

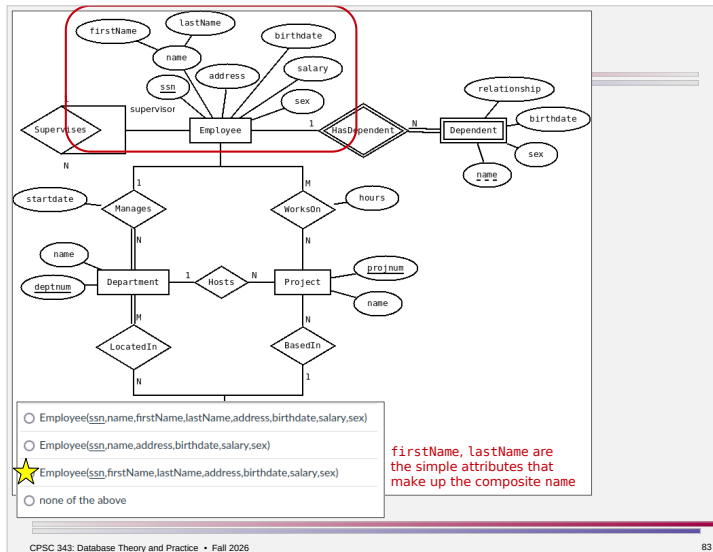
ER/EER → Relational Mapping

A series of steps –

- entity types (except multivalued attributes)
- weak entity types
- binary relationships: 1:1, 1:N, M:N
- multivalued attributes
- n-ary relationships
- specialization
- category types

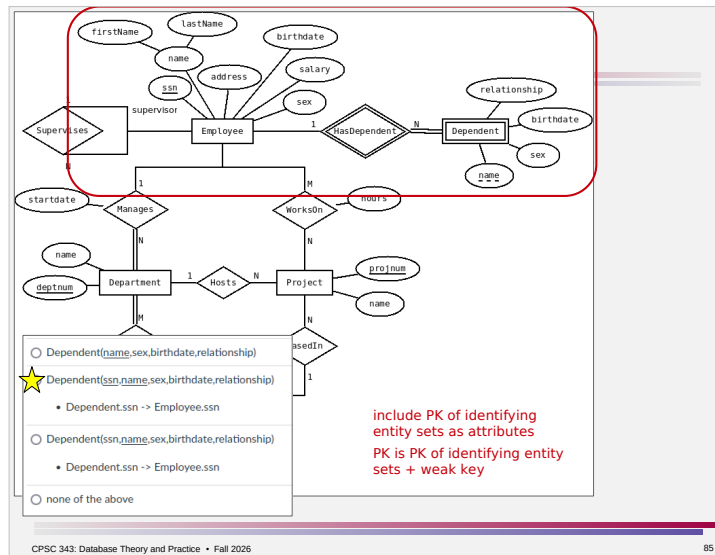
ER → Relational: Entity Types

1. entity type $E \rightarrow$ relation with all of the simple attributes of E
 - key of $E \rightarrow$ primary key



ER → Relational: Weak Entity Types

2. weak entity type $W \rightarrow$ relation with all of the simple attributes of W + all of the simple attributes of identifying relationships R + all key attributes of identifying entity sets E
 - partial key of W + key attributes of $E \rightarrow$ primary key
 - attributes stemming from key attributes of E are foreign keys referring to E 's relation



ER → Relational: Binary Relationship Types

3. **binary 1:1 relationship type**
 - if both entity types have total participation → use merged relation
 - if one entity type has total participation → use foreign key
 - if neither entity type has total participation → use foreign key or cross-reference
4. **binary 1:N relationship type** → use foreign key (with the relationship on the N side) or cross-reference
5. **binary M:N relationship type** → use cross-reference

merged relation approach – combine the relations for the two participating entity sets into one, along with the simple attributes of the relationship

foreign key approach – add the simple attributes of the relationship to the relation for one of the participating entity sets, along with the primary key of the relation for the other participating entity set

cross-reference approach – create a new relation with the simple attributes of the relationship and the primary keys of the relations for the participating entity sets ■

ER → Relational: Binary Relationship Types

Use the choice of primary key to enforce cardinality constraints.

