

## 第二章 数据模型

---

1

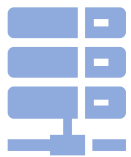
什么是数据模型

2

概念模型

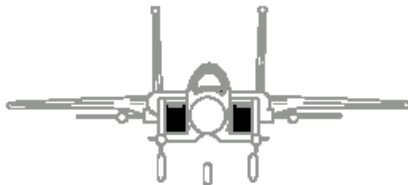
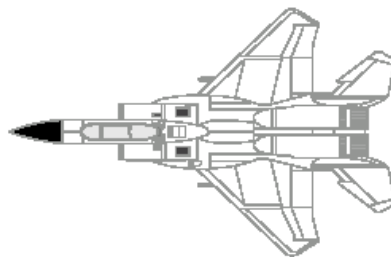
3

逻辑模型



# 什么是数据模型

- 模型是对现实世界的抽象
  - 模型常常用于帮助人们提取问题本质，从而更容易找到解决办法
- ✓ 飞机模型 → 飞机





# 什么是数据模型

---

- 数据模型是模型的一种，是现实世界数据特征的抽象，是用来描述数据的一组概念和定义。
  - 由于计算机不能直接处理现实世界中的具体事物，所以人们必须将具体事物转换成计算机能够处理的数据。
  - 数据本身是没有含义的，需要将它组织成固定格式，并用相应的语义去解释它，才能还原成人们想要的信息
- 数据在计算机中的组织结构，和施加在这个结构上的处理方法，即是数据模型



# 什么是数据模型

---

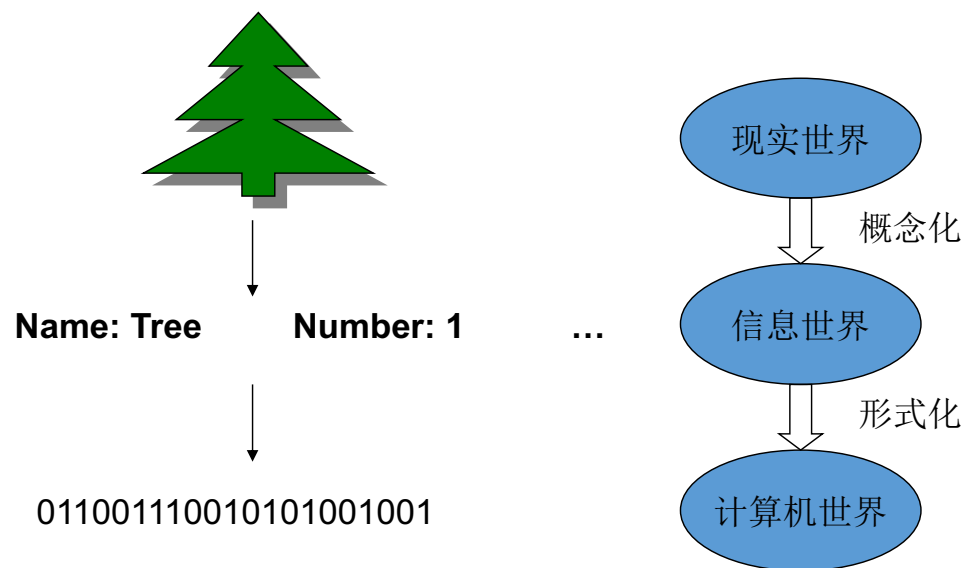
- 不同的数据模型代表了不同的观察世界的角度，代表了不同的抽象方法
- 数据模型是数据库的核心和基础，是数据库科学领域的研究重点
  - 数据库相对于文件系统的诸多优点，都源于它采用了更符合现实世界的数据模型
  - 数据模型是划分数据库种类的依据，因为它根本性的决定了数据库的功能/性能特点



# 什么是数据模型

## ■ 信息世界

- 现实世界的问题，要抽象为计算机能理解的数据才可以被处理
- 从现实世界到计算机世界的直接转换是非常困难的，我们引入一个信息世界作为现实世界通向计算机实现的桥梁
- 一方面，信息世界是对现实世界的抽象，从纷繁的现实世界中抽取能反映现实本质的概念和基本关系；另一方面，信息世界中的概念和关系，要以一定的方式映射到计算机世界中去，在计算机系统上最终实现。信息世界起到了承上启下的作用。





# 什么是数据模型

---

- 数据模型是对现实世界的抽象，属于信息世界。
- 现实世界是复杂的，需要从不同层次去考察
- 因此信息世界的的数据抽象过程又可细分为几个阶段，每个阶段用不同的模型去描述抽象结果，从而产生了如下的数据模型分类：
  - 概念数据模型
  - 逻辑数据模型
  - 物理数据模型



# 什么是数据模型

## ■概念数据模型：

- 是独立于计算机系统的数据模型，不涉及信息在计算机中的表示，只用来描述某个特定组织所关心的信息结构，是对现实世界的第一层**抽象**

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



- 概念模型是按用户的观点对数据建模，强调其语义表达能力，是用户和数据库设计人员之间进行交流的语言和工具。



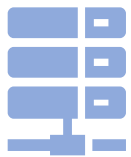
# 什么是数据模型

---

## ■ 逻辑数据模型:

- 直接面向数据库的逻辑结构,是对现实世界的第二层抽象。它直接与DBMS有关,有严格的形式化定义,以便在计算机系统中实现。
- 它通常有一组严格定义的无二义性语法和语义的DB语言,人们可以用这种语言来定义、操纵DB中的数据。
- 该类数据模型有层次模型、网状模型、关系模型等。
- 逻辑数据模型是区分不同类型数据库的依据,并在很大程度上决定了数据库的性能和应用范围。
- 通常所说的数据模型即指逻辑数据模型



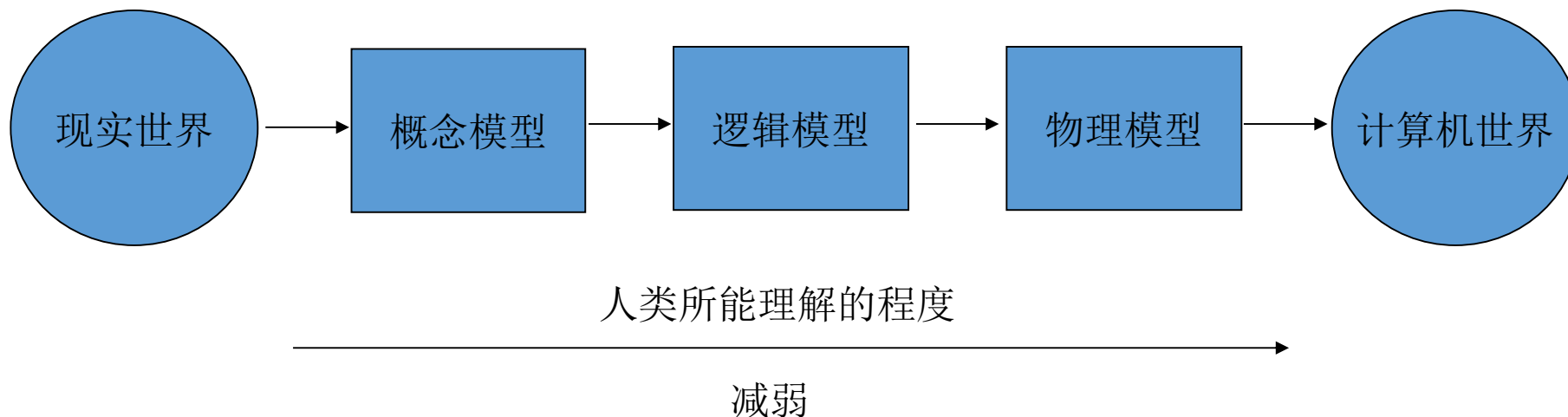


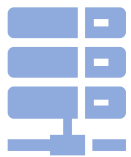
# 什么是数据模型

## ■ 物理数据模型

- 反映了数据在存储介质上的组成结构，并描述了访问机制
- 如何表达记录结构、记录顺序和访问路径等信息

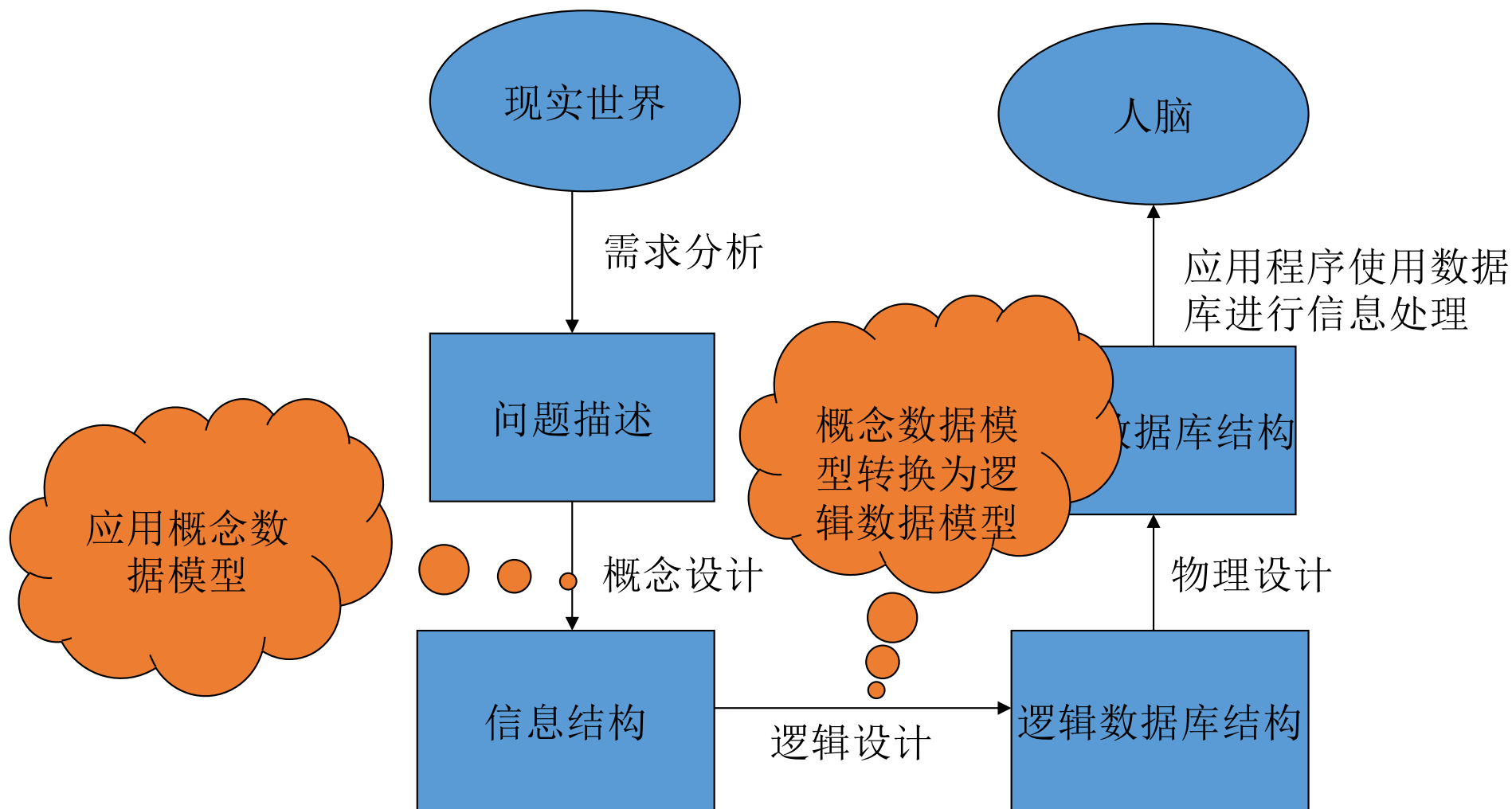
## ■ 数据模型的抽象级别





# 什么是数据模型

## ■ 基于数据模型开发数据库系统





# 概念模型

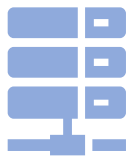
---

## ■ 概念模型

- 概念模型用于信息世界的建模，对现实世界进行第一层的抽象
- 独立于计算机系统
- 是数据库设计的有力工具
- 数据库设计人员和用户之间进行交流的语言

## ■ 对概念模型的基本要求

- 较强的语义表达能力，能够方便、直接地表达应用中的各种语义知识
- 简单、清晰、易于用户理解。



# 概念模型

## ■ ER数据模型：最常用的一种概念数据模型

- Entity-Relationship Data Model
- 由华人计算机科学家陈品山发明
- 从汉字起源中得到灵感，用图形化方法为现实世界的实体建模




<u>Original Form</u>	<u>Current Form</u>	<u>Meaning</u>
	日	Sun
	月	Moon
	人	Person
$\text{日 (sun)} + \text{月 (moon)} = \text{明 (Bright/ Brightness by light)}$		

