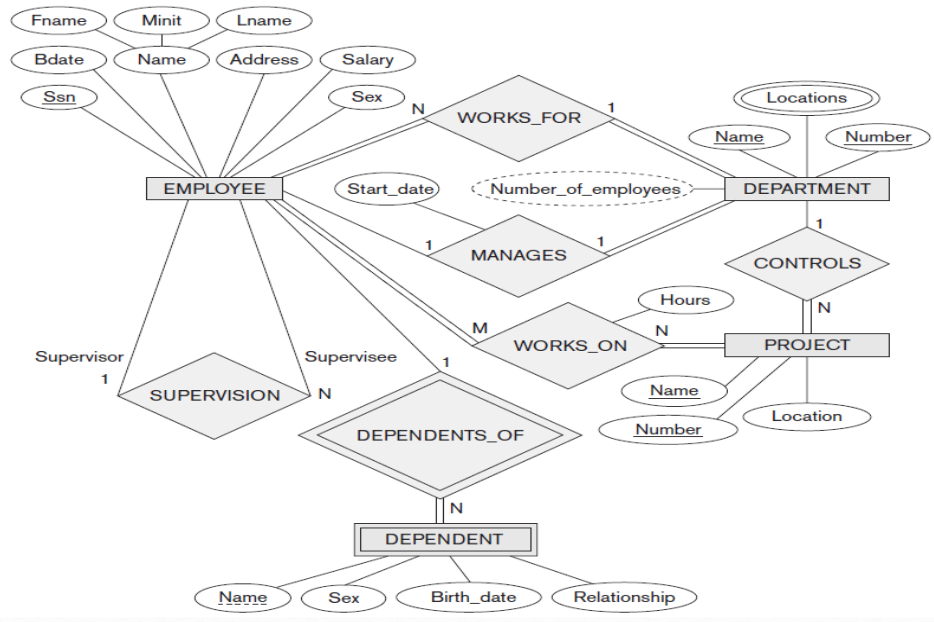
CSIE30600/CSIEB0290 Database Systems

Lecture 9: ER/EER to Relational Mapping

Outline

- **ER-to-Relational Mapping Algorithm**
  - Step 1: Mapping of Regular Entity Types
  - Step 2: Mapping of Weak Entity Types
  - Step 3: Mapping of Binary 1:1 Relation Types
  - Step 4: Mapping of Binary 1:N Relationship Types.
  - Step 5: Mapping of Binary M:N Relationship Types.
  - Step 6: Mapping of Multivalued attributes.
  - Step 7: Mapping of N-ary Relationship Types.
- **Mapping EER Model Constructs to Relations**
  - Step 8: Options for Mapping Specialization or Generalization.
  - Step 9: Mapping of Union Types (Categories).

Figure 9.1
The ER conceptual schema diagram for the COMPANY database.



