



数据库系统概论

第7章 数据库设计

- ❖ 第一节 数据库设计概述
- ❖ 第二节 需求分析
- ❖ 第三节 概念结构设计
- ❖ 第四节 逻辑结构设计
- ❖ 第五节 数据库的物理设计
- ❖ 第六节 数据库实施和维护



❖ 掌握

- ◆ 概念结构设计、逻辑结构设计

❖ 了解

- ◆ 需求分析、物理设计、运行维护

❖ 重点

- ◆ 概念结构设计、逻辑结构设计

❖ 难点

- ◆ 概念结构向逻辑结构转换



- ❖ 第一节 数据库设计概述
- ❖ 第二节 需求分析
- ❖ 第三节 概念结构设计
- ❖ 第四节 逻辑结构设计
- ❖ 第五节 数据库的物理设计
- ❖ 第六节 数据库实施和维护



❖ 数据库设计

- ◆ 数据库设计是指对于一个给定的应用环境，构造（设计）优化的数据库逻辑模式和物理结构，并据此建立数据库及其应用系统，使之能够有效地存储和管理数据，满足各种用户的应用需求，包括信息管理要求和数据操作要求
- ◆ 目标：为用户和各种应用系统提供一个信息基础设施和高效率的运行环境



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



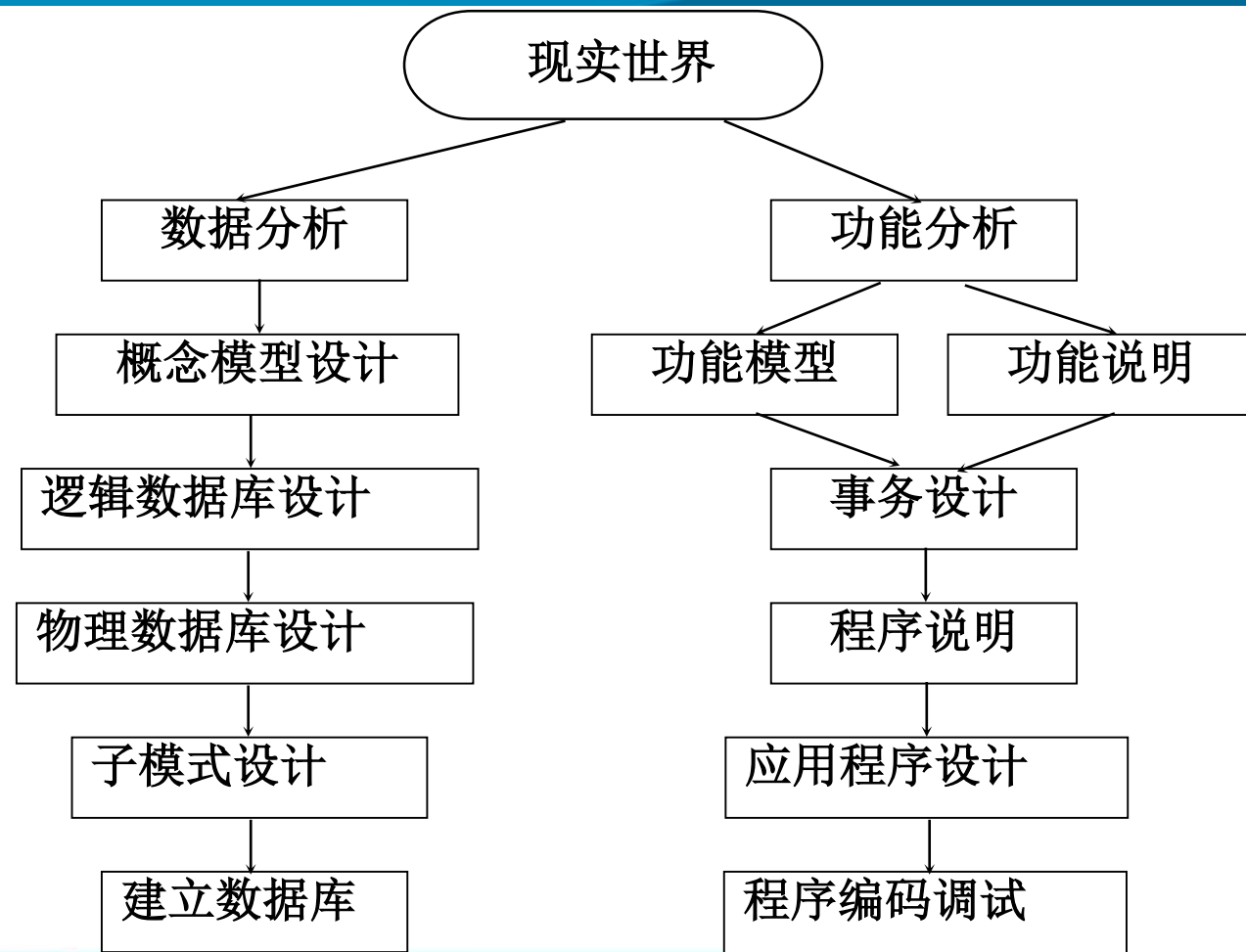
❖ 数据库建设的基本规律

- ◆ 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

应用系统设计



❖ 手工与经验相结合方法

❖ 规范设计法

- ◆ 基本思想：过程迭代和逐步求精

❖ 常见的设计方法

- ◆ 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
- ◆ ODL (Object Definition Language) 方法



❖ 数据库设计分6个阶段

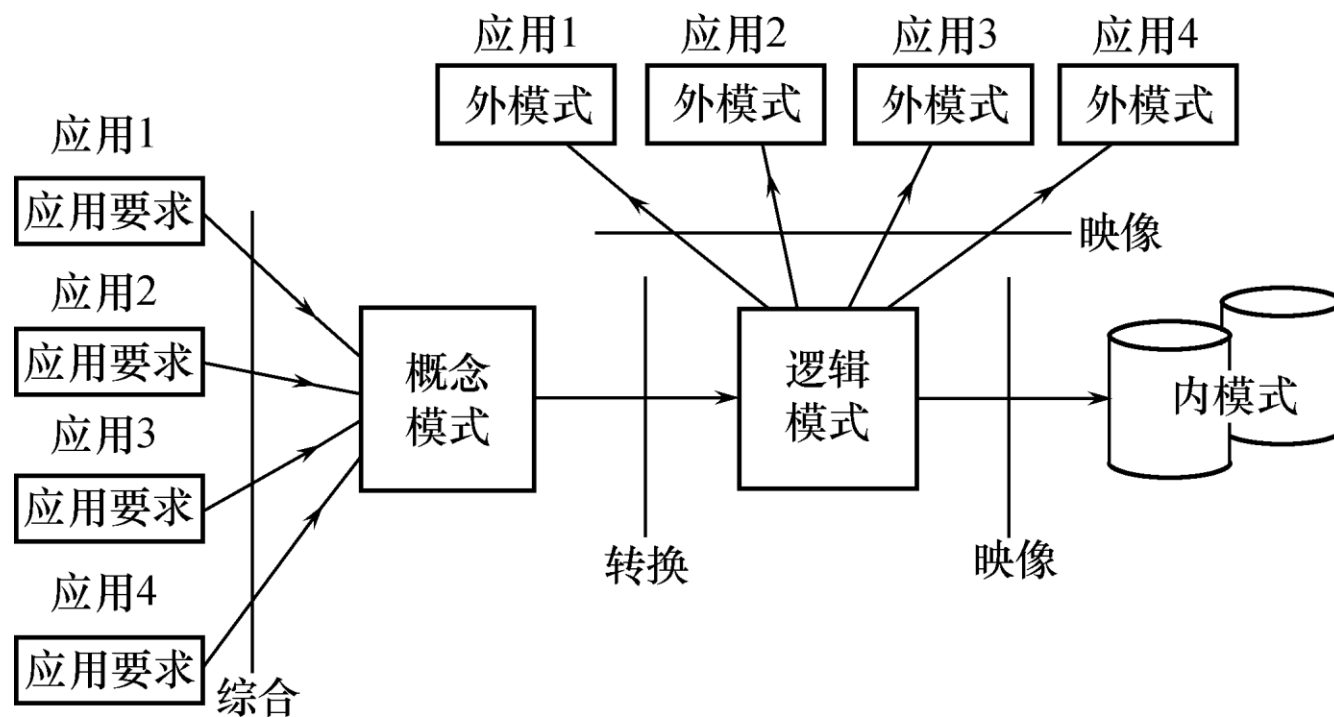
- ◆ 需求分析
- ◆ 概念结构设计
- ◆ 逻辑结构设计
- ◆ 物理结构设计
- ◆ 数据库实施
- ◆ 数据库运行和维护

❖ 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



❖ 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



- ❖ 第一节 数据库设计概述
- ❖ 第二节 需求分析
- ❖ 第三节 概念结构设计
- ❖ 第四节 逻辑结构设计
- ❖ 第五节 数据库的物理设计
- ❖ 第六节 数据库实施和维护