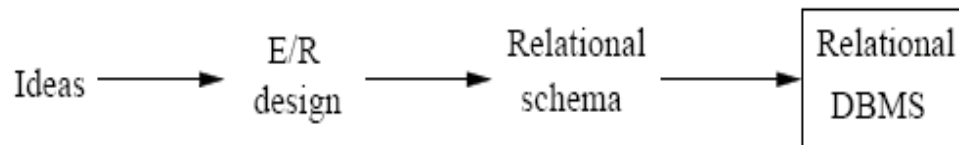
Fig. 6. Composition of Two Chinese Characters into a New Chinese Character

Chen, Peter P. (1976). "The Entity-Relationship Model - Toward a Unified View of Data". ACM Transactions on Database Systems 1 (1): 9-36.



# ER模型

- 对现实世界进行抽象
- 应用ER模型表达抽象结果
- 两种常用抽象手段



- 分类 (Classification)

- 定义某一类概念作为现实世界中一组对象的类型
- 在特定的上下文里，这些对象具有某些共同的特性和行为
- 大学里的学生，奥运会的赛事，公交公司的司机...

- 聚集 (Aggregation)

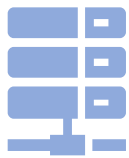
- 定义某一类型的组成成分
- 根据上下文管理需求，给出指定类型的特征集合
- 学生：学号、姓名、年龄等特征组成



# ER模型

---

- 观点：世界是由一组称作实体的基本对象和这些对象之间的联系构成的
- 给我们一种语言去描述如下信息：
  - 数据库中应该存储什么对象信息
  - 对象之间存在什么样的联系
- 三个基本元素
  - 实体 (Entity)
  - 属性 (Attribute)
  - 联系 (Relation)



# ER模型

## ■ 实体

- 客观存在并可以相互区分的客观事物或抽象事件称为实体

- 可以触及的客观对象仓库、器件、职工.....等是实体

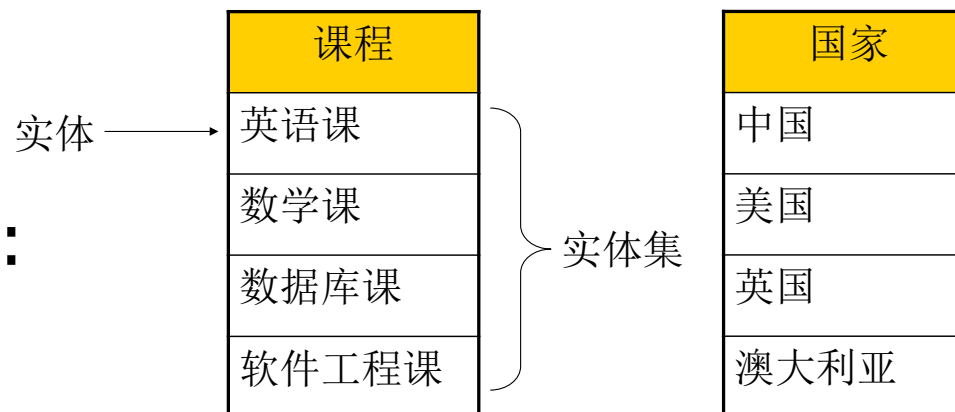
- 客观存在的抽象事件订货、演出、足球赛.....等也是实体

- 实体集

- 同型实体的集合

- 经常把实体集简称为实体。因此 ‘实体’ 的具体含义要看上下文

“分类”的抽象方法:





# ER模型

## ■ 属性

- 实体所具有的某一特性称为属性
- 一个实体可由若干个属性来刻画
  - ✓ 学生实体可由学号、姓名、年龄、系等属性来描述
- 属性类型不同的实体属于不同的实体集
- 属性类型相同，仅取值不同的实体属于同一个实体集

属性

课程名	任课老师	课时
英语	张老师	60
数学	李老师	60
数据库	马老师	50
软件工程	任老师	30

实体 →

“聚集”的抽象方法:





# ER模型

## ■ 属性

- 域 (Domain)

- ✓ 属性的取值范围。所有可能的取值的集合。

- 属性的类型 (Type)

- 简单属性

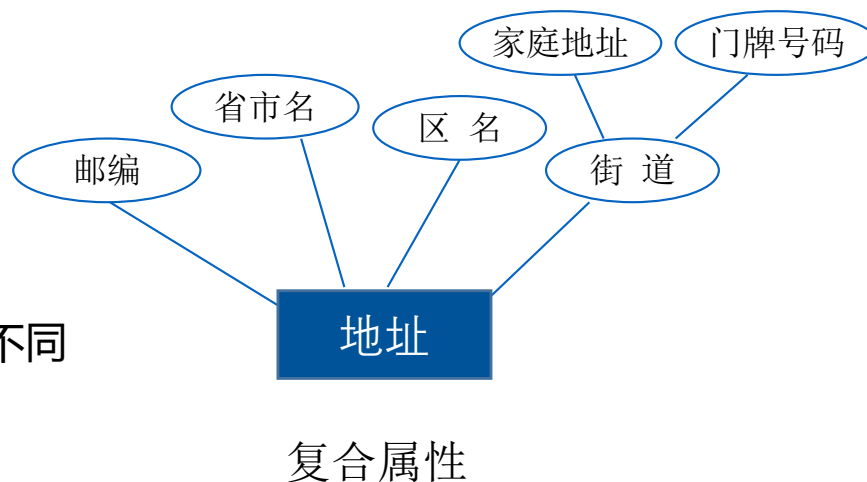
- ✓ 不可再分的属性，如学号、年龄等

- 复合属性

- ✓ 可以划分为更小的属性，每个属性的域不同

- ✓ 如：街道 = 家庭地址 + 门牌号码

- ✓ 用于把相关属性聚集起来，使ER模型更清晰





# ER模型

## ■ 属性

- 域 (Domain)

- ✓ 属性的取值范围。所有可能的取值的集合。

- 属性的类型 (Type)

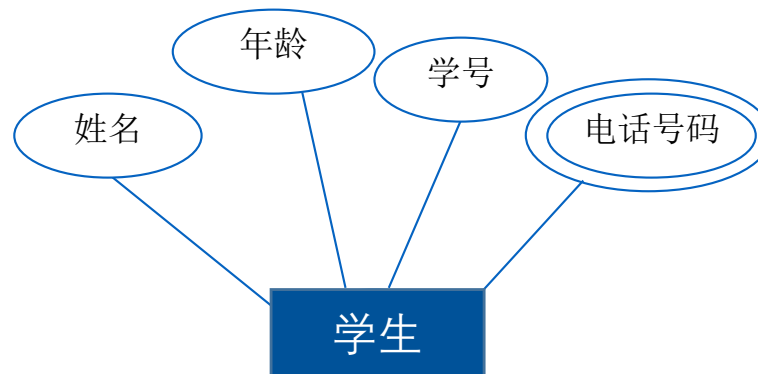
- 单值属性

- ✓ 每一个特定的实体在该属性上的取值唯一，如学生的学号、年龄等

- 多值属性

- ✓ 某特定实体在该属性上有多于一个的取值，如某人的电话号码

多值属性





# ER模型

## ■ 属性

- 域 (Domain)

- ✓ 属性的取值范围。所有可能的取值的集合。

- 属性的类型 (Type)

- 键 (key)

- ✓ 实体集中能唯一标识实体的属性或属性组称为实体集的键

- ✓ 两个以上属性组成的键称为复合键(compound key)

- ✓ 键属性用下划线标示出来

- 例：学生 (学号, 年龄, 性别, 班级)

- 成绩单 (学生, 课程, 学期, 成绩) ---复合键

- ✓ 实体集中的每一个实体，在键属性上的取值各不相同



# ER模型

## ■ 联系

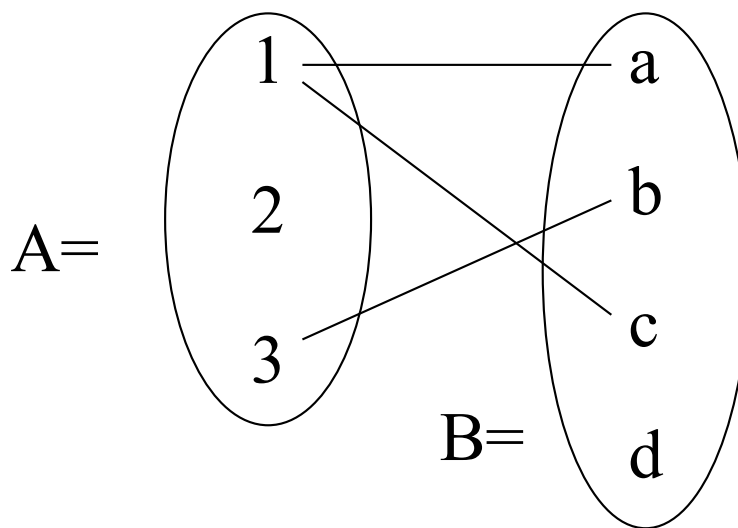
### ● 联系的数学定义

✓ 如果A、B是集合 (set) , 则联系 R 是  $A \times B$  的一个子集

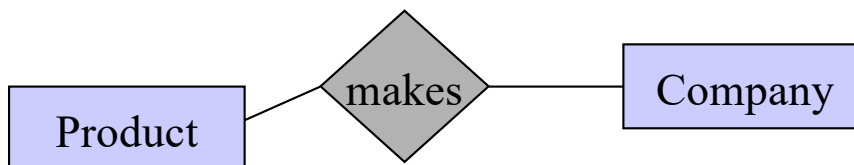
$A \times B$  是A和B的笛卡尔积

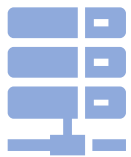
$A = \{1, 2, 3\}$ ,  $B = \{a, b, c, d\}$ ,

