

课程名称: 数据库系统

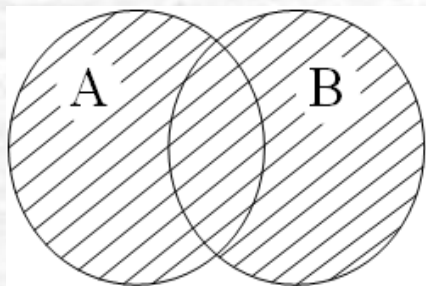
---

# 关系模型的其它相关知识

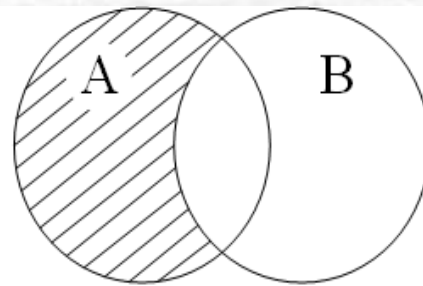
单位: 重庆大学计算机学院



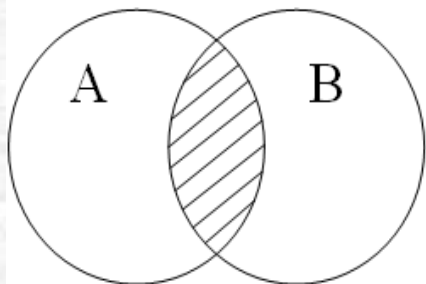
# 集合的基本操作



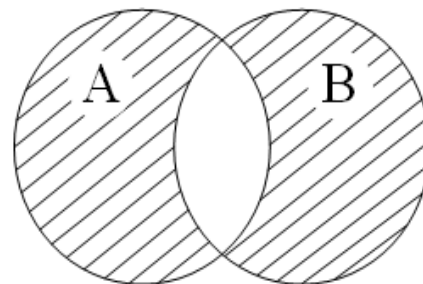
**$A \cup B$**



**$A - B$**



**$A \& B$**



**$A \wedge B$**

# 主要学习目标

- 基本的关系代数运算  
并、差、积、选择、投影
- 扩展的关系代数运算  
自然连接、外连接



# 前测小问题

关系模型需要什么操作？



# 一 关系代数

## 1.1 基本的关系代数运算

基本关系代数指  
哪几种运算，为  
何叫基本？

- 包含六种运算
- select:  $\sigma$  (选择)
- project:  $\pi$  (投影)
- union:  $\cup$  (并)
- set difference:  $-$  (差)
- Cartesian product:  $\times$  (笛卡尔积)
- rename:  $\rho$  (重命名)

(上节课已介绍，简略讨论和总结)

# 一 关系代数

## 1.1 基本的关系代数运算

n Relation r

A	B	C	D
$\alpha$	$\alpha$	1	7
$\alpha$	$\beta$	5	7
$\beta$	$\beta$	12	3
$\beta$	$\beta$	23	10



A	B	C	D
$\alpha$	$\alpha$	1	7
$\beta$	$\beta$	23	10

- $\sigma_{A=B \wedge D > 5}(r)$

# 一 关系代数

## 1.1 基本的关系代数运算

n Relation r

<i>A</i>	<i>B</i>	<i>C</i>
$\alpha$	10	1
$\alpha$	20	1
$\beta$	30	1
$\beta$	40	2



<i>A</i>	<i>C</i>
$\alpha$	1
$\beta$	1
$\beta$	2

- $\pi_{A, C}(r)$



# 一 关系代数

## 1.1 基本的关系代数运算

$r:$

$A$	$B$
$\alpha$	1
$\alpha$	2
$\beta$	1

$r$

$s:$

$A$	$B$
$\alpha$	2
$\beta$	3

$s$



$A$	$B$
$\alpha$	1
$\alpha$	2
$\beta$	1
$\beta$	3

•  $r \bowtie s$

# 一 关系代数

## 1.1 基本的关系代数运算

<i>A</i>	<i>B</i>
$\alpha$	1
$\alpha$	2
$\beta$	1

*r*

<i>A</i>	<i>B</i>
$\alpha$	2
$\beta$	3

*s*

■  $r - s:$

<i>A</i>	<i>B</i>
$\alpha$	1
$\beta$	1

# 一 关系代数

## 1.1 基本的关系代数运算

<i>A</i>	<i>B</i>
$\alpha$	1
$\beta$	2

*r*

*s*:

<i>C</i>	<i>D</i>	<i>E</i>
$\alpha$	10	<i>a</i>
$\beta$	10	<i>a</i>
$\beta$	20	<i>b</i>
$\gamma$	10	<i>b</i>

*S*



*r* x *s*:

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>
$\alpha$	1	$\alpha$	10	<i>a</i>
$\alpha$	1	$\beta$	10	<i>a</i>
$\alpha$	1	$\beta$	20	<i>b</i>
$\alpha$	1	$\gamma$	10	<i>b</i>
$\beta$	2	$\alpha$	10	<i>a</i>
$\beta$	2	$\beta$	10	<i>a</i>
$\beta$	2	$\beta$	20	<i>b</i>
$\beta$	2	$\gamma$	10	<i>b</i>

# 一 关系代数

## 1.1 基本的关系代数运算

$r \times s$ :

$A$	$B$	$C$	$D$	$E$
$\alpha$	1	$\alpha$	10	$a$
$\alpha$	1	$\beta$	10	$a$
$\alpha$	1	$\beta$	20	$b$
$\alpha$	1	$\gamma$	10	$b$
$\beta$	2	$\alpha$	10	$a$
$\beta$	2	$\beta$	10	$a$
$\beta$	2	$\beta$	20	$b$
$\beta$	2	$\gamma$	10	$b$

故  $\sigma_{A=C}(r \times s)$ :

$A$	$B$	$C$	$D$	$E$
$\alpha$	1	$\alpha$	10	$a$
$\beta$	2	$\beta$	10	$a$
$\beta$	2	$\beta$	20	$b$



## 1.2 基本运算的复合

## 一 关系代数

### 案例1

可以，因为结果仍为关系 (基本运算的查询能力已很强)

根据instructor找出最高的工资

Step 1: find instructor salaries that are less than some other instructor salary (i.e. not maximum)

using a copy of *instructor* under a new name *d*

$\sigma_{instructor.salary < d.salary} (instructor \times d)$

(*instructor*  $\times$  *d*)

Step 2: Find the largest salary

$\sigma_{salary = \max_{salary} (instructor)}$

$\sigma_{instructor.salary < d.salary} (instructor \times d)$

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

\* 基本运算可用  
复合使用吗?

找出非最大工资的工资集合!  
(如: 因  $instructor.90000 < d.92000$   
故90000在此集合中)

\* 这两个复合运  
算的结果是什

找出最大工资! (95000)

关系代数是一种过程化的查询语言(为什么这么说?)



## 案例2

Find the names of all instructors in the Physics department, along with the *course\_id* of all courses they have taught

(找出物理学院的老师ID和他们所教授的课程ID)

Query 1 (方式1)

$\square instructor.ID, course\_id (\square dept\_name$

$\square instructor.ID=teaches.ID (instructor \times teaches)$

Query 2 \* (方式2)

$\square instructor.ID, course\_id (\square instructor.ID$

$\square dept\_name="Physics" (instructor) \times teaches$

instructor

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

teaches

ID	course_id	sec_id	semester	year
10101	CS-101	1	Fall	2009
10101	CS-315	1	Spring	2010
10101	CS-347	1	Fall	2009
12121	FIN-201	1	Spring	2010
15151	MU-199	1	Spring	2010
22222	PHY-101	1	Fall	2009
32343	HIS-351	1	Spring	2010
45565	CS-101	1	Spring	2010
45565	CS-319	1	Spring	2010
76766	BIO-101	1	Summer	2009
76766	BIO-301	1	Summer	2010
83821	CS-190	1	Spring	2009
83821	CS-190	2	Spring	2009
83821	CS-319	2	Spring	2010
98345	EE-181	1	Spring	2009

\* 这两个复合运算的结果是什

**[例]设教学数据库中有3个关系:**

**学生关系 S(SNO,SNAME,AGE,SEX)**

**学习关系 SC(SNO,CNO,GRADE)**

**课程关系 C(CNO,CNAME,TEACHER)**

下面用关系代数表达式表达每个查询语句。

- (1) 检索学习课程号为**C2**的学生学号与成绩。
- (2) 检索学习课程号为**C2**的学生学号与姓名
- (3) 检索选修课程名为**MATHS**的学生学号与姓名。
- (4) 检索选修课程号为**C2**或**C4**的学生学号。

何谓交运算，  
结果是什么？

### (一) 集合交运算

(Set Intersection Operation)

- 案例3  
:

A	B
□	1
□	2
□	1

$r$

A	B
□	2
□	3

$s$

$r \cap s$

A	B
□	2

保留相同元组(仍为关系)!

- 形式化定义:  $r \cap s = \{ t \mid t \in r \text{ and } t \in s \}$
- **限制条件(Assume)?**
- $r, s$  have the *same arity* (相同属性数量)
- attributes of  $r$  and  $s$  are compatible (属性相匹配)

增强查询  
操作能  
力?

不会增强, 因为:

$$r \cap s = r - (r - s)$$



何谓自然连接运算?