需求分析的任务

需求分析的方法

数据字典

- ❖ 详细调查现实世界要处理的对象
- ❖ 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



❖ 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



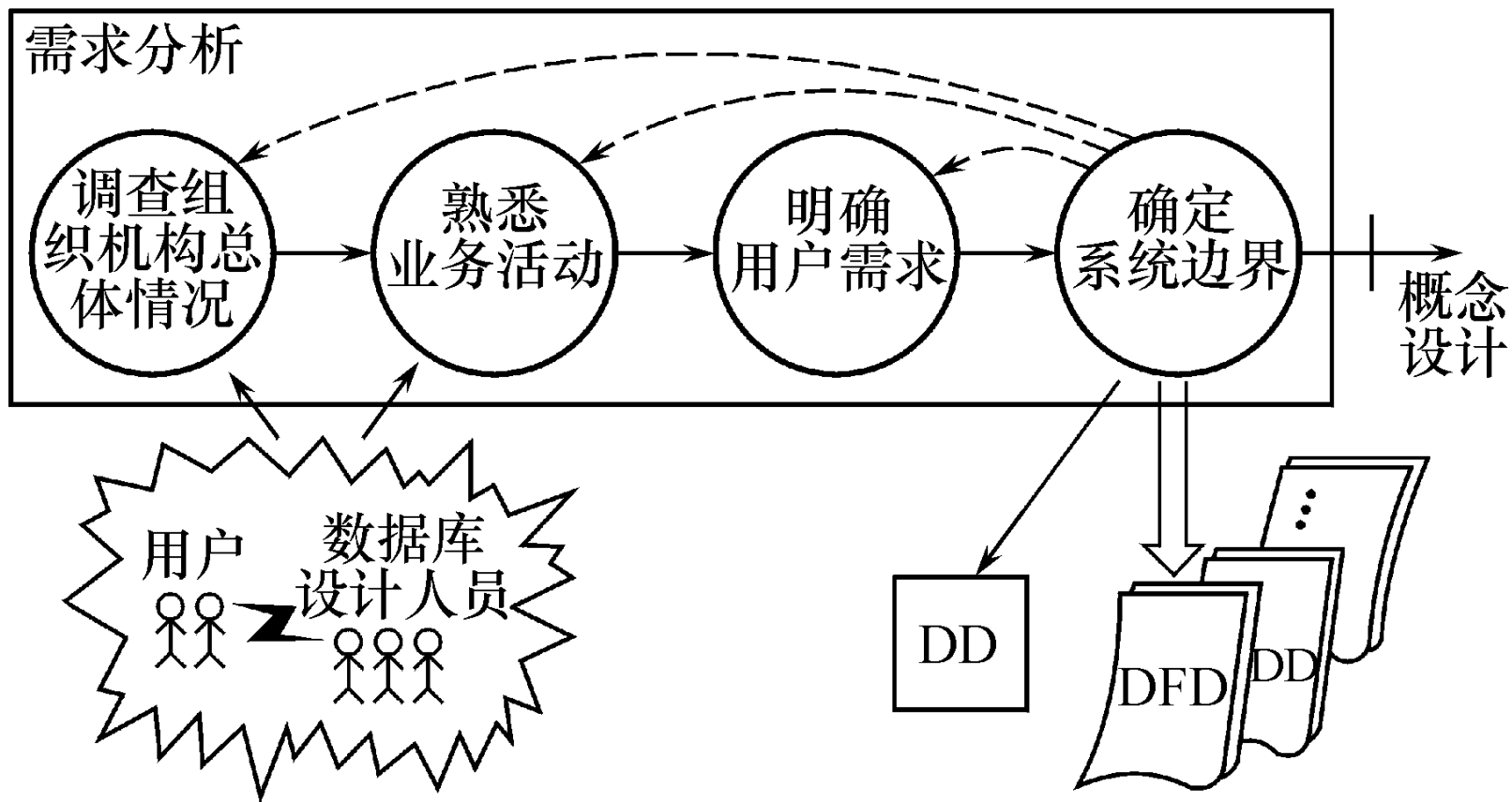
❖ 调查用户需求，与用户达成共识，然后分析表达这些需求

❖ 调查用户需求的具体步骤

- ◆ 调查组织机构情况
- ◆ 调查各部门的业务活动情况
- ◆ 在熟悉业务活动的基础上，协助用户明确对新系统的各种要求
- ◆ 确定新系统的边界



需求分析过程



❖ 数据字典的用途

- ◆ 进行详细的数据收集和数据分析所获得的主要结果

❖ 数据字典的内容

- ◆ 数据项
- ◆ 数据结构
- ◆ 数据流
- ◆ 数据存储
- ◆ 处理过程



- ❖ 第一节 数据库设计概述
- ❖ 第二节 需求分析
- ❖ 第三节 概念结构设计
- ❖ 第四节 逻辑结构设计
- ❖ 第五节 数据库的物理设计
- ❖ 第六节 数据库实施和维护



概念结构设计概述

概念结构设计的方法与步骤

数据抽象与局部视图设计

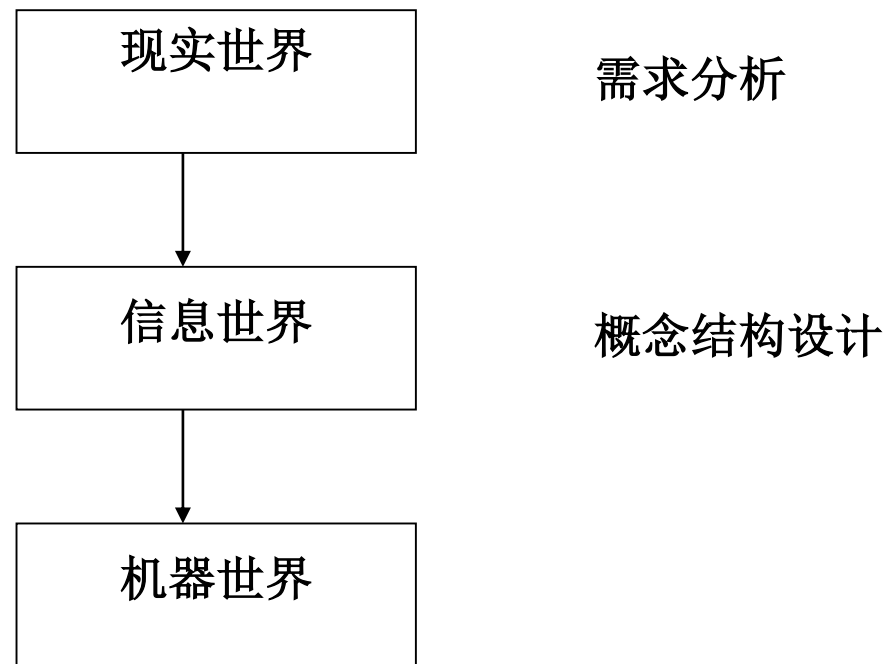
视图的集成

❖什么是概念结构设计

- ◆ 需求分析阶段描述的用户应用需求是现实世界的具体需求
- ◆ 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
- ◆ 概念结构设计是整个数据库设计的**关键**

❖ 概念结构设计的特点

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



❖ 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

❖ 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
- ◆ 例如：(李明, 男, 1972, 江苏, 计算机系, 1990)

❖ 码 (Key)

- ◆ 唯一标识实体的属性集称为码



❖ 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内部的联系和实体之间的联系
 - ◆ 根据联系涉及的实体数量可分为
 - 一个实体型
 - 多个实体型
 - 两个实体型
- 一对一联系 (1:1)
一对多联系 (1:n)
多对多联系 (m:n)



两个实体型之间的联系

❖ 一对一联系 (1:1)

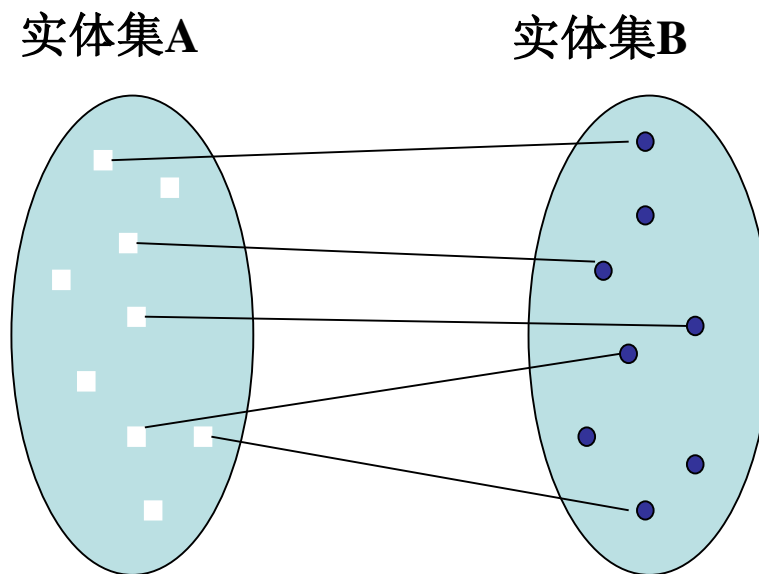
◆ 定义

如果对于实体集A中的每一个实体，实体集B中至多有一个（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

◆ 实例

一个班级只有一个正班长

一个班长只在一个班中任职



1: 1联系

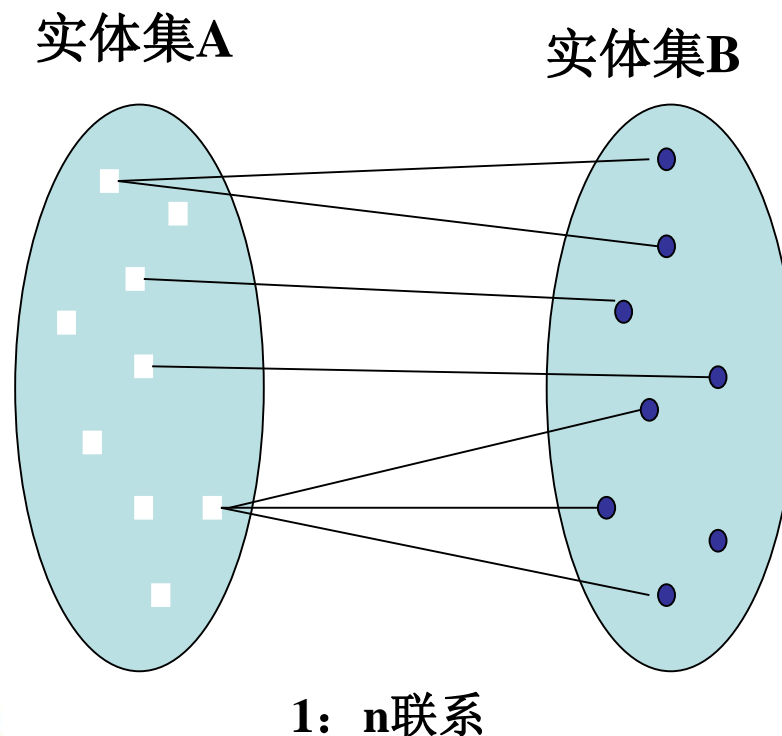
❖ 一对多联系 (1: n)

◆ 定义

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 ($n \geq 0$) 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

◆ 实例

一个班级中有若干名学生，
每个学生只在一个班级中学习



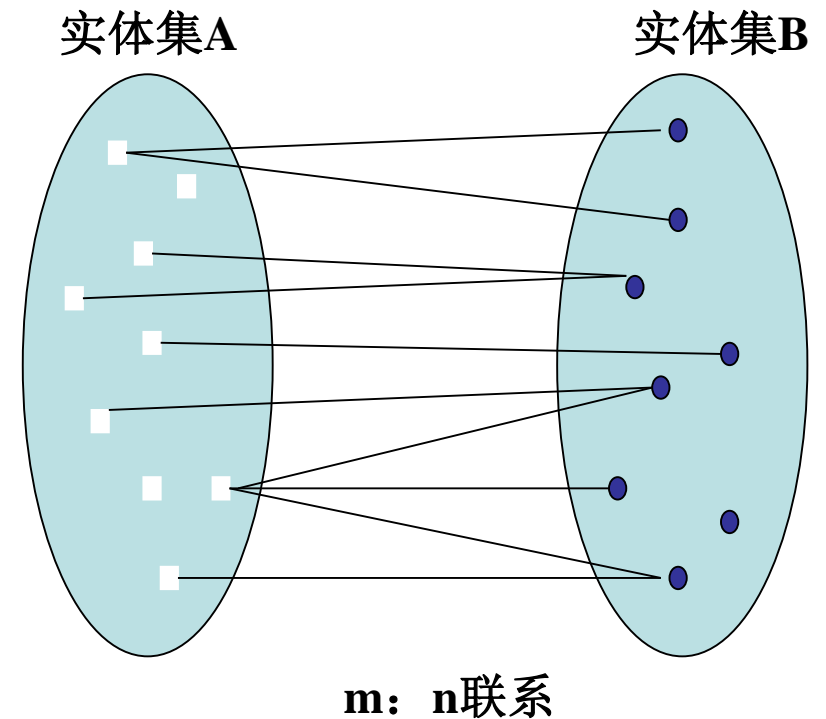
❖ 多对多联系 (m:n)

◆ 定义

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 ($n \geq 0$) 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 ($m \geq 0$) 与之联系，则称实体集A与实体B具有多对多联系，记为 $m:n$

