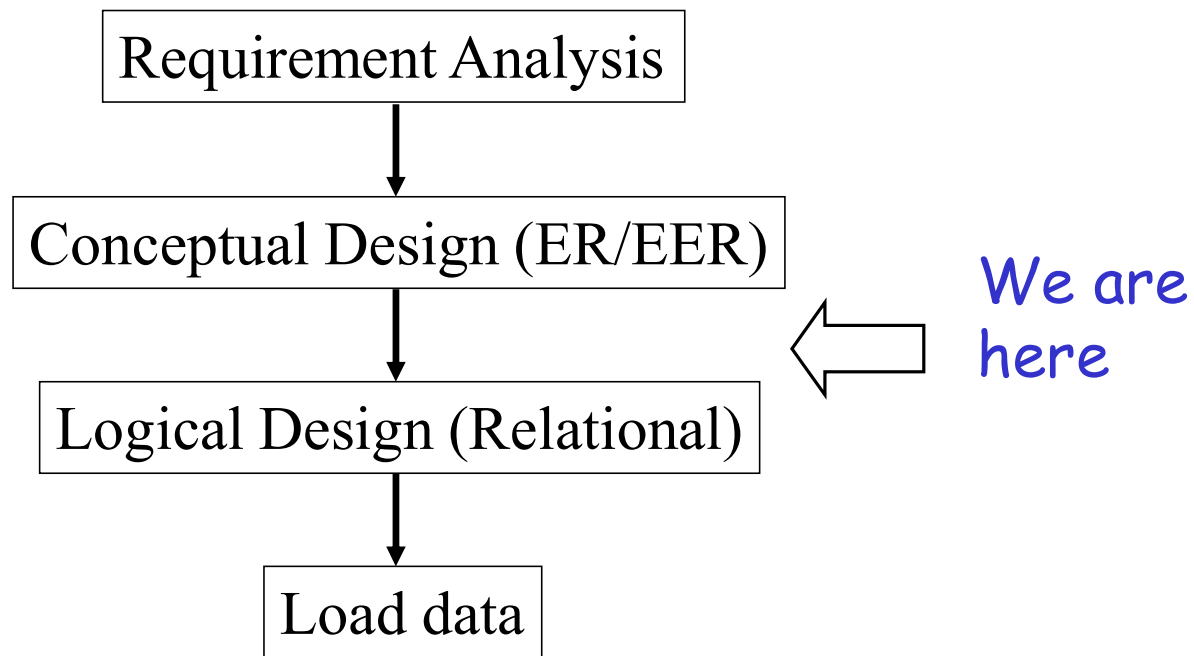


ER to Relational Translation

Recall Database Design

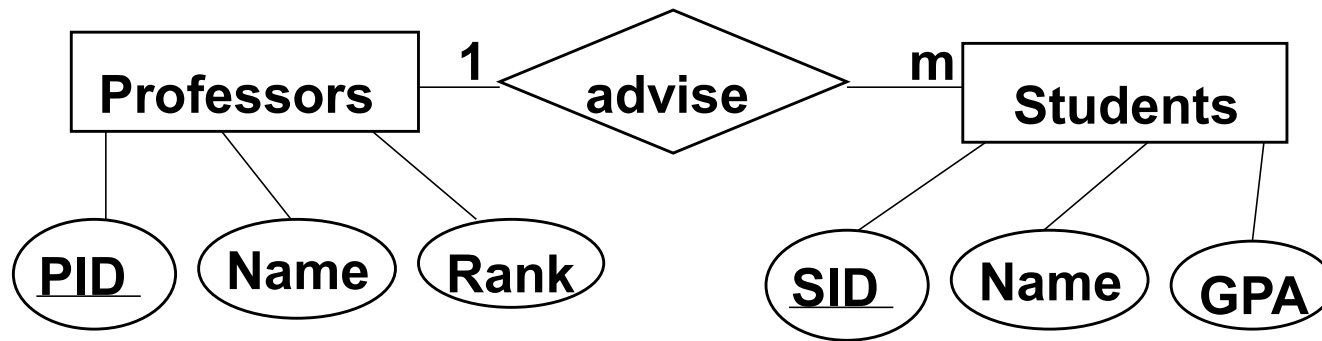
- ◆ Steps in database design



General Rules of Translation

- ◆ Translate each strong entity type to a relation.
 - ◆ Translate each relationship type into a relation or into a foreign key.
 - ◆ Translate attributes in ER to attributes in relations
 - ◆ Translate keys in ER to keys in relations
- ☛ Need rules for specific situations

Simple Translation: An Example



| Professors | Advise | Students |
|----------------------|---------------|--------------------|
| p1: 123, Jack, Prof. | p1 advises s1 | s1: 456, John, 3.4 |
| p2: 234, Ann, Prof. | p1 advises s2 | s2: 567, Carl, 3.2 |
| p3: 345, Bob, Prof. | p3 advises s3 | s3: 678, Ken, 3.5 |

Simple Translation: An Example (cont.)

Translate into three relations:

Professors

| <u>PID</u> | Name | Rank |
|------------|------|-------|
| 123 | Jack | Prof. |
| 234 | Ann | Prof. |
| 345 | Bob | Prof. |

Advises

| <u>PID</u> | <u>SID</u> |
|------------|------------|
| 123 | 456 |
| 123 | 567 |
| 345 | 678 |

Students

| <u>SID</u> | Name | GPA |
|------------|------|-----|
| 456 | John | 3.4 |
| 457 | Carl | 3.2 |
| 678 | Ken | 3.5 |

Actually, two relations are sufficient:

Professors

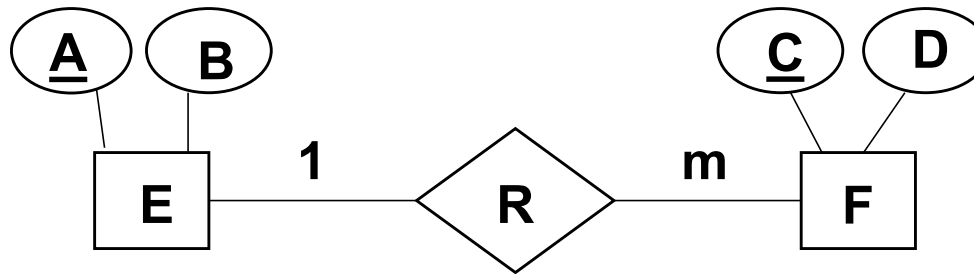
| <u>PID</u> | Name | Rank |
|------------|------|-------|
| 123 | Jack | Prof. |
| 234 | Ann | Prof. |
| 345 | Bob | Prof. |

