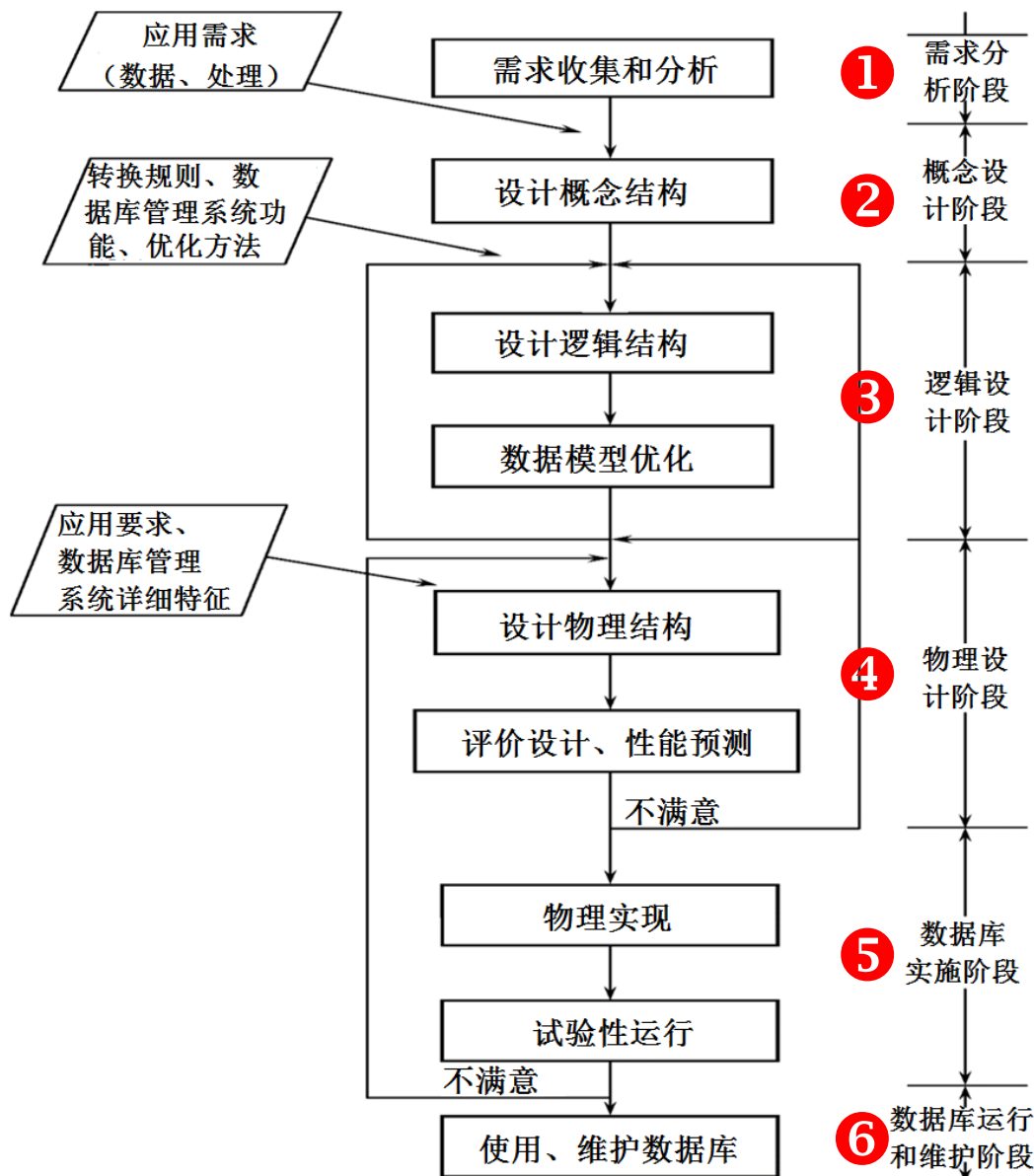
■ 常用数据库设计方法

- 新奥尔良 (New Orleans) 方法
- 基于E-R模型的数据库设计方法
- 3NF (第三范式) 的设计方法
- 面向对象的数据库设计方法
- 统一建模语言 (UML) 方法

- 为保证数据库系统的开发质量，提高开发效率，市场上有很多的数据库设计工具被普遍用于大型数据库的设计中。

- 工具列表参考：<https://www.databasestar.com/data-modeling-tools/>

数据库设计的基本步骤



- 整个设计的**基础**
- 是否做得充分与准确，决定了构建数据库的速度和质量，独立于具体DBMS
- 整个设计的**关键**
- 通过对用户需求进行综合、归纳和抽象，形成一个独立于具体DBMS的概念模型
- 将概念结构转换为某个数据库管理系统所支持的数据模型，并对其进行优化
- 与具体DBMS类型相关
- 为逻辑数据结构选取一个最适合应用环境的物理结构，与具体DBMS类型相关
- 包括存储结构和存取方法
- 由DBMS自动完成
- 设计人员运用DBMS提供的数据库语言及其宿主语言，根据逻辑设计和物理设计的结果建立数据库
- 编写与调试应用程序，组织数据入库，试运行
- 应用程序由程序员完成
- 数据库运行过程中的评估、调整与修改
- 由DBA负责

- 设计一个完善的数据库应用系统往往是上述6个阶段的**不断反复**。
- 上述设计步骤既是数据库设计的过程，也包括了数据库应用系统的设计过程。

- **参加数据库设计的人员**

- | | | | |
|-----------|---|---|------------------------------|
| – 系统分析员 | } | ➡ | • 数据库设计的核心人员 |
| – 数据库设计人员 | | | • 自始至终参与数据库设计，水平高低决定数据库系统的质量 |
| – 应用开发人员 | | ➡ | • 程序员：负责编写程序 |
| | | | • 操作员：负责准备软硬件环境 |
| – DBA | } | ➡ | • 参与需求分析 |
| – 用户代表 | | | • 数据库的运行和维护 |

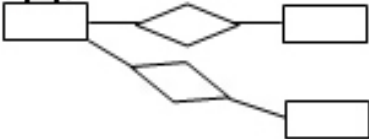
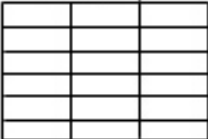
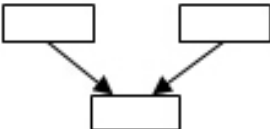
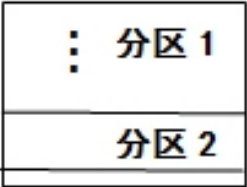

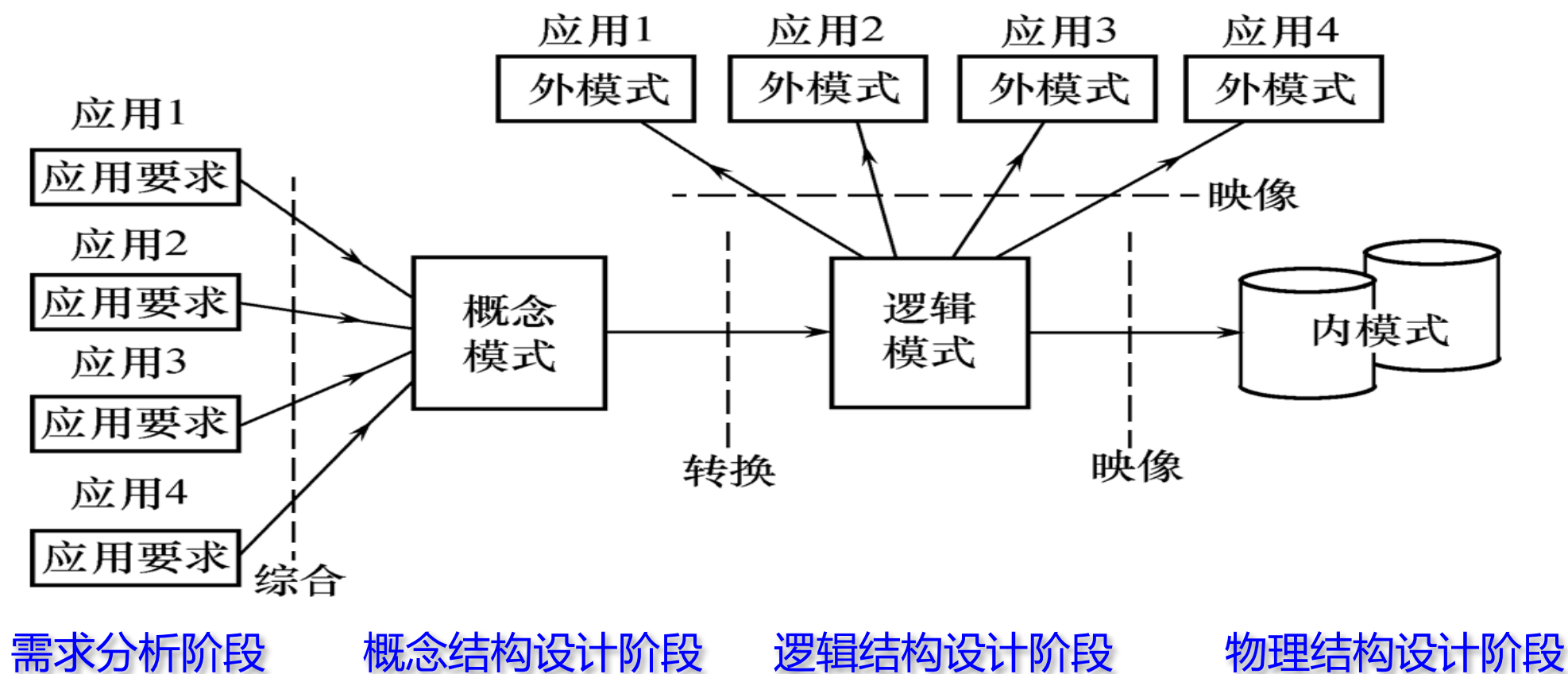
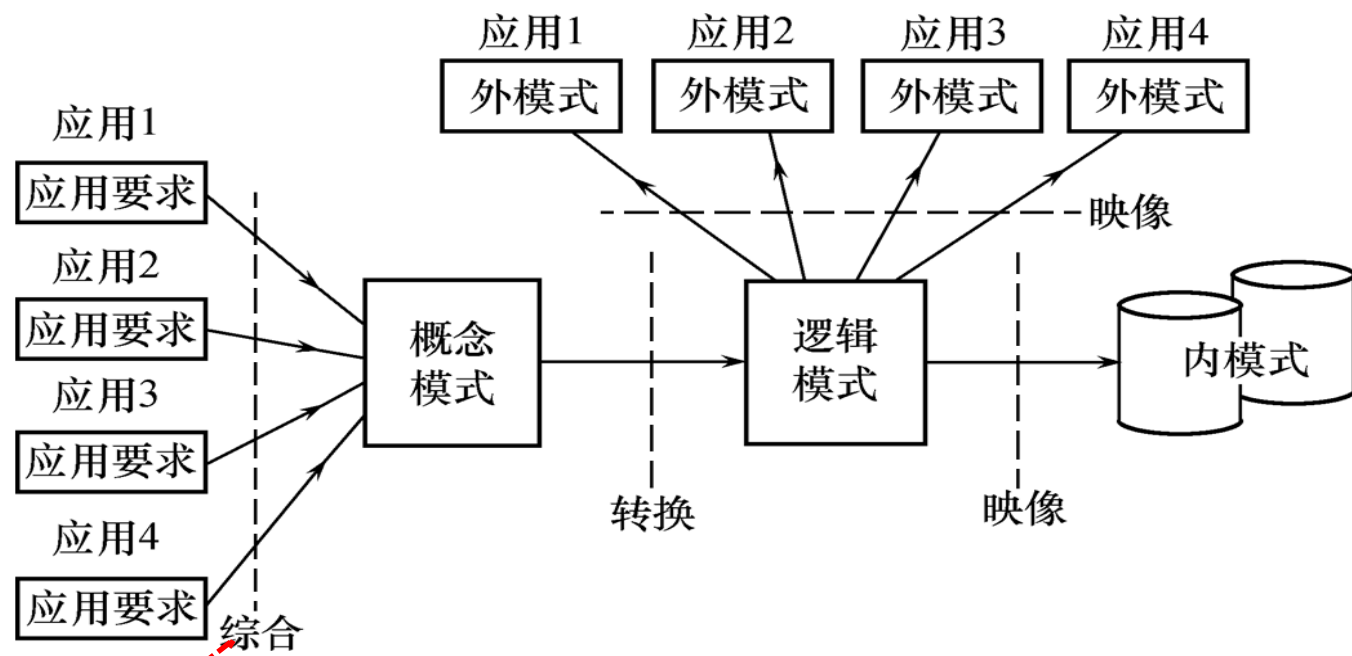
设计阶段	设计描述
需求分析	数据字典 全系统中数据项、数据结构、数据流、数据存储的描述
概念结构设计	概念模型 (E-R 图)  数据字典
逻辑结构设计	某种数据模型 <div> 关系  </div> <div> 非关系  </div>
物理结构设计	存储安排 存取方法选择 存取路径建立 
数据库实施	创建数据库模式 装入数据 数据库试运行 
数据库运行和维护	性能监测、转储/恢复、数据库重组和重构

