



# Chapter 3

## Relational Model

Department: Computer  
Course: DBMS  
Faculty: Sana Shaikh

## Quick Recap:

- EER Model Concepts (Specialization, Generalization, Aggregation)
- Introduction to the Relational Model
- Relational Model Concepts
- Relational schema
- Concept of Relational keys

## **Topics to be covered:**

- Mapping the ER and EER Model to the Relational Model
- Solving Examples

## **Learning Outcomes:**

Students should be able to:

- Describe the rules for Mapping ER and EER Model to the Relational Model
- Convert ER and EER Model to the Relational Model for any given problem

# Relational Model Concepts **Quick Recap:**

## Relation:

- We shall represent a relation as a table with columns and rows.
- Each column of the table has a name, or attribute.
- Each row is called a tuple.

**Domain:** a set of atomic values that an attribute can take

## Attribute:

- Name of a column in a particular table (all data is stored in tables).
- Each attribute  $A_i$  must have a domain,  $\text{dom}(A_i)$ .

## Relational Schema:

The design of one table, containing the name of the table (i.e. the name of the relation), and the names of all the columns, or attributes. Example: STUDENT( Name, SID, Age, GPA)

**Degree of a Relation:** the number of attributes in the relation's schema.

**Tuple:**  $t$ , of  $R(A_1, A_2, A_3, \dots, A_n)$ : an ORDERED set of values,  $\langle v_1, v_2, v_3, \dots, v_n \rangle$ , where each  $v_i$  is a value from  $\text{dom}(A_i)$ .