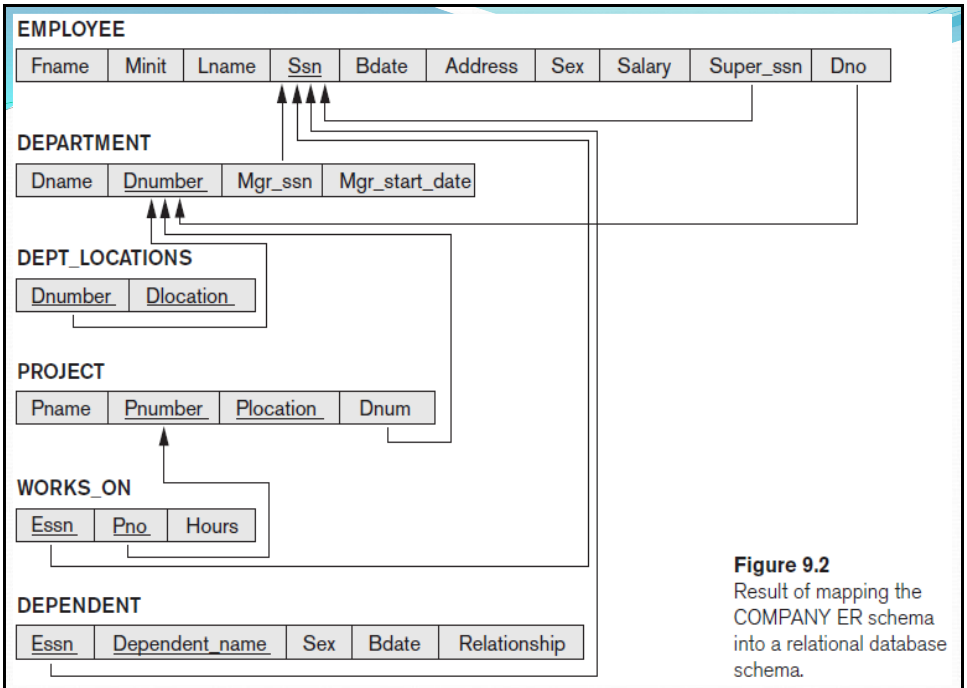


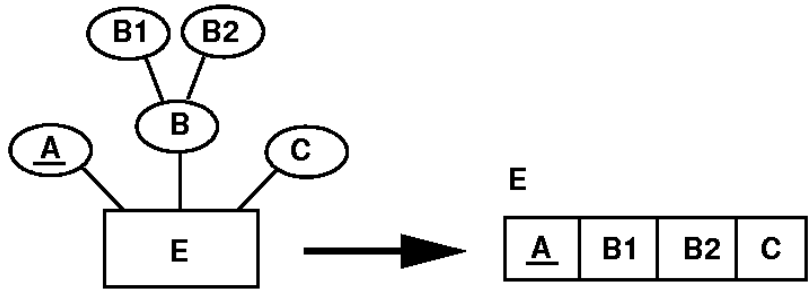
Figure 9.2
Result of mapping the COMPANY ER schema into a relational database schema.

Mapping Algorithm (Step 1)

- **Step 1: Mapping of Regular Entity Types.**
  - For each regular (strong) entity type E, create a relation R that includes all the simple attributes of E.
  - Choose one of the keys of E as the primary key for R.
  - If the chosen key of E is composite, the set of simple attributes together form the primary key of R.
- Example: Create the relations EMPLOYEE, DEPARTMENT, and PROJECT corresponding to the regular entities in the ER diagram.
  - SSN, DNUMBER, and PNUMBER are the primary keys for the relations EMPLOYEE, DEPARTMENT, and PROJECT as shown.

ER to Relational - Step 1

- **Step 1**: Process regular entity types. (**entity relations**)

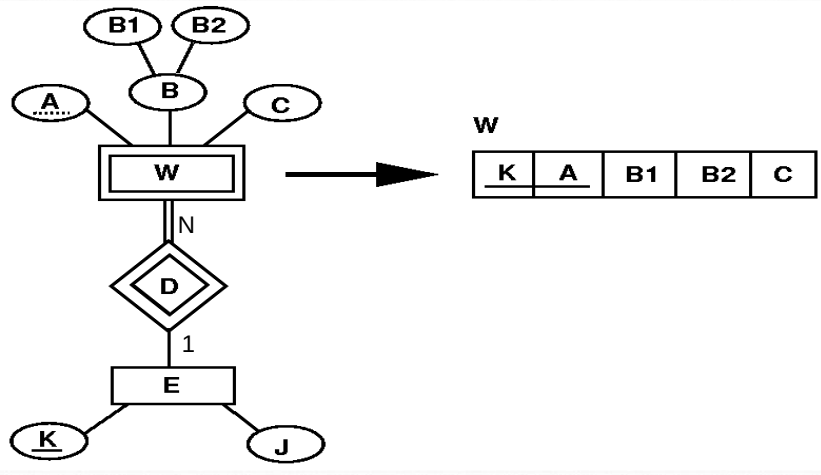


Mapping Algorithm (Step 2)

- **Step 2: Mapping of Weak Entity Types**
  - For each weak entity type W in the ER schema with owner entity type E, create a relation R & include all simple attributes (or simple components of composite attributes) of W as attributes of R.
  - Also, include as foreign key attributes of R the primary key attribute(s) of the owner(s).
  - The primary key of R is the combination of the primary key(s) of the owner(s) and the partial key of the weak entity type W, if any.

ER to Relational - Step 2

- **Step 2**: Process weak entity types.



Step 2 (contd.)

- **Example:** Create the relation DEPENDENT in this step to correspond to the weak entity type DEPENDENT.
  - Include the primary key SSN of the EMPLOYEE relation as a foreign key attribute of DEPENDENT (renamed to ESSN).
  - The primary key of the DEPENDENT relation is the combination {ESSN, DEPENDENT_NAME} because DEPENDENT_NAME is the partial key of DEPENDENT.

ER-to-Relational Mapping Algorithm (cont.)

Figure 9.3
Illustration of some mapping steps.
(a) *Entity* relations after step 1.
(b) Additional *weak entity* relation after step 2.
(c) *Relationship* relations after step 5.
(d) Relation representing multivalued attribute after step 6.

