Database Principles

CSC 3400

Relational Database

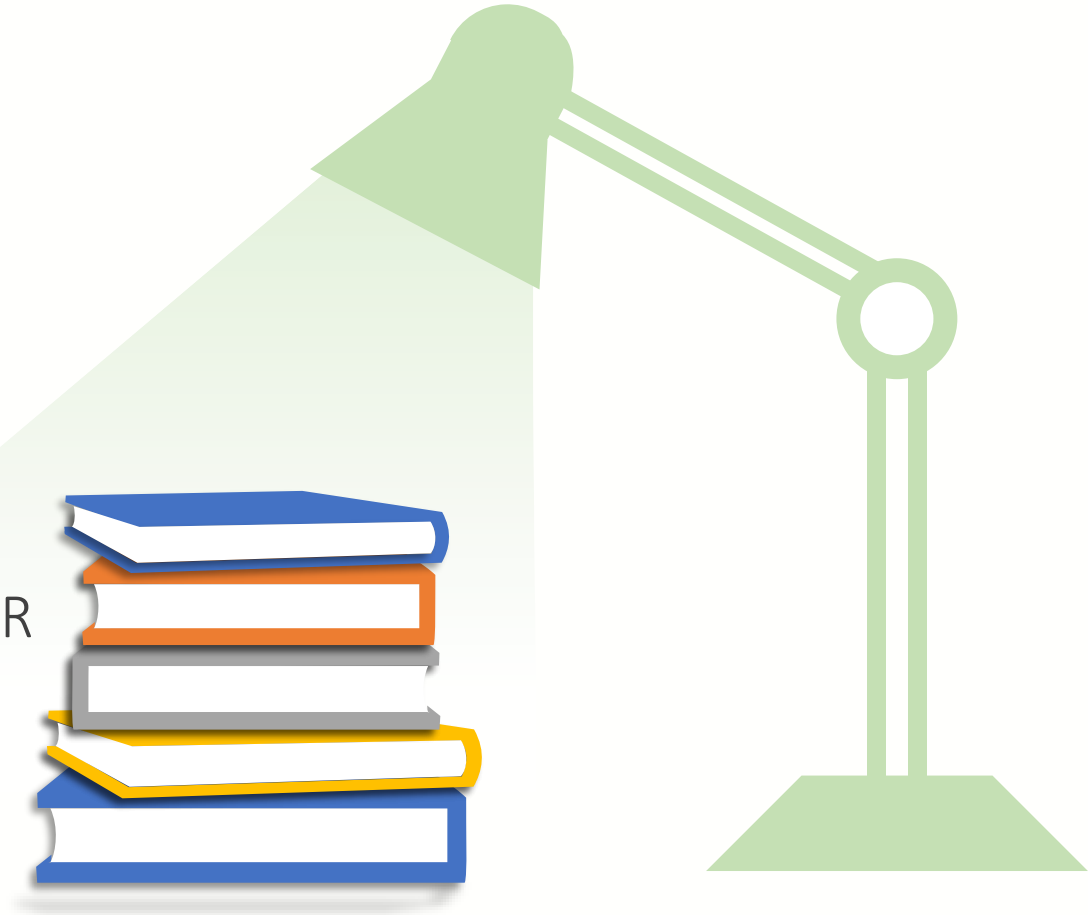
Learning Objectives

01 The concept of a relational model

02 Constraints of relational models and database schemas

03 Design a relational database using ER and EER

04 Mapping to a relational database



Relation

- **Definition**: A relation is a named, two-dimensional table of data
  - Table is made up of rows (records), and columns (attribute or field)
- Not all tables qualify as relations
- Requirements:
  - Every relation has a unique name.
  - Every attribute value is atomic (not multivalued, not composite)
  - Every row is unique (can't have two rows with exactly the same values for all their fields)
  - Attributes (columns) in a table have unique names
  - The order of the columns is irrelevant
  - The order of the rows is irrelevant

What is atomic value?

- This mean that in your table, for every row-by-column position (cell), there exists only one value - not an array or list of values:

OrderID	CustID	Date	Items
1	4	**4/11/02**	5 Pencils, 3 Erasers, 6 Rulers
2	23	**6/11/02**	1 Scissor
3	15	**7/11/02**	2 Pen, 2 Notebook
4	2	**7/11/02**	15 5" Magazine File
5	23	**7/11/02**	1 Stapler
6	2	**8/11/02**	5 Kingston USB Flash Drive 8GB

NOT a Relation, since Items cell has many values

Is this a relation?

Still NOT a relation, since Items cell multi-value

OrderID	CustID	Date	Quantity	Items
1	4	4/11/02	5	**Pencils, Erasers, Rulers**
2	23	6/11/02	1	Scissor
3	15	7/11/02	2	**Pen, Notebook**
4	2	7/11/02	15	5" Magazine File
5	23	7/11/02	1	Stapler
6	2	8/11/02	5	Kingston USB Flash Drive 8GB

Is this a relation?

A Relation, since Items cell has single value, and quantity is separated as a new column

OrderID	CustID	Date	Quantity	Items
1	4	4/11/02	5	Pencils
1	4	4/11/02	3	Erasers
1	4	4/11/02	6	Rulers
2	23	6/11/02	1	Scissor
3	15	7/11/02	2	Pen
3	15	7/11/02	2	Notebook
4	2	7/11/02	15	5" Magazine File
5	23	7/11/02	1	Stapler
6	2	8/11/02	5	Kingston USB Flash Drive 8GB

Example

Not a Relation, since EMP_No and EMP_Name cells have many values

DEPT_NO	MANAGER_NO	EMP_NO	EMP_NAME
D101	12345	2000 2001 2002	Carl Sagan Magic Johnson Larry Bird
D102	13456	3000 3001	Jimmy Carter Paul Simon

…and now, is this a relation?

Dept_No	Manager_No	Emp_No	Emp_Name
D101	12345	2000	Carl Sagan
D101	12345	2001	Magic Johnson
D101	12345	2002	Larry Bird
D102	13456	3000	Jimmy Carter
D102	13456	3001	Paul Simon

A Relation, since EMP_No and EMP_Name cells have single values

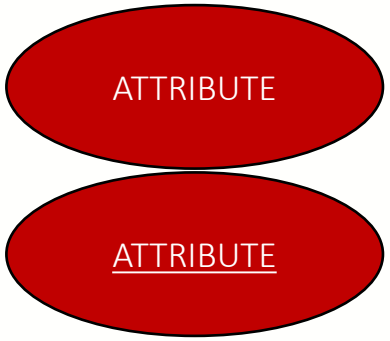
Key

- Keys are special fields that serve two main purposes:
  - **Primary keys** (PK)are unique identifiers of the relation in question. Examples include employee numbers, social security numbers, etc. *This is how we can guarantee that all rows are unique*
  - **Foreign keys** (FK)are identifiers that enable a dependent relation (on the many side of a relationship) to refer to its parent relation (on the one side of the relationship)
- Keys can be **simple** (a single field) or **composite** (more than one field)
- Keys usually are used as indexes to speed up the response to user queries.

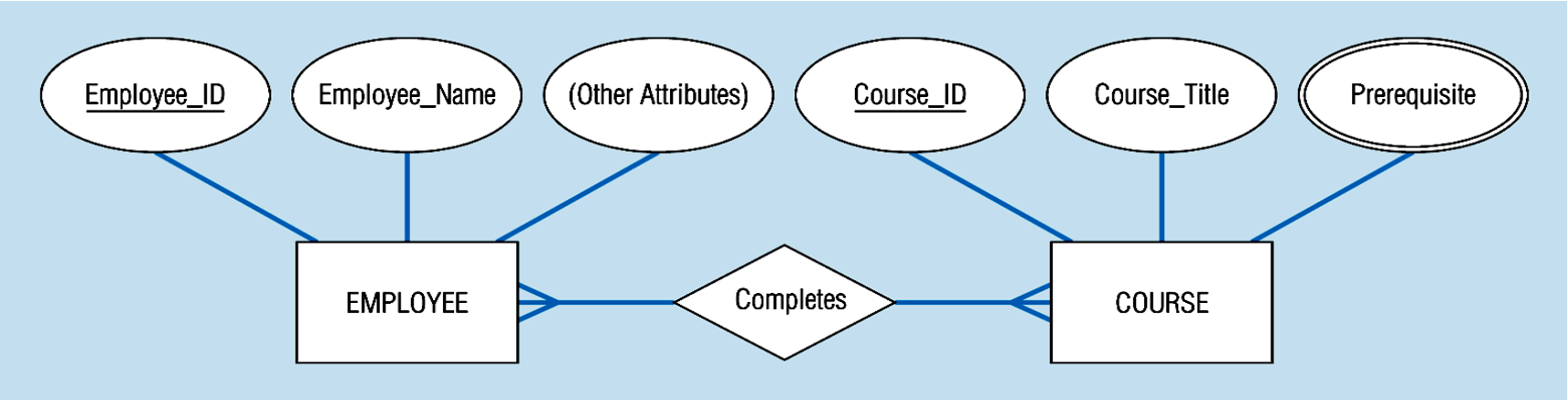
Correspondence with ER Model

- Relations (tables) correspond with **entity types**, multivalued attribute and with many-to-many relationship types

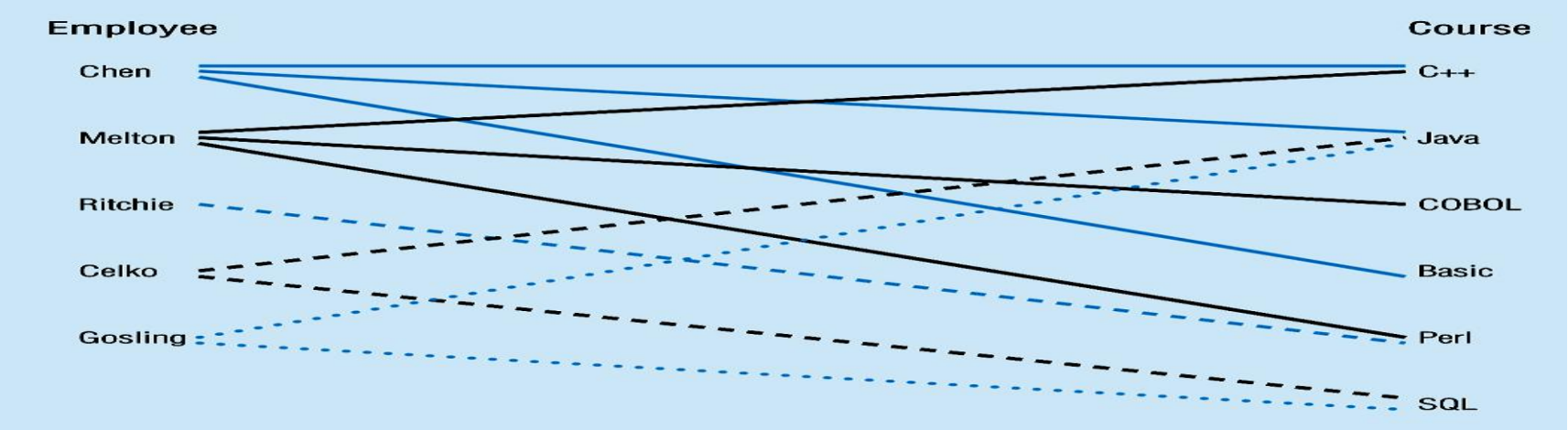
- Rows correspond with entity instances and with many-to-many relationship instances
- Columns correspond with attributes
- Primary key are underlined attribute(s)



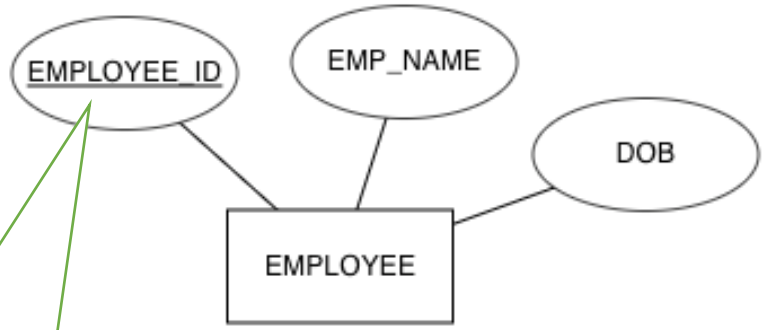
EXAMPLE



Entity and Relationship instances



ENTITY TYPE → EMPLOYEE



Primary key is underlined

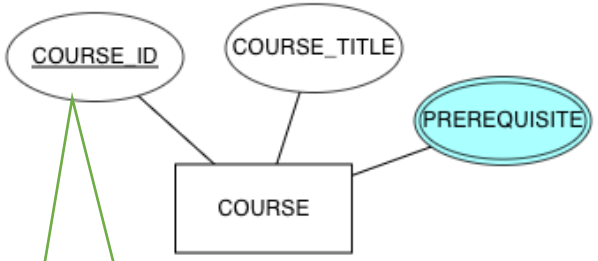
Primary key is underlined

Entity Instances (rows)

EMPLOYEE

EMPLOYEE_ID	EMP_NAME	DOB
1001	CHEN	12-FEB-1970
1002	MELTON	19-AUG-1986
1003	RITCHIE	26-MAR-1992
1004	CELKO	10-JUL-1990
1005	GOSLING	03-MAY-1976

MULTIVALUED→PREREQUISITE



Primary key is underlined

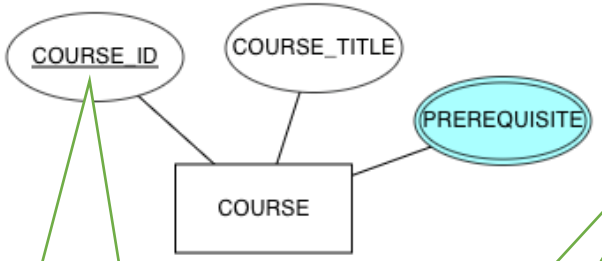
Primary key is underlined

COURSE

COURSE_ID	COURSE_TITLE	PREREQUISITE
C1	PROBLEM SOLVING	NULL
C2	JAVA	C1
C3	C#	C1
C4	DATA STRUCTURE	C2, C3
C5	RELATIONAL DATABASE	C4
C6	SQL	C5
C7	WEB APPLICATION DEVELOPMENT	C6, C8
C8	SPRINGBOOT	C2

multivalued

MULTIVALUED→PREREQUISITE



Primary key is underlined

Primary key is underlined

COURSE

COURSE_ID	COURSE_TITLE
C1	PROBLEM SOLVING
C2	JAVA
C3	C#
C4	DATA STRUCTURE
C5	RELATIONAL DATABASE
C6	SQL
C7	WEB APPLICATION DEVELOPMENT
C8	SPRINGBOOT

Composite primary keys COURSE_ID, PREQUISITE are underlined

COURSE-PRE EQUISITE

COURSE_ID	PREREQUISITE
C1	NULL
C2	C1
C3	C1
C4	C2
C4	C3
C5	C4
C6	C5
C7	C6
C7	C8
C8	C2

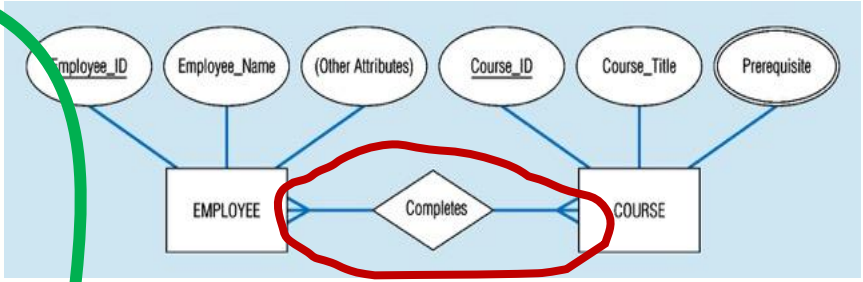
Foreign key COURSE_ID is dashed underlined

