

Relational Database Design by ER- and EER-to- Relational Mapping

Design a relational database schema

- تصميم مخطط قاعدة بيانات علائقية

- Based on a conceptual schema design

- بناء على تصميم مخطط مفاهيمي

- Seven-step algorithm to convert the basic ER model constructs into relations

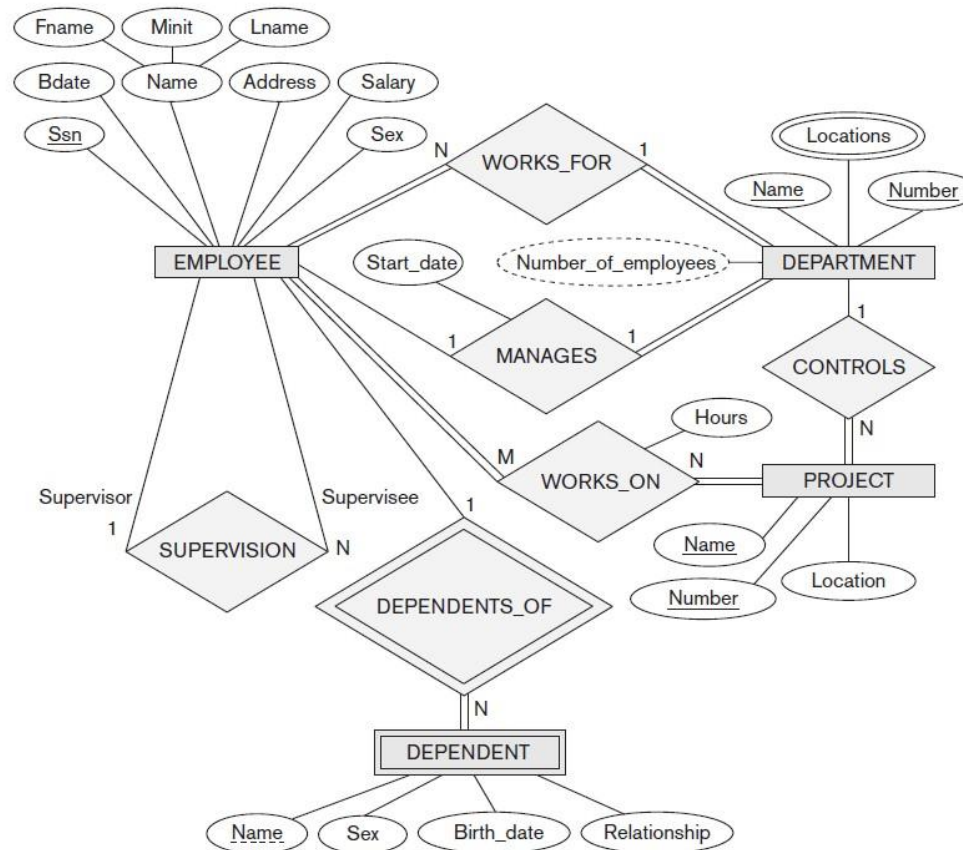
- خوارزمية من سبع خطوات لتحويل تركيبات نموذج ER الأساسي إلى علاقات

- Additional steps for EER model

Relational Database Design Using ER-to-Relational Mapping

Figure 9.1

The ER conceptual schema diagram for the COMPANY database.



EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

DEPARTMENT

Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
-------	----------------	---------	----------------

DEPT_LOCATIONS

<u>Dnumber</u>	<u>Dlocation</u>
----------------	------------------

PROJECT

Pname	<u>Pnumber</u>	<u>Plocation</u>	Dnum
-------	----------------	------------------	------

WORKS_ON

<u>Essn</u>	<u>Pno</u>	Hours
-------------	------------	-------

DEPENDENT

<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------

Figure 9.2

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

ER-to-Relational Mapping Algorithm

COMPANY database example ■

Assume that the mapping will create tables ■
with simple single-valued attributes

Step 1: Mapping of Regular Entity Types ■

For each regular entity type, create a relation R ■
that includes all the simple attributes of E

Called **entity relations** ■

Each tuple represents an entity instance •

ER-to-Relational Mapping Algorithm (cont'd.)

