第5章 数据库设计

5.1 数据库设计概述

5.2 需求分析

5.3 概念设计

5.4 逻辑设计

5.5 物理设计

5.6 IDEF方法简介

5.7 计算机辅助数据库设计

5.8 数据库实施

5.9 数据库运行与维护

5.1 概述

- 数据库设计是数据库生命周期中的最重要的阶段，其好坏直接影响整个数据库系统的成败
- 数据库设计是指在一个特定应用环境下，根据用户的信息需求、处理需求和数据库支撑环境（包括DBMS、OS和硬件）， 构造最合理的数据库模式，创建数据库及其相关应用系统的过程

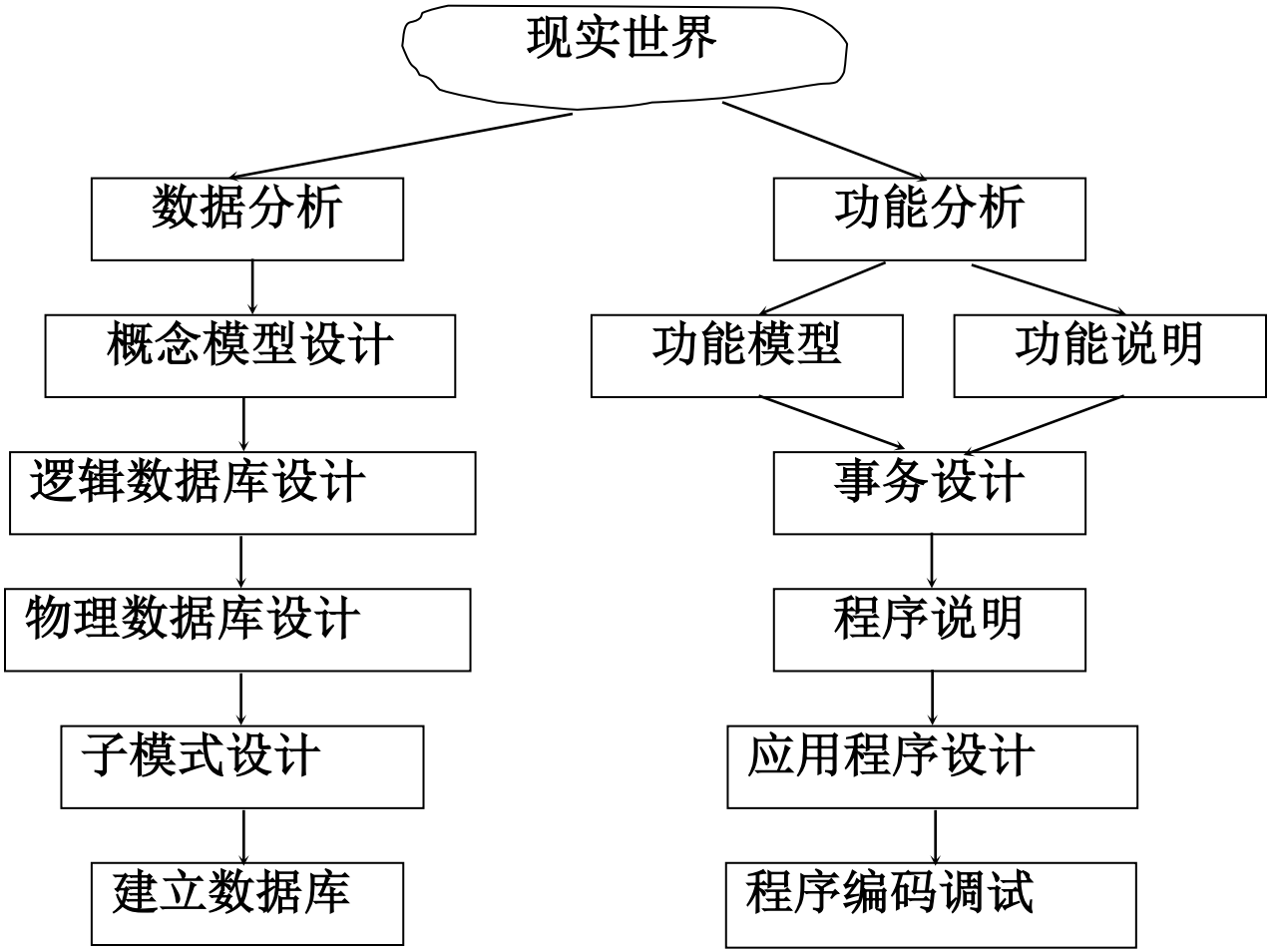
数据库系统生命周期

- 指数据库应用系统从开始规划、分析、设计、实现、投入运行后的维护到最后被新的系统所取代而停止使用的整个期间
- 数据库系统的生存期：
  （1）系统规划：
  （2）数据库设计：
  需求分析，概念设计，逻辑设计，物理设计
  （3）系统实现：
  应用程序编码、调试
  （4）运行和维护：

数据库设计的特点

- 结构（数据）设计和行为（处理）设计相结合
  - 将数据库结构设计和数据处理设计密切结合

数据库设计的特点



结构和行为分离的设计

数据库设计方法

- 数据库设计主要有以下两种方法：
  1) 面向数据的设计方法(data-oriented approach)：以信息需求为主，兼顾处理需求;
  2) 面向过程的设计方法(process-oriented approach)：以处理需求为主，兼顾信息需求;
- 一般来说，数据相对比较稳定，而处理则相对容易变动，所以为了设计一个相对稳定的数据库，数据库设计主要采用面向数据的设计方法

数据库设计方法

- 手工与经验相结合方法
  - 设计质量与设计人员的经验和水平有直接关系
  - 数据库运行一段时间后常常不同程度地发现各种问题，增加了维护代价
- 规范设计法
  - 基本思想：过程迭代和逐步求精

数据库设计方法

- 新奥尔良（New Orleans）方法
  - 将数据库设计分为若干阶段和步骤
- 基于E-R模型的数据库设计方法
  - 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
- IDEF 1X方法
- ODL（Object Definition Language）方法
  - 面向对象的数据库设计方法

数据库设计的基本步骤

- 数据库设计分6个阶段
  - 系统规划
  - 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

数据库设计的基本步骤

1.需求分析阶段

- 准确了解与分析用户需求（包括数据与处理）
- 最困难、最耗费时间的一步

2.概念结构设计阶段

- 整个数据库设计的关键
- 通过对用户需求进行综合、归纳与抽象，形成一个独立于具体DBMS的概念模型

数据库设计的基本步骤

3.逻辑结构设计阶段

- 将概念结构转换为某个DBMS所支持的数据模型
- 对其进行优化

4.数据库物理设计阶段

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

数据库设计的基本步骤

5.数据库实施阶段

- 运用DBMS提供的数据库语言（如SQL）及宿主语言，根据逻辑设计和物理设计的结果
  - 建立数据库
  - 编制与调试应用程序
  - 组织数据入库
  - 进行试运行

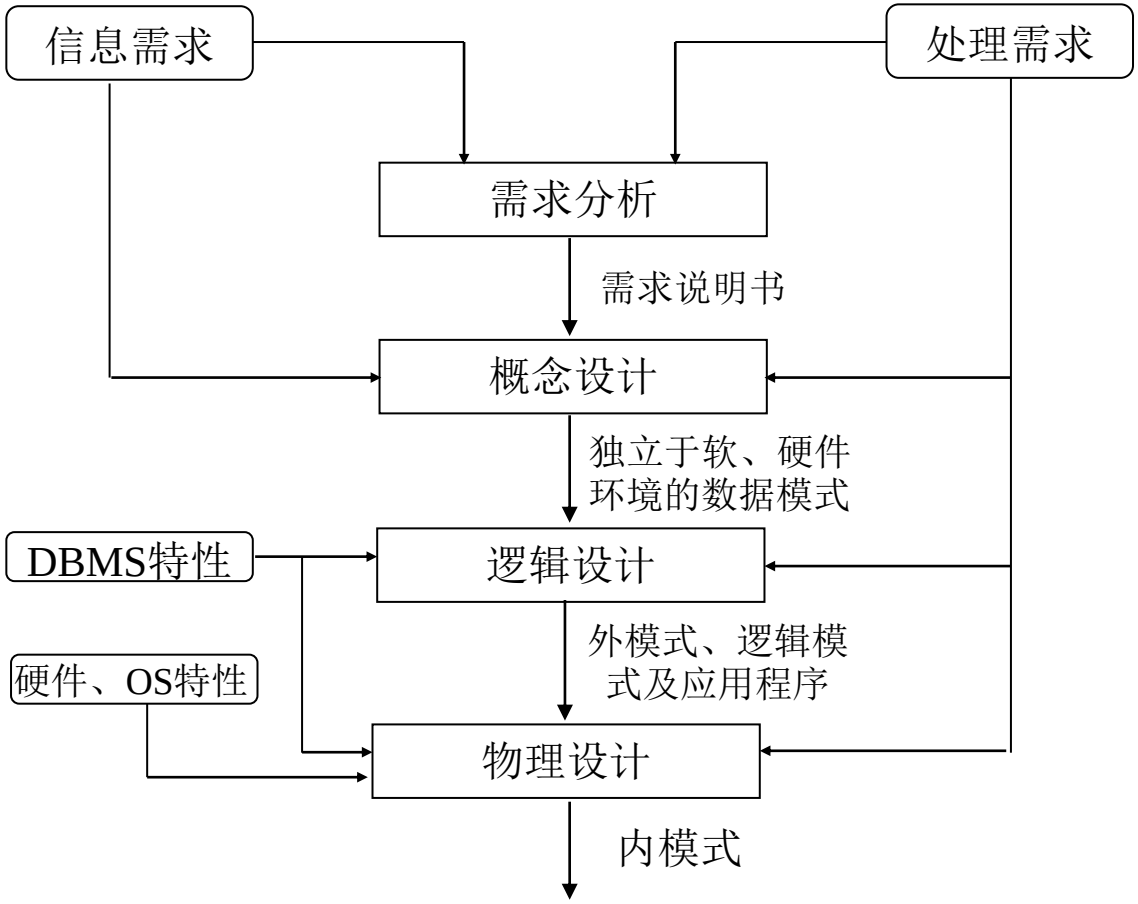
数据库设计的基本步骤

6.数据库运行和维护阶段

- 数据库应用系统经过试运行后即可投入正式运行
- 在数据库系统运行过程中必须不断地对其进行评价、调整与修改

- 设计一个完善的数据库应用系统往往是上述六个阶段的不断反复

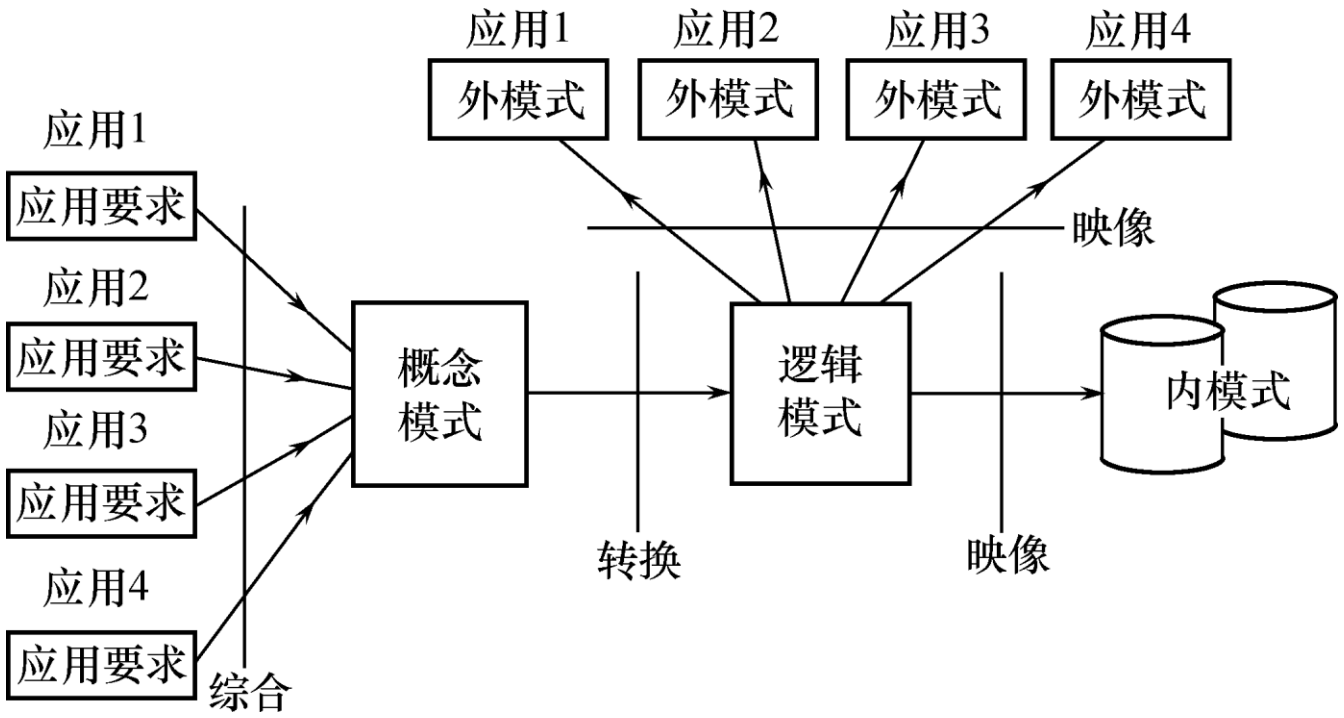
数据库设计的基本步骤



数据库设计基本过程

数据库设计过程中的各级模式

数据库设计不同阶段形成的数据库各级模式



数据库的各级模式

5.2 需求分析

- 需求分析阶段主要负责收集用户的信息需求及处理需求，并加以分析、整理、统一，最终形成用户需求分析说明书

需求分析的任务

- 详细调查现实世界要处理的对象（组织、部门、企业等）
- 充分了解原系统（手工系统或计算机系统）
- 明确用户的各种需求
- 确定新系统的功能
- 充分考虑今后可能的扩充和改变
- 编写需求分析说明书

需求分析的重点

- 需求分析的重点是用户的“数据”和“处理”，即获得用户对数据库的“真实”要求，包括：
  - 信息要求
  - 处理要求
  - 安全性与完整性要求

需求分析的难点

- 确定用户的真实需求，原因：
  - 用户缺少计算机知识
  - 设计人员缺少用户的专业知识
- 解决方法
  - 设计人员必须不断深入地与用户进行交流

需求分析的步骤

1. 需求信息的收集（了解用户需求）
   (1) 信息需求,用户要从数据库获得的信息内容
   (2) 处理需求,完成什么处理功能及处理方式
   (3) 安全性和完整性要求
2. 需求信息的分析整理
   - 对收集到的数据进行抽象,即对实际事物或事件的人为处理,抽取共同的本质特性,并用各种概念精确地加以描述
   - 要想把收集到的信息（如文件、图表、票据等）转换为下一阶段工作可用的形式信息，必须对需求信息作分析整理的工作
3. 编写需求分析说明