(a) **EMPLOYEE**

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary

DEPARTMENT

Dname	Dnumber

PROJECT

Pname	Pnumber	Plocation

(b) **DEPENDENT**

Essn	Dependent_name	Sex	Bdate	Relationship

(c) **WORKS_ON**

Essn	Pno	Hours

(d) **DEPT_LOCATIONS**

Dnumber	Dlocation

Mapping Algorithm (Step 3)

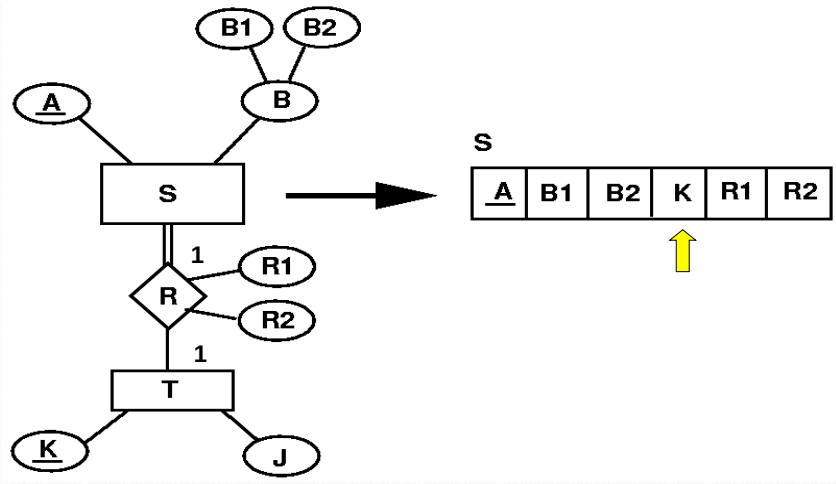
- **Step 3: Mapping of Binary 1:1 Relation Types**
  - For each binary 1:1 relationship type R in the ER schema, identify the relations S and T that correspond to the entity types participating in R.
- Three possible approaches:
  1. **Foreign Key approach**: Choose one of the relations-say S-and include a foreign key in S the primary key of T. It is better to choose an entity type with total participation in R in the role of S.
     - **Example**: 1:1 relation MANAGES is mapped by choosing the participating entity type DEPARTMENT to serve in the role of S, because its participation in the MANAGES relationship type is total.

Step 3 (cont.)

2. **Merged relation option**: An alternate mapping of a 1:1 relationship type is possible by merging the two entity types and the relationship into a single relation. This may be appropriate when **both participations are total**.
3. **Cross-reference or relationship relation option**: The third alternative is to set up a **third relation** R for the purpose of cross-referencing the primary keys of the two relations S and T representing the entity types.

ER to Relational - Step 3

- **Step 3**: Process 1:1 relationships.

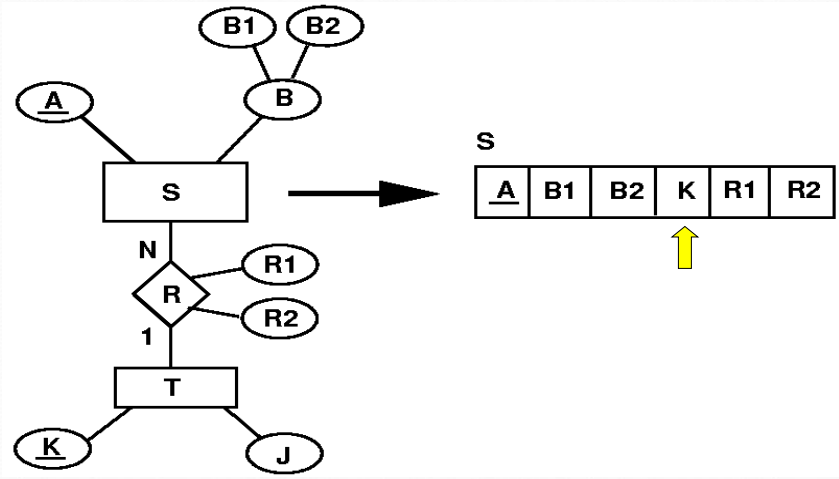


Mapping Algorithm (Step 4)

- **Step 4: Mapping of Binary 1:N Relationship Types.**
  - For each regular binary 1:N relationship type R, identify the relation S that represent the participating entity type at the N-side of the relationship type.
  - Include as foreign key in S the primary key of the relation T that represents the other entity type participating in R.
  - Include any simple attributes of the 1:N relation type as attributes of S.

ER to Relational - Step 4

- **Step 4**: Process 1:N relationships.



Step 4 (cont.)

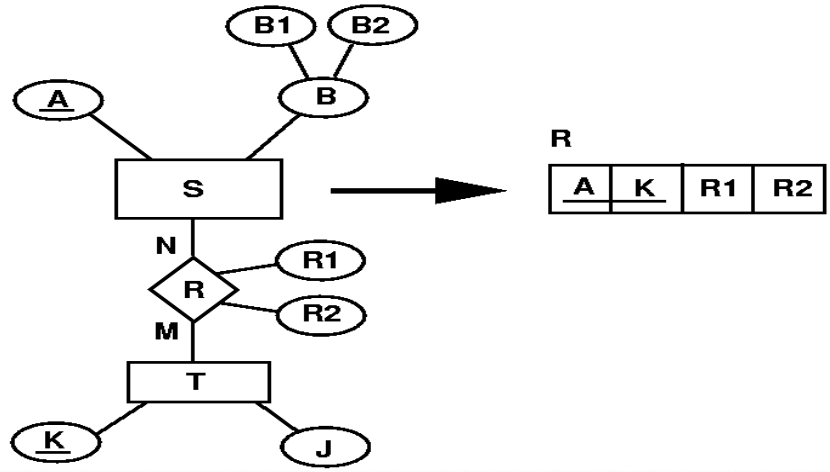
- Example: 1:N relationship types WORKS_FOR, CONTROLS, and SUPERVISION in the figure.
  - For WORKS_FOR we include the primary key DNUMBER of the DEPARTMENT relation as foreign key in the EMPLOYEE relation and call it DNO.