$R = \{(1, a), (1, c), (3, b)\}$



又例: **makes** 是 **Product**  $\times$  **Company** 的子集





# ER模型

## ■ 联系

- 联系由一系列的实体间对应关系所组成

教师实体集

教师	职称
张老师	副教授
李老师	教授

指导联系

教师	学生
张老师	王楠
张老师	李飞
李老师	张雨

学生实体集

学生	班级
王楠	3301
李飞	3302
张雨	3302

联系的键是复合键

其键由来自于各相关实体集的键属性组成



# ER模型

## ■ 联系

- 也可以用属性来刻画联系的特性

指导联系

教师实体集 ←	教师	学生	学科	→ 学生实体集
	张老师	王楠	数学	
	张老师	李飞	数学	
	李老师	张雨	语文	

上面联系中的“学科”是联系自身的属性，不属于相关实体集



# ER模型

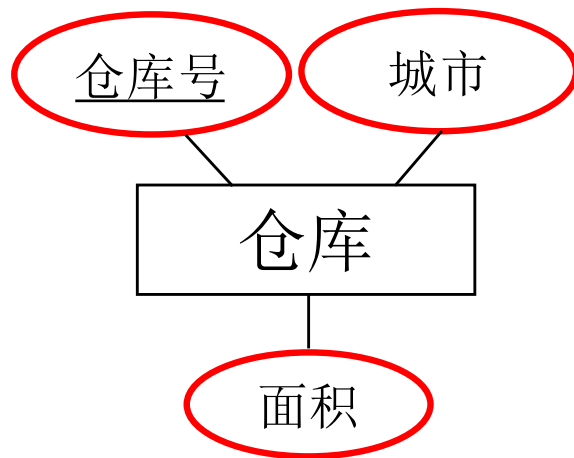
## ■ ER图(ER diagram)

### ●用图形化的方法来表示ER数据模式

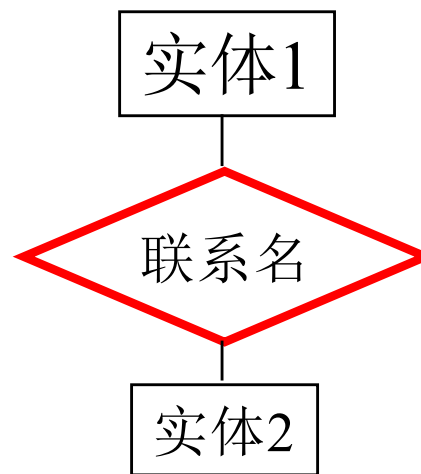
- ✓用矩形框表示实体，把实体名写在框内
- ✓实体的属性用椭圆框表示，框内写上属性名，并用连线连到相应实体。
- ✓实体之间的联系用菱形框表示，框内写上联系名，并用连线与有关的实体相连。



实体



属性



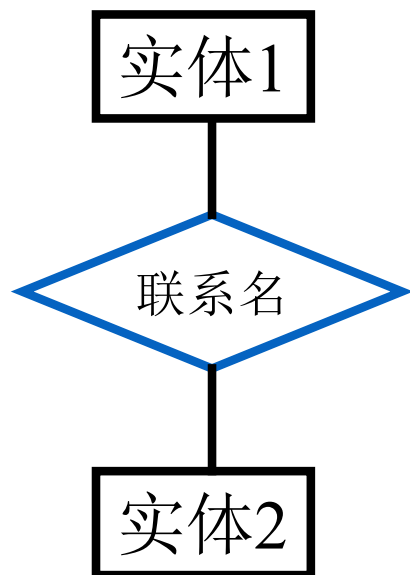
联系



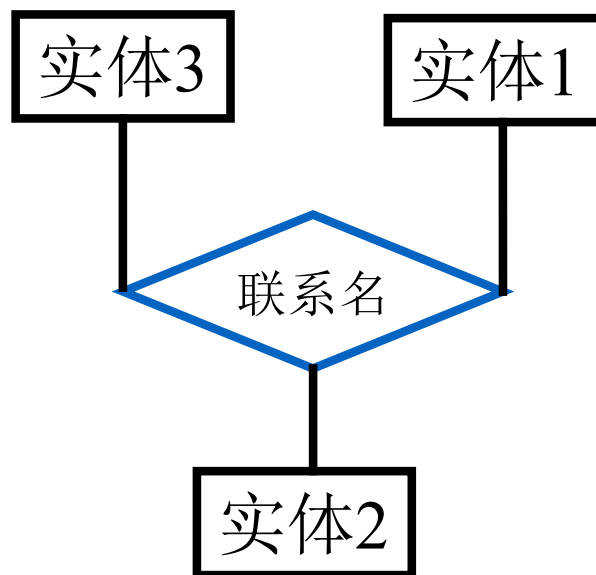
# ER模型

## ■ 联系的元 (degree)

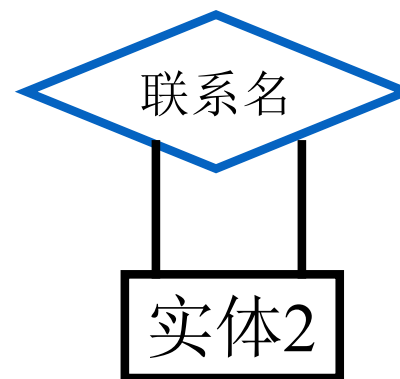
- 参与联系的实体的个数称为联系的元



二元联系  
Binary relationship



三元联系  
Ternary relationship



一元联系  
Unitary relationship





# ER模型

## ■ 联系的基数比约束(Mapping Cardinality Constraints)

- 表示一个实体可以通过联系关联到的另一个实体集中实体的数量。

- **一对一联系 (1:1)**

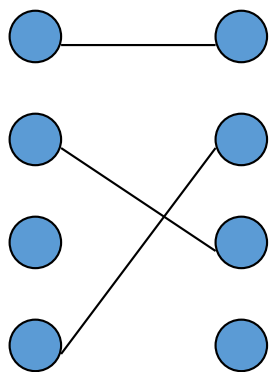
✓ 实体集A中的一个实体至多与实体集B中的一个实体相对应，反之亦然，则称实体集A与实体集B为一对一的联系。记作1:1。

示例:

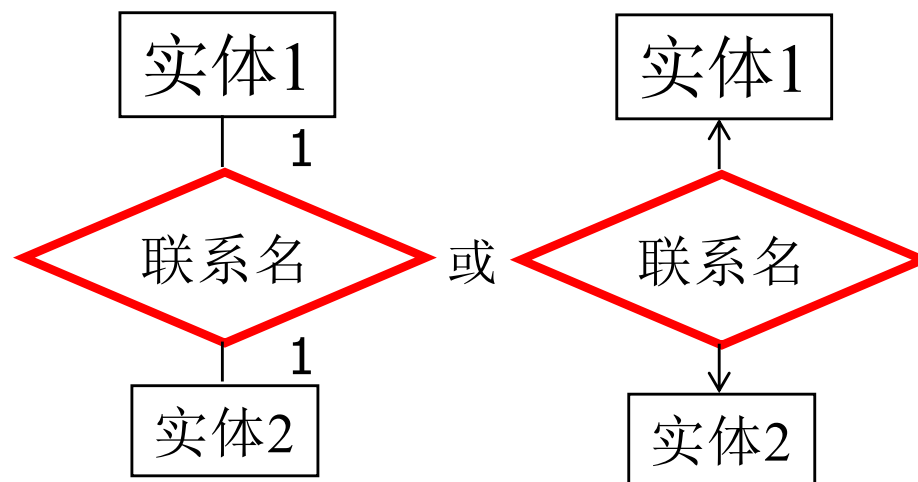
班级与班长

观众与座位

夫与妻



**1:1**联系图示



**1:1**联系表示法



# ER模型

## ■ 联系的基数比约束(Mapping Cardinality Constraints)

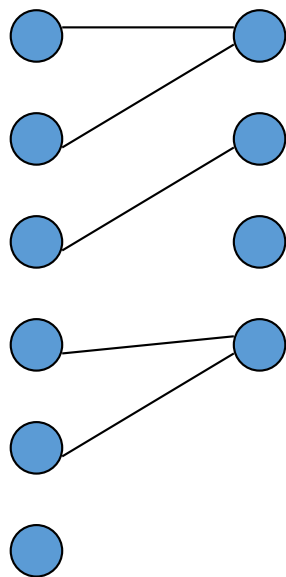
- 表示一个实体可以通过联系关联到的另一个实体集中实体的数量。
- 一对一联系 (1:1) 实体集A中的一个实体可以与实体集B中的多个实体相对应，反之，实体集B中的一个实体至多与实体集A中的一个实体相对应
- 一对多联系 (1:n)

示例:

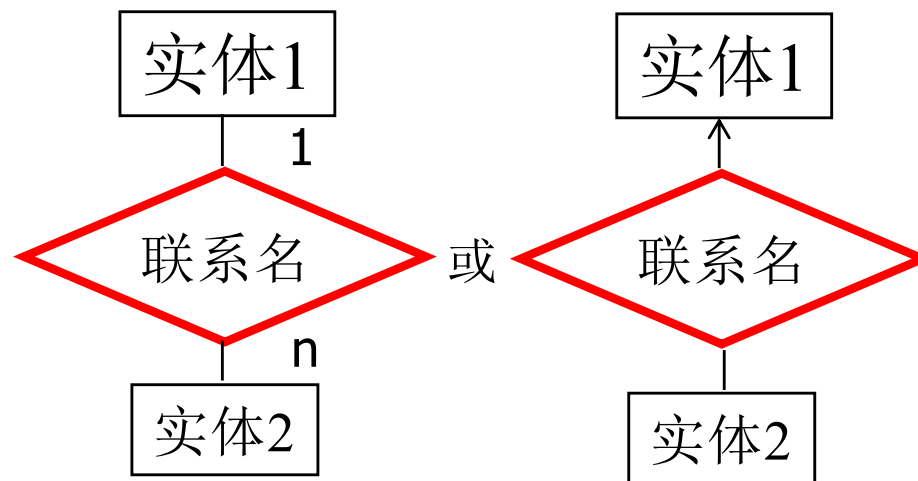
班级与学生

公司与职员

省与市



1:n联系图示



1:n联系表示法



# ER模型

## ■ 联系的基数比约束(Mapping Cardinality Constraints)

- 表示一个实体可以通过联系关联到的另一个实体集中实体的数量。

- 一对一联系 (1:1)

- 一对多联系 (1:n)

- 多对多联系 (m:n)

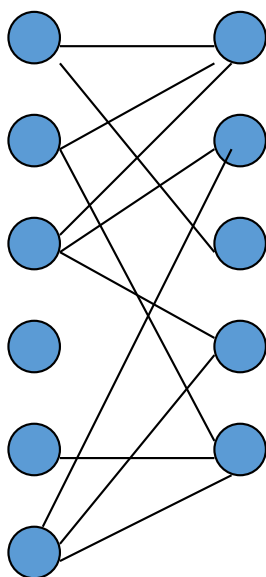
实体集A中的一个实体可以与实体集B中的多个实体相对应，反之亦然

示例:

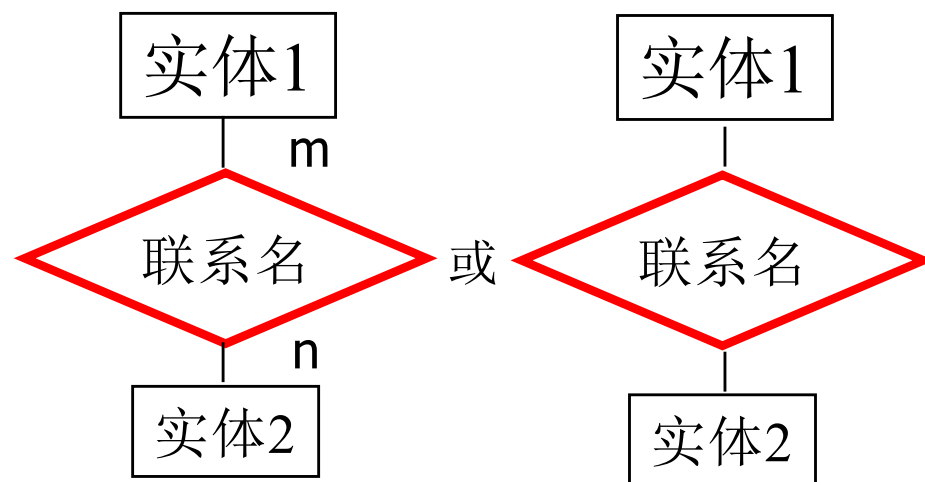
教师与学生

学生与课程

工厂与产品



**m:n**联系图示

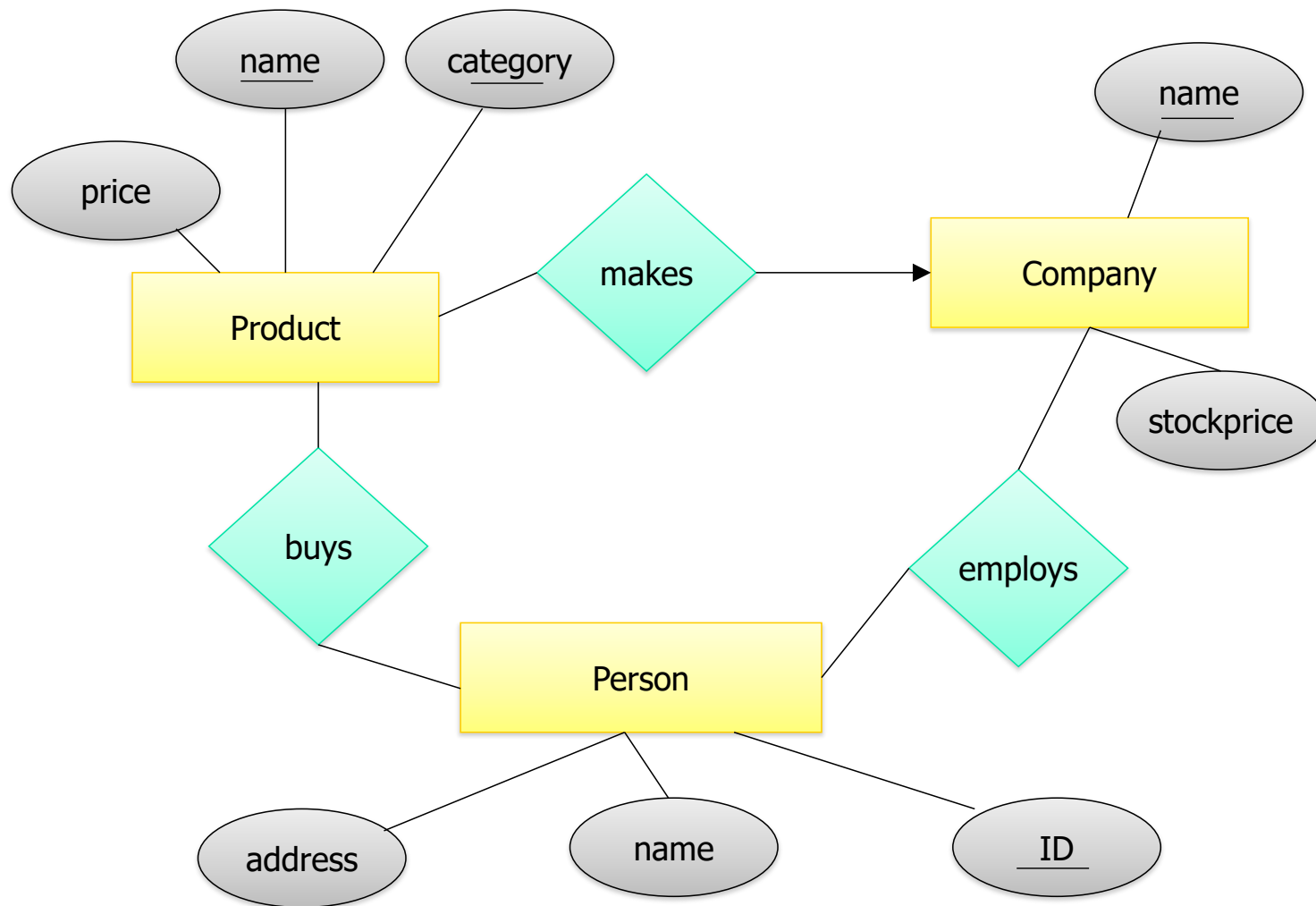


**1:n**联系表示法



# ER模型

解读一下？

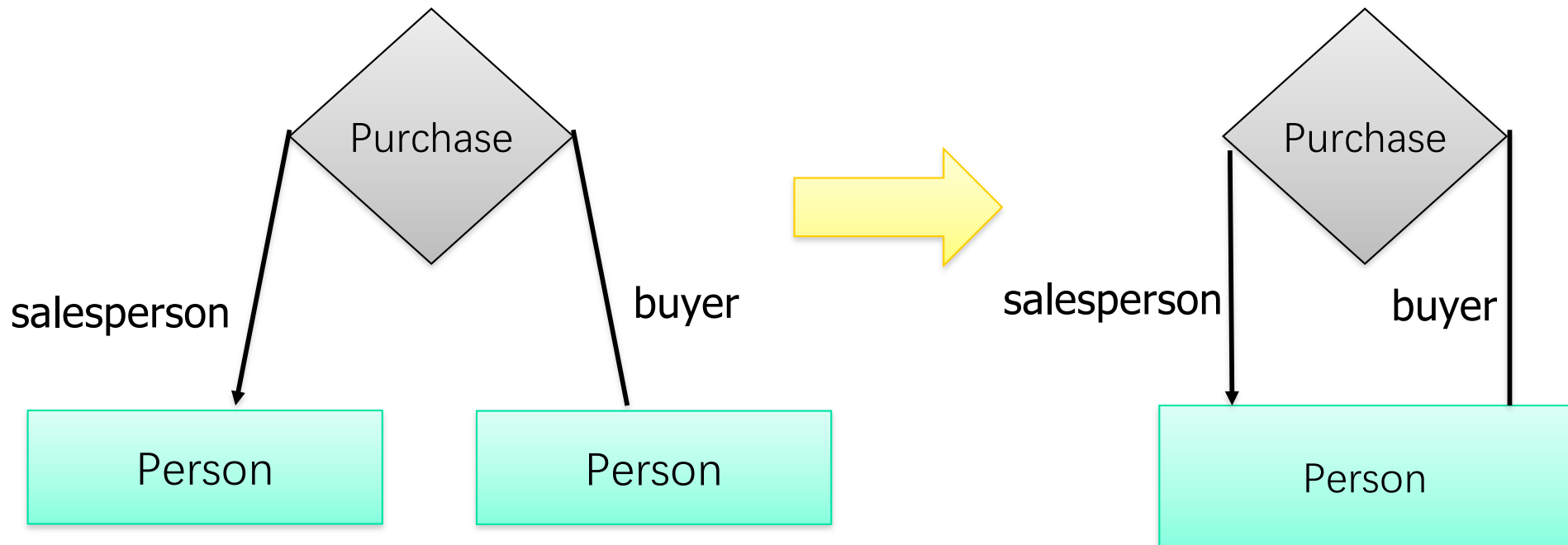


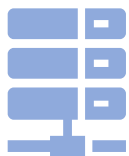


# ER模型

## ■一元联系

- ✓ 有时二元联系关联的两个实体集实际上属于同一个实体集的两个副本
- ✓ 可以将这二个实体集合并，形成一元联系





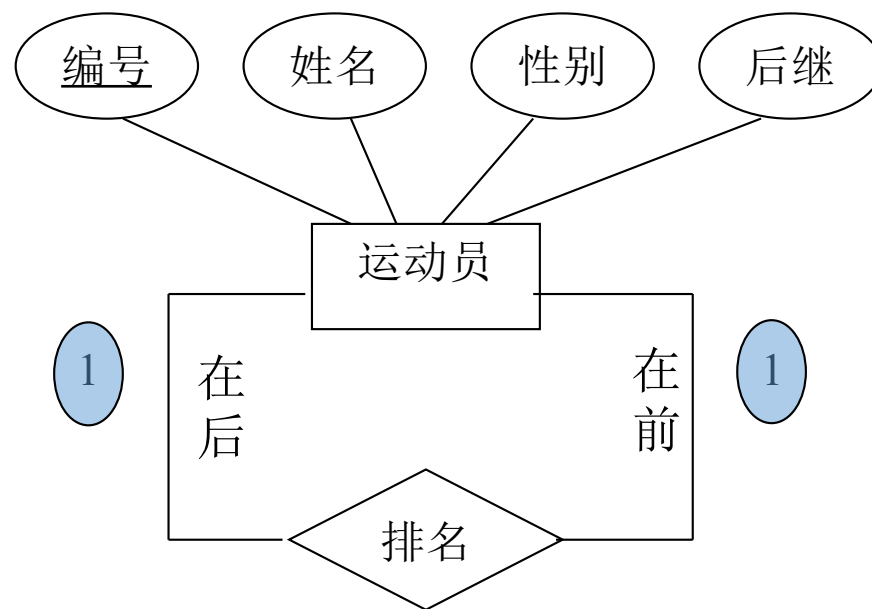
# ER模型

## ■一元联系

### ✓ 一元联系中的1:1联系

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

运动员	后继运动员	说明
A	B	A在B前
B	C	B在C前
C		



运动员之间的排名联系



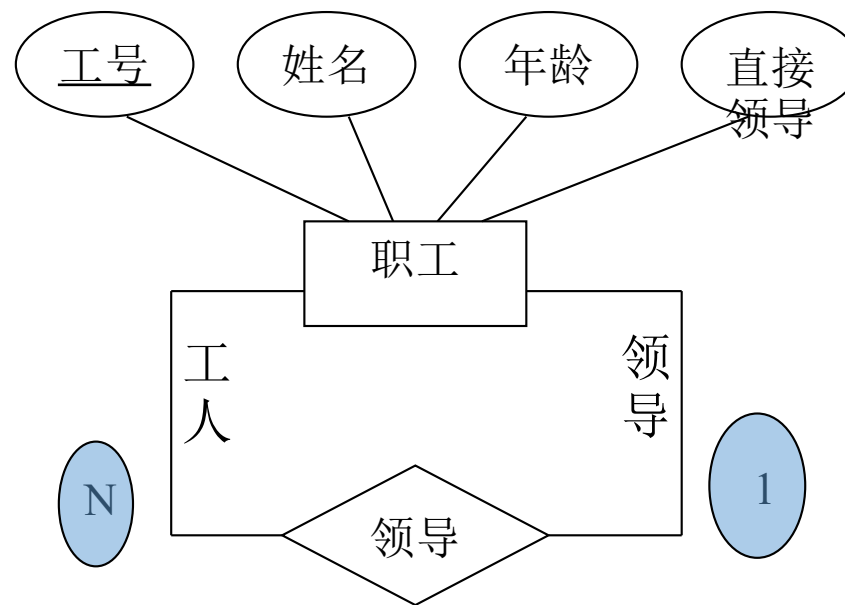
