

教师之为教，不在全盘授予，而在相机
诱导。必令学生运其才智，勤其练习，领悟
之源广开，纯熟之功弥深，乃善教者也。

——叶圣陶先生

第七章 数据库设计

7.1 数据库设计概述

7.2 需求分析

7.3 概念结构设计

7.4 逻辑结构设计

7.5 物理设计

7.6 数据库实施与维护

7.7 数据库设计实例（自学）

一. 数据库设计DBD (DataBase Design)

☞ 又称数据库分析与设计

1. 主要目标：设计一个“好”的数据模式

❖ 具体要求：

☞ 满足用户需求

☞ 符合环境要求

2. 满足用户需求：

❖ 信息管理需求

☞ 是指在数据库中应该存储和管理哪些数据对象

❖ 数据操作要求

☞ 是指对数据对象需要进行哪些操作

3.符合环境要求:

- ❖ 硬件平台
- ❖ 软件平台

二. 数据库设计的特点

- ❖ 是软件工程的一部分称**数据工程**

1. 数据库建设的基本规律

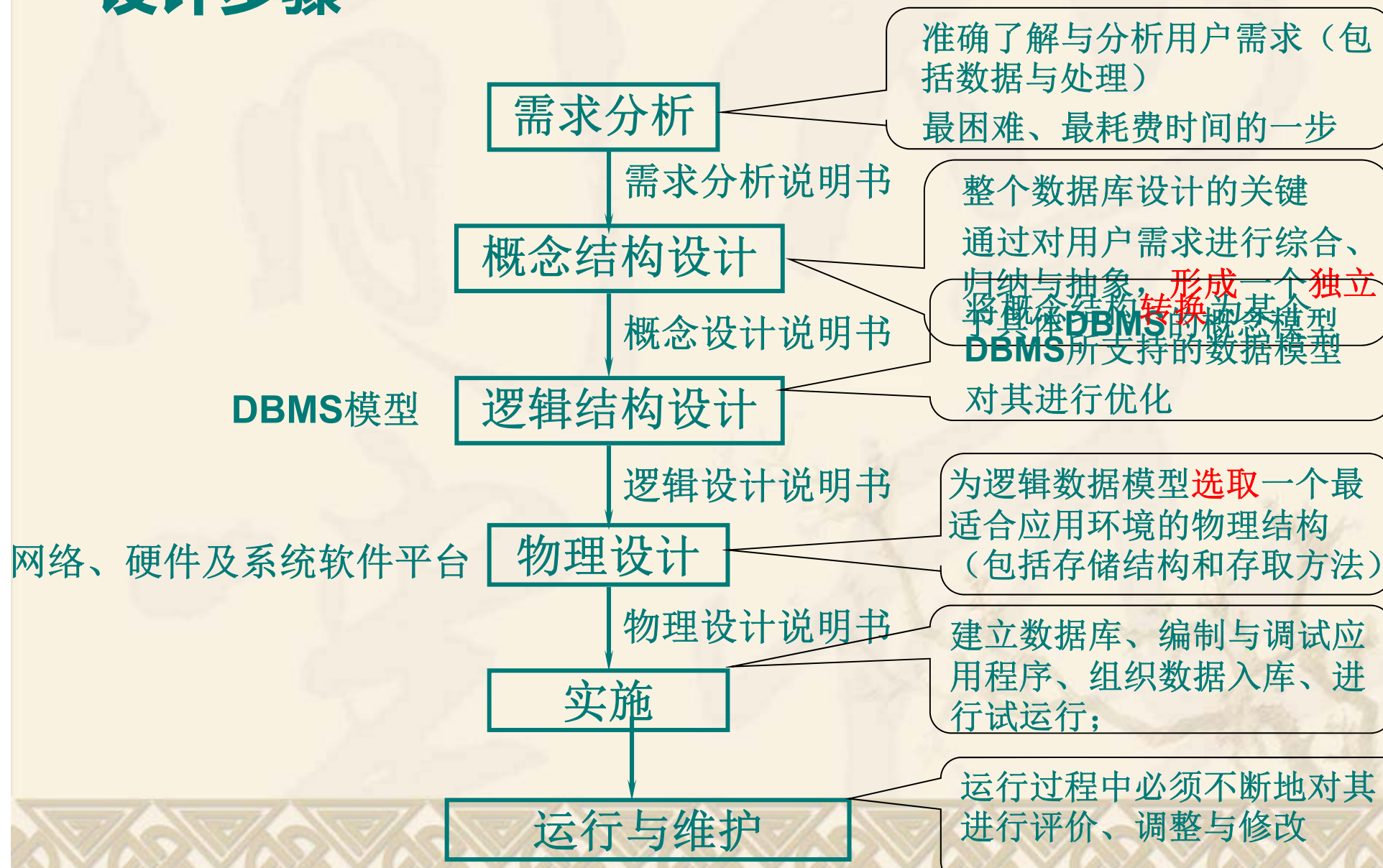
☞ “三分技术，七分管理，十二分基础数据”

2. 结构（数据）设计和行为（处理）设计相结合

三. 数据库设计方法

- ❖ 手工与经验相结合方法
- ❖ 规范设计法

设计步骤



7.2 需求分析

1.需求分析的任务

- ❖ 通过详细调查现实世界要处理的对象，充分了解原系统的工作概况，明确用户的各种需求，然后在此基础上确定新系统的功能。



图7.2 需求分析结构图

2. 调查的重点

—— “数据和处理”

- ❧ 信息要求。指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。
- ❧ 处理要求。指用户要完成的数据处理功能和性能。
- ❧ 安全性与完整性要求。

3. 常用的调查方法

- ❧ 跟班作业
- ❧ 开调查会
- ❧ 请专人介绍
- ❧ 询问
- ❧ 设计调查表请用户填写
- ❧ 查阅记录

4. 需求分析的方法

- ❧ 首先是**调查**清楚用户的实际需求，与用户**达成共识**
- ❧ **分析和表达需求**
- ❧ 需求分析**报告**提交用户，征得用户的**认可**

5 . 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
- ❧ 协助用户明确对新系统的各种要求
- ❧ 确定新系统的边界

6. 数据字典DD

- ❖ 数据字典是进行详细收集和数据分析所获得的主要成果。关于数据库中数据的描述，是元数据，而不是数据本身
- ❖ 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
- ❖ 数据字典的内容
 - ❖ 数据项（数据的最小组成单位）
 - ❖ 数据结构
 - ❖ 数据流
 - ❖ 数据存储
 - ❖ 处理过程

❖ 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}

❧ 其中, 取值范围、与其他数据项的逻辑关系 (如等于其他数据项之和等), 定义了数据的完整性约束条件。

❖ 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} }

❖ 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, 高峰期流量 }
- ❧ 说明: 来源即来自哪个过程; 平均流量即单位时间里传输次数。

❖ 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。
- ❧ 可以是手工文档或手工凭单，也可是计算机文档；
- ❧ 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}
- ❧ 说明：存取频度指单位时间存取次数及数据量等信息；存取方式指批处理还是联机处理，顺序存取还是随机存取等

❖ 处理过程

- ❧ 具体处理逻辑一般用判定表或判定树来描述
- ❧ 数据字典只需要描述处理过程的说明性信息即可
- ❧ 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} }

8. 数据字典举例

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： 8

取值范围： 00000000至99999999

取值含义： 前两位标别该学生所在年级，
后六位按顺序编号

与其他数据项的逻辑关系：

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： 是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

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结果

数据流来源： 体检

数据流去向： 批准

组成：

平均流量：

高峰期流量：

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：

流出数据流：

组成：

数据量： 每年3000张

存取方式： 随机存取

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性别的学生，
同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3平方米。
安排新生宿舍其处理时间应不超过15分钟。

8.数据性能需求

- ∞精度
- ∞时间
- ∞灵活性
- ∞安全性
- ∞完整性
- ∞可靠性
- ∞运行环境

×××项目需求分析说明书

1. 前言

1.1 编写目的

1.2 背 景

1.3 名词定义

1.4 参考资料

2. 数据关系分析

2.1 数据边界分析

2.2 数据内部关系分析

2.3 数据环境分析

3. 数据字典

4.1 数据项分析

4.2 数据类分析

4.3 数据性能需求分析

4. 原始资料汇编

4.1 原始资料汇编之1

.....

