DATABASE FUNDAMENTALS & DESIGN

Presented by

Josephine Boles

Outline

- What is a Relational Database?
- Basic Database Structure.
- Entity Relationship Modeling.
- ERD.
- Foreign Key
- Mapping

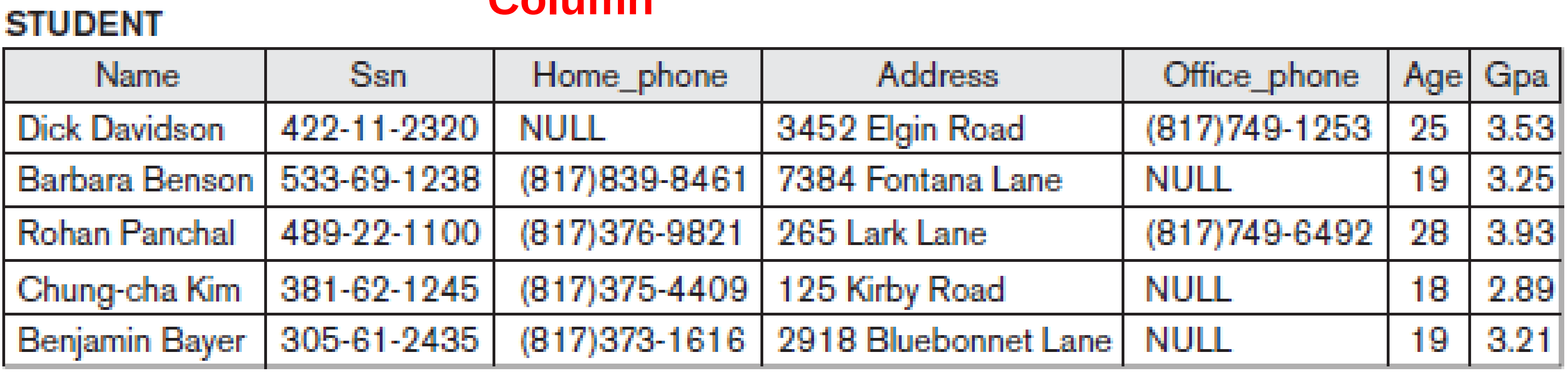
What is a Relational Database?

- A data structure through which data is stored in tables that are **related** to one another in some way.

- The way the tables are related is described through a **relationship**

Basic Database Structure

Column



STUDENT

Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	(817)749-1253	25	3.53
Barbara Benson	533-69-1238	(817)839-8461	7384 Fontana Lane	NULL	19	3.25
Rohan Panchal	489-22-1100	(817)376-9821	265 Lark Lane	(817)749-6492	28	3.93
Chung-cha Kim	381-62-1245	(817)375-4409	125 Kirby Road	NULL	18	2.89
Benjamin Bayer	305-61-2435	(817)373-1616	2918 Bluebonnet Lane	NULL	19	3.21

Row

Entity Relationship Modeling

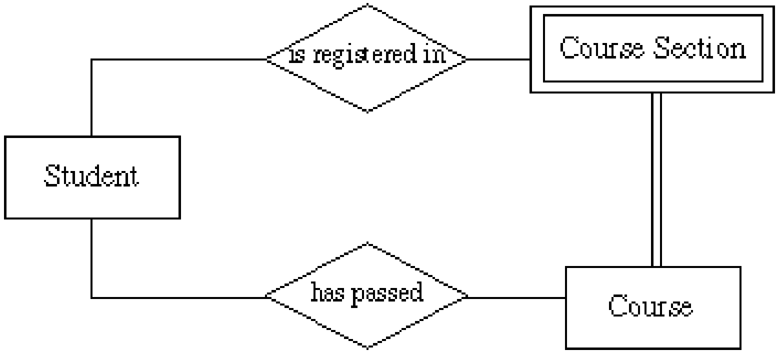
Entity-Relationship Diagram (ERD): identifies information required by the business by displaying the relevant entities and the relationships between them.

Definitions

- **Entity :**

  It is any thing about **which data is collected**

  (any thing a user want to track)



- **Weak Entity :**

  It is an entity whose existence is **dependent** on another entity.

Existence of course_section depend on Course

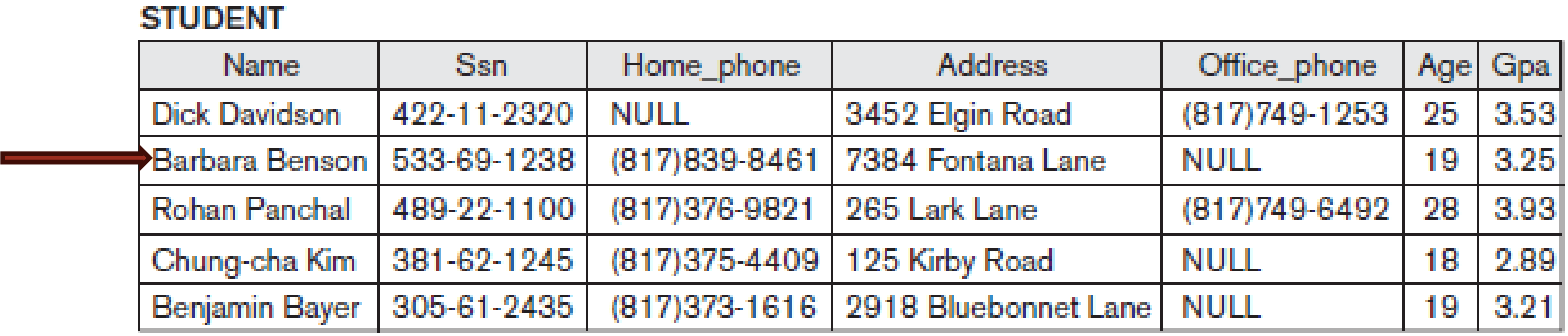
Definitions

- **Entity instance:** **An instance is a particular occurrence of an entity.**

For example:

each Student is an instance of an entity

each car is an instance of an entity, etc.



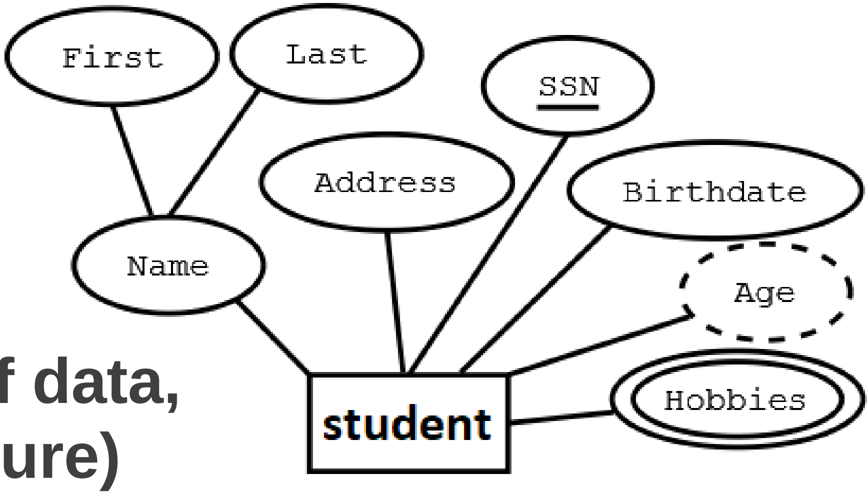
STUDENT

Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	(817)749-1253	25	3.53
Barbara Benson	533-69-1238	(817)839-8461	7384 Fontana Lane	NULL	19	3.25
Rohan Panchal	489-22-1100	(817)376-9821	265 Lark Lane	(817)749-6492	28	3.93
Chung-cha Kim	381-62-1245	(817)375-4409	125 Kirby Road	NULL	18	2.89
Benjamin Bayer	305-61-2435	(817)373-1616	2918 Bluebonnet Lane	NULL	19	3.21

Definitions (cont.)

Attributes :

They are the Characteristics of entities.



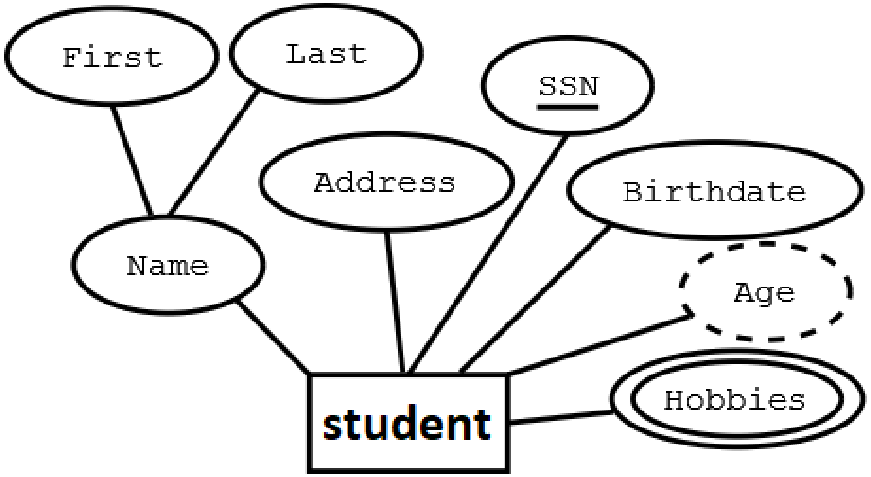
Types of attributes :

- **Simple (Scalars) –**
  - **smallest semantic unit of data,**
  - **atomic (no internal structure)**
  - **singular e.g. ssn,birthdate**

Definitions (cont.)