◆ 实例

一门课程同时有若干个学生选修
一个学生可以同时选修多门课程



❖ E-R图

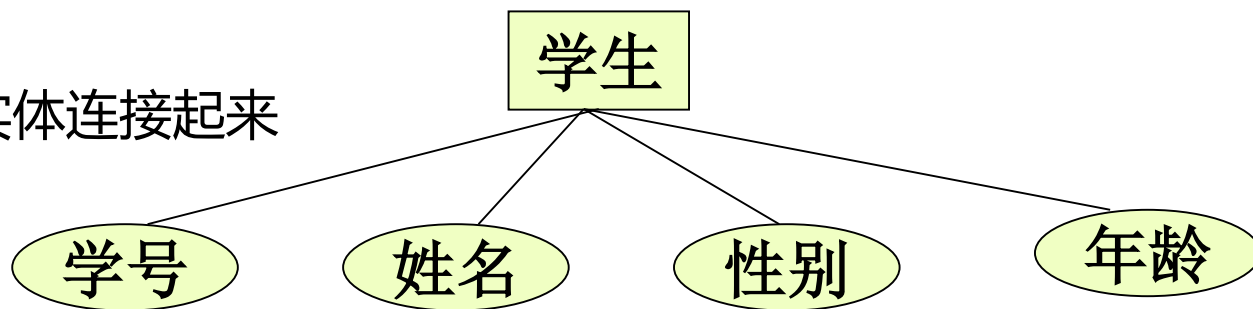
◆ 实体型

- 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连接起来



❖ 联系

◆ 联系本身：

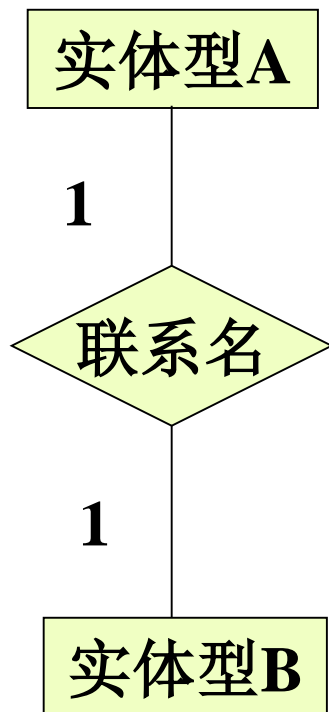
用菱形表示，菱形框内写明联系名，并用无向边分别与有关实体连接起来，同时
在无向边旁标上联系的类型（1:1、1:n或m:n）

◆ 联系的属性：

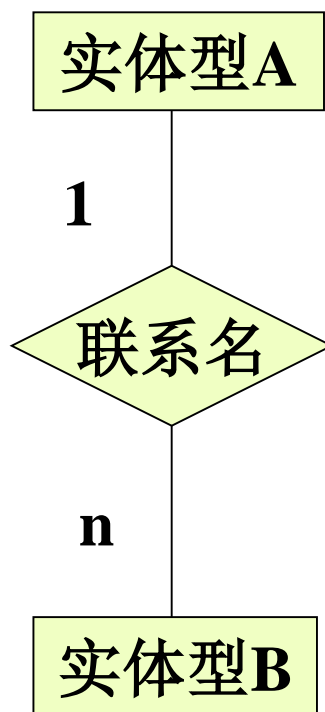
联系本身也是一种实体型，也可以有属性。如果一个联系具有属性，则这些属性
也要用无向边与该联系连接起来



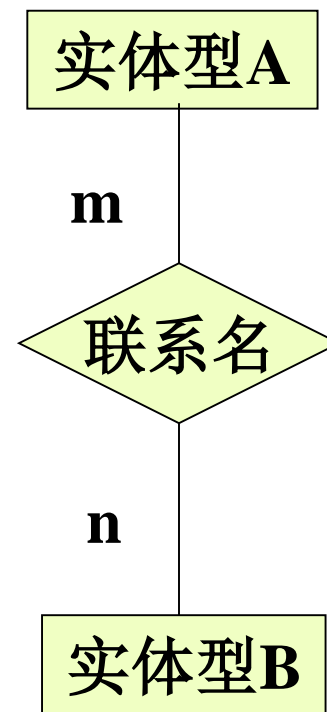
联系的表示方法



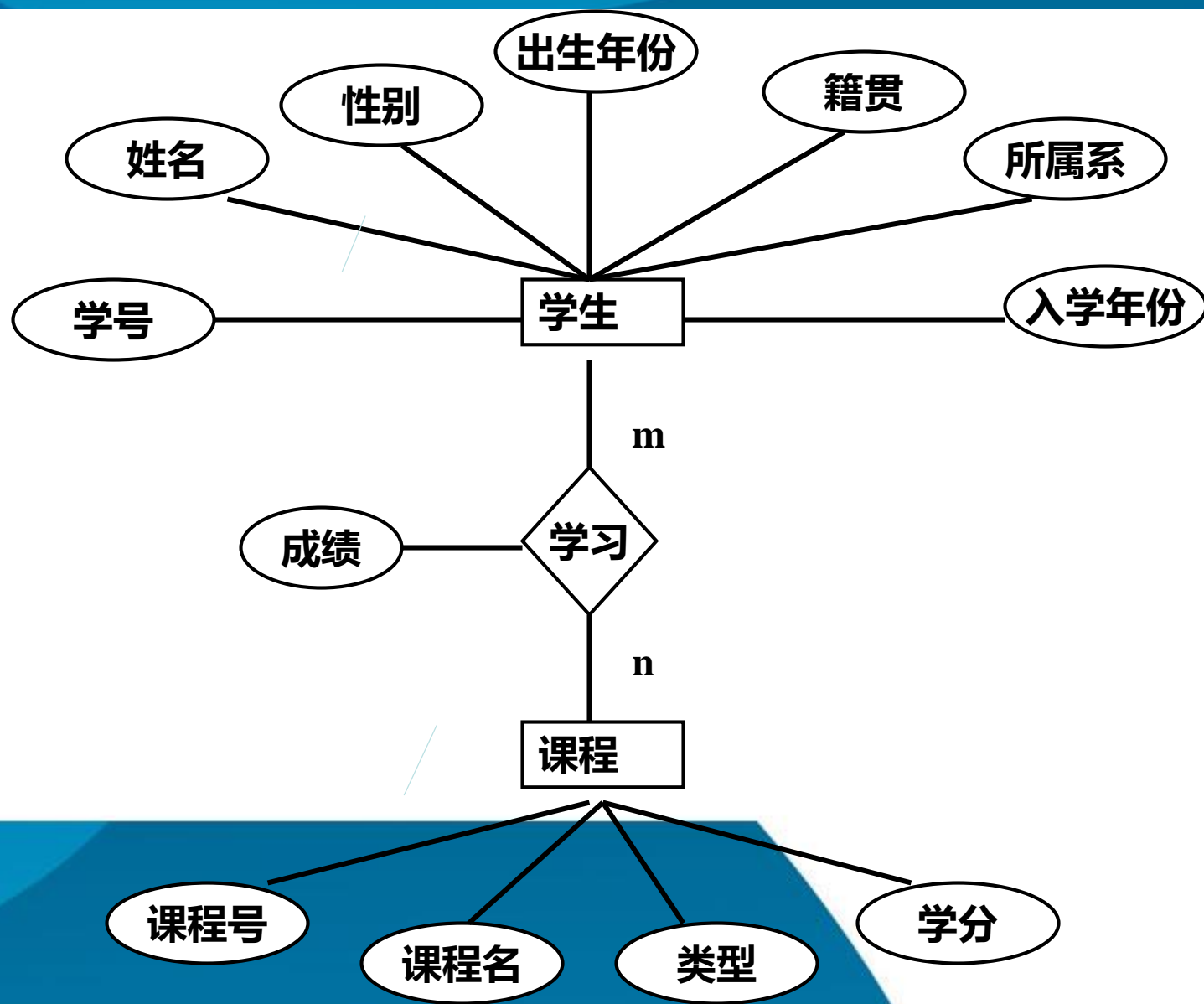
1:1联系



1:n联系



m:n联系



❖ 根据描述画出学校管理的E-R图

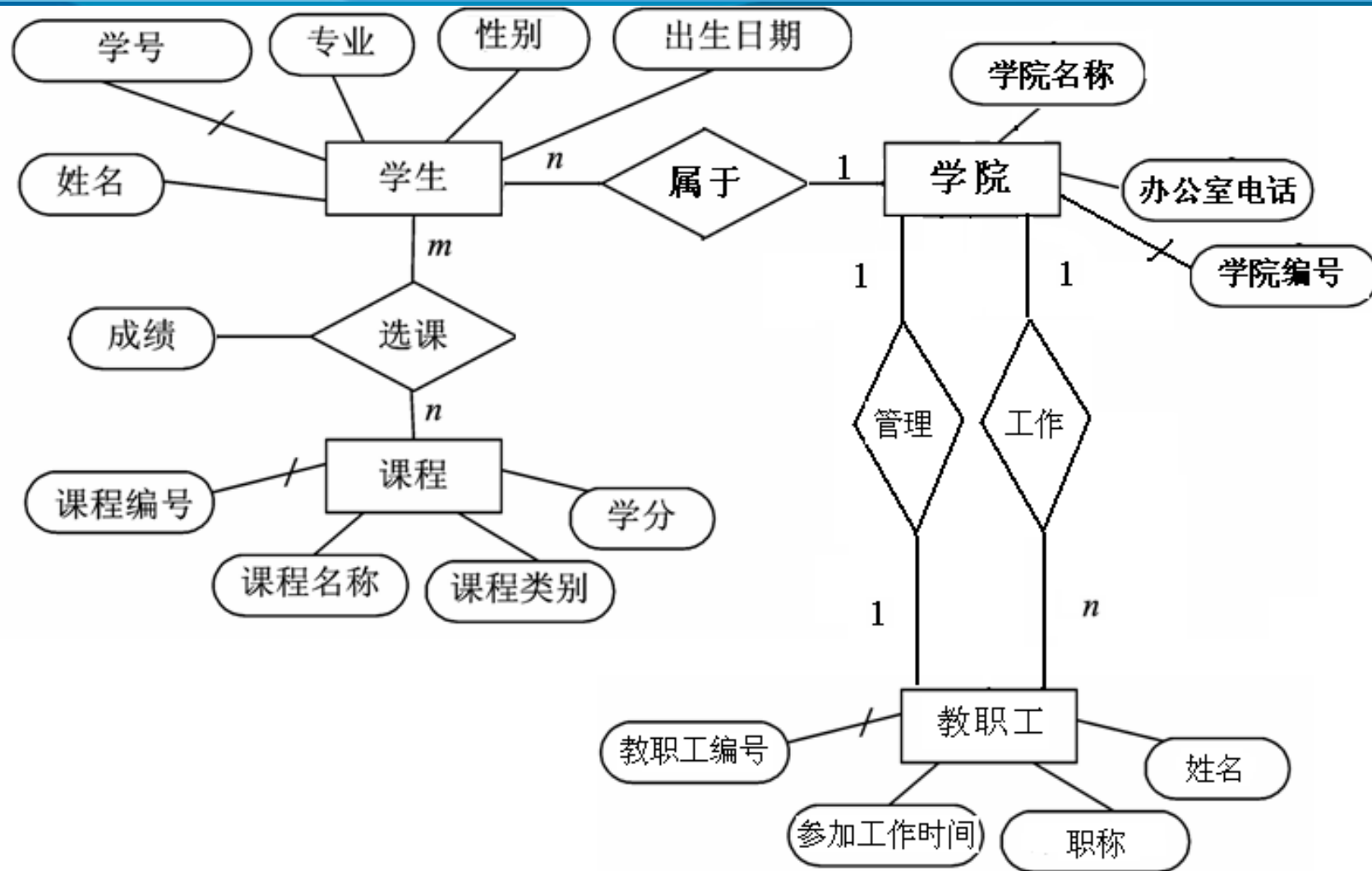
❖ 涉及的实体型有：

- ◆ 学生：属性有学号、姓名、性别、专业、出生日期
- ◆ 课程：属性有课程号、课程名、课程类别、学分
- ◆ 学院：属性有学院编号、学院名称、办公室电话
- ◆ 教职工：属性有教职工编号、姓名、职称、参加工作日期

❖ 这些实体之间的联系如下：

- ◆ 一个学生可以选修多门课程，一门课程可以由多个学生选修
- ◆ 一个学生只能属于一个学院，一个学院包括若干学生
- ◆ 一个教职工只能属于一个学院，一个学院包括若干教职工
- ◆ 一个学院只有一个正院长，一个正院长只能在一个学院任正院长职务

