

分布式文件系统及数据库技术

第七讲

主讲人：曹仔科 彭希羨
浙江大学管理学院
数据科学与工程管理系

1、数据库系统开发生命周期

2、ER模型

数据库设计概念

- 软件工程

用科学知识、工程方面的纪律指导软件开发的过程，以提高软件质量和开发效率，降低开发成本。

- 软件生存期

从软件的规划、研制、实现、投入运行后的维护，直到它被新的软件所取代而停止使用的整个期间。

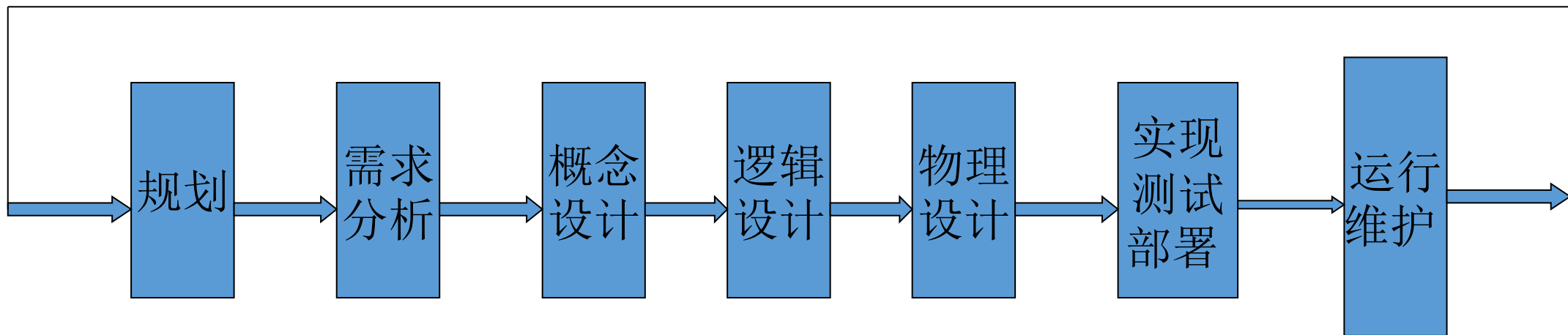
- 数据库工程

- 数据库系统生存期

数据库应用系统从开始规划、设计、实现、维护到最后被新的系统取代而停止使用的整个期间。

隶属关系

数据库系统开发生命周期



规划阶段

- **规划阶段的三个步骤**

- **系统调查：**

- 例如：对企业组织作全面的调查，画出组织层次图，以了解企业的组织结构

- **可行性分析**

- 从技术、经济、效益、法律等方面对建立数据库的可行性进行分析；写出可行性分析报告；组织专家进行讨论其可行性

- **确定数据库系统的总目标和制定项目开发计划**

- 公司的目标是什么？需要一个什么样的数据库？帮助解决什么问题？

需求分析

- 这个阶段专注于详细、精确地收集和定义系统必须做什么，特别是**数据需求**和**功能需求**。

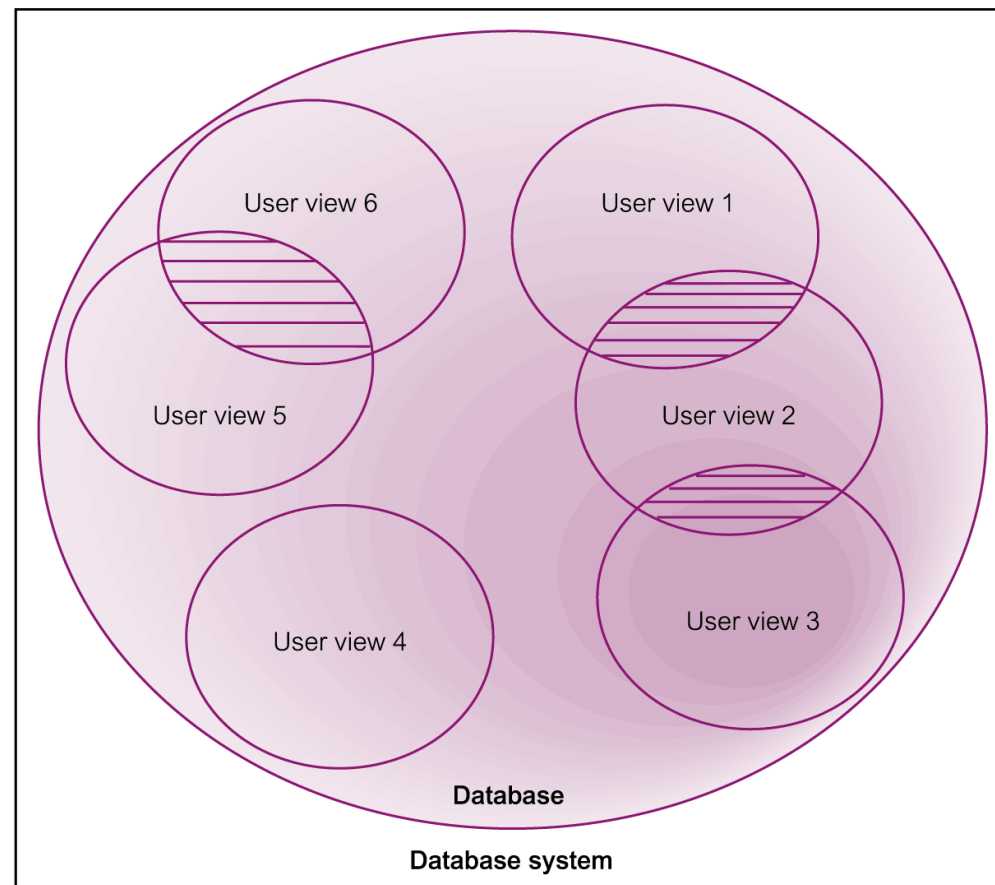
主要任务：

- **识别数据需求**： 确定系统需要存储哪些数据实体（如客户、产品、订单）以及它们的属性。
- **识别功能需求**： 确定系统需要对数据执行哪些操作（如新增订单、查询库存、更新客户信息）。
- **识别约束和规则**： 如数据完整性规则（唯一性、非空等）、业务规则和安全要求。

需求分析

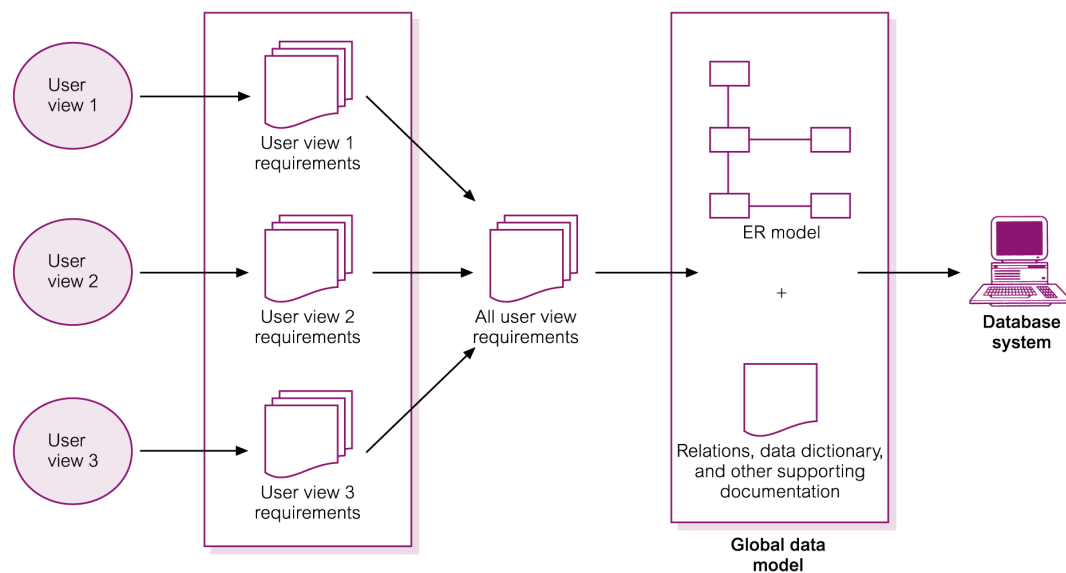
• 明确用户视图

- 从用户（角色）的角度出发，描述他们需要什么以及为什么需要。
- “作为一个【角色】，我想要【完成某个活动】，以便于【实现某个价值】。”

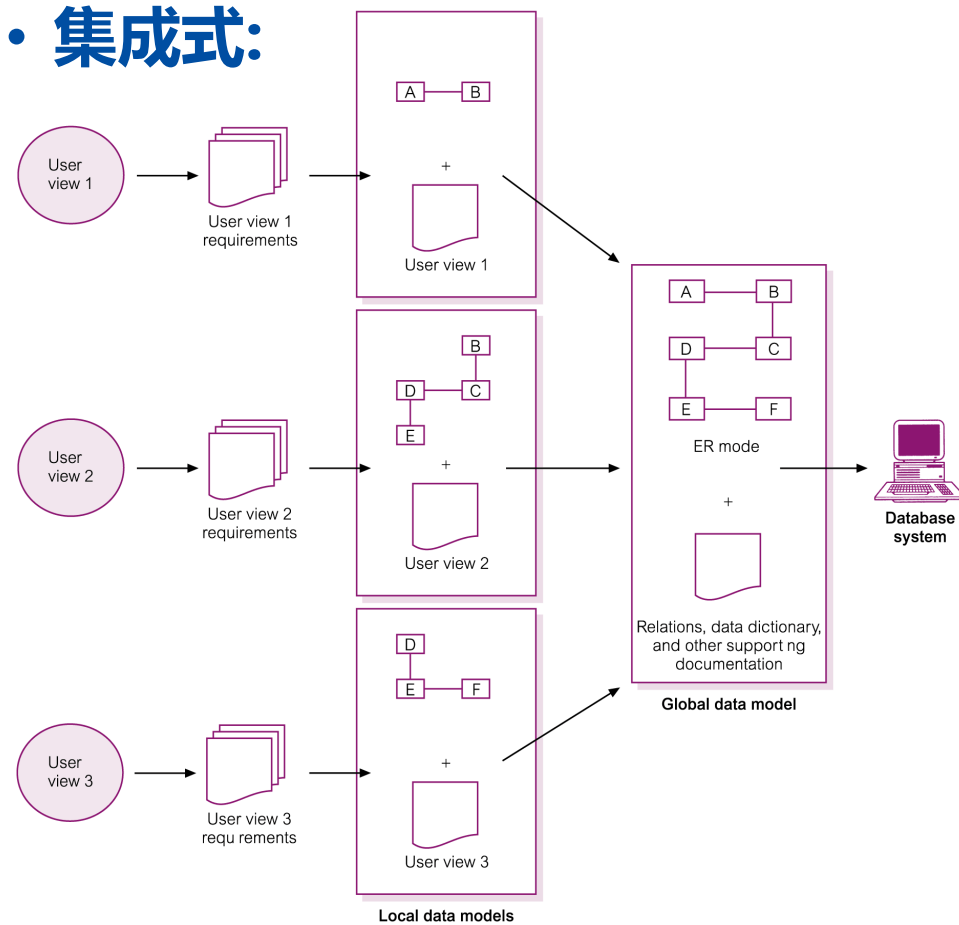


需求收集与分析

• 集中式:



• 集成式:



• 现实中常使用两种方式的结合

需求收集与分析

- 可先利用集中式将**负责人和经理**的需求合并在一起，作为分支表
 - 利用集中式将**主管、助理和客户**的需求合并在一起，作为员工表
 - 最后使用集成方法将它们合并
-
- 集中和集成方法的使用根据系统复杂度和视图重叠度而定

用户视图及视图对应使用的主要数据

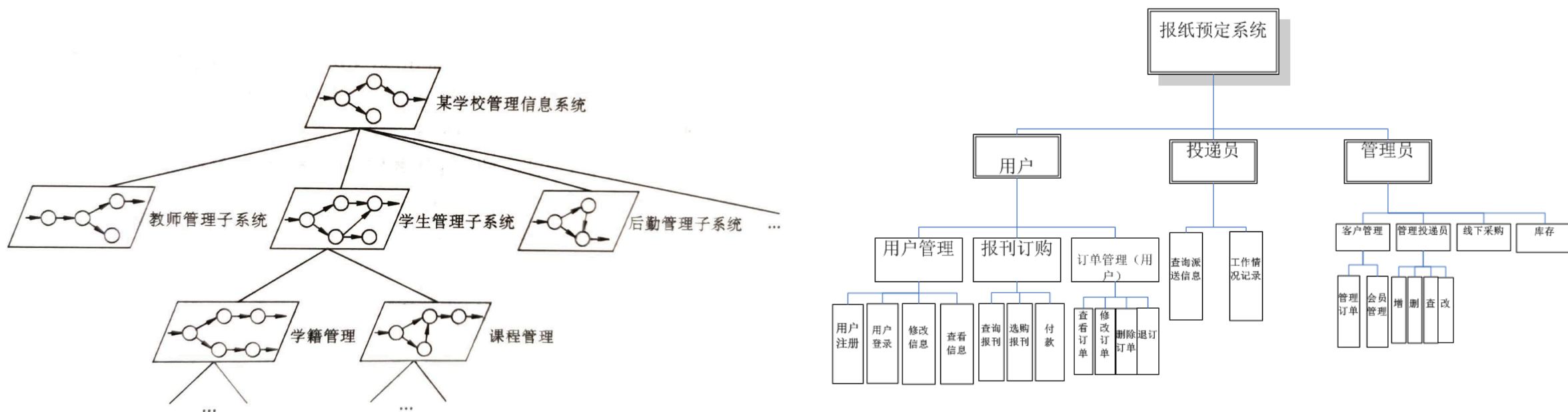
	Director	Manager	Supervisor	Assistant
branch	X	X		
staff	X	X	X	
property for rent	X	X	X	X
owner	X	X	X	X
client	X	X	X	X
property viewing			X	X
lease	X	X	X	X
newspaper	X	X		

需求分析

- 确定系统范围和组成，产生**系统模块图**。
 - 分析用户活动过程，产生**业务流程图**。
 - 分析用户活动涉及的数据，产生**数据流程图**。
 - 如使用同一建模语言（UML），可以绘制**用例图**，从系统外部视角，图形化地展示系统的功能单元（用例）、与系统交互的外部参与者（角色）以及它们之间的关系。
 - 分析系统数据，产生**数据字典**。
-
- 无论使用什么方法，其最终目的都是相同的：**清晰地捕获功能需求，从而推导出准确、完整的数据库数据需求与业务规则，为后续的数据库概念设计和逻辑设计奠定坚实的基础。**

系统流程图（最高层）

- 系统流程图是在系统分析员在做系统构架阶段对未来构建的信息处理系统的一种描述
- 这种描述是相对简单但完备，涉及到未来系统中使用的处理部件/模块



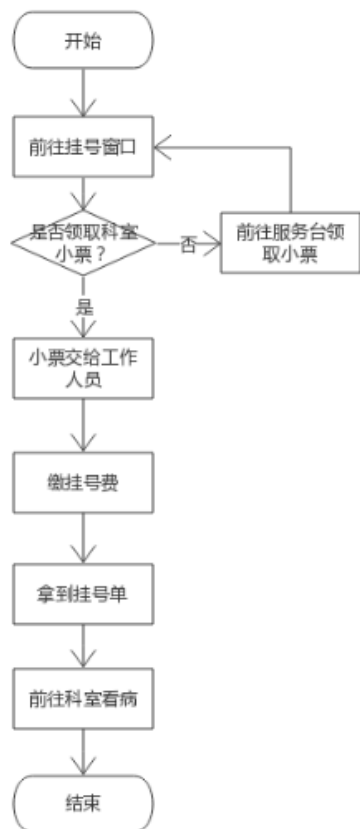
业务流程图

- 业务流程图是一种描述系统内各单位、人员之间业务关系、作业顺序和管理信息流向的图表
- 业务流程图重在描述业务走向
 - 如一般校医院看病流程：首先去挂号，然后到相应医生那里看病开药，然后再到药房领药，最后离开
- 业务流程图描述的是完整的业务流程，以业务处理过程为中心，一般没有数据的概念

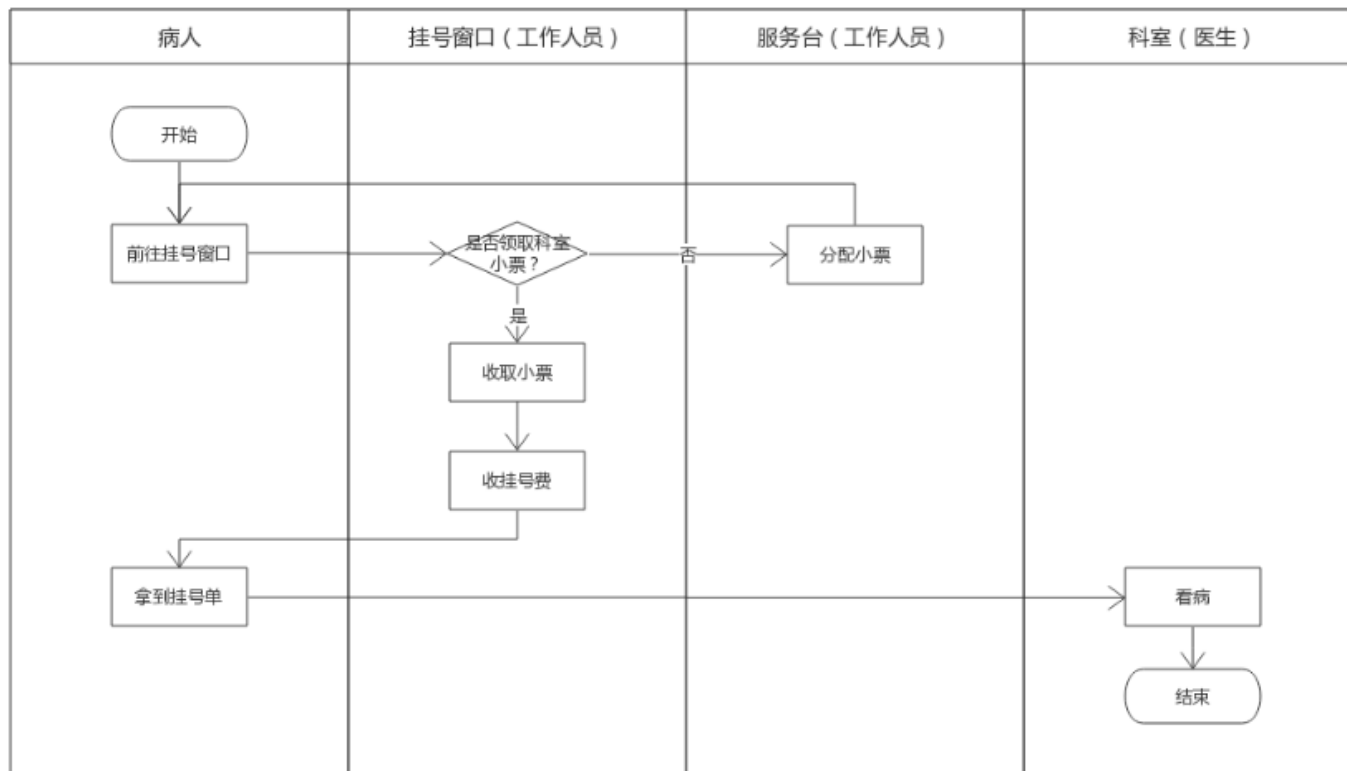
基本流程图

业务流程描述：去医院看病，你需要先去服务台领个具体要去看病的某个科室的小票（假设必须要领），再前往挂号窗口将小票递给工作人员，缴完挂号费之后拿到挂号单，再前往具体科室去看病。

