COS221
L14 - (E)ER to Relational Mapping
(Chapter 9 in Editions 6 and 7)

Linda Marshall

13 March 2023

(E)ER-to-Relational Mapping Algorithm

Step 1: Mapping of regular (strong) entity types

Step 2: Mapping of weak entity types

Step 3: Mapping of binary 1:1 relationships

Step 4: Mapping of binary 1:N relationships

Step 5: Mapping of binary M:N relationships

Step 6: Mapping of multivalued attributes

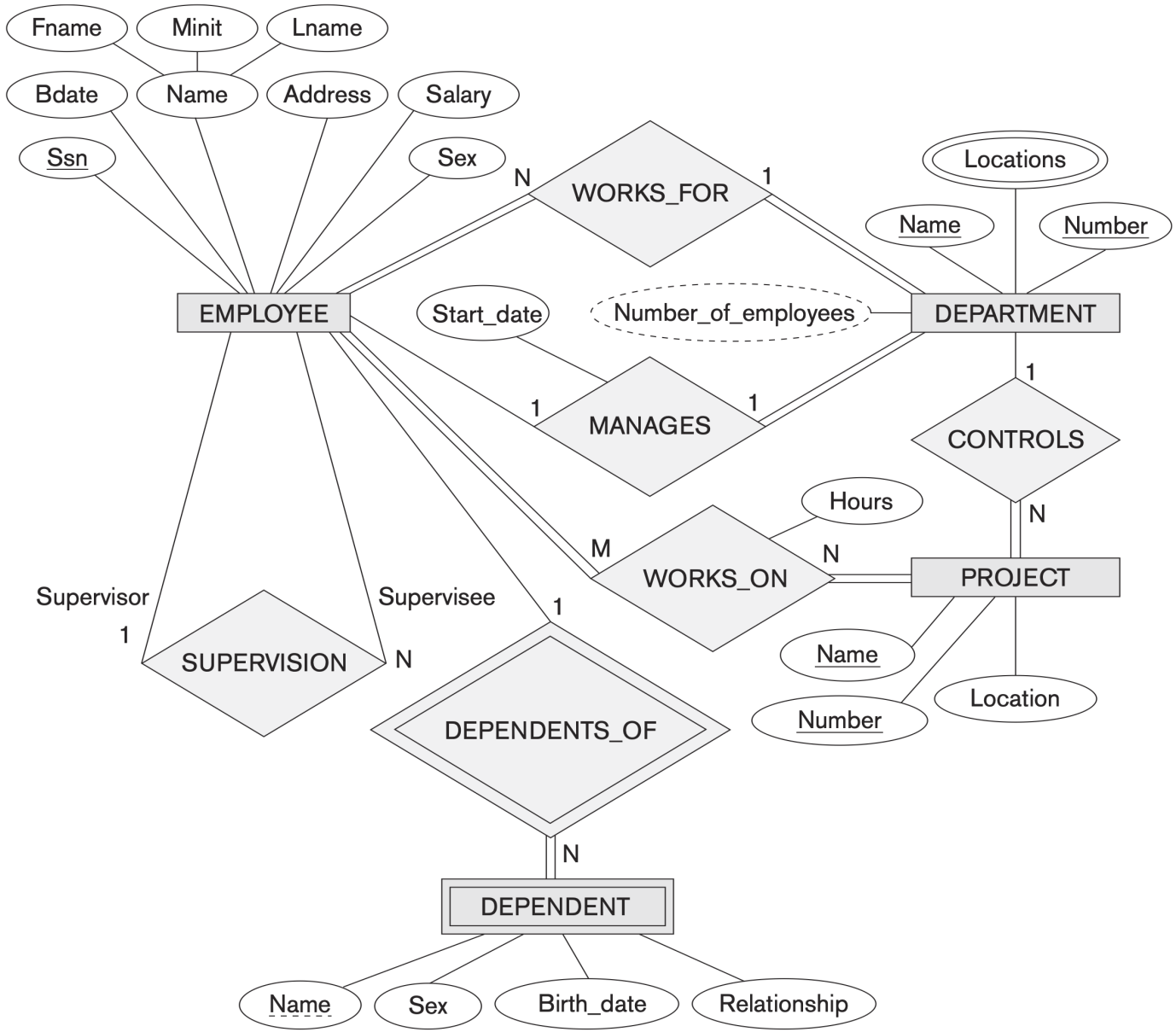
Step 7: Mapping of N-ary relationships

Step 8: Mapping specialisation and generalisation

Step 9: Mapping unions

The example

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

