

将E-R模式转换为数据库表

- ❑ E-R图转换成表格式是从E-R图导出关系数据库设计的基础
- ❑ 符合E-R图的数据库可以表示成若干表的集合
- ❑ 对每一个实体集及联系集都有一个唯一的表，该表的名字就是对应实体集或联系集的名字

实体集表示为表

□ 强实体集转换到具有相同属性的表

<i>dept_name</i>	<i>building</i>	<i>budget</i>
Biology	Watson	90000
Comp. Sci.	Taylor	100000
Elec. Eng.	Taylor	85000
Finance	Painter	120000
History	Painter	50000
Music	Packard	80000
Physics	Watson	70000

department (*dept name*, *building*, *budget*)

还有: *classroom*(*building*, *room number*, *capacity*)

course(*course id*, *title*, *credits*)

instructor(*ID*, *name*, *salary*)

student(*ID*, *name*, *tot cred*)

有复合属性的实体集

- 复合属性可通过为每个组成属性创建一个单独属性
- 例，给定带有复合属性 *name* (组成属性为 *first-name* 和 *last-name*) 的实体集 *instructor*，对应的表具有两个属性

name.first-name 和 *name.last-name*

或是，*first-name* 和 *last-name*

有多值属性的实体集

□ 实体集E的多值属性 M 用一个单独的表EM表示

■ 例, *instructor*的多值属性`phone_number`表示为表
`instructor(ID, name, salary, phone_number)`

⇒ `instructor(ID, name, salary)`

`instructor_phone(ID, phone_number)`

□ 多值属性的每个值映射到表EM中的单独行

■ 主键为“22222”的*instructor*实体及其电话号码“555-1234”和“555-4321”映射到两行:

— `instructor_phone`联系中的(22222 , 555-1234)和(22222 , 555-4321)

弱实体集表示

- 弱实体集转换成的表还包含对应于其标识强实体集的主键的列

<u>course_id</u>	<u>sec_id</u>	<u>semester</u>	<u>year</u>	<u>building</u>	<u>room_number</u>	<u>time_slot_id</u>
BIO-101	1	Summer	2009	Painter	514	B
BIO-301	1	Summer	2010	Painter	514	A
CS-101	1	Fall	2009	Packard	101	H
CS-101	1	Spring	2010	Packard	101	F
CS-190	1	Spring	2009	Taylor	3128	E
CS-190	2	Spring	2009	Taylor	3128	A
CS-315	1	Spring	2010	Watson	120	D
CS-319	1	Spring	2010	Watson	100	B
CS-319	2	Spring	2010	Taylor	3128	C
CS-347	1	Fall	2009	Taylor	3128	A
EE-181	1	Spring	2009	Taylor	3128	C
FIN-201	1	Spring	2010	Packard	101	B
HIS-351	1	Spring	2010	Painter	514	C
MU-199	1	Spring	2010	Packard	101	D
PHY-101	1	Fall	2009	Watson	100	A

section(course_id, sec_id, semester, year, building,
room_number, time_slot_id)

联系集表示成表

- ❑ 联系集表示成的表具有对应于两个参加实体集的主键的列，以及对应于联系集自己的描述性属性的列
- ❑ 对于多对多的联系集
 - *takes*(*ID*, *course_id*, *sec_id*, *semester*, *year*, *grade*)
 - *teaches*(*ID*, *course_id*, *sec_id*, *semester*, *year*)
 - *prereq*(*course_id*, *prereq_id*)

联系集表示成表

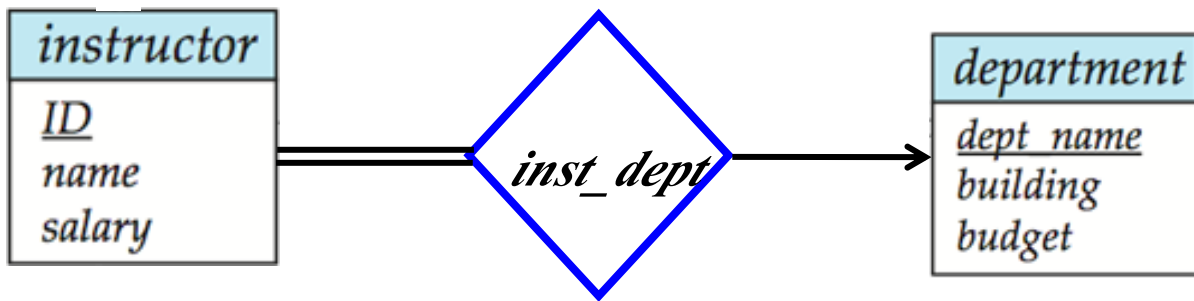
□ 对于多对一的联系集

- *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*)
- *stud_dept*(*ID*, *dept_name*)
- *course_dept*(*course_id*, *dept_name*)