图7.3 数据库设计各个阶段的数据设计描述

数据库设计过程中的各级模式

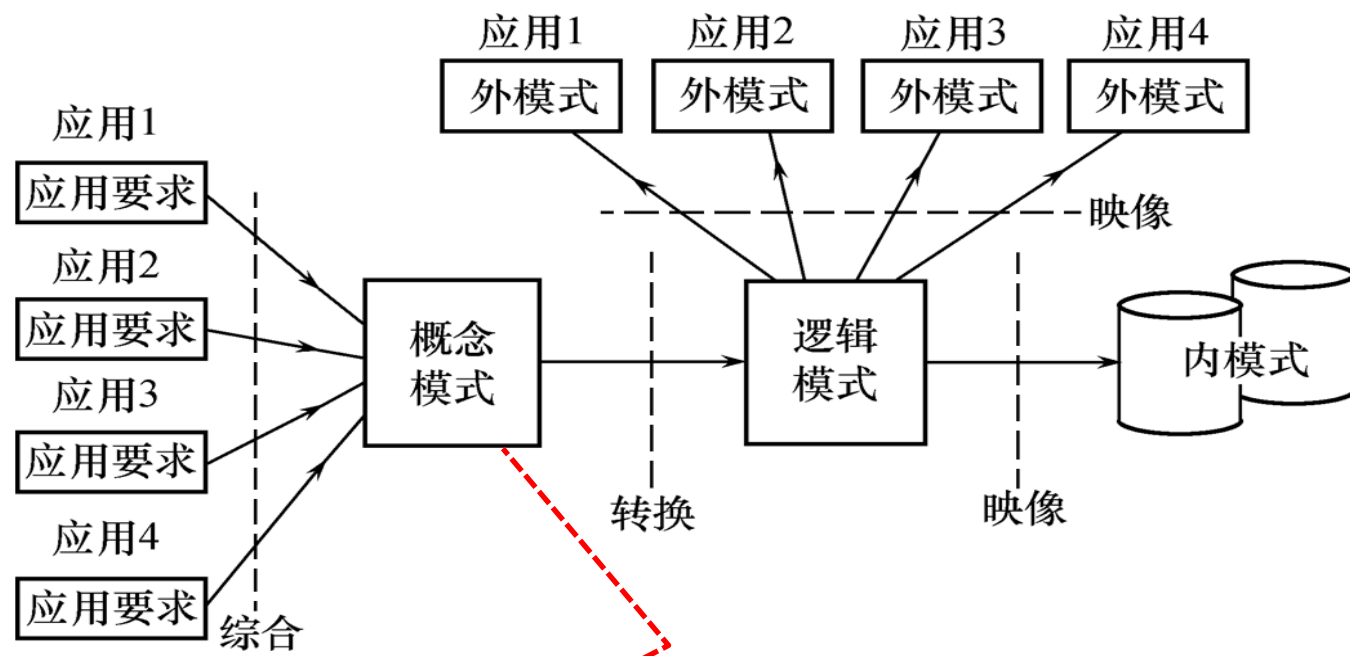
- 数据库设计不同阶段形成的数据库各级模式





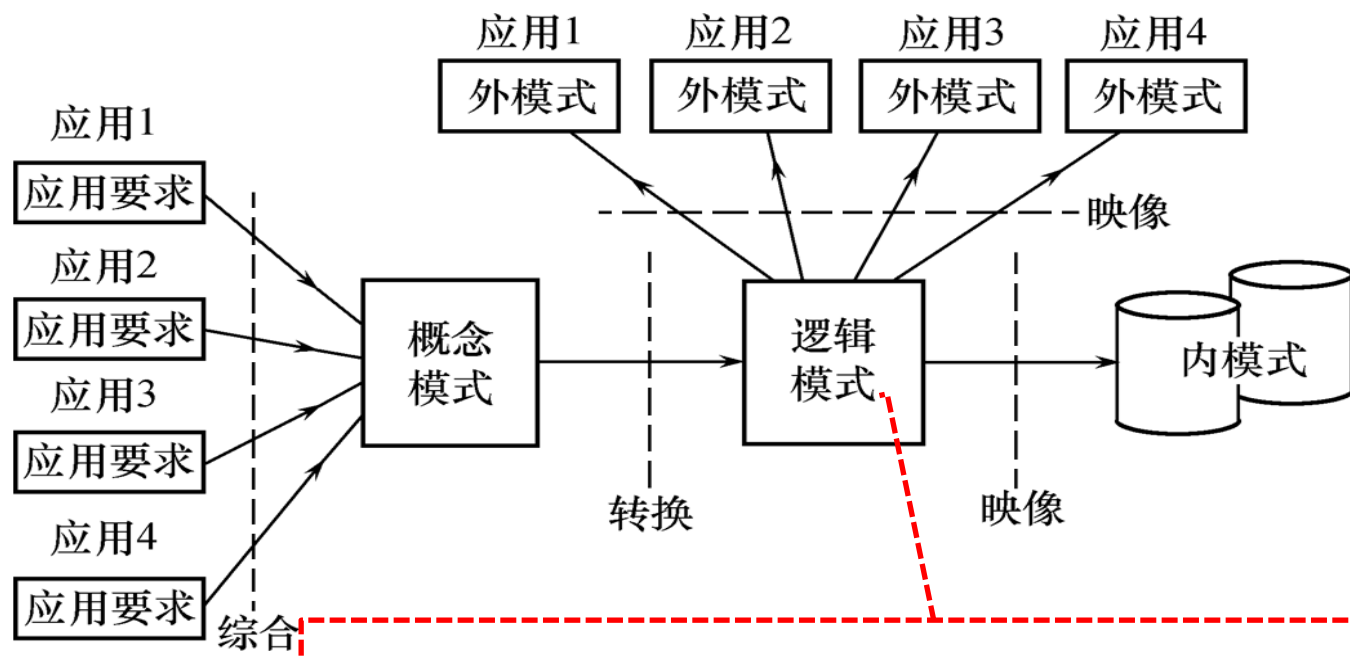
需求分析阶段:

综合各个用户的应用需求



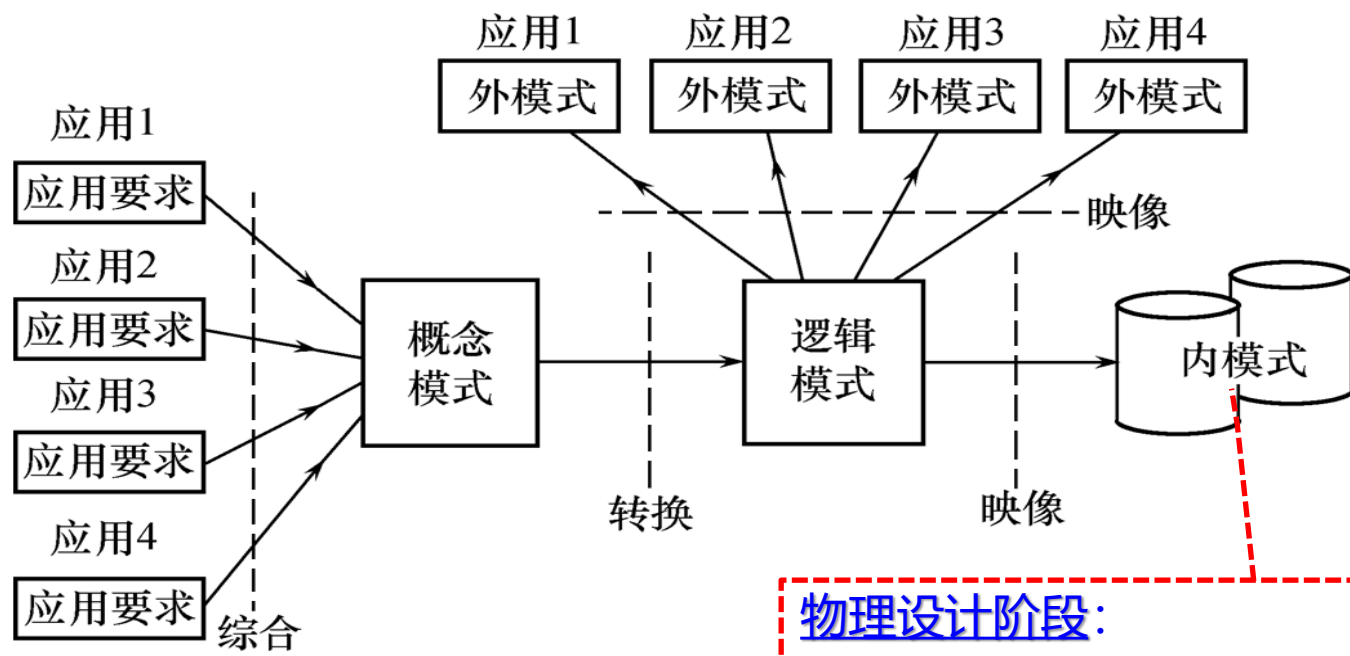
概念设计阶段:

形成独立于机器特点, 独立于各个数据库管理系统产品的概念模式(E-R图)



逻辑设计阶段:

- 首先将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



物理设计阶段:

- 根据数据库管理系统特点和处理的需要, 进行物理存储安排, 建立索引, 形成数据库内模式

大纲

- 数据库设计概述
- **需求分析**
- 概念结构设计
- 逻辑结构设计
- 物理结构设计
- 数据库的实施和维护
- 本章小结

需求分析

- 需求分析就是分析用户的要求
 - 是设计数据库的起点
 - 需求分析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
- 需求分析的方法
- 数据字典

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）工作概况。
- 明确用户的各种需求。
- 在此基础上确定新系统的功能。
- 新系统必须充分考虑今后可能的扩充和改变。
- 调查的**重点**是 “**数据**” 和 “**处理**”，获得用户对数据库的要求。
 - **信息要求**：需要存储的数据
 - **处理要求**：功能和性能
 - **安全性与完整性要求**

需求分析的方法

调查清楚用户的实际要求 ➡ 与用户达成共识 ➡ 分析与表达需求

- 调查步骤:

- ① 调查组织机构情况
- ② 调查各部门的业务活动情况
- ③ 明确新系统的各项要求: 信息要求、处理要求、安全性与完整性要求(调查重点)
- ④ 确定新系统边界: 哪些由计算机完成, 哪些由人工完成

- 调查方法:

- ① 跟班作业
- ② 开调查会
- ③ 请专人介绍
- ④ 询问
- ⑤ 设计调查表请用户填写
- ⑥ 查阅记录

- 调查完用户需要后, 还需进一步分析和表达用户的需求
- 结构化分析(Structured Analysis, SA)方法
 - 简单实用
 - 从最上层的系统组织机构入手
 - 自顶向下、逐层分解方式
- 需求分析报告必须提交给用户, 征得用户的认可

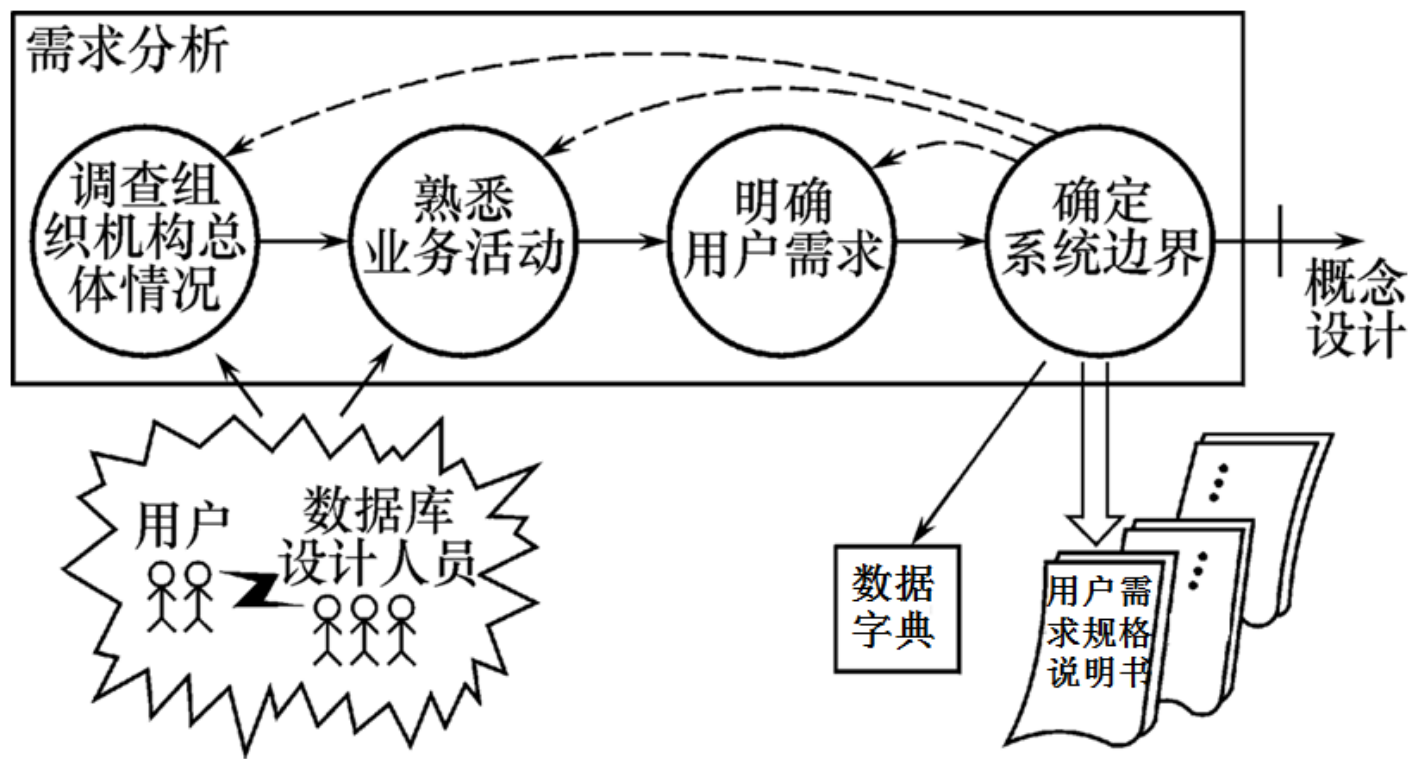


图7.5 需求分析过程

数据字典

- 数据字典是进行详细的数据收集和分析所获得的主要结果。
- 字典是**关于数据库中数据的描述**，即**元数据**，不是数据本身。
- 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。
- 在数据库设计占有很重要的地位。
- **数据字典的内容**
 1. 数据项(data item)
 2. 数据结构
 3. 数据流
 4. 数据存储
 5. 处理过程