Mapping Algorithm (Step 5)

- **Step 5: Mapping of Binary M:N Relationship Types.**
  - For each regular binary M:N relationship type R, create a new relation S to represent R.
  - Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types; *their combination will form the primary key* of S.
  - Also include any simple attributes of the M:N relationship type (or simple components of composite attributes) as attributes of S.

ER to Relational - Step 5

- **Step 5**: Process M:N relationships.



Step 5 (contd.)

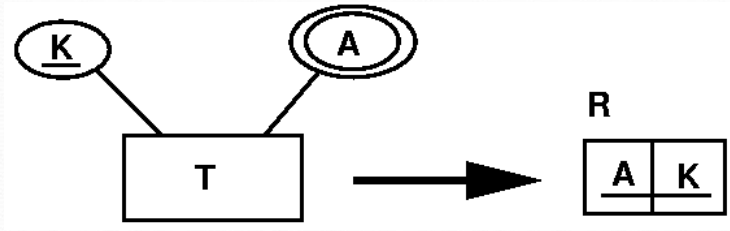
- Example: The M:N relationship type WORKS_ON from the ER diagram is mapped by creating a relation WORKS_ON in the relational database schema.
  - The primary keys of the PROJECT and EMPLOYEE relations are included as foreign keys in WORKS_ON and renamed PNO and ESSN, respectively.
  - Attribute HOURS in WORKS_ON represents the HOURS attribute of the relation type. The primary key of the WORKS_ON relation is the combination of the foreign key attributes {ESSN, PNO}.

Mapping Algorithm (Step 6)

- **Step 6: Mapping of Multivalued attributes.**
  - For each multivalued attribute A, create a new relation R.
  - This relation R will include an attribute corresponding to A, plus the primary key attribute K-as a foreign key in R-of the relation that represents the entity type of relationship type that has A as an attribute.
  - The primary key of R is the combination of A and K. If the multivalued attribute is composite, we include its simple components.

ER to Relational - Step 6

- **Step 6**: Process multivalued attributes.



Step 6 (contd.)

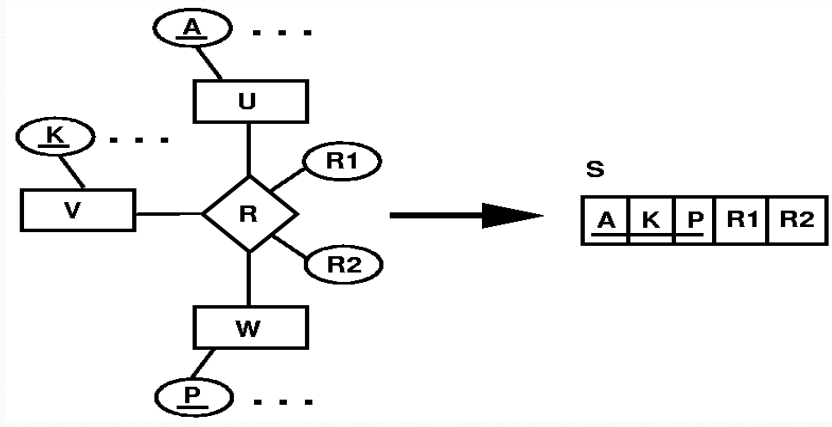
- **Example:** The relation DEPT_LOCATIONS is created.
  - The attribute DLOCATION represents the multivalued attribute LOCATIONS of DEPARTMENT, while DNUMBER-as foreign key-represents the primary key of the DEPARTMENT relation.
  - The primary key of R is the combination of {DNUMBER, DLOCATION}.

Mapping Algorithm (Step 7)

- **Step 7: Mapping of N-ary Relationship Types.**
  - For each n-ary relationship type R, where n>2, create a new relation S to represent R.
  - Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types.
  - Also include any simple attributes of the n-ary relationship type (or simple components of composite attributes) as attributes of S.

ER to Relational - Step 7

- **Step 7**: Process n-array relationship type for n > 2.



