UNIT-V EER and ER to relational mapping:

UNIT-V

EER and ER to relational mapping:

Data base design using EER to relational language.

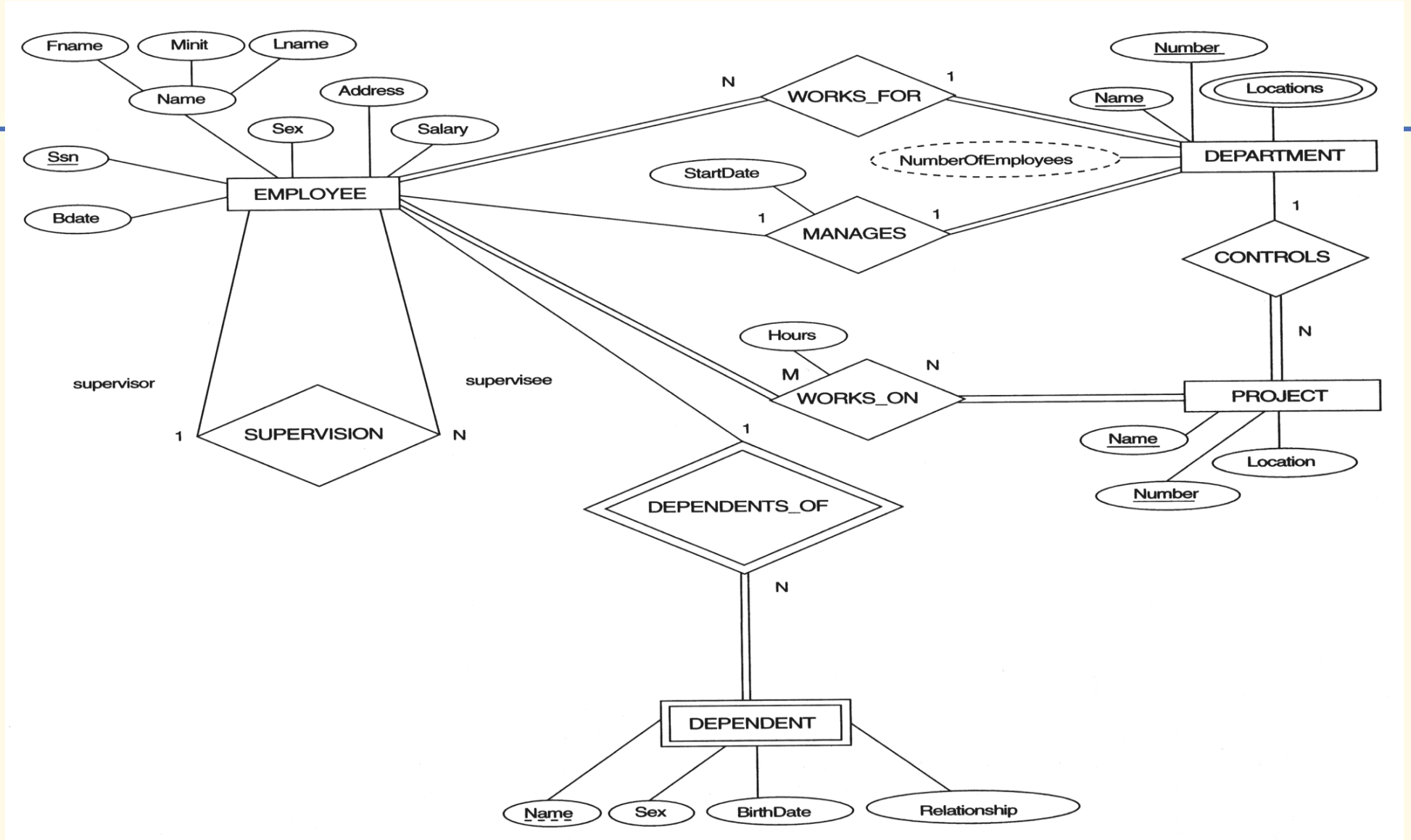
UNIT-V EER and ER to relational mapping:

- **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.