需求分析的结果

- 确定系统范围，产生系统范围图
- 分析用户活动，产生业务流程图
- 分析用户活动涉及的数据，产生数据流图
- 分析系统数据，产生数据字典

5.3 概念设计

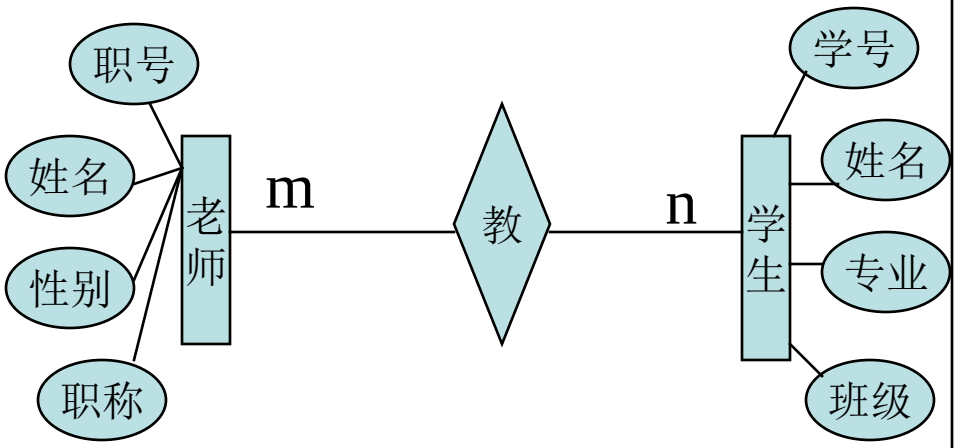
- 将需求分析阶段得到的用户需求抽象为信息结构即概念模型的过程就是数据库的概念设计
- 概念模型是对现实世界的一种抽象，即对实际的人、物、事和概念进行人为处理，抽取人们关心的共同特性，忽略非本质的细节，并把这些特性用各种概念精确地加以描述

概念设计的特征

- 概念设计阶段将得到系统的整体数据结构描述，该阶段通常会采用某种概念数据模型(如ER模型)来进行，由于概念数据模型是面向现实世界的，所以经过概念设计所得到的数据模式应当是与DBMS无关的，且独立于软硬件应用环境

采用ER模型的概念设计

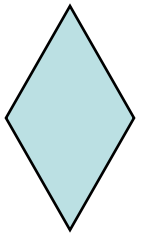
- ER模型的基本元素
  - 实体
  - 联系
  - 属性
- 例子



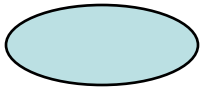
•ER模型图例

实体

联系



属性



对ER模型的理解

- ER模型是人们认识客观世界的一种方法和工具
- ER模型具有客观性和主观性两重含义
- ER模型是在客观事物或系统的基础上形成的，在某种程度上反映了客观现实，反映了用户的数据需求，因此ER模型具有客观性
- 但ER模型又不等同于客观事物的本身，它往往反映事物的某一方面，至于选取哪个方面或哪些属性，如何表达则决定于观察者本身的目的与状态，从这个意义上说，ER模型又具有主观性

对ER模型的理解

- 基于ER模型的设计过程，基本上是两大步：
  - 设计实体类型（此时不要涉及到“联系”）
  - 设计联系类型（此时考虑实体间的联系）
- 具体设计时，有时“实体”与“联系”两者之间的界线是模糊的,数据库设计者的任务就是要把现实世界中的数据以及数据间的联系抽象出来，用“实体”与“联系”来表示
- 另外，设计者应注意，ER模型应该充分反映用户需求，ER模型要得到用户的认可才能确定下来

实体和属性的设计

- 基本属性和复合属性（可否再分）
- 单值属性和多值属性（对一个实体对象是否只能取一个值）
- 多值属性的处理
  - 将原来的多值属性用几个新的单值属性来表示。
  - 将原来的多值属性用一个新的实体类型表示
- 导出属性
- 空值

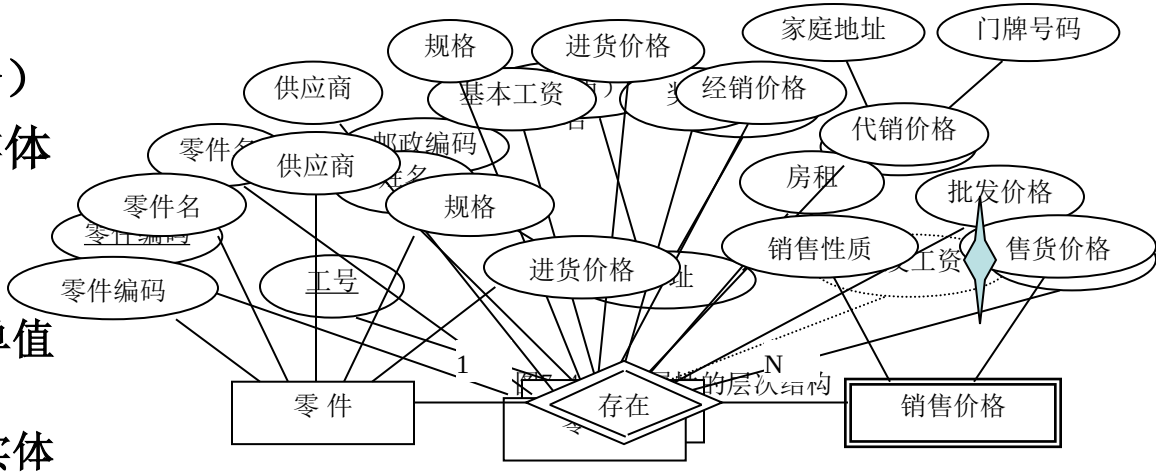


图7.6 多值属性的变换（1）

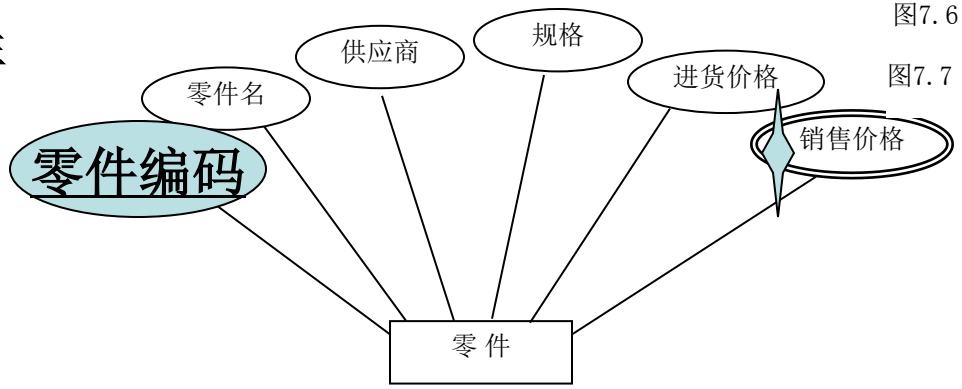


图7.5 多值属性的表示

图7.7 多值属性的变换（2）

联系的设计

- **联系集**

  联系集是n（n≥2）个实体集上的数学关系，这些实体集不必互异。如果E1，E2，…，En为n个实体集，那么联系集R是{(e1，e2，…，en）|e1∈E1 ,e2∈E2，…，en∈En}的一个子集，而（e1，e2，…，en）是一个联系。

- **联系的元数**

  一个联系涉及到的实体集个数

- **联系的连通词（基数比约束）**

  联系涉及到的实体集之间实体对应的方式

- **实体的基数（参与约束）**

  有两个实体集E1和E2，E1中每个实体与E2中有联系实体的数目的最小值min和最大值max，称为E1的基数，用（min，max）形式表示

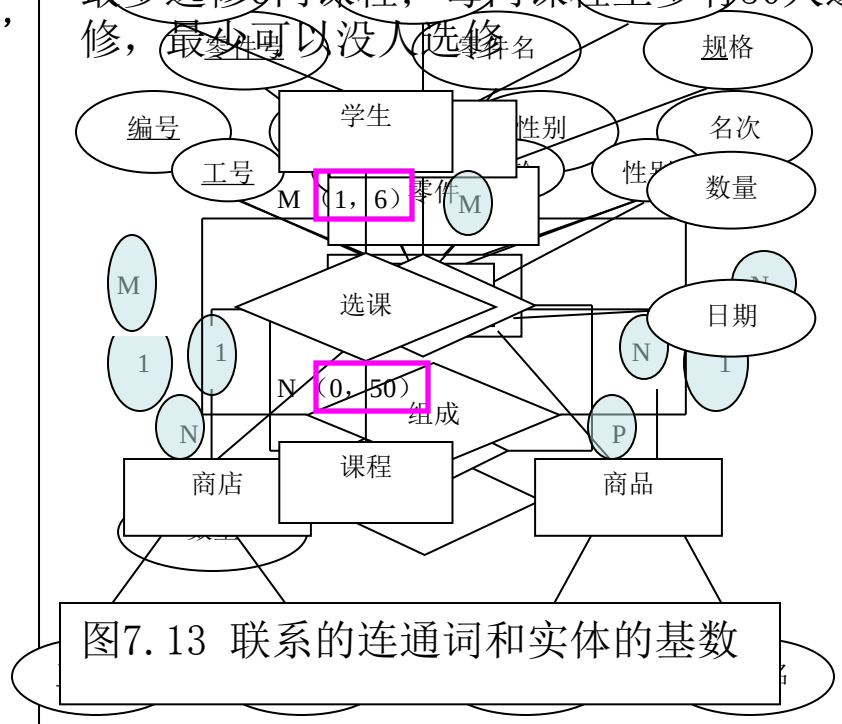


图7.13 联系的连通词和实体的基数

图7.12 三元联系中的M:N:P联系

ER模型的操作

包括实体类型、联系类型和属性的**分裂**、**合并**、**增删**等等

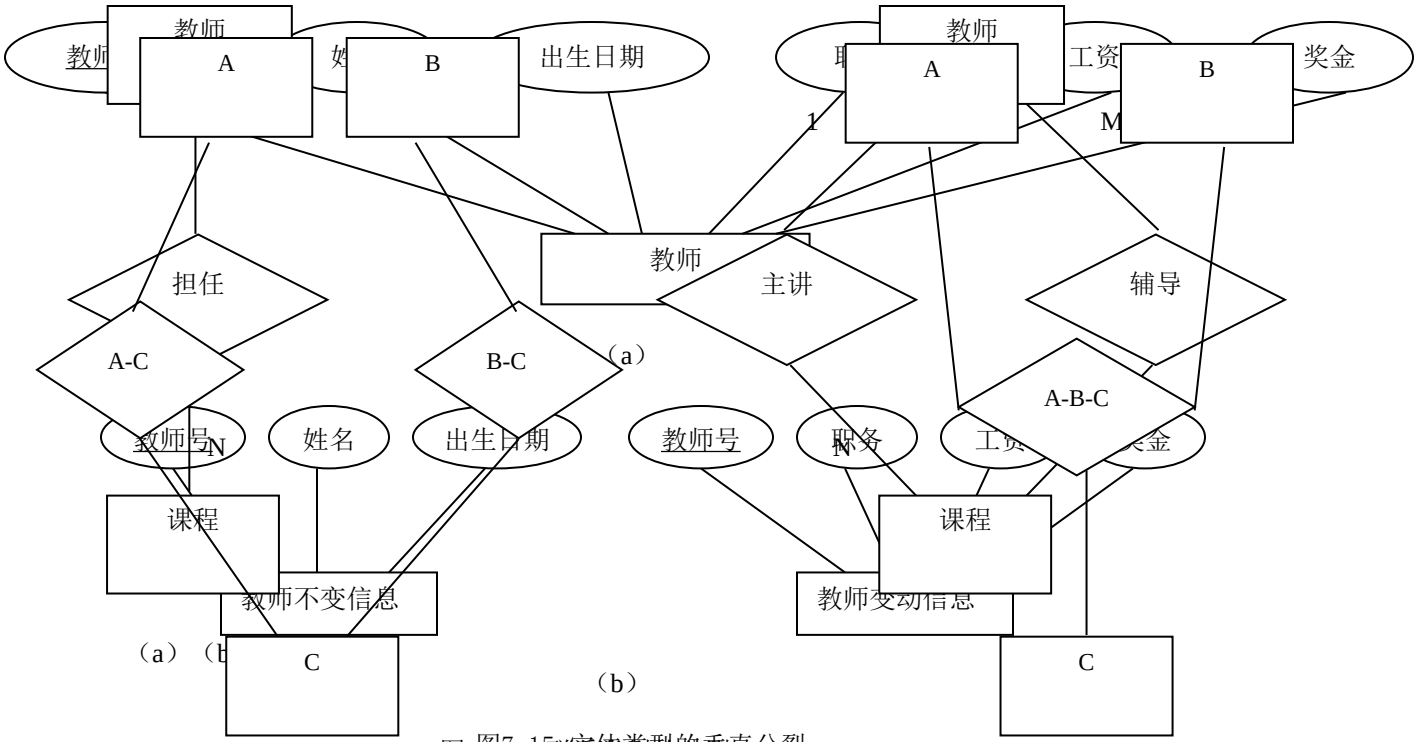


图7.15 实体类型的垂直分裂

图7.16 联系类型的分裂

(a)

(b)

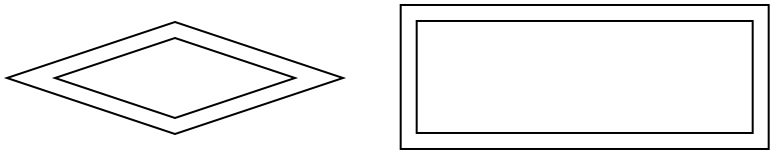
图7.17　不合法的合并

扩充的ER模型

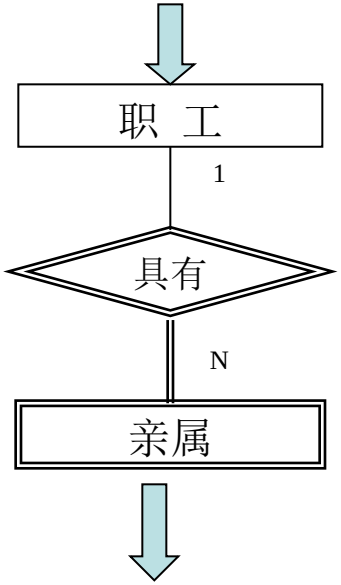
- 由于现实世界中客观存在的一些事物和概念无法通过基本ER模型进行准确描述，因此对基本E－R模型进行了必要的扩展，即扩充的ER模型（Extended ER，EER），包括:弱实体、普遍化(Generalization)/特殊化(Specialization)、范畴(Category)、聚集(Aggregation)等概念。