MANY-TO-MANY RELATIONSHIP

Identify the PK and FK

COMPLETED_COURSE

EMPLOYEE_ID	COURSE_ID
1001	C1
1001	C2
1001	C4
1002	C1
1002	C3
1002	C5
1003	C5
1004	C2
1004	C6
1005	C2
1005	C6



EMPLOYEE

EMPLOYEE_ID	EMPLOYEE_NAME	(… other attributes)
1001	CHEN	
1002	MELTON	
1003	RITCHIE	
1004	CELKO	
1005	GOSLING	

COURSE

COURSE_ID	COURSE_TITLE
C1	PROBLEM SOLVING
C2	JAVA
C3	C#
C4	DATA STRUCTURE
C5	RELATIONAL DATABASE
C6	SQL
C7	WEB APPLICATION DEVELOPMENT
C8	SPRINGBOOT

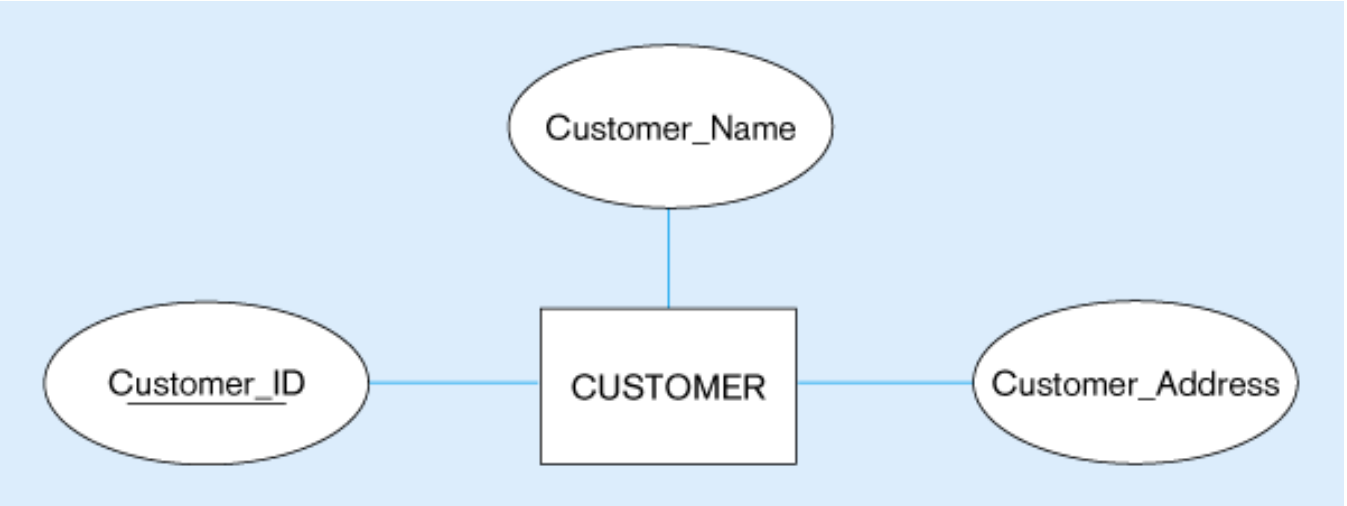
Transforming ER into Relation

Step 1: Mapping Regular Entities to Relations

i. Simple attributes: E-R attributes map directly onto the relation
ii. Composite attributes: Use only their simple, component attributes
iii. Multi-valued Attribute - Becomes a separate relation with a foreign key taken from the superior entity