Types of attributes :

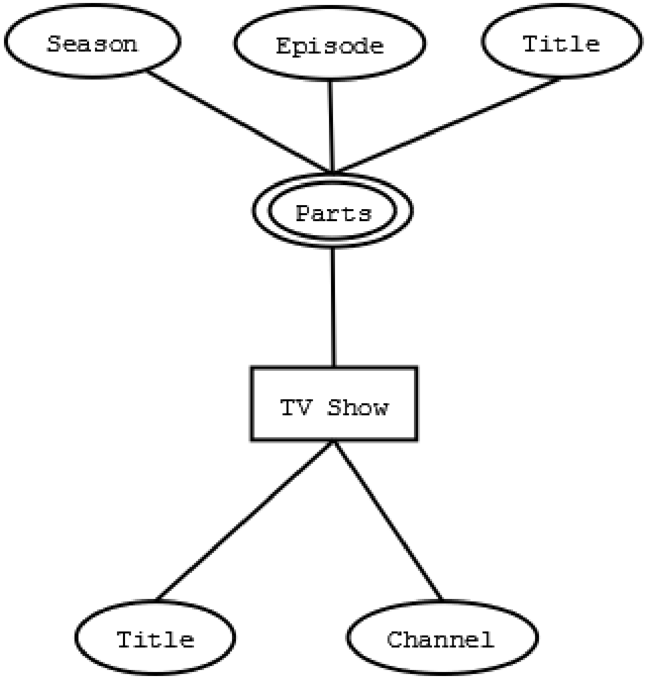
- Composite - group of attributes
  - e.g. address (street, city, state, zip)
- Multi-valued (list)
  - multiple values e.g. phone numbers.
- Derived.



Definitions (cont.)

Types of attributes :

- Complex attribute:
  - Those attributes, which can be formed by the nesting of composite and multi-valued attributes



Definitions (cont.)

- **Attribute Values**
    - ✓ Sometimes attribute values is set to null.
    - ✓ There are two meanings of null
        - either not applicable
        - unknown values.
    - ✓ Default Value.

Definitions (cont.)

- Primary Key:

  Identifier used to uniquely identify one particular instance of an entity.

  - ✓ Can be one or more attributes. (consider substituting a single concatenated key attribute for multiple attribute key).
  - ✓ Must be unique .
  - ✓ Value should not change over time.
  - ✓ Must always have a value .

Definitions (cont.)

- Candidate Key :

  when multiple possible identifiers exist, each is a candidate key.

- Foreign Keys :

  Foreign keys reference a related table through the primary key of that related table.

- Referential Integrity Constraint:

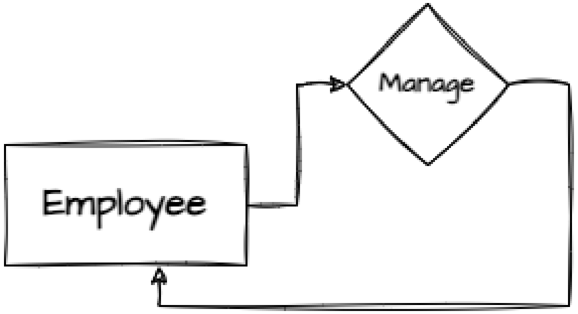
  For every value of a foreign key there is a primary key with that value in the referenced table e.g.

  if student name is to be used in a dormitory table then that name must exist in the student table.

Relationships

- A relationship is a connection between entity

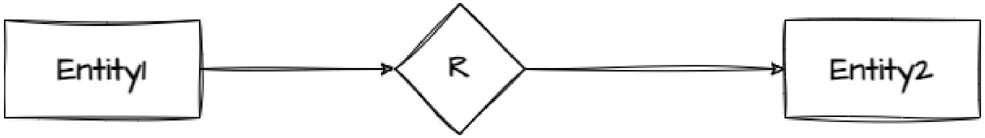
  classes.



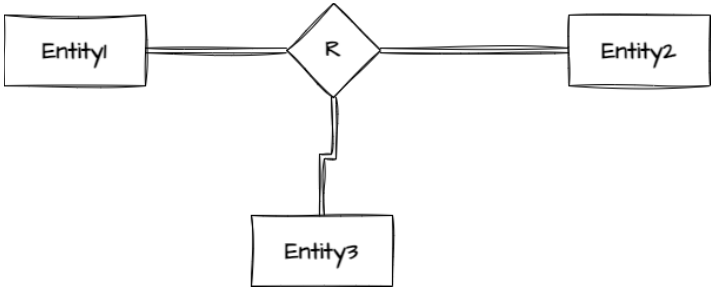
- Degree of Relationship:

  ✓ **Unary :**relationships exist when both entity types are the same and we call them the degree of relationship is 1

  ✓**Binary:**relationship exists when there are two types of entity and we call them a degree of relationship is 2



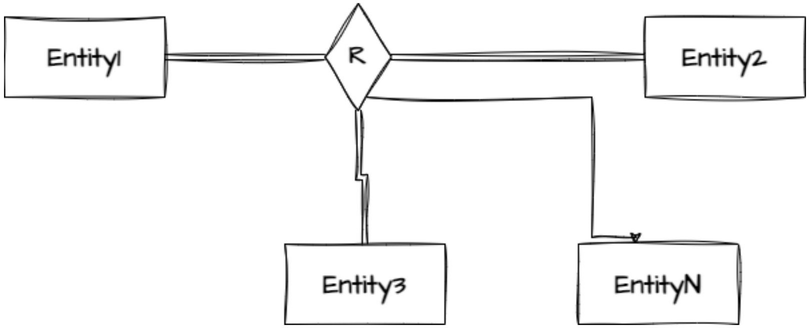
Relationships



- Degree of Relationship:

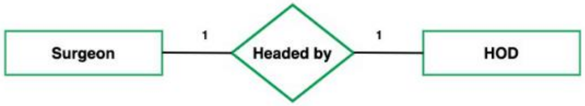
✓**Ternary:** relationship exists when there are three types of entity and we call them a degree of relationship is 3

✓**N-ary:** relationship exists when there are n types of entities. There is one limitation of the N-ary relationship

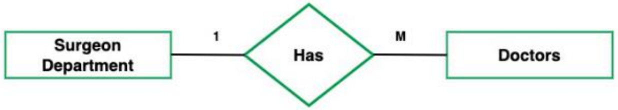


Relationships

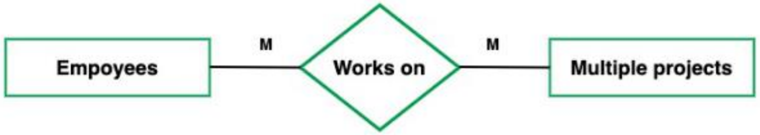
- Cardinality represent **maximum** number of relationships that can occur with these instance .



- Types of relationships (cardinality)
  - ✓ One-to-one relationship (1:1)



  - ✓ One-to-many relationship (1:M)
  - ✓ Many-to-many relationship
  - ✓ (N:M)

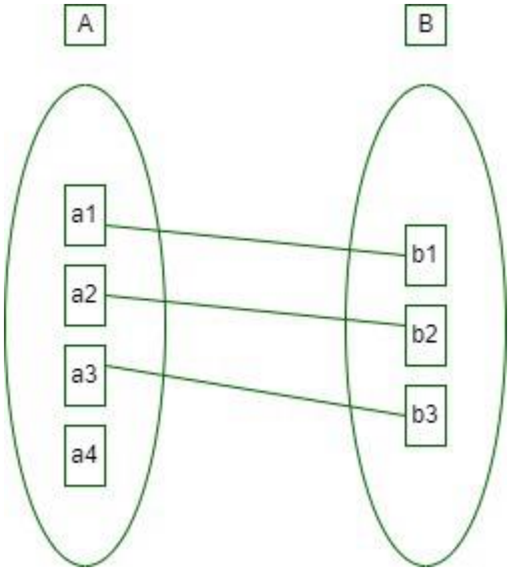


Relationships (cont.)

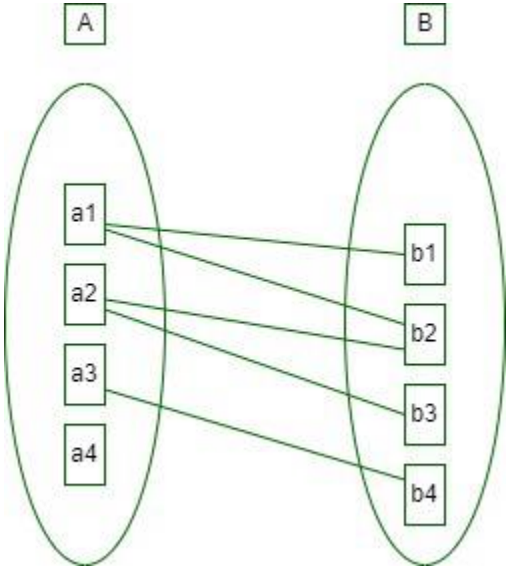
- **One-to-one relationship (1:1) :**

  A single record in table A is related to only one record in table B, and vice versa.

Ex. : Emp. Uses at most one car, a car is used at most by one emp.



Relationships (cont.)



- **One-to-many relationship (1:M) :**

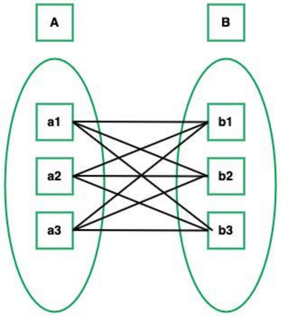
A single record in table (A) can be related to one or more records in table (B), but a single record in table (B) can be related to only one record in table (A).

Ex. : Emp. Uses at most one car, a car is used by many or several employees, student-advisor, customer-order

Relationships (cont.)

- **Many-to-many relationship (M:M) :**

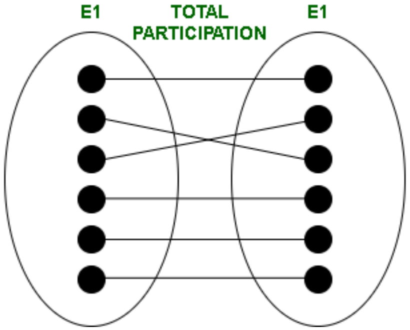
A single record in table A can be related to one or more records in table B, and vice versa.