Step 2: Mapping of Weak Entity Types ■

For each weak entity type, create a relation R ■
and include all simple attributes of the entity
type as attributes of E

Include primary key attribute of owner as ■
foreign key attributes of R

ER-to-Relational Mapping Algorithm (cont'd.)

Figure 9.3

Illustration of some mapping steps.

- a. *Entity* relations after step 1.
- b. Additional *weak entity* relation after step 2.
- c. *Relationship* relation after step 5.
- d. Relation representing multivalued attribute after step 6.

(a) **EMPLOYEE**

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary
-------	-------	-------	------------	-------	---------	-----	--------

DEPARTMENT

Dname	<u>Dnumber</u>
-------	----------------

PROJECT

Pname	<u>Pnumber</u>	Plocation
-------	----------------	-----------

(b) **DEPENDENT**

<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------

(c) **WORKS_ON**

<u>Essn</u>	<u>Pno</u>	Hours
-------------	------------	-------

(d) **DEPT_LOCATIONS**

<u>Dnumber</u>	<u>Dlocation</u>
----------------	------------------

ER-to-Relational Mapping Algorithm (cont'd.)

Step 3: Mapping of Binary 1:1 Relationship Types

For each binary 1:1 relationship type

Identify relations that correspond to entity types
participating in R

Possible approaches:

Foreign key approach

Merged relationship approach

Crossreference or relationship relation approach

ER-to-Relational Mapping Algorithm (cont'd.)

Step 4: Mapping of Binary 1: N Relationship Types

- For each regular binary 1: N relationship type
- Identify relation that represents participating entity type at N -side of relationship type
- Include primary key of other entity type as foreign key in S
- Include simple attributes of 1: N relationship type as attributes of S

ER-to-Relational Mapping Algorithm (cont'd.)

Alternative approach ■

- Use the **relationship relation** (cross-reference) option as in the third option for binary 1:1 relationships

ER-to-Relational Mapping Algorithm (cont'd.)

Step 5: Mapping of Binary $M:N$ Relationship Types

For each binary $M:N$ relationship type

- Create a new relation S
- Include primary key of participating entity types as foreign key attributes in S
- Include any simple attributes of $M:N$ relationship type

ER-to-Relational Mapping Algorithm (cont'd.)

- Step 6: Mapping of Multivalued Attributes
 - For each multivalued attribute
 - Create a new relation
 - Primary key of R is the combination of A and K
 - If the multivalued attribute is composite, include its simple components

ER-to-Relational Mapping Algorithm (cont'd.)

Step 7: Mapping of N -ary Relationship Types

For each n -ary relationship type R ■

Create a new relation S to represent R •

Include primary keys of participating entity types as
foreign keys •

Include any simple attributes as attributes •

Discussion and Summary of Mapping for ER Model Constructs

Table 9.1 Correspondence between ER and Relational Models

ER MODEL	RELATIONAL MODEL
Entity type	<i>Entity</i> relation
1:1 or 1:N relationship type	Foreign key (or <i>relationship</i> relation)
M:N relationship type	<i>Relationship</i> relation and <i>two</i> foreign keys
<i>n</i> -ary relationship type	<i>Relationship</i> relation and <i>n</i> 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

Discussion and Summary of Mapping for ER Model Constructs (cont'd.)

In a relational schema relationship, types are not represented explicitly ■

Represented by having two attributes A and B : ■
one a primary key and the other a foreign key

Mapping EER Model Constructs to Relations

Extending ER-to-relational mapping algorithm ■

Mapping of Specialization or Generalization

Step 8: Options for Mapping Specialization or Generalization (see pages 294-295) ■

Option 8A: Multiple relations—superclass and subclasses ■

For any specialization (total or partial, disjoint or overlapping) •

Option 8B: Multiple relations—subclass relations only ■

Subclasses are total •

Specialization has disjointedness constraint •

Mapping of Specialization or Generalization (cont'd.)