ER-to-Relational Mapping Algorithm

Step 1: Mapping of Regular Entity Types
For every **strong (regular) entity**, create a **table**.
•Attributes become columns.
•**Primary key** of the entity becomes the **primary key** of the table.
Example:
Entity: Student (RollNo, Name, Class)
→ Relation: **Student(RollNo, Name, Class)**
PK = RollNo



The ER conceptual schema diagram for the COMPANY database.

UNIT-V EER and ER to relational mapping:

EMPLOYEE
Stores employee details
Primary Key → **Ssn (Social Security Number)**
Foreign Keys → **Super_ssn** (supervisor), **Dno** (department)

DEPARTMENT
Stores department details
Primary Key → **Dnumber**
Foreign Key → **Mgr_ssn** (manager)

DEPT_LOCATIONS
Stores all locations of each department
Foreign Key → **Dnumber**
(Department can have many locations)

PROJECT
Stores project details
Primary Key → **Pnumber**
Foreign Key → **Dnum** (department controlling project)

WORKS_ON
Links employees to projects
Foreign Keys → **Essn** (employee), **Pno** (project)
Shows how many **hours** employee works on each project

DEPENDENT
Stores family members of employees
Foreign Key → **Essn**
Dependent exists only for an employee

Step 2: Mapping of Weak Entity Types

For every **weak entity**, create a **separate table**.

- Include the **partial key** and the **primary key of its owner entity**.
- **Primary key = (Owner PK + Partial key)**.

Example:

Weak Entity: Dependent (Name, Age), depends on Employee(EID)

→ Relation: **Dependent(EID, Name, Age)**

PK = {EID + Name}

EID is also a **foreign key (FK)**.

Step 3: Mapping of Binary 1:1 Relationship Types
Two strong entities with **1:1**
→ Add the **primary key of one entity** as a **foreign key** in the other.
(Prefer side with **total participation**)

Example:
Employee(EMPID) ↔ Locker(LID)
If Locker has total participation → add EMPID to Locker.

Step 4: Mapping of Binary 1:N Relationship Types
For **1:N relationship**, add a **foreign key of the "1-side" entity into the "N-side" table**.

Example:
Department(DeptID) — 1:N — Employee(EID)
→ Employee table gets DeptID as **FK**.

Step 5: Mapping of Binary M:N Relationship Types
For **M:N relationship**, create a **new table**.
•Include **PKs of both entities** (as foreign keys) and any relationship attributes.
•**Joint PK = both foreign keys**.

Example:
Student(SID) — M:N — Course(CID)
→ new relation: **Enroll(SID, CID, Grade)**
PK = (SID, CID)

Step 6: Mapping of Multivalued Attributes

For every **multivalued attribute**, create a **separate table**.

- Include **PK of main entity** + **multivalued attribute**.
- PK = **combination**.

Example:

Employee(EID, PhoneNumber multivalued)