Summary of Mapping for ER Model Constructs

Table 9.1 Correspondence between ER and Relational Models

ER MODEL	RELATIONAL MODEL
Entity type	*Entity* relation
1:1 or 1:N relationship type	Foreign key (or *relationship* relation)
M:N relationship type	*Relationship* relation and *two* foreign keys
n-ary relationship type	*Relationship* relation and *n* foreign keys
Simple attribute	Attribute
Composite attribute	Set of simple component attributes
Multivalued attribute	Relation and foreign key
Value set	Domain
Key attribute	Primary (or secondary) key

Mapping EER Model Constructs to Relations

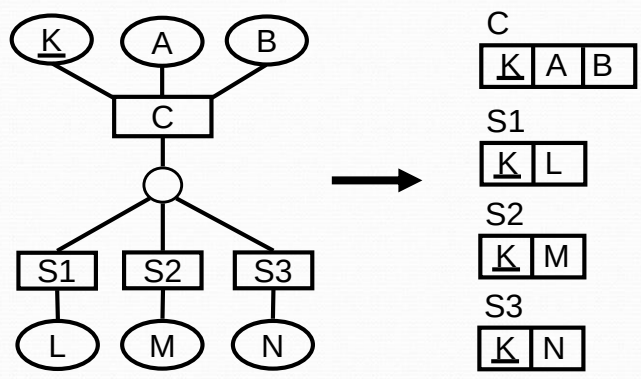
- **Step 8: Mapping Specialization or Generalization.**
  - Convert each specialization with m subclasses {S1,....,Sm} and generalized superclass C with attributes {k,a1,...an} and k is the (primary) key, into relational schemas:
    - Option 8A: Multiple relations-Superclass and subclasses
    - Option 8B: Multiple relations-Subclass relations only
    - Option 8C: Single relation with one type attribute
    - Option 8D: Single relation with multiple type attributes

Mapping EER Model Constructs to Relations

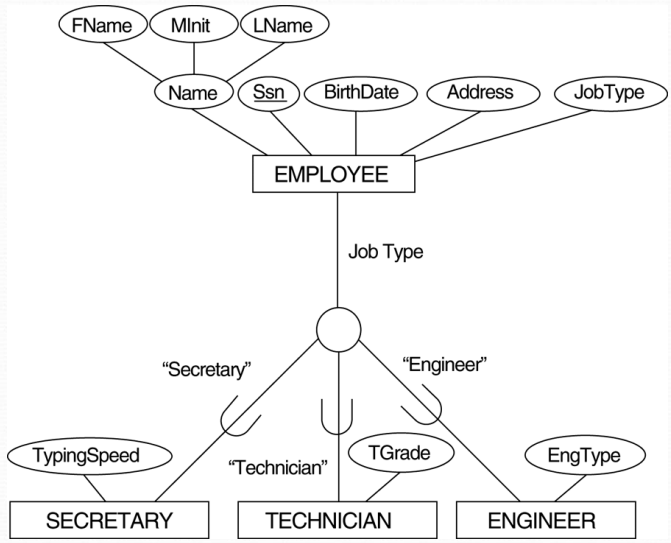
- **Option 8A: Multiple relations-Superclass and subclasses**
  - Create a relation L for C with attributes Attrs(L) = {k,a1,...an} and PK(L) = k. Create a relation Li for each subclass Si, 1 < i < m, with the attributes Attrs(Li) = {k} U {attributes of Si} and PK(Li)=k. This option works for any specialization (total or partial, disjoint of over-lapping).

EER to Relational - Step 8

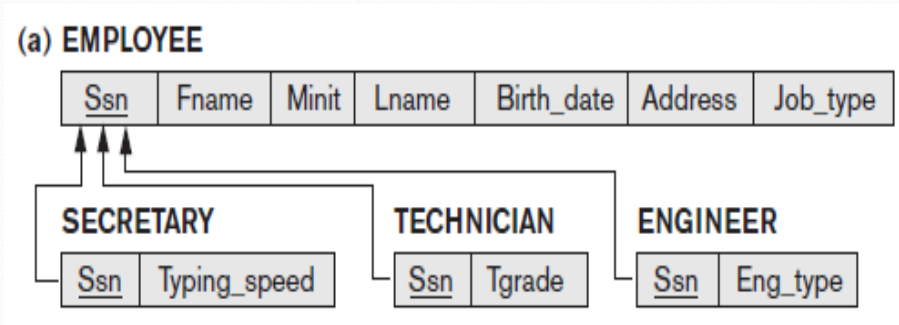
- **Step 8**: Process superclass/subclass relationships.
  - **Option 8A**: Create a relation for the superclass and a relation for each subclass with key attributes from the superclass.



EER diagram notation for an attribute-defined specialization on JobType.



Options for mapping specialization or generalization. (a) Mapping the EER schema in last slide using option 8A.