4. nn

编写人员：_____

审核人员：_____

审批人员：_____

日 期：_____

需求分析小结

- ❖ 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**

7.3 概念结构设计

1. 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。即目标—ER图

2. 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

3.E-R模型

❖ 实体

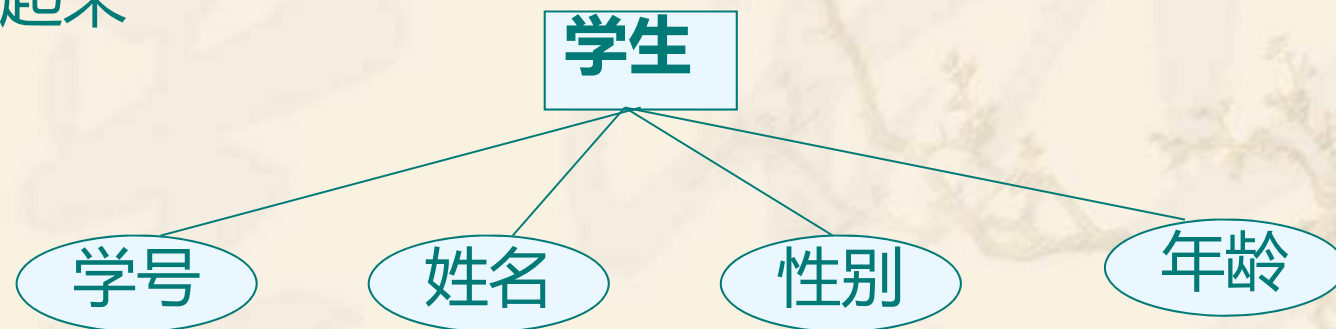
∞ 用矩形表示，矩形框内写明实体名。

学生

教师

❖ 属性

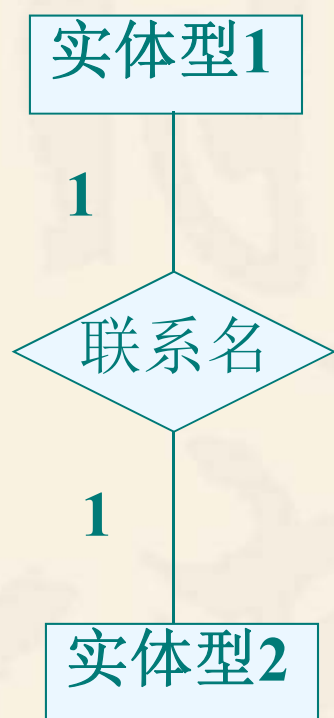
∞ 用椭圆形表示，并用无向边将其与相应的实体连接起来



❖ 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）

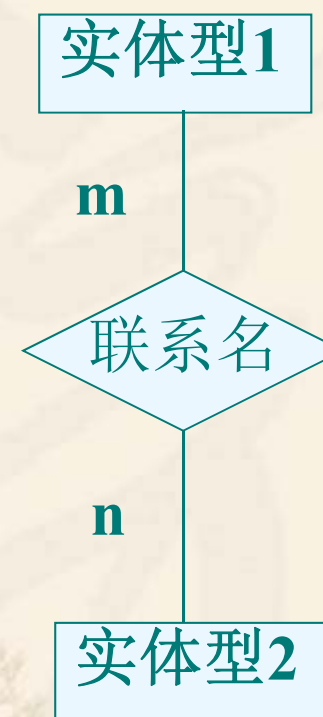
联系的表示方法



1:1联系



1:n联系

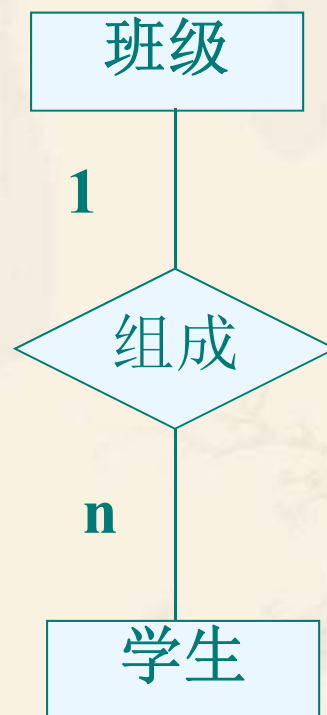


m:n联系

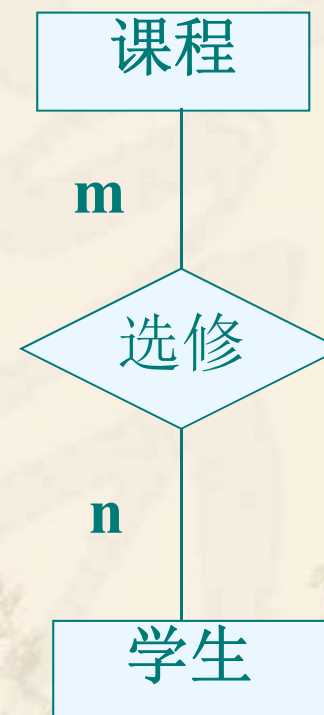
联系的表示方法示例



1:1联系



1:n联系



m:n联系

两个以上实体型之间的联系

❖ 实例

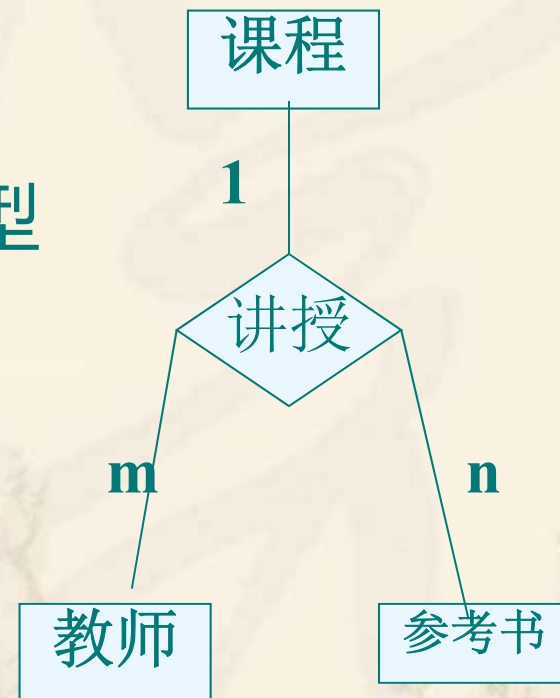
课程、教师与参考书三个实体型

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

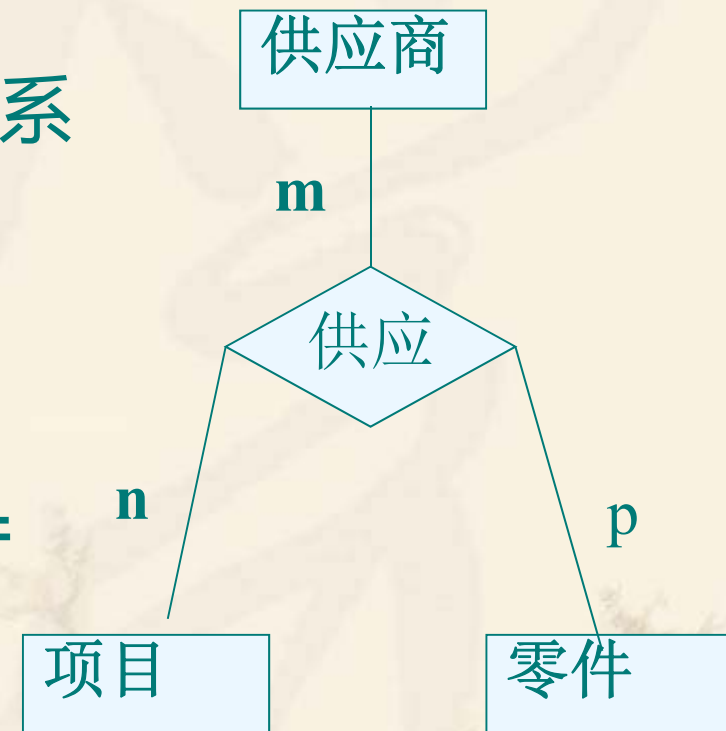


两个以上实体型间1:n联系

- ❖ 多个实体型间的一对一联系
- ❖ 两个以上实体型间的多对多联系

实例

供应商、项目、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



两个以上实体型间m:n联系

单个实体型内的联系

❖ 一对多联系

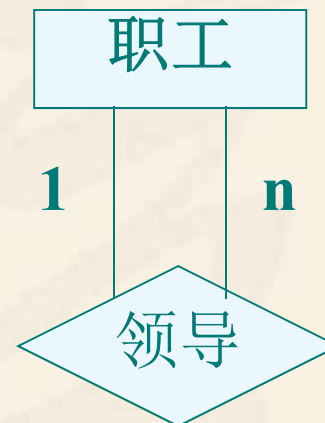
∞ 实例

职工实体型内部具有领导与被领导的联系

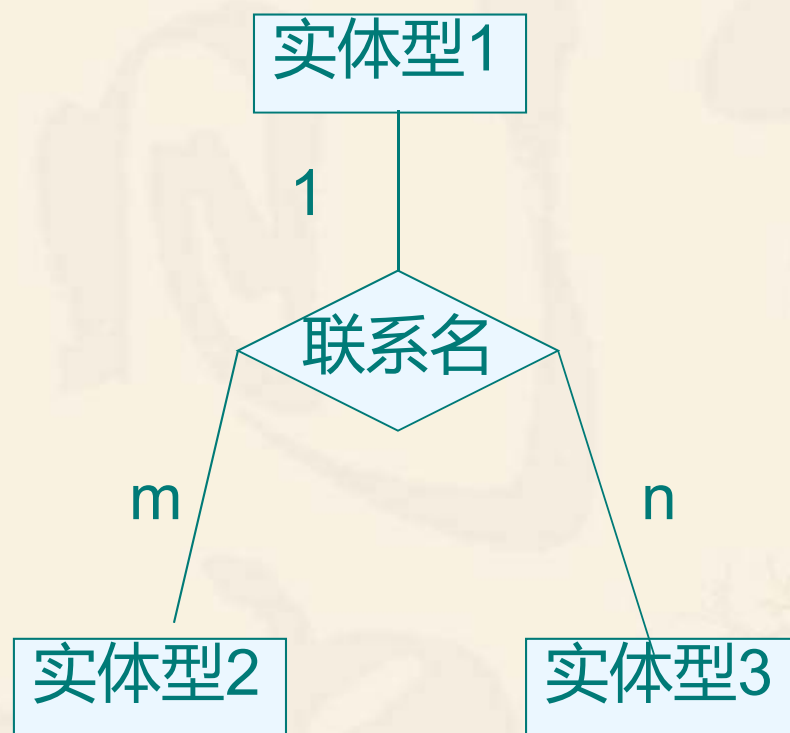
某一职工（干部）“领导”若干名职工

一个职工仅被另外一个职工直接领导

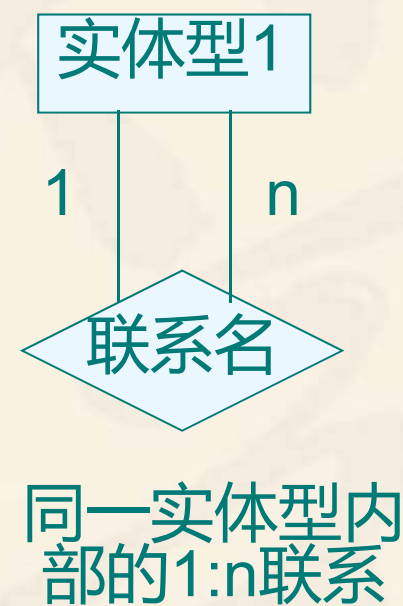
这是一对多的联系



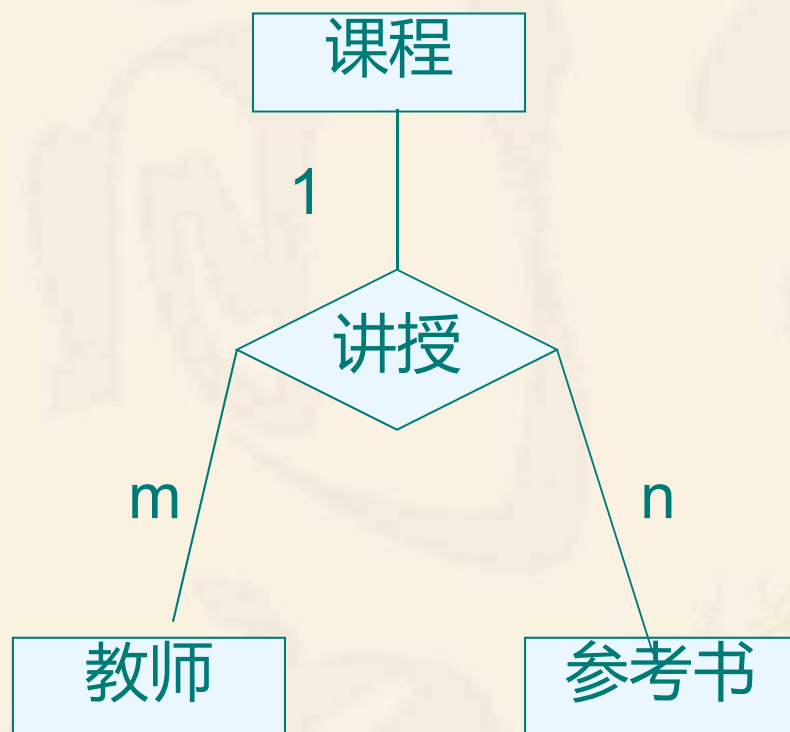
单个实体型内部
1:n联系



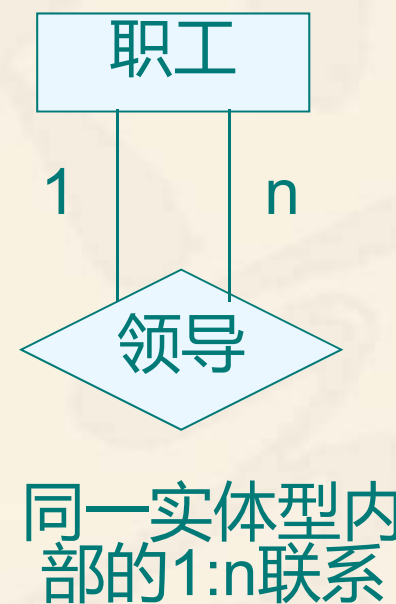
多个实体型间的1:n联系



同一实体型内部的1:n联系



多个实体型间的1:n联系



❖ 联系

∞ 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。

❖ 联系属性的表示方法



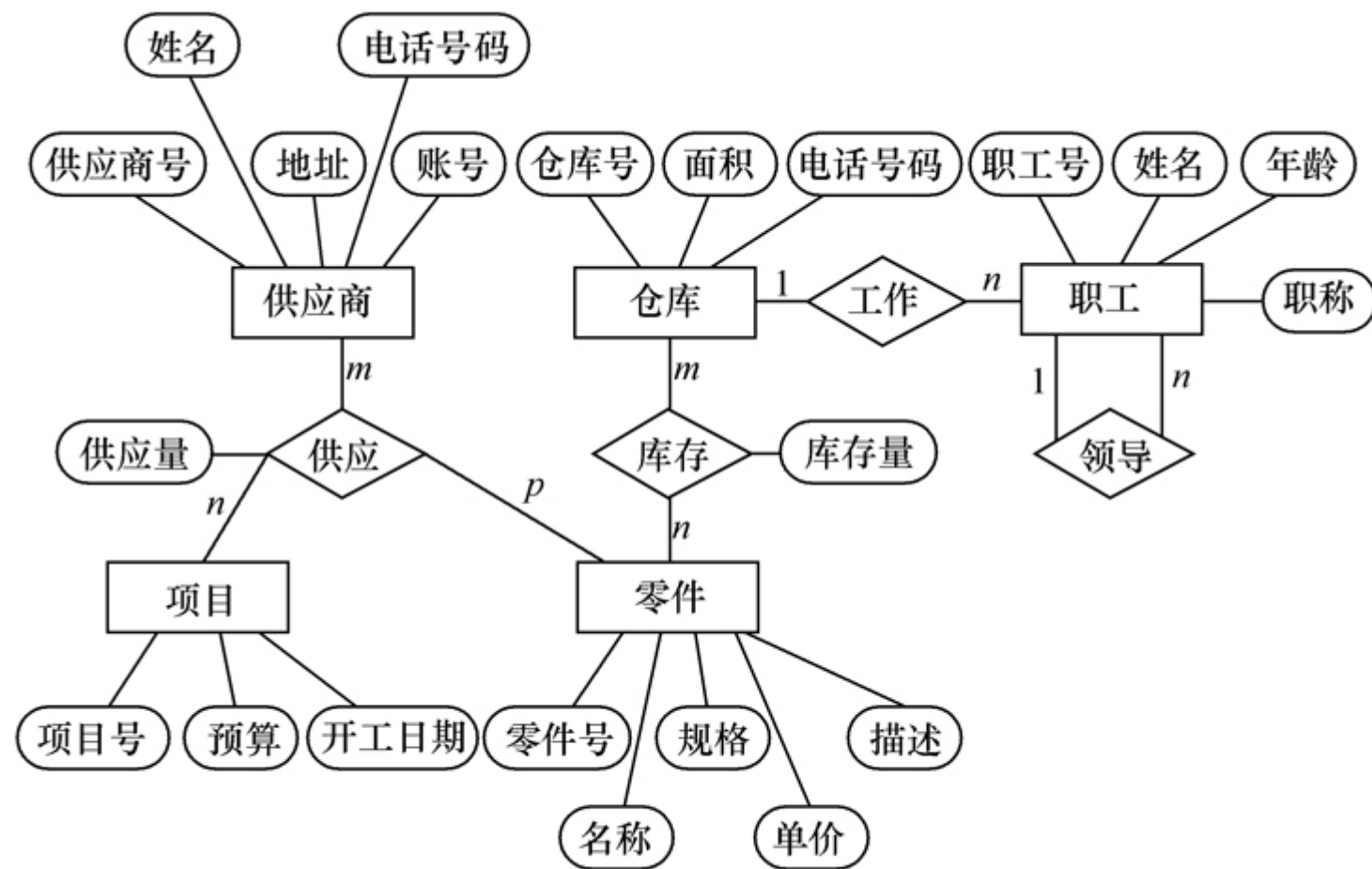
E-R图实例:某个工厂物资管理的概念模型

❖ 物资管理涉及的实体:

- ❧ 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

- ❧ 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。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。
- ❧ 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。
- ❧ 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。