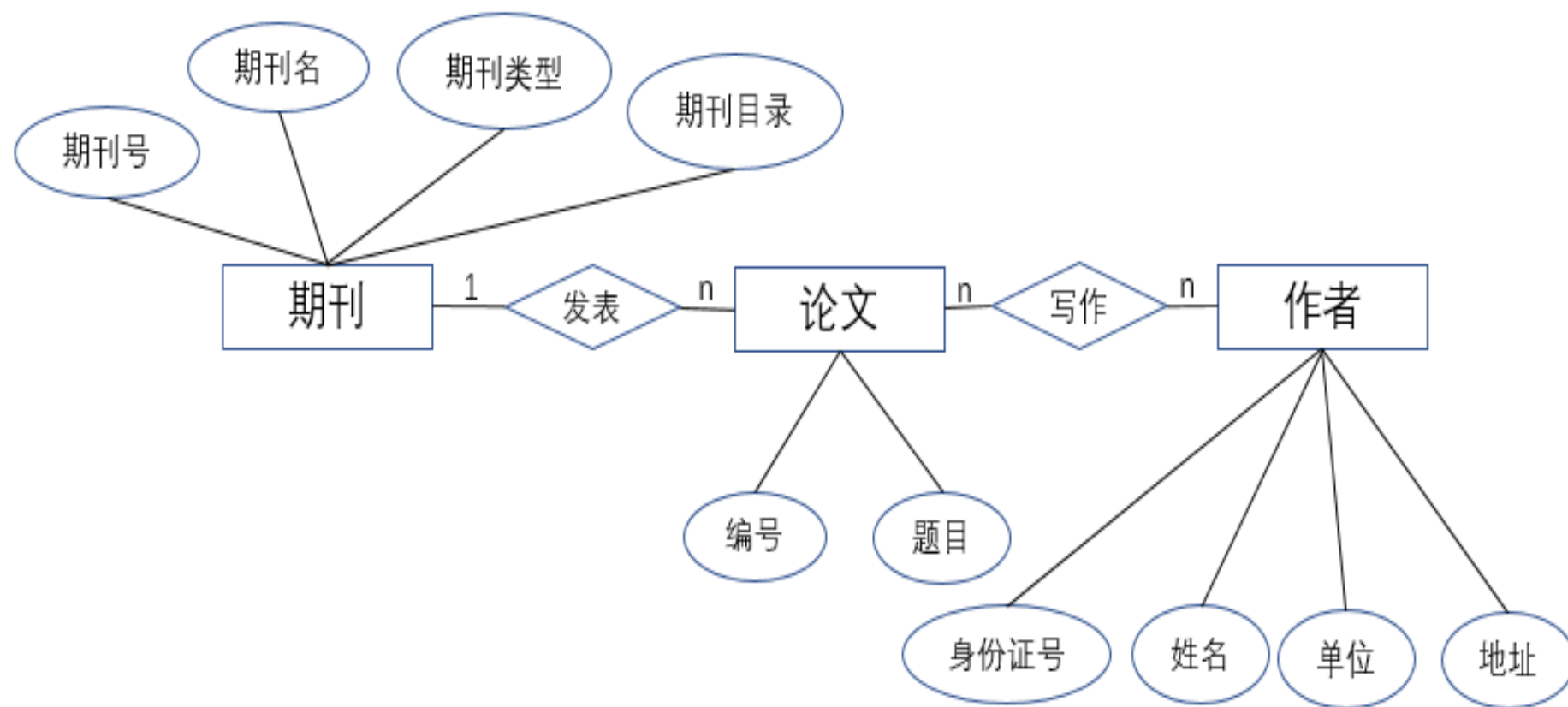




❖ 期刊论文模型

- ◆ 有若干种期刊，每种期刊包括：期刊号、期刊名、期刊类型、期刊目录
- ◆ 每种期刊有若干篇论文，每篇论文只能在一种期刊上发表，论文包括：编号、题目
- ◆ 一篇论文可能有多个作者，且一位作者可能写过多篇论文，在每一篇论文中有作者的顺序，作者包括：身份证号、姓名、单位、地址

❖ 请画出ER图，要求画出属性和注明联系类型



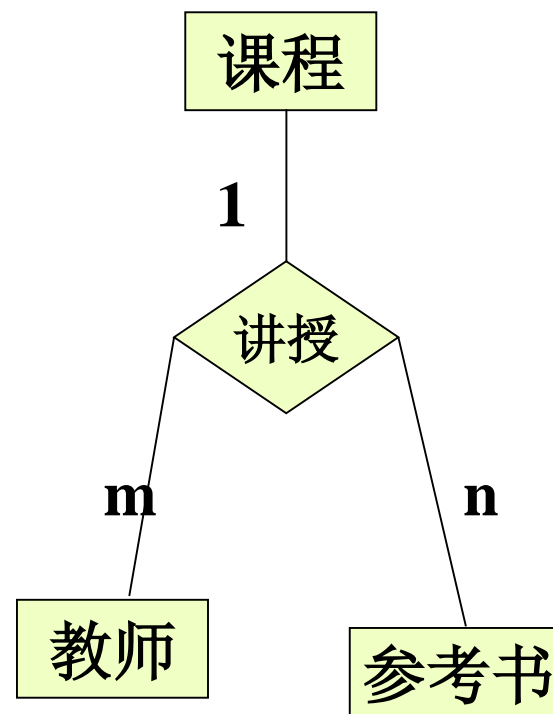
❖ 多个实体型间的一对多联系

- ◆ 若实体集 E_1, E_2, \dots, E_n 存在联系, 对于实体集 E_j ($j=1, 2, \dots, i-1, i+1, \dots, n$) 中的给定实体, 最多只和 E_i 中的一个实体相联系, 则我们说 E_i 与 $E_1, E_2, \dots, E_{i-1}, E_{i+1}, \dots, E_n$ 之间的联系是一对多的

两个以上实体型之间的联系

❖实例

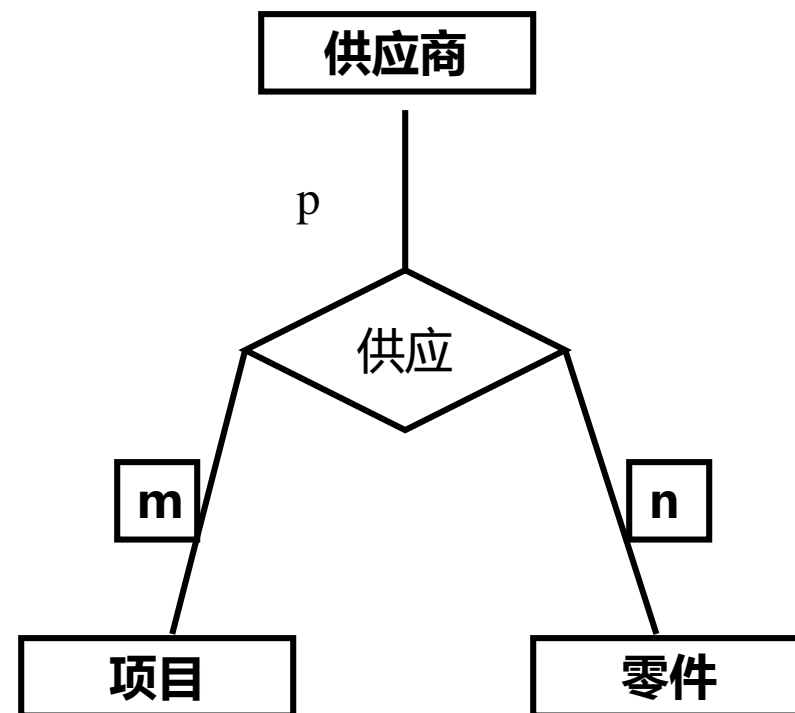
课程、教师与参考书三个实体型
一门课程可以有若干个教师讲授，
使用若干本参考书，
每一个教师只讲授一门课程，
每一本参考书只供一门课程使用



两个以上实体型间1:n联系

❖ 习题

- ◆ 供应商、项目与零件三个实体型
- ◆ 一个供应商可以供给多个项目多种零件，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



多个实体型间的联系

单个实体型内的联系

❖ 一对多联系

❖ 一对一联系

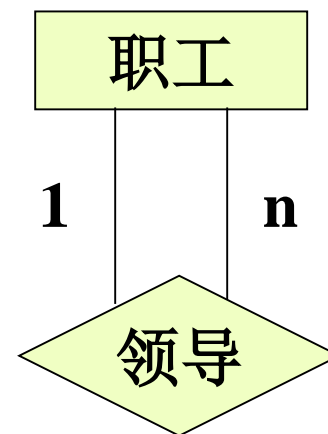
❖ 多对多联系

❖ 实例：职工实体型内部具有领导与被领导的联系

某一职工（干部）“领导”若干名职工

一个职工仅被另外一个职工直接领导

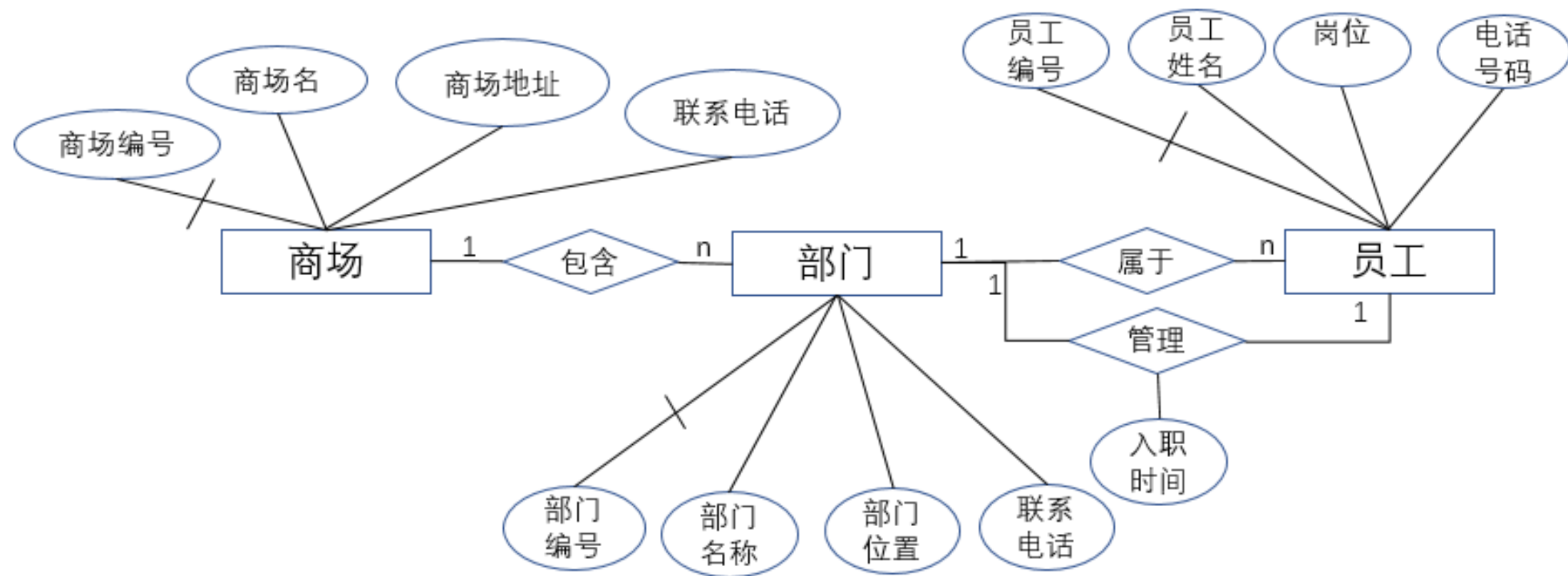
这是一对多的联系



单个实体型内部
1:n联系

❖ 某公司拥有多个大型连锁商场，公司需要构建一个数据库系统便于管理其业务，请根据描述构建E-R图

- ◆ 商场需要记录商场编号（不能重复）、商场名称、商场地址、联系电话
- ◆ 每个商场包含很多不同部门，部门需要记录部门的编号、部门名称、位置分布、联系电话
- ◆ 每个部门雇佣了很多员工，每个员工只属于一个部门。员工需要记录员工编号、员工姓名、岗位、电话号码
- ◆ 每个部门有一个经理，每个经理只管理一个部门，经理需要记录入职时间