i. Mapping a regular entity with simple attribute

(a) CUSTOMER entity type with simple attributes



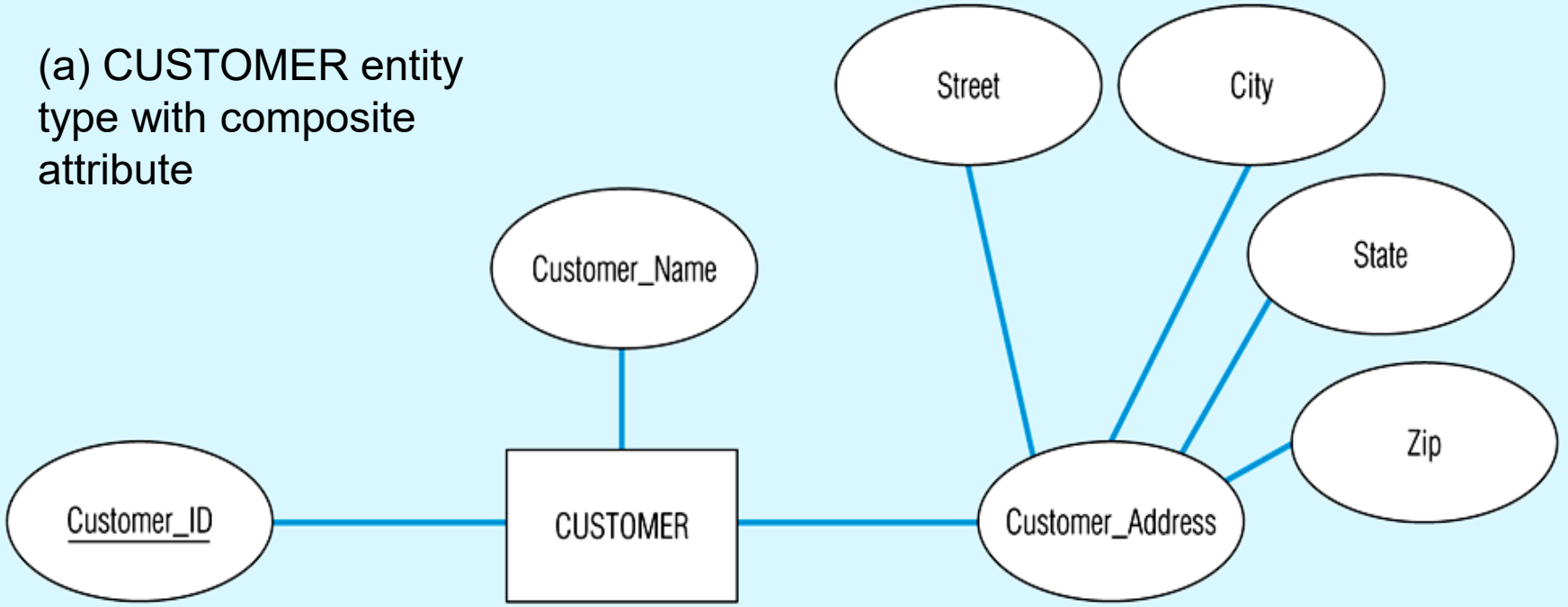
(b) CUSTOMER relation

CUSTOMER

Customer_ID	Customer_Name	Customer_Address

ii. Mapping a regular entity with composite attributes

(a) CUSTOMER entity type with composite attribute

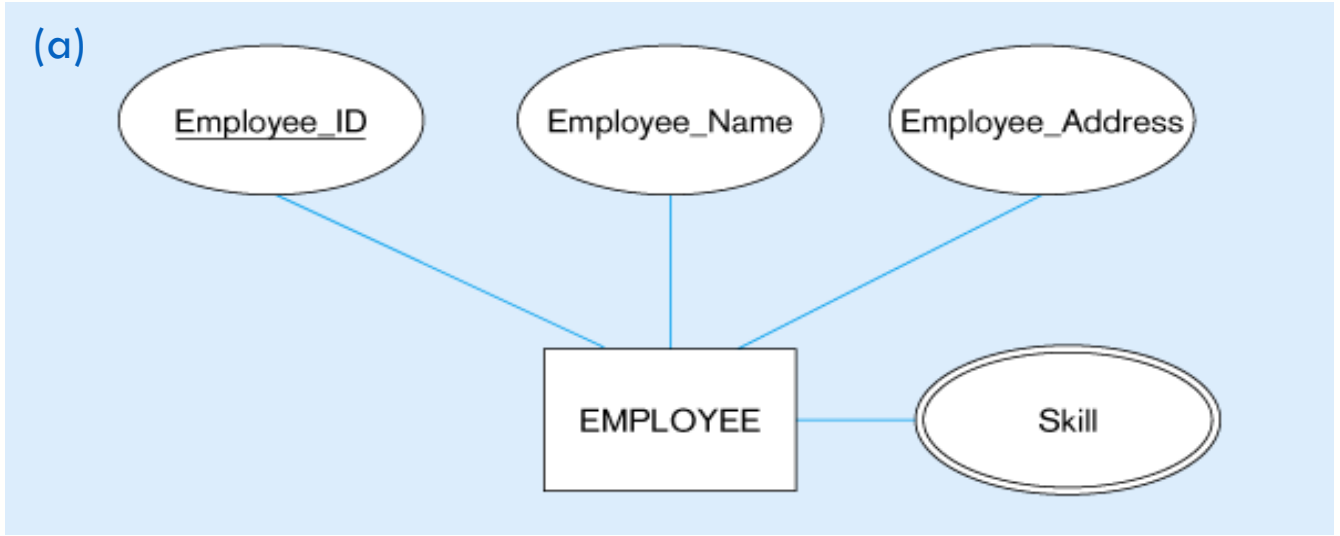


(b) CUSTOMER relation with address detail

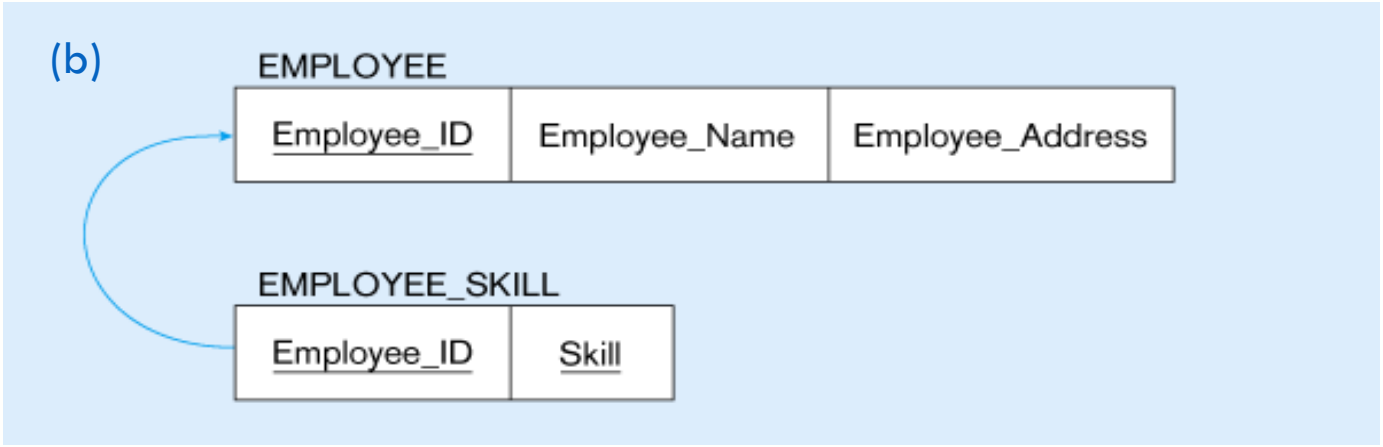
CUSTOMER

Customer_ID	Customer_Name	Street	City	State	Zip

iii. Mapping regular entity with a multivalued attribute



Multivalued attribute becomes a separate relation with foreign key



1 – to – many relationship between original entity and new relation

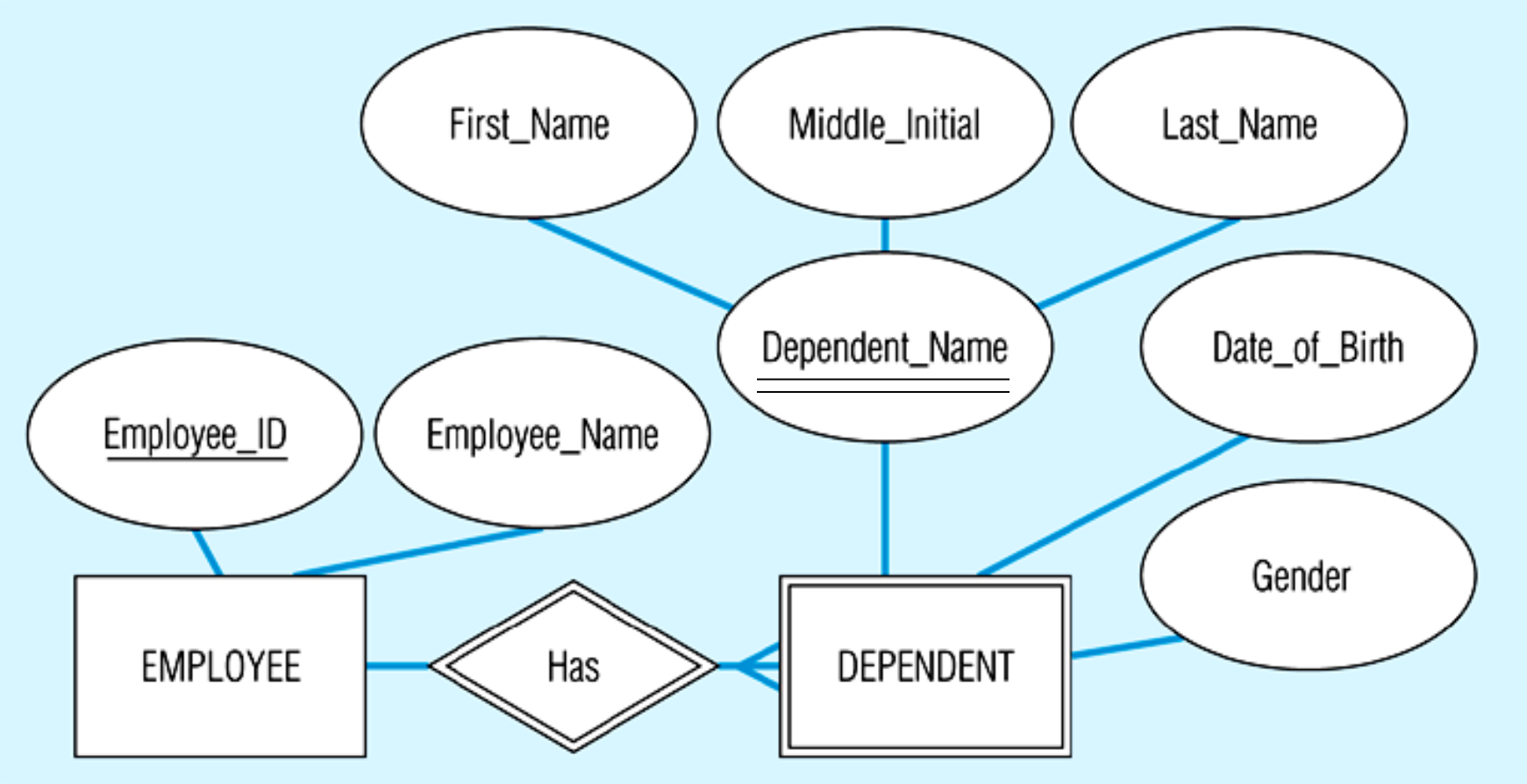
Transforming ER into Relation

Step 2: Mapping Weak Entities

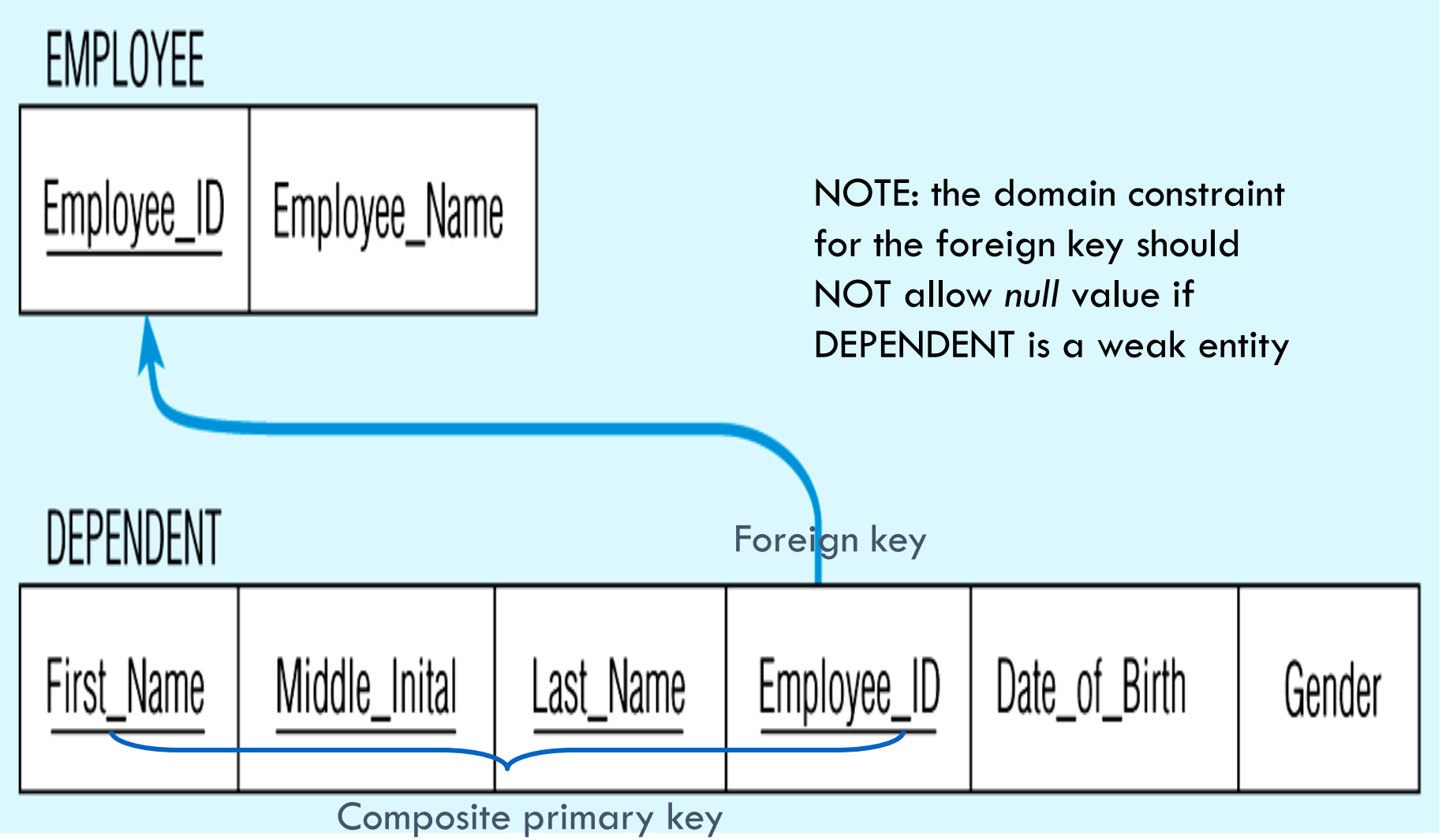
- Becomes a separate relation with a foreign key taken from the superior entity
- Primary key composed of:
  - Partial identifier of weak entity
  - Primary key of identifying relation (strong entity)

Mapping a weak entity

(a) Weak entity DEPENDENT



(b) Relations resulting from weak entity



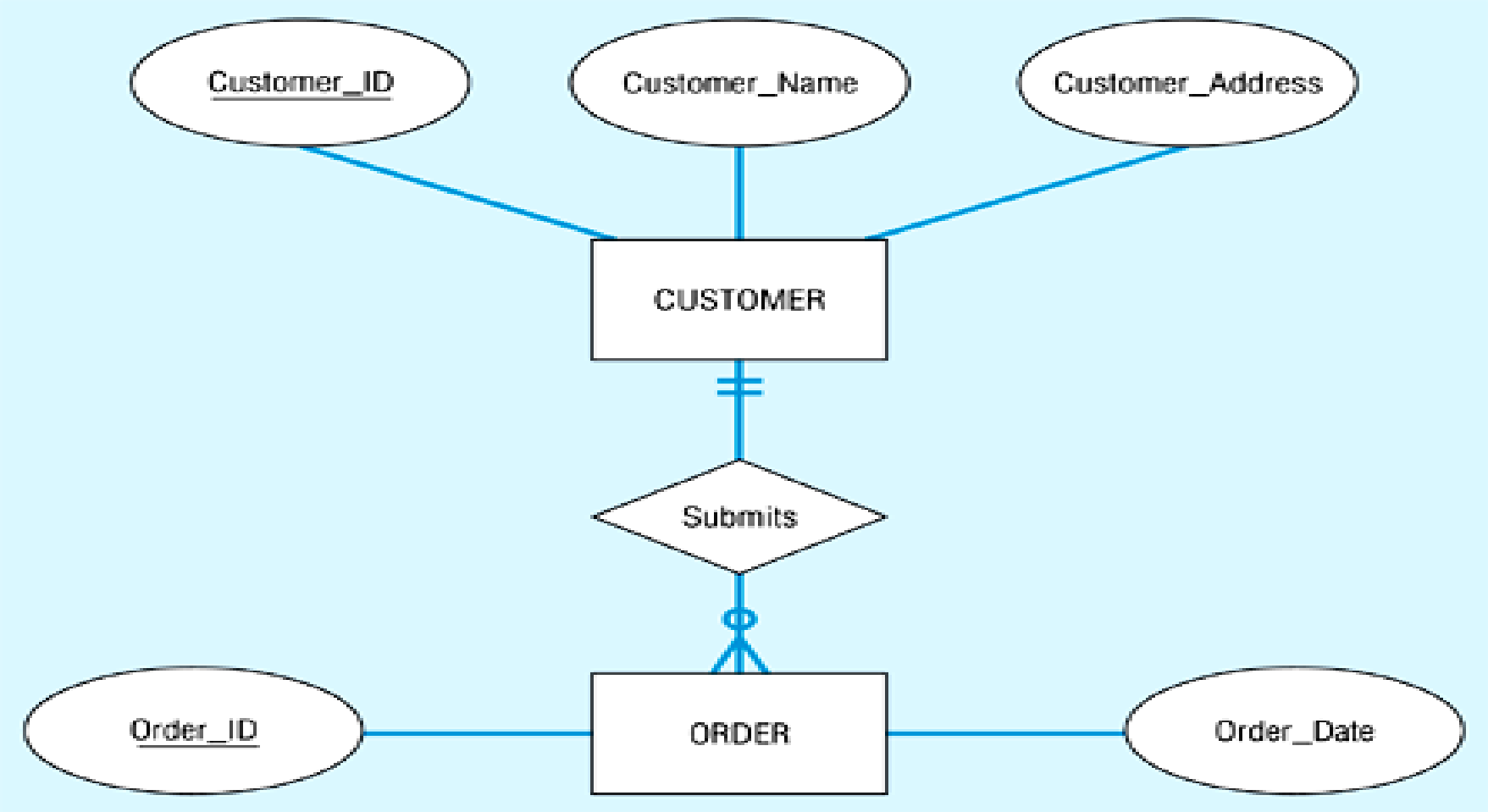
Transforming ER into Relation

Step 3: Mapping Binary Relationships

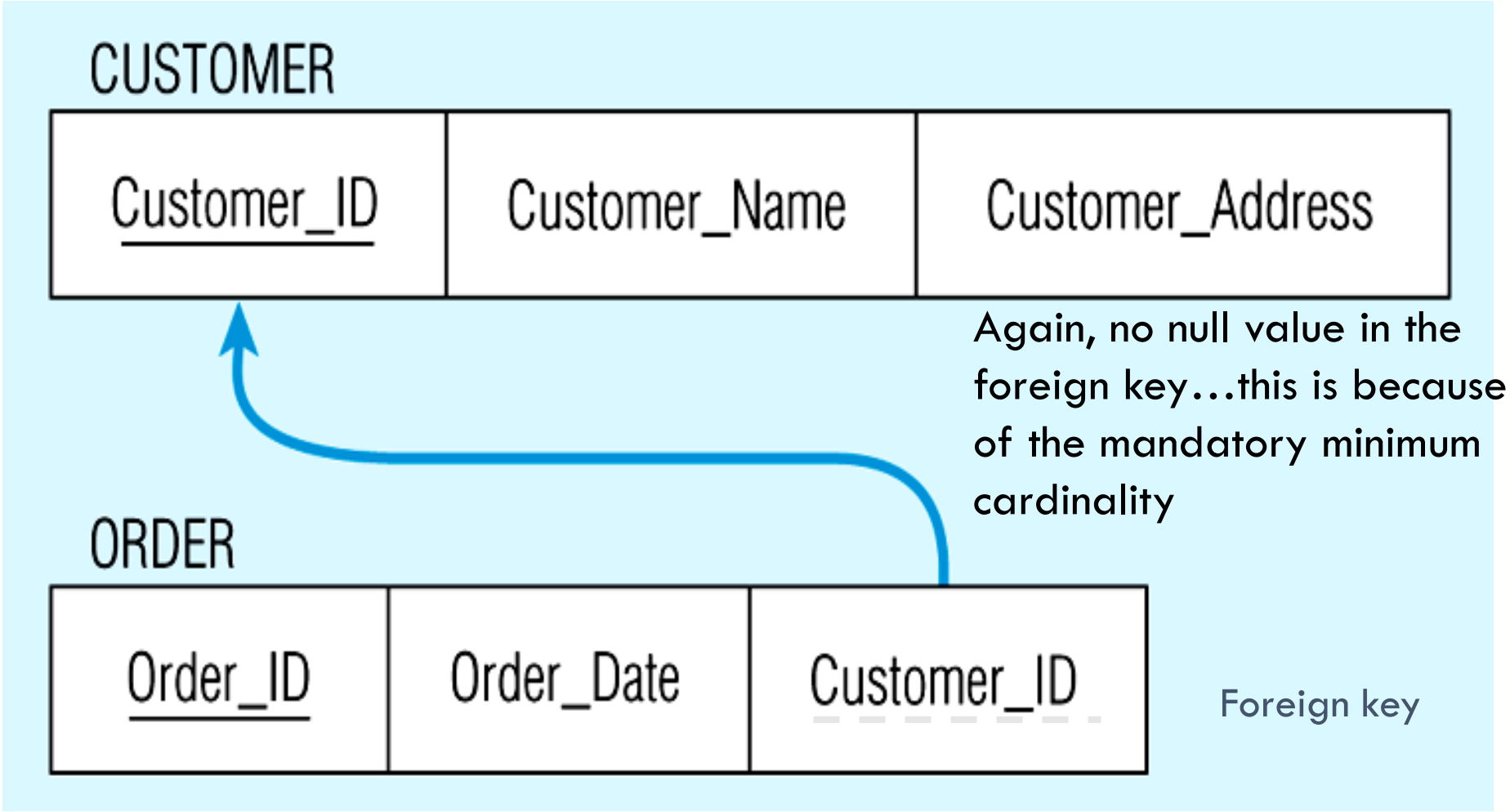
- One-to-Many - **Primary key** on the one side becomes a **foreign key** on the many side
- Many-to-Many - Create a *new relation* with the primary keys of the two entities as its primary key
- One-to-One - **Primary key** on the mandatory side becomes a **foreign key** on the optional side