(c) 完整的实体-联系图

4.设计方法

❧ 自顶向下

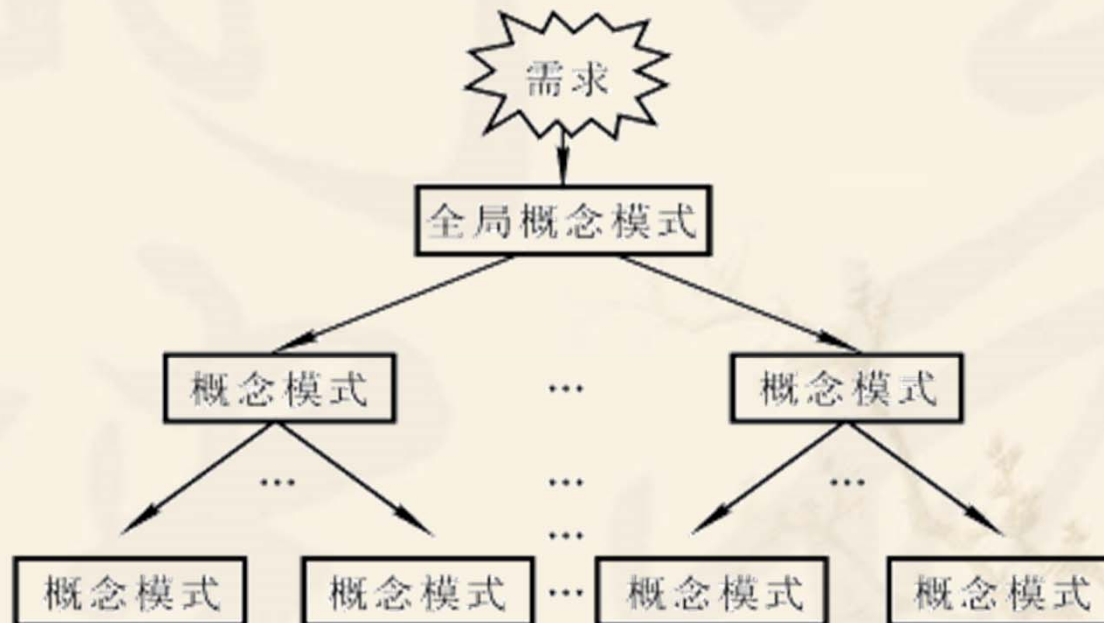
❧ 自底向上

❧ 逐步扩张

❧ 混合策略

❖ 自顶向下

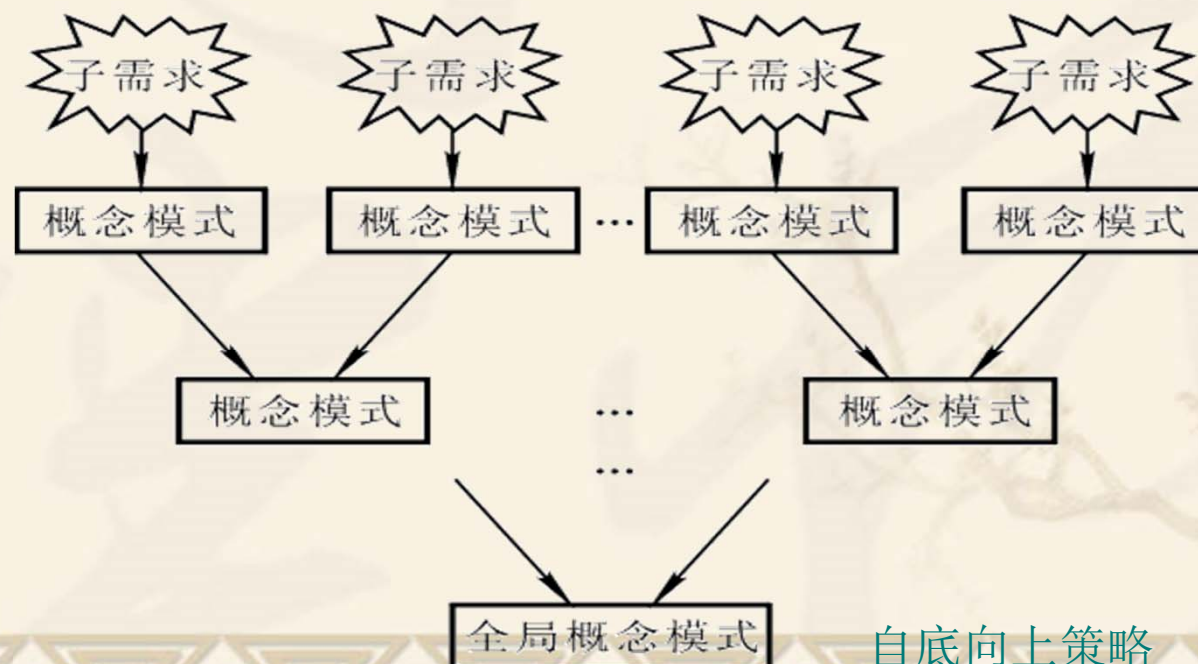
- 首先定义全局概念结构的框架，然后逐步细化



自顶向下策略

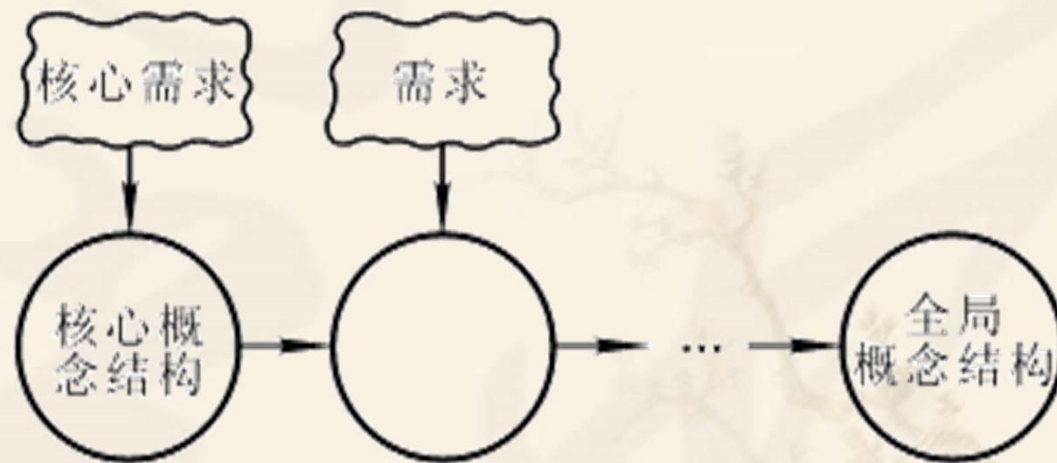
❖ 自底向上

- 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



❖ 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

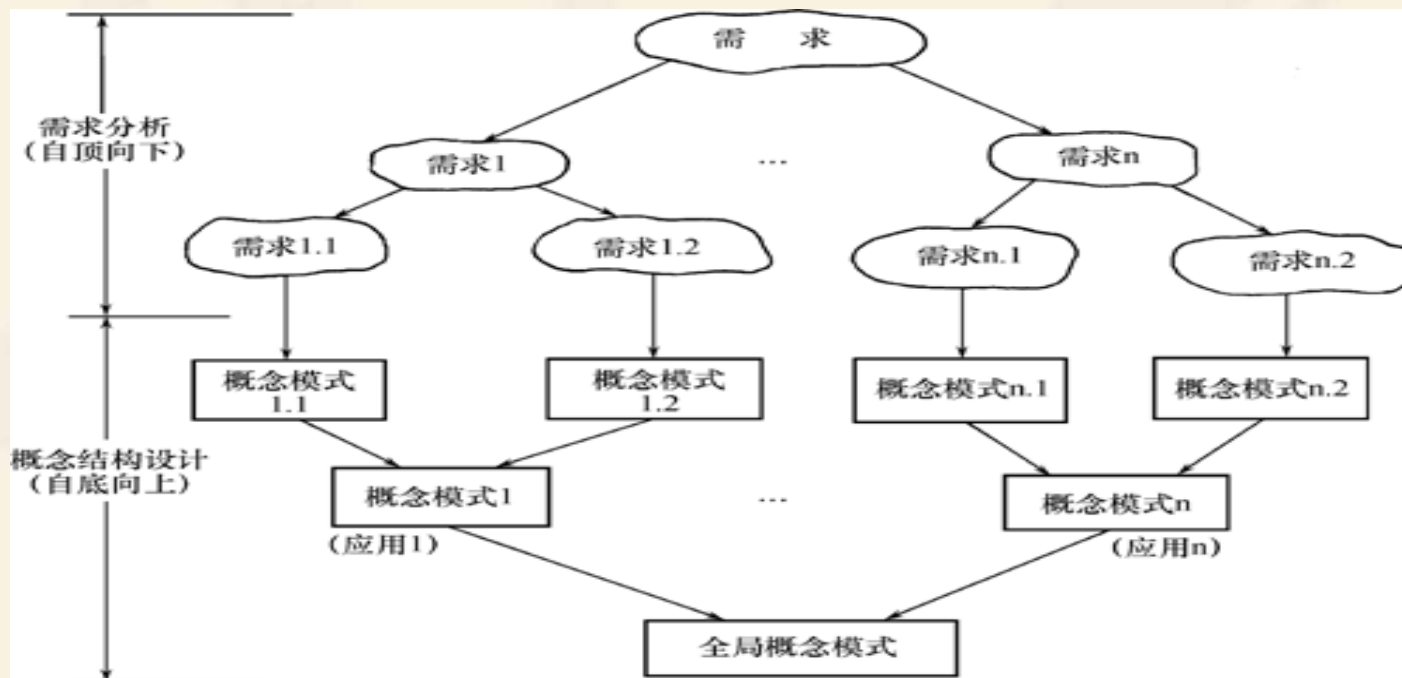
❖ 混合策略

✧ 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。

❖ 常用策略

❧ 自顶向下地进行需求分析

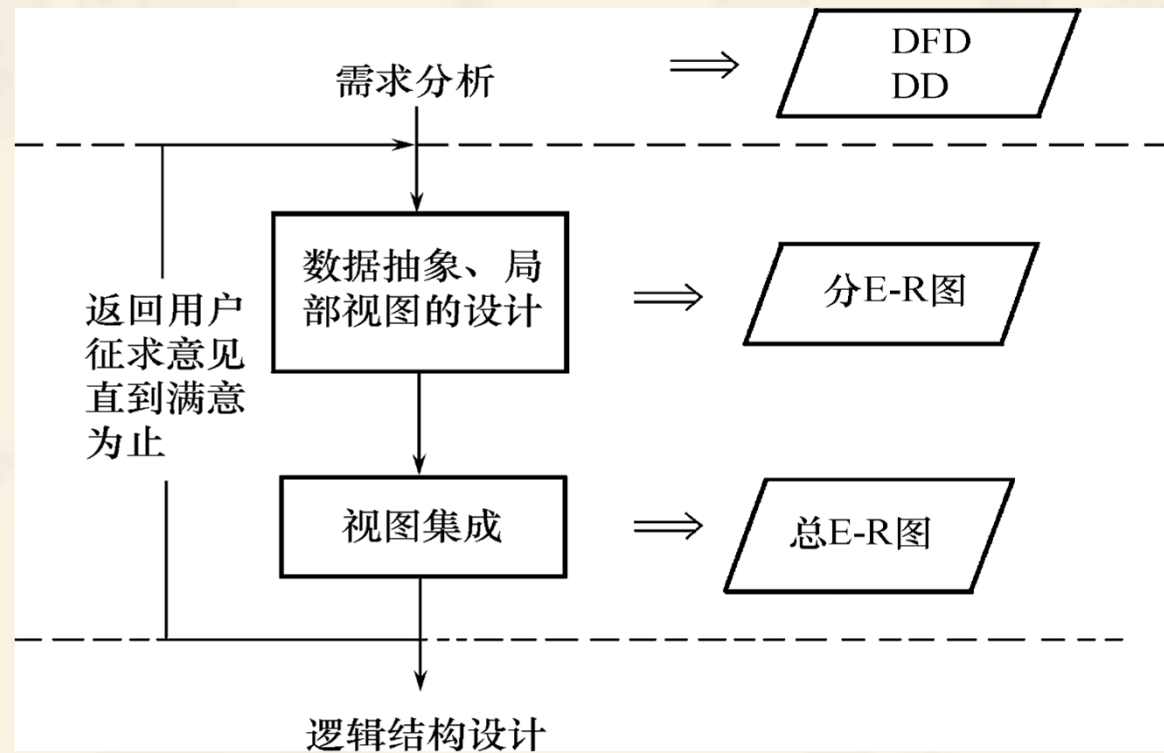
❧ 自底向上地设计概念结构



5.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



6.数据抽象

∞对实际的人、物、事和概念中抽取所关心的**共同特性**，忽略非本质的细节，并把这些特性用各种概念精确地加以描述。

∞三种常用抽象

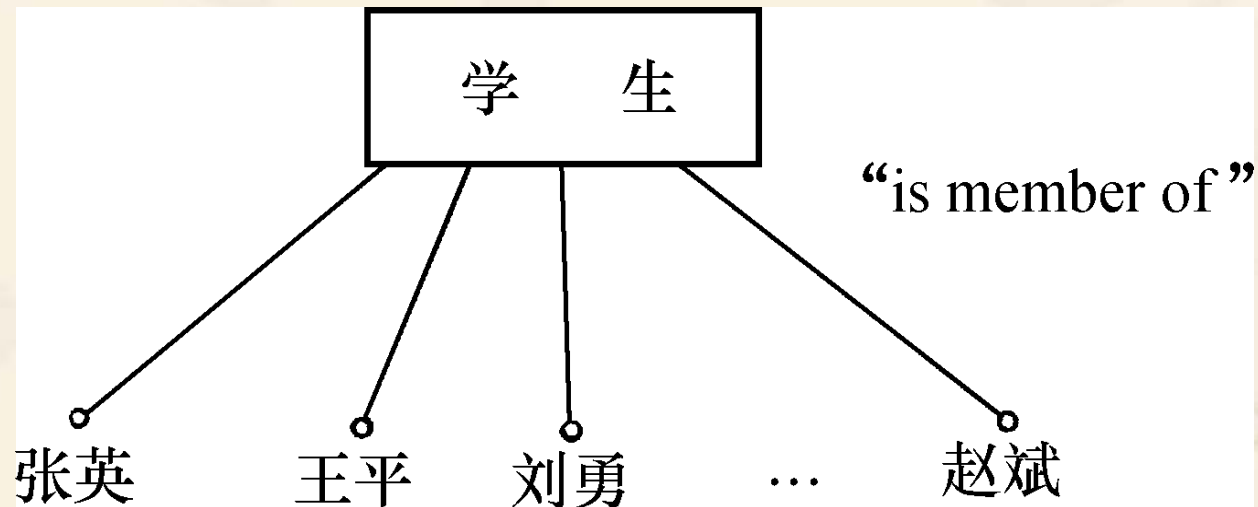
- 分类 (Classification)
- 聚集 (Aggregation)
- 概括 (Generalization)