1.数据项

- 数据项是不可再分的数据单位，是数据的最小组成单位。

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 }

- “取值范围、与其他数据项的逻辑关系” 定义了数据的完整性约束条件，是设计数据检验功能的依据。
- 数据项之间的关系可以用数据依赖的概念进行分析和表示。
 - 按实际语义写出每个数据项之间的数据依赖，这些数据依赖是数据库逻辑设计阶段数据模型优化的依据。

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。

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。

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, 最高峰期流量 }

- 数据流来源: 说明该数据流来自哪个过程
- 数据流去向: 说明该数据流将到哪个过程去
- 平均流量: 在单位时间 (每天、每周、每月等) 里的传输次数
- 高峰期流量: 在高峰时期的数据流量

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。
 - 手工文档或手工凭单，计算机文档

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}

- 存取频度: 每小时、每天或每周存取次数及每次存取的数据量
- 存取方式: 批处理 / 联机处理, 检索 / 更新, 顺序检索 / 随机检索
- 输入的数据: 数据来源
- 输出的数据: 数据去向

5.处理过程

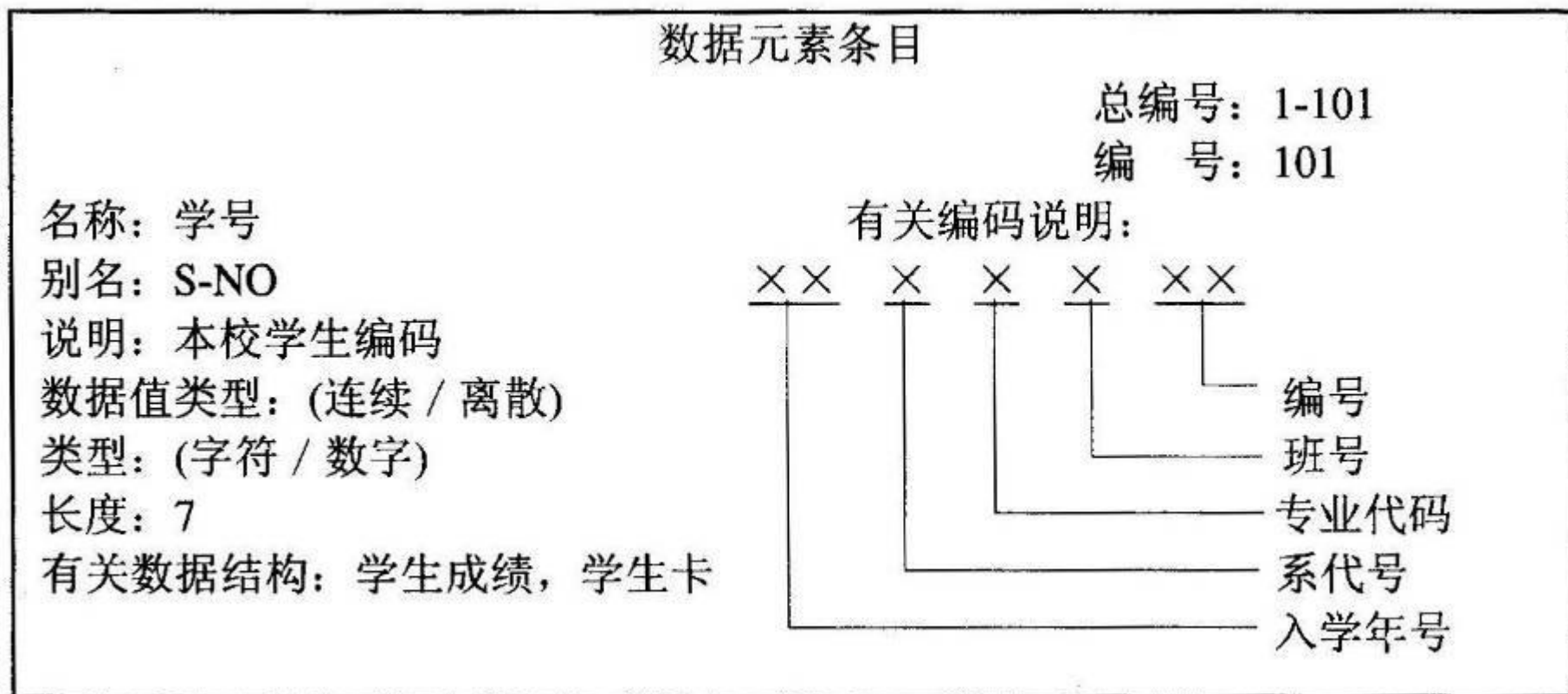
- 处理过程的具体处理逻辑一般用判定表或判定树来描述。
- 数据字典中只需要描述处理过程的说明性信息。

处理过程描述 = { 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: 说明该处理过程的功能及处理要求

- 功能: 该处理过程用来做什么 (不是怎么做)
- 处理要求: 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
- 处理要求是后面物理设计的输入及性能评价的标准

举例：学生学籍管理子系统的数据字典



数据结构条目

名称：学生登记卡
说明：新生入学时填写的卡片
结构：

学号	总编号：2-03
姓名	编号：008
[曾用名]	有关的数据流、数据存储：
入学日期	新生登记表
出生日期	学籍表
性别	数量
民族	每年约 1000 份
家庭地址	
本人简历*	
开始时间	
终止时间	
单位	
职务	

数据流条目

名称：期末成绩单
简要说明：学期结束时，由任课教师填写的成绩单
数据流来源：教师
数据流去向：P2.1、P2.2
包含的数据结构：

科目名称	总编号：3-05
{ 考试 }	编号：005
{ 考查 }	
学生成绩*	流通量：200 份 / 学期
学号	
姓名	
成绩	
任课教师	

数据存储条目

名称：学习成绩一览表
说明：学期结束，按班汇集学生各科成绩
结构：

班级	总编号：4-02
学生成绩*	编号：D2
学号	有关的数据流：
姓名	P2.1.1→D2
成绩*	D2→P2.1.2
科目名称	D2→P2.1.4
{ 考试 }	D2→P2.1.3
{ 考查 }	D2→P2.1.5
成绩	信息量：150 份 / 学期
	有无立即查询：有

处理功能条目

名称：填写成绩单
说明：通知学生成绩，有补考科目的说明补考日期
输入：D2→P2.1.5
输出：P2.1.5→学生(成绩通知单)
处理：查 D2(成绩一览表)，打印每个学生的成绩通知单，若有不及格科目，不够直接留级，则在“成绩通知”中填写补考科目、时间，若直接留级则注明留级。

总编号：5-007
编号：P2.1.4

需求分析小结

- 把需求收集和分析作为数据库设计的第一阶段是十分重要的。
- 第一阶段收集的基础数据(用数据字典来表达)是下一步进行概念设计的基础。
- **强调两点**
 - 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
 - 用户的参与是数据库设计不可分割的一部分
 - 任何调查研究没有用户的积极参与都是寸步难行的
 - 设计人员帮助用户建立数据库环境下的共同概念，对设计的最终结果承担共同责任

大纲

- 数据库设计概述
- 需求分析
- **概念结构设计**
- 逻辑结构设计
- 物理结构设计
- 数据库的实施和维护
- 本章小结

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。
 - 概念结构设计是整个数据库设计的关键
- 本节主要内容：
 1. 概念模型
 2. E-R模型
 3. 扩展的E-R模型(自学，课堂不讲)
 4. UML(自学，课堂不讲)
 5. 概念结构设计过程

1.概念模型

- 概念模型的特点:

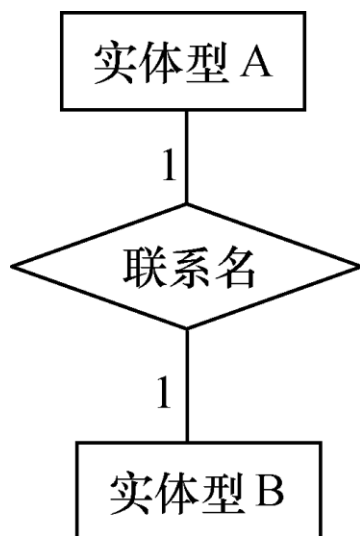
- 能真实、充分地反映现实世界，是现实世界的一个真实模型；
- 易于理解，从而可以用它和不熟悉计算机的用户交换意见；
- 易于更改，当应用环境和应用要求改变时，容易对概念模型修改和扩充；
- 易于向关系、网状、层次等各种数据模型转换。

- 描述工具

- E-R模型(Model)/图(Diagram)/方法(Approach)

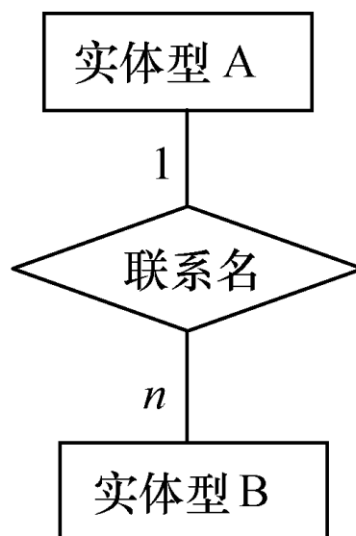
2.E-R模型

- 两个实体型之间的联系
 - 1:1联系; 1:n联系; m:n联系



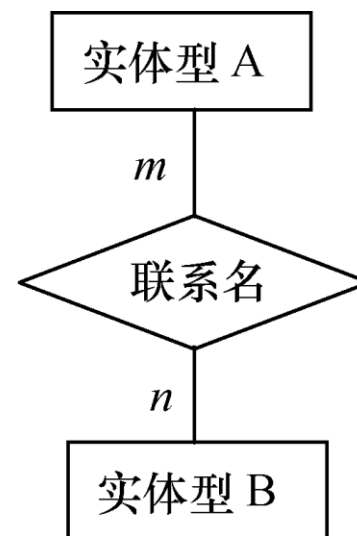
(a) 1:1 联系

例：班级与班长



(b) 1:n 联系

例：班级与学生

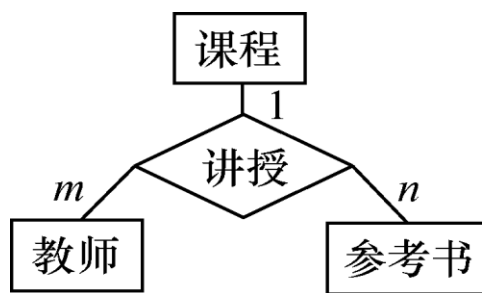


(c) m:n 联系

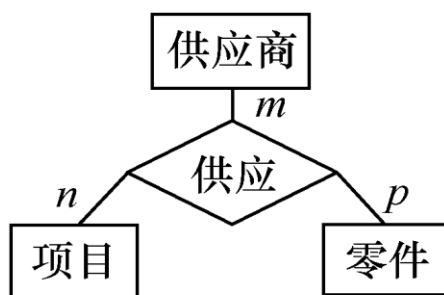
例：课程与学生

■ 两个以上的实体型之间的联系

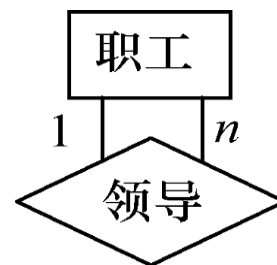
- 两个以上的实体型之间也存在着一对一、一对多、多对多联系



(a)



(b)