## (二) 自然连接运算 (Natural Join Operation)

n

案例4:

A	B	C	D
	1		a
	2		a
	4		b
	1		a
	2		b

$r$

B	D	E
1	a	
3	a	
1	a	
2	b	
3	b	

$s$

$r \bowtie s$ :

A	B	C	D	E
	1		a	
	1		a	
	1		a	
	1		a	
	2		b	

消除重复连接属性!

n 形式化定义:

元组按属性值相同粘贴(仍为关系)!

Let  $r$  and  $s$  be relations on schemas  $R$  and  $S$  respectively.

Then,  $r \bowtie s$  is a relation on schema  $R \sqcup S$  (属性集合的并) obtained as follows:

- Consider each pair of tuples  $tr$  from  $r$  and  $ts$  from  $s$ .
- If  $tr$  and  $ts$  have the same value on each of the attributes in  $R \cap S$ , add a tuple  $t$  to the result, where
  - $t$  has the same value as  $tr$  on  $r$
  - $t$  has the same value as  $ts$  on  $s$

增强查询操作能力吗?

不会增强(用例示说明):

假设:  $R = (A, B, C, D)$ ,  $S = (E, B, D)$ , 则有:

$$r \bowtie s = \pi_{r.A, r.B, r.C, r.D, s.E} (\sigma_{r.B = s.B \wedge r.D = s.D} (r \times s))$$

\*何谓赋值运算?

## (三) \*赋值运算 (Assignment Operation)

- The assignment operation ( $\leftarrow$ ) provides a convenient way to express **complex queries**.
  - Write query as a sequential program consisting of
    - a series of assignments
    - followed by an expression whose value is displayed as a result of the query.
  - Assignment must always be made to a **temporary relation variable**.
  - 案例5:** Write  $r \bowtie s$  as :
    - $temp1 \leftarrow R-S(r);$
    - $temp2 \leftarrow R-S((temp1 \times s) - R-S, S(r));$
    - $result = temp1 - temp2$
  - The result to the right of the  $\leftarrow$  is assigned to the relation variable on the left of the  $\leftarrow$ .
  - May use variable in subsequent expressions.
- 运算结果可赋值给临时关系变量(仍为关系), 并且可继续参与运算!

增强查询操作能力吗?

不会增强(用例示说明)! 因为对关系 $r$ 和 $s$ 有 $r \bowtie s = R-S(r) - (R-S((R-S(r) \times s) - R-S, S(r)))$



有几种外连接运算，与自然连接的区别？

增强查询操作能力吗？

## (四) 外连接运算 (Outer Join)

*Instructor1:*

<i>ID</i>	<i>name</i>	<i>dept_name</i>
10101	Srinivasan	Comp. Sci.
12121	Wu	Finance
15151	Mozart	Music

*teaches1:*

<i>ID</i>	<i>course_id</i>
10101	CS-101
12121	FIN-201
76766	BIO-101

案例

6

n Join(自然连接):

*instructor*  $\bowtie$  *teaches*

n Left Outer Join(左外连接):

*instructor*  $\leftarrow$  *teaches*

n Right Outer Join(右外连接):

*instructor*  $\rightarrow$  *teaches*

n Full Outer Join(全外连接):

*instructor*  $\bar{\bowtie}$  *teaches*

<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>course_id</i>
10101	Srinivasan	Comp. Sci.	CS-101
12121	Wu	Finance	FIN-201

<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>course_id</i>
10101	Srinivasan	Comp. Sci.	CS-101
12121	Wu	Finance	FIN-201
15151	Mozart	Music	null

<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>course_id</i>
10101	Srinivasan	Comp. Sci.	CS-101
12121	Wu	Finance	FIN-201
76766	null	null	BIO-101

<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>course_id</i>
10101	Srinivasan	Comp. Sci.	CS-101
12121	Wu	Finance	FIN-201
15151	Mozart	Music	null
76766	null	null	BIO-101

查询能力不会增强！

$$s = (r \bowtie s) \cup (r - \Pi R(r \bowtie s) \times \{(null, \dots, null)\})$$

### (一) 广义投影运算 (Generalized Projection)

何谓广义投影运算?

- 形式化定义:  $\Pi_{F_1, F_2, \dots, F_n}(E)$

It allows **arithmetic functions** to be used in the projection list.

- $E$  is any relational-algebra expression
- Each of  $F_1, F_2, \dots, F_n$  are arithmetic expressions involving constants and attributes in the schema of  $E$ .

与投影不同在于: 允许属性表达式(仍为关系)!

#### 案例7

给定关系 *instructor*(*ID*, *name*, *dept\_name*, *salary*) , *salary* 是年薪, 如何获得月薪呢?

$\Pi_{ID, name, dept\_name, salary/12}(instructor)$

增强查询操作能力吗?

不能采用基本关系代数运算来表达!

### 聚集函数运算的形式化定义

- Aggregation function(聚集函数) takes a collection of values and returns a single value as a result. (计算一个集合的数值, 得到一个值做为结果)
  - avg: average value – 求平均值
  - min: minimum value – 求最小值
  - max: maximum value – 求最大值
  - sum: sum of values – 求和
  - count: number of values – 求元组数
- Aggregate operation: in relational algebra  
$$G_1, G_2, \dots, G_n \quad F_1(A_1), F_2(A_2), \dots, F_n(A_n) \quad (E)$$
  - $E$  is any relational-algebra expression
  - $G_1, G_2, \dots, G_n$  is a list of attributes on which **to group** (can be empty)
  - Each  $F_i$  is an **aggregate function**
  - Each  $A_i$  is an **attribute name**



# 1.4 扩展的关系代数运算

## (二) 聚集函数运算

### (Aggregate Operation Example)

聚集函数的作用(如sum,avg为例)?

- Relation  $r$ :

A	B	C
		7
		7
		3
		10

求和!

$g \text{ sum}(c) (r):$

sum(c)
27

avg  
Min  
Max  
sum  
count

- 找出每个部门的平均工资

$dept\_name \text{ gavg}(salary) (instructor)$

dept_name	
Biology	72000
Comp. Sci.	77333
Elec. Eng.	80000
Finance	85000
History	61000
Music	40000
Physics	91000

分组求平均值!

ID	name	dept_name	salary
76766	Crick	Biology	72000
45565	Katz	Comp. Sci.	75000
10101	Srinivasan	Comp. Sci.	65000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000
12121	Wu	Finance	90000
76543	Singh	Finance	80000
32343	El Said	History	60000
58583	Califieri	History	62000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
22222	Einstein	Physics	95000

- Result of aggregation does not have a name
- Can use rename operation to give it a name
- For convenience, we permit renaming as part of aggregate operation

案例  
8

$dept\_name \text{ gavg}(salary) \text{ as avg\_sal} (instructor)$

(形式化定义见下页)

## 二 基本E-R图到关系模式的转换

### 2.1 关系数据库设计的起点

一个应用的关系模式集合应当如何得到?

三种可选方式:

- 1) 直接与用户交互, 发现数据对象, 形成关系模式集合  
(适合于原型开发方法, 但需与用户不断交互, 工作量巨大)
- 2) 根据概念模型设计结果(E-R图&数据字典), 形成关系模式集合  
(方便且高效, 避免与用户交互, 适合与周期开发方法)
- 3) 根据面向对象方法的分析结果 (UML类图), 形成关系模式集合  
(优点类同上一方法)

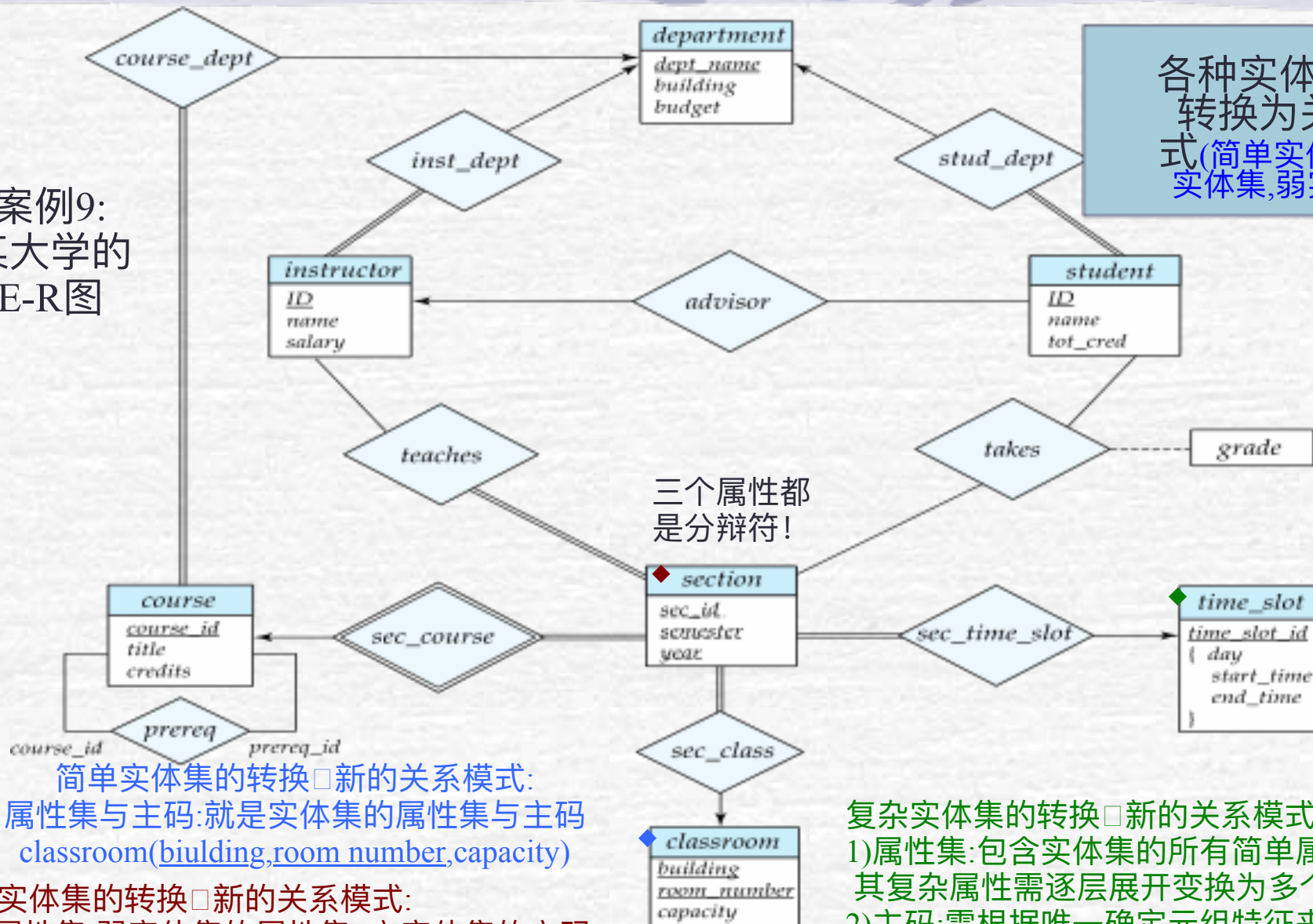
(下面主要介绍第2种方法, 从E-R图转换得当关系模式)