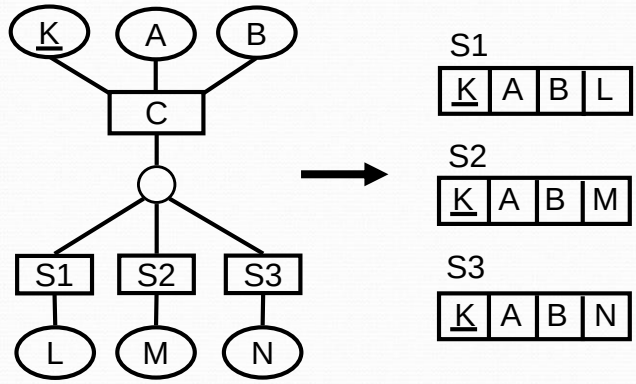


EER to Relational - Option 8B

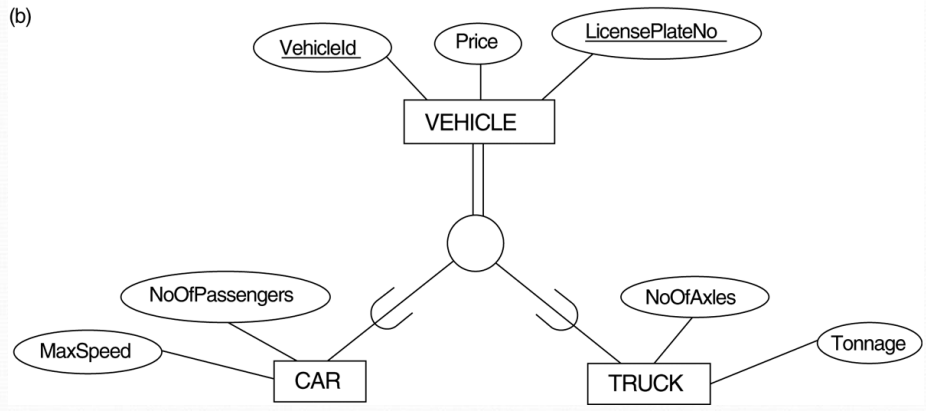
- **Option 8B: Multiple relations-Subclass relations only**
  - Create a relation Li for each subclass Si, 1 < i < m, with the attributes Attr(Li) = {attributes of Si} U {k,a1...,an} and PK(Li) = k. This option only works for a specialization whose subclasses are total (every entity in the superclass must belong to (at least) one of the subclasses).

EER to Relational - Option 8B

- **Option 8B**: Create a relation for each subclass with all attributes of the superclass.



Generalization. (b) Generalizing CAR and TRUCK into the superclass VEHICLE.



Options for mapping specialization or generalization.
(b) Mapping the EER schema using option 8B.

(b) **CAR**

Vehicle_id	License_plate_no	Price	Max_speed	No_of_passengers

TRUCK

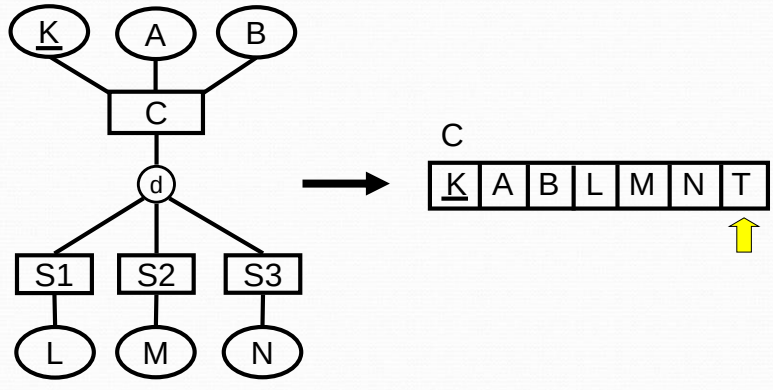
Vehicle_id	License_plate_no	Price	No_of_axles	Tonnage

EER to Relational - Option 8C

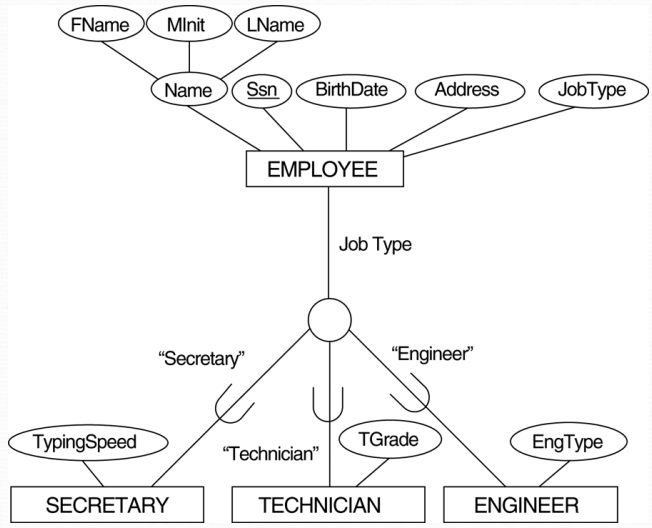
- **Option 8C: Single relation with one type attribute**
  - Create a single relation L with attributes Attrs(L) = {k,a_1,...a_n} U {attributes of S_1} U…U {attributes of S_m} U {t} and PK(L) = k. The attribute t is called a type (or **discriminating**) attribute that indicates the subclass to which each tuple belongs

EER to Relational - Option 8C

- **Option 8C**: Create a single relation with all attributes of superclass and subclasses, and a **type attribute**. (for *disjoin specialization*)



EER diagram notation for an attribute-defined specialization on JobType.



