



数据库系统原理

李瑞轩

华中科技大学计算机学院

第七章 数据库设计

- 7.1 数据库设计概述(Summarize)
- 7.2 需求分析(Requirement Analysis)
- 7.3 概念设计(Conception Design)
- 7.4 逻辑设计(Logical Design)
- 7.5 物理设计(Physical Design)
- 7.6 实施与维护(Running & Maintain)



7.1 数据库设计概述

- 7.1.1 数据库设计的特点
- 7.1.2 数据库设计方法
- 7.1.3 数据库设计的基本步骤

数据库设计与MIS

- 管理信息系统MIS

- ☐ 数据库系统的规划
- ☐ 数据库设计
- ☐ 数据库实现
- ☐ 数据库的运行、管理和维护
- ☐ 数据库的扩充和重构

- 数据库设计是指对于一个给定的应用环境，构造最优的数据库模式，建立数据库及其应用系统，有效存储数据，满足用户信息要求和处理要求

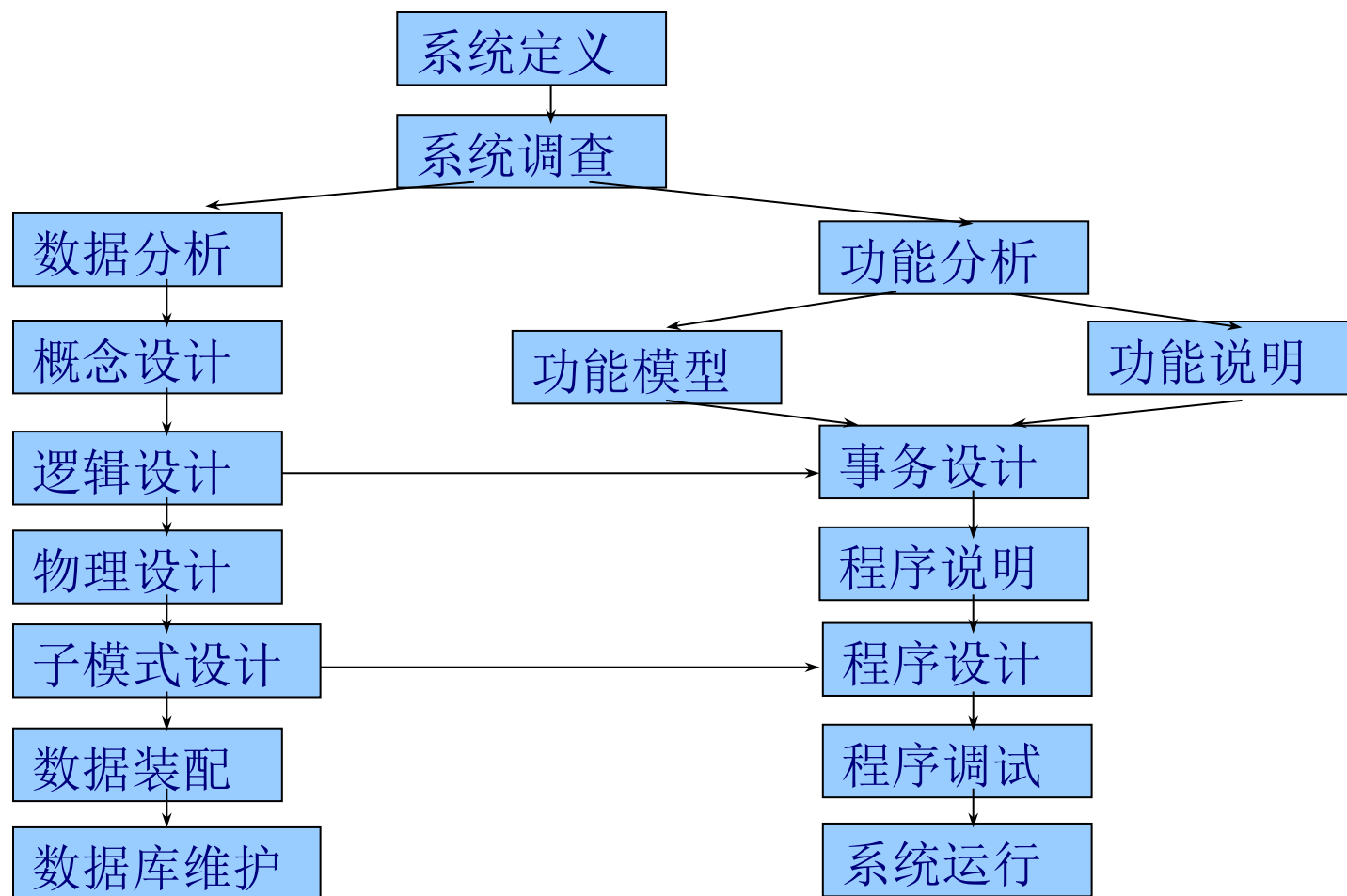
- 一个好的数据库产品不等于就有一个好的应用系统

7.1.1 数据库设计的特点

■ 设计特点

- “三分技术，七分管理，十二分基础数据”
- 静态数据设计与动态行为设计相结合
- 反复性、试探性、分步进行
- 数据库设计具有多解性

静态数据设计与动态行为设计 分离->结合



静态数据设计

动态行为设计

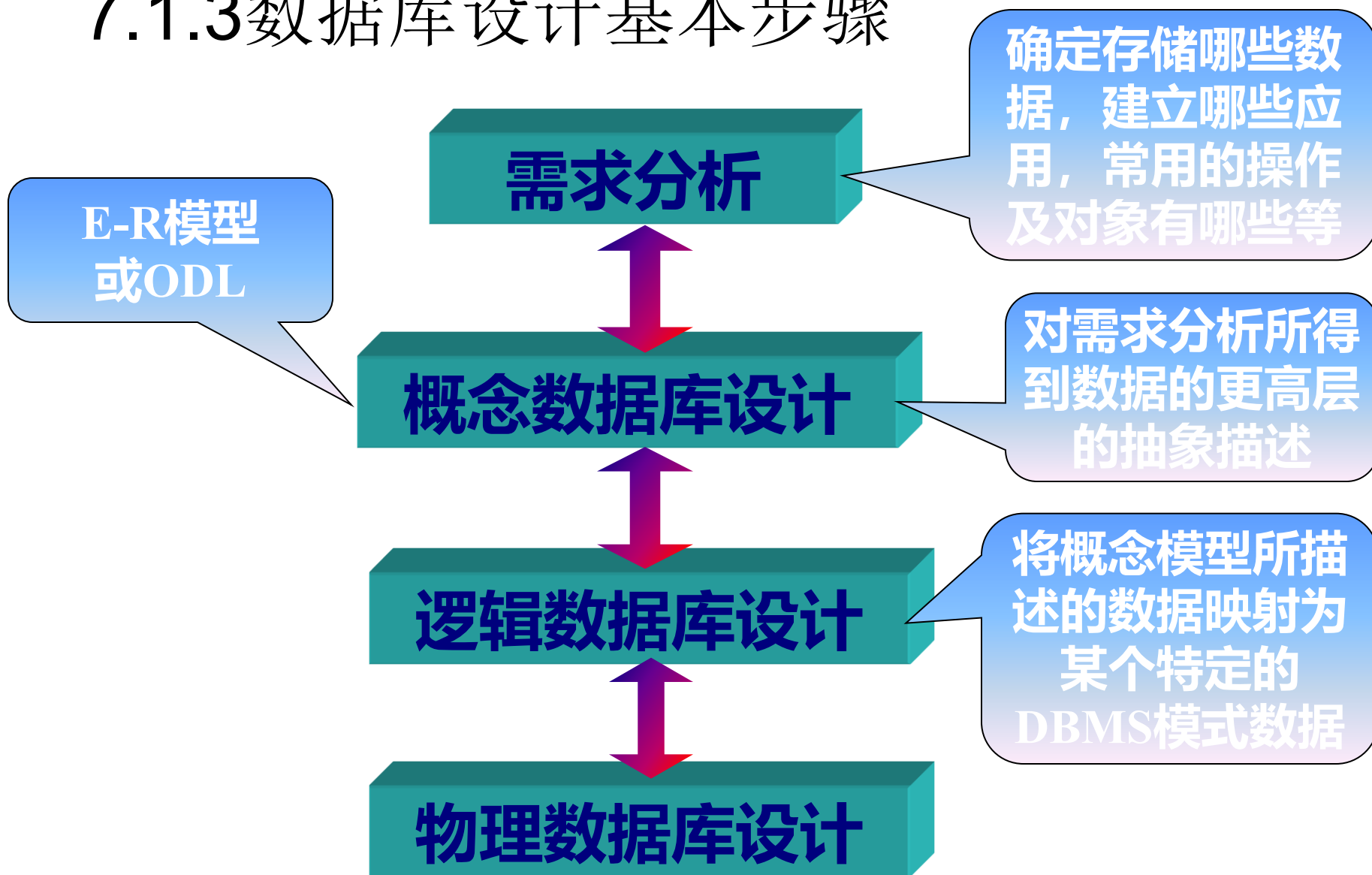
7.1.2 数据库设计方法

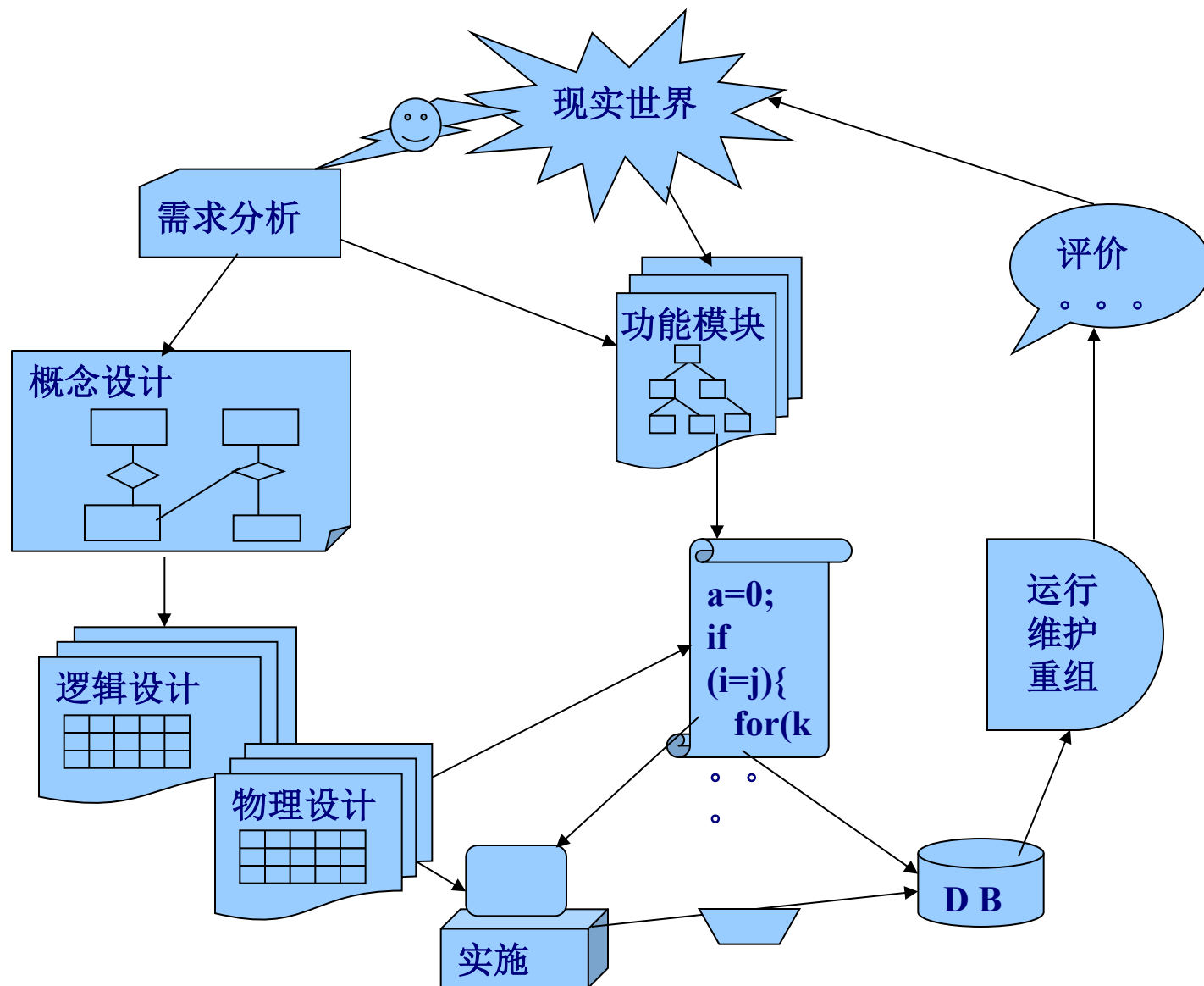
- 新奥尔良（New Orleans）方法
 - 将数据库设计分为若干阶段和步骤
- 基于E-R模型的数据库设计方法
 - 概念设计阶段广泛采用
- 3NF（第三范式）的设计方法
 - 逻辑阶段可采用的有效方法
- ODL（Object Definition Language）方法
 - 面向对象的数据库设计方法

■ 数据库设计工具

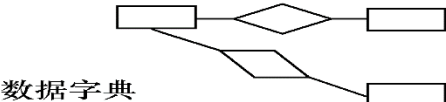
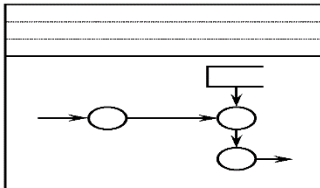
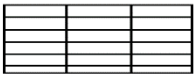
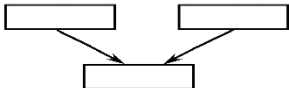
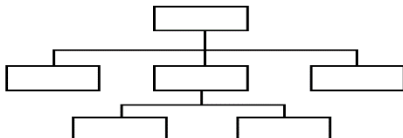


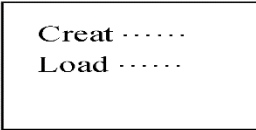
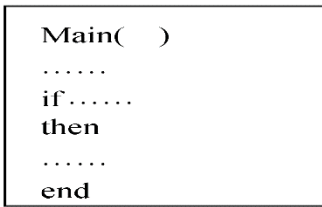
- Sybase *PowerDesigner*
- Oracle *Designer 2000*
- ERWin

7.1.3 数据库设计基本步骤

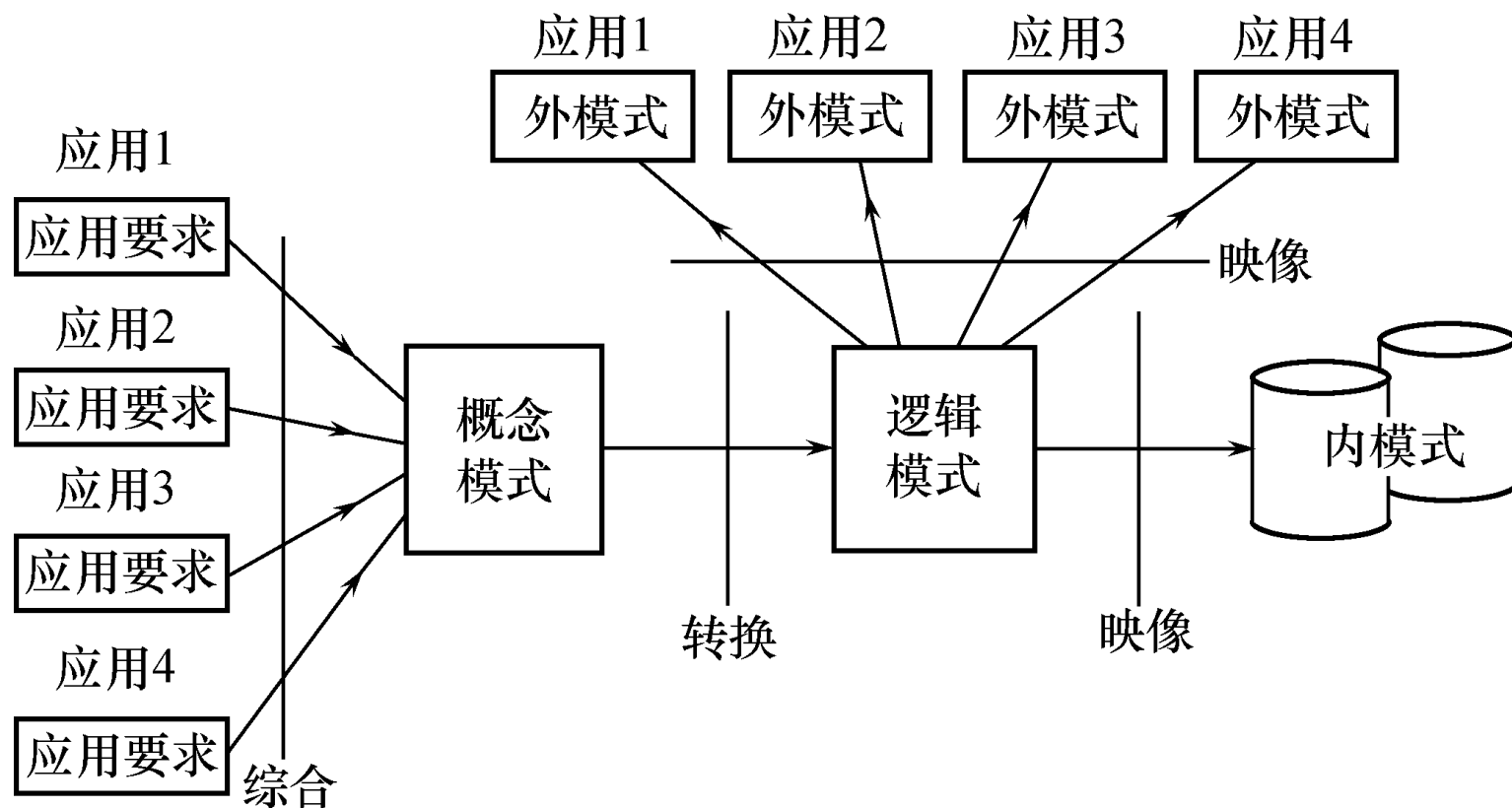




数据库设计各个阶段的设计描述

设计阶段	设计描述	
	数 据	处 理
需求分析	数据字典、全系统中数据项、数据流、数据存储的描述	数据流图和判定表（判定树）、数据字典中处理过程的描述
概念结构设计	概念模型（E-R图） 	系统说明书包括： ① 新系统要求、方案和概图 ② 反映新系统信息流的数据流图 
逻辑结构设计	某种数据模型 关系  非关系 	系统结构图 （模块结构） 
物理设计	存储安排 方法选择 存取路径建立 	模块设计 IPO 表 
数据库实施阶段	编写模式 装入数据 数据库试运行 	程序编码、编译联结、测试 
数据库运行和维护	性能监测、转储 / 恢复 数据库重组和重构	新旧系统转换、运行、维护（修正性、适应性、改善性维护）