Students

| <u>SID</u> | Name | GPA | Advisor |
|------------|------|-----|---------|
| 456 | John | 3.4 | 123 |
| 457 | Carl | 3.2 | 123 |
| 678 | Ken | 3.5 | 345 |

Translate 1-to-m Relationships

Translate one-to-many relationship

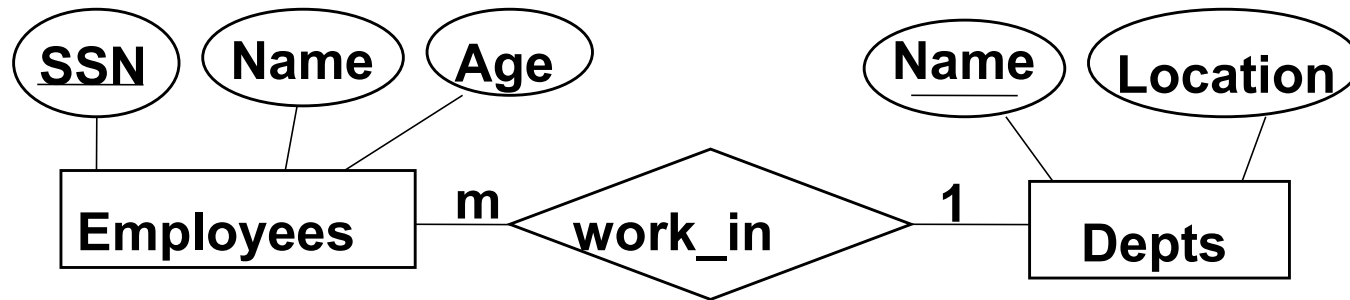


into relations: E(A, B), F(C, D, A)

- ◆ Relationship R is translated into a foreign key.

Translate 1-to-m Relationship

Example: Translating work_in relationship.

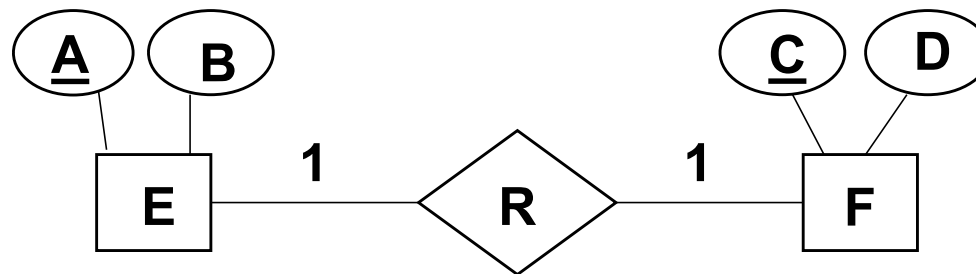


Employees(SSN, Name, Age, Dept_name)
Depts(Name, Location)

- Give meaningful attribute names to indicate relationships.

Translate 1-to-1 Relationships

Translate one-to-one relationship

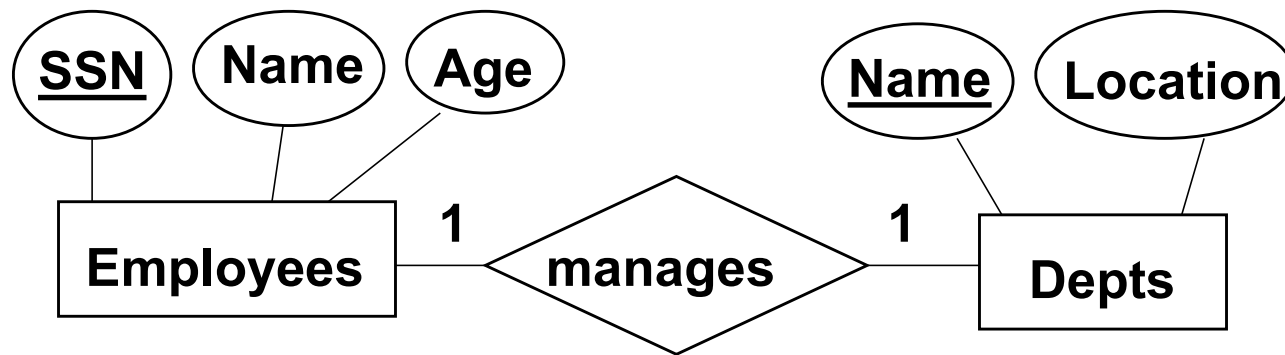


into either
or

$E(\underline{A}, B), F(\underline{C}, D, A)$
 $E(\underline{A}, B, C), F(\underline{C}, D)$

Translate 1-to-1 Relationship

Example: Translating "manages" relationship.



Either : Employees(SSN, Name, Age, ManagerOf)

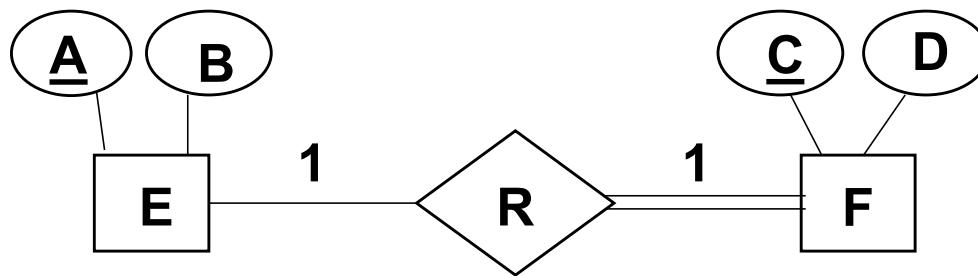
Depts(Name, Location)

Or: Employees(SSN, Name, Age)

Depts(Name, Location, ManagerID)

Translate 1-to-1 Relationship (cont.)

Translate one-to-one relationship

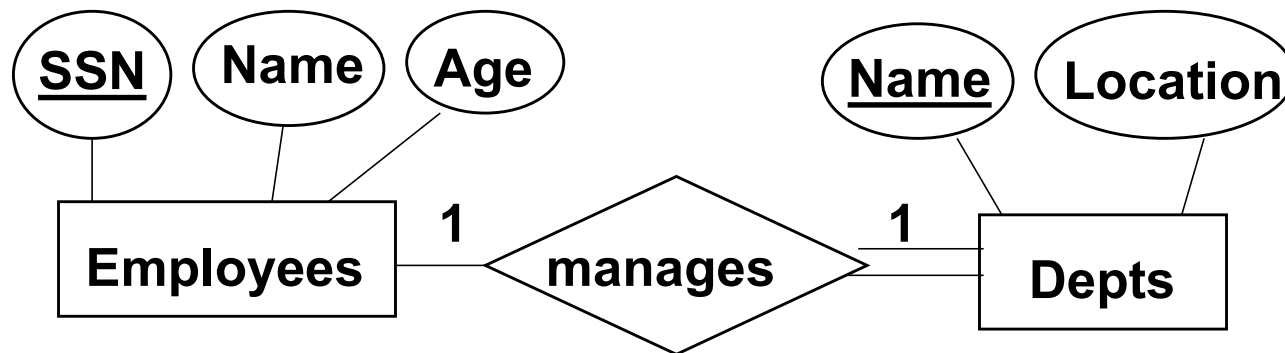


into $E(\underline{A}, B), F(\underline{C}, D, A)$

- The entity set with the total participation is transformed to a relation with a foreign key.