□ 对于一对多的联系集（和多对一类似）

表的冗余

- 如果多对一和一对多联系集在“多”端是完全的，则可不必要为联系集创建表，而是在对应于“多”端的表中加入对应于“一”端表的主键的额外属性



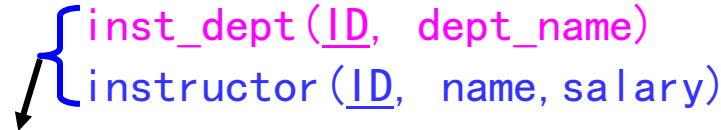
instructor(*ID*, *name*, *salary*); *department*(*dept_name*, *building*, *budget*)

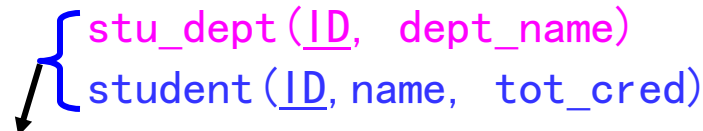
inst_dept(*ID*, *dept_name*)

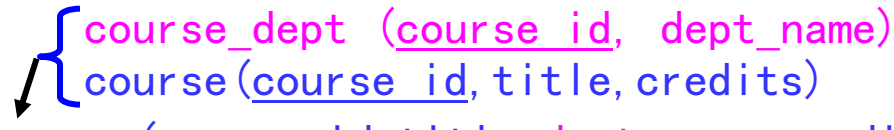
instructor(*ID*, *name*, *dept_name*, *salary*);

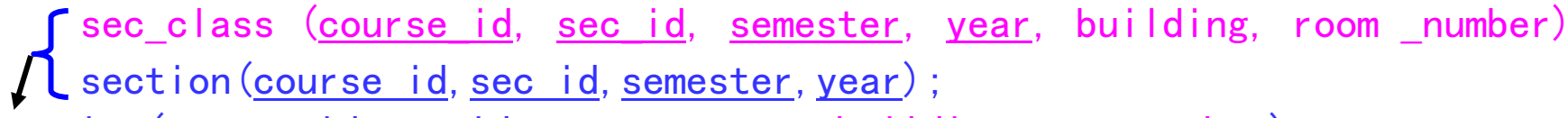
- 对于一对一联系集，任何一边都可选作为“多”端
 - 即，额外属性可加入到任何一个对应于两个实体集的表中
- 如果“多”端参加联系是部分的，上述方法可导致空值
 - 例，如果 *inst_dept* 是部分参与的，那么我们将为那些没有相关联的系的教师在属性 *dept_name* 中存放空值

表的冗余


instructor(ID, name, dept_name, salary)


student(ID, name, dept_name, tot_cred)

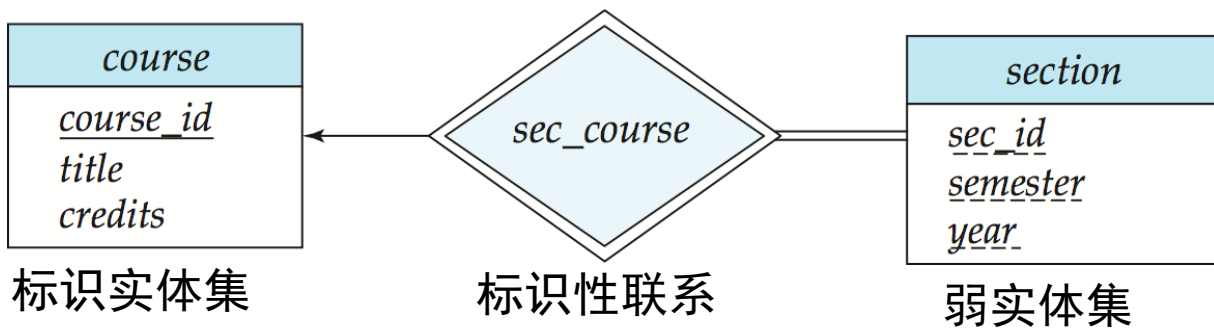

course(course_id, title, dept_name, credits)


section(course_id, sec_id, semester, year, building, room_number)