7.1.4 数据库设计过程中的各级模式



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 需求分析的步骤
- 7.2.4 需求分析阶段文档

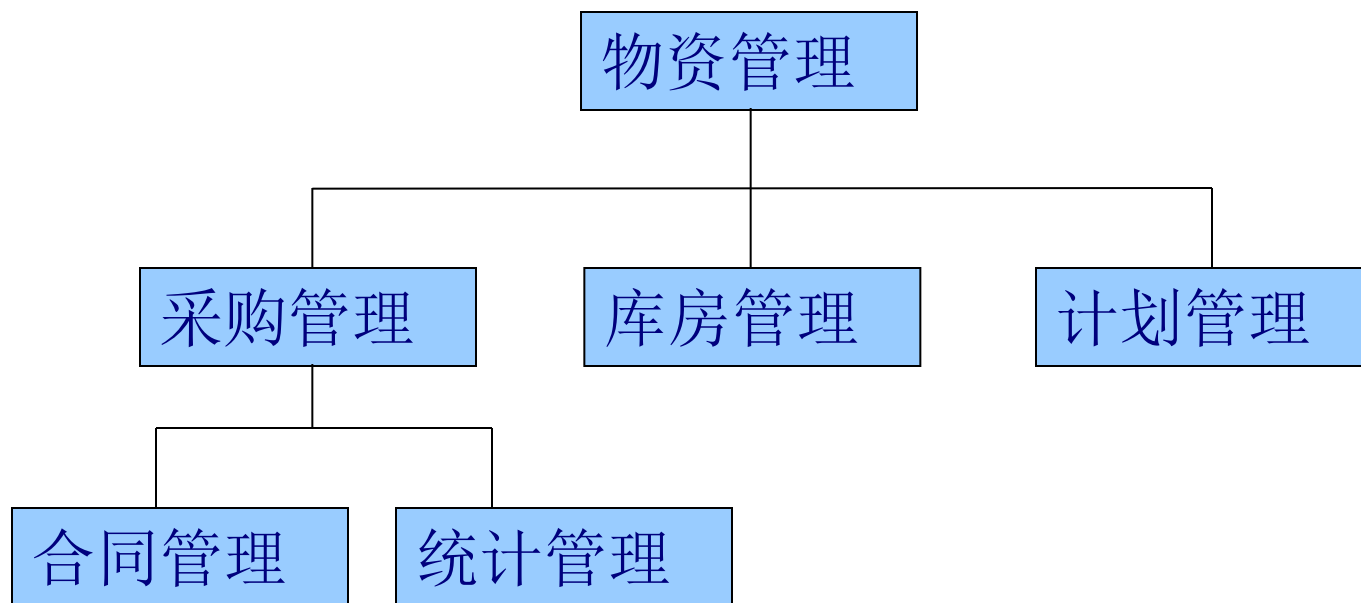
7.2 需求分析

■ 7.2.1 需求分析的任务

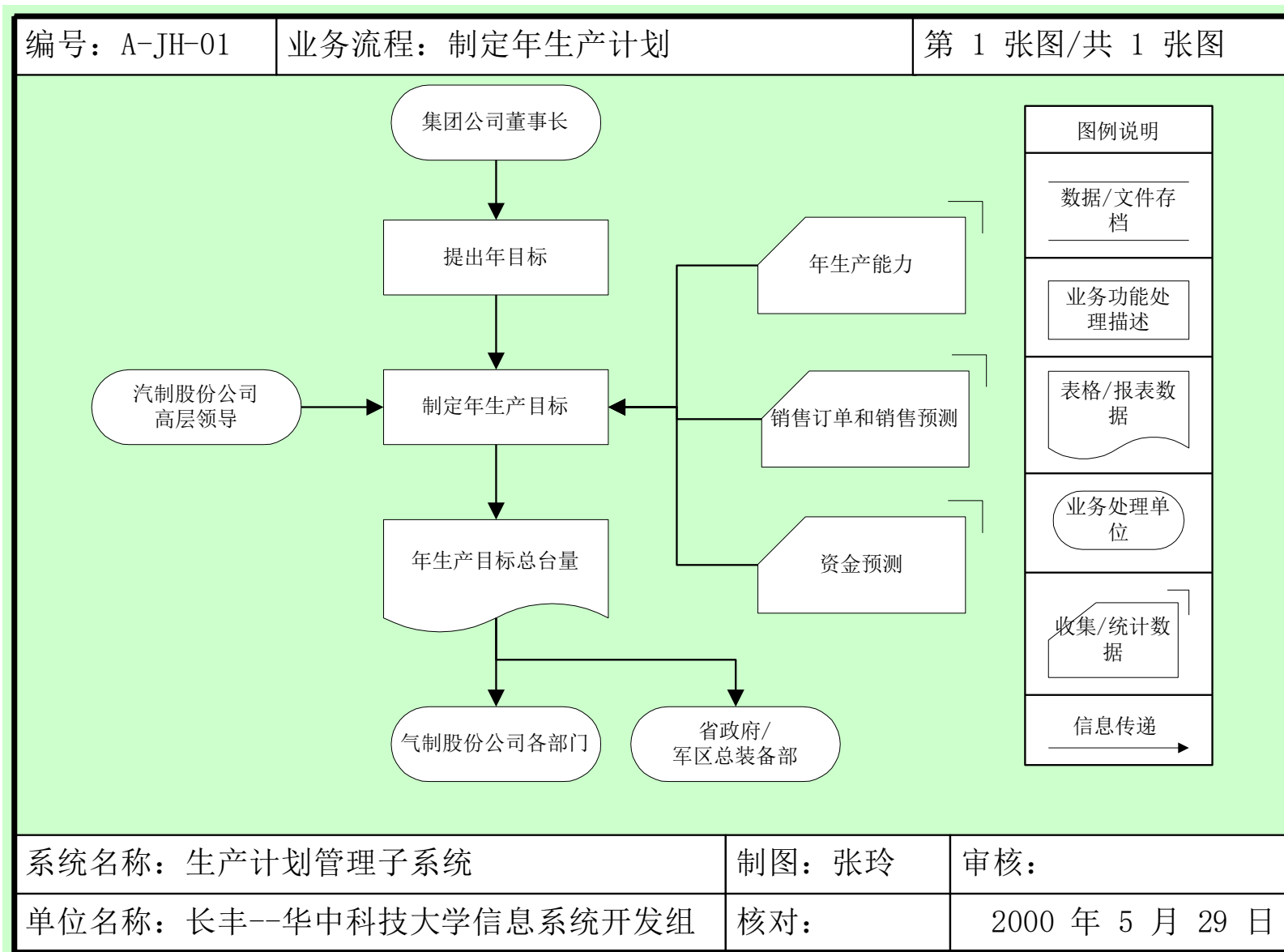
- 需求分析就是从系统数据处理加工过程中抽象并描述出系统的概念模型，为系统的实现做准备。
- 了解系统数据来源、流向、处理过程、处理结果均是需求分析阶段必须完成的工作。

- 信息需求
- 处理需求（功能需求）
- 完整性、安全性需求
- 提交需求说明文档

功能树



功能间的数据关联



需求说明文档

级别 编号	模块名称	功能说明	子模块	操作步骤	操纵数据
01# #	用户登录 子模块	<p>1.通过用户名和用户口令来控制该系统的合法用户，以及这些用户相应的权限</p> <p>2.用户分成高级用户和普通用户两类，其中高级用户为教务管理科工作人员，普通用户包括管理科工作人员和各院系教务员</p>		<p>1.选择用户名；</p> <p>2.输入相应口令；</p> <p>3.系统判断该用户的合法性以及相应权限，并进入相应操作界面</p>	系统用户功能权限表（**）

7.2 需求分析（续）

■ 7.2.2 需求分析的方法

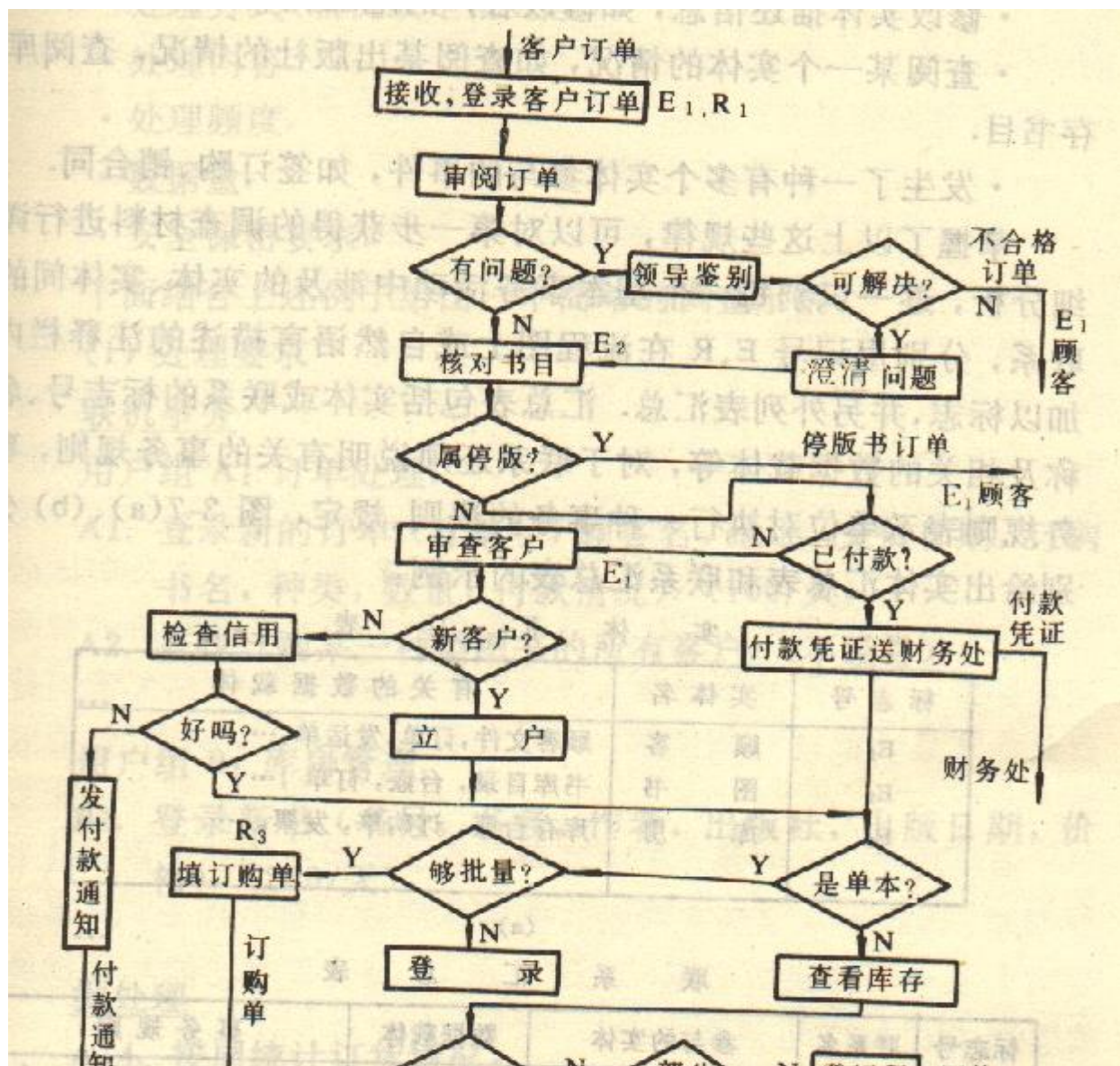
□ 1. 方法分类

■ 面向数据的方法

- 着眼于数据对现实世界的描述作用

■ 面向过程的方法

- 着眼于数据在各项功能活动中被加工变换的流程



7.2.2 需求分析的方法(续)

■ 2. 结构化分析(SA)方法简介

□ 工具：数据流程图、数据字典

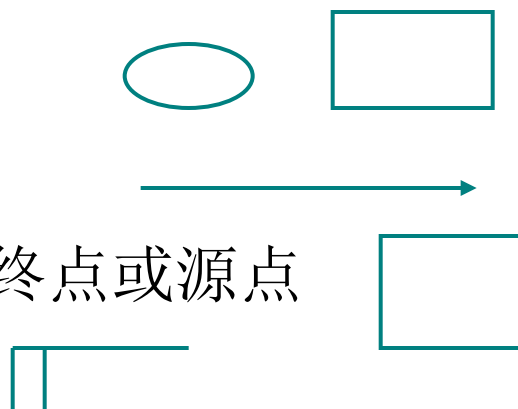
● 数据流程图

● 处理过程

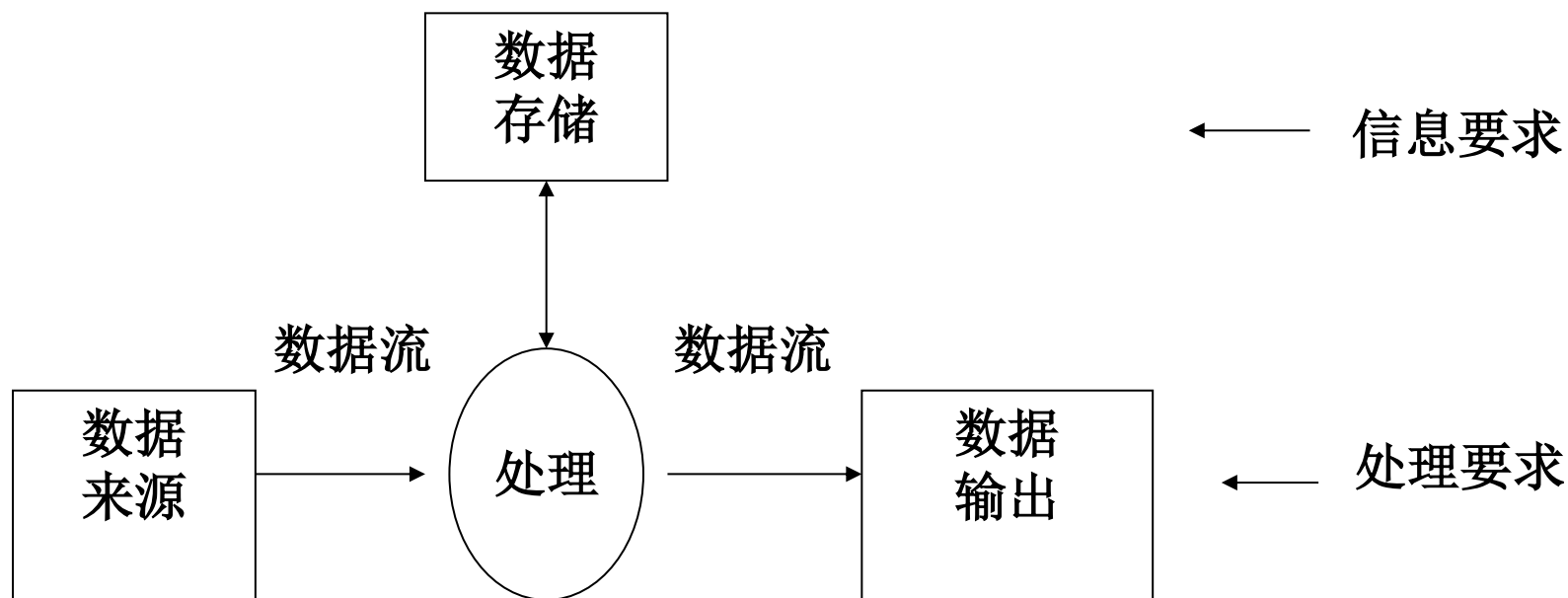
● 数据流

● 数据流的终点或源点

● 存储池



系统高层抽象图



7.2.3 需求分析的步骤

■ 1.需求分析的步骤

- 调查组织结构 → 系统的管理模式 → 各部门功能
- 调查各部门业务活动、职责 → 信息流程
- 收集各种静态信息 → 原系统的信息存储
- 新系统的功能
- 新系统的边界