基本流程图



跨职能流程图



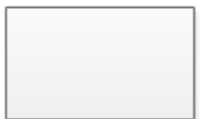
业务流程图常用符号

管理业务流程图常用符号



开始或结束

流程的开始或结束，流程图只有一个开始，可有多个结束。



流程

要执行的处理步骤。如果是画页面的框架图，该矩形框可以用来指代一个页面，当流程图中有指代页面又有指代流程的，可将这两类矩形框做色彩区别，再做下注释即可。



判断

判定条件（If...Then...Else），有一个输入，两个输出



流程线

表示流程执行的方向与顺序



文档

以文件的方式输入/输出。表达为一个文件，可以是生成的文件，或者是调用的文件，文件的题目或说明写在符号内。



数据

表示数据的输入/输出



页面内引用

同一个流程图中从一个进程到另一个进程的交叉引用。圈内有一个字母或数字，在相互联系的流程图内，连接符号使用同样的字母或数字，以表示各个过程是如何连接的。



注释/说明

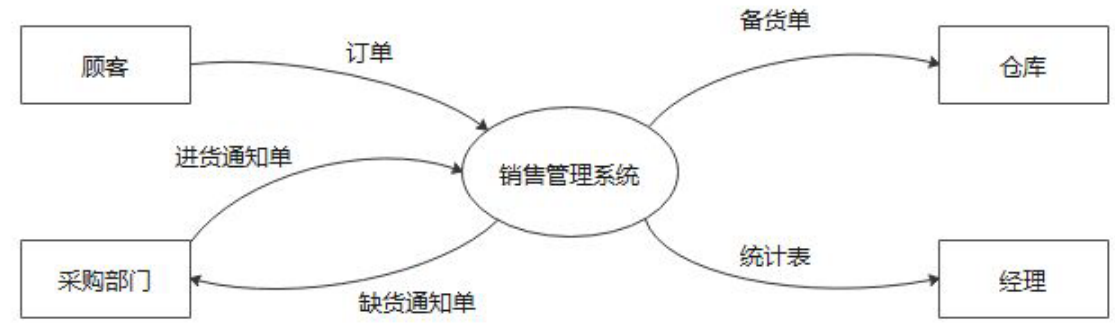
一般流程到一个位置，做一段执行说明，或者特殊行为时，会用到它。

数据流程图

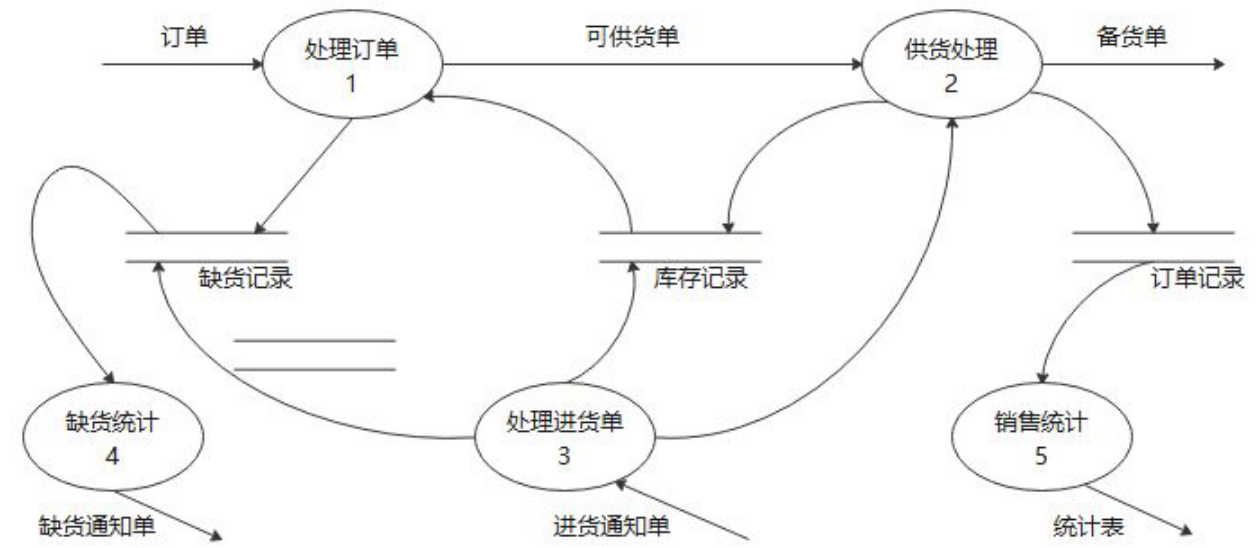
- 数据流程图综合地反映出信息（数据）在系统中的流动、处理和存储情况
- 数据流程图则是描述数据的走向
 - 继续以校医院看病为例：数据流程图主要刻画病人挂号系统需要哪些表，数据怎么存，怎么更新，怎么检查等
- 数据流图描述的是各个子块之间如何进行数据传递，以处理数据流、数据存储为核心

数据流程图-示例

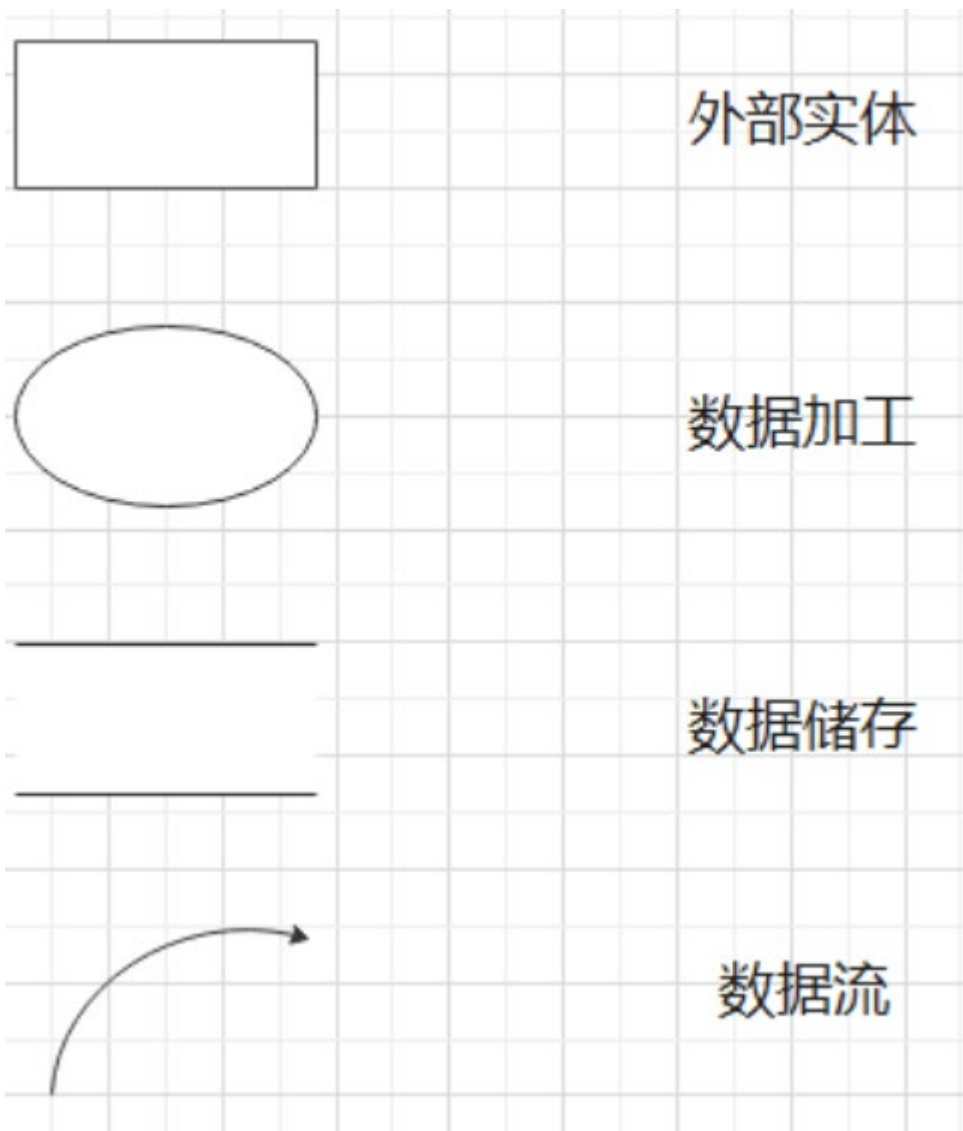
顶层图



0层图



数据流程图图常用符号



数据字典

- 数据字典主要包括：
 - 数据项
 - 数据结构
- 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
- 在数据字典中可以通过对**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。

数据字典

- **数据项**是不可再分的数据单位

- 对数据项的描述

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}

- 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

数据字典

- 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。

- 对数据结构的描述

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 } }

数据字典

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： 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

- 数据字典的作用在于方便其他用户对于数据库数据的理解，也可以用表格形式组织。

设计阶段

这是整个生命周期中最关键的技术阶段，它将需求转化为一个详细的、可实现的数据库蓝图。通常分为三个子阶段：

(1) 概念设计：

- **目标：** 创建一个高层次、技术中立的的数据模型，专注于数据本身及其关系，而不考虑具体的DBMS实现。
- 使用**实体-关系模型** (E-R model) 。
- 产出**E-R图**，展示了实体、实体的属性以及实体之间的关系 (1:1, 1:N, M:N) 。

设计阶段

(2) 逻辑设计

- **目标**：将概念模型映射到所选数据模型（最常用的是**关系模型**）的结构上。
- 将E-R图转换为**关系模式**（即一系列的表结构）。
- 应用**规范化理论**（通常是第三范式3NF或巴斯-科德范式BCNF）来消除数据冗余和插入/更新/删除异常，确保数据完整性。
- 定义**主键、外键**。
- 产出完整的数据库逻辑模式（表结构清单），包括每个表的字段名、数据类型、主外键关系。

设计阶段

(3) 物理设计:

- **目标**: 为特定的 **数据库管理系统** (如MySQL, Oracle, SQL Server) 设计底层的存储结构和访问机制, 以优化性能。
- 决定文件的存储位置、分区策略。
- 设计**索引** (哪些列需要创建索引, 是什么类型的索引) 以加速查询。
- 确定磁盘存储结构和访问方法。

总结

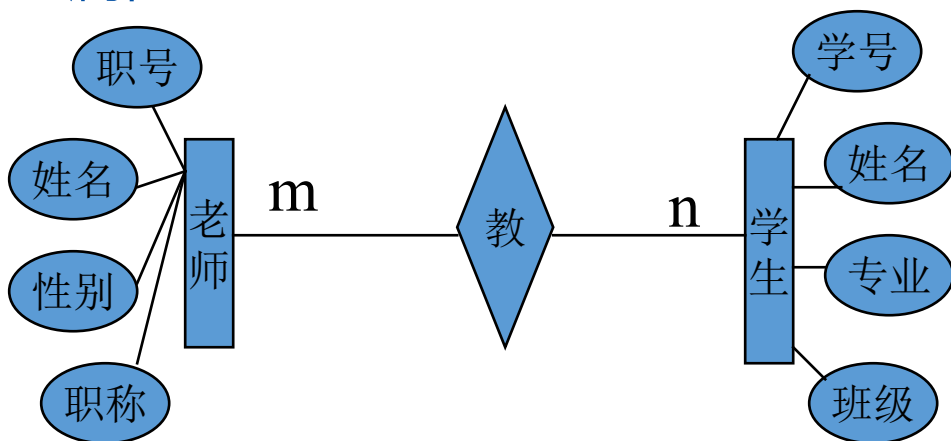
阶段	核心问题	关键产出
规划	为什么要做？是否可行？	项目计划、可行性报告
需求分析	系统需要做什么？	需求规格说明书、数据字典
数据库设计	数据如何组织？	E-R图、规范化后的表结构、DDL脚本
实现	如何构建它？	创建好的数据库、应用程序
测试	它是否正常工作？	测试报告、性能分析
部署	如何交付给用户？	上线的系统、培训材料
运维	如何长期支持？	系统优化、备份恢复记录

基础ER 模型

- ER模型：概念设计阶段的主要描述工具，起到至关重要的“承上启下”的过渡作用。用直观的图形化方式描述现实世界中的**数据**及其**关联**，而不涉及任何具体的技术实现。

- ER模型的基本元素

- 实体
- 联系
- 属性

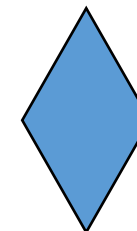


ER模型图例

实体



联系



属性



ER图的三个基本元素

1. 实体

- **定义：** 现实世界中可区别于其他对象的“事物”或“对象”。
例如：学生、课程、订单、产品。
- **表示：** 用**矩形**表示。
- **强实体：** 不依赖于其他实体而存在，拥有自己的主键。如学生。
- **弱实体：** 其存在依赖于另一个实体（称为标识实体）。如家属依赖于员工而存在。用**双线矩形**表示。