数据抽象

(1) . 分类

定义某一类概念作为现实世界中**一组对象的类型**

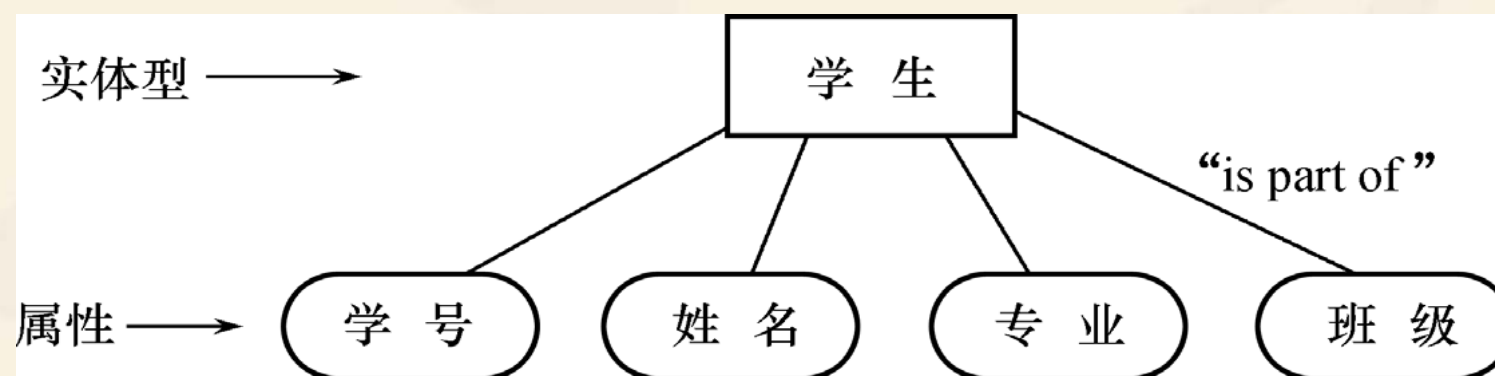
抽象了对象**值和型**之间的 “**is member of**” 的语义



数据抽象

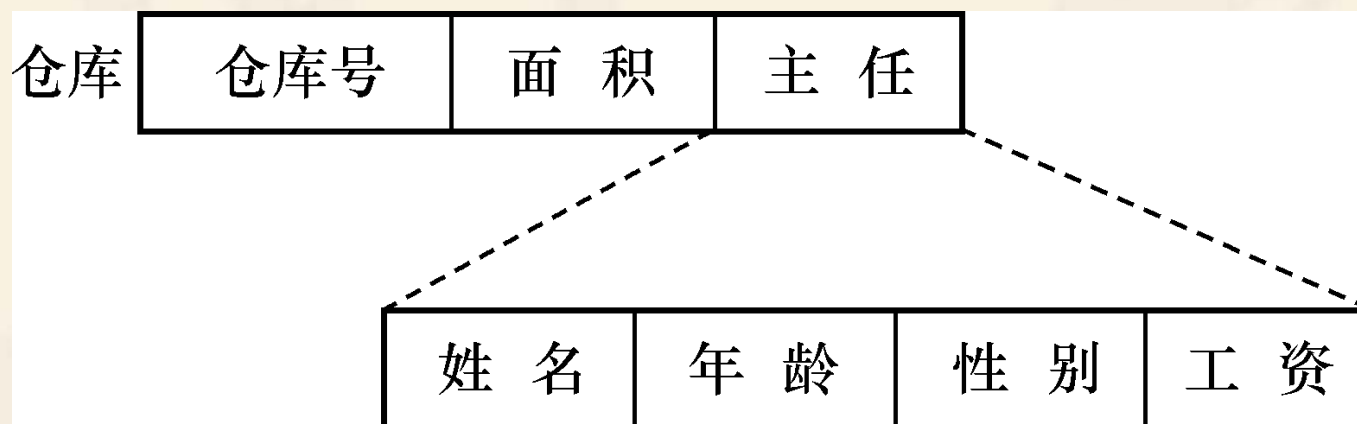
(2) . 聚集

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



数据抽象

- 复杂的聚集，某一类型的成分仍是一个聚集

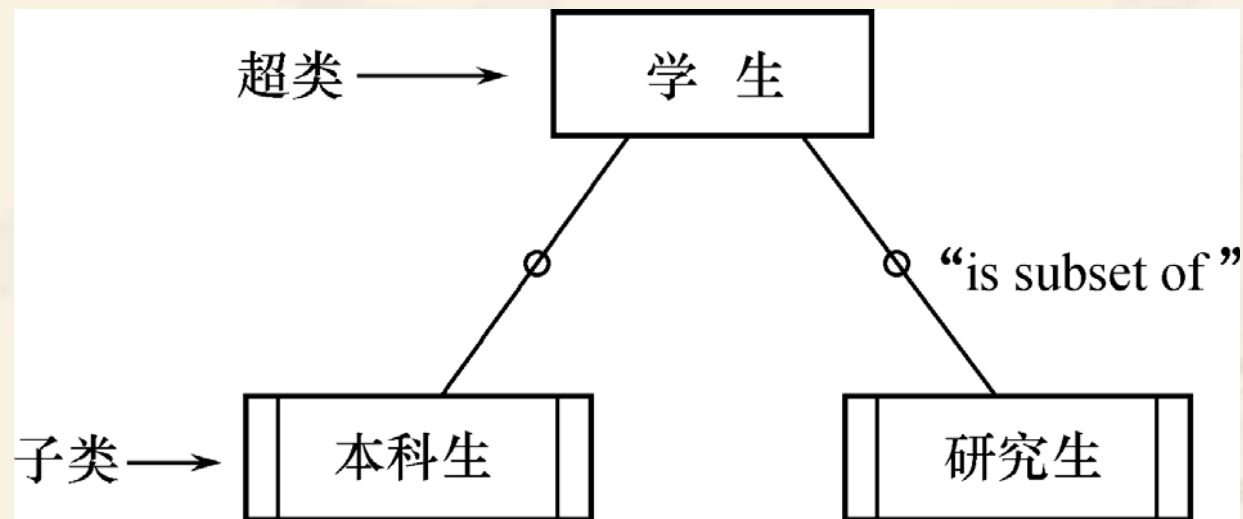


更复杂的聚集

数据抽象

(3) . 概括

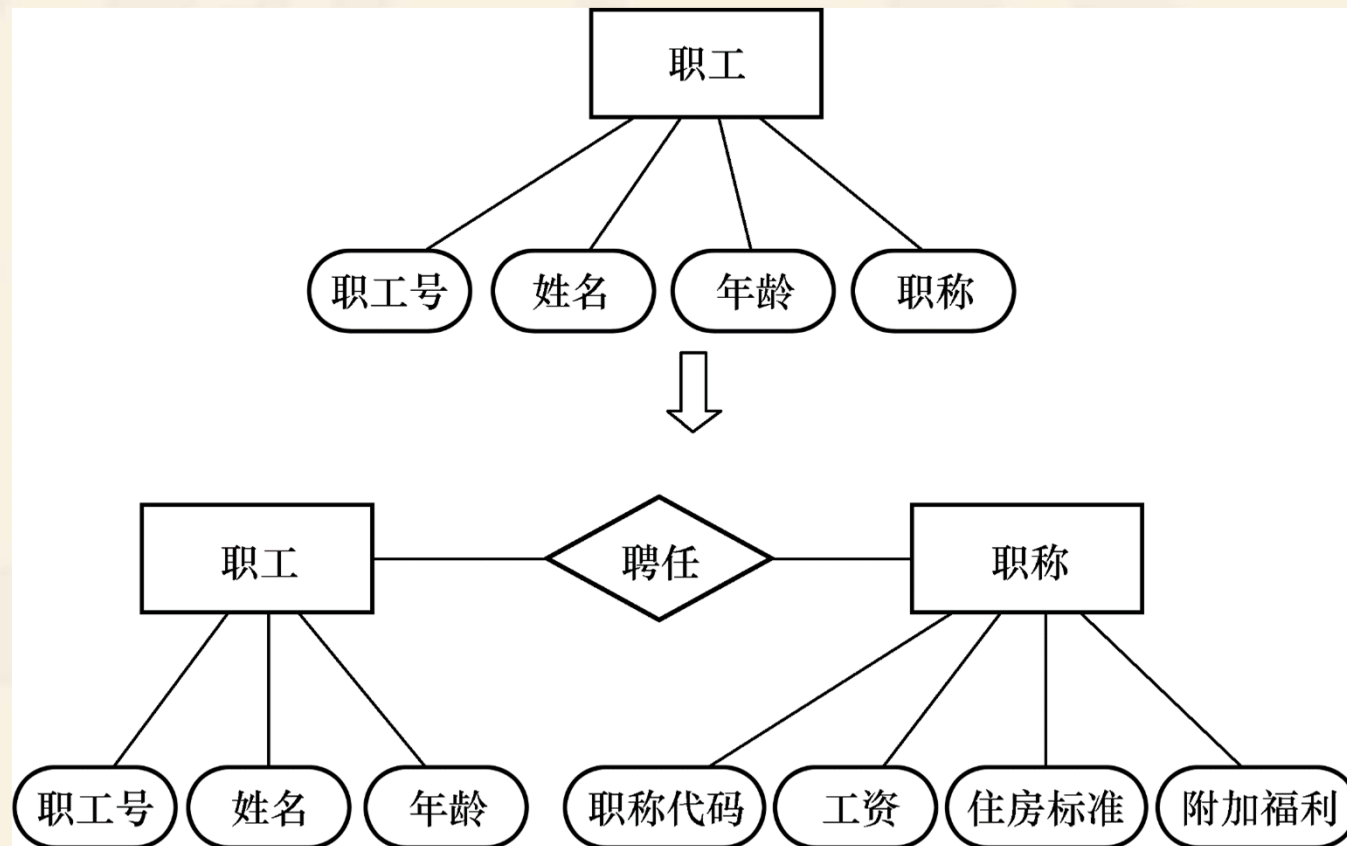
- 定义类型之间的一种子集联系
- 抽象了类型之间的 “is subset of” 的语义
- 继承性



7. 实体与属性的划分原则

- ❖ 为了简化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。**
- ❖ 两条准则：
 - (1) 属性,不能再具有需要描述的性质。即**属性必须是不可分的数据项，不能再由另一些属性组成**
 - (2) 属性不能与其他实体具有联系。**联系只发生在实体之间**

局部视图设计



职称作为一个实体

8.视图的集成