Ex. An emp. Uses several cars, a car can be used by several employees. Student-Club, order-products.

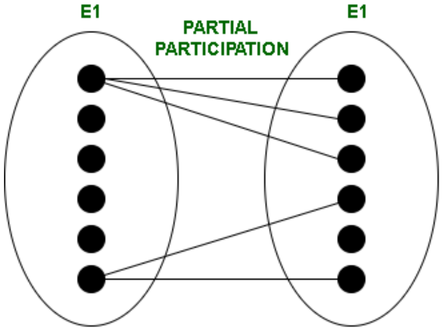
Relationships (cont.)

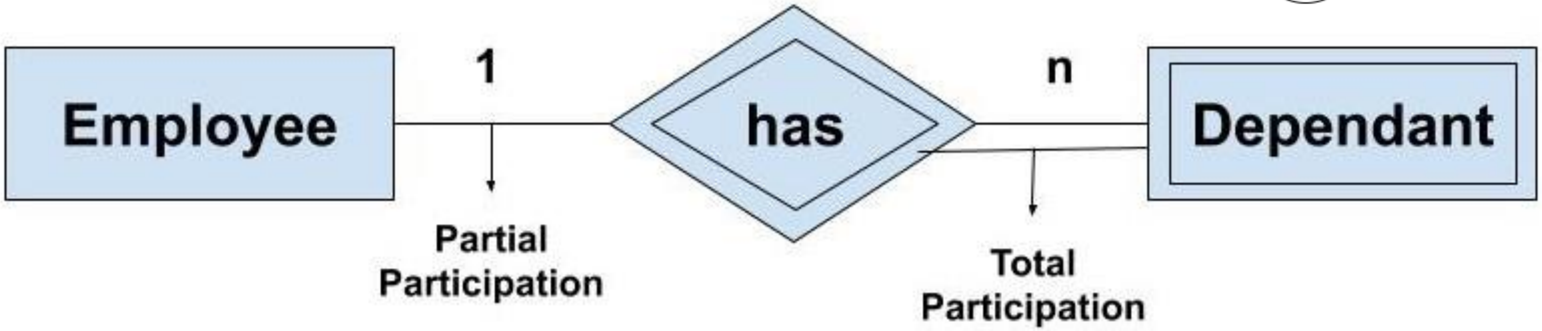
- **Participation Constraints (opposite of cardinality)**
  - Represent **minimum** number of relationships that can occur with these instance .
  - **Total Participation – Each Row is involved in the relationship. Total participation is represented by double lines.**



Relationships (cont.)

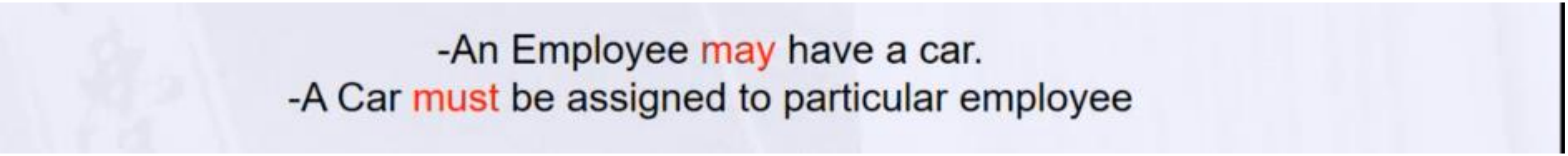
- **Participation Constraints**
  - **Partial participation** − Not all Rows are involved in the relationship. Partial participation is represented by **single lines.**



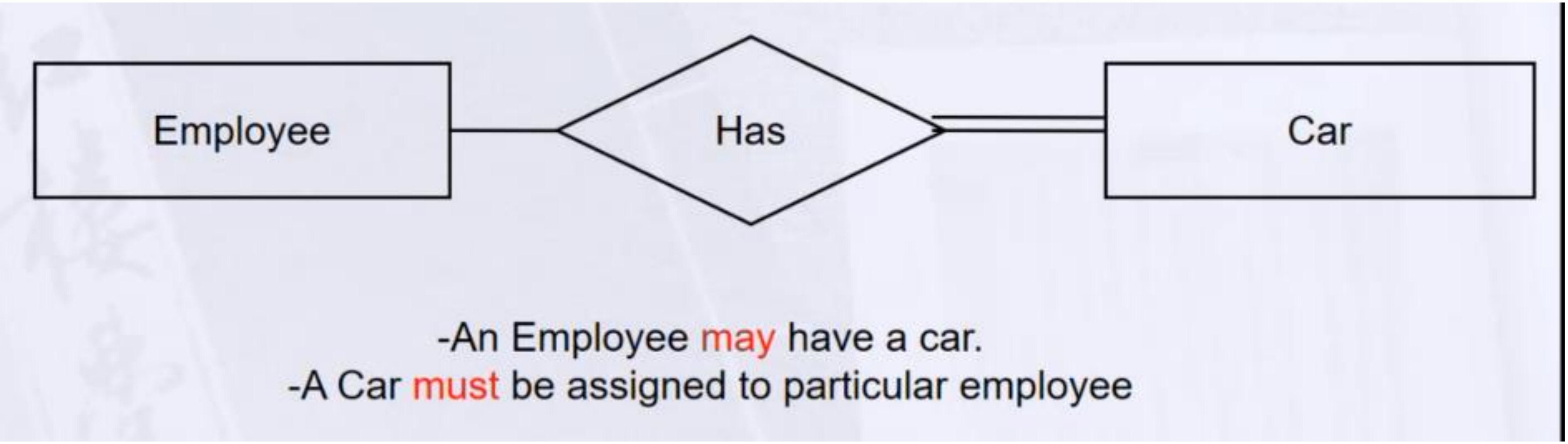


Any weak entity must be totally Participation

Relationships (cont.)



-An Employee may have a car.
-A Car must be assigned to particular employee

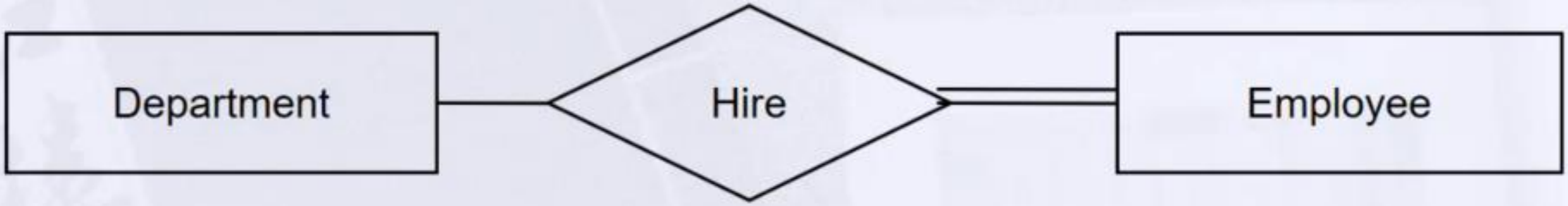


-An Employee may have a car.
-A Car must be assigned to particular employee

Relationships (cont.)

- A department may hire many employees (Zero or more)
- An employee must be employed by a department

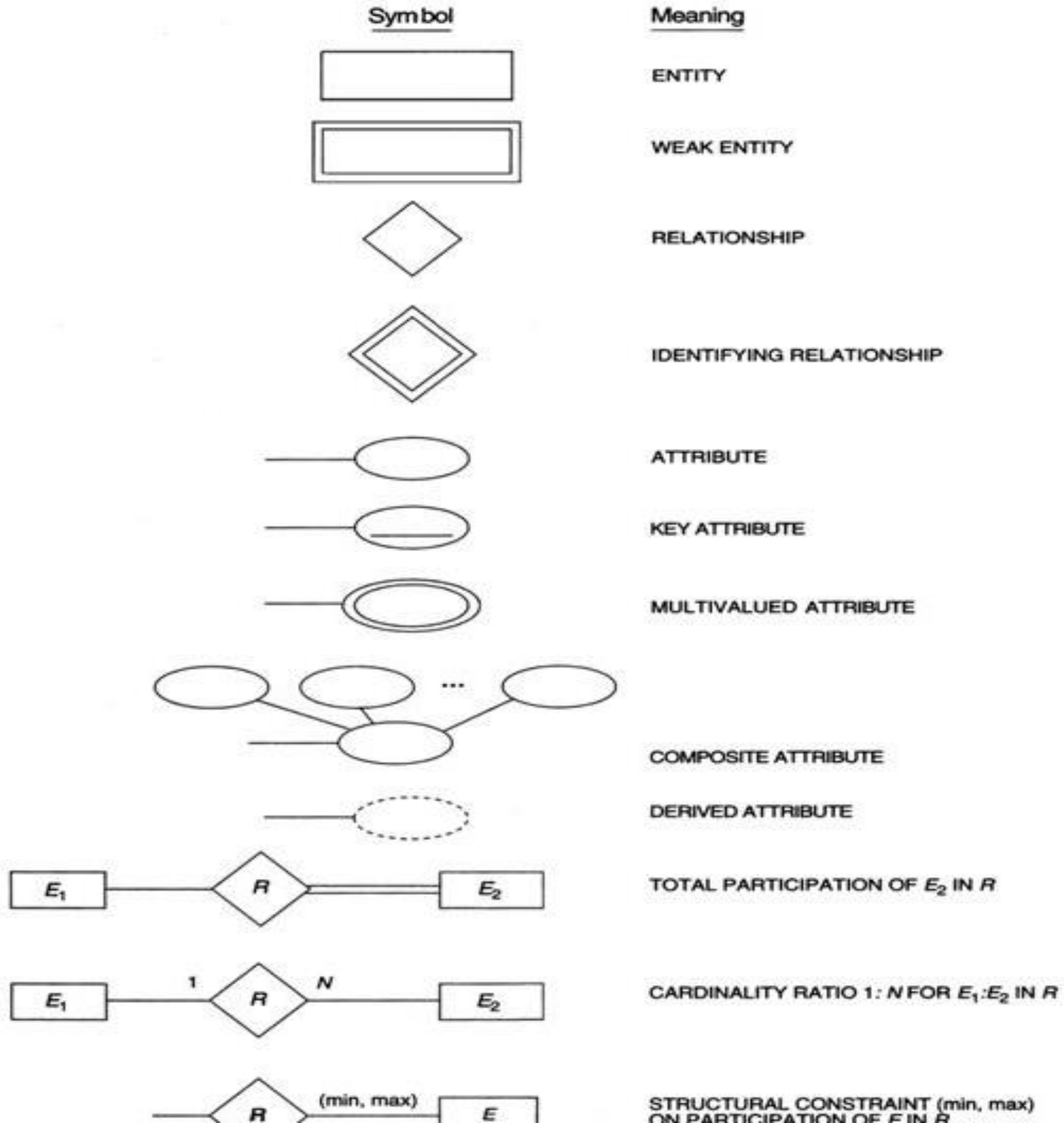
(Department membership is Optional, Employee membership is Mandatory)