ER图的三个基本元素

2.属性

- **定义**：实体的特征或性质。例如：学生实体有学号、姓名、年龄等属性。
- **表示**：用**椭圆**表示，并用无向边连接到其所属的实体。
- **键属性/主键**：能唯一标识实体中每一个实例的属性（如 学号）。在图形中，属性名下加**下划线**。

ER图的三个基本元素

3. 联系

- **定义：** 实体之间的关联。例如：学生 **选修** 课程。
- **表示：** 用**菱形**表示，菱形内写明联系名，并用无向边与相关的实体连接。
- **联系的度数（参与实体的个数）：**
 - **一元联系/递归联系：** 同一实体集内的实例之间的联系。如 员工实体内部的 领导联系（一个员工领导多个员工，一个员工被一个领导管理）。
 - **二元联系：** 两个实体集之间的联系。这是最常见的联系，如 学生和 课程之间的 选修联系。
 - **多元联系：** 三个或以上实体集之间的联系。如 供应商、项目、零件三个实体之间的 供应联系。

联系的基数约束

- 基数约束定义了一个实体通过联系能与另一个实体的多少个实例相关联。这是ER图设计的重中之重。
- **一对一联系 (1:1) :**
 - 实体集A中的一个实例最多与实体集B中的一个实例相关联，反之亦然。
 - **示例：公司和 CEO**（假设一个公司只有一个CEO，一个CEO只能领导一个公司）。
 - 在联系的两端都写上1。

联系的基数约束

➤ 一对多联系 (1:N)

- 实体集A中的一个实例可以与实体集B中的任意多个实例相关联，而B中的一个实例最多与A中的一个实例相关联。
- 示例：部门和 员工（一个部门有多个员工，一个员工只属于一个部门）。
- 在“一”端写 1，在“多”端写 N或 M。

➤ 多对多联系 (M:N)

- 实体集A中的一个实例可以与实体集B中的任意多个实例相关联，反之亦然。
- 示例：学生和 课程（一个学生可以选多门课，一门课可以被多个学生选）。
- 在联系的两端都写上 N或 M。

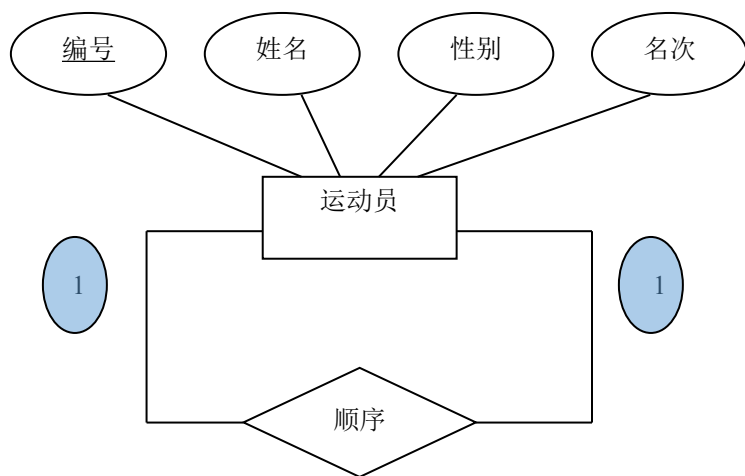
建立ER图的步骤

设计E-R图的步骤:

- 1. 识别实体：** 从需求中找出核心对象，例如：学生、课程、教师、学院。
- 2. 识别实体的属性并确定主键。**
- 3. 识别实体之间的联系并确定基数约束。**
- 4. 绘制ER图：** 将上述分析结果用图形元素组合起来。