## 2.2 实体集的转换

## 二 基本E-R图到关系模式的转换

案例9:  
某大学的  
E-R图



简单实体集的转换 □ 新的关系模式:  
属性集与主码: 就是实体集的属性集与主码  
**classroom**(building, room number, capacity)

弱实体集的转换 □ 新的关系模式:  
1) 属性集: 弱实体集的属性集 + 主实体集的主码  
2) 主码: 主实体集的主码 + 弱实体集的分辩符  
**section**(course\_id, sec\_id, semester, year)

复杂实体集的转换 □ 新的关系模式(1NF):  
1) 属性集: 包含实体集的所有简单属性, 此外其复杂属性需逐层展开变换为多个简单属性  
2) 主码: 需根据唯一确定元组特征来确定 **time\_slot** (详见下页)

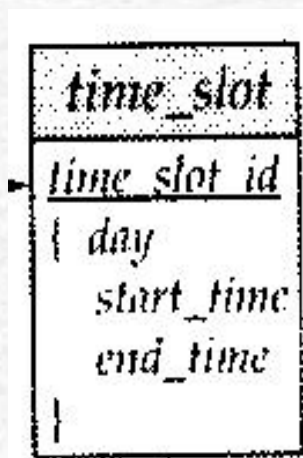
### 复杂实体集转换示例

该复杂实体集如何转换为关系模式，理由？

复杂属性time\_slot\_id:纵向展开+横向展开!

time\_slot实体集□新的关系模式:

time\_slot(time\_slot\_id, day, start\_time, end\_time)



注释:

纵向展开(多值属性□多个元组):1层day和2层start\_time  
(第1层: 每个元组表示一天)

(第2层: 每个元组表示一个时段)

横向展开(结构属性□多个属性):day,start\_time,end\_time

例如该元组表示: A时段, 周一, 8:00-8:50

<u>time_slot_id</u>	<u>day</u>	<u>start_hr</u>	<u>start_min</u>	<u>end_hr</u>	<u>end_min</u>
A	M	8	0	8	50
A	W	8	0	8	50
A	F	8	0	8	50
B	M	9	0	9	50
B	W	9	0	9	50
B	F	9	0	9	50
C	M	11	0	11	50
C	W	11	0	11	50
C	F	11	0	11	50
D	M	13	0	13	50
D	W	13	0	13	50
D	F	13	0	13	50
E	T	10	30	11	45
E	R	10	30	11	45
F	T	14	30	15	45
F	R	14	30	15	45
G	M	16	0	16	50
G	W	16	0	16	50
G	F	16	0	16	50
H	W	10	0	12	30

图 A-14 具有起始和终止时间被拆分成小时和分钟的 time\_slot 关系



## 2.3 联系集的转换

## 二 基本E-R图到关系模式的转换

多对多联系 □ 新的关系模式:

属性集: 两方主码+联系的属性; 主码: 两方主码

takes(ID, course\_id, secid, semester, year, grade)

prereq(course\_id, prereq\_id)

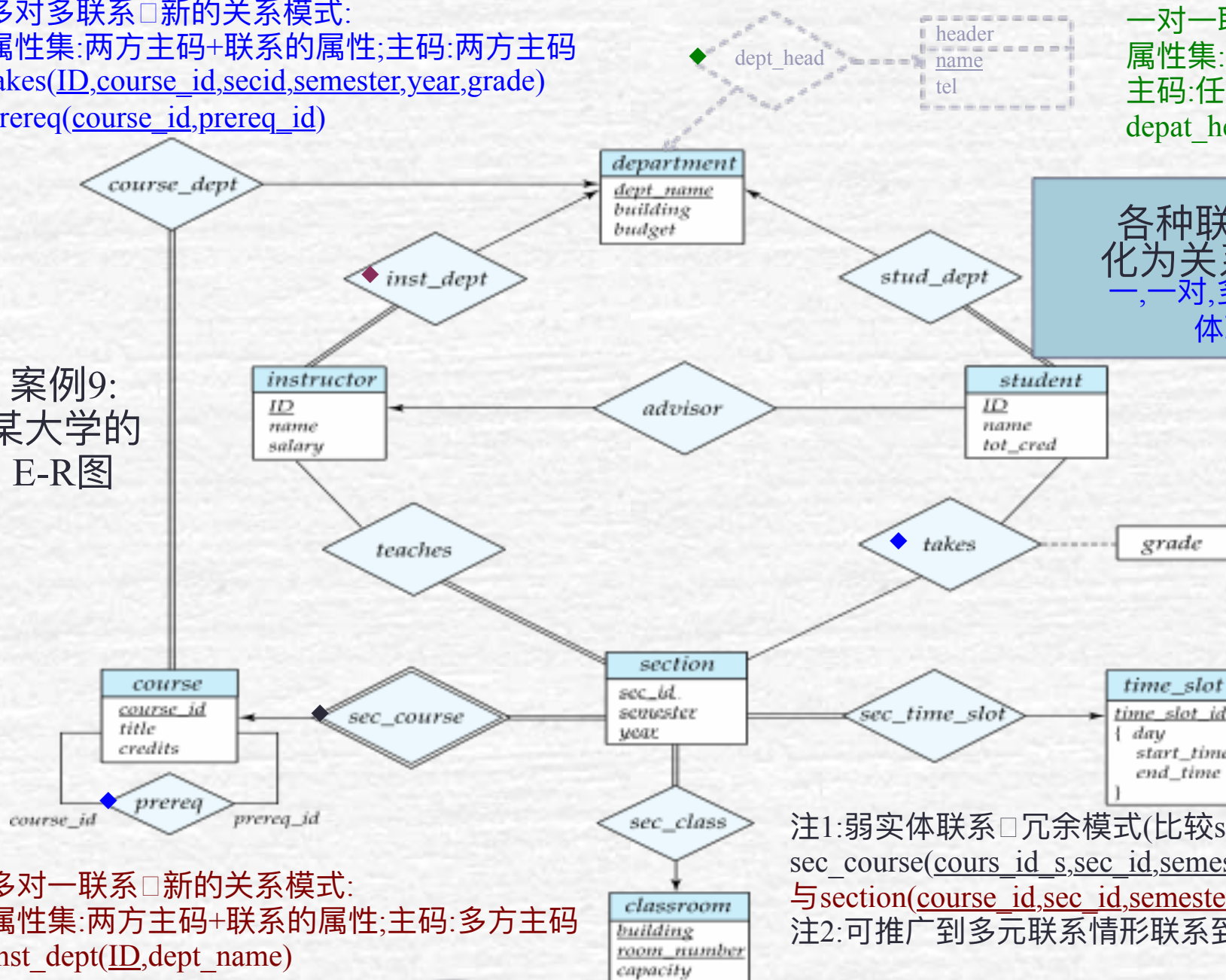
一对一联系 □ 新的关系模式:

属性集: 两方主码+联系的属性

主码: 任一方的主码均可

dept\_head(dept\_name, name)

案例9:  
某大学的  
E-R图



各种联系如何转  
化为关系模式(一对  
一, 一对, 多对多, 弱实  
体联系)?