- A department may hire many employees (Zero or more)
- An employee must be employed by a department

(Department membership is Optional, Employee membership is Mandatory)

Summary of the Notation for ER Diagrams



ENTITY

WEAK ENTITY

RELATIONSHIP

IDENTIFYING RELATIONSHIP

ATTRIBUTE

KEY ATTRIBUTE

MULTIVALUED ATTRIBUTE

COMPOSITE ATTRIBUTE

DERIVED ATTRIBUTE

TOTAL PARTICIPATION OF E_2 IN R

CARDINALITY RATIO 1: N FOR E_1:E_2 IN R

STRUCTURAL CONSTRAINT (min, max) ON PARTICIPATION OF E IN R

Entity Relationship Modeling

- In building a data model a number of questions must be addressed:

1- What entities need to be described in the model?

2- What characteristics or attributes of those entities need to be recorded?

3- Can an attribute or a set of attributes be identified that will uniquely identify one specific occurrence of an entity?

4- What associations or relationships exist between entities?

An Example

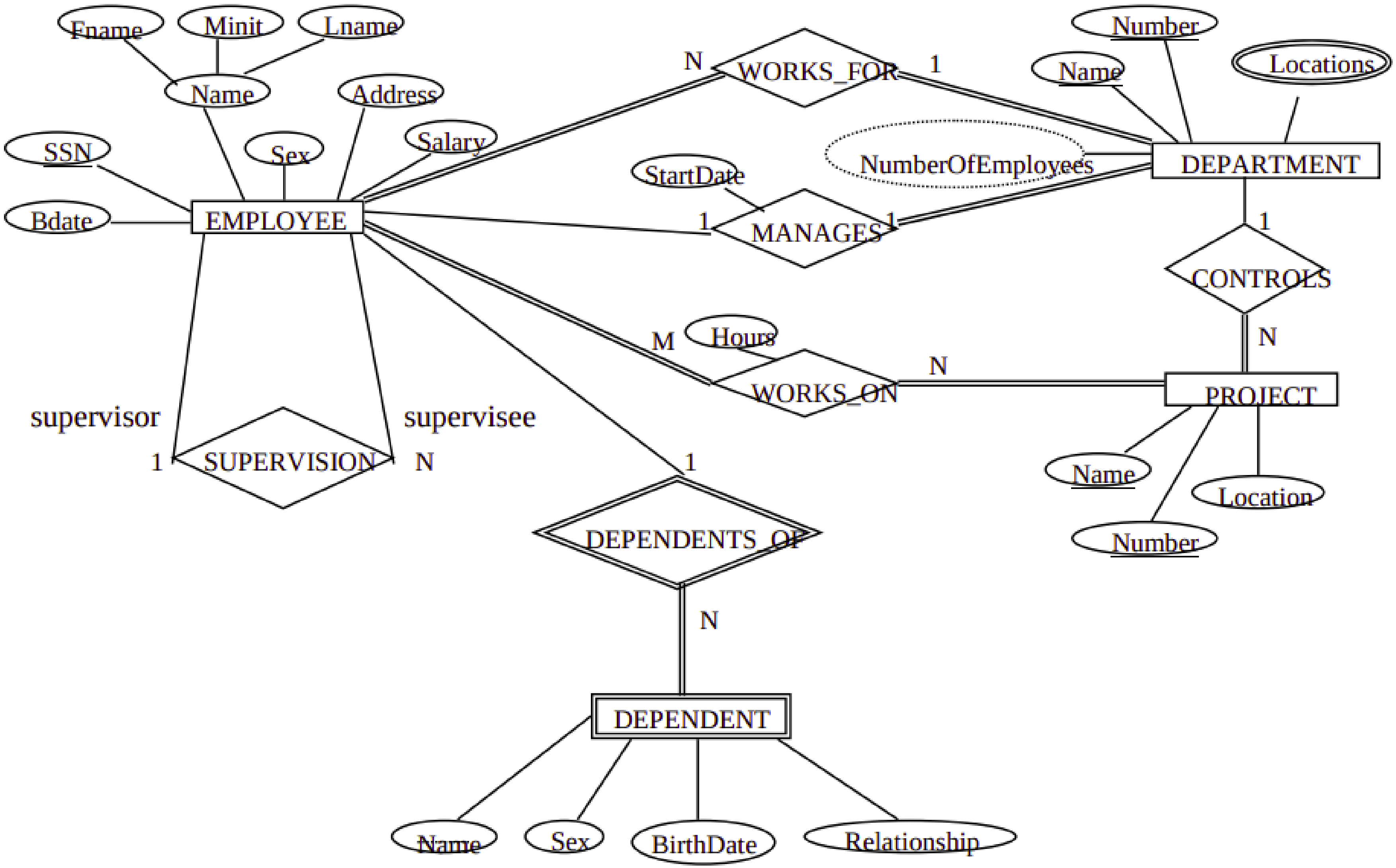
- A company is organized into departments. Each department has a unique name, a unique number, and a particular employee who manages the department. A department may have several locations.

- A department may control a number of projects, each of which has a unique name, a unique number, and a single location. A project must controlled by department

An Example (Cont’d)

- We store employee’s name, social security number, address, salary, gender and birth date. An employee must be assigned to one department and must work on one or more projects, which are not necessarily controlled by the same department. We keep track of the number of hours per week that an employee works on each project. We also keep track of the direct supervisor of each employee.

- We want to keep track of the dependents of each employee for insurance purposes. We keep each dependent’s first name, gender, birth date and relationship to that employee.


Fname
Minit
Lname
Name
Address
SSN
Sex
Salary
Bdate
EMPLOYEE
N
WORKS_FOR
1
Number
Name
Locations
NumberOfEmployees
DEPARTMENT
StartDate
1
MANAGES
1
1
CONTROLS
N
M
Hours
WORKS_ON
N
PROJECT
Name
Location
Number
supervisor
supervisee
1
SUPERVISION
N
1
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
BirthDate
Relationship

Definitions (cont.)

- **Foreign Keys :**
  - is a column that refers to the primary key/unique key of other table. So it demonstrates relationship between tables and act as cross reference among them
  - Fk in A must be same type of pk in B

- **Referential Integrity Constraint:**

For every value of a foreign key there is a primary key with that value in the referenced table

Each value of fk in A must match value in pk in B

e.g. if student name is to be used in a dormitory table then that name must exist in the student table.

ER-to-Relational Mapping

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 Multi-valued attributes.

Step 7: Mapping of N-ary Relationship Types.

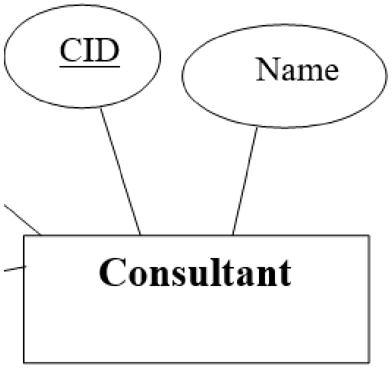
Step 1: Mapping of Regular Entity Types

Create table for each entity type

Choose one of key attributes to be the primary key

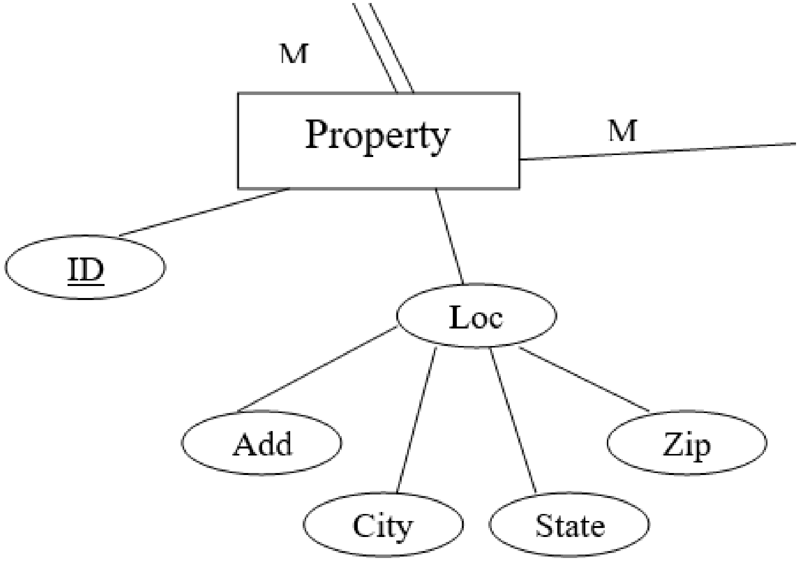
Consultant

CID	Name



Step 1: Mapping of Regular Entity Types

Regular entity with composite attribute



Property

ID	Add	City	State	Zip

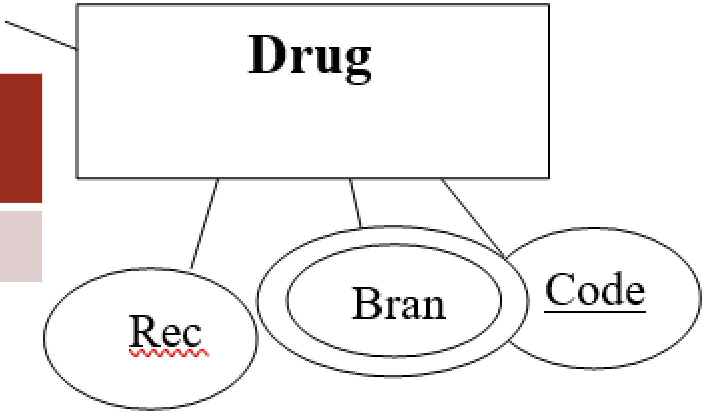
Leave note about Loc

Step 6: Mapping of Multi-valued attributes.