Translate 1-to-1 Relationship (cont.)

Example: Translating "manages" relationship.



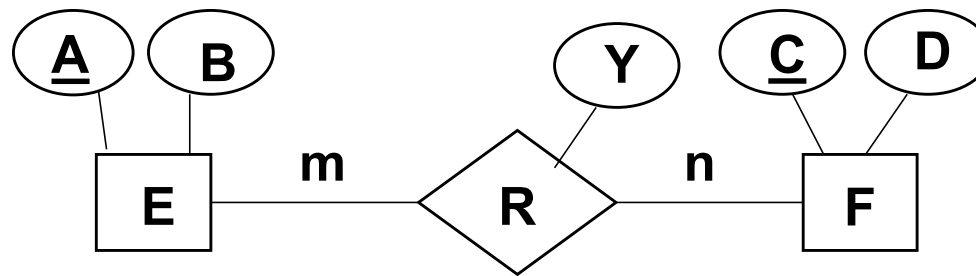
Employees(SSN, Name, Age)

Depts(Name, Location, ManagerID)

☛ Why not let Employees have the foreign key?

Translate m-to-n Relationships

Translate many-to-many relationship

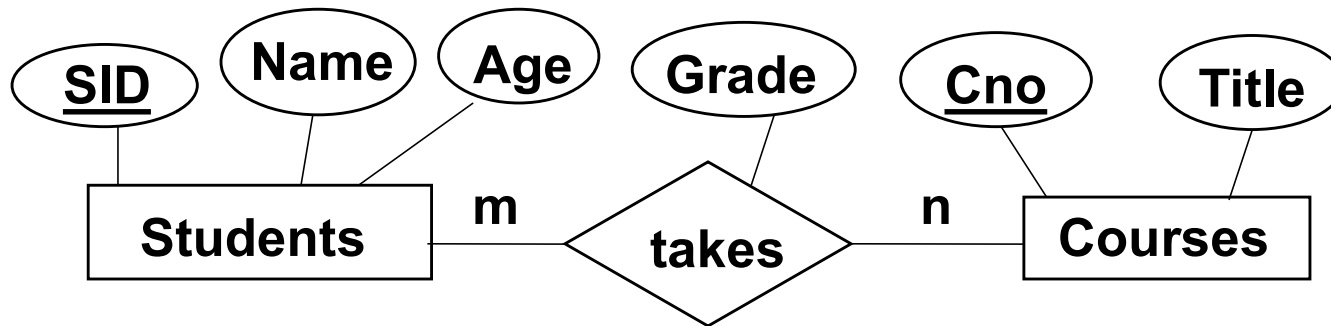


into $E(\underline{A}, B), F(\underline{C}, D), R(\underline{A}, \underline{C}, Y)$

- ☛ Transform the m-to-n relationship to a separate relation.
- ☛ R has two foreign keys & they form the primary key of R.

Translate m-to-n Relationship

Example: Translating "takes" relationship.



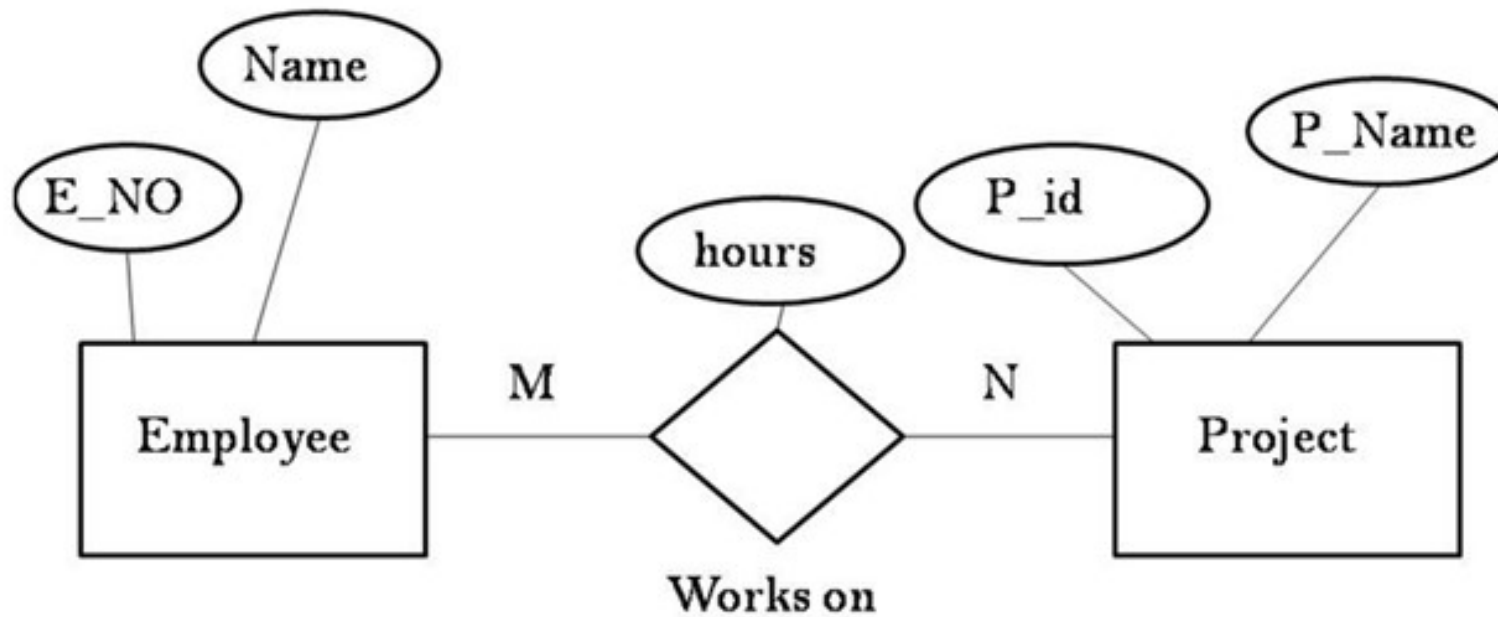
Students(SID, Name, Age)

Courses(Cno, Title)

Takes(SID, Cno, Grade)