Mapping a 1:M relationship

(a) Relationship between customers and orders

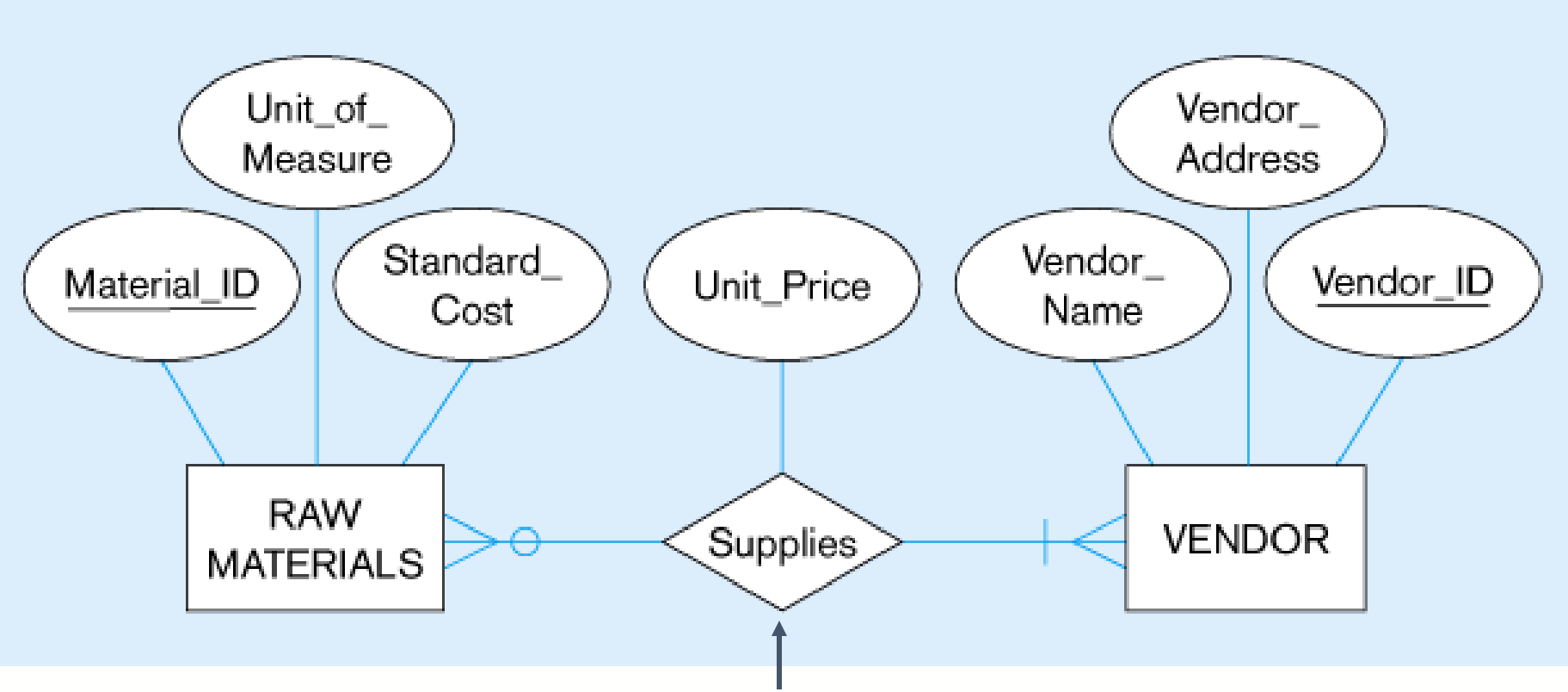


(b) Mapping the relationship



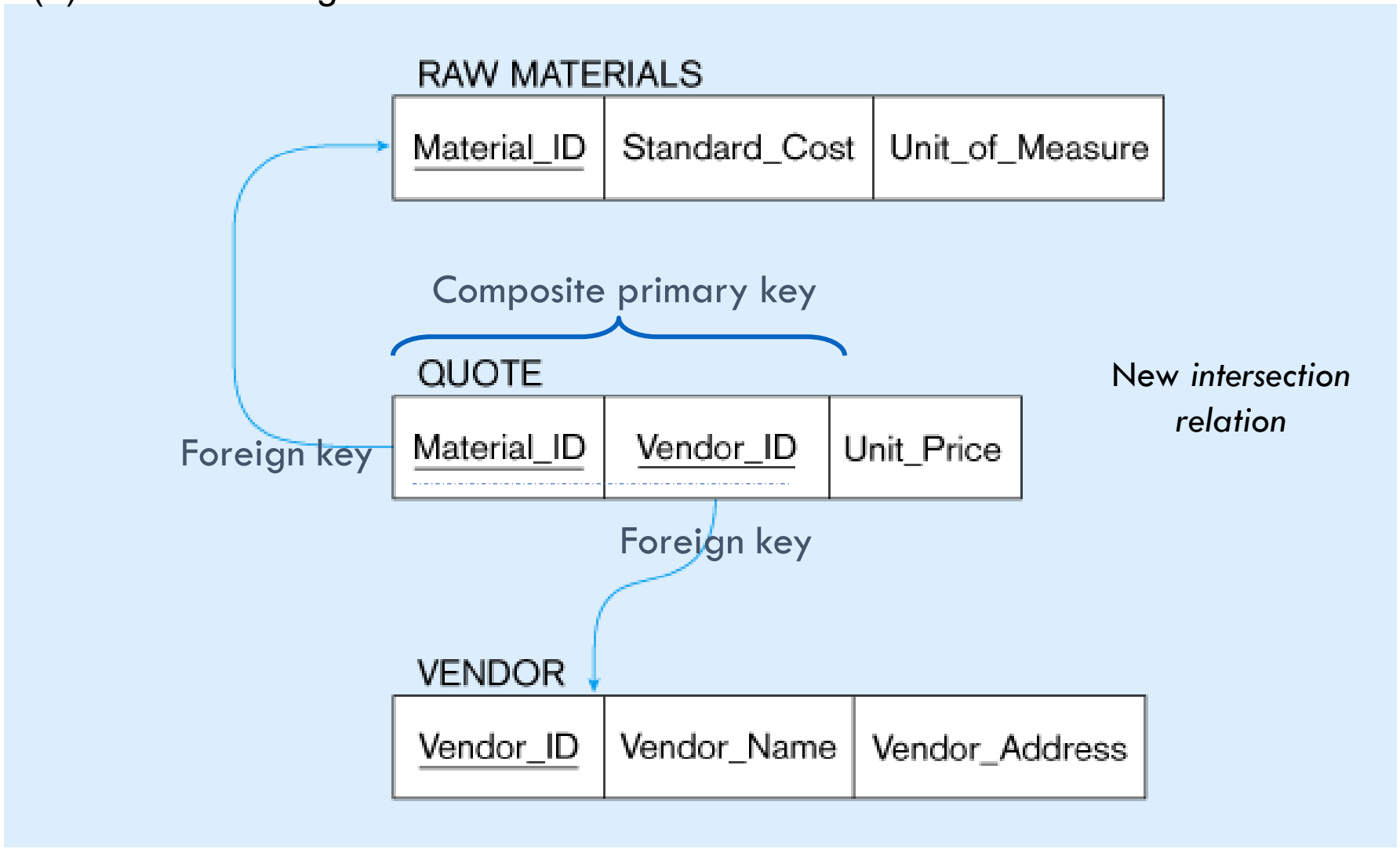
Mapping M:N relationship

(a) ER diagram (M:N)



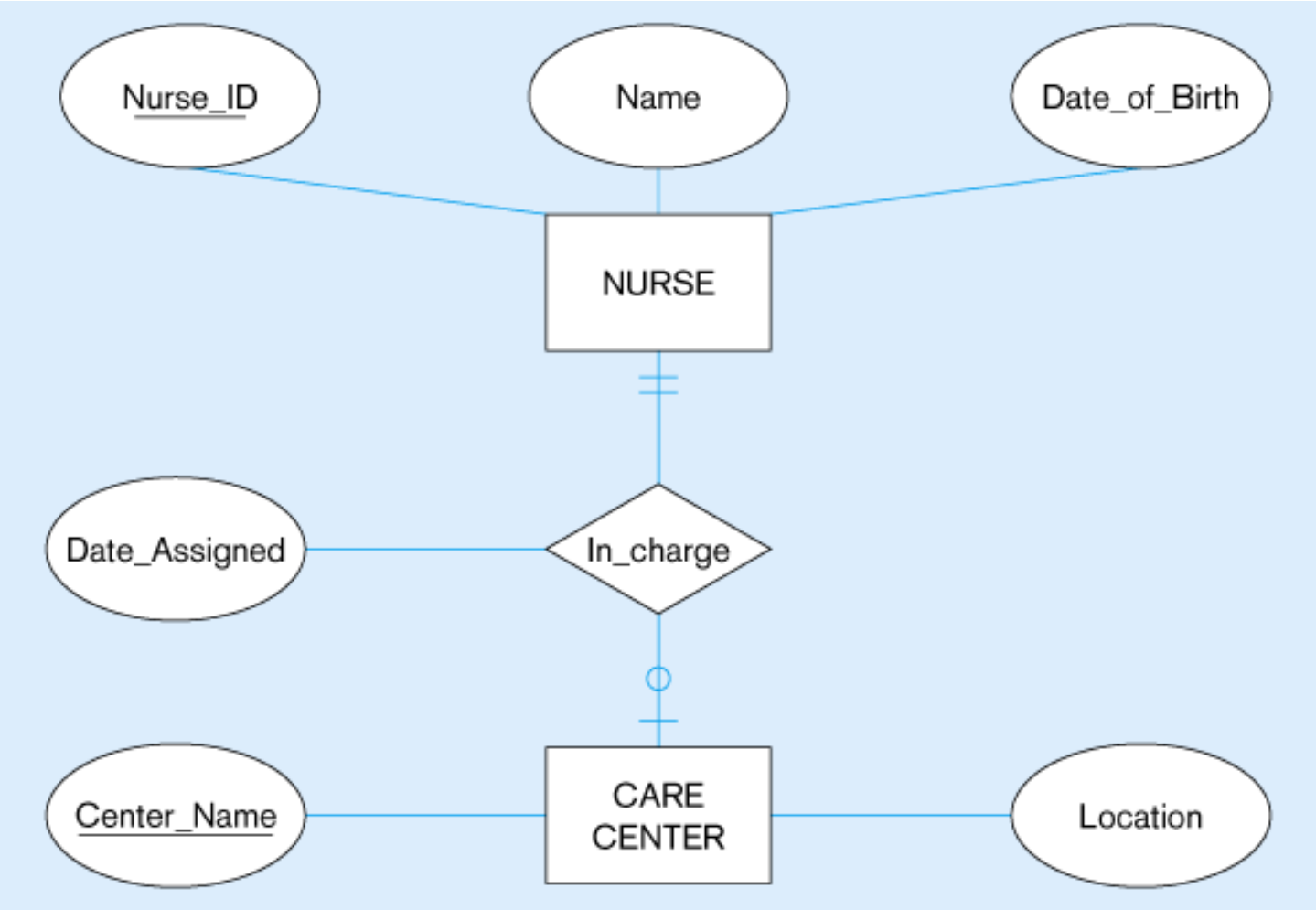
The *Supplies* relationship will need to become a separate relation