Create new table for each multi-valued attribute
Table will include two columns.
one for multi-valued attribute + FK column.

Drug

Code	Rec	City	State	Zip



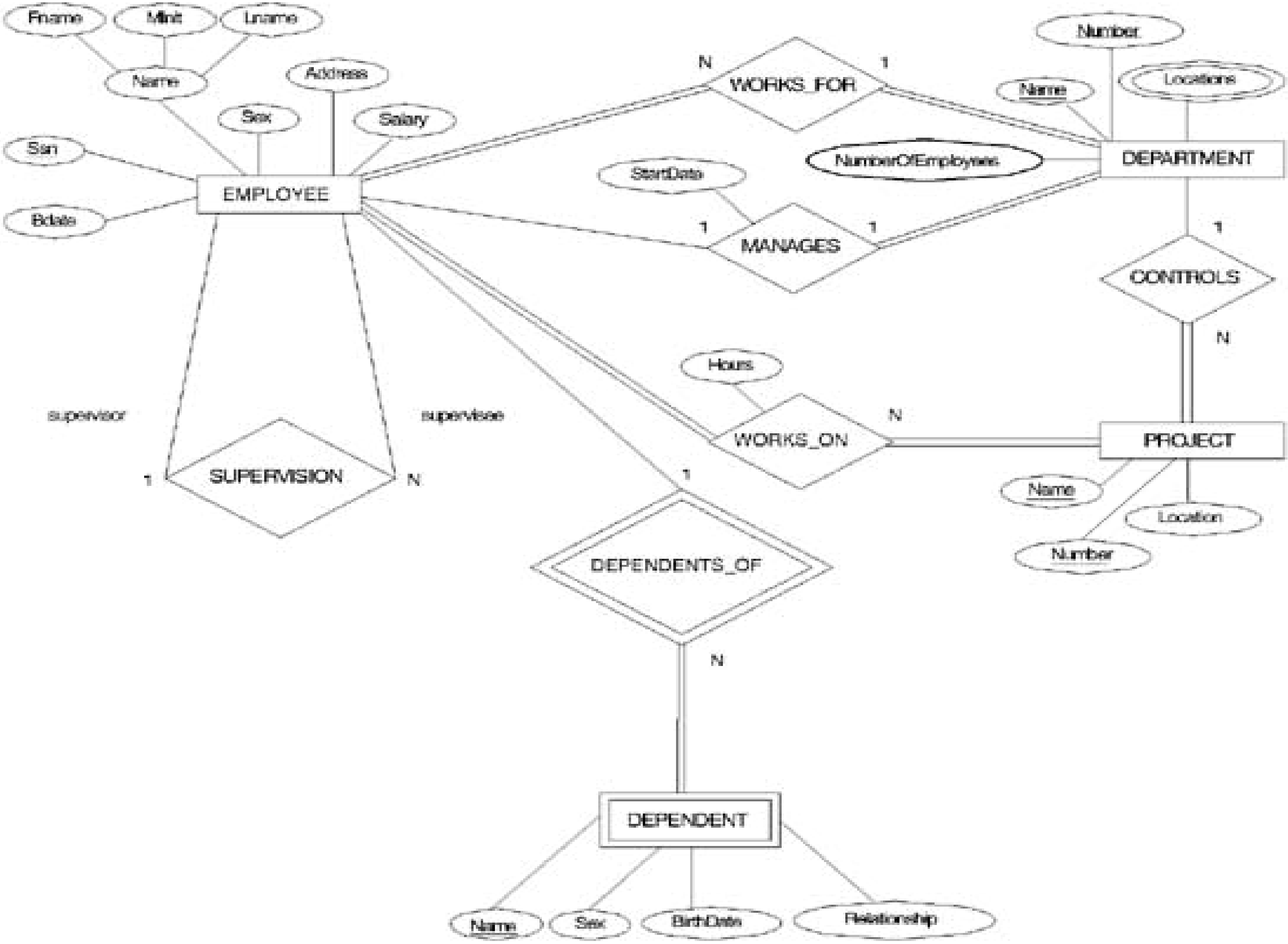
Drug_Brand

Code - - - -	Brand
Composite pk	

Step 2: Mapping of Weak Entity Types

Create table for each weak entity.
Add foreign key that correspond to the owner entity type.
Choose the primary key : (FK + weak entity Partial PK if any)

Step 2: Mapping of Weak Entity Types



Step 2: Mapping of Weak Entity Types

Employee

SSN	Bdate	Fname	Lname	mname	Gender	Salary	address

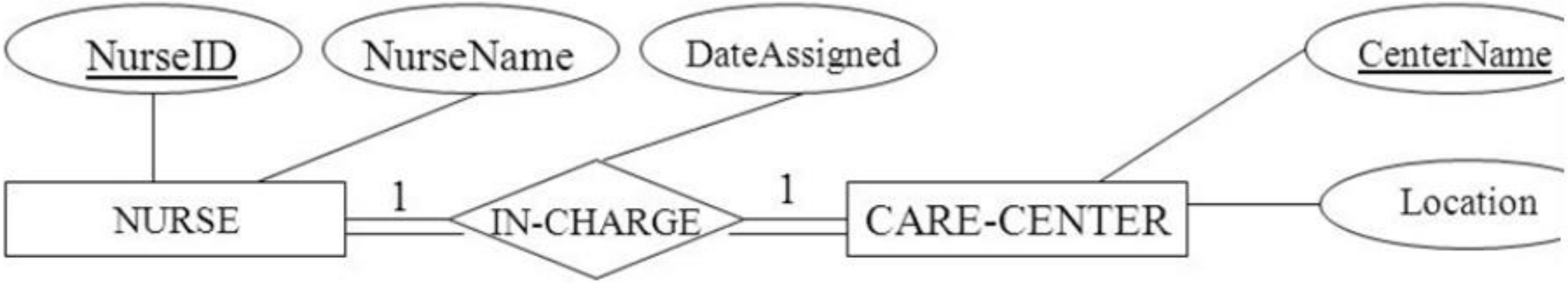
Dependent

Name	SSN ---------	Gender	Bdate	Relation
Composite pk				

Step 3: Mapping of Binary 1:1 Relation Types

Merged two tables if both sides are Mandatory.

- one-to-one relationship:



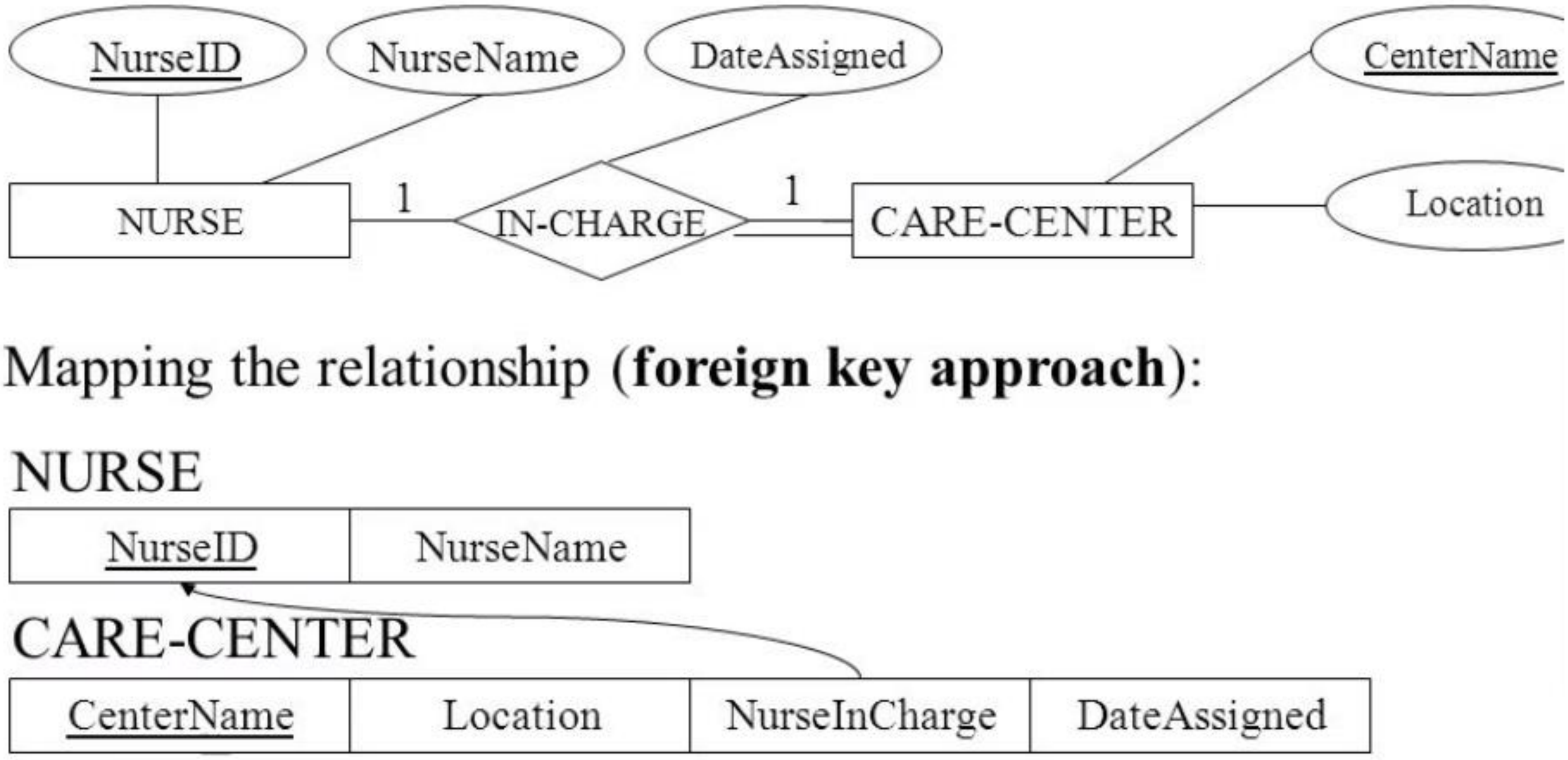
- Mapping the relationship (**Merged relation option**):

NURSE

<u>NurseID</u>	NurseName	CenterName	Location	DateAssigned

Step 3: Mapping of Binary 1:1 Relation cont.