扩充的ER模型:弱实体

- 什么是弱实体
- 弱实体的表示方法



问题：在人事管理系统中，亲属的存在是以职工的存在为前提，即亲属对于职工具有依赖联系，因此，亲属是弱实体



ER图

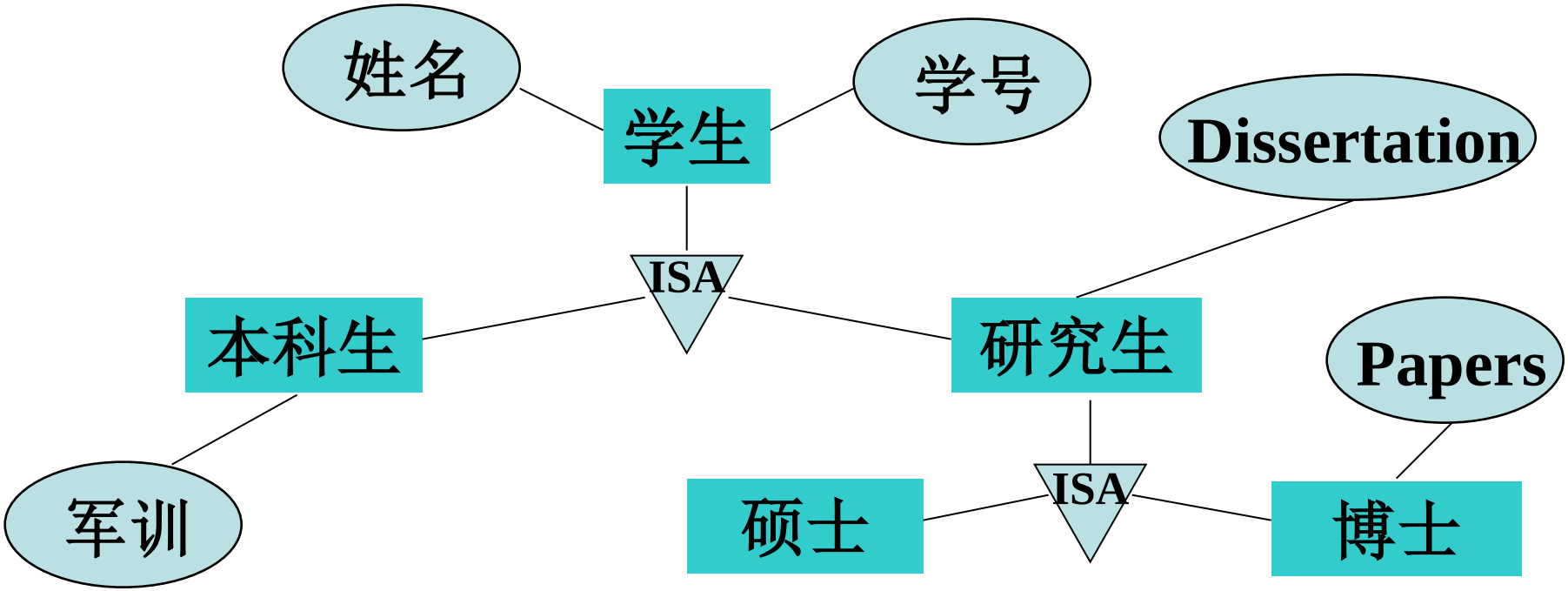
关系模式

职工（职工号，职工姓名，性别，年龄）

亲属（职工号，称呼，姓名，工作单位）

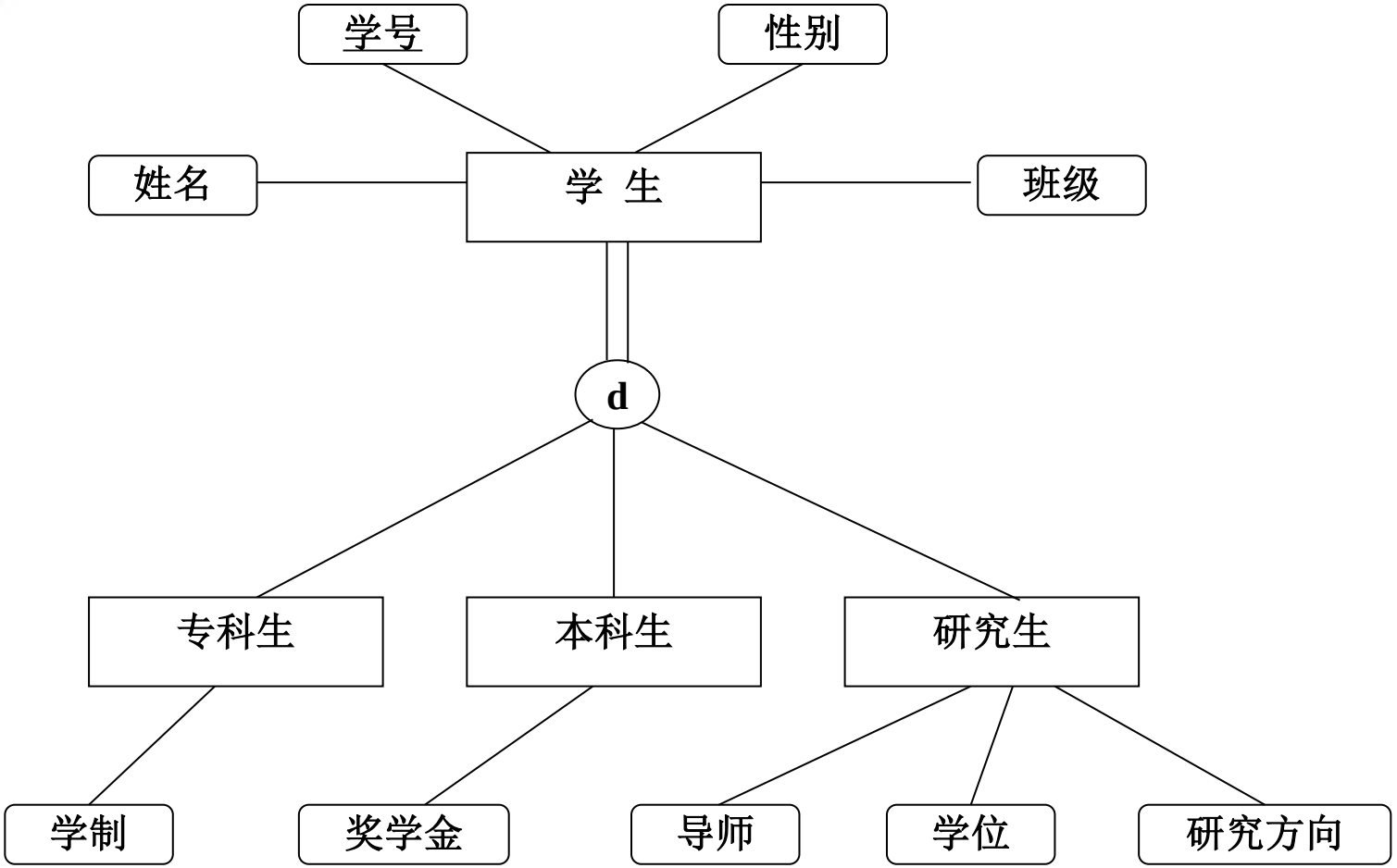
扩充的ER模型:普遍化/特殊化

- 表示“是……的一种”的关系抽象



如果概括是不相交并且是全部的，即一个高层实体最多并且只能属于一个低层实体集，则可以不为高层实体集建立关系码。

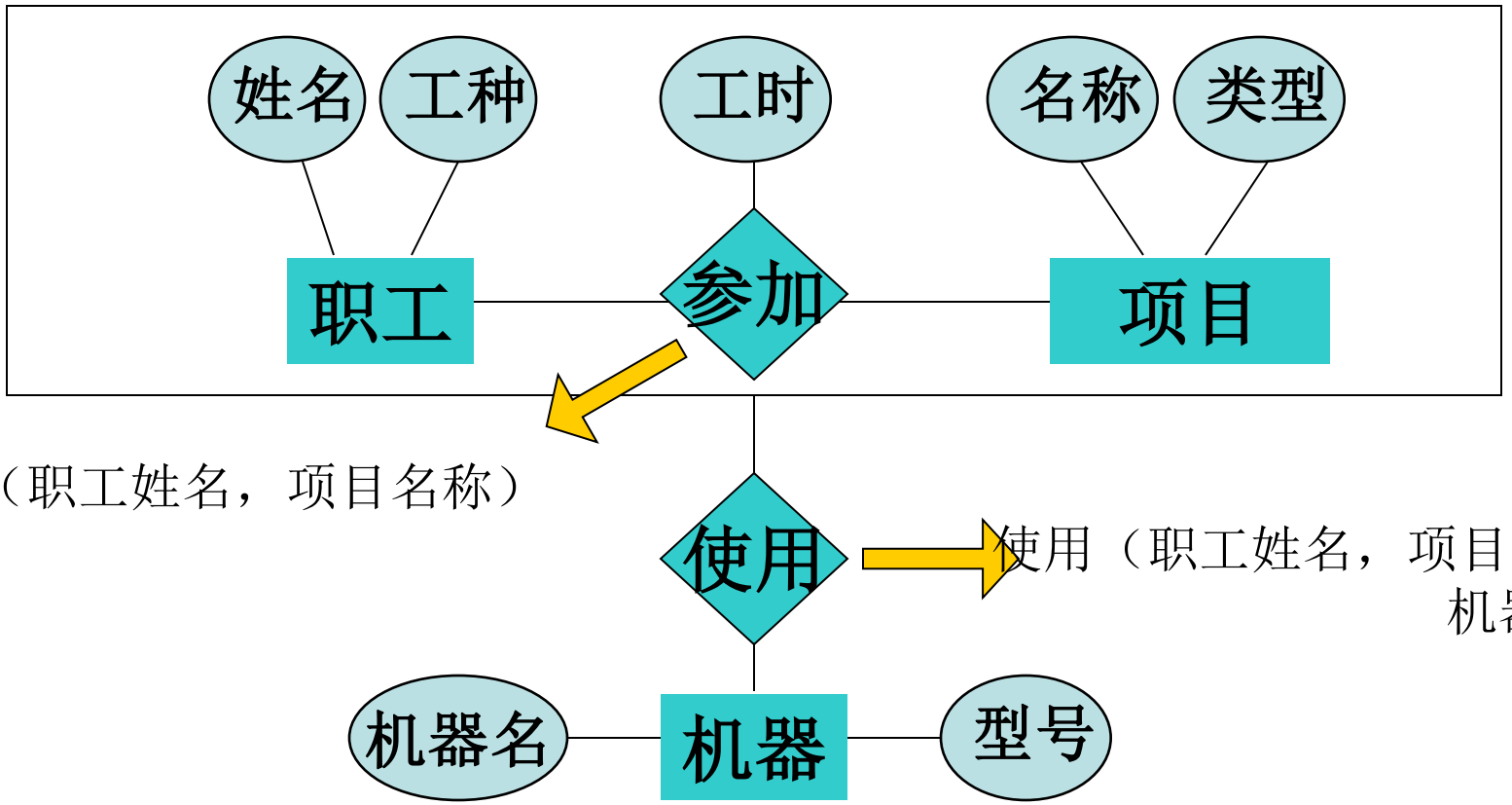
扩充的ER模型:普遍化/特殊化



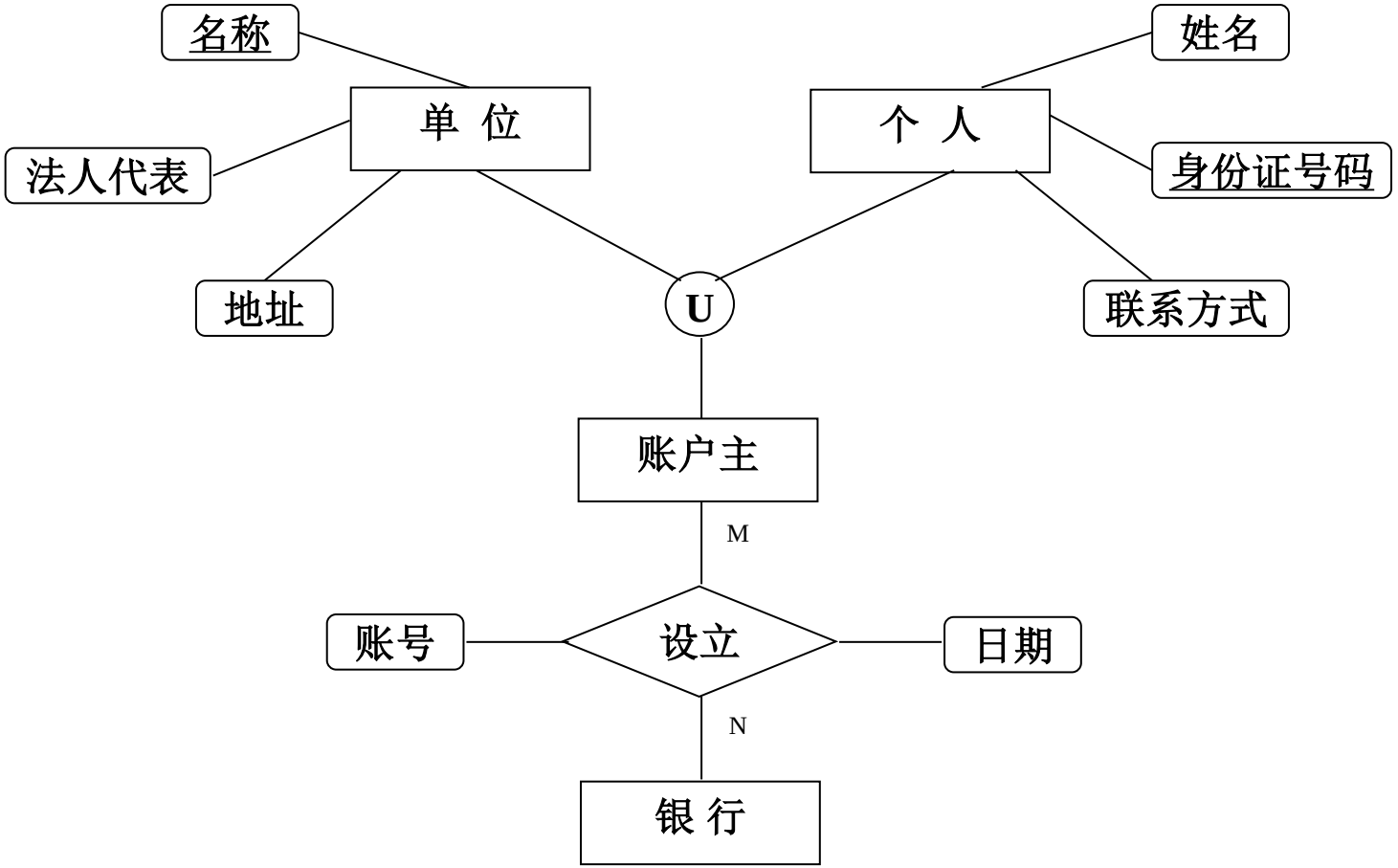
“特殊化”E－R图示例

扩充的ER模型:聚集

- 表示“是……的一部分”的关系抽象



扩充的ER模型:范畴



“范畴”E－R图示例

概念设计的方法

- 根据对用户需求处理方式的不同，可以有集中式模式设计法和视图集成法两种概念结构设计方式

集中式模式设计法

- 集中式模式设计法首先对需求分析阶段得到的应用需求进行合并，形成一个统一的整体需求说明。
- 系统的概念结构设计将以合并后的总体需求为蓝本，设计出全局的概念模式，然后再按照不同的特定应用和用户设计各自的外模式。
- 集中式设计法较适合于小规模的应用

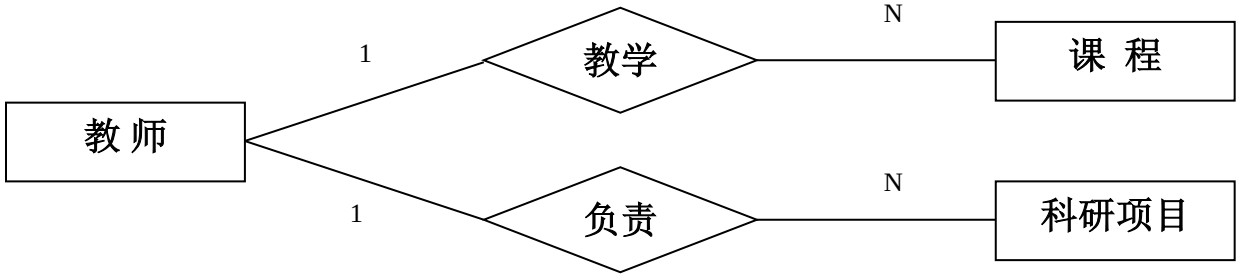
视图集成法

- 视图集成法主要包括：
  - 局部视图设计
  - 视图集成
- 两个设计步骤

概念设计的策略

(1) 自顶向下：首先建立较高抽象层次的模式（视图），然后再逐步细化，求精，直至得到更为具体的模式（视图）

(a)