表的冗余

- ❑ 联系弱实体集及其标识性实体集的联系集对应的表是冗余的
- ❑ 例，*section*表已经包含了出现在*sec_course*表中的信息（如，*course_id*，*sec_id*, *semester*和*year*列）



表的冗余

`section` (course_id, sec_id, semester, year, building, room_number, time_slot_id)

<u>course_id</u>	<u>sec_id</u>	<u>semester</u>	<u>year</u>	building	room_number	time_slot_id
BIO-101	1	Summer	2009	Painter	514	B
BIO-301	1	Summer	2010	Painter	514	A
CS-101	1	Fall	2009	Packard	101	H
CS-101	1	Spring	2010	Packard	101	F
CS-190	1	Spring	2009	Taylor	3128	E
CS-190	2	Spring	2009	Taylor	3128	A
CS-315	1	Spring	2010	Watson	120	D
CS-319	1	Spring	2010	Watson	100	B
CS-319	2	Spring	2010	Taylor	3128	C
CS-347	1	Fall	2009	Taylor	3128	A
EE-181	1	Spring	2009	Taylor	3128	C
FIN-201	1	Spring	2010	Packard	101	B
HIS-351	1	Spring	2010	Painter	514	C
MU-199	1	Spring	2010	Packard	101	D
PHY-101	1	Fall	2009	Watson	100	A

删除: `sec_course` (course_id, sec_id, semester, year)



□ 方法1:

- 为高层实体集构造表
- 为每个低层实体集构造表，包括高层实体集的主键和局部属性

表	表属性
<i>person</i>	<i>ID, name, street, city</i>
<i>student</i>	<i>ID, tot_cred</i>
<i>employee</i>	<i>ID, salary</i>

- 缺点：获得 *employee*之类的实体的信息需要访问两个表

□ 方法2:

- 为每个实体集构造表，其属性包括所有局部属性和继承来的属性

表	表属性
<i>person</i>	<i>ID, name, street, city</i>
<i>student</i>	<i>ID, name, street, city, tot_cred</i>
<i>employee</i>	<i>ID, name, street, city, salary</i>

- 如果特化是全部特化，则没有必要为一般实体 *person* 创建表
 - 可以被定义为包含特化联系的“视图”
 - 由于外键约束的需要，可能仍然需要定义关系模式 *person*
- 缺点：对于既是学生又是雇员的人，其 *name*、*street* 和 *city* 被冗余存储

