第七章　数据库设计

第七章 数据库设计

7.1 数据库设计概述

- 广义地讲，是数据库及其应用系统的设计，即设计整个数据库应用系统；
- 狭义地讲，是设计数据库本身，即设计数据库的各级模式并建立数据库，这是数据库应用系统设计的一部分。
- 什么是数据库设计？

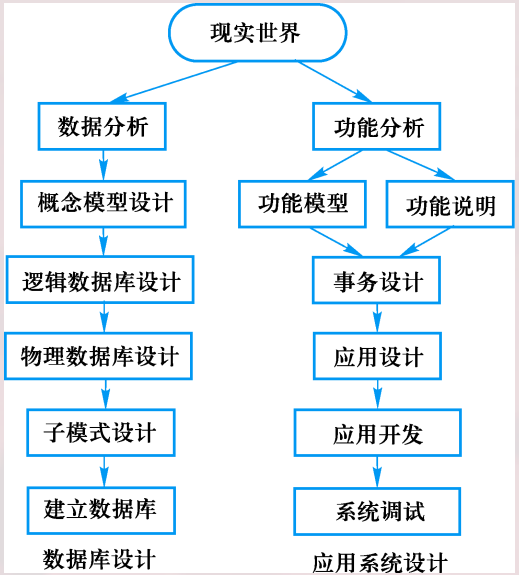
数据库设计是指对于一个给定的应用环境，设计一个优良的数据库逻辑模式和物理结构，并据此建立数据库及其应用系统，使之能够有效地存储和管理数据，满足各种用户的应用需求，包括信息管理要求和数据处理要求：

- 信息管理要求：在数据库中存储和管理需要的数据对象 。
- 数据处理要求：对数据对象需要进行的处理，如查询、增删改、统计和分析等。

7.1.1 数据库设计的特点

1. 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



2. 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
- 传统的软件工程：重 行为设计
  - 忽视对应用中数据语义的分析和抽象，只要有可能就尽量推迟数据结构设计
- 早期的数据库设计：重 结构设计
  - 致力于数据模型和数据库建模方法研究，忽视了行为设计对结构设计的影响

7.1.2 数据库设计方法

- 大型数据库设计是涉及多学科的综合性技术，又是一项庞大的工程项目。
- 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

7.1.2 数据库设计方法（续）

- 手工设计法
  - 设计质量与设计人员的经验和水平有直接关系
  - 缺乏科学理论和工程方法的支持，工程的质量难以保证
  - 数据库运行一段时间后常常又不同程度地发现各种问题，增加了维护代价
- 规范设计法
  - 典型方法——新奥尔良（New Orleans）方法
    - 将数据库设计分为若干阶段和步骤
    - 采用辅助手段实现每一过程
    - 按设计规程用工程化方法设计数据库

7.1.2 数据库设计方法（续）

- 基于E-R模型的设计方法

  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

- UML(Unified Modeling Language)方法

  面向对象的建模方法

数据库设计方法（续）

❖ 数据库设计工具

- SYBASE PowerDesigner

  数据库建模—UML工具

- Rational Rose

  UML工具—数据库建模

- CA ERWin

  ERwin全称是ERwin Data Modeler

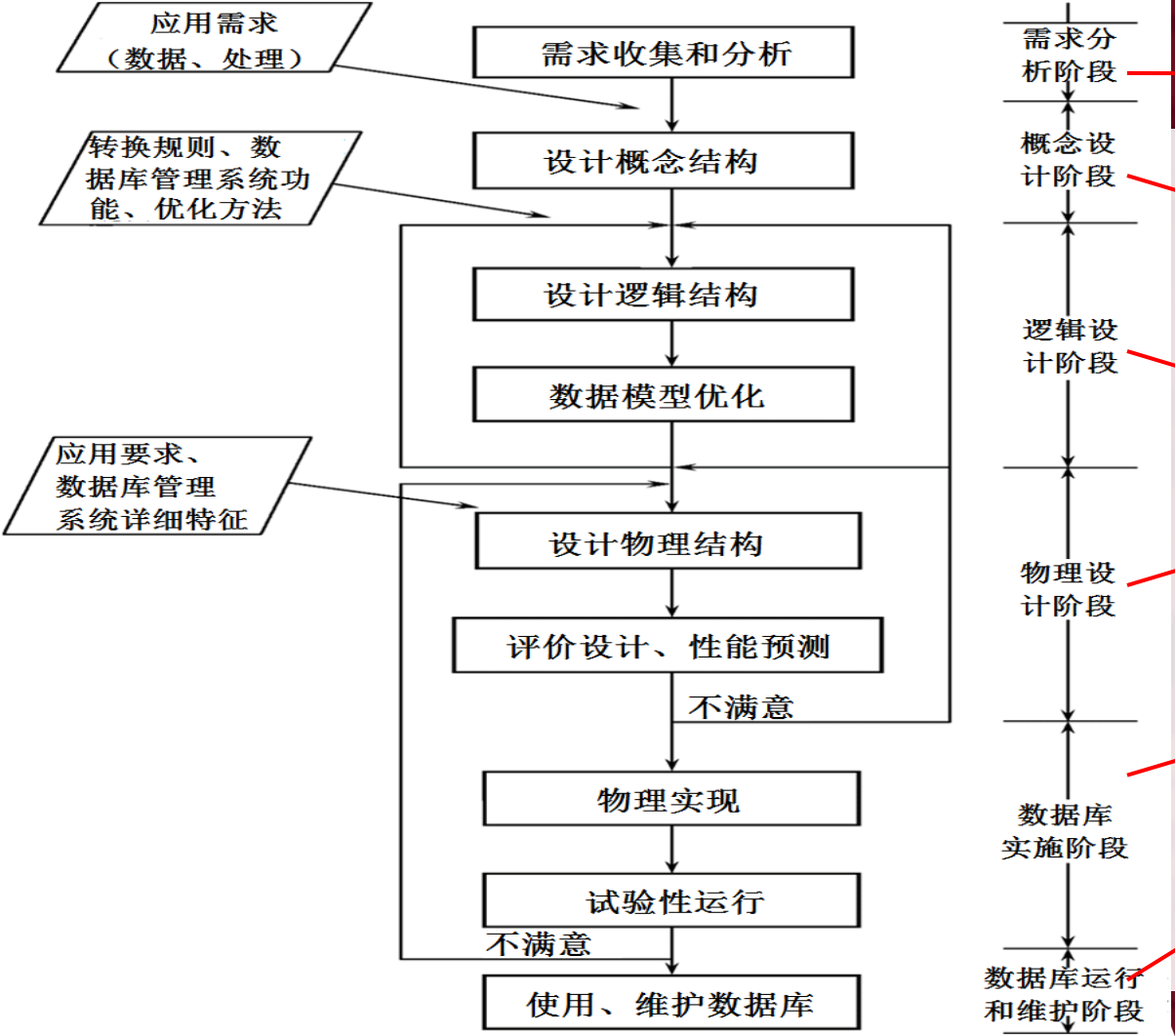
  功能强大、易于使用的数据建模、数据库设计与开发工具

7.1.3 数据库设计的基本步骤

- 数据库设计分6个阶段
  - 需求分析
  - 概念结构设计

  独立于任何数据库管理系统
  - 逻辑结构设计
  - 物理结构设计

  与选用的数据库管理系统密切相关
  - 数据库实施
  - 数据库运行和维护



是否做得充分与准确，决定了构建数据库的速度和质量

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

将概念结构转换为某个DBMS所支持的数据模型，并对其进行优化

为逻辑数据结构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

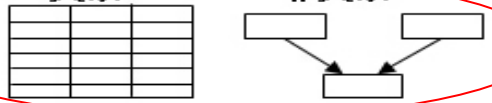
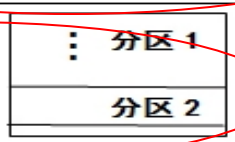
根据逻辑设计和物理设计的结果构建数据库，编写与调试应用程序，组织数据入库并进行试运行

经过试运行后即可投入正式运行。
在运行过程中必须不断对数据库设计进行评估、调整与修改。

数据库设计的基本步骤（续）

- 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。
- 这个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。
- 把数据库的设计和对数据库中数据处理的设计紧密结合起来，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，以完善两方面的设计。

数据库设计的基本步骤（续）

设计阶段	设计描述
需求分析	数字字典、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 关系 非关系 
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

数据库设计各个阶段的数据设计描述

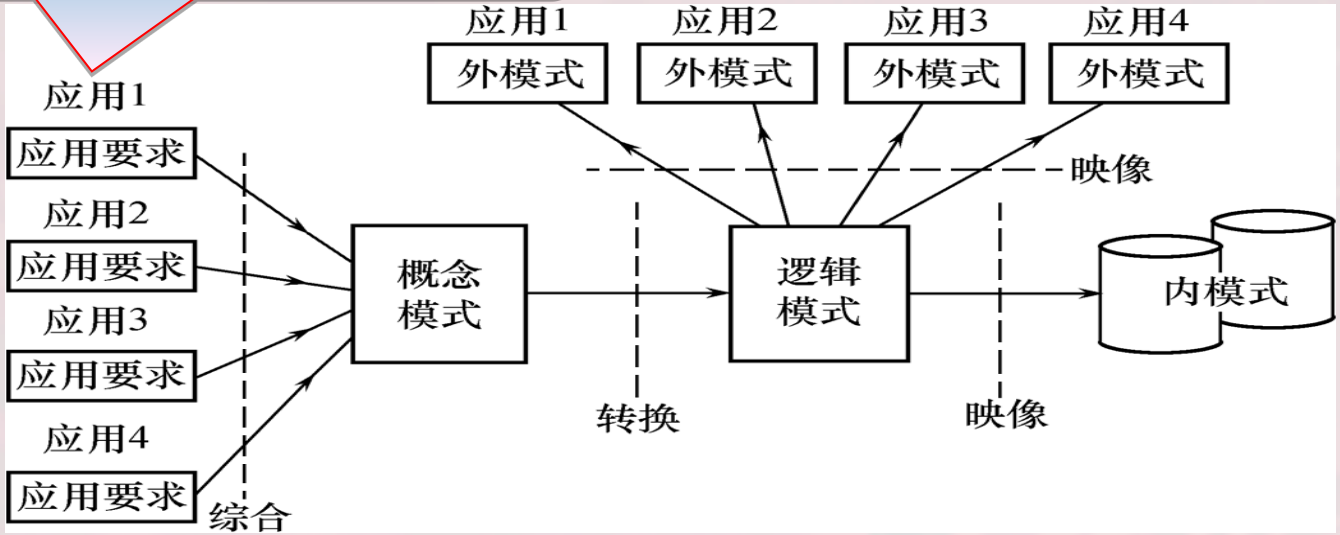
数据库设计各个阶段产生的
设计文档/设计说明

数据库设计的基本步骤（续）

- 参加数据库设计的人员
  - 系统分析人员和数据库设计人员
    - 自始至终参与数据库设计
  - 数据库管理员和用户代表
    - 主要参加需求分析与数据库的运行和维护
  - 应用开发人员
    - 包括程序员和操作员
    - 在实施阶段参与进来，分别负责编制程序和准备软硬件环境

7.1.4 数据库设计过程中的各级模式

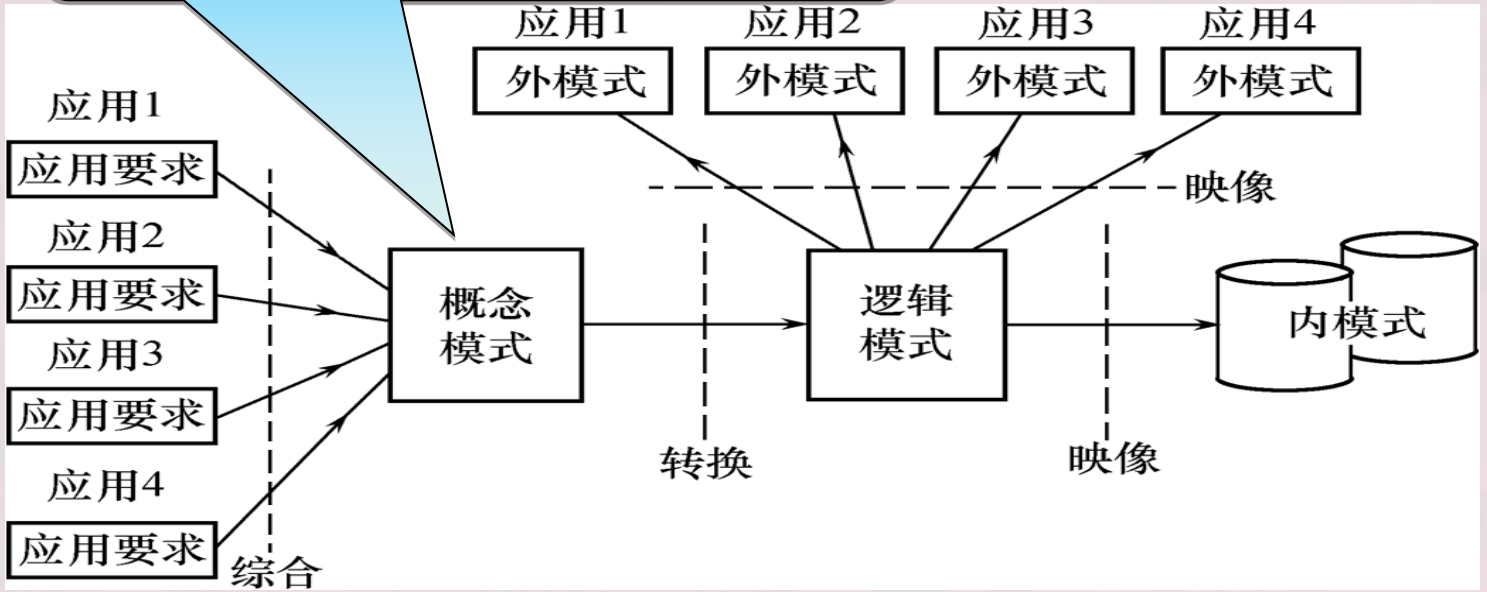
需求分析阶段：
综合各个用户的应用需求



数据库的各级模式

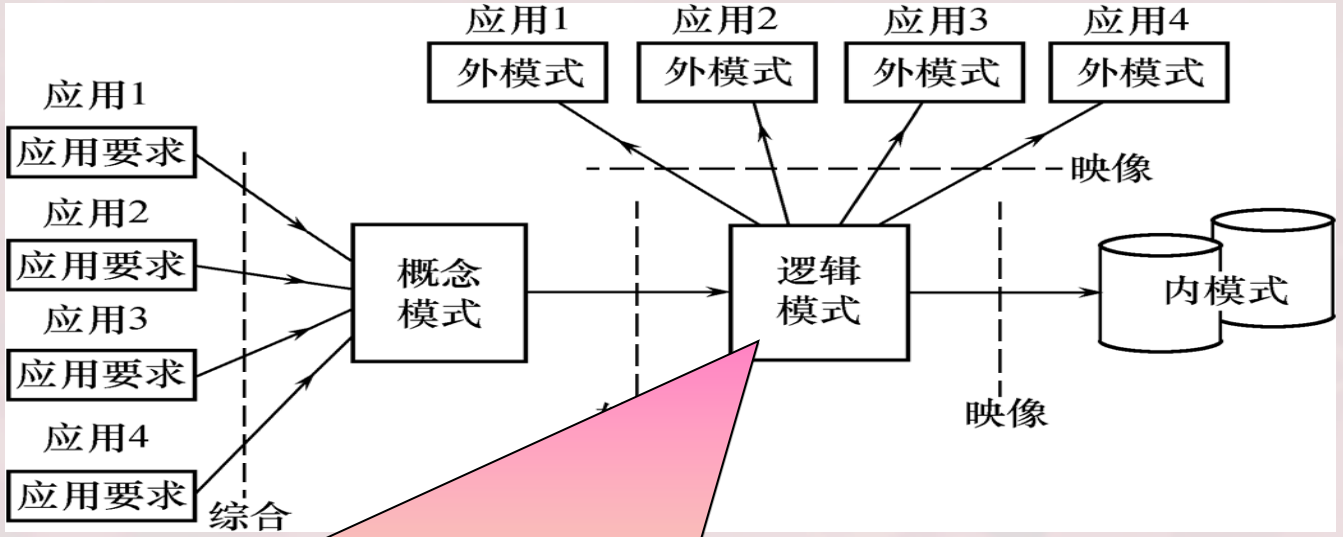
数据库设计过程中的各级模式（续）

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（E-R图）



数据库的各级模式

数据库设计过程中的各级模式（续）



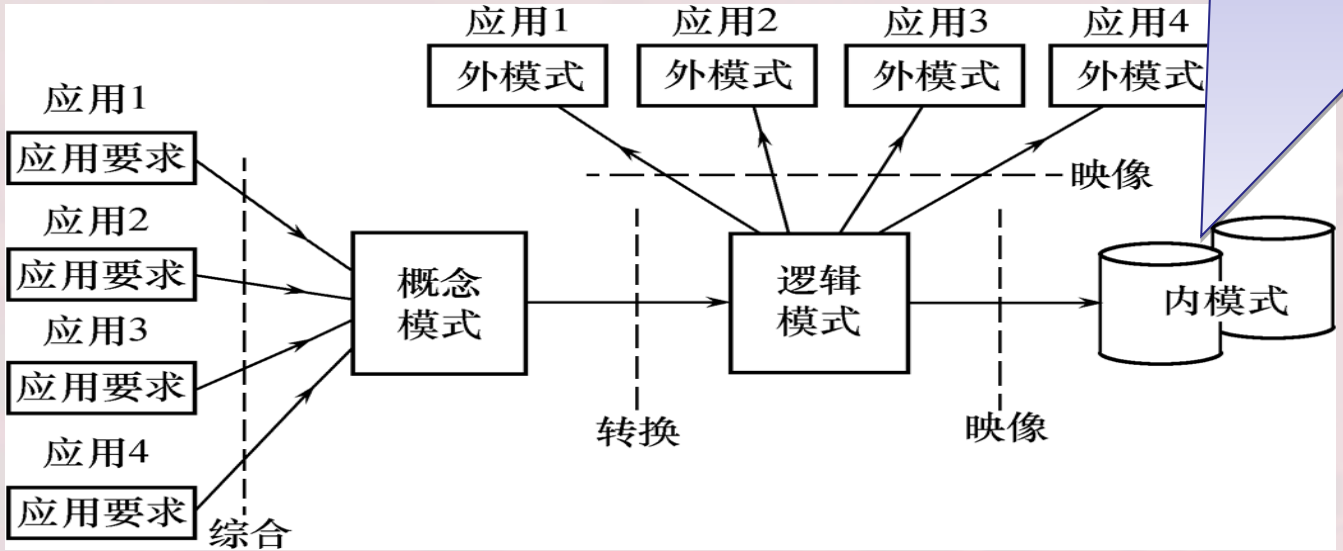
逻辑设计阶段：

1. 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

2. 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

数据库设计过程中的各级模式（续）

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



数据库的各级模式

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 数据字典

7.2 需求分析

- 什么是需求分析——分析用户的要求
  - 是设计数据库的起点
- 需求分析的重要性
  - 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
- 需求分析常常被忽视
  - 设计人员认为这是软任务，急于进行具体设计
  - 用户嫌麻烦
  - 领导不重视