Add FK into table with the total participation relationship to represent optional side.



- Mapping the relationship (**foreign key approach**):

NURSE

NurseID	NurseName

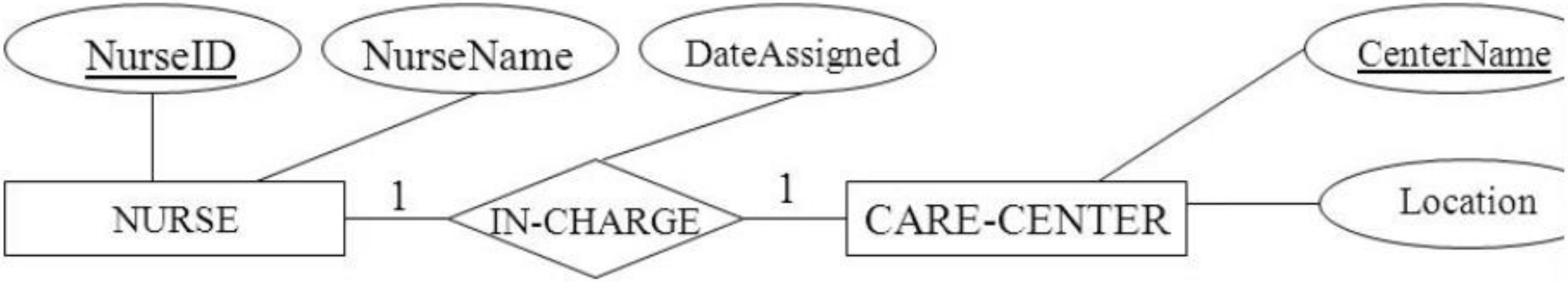
CARE-CENTER

CenterName	Location	NurseInCharge	DateAssigned

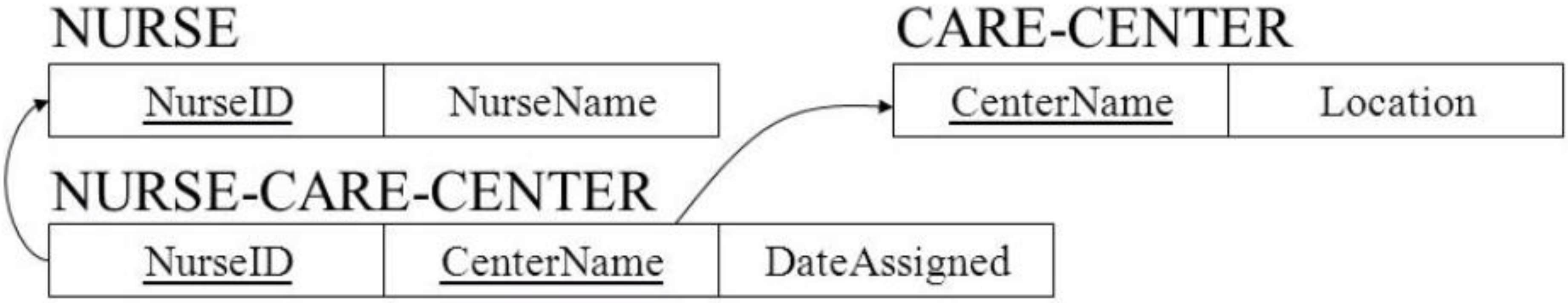
Step 3: Mapping of Binary 1:1 Relation cont.

Create third table if both sides are optional.

- one-to-one relationship:



- Mapping the relationship (**Cross-reference option**):


NURSE

NurseID	NurseName

CARE-CENTER

CenterName	Location

NURSE-CARE-CENTER

NurseID	CenterName	DateAssigned

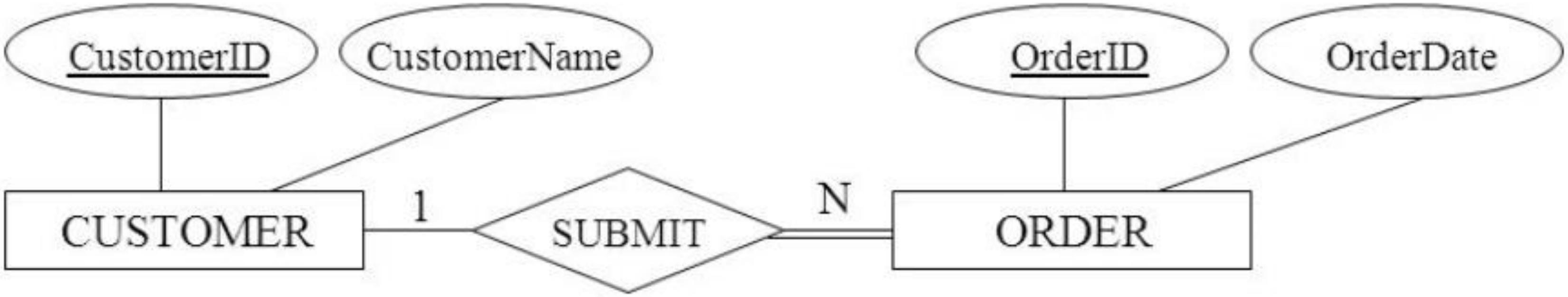
Step 4: Mapping of Binary 1:N Relationship Types.

Add FK to N-side table

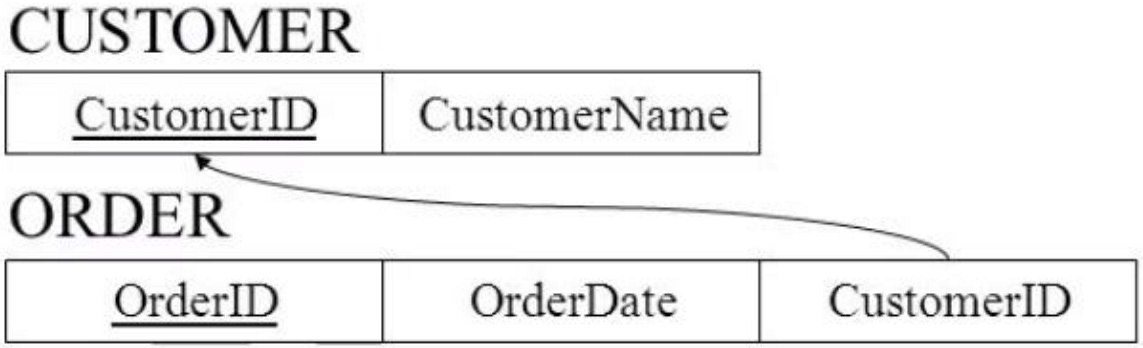
Add any simple attributes of relationship as column to N-side table.

Step 4: Mapping of Binary 1:N Relationship Types.

- One-to-many relationship:



- Mapping the relationship (**foreign key approach**):