多对一联系 □ 新的关系模式:

属性集: 两方主码+联系的属性; 主码: 多方主码

inst\_dept(ID, dept\_name)

注1: 弱实体联系 □ 冗余模式(比较section), 可忽略掉!  
sec\_course(course\_id\_s, sec\_id, semester, year, course\_id\_c);

与section(course\_id, sec\_id, semester, year)冗余

注2: 可推广到多元联系情形联系到关系模式的转换

# 三 关系模式合并与约束表示

## 3.1 模式合并(优化)

由实体集转换得到的关系模式

```
classroom(building, room_number, capacity)
department(dept_name, building, budget)
course(course_id, title, dept_name, credits)
instructor(ID, name, dept_name, salary) i_id
section(course_id, sec_id, semester, year, building, room_number, time_slot_id)
teaches(ID, course_id, sec_id, semester, year)
student(ID, name, dept_name, tot_cred) s_id
takes(ID, course_id, sec_id, semester, year, grade) s_id
advisor(s_ID, i_ID)
time_slot(time_slot_id, day, start_time, end_time)
prereq(course_id, prereq_id)
```

注: Teaches和takes不能合并!

```
i_id teaches (ID, course_id, sec_id, semester, year)
s_id takes (ID, course_id, sec_id, semester, year, grade)
prereq (course_id, prereq_id)
advisor (s_ID, i_ID)
sec_course (course_id, sec_id, semester, year)
sec_time_slot (course_id, sec_id, semester, year, time_slot_id)
sec_class (course_id, sec_id, semester, year, building, room_number)
inst_dept (ID, dept_name) i_id
stud_dept (ID, dept_name) s_id
course_dept (course_id, dept_name)
```

由联系集转换得到的关系模式

合并得到的关系模式

```
classroom(building, room_number, capacity)
department(dept_name, building, budget)
course(course_id, title, dept_name, credits)
instructor(ID, name, dept_name, salary) i_id
section(course_id, sec_id, semester, year, building, room_number, time_slot_id)
teaches(ID, course_id, sec_id, semester, year) i_id
student(ID, name, dept_name, tot_cred) s_id
takes(ID, course_id, sec_id, semester, year, grade) s_id
advisor(s_ID, i_ID)
time_slot(time_slot_id, day, start_time, end_time)
prereq(course_id, prereq_id)
```

注:这样,二者还可合并

转换得到的关系模式需要合并吗, 合并原则?

合并主码相同的关系模式, 因这些属性均由相同主码唯一确定! (理由见第8章中模式优化)

关系模式如何区分全参与(描述双线)与部分参与特征?

◆是否允许属性取空值:  
全参与: not NULL-非空约束!

案例  
10



## 3.2 特别说明

## 三 关系模式合并与约束表示

由实体集转换得到的关系模式

```
classroom(building, room_number, capacity)
department(dept_name, building, budget)
course(course_id, title, dept_name, credits)
instructor(ID, name, dept_name, salary) i_id
section(course_id, sec_id, semester, year, building, room_number, time_slot_id)
teaches(ID, course_id, sec_id, semester, year)
student(ID, name, dept_name, tot_cred) s_id
takes(ID, course_id, sec_id, semester, year, grade) s_id
advisor(s_ID, i_ID)
time_slot(time_slot_id, day, start_time, end_time)
prereq(course_id, prereq_id)
```

注: Teaches和takes不能合并!

```
i_id teaches (ID, course_id, sec_id, semester, year)
s_id takes (ID, course_id, sec_id, semester, year, grade)
prereq (course_id, prereq_id)
advisor (s_ID, i_ID)
sec_course (course_id, sec_id, semester, year)
sec_time_slot (course_id, sec_id, semester, year, time_slot_id)
sec_class (course_id, sec_id, semester, year, building, room_number)
inst_dept (ID, dept_name) i_id
stud_dept (ID, dept_name) s_id
course_dept (course_id, dept_name) d_id
```

由联系集转换得到的关系模式

合并得到的关系模式

```
classroom(building, room_number, capacity)
department(dept_name, building, budget)
course(course_id, title, dept_name, credits)
instructor(ID, name, dept_name, salary) i_id
section(course_id, sec_id, semester, year, building, room_number, time_slot_id)
teaches(ID, course_id, sec_id, semester, year) i_id
student(ID, name, dept_name, tot_cred) s_id
takes(ID, course_id, sec_id, semester, year, grade) s_id
advisor(s_ID, i_ID) ♦
time_slot(time_slot_id, day, start_time, end_time)
prereq(course_id, prereq_id)
```

注:二者还可合并

转换后的关系模式,应确定哪些参照约束?

1) 从上合并过程可以看出:  
一对多 (及一对一) 联系都不需要单独增加关系模式,

只需要在多方实体集中添加一方主码和联系的属性即可

2) 参照完整性约束:  
涉及所有因描述联系增加的主码属性上!

a) 因多对一联系而增加的一方主码属性:  
(即蓝色虚线属性)

b) 多对多联系得到模式的两方主码属



# 四 扩展E-R图的转换

## 4.1 特殊化与概化的转换

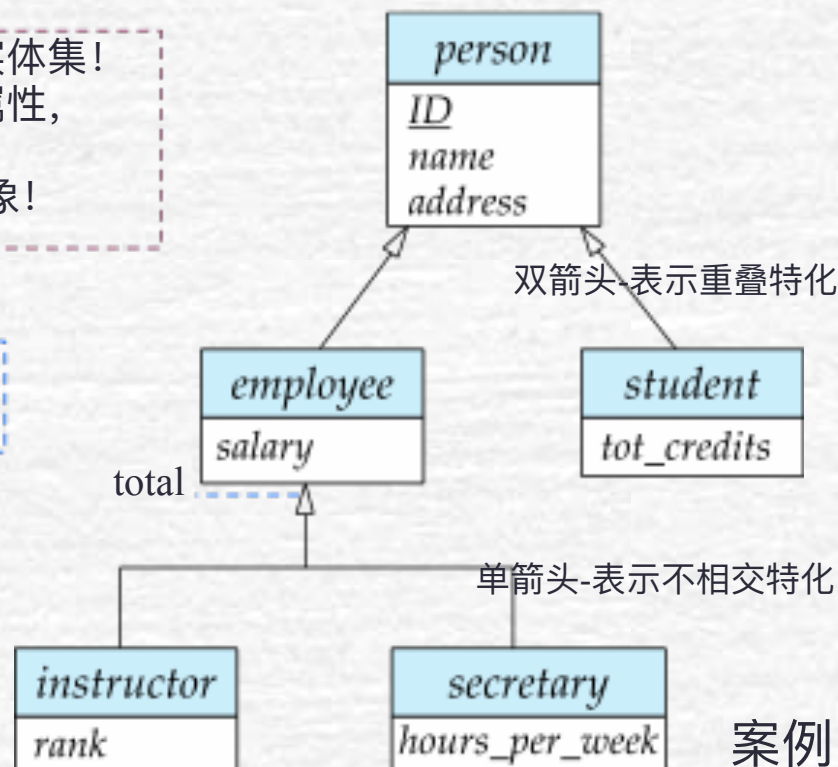
一个实体集(父)可能包含一个子实体集!  
子实体集具有(继承)父实体集的属性,  
还可能具有其它一些独特属性。  
特化与概化专门用于描述这种现象!

如何采用关系  
模式描述特化  
与概化现象?

特化: 自动向下的设计过程  
概化: 自底向上的设计过程

由下向上的箭头连线  
为一种‘is a’联系

部分概化: 允许父实体不属于任何子实体集  
(缺省表示)  
全部概化: 每个父实体必属于某一子实体集  
(采用标识total)



案例  
11

方法1(父、子实体集各用1个关系模式)

person(ID,name,street,city)

employee(ID,salary)

student(ID,tot\_credits)

不足:查询employee信息涉及多表连接

方法2(当概化不相交且完全时, 仅子实体集各用1个关系模式(父实体元组分属于两关系))

employee(ID,name,street,city,salary)

student(ID,name,street,city,tot\_credits)

不足:如另一关系模式R有参照person的ID约束时,  
无法支持, 需增加一个person关系模式(冗余大)

## 4.2 聚集的转换

## 四 扩展E-R图的转换

聚集：是一种抽象！  
将联系集(及其相关实体集)，  
看着是一个更高层的抽象实体集。

例如，抽象的proj\_guide实体集：  
包含了project\_guide联系集P\_I\_S以及  
instructor, student, project 实体集。  
看着是一个具有复杂构造的实体集！

故：在抽象实体集P\_I\_S与实体集之间，  
可以有(多对多)联系集eval\_for。

如何采用关  
系模式描述  
聚集现象？

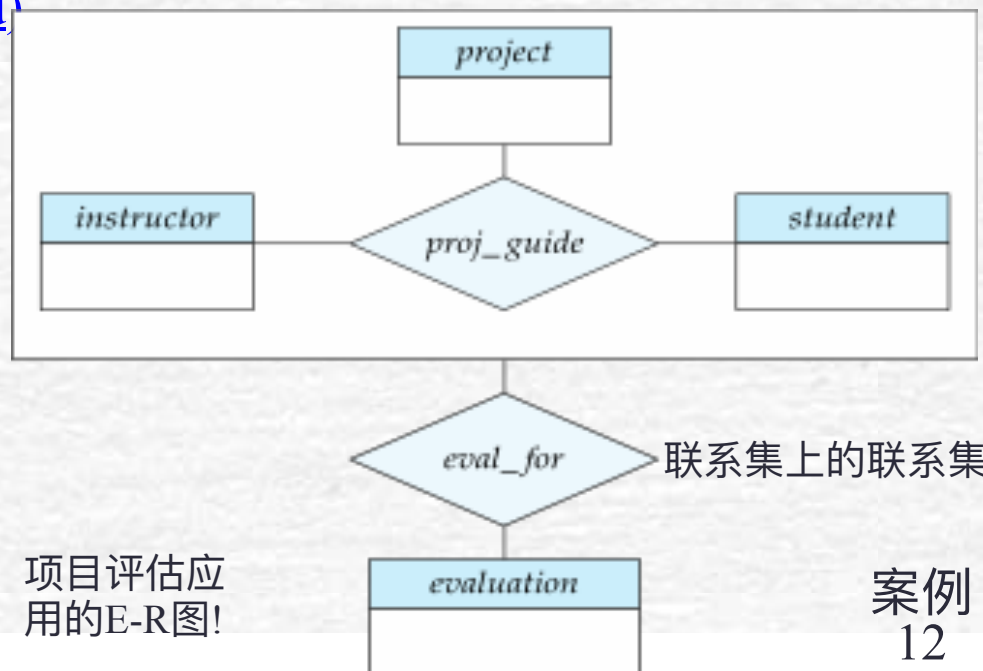
示例：project\_guide抽象联系集可表示为：

P\_I\_S(instructor-id, student-id, project-id)

evaluation 实体集表示为：EV(evaluation-id, ...)