Option 8C: Single relation with one type attribute

Type or discriminating attribute indicates subclass of •
tuple

Subclasses are disjoint •

Potential for generating many NULL values if many specific •
attributes exist in the subclasses

Option 8D: Single relation with multiple type attributes

Subclasses are overlapping •

Will also work for a disjoint specialization •

Mapping of Shared Subclasses (Multiple Inheritance)

Apply any of the options discussed in step 8 to a shared subclass

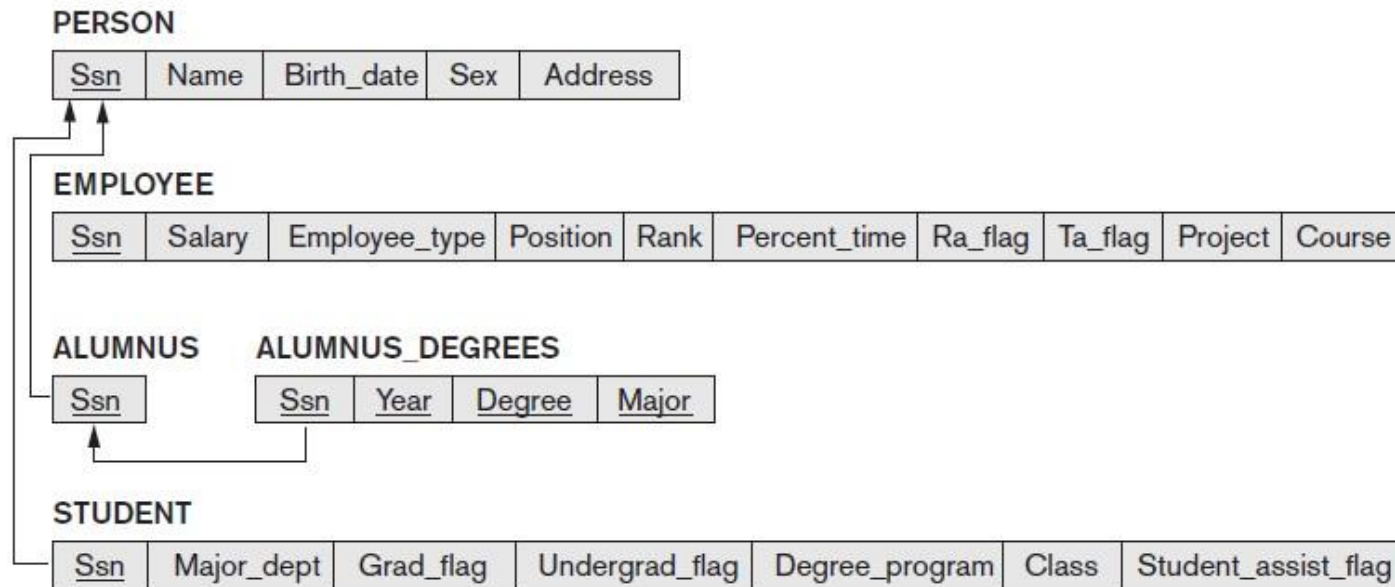


Figure 9.6

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

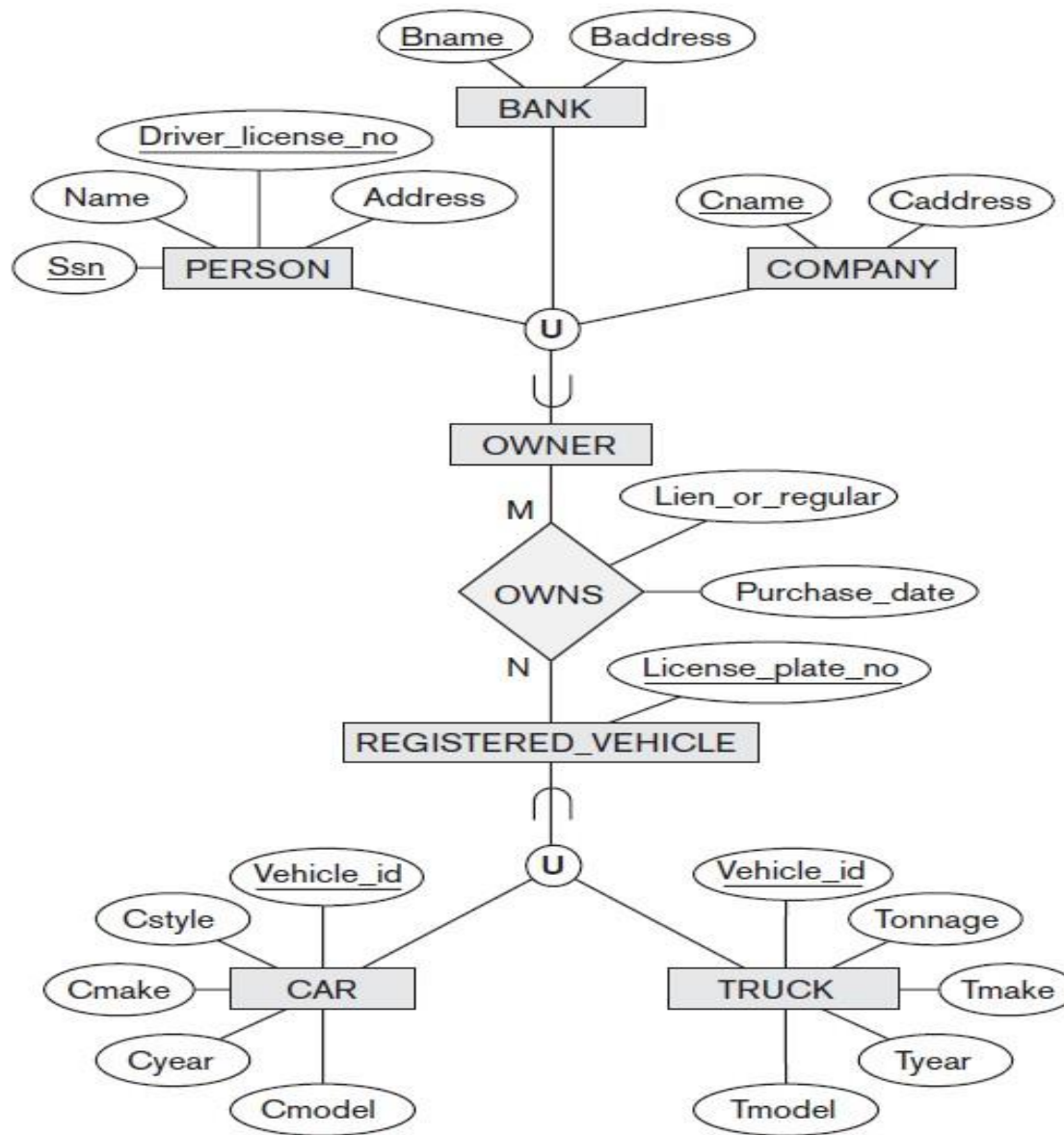
Mapping of Categories (Union Types)