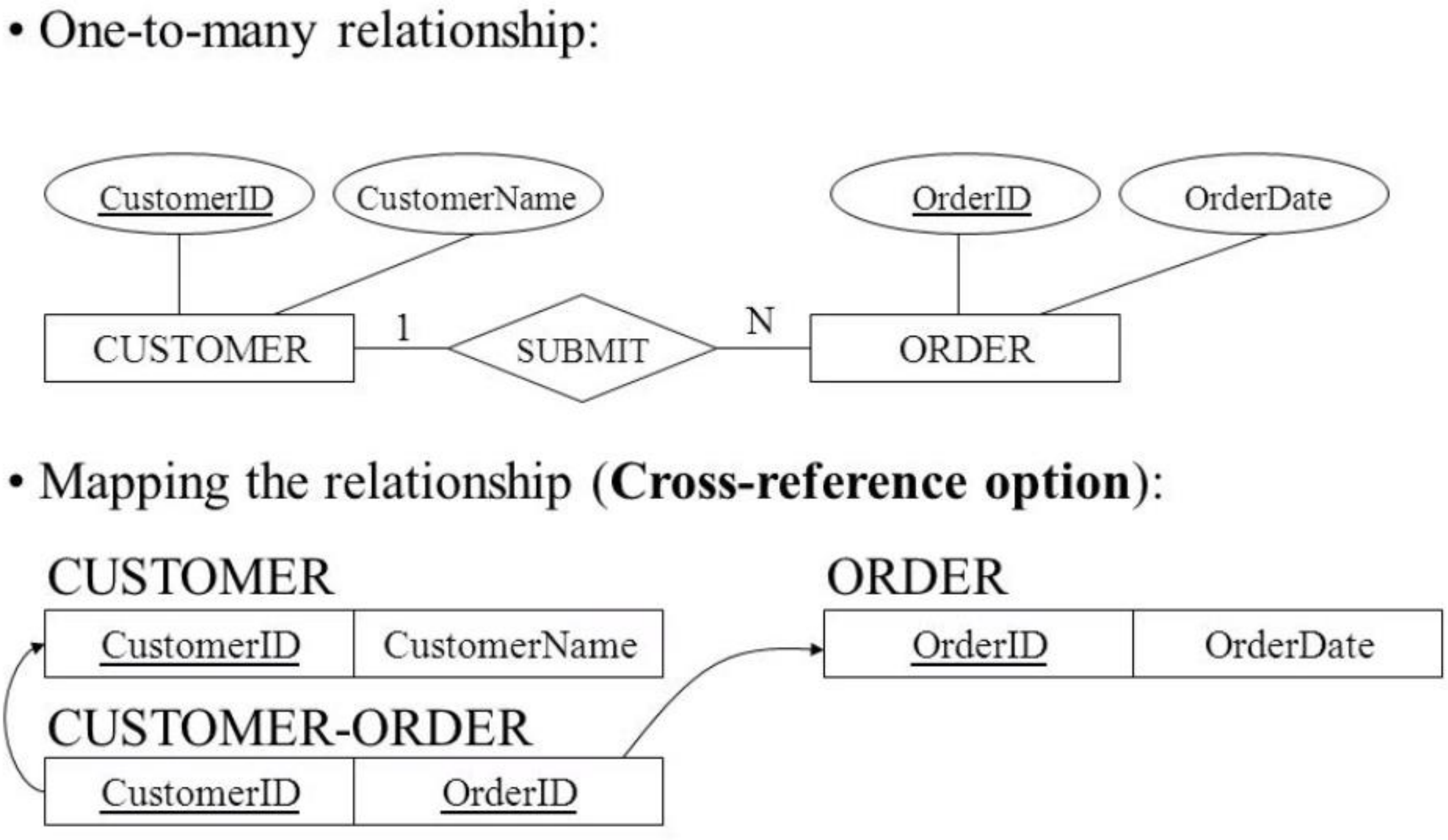
CUSTOMER

CustomerID	CustomerName

ORDER

OrderID	OrderDate	CustomerID

Step 4: Mapping of Binary 1:N Relationship Types.



- One-to-many relationship:

- Mapping the relationship (**Cross-reference option**):

CUSTOMER

CustomerID	CustomerName

ORDER

OrderID	OrderDate

CUSTOMER-ORDER

CustomerID	OrderID

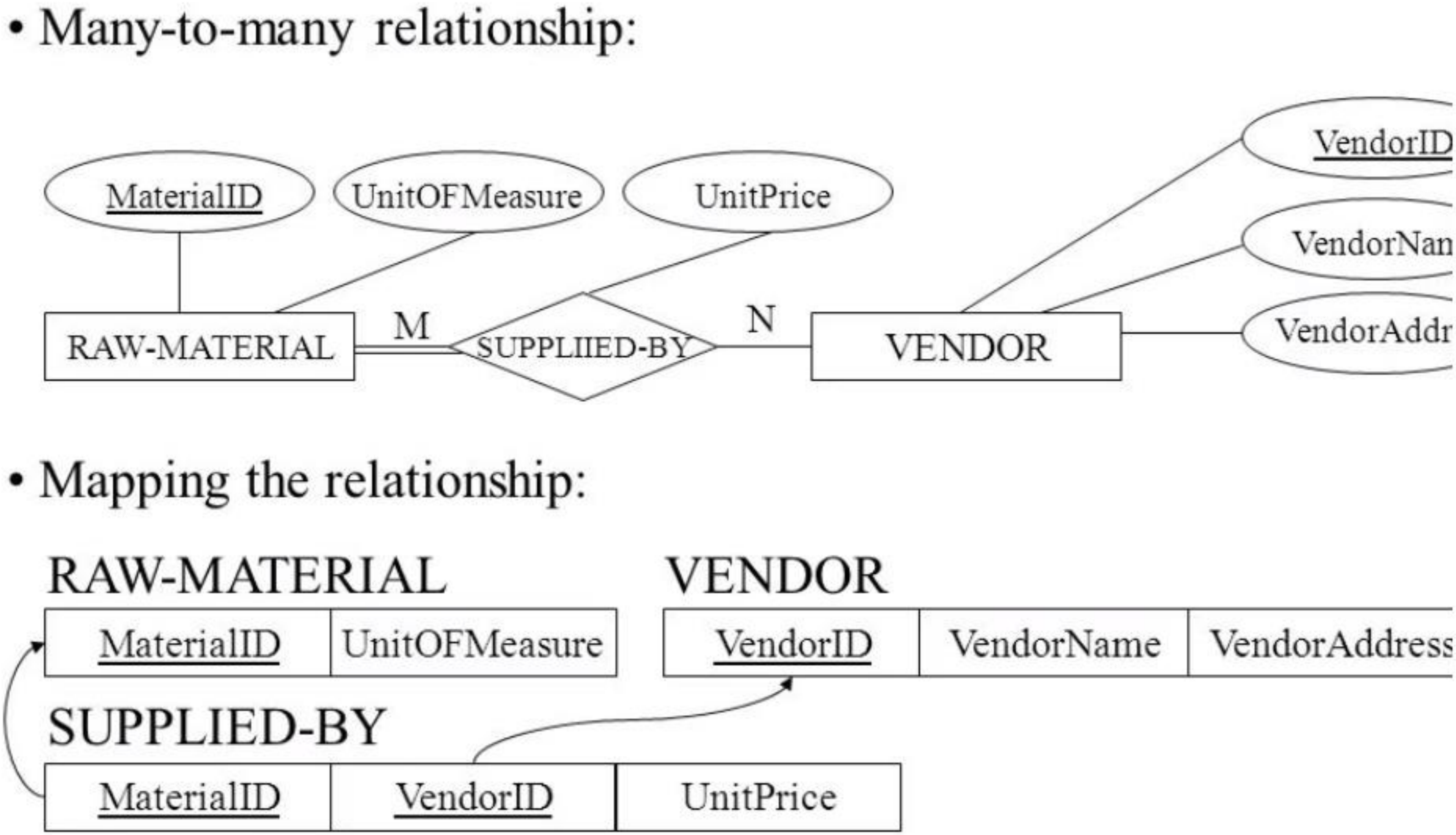
Step5: Mapping of Binary M:N Relationship

Create a new third table

Add FKs to the new table for both parent tables

Add simple attributes of relationship to the new table if any .

Step5: Mapping of Binary M:N Relationship



- Many-to-many relationship:

- Mapping the relationship:

RAW-MATERIAL

MaterialID	UnitOFMeasure

VENDOR

VendorID	VendorName	VendorAddress

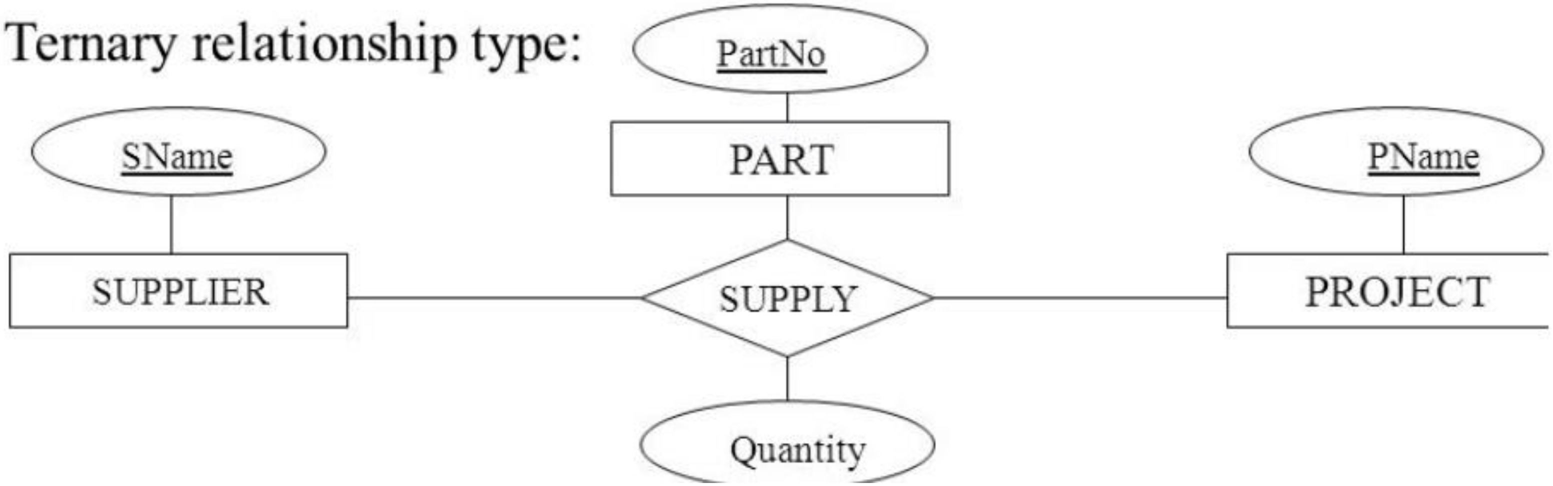
SUPPLIED-BY

MaterialID	VendorID	UnitPrice

Step7: Mapping of N-ary Relationship

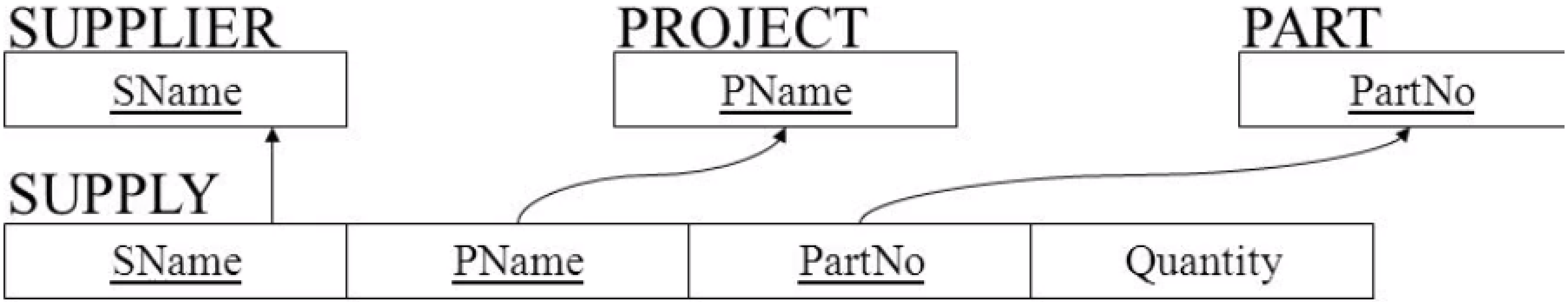
If n > 2 then :
Create a new third table
Add FKs to the new table for all parent tables
Add simple attributes of relationship to the new table if any .