故eval\_for聚集联系集可表示为：

eval\_for(instructor-id, student-id, project-id, evaluation-id)



### 7.8.6.2 聚集的表示

为包含聚集的 E-R 图进行模式设计是很直接的。考虑图 7-23 中的 E-R 图。表示聚集 *proj\_guide* 和实体集 *evaluation* 之间联系的联系集 *eval\_for* 的模式包含对应实体集 *evaluation* 主码中的每个属性以及联系集 *proj\_guide* 主码中的每个属性。它还包含对应于联系集 *eval\_for* 的任意描述性属性(如果存在的话)。然后，根据我们已经定义的规则，在聚集的实体集中转换联系集和实体集。

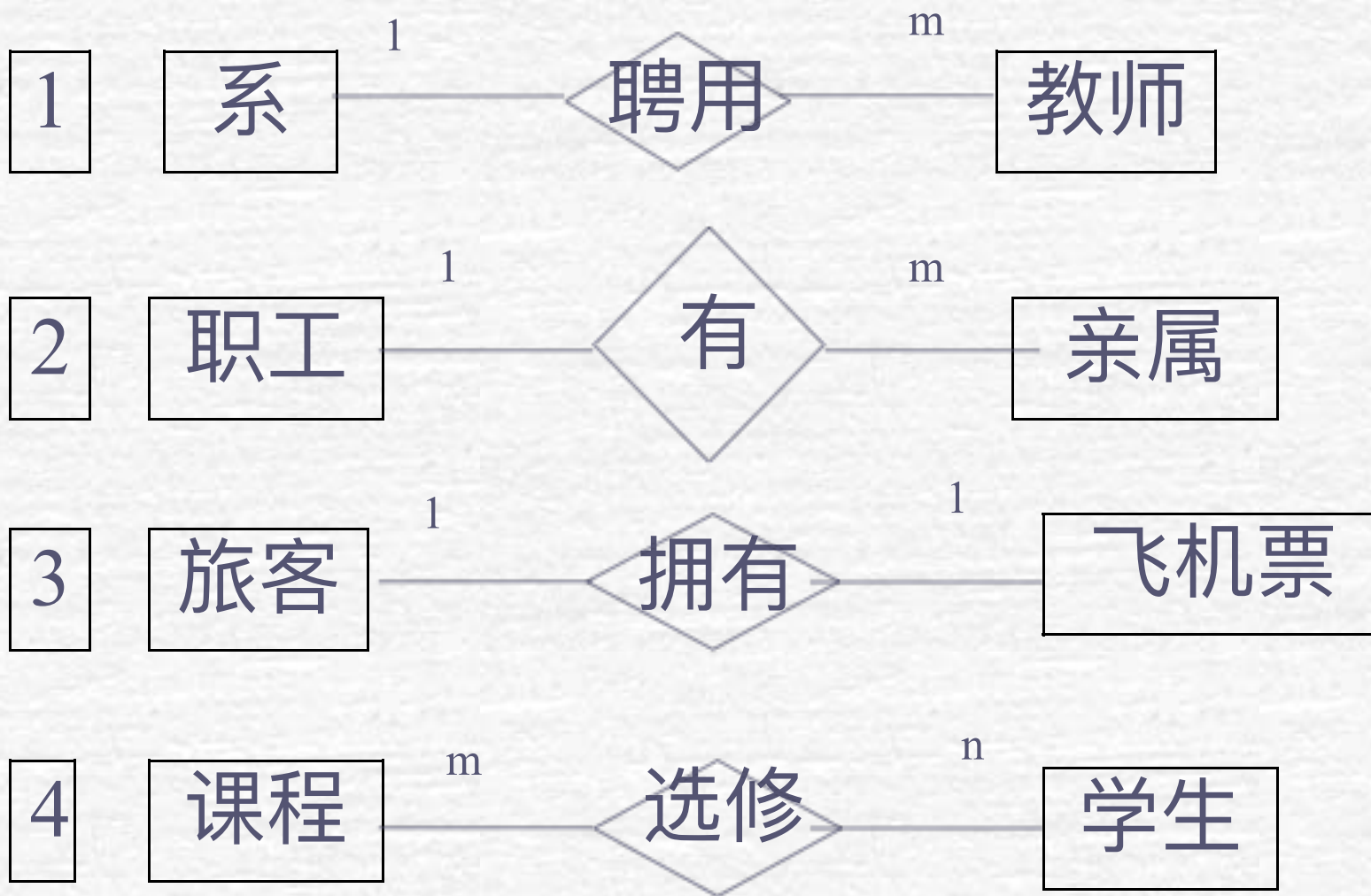
# ER图转换为关系模式的规则

- 每个实体建一个关系，属性转为关系的属性，关键属性为主键
- 联系转换
  - 1:M  
1方主键出现在M方成为外键
  - 1:1  
任意一方主键出现在另一方中，成外键
  - M:N  
联系建为一新关系，其主键由两个父实体的主键复合组成