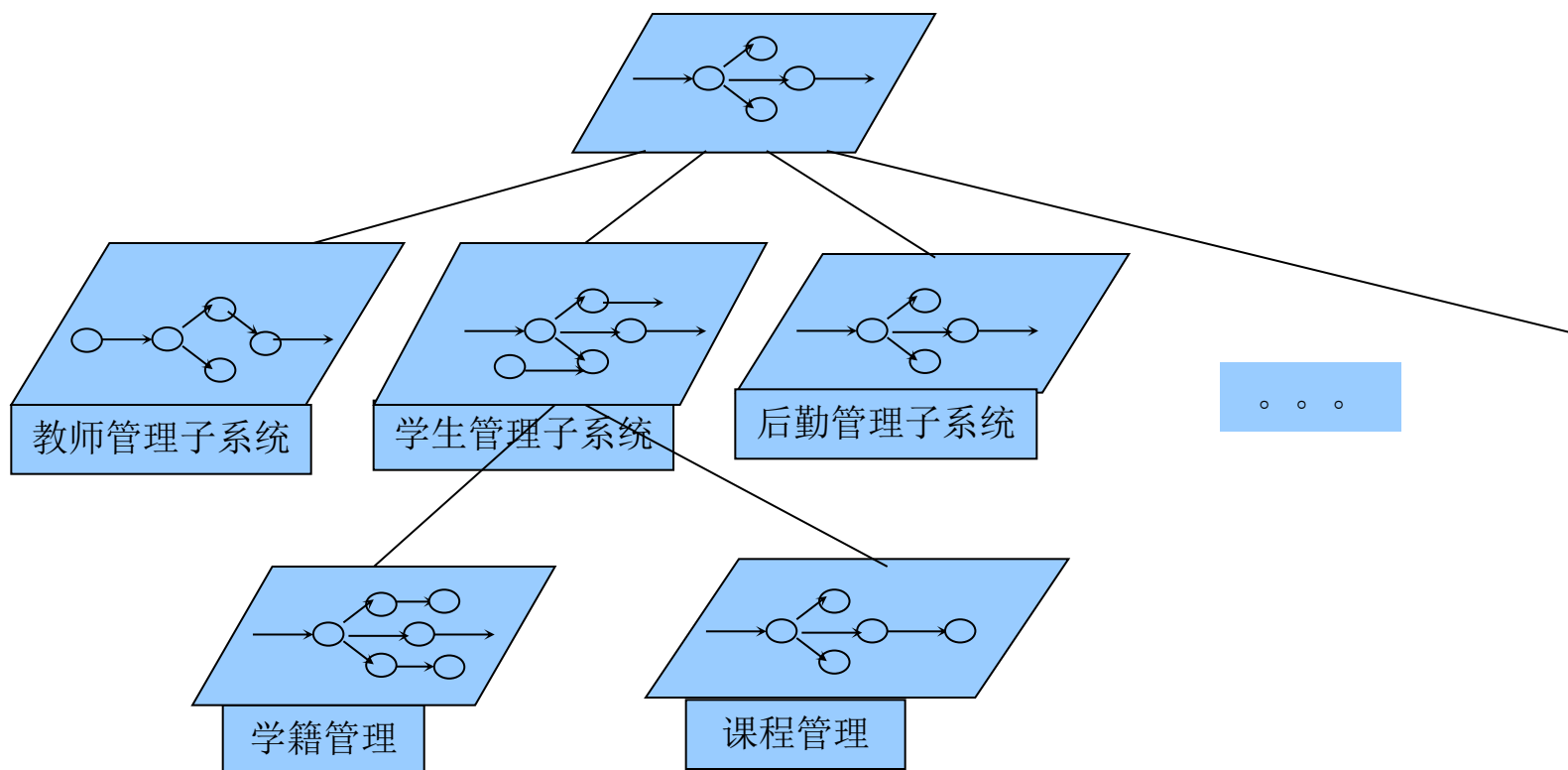
7.2.3 需求分析的步骤(续)

■ 2. 常用的需求分析调查方法

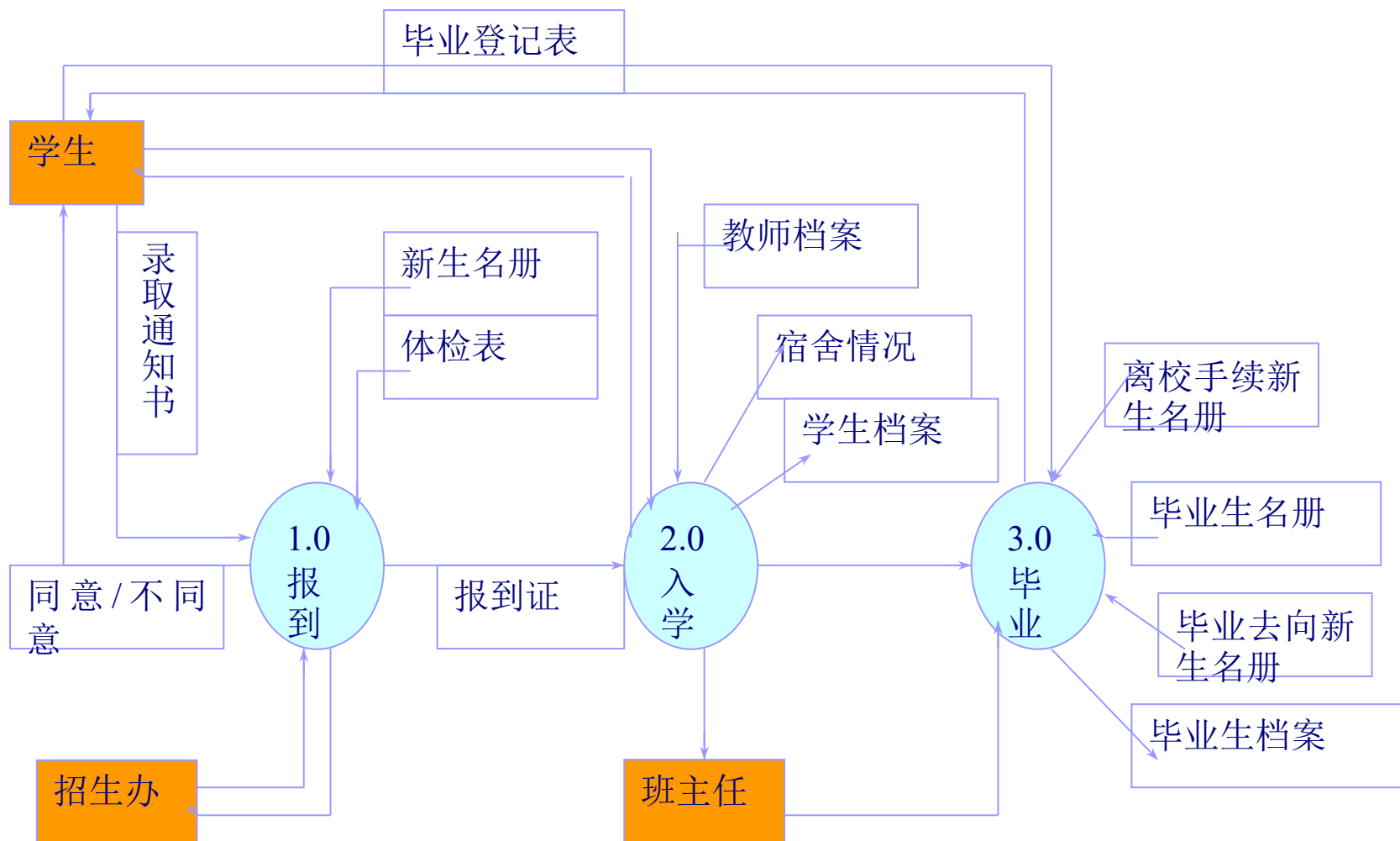
- 跟班作业
- 开调查座谈会
- 请用户介绍
- 提问
- 设计调查问卷请用户填写
- 查阅历史记录
- 在实际操作中往往是若干种方式同时进行

需求分析实例

- 假设开发某学校信息管理系统，经过可行性分析和初步调查，采用自顶向下的方法可以抽象出该系统高层数据流图。逐步往下求精得出：教师管理子系统、学生管理子系统、后勤管理子系统、科研管理子系统、产业管理子系统。限于篇幅我们以学生管理子系统为例向下求精。



学籍管理的数据流图



7.2.4 需求分析阶段文档

- 需求分析阶段结束时，应提供的文档包括：
 - 系统组织结构图；
 - 数据流图；
 - 数据字典（DD）；
 - 数据处理流程图（DFD）等。

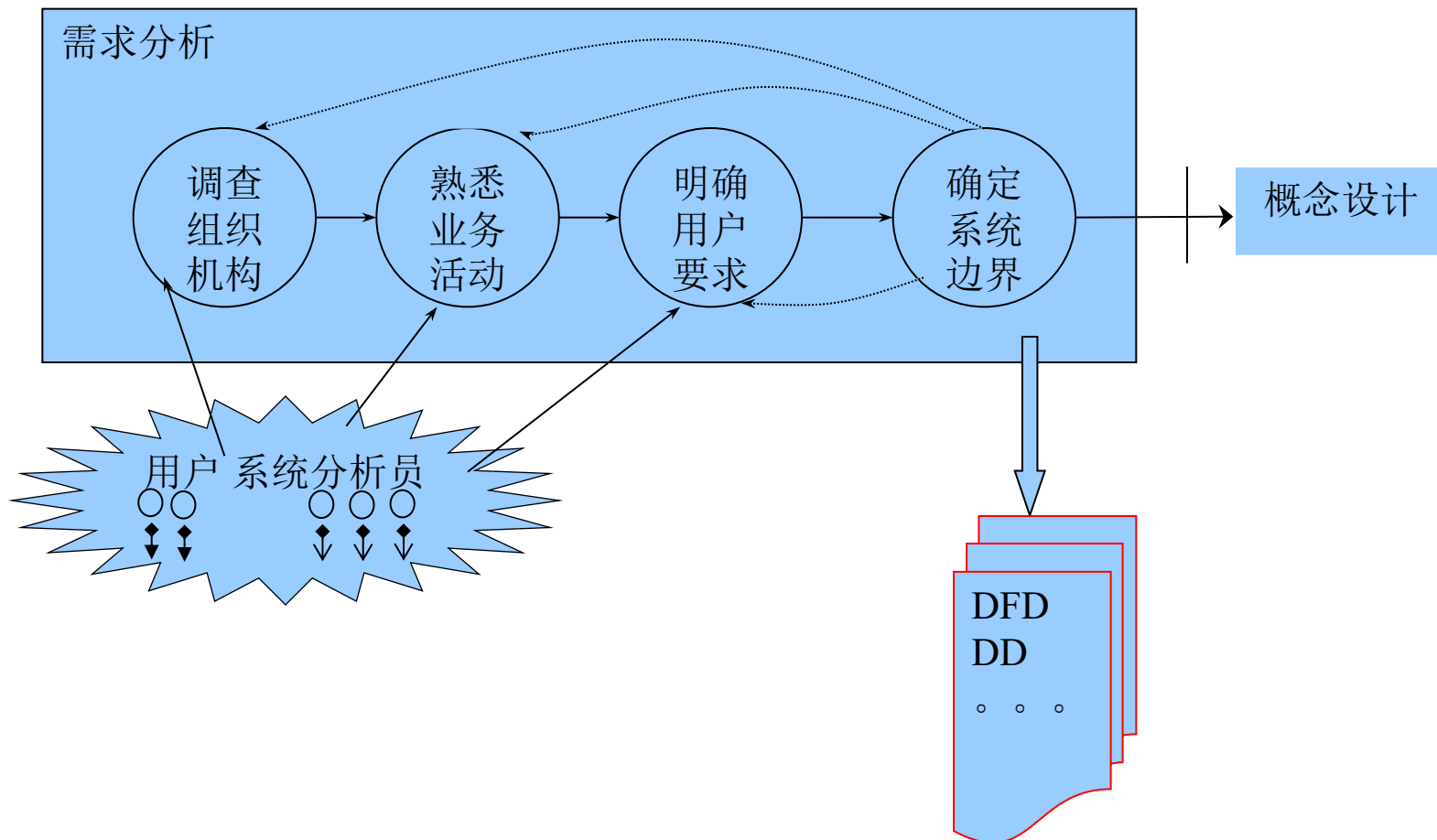
数据字典

- 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

数据字典

- 数据字典是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

需求分析小结



7.3 概念数据库设计

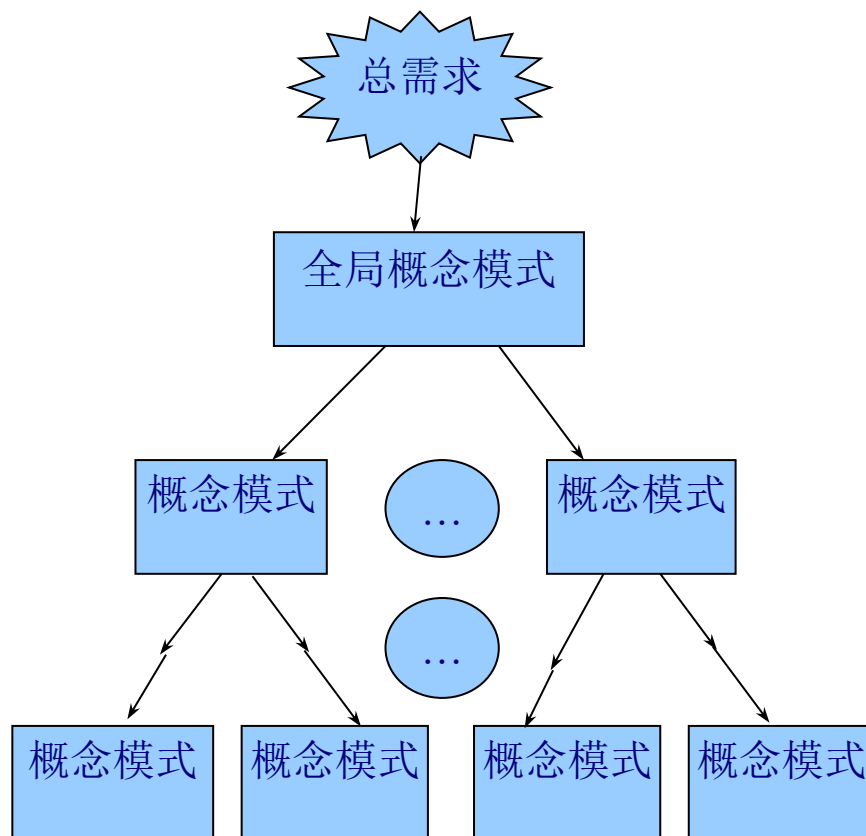
- 7.3.1 概述
- 7.3.2 概念数据库设计方法与步骤
- 7.3.3 数据抽象与局部ER图设计
- 7.3.4 视图集成

7.3.2 概念数据库设计方法与步骤

■ 概念数据库设计方法与步骤

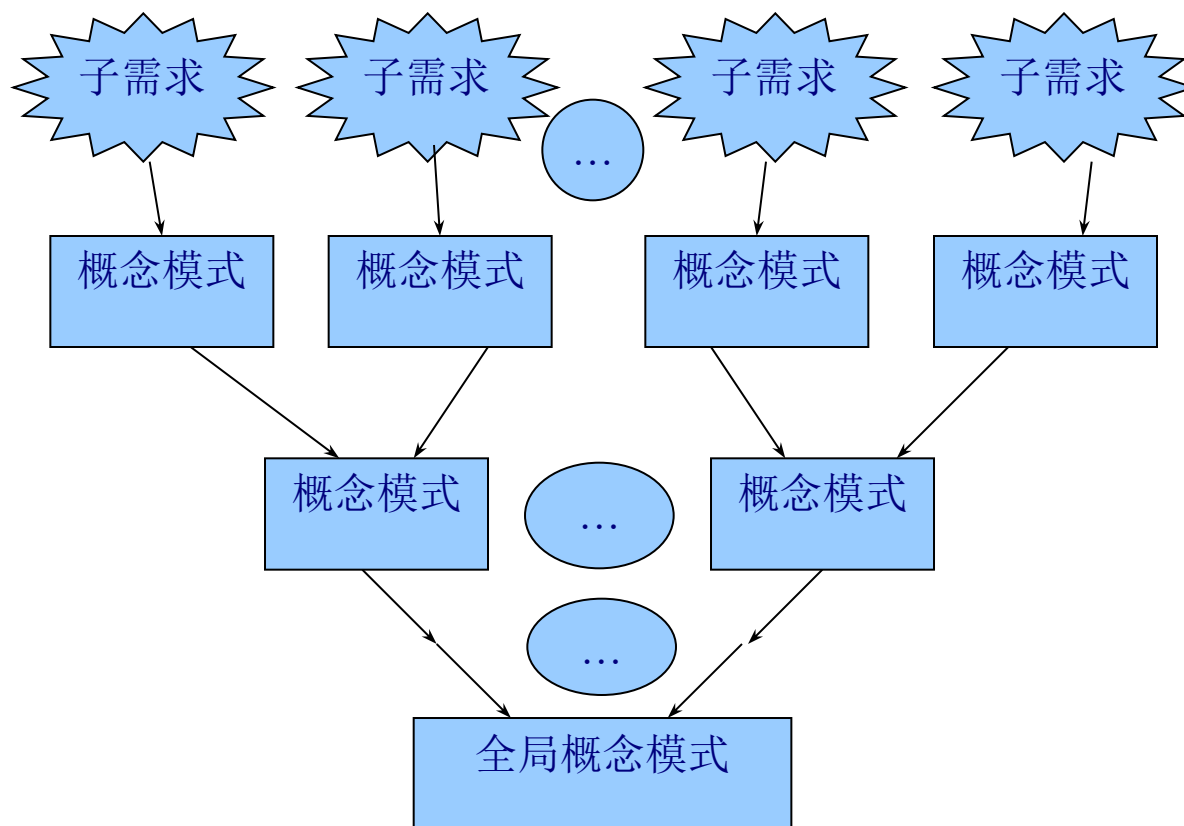
- 自顶向下：首先定义全局的概念模型，然后逐步细化得到局部的概念模型.
- 自底向上：首先定义各局部应用的概念结构，然后将其集成得到全局概念模型.
- 逐步扩张：首先定义最重要的核心概念结构，然后向外扩充，逐步生成其他的概念结构和总体概念结构.
- 混合策略