7.2.1 需求分析的任务

- 充分了解原系统（手工系统或计算机系统）工作概况
- 详细调查要开发应用系统的组织/部门/企业等
- 明确用户的各种需求

确定新系统的功能

注意：新系统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

1.信息要求

- 用户需要从数据库中获得信息的内容与性质
- 由信息要求可以导出数据要求，即在数据库中需要存储哪些数据

2.处理要求

- 用户要什么处理功能、对处理性能、处理方式、处理周期等的要求

（批处理 / 联机处理/ 发布处理/每月一次/……）

3.安全性与完整性要求

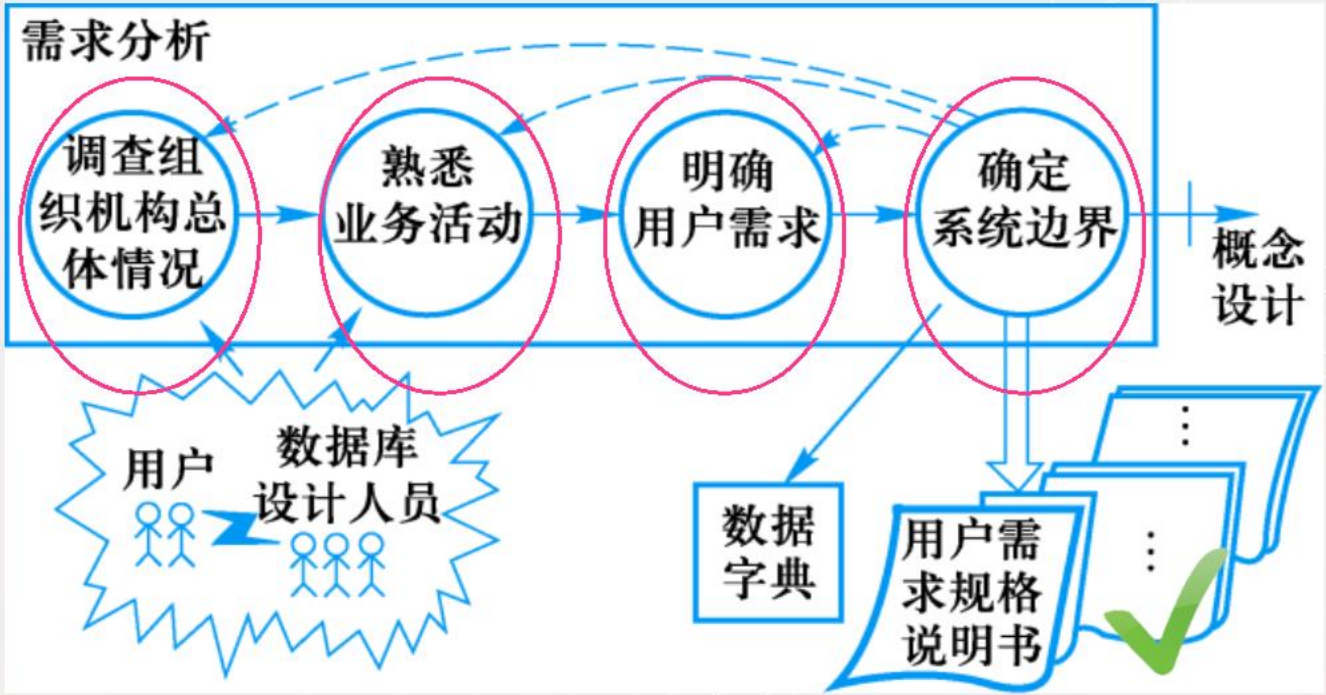
确定用户需求的难点

- 用户缺少计算机知识，不能准确地表达自己的需求，提出的需求往往不断地变化。
- 设计人员缺少用户的专业知识，不易理解用户的真正需求，甚至误解用户的需求。

解决方法

- 设计人员必须不断深入地与用户进行交流，才能逐步确定用户的实际需求

7.2.2 需求分析的方法


需求分析
调查组织机构总体情况
熟悉业务活动
明确用户需求
确定系统边界
概念设计
用户
数据库设计人员
数据字典
用户需求规格说明书

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 数据字典

7.2.3 数据字典

- 什么是数据字典？

  数据字典是关于数据库中数据的描述，称为元数据。

  它不是数据本身，而是数据的数据。
- 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。
- 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。

注意：与 DBMS中的数据字典的区别和联系

数据字典（续）

- 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
- 数据项是数据的最小组成单位
- 若干个数据项可以组成一个数据结构
- 通过对数据项和数据结构的定义来描述数据流、数据存储的逻辑内容

1. 数据项

- 数据项是不可再分的数据单位
- 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 }
- 关系规范化理论为指导，用数据依赖的概念分析和抽象数据项之间的联系——函数依赖
- “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，是模式设计、完整性检查条件、触发器、存储过程的依据

实例

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： 9

取值范围：0000 00 000至9999 99 999

取值含义：前4位标别该学生入学年份， 第5第6位所在专业系编号，后3位按顺序编号， 例如201615008

与其他数据项的逻辑关系：学号的值确定了其他数据项的值

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}

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} }

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

3. 数据流

- 数据流是数据结构在系统内部传输的路径。
- 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 }

- 数据流来源：说明该数据流来自哪个处理过程/数据存储
- 数据流去向：说明该数据流将到哪个处理过程/数据存储去
- 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

实例

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报告

数据流来源：体检（处理过程）

数据流去向：批准（处理过程）

组成： { 学号， {血常规}，{尿常规}，{血液生化}，{心电图}，
{B超}，… … {其他体检} }

平均流量： 每天200

高峰期流量：每天400

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。

- 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，每次存取的数据量等信息
  - 存取方法：批处理 / 联机处理；检索 / 更新；顺序检索 / 随机检索
  - 输入的数据流：数据来源
  - 输出的数据流：数据去向

实例

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：每学期5000

流出数据流：每学期5000

组成： {学号，姓名，性别，年龄，所在系，年级，{学习成绩}，{体检结果}，{奖惩记录} … … }

数据量： 每年10000张

存取方式： 随机存取+按照专业系/班级打印

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

- 处理过程

  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
  - 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
  - 处理要求是物理设计的输入及性能评价的标准

实例

处理过程“分配宿舍”可如下描述：

处理过程：分配宿舍

说明：为所有新生分配学生宿舍

输入：学生，宿舍

输出：宿舍安排

处理：在新生报到后，为所有新生分配学生宿舍。

要求同一间宿舍只能安排同一年级同一性别的学生

一个学生只能安排在一个宿舍中。每个学生的居住面积不小于6平方米。

安排新生宿舍其处理时间应不超过15分钟。

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} }

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描述

  是下一步进行概念设计的基础。
- 强调两点

  （1）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

  （2）必须强调用户的参与，领导的重视

7.3 概念结构设计

7.3.1 概念模型

7.3.2 E-R模型

*7.3.3 扩展的E-R模型

*7.3.4 UML

7.3.5 概念结构设计

7.3.1 概念模型

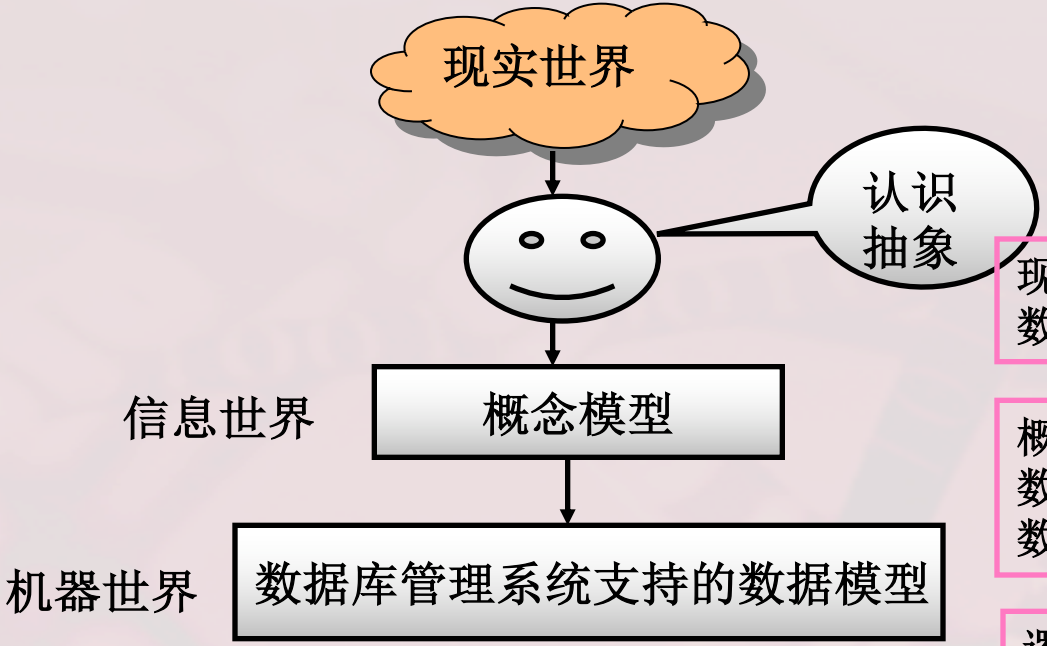
❖什么是概念结构设计

➢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；

➢概念结构是现实世界的一个真实模型。是各种数据模型的共同基础，它比数据模型更独立于机器、更抽象，从而更加稳定；

➢概念结构设计是数据库设计的关键。

回顾：两类数据模型



现实世界 ⇨ 概念模型
数据库设计人员完成

概念模型 ⇨ 逻辑模型
数据库设计人员完成
数据库设计工具协助完成

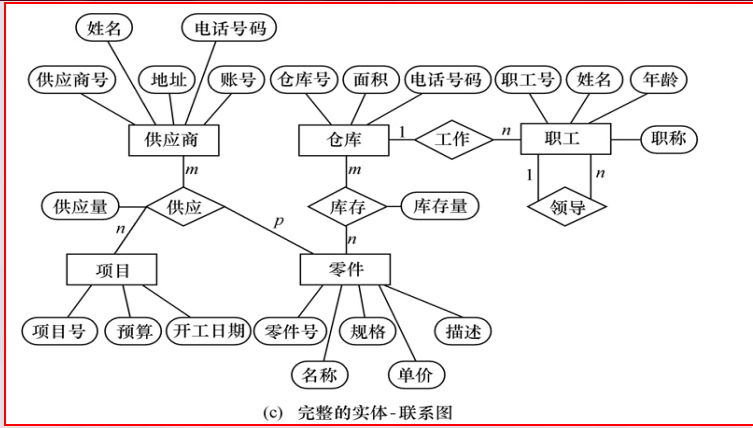
逻辑模型 ⇨ 物理模型
由DBMS完成

现实世界中客观对象的抽象过程

回顾：1.2.2 概念模型

❖ 概念模型的用途

- 概念模型用于信息世界的建模
- 是现实世界到机器世界的一个中间层次
- 是数据库设计的有力工具
- 数据库设计人员和用户之间进行交流的语言



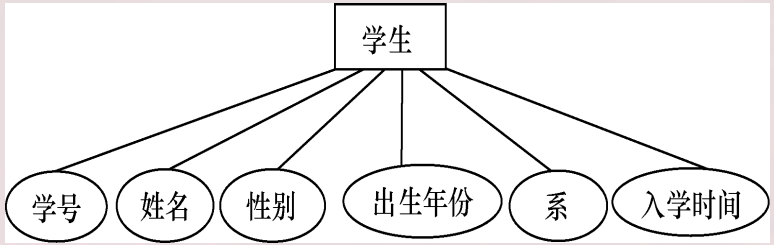
(c) 完整的实体-联系图

例：工厂物质管理的概念模型

❖ 对概念模型的基本要求

- 较强的语义表达能力
- 简单、清晰、易于用户理解

回顾：信息世界中的基本概念



（1）实体（Entity）

客观存在并可相互区别的事物称为实体。

可以是具体的人、事、物或抽象的概念。

（2）属性（Attribute）

实体所具有的某一特性称为属性。一个实体可以由若干个属性来刻画。

（3）码（Key）

唯一标识实体的属性集称为码。

（4）实体型（Entity Type）

用实体名及其属性名集合来抽象和刻画同类实体称为实体型

（5）实体集（Entity Set）

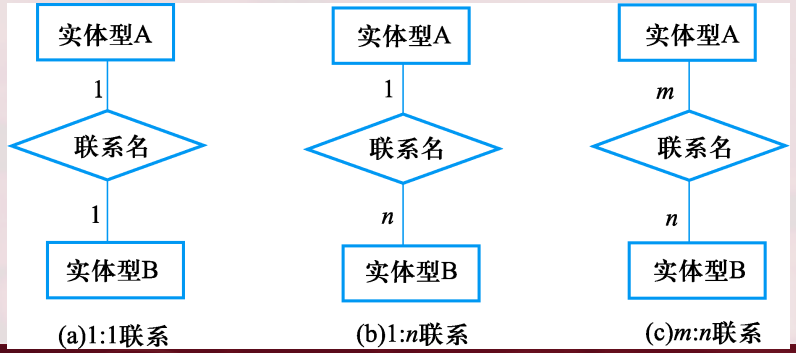
同一类型实体的集合称为实体集

回顾：信息世界中的基本概念（续）

（6）联系（Relationship）

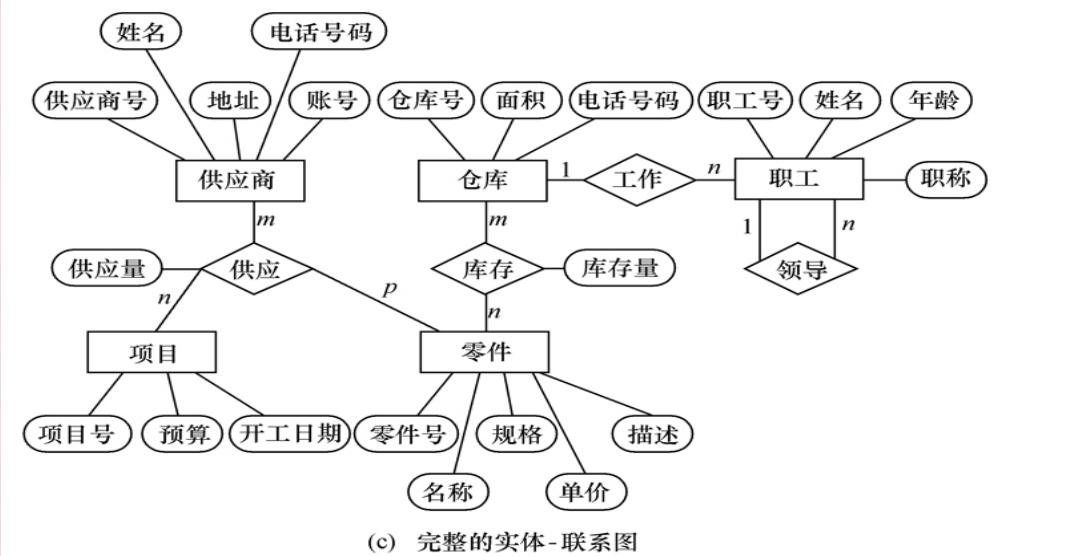
- 现实世界中事物内部以及事物之间的联系在信息世界中反映为实体（型）内部的联系和实体（型）之间的联系。
- 实体内部的联系: 是指组成实体的各属性之间的联系
- 实体之间的联系: 通常是指不同实体集之间的联系

实体之间的联系有一对一（1:1）、一对多（1:m）和多对多（m:n）等多种类型



回顾：实体-联系方法

- 概念模型的一种表示方法：
- 实体-联系方法（Entity-Relationship Approach）
  - 用E-R图来描述现实世界的概念模型
  - E-R方法也称为E-R模型