∞ 分E-R图集成为一个整体的数据概念结构即总E-R图

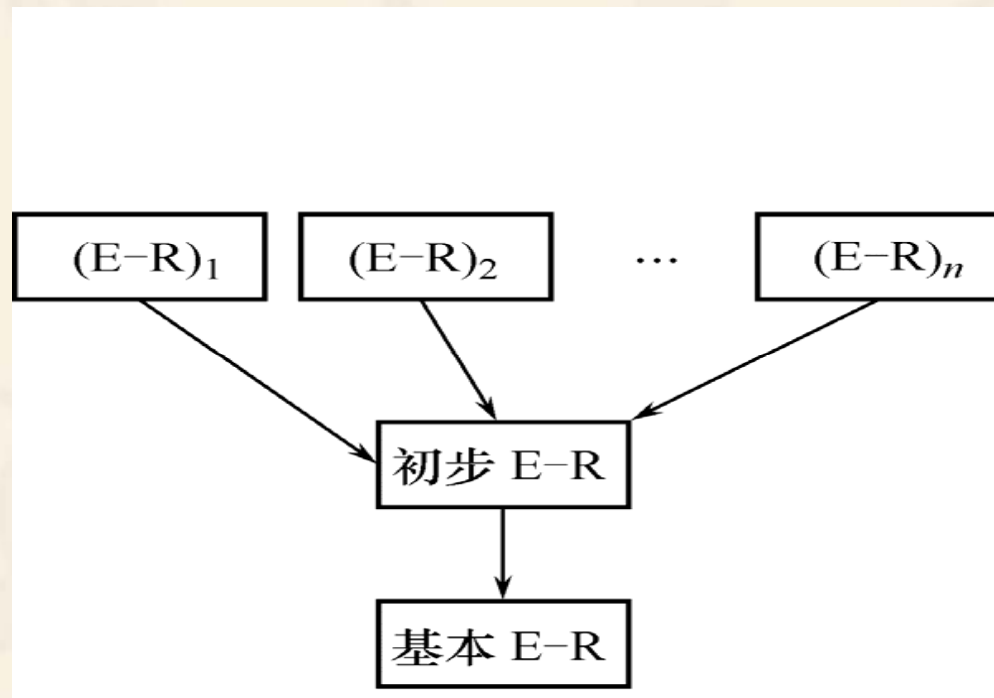
∞ 两种方式

➤ 多个分E-R图一次集成

➤ 逐步集成

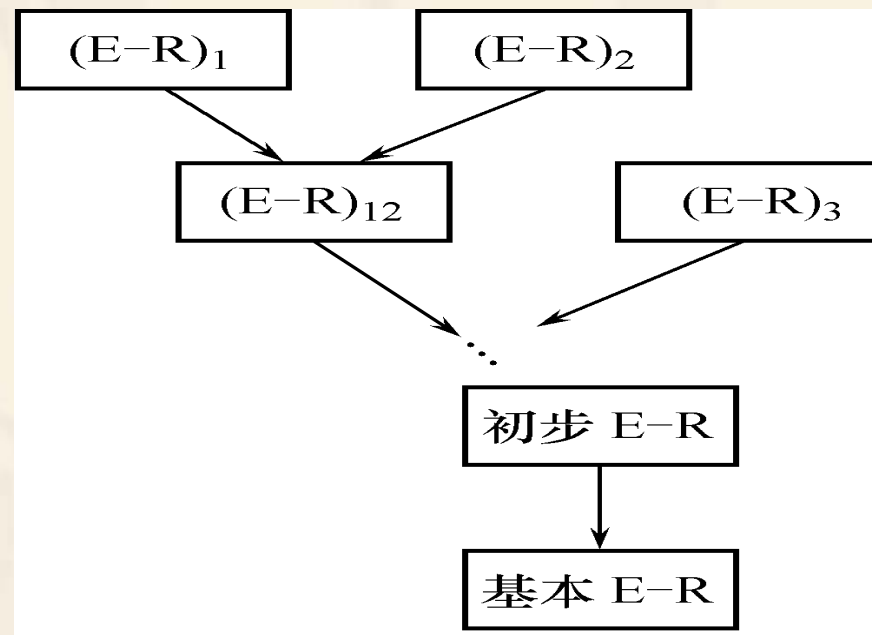
❖ 多个分E-R图一次集成

- 一次集成多个分E-R图
- 通常用于局部视图比较简单时



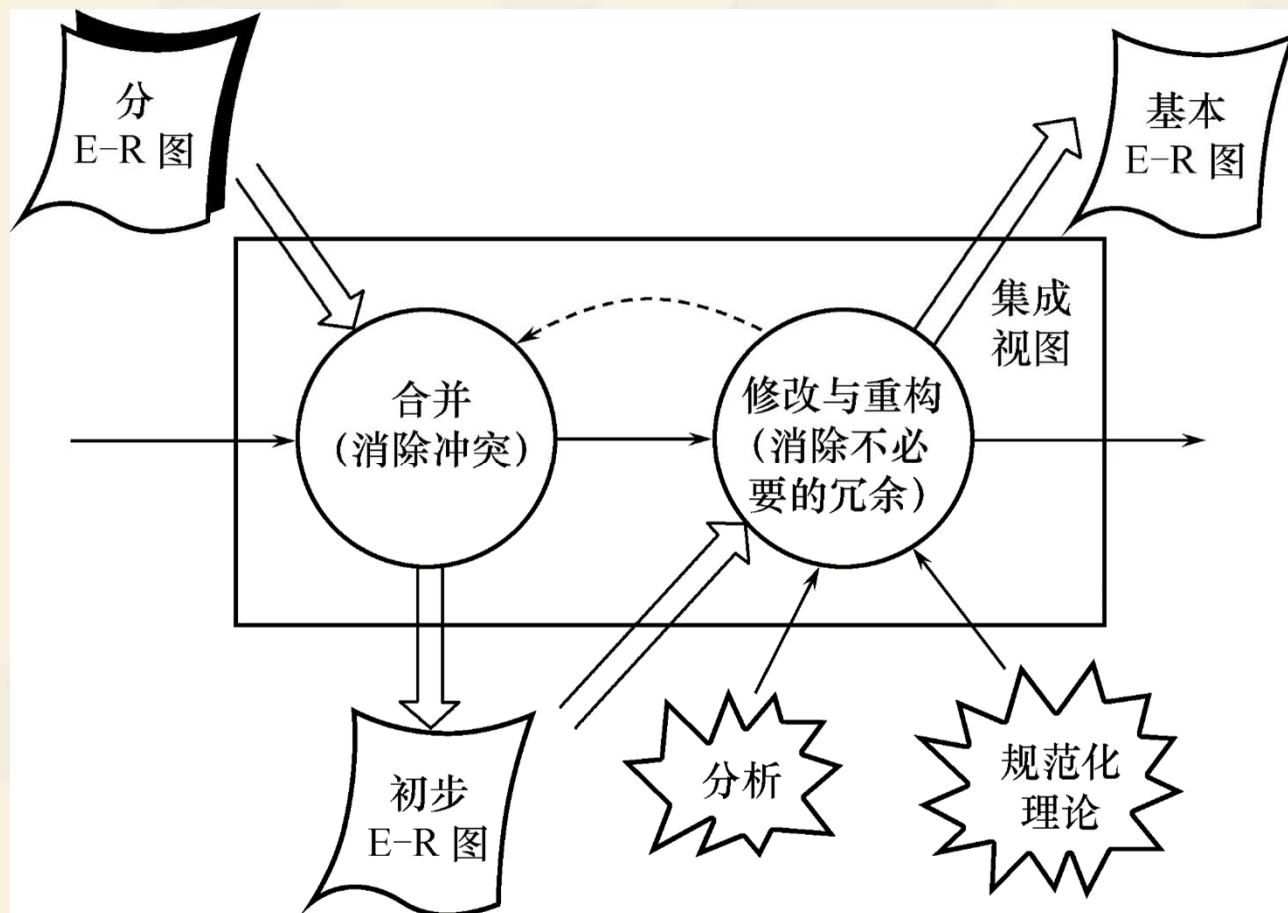
❖ 逐步集成

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

- (1) 合并：合理消除各分E-R图的冲突
- (2) 修改与重构：消除不必要的冗余



视图集成

❖ 冲突的种类

❧ 属性冲突

❧ 命名冲突

❧ 结构冲突

(1) 属性冲突

❖ 两类属性冲突

∞ 属性域冲突

- 属性值的类型
- 取值范围
- 取值集合不同

∞ 属性取值单位冲突

(2) 命名冲突

❖ 两类命名冲突

∞ **同名异义**：不同意义的对象在不同的局部应用中具有相同的名字

∞ **异名同义（一义多名）**：同一意义的对象在不同的局部应用中具有不同的名字

(3) 结构冲突

❖ 三类结构冲突

- ❧ 同一对象在不同应用中具有不同的抽象
- ❧ 同一实体在不同分E-R图所包含的属性个数和属性排列次序不完全相同
- ❧ 实体之间的联系在不同局部视图中呈现不同的类型（如P229-图7.25）