概念数据库设计方法与步骤



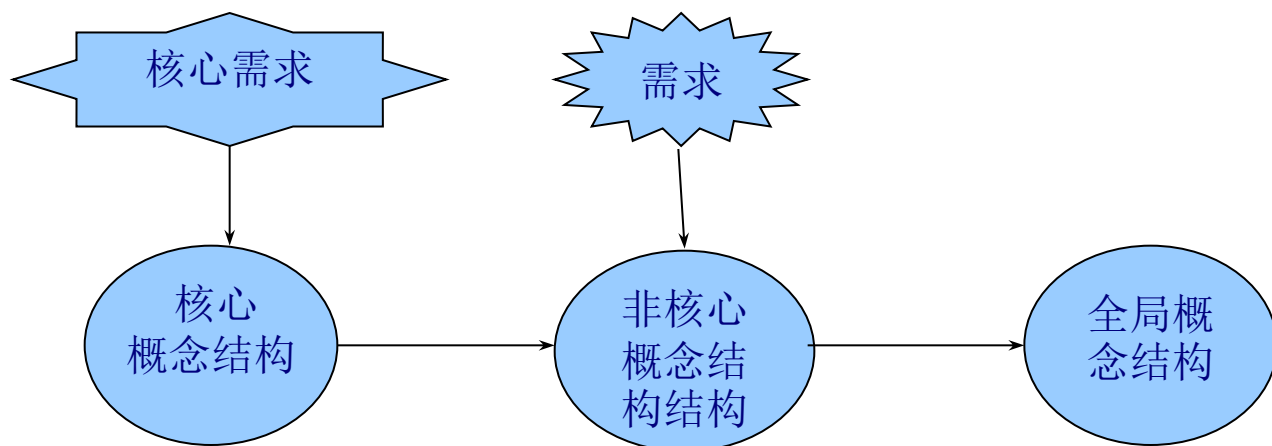
自顶向下

概念数据库设计方法与步骤



自底向上

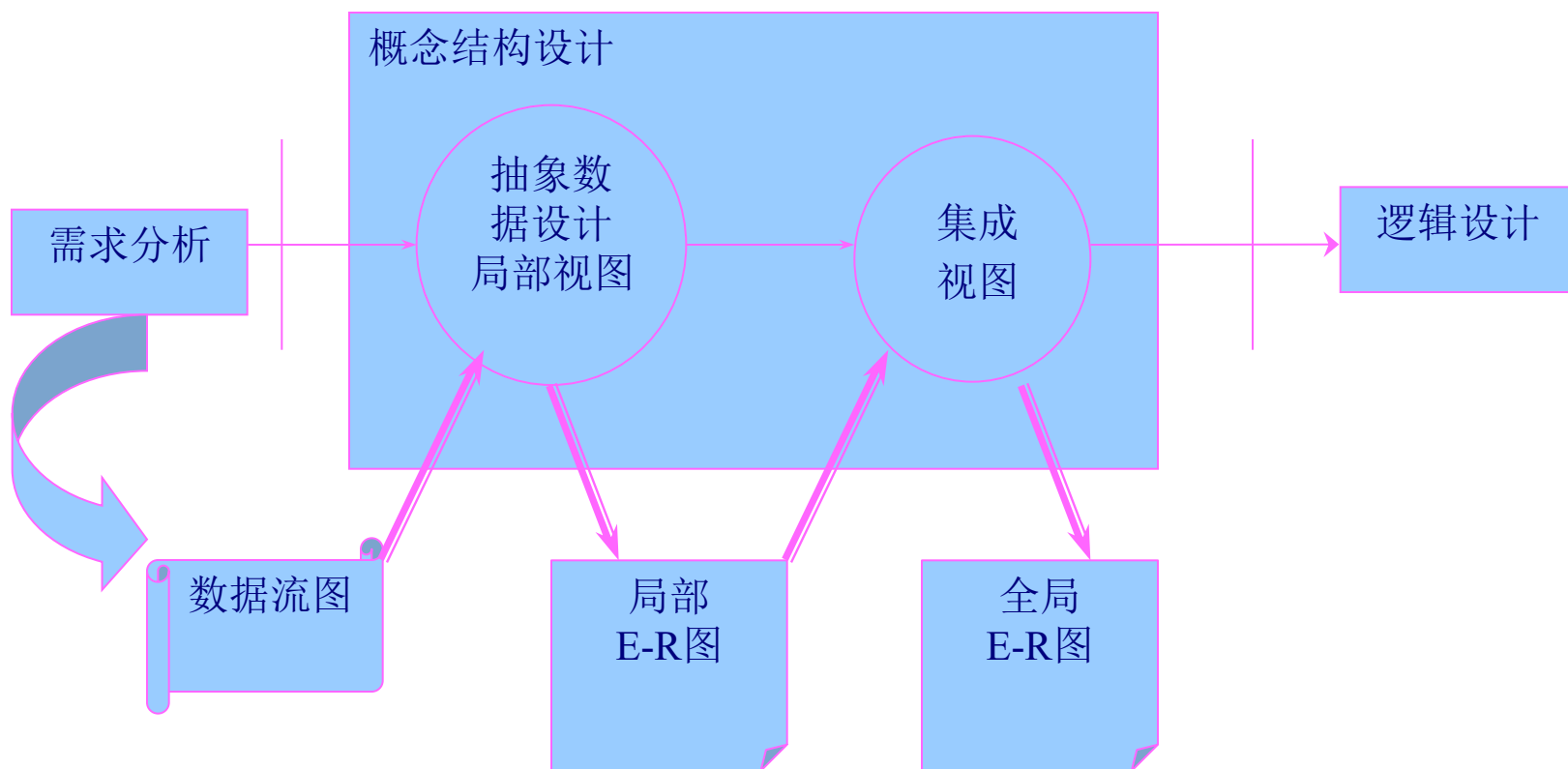
概念数据库设计方法与步骤



逐步扩张

选用哪一种设计策略主要由系统分析员选择，但通常要与需求分析的策略相一致。

采用自底向上策略的设计过程示意图

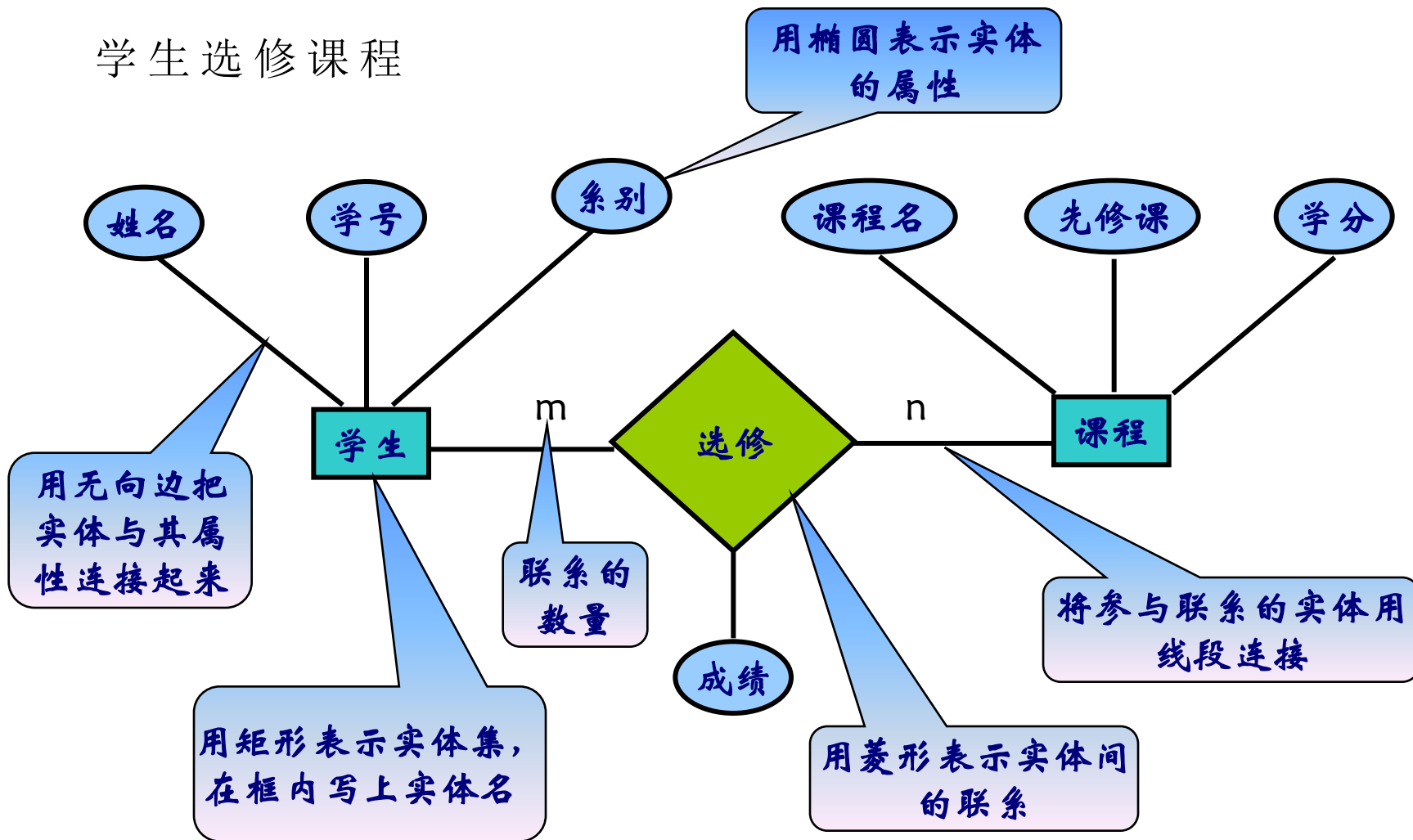


7.3.3 数据抽象与局部E-R图设计

- 1. E-R图基本要点复习
- 2. E-R图基本概念的补充说明
- 3. E-R图设计要点
- 4. 练习

7.3.3.1 E-R图基本要点复习

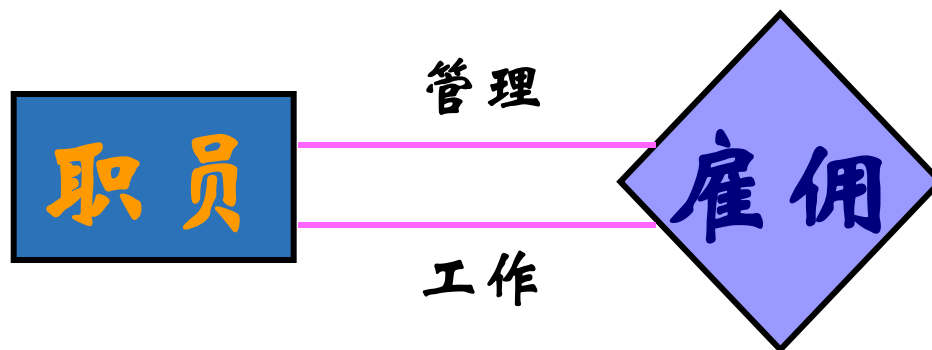
学生选修课程



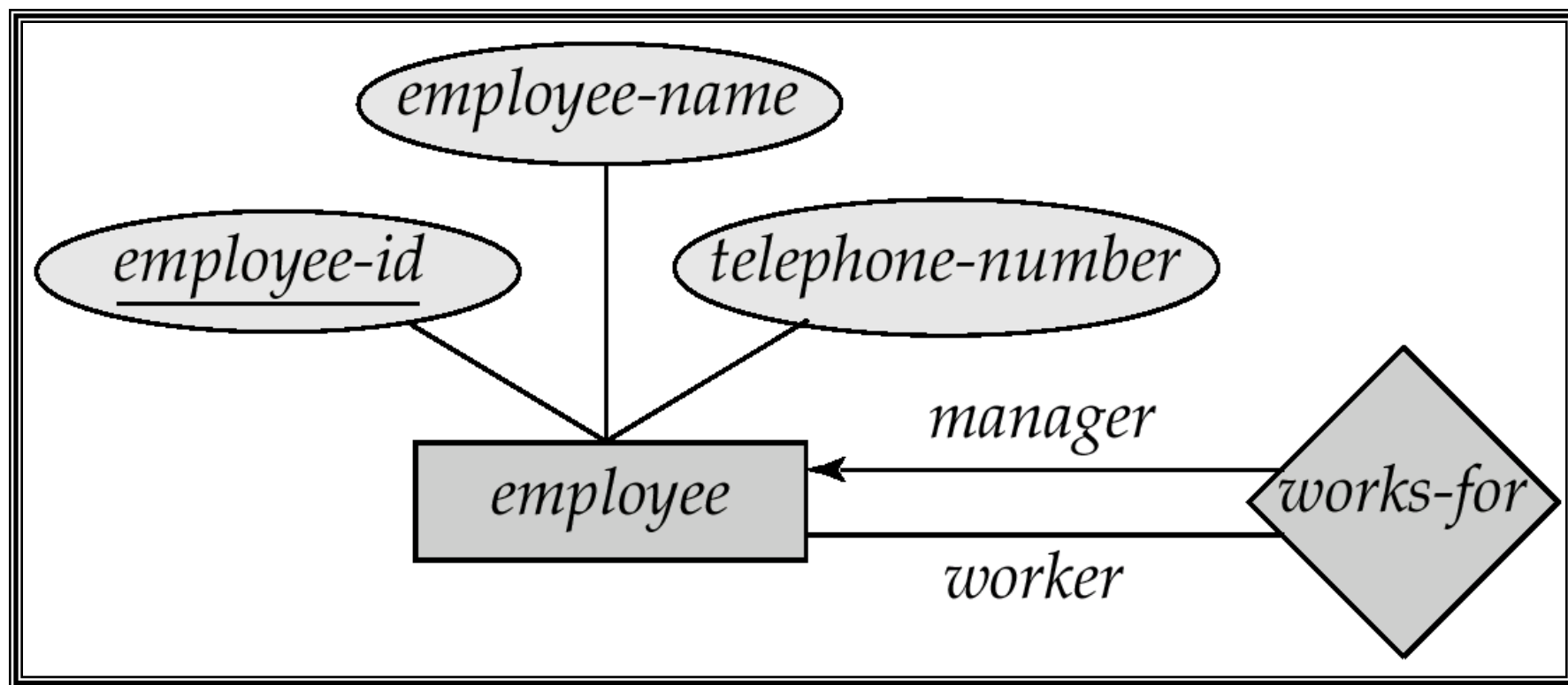
7.3.3.2 E-R图基本概念的补充说明

■ 1. 角色 (Role)

- 实体在联系中的作用称为实体的角色
- 当同一个实体集不止一次参与一个联系集时，为区别各实体的参与联系的方式，需要显式指明其角色
- 当需要显式区分角色时，在连接菱形和矩形的线上加上说明性标注以区别不同的角色



角色在E-R图中的表示



7.3.3.2 E-R图基本概念的补充说明

■ 2. 属性的类型

□ 简单属性

- 不可再分的属性
- 如学号、年龄、性别

□ 复合（Composite）属性

- 可以划分为更小的属性
- 可以把相关属性聚集起来，使模型更清晰

7.3.3.2 E-R图基本概念的补充说明

■ 2. 属性的类型

□ 单值属性

- 每一个特定的实体在该属性上的取值唯一
- 如学生的学号，年龄、性别、系别等

□ 多值属性

- 某个特定的实体在该属性上有多于一个的取值
- 如学生（学号，所选课程，联系电话）

7.3.3.2 E-R图基本概念补充说明

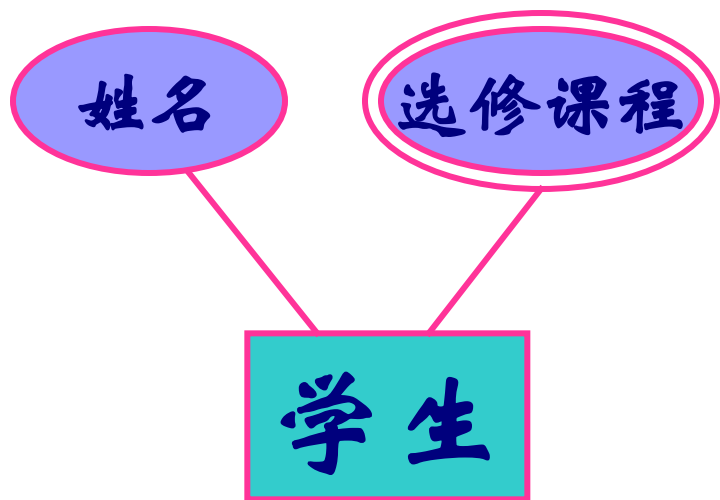
■ 2. 属性的类型

□ 派生属性与基属性

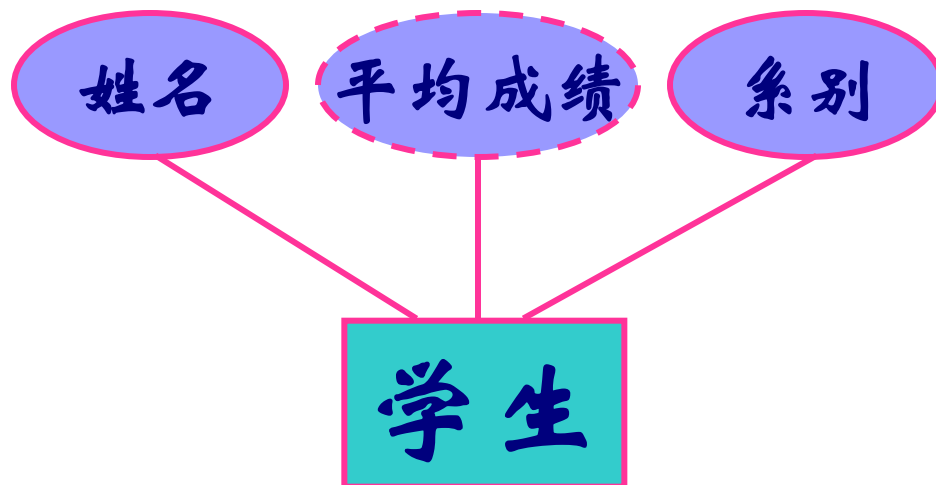
- 派生属性可以从其他相关的属性或实体派生出来的属性值
- 数据库中，一般只存基属性值，而派生属性只存其定义或依赖关系，用到时再从基属性中计算出来
- 勿轻易引入派生属性

属性在E-R图中的表示

多值属性用双椭圆表示



派生属性用虚椭圆表示



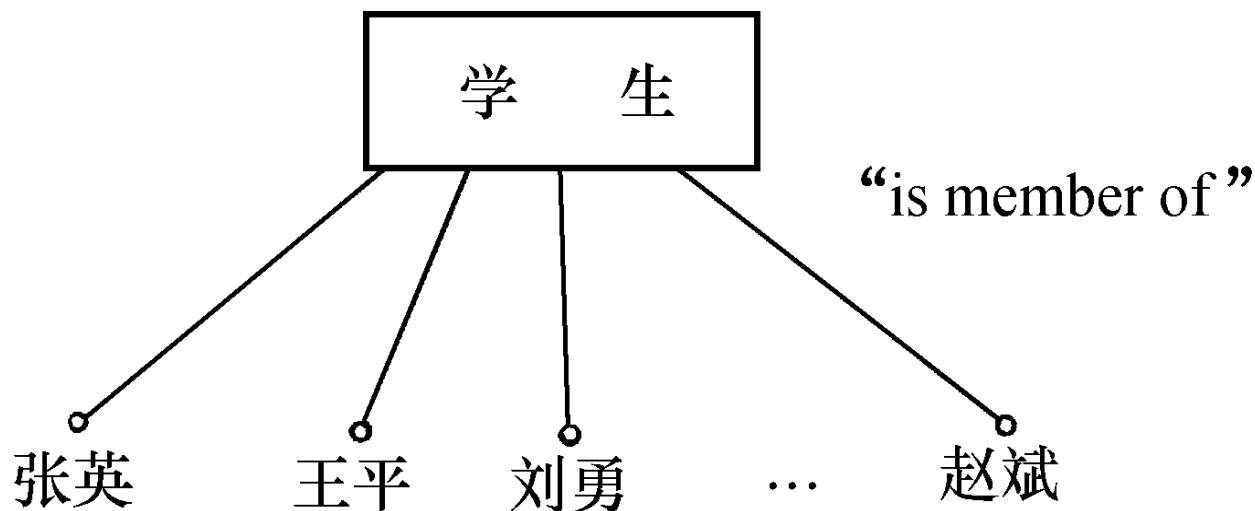
7.3.3.2 E-R图基本概念补充说明

- 扩展ER特性（三种常用数据抽象）
 - 分类（is member of）
 - 聚集（is part of）
 - 概括/特殊化（is subset of / isa）