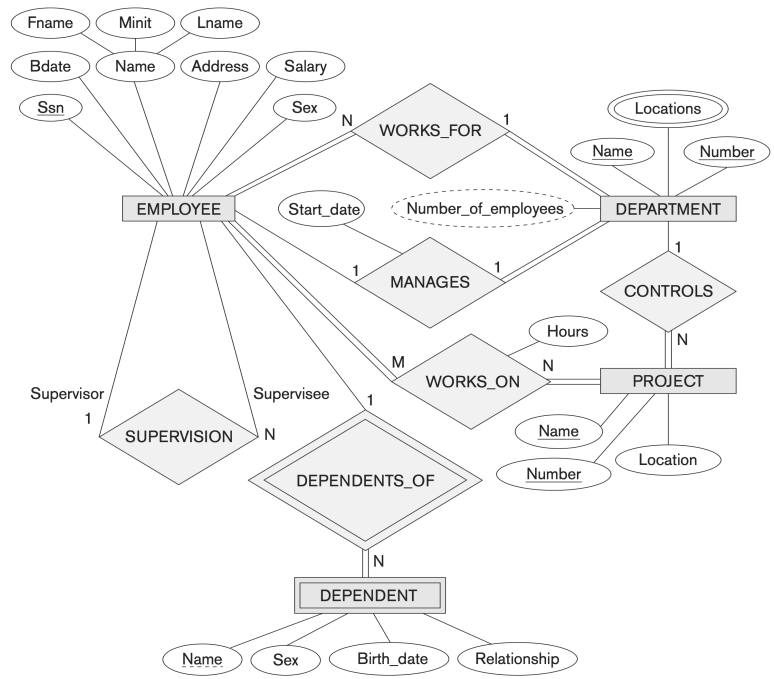


Step 1: Mapping of regular (strong) entity types

For each regular entity type, E, in the ER schema

- Create a relation R
- Add simple attributes of E to R
- Include the simple component attributes of composite attributes
- Choose one of the key attributes of E for R. If the attribute is a composite, then use all the simple attributes of the composite for the primary key of R.

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



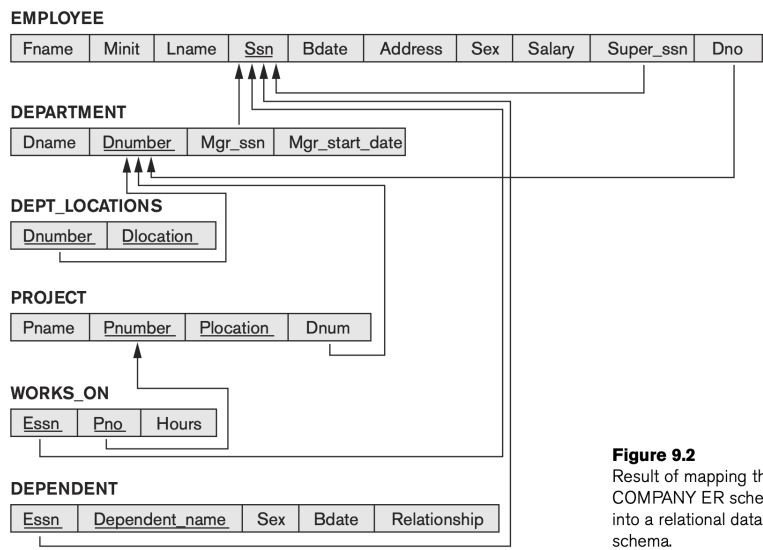


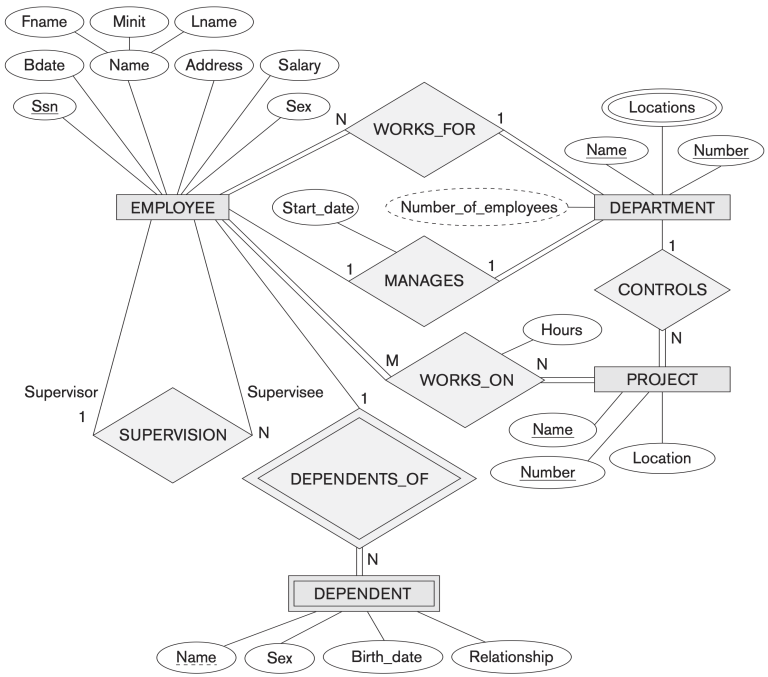
Figure 9.2
Result of mapping the COMPANY ER schema into a relational database schema.