→ **EmployeePhone(EID, PhoneNumber)**

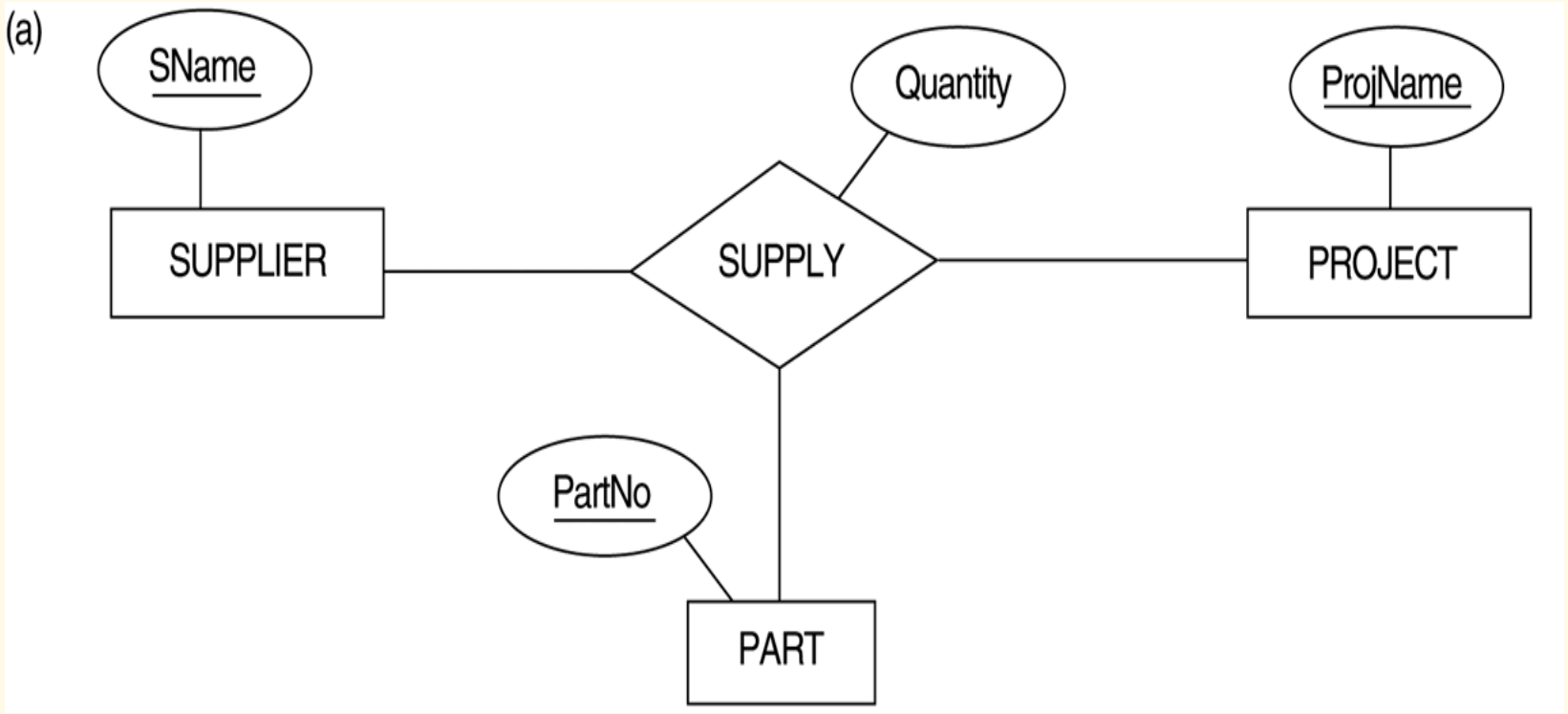
PK = (EID, PhoneNumber)

Step 7: Mapping of N-ary Relationship Types (N > 2)
For a relationship involving **3 or more entities**, create a **new table**.
•Include **PKs of all participating entities**.
•Add relationship attributes.
•**Primary key = combination of all FKs**.

Example:
Supplier — Parts — Project
→ **Supply(S ID, P ID, PRJ ID, Quantity)**
PK = (SID, PID, PRJID)

UNIT-V EER and ER to relational mapping:

Ternary relationship types. (a) The SUPPLY relationship.



Step 8: Mapping Specialization / Generalization

Approach	Mapping Rule
A) Multiple tables (Top-Down)	Create **superclass table** + **one table per subclass**, subclass table contains **PK of superclass as FK + subclass attributes**
B) Single table (Flattening)	Create **one table** combining all **superclass and subclass attributes**, add **type/discriminator attribute**
C) Only subclass tables	No table for superclass, each subclass table contains **all superclass attributes**