❖ 用E-R图表示某个工厂物资管理的概念模型 (p218)



❖ 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

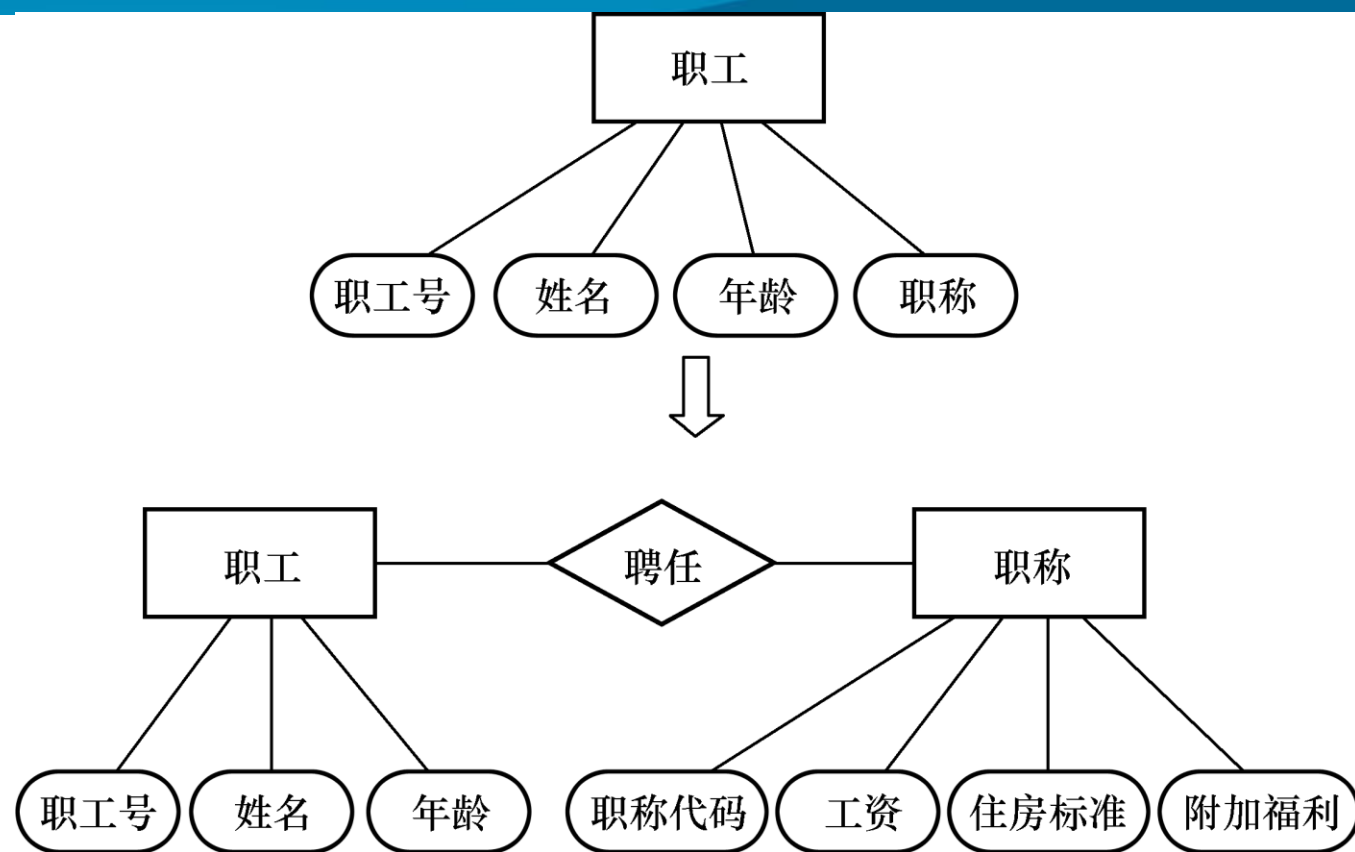
❖ 两条准则

- ◆ 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

◆ 举例

例1：“学生”由学号、姓名等属性进一步描述，根据准则1，“学生”只能作为实体，不能作为属性

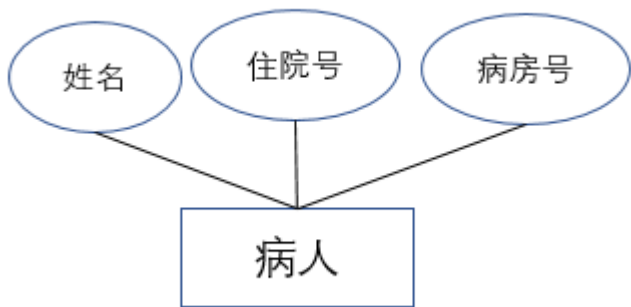
例2：教师是一个实体，有教工号、姓名、年龄等属性，通常职称也作为教师实体的属性，但在如果职称与教师的工资、岗位津贴、福利等挂钩，即职称与其他的实体之间有联系，则根据准则2，把职称作为实体来处理会更合适些