■ 单个实体型内的联系

- 同一个实体集内的各实体之间也可以存在一对一、一对多、多对多的联系

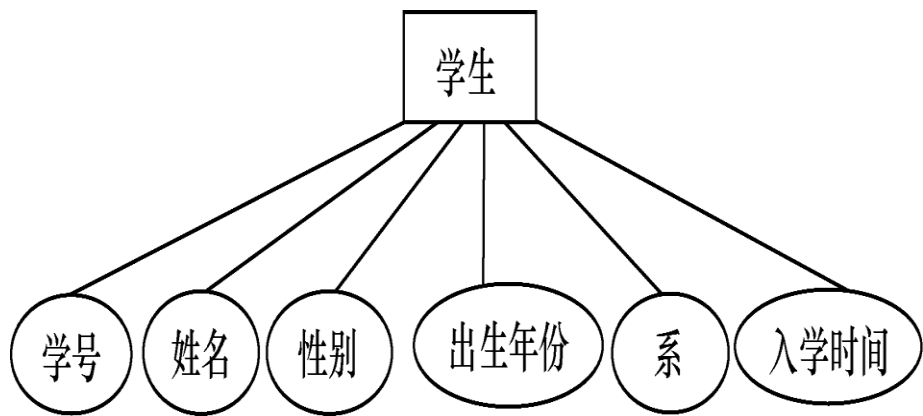
■ 联系的度

- 参与联系的实体型的数目称为联系的度
- 单元联系；二元联系；三元联系；N元联系

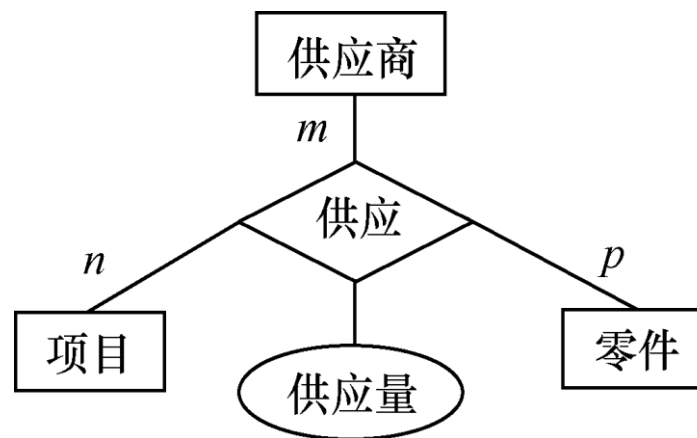
■ E-R图

– E-R图提供了表示实体型、属性和联系的方法

- 实体型用矩形表示
- 属性用椭圆表示
- 联系用菱形表示



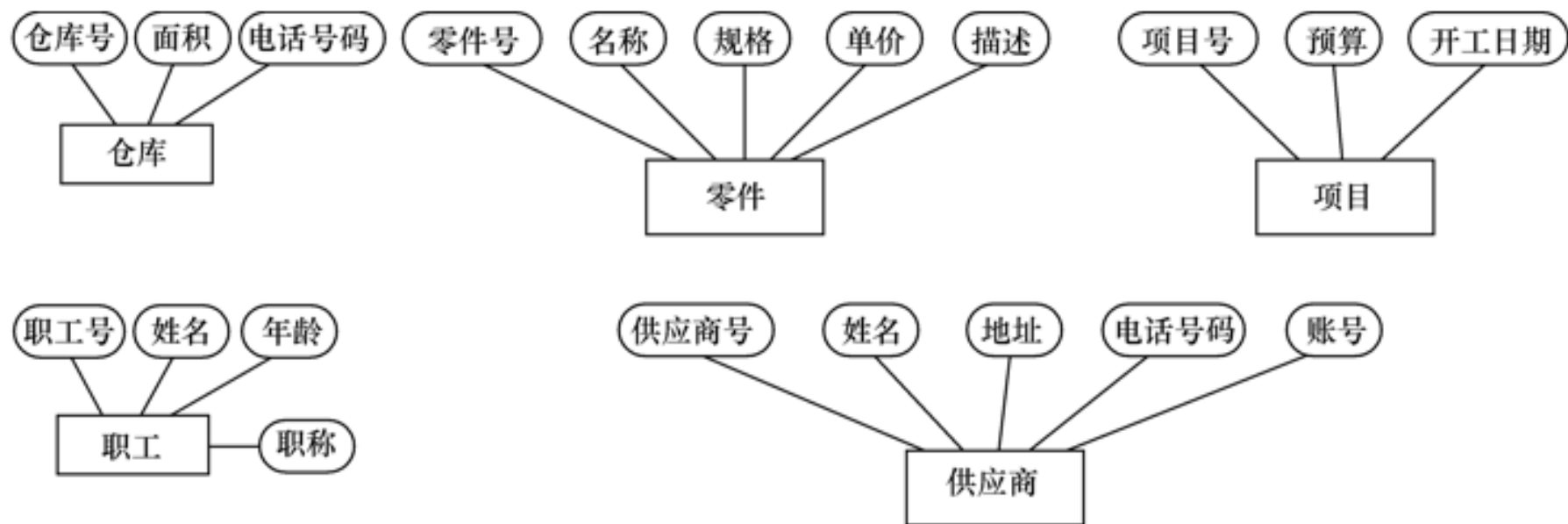
学生实体及属性



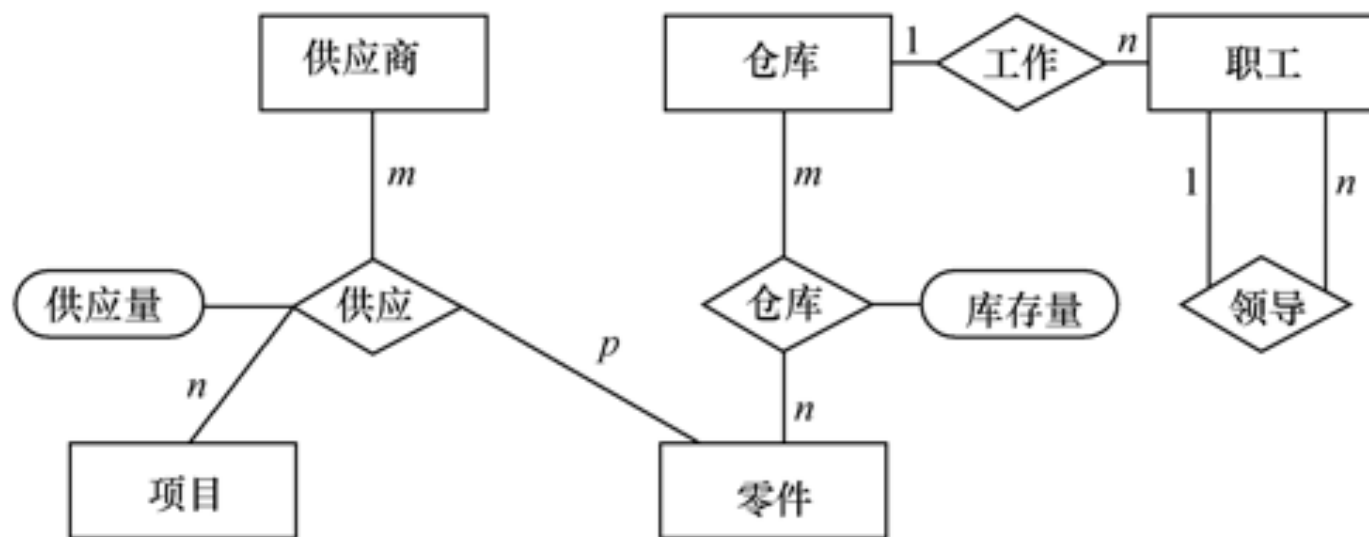
联系的属性

一个实例

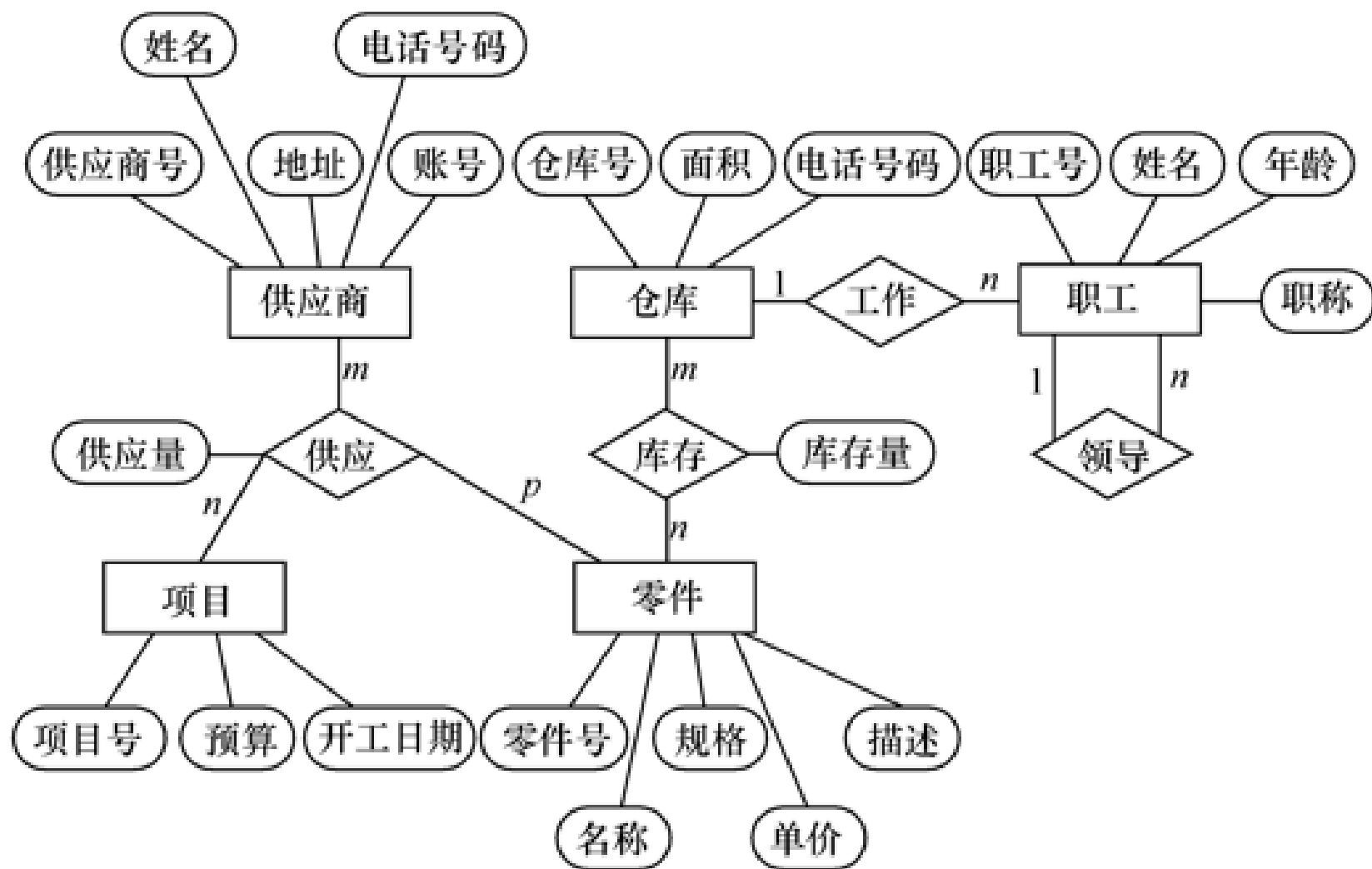
- 某个工厂物资管理的概念模型。物资管理涉及的**实体**有：
 - **仓库**：属性有仓库号、面积、电话号码
 - **零件**：属性有零件号、名称、规格、单价、描述
 - **供应商**：属性有供应商号、姓名、地址、电话号码、账号
 - **项目**：属性有项目号、预算、开工日期
 - **职工**：属性有职工号、姓名、年龄、职称
- 实体之间的**联系**如下：
 - 一个仓库可以存放多种零件，一种零件可以存放在多个仓库中，因此仓库和零件具有多对多的联系。用库存量来表示某种零件在某个仓库中的数量；
 - 一个仓库有多个职工当仓库保管员，一个职工只能在一个仓库工作，因此仓库和职工之间是一对多的联系；
 - 职工之间具有领导与被领导关系。即仓库主任领导若干保管员，因此职工实体型中具有一对多的联系；
 - 供应商、项目和零件三者之间具有多对多的联系。即一个供应商可以供给若干项目多种零件，每个项目可以使用不同供应商供应的零件，每种零件可由不同供应商供给。



(a) 实体及其属性图



(b) 实体及其联系图



(c) 完整的实体-联系图

课堂练习

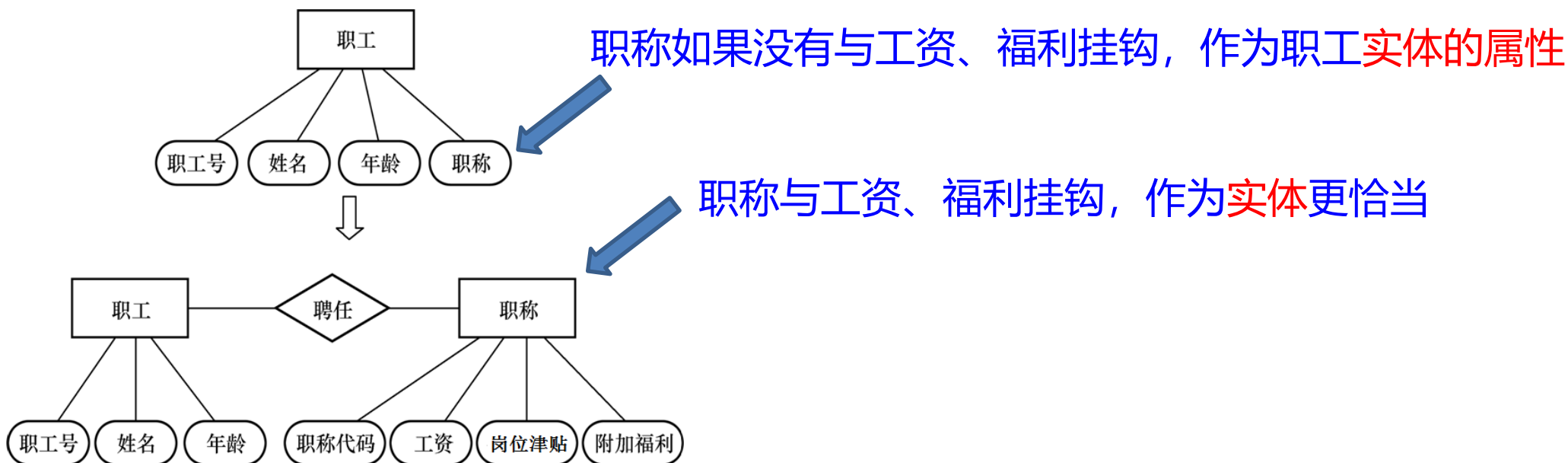
- 某商场可以为顾客办理会员卡，每个顾客只能办理一张会员卡，顾客信息包括顾客姓名、地址、固定电话、身份证号，会员卡信息包括号码、等级、积分，给出该系统的E-R图。
 - 说明，设计E-R图时应尽可能贴近实际应用，完善相应的实体、属性或联系