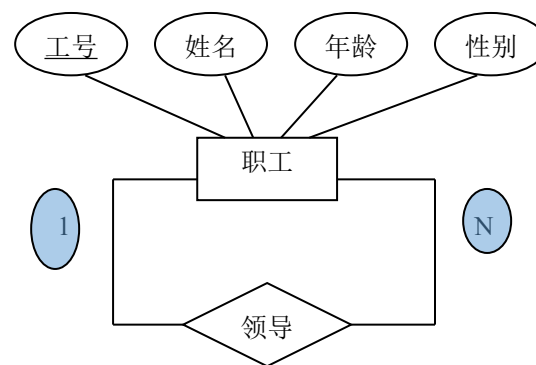
例子

运动员根据其得分来排定名次。在名次排列中，排在他前面只有一个人排在他后面也只有一个人



一元联系中的1:1联系

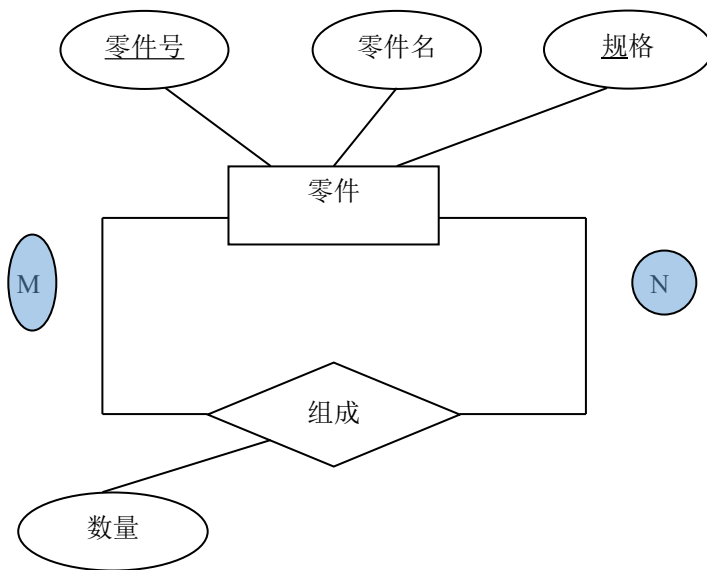
职工之间的上下级联系



一元联系中的1:N联系

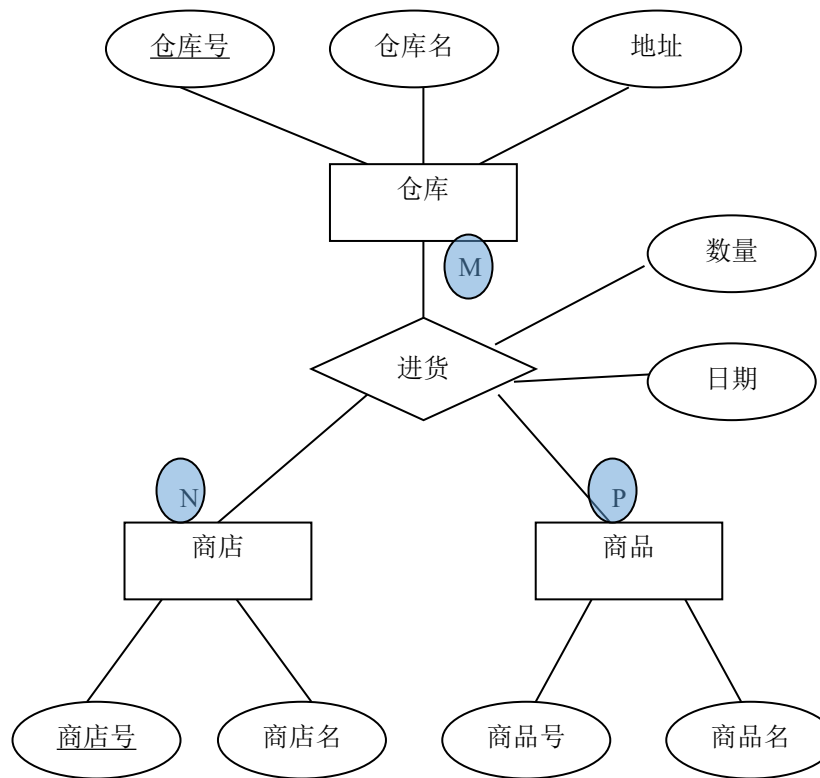
例子

工厂的零件之间存在着组合关系，一种零件由许多种子零件组成，而一种零件也可以是其他零件的子零件



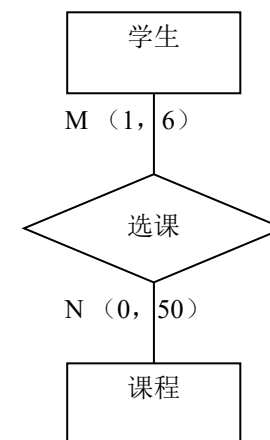
一元联系中的M:N联系

某商业集团中，商店、仓库、商品之间的进货联系



三元联系中的M:N:P联系

学校里规定每学期学生至少选修1门课程，最多选修6门课程；每门课程至多有50人选修，最少可以没人选修



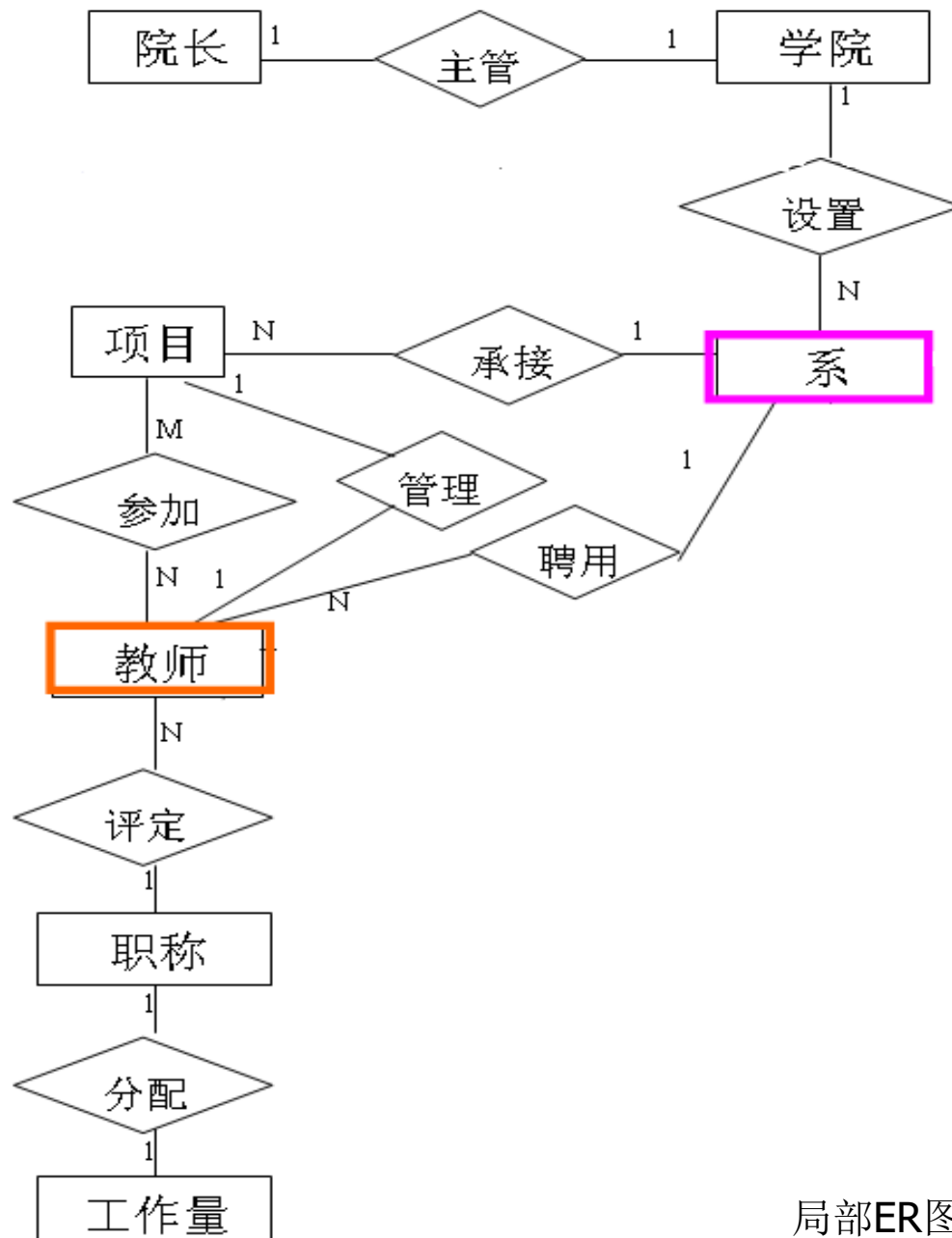
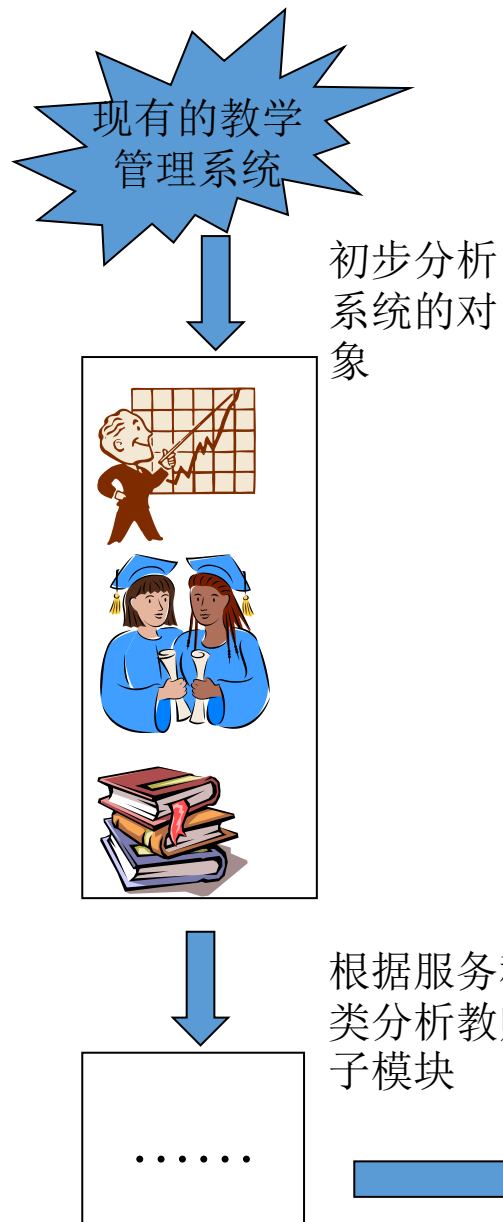
联系的连通词和实体的基数

局部vs全局ER图

设计一个庞大系统的数据库时，如果试图一次性画出所有实体和联系，会变得无比复杂，容易出错。因此，数据库设计师采用“**分而治之**”的策略：

- 首先，将整个系统按功能或业务流程划分为多个较小的、易于管理的子系统。
- 然后，为每个子系统独立设计一个ER图，这就是**局部ER图**。
- 最后，将所有局部ER图合并、集成，形成一个统一的、完整的ER图，这就是**全局ER图**。