Step 2: Mapping of weak entity types

For each weak entity type, W, in the ER schema with owner entity E

- Create a relation R
- Add the simple attributes of W to R
- Include as foreign key attributes of R, the primary key of the relations that correspond to the owner entity type
- The primary key of R is a combination of the primary key(s) of the owner(s) and the partial key of W, if any
- If there is a weak entity type E_1, whose owner is also a weak entity type, E_2, then E_1 should be mapped before E_2 to determine the primary keys first.

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



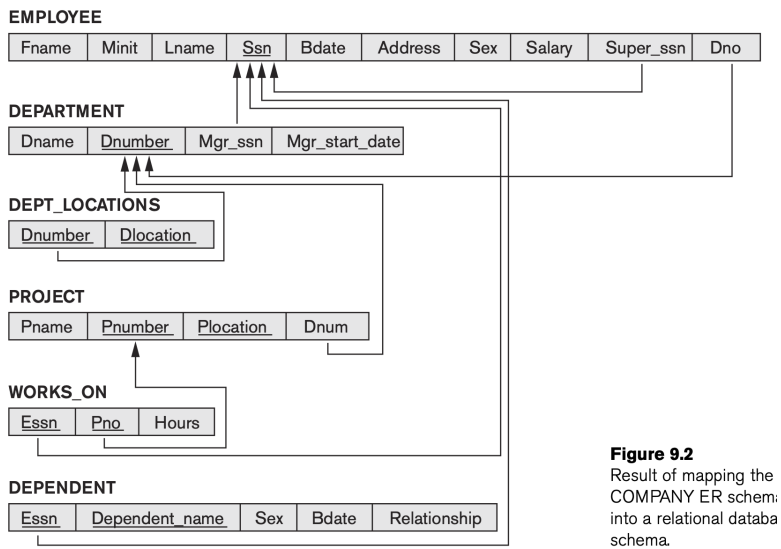


Figure 9.2
Result of mapping the COMPANY ER schema into a relational database schema.

Step 3: Mapping of binary 1:1 relationships

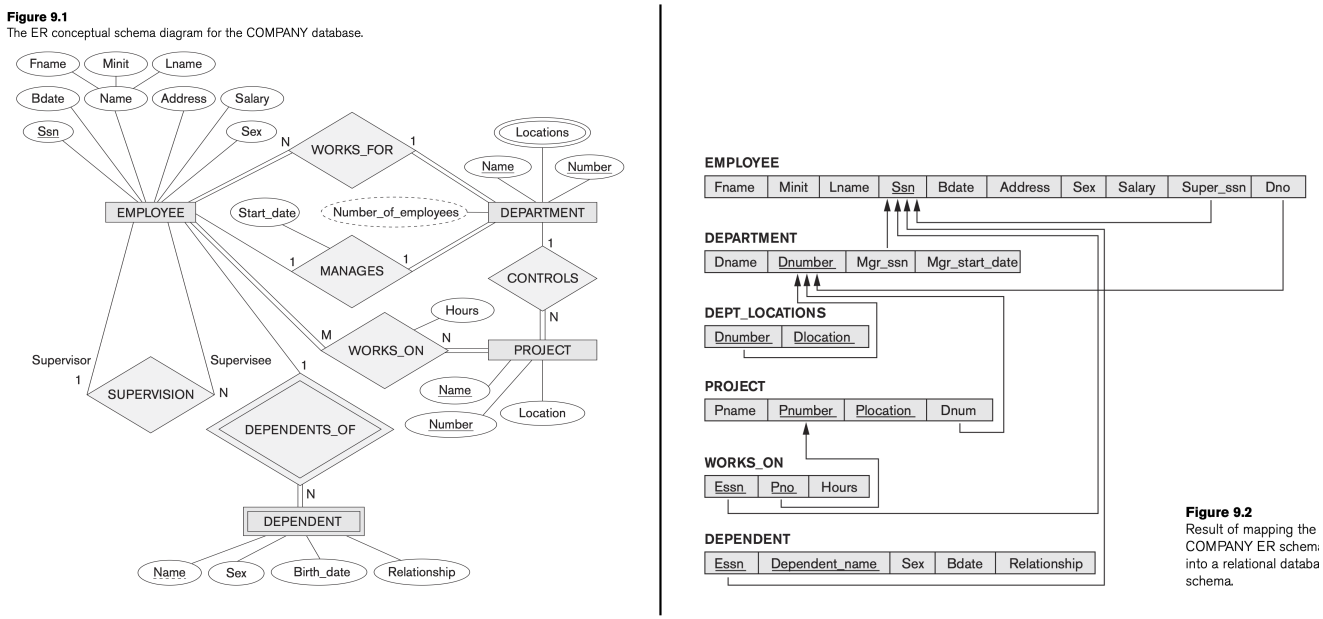
For each binary 1:1 relationship type R, identify the relations S and T that correspond to the entity types participating in R

1. *Foreign key approach* - most useful and followed unless special conditions exist
2. *Merged relation approach* - Works when both relations are total and can be merged. That is, when both relations have the same number of tuples at all times.
3. *Cross-reference or relationship relation approach* - Sets up a third relation, U, for the purpose of cross-referencing the primary keys of S and T. The relation U becomes a look-up table because it relates one tuple from S with one tuple from T.