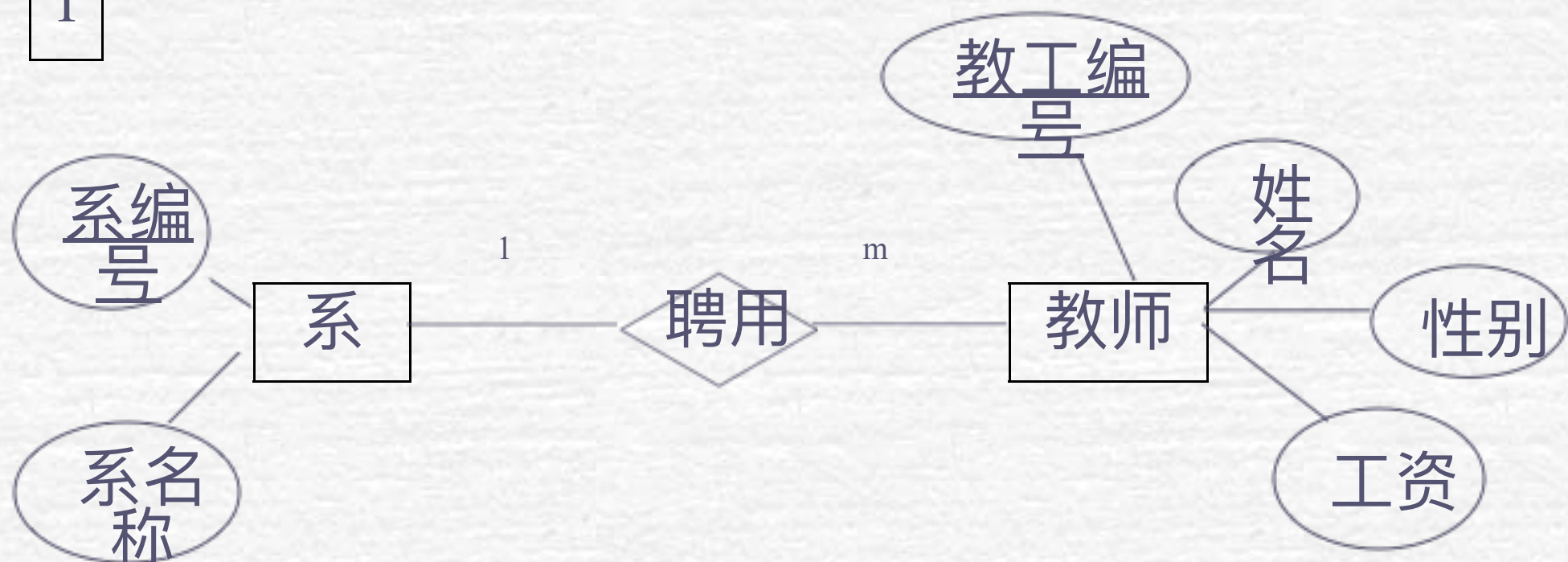
E-R 图

关系模式





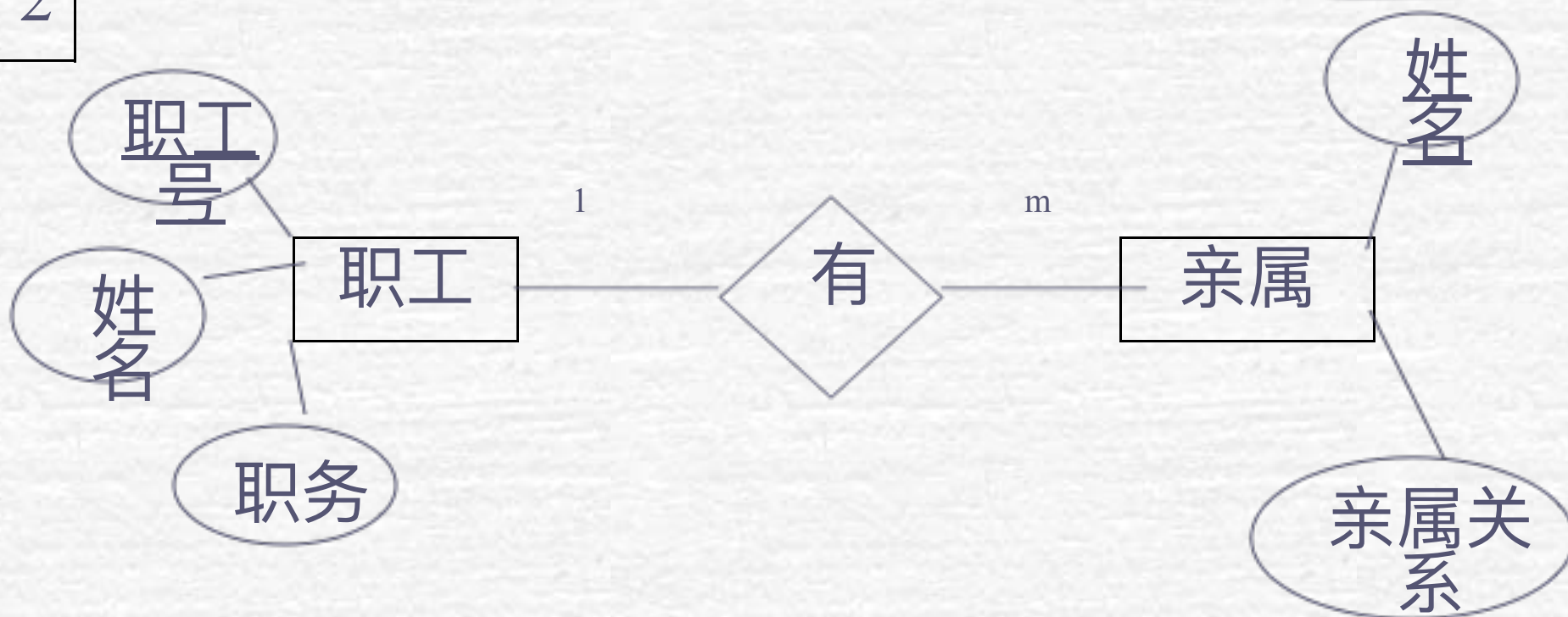
1



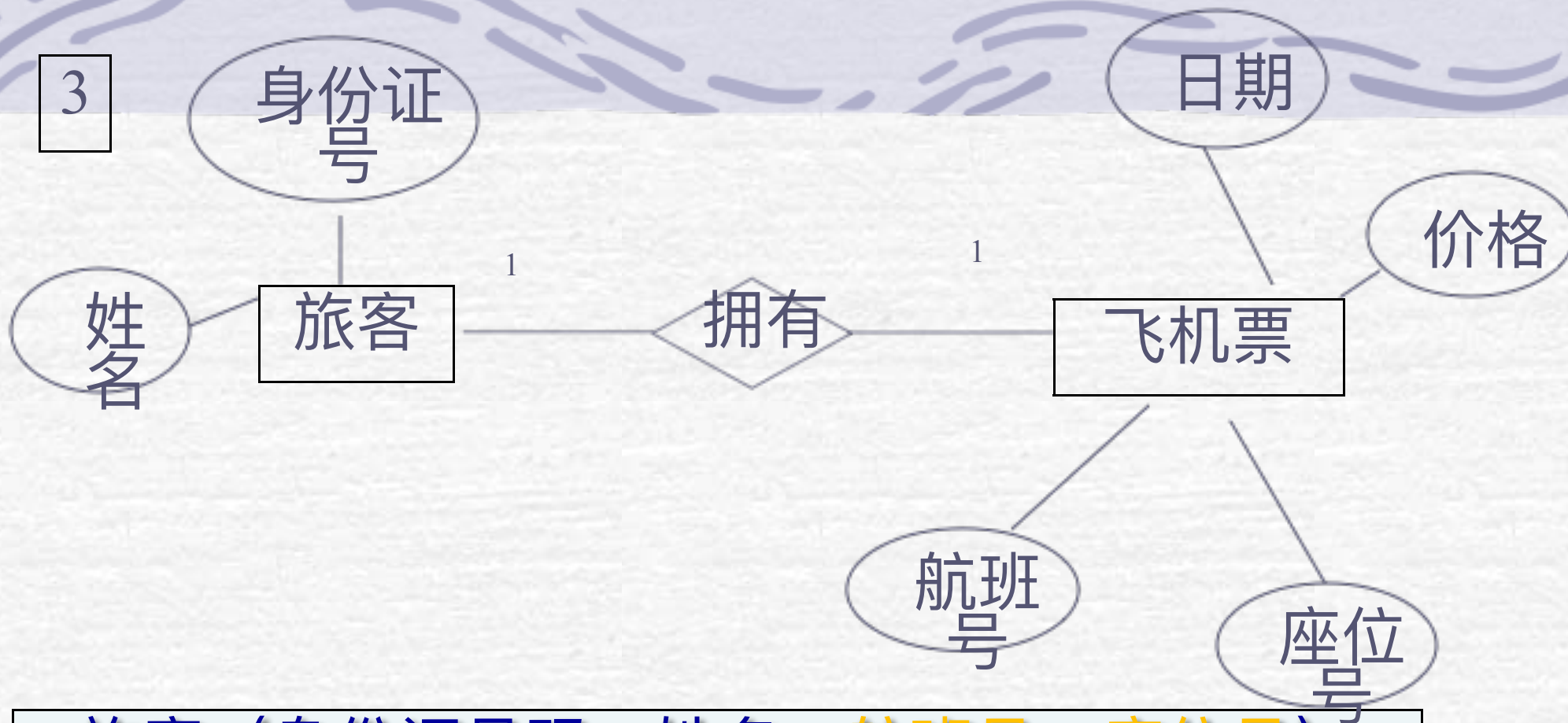
系（系编号，系名称）

教师（教工编号，姓名，性别，工资，系编号）

2



职工 (职工号, 姓名, 职务)  
亲属 (亲属姓名, 亲属关系, 职工号)

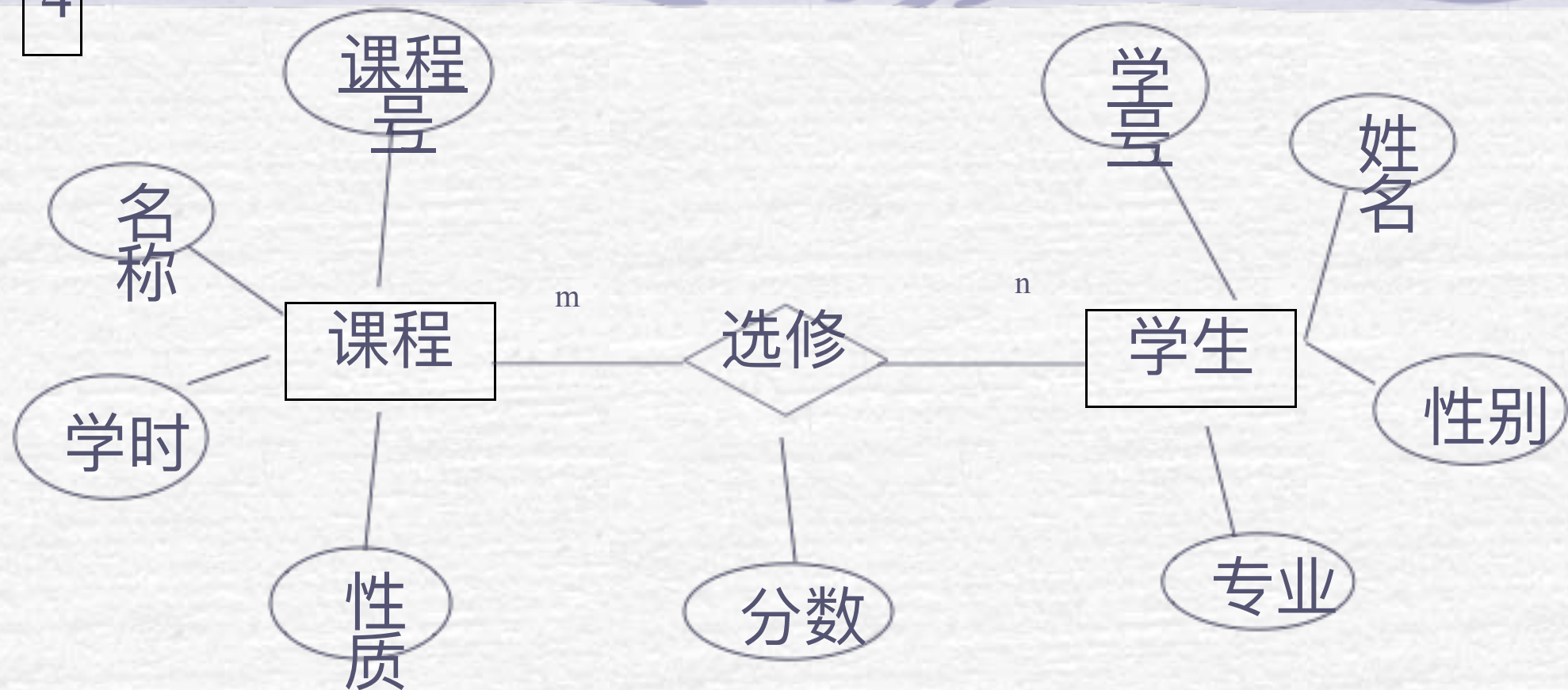


旅客 (身份证号, 姓名, 航班号, 座位号)  
 飞机票 (航班号, 座位号, 日期, 价格)