Step 9: Mapping of Union Types (Categories) ■

Defining superclasses have different keys ■

Specify a new key attribute ■

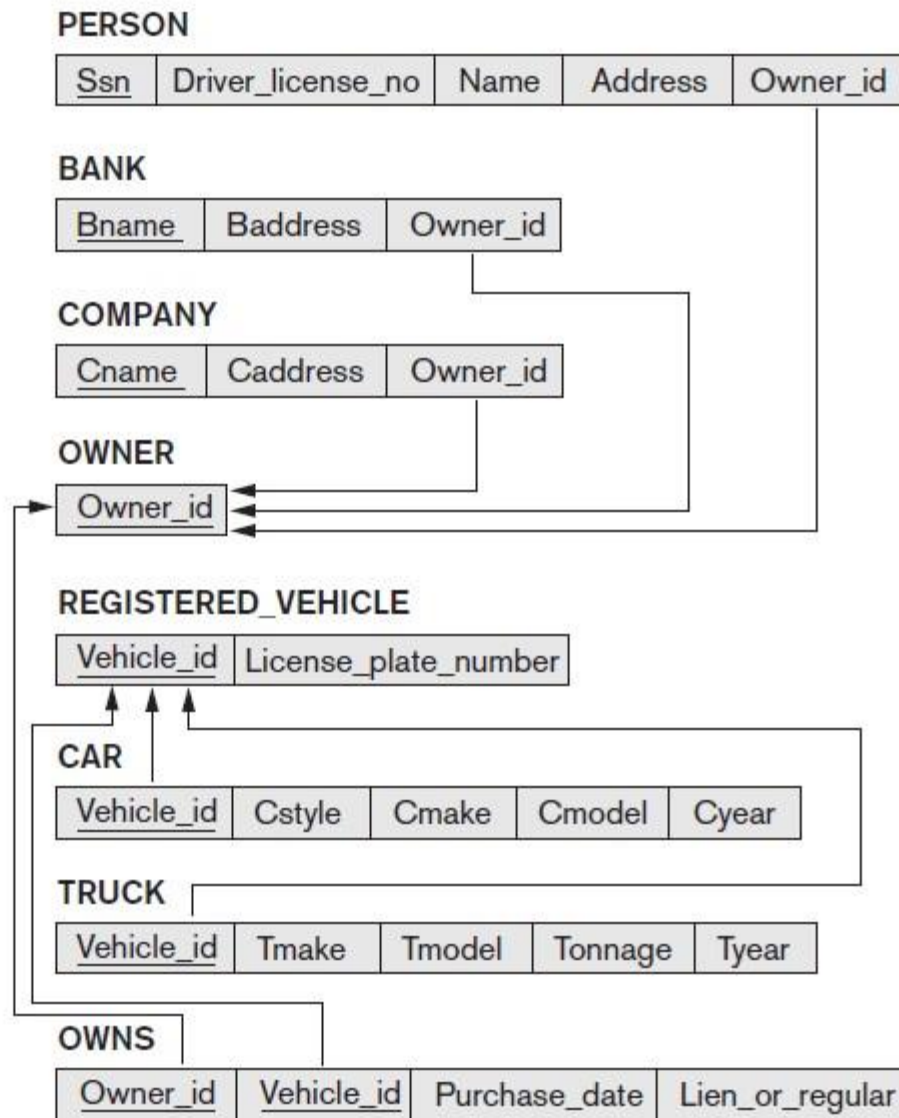
Surrogate key •

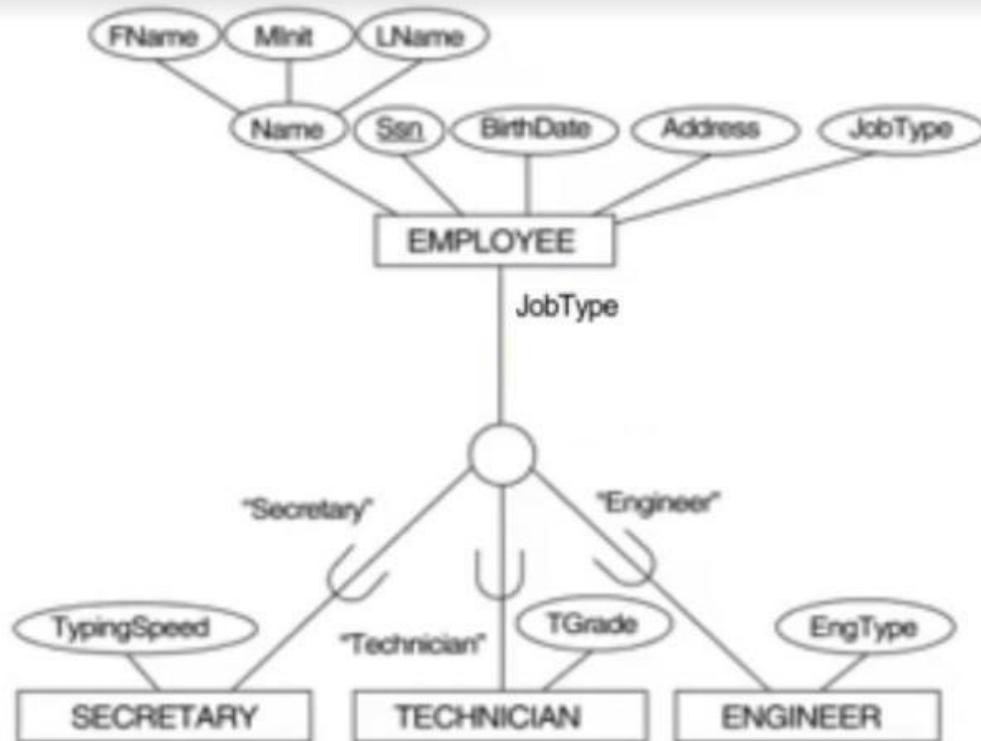
**Figure 8.8**

Two categories (union types): OWNER and REGISTERED_VEHICLE.

Figure 9.7

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





(a)

EMPLOYEE

<u>SSN</u>	FName	Minit	LName	BirthDate	Address	JobType
------------	-------	-------	-------	-----------	---------	---------