Step 3: Mapping of binary 1:1 relationships (Approach 1)

For each binary 1:1 relationship type R, identify the relations S and T that correspond to the entity types participating in R, Choose one of the relations, say S - preferably the one with the total participation in R

- include as foreign key in S, the primary key of T
- include all the simple attributes (or simple components of the composite attributes) of the 1:1 relationship type R in S



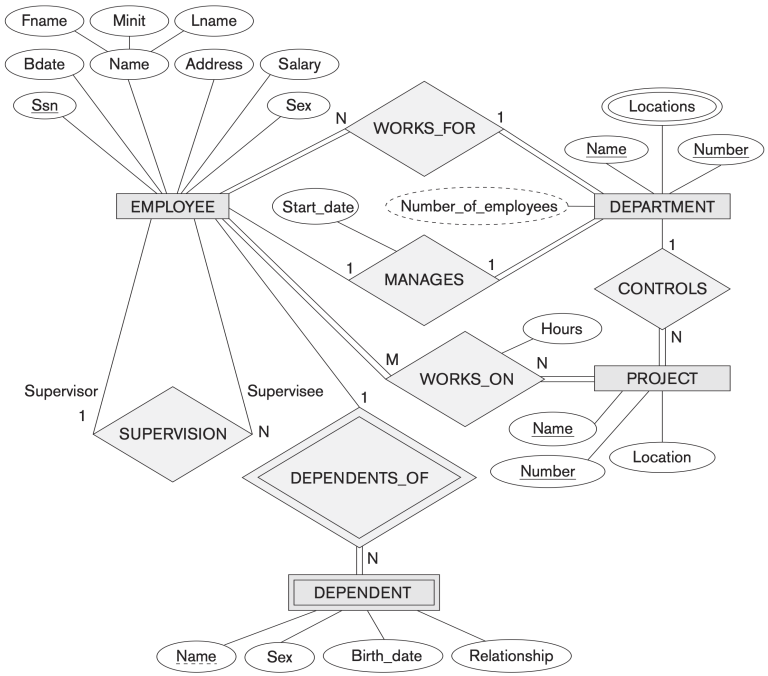
Choose DEPARTMENT as S because total in the MANAGES relation (every department has a manager). If EMPLOYEE was chosen as S, `Dept_managed` would have been included in EMPLOYEE. Where an employee is not a manager, a NULL value would need to have been included.

Step 4: Mapping of binary 1:N relationships

For each binary 1:N relationship type R, identify the relation S on the N side

- include in S the primary key of relation T (the other entity on the relationship).
- include any simple attributes (or simple components of composite attributes) of the 1:N relationship type as attributes of S.

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



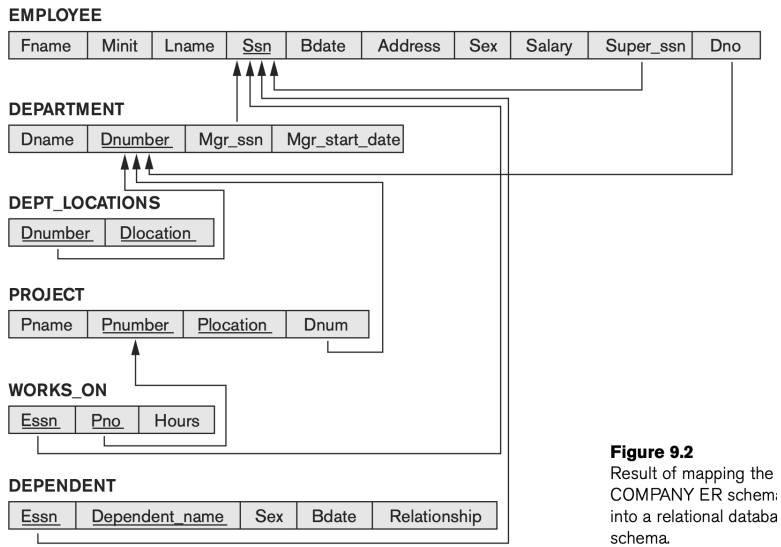


Figure 9.2
Result of mapping the COMPANY ER schema into a relational database schema.