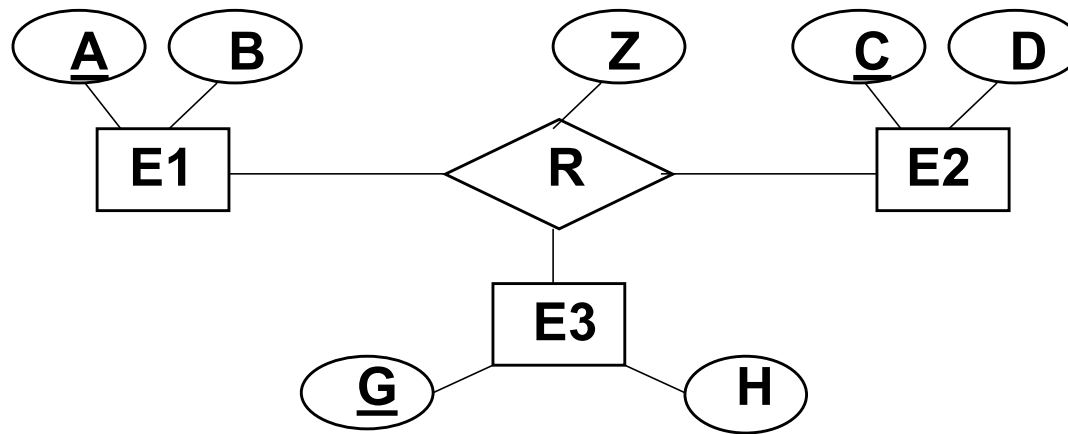
Exercise

Convert to a Relational Table



Translate Ternary Relationship

Translate ternary relationship

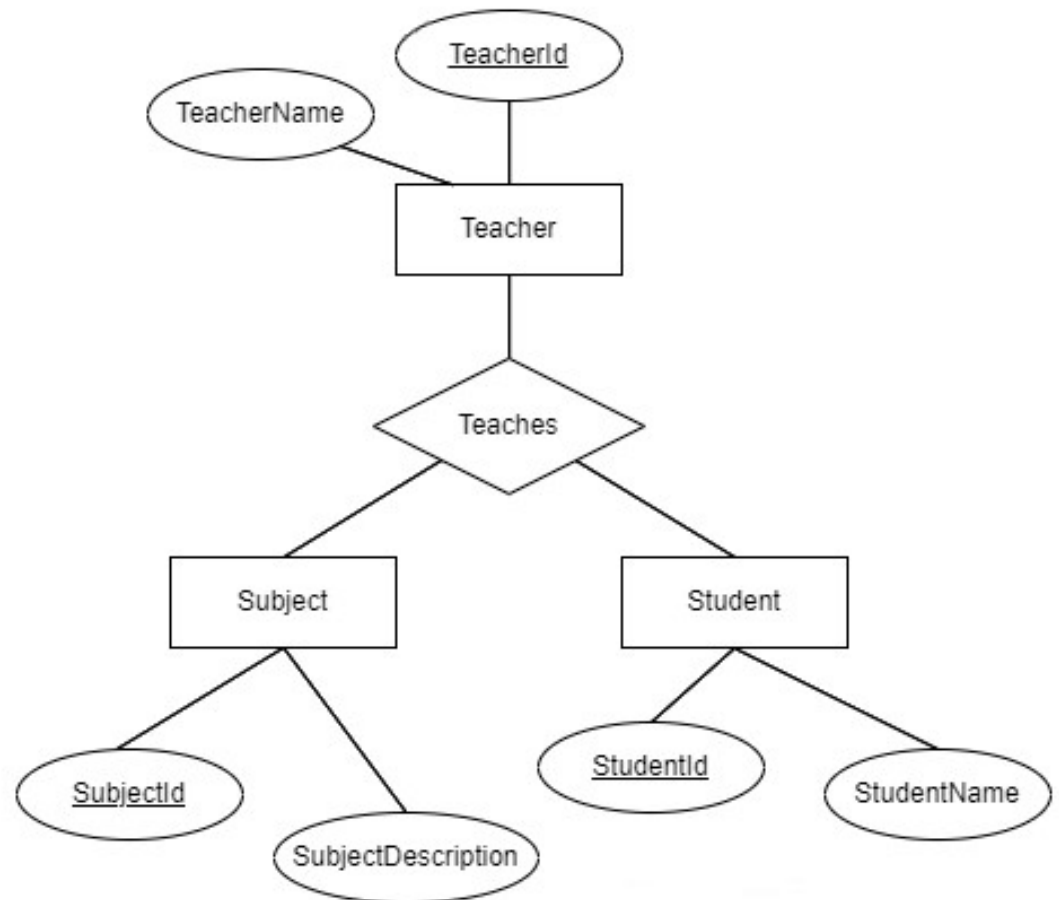


into $E1(\underline{A}, B), E2(\underline{C}, D), E3(\underline{G}, H), R(\underline{A}, \underline{C}, \underline{G}, Z)$

- Translate n-ary relationship to a relation with n foreign keys.

Exercise

Convert to a Relational Table



Teacher(TeacherId, TeacherName)

Subject(SubjectId, SubjectDescription)

Student(StudentId, StudentName)

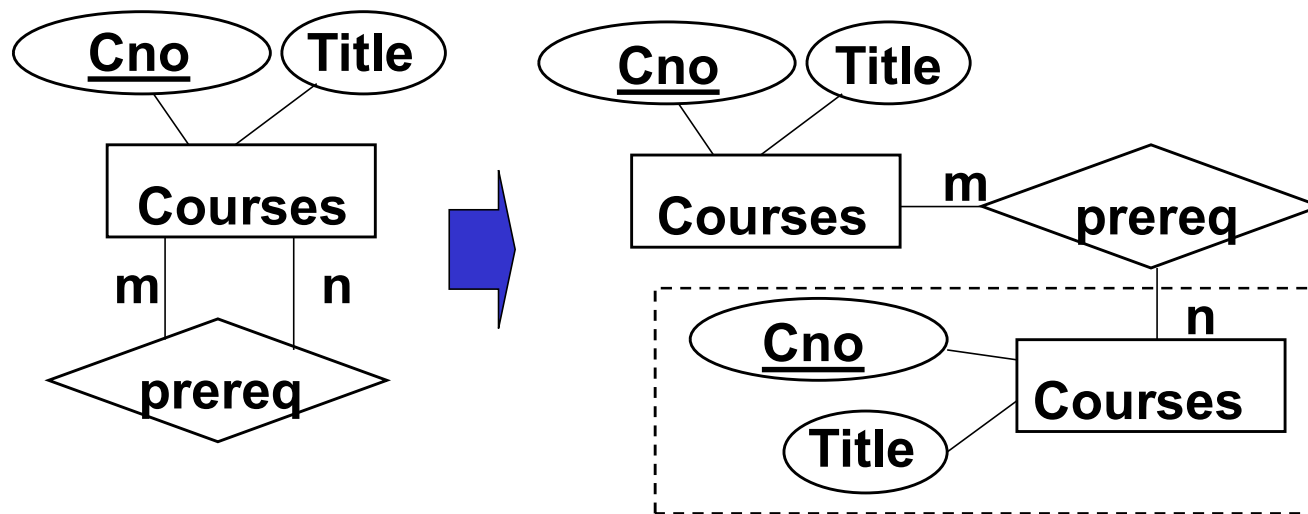
TeachSchedule(TeacherId,SubjectId,StudentId)

Translate Unary Relationship

- ◆ Create a *shadow* entity type and translate the unary relationship into a binary relationship.
- ◆ Apply the rules for translating binary relationships.
- ◆ After the translation, remove one redundant relation, or if there is no redundant relation, remove the relation with fewer attributes.