**Cardinality of relation:** the number of tuples in the relation

**Relation Instance:**  $r(R)$ : a set of tuples; thus,  $r(R) = \{t_1, t_2, t_3, \dots, t_m\}$

# RDBMS Terminologies

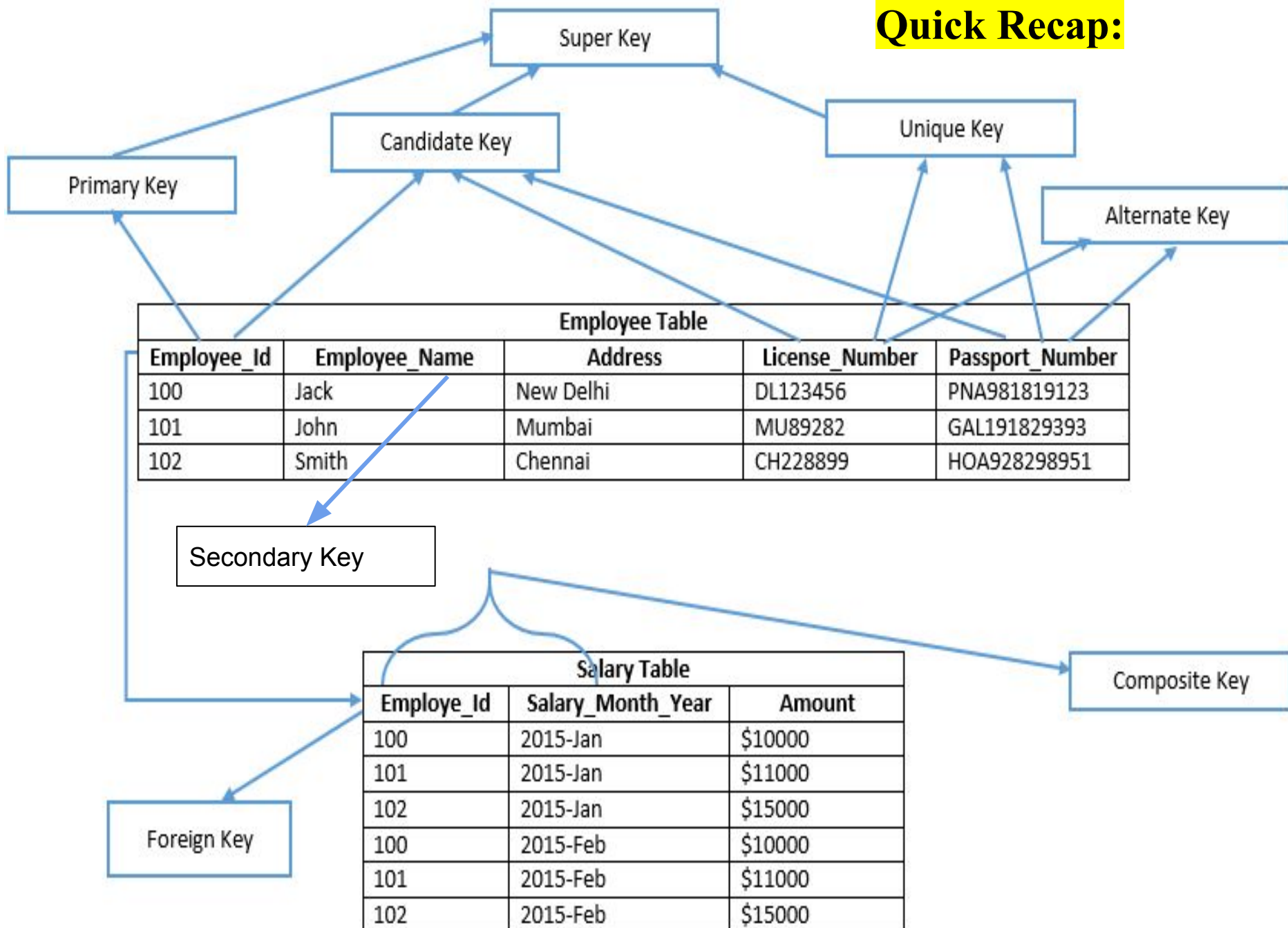
**Quick Recap:**

Informal Terms	Formal Relational Terms
Table	Relation
Row / Record	Tuple
No. of rows	Cardinality
Column / Field	Attribute
No. of Columns	Degree
Unique Identifier	Primary Key
Set of Legal Values	Domain

# Summary

- Super key:- Set of an attribute which can uniquely identify a tuple
- Primary key :- The attribute or combination of attributes that uniquely identifies a row or record.
- Unique key:- ensures that all values in a column are different.
- Foreign Key:- an attribute or combination of attribute in a table whose value match a primary key in another table.
- Composite key:- A primary key that consists of two or more attributes is known as composite key.
- Candidate key:- is a column in a table which has the ability to become a primary key.
- Alternate Key:- Any of the candidate keys that is not part of the primary key is called an alternate key.
- Secondary key:- alternate of primary key.

## Quick Recap:



# **Mapping the ER and EER Model to the Relational Model**

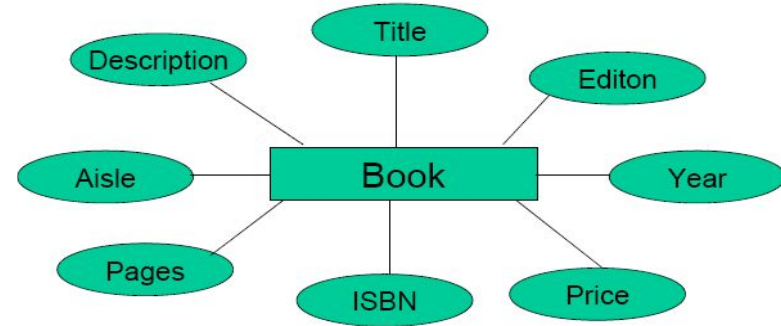


# Mapping Entity

- **An entity is a real-world object with some attributes.**

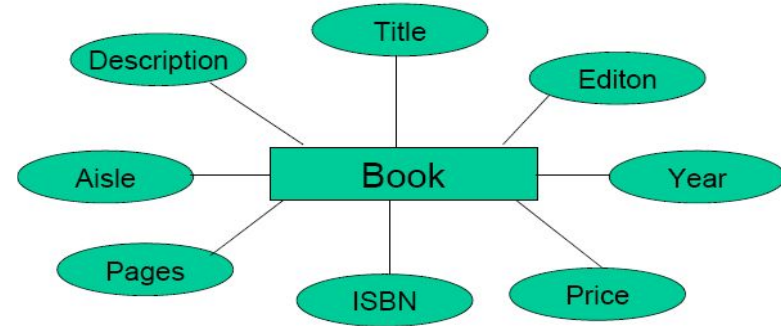
# Mapping Entity

- **An entity is a real-world object with some attributes.**



# Mapping Entity

- An entity is a real-world object with some attributes.