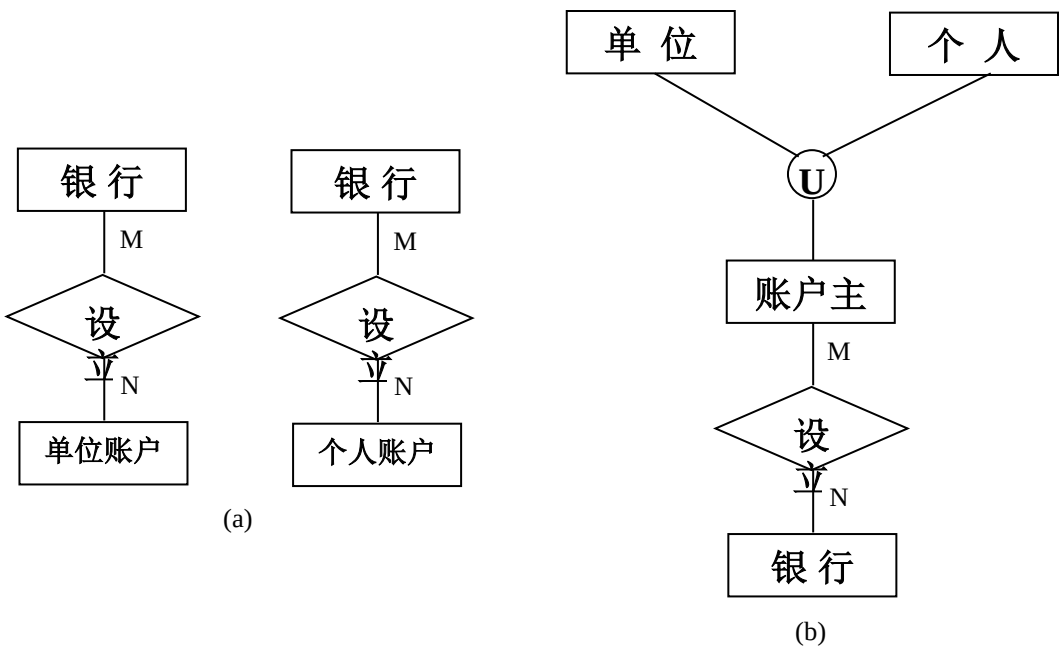


(b)

自顶向下设计过程示例

概念设计的策略

(2) 自底向上：首先从具体的基本对象开始，建立一个包含有基本抽象的模式，然后再在此基础上进行组合、修改、抽象



自底向上设计过程示例

概念设计的策略

(3)由内向外：该策略是自底向上策略的一个特例，首先从最基本的概念出发，建立一个仅包含那些具有明显特征的实体类型的初步模型，然后再逐步引入其它相关对象，因此整个建模过程是一个由内向外的扩张过程

概念设计的策略

(4)混合策略：该策略是自顶向下方式和自底向上方式的一种有效结合，首先按照自顶向下思想建立系统的全局结构框架，然后对于框架内的每一部分需求再按照自底向上方式进行详细设计

局部视图设计

确定属性的原则：

属性应该是不可再分解的；

单位；实体和属性之间的关系是1:N的；

应无直接关联关系；

属性分配的原则：

当多个实体类型用到同一属性时，一般把属性分配给那些使用频率最高的实体类型，或分配给实体值少的实体类型。

有些属性不宜归属于任一实体类型，只说明实体之间联系的特性

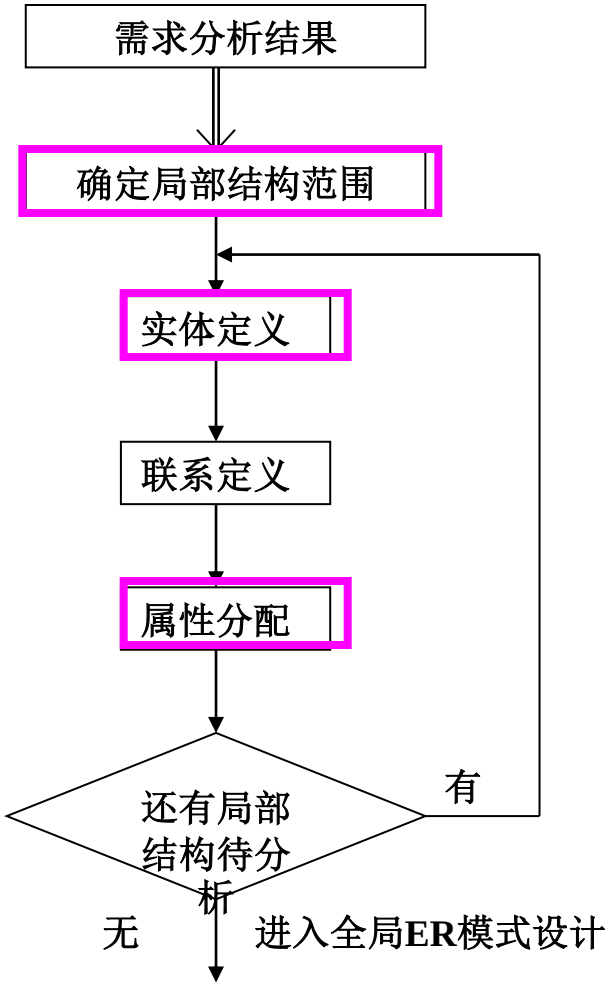


图 局部ER模式设计

E-R模型设计举例

- 某学校的管理信息系统。学校有4个部门要求实现计算机管理：
  - 人事处：教职工管理
  - 学生处：学生学籍管理
  - 教务处：教学管理
  - 后勤处：住宅、宿舍管理

E-R模型设计举例

- 假定在设计之前，已进行了调研和需求分析。主要信息如下：
  - 学校包括多个管理部门和多个系；
  - 每个部门或系有多名教职工，一名教职工只能属于一个部门或一个系；
  - 每个系有多个班级，一个班级只能属于一个系；
  - 每个班级有若干名学生，一名学生只能属于一个班级；
  - 学校开设多门课程，一门课程可被多名学生选修，一名学生也可选修多门课程；
  - 一门课程可由多名教师讲授，一名教师也可讲授多门课程；
  - 学校有多间教室，一间教室同一时间只能安排一门课程，而一门课程在同一时间可安排在多个教室(由多名教师讲授,见上)；
  - 学校有多座教工住宅楼，每个住宅楼有多套住房，每套住房只能分配给一名教工，每名教职工只能分配一套住房；
  - 学校有多座学生宿舍楼，每个宿舍楼有多个房间，每个房间可安排多名学生住宿，每个学生只能安排一个房间

E-R模型设计举例

1. 设计局部E-R模型

1）确定局部应用范围

- 通常情况下可按系统的使用部门划分：本例中划分为四个模块。
  - 人事管理——人事处
  - 学生管理——学生处
  - 教学管理——教务处
  - 住房管理——后勤处
- 通常，校长需要了解整个学校的运行情况，所以还应有一个校长查询模块，提供决策指导信息。

E-R模型设计举例

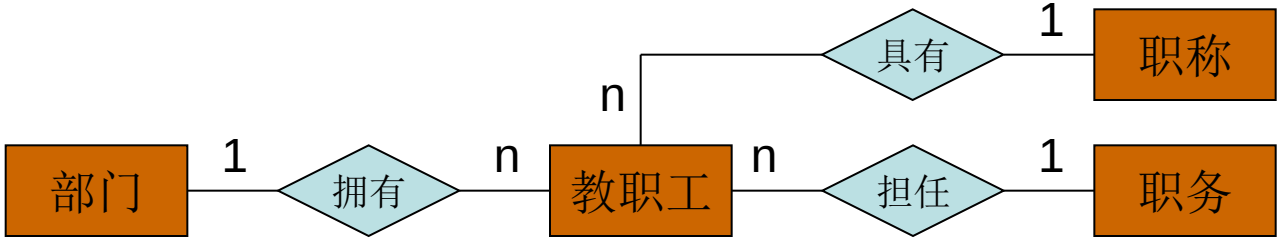
2）确定实体集（以“人事管理”为例）

- 通过调研和需求分析，人事部门需要管理教职工、部门、职称和职务，所以实体集有：教职工、部门、职称和职务。

3）确定联系集

- 决定各实体集间的联系：
  - 部门—教职工：1:N
  - 部门—职称：没有联系
  - 部门—职务：没有联系
  - 教职工—职称：N:1
  - 教职工—职务：N:1
  - 职称—职务：没有联系