(b) Three resulting relations

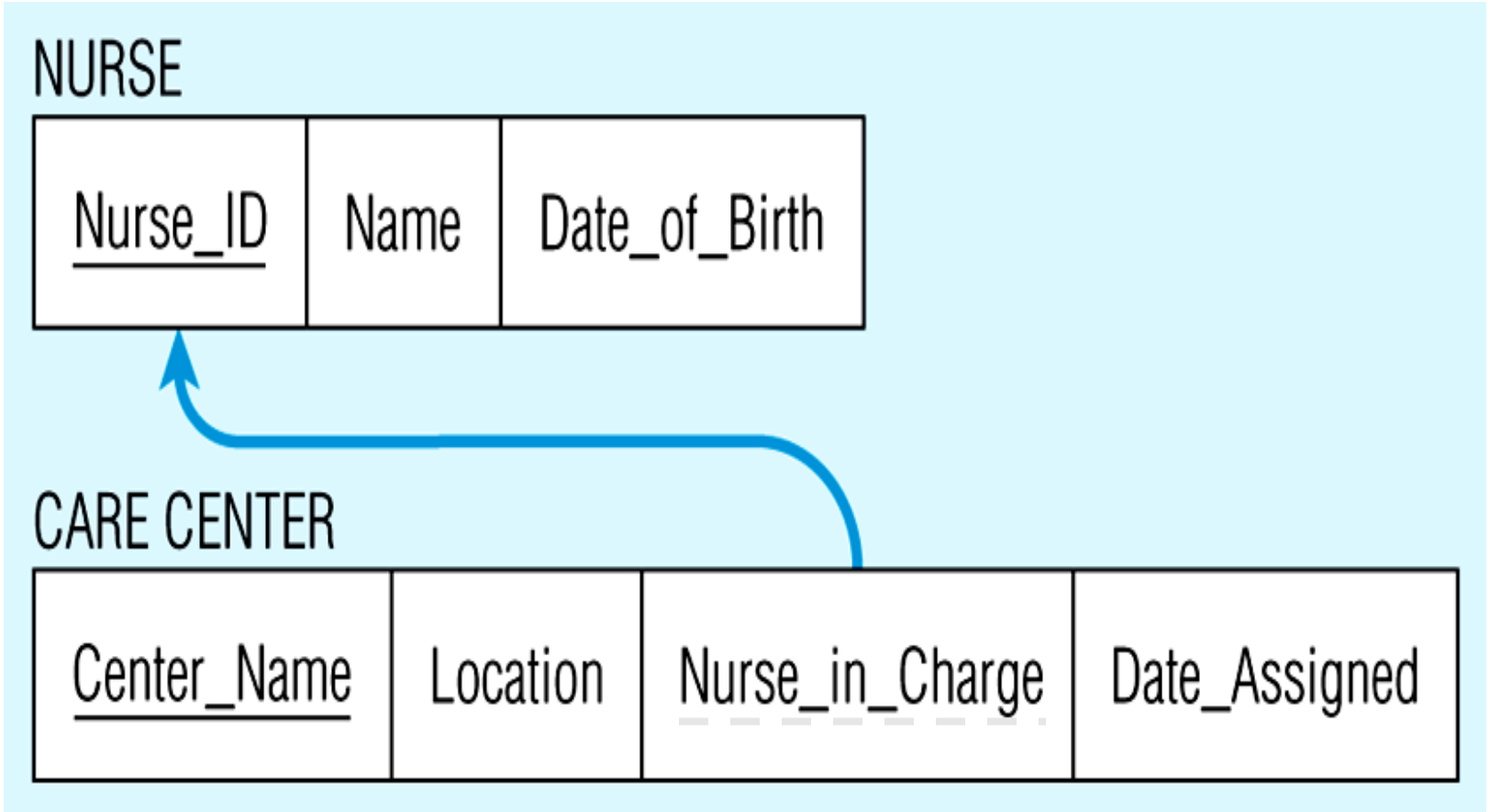


Mapping a binary 1:1 relationship

(a) Binary 1:1 relationship



(b) Resulting relations



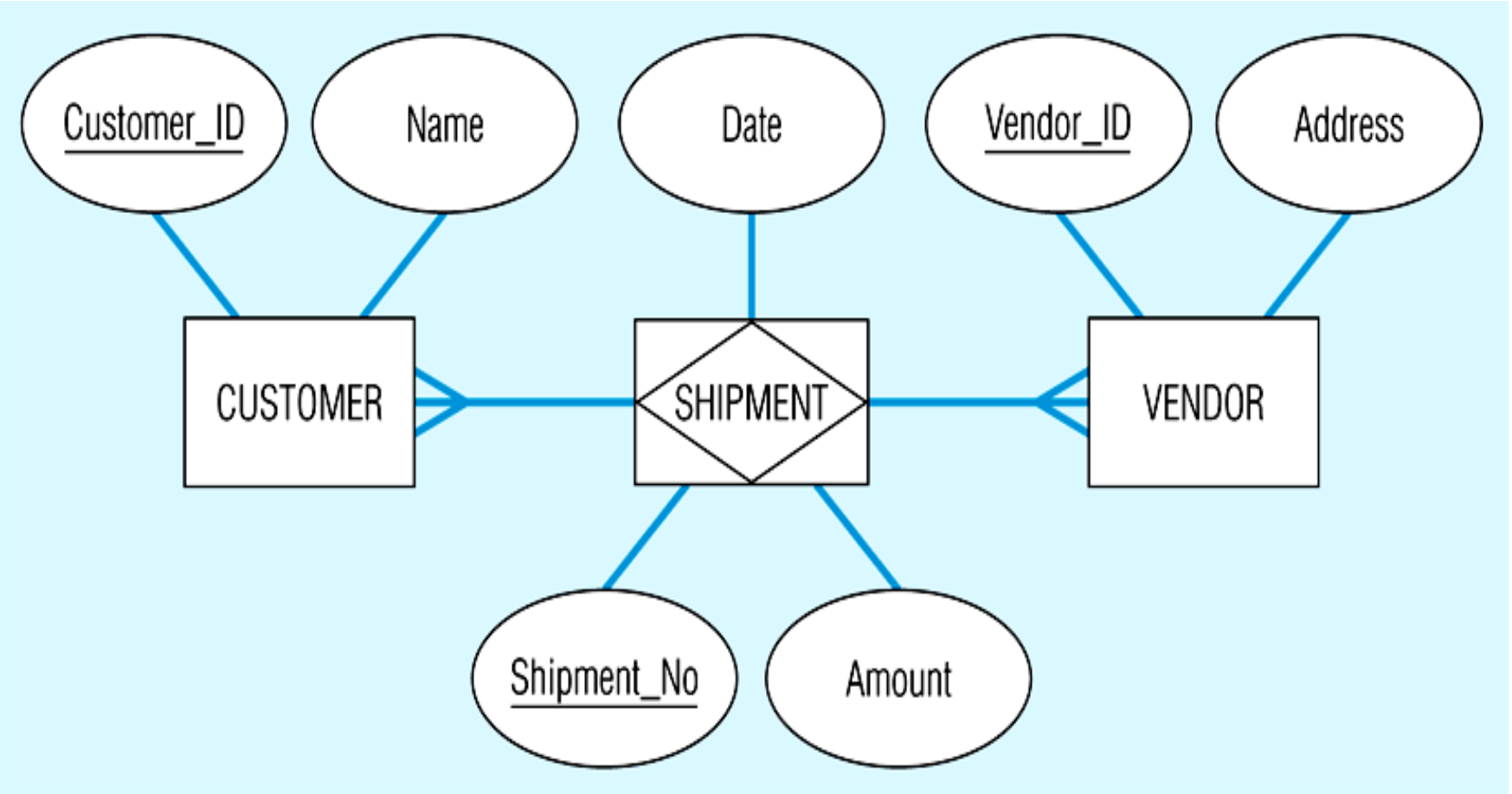
Transforming ER into Relation

Step 4: Mapping Associative Entities

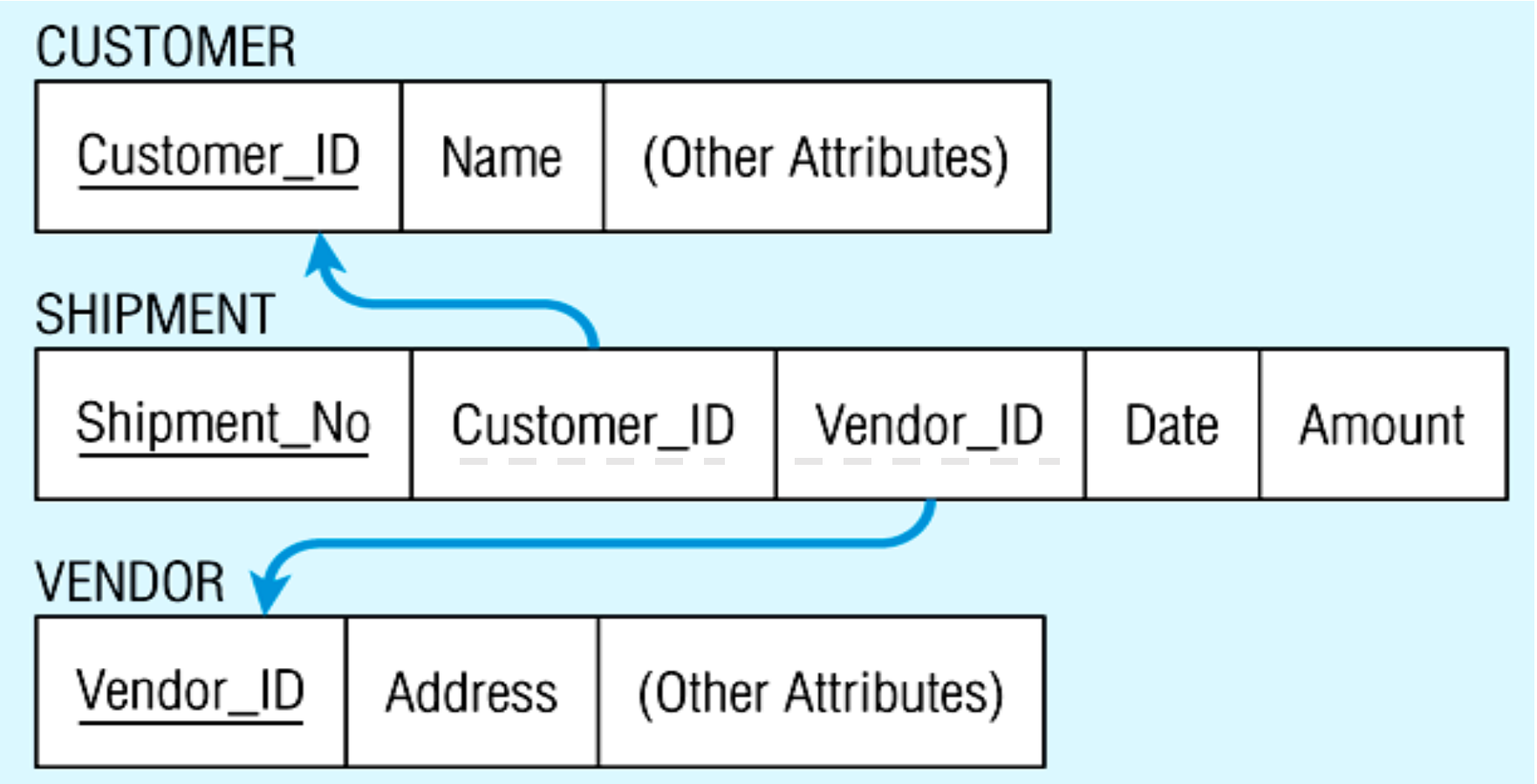
- Identifier Not Assigned
  - Default primary key for the association relation is composed of the primary keys of the two entities (as in M:N relationship)
- Identifier Assigned
  - It is natural and familiar to end-users
  - Default identifier may not be unique

Mapping an associative entity

(a) Associative entity



(b) Three resulting relations

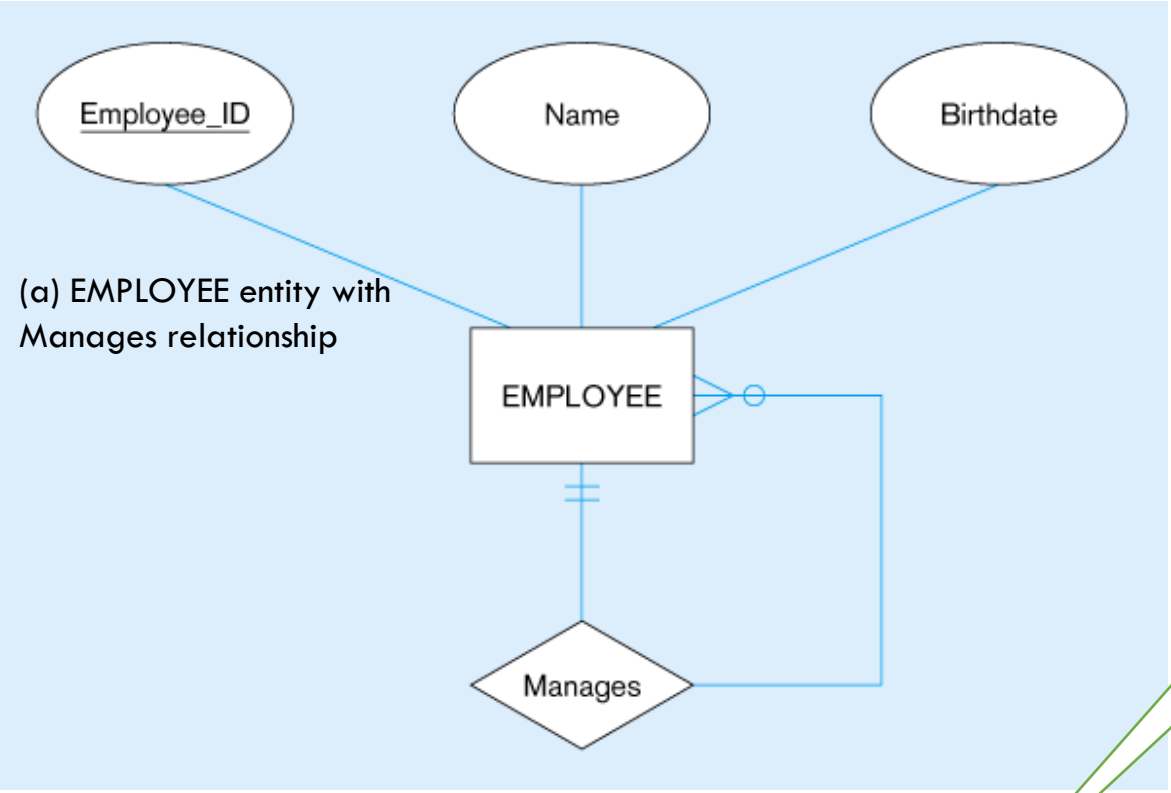


Transforming ER into Relation

Step 5: Mapping Unary Relationships

- One-to-Many - Recursive foreign key in the same relation
- Many-to-Many - Two relations:
  - One for the entity type
  - One for an associative relation in which the primary key has two attributes, both taken from the primary key of the entity

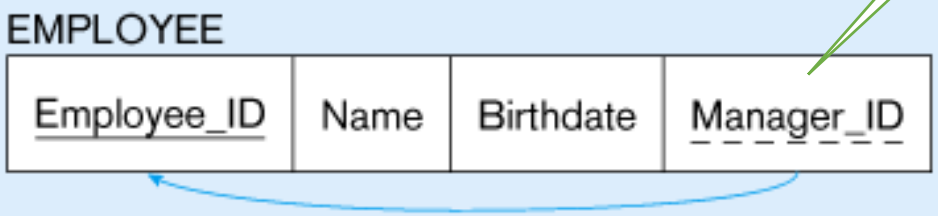
Mapping a unary 1:N relationship



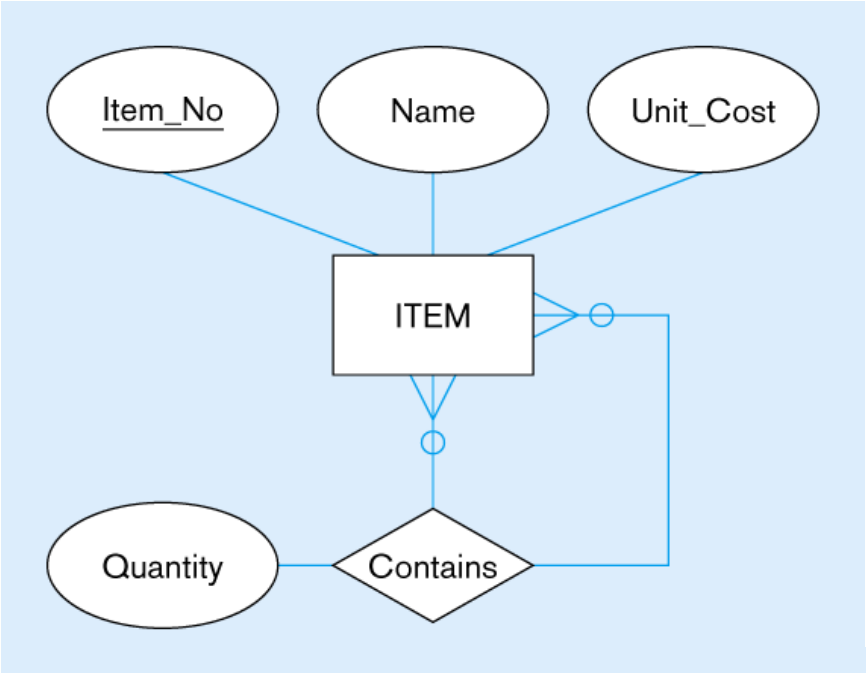
(a) EMPLOYEE entity with Manages relationship

PK Employee_ID renamed as Manager_ID to be a FK

(b) EMPLOYEE relation with recursive foreign key

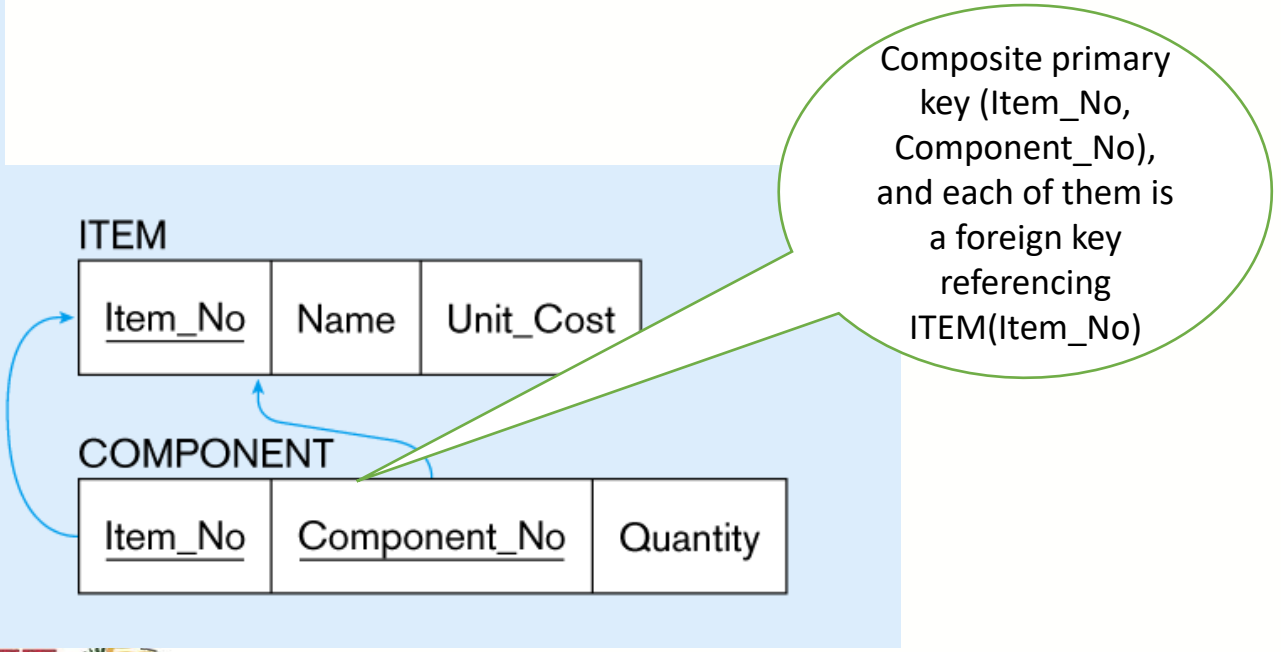


Mapping a unary M:N relationship

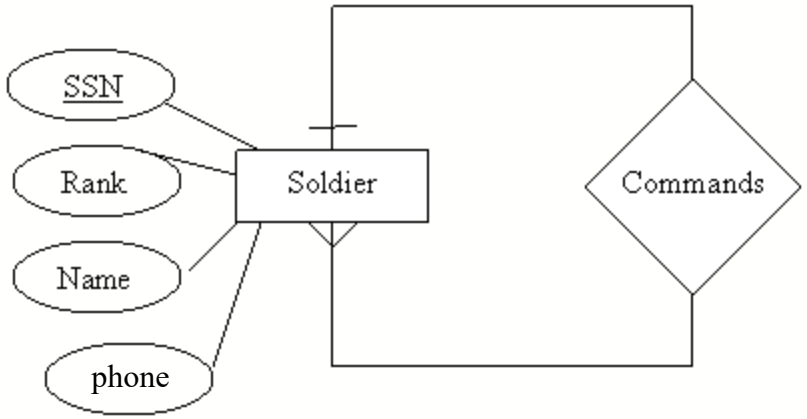


(a) Bill-of-materials relationships (M:N)