# ER模型

## ■一元联系

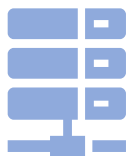
### ✓ 一元联系中的1:n联系

职工之间的上下级关系中，一个领导可以管理多个职工，每个职工只隶属于一个领导

职工	上级领导	说明
A	B	B领导A
B		
C	B	B领导C
D	C	C领导D
E	C	C领导E



职工之间的上下级联系



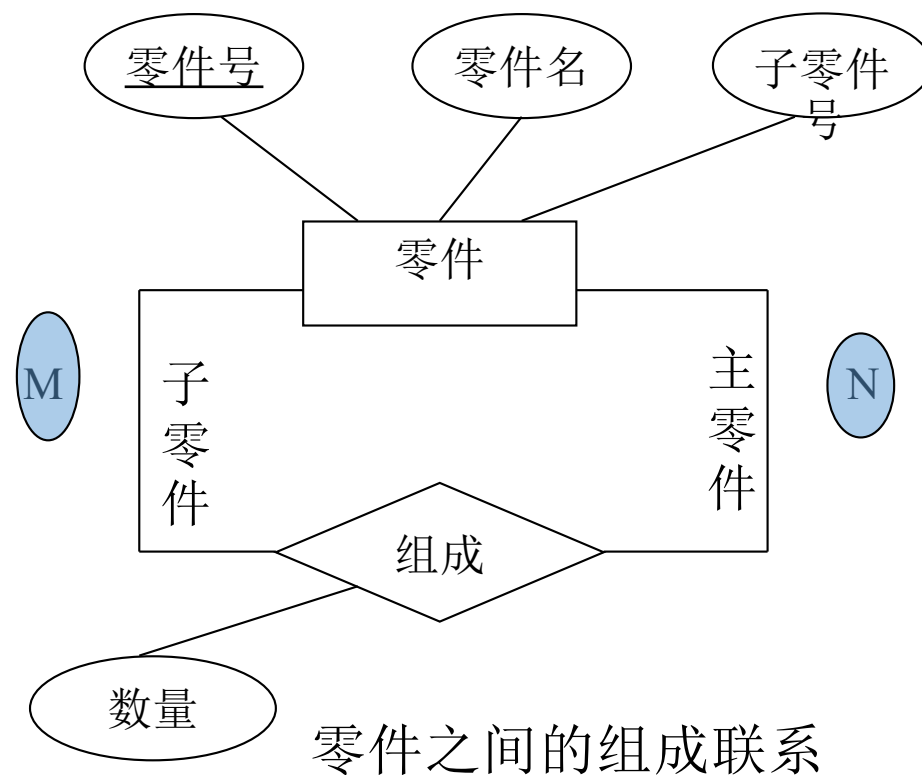
# ER模型

## ■一元联系

### ✓ 一元联系中的m:n联系

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

零件	子零件	组成关系
A	B	B属于A
A	C	C属于A
D	A	A属于D
E	A	A属于E



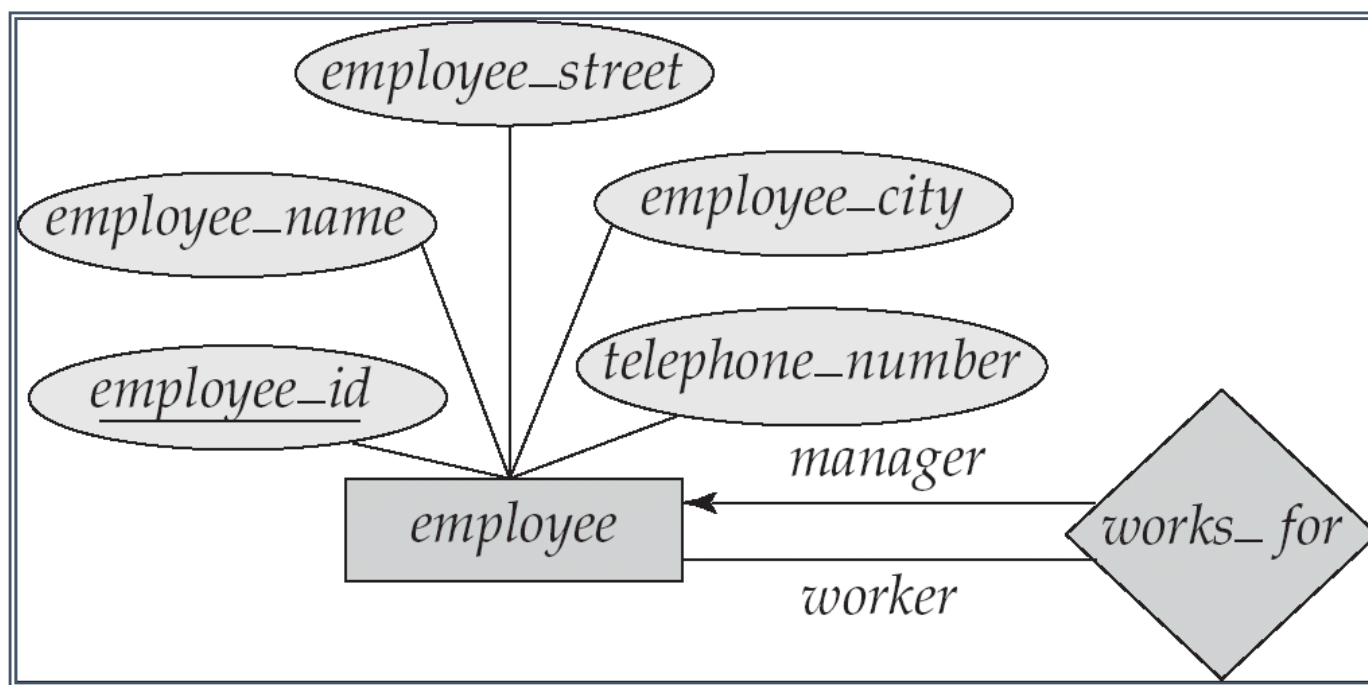




# ER模型

## ■ 联系中的角色 (Roles)

- ✓ 在联系中关联的每个实体集应是相互区别的。在一元联系中，需要用角色来区分同一实体集的不同副本
- ✓ 下图中的标签 “manager” 和 “worker” 被称为角色，它们说明了 employee 实体是如何通过 works\_for 联系相互关联的