$$\begin{matrix} S(\underline{s1}, s2) \\ T(\underline{t1}, t2) \end{matrix}$$

- merged relation approach

supports N:1 with repetition of T

supports 1:N with repetition of S

supports M:N with repetition of both – or **1:1 (no repetition)**

ST(s1,s2,t1,t2)
ST(s1,s2,t1,t2)
ST(s1,s2,t1,t2)

- foreign key approach

supports N:1

supports 1:N with repetition of S

supports M:N with repetition of S

$$\left. \begin{array}{l} S(s_1, s_2, t_1) \\ S(s_1, s_2, \underline{t_1}) \\ S(s_1, s_2, \underline{t_1}) \end{array} \right\} T(\underline{t_1}, t_2)$$

- cross-reference approach

supports N:1

supports 1:N

supports M:N

$$\left. \begin{array}{l} ST(\underline{s1}, \underline{t1}) \\ ST(\underline{s1}, \underline{t1}) \\ \mathbf{ST(\underline{s1}, \underline{t1})} \end{array} \right\} \begin{array}{l} S(\underline{s1}, s2) \\ T(\underline{t1}, t2) \end{array}$$

ER → Relational: Binary Relationship Types

Participation.

$$\begin{array}{l} S(\underline{s1}, s2) \\ T(\underline{t1}, t2) \end{array}$$

- merged relation approach

requires total participation of both

ST(s1, s2, t1, t2)

- foreign key approach

partial participation of S results in NULL values

$$S(\underline{s1}, s2, t1) \quad T(\underline{t1}, t2)$$

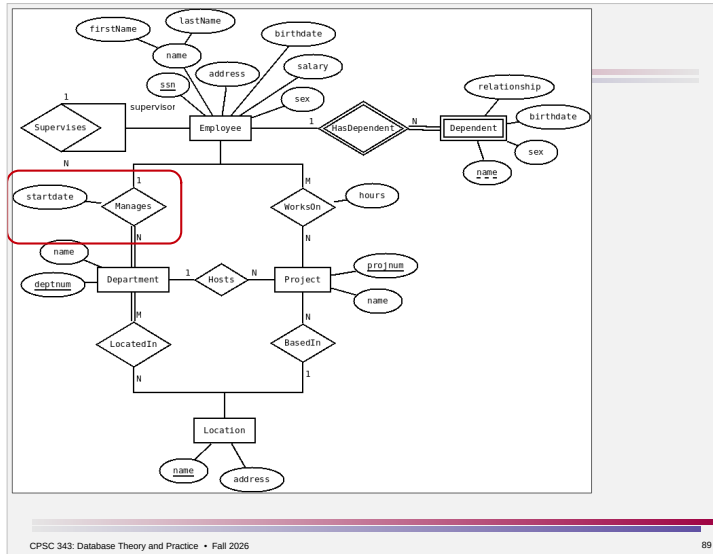
cannot require total participation of T

- cross-reference approach

no NULLs – ST only contains entries if there is an instance of the relationship type

$$\text{ST}(\underline{s1}, \underline{t1}) \quad \begin{array}{l} S(\underline{s1}, s2) \\ T(\underline{t1}, t2) \end{array}$$

cannot require total participation of S or T



Step 4 of the ER-to-relational mapping process means that Manages will end up as: (choose all that are appropriate - the constraints from the ER diagram are captured without drawbacks like redundant information or unnecessary NULL values)

- ☐ A single relation for Employee, Department, and Manages:
Employee(ssn,empname,address,birthdate,salary,sex,deptnum,deptname,startdate)
- ☐ A single relation for Employee, Department, and Manages:
Department(deptnum,deptname,ssn,empname,address,birthdate,salary,sex,startdate)
• Department.ssn NOT NULL
- ☐ Department(deptnum,deptname,ssn,startdate)
• Department.ssn -> Employee.ssn
- ☐ Department(deptnum,deptname,ssn,startdate)
• Department.ssn -> Employee.ssn
• Department.ssn NOT NULL
- ☐ Employee(ssn,firstName,lastName,address,birthdate,salary,sex,deptnum,startdate)
• Employee.deptnum -> Department.deptnum
• Employee.deptnum NOT NULL
- ☐ Employee(ssn,firstName,lastName,address,birthdate,salary,sex,deptnum,startdate)
• Employee.deptnum -> Department.deptnum
• Employee.deptnum NOT NULL

merged relation is not suitable
- cannot accommodate employees who aren't managers

foreign key approach
- relationship goes with N side (Department)
- at most 1 manager is enforced by PK being PK of N side (Department)
- total participation of Department means no NULLs for ssn, startdate

cross-reference approach
- at most 1 manager is enforced by PK being PK of N side (Department)
- does not enforce total participation of Department