- **Mapping Process**

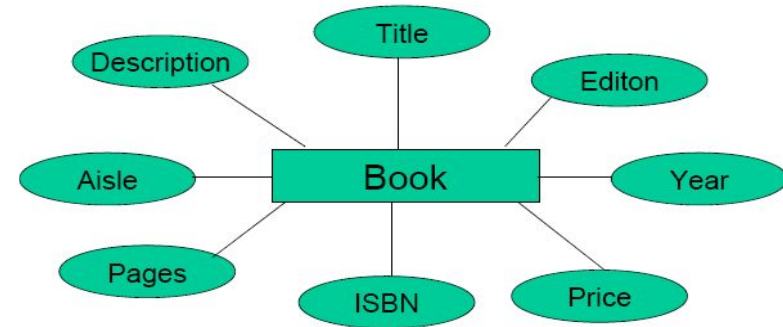
1. Create table for each entity.
2. Entity's attributes should become fields of tables with their respective data types.
3. Declare primary key.

# Mapping Entity

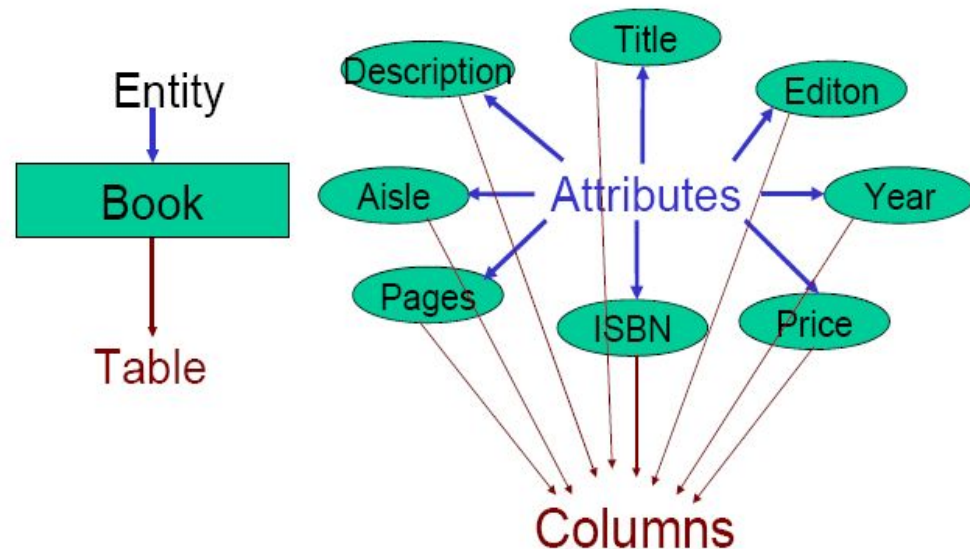
- An entity is a real-world object with some attributes.

