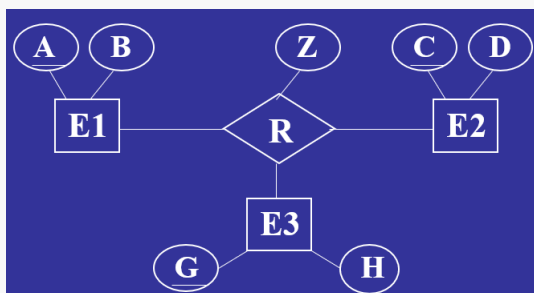


数据库系统 期末速通教程

6. ER 图

6.1 多元关系

[Ternary Relationship]



$E1(\underline{A}, B),$
 $E2(\underline{C}, D),$
 $E3(\underline{G}, H),$
 $R(\underline{A}, \underline{C}, \underline{G}, Z),$
 其中 $R.A$ 、 $R.C$ 、 $R.G$ 是外键

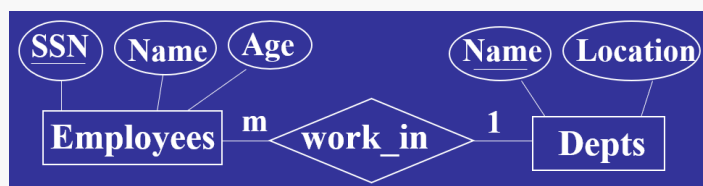
[Binary Relationship]



(1) 一对多: $x = 1, y = m$ 或 $x = (?, m), y = (?, 1)$.

关系表: $E(\underline{A}, B), F(\underline{C}, D, A)$, 其中 $F.A$ 是外键.

[记] 1 的一端作外键.



$Depts(\underline{Name}, Location),$
 $Employees(\underline{SSN}, Name, Age, Dept_name)$
 其中 $Employees.Dept_name$ 是外键

(2) 一对一:

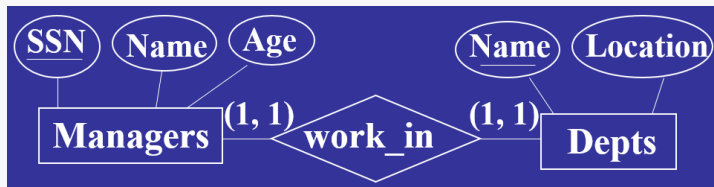
[记] 少的一方作外键.

① $x = (1, 1), y = (1, 1)$ 或 $x = (0, 1), y = (0, 1)$.

关系表: $E(\underline{A}, B), F(\underline{C}, D, A)$, 其中 $F.A$ 是外键.

或: $E(\underline{A}, B, C), F(\underline{C}, D)$, 其中 $E.C$ 是外键.

[记] 双方都强制参与或都部分参与, 则任一方作外键.



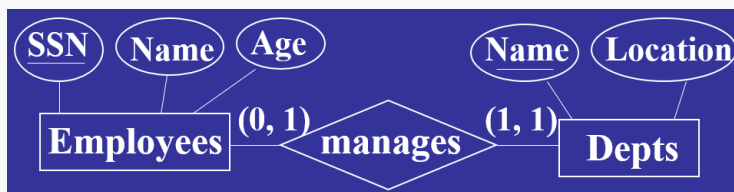
Depts(Name, Location),
Managers(SSN, Name, Age, Dept_name)

其中 Managers.Dept_name 是外键
或

Depts(Name, Location, Manager_SSN),
Managers(SSN, Name, Age),
其中 Depts.Name 是外键

② $x = (0, 1), y = (1, 1)$.

关系表: $E(\underline{A}, B), F(\underline{C}, D, A)$, 其中 $F.A$ 是外键.