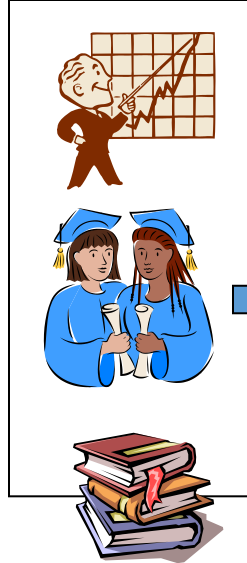
局部模式



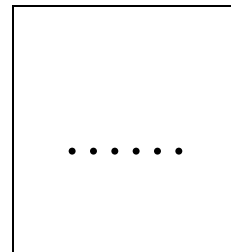
其他局部模式

现有的教学
管理系统

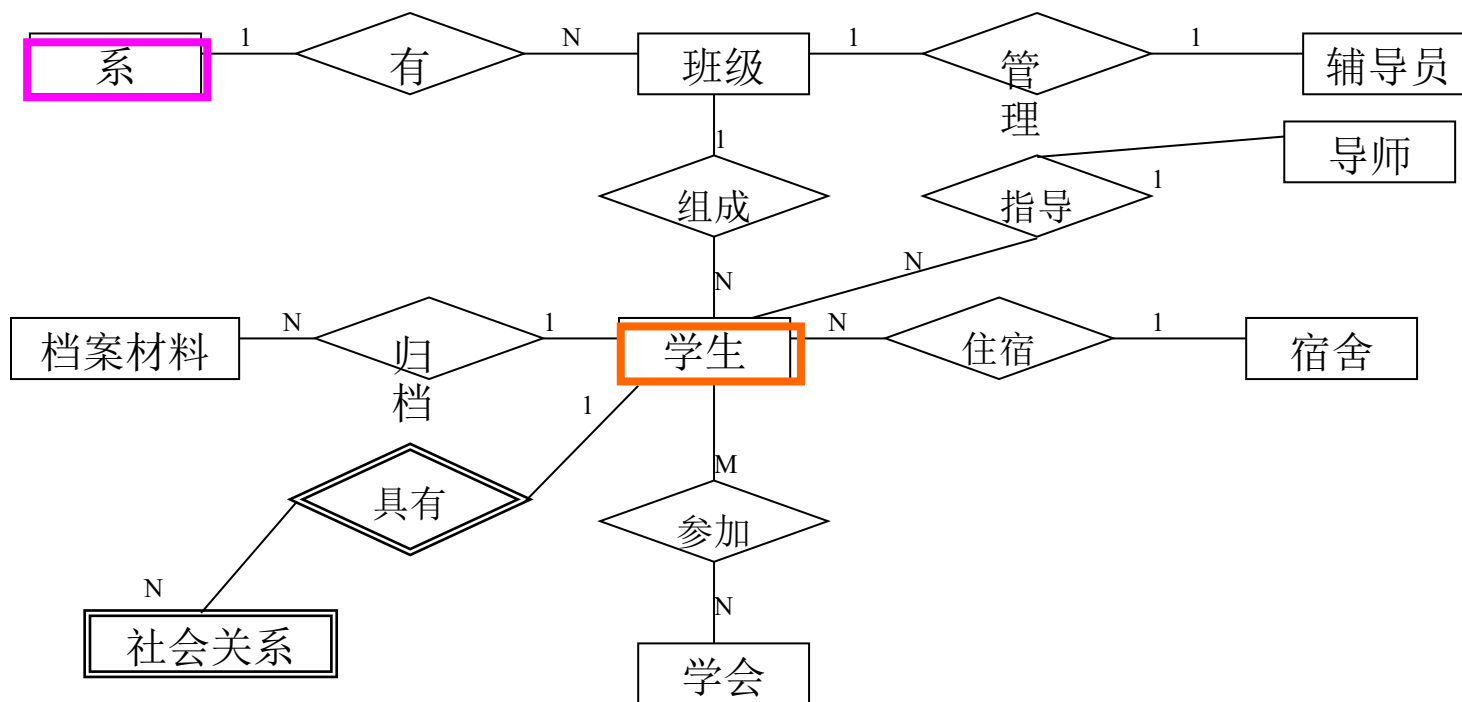
初步分析
系统的对
象



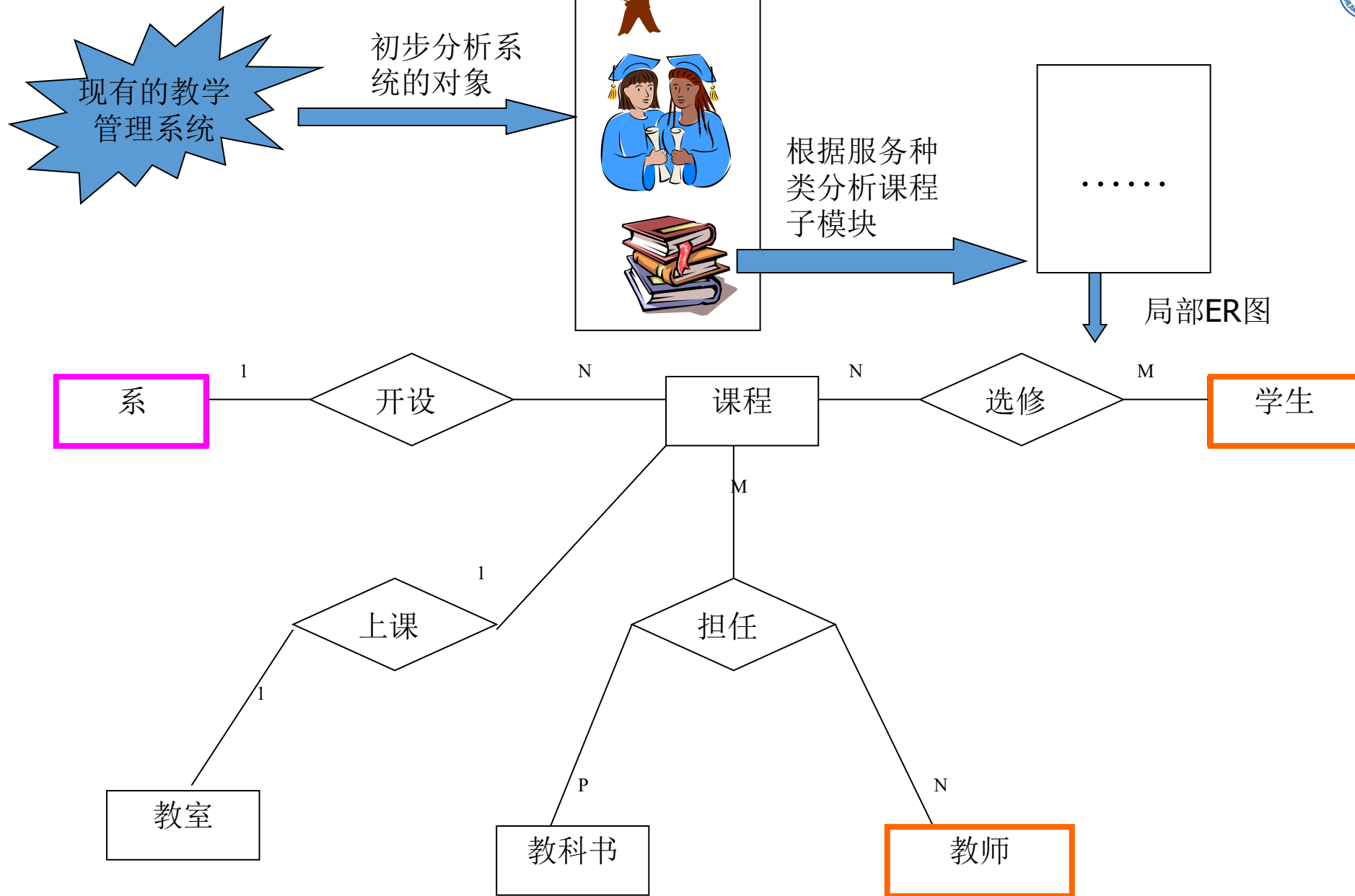
根据服务种
类分析学生
子模块



局部ER图



其他局部模式



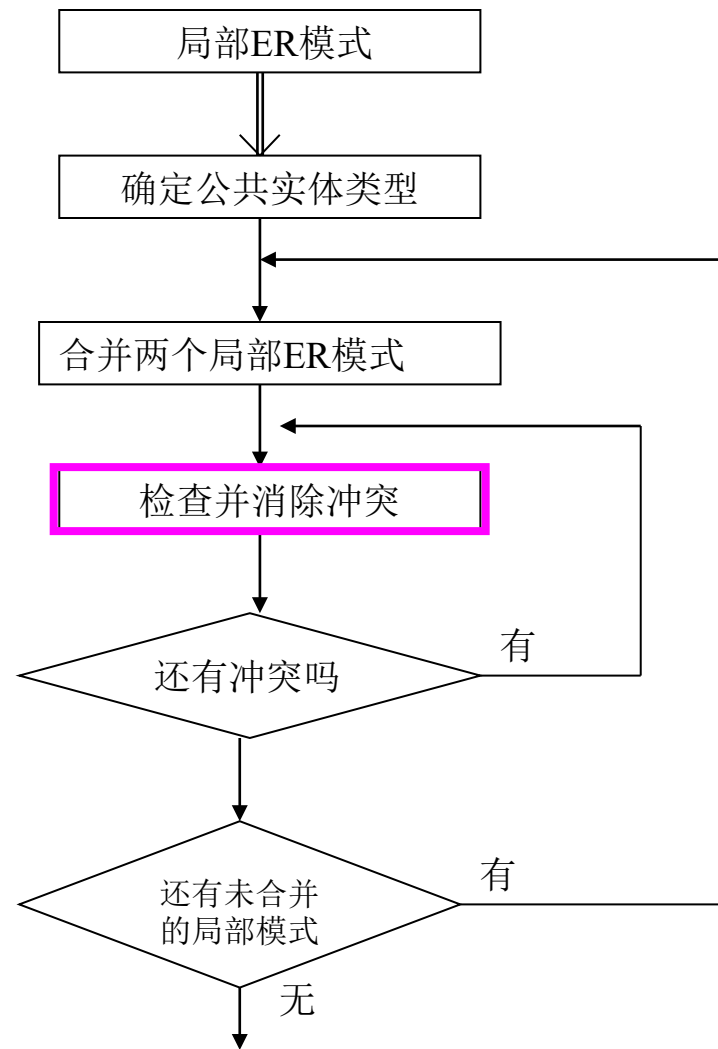
设计全局ER图

常见冲突：

1. **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。通过统一名字解决。

2. **结构冲突**：同一对象在不同应用中抽象为不同类型的组件。如“地址”可能在用户实体中为属性，但在复杂订单逻辑中，可能为独立实体。一般以支持复杂业务需求为主。

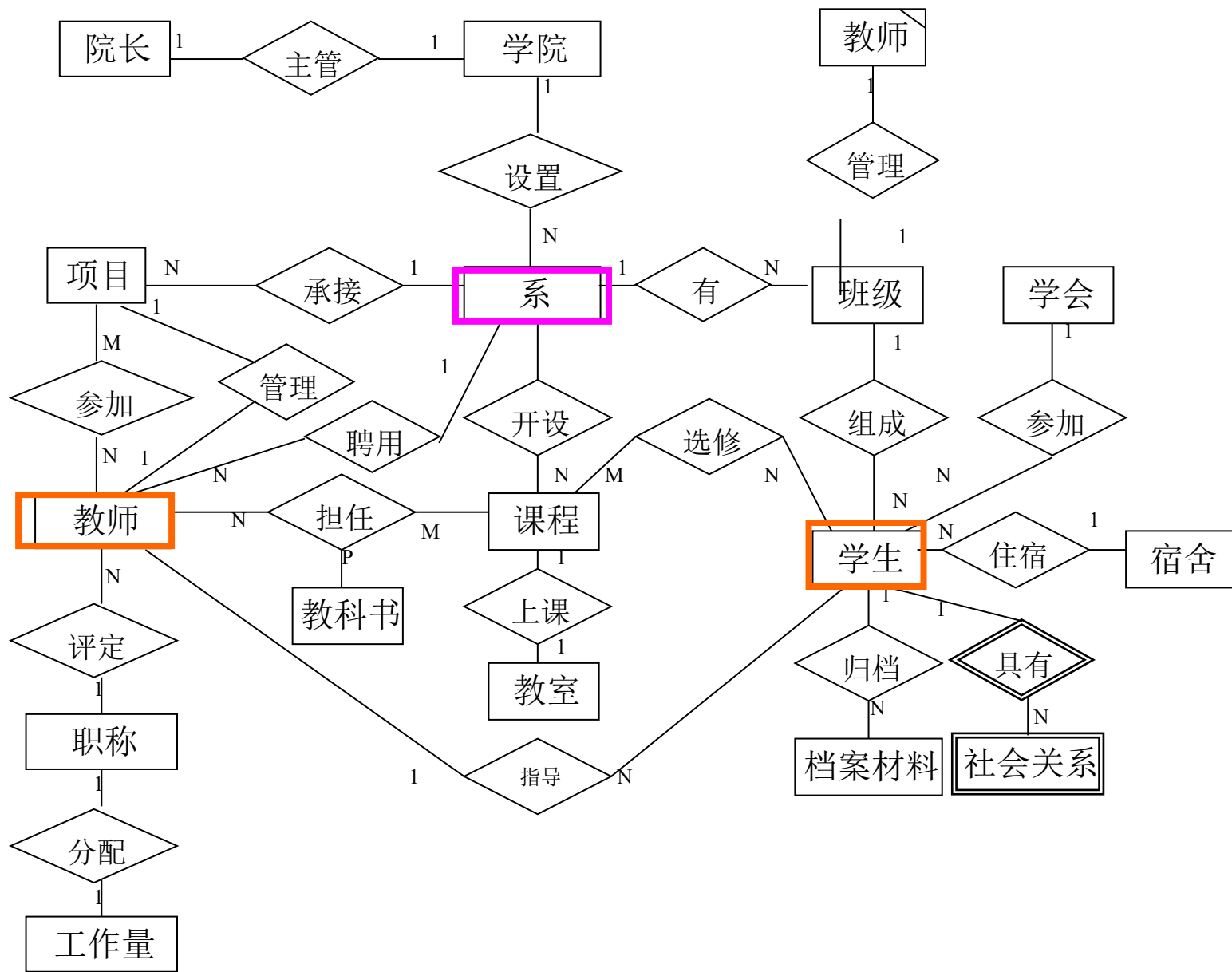
3. **联系冲突**：同一实体间的联系在不同局部ER图中表现出不同的类型（基数）。视情况可保留两个联系。



全局ER模式的优化

- 冗余属性/数据项的消除
- 冗余联系的消除
- 冗余实体类型的合并
 - 具有相同主键的实体类型

例子：三个局部ER图合并成一个ER图



ER图转换成关系模式

(1) 主体类型的转换规则:

- 每一个实体类型都需要转换成一个关系模式，实体的属性即为关系模式的属性，实体标识符即为关系模式的主键

(2) 联系类型的转换规则:

- 二元联系类型的转换
 - 若实体间联系是1:1，可以在两个实体类型转换成的两个关系模式中任意一个关系模式的属性中加入另一个关系模式的主键和联系类型的属性，一般无需单独创建关系。
 - 若实体间联系是1:N，则在N端实体类型转换成的关系模式中加入1端实体类型的主键和联系类型的属性，无需单独创建关系。
 - 若实体间联系是M:N，则需要将此联系类型也转换成关系模式，其属性为两端实体类型的主键加上联系类型的属性，而转换的关系模式的主键为两端实体键的组合。

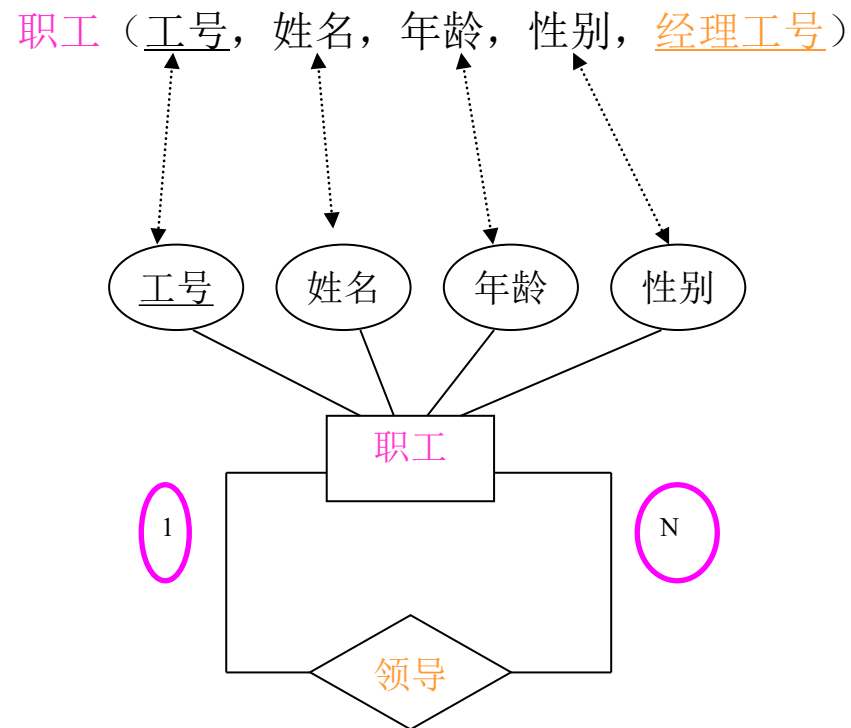
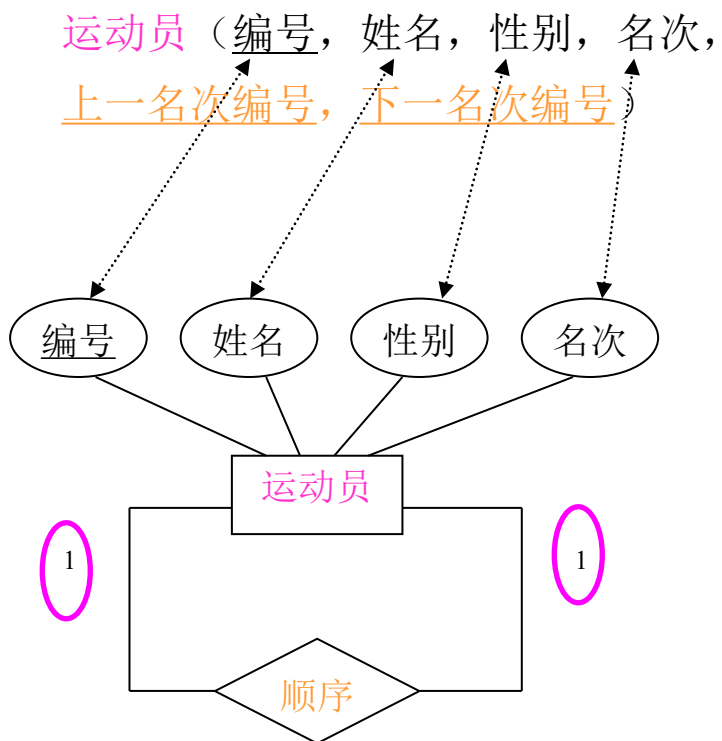
二元联系核心转换规则总结