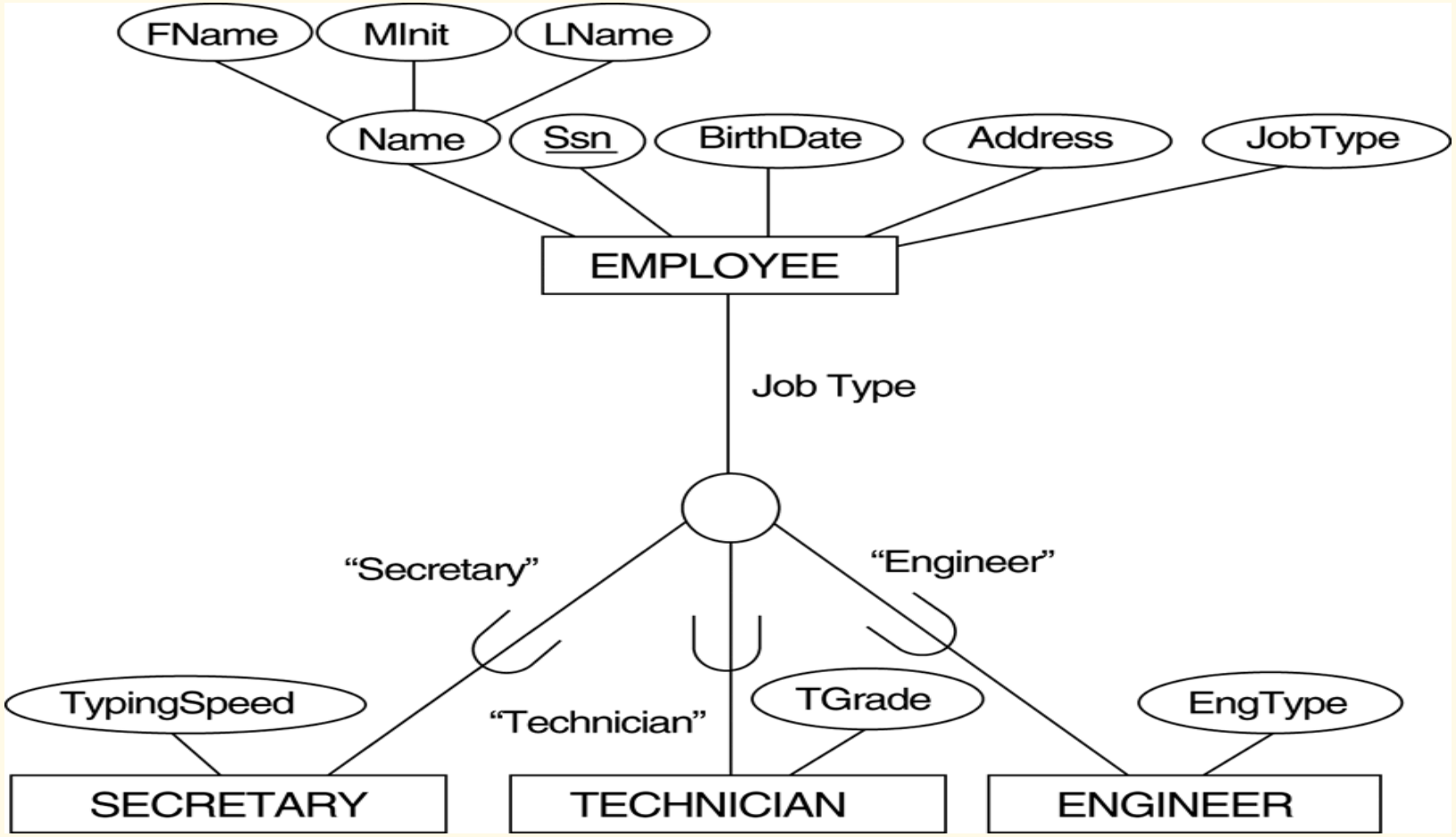


FIGURE 4.4
EER diagram notation for an attribute-defined specialization on JobType.

Enhanced **E**ntity–**R**elationship

Options for mapping specialization or generalization.
(a) Mapping the EER schema in Figure 4.4 using option 8A.