Options for mapping specialization or generalization.
(c) Mapping the EER schema in last slide using option 8C.

(c) **EMPLOYEE**

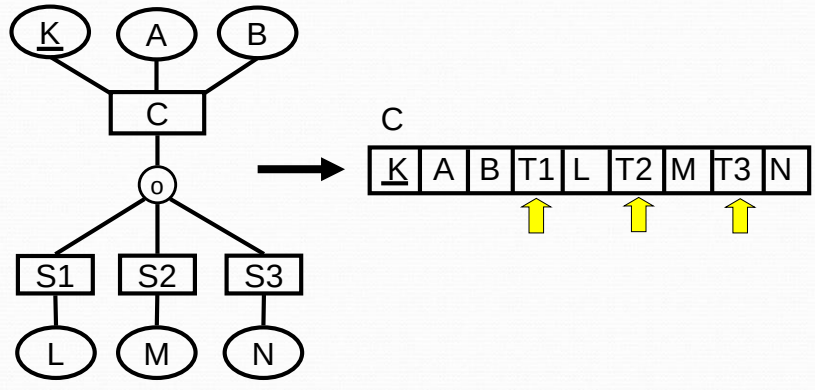
Ssn	Fname	Minit	Lname	Birth_date	Address	Job_type	Typing_speed	Tgrade	Eng_type

EER to Relational - Option 8D

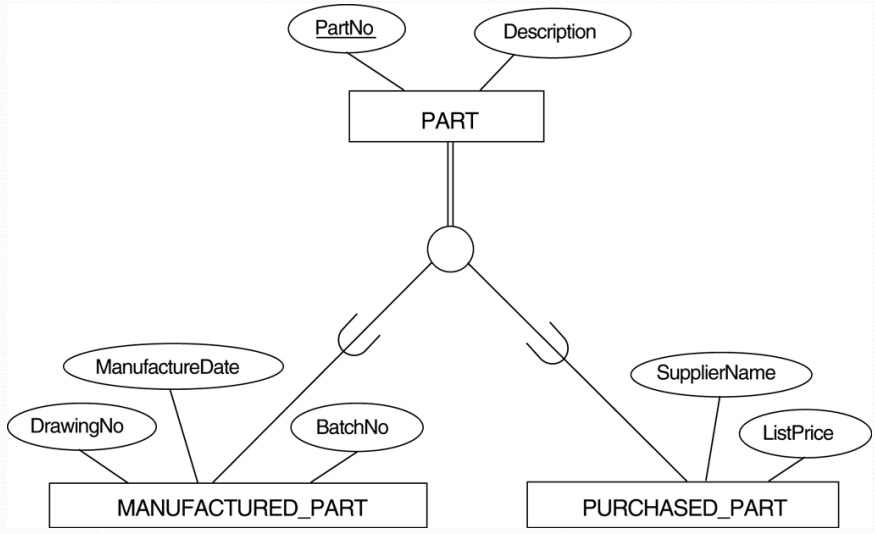
- **Option 8D: Single relation with multiple type attributes**
  - Create a single relation L with attributes Attrs(L) = {k,a_1,...a_n} U {attributes of S_1} U...U {attributes of S_m} U {t_1, t_2,…,t_m} and PK(L) = k. Each t_i, 1 < I < m, is a Boolean type attribute indicating whether a tuple belongs to the subclass S_i.

EER to Relational - Option 8D

- **Option 8D**: Create a single relation with all attributes of superclass and subclasses, and a set of **Boolean attributes**, one for each subclass. (for *overlapping specialization*)



EER diagram notation for an overlapping (non-disjoint) specialization.



Options for mapping specialization or generalization.
(d) Using option 8D with Boolean type fields Mflag and Pflag.