Translate Unary Relationship (cont.)

- ◆ Translate many-to-many unary relationship

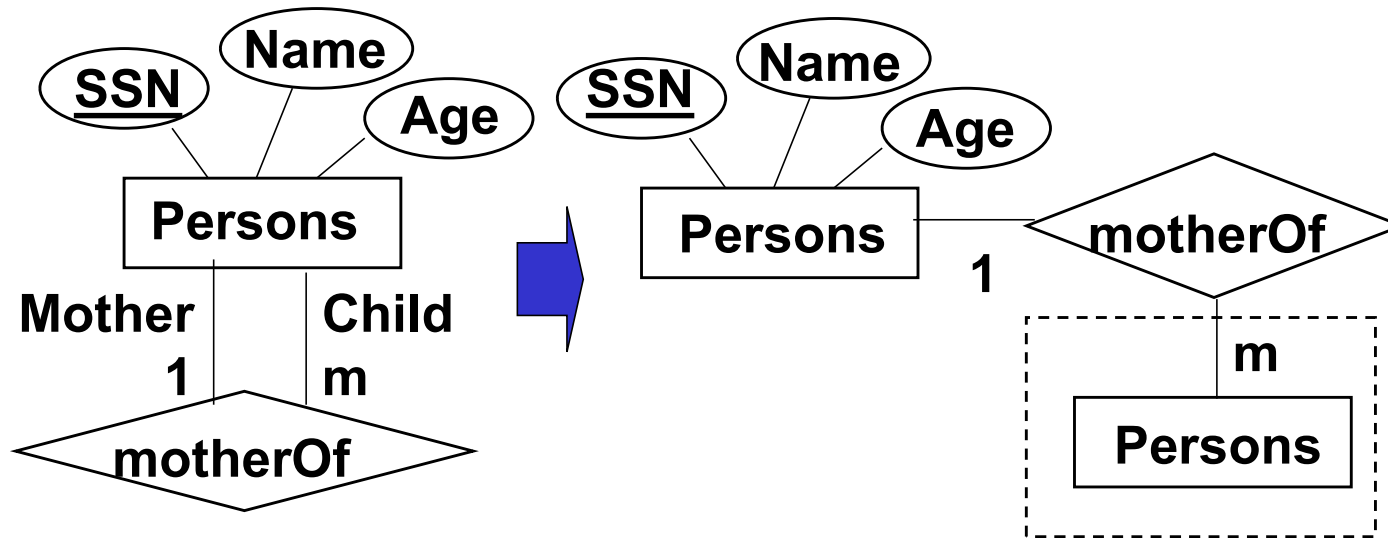


into `Courses(Cno, Title)`, `Prereq(Cno, Prereq_Cno)`

- ☛ Again, use a separate relation.

Translate Unary Relationship (cont.)

- ◆ Translate one-to-many unary relationship

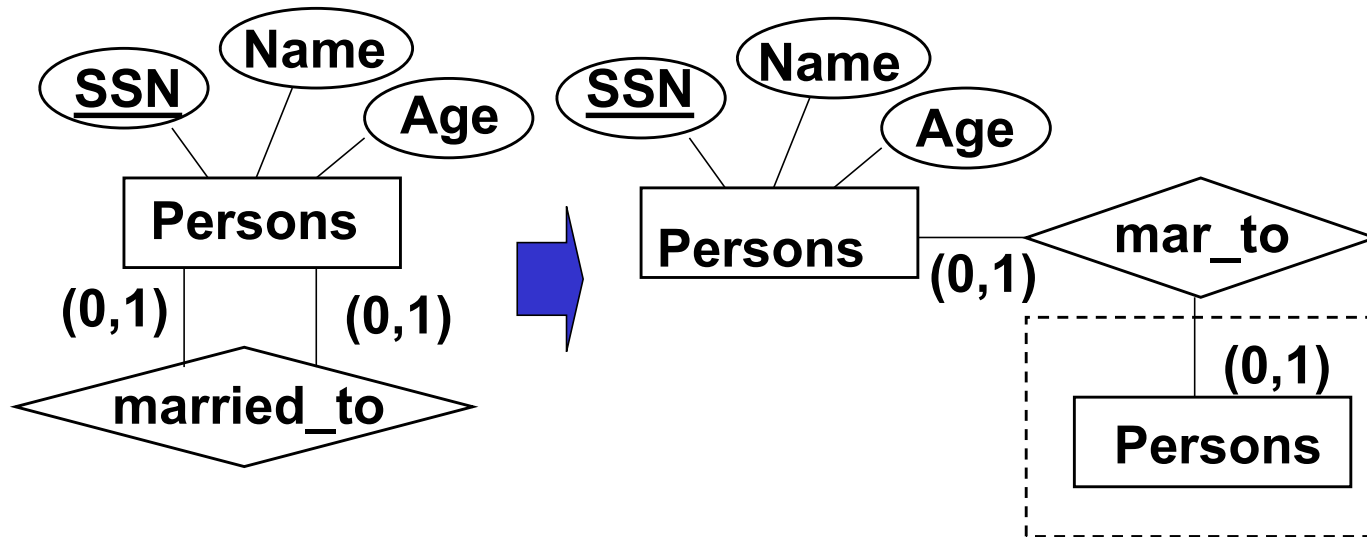


into `Persons(SSN, Name, Age, MotherSSN)`

☛ Again, use a foreign key.

Translate Unary Relationship (cont.)