大学数据库模式

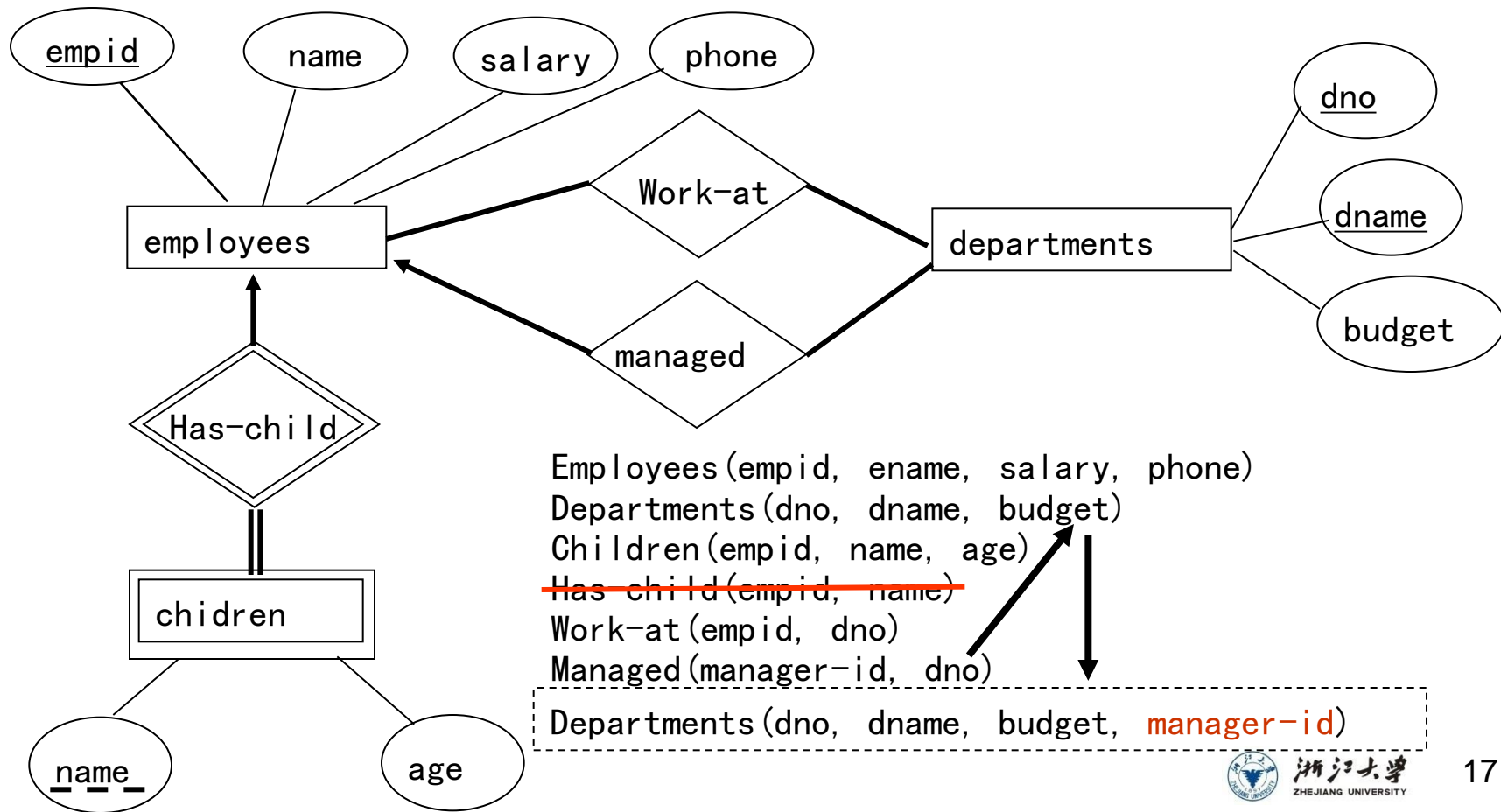
- ❑ *classroom*(building, room_number, capacity)
- ❑ *department*(dept_name, building, budget)
- ❑ *course*(course_id, title, dept_name, credits)
- ❑ *instructor*(ID, name, dept_name, salary)
- ❑ *section*(course_id, sec_id, semester, year, building, room_number, time_slot_id)
- ❑ *teaches*(ID, course_id,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



示例一

- ❑ 一个公司的数据库需要存储有关雇员（由empid唯一标识，有姓名、工资和手机号码作为属性），部门（由dno唯一标识，有部门名称与预算作为属性）和雇员的孩子信息（有姓名和年龄属性）。员工在部门工作，每个部门由一个管理员管理，孩子必须被唯一名称标识，其父母（假定只有父母中的一位在公司工作）是已知的。一旦父母离开公司，我们将不再关注孩子的信息。绘制ER图，然后将其转换为关系模式。