Depts(Name, Location, Manager_SSN)

Employees(SSN, Name, Age),
其中 Depts.Manager_SSN 是外键

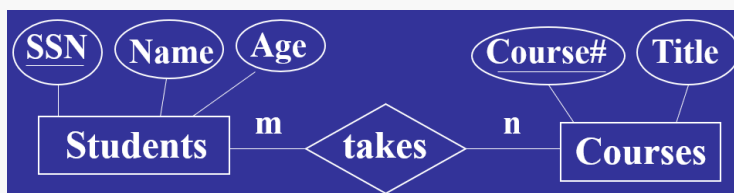
③ $x = (1, 1), y = (0, 1)$.

关系表: $E(\underline{A}, B, C), F(\underline{C}, D)$, 其中 $E.C$ 是外键.

(3) 多对多: $x = m, y = n$.

① 若关系 R 无属性:

关系表: $E(\underline{A}, B), F(\underline{C}, D), R(\underline{A}, \underline{C})$, 其中 $R.A$ 和 $R.C$ 是外键.

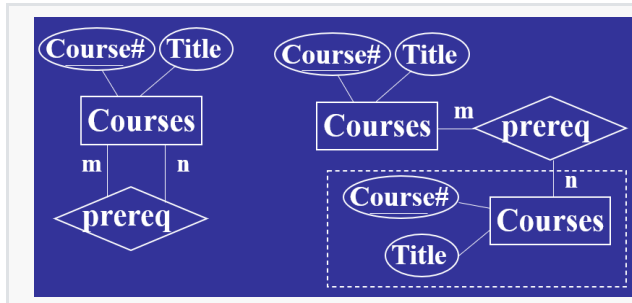


Students(SSN, Name, Age),
Courses(Course#, Title),
Takes(SSN, Course#),
其中 Takes.SSN 和 Takes.Course# 是外键

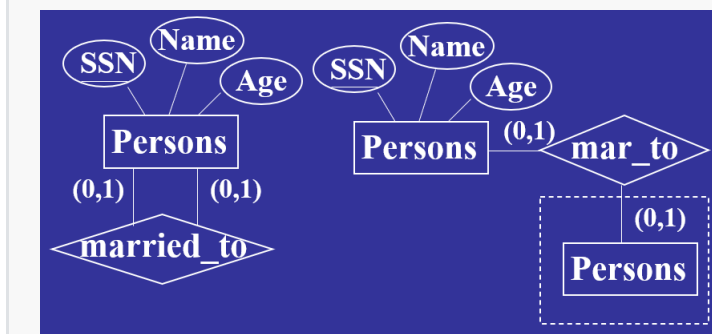
② 若关系 R 有属性 Z :

关系表: $E(\underline{A}, B), F(\underline{C}, D), R(\underline{A}, \underline{C}, Z)$, 其中 $R.A$ 和 $R.C$ 是外键.

[Unary Relationship] 将关系表复制一份, 将 Unary Relationship 转化为 Binary Relationship, 应用 Binary Relationship 的转换方法转化为关系表后, 删去冗余表. 若无冗余表, 删除属性较少的表.



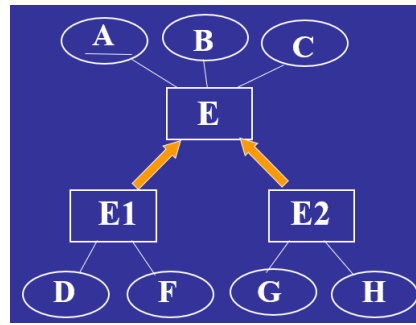
$Courses(\underline{Course\#}, Title),$
 $Prereq(\underline{Course\#}, \underline{Prereq_Course\#}),$
 其中 $Prereq.Course\#$ 和 $Prereq.Prereq_Course\#$
 是外键



$Persons(\underline{SSN}, Name, Age, Spouse_SSN),$
 其中 $Spouse_SSN$ 是外键

6.2 继承

[继承]

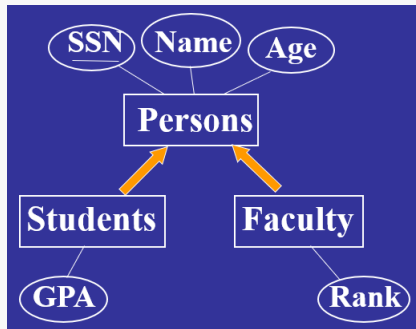


(1) 方法 1 : 子实体只保留父实体的主键作外键.