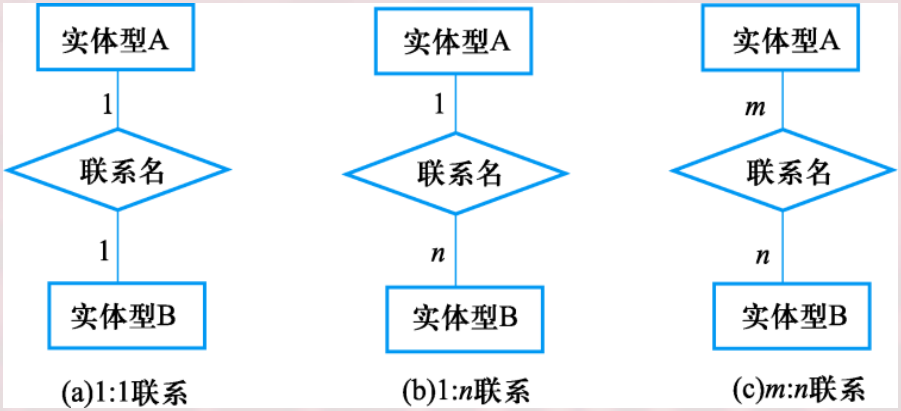
(c) 完整的实体-联系图

7.3.2 E-R模型

1. 实体之间的联系

（1）两个实体型之间的联系，可以分为三种：

- 一对一联系（1:1）
- 一对多联系（1:*n*）
- 多对多联系（*m*:*n*）



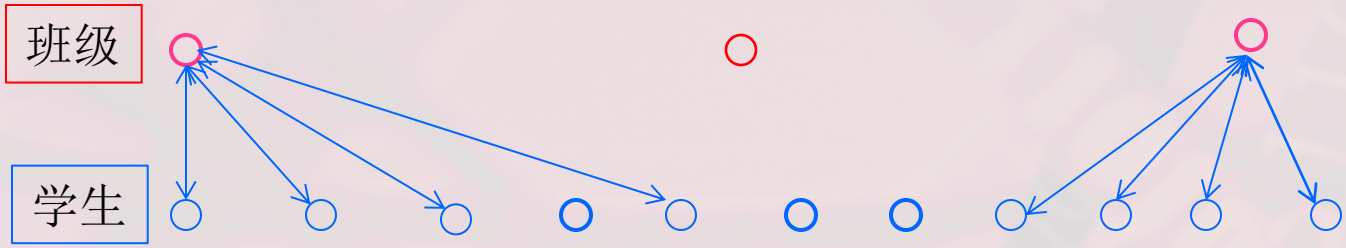
E-R模型（续）

- 一对一联系（1:1）
  - 如果对于实体集*A*中的每一个实体，实体集*B*中最多有一个（也可以没有）实体与之联系，反之亦然，则称实体集*A*与实体集*B*具有一对一联系，记为1:1。
  - 例如，学校里一个班级只有一个正班长，而一个班长只在一个班中任职，则班级与班长之间具有一对一联系。

E-R模型（续）

②一对多联系（1:n）

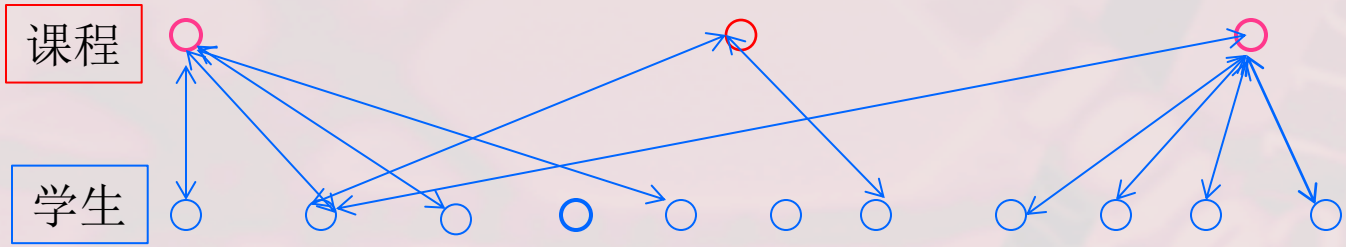
- 如果对于实体集*A*中的每一个实体，实体集*B*中有*n*个实体（*n*≥0）与之联系，反之，对于实体集*B*中的每一个实体，实体集*A*中至多只有一个实体与之联系，则称实体集*A*与实体集*B*有一对多联系，记为1:*n*。
- 例如，一个班级中有若干名学生，而每个学生只在一个班级中学习，则班级与学生之间具有一对多联系。



E-R模型（续）

③多对多联系（m:n）

- 如果对于实体集*A*中的每一个实体，实体集*B*中有*n*个实体（*n*≥0）与之联系，反之，对于实体集*B*中的每一个实体，实体集*A*中也有*m*个实体（*m*≥0）与之联系，则称实体集*A*与实体集*B*具有多对多联系，记为*m*:*n*。
- 例如，一门课程同时有若干个学生选修，而一个学生可以同时选修多门课程，则课程与学生之间具有多对多联系。

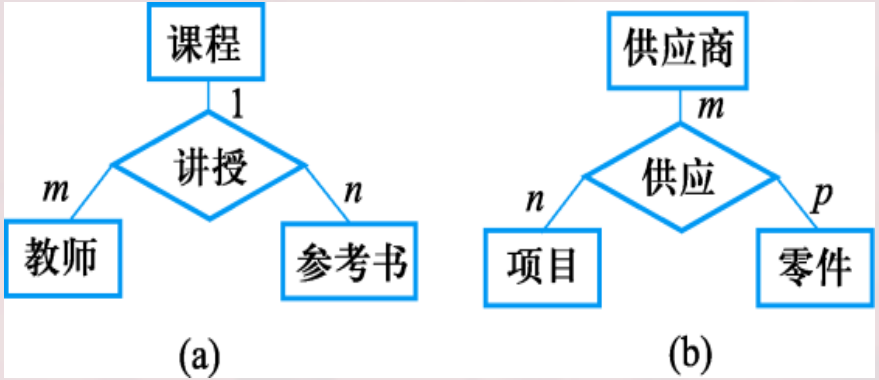


E-R模型（续）

（2）两个以上的实体型之间的联系：也存在着一对一、一对多、多对多联系。

对于课程、教师与参考书，如果一门课程可以有若干个教师讲授，使用若干本参考书，而每一个教师只讲授一门课程，每一本参考书只供一门课程使用。

则课程与教师、参考书之间的联系是一对多的。



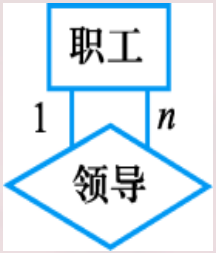
对于供应商、项目、零件，一个供应商可以供给多个项目多种零件，而每个项目可以使用多个供应商供应的零件，每种零件可由不同供应商供给。

可以看出供应商、项目、零件三者之间是多对多的联系。

E-R模型（续）

（3）单个实体型内的联系，也存在一对一、一对多、多对多的联系。

- 例如，职工实体型内部具有领导与被领导的联系，即某一职工（干部）“领导”若干名职工，而一个职工仅被另外一个职工直接领导，因此这是一对多的联系。



把参与联系的实体型的数目称为联系的度。

两个实体型之间的联系度为2，也称为二元联系；

三个实体型之间的联系度为3，也称为 三元联系；

*N*个实体型之间的联系度为*N*，也称为N元联系。

E-R模型（续）

2. 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，矩形框内写明实体名。
- 属性：用椭圆形表示，并用无向边将其与相应的实体型连接起来。

例，学生实体具有学号、姓名、性别、出生年份、系、入学时间等属性，

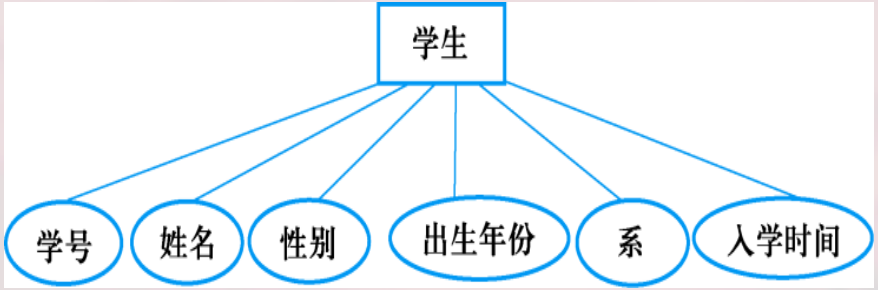


图7.9 学生实体及属性

E-R模型（续）

- 联系：用菱形表示，菱形框内写明联系名，并用无向边分别与有关实体型连接起来，同时在无向边旁标上联系的类型（1:1，1:*n* 或*m*:*n*）。
- 联系可以具有属性

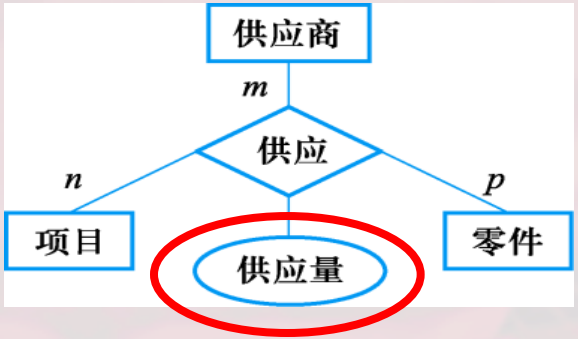


图7.10 联系的属性

E-R图 例题解析（1）

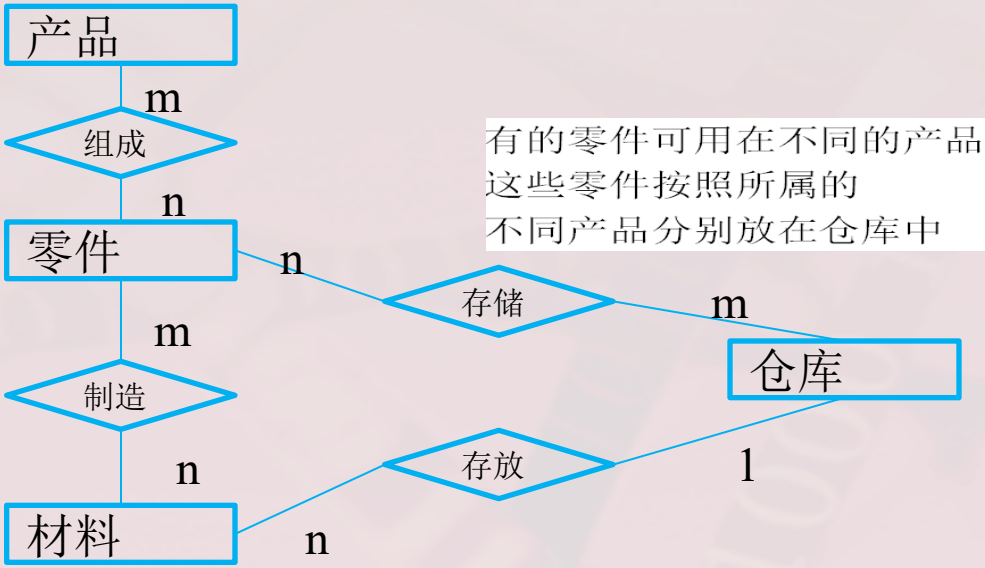
例题1

- 某工厂生产若干产品，每种产品由不同的零件组成，有的零件可用在不同的产品。这些零件由不同的原材料制成，不同零件所用的材料可以相同。有的零件可用在不同的产品，这些零件按照所属的不同产品分别放在仓库中，原材料按照类别放在若干仓库中。
- 请用E-R图画出此工厂产品、零件、材料、仓库的概念模型。

解答

每种产品由不同的零件组成，
有的零件可用在不同的产品。

这些零件由不同的原材料制成，
不同零件所用的材料可以相同。



有的零件可用在不同的产品，
这些零件按照所属的
不同产品分别放在仓库中

原材料按照类别放在若干仓库中

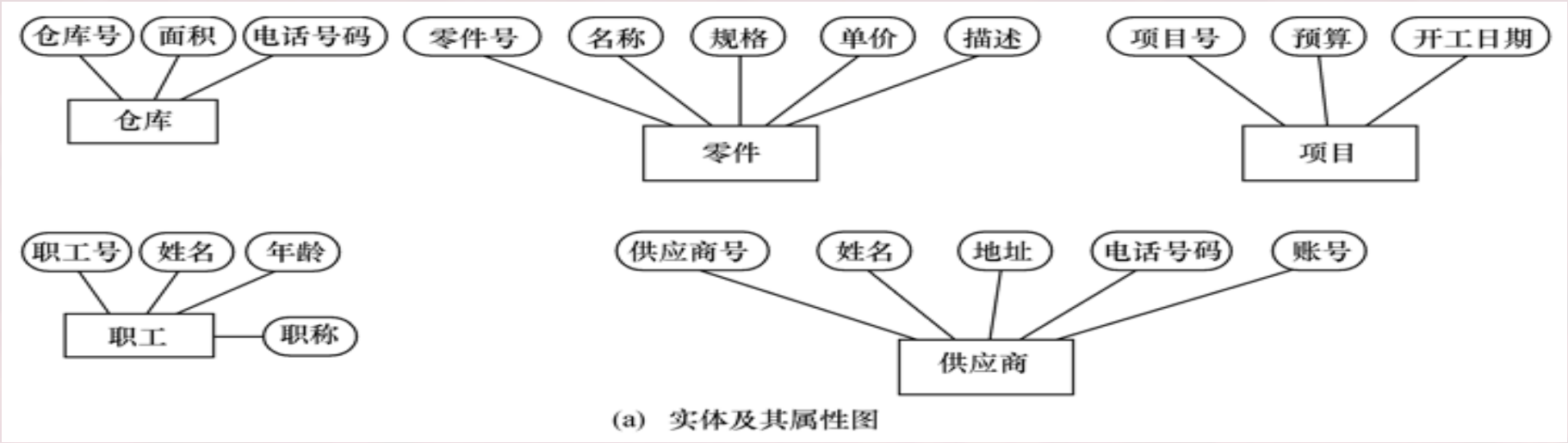
工厂产品、零件、材料、仓库的E-R图

E-R图例题解析（2）

例题2

- 画出某个工厂物资管理的概念模型（教科书P.218）
- 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

实体属性图



(a) 实体及其属性图

实体之间的联系

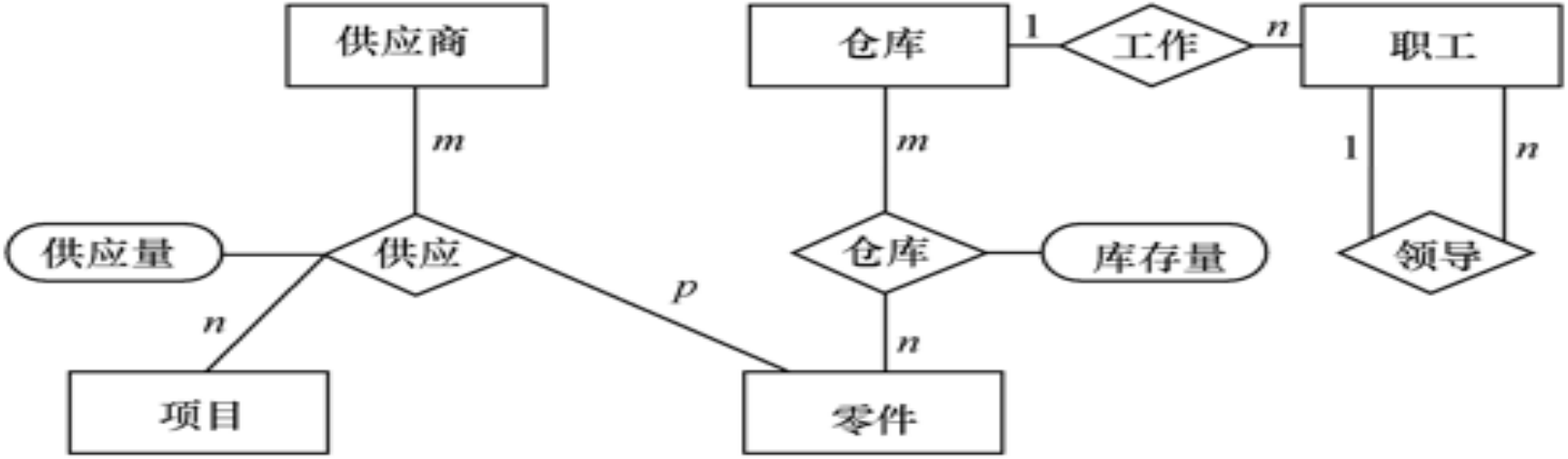
（1）一个仓库可以存放多种零件，一种零件可以存放在多个仓库中，因此仓库和零件具有多对多的联系。**用库存量来表示某种零件在某个仓库中的数量。**