旅客 (身份证号, 姓名)  
 飞机票 (航班号, 座位号, 日期, 价格, 身份证号)



4



课程 (课程号, 课程名称, 学时, 性质)  
学生 (学号, 姓名, 性别, 专业)  
成绩 (学号, 课程号, 分数)

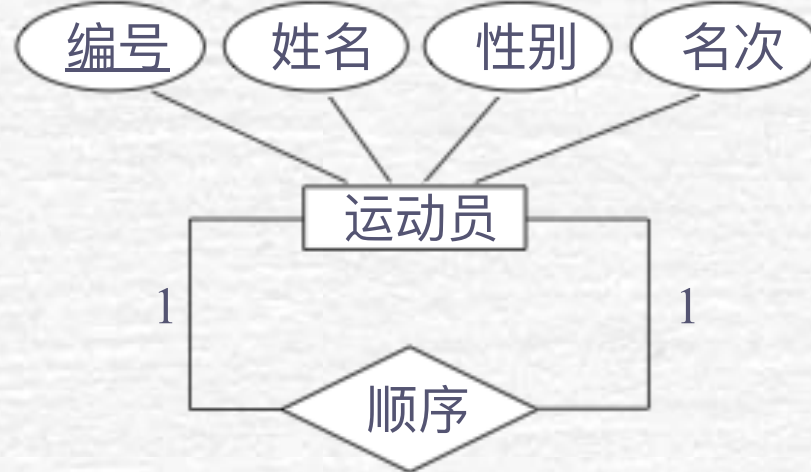


- 一元、二元、三元联系的转换规则：
- 二元联系的转换规则
- 一元联系的转换规则：同二元联系的转换规则。
- 三元联系的转换规则：

不管三元联系是哪一种类型（1:1:1, 1:1:n, 1:m:n 和 m:n:p），总是将三元联系也转换成关系模式，其属性为三端实体类型的键加上联系自己的属性，而联系关系模式的键为三端实体键的组合。

【例】一元联系的转换例子。

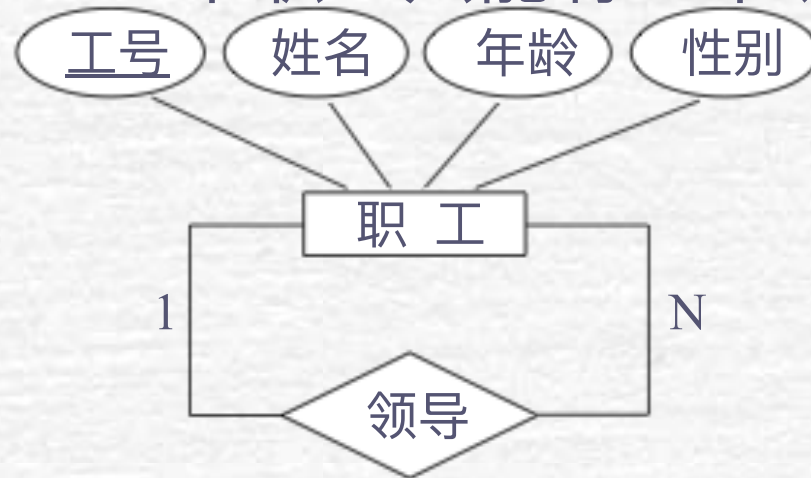
**1:1联系：**运动员根据得分来排定名次。在名次排列中，排在一个运动员前面的只有一个人。



运动员(编号, 姓名, 性别, 名次, 上一名次编号)  
或 运动员(编号, 姓名, 性别, 名次, 下一名次编号)

【例】一元联系的转换例子。

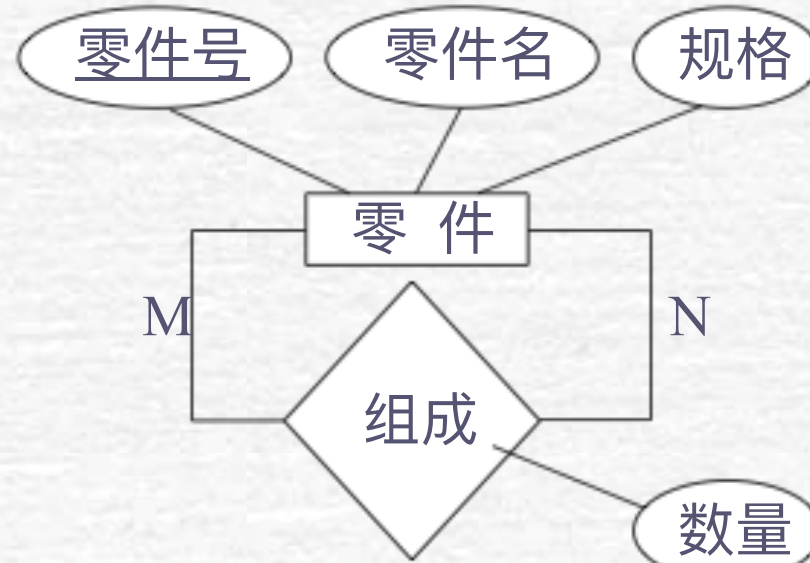
**1:N联系：**职工之间的上下级关系。一个职工可以领导多名职工；一个职工只能有一个领导。



职工(工号, 姓名, 年龄, 性别, 领导工号)

【例】一元联系的转换例子。

**M:N联系：**零件之间的组成关系。一个零件可以由多个零件组成；一个零件（作为子零件）也可以包括在多个零件中。



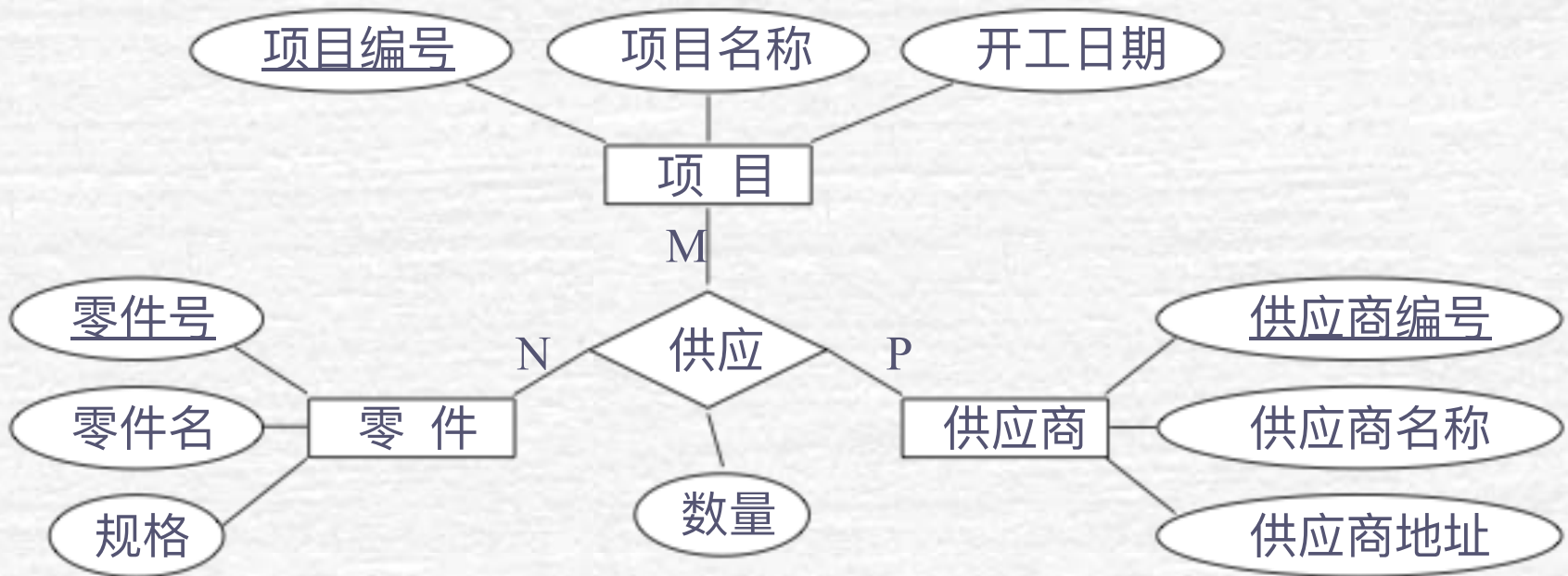
零件(零件号, 零件名, 规格)

组成(零件号, 子零件号, 数量)