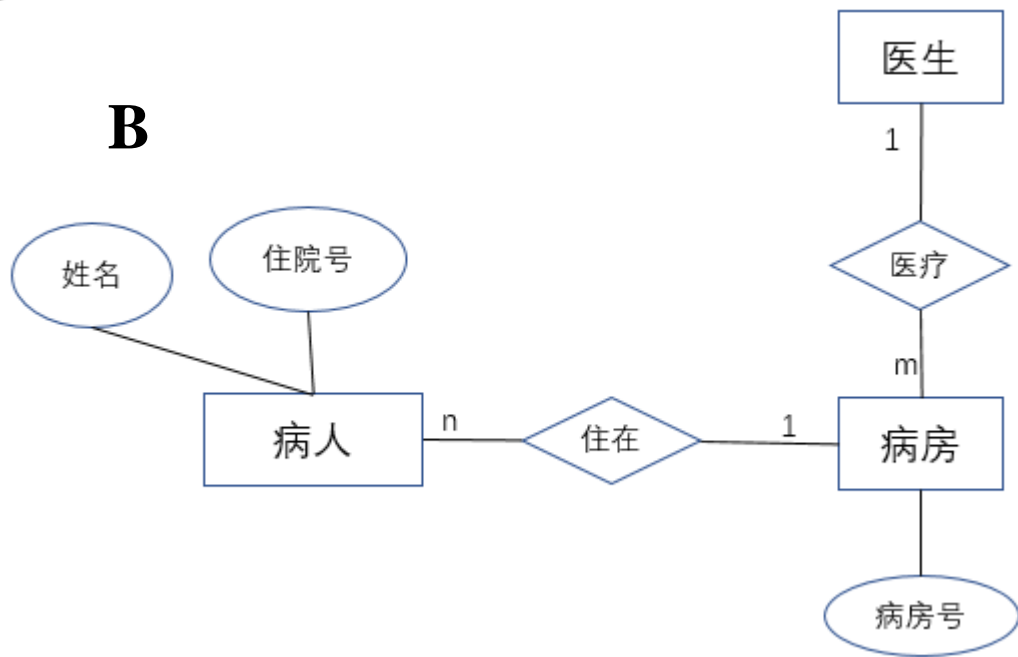
职称作为一个实体

❖ 医院中有病人，每个病人有住院号、姓名、籍贯、病房号等属性，医院里的每一个医生需要负责几个病房的病人的医疗工作，则病房号应该如何处理？

A



B



❖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

❖视图集成方法

◆ 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

◆ 逐步集成

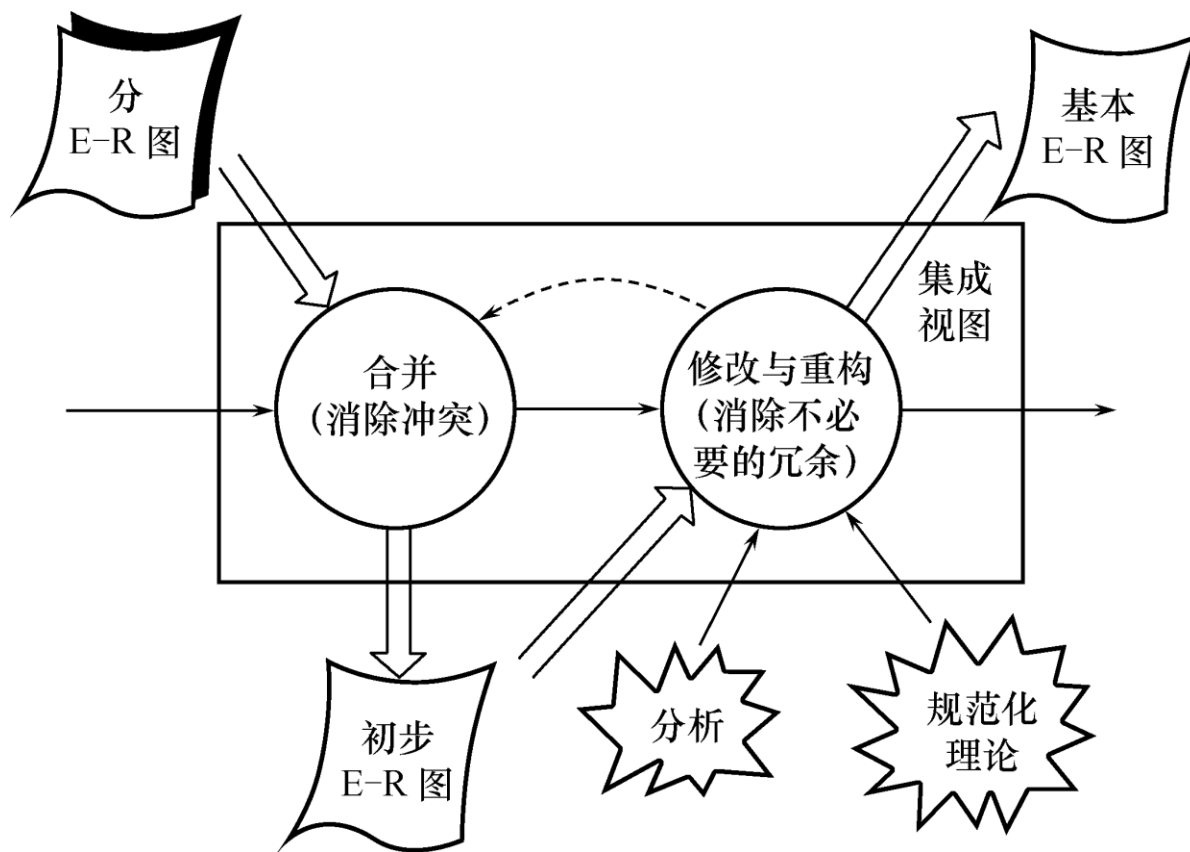
- 用累加的方式一次集成两个分E-R图



集成局部E-R图的步骤

❖ 合并

❖ 修改与重构

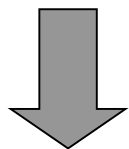


❖ 合并分E-R图，生成初步E-R图

◆ 各分 E - R 图存在冲突

➤ 各个局部应用所面向的问题不同

由不同的设计人员进行设计



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所在：合理消除各分E-R图的**冲突**

❖ 冲突的种类

- ◆ 属性冲突
- ◆ 命名冲突
- ◆ 结构冲突



❖ 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

例1：由于学号是数字，因此某些部门（即局部应用）将学号定义为整数形式，而由于学号不用参与运算，因此另一些部门（即局部应用）将学号定义为字符型形式

◆ 属性取值单位冲突

例2：学生的身高，有的以米为单位，有的以厘米为单位，有的以尺为单位

❖ 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

例：局部应用A中将教室称为房间

局部应用B中将学生宿舍称为房间



❖ 三类结构冲突

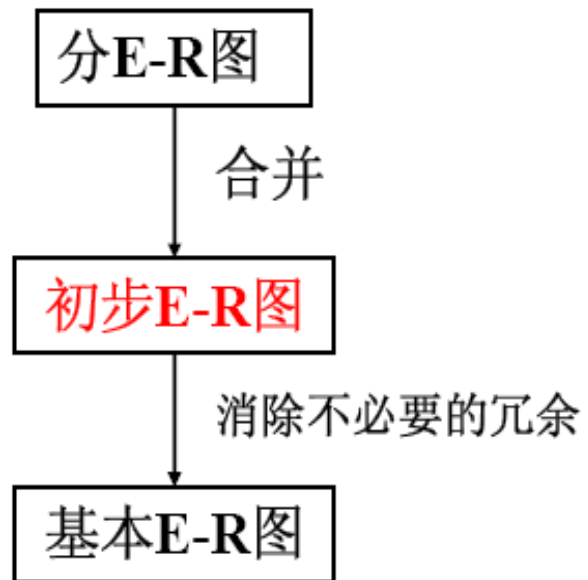
- ◆ 同一对象在不同应用中具有不同的抽象
- ◆ 同一实体在不同分E-R图所包含的属性个数和属性排列次序不完全相同
- ◆ 实体之间的联系在不同局部视图中呈现不同的类型



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



- ❖ 第一节 数据库设计概述
- ❖ 第二节 需求分析
- ❖ 第三节 概念结构设计
- ❖ 第四节 逻辑结构设计
- ❖ 第五节 数据库的物理设计
- ❖ 第六节 数据库实施和维护

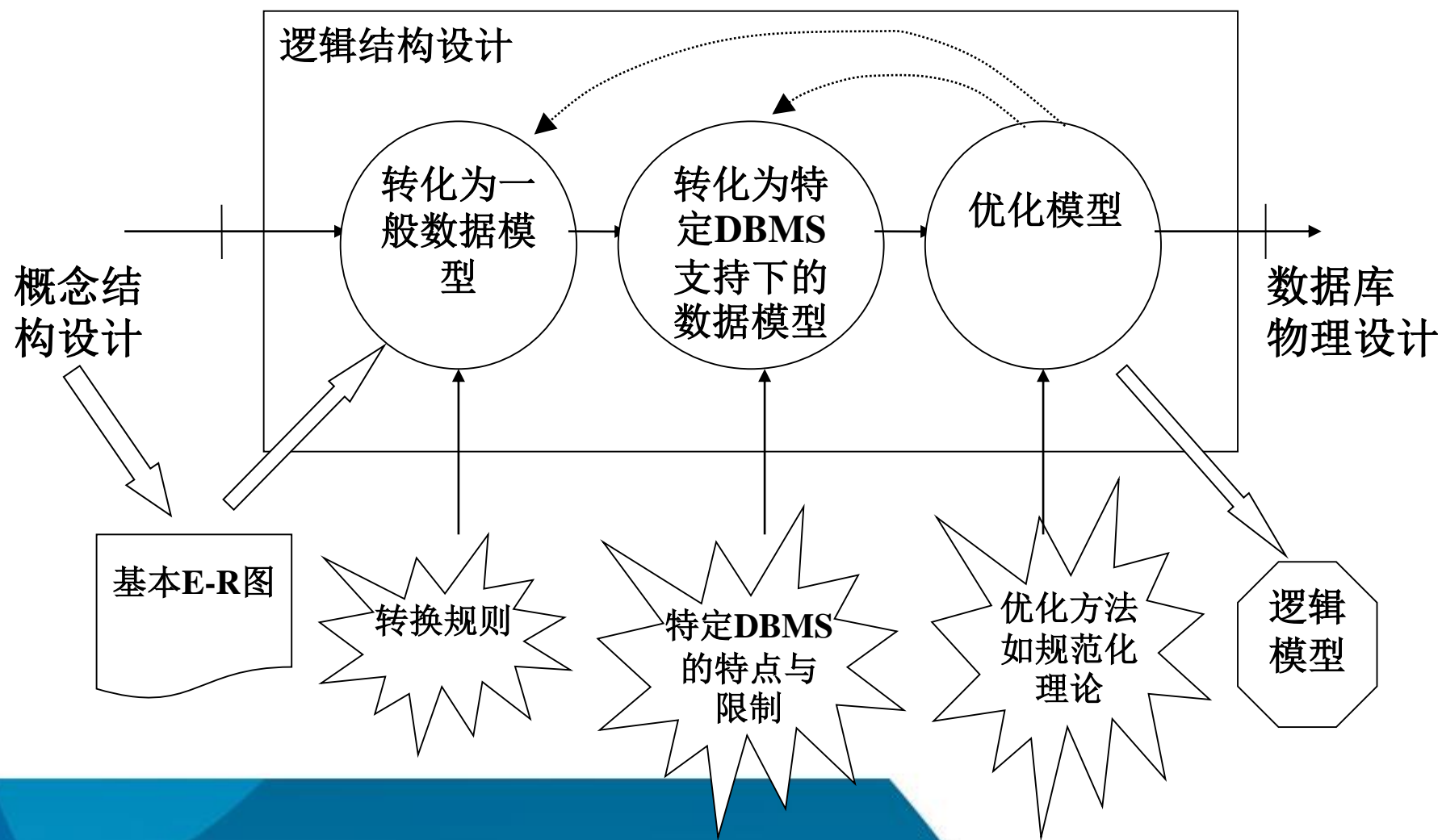


❖ 逻辑结构设计的任务

- ◆ 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



E-R图向关系模型的转换

数据模型的优化

设计用户子模式

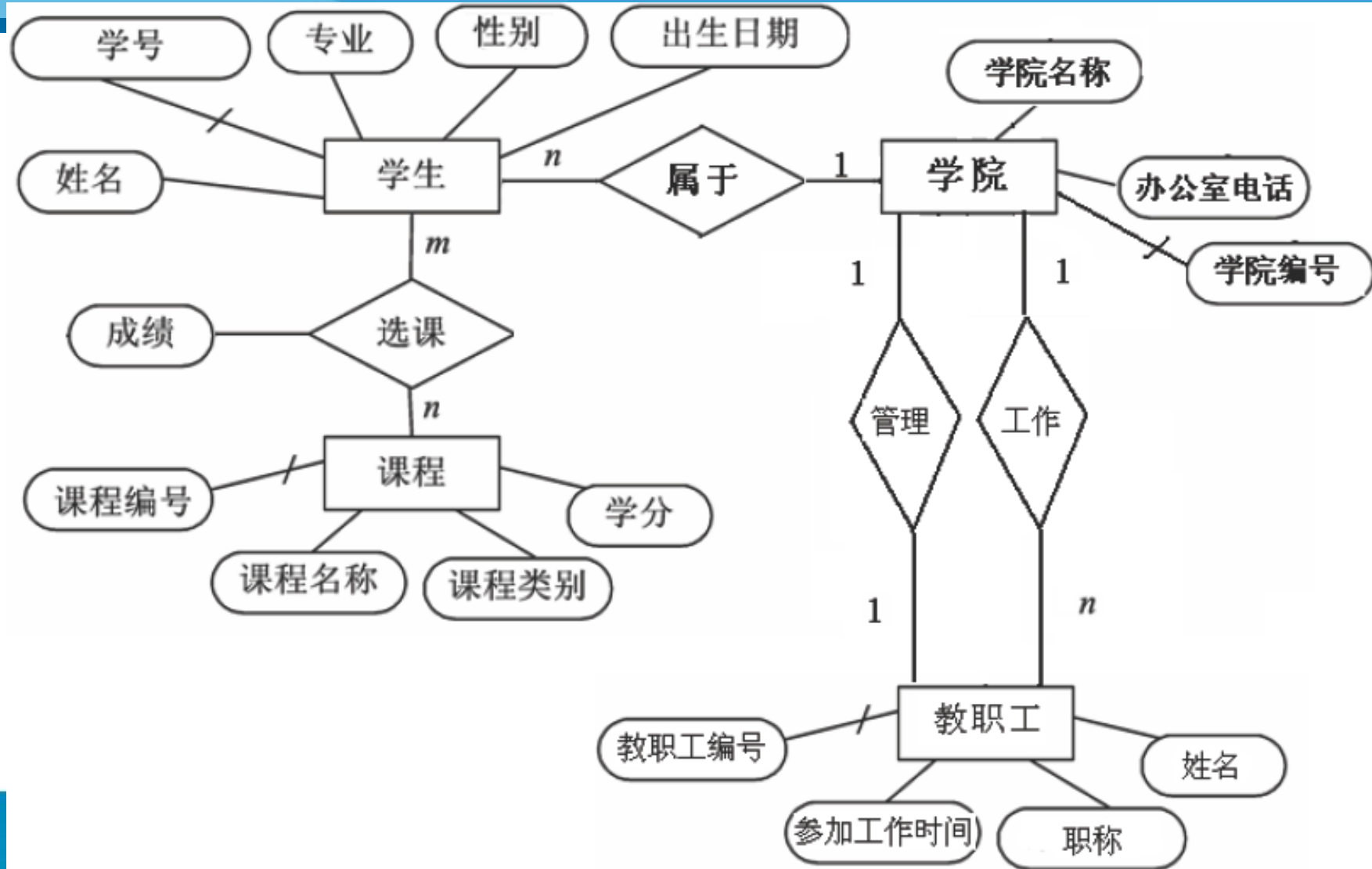
❖ E-R图向关系模型的转换要解决的问题

- ◆ 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





转换原则

❖ 一个实体型转换为一个关系模式

- ◆ 关系的属性：实体型的属性
- ◆ 关系的码：实体型的码

关系模式：

学生 (学号, 姓名, 性别, 专业, 出生日期)

课程 (课程编号, 课程名称, 课程类别, 学分)

学院 (学院编号, 学院名称, 办公室电话)

教职工 (教职工编号, 姓名, 参加工作时间, 职称)

❖ 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以及联系本身的属性

选课

学号	课程号	成绩
	· · ·	

- 关系的码：各实体码的组合
- 选课 (学号, 课程号, 成绩)



❖ 一个1:n联系可以转换为一个独立的关系模式，也可以与n端对应的关系模式合并

◆ 转换为一个独立的关系模式

➤ 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**

学院学生（学院编号，学号）

学院教职工（学院编号，教职工编号）



- ◆ 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

- ◆ 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



E-R图向关系模型的转换

学生 (学号, 姓名, 性别, 专业, 出生日期, 学院编号)

课程 (课程编号, 课程名称, 课程类别, 学分)

学院 (学院编号, 学院名称, 办公室电话, 院长教职工编号)

教职工 (教职工编号, 姓名, 参加工作时间, 职称, 学院编号)

选课 (学号, 课程号, 成绩)