(a)

EMPLOYEE

SSN	FName	MInit	LName	BirthDate	Address	JobType

SECRETARY

SSN	TypingSpeed

TECHNICIAN

SSN	TGrade

ENGINEER

SSN	EngType

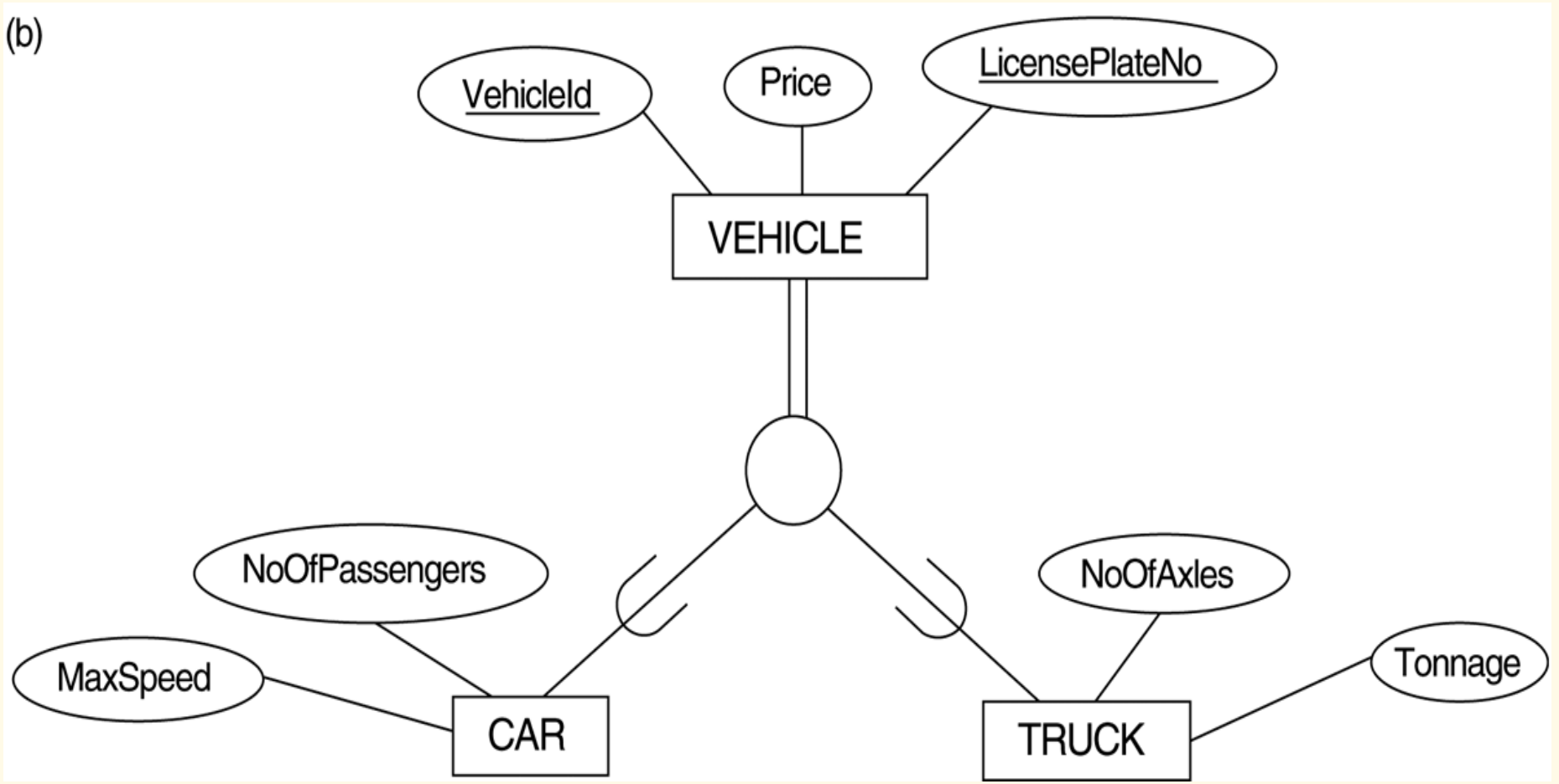


FIGURE 4.3
Generalization. (b) Generalizing CAR and TRUCK into the superclass VEHICLE.

FIGURE 7.4

Options for mapping specialization or generalization.
(b) Mapping the EER schema in Figure 4.3b using option 8B.