- 根据以上两个步骤，画出初步E-R图如下：



E-R模型设计举例

4）确定实体集的属性. 通过调研和需求分析，各实体的属性如下：

教职工：教职工号，姓名，性别，出生日期，学历

部门(包括管理部门和教学院系)：部门号，类型，名称，办公电话

职务：代号，名称

职称：代号，名称

5）确定联系集的属性

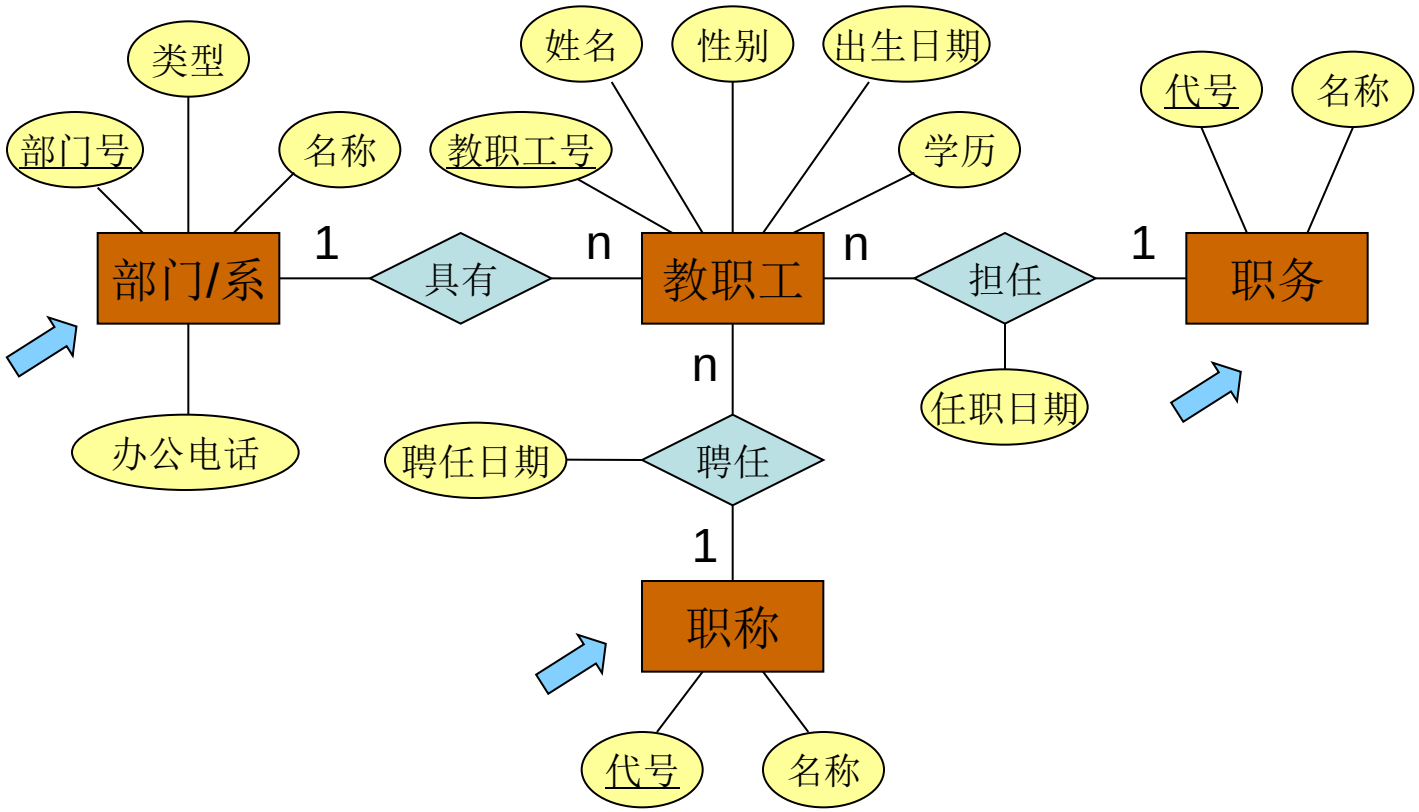
部门－教职工：无

教职工－职称：聘任日期

教职工－职务：任职日期

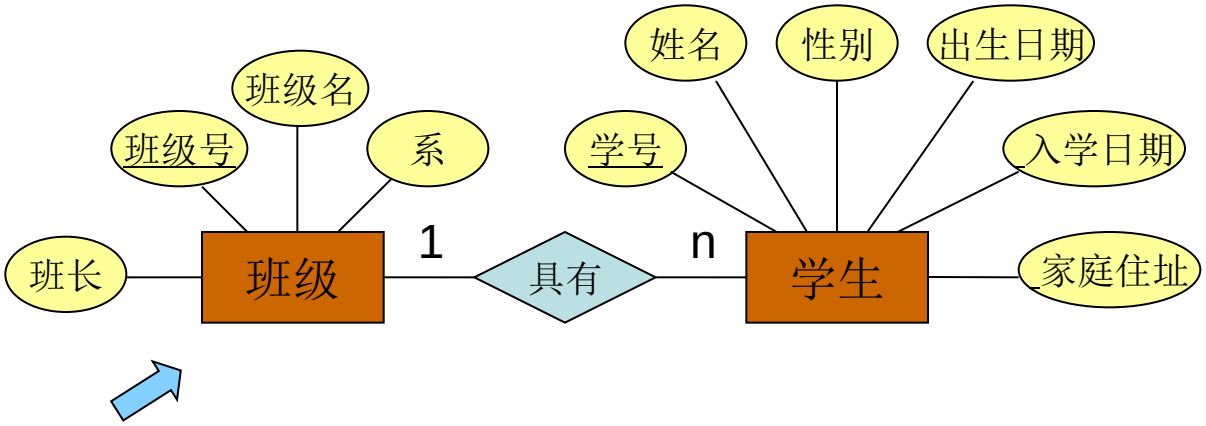
E-R模型设计举例

6）画出局部E-R模型



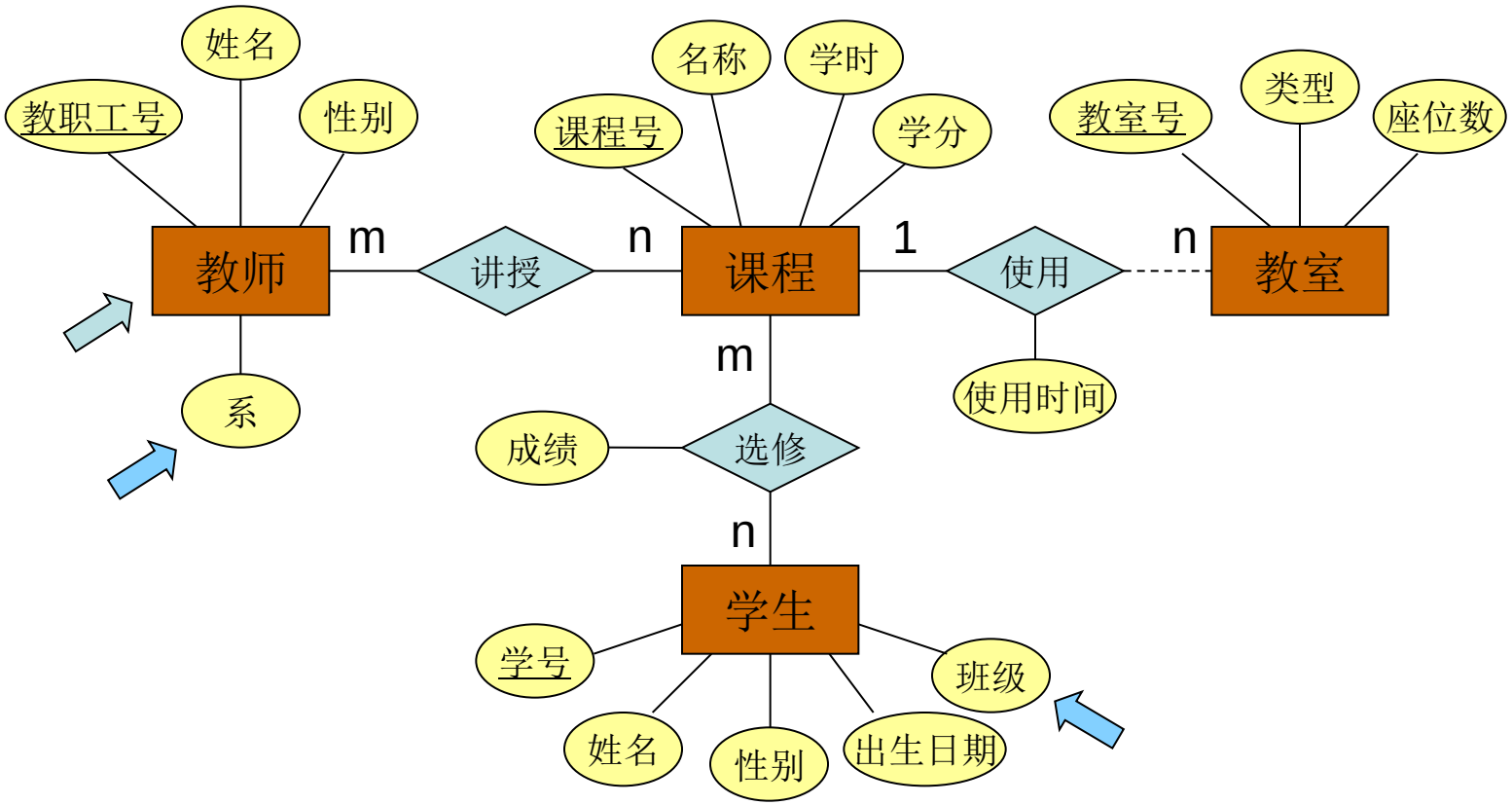
E-R模型设计举例

- 其他三个子模块的局部E-R模型的设计基本类似。
  - 学生管理的局部E-R模型



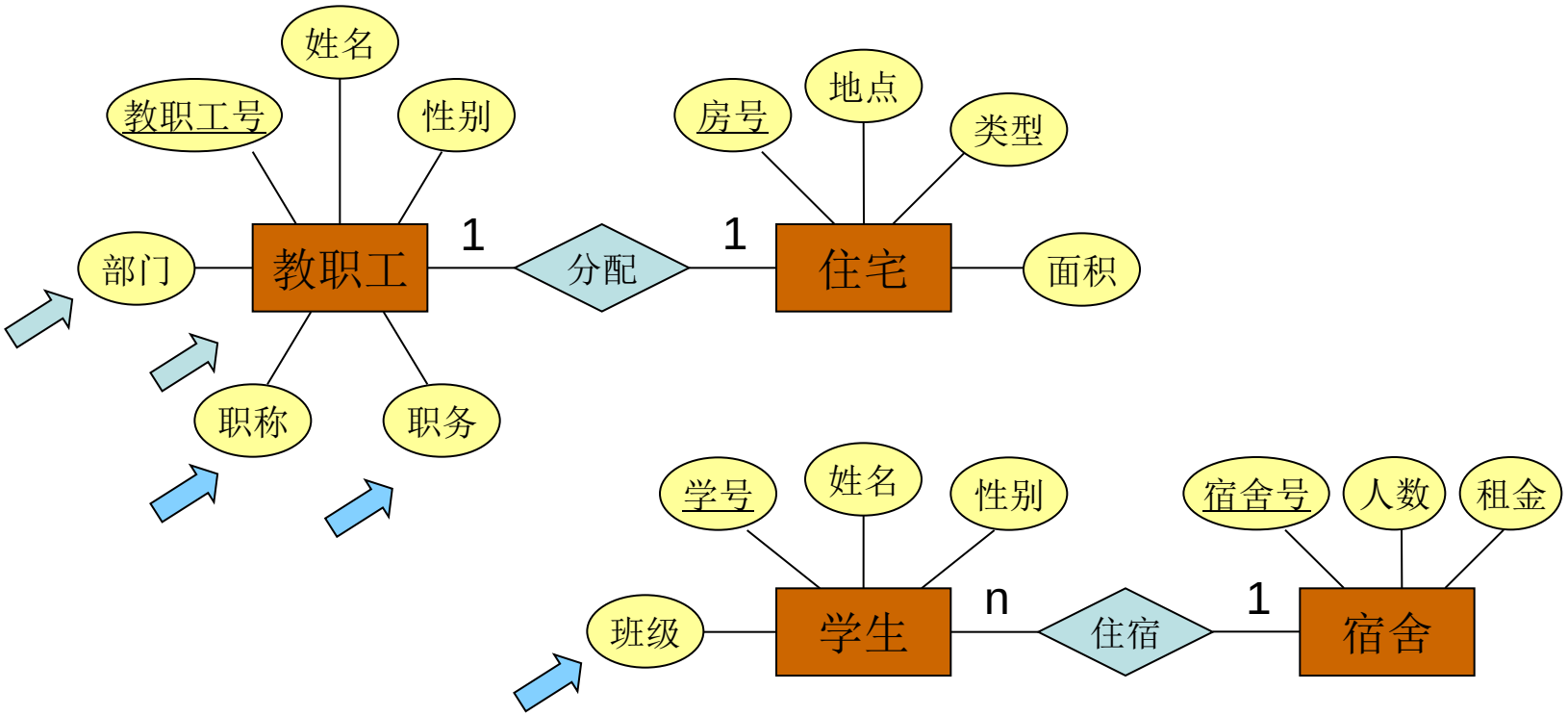
E-R模型设计举例

- 教学管理的局部E-R模型



E-R模型设计举例

- 住房管理的局部E-R模型



视图集成

- 各局部应用的E－R图设计完成后，需要将其合并形成一个全局的E－R图模式，以此作为逻辑设计阶段的设计依据，这是视图集成阶段的主要工作
- 全局E－R图不是各个局部E－R图的简单拼凑

视图集成

属性冲突：如，重量单位有的用公斤，有的用克。

结构冲突：同一对象在不同应用中的不同抽象；同一实体在不同局部ER图中属性的个数或次序不同；实体之间的联系在不同的局部ER图中呈现不同的类型

命名冲突：属性名、实体名、联系名之间存在同名异义或异名同义冲突

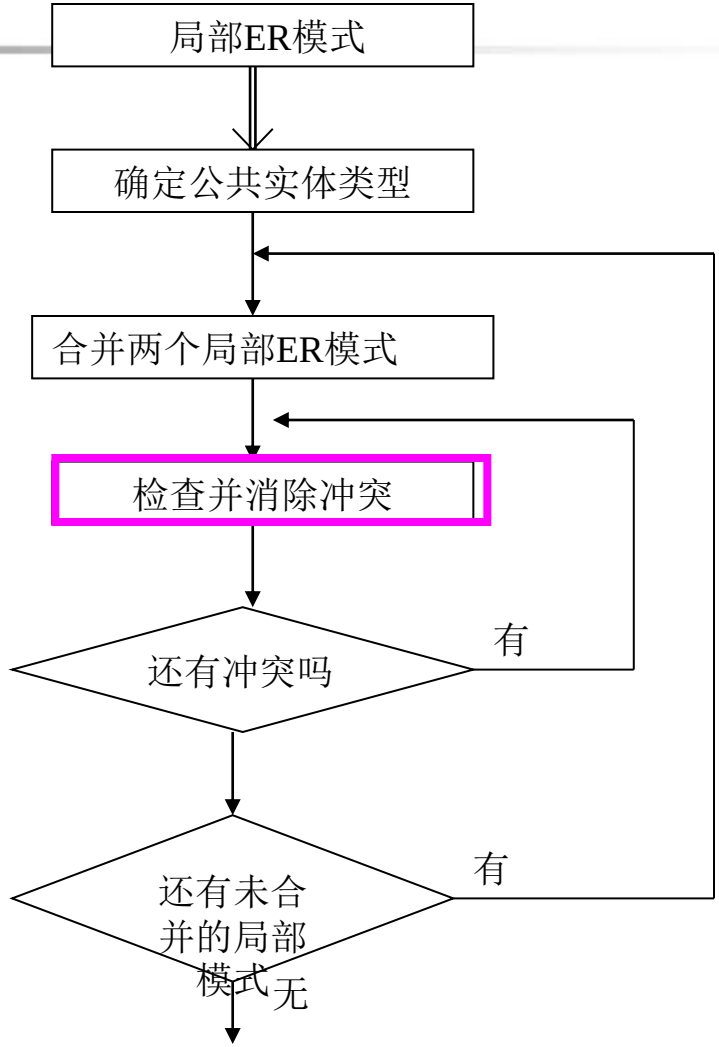


图 视图集成

视图集成

➢解决冲突，生成初步全局E－R图

(1) 命名冲突：主要指实体、属性、联系等对象在名字方面的冲突，一般表现为同名异义和同义异名两种情况。

(2) 属性冲突：包括属性域冲突和属性取值单位冲突两种情况。

(3) 类型冲突：同一个概念在不同视图中被抽象成了不同类型的对象，实际应用中的类型冲突主要是实体和属性间的冲突

(4) 约束冲突：约束冲突主要指语义约束方面的不一致

视图集成

- 消除冗余，完成全局E－R图
  - 冗余主要包括数据冗余和联系冗余两种情况，冗余的数据是指可以由其它基本数据导出的数据，冗余的联系是指可以由其它联系导出的联系
  - 数据和联系的冗余将会给数据库完整性的维护带来困难，容易引起数据的不一致，消除冗余的工作可以通过分析数据字典和数据流图中的数据说明及数据间的关系完成

视图集成:例

- 例如，对于教学管理E－R图和学生管理E－R图
- 教师实体和班主任实体属于同义异名，可将二者统一为教师实体，并将两个属性集的并作为新的属性集合，但是两实体的属性集存在冲突，班主任的教师编号属性和教师的工作证号属性其实表达了相同的概念，教师的年龄属性可以根据班主任实体的出生日期属性计算得出，而且年龄会随时间变化，需要做出相应的修改，可将年龄属性消去，得到教师实体的属性集为

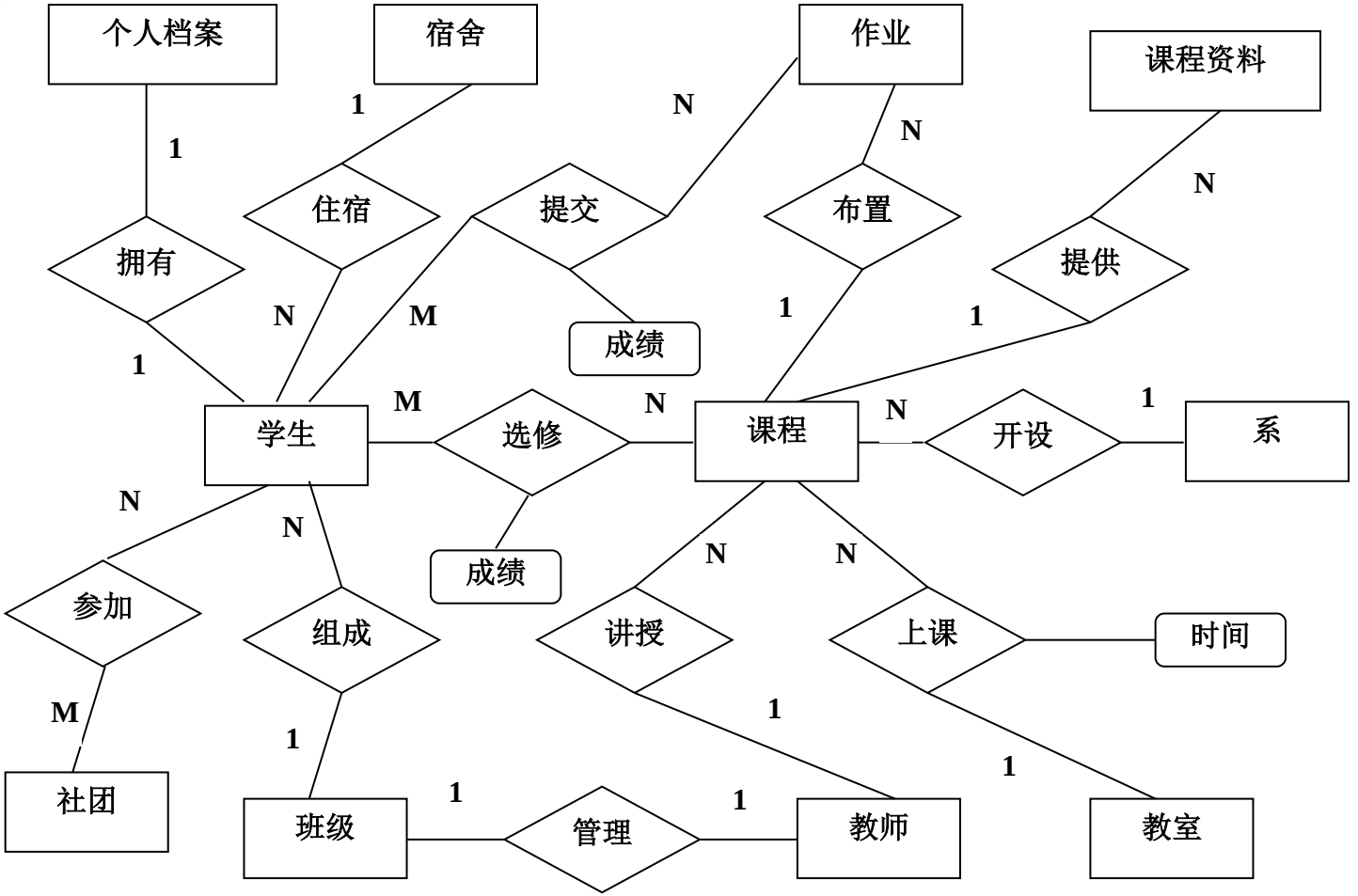
  {工作证号，姓名，性别，出生日期，职称，是否班主任}

视图集成:例

- 对于教学管理E－R图和学生管理E－R图
- 在教学管理E－R图中班级是作为学生实体的属性出现的，而在学生管理E－R图中班级作为实体出现，二者存在矛盾，将学生实体的班级属性去除，得到新的学生实体的属性集为：

  {学号，姓名，性别，出生日期，Email}

视图集成:例


个人档案
宿舍
作业
课程资料
住宿
提交
布置
提供
拥有
成绩
学生
选修
课程
开设
系
参加
组成
成绩
讲授
上课
时间
社团
班级
管理
教师
教室
1
N
M