（2）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。

（3）职工之间具有领导-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。

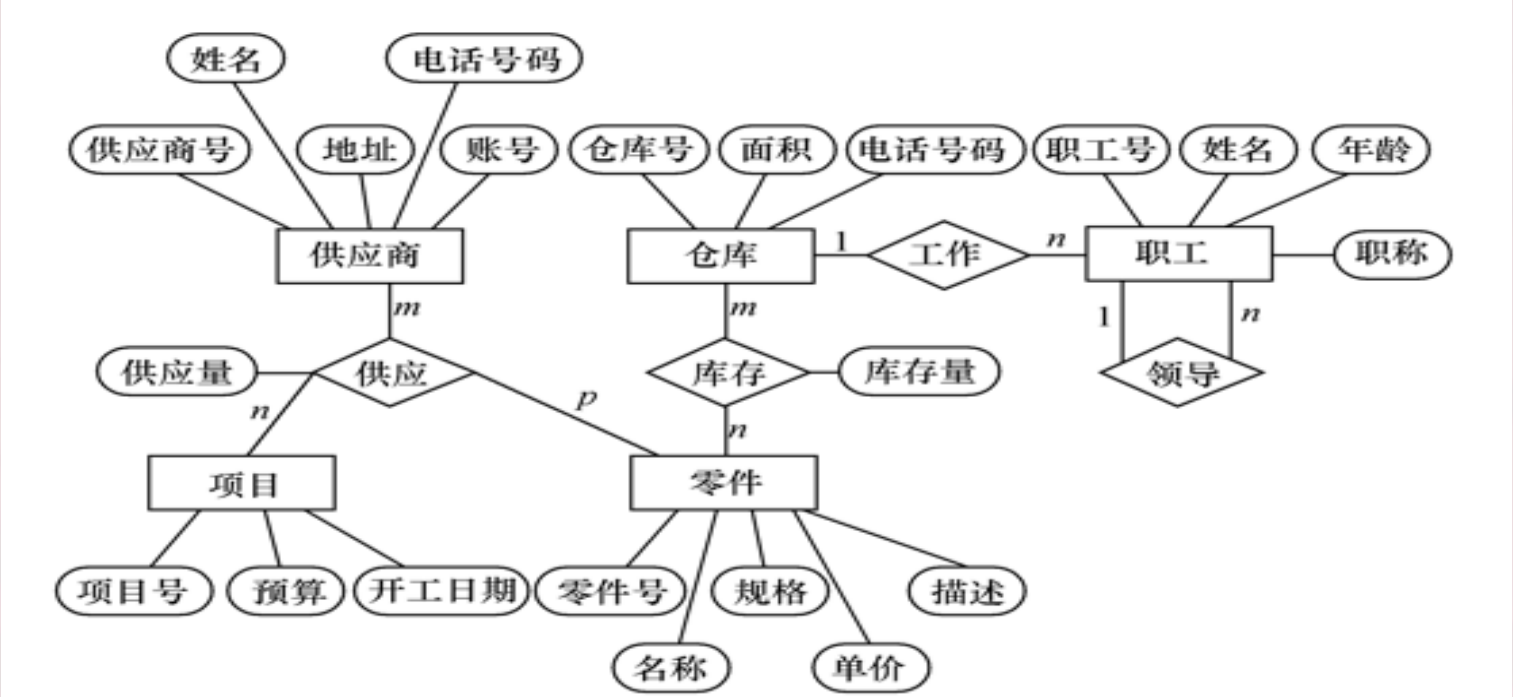
（4）供应商、项目和零件三者之间具有多对多的联系。即一个供应商可以供给若干项目多种零件，每个项目可以使用不同供应商供应的零件，每种零件可由不同供应商供给。

实体与实体之间联系图



(b) 实体及其联系图

工厂物资管理的E-R图



(c) 完整的实体-联系图

7.3 概念结构设计

7.3.1 概念结构

7.3.2 E-R模型

*7.3.3 扩展的E-R模型

*7.3.4 UML

7.3.5 概念结构设计

讲解：

1. 实体与属性的划分原则

➢对需求分析阶段收集到的数据进行分类、组织

➢确定实体、实体的属性、实体之间的联系类型

2. E-R图的集成

➢设计各个子系统的分E-R图

➢消除冲突，进行集成

➢设计基本E-R图

7.3.5 概念结构设计

1. 实体与属性的划分原则

❖ 现实世界的事物能作为属性对待的，尽量作为属性对待。

可以简化E-R图的处置

❖ **划分实体与属性的两条准则：**

（1）作为属性，不能再具有需要描述的性质。

属性必须是不可分的数据项，不能包含其他属性。

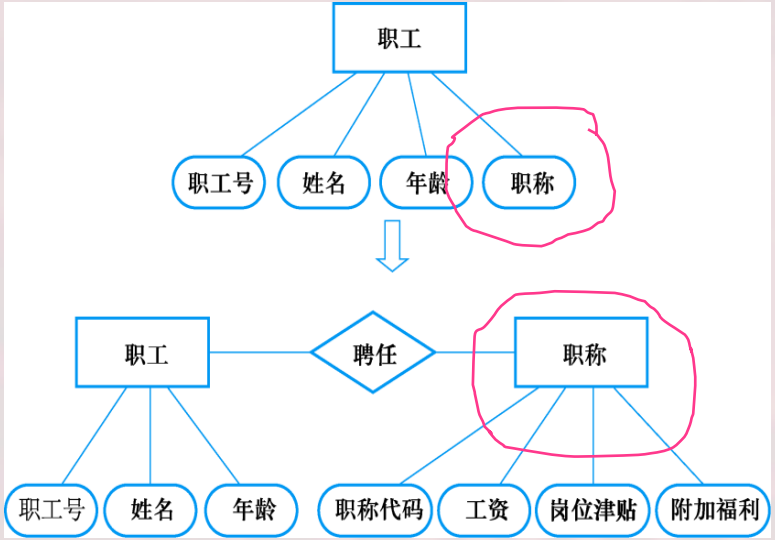
（2）属性不能与其他实体具有联系。

E-R图中所表示的联系是实体与实体之间的联系。

实体与属性的划分

[例1] 职工实体：职工号、姓名、年龄是职工的属性，如何设计职称？

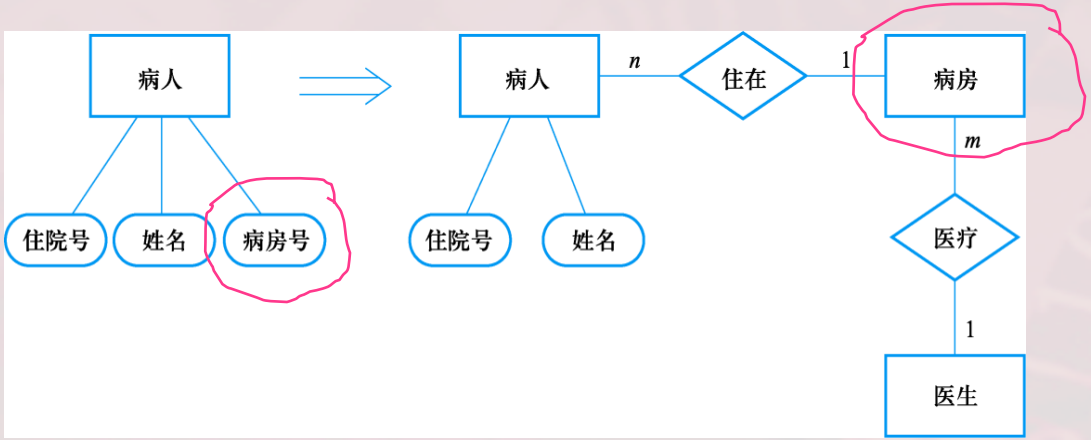
- 如果不存储和处理与职称相关的工资、福利，根据准则（1）设计为职工的属性。
- 如果需要存储或处理：与职称相关的工资、津贴、附加福利等，则职称应设计为一个实体。



实体与属性的划分

[例2] 医院中病人实体，如何设计病房？

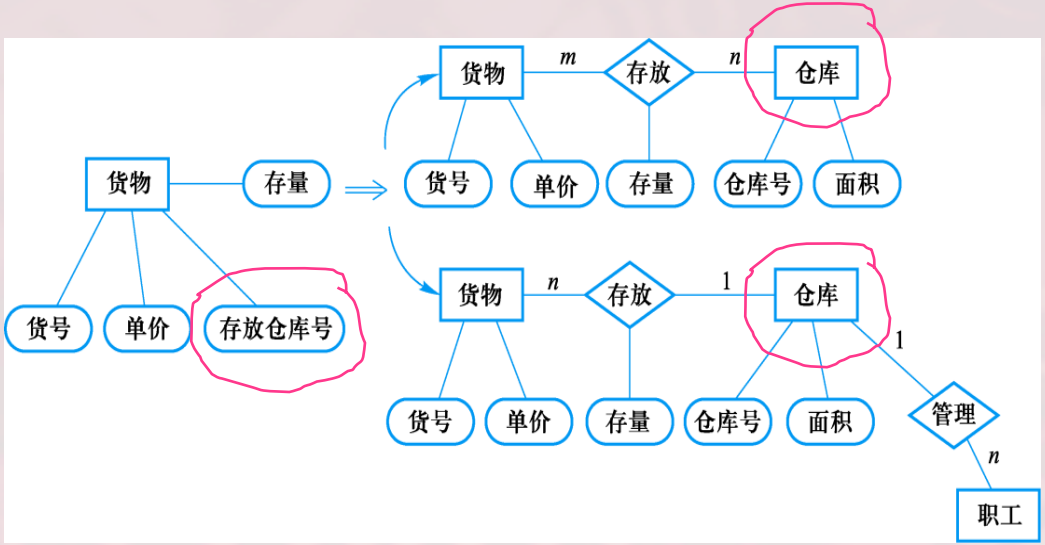
- 一个病人只能住在一个病房，病房号可以作为病人实体的一个属性；
- 如果病房还要与医生实体发生联系，即一个医生负责若干病房的病人的医疗工作则根据准则（2） 病房应作为一个实体。



实体与属性的划分

[例3] 货物实体，如何设计仓库？

- 如果一种货物只存放在一个仓库，可以把仓库号作为描述货物存放地点的属性。
- 如果一种货物可以存放在多个仓库中，或者仓库本身又用面积作为属性，或者仓库与职工发生管理上的联系，那么就应把仓库作为一个实体。



设计各个子系统的分E-R图

- 某工厂开发信息系统，经过可行性分析，详细调查确定了该系统由物资管理、销售管理、劳动人事管理等子系统组成。
- **[例7.1] 销售管理子系统E-R图的设计**
- 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

通过需求分析，知道销售子系统功能围绕“订单”和“应收账款”的处理来实现。

设计销售子系统的分E-R图

先设计该E-R图的草图：

数据结构中订单、顾客、顾客应收账目、产品是许多子功能、数据流共享的数据，因此设计为实体。（如图7.22所示）

再对E-R图进行详细设计和调整：

参照需求分析和数据字典中的详尽描述，遵循前面给出的两个准则。

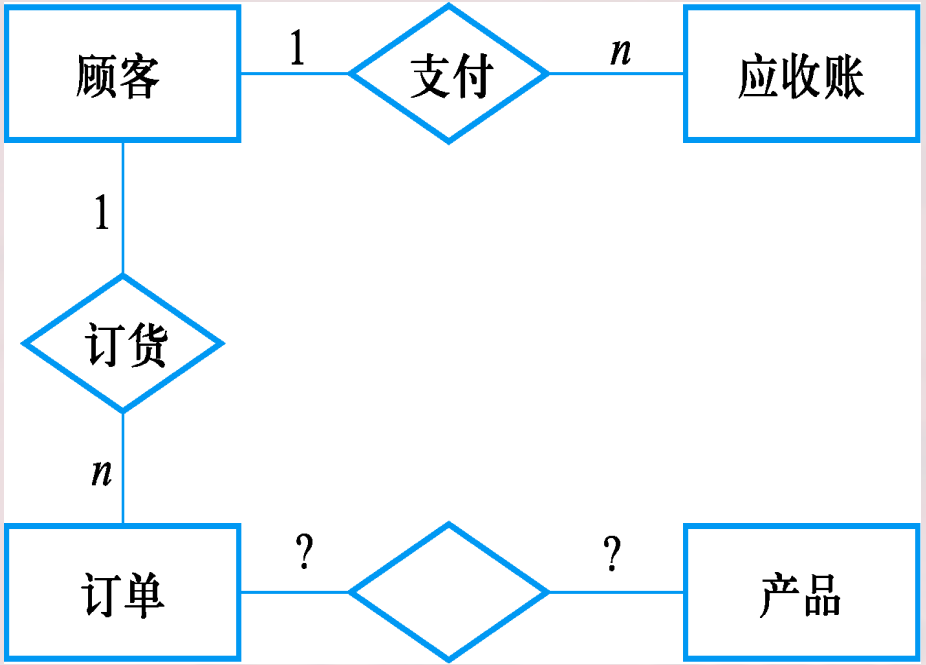
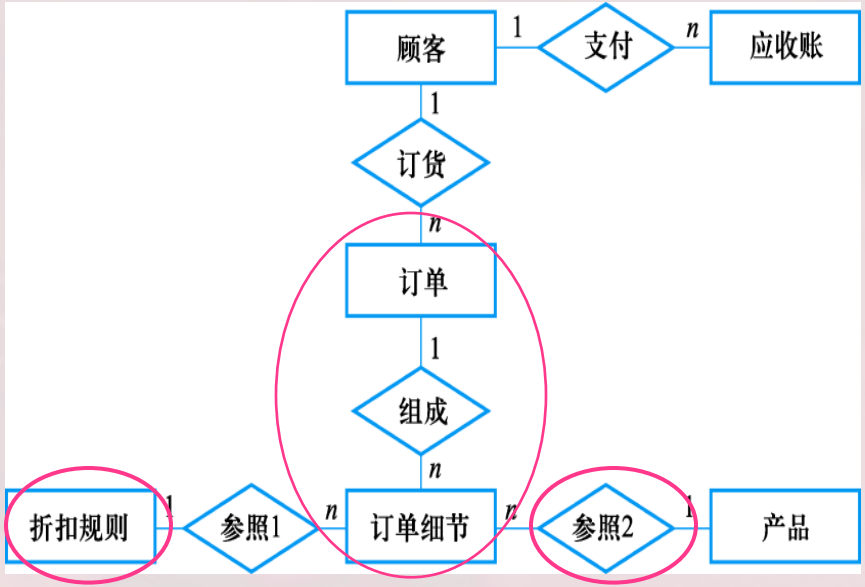


图7.22 分E-R图的框架

设计销售子系统的分E-R图



（2）原订单和产品的联系实际上是订单细节和产品的联系，增加了一个联系—参照2。产品描述，进一步细化，包括当前单价、产品重量等信息。

（3）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，应增加一个实体—“折扣规则”存放这些信息。

（1）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。按照准则（2），订单细节就应该上升为实体。一张订单可以订若干产品，订单与订单细节两个实体之间是1:n的联系。

设计销售子系统的分E-R图

❖ 最后得到销售管理子系统E-R图：

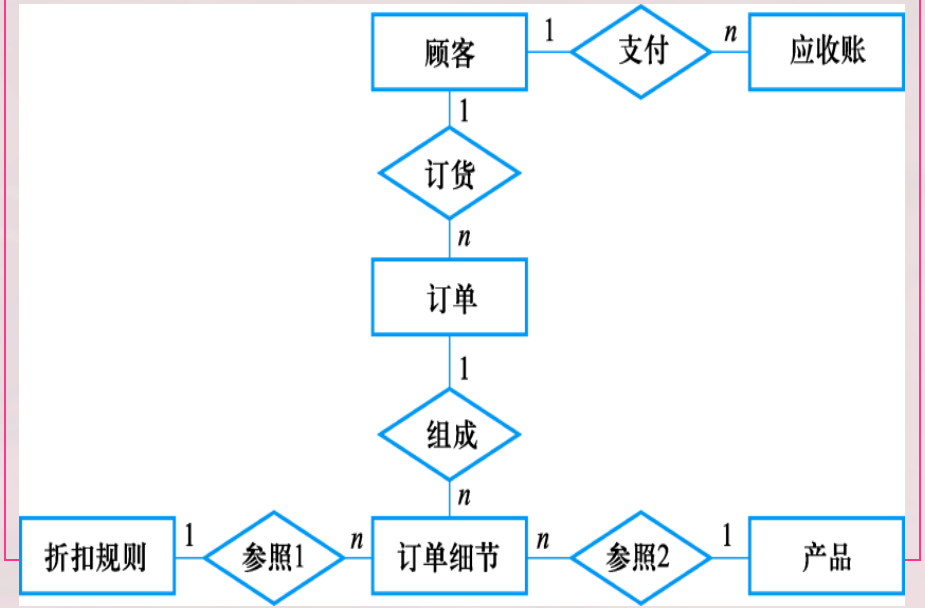


图7.23 销售管理子系统分E-R图

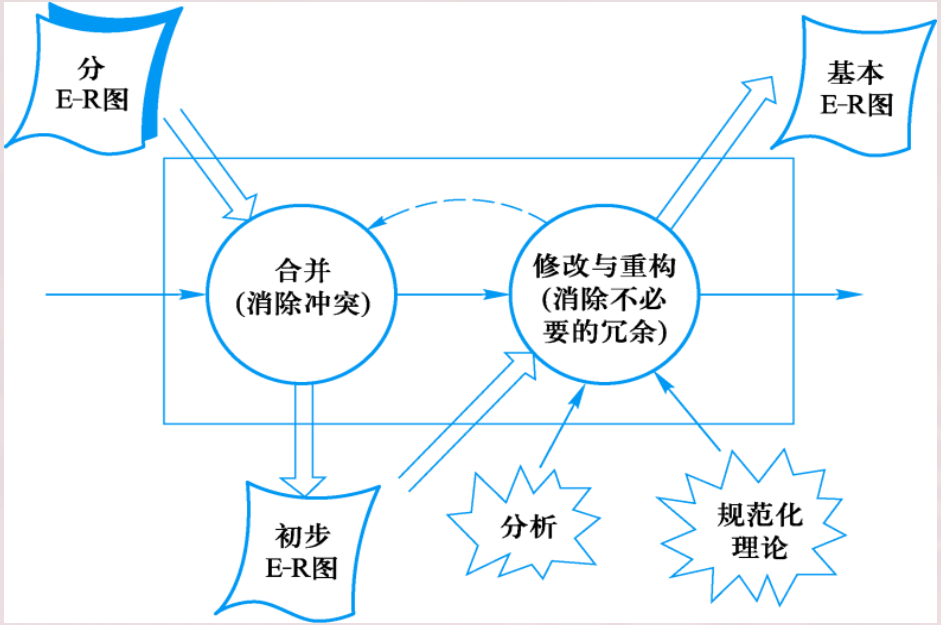
❖ 对每个实体定义的属性如下：

- 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：{产品号，产品名，单价，重量}
- 折扣规则：{产品号，订货量，折扣}