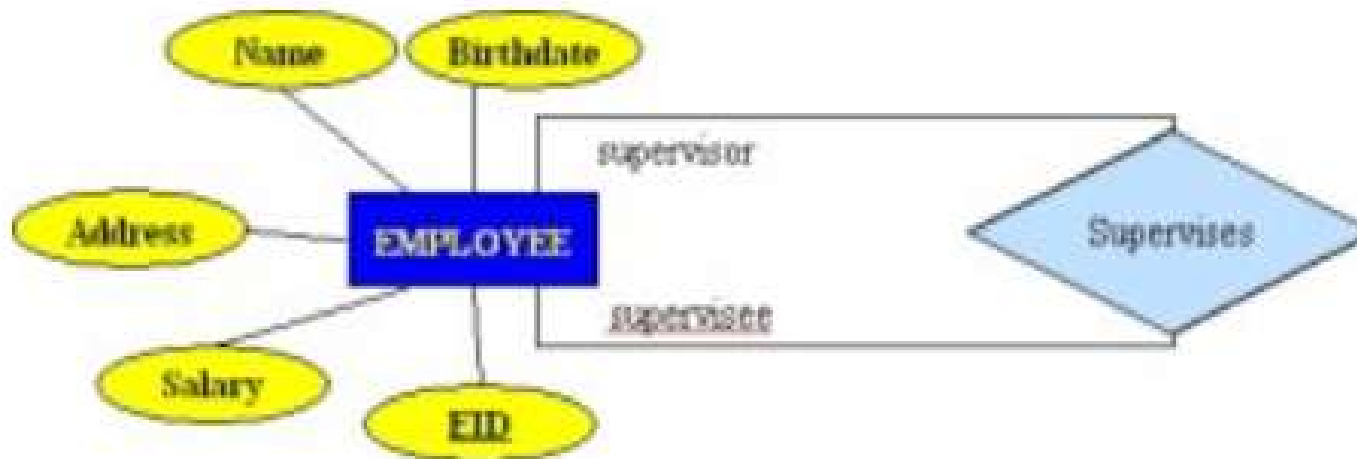
- ◆ Translate one-to-one unary relationship (special case of 1-to-m)



into **Persons**(SSN, Name, Age, SpouseSSN)

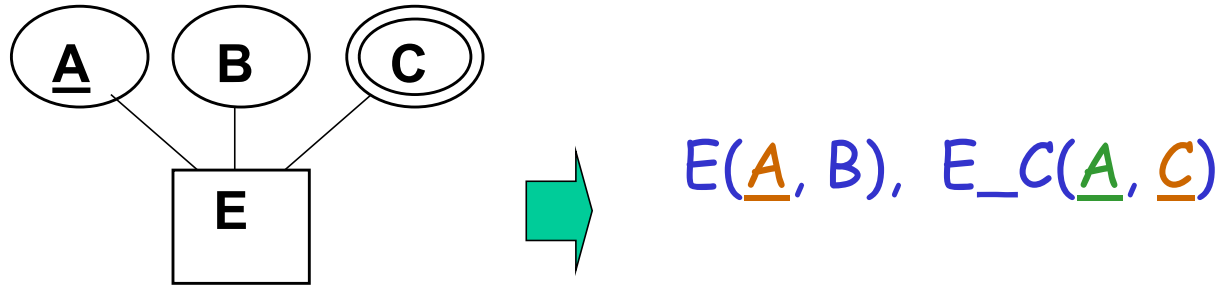
Exercise

Convert to a Relational Table



Translate Multi-valued Attribute

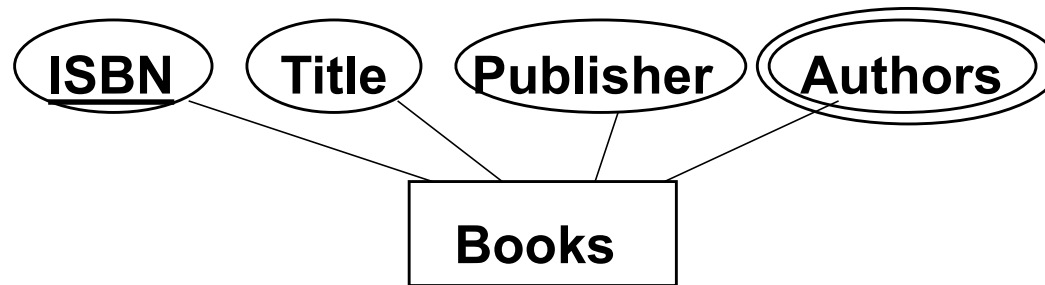
- ◆ Create a separate relation.



- $E_C.A$ is a foreign key referencing $E.A$.
- $E_C.C$ must be a part of primary key.

Translate Multi-valued Attribute

- ◆ Translate attribute authors to a new relation



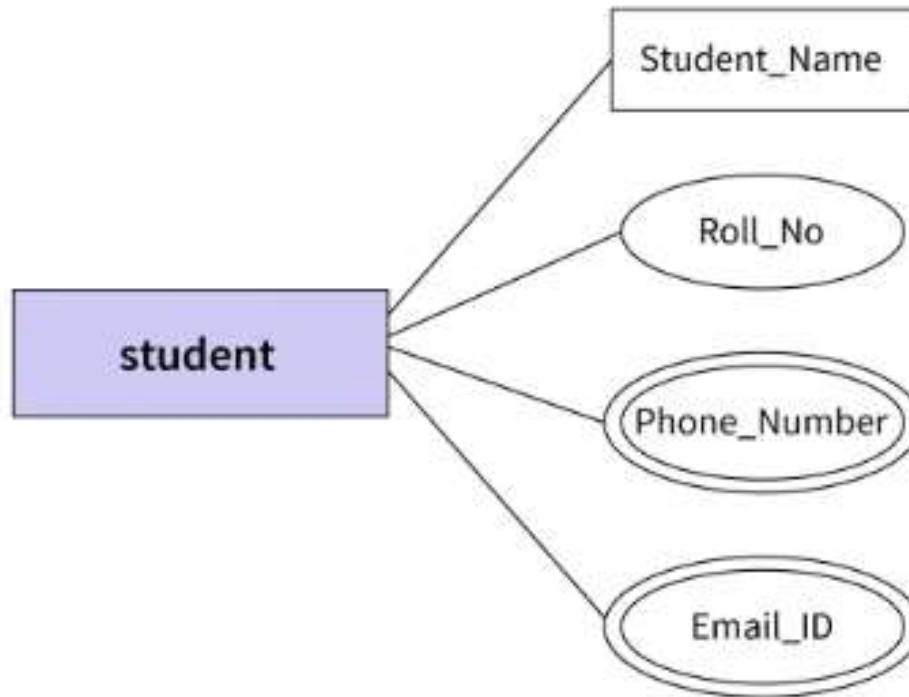
Books (ISBN, Title, Publisher)

Book_Authors (ISBN, Author)