消除冗余

❖ 冗余

❧ 冗余的数据是指可由基本数据导出的数据

冗余的联系是指可由其他联系导出的联系

❧ 冗余数据和冗余联系容易破坏数据库的完整性，给数据库维护增加困难

❧ 消除不必要的冗余后的初步E-R图称为基本E-R图

消除冗余

❖ 消除冗余的方法

1、分析方法

- ❧ 以数据字典和数据流图为依据
- ❧ 根据数据字典中关于数据项之间的逻辑关系

2、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

9. 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**

- ❖ 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。

10.一个实例

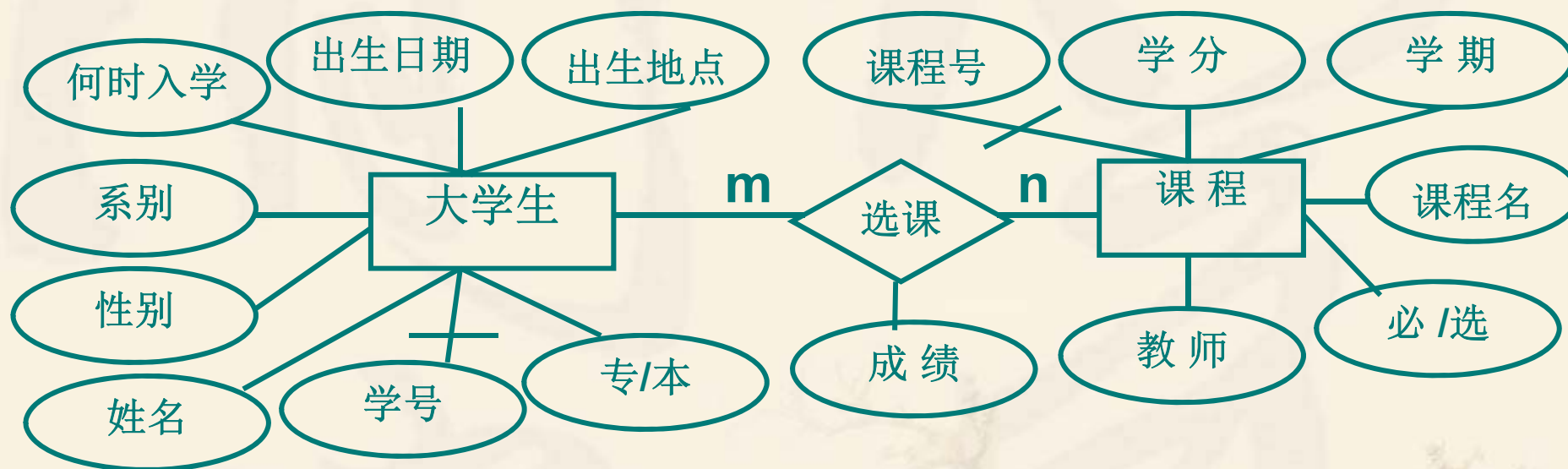


图7.5 教务处关于学生的视图

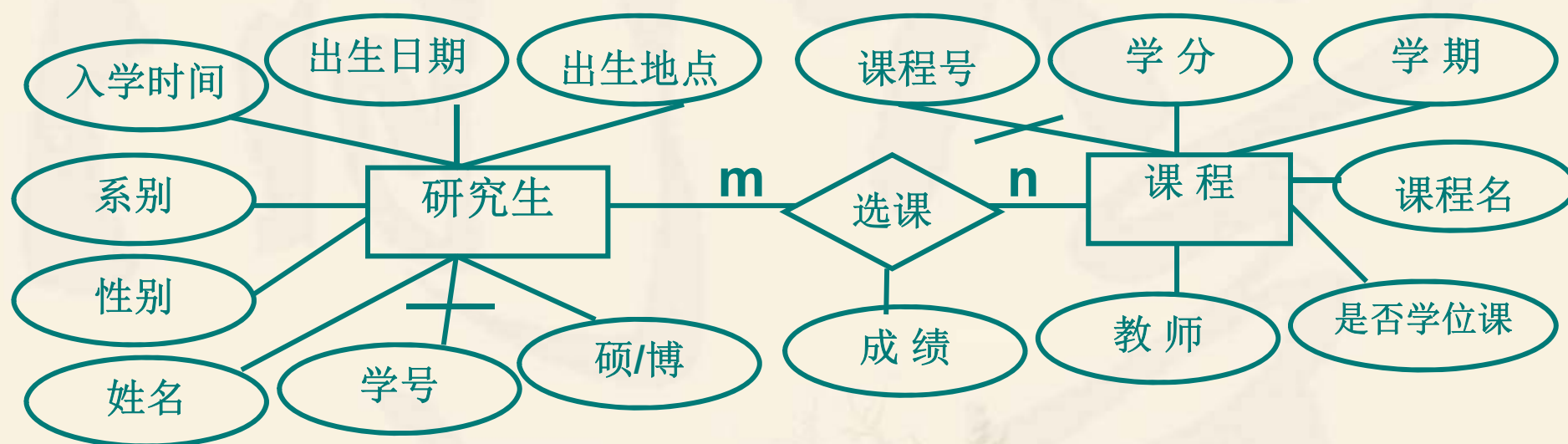


图7.6 研究生院关于研究生的视图

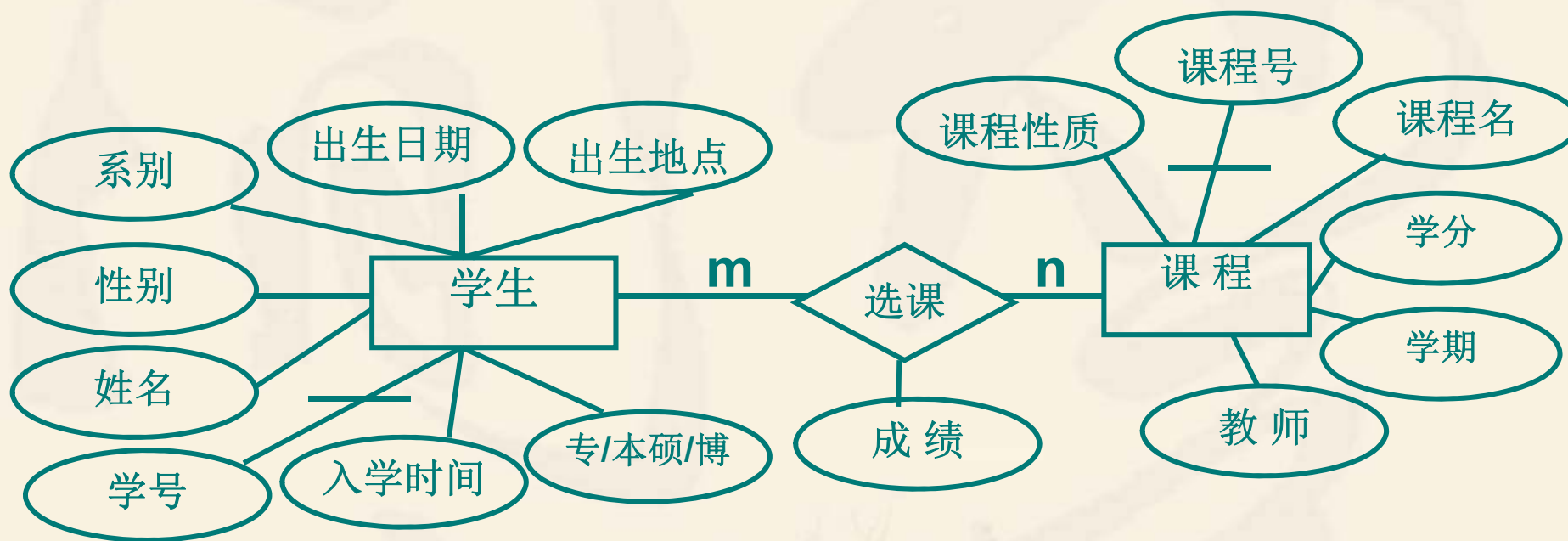


图7.7 两个视图集成

×××项目概念设计说明书

1. 前言

1.1 编写目的

1.2 背 景

1.3 名词定义

1.4 参考资料

2. 属性设计

2.1 属性一览

2.2 属性说明

3. 实体设计

3.1 实体一览

3.2 实体说明

3.3 候选键及主键

3.4 实体与属性关系表

4. 联系设计

4.1 联系一览

4.2 联系说明

5. E - R图

编写人员：_____ 审核人员：_____

审批人员：_____ 日 期：_____

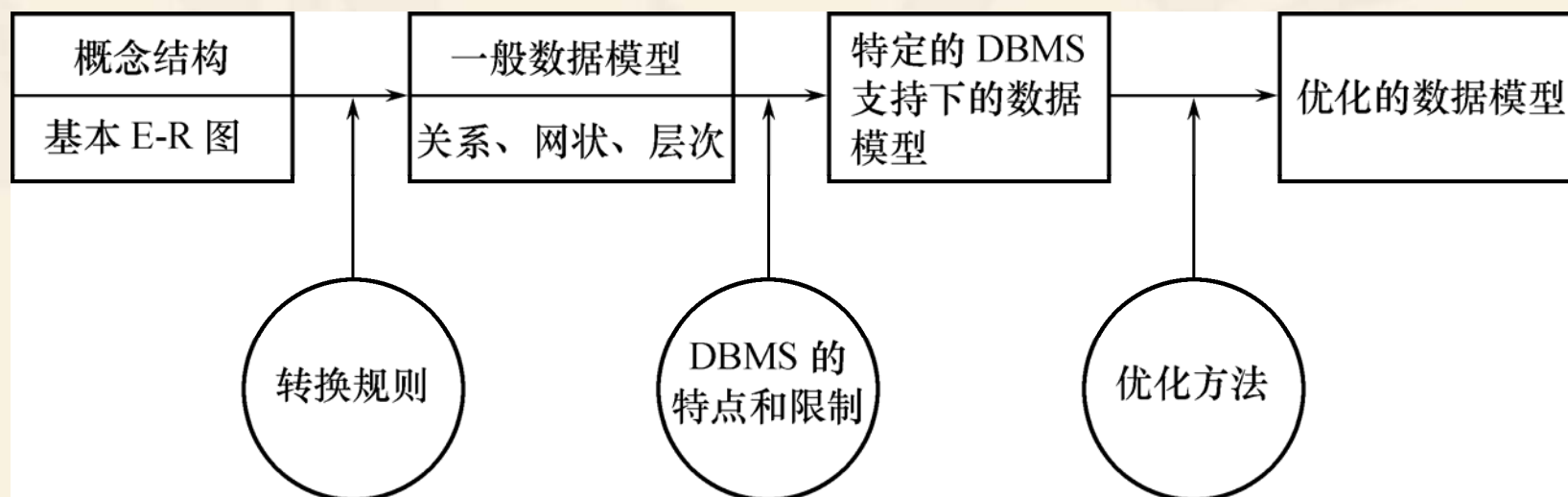
7.4 逻辑结构设计

1.逻辑结构设计的任务

- ❧ 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

2.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



逻辑结构设计的3个步骤

3. E-R图向关系模型的转换

∞ 将实体、实体的属性和实体之间的联系**转**
换为关系模式

实体型间的联系有以下不同情况：

(1) 一个1:1联系

- ❧ 转换为一个独立的关系模式
- ❧ 与某一端实体对应的关系模式合并

(2) 一个1:n联系

- ❧ 转换为一个独立的关系模式
- ❧ 与n端对应的关系模式合并

(3) 一个m:n联系转换为一个关系模式。

例，“选修”联系是一个m:n联系，可以将它转换为如下关系模式，其中学号与课程号为关系的组合码：

选修（学号，课程号，成绩）

(4) 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。

例，如下关系“讲授”联系是一个三元联系，可以将它转换为模式，其中课程号、职工号和书号为关系的组合码：

讲授（课程号，职工号，书号）

(5) 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

4. 数据模型的优化

- ❧ 数据库逻辑设计的结果不是唯一的
- ❧ 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
- ❧ 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