数据抽象（续）

1. 分类（Classification）

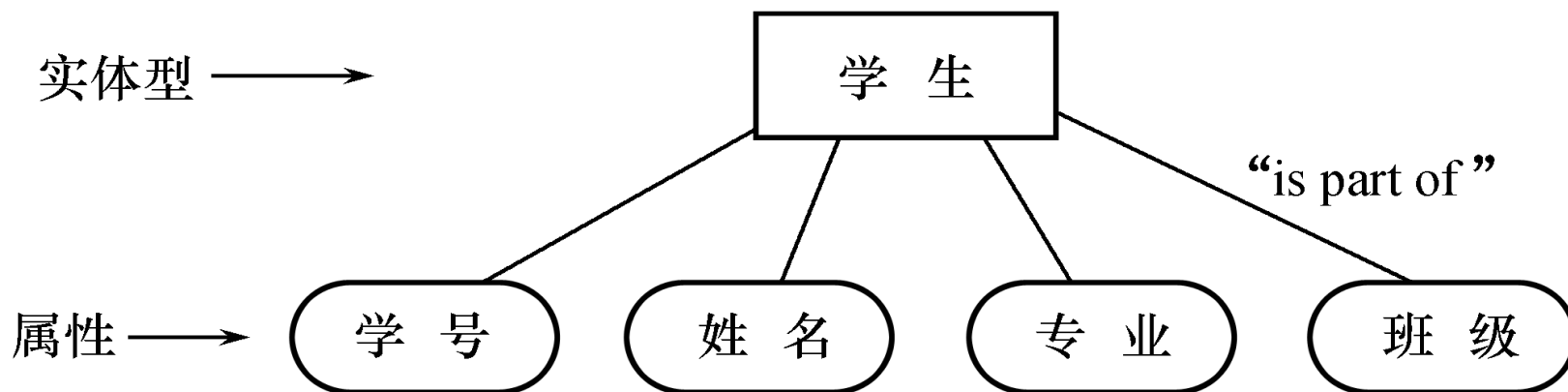
- 定义某一类概念作为现实世界中一组对象的类型
- 抽象了对象值和型之间的“is member of”的语义



数据抽象（续）

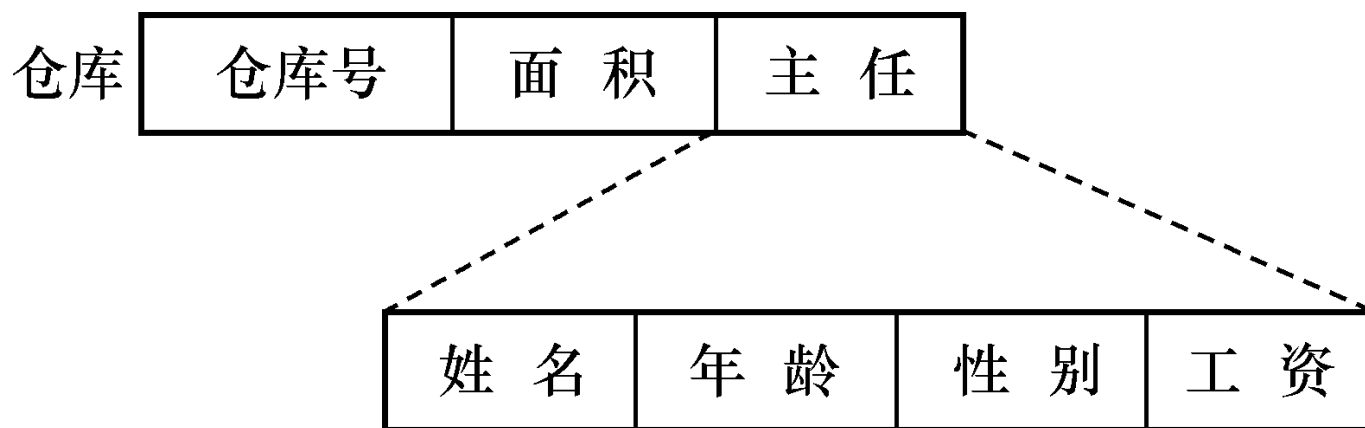
2. 聚集（Aggregation）

- 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（续）

- 复杂的聚集，某一类型的成分仍是一个聚集

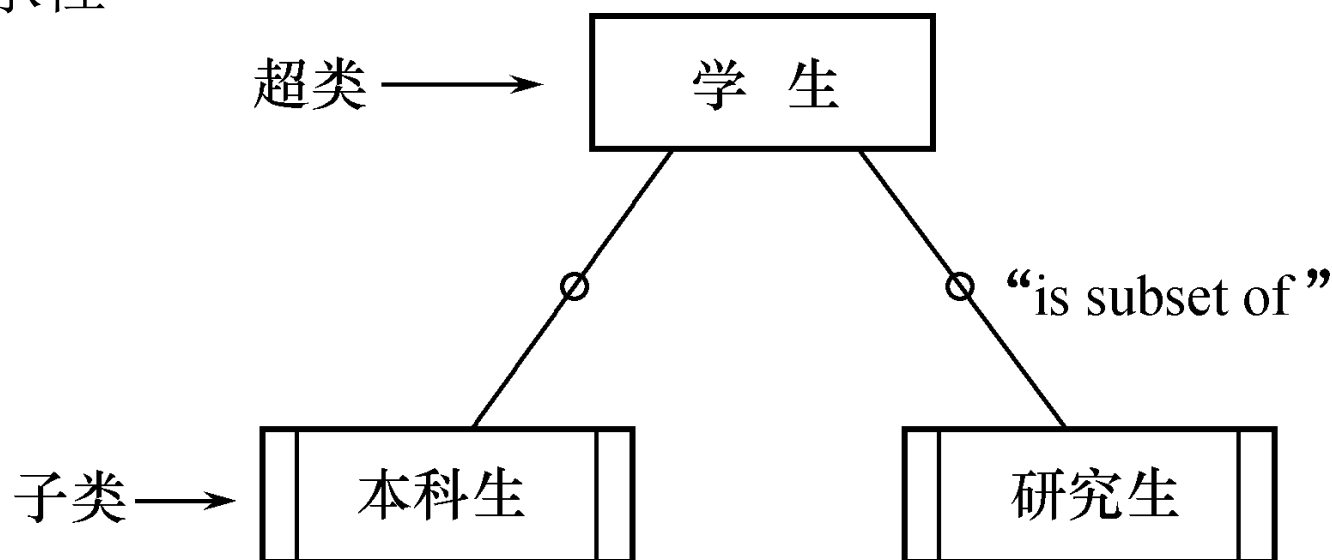


更复杂的聚集

数据抽象（续）

3. 概括（Generalization）

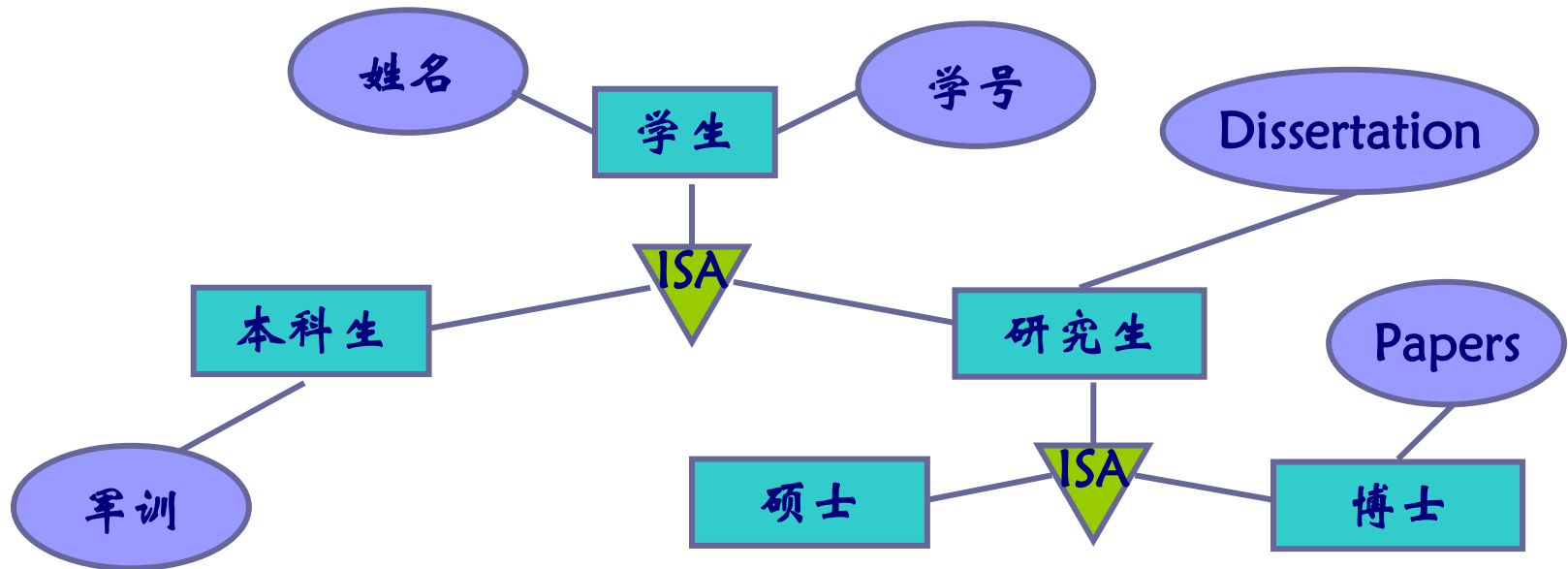
- 定义类型之间的一种子集联系
- 抽象了类型之间的“is subset of”的语义
- 继承性



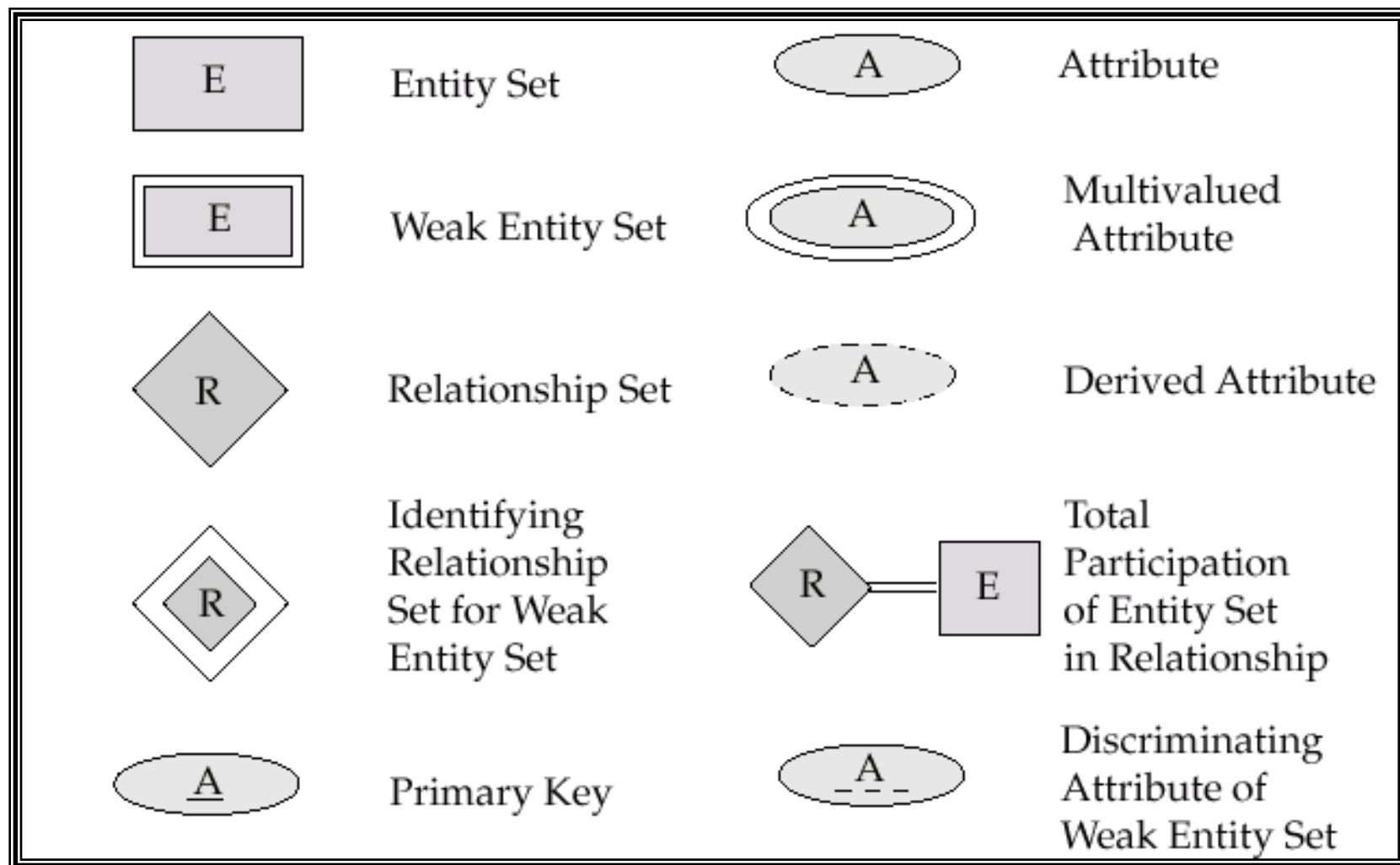
某些扩展E-R特性的表示

■ 概括（Generalization）

- 特殊化用标记为ISA的三角形来表示，表示高层实体和低层实体之间的“父类—子类”联系（或教材P218-221方法）

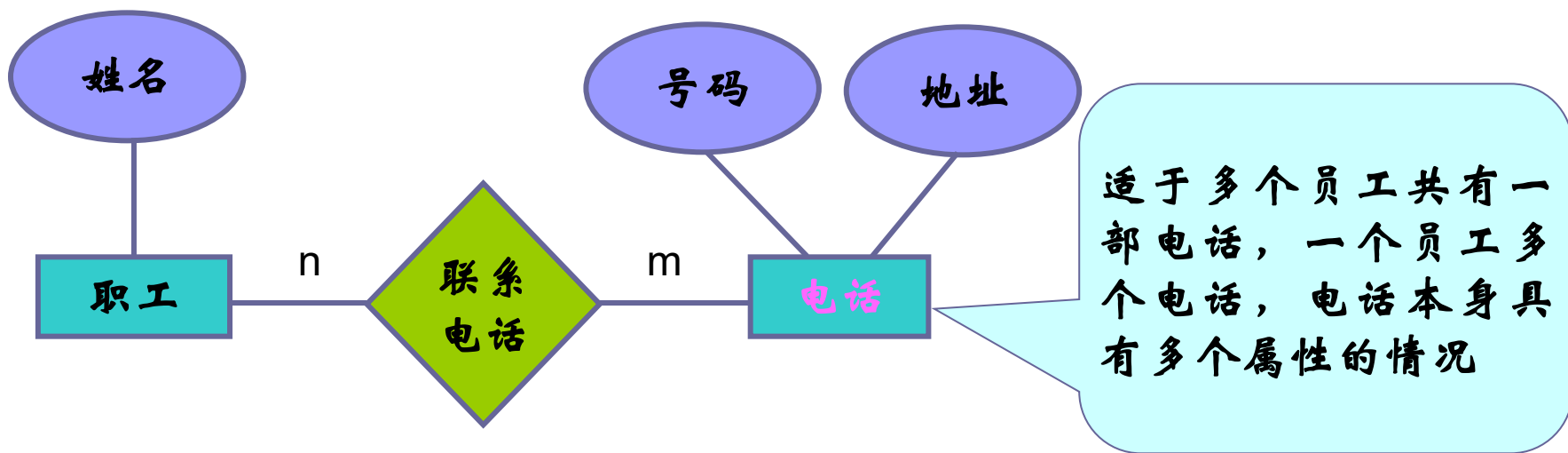
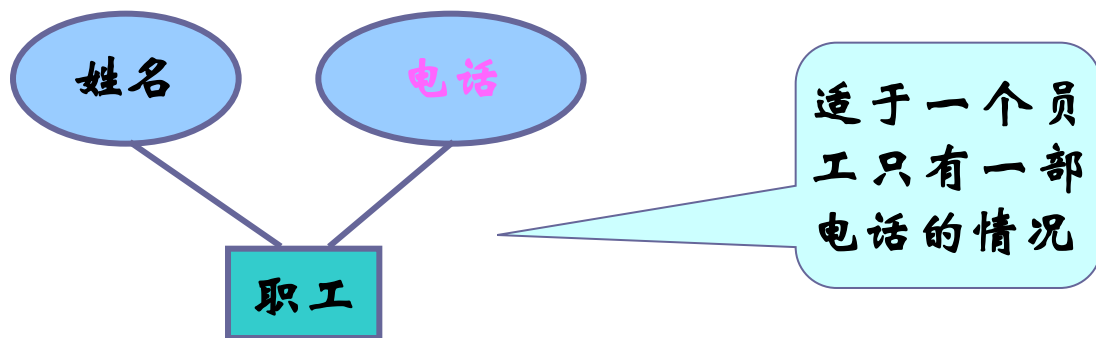