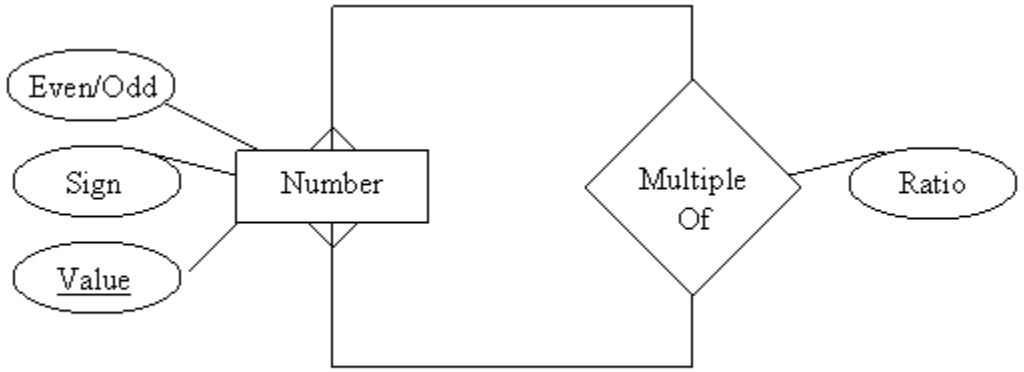
(b) ITEM and COMPONENT relations



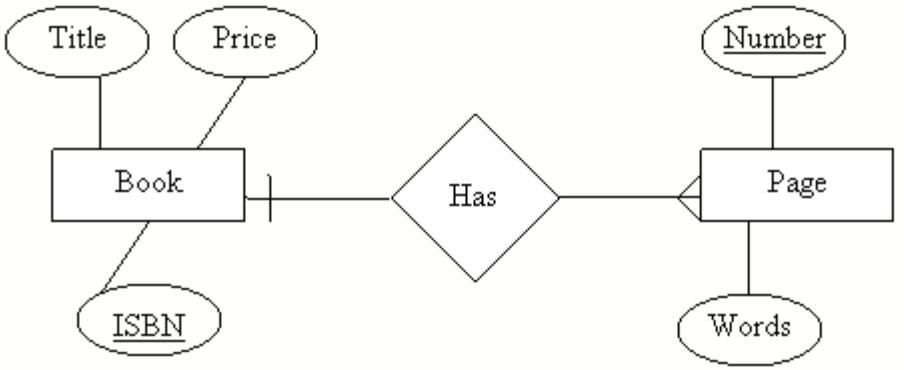
Practice 1



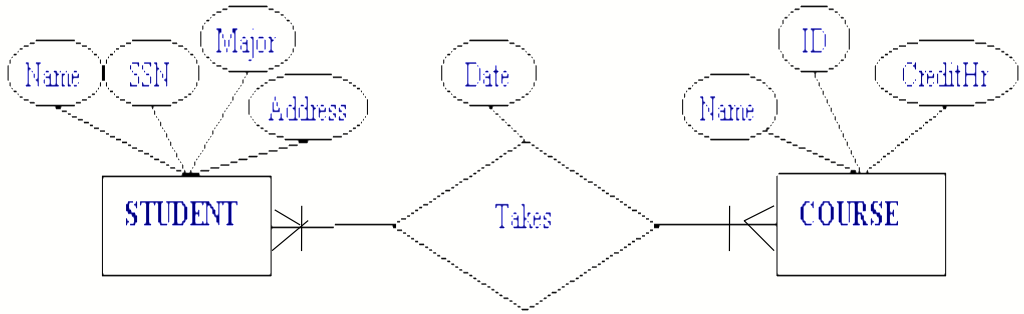
Practice 2



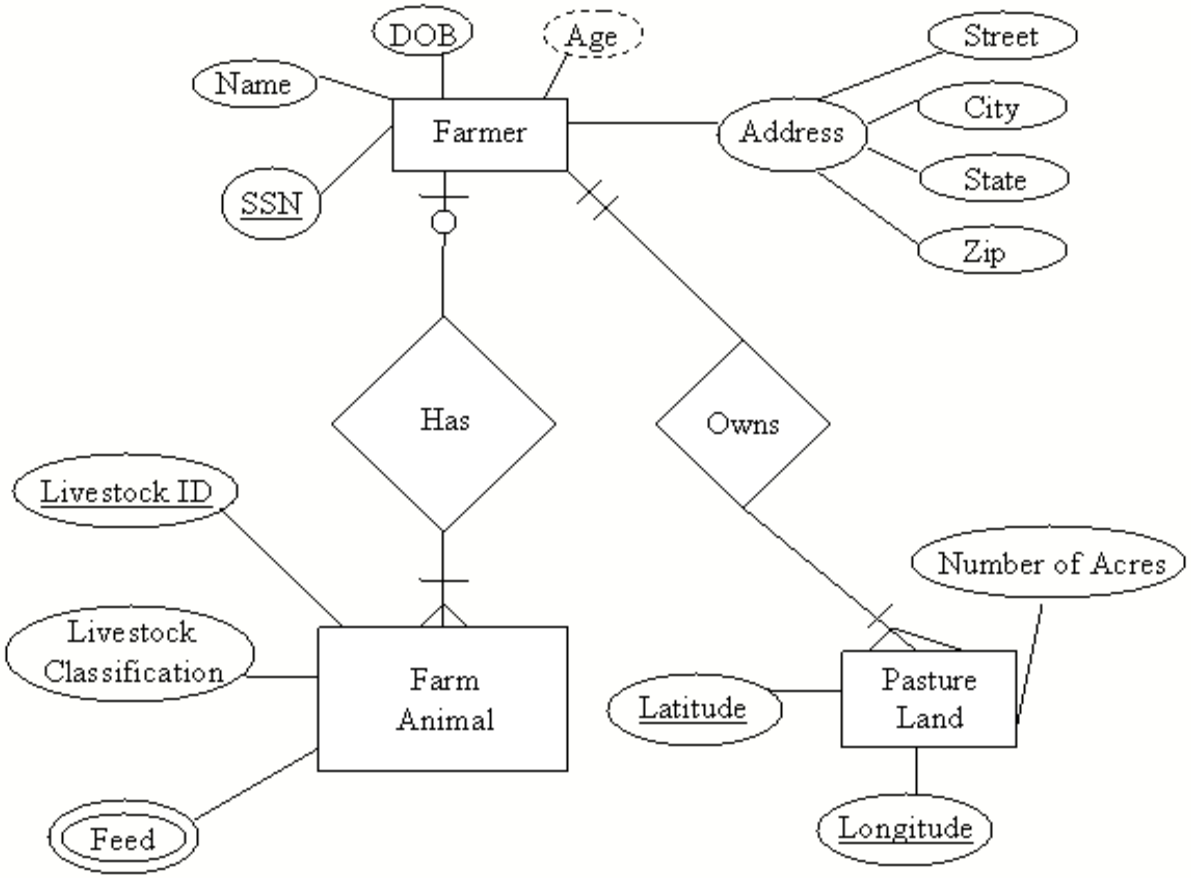
Practice 3



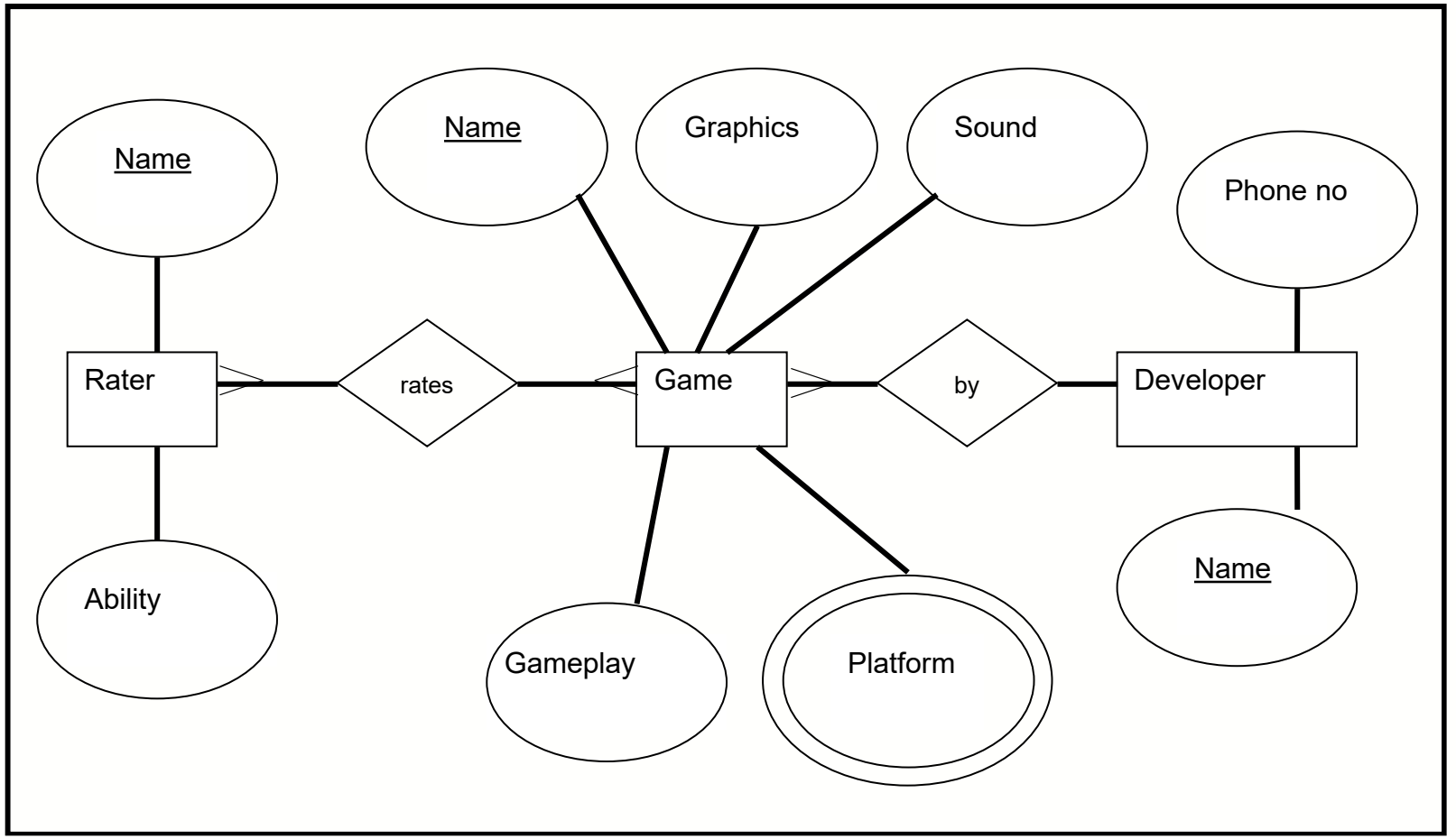
Practice 4


Name
SSN
Major
Address
STUDENT
Date
Takes
ID
Name
CreditHr
COURSE

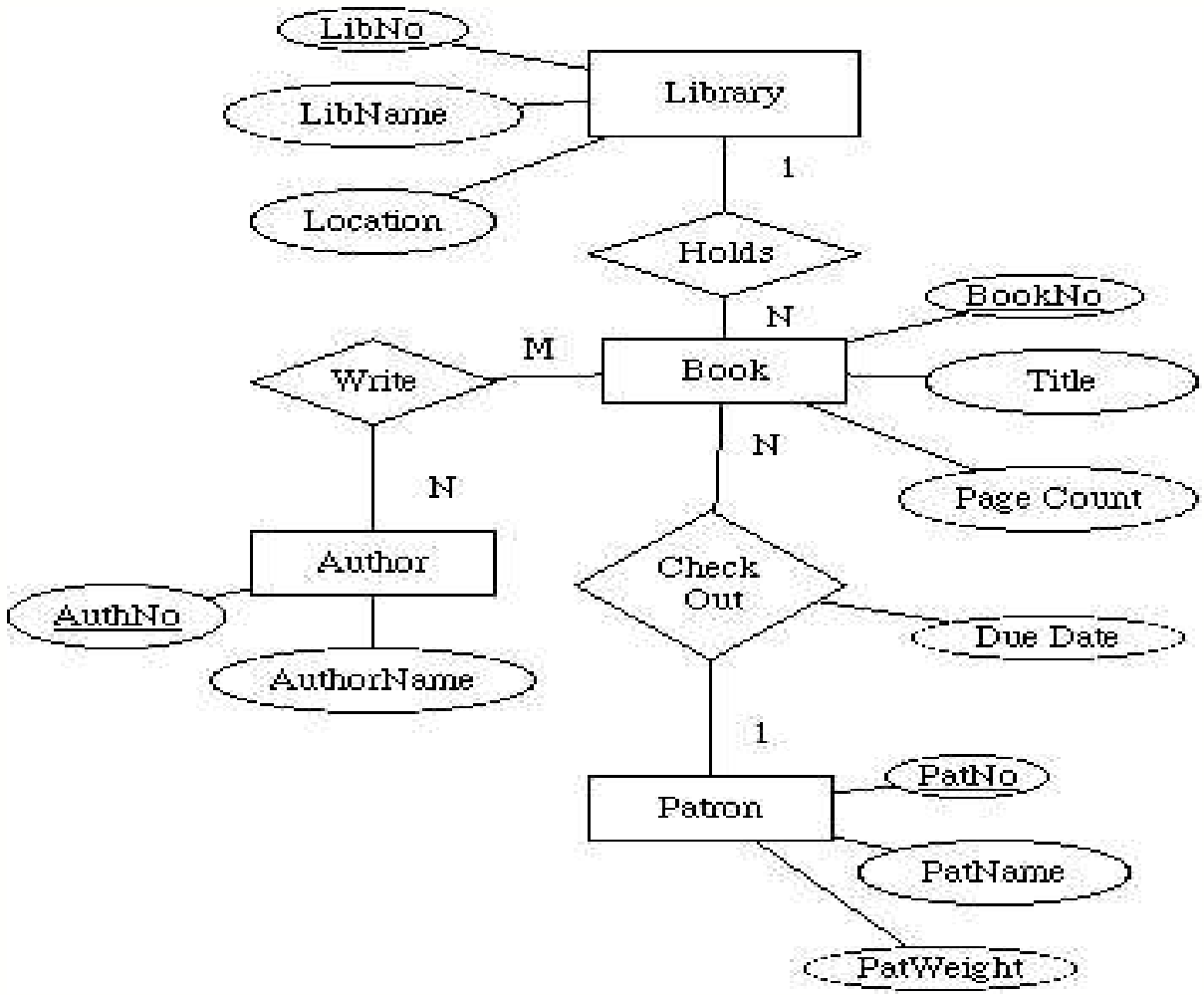
Practice 5


DOB
Age
Name
Farmer
SSN
Street
City
Address
State
Zip
Has
Owns
Livestock ID
Number of Acres
Livestock Classification
Farm Animal
Pasture Land
Latitude
Feed
Longitude

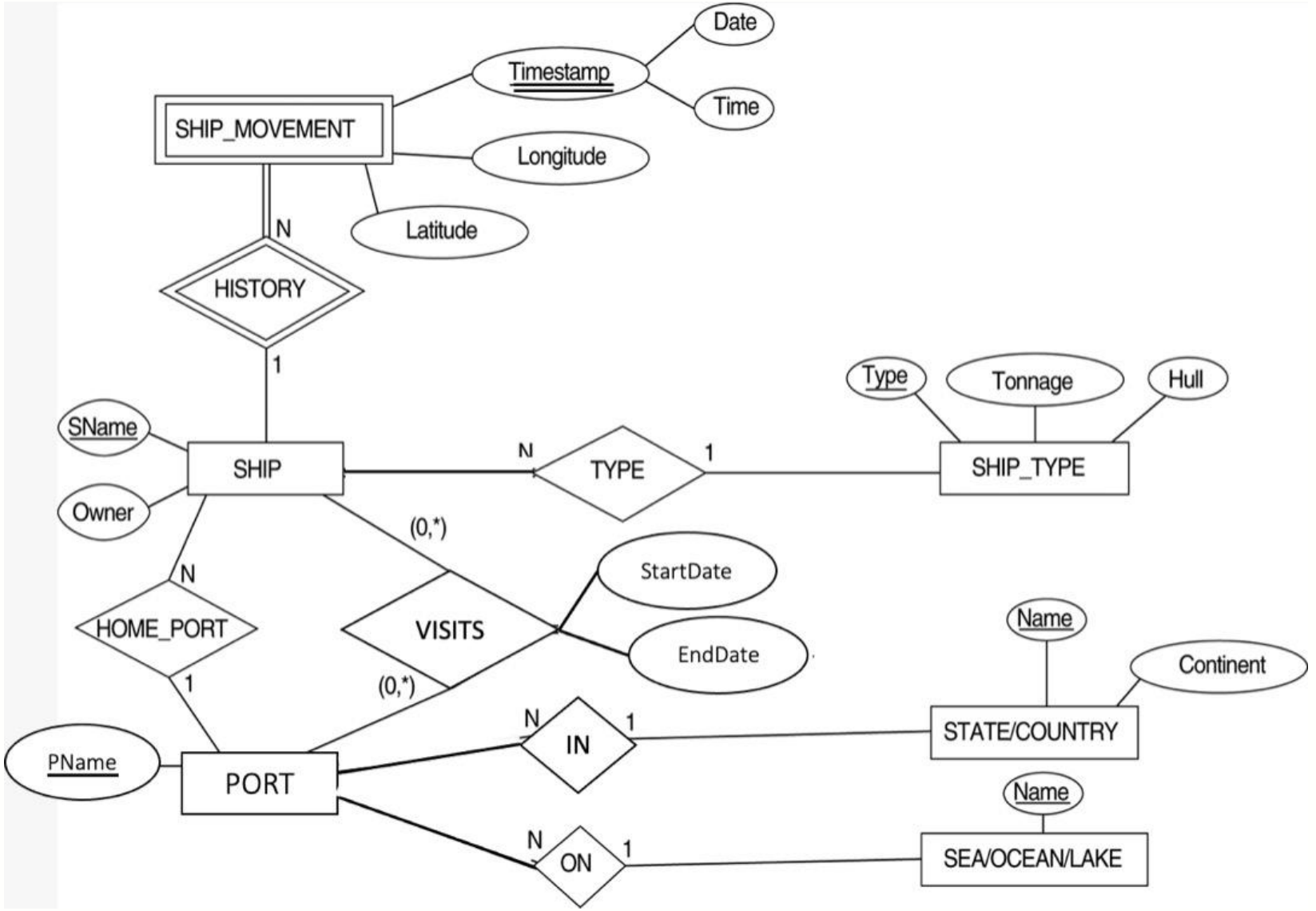
Practice 6


Name
Name
Graphics
Sound
Phone no
Rater
rates
Game
by
Developer
Ability
Gameplay
Platform
Name