(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。

(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

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

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

(5) 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

∞ 常用分解方法

- **水平分解**
- **垂直分解**

5. 设计用户模式

❖ 定义用户模式时应该注重的方面包括三个方面：

- (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

×××项目逻辑设计说明书

1. 前言

1.1 编写目的

1.2 背景

1.3 名词定义

1.4 参考资料

2. 表设计

2.1 表一览

2.2 表结构

2.3 候选键、主键与外键

2.4 表说明

3. 属性设计

3.1 属性一览

3.2 属性说明

3.3 属性与表的关系

4. 数据约束设计

4.1 数据完整性设计

4.2 数据安全性设计

5. 视图设计

5.1 视图一览

5.2 视图说明

5.3 视图定义

编写人员：_____ 审核人员：_____
审批人员：_____ 日期：_____

7.5 物理结构设计

1. 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。**

2. 步骤

- ❧ 确定数据库的物理结构，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

显然，物理设计完全依赖于给定的硬件环境和数据库产品。

通常，关系数据库物理设计的内容主要包括为关系模式选择存取方法，以及设计关系、索引等数据库文件的物理存储结构。

3. 存取方法是快速存取数据库中数据的技术

常用的存取方法:

❖ 索引方法

- 目前主要是B+树索引和Hash索引
- 经典存取方法，使用最普遍

❖ 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

索引存取方法的选择

❖ 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

B+ 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(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

聚簇存取方法的选择

❖ 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。即对应每个聚簇码值的平均元组数不太少。太少了，聚簇的效果不明显

HASH存取方法的选择

❖ 选择HASH存取方法的规则

☞ 当一个关系满足下列两个条件时，可以选择HASH存取方法

➤ 该关系的属性主要出现在等值连接条件中或主要出现在相等比较选择条件中

➤ 该关系的大小可预知，而且不变；

或

该关系的大小动态改变，但所选用的DBMS提供了动态HASH存取方法

4. 存储结构设计

- ∞ 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，确定系统配置等。
- ∞ 综合考虑存取时间、存储空间利用率和维护代价三方面的因素。
- ∞ 为提高性能，可根据应用情况将数据的易变与稳定部分、经常存取部分和存取频率较低部分分开存放。

×××项目物理设计说明书

- 1. 前言
 - 1.1 编写目的
 - 1.2 背 景
 - 1.3 名词定义
 - 1.4 参考文献
- 2. 存取方法选择
 - 2.1 索引设计
 - 2.1.1索引一览
 - 2.1.2 索引说明
 - 2.1.3 索引定义
 - 2.2 集簇设计
 - 2.2.1 集簇一览
 - 2.2.2 集簇说明
 - 2.2.3 集簇定义
 - 2.3 HASH设计
 - 2.3.1 HASH一览
 - 2.3.2 HASH说明
 - 2.3.3 HASH定义
- 3. 存储结构设计
 - 3.1 分区设计
 - 3.1.1 分区一览
 - 3.1.2 分区说明
 - 3.1.3 分区定义
 - 3.2 系统参数配置
 - 3.2.1 系统参数配置一览
 - 3.2.2 系统参数配置说明
 - 3.2.3 系统参数配置定义

编写人员：_____ 审核人员：_____
审批人员：_____ 日 期：_____

7.6 数据库实施与维护

实施阶段重要工作：一是数据载入；二是应用程序的编码与调试。

1、数据的载入

- ❧ 数据库实施阶段最主要的工作

- ❧ 数据装载方法

 - 人工方法（筛选，转换格式，输入，校验）

 - 计算机辅助数据入库

2、应用程序的编码与调试

- ❧ 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

3、数据库的试运行（又称联合调试）

∞ 主要工作

- 功能测试：实际运行应用程序，执行对数据库的各种操作，测试应用程序的各种功能；
- 性能测试：测试系统的性能指标，分析是否符合设计目标。

∞ 分期分批组织数据入库

∞ 由于系统不稳定，人员不熟悉，因此需要做好数据库的转储和恢复

4、数据库的运行与维护

∞ 长期任务

∞ 由DBA完成

- ❖ 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
 - ❖ 数据库的重组，并不修改原计划的数据逻辑和物理结构
 - ❖ 重构造，需要调整数据库的模式和内模式，当然数据库的重构也是有限的，只能做部分修改。

7.7 数据库设计实例（自学）

1、需求分析

(1) 需求调查

该数据库设计的客观世界目标对象是一个期刊编排系统，经调查其数据边界、环境及内部关系是：

- ❖ 一个期刊社有若干个编辑部与若干个排版部，它们分别负责对期刊的编辑与排版。
- ❖ 由编辑部的编辑人员对稿件作编辑并决定稿件收录的期刊、刊次与栏目。
- ❖ 由排版部的设计人员负责期刊的设计排版工作。
- ❖ 该期刊编排系统与外界环境无关。

(2) 数据字典

根据需求调查结果，该系统的数据字典是：

❖ **数据类：** 期刊编排系统共有6个数据类。

❧ 期刊编排C1：部门

❧ 期刊编排C2：职工

❧ 期刊编排C3：期刊

❧ 期刊编排C4：稿件

❧ 期刊编排C5：编辑

❧ 期刊编排C6：排版

❖ **数据项：**

❧ 期刊编排系统共有27个数据项。

数据项表

编号	数据项名	对应数据类	数据类型、长度	附注
I1	编号	C1	整型	非空
I2	名称	C1	字符、可变长50	非空
I3	负责人	C1	字符、可变长50	
I4	电话	C1	字符、可变长24	
I5	编号	C2	整型	非空
I6	姓名	C2	字符、可变长30	
I7	年龄	C2	整型	
I8	职务	C2	字符、可变长20	
I9	性别	C2	字符、固定长度2	
I10	期刊编号	C3	字符、固定长度8	非空
I11	期刊名称	C3	字符、可变长度50	
I12	设计排版人员号	C3	整型	
I13	发行量	C3	整型	
I14	出版日期	C3	日期	

I15	编号	C4	整型	非空
I16	标题	C4	字符、可变长度150	
I17	作者	C4	字符、可变长度30	
I18	正文	C4	字符	
I19	字数	C4	整型	
I20	编辑人员号	C5	整型	非空
I21	稿件编号	C5	整型	
I22	完成日期	C5	日期	
I23	所属栏目	C5	字符、可变长40	
I24	部门编号	C6	整型	非空
I25	期刊编号	C6	字符、固定长8	
I26	完成日期	C6	日期	
I27	排版人员号	C6	整型	

(3) 语义约束

该期刊编排系统遵循有如下的约束：

- ❖ 期刊编排有若干个部门，它包括若干编辑部与若干排版部。
- ❖ 期刊编排有若干个职工，他们每人工作于一个部门，每个部门有若干人。
- ❖ 若干的编辑部中每个编辑负责若干稿件，每个稿件仅由一个编辑部的编辑负责。
- ❖ 期刊编排中的排版部有若干人员与若干期刊。每期期刊由多篇稿件组成，并由一个部门负责排版设计。
- ❖ 人员性别非男即女。
- ❖ 人员年龄为18-60。