- ☐ Manages(ssn,deptnum,startdate)
• Manages.ssn -> Employee.ssn
• Manages.deptnum -> Department.deptnum
- ☐ Manages(ssn,deptnum,startdate)
• Manages.ssn -> Employee.ssn
• Manages.deptnum -> Department.deptnum
• Manages.deptnum NOT NULL
- ☐ Manages(ssn,deptnum,startdate)
• Manages.ssn -> Employee.ssn
• Manages.deptnum -> Department.deptnum
• Manages.ssn NOT NULL

ER → Relational: Multivalued Attributes

6. multivalued attribute A on entity type E → relation with attribute A and primary key of E's relation
all attributes → primary key

- ☐ Club(clubname,role,name)
- ☐ Club(name,officers)
- ☐ Club(name)
Officers(clubname,role,name)
• Officers.clubname -> Club.name

entity set → relation with all of the simple attributes of the entity set
multivalued attribute → separate relation with the attribute and the PK of the relation for the entity set

CPSC 343: Database Theory and Practice • Fall 2026 92

ER → Relational: n-Ary Relationship Types

7. n-ary relationship type → extension of cross-reference approach for binary relationship types
- i.e. separate relation for the relationship, with attributes corresponding to the simple attributes of the relationship and the PK of all of the participating entity sets
 - can adjust PK to capture some cardinality constraints
 - can't capture total participation

EER → Relational: Specialization

Specialization → one of four options

- multiple relations**
 - superclass and subclass* – superclass relation has attributes of its entity type (PK = entity type's key); subclass relations have attributes of their entity types plus PKs of the superclass relation (as FK; PK = superclass PK)
 - subclass only* – subclass relations have attributes of their entity types and of the superclass entity type (PK = superclass key)
- single relation** – single relation with attributes from superclass and all subclasses (PK = superclass key)
 - single type attribute* – plus a single type attribute which identifies which subclass an entity belongs to
 - multiple type attributes* – plus a boolean type attribute for each subclass

EER → Relational: Specialization

8. specialization → one or more relations

		type of specialization				
		partial	total	disjoint	overlap	
multiple relations	superclass and subclass	y	y	y	y	
	subclass only	x	y	y	entities may be duplicated	
single relation	single type attribute	y	y	y	x	subclass-only attributes result in NULL values
	multiple type attributes	y	y	inconsistency potential	y	

```

graph TD
    Player((Player)) --- name((name))
    Player --- height((height))
    Player --- weight((weight))
    Player --| Offense((Offense))
    Player --| Defense((Defense))
    Offense --- goals((goals))
    Offense --- assists((assists))
    Defense --- tackles((tackles))
    Defense --- fouls((fouls))
    Defense --- offsides((offsides))
    
```

Player(name,height,weight)
 Offense(goals,assists,name)
 Defense(tackles,fouls,offsides,name)

- Offense.name → Player.name
- Defense.name → Player.name

multiple classes (subclass) should also have height, weight for both relations ~ redundant info for players who are both

Offense(goals,assists,name)
 Defense(tackles,fouls,offsides,name)

single relation (single type attribute) can't accommodate players who are both

Player(name,height,weight,goals,assists,tackles,fouls,offsides,isOffense,isDefense)

total specialization with overlap → multiple classes (superclass and subclass) and single relation (multiple type attributes)

EER → Relational: Category Types

9. category types → relation with any attributes of the category itself plus the surrogate or superclass key (as FKs in the superclasses)

surrogate/superclass key → primary key

(surrogate key is added to superclasses as FK if they do not have a common key)