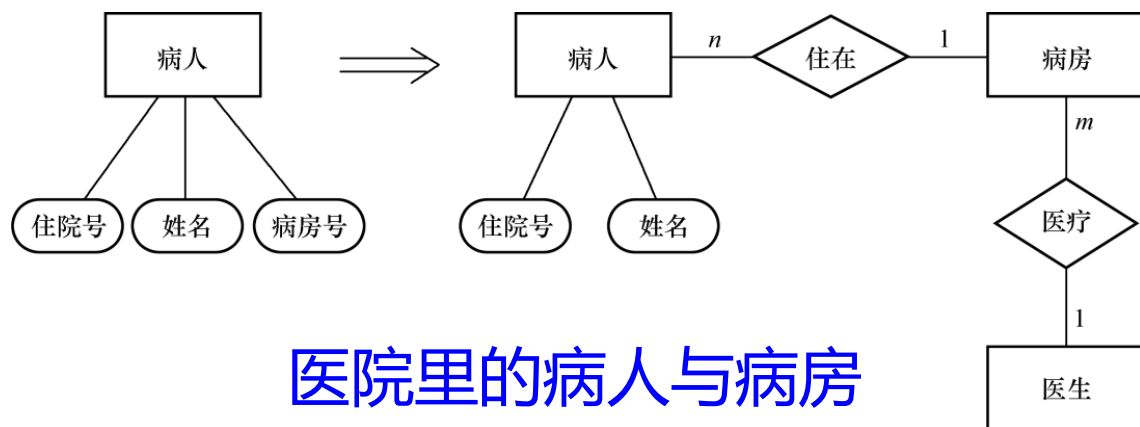
5.概念结构设计过程

- 本节目标：
 - 掌握在设计E-R图过程中如何确定实体与属性
 - 掌握在集成E-R图时如何解决冲突等关键技术
- 概念结构设计的第一步就是对需求分析阶段收集到的数据进行分类、组织，确定实体、实体属性、实体之间的联系类型，形成E-R图。
- 事实上，在现实世界中具体的应用环境已对实体和属性作了自然的大体划分。
 - 数据字典、数据结构、数据流和数据存储
- 为简化E-R图的处置，现实世界的事物能作为属性对待的，尽量作为属性对待。

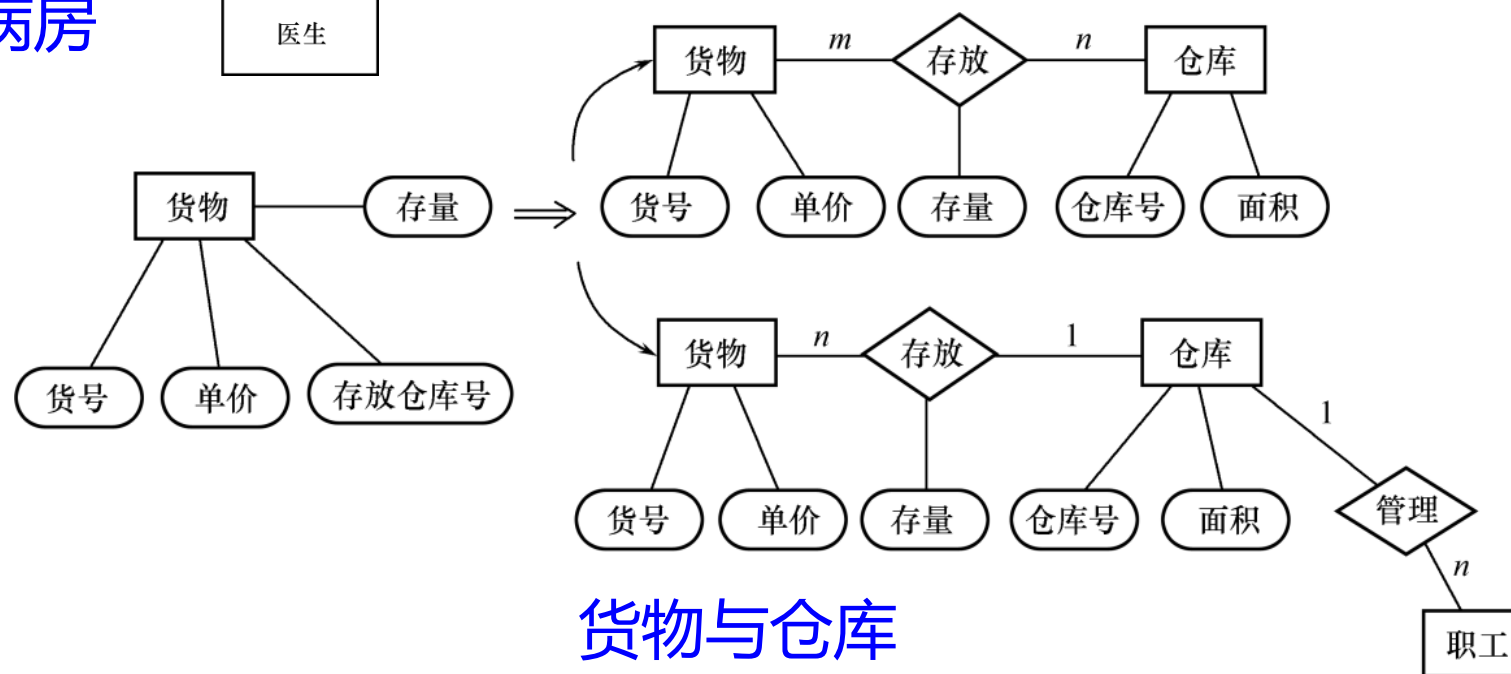
■ 实体与属性的划分原则(两条准则)

- 作为属性，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，不能包含其他属性
- 属性不能与其他实体具有联系
 - 即E-R图中所表示的联系是实体之间的联系



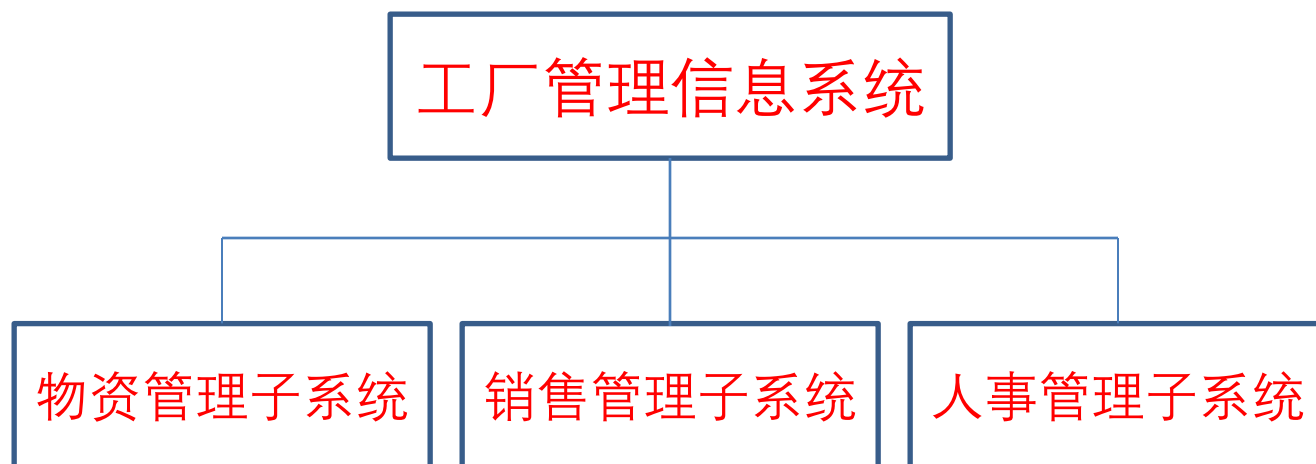


医院里的病人与病房



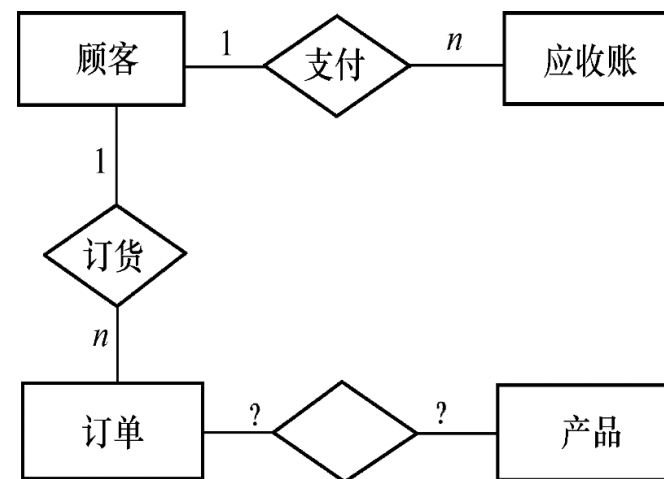
货物与仓库

■ [例7.1] 销售管理子系统E-R图的设计



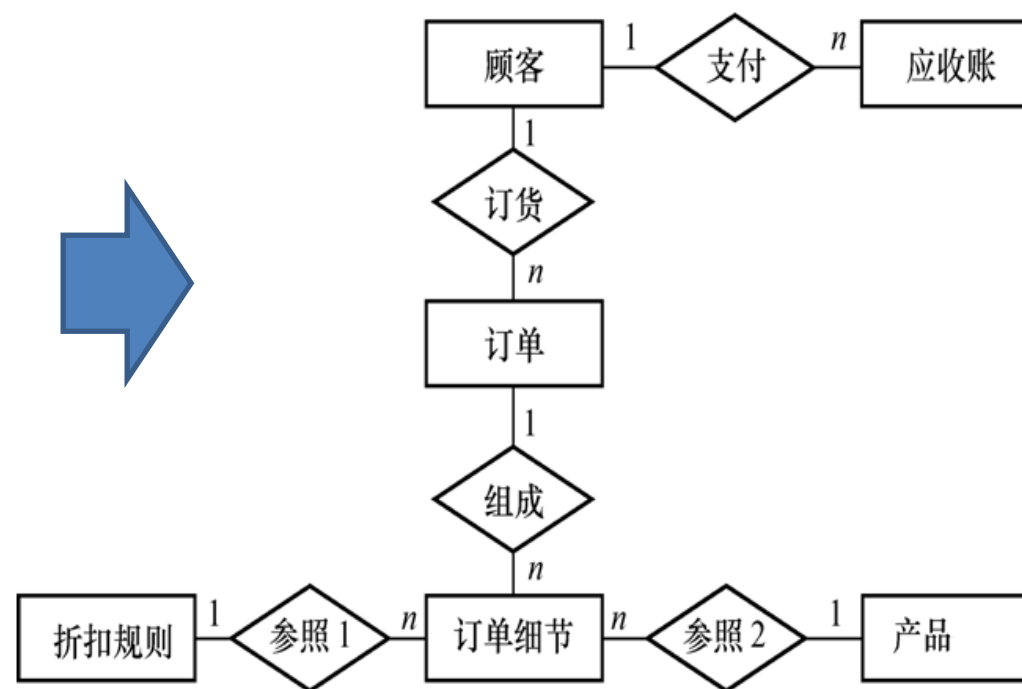
■ 经分析，该子系统的主要功能是：

- 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



■ 参照需求分析和数据字典中的详尽描述，遵循两个准则，进行如下调整：

- 每张订单由订单号、若干头信息和订单细节组成。
订单细节又有订货的零件号、数量等来描述
- 原订单和产品的联系实际上是订单细节和产品的联系。每条订货细节对应一个产品描述，订单处理时从中获得当前单价、产品重量等信息
- 工厂对大宗订货给予优惠。每种产品都规定了不同订货数量的折扣，应增加一个“折扣规则”实体存放这些信息，而不应把它们放在产品实体中



■ 对每个实体定义的属性如下：

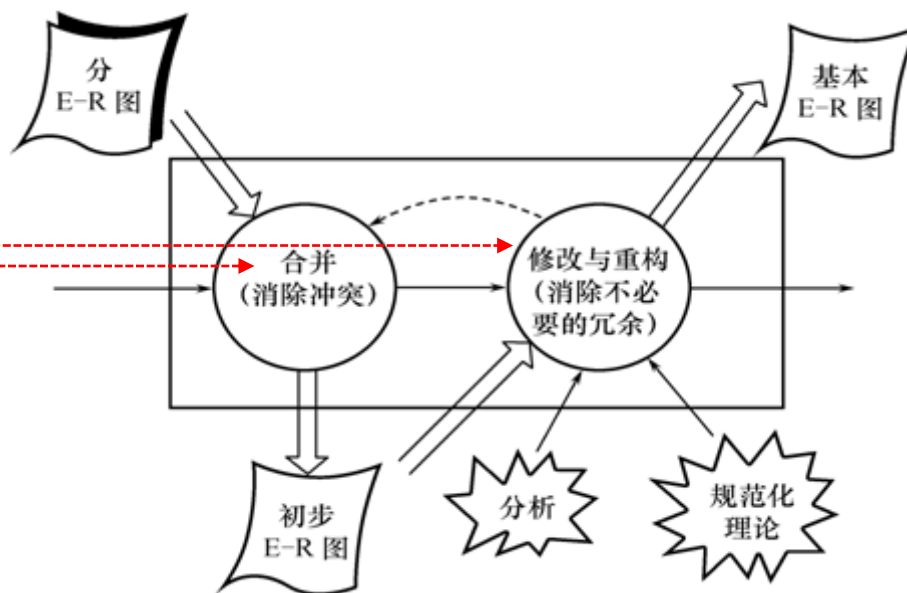
- 顾客：{顾客号，顾客名，地址，电话，信贷状况，账目余额}
- 订单：{订单号，顾客号，订货项数，订货日期，交货日期，工种号，生产地点}
- 订单细则：{订单号，细则号，零件号，订货数，金额}
- 应收账款：{顾客号，订单号，发票号，应收金额，支付日期，支付金额，当前余额，贷款限额}
- 产品：{产品号，产品名，单价，重量}
- 折扣规则：{产品号，订货量，折扣}