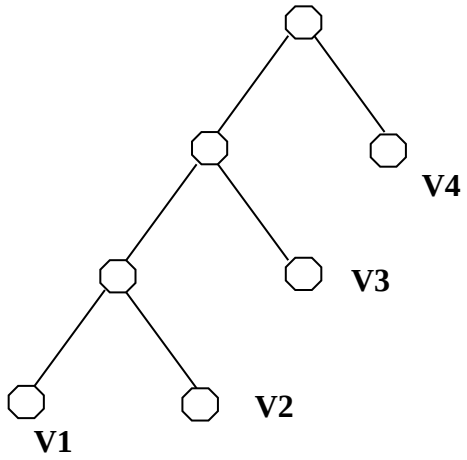
全局E－R图

视图集成策略

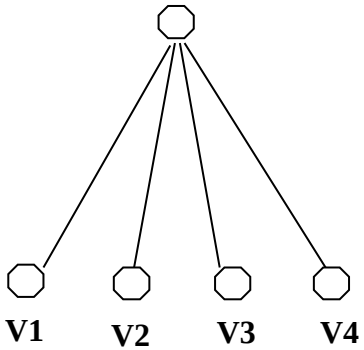
(1) 二元梯形集成

(2) n元集成

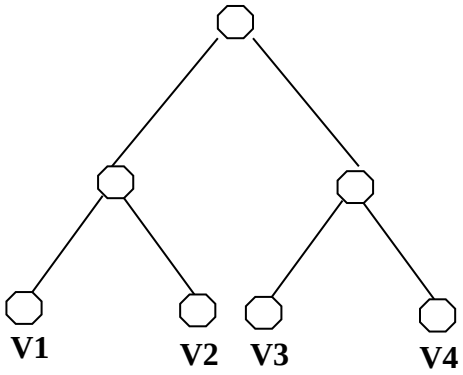
(3) 二元平衡集成



(a) 二元梯形集成



(b) n元集成



(c) 二元平衡集成

5.4 逻辑设计

- 逻辑设计的任务是在概念设计的基础上给出与DBMS相关的数据库逻辑模式
- 例如，将E-R图转换为某种具体的(如,关系)数据模型

基于ER模型的逻辑设计步骤

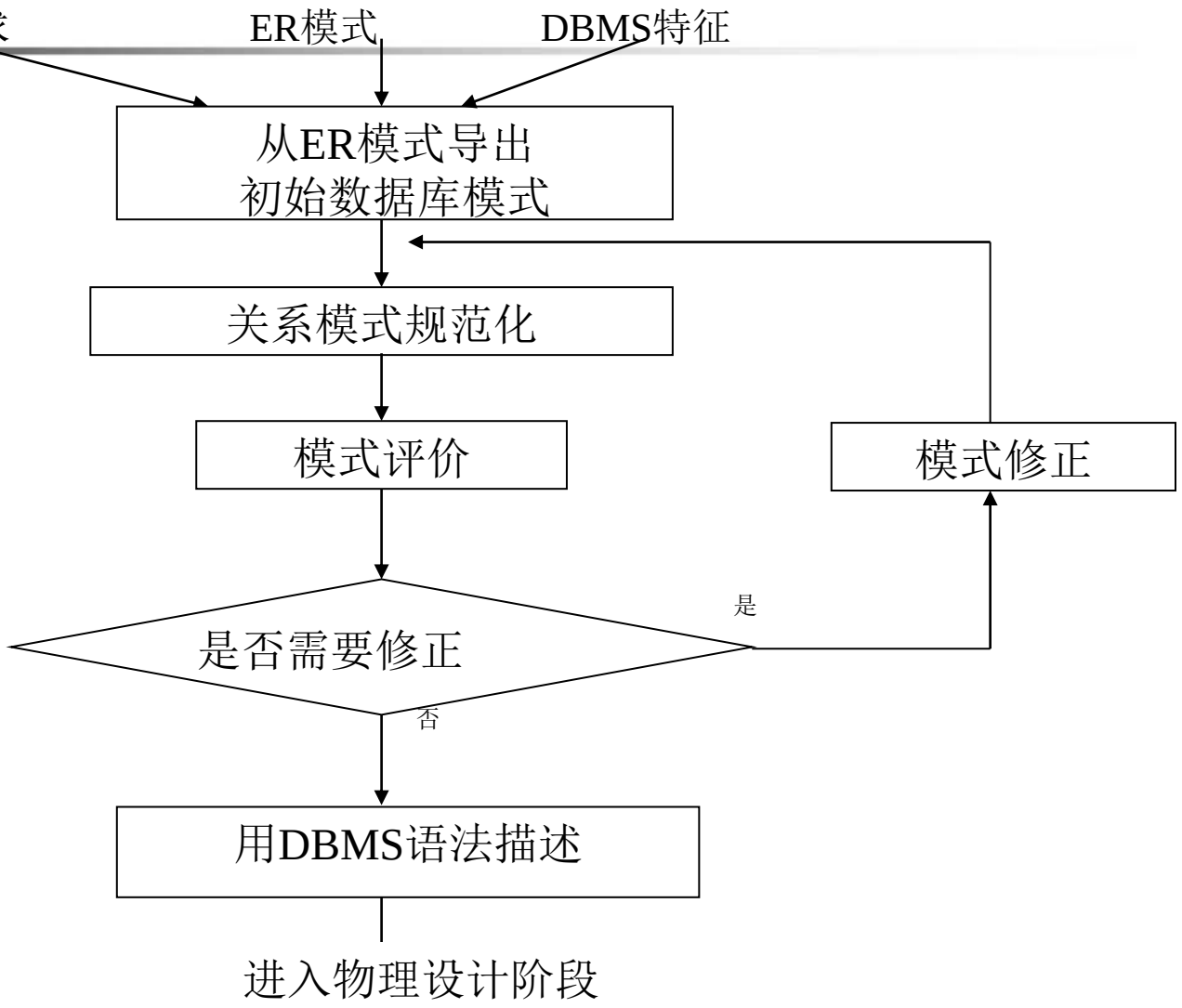


图 关系数据库的逻辑设计

E-R模型向关系模型转化

- E-R模型独立于DBMS——可用于任何一种数据库。
- 把E-R模型转换为某个具体的DBMS所能接受的关系数据模型，称为数据库的逻辑设计，或称为建立数据库逻辑模式。
- E-R模型转换为关系模型主要解决：
  - 如何用关系来表达实体和实体间的联系

ER模型向关系模型的转换

- E—R模型向关系模型的转换可以参照以下原则进行：

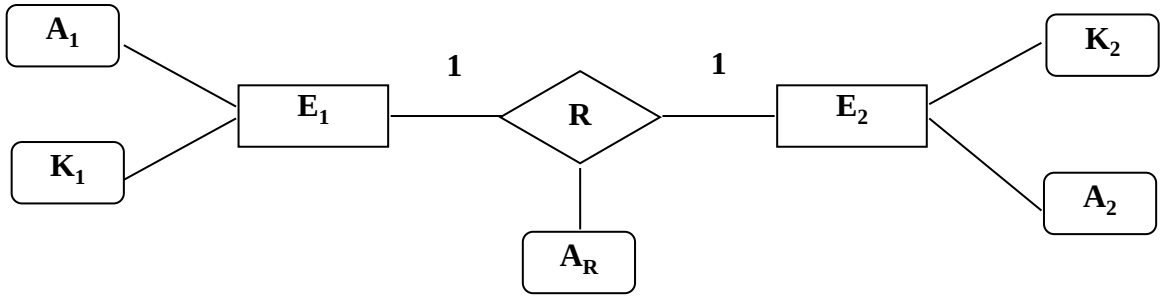
(1)实体转换为关系模式，关系模式的属性集由实体原有的属性集构成，实体的键是关系模式的键

(2)如果属性是非原子属性，可以按照纵向展开或横向展开的方式将其转化为原子属性。对于集合类型的非原子属性，需要纵向展开，而元组类型的属性则需要横向展开

(3)实体间的联系可以转换为关系模式，也可以与参与联系的实体所对应的关系模式合并，对于常见的二元联系和三元联系，有以下的具体转换方式：

ER模型向关系模型的转换

- 1：1联系



1:1联系

- 当E1和E2都是部分参与时：

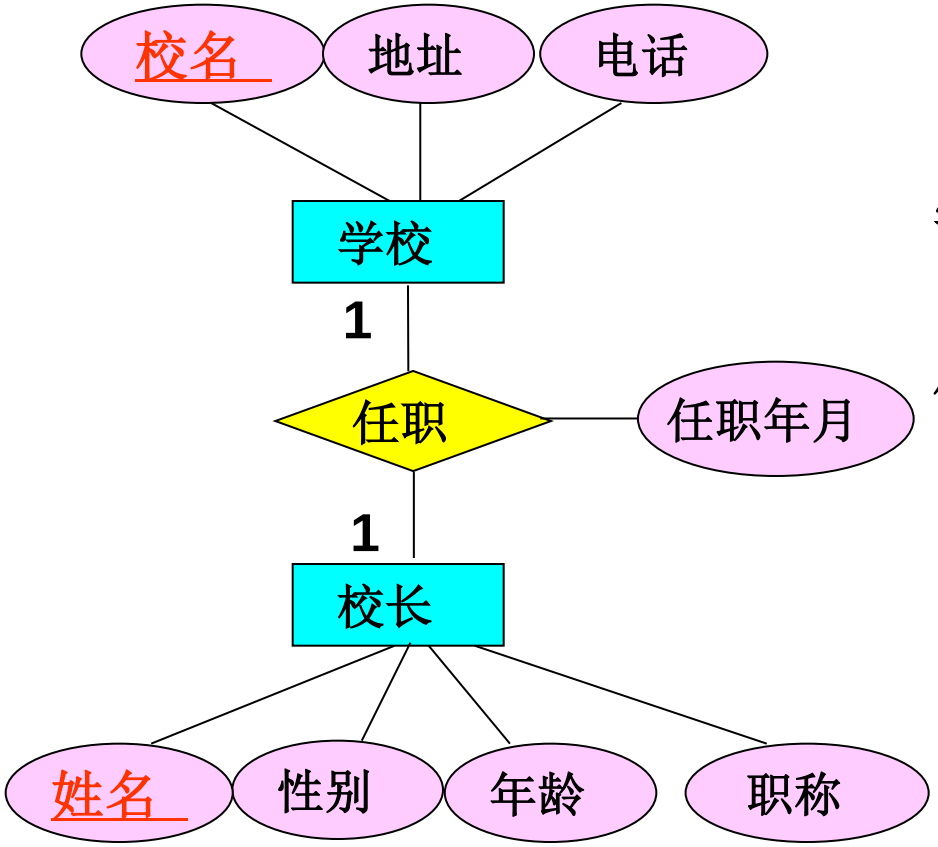
R1(K1，A1)，R2(K2，A2)，R3(K1，K2，AR)，其中K1，K2都可成为R3的键

- 当E1是全参与时：

R1(K1，A1，K2，AR)，R2(K2，A2)，其中K2是R1的外键

ER模型向关系模型的转换

举例1　转换为独立的关系模式(方法一)



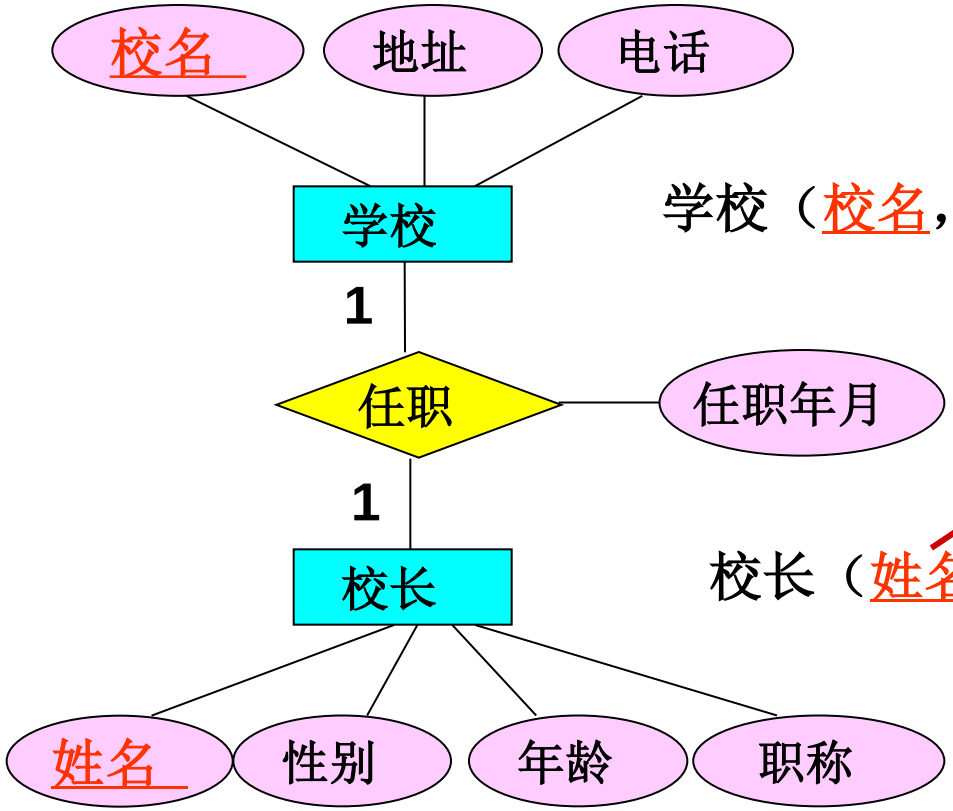
学校（校名，地址，电话）

任职（校名，校长名，任职年月）

校长（姓名，年龄，性别， 职称）

ER模型向关系模型的转换

举例1　与一端合并(方法二)

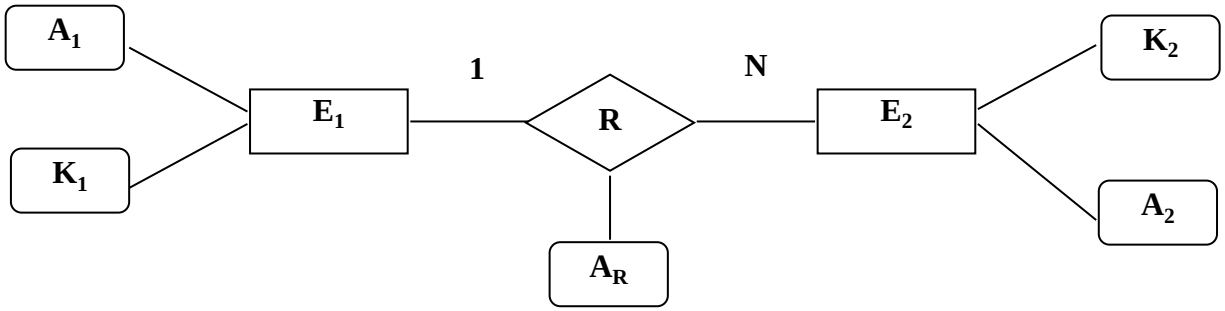


学校（校名，地址，电话,）校长名，任职年月）

校长（姓名，年龄，性别，职务）

ER模型向关系模型的转换

1：N联系



1：N联系

- 当E2是部分参与时：

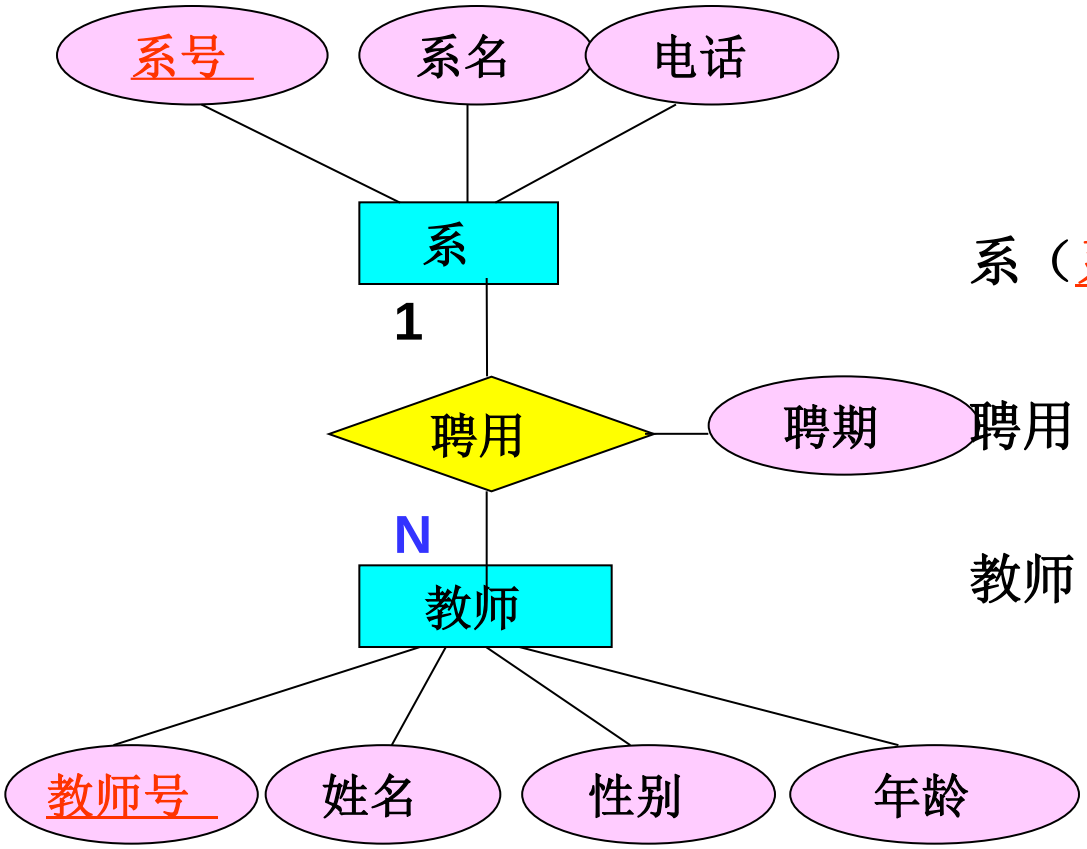
  R1(K1，A1)，R2(K2，A2)，R3(K2，K1，AR)，其中K1是R3的外键