Step 4: Mapping of binary 1:N relationships (Alternative approach)

An alternative approach is to use the relationship relation (cross-reference) approach, discussed as the third option of the 1:1 relationships. In this approach

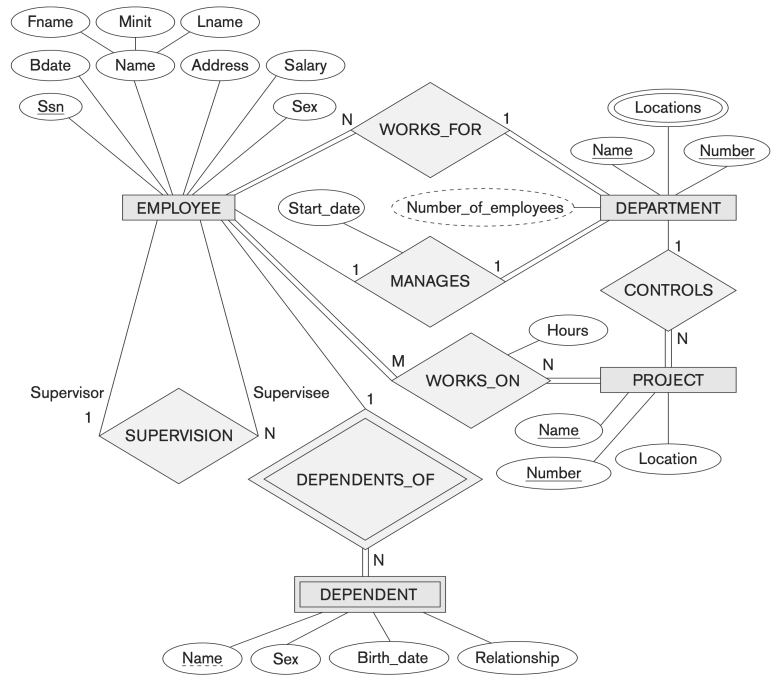
- Create a separate relation with attributes being the primary keys of the two relations involved in the relationship which will act as foreign keys for the respective relations.
- The primary key in this relation is the same as that of the first relation.
- This approach can lead to NULL values in the foreign keys.

Step 5: Mapping of binary M:N relationships

For each 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
- make the combination of these primary keys the primary key
- include the simple attributes (or the composite as simple components of the relationship

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



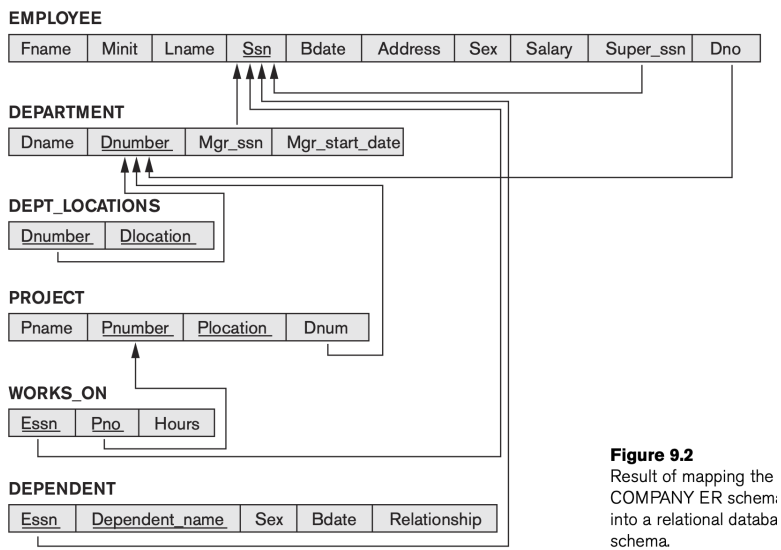


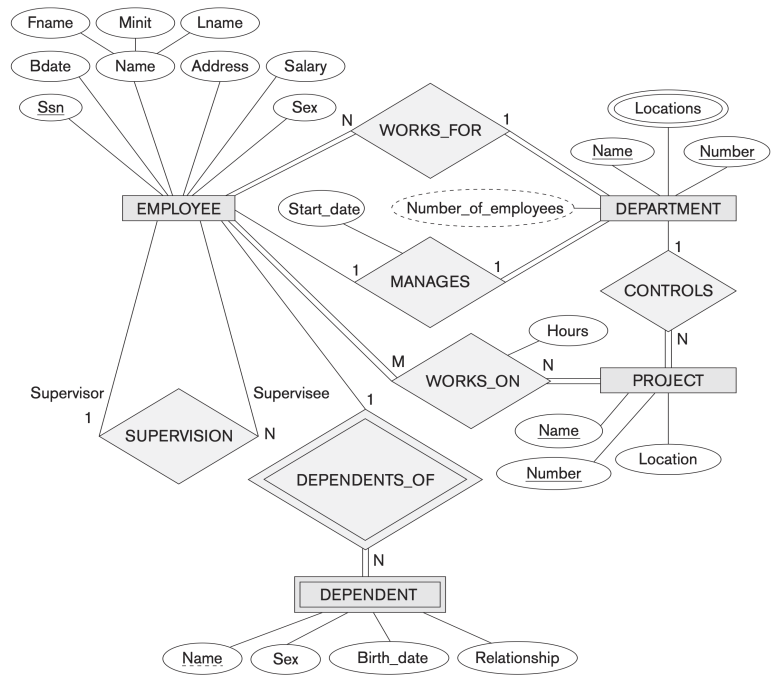
Figure 9.2
Result of mapping the COMPANY ER schema into a relational database schema.