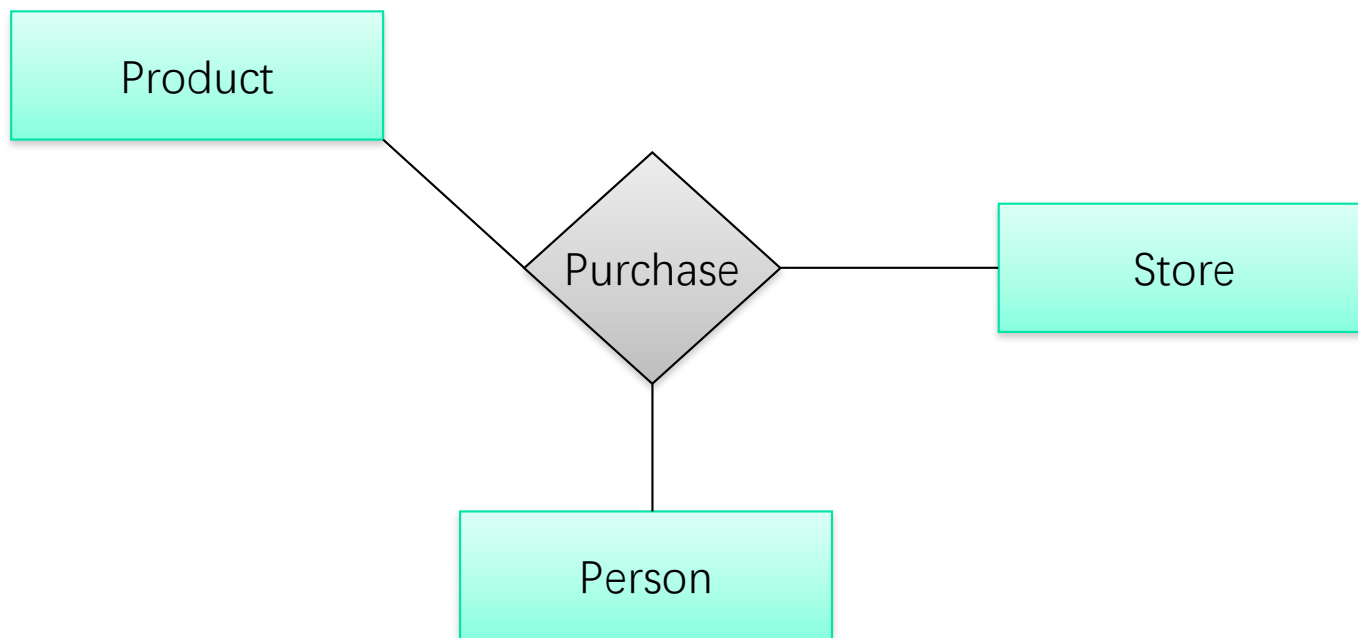


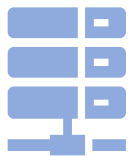
# ER模型

## ■ 多元联系

二元以上的联系称为多元联系

如何解读下面的多元联系？

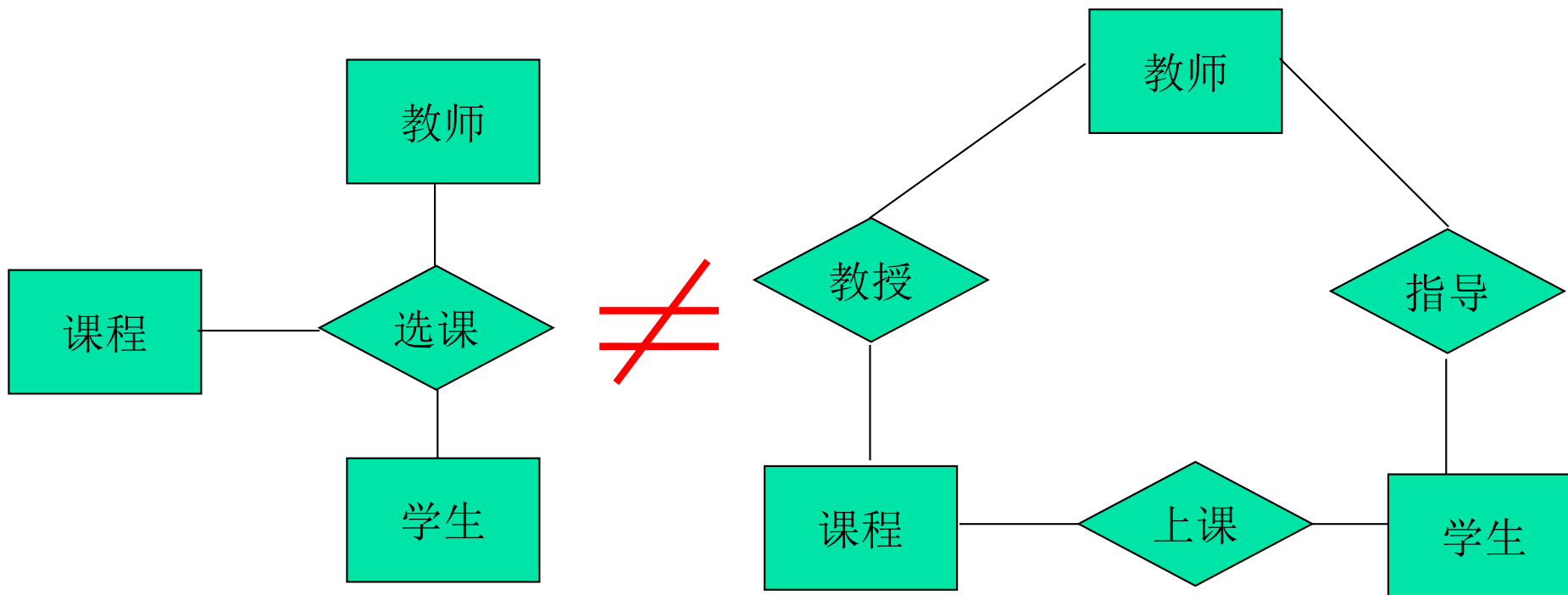




# ER模型

## ■ 多元联系

为什么？

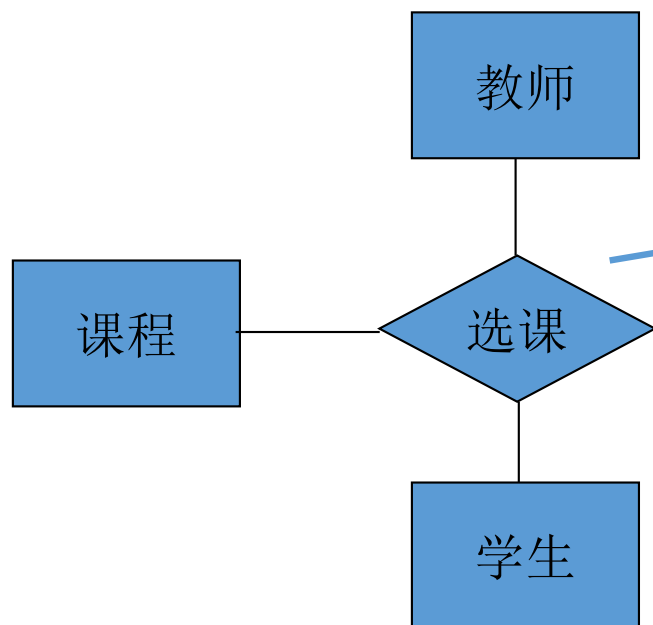




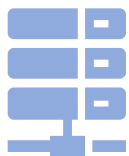
# ER模型

## ■ 多元联系

试查询：教授A数学课的老师姓名？



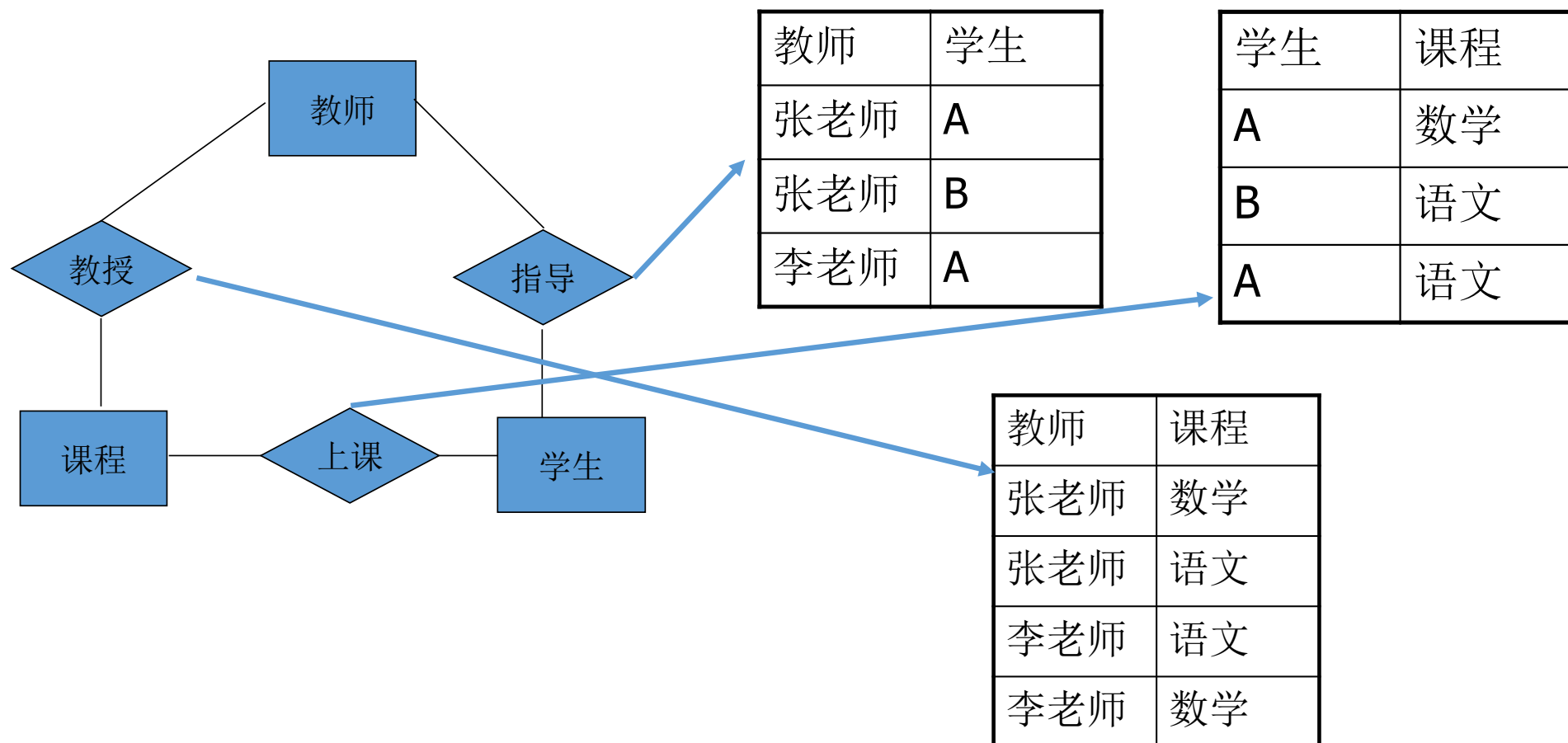
学生	教师	课程
A	张老师	数学
B	张老师	语文
A	李老师	语文



# ER模型

## ■ 多元联系

试查询：教授A数学课的老师姓名？

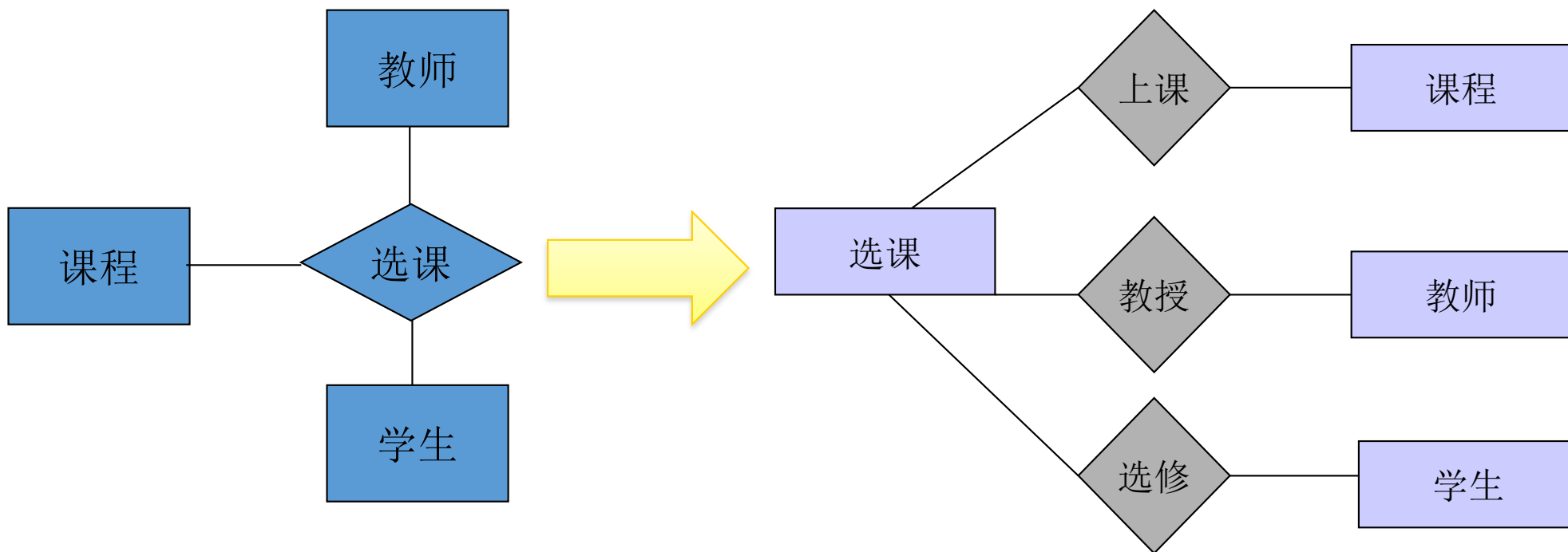




# ER模型

## ■ 多元联系

把三元联系转换为二元联系

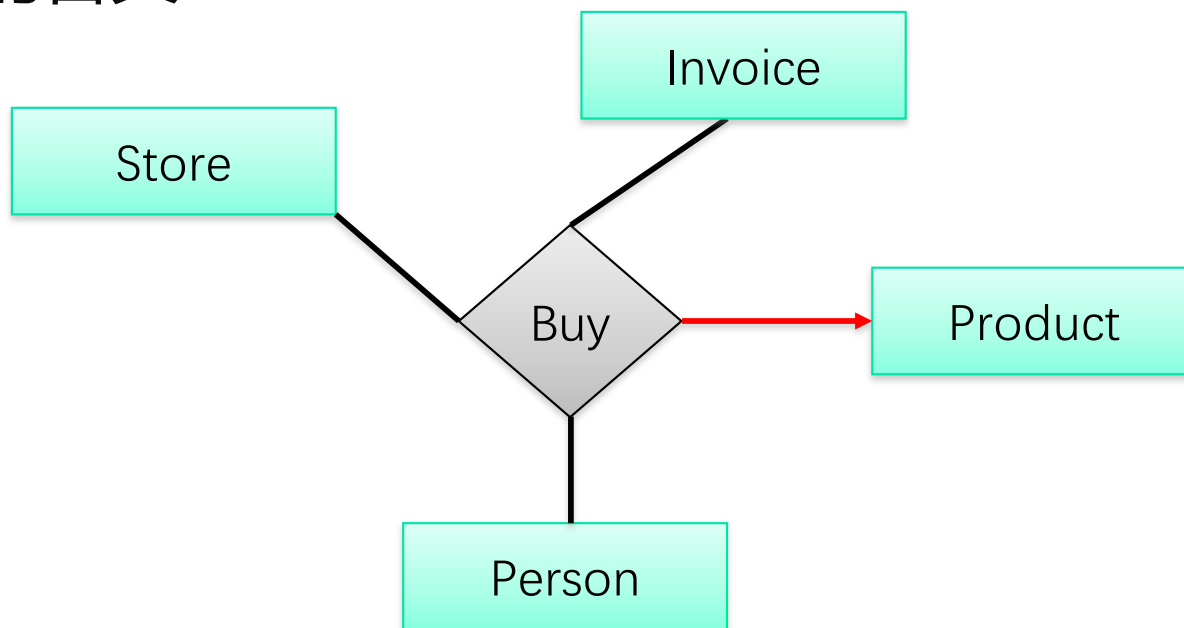




# ER模型

## ■ 多元联系中的箭头

Q: 图中箭头的含义?