■ E-R图的集成

- 在开发一个大型信息系统时，最经常采用的策略
 - 自顶向下进行需求分析；
 - 再自底向上设计概念结构
- 即：设计各子系统的分E-R图 \Rightarrow 集成分E-R图 \Rightarrow 得到全局E-R图

■ E-R图的集成步骤

- ① 合并
- ② 修改和重构



1. 合并E-R图，生成初步E-R图

- 冲突：各分E-R图之间存在的 inconsistent 的地方

■ 三类冲突：

– 属性冲突：

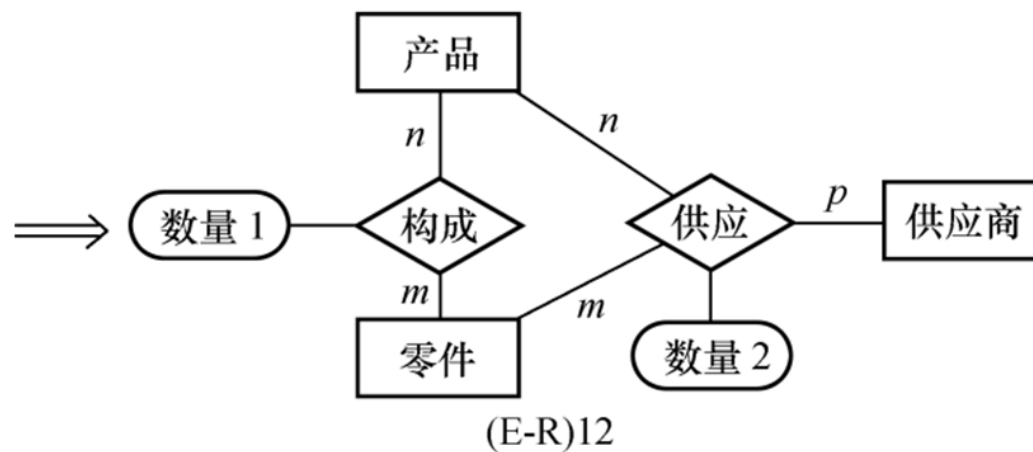
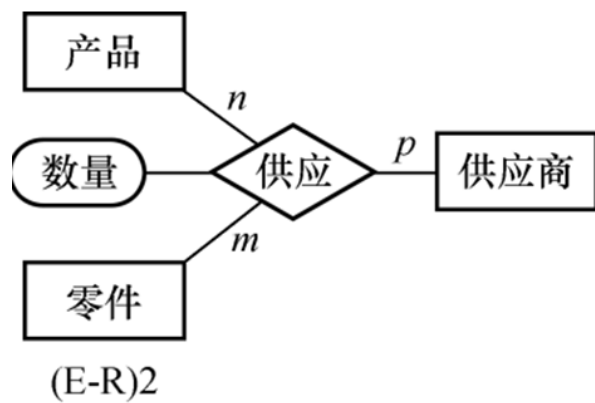
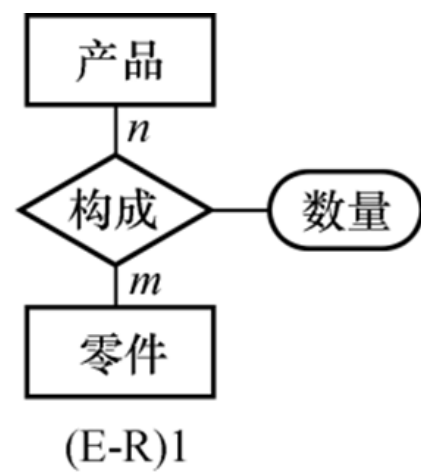
- 属性域冲突，即属性值的类型、取值范围或取值集合不同。如，零件号：字符型或整数
- 属性取值单位冲突，如，重量：公斤或磅或斤或吨或克

– 命名冲突：

- 同名异义，即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)，即同一意义的对象在不同的局部应用中具有不同的名字，如，科研项目/课题，
- 可发生在实体、属性或联系上，通过讨论、协商等行政手段加以解决

– 结构冲突：

- 同一对象在不同应用中具有不同的抽象。如，职工在一个局部应用中是实体，在另外一个应用中是属性
- 同一实体在不同子系统的E-R图所包含的属性个数和属性排列次序不完全相同。⇒合并属性，调整次序
- 实体间的联系在不同的E-R图中为不同的类型。⇒根据语义对联系的类型进行综合或调整



2. 消除不必要的冗余，设计基本E-R图

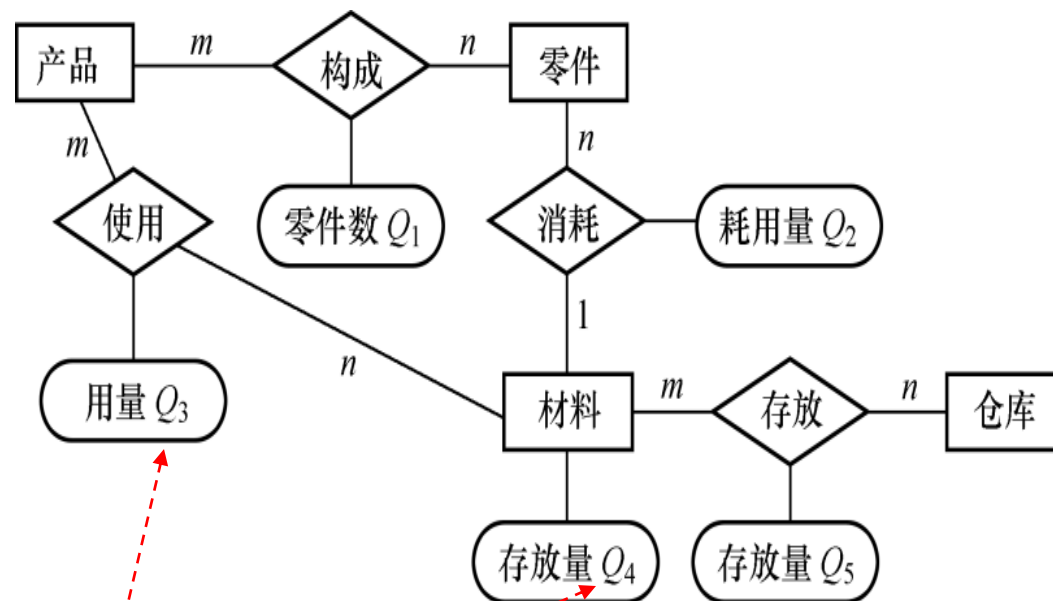
— 冗余的数据

- 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，应当予以消除。消除了冗余后的初步E-R图称为基本E-R图。
- 消除冗余的主要方法：分析法
 - 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逻辑关系的说明来消除冗余



$Q_3 = Q_1 \times Q_2$, $Q_4 = \sum Q_5 \Rightarrow Q_3$, Q_4 是冗余数据，可以消去，产品与材料间的 $m:n$ 联系也应消去

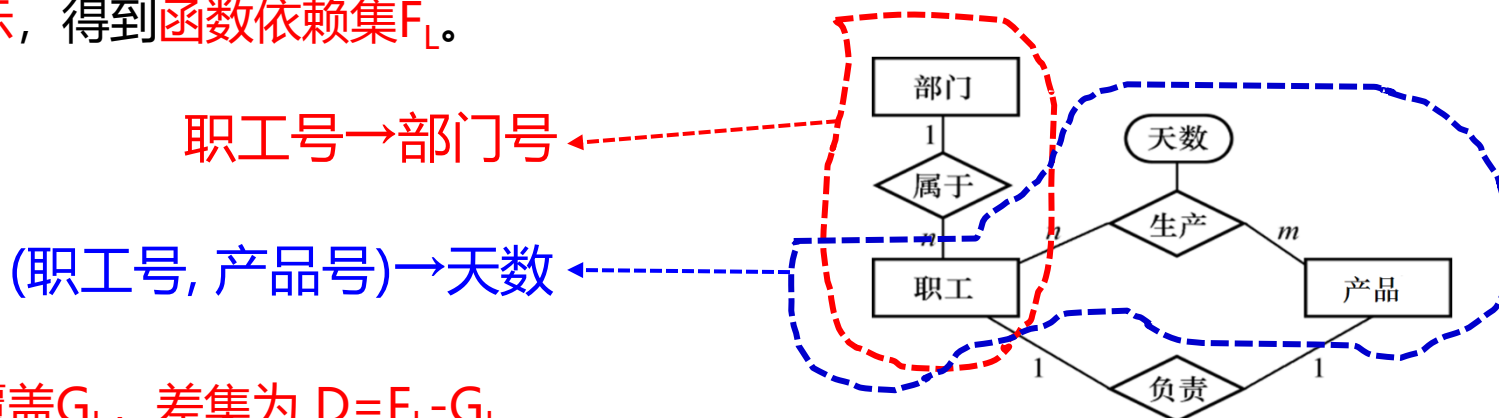
并不是所有的冗余数据与冗余联系都必须加以消除，有时为了提高效率，不得不以冗余信息作为代价