Practice 7


LibNo
LibName
Location
Library
1
Holds
N
BookNo
M
Write
Book
Title
N
N
Page Count
Author
Check
Out
AuthNo
Due Date
AuthorName
1
PatNo
Patron
PatName
PatWeight

Practice 8


Date
Timestamp
Time
SHIP_MOVEMENT
Longitude
Latitude
N
HISTORY
1
Type
Tonnage
Hull
SName
SHIP
N
TYPE
1
SHIP_TYPE
Owner
(0,*)
N
StartDate
HOME_PORT
VISITS
EndDate
Name
Continent
1
(0,*)
N
1
PName
PORT
IN
STATE/COUNTRY
Name
N
1
ON
SEA/OCEAN/LAKE

Transforming EER into Relation

Step1: Options for Mapping Specialization or Generalization.

- Convert each specialization with m subclasses {S1, S2,….,Sm} and generalized superclass C, where the attributes of C are {k,a1,…an} and k is the (primary) key, into relational schemas using one of the four following options:
  - Option 1A: Multiple relations-Superclass and subclasses (M-Sp-Sb)
  - Option 1B: Multiple relations-Subclass relations only (M-Sb)
  - Option 1C: Single relation with one type attribute (S-Ot)
  - Option 1D: Single relation with multiple type attributes (S-Mt)

Transforming EER into Relation

Option 1A: 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 or over-lapping).

Option 1B: 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).

Transforming EER into Relation

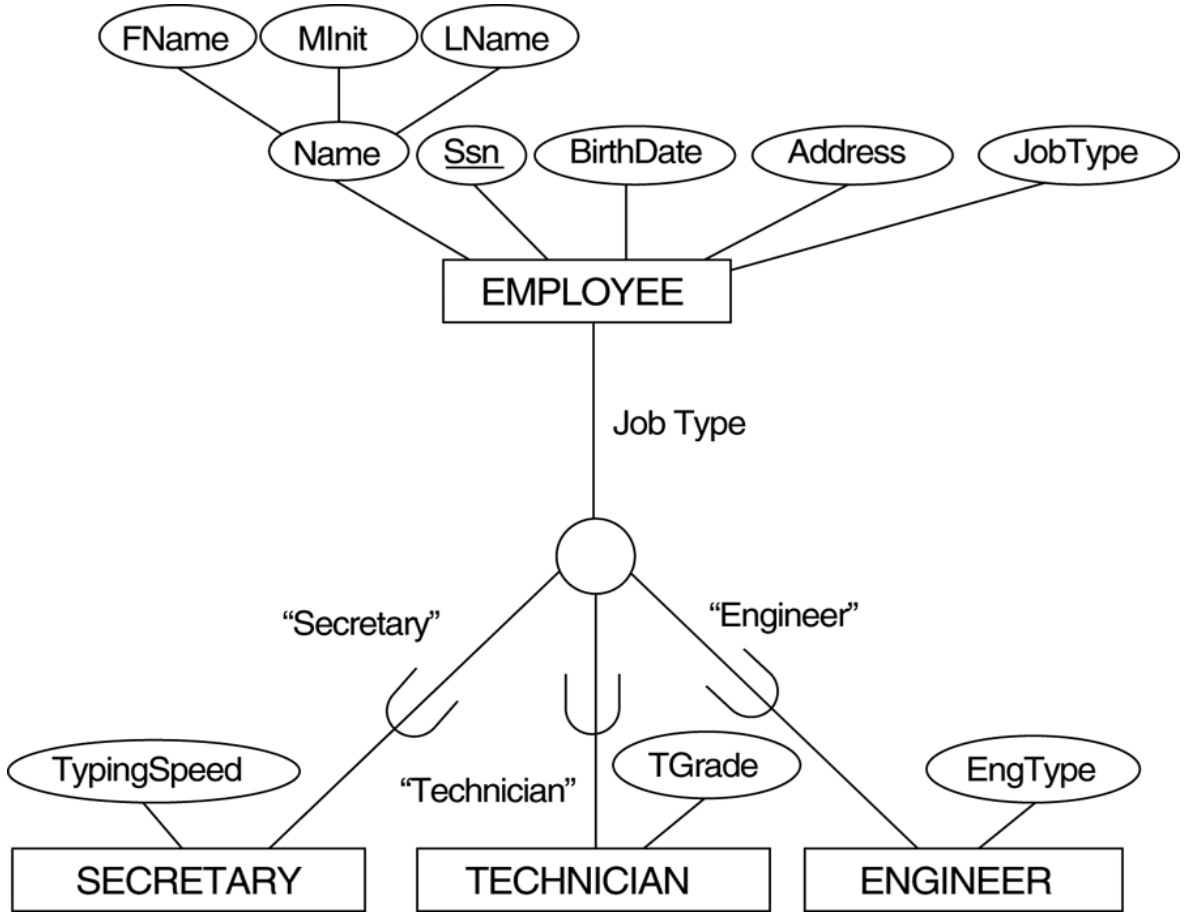
Option 1C: Single relation with one type attribute

- 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

Option 1D: Single relation with multiple type attributes

- Create a single relation schema 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 diagram notation for an attribute-defined specialization on JobType.



Mapping the EER schema using option 1A

(a)

EMPLOYEE

SSN	FName	MInit	LName	BirthDate	Address	JobType

SECRETARY

SSN	TypingSpeed

TECHNICIAN

SSN	TGrade

ENGINEER

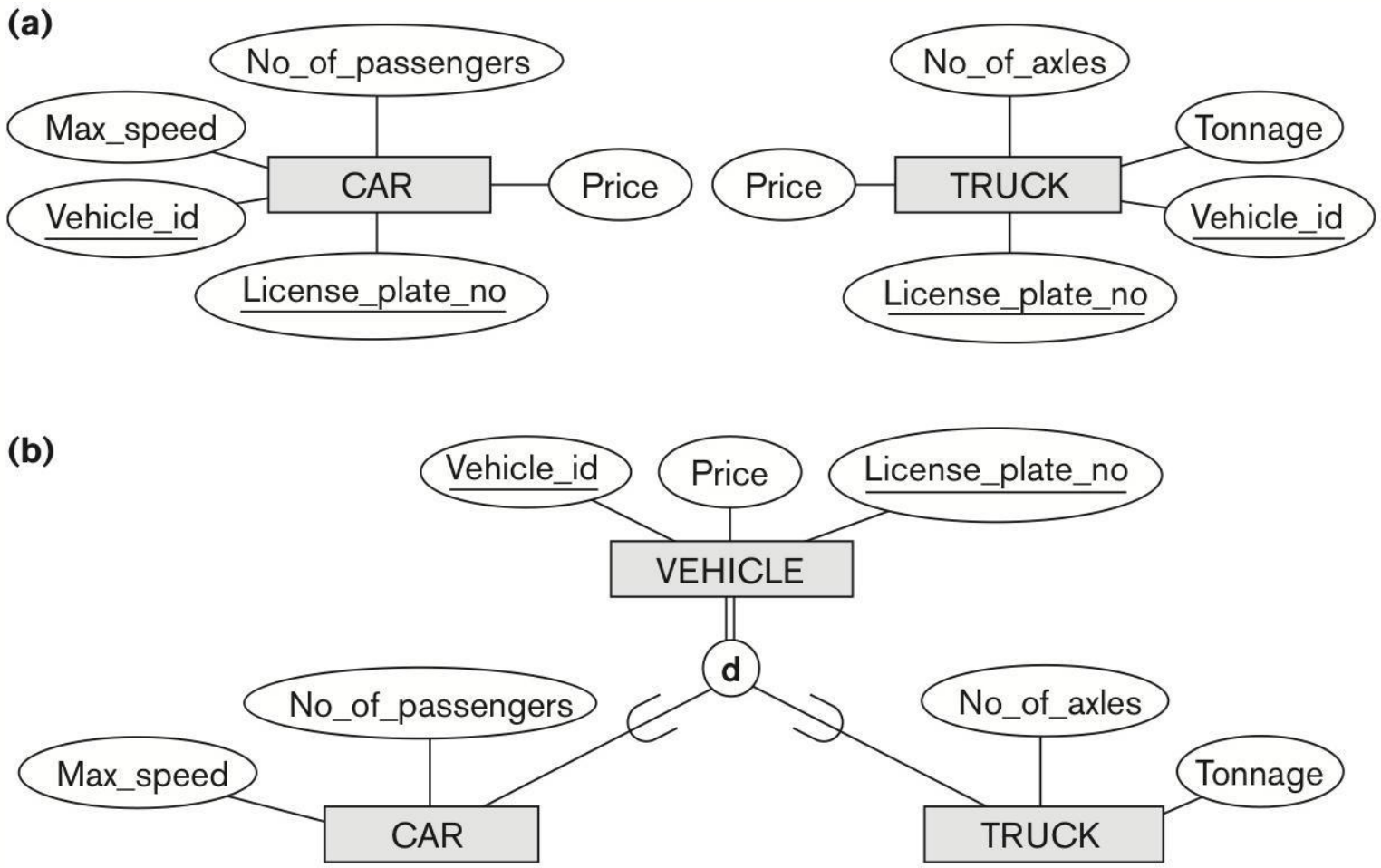
SSN	EngType

Mapping the EER schema using option 1C

(c) EMPLOYEE

SSN	FName	MInit	LName	BirthDate	Address	JobType	TypingSpeed	TGrade	EngType

Generalizing CAR and TRUCK into the superclass VEHICLE.



Mapping the EER schema using option 1B

(b)

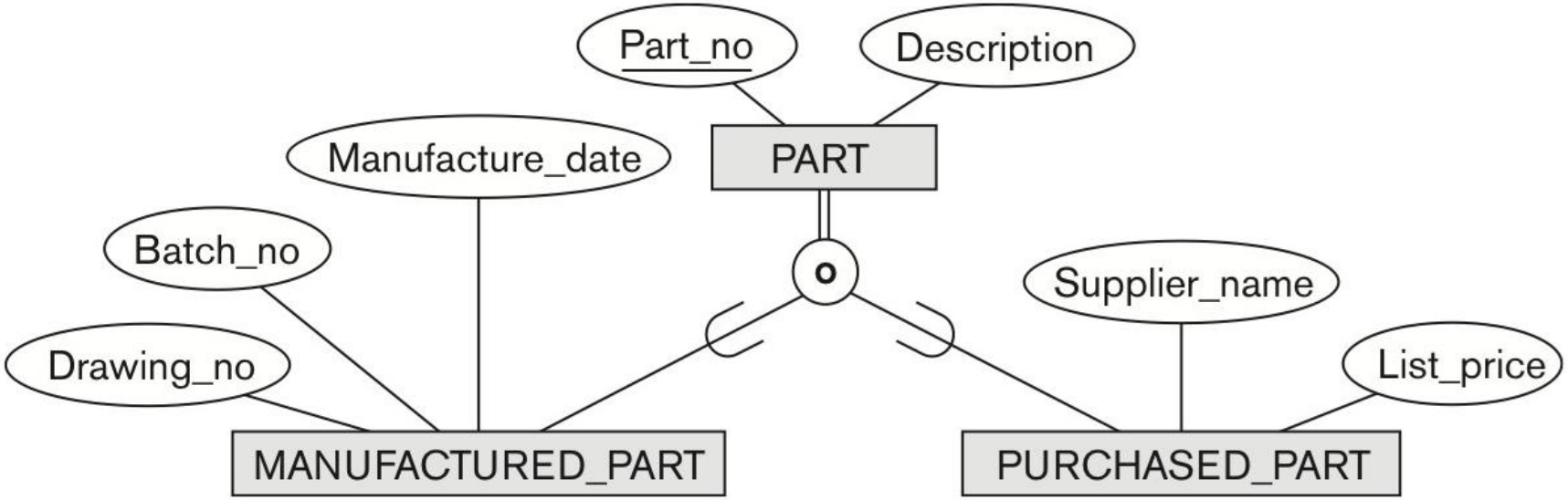
CAR

VehicleId	LicensePlateNo	Price	MaxSpeed	NoOfPassengers

TRUCK

VehicleId	LicensePlateNo	Price	NoOfAxles	Tonnage

An overlapping (non-disjoint) specialization.