这里省略了实体属性图，实体的码用下划线划出

E-R图的集成

2. E-R图的集成



E-R图的集成一般需要分两步

- 合并。解决各分E-R图之间的冲突，将分E-R图合并，生成初步E-R图。
- 修改和重构。消除不必要的冗余，生成基本E-R图。

合并E-R图，生成初步E-R图

（1）合并E-R图，生成初步E-R图

- 各个局部应用所面向的问题不同，各个子系统的E-R图之间必定会存在许多不一致的地方，称之为冲突。
- 子系统E-R图之间的冲突主要有三类：

①属性冲突

②命名冲突

③结构冲突

属性冲突

①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。
  - 例如零件号，有的部门把它定义为整数，有的部门把它定义为字符型。
  - 年龄，某些部门以出生日期形式表示职工的年龄，有的用整数表示职工的年龄。
- 属性取值单位冲突。
  - 例如零件的重量有的以公斤为单位，有的以斤为单位，有的以克为单位。

命名冲突

②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。
  - 如对科研项目，财务科称为项目，科研处称为课题，生产管理处称为工程。
- 命名冲突
  - 可能发生在实体、联系一级上
  - 也可能发生在属性一级上
  - 通过讨论、协商等行政手段加以解决

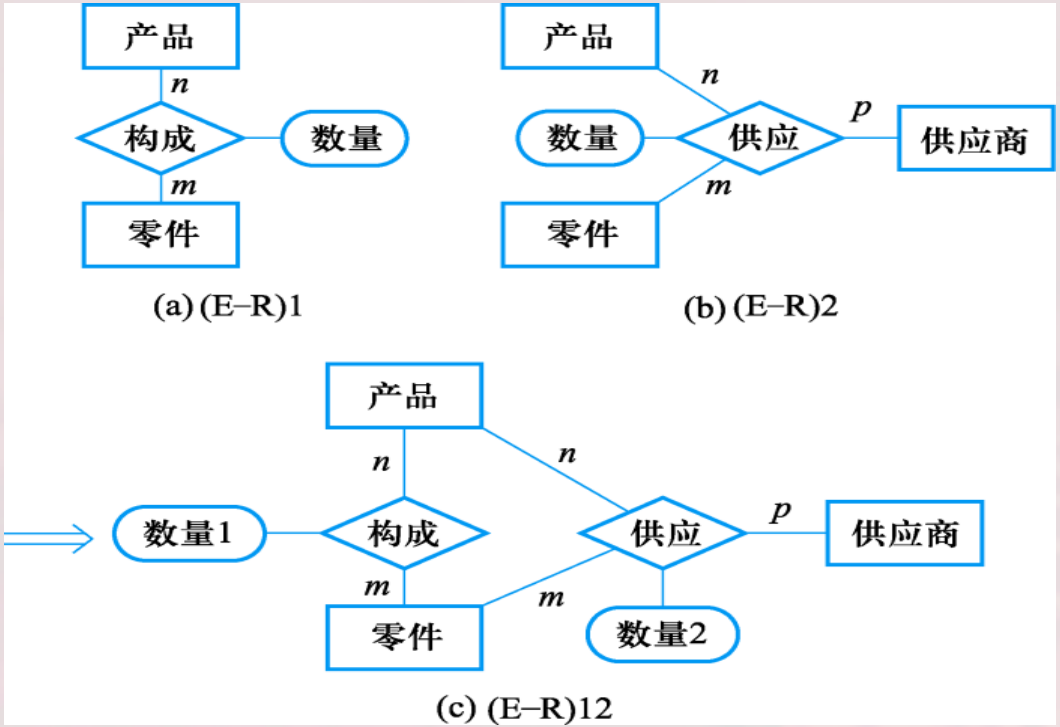
结构冲突

③结构冲突

- 同一对象在不同应用中具有不同的抽象。
  - 例如，职工在某一局部应用中被当作实体，而在另一局部应用中被当作属性。
  - 解决方法：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。
- 同一实体在不同子系统的E-R图中的属性个数和属性排列次序不完全相同。
  - 解决方法：取各子系统的E-R图中属性的并集，再适当调整属性的次序。
- 实体间的联系在不同的E-R图中为不同的类型。
  - 例如，实体E1与E2在一个E-R图中是多对多联系，在另一个E-R图中是一对多联系
  - 解决方法是根据应用的语义对实体联系的类型进行综合或调整。

合并E-R图 生成初步E-R图

图(a)中
零件与产品之间存在多对多的联系：
“构成”

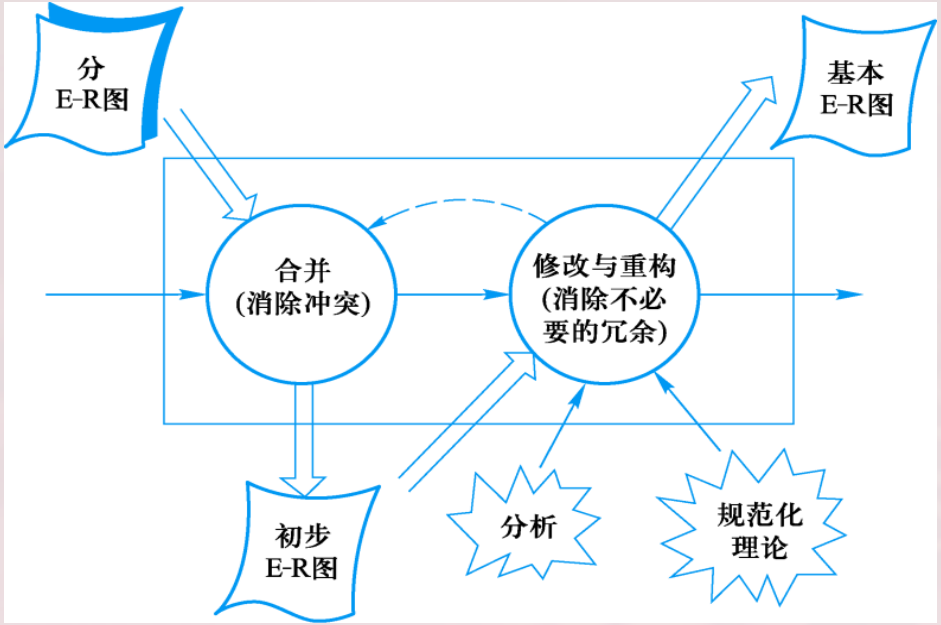


图(b)中
产品、零件与供应商三者之间还存在多对多的联系
“供应”

合并两个E-R图，生成图(c)

E-R图的集成

2. E-R图的集成



E-R图的集成一般需要分两步

- 合并。解决各分E-R图之间的冲突，将分E-R图合并，生成初步E-R图。
- 修改和重构。消除不必要的冗余，生成基本E-R图。

设计 基本E-R图

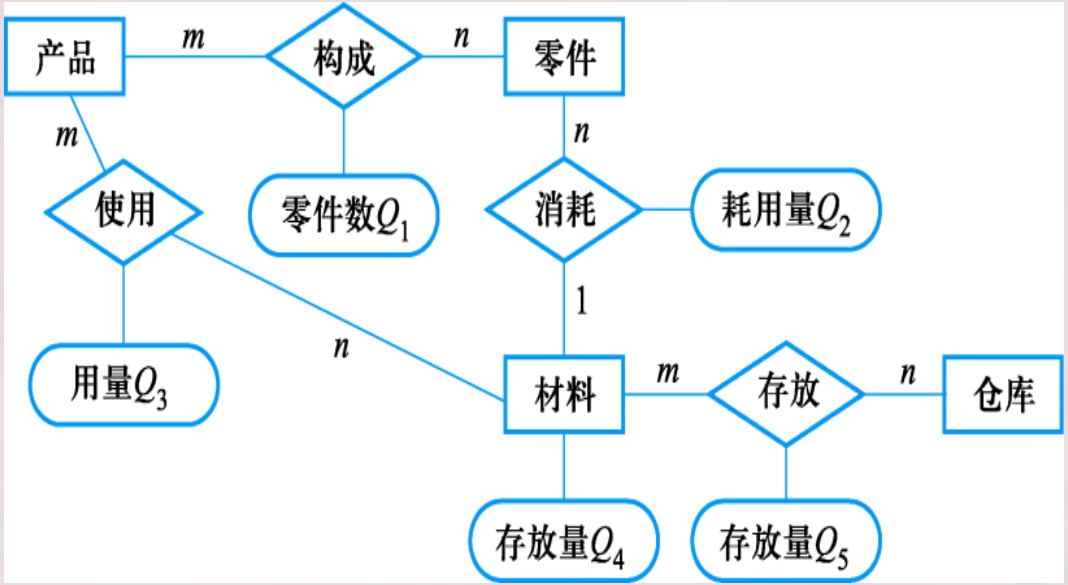
（1）合并E-R图，生成初步E-R图

（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
- 冗余带来的问题：破坏数据库的完整性，数据库维护困难，应当予以消除
- 消除冗余方法：
  - 分析方法
  - 规范化理论的方法

设计 基本E-R图

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。



$Q_3=Q_1\times Q_2$，$Q_4=\sum Q_5$

所以Q_3和Q_4是冗余数据

可以消去。

由于Q_3消去，产品与材料间

*m:n*的冗余联系也应消去。

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。

设计 基本E-R图

❖ 用规范化理论来消除冗余

①确定分E-R图实体之间的数据依赖。

- 实体之间一对一、一对多、多对多的联系可以用实体码之间的函数依赖来表示。于是有函数依赖集F_L

②求F_L的最小覆盖G_L，差集为 $D= F_L - G_L$

- 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

❖ 应注意的问题：

- 冗余的联系一定在D中，而D中的联系不一定是冗余的；
- 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。

设计 基本E-R图

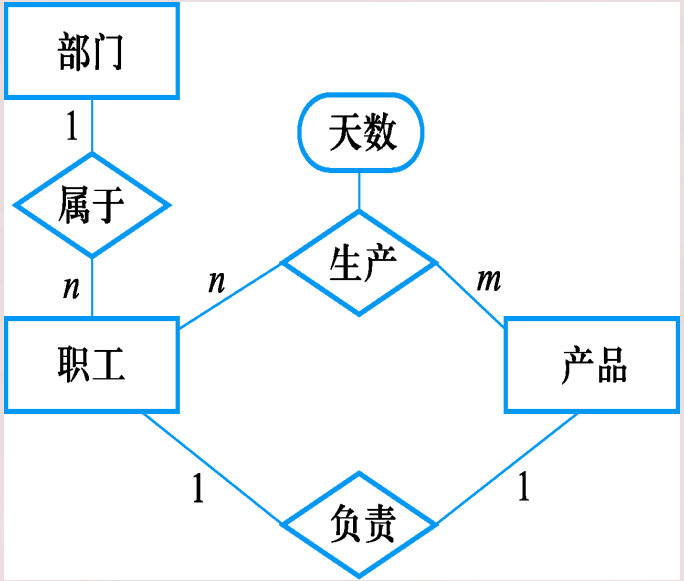


图7.27 劳动人事管理的分E-R图

部门和职工之间一对多的联系
可表示为：
职工号→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）→工作天数

职工和产品之间一对一的联系
负责人.职工号→产品号
产品号→负责人.职工号
等。

设计 基本E-R图

[例7.2] 某工厂管理信息系统的视图集成。

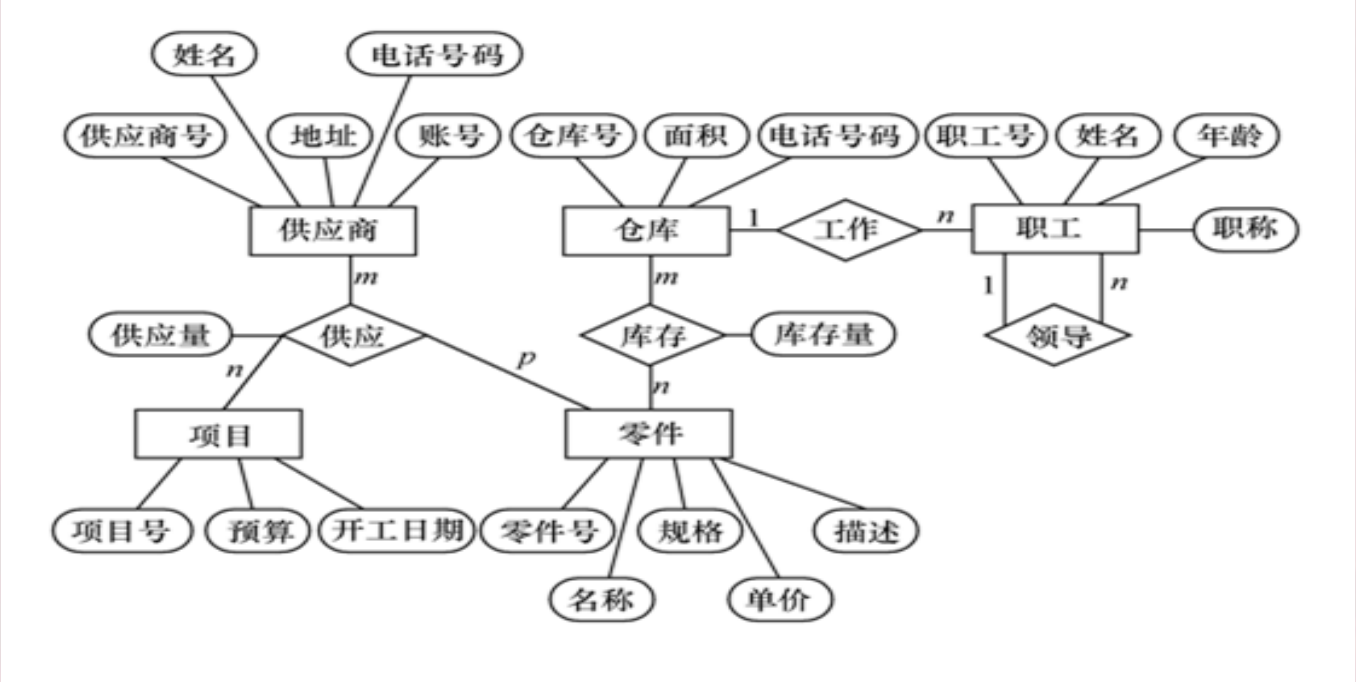


图7.11 工厂物资管理子系统的E-R图

设计 基本E-R图

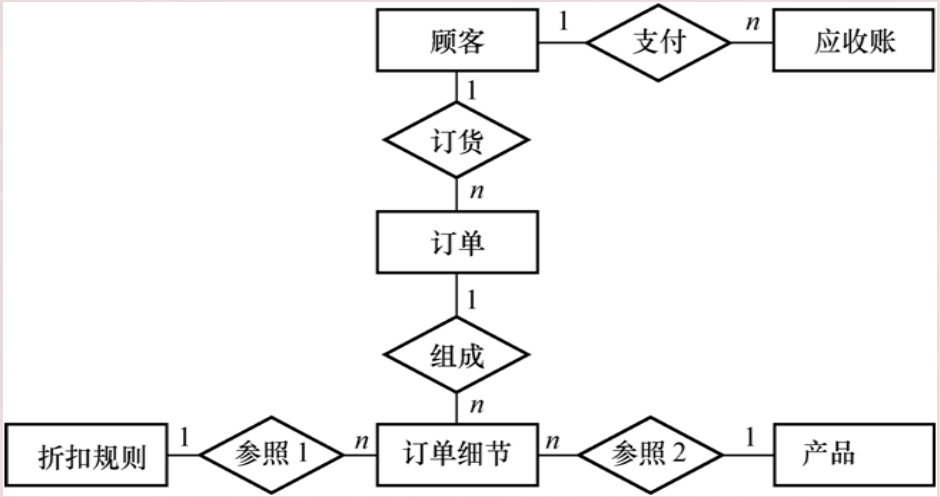


图7.23 销售管理子系统的E-R图

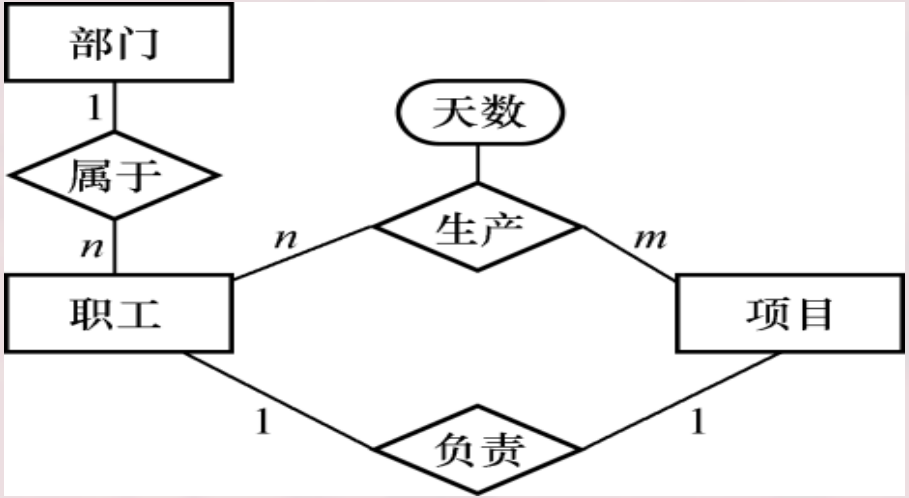


图7.27 劳动人事管理子系统的E-R图

设计 基本E-R图

[例7.2] 某工厂信息系统的E-R图集成

图7.28 某工厂信息系统的基本E-R图
==
图7.11 工厂物资管理子系统的E-R图
+图7.23 销售管理子系统的E-R图
+图7.27 劳动人事管理子系统的E-R图