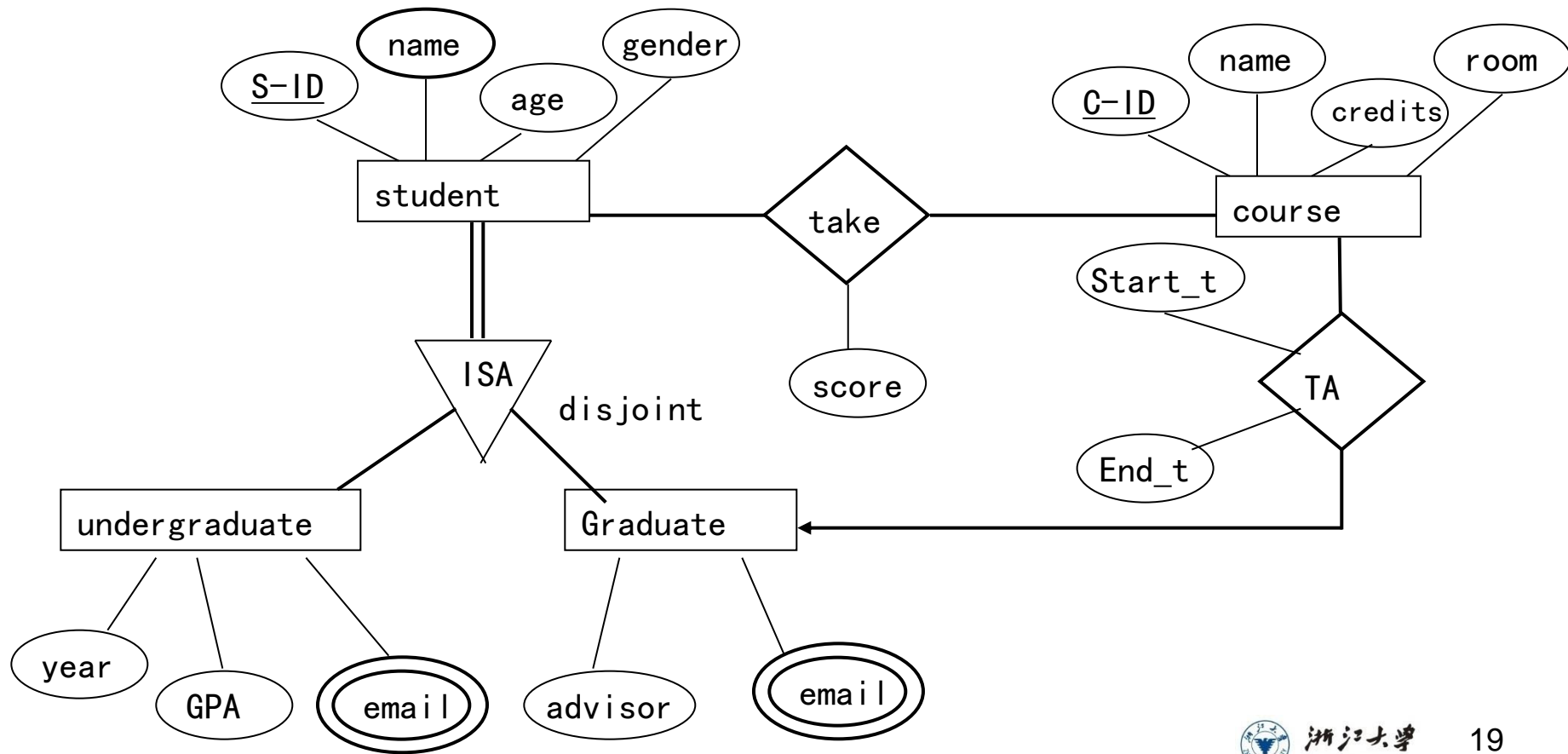
示例一



示例二

- ❑ 学生学习某门课程。每个学生都有学号，姓名，年龄和性别属性。每门课程都有课程编号，课程名，学分和上课教室编号属性。每个学生可以是本科生或研究生。对于每一个本科生，我们想记录他/她的学习年份，平均绩点，以及（可能是多个）电子邮箱。对于每一个研究生，我们要记录他/她的导师的名字，以及（可能是多个）电子邮箱。此外，每门课程有一名研究生作为课程助教，我们要记录助教的开始和结束时间（例如，开始于2015年3月10日，结束于2015年6月30日）。绘制E-R图。

示例二



- ❑ 数据库设计主要涉及数据库模式的设计
- ❑ E-R模型主要用于数据库设计过程、E-R图用于表示数据库模式的逻辑结构
- ❑ 实体、实体集
- ❑ 联系、联系集
- ❑ 超码、候选码以及主码同样适用于实体和联系集
- ❑ 映射基数
- ❑ 弱实体集、强实体集
- ❑ 将E-R图转换为关系模式

- ❑ 数据库设计主要涉及数据库模式的设计
- ❑ E-R模型主要用于数据库设计过程，E-R图用于表示数据库模式的逻辑结构，以及E-R图的其他表示方法
- ❑ 实体、实体集
- ❑ 联系、联系集
- ❑ 超码、候选码以及主码同样适用于实体和联系集
- ❑ 映射基数
- ❑ 弱实体集、强实体集
- ❑ 特化和概化定义了一个高层实体集和一个或多个低层实体集之间的包含关系

- ❑ 聚集是一种抽象，其中联系集（和跟它们相关的实体集一起）被看作高层实体集，并且可以参与联系
- ❑ E-R设计问题
- ❑ 将E-R图转换为关系模式

谢谢！