ER图组件	转换规则	示例
强实体	为每个强实体创建一个关系（表）。实体的属性转换为表的属性（列）。实体的主键转换为表的主键。	学生实体 -> 学生表（学号，姓名，性别，出生日期）
弱实体	为每个弱实体创建一个关系。属性包括弱实体自身的属性 + 所依赖的强实体的主键（作为外键）。表的主键是（强实体的主键 + 弱实体的部分键）。	家属（弱实体，依赖于员工） -> 家属表（员工工号，家属姓名，关系）
1:1 联系	方案1（推荐）：将联系合并到任意一方的实体对应的表中。通常选择参与度更完全（非空）的一方，或在查询频率高的一方增加外键。 方案2：单独创建一个关系，包含两个实体的主键（作为外键）和联系本身的属性。	公司(1) ---- (1)CEO -> 公司表（公司编号，公司名，CEO工号） 或 CEO表（CEO工号，姓名，公司编号）
1:N 联系	在“N”端（多方）的表中，加入“1”端（一方）的主键作为外键，并加上联系本身的属性（如果有）。	部门(1) ---- (N)员工 -> 员工表（工号，姓名，...，部门编号）
M:N 联系	必须创建一个新的关系（连接表/关联表）。新表的主键由两端实体的主键组合而成。新表的属性包括联系本身的属性。	学生(M) ---- (N)课程，联系有属性成绩 -> 选课表（学号，课程号，成绩） 主键：（学号，课程号）

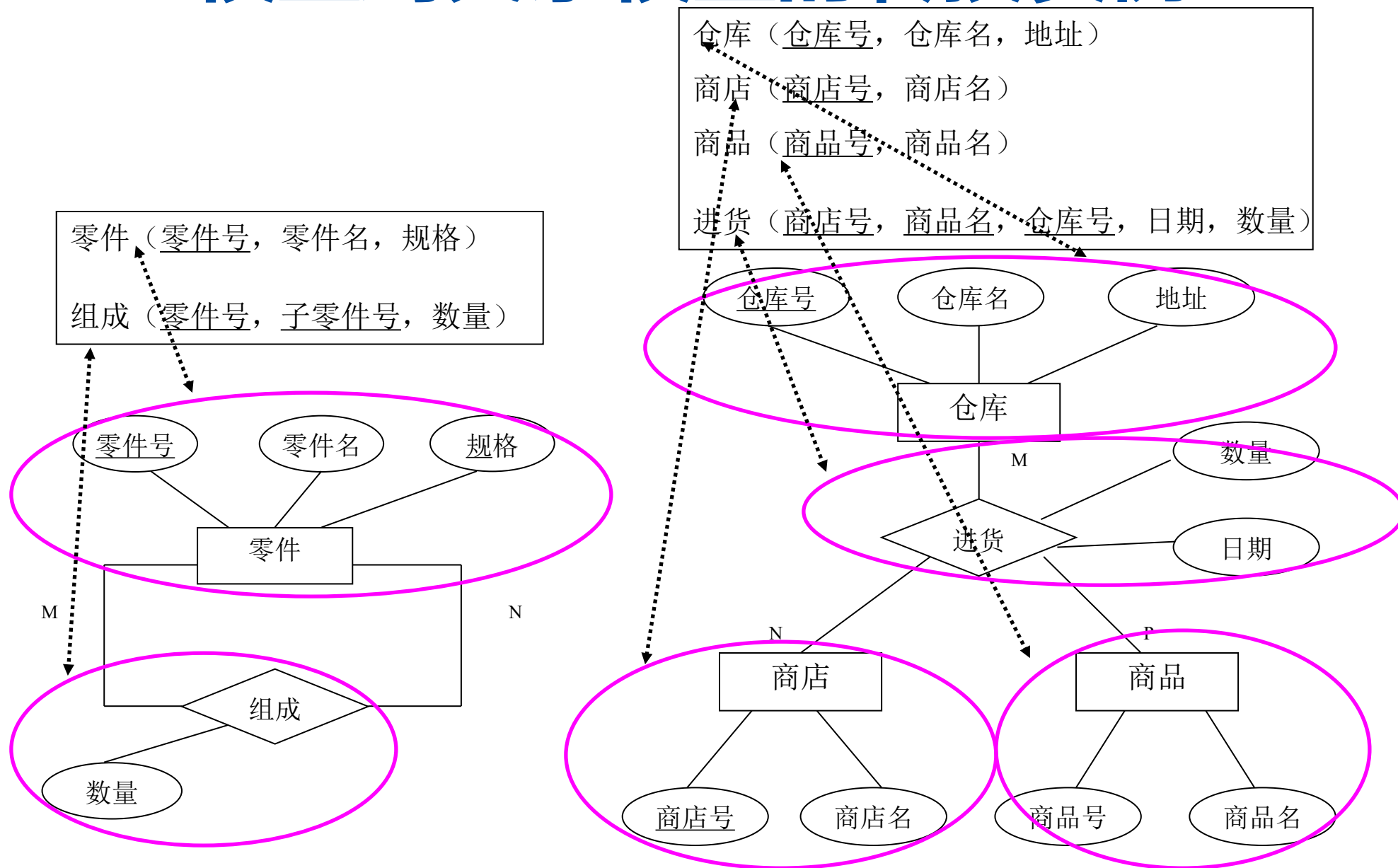
ER图转换成关系模式

- 一元联系类型的转换：同上述二元联系类型转换规则
- 三元（及以上）联系类型的转换
 - 永远都需要将大于或等于三元联系类型转换成关系模式，其属性为各端实体类型的主键加上联系类型的属性，而转换而来的关系模式的主键为各端实体键的组合。