- Ternary relationship type:



Step7: Mapping of N-ary Relationship

- Mapping the n-ary relationship type SUPPLY:



SUPPLIER

SName

PROJECT

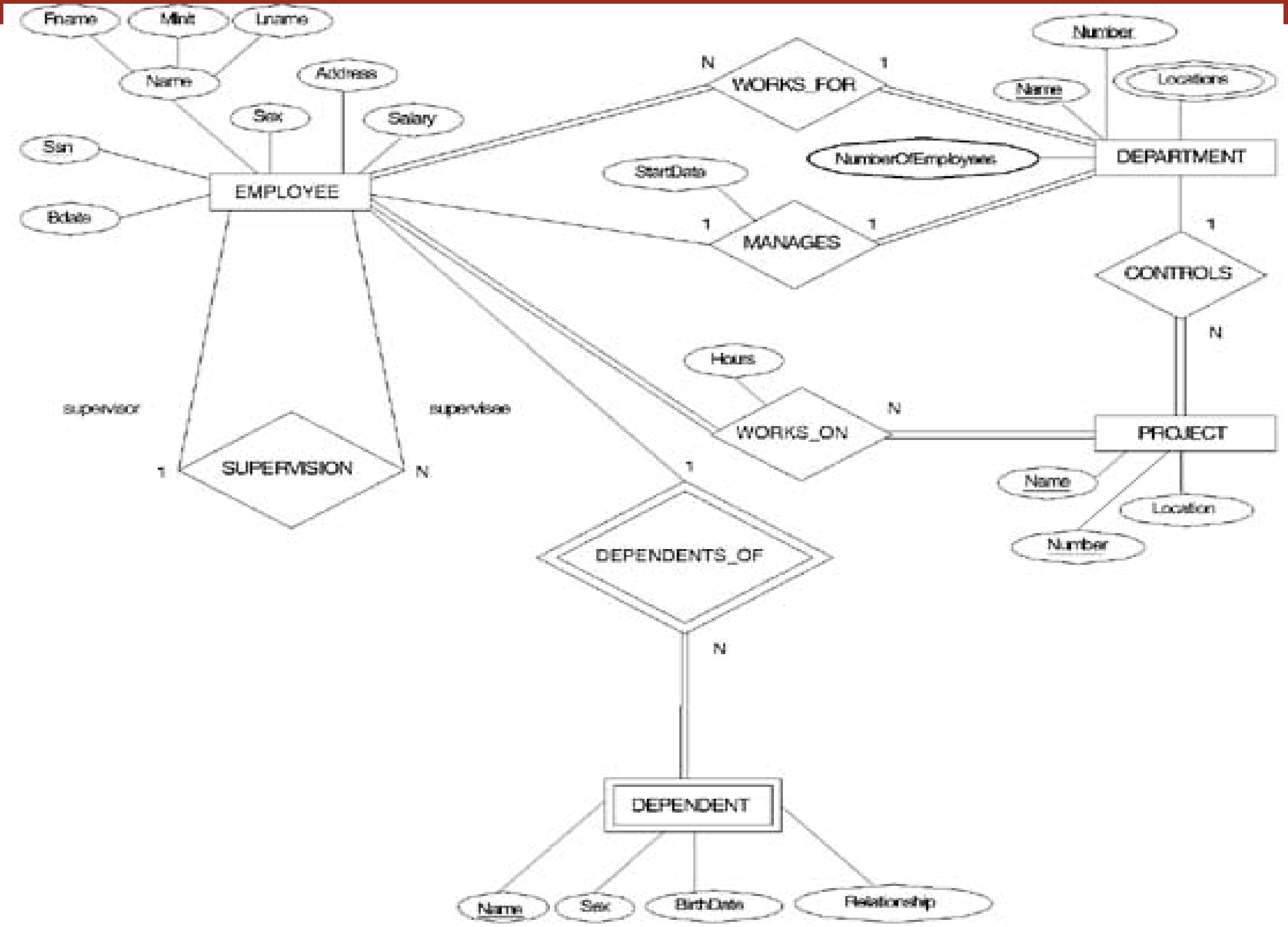
PName

PART

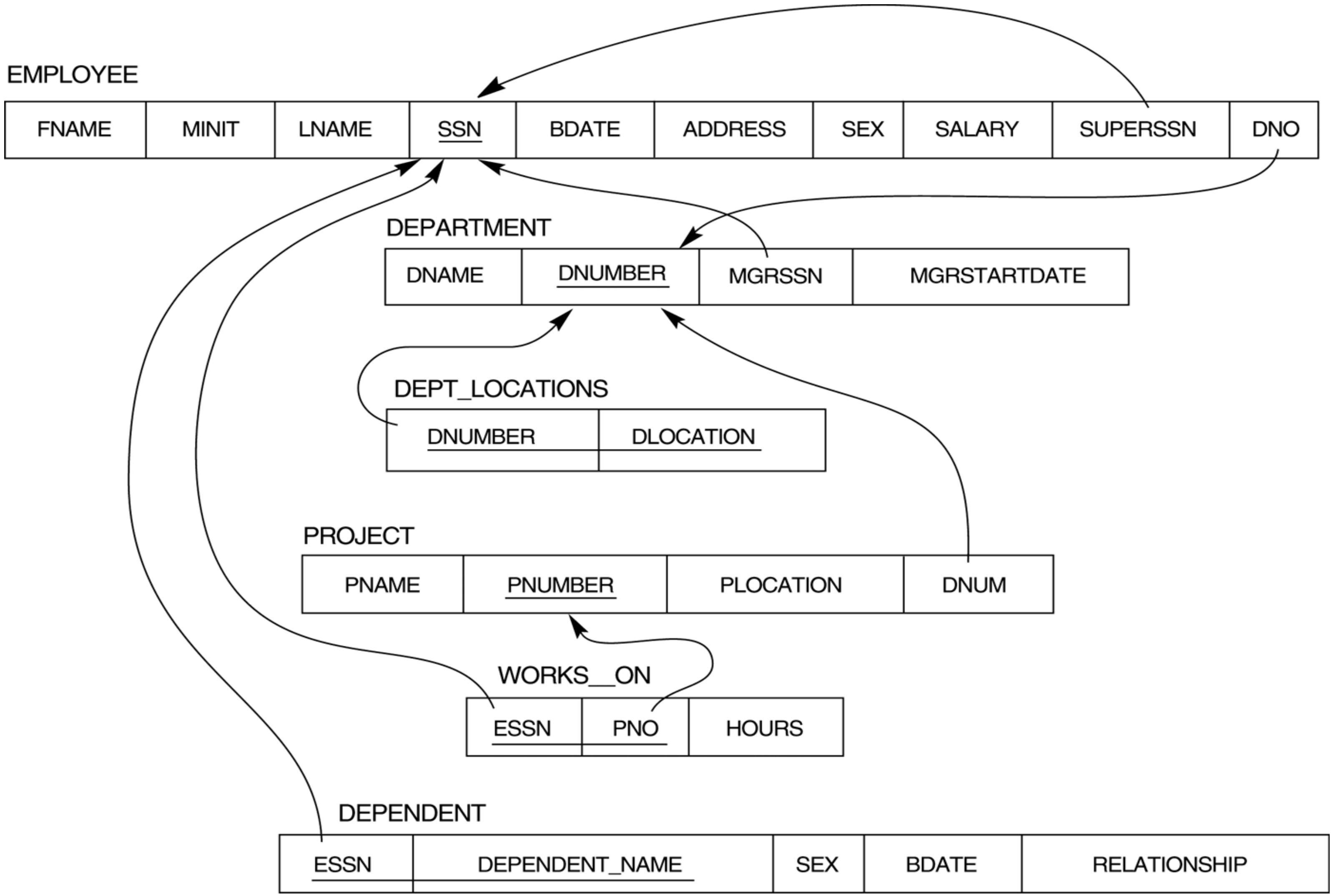
PartNo

SUPPLY

SName	PName	PartNo	Quantity


Fname
Minit
Lname
Name
Address
Sex
Salary
Ssn
Bdate
EMPLOYEE
N
WORKS_FOR
1
Number
Name
Locations
DEPARTMENT
NumberOfEmployees
StartDate
1
MANAGES
1
1
CONTROLS
N
supervisor
supervisee
Hours
WORKS_ON
N
PROJECT
1
SUPERVISION
N
1
DEPENDENTS_OF
N
Name
Location
Number
DEPENDENT
Name
Sex
BirthDate
Relationship

Mapping Result



EMPLOYEE

FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO

DEPARTMENT

DNAME	DNUMBER	MGRSSN	MGRSTARTDATE

DEPT_LOCATIONS

DNUMBER	DLOCATION

PROJECT

PNAME	PNUMBER	PLOCATION	DNUM

WORKS__ON

ESSN	PNO	HOURS

DEPENDENT

ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP

Self Study

- IS A Relation

Questions ?

Exercises