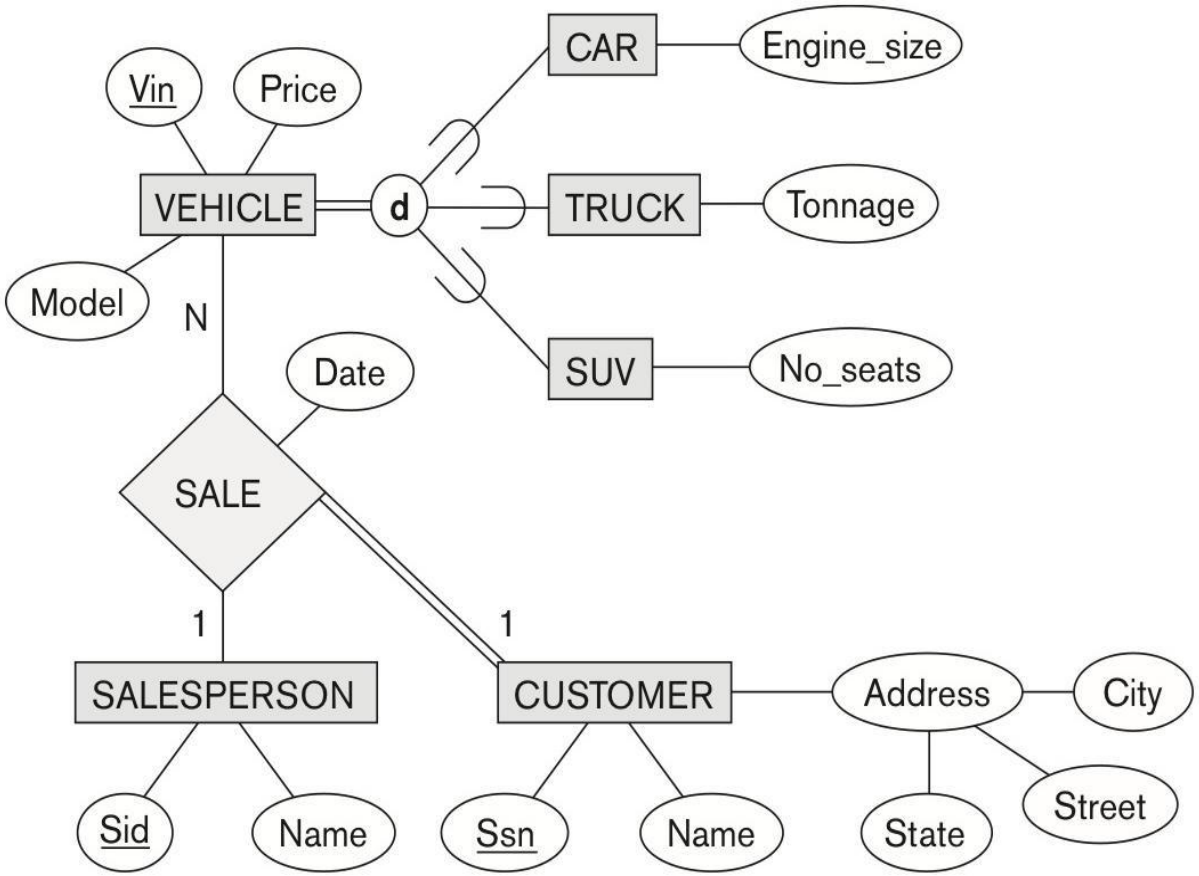
Mapping the EER schema using option 1D

(d) PART

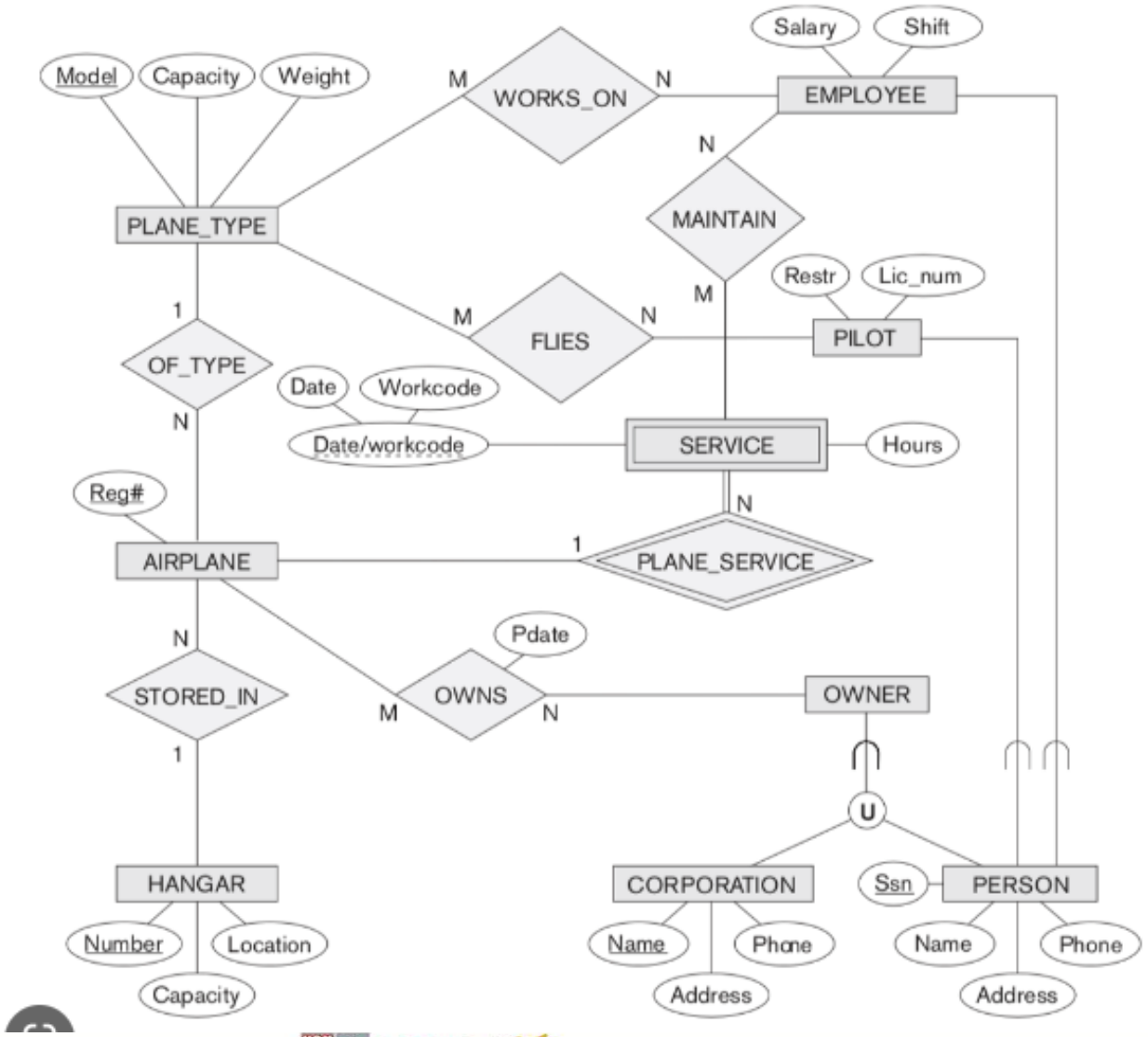
PartNo	Description	MFlag	DrawingNo	ManufactureDate	BatchNo	PFlag	SupplierName	ListPrice

*Mflag and PFlag are Boolean type fields.

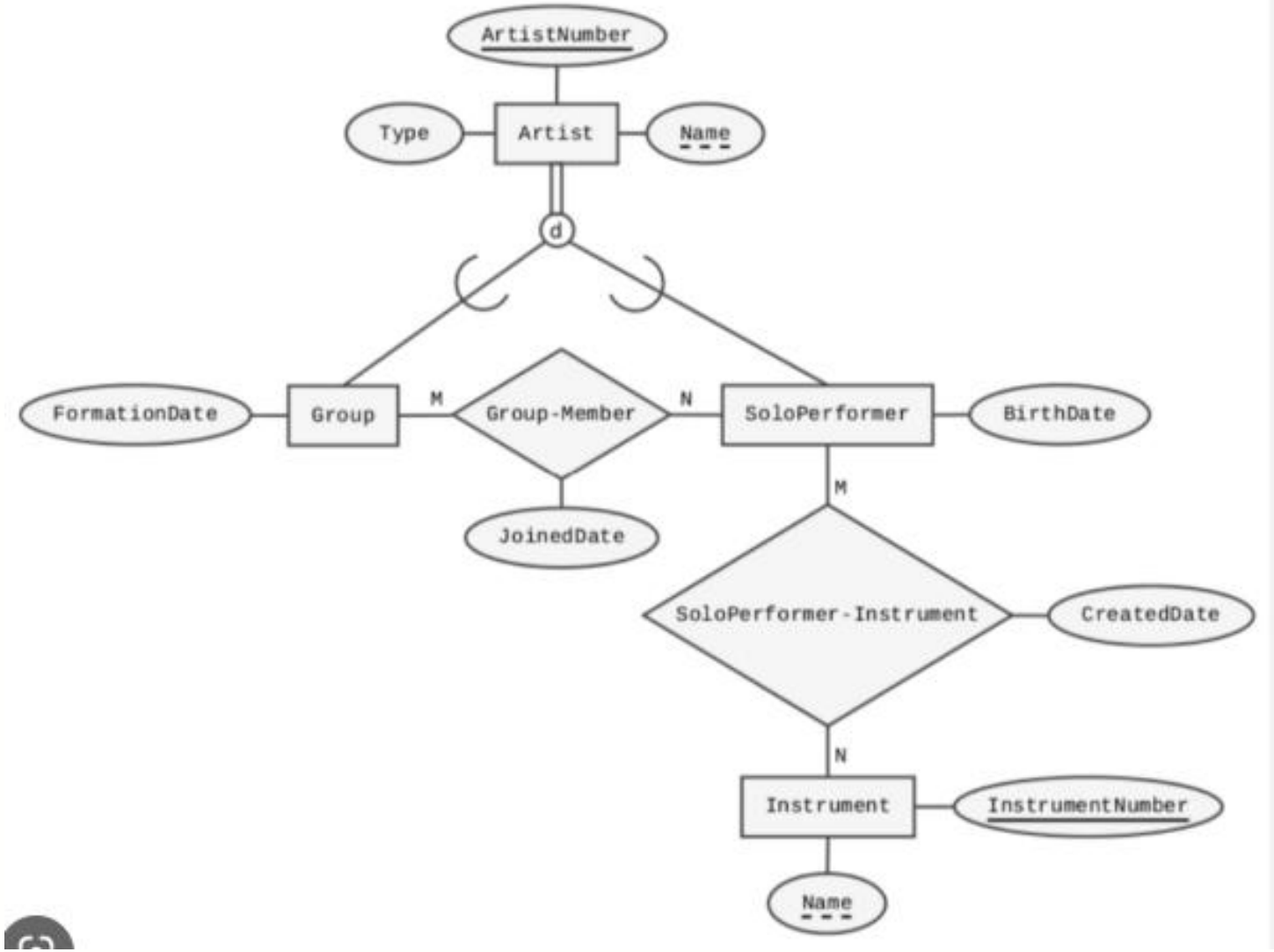
Practice 1


CAR
Engine_size
Vin
Price
VEHICLE
d
TRUCK
Tonnage
Model
N
SUV
No_seats
Date
SALE
1
1
SALESPERSON
CUSTOMER
Address
City
Sid
Name
Ssn
Name
State
Street

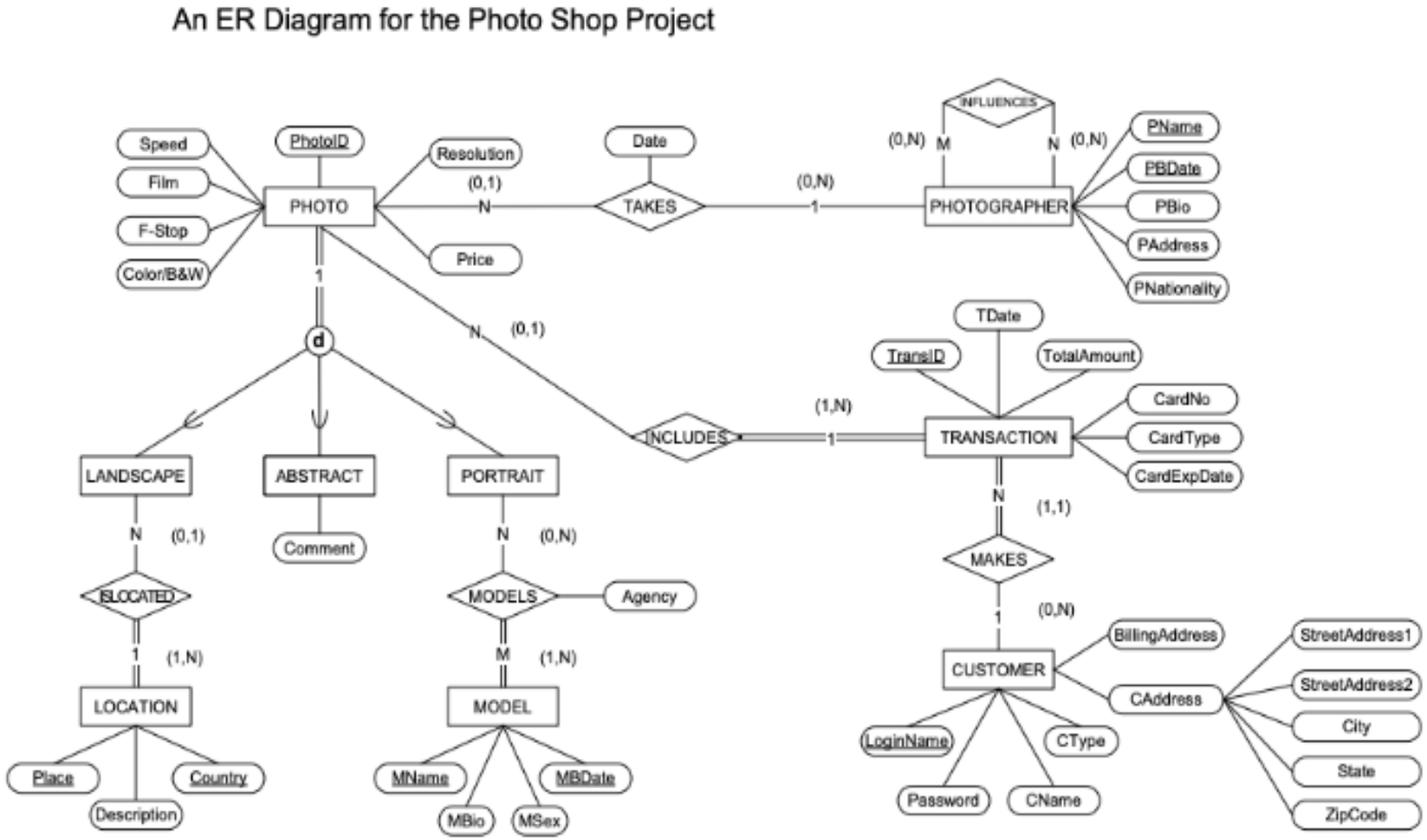
Practice 2


Salary
Shift
Model
Capacity
Weight
M
WORKS_ON
N
EMPLOYEE
N
MAINTAIN
PLANE_TYPE
M
Restr
Lic_num
1
M
N
FLIES
PILOT
OF_TYPE
Date
Workcode
N
Date/workcode
SERVICE
Hours
Reg#
N
1
AIRPLANE
PLANE_SERVICE
N
Pdate
STORED_IN
M
OWNS
N
OWNER
1
U
HANGAR
CORPORATION
Ssn
PERSON
Number
Location
Name
Phone
Name
Phone
Capacity
Address
Address

Practice 3


ArtistNumber
Type
Artist
Name
d
FormationDate
Group
M
Group-Member
N
SoloPerformer
BirthDate
JoinedDate
M
SoloPerformer-Instrument
CreatedDate
N
Instrument
InstrumentNumber
Name

Practice 4


An ER Diagram for the Photo Shop Project
Speed
Film
F-Stop
Color/B&W
PhotoID
PHOTO
Resolution
Price
(0,1)
N
Date
TAKES
(0,N)
1
INFLUENCES
(0,N)
M
N
(0,N)
PHOTOGRAPHER
PName
PBDate
PBio
PAddress
PNationality
1
d
N
(0,1)
LANDSCAPE
ABSTRACT
PORTRAIT
N
(0,1)
LOCATED
1
(1,N)
LOCATION
Place
Country
Description
Comment
N
(0,N)
MODELS
Agency
M
(1,N)
MODEL
MName
MBDate
MBio
MSex
INCLUDES
(1,N)
1
TDate
TransID
TotalAmount
CardNo
CardType
CardExpDate
TRANSACTION
N
(1,1)
MAKES
1
(0,N)
CUSTOMER
BillingAddress
CAddress
StreetAddress1
StreetAddress2
City
State
ZipCode
LoginName
CType
Password
CName

End of Chapter