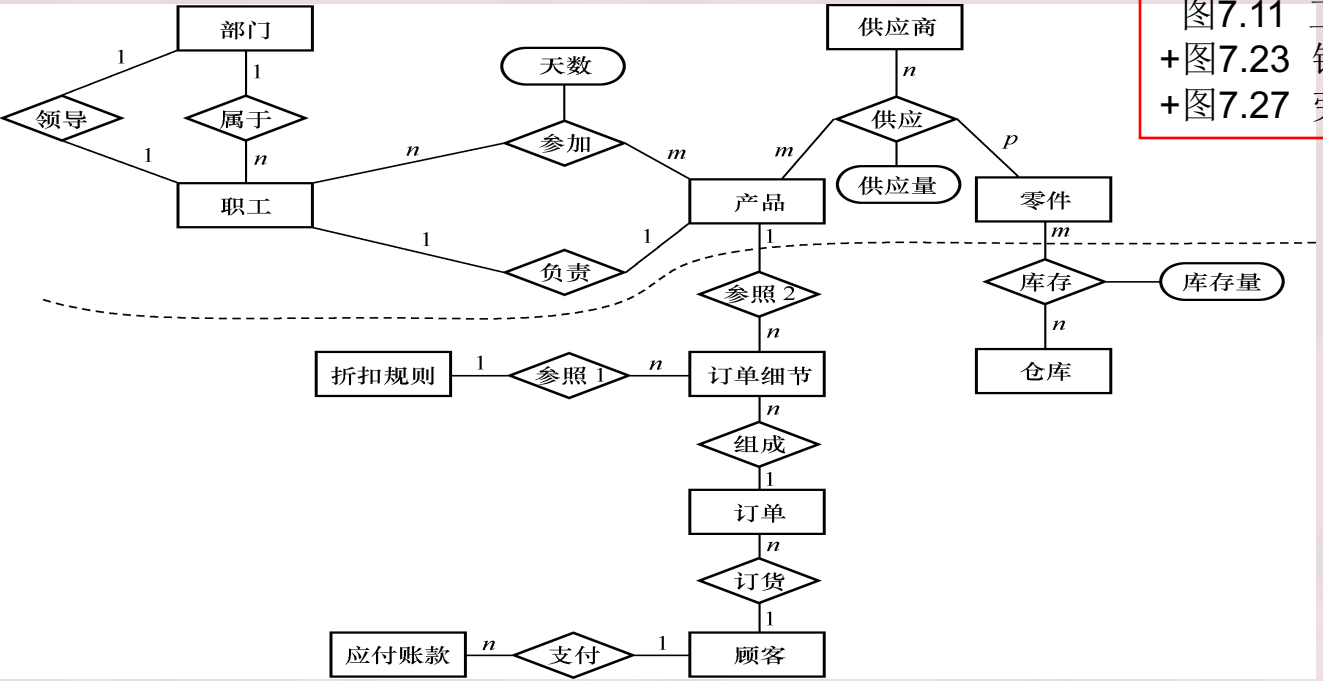


图7.28 某工厂信息系统的基本E-R图

设计 基本E-R图

[例7.2] 某工厂信息系统的E-R图集成

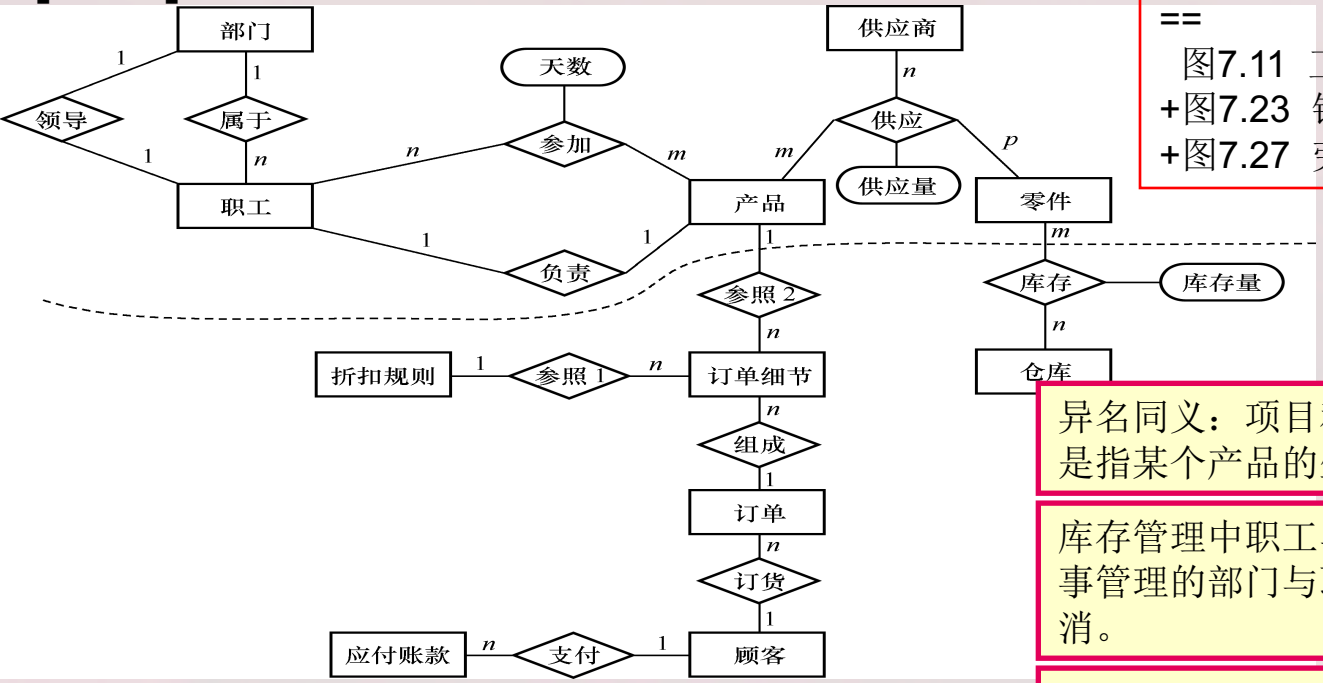


图7.28 某工厂信息系统的基本E-R图

图7.28 某工厂信息系统的基本E-R图

==

图7.11 工厂物资管理子系统的E-R图

+图7.23 销售管理子系统的E-R图

+图7.27 劳动人事管理子系统的E-R图

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。统一用产品作实体名。

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。

7.3 概念结构设计

7.3.1 概念结构

7.3.2 E-R模型

*7.3.3 扩展的E-R模型

*7.3.4 UML

7.3.5 概念结构设计

讲解：

1. 实体与属性的划分原则

（1）作为属性，不能再具有需要描述的性质。

（2）属性不能与其他实体具有联系。

2. E-R图的集成

- 设计各个子系统的分E-R图
- 消除冲突，进行集成
- 设计基本E-R图

7.4 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产品所支持的逻辑结构
- 目前主要使用关系模型，关系模型的逻辑结构是一组关系模式的集合

7.4 逻辑结构设计

7.4.1 E-R图向关系模型的转换

- 转换内容

将E-R图转换为关系模型：

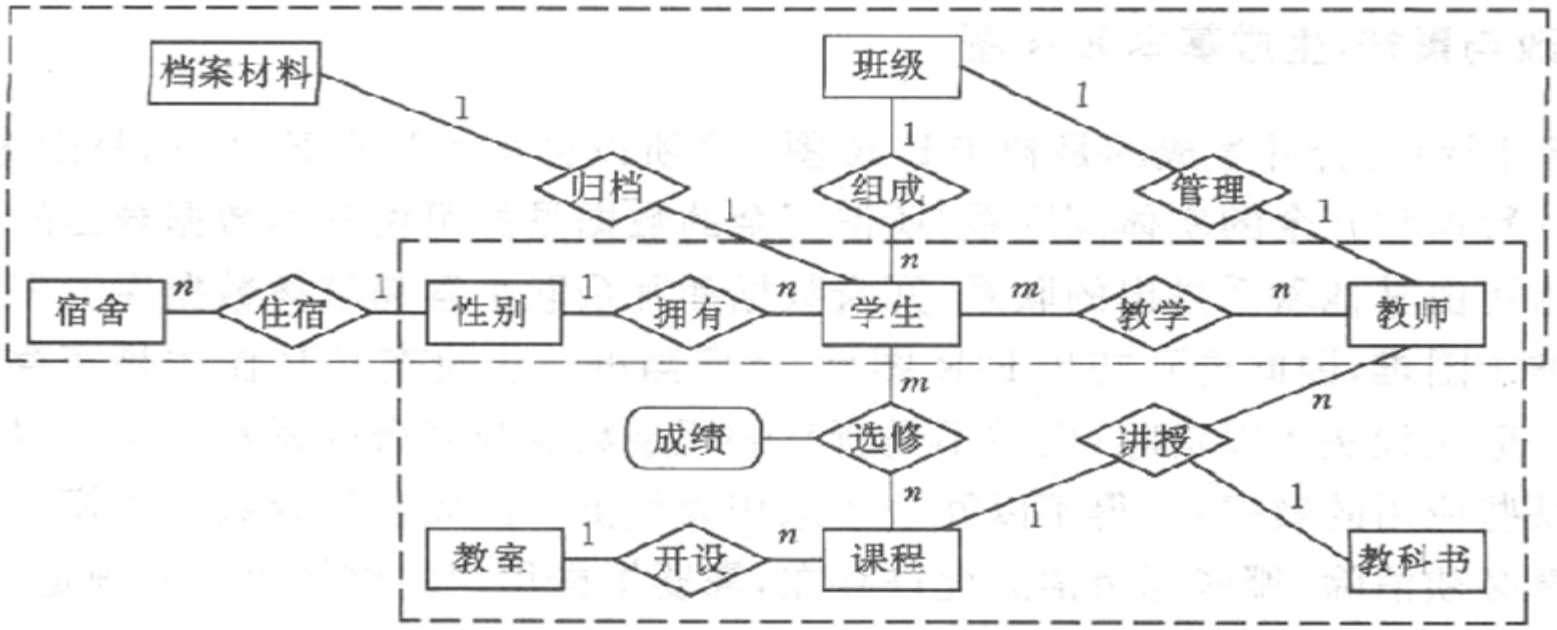
将实体型、实体的属性和实体型之间的联系转化为关系模式。

- 转换原则

7.4.2 数据模型的优化

7.4.3 设计用户子模式

E-R图向关系模型的转换原则

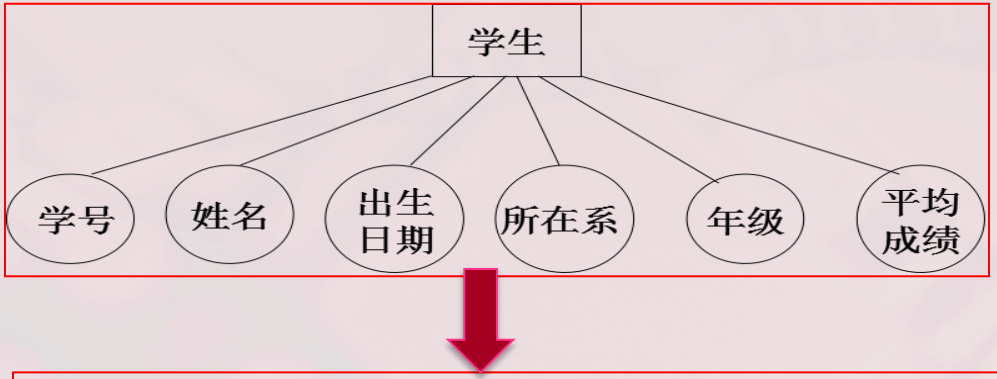


学生管理子系统基本E-R图

E-R图向关系模型的转换原则

1. 实体型的转换：一个实体型转换为一个关系模式

- 关系模式的属性：实体的属性
- 关系模式的码：实体的码



学生（学号，姓名，出生日期，所在系，年级，平均成绩）

E-R图向关系模型的转换原则

2. 实体型间的1:1联系：

■可以转换为一个独立的关系模式

- 关系模式的属性：与该联系相连的各实体的码以及联系本身的属性
- 关系模式的候选码：每个实体的码均是该关系模式的候选码

■也可以与相连的任意一端对应的关系模式合并

➢关系模式的属性：

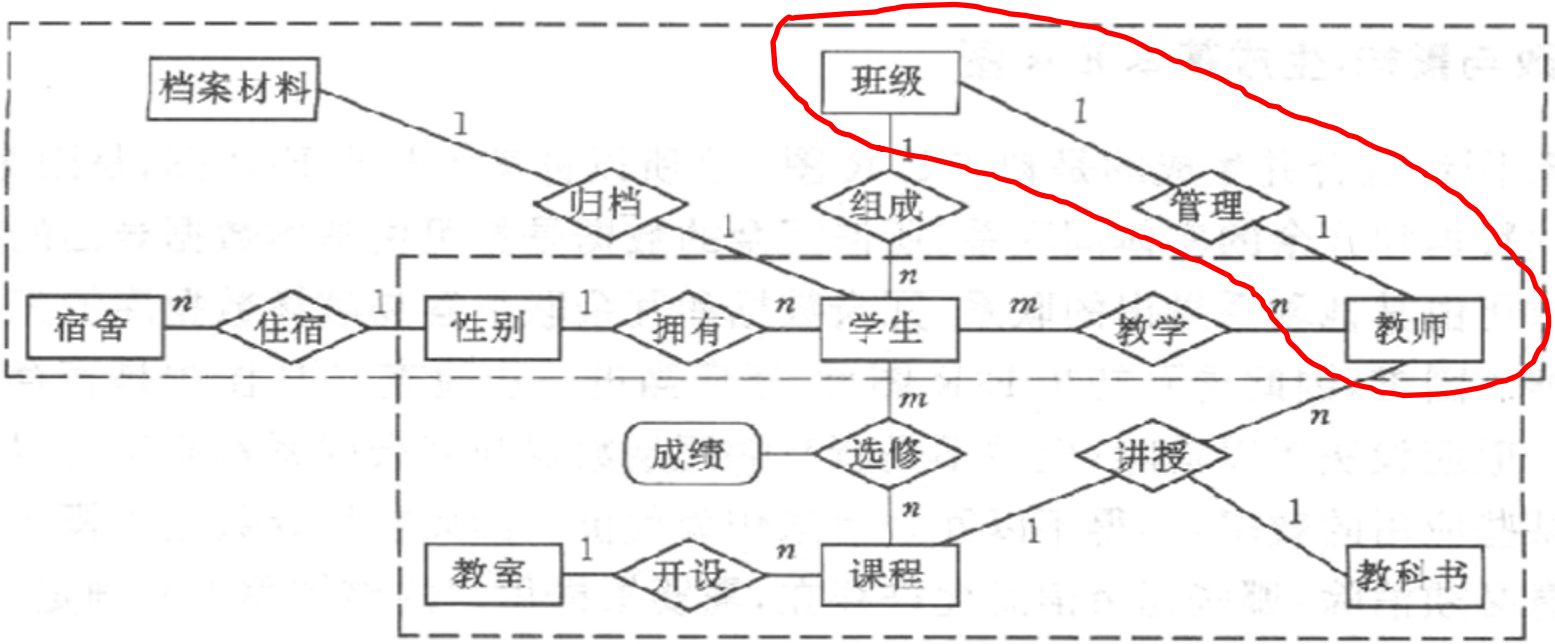
与某一端关系模式合并，则在该关系模式的属性中加入另一端关系模式的码和联系的属性

作为外码

➢合并后关系模式的码：不变

➢可以减少系统模式中的关系个数，一般情况下更倾向于采用这种方法

E-R图向关系模型的转换示例



学生管理子系统基本E-R图

E-R图向关系模型的转换示例

[例]“管理”联系为1:1联系：

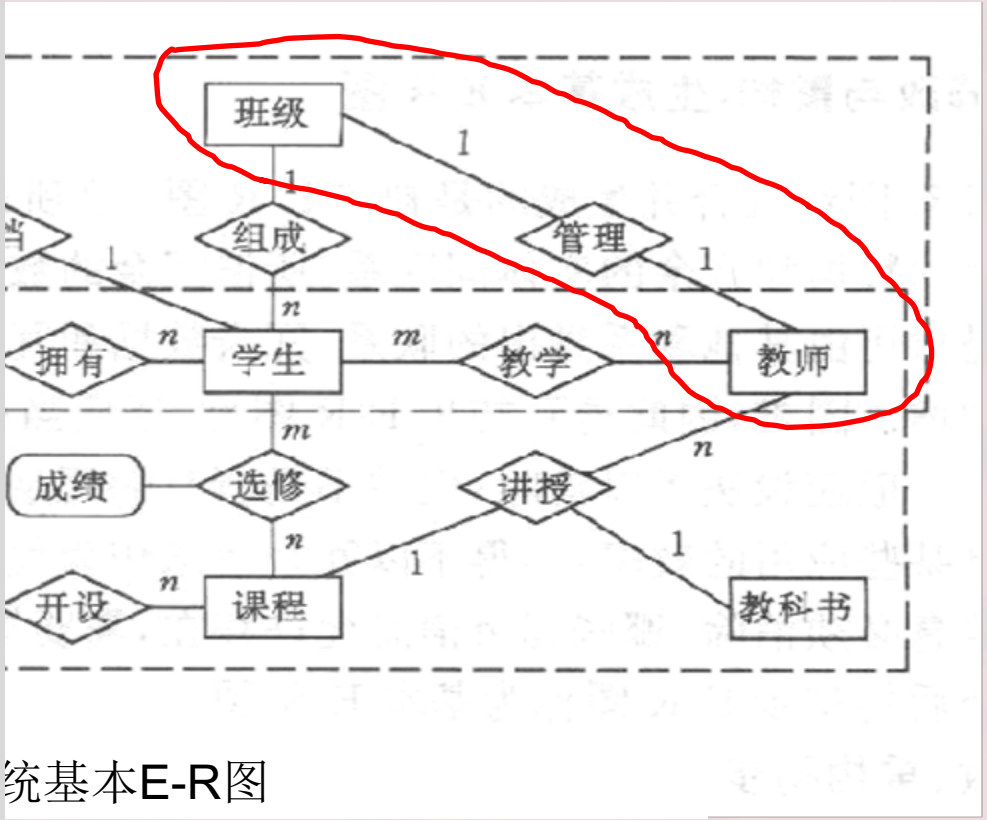
(1)转换为一个独立的关系模式：
管理(职工号，班级号) 或
管理(职工号，班级号)