## 【例】三元联系的转换例子。

项目—零件—供应商之间的M:N:P联系。

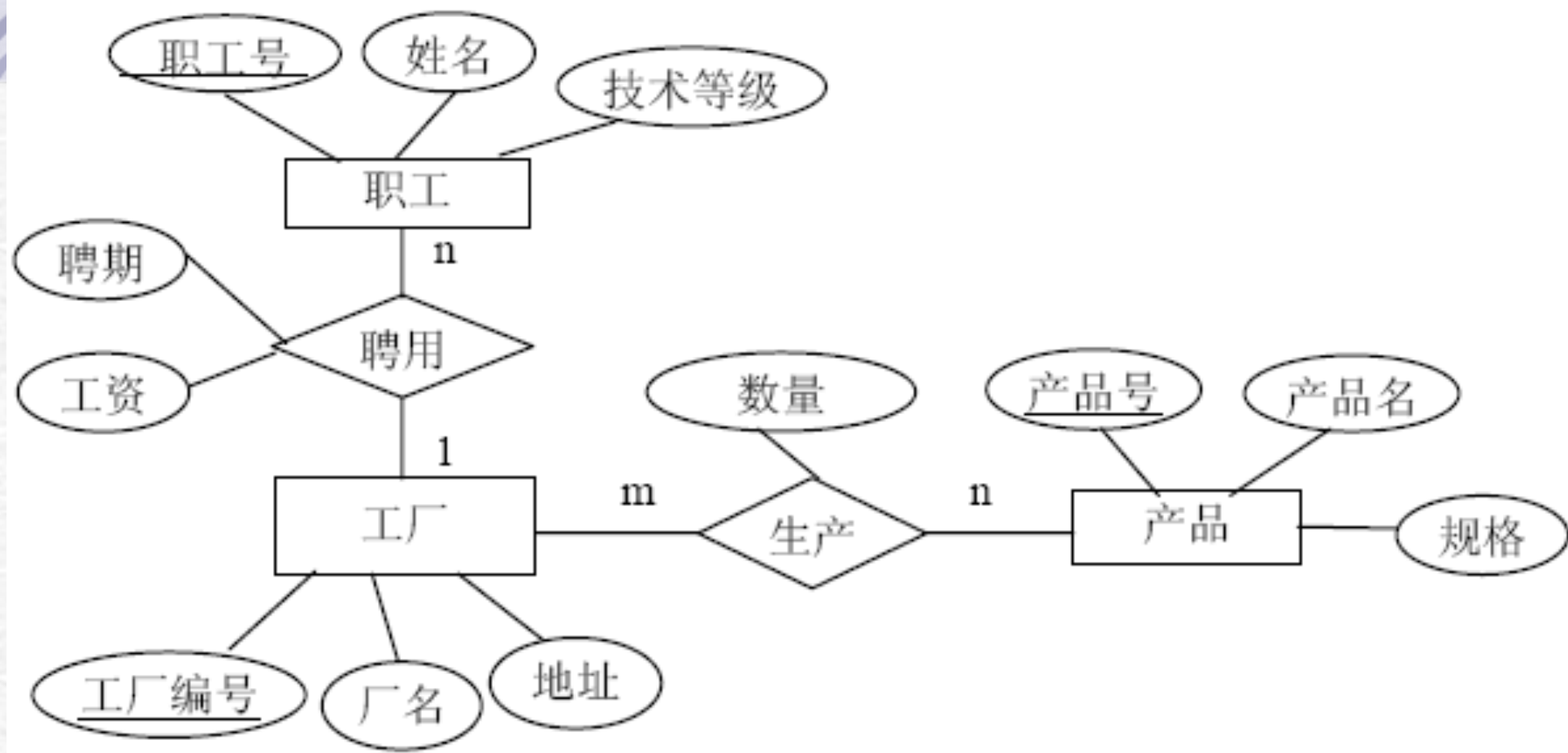


项目(项目编号, 项目名称, 开工日期)

零件(零件号, 零件名, 规格)

供应商(供应商编号, 供应商名称, 供应商地址)

供应(项目编号, 零件号, 供应商编号, 数量)



职工 (职工号, 姓名, 技术等级, 工厂编号, 聘期, 工资)

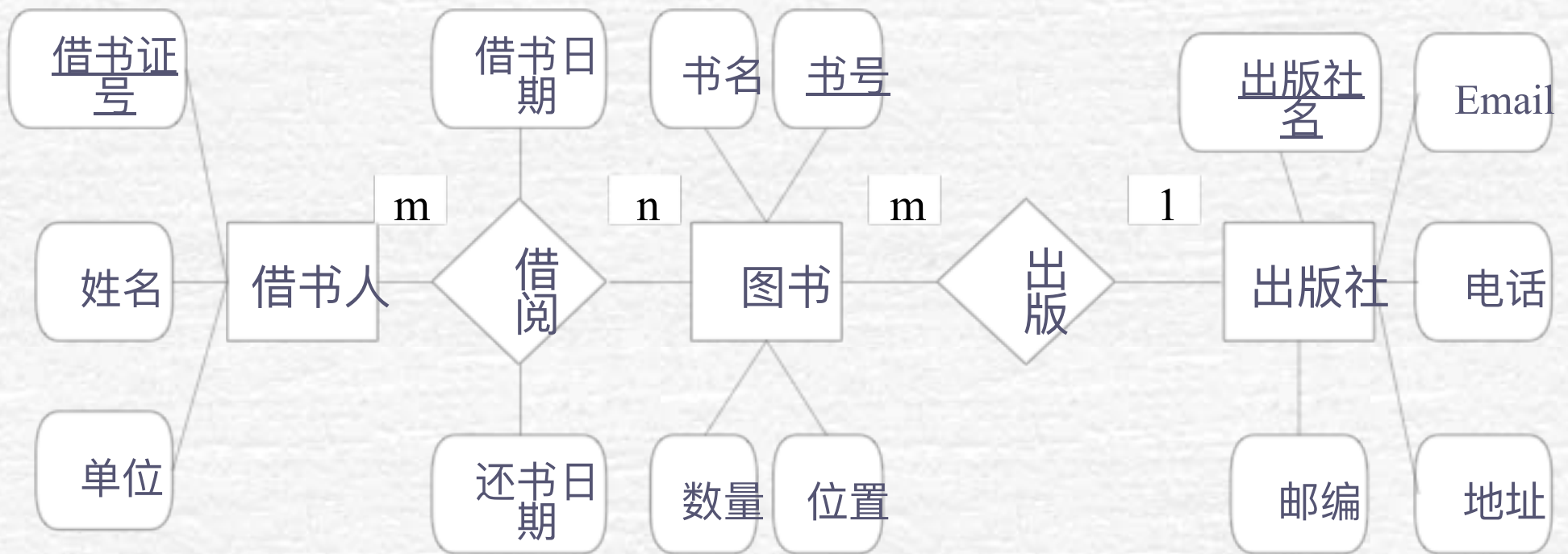
工厂 (工厂编号, 厂名, 地址)

产品 (产品号, 产品名, 规格)

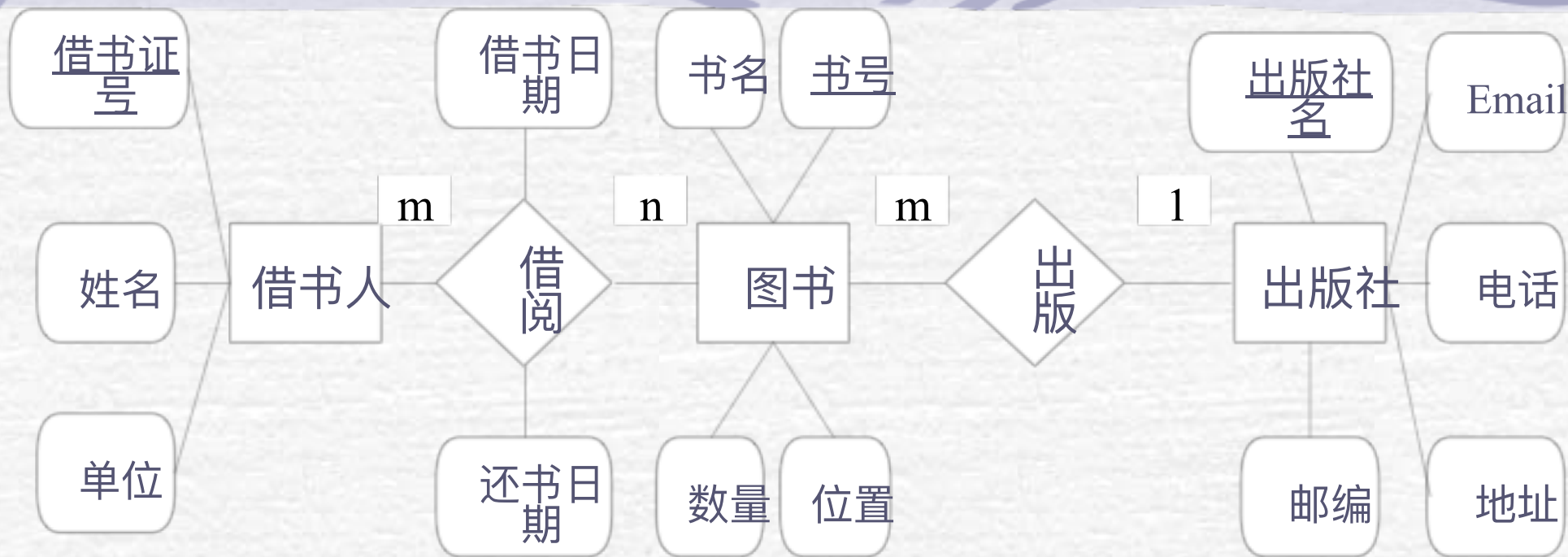
生产 (工厂编号, 产品号, 数量)

# 课堂小测试

## 转换为关系模型







借书人 (借书证号, 姓名, 单位)

图书 (书号, 书名, 数量, 位置, 出版社名)

出版社 (出版社名, Email, 电话, 邮编, 地址)

借阅 (借书证号, 书号, 借书日期, 还书日期)

# 课后作业安排

- **作业7**

将P159的7.15某大学的E-R图转换为关系模式

- **预习**

第8讲-SQL语言（课前预习要求）