■ 消除冗余方法之二：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

– 具体步骤：

- ① 确定分E-R图实体间的数据依赖，实体之间一对一、一对多、多对多的联系用实体码之间的函数依赖来表示，得到函数依赖集 F_L 。



- ② 求 F_L 的最小覆盖 G_L ，差集为 $D=F_L-G_L$

- ③ 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。

• 使用规范化理论消除冗余时应注意两个问题：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。

■ [例7.2] 某工厂管理信息系统的视图集成

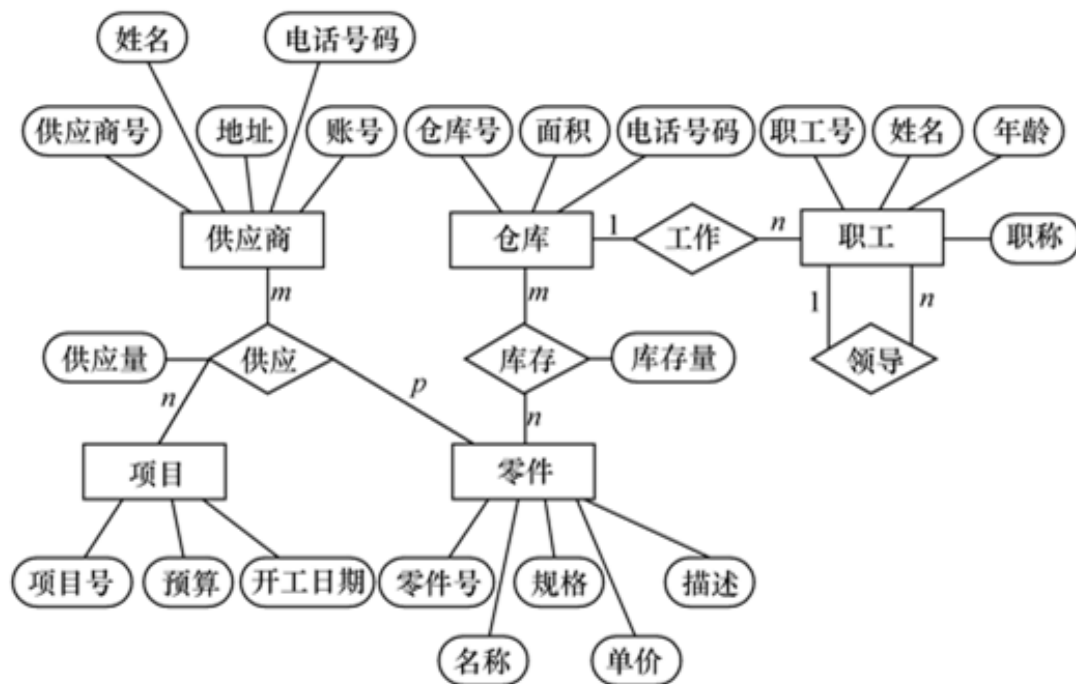


图7.11 工厂物资管理E-R图

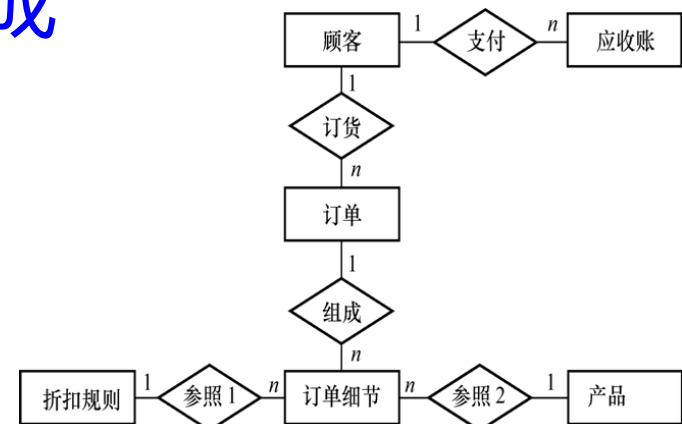


图7.23 销售管理子系统的E-R图

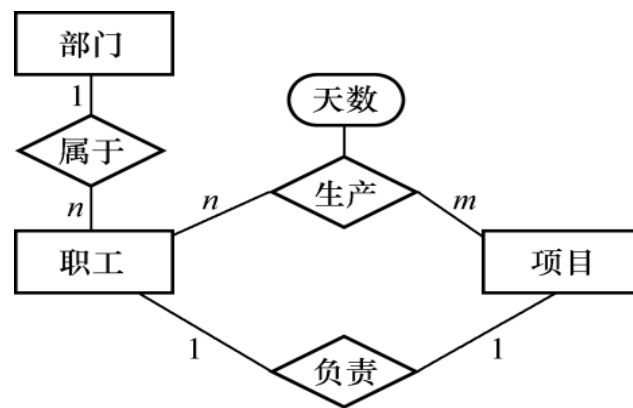
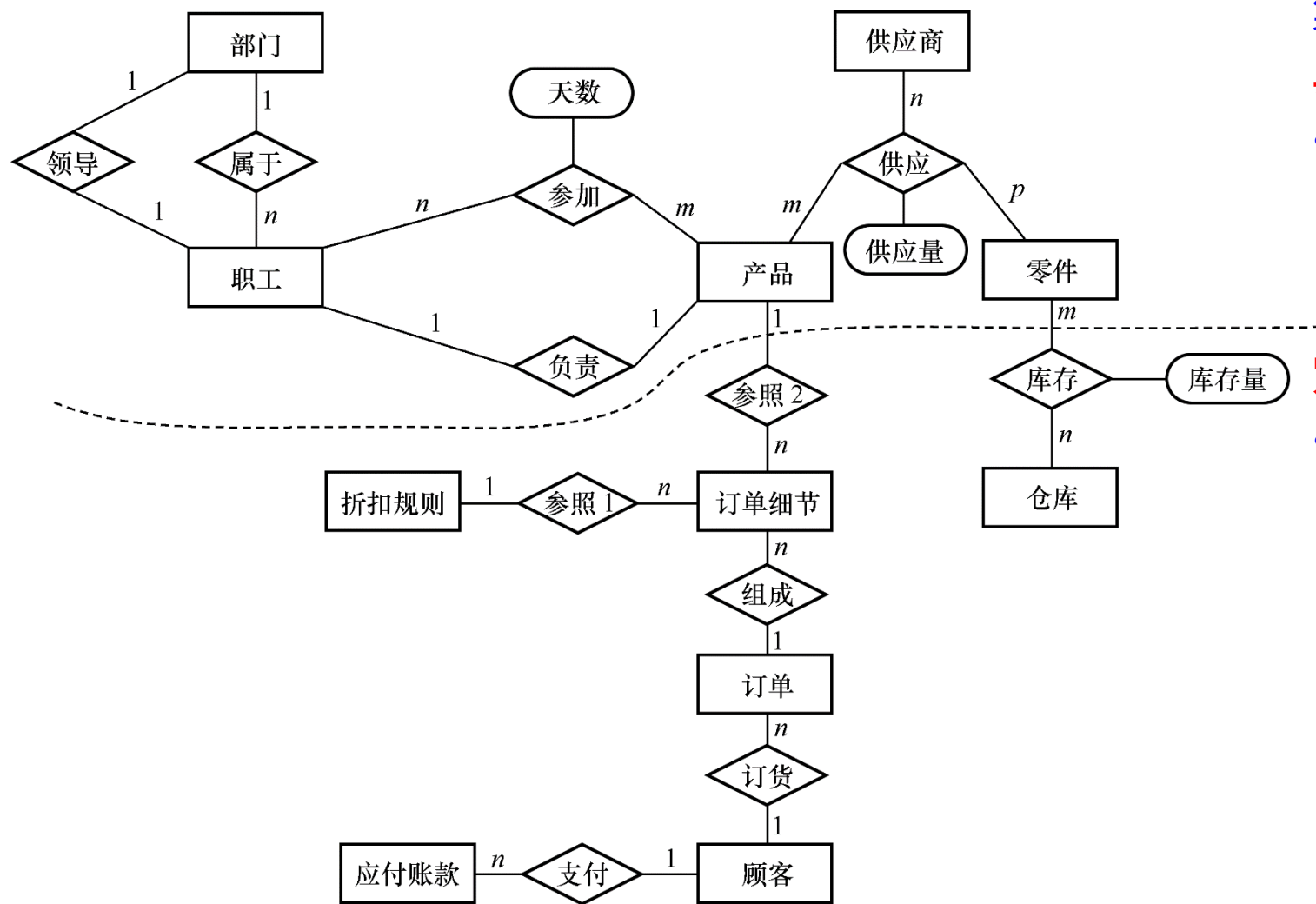


图7.27 劳动人事管理的分E-R图



集成过程中解决了如下问题:

异名同义:

- 项目和产品含义相同, 某个项目实质上是指某个产品的生产

冗余的联系:

- 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, 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, 也可以取消。

大纲

- 数据库设计概述
- 需求分析
- 概念结构设计
- **逻辑结构设计**
- 物理结构设计
- 数据库的实施和维护
- 本章小结

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图转换为与选用的DBMS产品所支持的数据模型相符合的逻辑结构
- 主要介绍E-R图向关系数据模型的转换

■ 本节主要内容

1. E-R图向关系模型的转换
2. 数据模型的优化
3. 设计用户子模式

1. E-R图向关系模型的转换

■ 任务：

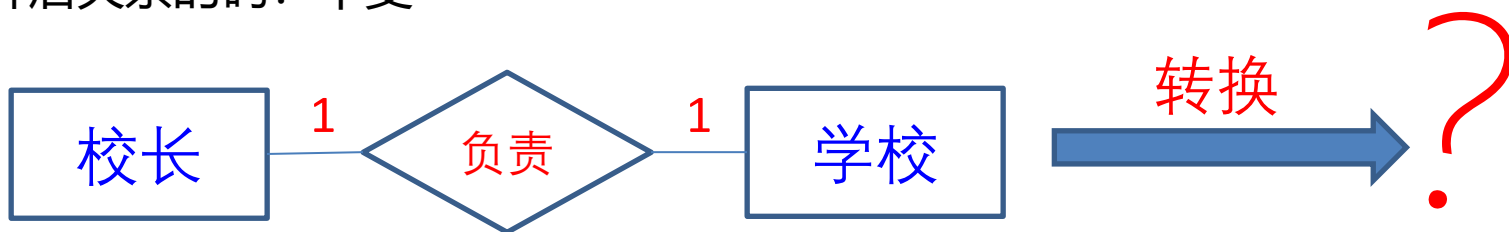
- 如何将实体型和实体间的联系转换为关系模式
- 如何确定这些关系模式的属性和码

■ 转换原则：

- 一个实体型转换为一个关系模式
 - 关系的属性：实体的属性
 - 关系的码：实体的码
- 实体间联系的转换根据基数约束不同而采用不同的策略

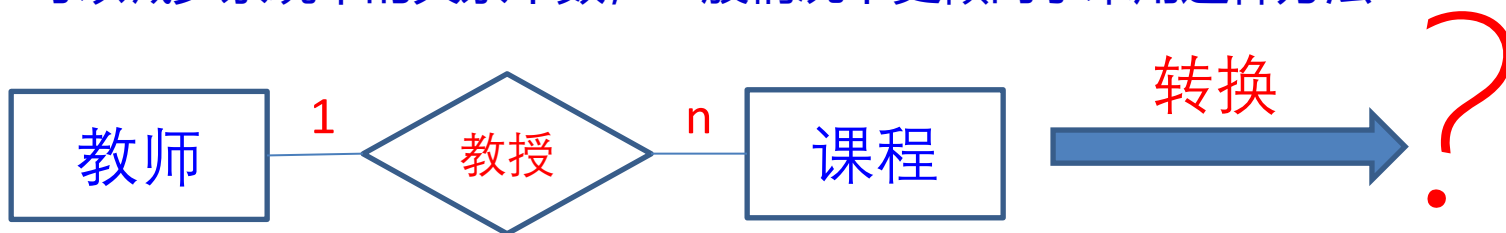
■ 1:1联系

- 可将联系转换为一个**独立的关系模式**，或与任意一端对应的**关系模式合并**
- **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
- **合并的关系模式**
 - 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



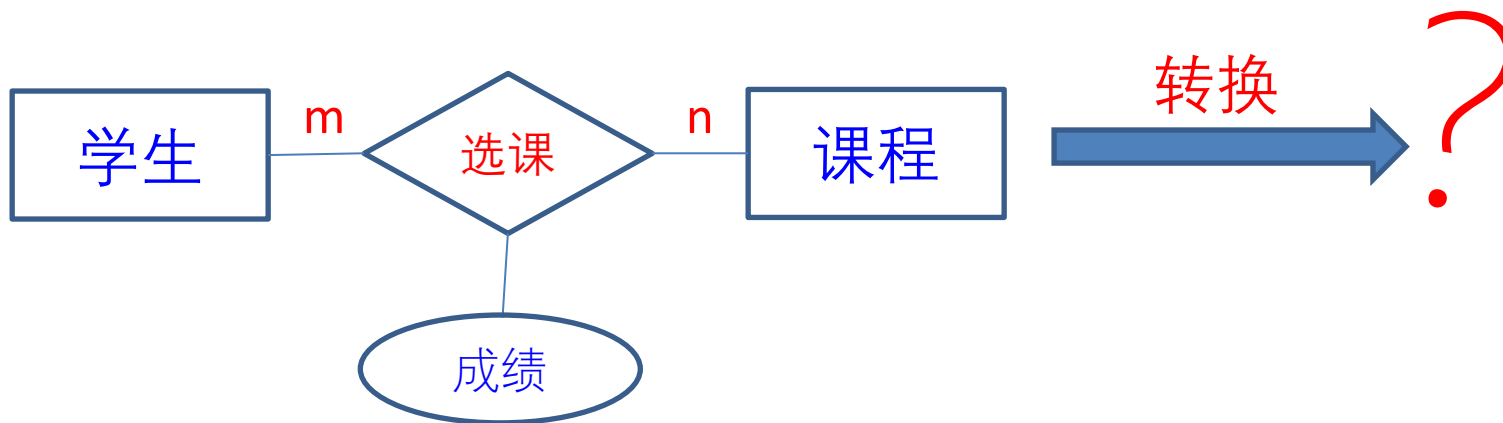
■ 1:N联系

- 可将联系转换为一个独立的关系模式，或与n端对应的关系模式合并
- 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
- 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
 - 注：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



■ m:n联系

- 只能将联系转换为一个**独立的关系模式**
- **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：**各实体码的组合**

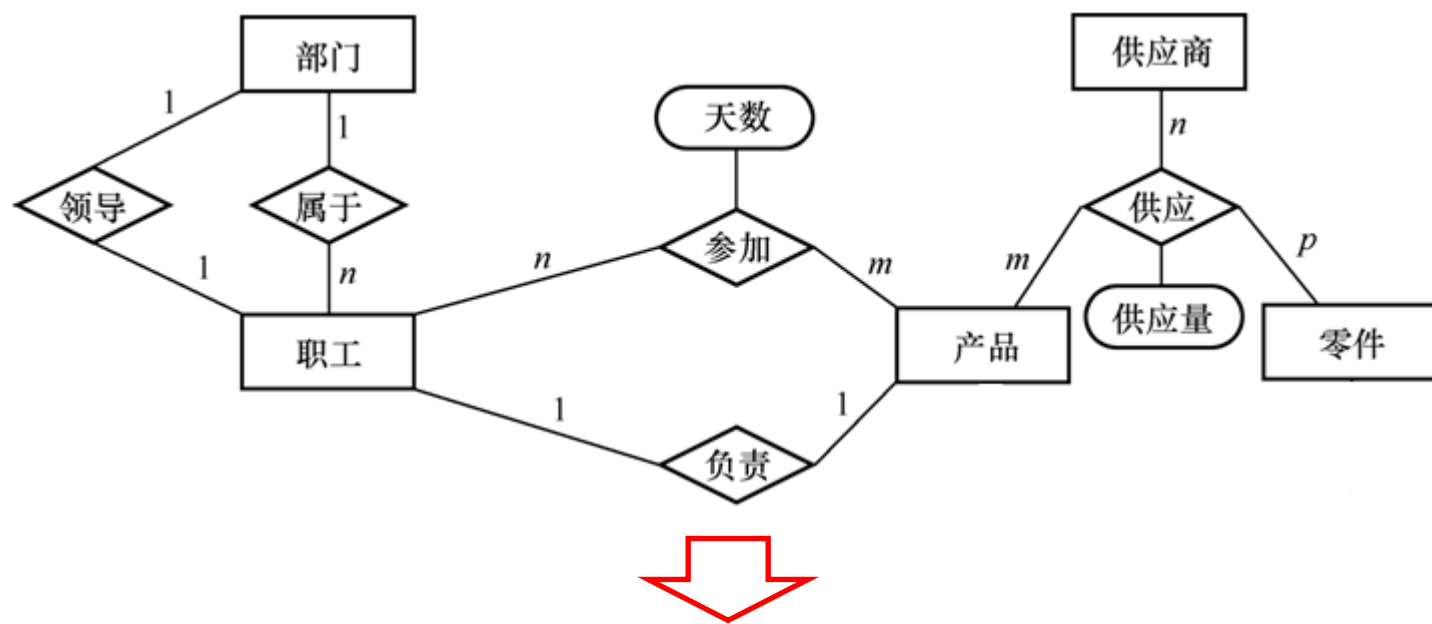


■ 三个或三个以上实体间的一个多元联系

- 只能将联系转换为一个独立的关系模式
- 独立的关系模式
 - 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

■ 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
 - 去掉其中的同义属性（可能同名也可能不同名）
 - 适当调整属性的次序



- 部门(部门号, 部门名, 经理的职工号, ...)
- 职工(职工号, 部门号, 职工名, 职务, ...)
- 产品(产品号, 产品名, 产品组长的职工号, ...)
- 供应商(供应商号, 姓名, ...)
- 零件(零件号, 零件名, ...)
- 职工工作(职工号, 产品号, 工作天数, ...)
- 供应(产品号, 供应商号, 零件号, 供应量)