(2)“管理”与“班级”关系模式合并：

班级(班级号，学生人数，职工号)
在“班级”中加入“教师”的码，即职工号

(3)“管理”与“教师”关系模式合并

教师(职工号，姓名，性别，职称，班级号，是否为优秀班主任)
在“教师”中加入“班级”的码，即班级号



统基本E-R图

E-R图向关系模型的转换示例

[例]“管理”联系为1:1联系：

(1)转换为一个独立的关系模式：
管理(职工号，班级号) 或
管理(职工号，班级号)

(2)“管理”与“班级”关系模式合并：

班级(班级号，学生人数，职工号)
在“班级”中加入“教师”的码，即职工号

(3)“管理”与“教师”关系模式合并

教师(职工号，姓名，性别，职称，班级号，是否为优秀班主任)
在“教师”中加入“班级”的码，即班级号

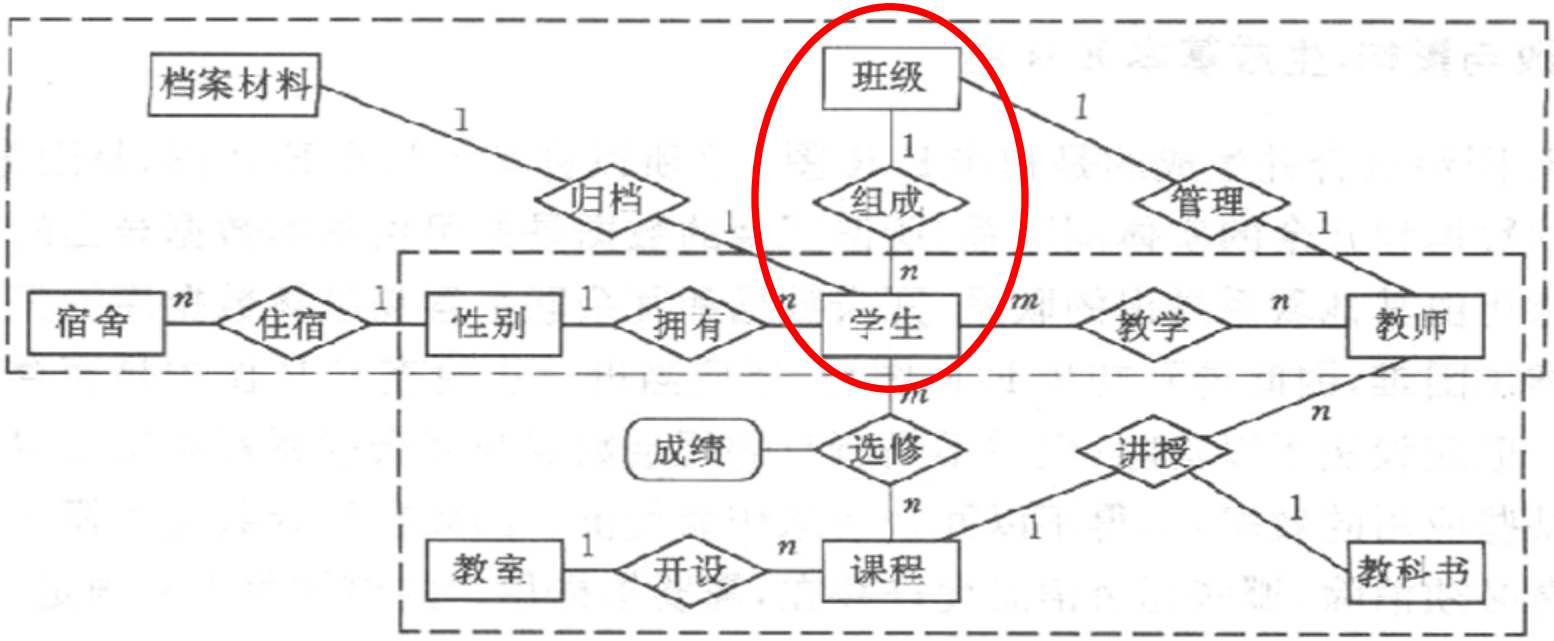
（1）转换为一个独立的关系模式

- 关系模式的属性：与该联系相连的各实体的码以及联系本身的属性
- 关系模式的候选码：每个实体的码均可以是该关系模式的候选码

（2）（3）与任意一端实体所对应的关系模式合并

- 合并后关系模式的属性：原模式属性+另一个关系模式的码（本关系的外码）+联系的属性
- 合并后关系模式的码：不变

E-R图向关系模型的转换原则



学生管理子系统基本E-R图

E-R图向关系模型的转换原则

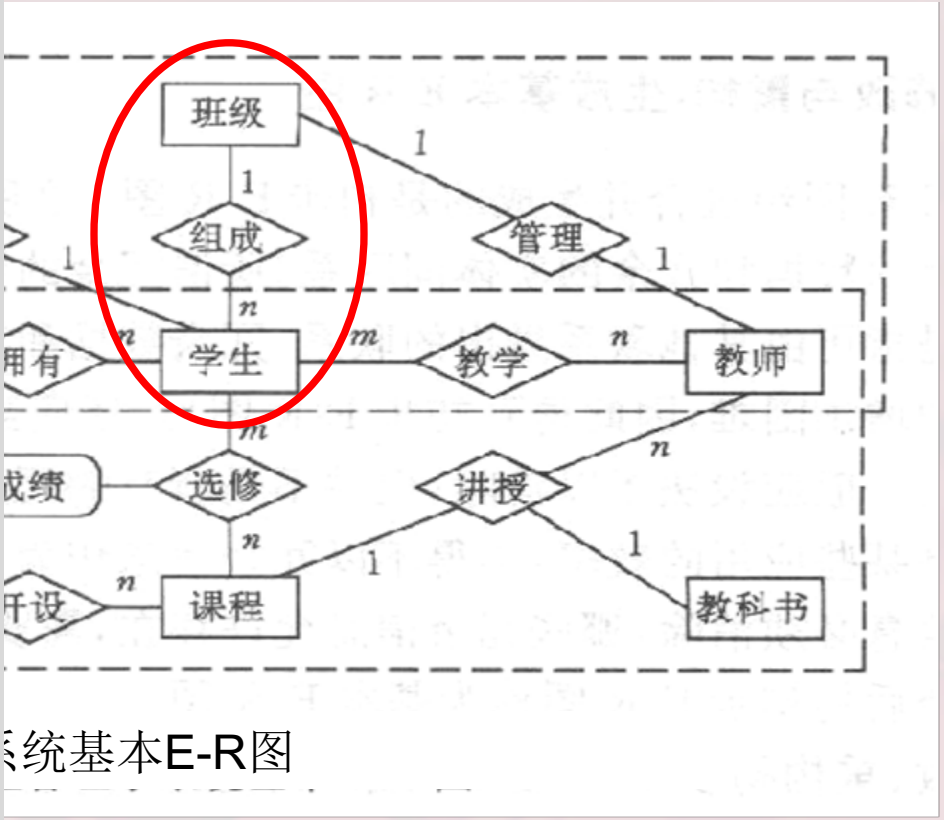
[例]“组成”联系为1:n联系。

将其转换为关系模式的两种方法：

(1)使其成为一个独立的关系模式：
组成（学号，班级号）

(2)将其与“学生”合并：
学生（学号，姓名，出生日期，
所在系，年级，班级号，
平均成绩）

作为
外码



系统基本E-R图

E-R图向关系模型的转换原则

3. 实体型间的1:n联系：

■转换为一个独立的关系模式

- 关系模式的属性：与该联系相连的各实体的码 + 联系本身的属性
- 关系模式的码：n端的 实体的码

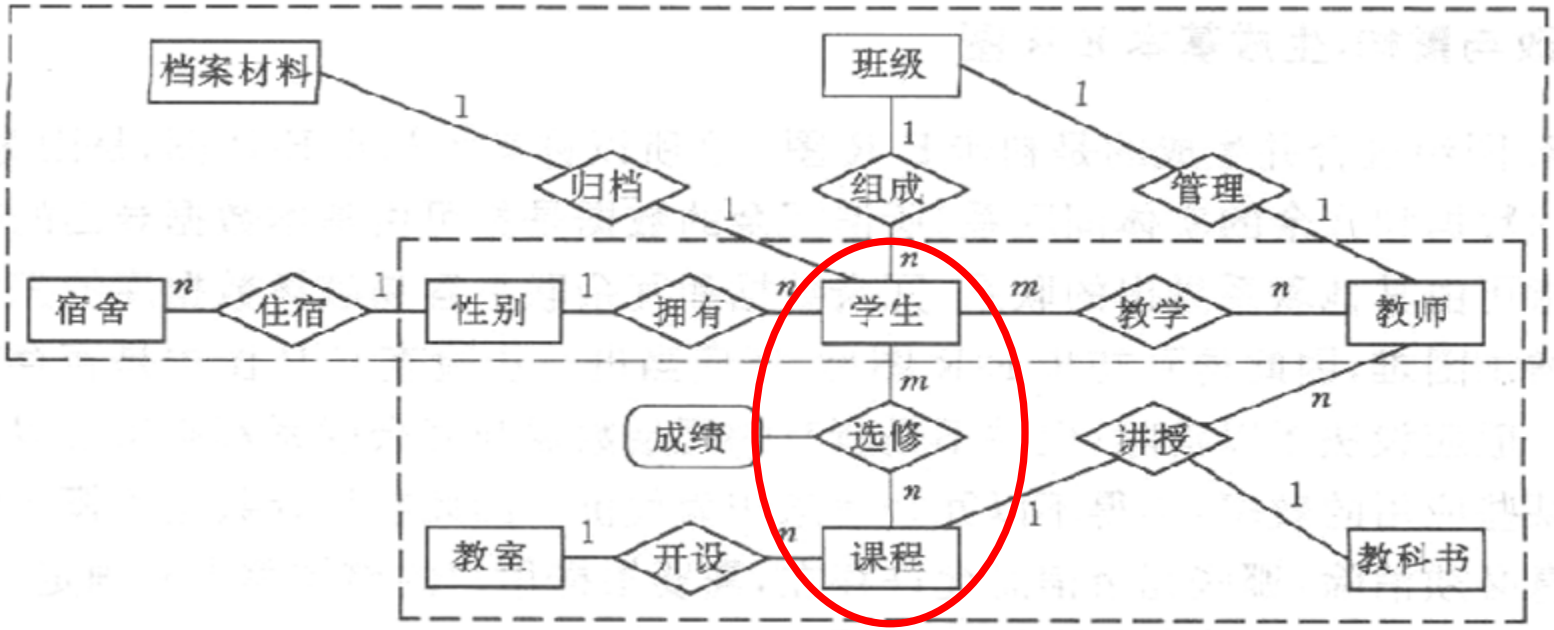
■与n端对应的关系模式合并

- 合并后关系模式的属性：在n端关系模式中 + 1端关系的码 + 联系本身的属性
- 合并后关系模式的码：不变

作为外码

可以减少系统模式中的关系个数，一般情况下更倾向于采用这种方法

E-R图向关系模型的转换（续）

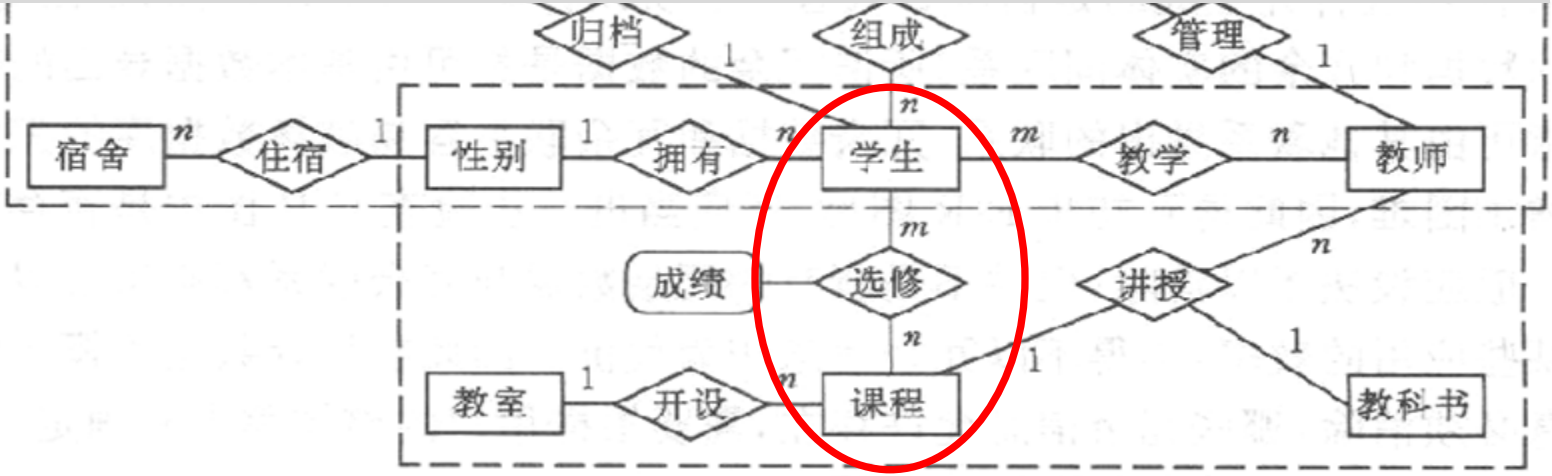


学生管理子系统基本E-R图

E-R图向关系模型的转换（续）

[例]“选修”联系是一个m:n联系，将它转换为:

选修（学号，课程号，成绩），其中学号与课程号为关系的组合码



学生管理子系统基本E-R图

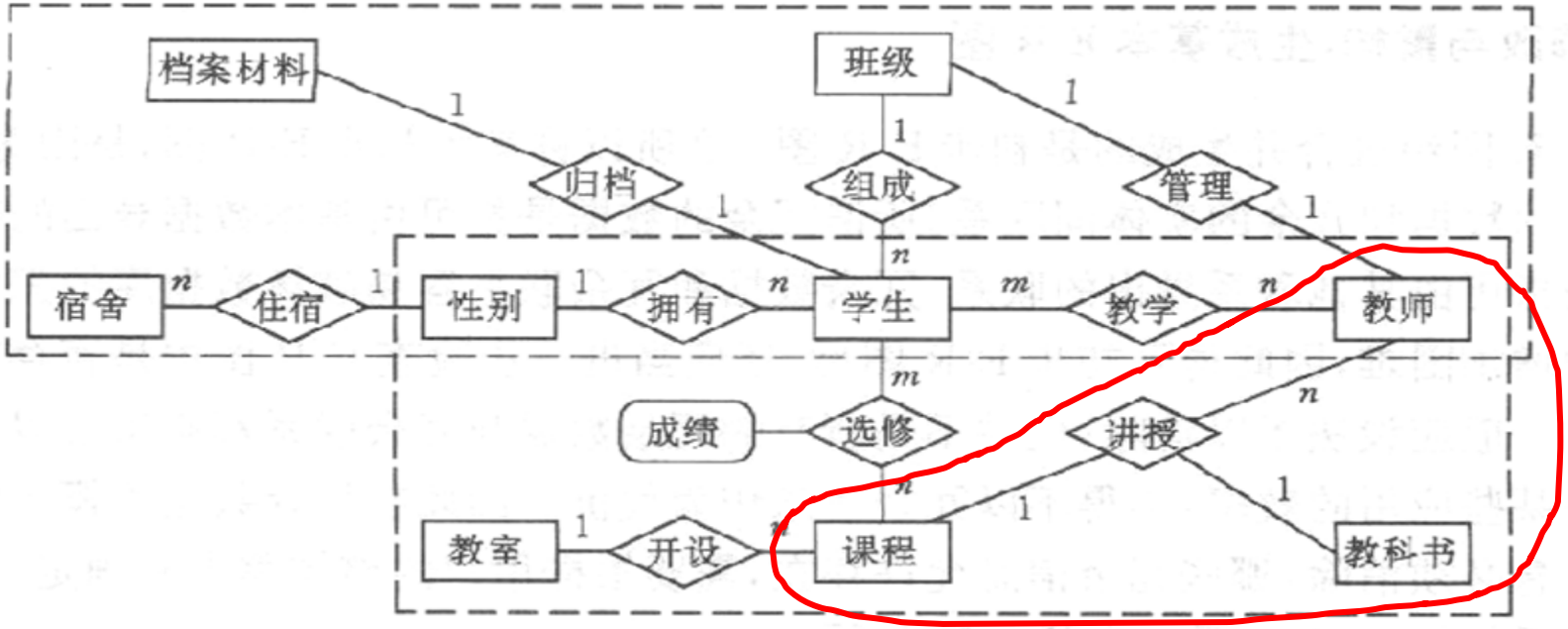
E-R图向关系模型的转换（续）

4. 实体型间的*m:n*联系：

■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（本关系的外码）以及联系本身的属性
- 关系的码：各实体码的组合