基本E-R图表示汇总



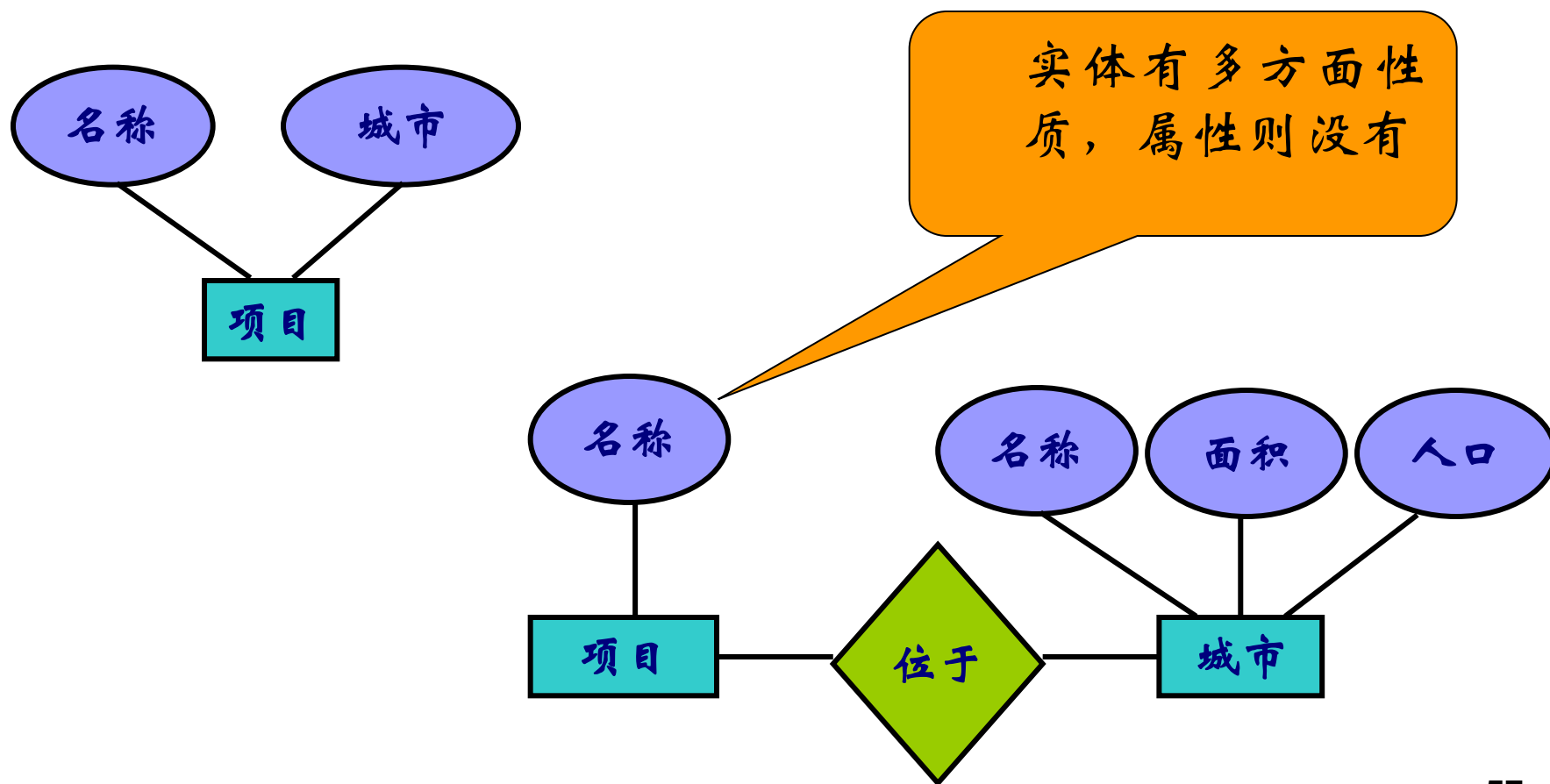
7.3.3.3 E-R图设计要点

■ 1. 实体集 VS 属性



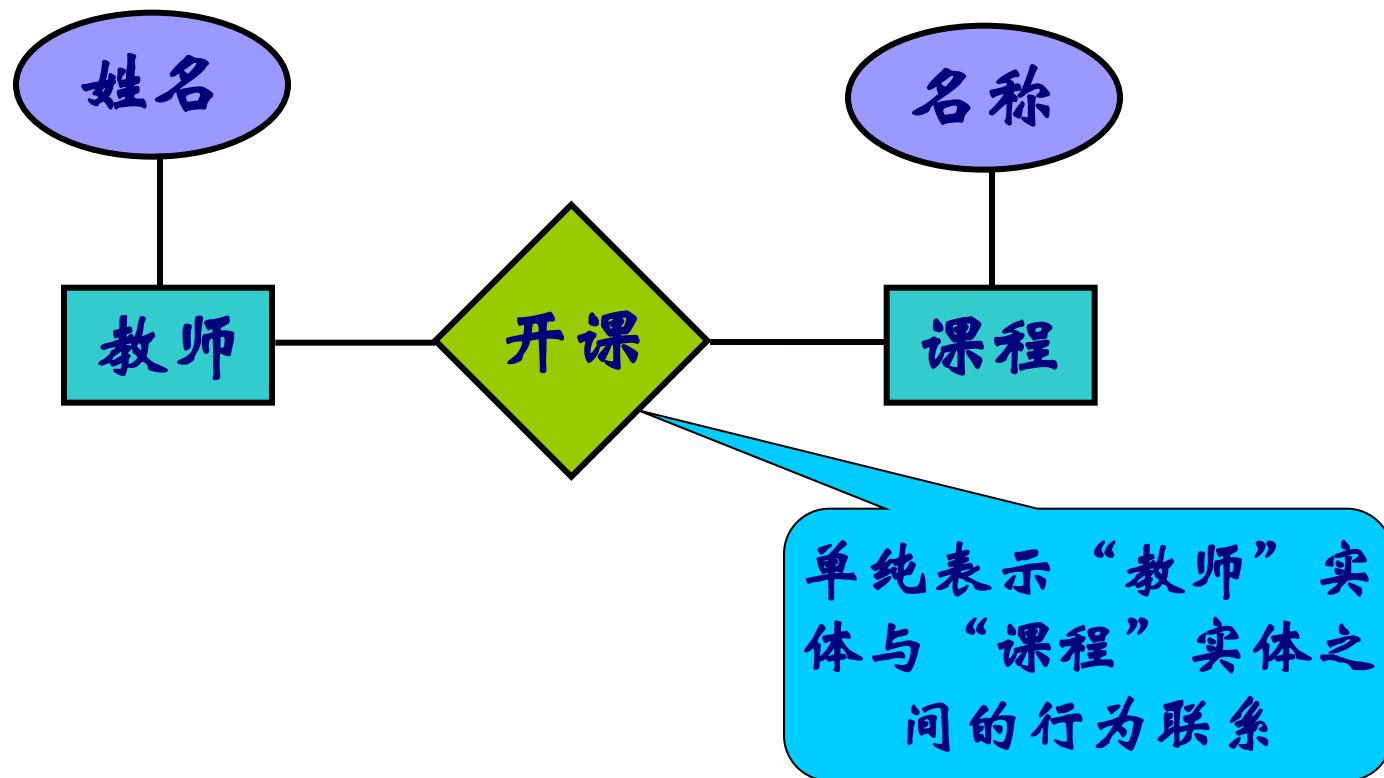
7.3.3.3 E-R图设计要点(续)

■ 1. 实体集 VS 属性



7.3.3.3 E-R图设计要点(续)

■ 1. 实体集 VS 属性



7.3.3.3 E-R图设计要点(续)

■ 2. 实体集与属性一般的区分原则

- 实体一般都有描述信息，而属性不一定
- 多值的对象类可考虑作为实体
- 一个对象类的某一个描述项如果和另一个对象类存在多对一的关系，那么即使它本身没有描述信息，也宜将这个描述项作为实体
- 使用组合标识的对象类，如果组成这个标识的成分都是其他对象类的标识，一般应定义为联系，但如果不是，则可根据情况定义为实体

7.3.3.3 E-R图设计要点(续)

■ 3. 实体集 VS 联系

□ 存在性联系

- 系有学生，学生有课程，课程有成绩

□ 功能性联系

- 教师教学生，工程师参与工程

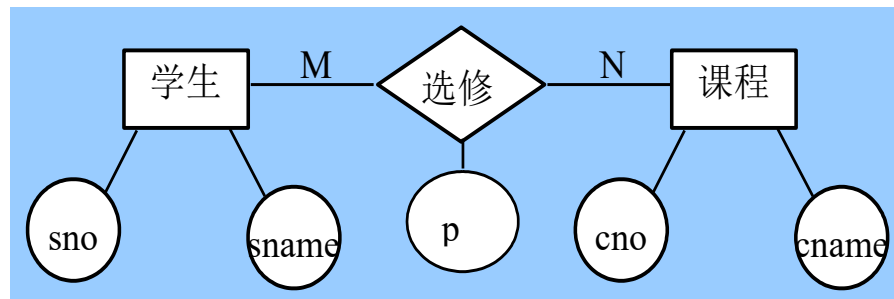
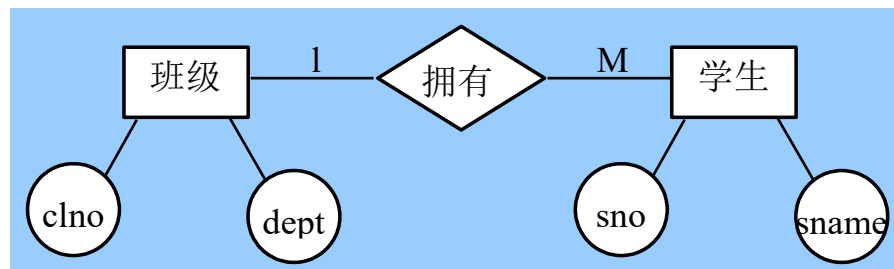
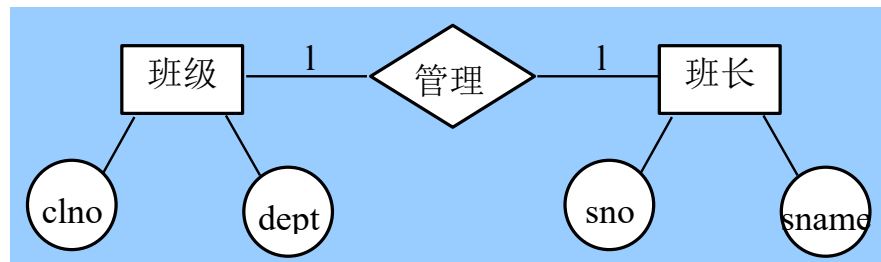
□ 事件联系

- 顾客发出订单，学生借书

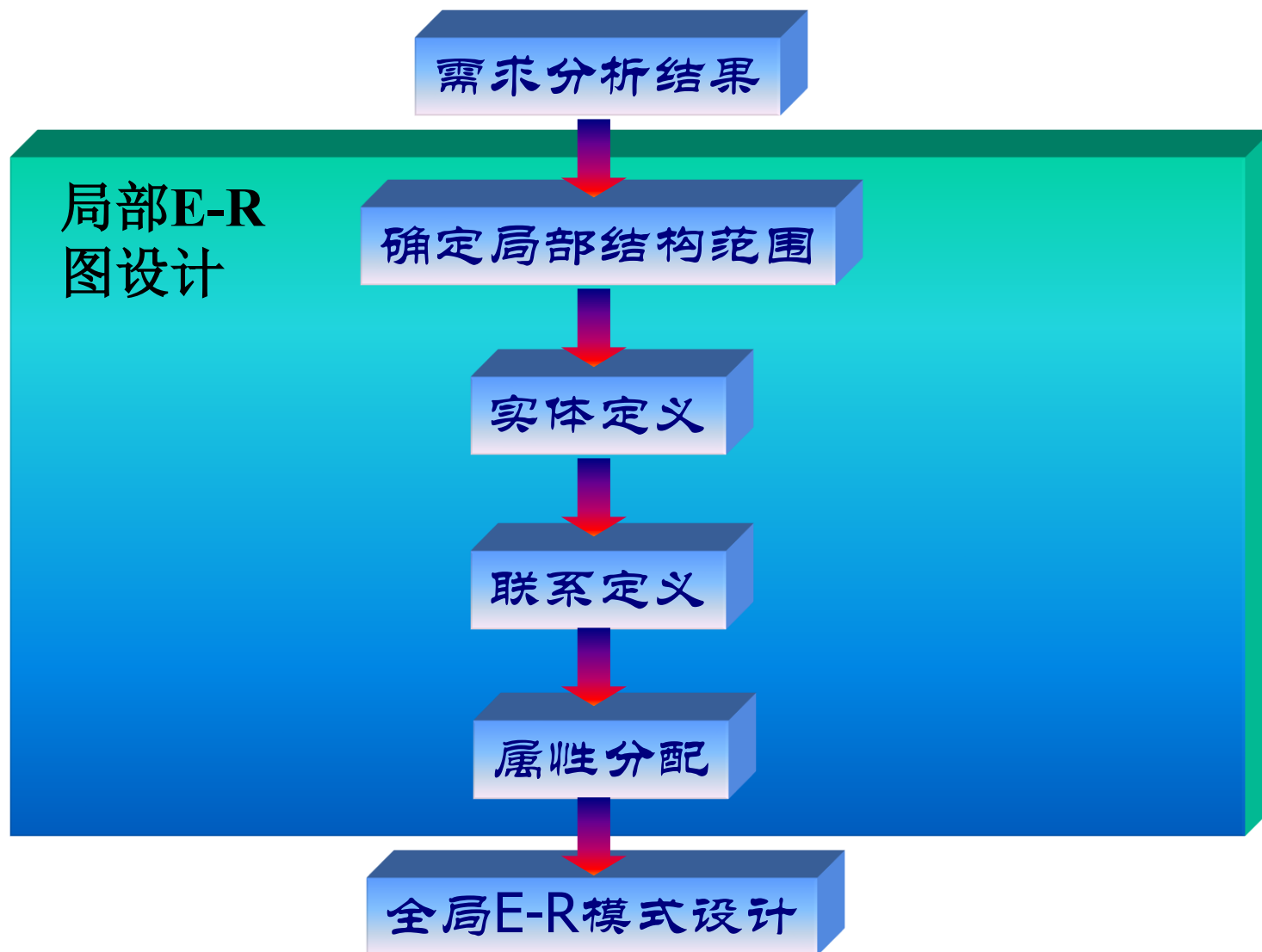
7.3.3.3 E-R图设计要点(续)

■ 4. 联系的确定

- ① **1: 1**时：取任何一方的KEY作为联系的KEY
- ② **1: m**时：取“m”方的KEY作为联系实体的KEY
- ③ **m: n**时：取“双方”的KEY作为联系实体的KEY



局部E-R图设计总结



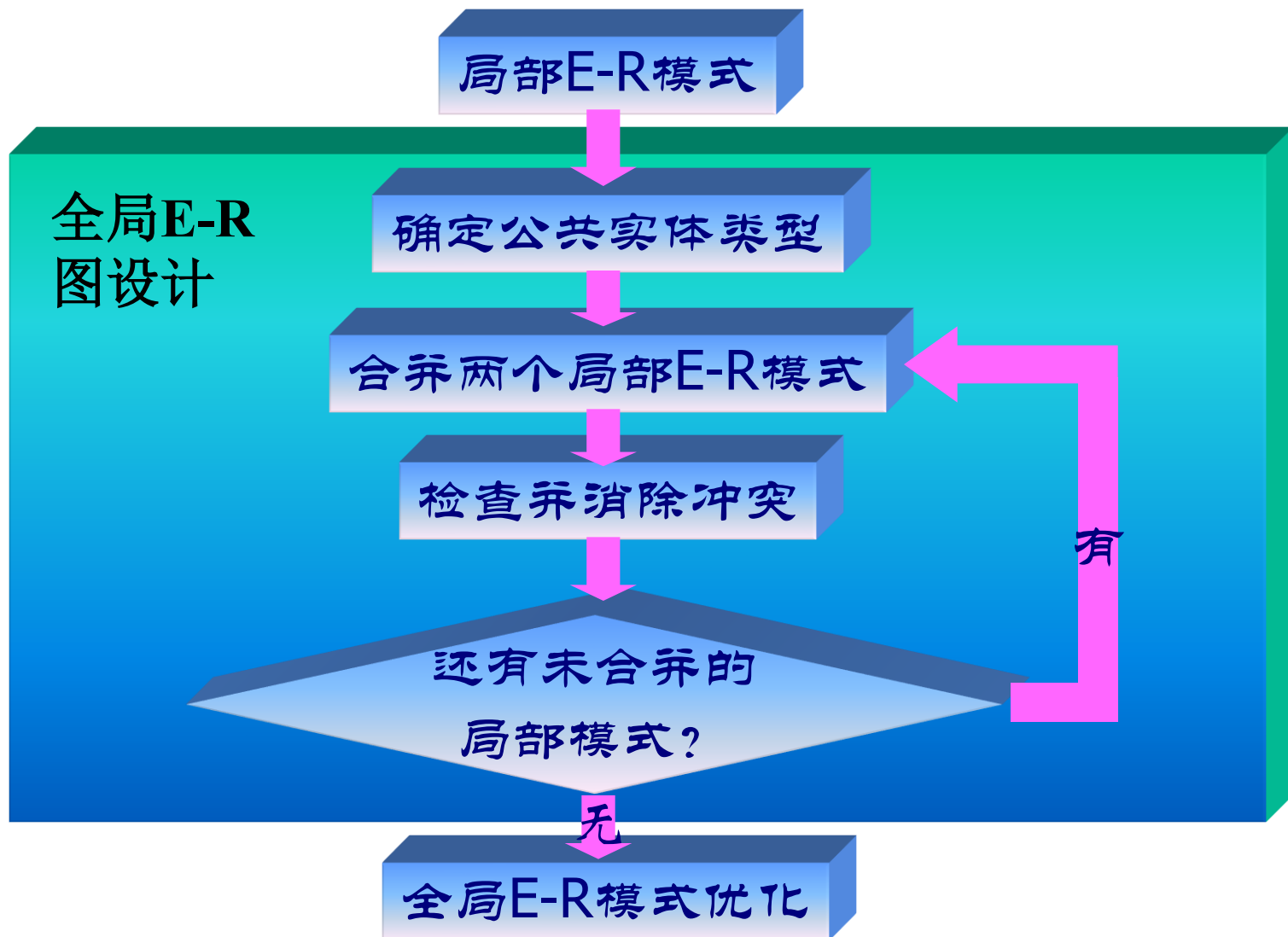
练习

- 请设计一个图书馆数据库，此数据库对每个借阅者保持读者记录，包括：姓名、地址、性别、年龄和单位；对每本书有：书号、书名、作者、出版社；对每本被借出的书有：读者号、借出的日期、归还日期。要求给出**E-R**图，再将其转换为关系模型。