SECRETARY

<u>SSN</u>	TypingSpeed
------------	-------------

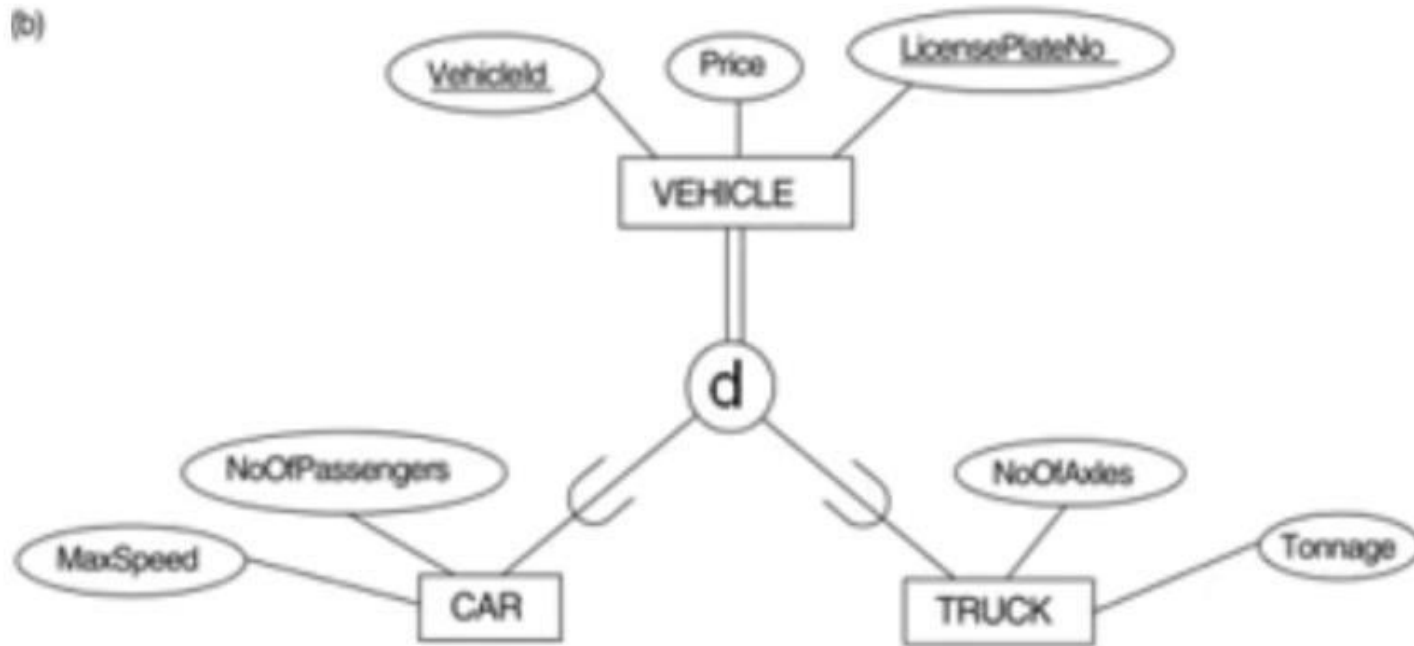
TECHNICIAN

<u>SSN</u>	TGrade
------------	--------

ENGINEER

<u>SSN</u>	EngType
------------	---------

(b)



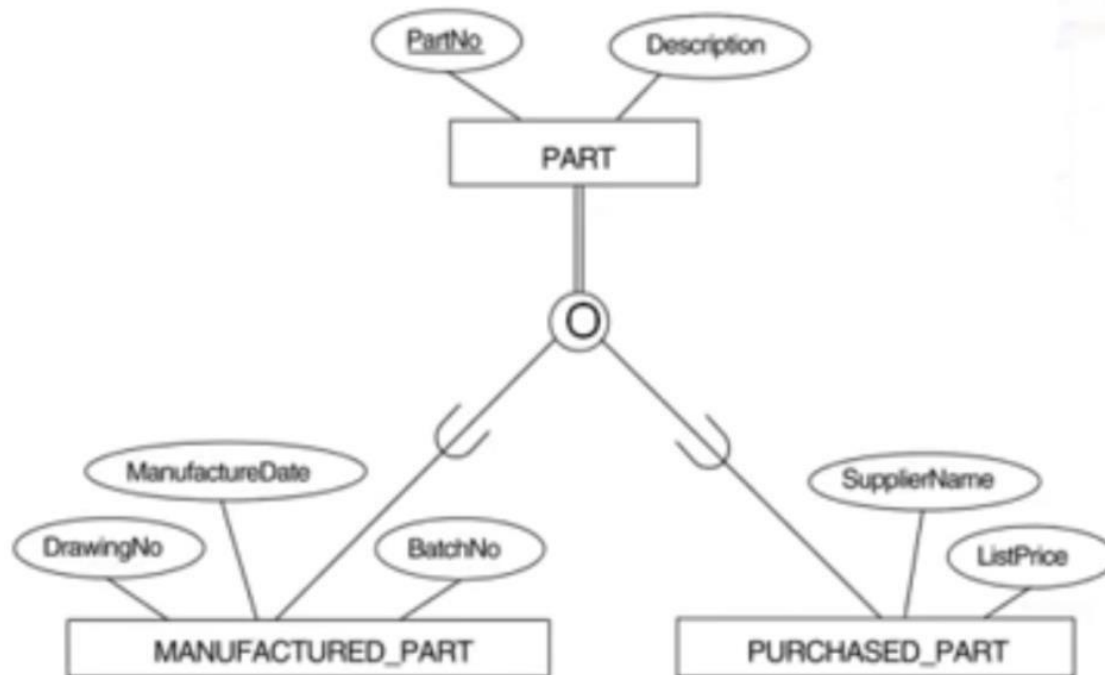
(b)

CAR

<u>VehicleId</u>	LicensePlateNo	Price	MaxSpeed	NoOfPassengers
------------------	----------------	-------	----------	----------------

TRUCK

<u>VehicleId</u>	LicensePlateNo	Price	NoOfAxes	Tonnage
------------------	----------------	-------	----------	---------



(d)

PART								
<u>PartNo</u>	Description	MFlag	DrawingNo	ManufactureDate	BatchNo	PFlag	SupplierName	ListPrice