❖ 一个1:1联系可以转换为一个独立的关系模式，也可以与任意一端对应的关系模式合并

◆ 转换为一个独立的关系模式

- 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：每个实体的码均是该关系的候选码

◆ 与某一端的关系模式合并

- 合并后关系的属性：加入另一端关系的码和联系本身的属性
- 合并后关系的码：不变



❖ 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），并适当调整属性的次序



❖ 期刊论文E-R模型转换为关系模型

- ◆ 期刊 (期刊号, 期刊名, 期刊类型, 期刊目录)
- ◆ 论文 (编号, 题目, 发表期刊号)
- ◆ 作者 (身份证号, 姓名, 单位, 地址)
- ◆ 写作 (作者身份证号, 论文编号)

❖ 连锁商场E-R模型转换为关系模型

- 商场 (商场编号, 商场名, 商场地址, 联系电话)
- 部门 (部门编号, 部门名称, 部门位置, 联系电话, 所属商场编号)
- 员工 (员工编号, 员工姓名, 岗位, 电话号码, 所属部门编号)
- 管理 (员工编号, 部门编号, 入职时间)



- ❖ 数据库逻辑设计的结果不是唯一的
- ❖ 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



❖ 优化数据模型的方法

- ◆ 确定数据依赖
- ◆ 消除冗余的联系
- ◆ 确定所属范式
- ◆ 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
- ◆ 按照需求分析阶段得到的各种应用对数据处理的要求，对关系模式进行必要的分解，以提高数据操作的效率和存储空间的利用率

设计用户子模式（外模式）

❖ 定义用户外模式时应该注重的问題,包括三个方面

- ◆ 使用更符合用户习惯的别名
- ◆ 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
- ◆ 简化用户对系统的使用



❖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

❖ 设有系、教师、学生、课程等实体，其中：

系（系名、系址、系主任姓名、办公电话）

教师（工作证号码、教师名、出生日期、党派）

学生（学号、姓名、出生日期、性别）

课程（课程号、课程名）

实体间的关系：

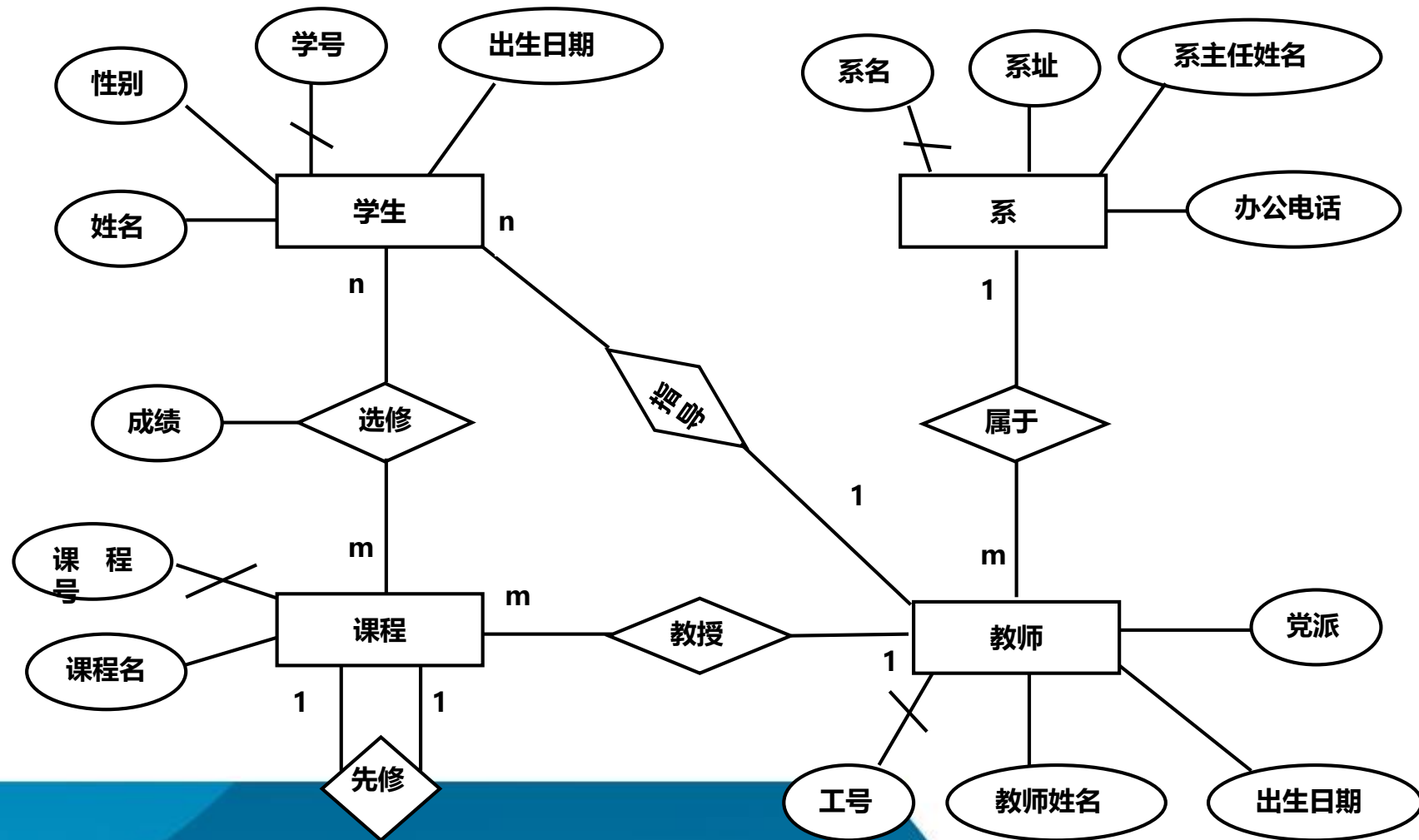
- (1) 一个系可以有多名教师，一个教师只属于一个系；
- (2) 每个教师教多门课程，一门课程由一个教师教；
- (3) 教师指导多个研究生，一个研究生只能有一个导师
- (4) 一个学生可选多门课程，每门课程可以由多个学生选，学生选修一门课程有一个成绩
- (5) 每门课程只有一个预修课程

试根据以上语义完成下述要

(1) 画出E-R图

(2) 将以上ER图用关系模型表示出来

(3) 标出各关系的主键、外键



系 (系名, 系址, 系主任姓名, 办公电话)

教师 (工作证号码, 教师名, 出生日期, 党派, 系名)

学生 (学号, 姓名, 出生日期、性别、指导教师工号)

课程 (课程号, 课程名, 预修课号, 任课教师工号)

选修 (学号, 课程号, 成绩)

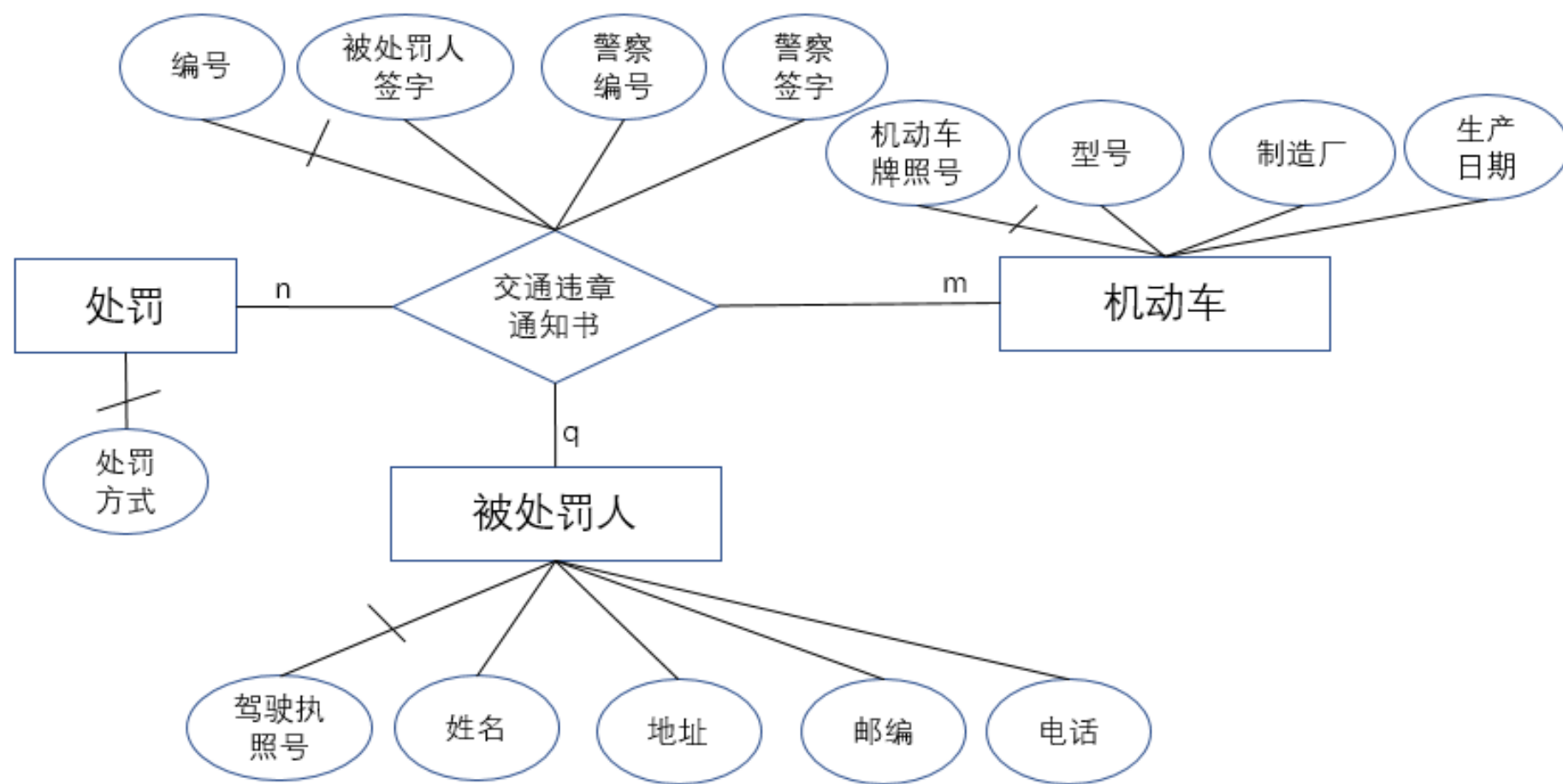


❖ 下图中显示一张交通违章处罚通知书，根据这张通知书所提供的信息，设计一个E-R模型，并将这个E-R模型转换成关系数据模型，要求标注主码和外码。注：驾驶执照号和机动车牌照号是不同的

- (1) 根据以上描述画出E-R图，要求画出属性和注明联系类型
- (2) 将E-R图转换为一组关系模式（表尽可能少），并给出每个关系模式的主键和外键（如果存在）

综合练习二

交通违章通知书：		编号：TZ11719.
1.	姓名：××× 驾驶执照号：××××××. 地址：×××××××××× . 邮编：×××××× 电话：××××××.	
2.	机动车牌照号：×××××× . 型号：××××××. 制造厂：×××××× 生产日期：××××××.	
3.	违章日期：×××××× 时间：×××××× . 地点：×××××× . 违章记载：××××××.	
4.	处罚方式. <input checked="" type="checkbox"/> 警告. <input checked="" type="checkbox"/> 罚款 . <input type="checkbox"/> 暂扣驾驶执照.	
5.	警察签字：××× 警察编号：×××.	
6.	被处罚人签字：×××.	