UNIT-V EER and ER to relational mapping:

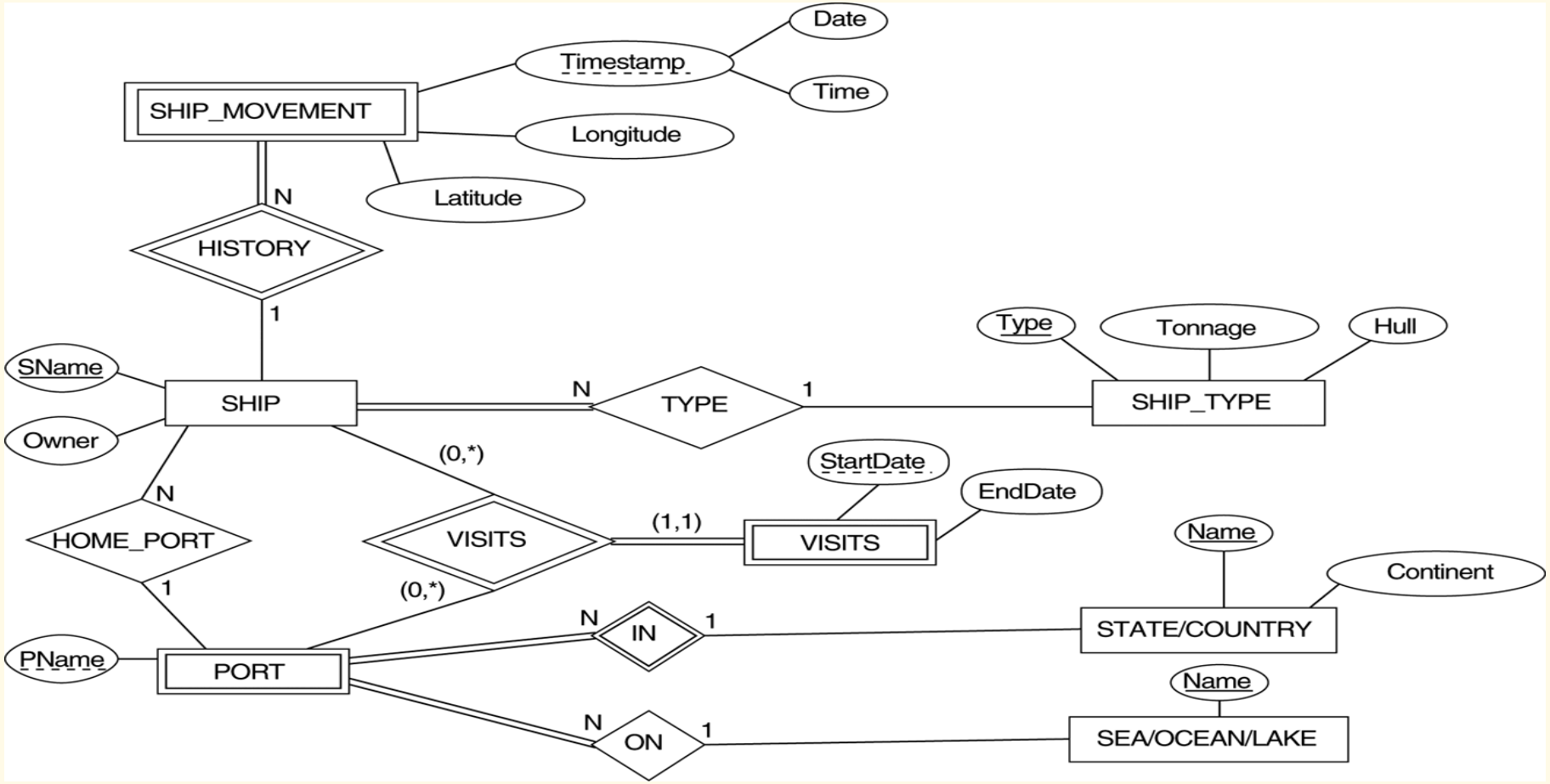


FIGURE 7.7
An ER schema for a SHIP_TRACKING database.