- ◆ Define Book_Authors.ISBN as a foreign key referencing Books.ISBN

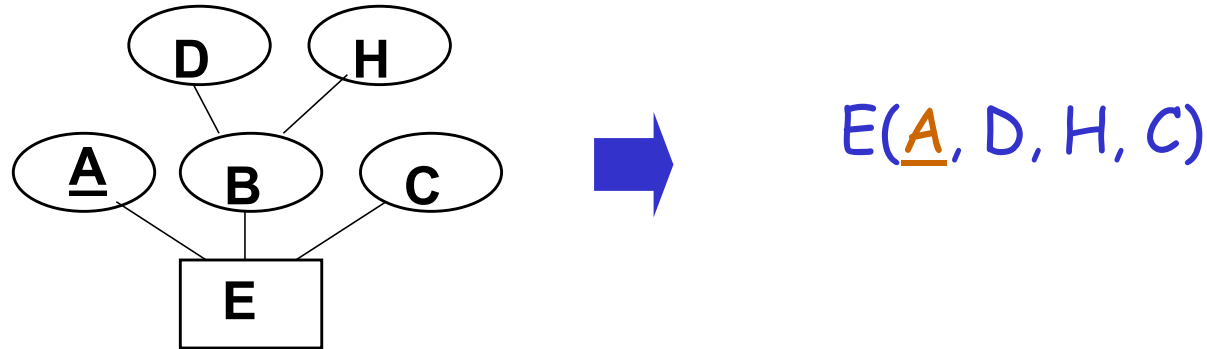
Exercise

Convert to a Relational Table



Translate Composite Attribute

Method 1: Keep all simple attributes & discard the composite attribute (in the same relation)

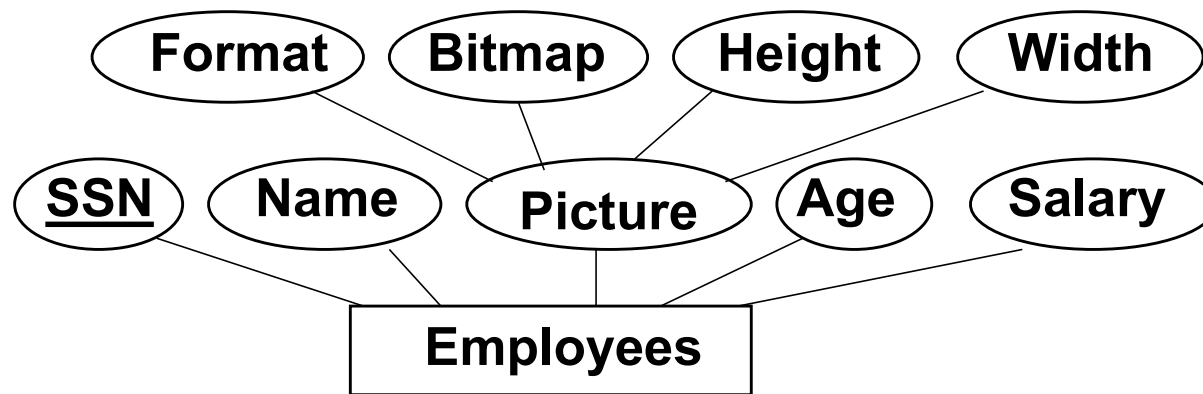


Method 2: Translate the composite attribute to a separate relation.

$E(\underline{A}, C)$
 $E_B(\underline{A}, D, H)$

Translate Composite Attribute (cont.)

An Example Using Method 2:

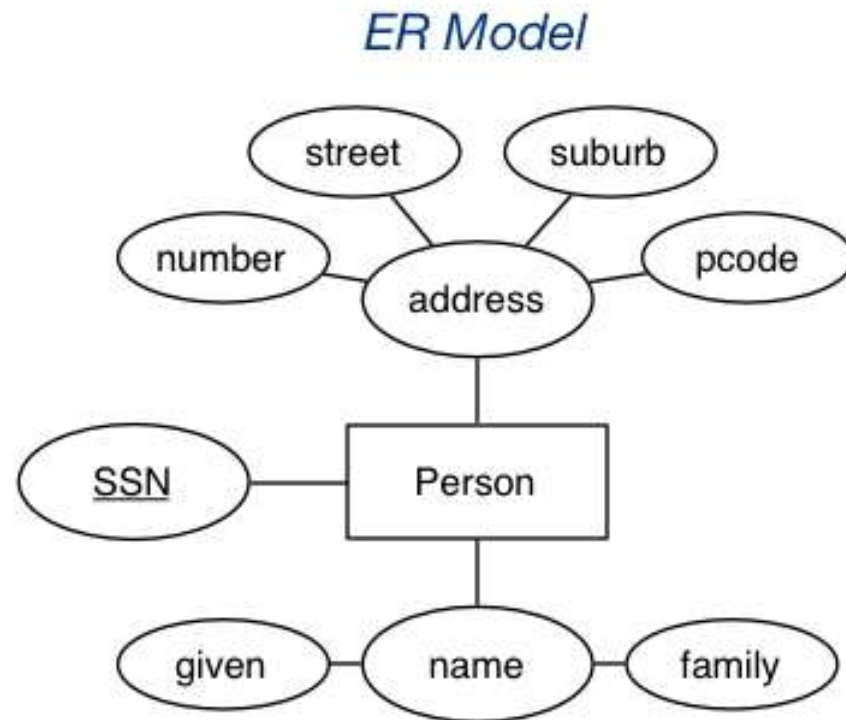


Employees (SSN, Name, Age, Salary)

Emp_Pic (SSN, Bitmap, Format, Height, Width)