- 当E2是全参与时：

  R1(K1，A1)，R2(K2，A2，K1，AR)，其中K1是R2的外键

ER模型向关系模型的转换

举例2 转换为独立的关系模式(方法一)



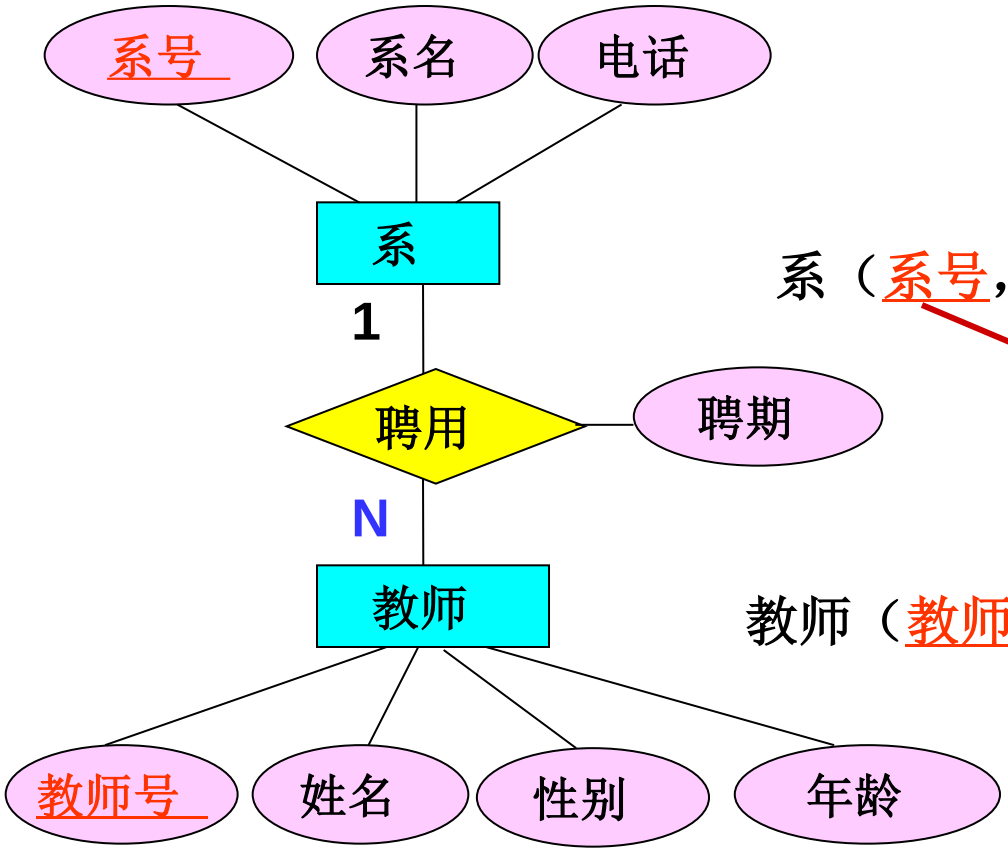
系（系号，系名，电话）

聘用（教师号，系号，聘期）

教师（教师号，姓名，年龄，性别）

ER模型向关系模型的转换

举例2　与n端合并(方法二)

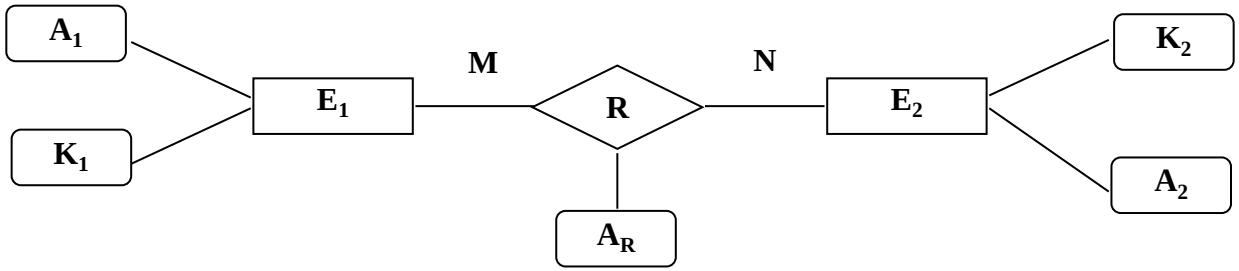


系（系号，系名，电话）

教师（教师号，姓名，年龄，性别，系号，聘期）

ER模型向关系模型的转换

- M：N联系



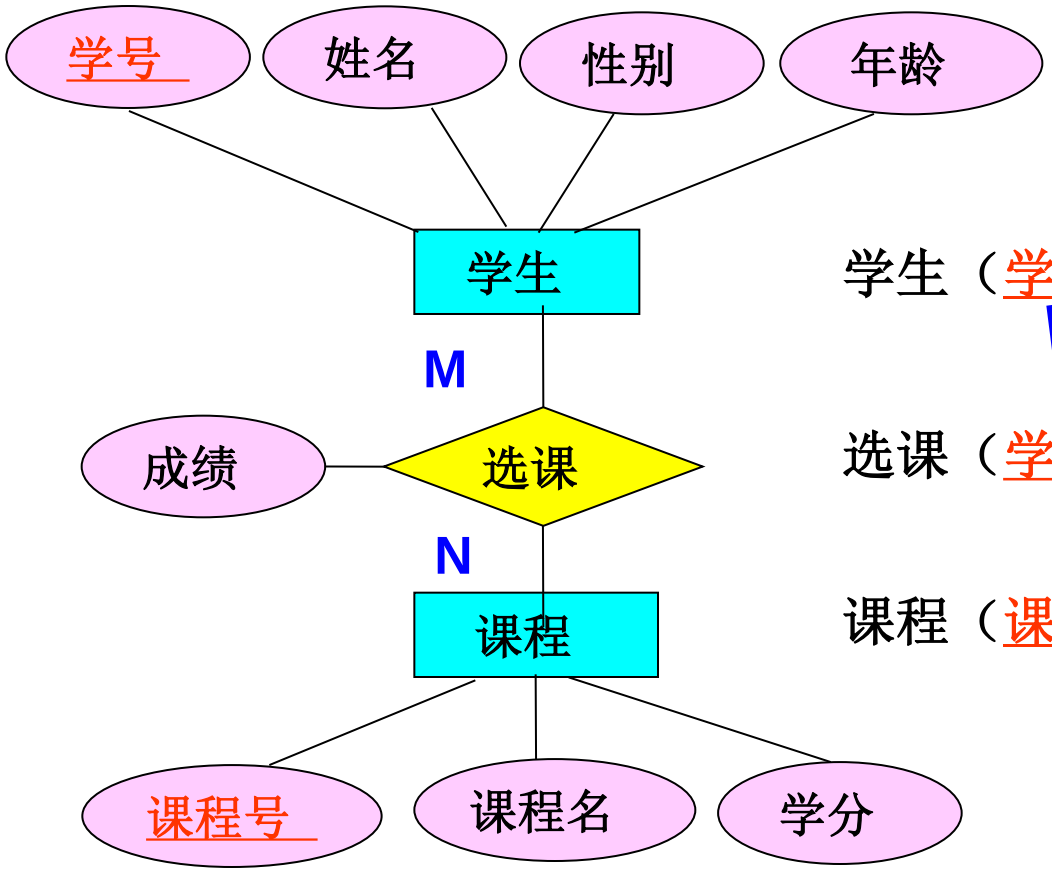
M：N联系

➢

➢R1(K1，A1)，R2(K2，A2)，R3(K1，K2，AR)，其中K1，K2联合构成R3的键，且K1，K2又同时都是外键

ER模型向关系模型的转换

举例3　转换为独立的关系模式



学生（学号，姓名，年龄，性别）

选课（学号，课程号，成绩）

课程（课程号，课程名，学分）

ER模型向关系模型的转换

- 三元联系
  - 可以按照M：N二元联系的情况处理，因此包括实体转换的关系模式在内一共会得到四个关系模式

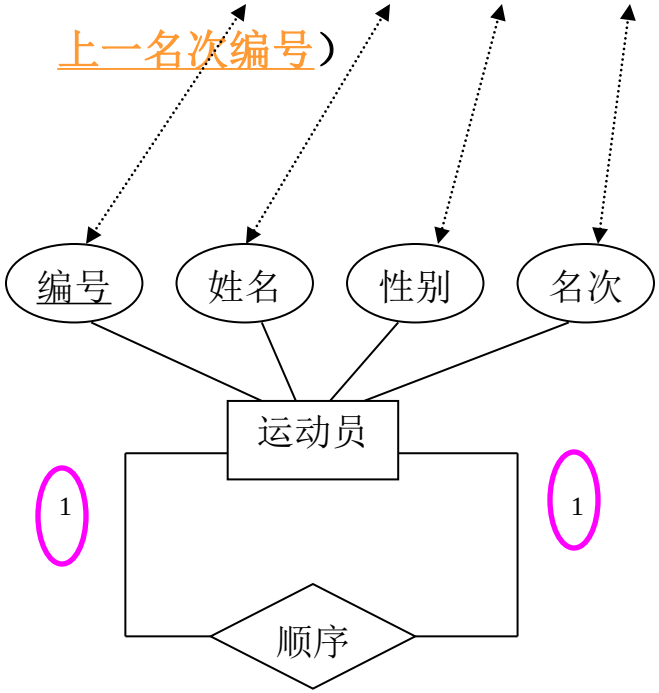
ER模型向关系模型的转换

(4)如果存在键完全相同的若干个关系模式，则可以考虑将其合并为一个新的关系模式

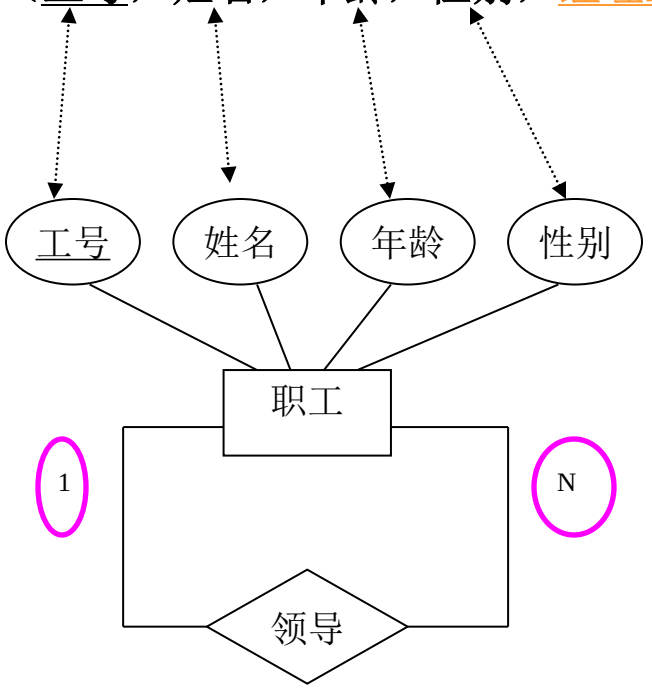
(5)对于同一类型实体的自联系，可以参照(3)中给出的不同实体间1：1、1：N、M：N联系的情况进行转换，但应给予不同的命名加以区分