E-R图向关系模型的转换（续）



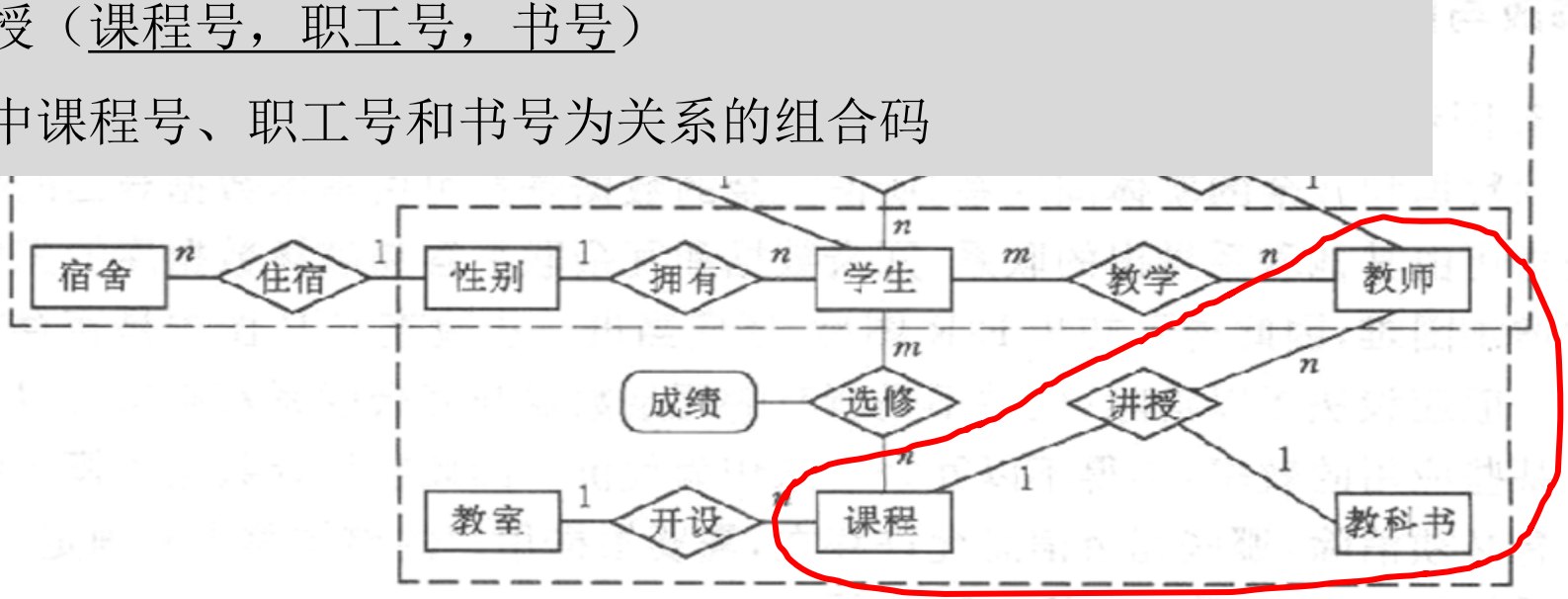
学生管理子系统基本E-R图

E-R图向关系模型的转换（续）

[例]“讲授”联系是一个三元联系，可以将它转换为：

讲授（课程号，职工号，书号）

其中课程号、职工号和书号为关系的组合码



学生管理子系统基本E-R图

E-R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

- 关系模式的属性：与该多元联系相连的各实体的码（分别作为外码） + 联系本身的属性
- 关系模式的码：各实体码的组合

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）
  - 适当调整属性的次序

E-R图向关系模型的转换（续）

按照上述转换原则，学生管理子系统中的实体（9）和联系（9）转换为下列关系模型：

学生（学号，姓名，性别，出生日期，所在系， 年级，班级号，平均成绩，档案号）

性别（性别，宿舍楼）

宿舍（宿舍编号，地址，性别，人数）

班级（班级号，学生人数）

教师（职工号，姓名，性别，职称，班级号，是否优秀班主任）

教学（职工号，学号）

课程（课程号，课程名，学分，教室号）

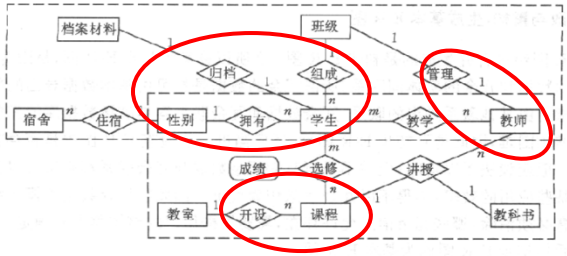
选修（学号，课程号，成绩）

教科书（书号，书名，价钱）

教室（教室编号，地址，容量）

讲授（课程号，教师号，书号）

档案材料（档案号，……）



学生管理子系统基本E-R图

该关系模型由12个关系模式组成。其中：

- 学生关系模式包含了“拥有”联系、“组成”联系、“归档”联系所对应的关系模式
- 教师关系模式包含了“管理”联系所对应的关系模式；
- 宿舍关系模式包含了“住宿”联系所对应的关系模式；
- 课程关系模式包含了“开设”联系所对应的关系模式。

7.4 逻辑结构设计

7.4.1 E-R图向关系模型的转换

7.4.2 数据模型的优化

7.4.3 设计用户子模式

7.4.2 数据模型的优化

- 数据库逻辑设计的结果不是唯一的。
- 得到初步数据据模式后，还应该适当地修改、调整数据库逻辑结构，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）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）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）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。

包括水平分解和垂直分解。

数据模型的优化（续）

几点注意

- 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来决定。
- 当查询经常涉及两个或多个关系模式的属性时，系统必须经常地进行连接运算，连接运算的代价是相当高的。这种情况下，需要降低规范化程度。
- 非BCNF的关系模式会存在不同程度的更新异常。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，就不会产生实际影响。（即结合应用环境的具体情况，合理地设计数据库模式。）

7.4 逻辑结构设计

7.4.1 E-R图向关系模型的转换

7.4.2 数据模型的优化

7.4.3 设计用户子模式

7.4.3 设计用户子模式

- 数据库模式——全局模式。

  考虑系统全局应用需求，时间效率、空间效率、易维护等。

- 用户子模式——视图机制

  考虑局部应用的特殊需求和用户体验。

（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。
- 在设计用户子模式时可以设计子模式时重新定义某些属性名，使其与用户习惯一致，以方便使用。

设计用户子模式（续）

（2）针对不同级别的用户定义不同的视图，提高系统的安全性

假设有关系模式:

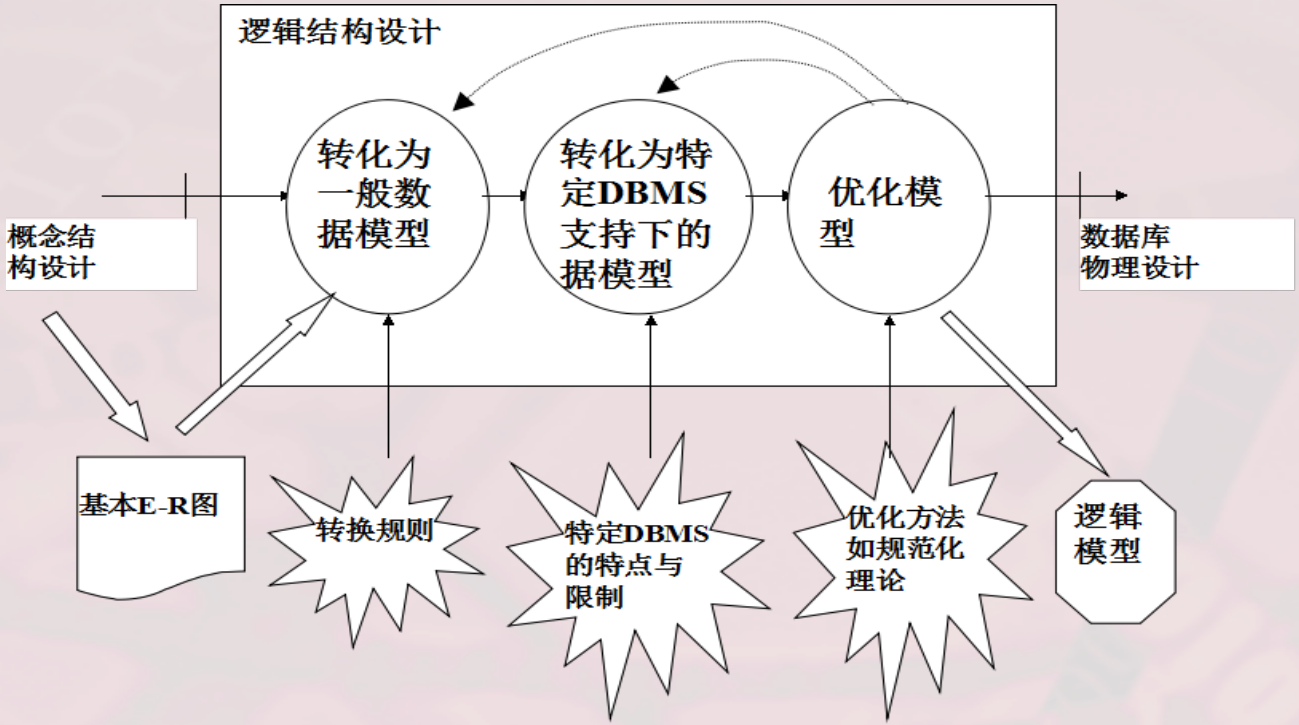
产品（产品号，产品名，规格，单价，生产车间，生产负责人，产品成本，产品合格率，质量等级）

为一般顾客、为产品销售部门和管理部门建立不同的视图。

（3）简化用户对系统的使用

某些局部应用中经常要使用一些很复杂的查询，为了方便用户，可以将这些复杂查询定义为视图。

7.4 逻辑结构设计

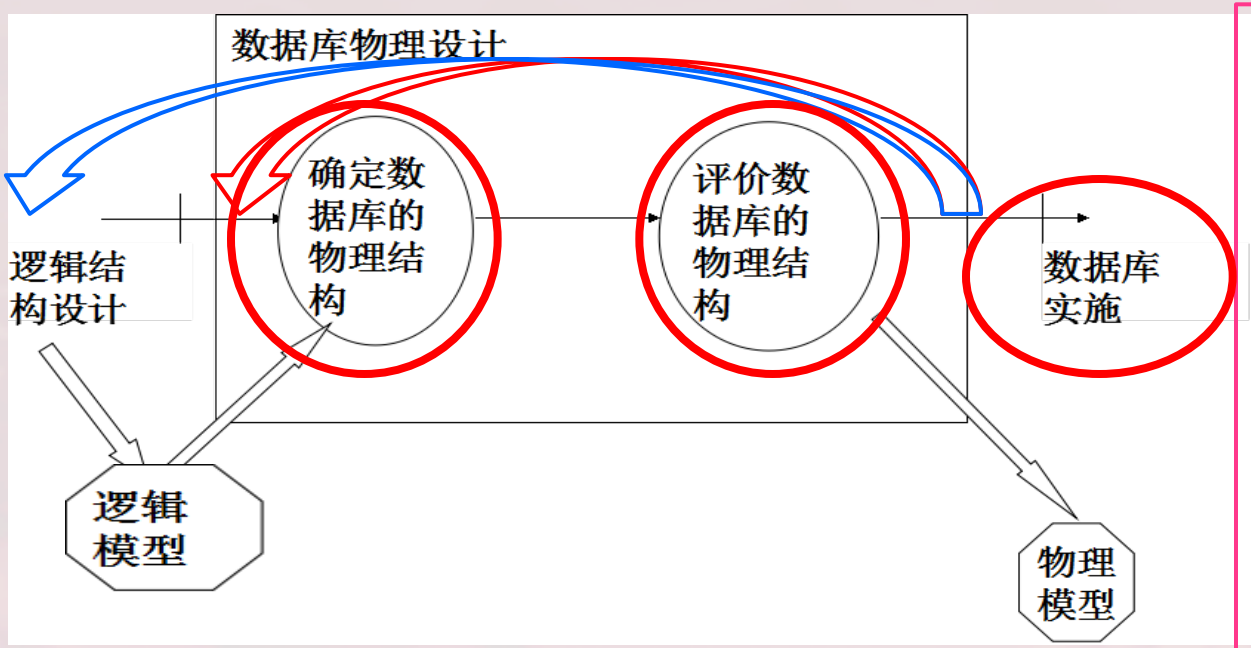

逻辑结构设计
转化为一般数据模型
转化为特定DBMS支持下的据模型
优化模型
概念结构设计
数据库物理设计
基本E-R图
转换规则
特定DBMS的特点与限制
优化方法如规范化理论
逻辑模型

7.5 数据库的物理设计

❖ 什么是数据库的物理设计

- 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DBMS。

数据库物理设计的步骤



◆ 确定数据库的物理结构

RDBMS中主要指存取方法和存储结构;

◆对物理结构进行评价

重点是时间和空间效率

IF 满足原设计要求

THEN 进入到物理实施阶段

ELSE

(重新设计

OR 修改物理结构

OR 返回逻辑设计阶段

修改数据模型)

7.5.1 数据库物理设计的内容和方法

❖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）

➢为关系、索引、日志、备份等数据库文件选择物理存储结构

7.5.1 数据库物理设计的内容和方法

不同的DBMS产品

- 物理环境、存取方法和存储结构有很大差别
- 能供设计人员使用的设计变量、参数范围很不相同
- 没有通用的物理设计方法，只能给出**一般的设计内容和原则。**

设计物理数据库结构的准备工作

- 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。
  - 有哪些索引（例如B+树索引，HASH索引，BITMAP索引等），如何建立索引；
  - 有哪些存储结构（行存储，列存储，聚簇存储），如何选择。

7.5 数据库的物理设计

7.5.1 数据库物理设计的内容和方法

7.5.2 关系模式存取方法选择

索引方法、聚簇方法

7.5.3 确定数据库的存储结构

7.5.4 评价物理结构

索引存取方法的选择

❖ 为什么要建立索引

提高存取的效率——查询、插入、删除、更新的效率

❖ 如何选择索引存取方法

根据应用要求确定：

- 对哪些属性列建立索引
- 对哪些索引要设计为**唯一索引、组合索引、**
- 选择合适的索引方法

索引存取方法的选择

- **如何创建索引**

  CREATE [UNIQUE] INDEX 索引名字 ON 表名 [USING 索引方法] （列名1，列名2，[, ...]);

  CREATE UNIQUE INDEX studentname ON student USING Hash (sname）;

- **RDBMS提供的索引方法：**
  - B-tree、B+树、hash（散列）、R-tree 、Bitmap等。
  - 如果不指定，缺省一般是B-tree。

索引存取方法的选择

❖ 选择索引存取方法的一般规则

- 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（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

索引存取方法的选择（续）

- 索引带来的额外开销
  - 维护索引的开销
  - 查找索引的开销
  - 存储索引的开销
- 确定是否需要建立索引，选择哪种索引

3. 聚簇存取方法的选择

❖什么是聚簇

- 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）
- 许多RDBMS都提供了聚簇功能

聚簇存取方法的选择

❖聚簇的用途

1. 大大提高按聚簇属性进行查询的效率

[例] 假设要查询计算机系的所有学生。

- 学生数据表随机存放，计算机系的500名学生分散存储在500个不同的物理块上，则至少要执行500次 I/O操作。
- 如果按照专业系名**聚簇存放**，将同一系的学生元组聚簇在一起存放，则可以显著地减少了访问磁盘的次数。计算机系的500名学生**聚簇**存储在50个不同的物理块上，只要执行50次 I/O操作。

聚簇存取方法的选择（续）

❖聚簇的适用范围

➢既适用于单个关系独立聚簇，也适用于多个关系组合聚簇

[例] 假设用户经常要按姓名查询学生成绩单。

SELECT sname, cno, grade from student, sc where student.sno=sc.sno

这一查询涉及学生关系和选修关系的连接操作，按学号连接这两个关系。

■按照学号把学生表和选修表聚簇在一起。

■相当于把多个关系按“预连接”的形式存放。

■大大提高连接操作的效率。

聚簇存取方法的选择（续）

❖聚簇的适用范围

➢当SQL语句中包含有与聚簇码有关的ORDER BY, GROUP BY, UNION, DISTINCT等子句或短语时，使用聚簇特别有利，可以省去或减少对结果集的排序操作

聚簇存取方法的选择（续）

❖聚簇的局限性

- ➢在一个基本表上最多只能建立一个聚簇索引
- ➢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相应改变。

聚簇存取方法的选择（续）

- 聚簇索引的适用条件
  - 很少对基表进行增删操作
  - 很少对其中的变长列进行修改操作

7.5 数据库的物理设计

7.5.1 数据库物理设计的内容和方法

7.5.2 关系模式存取方法选择

7.5.3 确定数据库的存储结构

7.5.4 评价物理结构

7.5.3 确定数据库的存储结构

- 确定数据的存储安排和存储结构

  - 关系
  - 索引
  - 数据库缓冲区
  - 日志
  - 备份

  - 内存/磁盘
  - 行存储/列存储
  - 集中/分散 存放
  - 顺序/随机/聚簇 存放

- 确定系统参数配置

各个系统所能提供的对数据进行物理安排的手段、方法差异很大；
设计人员应仔细了解给定的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。

7.5 数据库的物理设计

7.5.1 数据库物理设计的内容和方法

7.5.2 关系模式存取方法选择

7.5.3 确定数据库的存储结构

7.5.4 评价物理结构

7.5.4 评价物理结构

- 对数据库物理设计过程中产生的多种方案进行评价，从中选择一个较优的方案作为数据库的物理结构。
- 评价方法
  - 定量估算各种方案
    - 存储空间
    - 存取时间
    - 维护代价
  - 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
  - 返回用户 征求意见 修改设计

7.6 数据库的实施和维护(自学）

7.6.1 数据的载入和应用程序的调试

7.6.2 数据库的试运行

7.6.3 数据库的运行和维护

7.7 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
  - 物理结构设计
  - 数据库实施
  - 数据库运行维护
  - 设计过程中往往还会有许多反复

小结（续）

- 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。

小结（续）

- 概念结构设计
  - E-R模型的基本概念和图示方法
  - E-R模型的设计
  - 把E-R模型转换为关系模型的方法

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，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

课后作业

在校田径运动会中设置了各类比赛，每一比赛类别有类别编号、类别名称和主管等属性，每一比赛类别包含很多比赛项目；每一比赛项目有项目编号、项目名称、比赛时间和级别等属性；各个系团队有团编号、团名称、领队等属性，每一代表团有多名运动员组成，运动员有编号，姓名，年龄，性别等属性；每一名运动员可以参加多个比赛项目，每一比赛项目也有多名运动员参加，运动员参加比赛有成绩属性，成绩限定在0~7分。

1) 根据上述语义画出ER图，

2) 将ER图转换成关系模式，并指出每个关系模式的主键和外键。