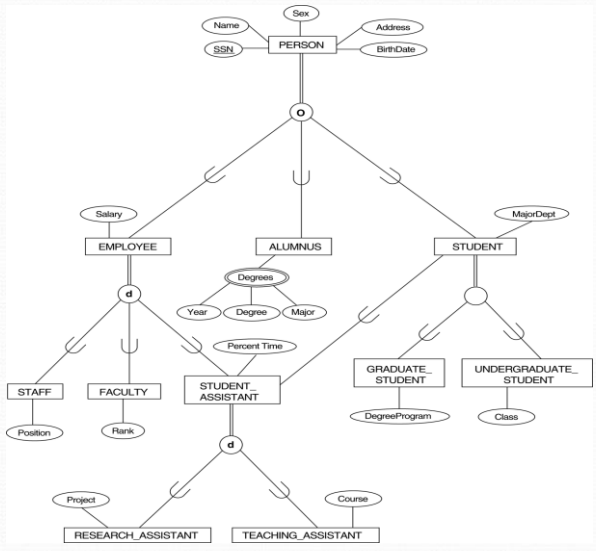
(d) PART

Part_no	Description	Mflag	Drawing_no	Manufacture_date	Batch_no	Pflag	Supplier_name	List_price

Mapping of Shared Subclasses (Multiple Inheritance)

- **Mapping of Shared Subclasses**
  - A shared subclass is a subclass of several classes, indicating multiple inheritance. These classes must all have the same key attribute; otherwise, the shared subclass would be modeled as a category.
  - We can apply any of the options discussed in Step 8 to a shared subclass, subject to the restriction discussed in Step 8 of the mapping algorithm. Below both 8C and 8D are used for the shared class STUDENT_ASSISTANT.

A specialization lattice with multiple inheritance for a UNIVERSITY database.



Mapping the EER specialization lattice using multiple options.

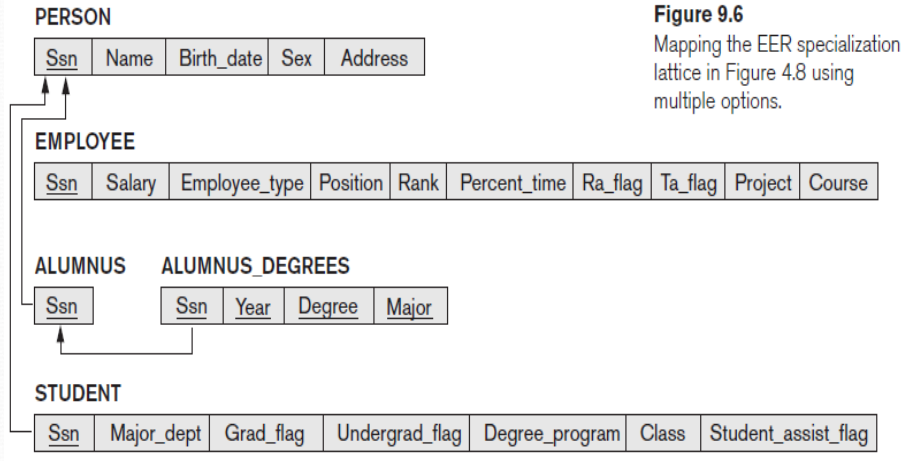
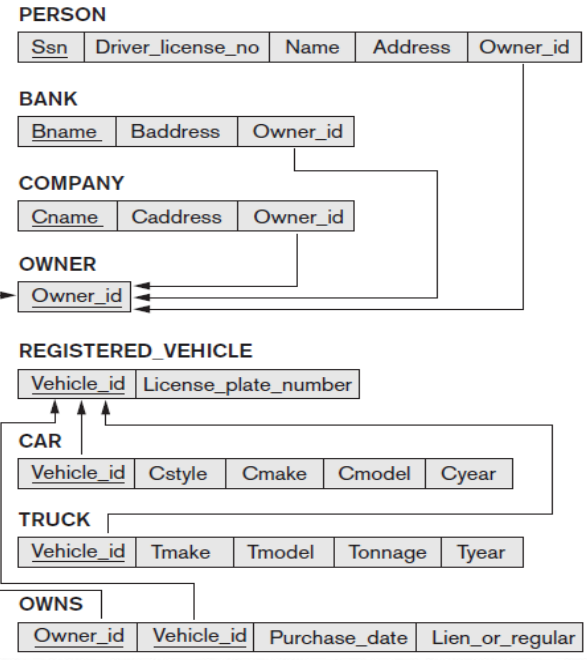


Figure 9.6
Mapping the EER specialization lattice in Figure 4.8 using multiple options.

Mapping of Categories (Union Types) – Step 9

- **Step 9: Mapping of Union Types (Categories)**
  - Defining superclasses have different keys
  - Specify a new key attribute
    - **Surrogate key**

Figure 9.7
Mapping the EER categories (union types) in Figure 4.8 to relations.



Summary

- **ER-to-Relational Mapping Algorithm**
  - Step 1: Mapping of Regular Entity Types
  - Step 2: Mapping of Weak Entity Types
  - Step 3: Mapping of Binary 1:1 Relation Types
  - Step 4: Mapping of Binary 1:N Relationship Types.
  - Step 5: Mapping of Binary M:N Relationship Types.
  - Step 6: Mapping of Multivalued attributes.
  - Step 7: Mapping of N-ary Relationship Types.
- **EER Model Constructs to Relations**
  - Step 8: Options for Mapping Specialization or Generalization.
  - Step 9: Mapping of Union Types (Categories).