ER模型到关系模型的转换：实例

运动员（编号，姓名，性别，名次，上一名次编号）



职工（工号，姓名，年龄，性别，经理工号）



ER模型到关系模型的转换：实例

仓库（仓库号，仓库名，地址）

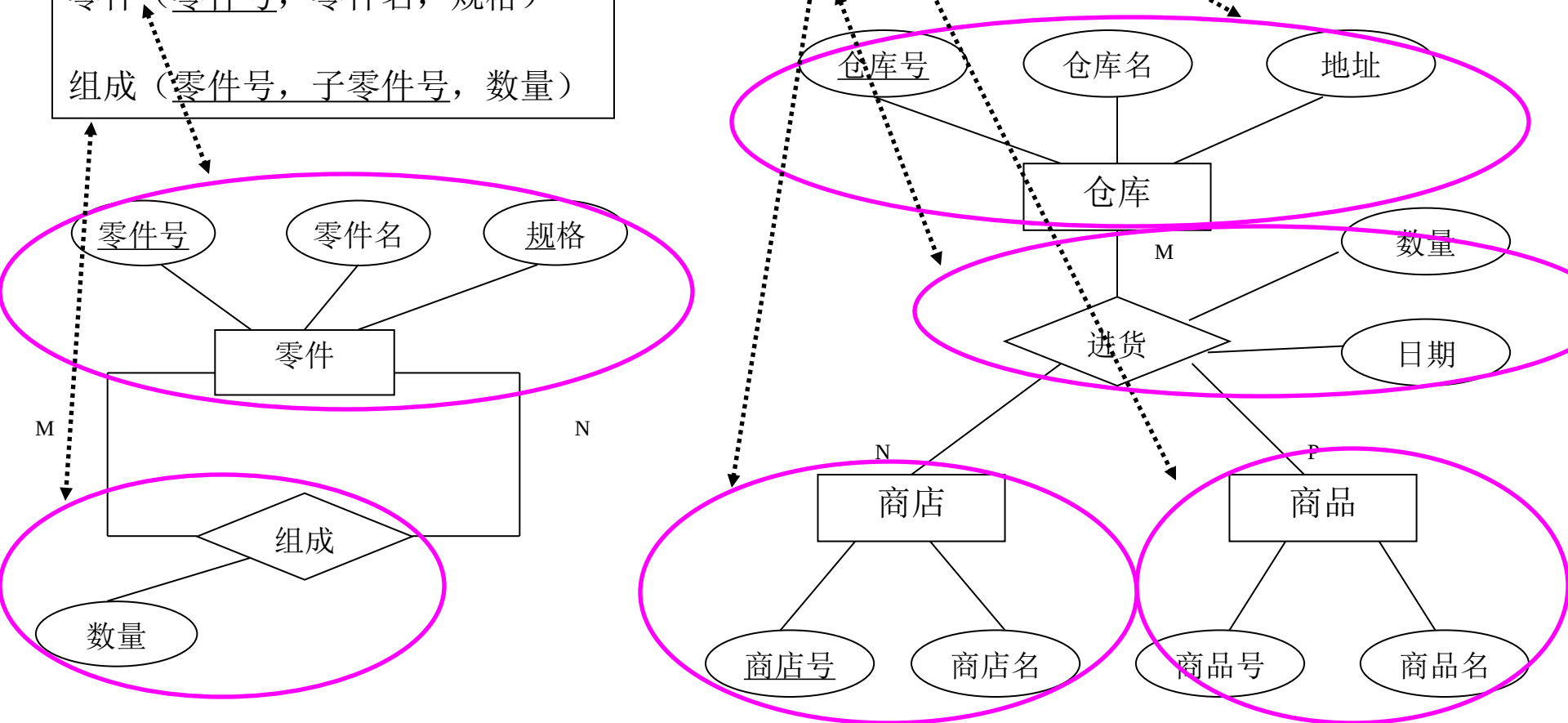
商店（商店号，商店名）

商品（商品号，商品名）

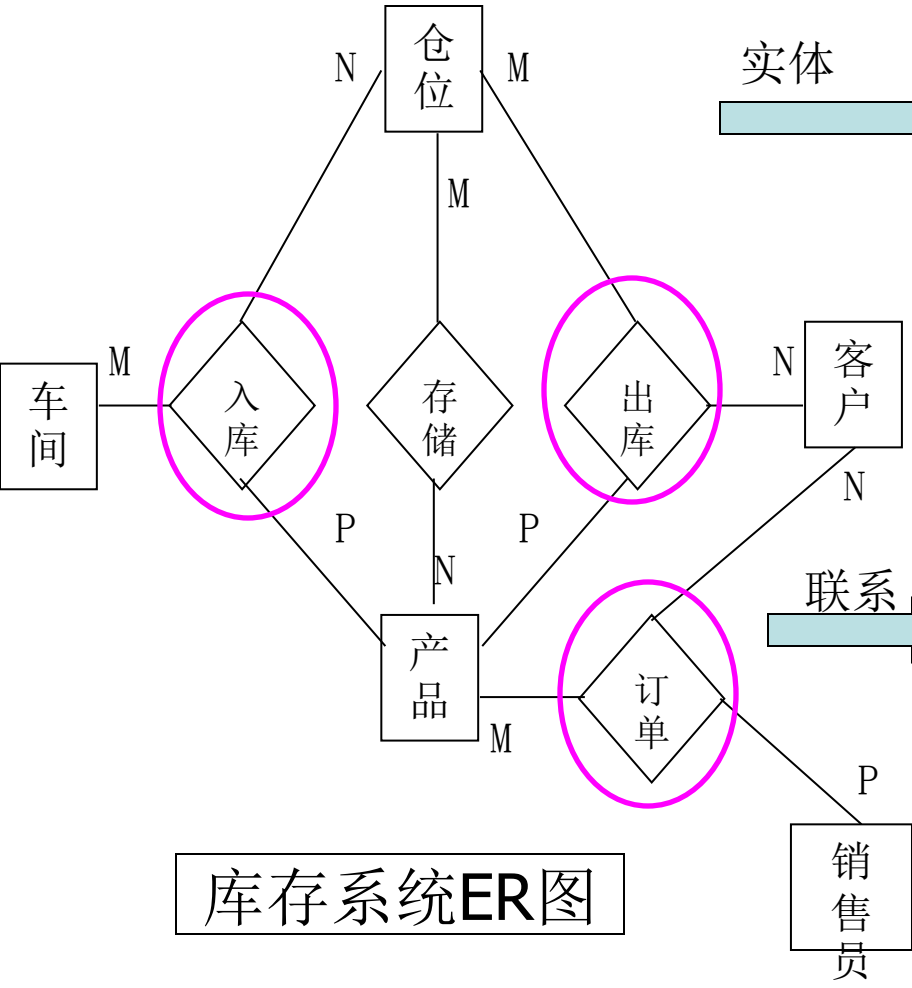
进货（商店号，商品名，仓库号，日期，数量）

零件（零件号，零件名，规格）

组成（零件号，子零件号，数量）



例：库存销售信息管理系统的ER模型及转换



库存系统ER图

实体

车间(车间号, 车间名, 主任名)
产品(产品号，产品名，单价)
仓位(仓位号，地址，主任名)
客户(客户号，客户名，联系人，电话，地址，税号，账号)
销售员(销售员号, 姓名, 性别, 学历, 业绩)

联系

入库（入库单号, 入库量, 入库日期, 经手人, 车间号, 仓位号, 产品名）
出库（出库单号，出库量，出库日期，经手人，客户号，产品号，仓位号）
订单（订单号，数量，折扣，总价，订单日期，产品号，客户号，销售员号）
存储(仓位号, 产品号, 核对日期, 核对员, 存储量)

ER模型到关系模型的转换：实例

学生（学号，姓名，性别，出生日期，Email）

个人档案（档案编号，姓名，政治面貌，个人经历，奖惩情况，社会关系）

宿舍（宿舍号，位置，床位数，描述）

班级（名称，学生人数，班长）

社团（社团编号，名称，负责人，简介）

课程（课程编号，课程名称，学分，学期，教材）

课程资料（资料编号，文件名称，文件位置，资料类型，资料说明）

作业（作业编号，文件名称，文件位置，作业说明，上交时间）

教室（教室编号，位置，座位数）

教师（工作证号，姓名，性别，出生日期，职称，是否班主任）

系（系编号，系名）

ER模型到关系模型的转换：实例

不将1：1和1：N联系转化为关系模式，而是合并到参与联系实体所对应的关系模式中，得到如下新的关系模式：

学生（学号，姓名，班级，性别，出生日期，Email，档案编号，宿舍编号）

个人档案（档案编号，姓名，政治面貌，个人经历，奖惩情况，社会关系）

宿舍（宿舍号，位置，床位数，描述）

班级（名称，学生人数，班长，班主任）

社团（社团编号，名称，负责人，简介）

课程（课程编号，课程名称，任课教师，系编号，学分，学期，教材，教室，时间）

课程资料（资料编号，课程编号，文件名称，文件位置，资料类型，资料说明）

作业（作业编号，课程编号，文件名称，文件位置，作业说明，上交时间）

教室（教室编号，位置，座位数）

教师（工作证号，姓名，性别，出生日期，职称，是否班主任）

系（系编号，系名）

优化与调整

- 对于关系模式进行的优化与调整主要集中在数据结构与性能两个方面：
  - 结构方面的调整主要是指通过关系规范化理论对关系模式进行优化，以减少数据冗余和更新异常问题的出现
  - 性能的调整主要是指通过减少查询时连接运算次数和改变关系大小等方式提高数据处理速度

优化与调整

- 关系模式的规范化过程主要包括以下几个步骤：

(1)确定数据依赖

(2)参照最小覆盖算法对每个关系模式内及关系模式间的数据依赖进行极小化处理，消除冗余依赖

(3)逐一分析每个关系模式属于第几范式，一般只要达到3NF或BCNF即可

优化与调整

- 由于性能方面的原因而对关系模式进行的调整主要包括：

  (1)减少连接运算

  (2)从水平方向和垂直方向对关系进行分解，避免出现过大的关系

用户子模式设计

- 得到优化的系统全局模式后，还需针对不同局部应用设计出用户子模式即外模式。设计时应注意：

  (1)能适应不同用户的使用需求

  (2)提供一定的逻辑数据独立性

  (3)数据安全性

5.5 物理设计

- 任何数据库最终都要存储在物理设备上。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（存储结构与存取方法）的过程，就是数据库的物理设计
- 数据库物理结构依赖于给定的DBMS、OS及硬件系统，因此设计人员必须(1)充分了解所用DBMS的内部特征，特别是存储结构和存取方法；(2)充分了解应用环境，特别是应用的处理频率和响应时间要求；(3)充分了解外存设备的特性

物理设计的步骤

- 确定数据的存储结构
- 确定数据的存放位置
- 确定数据的存取方法
- 确定系统配置
- 对物理结构进行评价

确定数据的存储结构

- 确定数据库存储结构时要综合考虑存取时间、存储空间利用率和维护代价三方面的因素

- 但是，这三个方面常常是相互矛盾的，例如消除一切冗余数据虽然能够节约存储空间，但往往会导致检索代价的增加，因此必须进行权衡，选择一个折中方案

确定数据的存放位置

- 目前许多计算机都有多个磁盘，因此进行物理设计时可以考虑将表和索引分别放在不同的磁盘上，在查询时，由于两个磁盘驱动器分别在工作，因而可以保证物理读写速度比较快
- 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。此外还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

确定数据的存放位置

- 数据可以按照使用频繁程度或者稳定程度进行划分并确定其存储位置，对于需要频繁访问的大关系，可以考虑将其按照一定的原则进行分片，然后放置在不同的磁盘上，此外也可以将索引和具体的数据文件放置在不同的磁盘上，同样可以提高I/O速度

确定数据的存放位置

- 将数据划分到不同的磁盘驱动器或磁盘阵列上的设计参考原则：

  (1)减少磁盘访问冲突，提高I/O并行性

  (2)分散需要频繁访问的数据，均衡I/O负载

  (3)提高关键数据存取速度

确定数据的存取方法

- 存取方式设计的任务就是确定采用何种存取路径以获得最快的数据存取速度，DBMS一般都会提供索引、聚簇、散列等常见的存取路径供选择，其中索引是使用最多的存取方法
- 存取路径的确定并不是惟一的，同一个数据对象上可以有满足不同应用需求的多种存取方式同时存在

确定数据的存取方法

- 在关系数据库中，选择存取路径主要是指确定如何建立索引

- 例如，应把哪些域作为次键建立次索引，建立单键索引还是组合索引，建立多少个为合适，是否建立聚集索引等

常用的存取方法

- 索引法
  - 为加快按某个属性(组)进行存取的效率，根据该属性(组)建立索引，如B+树
  - 索引建立在单个关系上
- 聚簇(Cluster)法
  - 为提高按聚簇键进行查询的效率，将聚簇键上具有相同值的元组存放在连续物理块
  - 一个数据库可以建立多个聚簇，但一个关系只能有一个聚簇
  - 聚簇可以建立在单表上，也可建立在进行连接操作的多个表上
  - SQL中与聚簇有关的操作如ORDER BY, GROUP BY, UNION, DISTINCT等

常用的存取方法

- HASH法
  - 设计合理的HASH函数，根据关键字值计算得到存储地址
  - 对可能出现的地址冲突现象设计合理的解决方案
  - 当某属性(组)主要出现在等连接条件或相等比较条件中，而且关系的大小可以预知，或关系大小动态变化而DBMS提供了HASH存取方法时，可考虑选用

确定数据的存取方法

- 设计数据的存取路径就是确定在哪些不同的属性或属性集合上应建立何种类型的索引，对于以下几种情况可以考虑建立索引：

  (1)关系的主键和外键上应该建立索引

  (2)对于经常作为连接条件或查询条件的属性或属性集

  (3)有些查询无需访问具体数据通过查找索引就可以直接给出结果，例如带有MIN、MAX、AVG、SUM、COUNT等聚集函数的查询以及带有存在谓词EXISTS的查询，可以在这些属性上建立索引。

  (4)对于以读操作为主的关系，可以多建立一些索引

确定数据的存取方法

- 对于以下的几种情况，不适宜建立索引：

(1)对于不经常使用的属性不需要建立索引，否则反而会增加维护开销。

(2)属性值很少或属性值分布不均匀的属性，不宜建立索引

(3)对于频繁变动的或较小的关系，不宜建立索引

确定数据的存取方法

- 聚簇可以有效提高在物理上连续存储的相关数据的存取速度，因此在实际当中常常也会采用聚簇的方法作为存取路径之一

确定数据的存取方法

- 对于以下的几种情况，可以考虑采用聚簇的存取方式：

(1)对于经常进行等值查询的属性，可以为其建立聚簇

(2)对聚簇属性的存取应该是关系上的主要应用，而且很少访问其它属性集，特别的，当SQL语句中含有与聚簇属性有关的ORDER BY、GROUP BY、UNION、DISTINCT等子句时，聚簇是非常有利的

(3)聚簇属性上每个取值所对应的元组个数应比较多，否则聚簇的性能优势并不明显

(4)聚簇属性上的取值应相对稳定，这样可以降低聚簇的维护开销

确定系统配置

- DBMS产品一般都提供了一些存储分配参数，供设计人员和DBA对数据库进行物理优化
- 初始情况下，系统都为这些变量赋予了合理的缺省值
- 但是这些值不一定适合每一种应用环境，在进行物理设计时，需要重新对这些变量赋值以改善系统的性能

确定系统配置

- 通常情况下，这些配置变量包括：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，锁的数目等等，这些参数值影响存取时间和存储空间的分配，在物理设计时就要根据应用环境确定这些参数值，以使系统性能最优

确定系统配置

- 在物理设计时对系统配置变量的调整只是初步的，在系统运行时还要根据系统实际运行情况做进一步的调整，以期切实改进系统性能

物理结构评价

- 数据库物理设计过程中需要对时间效率、空间效率、维护代价和各种用户要求进行权衡，其结果可以产生多种方案，数据库设计人员必须对这些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

物理结构评价

- 评价物理数据库的方法很大程度上依赖于所选用的DBMS，主要是从定量估算各种方案的存储空间、存取时间和维护代价入手，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

5.8 数据库实施

- 数据库物理设计阶段完成后即可以进入数据库实施阶段
- 数据库实施阶段主要包括以下工作：
  - ✓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

定义数据库结构

- 确定了数据库的逻辑结构与物理结构后，就可以用所选用的DBMS提供的数据定义语言（DDL）来严格描述数据库结构

数据装载

- 数据库结构建立好后，就可以向数据库中装载数据
- 组织数据入库是数据库实施阶段最繁重的工作

数据装载

- 对于数据量不是很大的系统，可以用人工方法完成数据的入库，其步骤为：

1. 数据筛选
   - 需要装入数据库中的数据通常都分散在各个部门的数据文件或原始凭证中，所以首先必须把需要入库的数据筛选出来
2. 数据格式转换
   - 筛选出来的需要入库的数据，其格式往往不符合数据库要求，还需要进行转换。这种转换有时可能很复杂
3. 数据输入
   - 将转换好的数据输入计算机中
4. 数据校验
   - 检查输入的数据是否有误

数据装载

- 对于中大型系统，由于数据量极大，用人工方式组织数据入库将会耗费大量人力物力，而且很难保证数据的正确性。因此应该设计一个数据输入子系统由计算机辅助数据的入库工作

编制与调试应用程序

- 数据库应用程序的设计应该与数据设计并行进行
- 在数据库实施阶段，当数据库结构建立好后，就可以开始编制与调试数据库的应用程序，也就是说，编制与调试应用程序是与组织数据入库同步进行的
- 调试应用程序时由于数据入库尚未完成，可先使用模拟数据

数据库试运行

- 应用程序调试完成，并且已有一小部分数据入库后，就可以开始数据库的试运行
- 数据库试运行也称为联合调试，其主要工作包括：
  - ✓功能测试：
    - 即实际运行应用程序，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
    - 即测量系统的性能指标，分析是否符合设计目标

5.9 数据库运行与维护

- 如果数据库试运行结果符合设计目标，数据库就可以真正投入运行了
- 数据库投入运行标着开发任务的基本完成和维护工作的开始，但并不意味着设计过程的终结，由于应用环境在不断变化，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

数据库运行与维护和主要内容

- 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，主要包括：
  1. 数据库的转储和恢复
  2. 数据库性能的监督、分析和改进
  3. 数据库的重组
  4. 数据库的重构

数据库的转储和恢复

- 定期对数据库和日志文件进行备份，以保证一旦发生故障，能利用数据库备份及日志文件备份，尽快将数据库恢复到某种一致性状态，并尽可能减少对数据库的破坏

数据库性能的监督分析和改进

- 目前,许多DBMS产品都提供了监测系统性能参数的工具，DBA可以利用这些工具方便地得到系统运行过程中一系列性能参数的值
- DBA应该仔细分析这些数据，通过调整某些参数来进一步改进数据库性能

数据库的重组

- 数据库运行一段时间后，由于记录的不断增、删、改，会使数据库的物理存储变坏，从而降低数据库存储空间的利用率和数据的存取效率，使数据库的性能下降。这时DBA就要对数据库进行重组，或部分重组（只对频繁增、删的表进行重组）
- 数据库的重组不会改变原设计的数据逻辑结构和物理结构，只是按原设计要求重新安排存储位置，回收垃圾，减少指针链，提高系统性能。DBMS一般都提供了供重组数据库使用的实用程序，帮助DBA重新组织数据库

数据库的重构

- 当数据库应用环境发生变化，会导致实体及实体间的联系也发生相应的变化，使原有的数据库设计不能很好地满足新的需求，从而不得不适当调整数据库的模式和内模式，这就是数据库的重构。DBMS都提供了修改数据库结构的功能。
- 重构数据库的程度是有限的，若应用变化太大，已无法通过重构数据库来满足新的需求，或重构数据库的代价太大，则表明现有数据库应用系统的生命周期已经结束，应该重新设计新的数据库系统，开始新数据库应用系统的生命周期