Step 6: Mapping of multivalued attributes

For each multivalued attribute A, create a new relation R

- R will include an attribute corresponding to A and a primary key K as foreign key in R of the relation that represents the entity type that has A as a multivalued attribute
- the primary key R is a combination of A and K (if the multivalued attribute is composite, include its simple components).

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



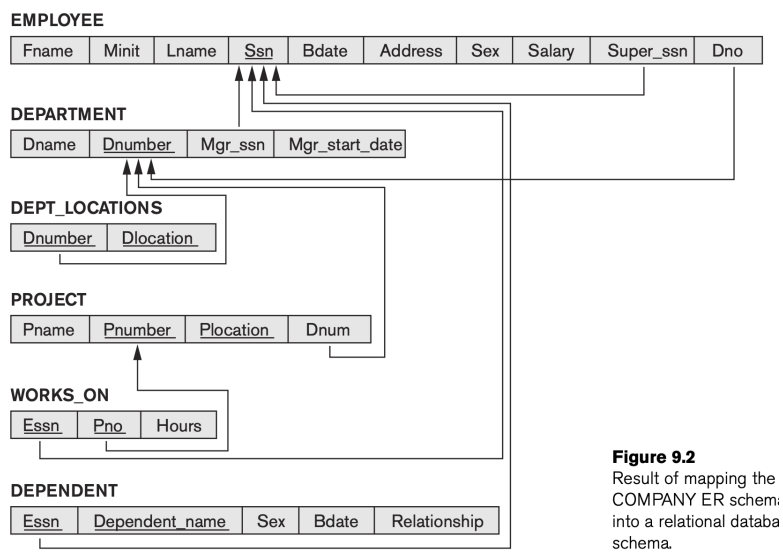


Figure 9.2
Result of mapping the COMPANY ER schema into a relational database schema.

Step 7: Mapping of N-ary relationships

For each N-ary relationship type R, where n ¿ 2, create a new relation S to represent R

- include as foreign attributes in S as primary key attributes that participate in the relationship
- include any simple attributes (or simple components of composites) as attributes of S
- the primary of S is a combination of all foreign keys that reference the relations participating in the relationship
- of the cardinality constraints on any of the entity types E participating in relation R is 1, then the primary key of S should not include the foreign key attribute that reference E' corresponding to E

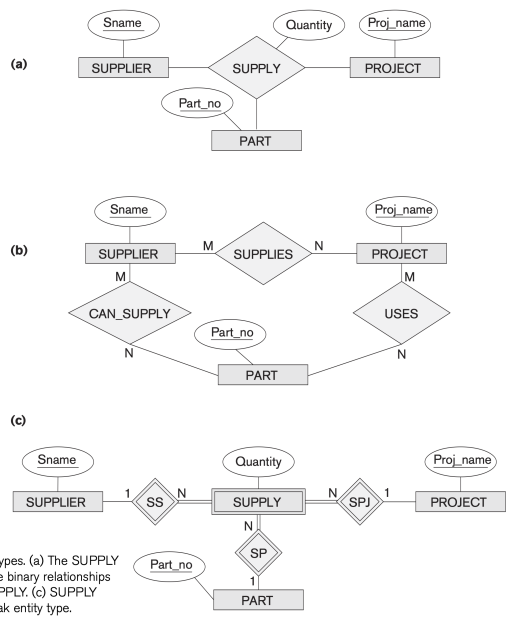
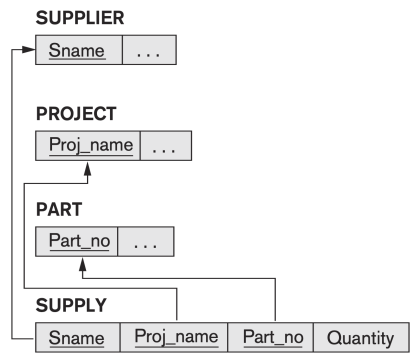


Figure 7.17
Ternary relationship types. (a) The SUPPLY relationship. (b) Three binary relationships not equivalent to SUPPLY. (c) SUPPLY represented as a weak entity type.

Figure 9.4
Mapping the *n*-ary relationship type SUPPLY from Figure 7.17(a).



Step 8: Mapping specialisation and generalisation

Convert each specialisation with m subclasses $\{S_1, S_2, \ldots, S_m\}$ and generalised superclass $C(k, a_1, \ldots, a_n)$, where k is the primary key into relation schemas using one of the following options:

- **Option 8A: Multiple relations - superclass and subclasses**
  - create a relation L for C with attributes of C and the primary key k
  - create a relation L_i for each subclass S_i with the attributes $\{k\} \cup \{\text{attributes of } S_i\}$ and k as primary key
  - works for total or partial, disjoint or overlapping specialisations
- **Option 8B: Multiple relations - subclass relations only**
  - create a relation L_i for each subclass S_i with attributes $\{\text{attributes of } S_i\} \cup \{k, a_1, \ldots, a_n\}$ with primary key of k
  - only works for specialisations where the subclasses are total, also recommended for specialisations with a disjointedness constraint
- **Option 8C: Single relation with one type attribute**
  - create a relation L with attributes $\{k, a_1, \ldots, a_n\} \cup \{\text{attributes of } S_i\} \cup \ldots. \{\text{attributes of } S_m\} \cup \{t\}$ and primary key of k
  - t is a type or discriminating attribute whose value indicates the subclass to which the tuple belongs
  - used for subclasses that are disjoint
  - can generate NULL values
- **Option 8D: Single relation with multiple type attributes**
  - create a relation L with attributes $\{k, a_1, \ldots, a_n\} \cup \{\text{attributes of } S_i\} \cup \ldots. \{\text{attributes of } S_m\} \cup \{t_1, t_2, \ldots, t_m\}$ and primary key of k, where t_i is a Boolean type attribute indicating whether the tuple belongs to subclass S_i
  - used for specialisations whose subclasses are overlapping, also works for disjoint

Step 8: Mapping specialisation and generalisation

Convert each specialisation with m subclasses $\{S_1, S_2, \ldots, S_m\}$ and generalised superclass $C(k, a_1, \ldots, a_n)$, where k is the primary key into relation schemas using one of the following options:

- **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**

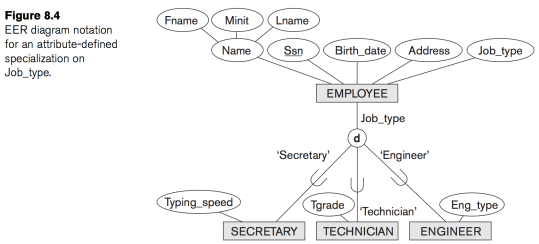


Figure 8.4
EER diagram notation for an attribute-defined specialization on Job_type.

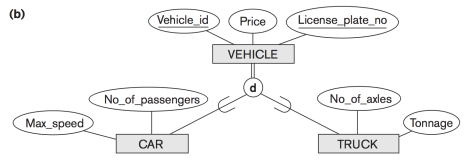


Figure 8.3
Generalization. (a) Two entity types, CAR and TRUCK. (b) Generalizing CAR and TRUCK into the superclass VEHICLE.

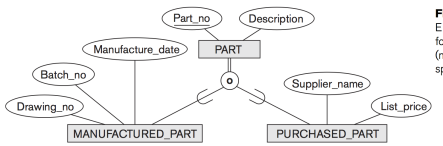


Figure 8.5
EER diagram notation for an overlapping (nondisjoint) specialization.

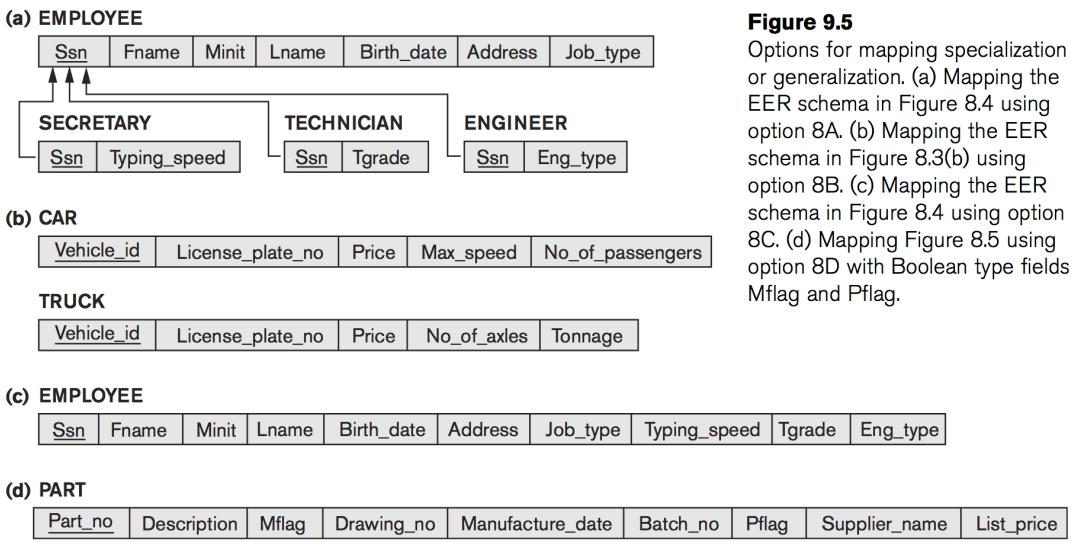


Figure 9.5
Options for mapping specialization or generalization. (a) Mapping the EER schema in Figure 8.4 using option 8A. (b) Mapping the EER schema in Figure 8.3(b) using option 8B. (c) Mapping the EER schema in Figure 8.4 using option 8C. (d) Mapping Figure 8.5 using option 8D with Boolean type fields Mflag and Pflag.