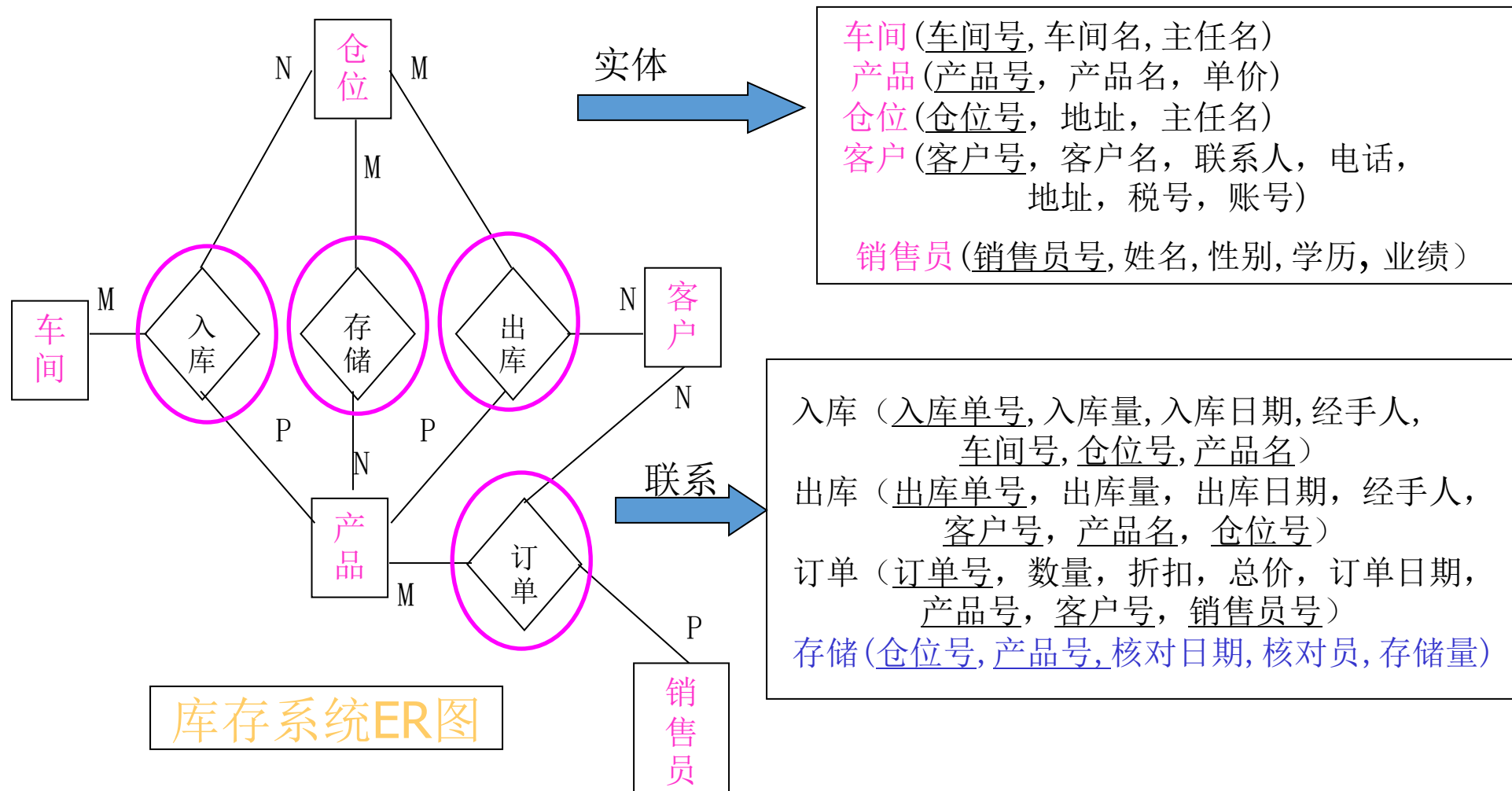
ER模型到关系模型的转换实例



ER模型到关系模型的转换实例

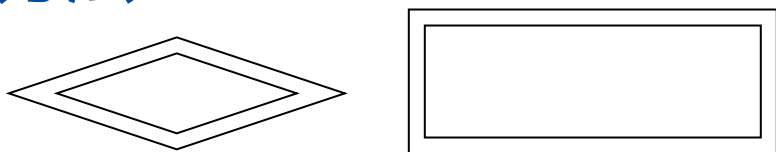


库存销售信息管理系统ER模型及转换



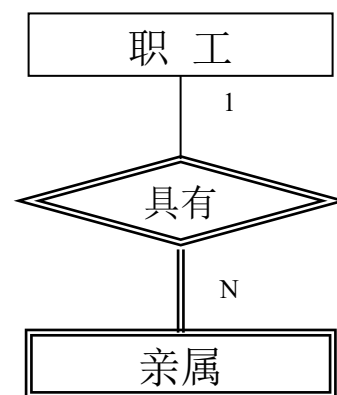
ER模型的扩充

- 弱实体：其存在必须依赖于其他实体
- 弱实体及其导致的联系表示方法：



- 包含弱实体的ER图转换成关系模式：
 - 弱实体的关系模式必须含有依赖实体的主键

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，所以说，亲属是弱实体



ER
图

职工（职工号，职工姓名，性别，年龄）

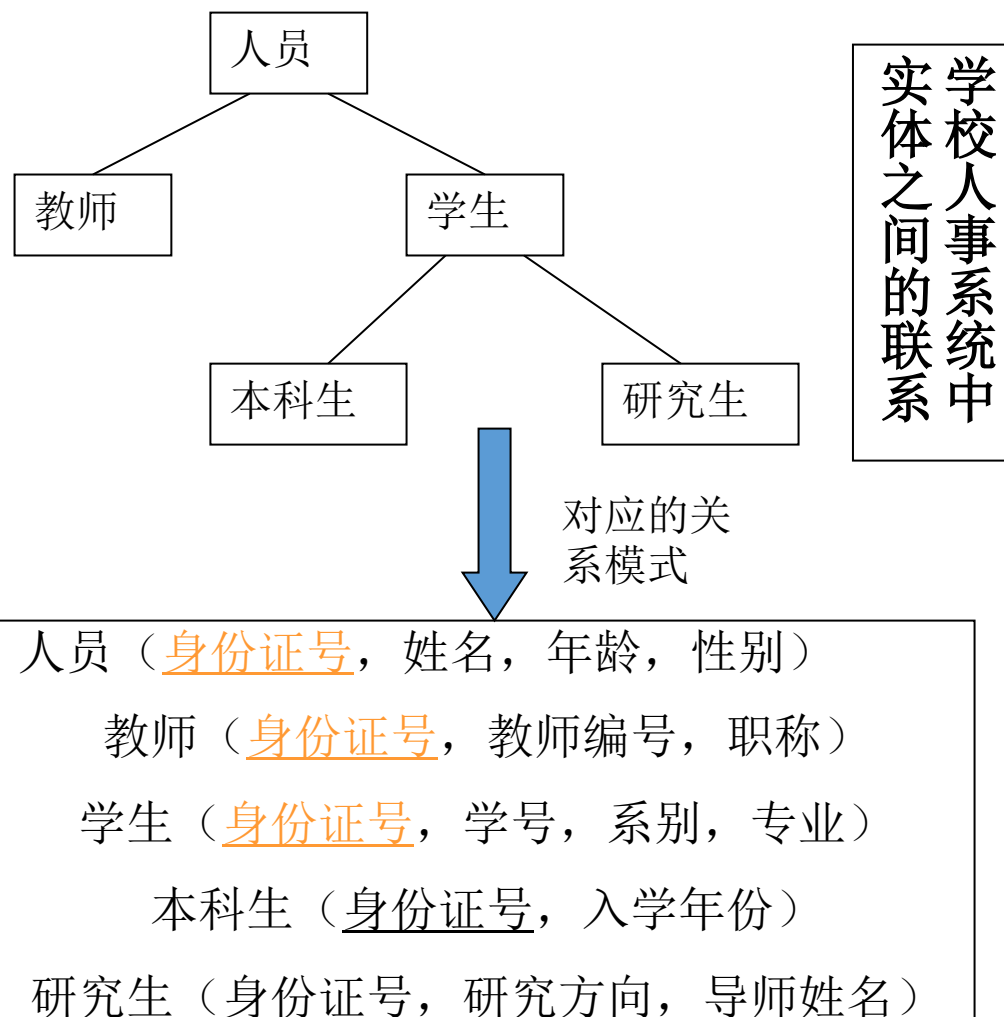
亲属（职工号，称呼，姓名，工作单位）

关系
模式

ER模型的扩充

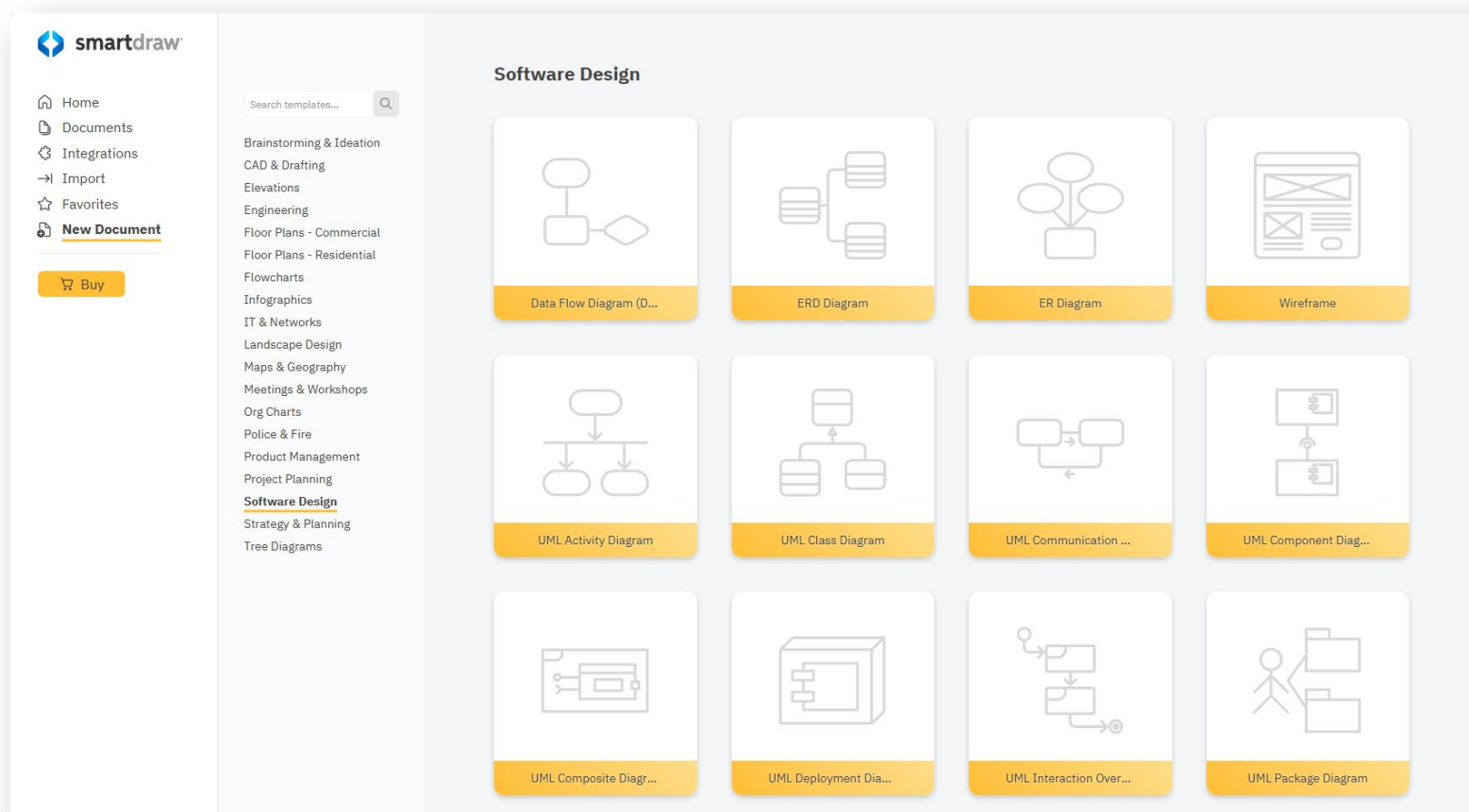
- 子类 and 超类的性质

- 子类与超类之间具有继承性，但子类本身还能包含比超类更多的属性。
- 子类和超类有相同的标识符（即主键）



做图工具

- SmartDraw: <https://app.smartdraw.com/>





浙江大学
ZHEJIANG UNIVERSITY

校园正版化软件平台

您好, 欢迎来到 浙江大学 正版化软件平台! 登录



浙江大学 管理学院
SCHOOL OF MANAGEMENT
ZHEJIANG UNIVERSITY

平台首页

微软应用下载

Adobe 下载

MATLAB 下载

WPS下载

JMP

JetBrains

帮助中心

安装激活说明



Microsoft® 产品

Visio 2021 专业版



Windows 7 专业版



Windows 8 专业版



Windows 8.1 专业版



Windows 10 专业版



Windows 11 专业版



Office 2016 专业增强版



Office 2019 专业增强版



Office 2021 LTSC



Visio 2021 专业版

选择 Visio 2021 专业版 版本:

系统架构: 本站下载32位 ▾

软件大小: 1.85G

MD5校验码: A20F397C69B4C8E847C524A2AC95DFF6

立即下载

商品介绍

配置要求

安装步骤

功能特点:

通过 Office Visio, 您可以使用结合了强大的搜索功能的预定义 Microsoft SmartShapes 符号来查找计算机上或网络上的合适形状, 从而轻松创建图表。

Office Visio 提供了特定工具来支持 IT 和商务专业人士的不同图表制作需要。

轻松刷新图表中的数据。Office Visio Professional 中新增的“刷新数据”功能可以自动刷新图表中的所有数据, 无需您手动刷新。

通过使用 Office Visio Professional 将图表链接至基础数据, 以提供更完整的画面, 从而使图表更智能、更有用。

使用 Office Visio, 可以通过多种图表, 包括业务流程图、软件界面、网络图、工作流图表、数据库模型和软件图表等直观地记录、设计和完全了解业务流程和系统的状态。

软件亮点:

- 1.支持限制对此关系图的权限
- 2.设置浏览器视图选项并检查数据连接

谢谢！ 下次见！