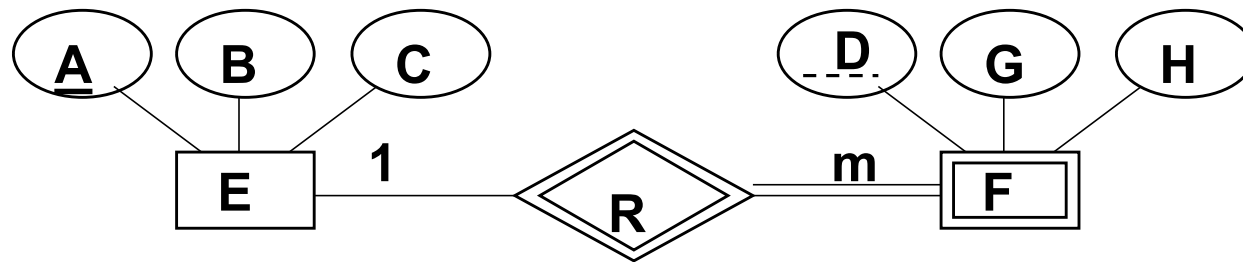
Exercise

Convert to a Relational Table



Translate Weak Entity Type

Translate weak entity type

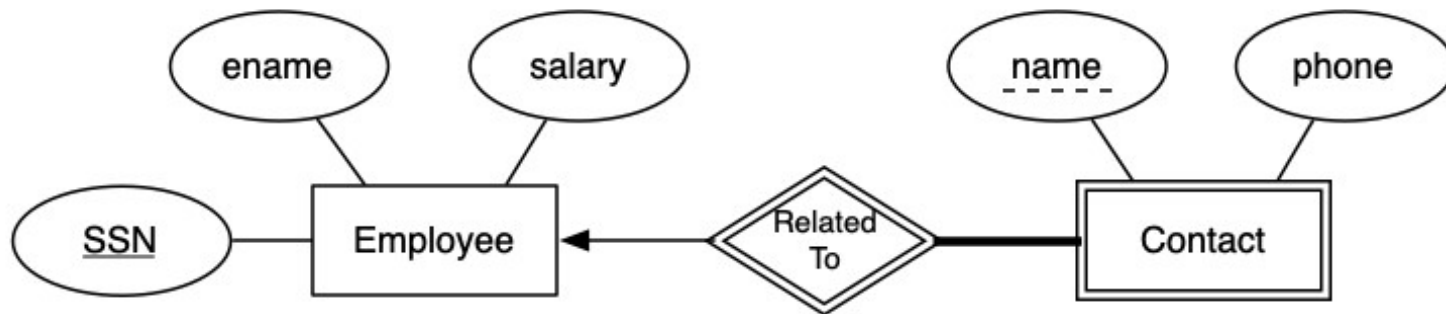


into $E(\underline{A}, B, C), F(\underline{A}, \underline{D}, G, H)$

- ☛ Use the key of strong entity & the partial key of weak entity to form the key.
- ☛ The key of strong entity is a foreign key.

Exercise

Convert to a Relational Table



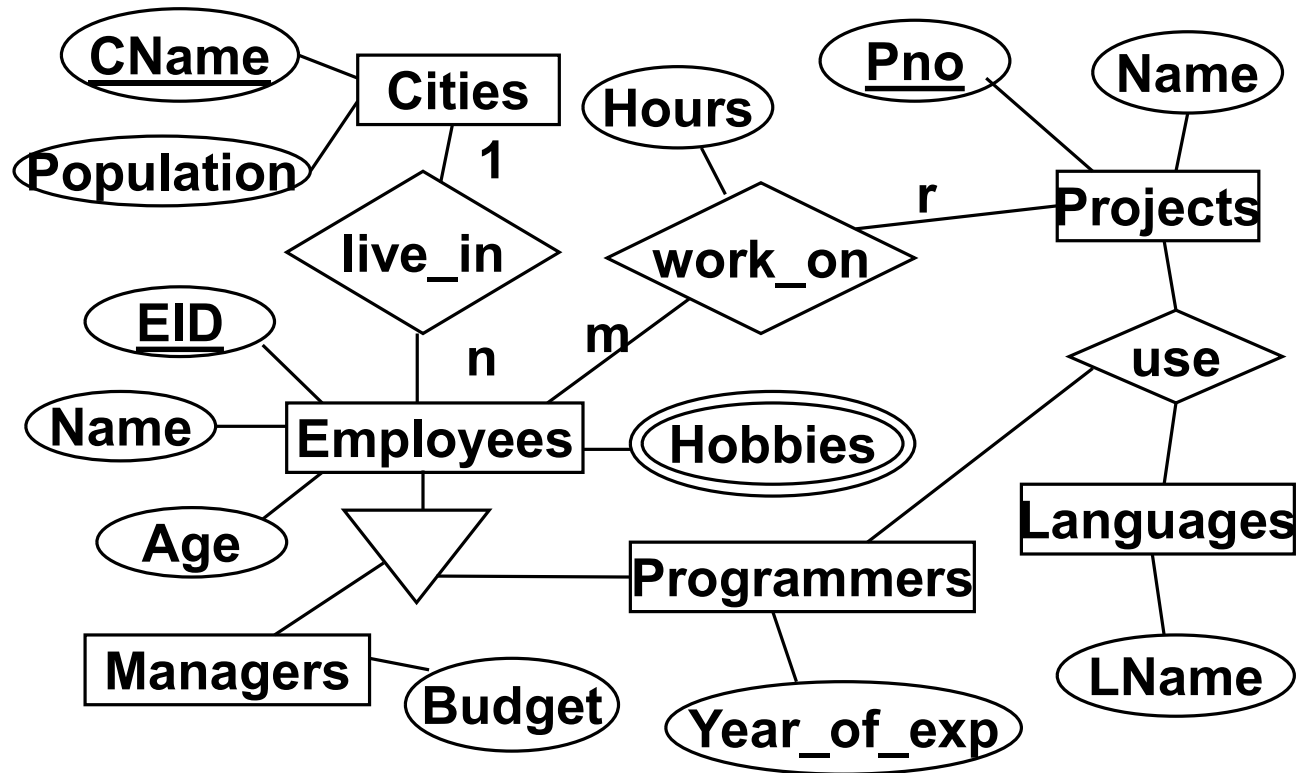
Translation Guidelines: Summary

- ◆ Translate entity types into relations (exclude multi-valued and composite attributes).
 - ▲ Translate multi-valued attributes into separate relations.
 - ▲ Translate composite attributes by keeping their simple attributes.
 - ▲ Specify a primary key for each relation.

Translation Guidelines: Summary

- ◆ Translate unary or binary 1-to-m (1-to-1) relationships into foreign keys
- ◆ Translate m-to-m or high degree (> 2) relationships into separate relations. Specify key and foreign key carefully.

Example



Example

◆ Use method 1:

Employees(EID Name, Age, Cname)

Employee-Hobby(EID, Hobby)

Managers(EID, Budget)

Programmers(EID, Years_of_exp)

Cities(CName, Population)

Projects(Pno, Name)

Languages(LName)

Work_on(EID, Pno, Hours)

Use(EID, Pno, LName)

Example

◆ Use method 2:

Employees(EID, Name, Age, CName)

Employee-Hobby(EID, Hobby)

Managers(Manager-EID, Name, Age, Budget, CName)

Manager-Hobby(Manager-EID, Hobby)

Programmers(Programmer-EID, Name, Age, Years_of_experience, CName)

Programmer-Hobby(Programmer-EID, Hobby)

Example

Cities(CName, Population)

Projects(Pno, Name)

Languages(LName)

Work_on(EID, Pno, Hours)

Manager-Work_on(Manager-EID, Pno, Hours)

Programmer-Work_on(Programmer-EID, Pno,
Hours)

Use(Programmer-EID, Pno, LName)

Exercise

Convert to a Relational Table