2.数据模型的优化

- 数据库逻辑设计的结果不是唯一的。
 - 问题：如何确保得到的数据模型满足数据库应用系统性能要求？
- 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。

■ 数据模型的优化过程：

- ① 确定数据依赖
- ② 对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 按照数据依赖的理论对关系模式进行分析，考察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 根据需要分析阶段得到的处理要求分析对于这样的应用环境这些模式是否合适，确定是否要对某些模式进行那个合并或分解
- ⑤ 对关系模式进行必要分解，提高数据操作效率和存储空间的利用率
 - 水平分解 遵循 “80/20原则”
 - 垂直分解 确保 “无损连接性和保持函数依赖”

3.设计用户子模式

- 将概念模式转换为全局逻辑模型之后，还应该根据局部应用需求，结合具体RDBMS的特点设计用户的外模式。
- 视图被用于实现设计用户子模式。
- 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，而定义用户外模式时应该更注重考虑用户的习惯与方便。
 - 使用更符合用户习惯的别名
 - 定义视图时重新定义某些属性名，使其与用户习惯一致，以方便使用
 - 针对不同级别的用户定义不同的视图，以保证系统的安全性
 - 如，顾客视图只包含允许顾客查询的属性；销售视图只包含允许销售部门查询的属性
 - 简化用户对系统的使用
 - 可以将某些局部应用中要经常使用的复杂查询定义为视图

大纲

- 数据库设计概述
- 需求分析
- 概念结构设计
- 逻辑结构设计
- **物理结构设计**
- 数据库的实施和维护
- 本章小结

物理结构设计

- 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要求的物理结构的过程，就是数据库的物理设计。
- 数据库的物理设计步骤：
 1. 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
 2. 对物理结构进行评价
 - 评价的重点是时间和空间效率
 3. 若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

物理结构设计(cont'd)

- 本节主要内容:
 - 数据库物理设计的内容和方法
 - 关系模式存取方法选择
 - 确定数据库的存储结构
 - 评价物理结构

1.数据库物理设计的内容和方法

- **没有通用**的物理设计方法，只有一般的设计内容和原则
 - 不同的数据库产品所提供的**物理环境、存取方法和存取结构**有很大的差别
 - 能供设计人员使用的**设计变量、参数范围**也很不相同
- **物理设计的总目标**:
 - 在数据库上运行的各种事务**响应时间小、存储空间利用率高、事务吞吐量率大**
- **物理设计的准备工作**
 - 充分了解**应用环境**，详细分析要**运行的事务**，以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
- **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

- 对于数据库查询事务需要知道

- 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

- 对于数据库更新事务需要知道

- 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，如，事务T必须在10s内结束

这些信息是确定关系的存取方法的依据

2.关系模式存取方法选择

- 数据库系统是多用户共享的系统，对同一个关系要建立多条存取路径才能满足多用户的多种应用要求。
- 物理结构设计任务之一是根据RDBMS支持的存取方法确定选择哪些存取方法。
- 数据库管理系统常用存取方法
 - B+树索引存取方法
 - Hash索引存取方法
 - 聚簇存取方法

■ B+树索引存取方法的选择

– 索引存取方法的选择

- 根据应用要求确定对关系的哪些属性列建立索引、哪些属性列建立组合索引、哪些索引要设计为唯一索引

– 选择的一般规则

- 如果一个(或一组)属性经常在查询条件中出现, 则考虑在这个(或这组)属性上建立索引(或组合索引)
- 如果一个属性经常作为最大值和最小值等聚集函数的参数, 则考虑在这个属性上建立索引
- 如果一个(或一组)属性经常在连接操作的连接条件中出现, 则考虑在这个(或这组)属性上建立索引

■ 关系上定义的索引数过多会带来较多的额外开销

– 维护索引的开销

– 查找索引的开销

■ Hash索引存取方法的选择

– 选择Hash存取方法的规则

- 如果一个关系的属性主要出现在等值连接条件中或主要出现在等值比较选择条件中，而且满足下列两个条件之一：
- 该关系的大小可预知，而且不变；
- 该关系的大小动态改变，但所选用的数据库管理系统提供了动态Hash存取方法

■ 聚簇存取方法的选择

– 聚簇与聚簇码

- 为了提高某个属性(或属性组)的查询速度，把这个或这些属性上具有相同值的元组集中存放在连续的物理块中称为聚簇
- 该属性(或属性组)称为聚簇码(cluster key)

- 聚簇功能可以大大提高按聚簇码进行查询的效率

[例] 假设学生关系按所在系建有索引，现在要查询信息系的所有学生名单。

- 计算机系的500名学生分布在500个不同的物理块上时，至少要执行500次I/O操作
- 如果将同一系的学生元组集中存放，则每读一个物理块可得到多个满足查询条件的元组，从而显著地减少了访问磁盘的次数

- 聚簇功能不但适用于单个关系，也适用于经常进行连接操作的多个关系

- 即把多个连接关系的元组按连接属性值聚集存放，相等于把多个关系按“预连接”的形式存放，大大提高连接操作的效率

- 一个数据库可以建立多个聚簇，一个关系只能加入一个聚簇

- 选择聚簇存取方法就是要确定需要建立多少个聚簇，每个聚簇包括哪些关系

- 具体选择步骤：

1. 设计候选聚簇

- 对经常在一起进行连接操作的关系可以建立聚簇
- 如果一个关系的一组属性经常出现在相等比较条件中，则该单个关系可建立聚簇
- 如果一个关系的一个(或一组)属性上的值重复率很高，则此单个关系可建立聚簇

2. 检查候选聚簇中的关系，取消其中不必要的关系

- 从聚簇中删除经常进行全表扫描的关系
- 从聚簇中删除更新操作远多于连接操作的关系
- 从聚簇中删除重复出现的关系：不同的聚簇中可能包含相同的关系，一个关系可以在某一个聚簇中，但不能同时加入多个聚簇

– 要从这多个聚簇方案(包括不建立聚簇)中选择一个运行各种事务的总代价最小的

■ 聚簇使用的特点：

- 聚簇只能提高某些特定应用的性能
- 建立与维护聚簇的开销相当大
 - 对已有关系建立聚簇，将导致关系中元组的物理存储位置移动，并使此关系上原有的索引无效，必须重建
 - 当一个元组的聚簇码改变时，该元组的存储位置也要做相应改变
 - 所以，聚簇码值要相对稳定，以减少修改聚簇码值引起的维护开销
- 当通过聚簇码进行访问或连接是该关系的主要应用，与聚簇码无关的其他访问很少或是次要的，这是可以使用聚簇
 - 尤其当SQL语句中包含有与聚簇码有关的子句或短语时：ORDER BY、GROUP BY、UNION、DISTINCT

3.确定数据库的存储结构

- 确定数据库物理结构主要指确定数据的存放位置和存储结构。
 - 包括确定关系、索引、聚簇、日志、备份等的存储安排和存储结构，确定系统配置等
- 确定数据的存放位置和存储结构要综合考虑存取时间、存储空间利用率和维护代价3个方面的因素，权衡择优。

1. 确定数据的存放位置

- 根据应用情况将数据的易变部分与稳定部分、经常存取部分和存取频率较低部分分开存放
 - 可以将比较大的表分别放在两个磁盘上，以加快存取速度，这在多用户环境下特别有效
 - 可以将日志文件与数据库对象(表、索引等)放在不同的磁盘以改进系统的性能
- 应仔细了解给定的RDBMS提供的方法和参数，针对应用环境的要求对数据进行适当的物理安排。

2. 确定系统配置

- RDBMS一般都提供了一些系统配置变量和存储分配参数，供设计人员和DBA对数据库进行物理优化。
 - 给出了默认值，但这些值不一定适合每一种应用环境
 - 需要根据应用环境重新调整默认值，以改善系统的性能
- 常见的系统配置变量
 - 同时使用数据库的用户数、同时打开的数据库对象数、内存分配参数
 - 缓冲区分配参数（使用的缓冲区长度、个数）、存储分配参数、物理块的大小
 - 物理块装填因子
 - 时间片大小
 - 数据库的大小
 - 锁的数目等
- 配置应根据系统后续实际运行情况做进一步的调整，以切实改进系统性能

3. 评价物理结构

- 设计过程中需要对时空效率、维护代价和各种用户要求进行权衡，结果可以产生多种方案。
- 数据库设计人员必须定量估算各种方案的存储空间、存取时间和维护代价，从中选择一个较优的、合理的物理结构
- 如果物理结构不符合用户需求，则需要修改设计
- 特别注意：物理结构的评价方法完全依赖于所选用的RDBMS

大纲

- 数据库设计概述
- 需求分析
- 概念结构设计
- 逻辑结构设计
- 物理结构设计
- **数据库的实施和维护**
- 本章小结

数据库的实施和维护

- 完成数据库的物理设计之后，设计人员就要用RDBMS提供的
数据定义语言和其他实用程序将数据库逻辑设计和物理设计结
果严格描述出来，成为RDBMS可以接受的源代码，再经过调
试产生目标模式，然后就可以组织数据入库，这就是数据库实
施阶段。
- 本节主要内容
 1. 数据的载入和应用程序的调试
 2. 数据库的试运行
 3. 数据库的运行和维护

1.数据的载入和应用程序的调试

- 数据库实施阶段包括两项重要的工作：
 - 数据的载入
 - 应用程序的编码和调试
- 组织数据载入就是要将各类源数据从各个局部应用中抽取出来，输入计算机，再分类转换，最后综合成符合新设计的数据库结构的形式，输入数据库。
 - 这样的数据转换、组织入库的工作是相当费力、费时的
 - 因为数据的组织方式、结构和格式都与新设计的数据库系统有相当的差距
 - 特别是原系统是手工数据处理系统时，各类数据分散在各种不同的原始表格、凭证和单据之中

- 为提高数据录入的工作的效率和质量，应该针对具体的应用环境设计一个数据录入子系统，由计算机来完成数据入库的任务。
- 现代RDBMS一般都提供不同RDBMS之间数据转换的工具，应充分利用新系统的数据转换工具。
 - ORACLE与SQL SERVER数据转换示例:
 - <http://www.cnblogs.com/jxgzCHforever/p/8650056.html>
- 数据库应用程序的设计应该与数据库设计同时进行，因此在组织数据入库的同时还要调试应用程序。
 - 应用程序的设计、编码和调试的方法、步骤参见软件工程相关课程

2.数据库的试运行

- 在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，这也称为数据库的试运行。
- 数据库的试运行包括：
 - 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，直至达到设计要求为止。
 - 测试系统的性能指标，分析其是否达到设计目标。
 - 原因：设计时得到的只是近似估计，与实际系统运行有一定差距
 - 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

■ 注意事项：

- 如果试运行后还要修改数据库的设计，则需要重新组织数据入库。因此，应分期分批组织数据入库。
 - 先输入小批量数据做调试用，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
- 要做好数据库的转储和恢复工作，一旦故障发生，能使数据库尽快恢复，尽量减少对数据库的破坏。
 - 这是因为，试运行阶段的数据库系统还不稳定，软硬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，误操作不可避免

3.数据库的运行和维护

- 数据库试运行合格后，数据库开发工作就基本完成，可以投入正式运行。但由于应用环境不断变化，数据库运行过程中物理存储也会不断变化，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。
- 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。
- 数据库的维护工作主要包括：
 1. 数据库的转储和恢复
 - 数据库的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，以保证发生故障后能尽快将数据库恢复到某种一致性的状态，尽可能减少对数据库的破坏

3.数据库的运行和维护

■ 数据库的维护工作主要包括：

2. 数据库的安全性、完整性控制

- 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应根据实际情况修改原有的安全性控制
- 完整性亦然

3. 数据库性能的监督、分析和改造

- 在数据库运行过程中，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是DBA的又一重要任务
- 主流RDBMS一般会提供监测系统性能参数的工具

4. 数据库的重组与重构造

- 数据库运行一段时间后，由于记录不断增删改，将会使数据库的物理存储情况变坏，降低数据的存储效率，使得数据库系统性能下降
- DBA要对数据库进行重组或部分重组(仅针对频繁增删的表)

本章小结

- 详细介绍了数据库设计各个阶段的目标、方法和步骤，重点是概念结构的设计和逻辑结构的设计。
- 数据库各级模式的形成。
 - 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：概念模式（信息世界模型），用E-R图来描述
 - 逻辑设计阶段：逻辑模式、外模式
 - 物理设计阶段：内模式
- 数据库应用程序的设计应与数据库设计同时进行。
- 数据库运维过程中DBA的主要职责和工作内容。

课堂练习

- 数据库外模式是在下列哪个阶段设计 ()
A.数据库概念结构设计 B.数据库逻辑结构设计 C.数据库物理结构设计 D.数据库实施和维护
- 生成DBMS系统支持的数据模型是在下列哪个阶段完成 ()
A.数据库概念结构设计 B.数据库逻辑结构设计 C.数据库物理结构设计 D.数据库实施和维护
- 根据应用需求建立索引是在下列哪个阶段完成 ()
A.数据库概念结构设计 B.数据库逻辑结构设计 C.数据库物理结构设计 D.数据库实施和维护
- 员工性别的取值，有的为“男”、“女”，有的为“1”、“0”，这种情况属于 ()
A.属性冲突 B.命名冲突 C.结构冲突 D.数据冗余

- 数据库设计方法包括_____、_____、_____、_____和统一建模语言（UML）方法等。
- 集成局部E-R图要分为两个步骤，分别是_____和_____。
- 数据库常见的存取方法主要有_____、_____和hash方法。
- 在进行概念结构设计时，将事物作为属性的基本准则是什么？
- 将E-R图转换为关系模式时，可以如何处理实体型之间的联系？

本章作业

- 教材第七章习题：1-15(全部).