关系表: $E(\underline{A}, B, C)$, $E1(\underline{A}, D, F)$, $E2(\underline{A}, G, H)$, 其中 $E1.A$ 和 $E2.A$ 是外键.

(2) 方法 2 : 子实体继承父实体的所有属性.

关系表: $E(\underline{A}, B, C)$, $E1(\underline{A}, B, C, D, F)$, $E2(\underline{A}, B, C, G, H)$, 其中 $E1.A$ 和 $E2.A$ 是外键.



方法 1:

Persons(SSN, Name, Age),

Students(SSN, GPA),

Faculty(SSN, Rank),

其中 Students.SSN 和 Faculty.SSN 是外键

方法 2:

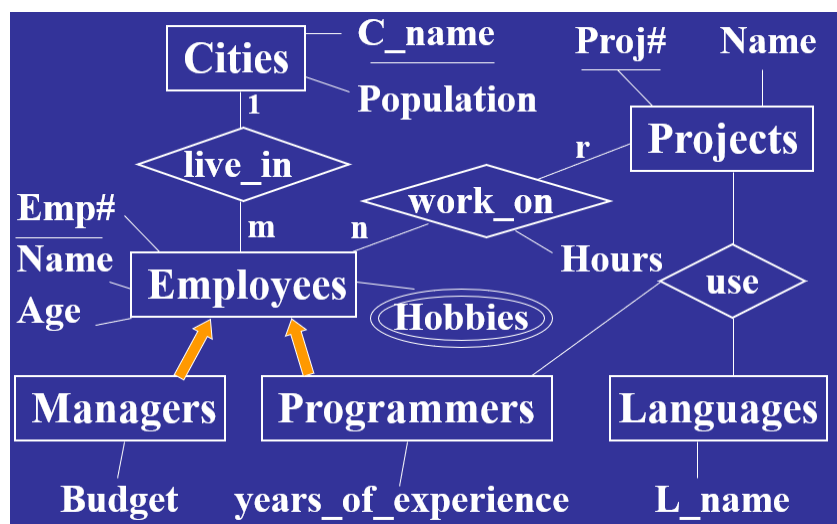
Persons(SSN, Name, Age),

Students(SSN, Name, Age, GPA),

Faculty(SSN, Name, Age, Rank),

其中 Students.SSN 和 Faculty.SSN 是外键

6.3 综合



方法 1:

```

Employees(Emp#, Name, Age, C_name),
Employee-Hobby(Emp#, Hobby),
Managers(Emp#, Budget),
Programmers(Emp#, Years_of_experience),
Cities(C_name, Population),
Projects(Proj#, Name),
Languages(L_name),
Work_on(Emp#, Proj#, Hours),
Use(Emp#, Proj#, L_name)
  
```

方法 2:

```

Employees(Emp#, Name, Age, C_name),
Employee-Hobby(Emp#, Hobby),
Managers(Managers-Emp#, Name, Age, C_name, Budget),
Manager-Hobby(Manager-Emp#, Hobby),
Programmers(Programmer-Emp#, Name, Age,
C_name, Years_of_experience),
Programmer-Hobby(Programmer-Emp#, Hobby),
Cities(C_name, Population),
Projects(Proj#, Name),
Languages(L_name),
Work_on(Emp#, Proj#, Hours),
Manager-Work_on(Manager-Emp#, Proj#, Hours),
Programmer-Work_on(Programmer-Emp#, Proj#, Hours),
Use(Programmer-Emp#, Proj#, L_name)
  
```