Step 8: Mapping specialisation and generalisation - Lattices

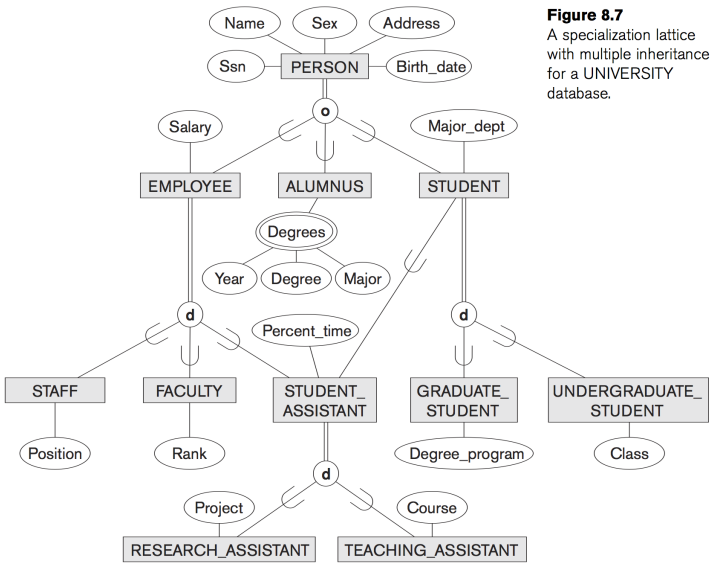


Figure 8.7
A specialization lattice with multiple inheritance for a UNIVERSITY database.

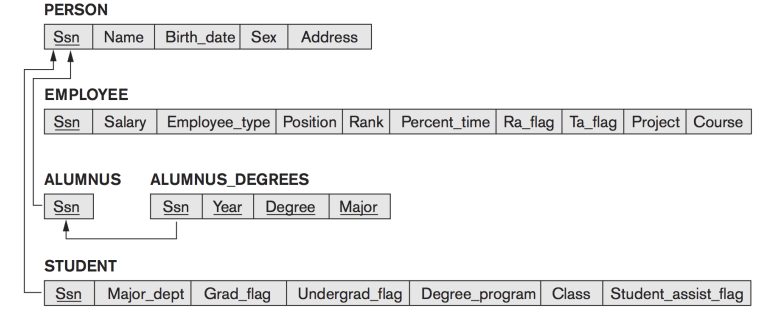


Figure 9.6
Mapping the EER specialization lattice in Figure 8.8 using multiple options.

Step 9: Mapping unions

When mapping classes whose superclasses have different keys, a surrogate key is defined when creating a relation which corresponds to the category. This key is the foreign key in the other relations participating in the relationship.

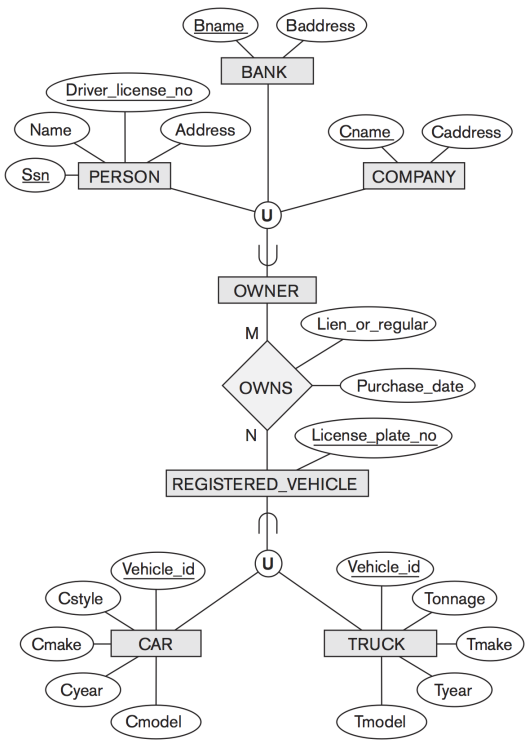
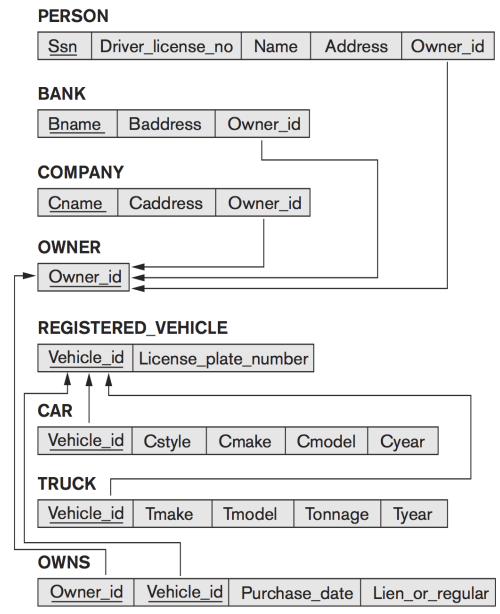


Figure 8.8
Two categories (union types): OWNER and REGISTERED_VEHICLE.

Figure 9.7
Mapping the EER categories (union types) in Figure 8.8 to relations.



`Owner_id` is the surrogate key.