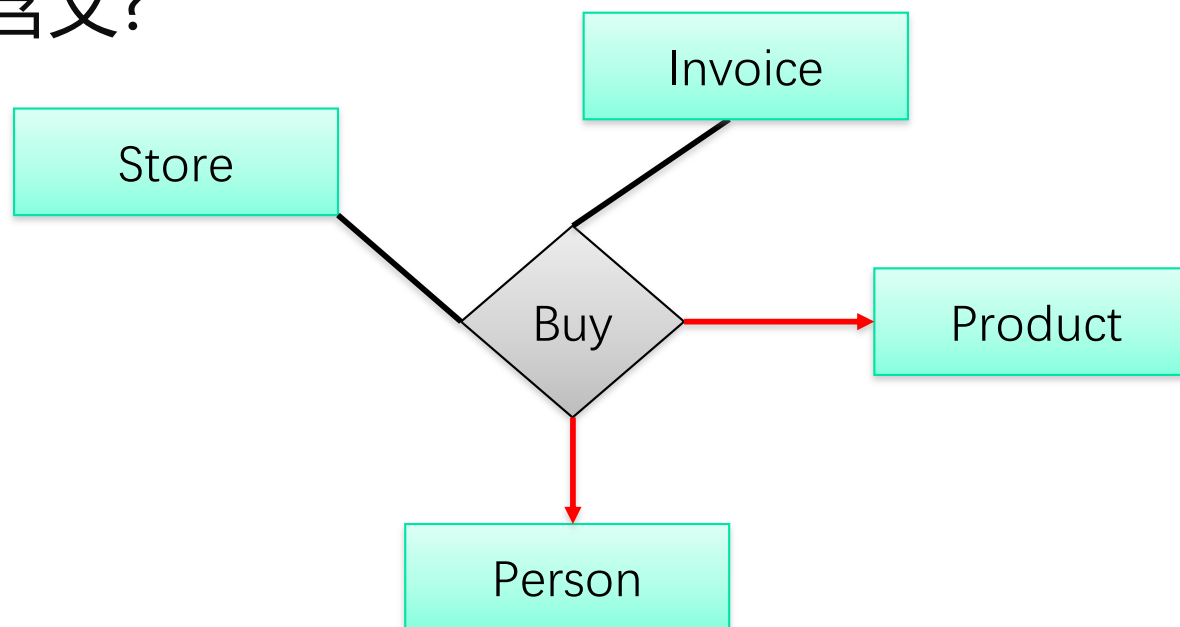
A: 如果知道了Store、Invoice（发票）、Person的信息，就能查到买的是哪件Product



# ER模型

## ■ 多元联系中的箭头

Q: 图中箭头的含义?



**A:** 知道Store、Invoice和Person的信息就能确定Product的信息；知道Store、Invoice和Product的信息就能确定Person的信息





# ER模型

---

## ■ 约束 (Constraints)

- 约束是一种对数据库的断言 (assertion) , 数据库应保证该断言成立
- 断言内容是对插入数据库中的数据进行限定。常见约束:
  - ✓ 非空约束; 唯一值约束; 主键约束; 外键约束; Check约束等等
- 约束是数据库模式的重要组成部分
  - ✓ 丰富了数据的语义
  - ✓ 保证数据的有效性和完整性
  - ✓ 实现更高效的存储和数据查询
- 关于约束, 将在后续《数据完整性》章节进一步阐述



# ER模型

## ■ 键约束 (Key Constraints)

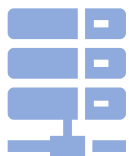
- 用键约束可以容易的引用实体
- **超键** (super key) 是实体集中的一个或一组属性, 用它们可以唯一的确定每一个实体
- **候选键** (candidate key) 是某个超键的最小集
- 若存在多个候选键, 则可以任选一个作为该实体集的**主键** (primary key)。主键会被数据库登记和使用。

✓ 例: 学生 (学号, 身份证号, 年龄, 性别, 班级)

超键: (学号, 身份证号, 年龄), (学号, 性别, 班级) ...

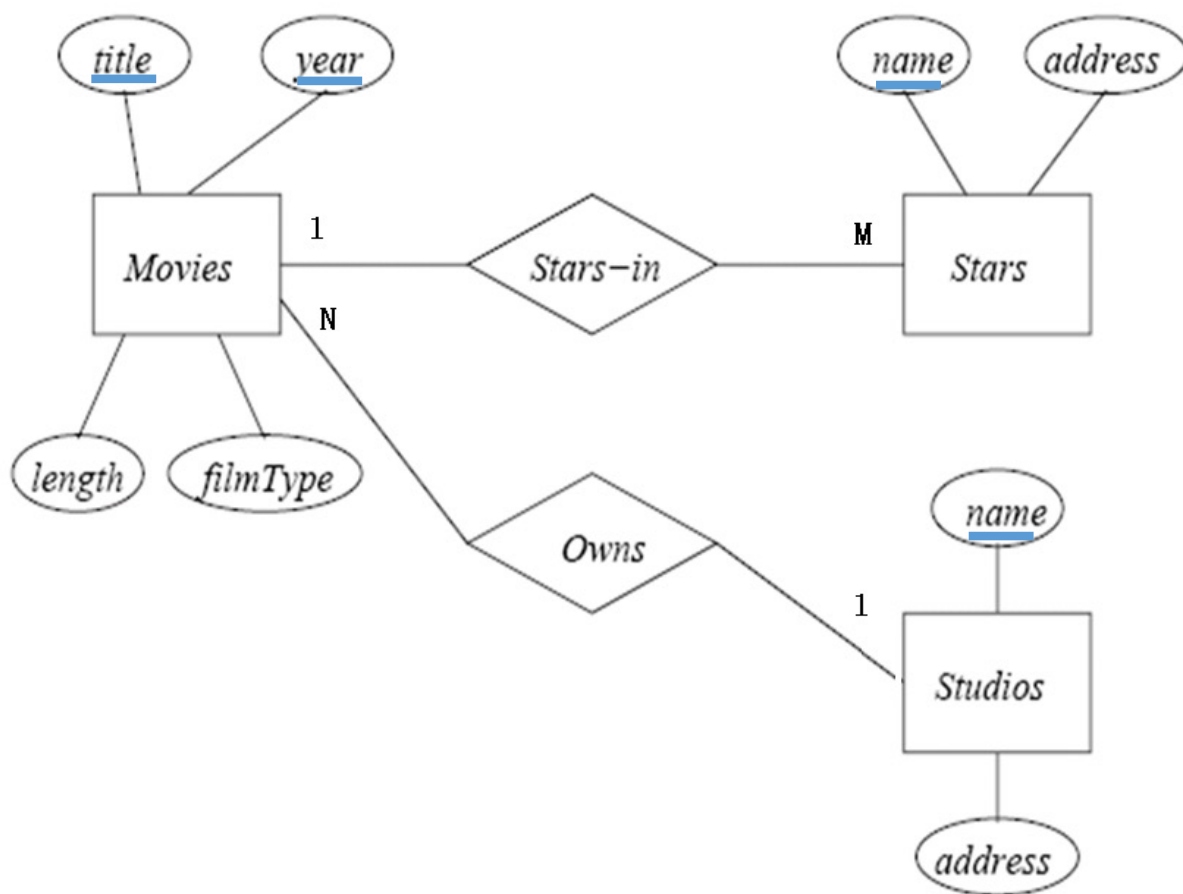
候选键: (学号), (身份证号)

主键: (学号)



# ER模型建模

## ■ 看图说话



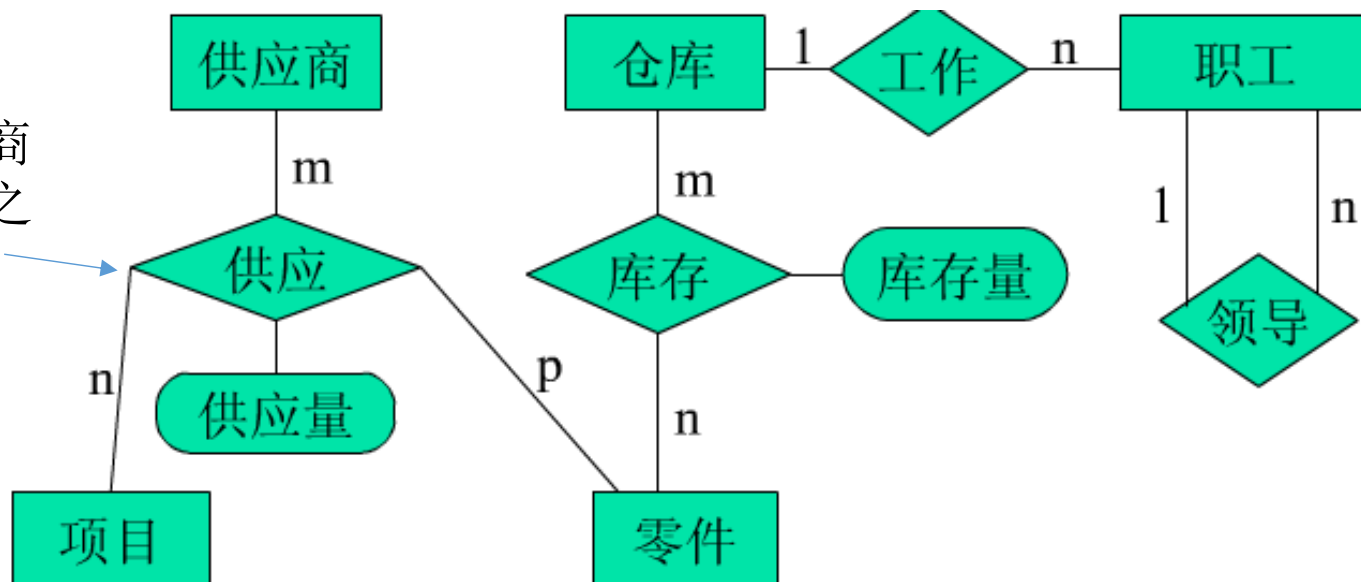


# ER模型建模

## ■ 根据需求描述建模

- ✓ 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。
- ✓ 某种零件在某个仓库中的数量用库存量描述。
- ✓ 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的关系，即仓库主任领导若干保管员

怎么描述供应商、项目、零件之间的联系？

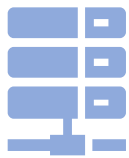




# ER模型建模

---

- 一些建模准则
  - ✓ 忠实于用户需求
  - ✓ **KISS**准则 (keep it simple and stupid)
  - ✓ 避免冗余
  - ✓ 能抽象为属性的，就不要抽象为实体