处罚 (处罚方式)

被处罚人 (驾驶执照号, 姓名, 地址, 邮编, 电话)

机动车 (机动车牌照号, 型号, 制造厂, 生产日期)

交通违章通知书 (编号, 被处罚人, 机动车, 处罚方式, 警察编号, 警察签字, 被处罚人签字)



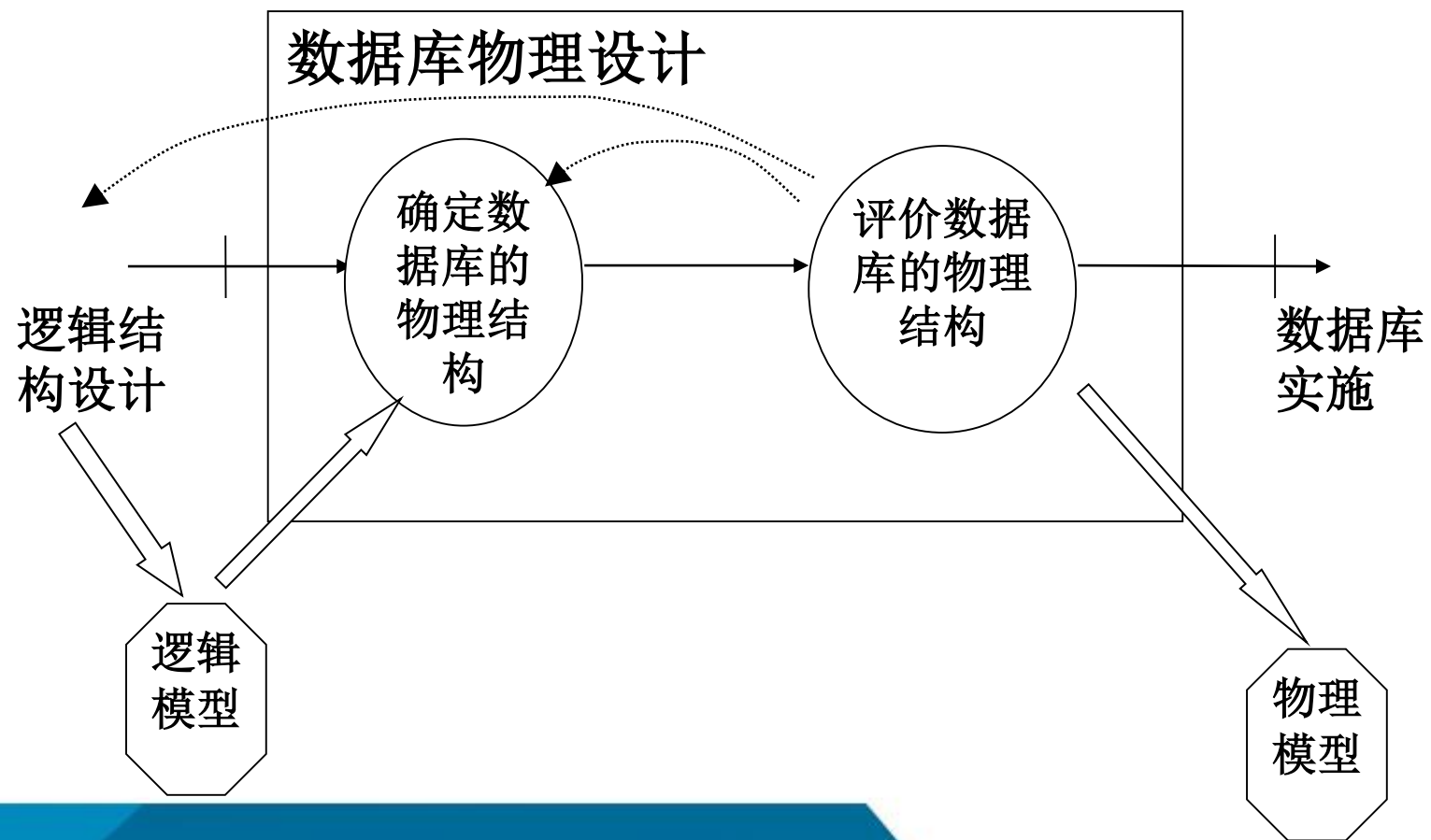
- ❖ 第一节 数据库设计概述
- ❖ 第二节 需求分析
- ❖ 第三节 概念结构设计
- ❖ 第四节 逻辑结构设计
- ❖ 第五节 数据库的物理设计
- ❖ 第六节 数据库实施和维护



❖ 什么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，它依赖于给定的计算机系统
- ◆ 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



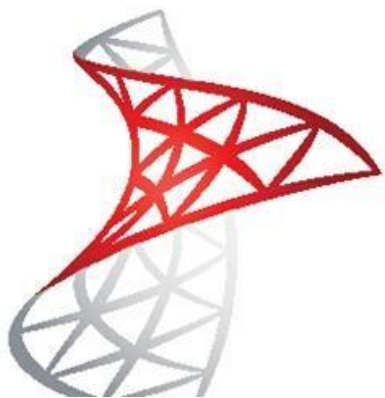


❖ 设计物理数据库结构的准备工作

- ◆ 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



❖ 关系数据库物理设计的内容



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

- ❖ 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
- ❖ 物理设计的第一个任务就是要确定选择哪些存取方法，即建立哪些存取路径



❖ DBMS常用存取方法

- ◆ 索引方法，目前主要是B+树索引方法

经典存取方法，使用最普遍

- ◆ 聚簇（Cluster）方法

- ◆ HASH方法



❖ 选择索引存取方法的主要内容

根据应用要求确定

- ◆ 对哪些属性列建立索引
- ◆ 对哪些属性列建立组合索引
- ◆ 对哪些索引要设计为唯一索引



❖ 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(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



❖ 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（称为聚簇码）上具有相同值的元组集中存放在连续的物理块**称为聚簇**
- ◆ 许多关系型DBMS都提供了聚簇功能



❖ 聚簇的用途

- ◆ 大大提高按聚簇属性进行查询的效率
- ◆ 节省存储空间

❖ 聚簇的局限性

- ◆ 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，将导致关系中元组移动其物理存储位置，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移动

❖ 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
- ◆ 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者是次要的时，可以使用聚簇



❖ 确定数据库物理结构的内容

◆ 确定数据的存放位置和存储结构

- 关系
- 索引
- 聚簇
- 日志
- 备份

◆ 确定系统配置



❖ DBMS产品一般都提供了一些存储分配参数

- ◆ 同时使用数据库的用户数
- ◆ 同时打开的数据库对象数
- ◆ 使用的缓冲区长度、个数
- ◆ 时间片大小
- ◆ 数据库的大小
- ◆ 装填因子
- ◆ 锁的数目



❖ 评价内容

- ◆ 对数据库物理设计过程中产生的多种方案进行细致的评价，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
- ◆ 如果该结构不符合用户需求，则需要修改设计

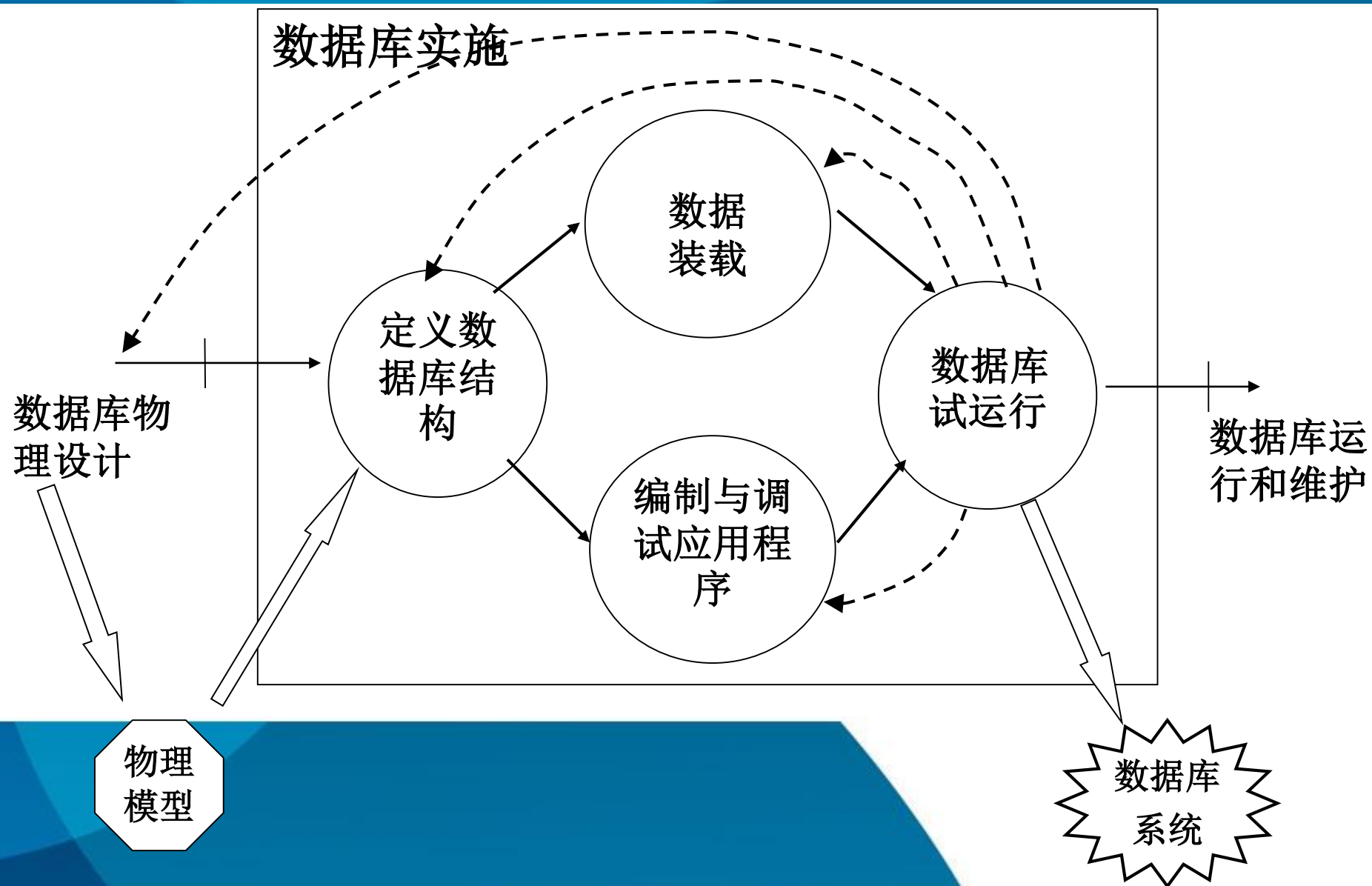
- ❖ 第一节 数据库设计概述
- ❖ 第二节 需求分析
- ❖ 第三节 概念结构设计
- ❖ 第四节 逻辑结构设计
- ❖ 第五节 数据库的物理设计
- ❖ 第六节 数据库实施和维护



❖ 数据库实施的工作内容

- ◆ 用DDL定义数据库结构
- ◆ 组织数据入库
- ◆ 编制与调试应用程序
- ◆ 数据库试运行





- ❖ 数据库试运行结果符合设计目标后，数据库就可以真正投入运行了
- ❖ 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

- ❖ 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
 - ◆ 数据库的转储和恢复
 - ◆ 数据库的安全性、完整性控制
 - ◆ 数据库性能的监督、分析和改进
 - ◆ 数据库的重组和重构造



❖ 实验

- ◆ 实验三 PowerDesigner自动生成PDM

❖ 第七章实验



子曰：「賜也！女以予
为多学而识之者与？」
对曰：「然，非与？」
曰：「非也。予一以贯
之。」