7.3.4 视图集成

- 目的要求：消除冗余、消除冲突、实施集成
 - 用分析法消除冗余属性
 - 用关系理论消除冗余联系
 - 消除属性冲突（包括域类型冲突和取值冲突）
 - 消除结构冲突
 - 消除命名冲突（包括同名异义和异名同义）

全局E-R图设计



7.3.4 消除冲突

■ 属性冲突

- 属性域的冲突：属性的类型、取值范围不同
- 属性取值单位冲突

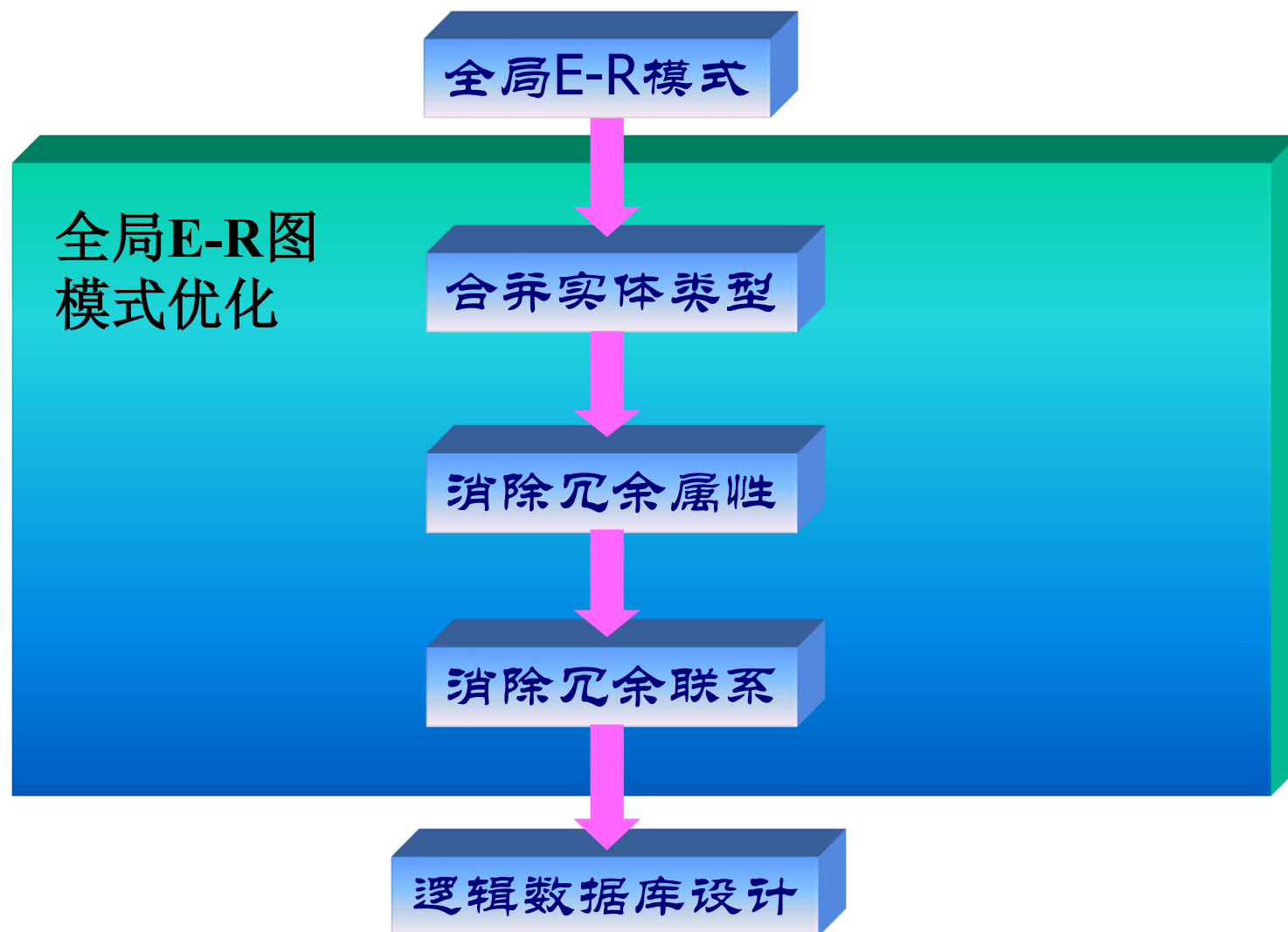
■ 命名冲突

- 同名异义：不同意义的对象具有相同的名字
- 异名同义：同一意义的对象具有不同的名字

■ 结构冲突

- 同一对象在不同应用中的抽象不同
- 同一实体在不同E-R图中属性组成不同
- 实体之间的联系在不同E-R图中呈现不同的类型

全局E-R图模式优化



7.4 逻辑数据库设计

■ 7.4.1 任务

- E-R图转换为等价的关系模型结构

■ 7.4.2 步骤

- E-R \Rightarrow 一般RDMS
- RDMS \Rightarrow 特定RDMS
- 优化

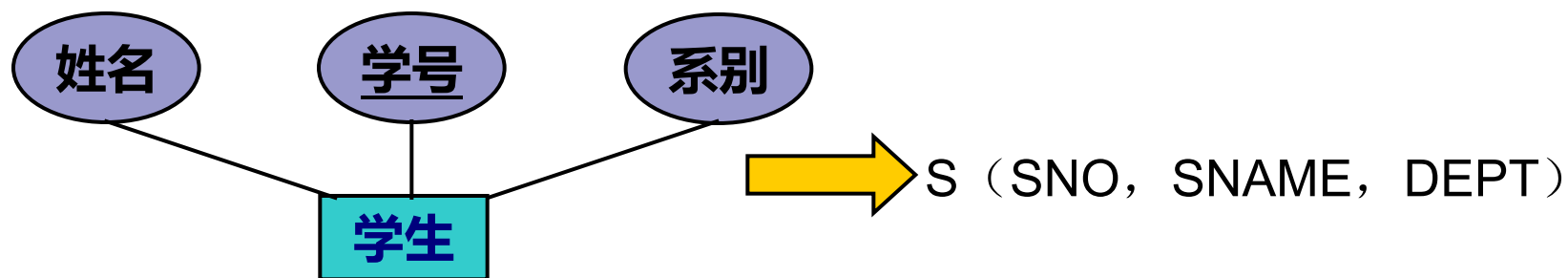
■ 7.4.3 E-R图向关系模型的转换

■ 7.4.4 设计用户子模式

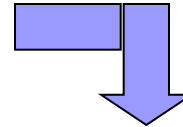
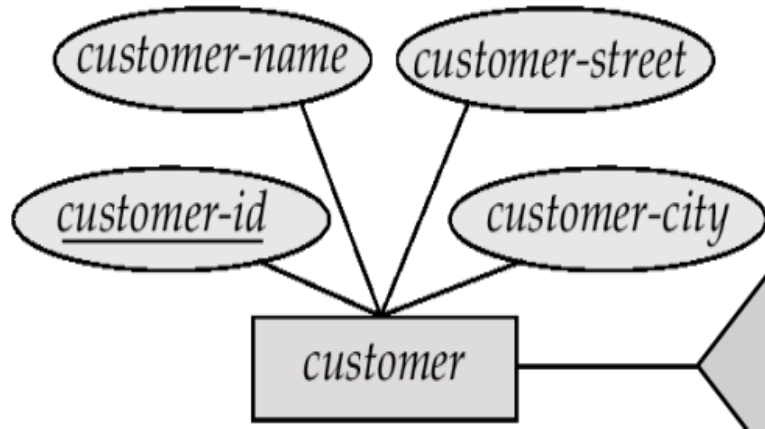
7.4.3 E-R图向关系模型的转换

■ 1. 实体和属性

- 实体 → 关系
- 属性 → 关系的属性



实体和属性

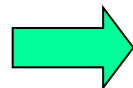
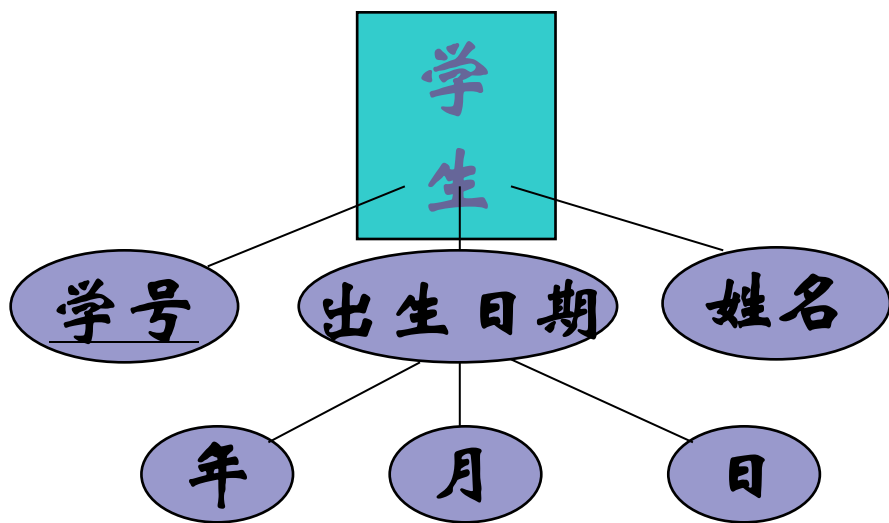


<i>customer-id</i>	<i>customer-name</i>	<i>customer-street</i>	<i>customer-city</i>
019-28-3746	Smith	North	Rye
182-73-6091	Turner	Putnam	Stamford
192-83-7465	Johnson	Alma	Palo Alto
244-66-8800	Curry	North	Rye
321-12-3123	Jones	Main	Harrison
335-57-7991	Adams	Spring	Pittsfield
336-66-9999	Lindsay	Park	Pittsfield
677-89-9011	Hayes	Main	Harrison
963-96-3963	Williams	Nassau	Princeton

7.4.3 E-R图向关系模型的转换(续)

■ 1. 实体和属性

- 复合属性 → 将每个组合属性作为复合属性所在实体的属性

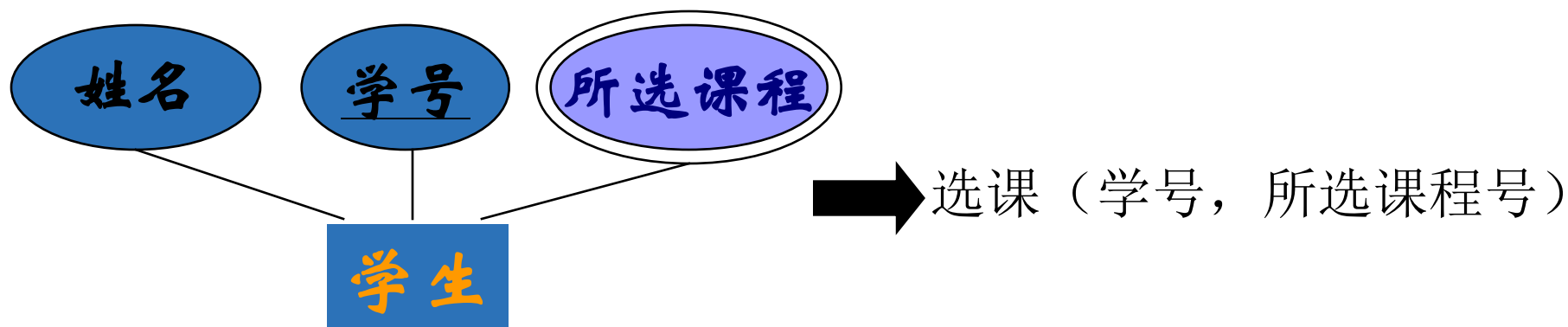


学生(学号, 姓名, 年, 月, 日)

7.4.3 E-R图向关系模型的转换(续)

■ 1. 实体和属性

□ 多值属性 → 新的关系+所在实体的码

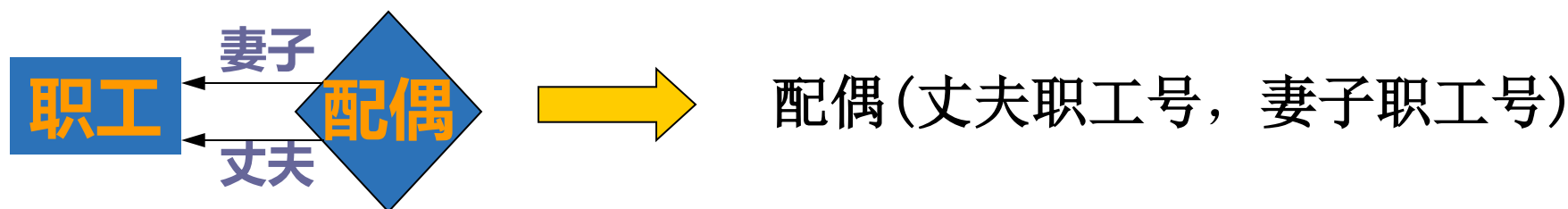


7.4.3 E-R图向关系模型的转换(续)

■ 2. 联系

□ 一对一联系（1:1）：

- 若联系双方均部分参与，则将联系定义为一个新的关系，属性为参与双方的码。

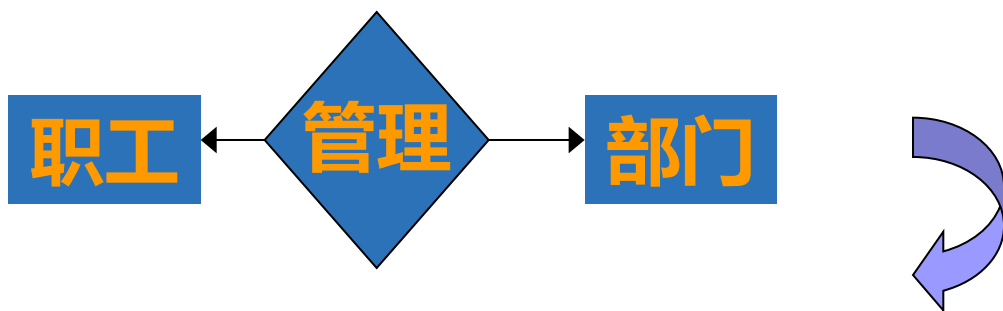


7.4.3 E-R图向关系模型的转换(续)

■ 2. 联系

□ 一对一联系（1:1）：

- 若联系一方全部参与，则将联系另一方的码作为全部参与一方的属性



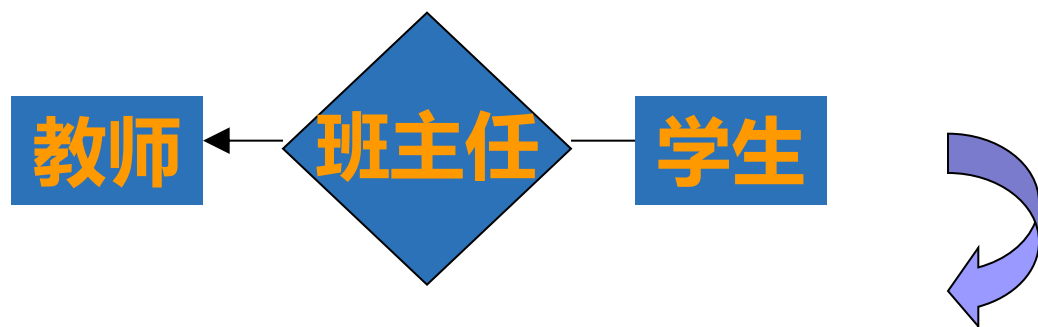
部门（部门号，部门名，**管理职工号**）

7.4.3 E-R图向关系模型的转换(续)

■ 2. 联系

□ 一对多联系（1:n）：

将单方参与实体的码作为多方参与实体的属性。



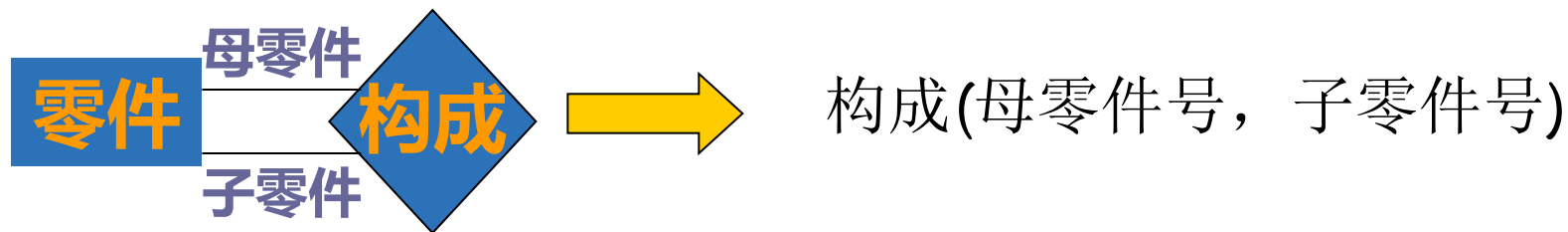
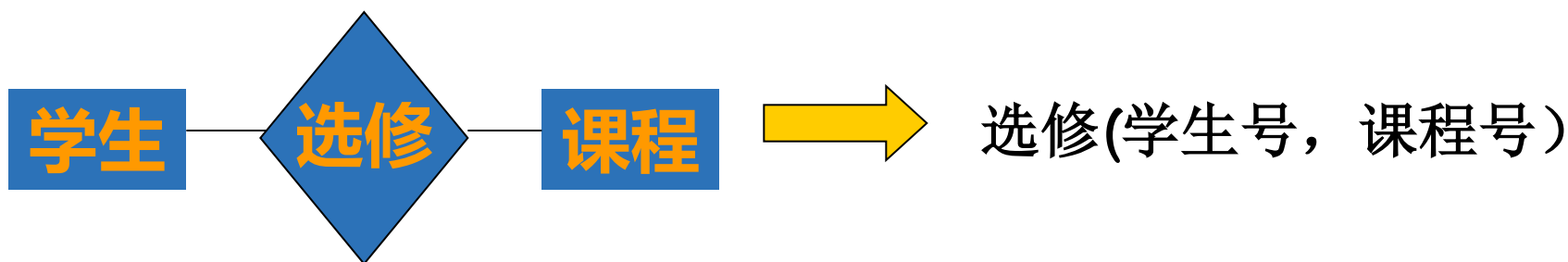
学生(学生号, 学生名, 系号, 班主任教工号)