# ER模型建模

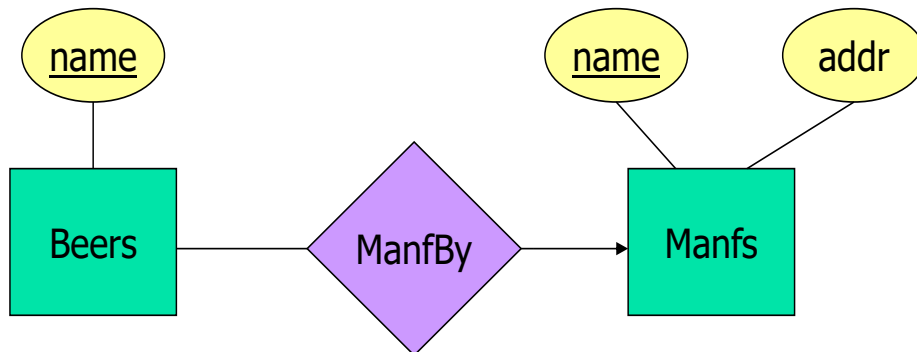
## ■ 一些建模准则

### ✓ 设计上避免冗余 (redundancy)

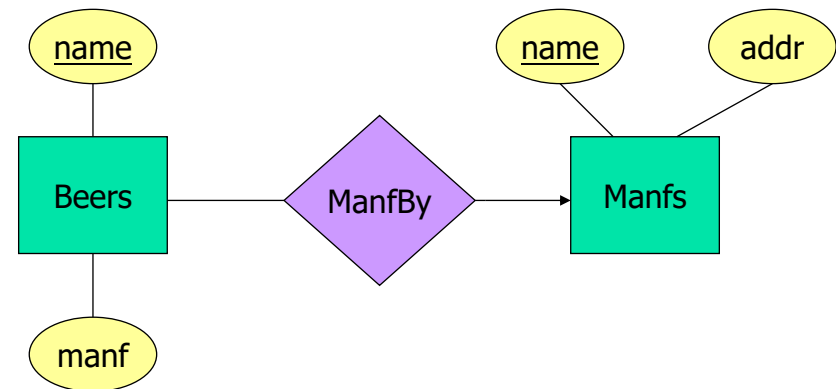
✓ 冗余：用两个以上不同的实例描述同一件事情(two instances of one fact)

✓ 冗余会导致空间浪费，更会带来数据不一致

✓ 在一个实例上修改了数据，而另一个实例没有修改，则会产生数据冲突



Good  
No redundancy



Bad  
states the manufacturer of a beer twice:  
as an attribute and as a related entity



# ER模型建模

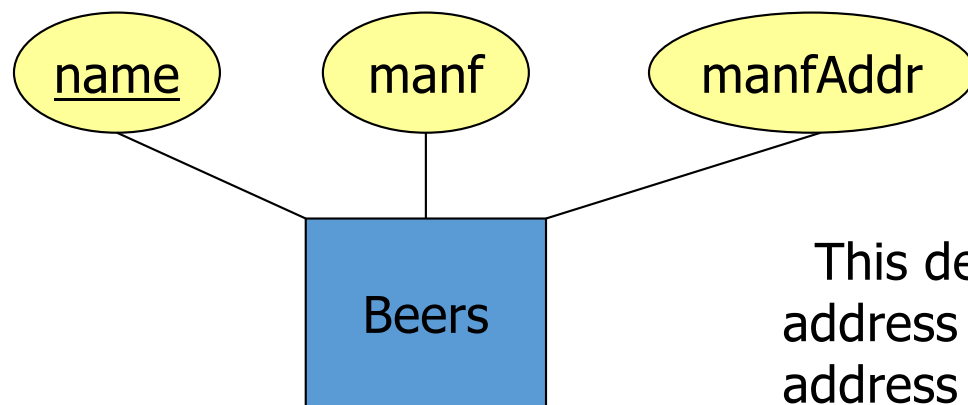
## ■ 一些建模准则

### ✓ 设计上避免冗余 (redundancy)

✓ 冗余：用两个以上不同的实例描述同一件事情(two instances of one fact)

✓ 冗余会导致空间浪费，更会带来数据不一致

✓ 在一个实例上修改了数据，而另一个实例没有修改，则会产生数据冲突



Bad

This design repeats the manufacturer's address once for each beer; loses the address if there are temporarily no beers for a manufacturer.



# ER模型建模

## ■ 一些建模准则

✓ 能抽象为属性的，不要抽象为实体

✓ 实体 VS 属性

✓ 理论上所有数据对象都可以被看成实体

✓ 但是只有在符合下面两个情况之一的时候才需要把事物抽象为实体

1. 事物有至少一个非键属性（不属于键的属性）

学生（姓名） VS 书（书名，页数，出版年代）

2. 事物处于“一对多”或“多对多”联系中“多”那一端

✓ 不符合以上情况的事物，都可以抽象为属性

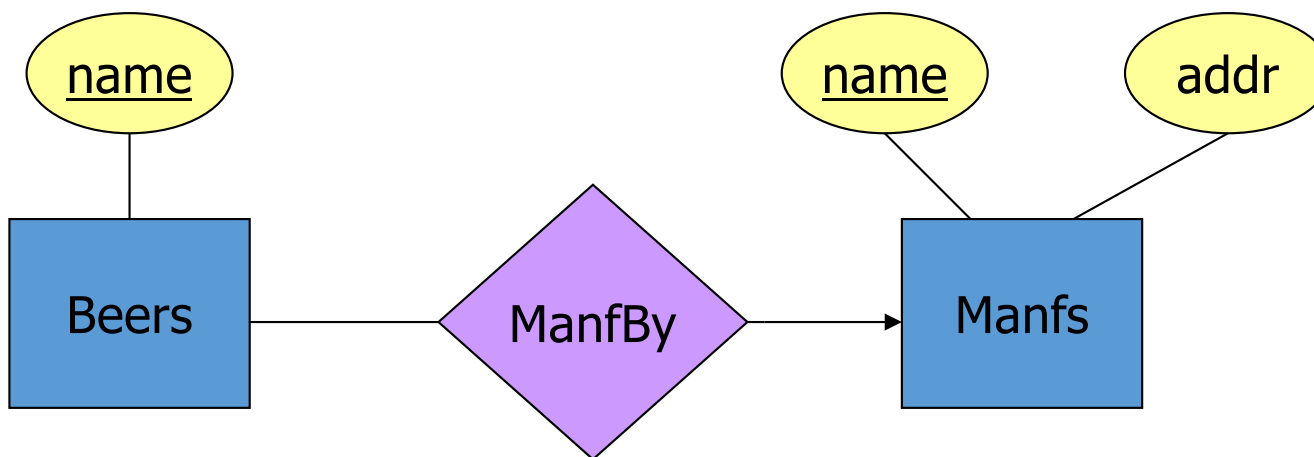




# ER模型建模

## ■ 一些建模准则

- ✓ 能抽象为属性的，不要抽象为实体



Good!

*Manfs* 具有一个非键属性 *addr*，所以它应该被抽象为实体

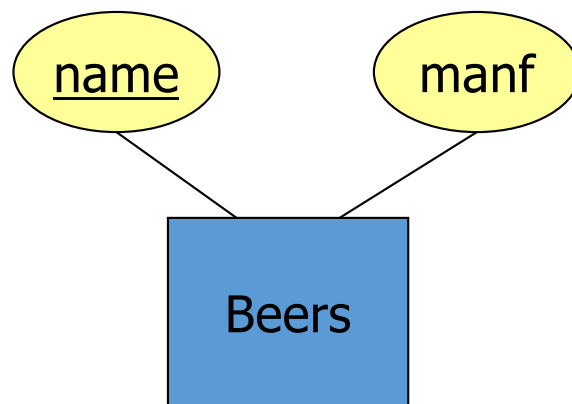
*Beers* 在一对多联系 *ManfBy* 的多端，所以它也应该被抽象为实体



# ER模型建模

## ■ 一些建模准则

- ✓ 能抽象为属性的，不要抽象为实体



Good !

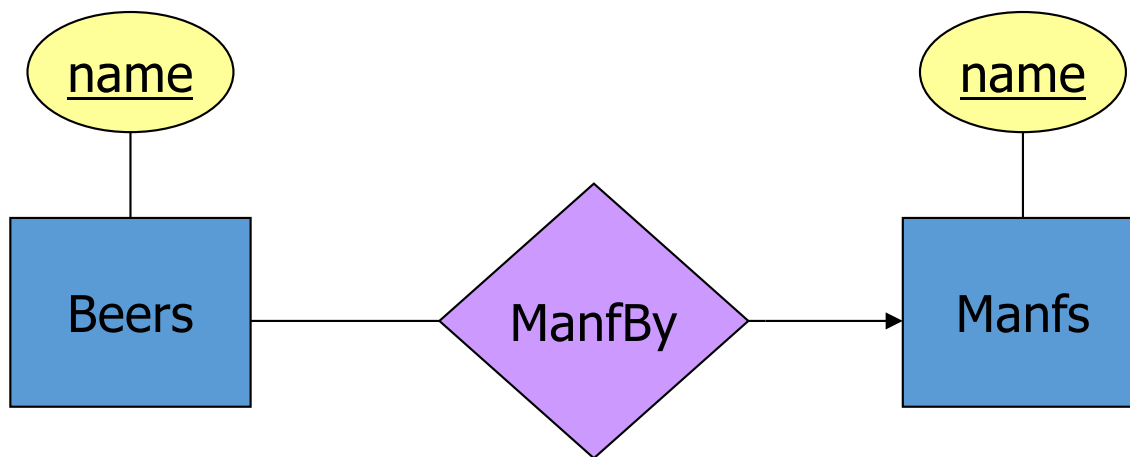
没必要把manufacture抽象为实体，因为它既没有非键属性，也不处于联系中



# ER模型建模

## ■ 一些建模准则

- ✓ 能抽象为属性的，不要抽象为实体



Bad !

Manufacture 只有一个名字属性，也不处于多端，所以不应该是一个实体。

它可以被简化为 **Beers** 的一个属性！

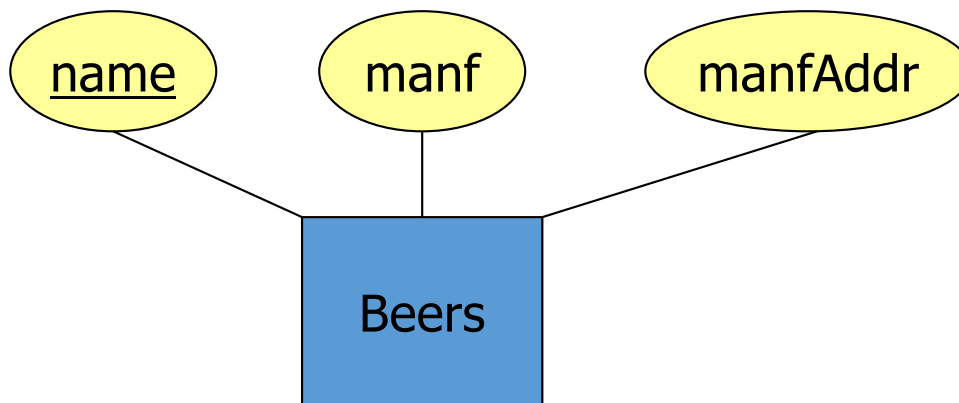


# ER模型建模

## ■ 一些建模准则

- ✓ 能抽象为属性的，不要抽象为实体

反过来，应该是实体的，也不要强行拆解为属性





# ER模型小结

---

- ER模型是人们认识客观世界的一种方法、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。



# ER模型小结

---

- ER模型的设计过程，基本上三步：
  - ✓先设计实体类型（找出逻辑独立的事物，即管理对象）；
  - ✓再设计联系类型（确定实体之间相互影响的关系）；
  - ✓最后为实体和联系确定属性。
- 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。



# 本章习题

---

- 课本第34页：7、8题
- 课本第241页：7、8题
- ER模型例题.pptx
- 提交时间：下周上课前