2、概念设计

根据需求分析，在概念设计阶段采用E-R方法与视图集成法。

(1) 分解

首先对期刊编采作分解，它可分解为如下三个视图：

- ❖ 人员组织；
- ❖ 稿件编辑；
- ❖ 设计排版；

(2) 视图设计

对每个视图作E-R图。

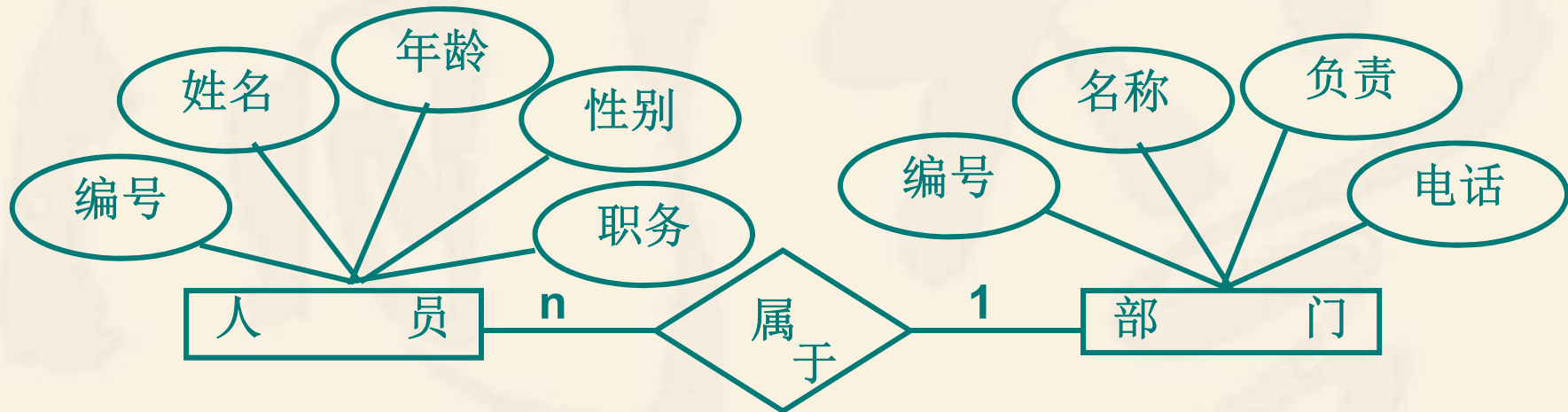


图7.12 人员组织E-R图

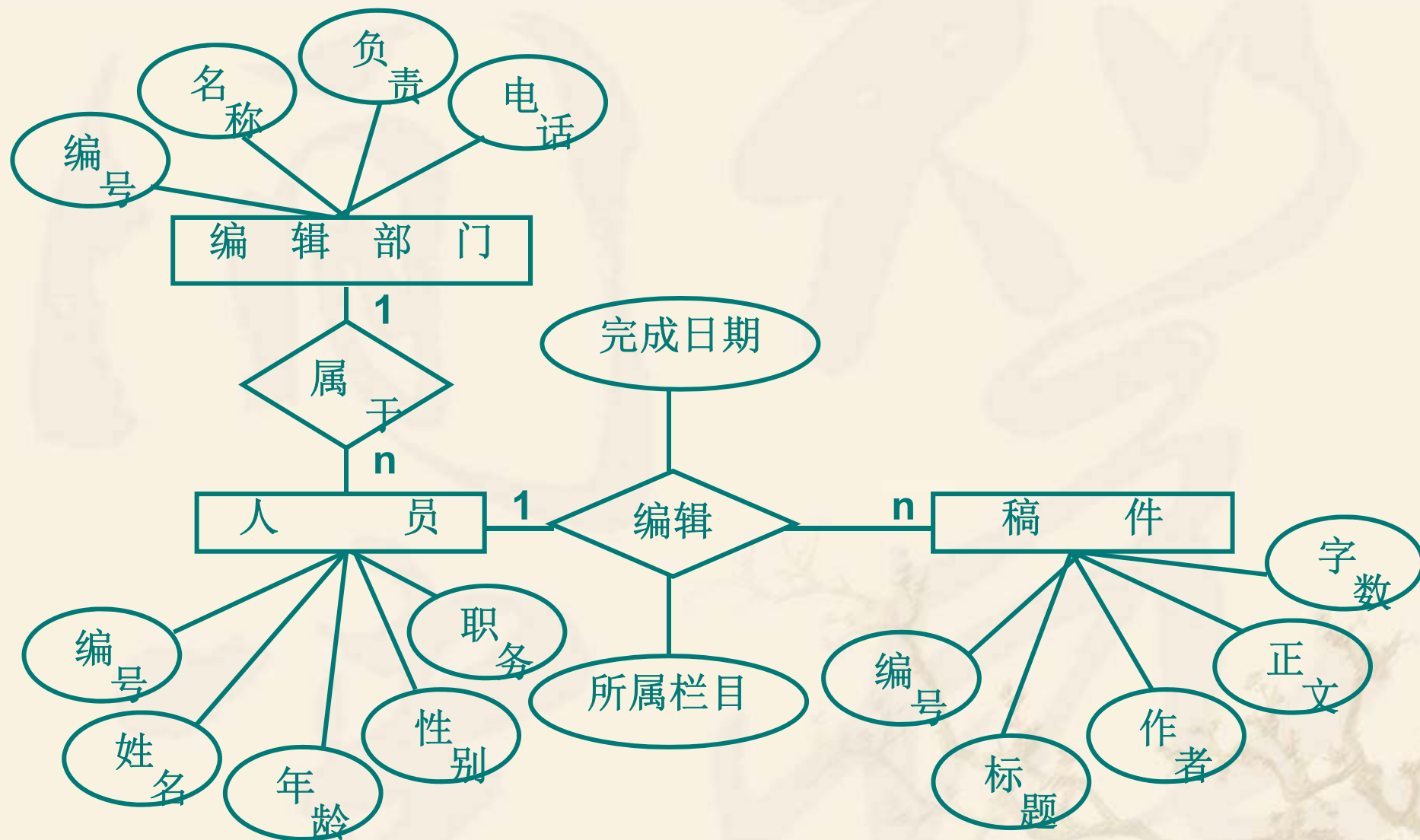


图7.13 稿件编辑

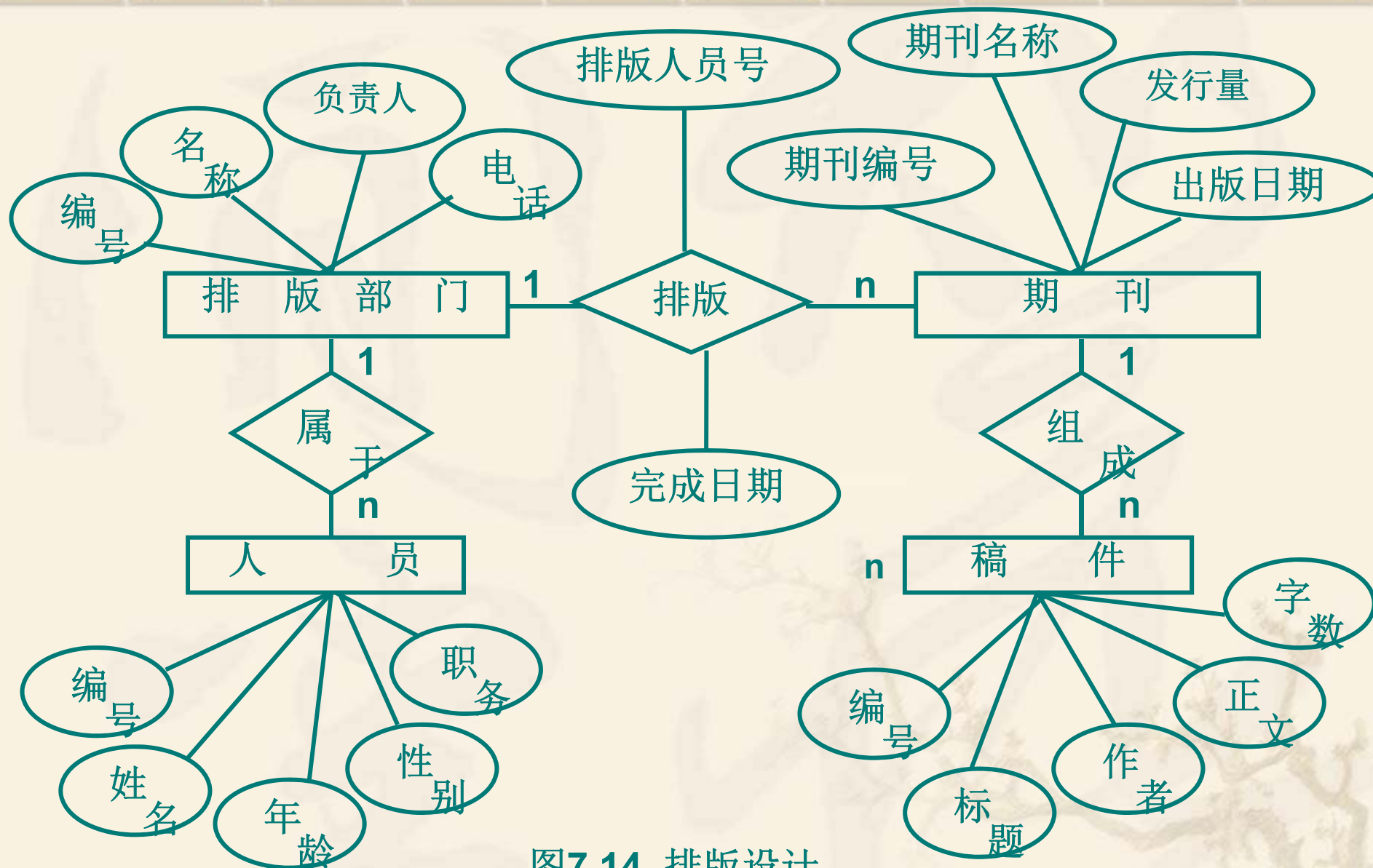


图7.14 排版设计

(3) 视图集成

视图设计后作视图集成，最终形成全局E-R图。在集成过程中存在冲突，它们是：

- ❖ 在实体集中编辑部与排版部均为期刊社的部门，因此可以合并为：部门。在合并中涉及属性的不一致与冲突，需作调整。
- ❖ 在集成中存在着属性的冲突性，如“完成日期”有冲突，“编号”有冲突，它们均须作调整。

经调整后最终形成全局的E-R图为图7.15所示。

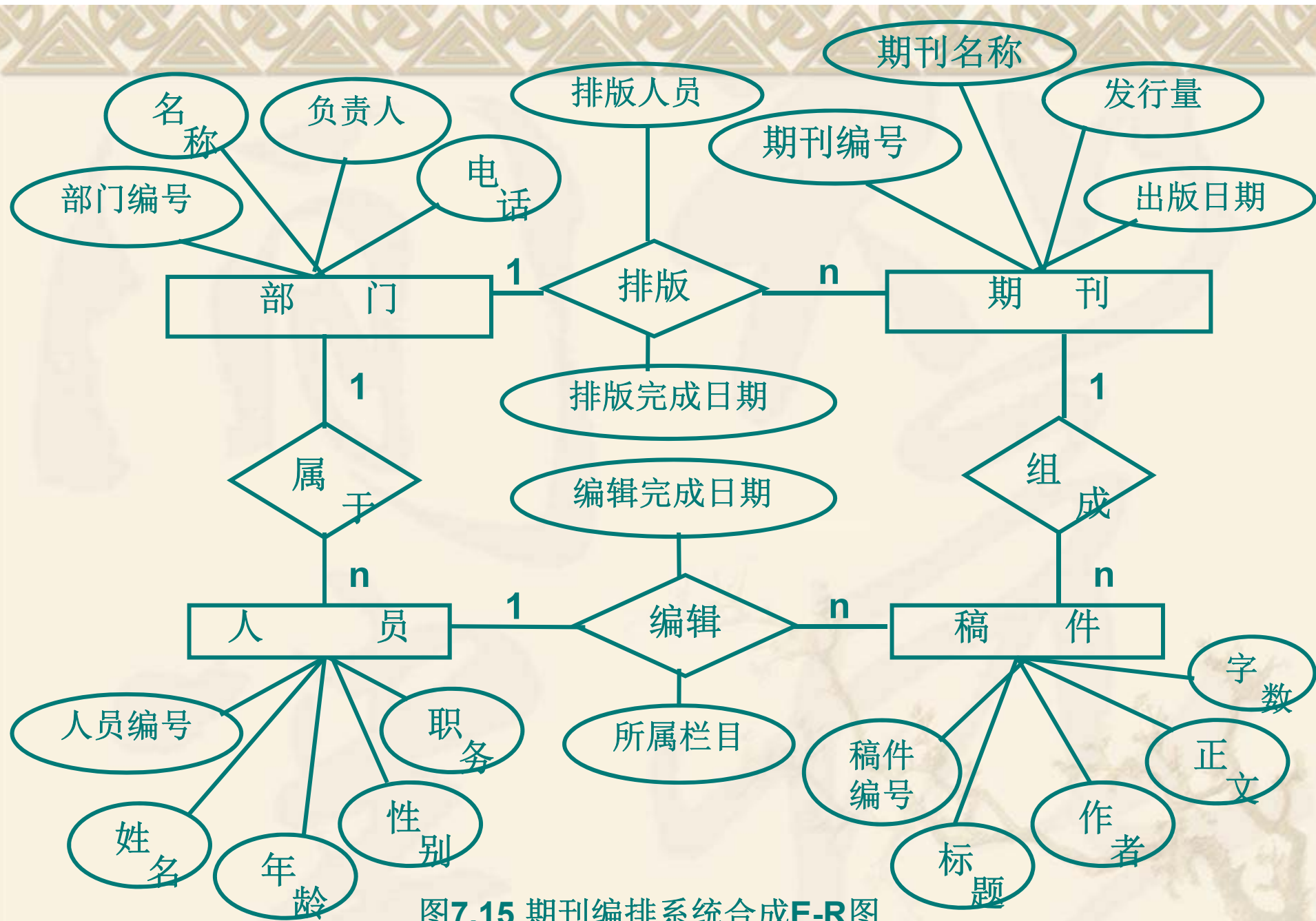


图7.15 期刊编排系统合成E-R图

3、逻辑结构设计

在概念设计基础上可以作逻辑设计，将全局E-R图转换成关系模式，这些模式均应符合第三范式，同时设计数据完整性约束与关系视图。

(1) 关系模式

在全局E-R图中有四个实体集，它们可构成四个关系，同时有四个联系。它们均为1:n联系。因此它们可以归并至相应四个关系中，从而组成如下的表。

- ❖ 部门 (部门编号, 部门名称, 负责人, 电话)
- ❖ 人员 (职工编号, 姓名、性别、年令、职务、部门编号)
- ❖ 期刊 (期刊编号, 期刊名称、出版日期、部门编号、排版人员号、排版完成日期、发行量)
- ❖ 稿件 (稿件编号, 标题、作者、字数、正文、编辑人员编号、编辑完成日期、期刊编号、所属栏目)

(2) 数据模式规范化

上述四个表都满足第三范式。

(3) 数据完整性

- ❖ 主键：已在关系模式中确定—分别是部门编号，职工编号，期刊编号及稿件编号。
- ❖ 外键：
 - ❧ 人员关系中的部门编号
 - ❧ 期刊关系中的部门编号
 - ❧ 稿件关系中的编辑人员编号及期刊编号

❖ 用户定义完整性:

- ❧ “人员” 中的 “性别” 值非男即女。
- ❧ “人员” 中的 “年龄” 约束为18-60。

(4) 关系中的视图

可在人员中构作: “编辑人员” 与 “排版人员” 两个视图。

4、物理结构设计

物理设计主要是建立索引，它包括：

- ❖ 在四个表的主键中建立索引。
- ❖ 在“人员”中的“年龄”和“稿件”中的“字数”

以及“期刊”的“发行量”上分别建立索引以提高
作统计时的运行效率。

本章小结

❖ 需求分析

- ❧ 需求调查

- ❧ 数据字典

- ❧ 需求分析说明书

❖ 概念结构设计

- ❧ E-R图

- ❧ 概念设计说明书

❖ 逻辑结构设计

- ❧ E-R图转换成关系模型

- ❧ 逻辑设计说明书

❖ 物理设计

- ❧ 存取方法设计

- ❧ 存取结构设计

- ❧ 物理设计说明书

课堂练习

- ❖ 设某商业集团数据库中有三个实体集。一是“**商店**”实体集，属性有商店编号、商店名、地址等；二是“**商品**”实体集，属性有商品号、商品名、规格、单价等；三是“**职工**”实体集，属性有职工编号、姓名、性别、业绩等。
 - ❖ 商店与商品间存在“销售”联系，每个商店可销售多种商品，每种商品也可放在多个商店销售，每个商店销售一种商品，有月销售量；商店与职工间存在着“聘用”联系，每个商店有许多职工，每个职工只能在一个商店工作，商店聘用职工有聘期和月薪。
- (1) 试画出ER图，并在图上注明属性、联系的类型。
 - (2) 将ER图转换成关系模型，并注明主键和外键。