7.4.3 E-R图向关系模型的转换(续)

■ 2. 联系

□ 多对多联系 (m:n) :

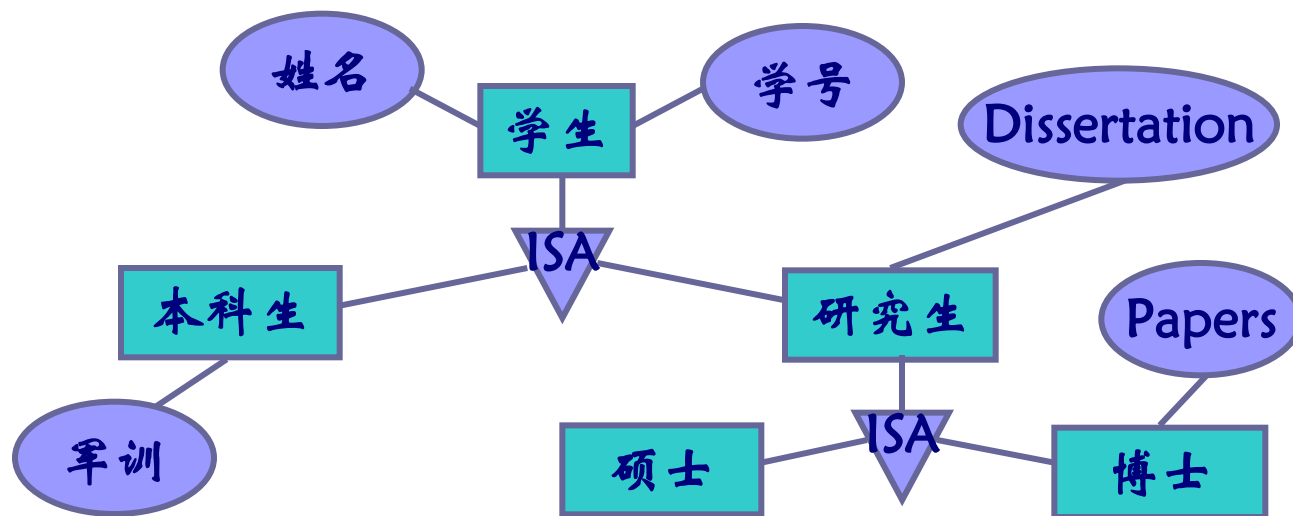
将联系定义为新的关系，属性为参与双方的码。



7.4.3 E-R图向关系模型的转换(续)

■ 3. 概括

- 高层实体集和低层实体集分别转为表
- 低层实体集所对应的关系包括高层实体集的码



学生(姓名, 学号)

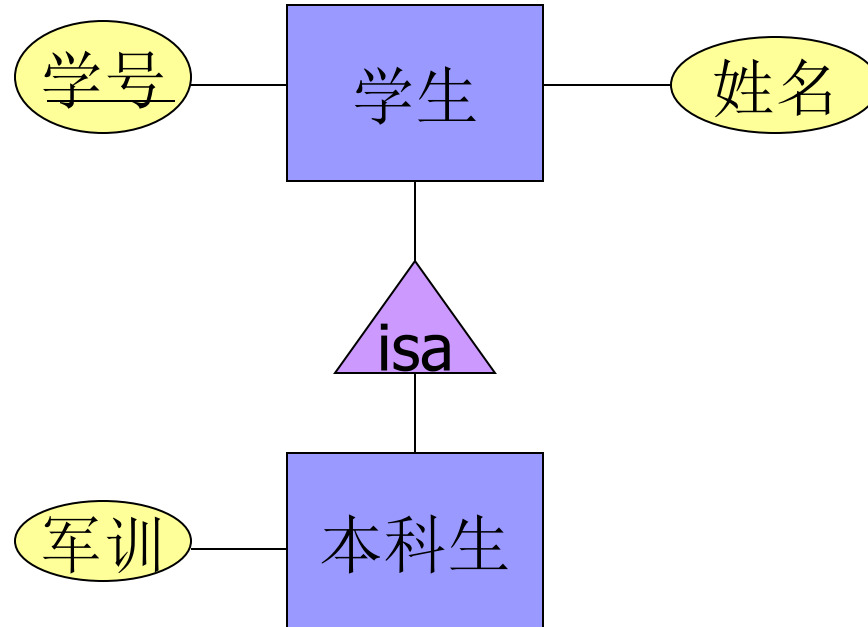
本科生(姓名, 学号, 军训)

研究生(姓名, 学号, 论文)

Subclasses: Three Approaches

1. *Object-oriented* : One relation per subset of subclasses, with all relevant attributes.
2. *Use nulls* : One relation; entities have NULL in attributes that don't belong to them.
3. *E/R style* : One relation for each subclass:
 - Key attribute(s).
 - Attributes of that subclass.

Example: Subclass -> Relations



Object-Oriented

学号	姓名
1001	张三

学生

学号	姓名	军训
1002	李四	是

本科生

Good for queries like “查找参加军训的本科生.”

E/R Style

学号	姓名
1001	张三
1002	李四

学生

学号	军训
1002	是

本科生

Good for queries like
“查找所有的学生.”

Using Nulls

学号	姓名	军训
1001	张三	NULL
1002	李四	是

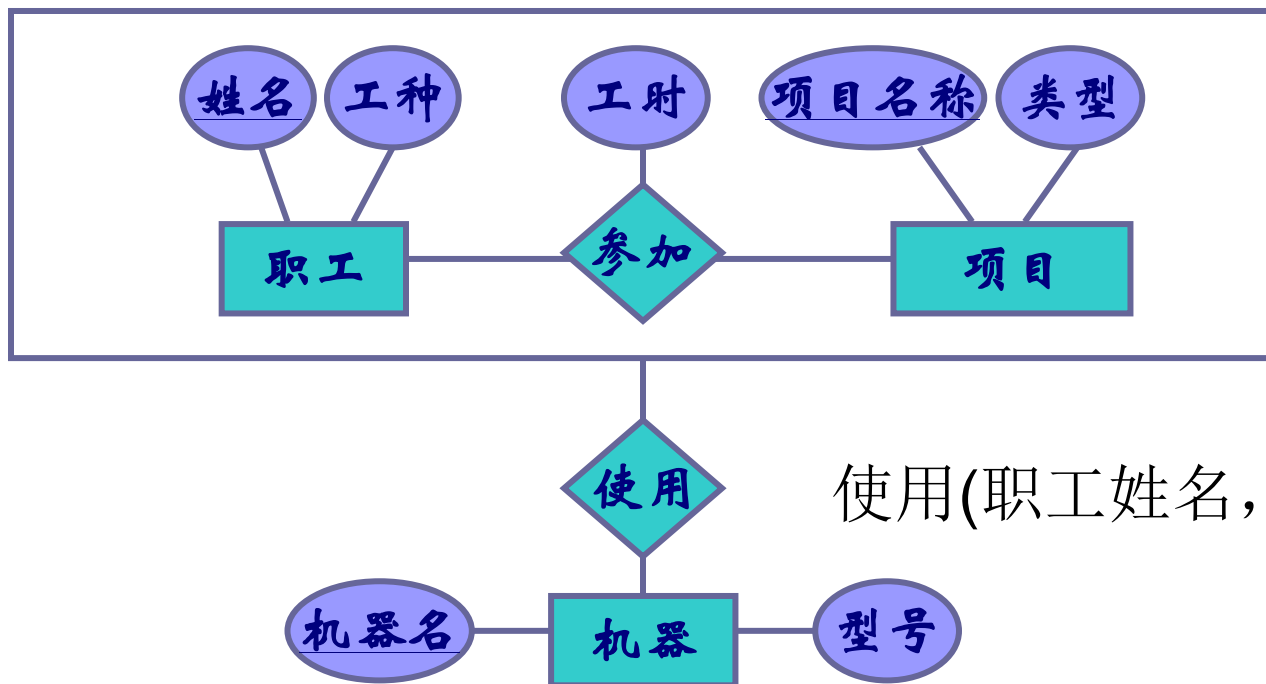
学生

Saves space unless there are *lots* of attributes that are usually NULL.

7.4.3 E-R图向关系模型的转换(续)

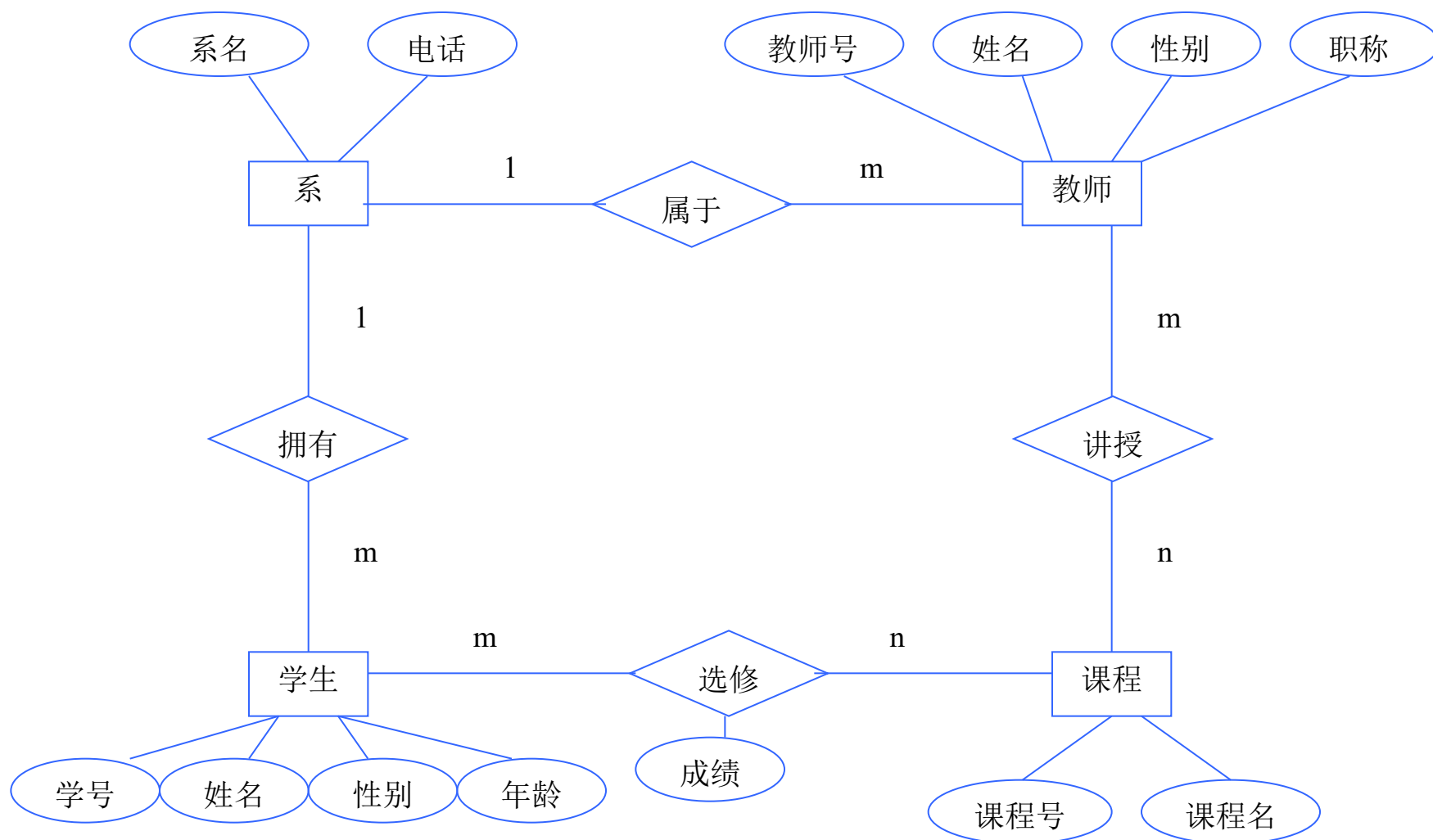
■ 4. 聚集

- 实体集A与B以及它们的联系R被看成实体集C，C与另一实体集D构成联系S，则S所对应的关系的码由R和D的码构成



使用(职工姓名，项目名称，机器名)

例：将如下E-R图转换为关系模式



转换过程

(1) 把每一个实体转换为一个关系

- 首先分析各实体的属性，从中确定其主键，然后分别用关系模式表示。例如，前页E-R图中的四个实体分别转换成四个关系模式：

- 学生（学号，姓名，性别，年龄）
- 课程（课程号，课程名）
- 教师（教师号，姓名，性别，职称）
- 系（系名，电话）

其中，有下划线者表示是主键。

转换过程(续)

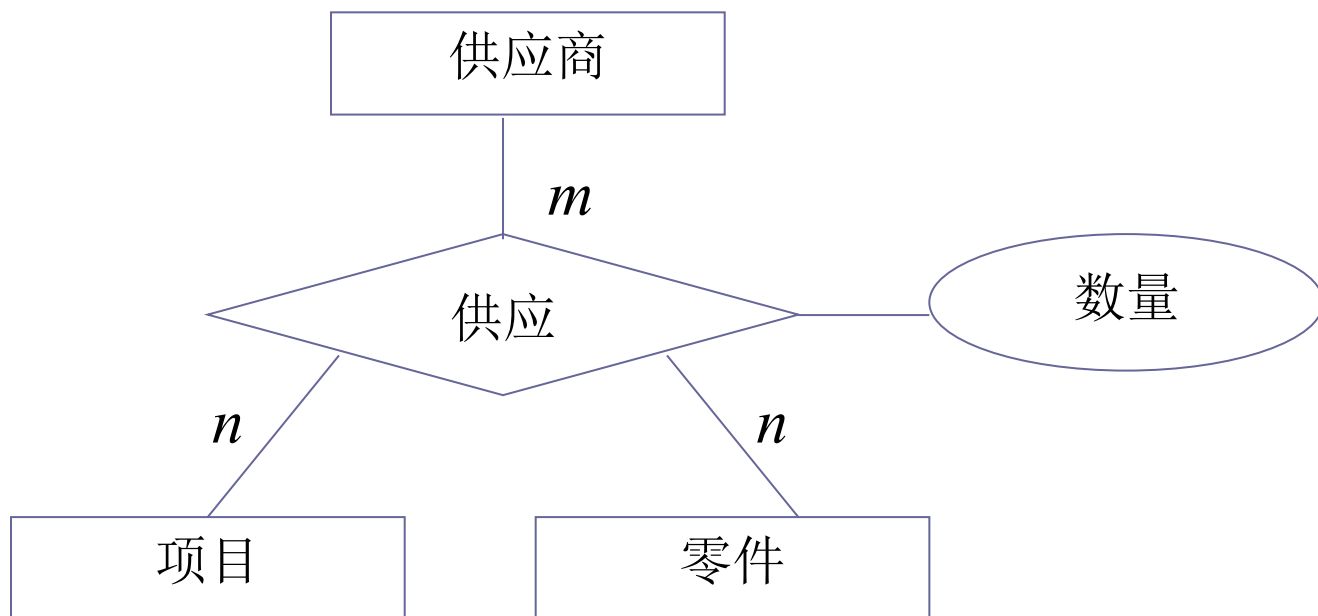
(2) 把每一个联系转换为关系模式

- 由联系转换得到的关系模式的属性集中，包含两个发生联系的实体中的主键以及联系本身的属性，其关系键的确定与联系的类型有关。
- 例如，前图四个联系也分别转换成四个关系模式：
 - 属于（教师号，系名）
 - 讲授（教师号，课程号）
 - 选修（学号，课程号，成绩）
 - 拥有（系名，学号）
- 1:1的联系和1:n的联系也可以与n端对应的关系模式合并，并在这一端加上1端关系模式的码和联系本身的属性。

转换过程(续)

(3) 特殊情况的处理

- 三个或三个以上实体间的一个多元联系在转换为一个关系模式时，与该多元联系相连的各实体的主键及联系本身的属性均转换为关系的属性，转换后所得到的关系的主键为各实体键的组合。
 - 供应（供应商号，项目号，零件号，数量）



7.5 物理数据库设计

■ 7.5.1 功能

- 确定DB存储结构
- 确定DB存取方法

■ 7.5.2 目标

- 提高速度
- 节约空间

7.5 物理数据库设计

■ 7.5.3 设计内容和方法

□ 存取方法

- 建立索引（B+树），提高速度
- 聚簇存取：减少I/O，适应单、多个关系集
- HASH方法

□ 存储结构

- 存放位置
- 系统配置（参数调整）

7.5 物理数据库设计

■ 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

7.5 物理数据库设计

■ 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

■ 聚簇的用途

- 1. 大大提高按聚簇码进行查询的效率
- 2. 节省存储空间

7.5 物理数据库设计

■ 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

7.5 物理数据库设计

■ 优化聚簇设计

- 从聚簇中删除经常进行全表扫描的关系；
- 从聚簇中删除更新操作远多于连接操作的关系；
- 不同的聚簇中可能包含相同的关系，一个关系可以在某一个聚簇中，但不能同时加入多个聚簇
 - 从这多个聚簇方案(包括不建立聚簇)中选择一个较优的，即在这个聚簇上运行各种事务的总代价最小

7.6 实施和维护

- 初始数据装载
- 应用程序编制与调试
- DB试运行
- 修改与完善
 - 由DBA完成

练习

- 学校由若干个系组成，每个系有若干名教师和学生（本科生，研究生），老师或者学生只能属于某一个特定院系；每个教师可以担任若干门课程，并参加多项科研项目，每位教授或副教授可以指导若干研究生；教师的工资由其职称决定，每位老师都拥有自己的工作证；每门课程可以由若干老师任教；每个学生（本科生，研究生）可以同时选修多门课程；每个研究生只有一位指导老师。请设计教学管理的**E-R**模型，并根据自己的理解标示实体、联系及其属性，并将得到的**E-R**图转换为关系模式。

下课了。。。

攀
登



休息一会儿。。。

*Darling,
We were meant to be !*