- **Mapping Process**

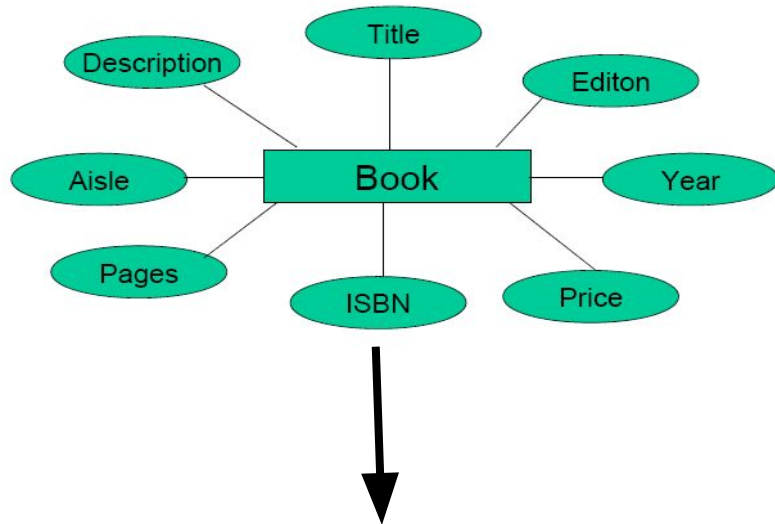
1. Create table for each entity.
2. Entity's attributes should become fields of tables with their respective data types.
3. Declare primary key.



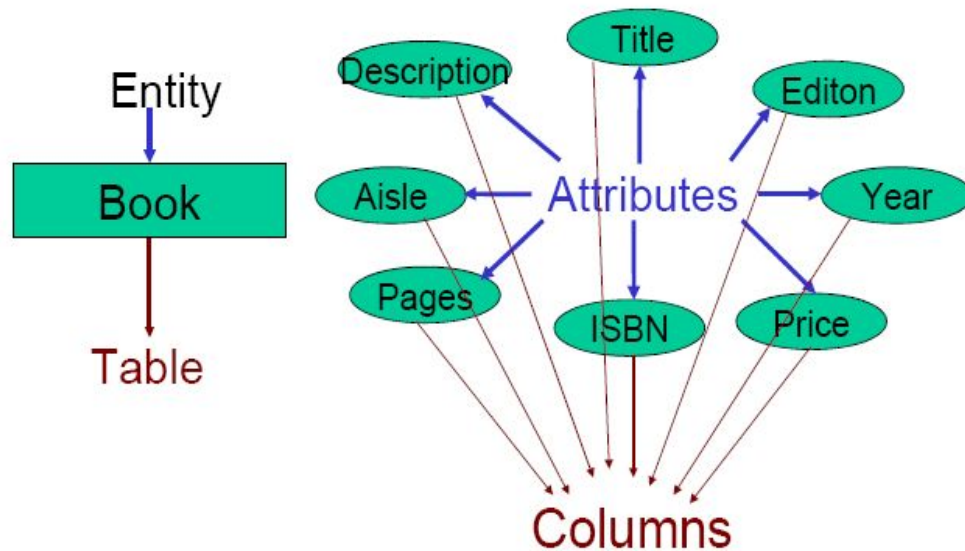
Mapping entity to a table



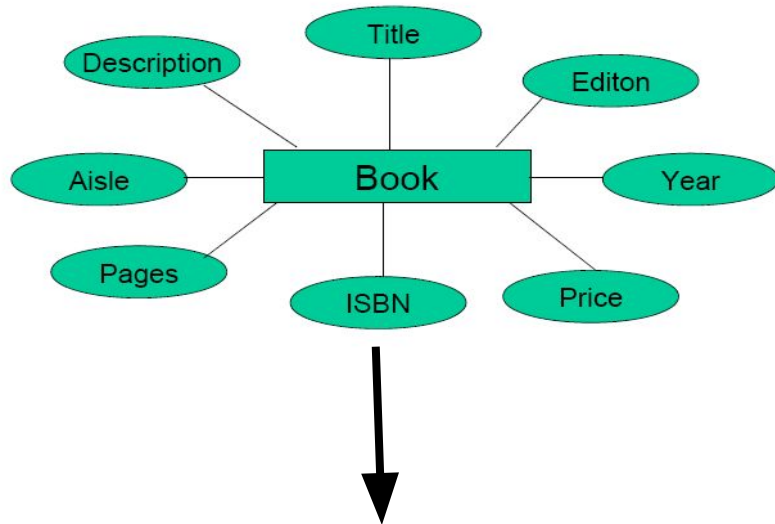
# Mapping Entity



Mapping entity to a table



# Mapping Entity

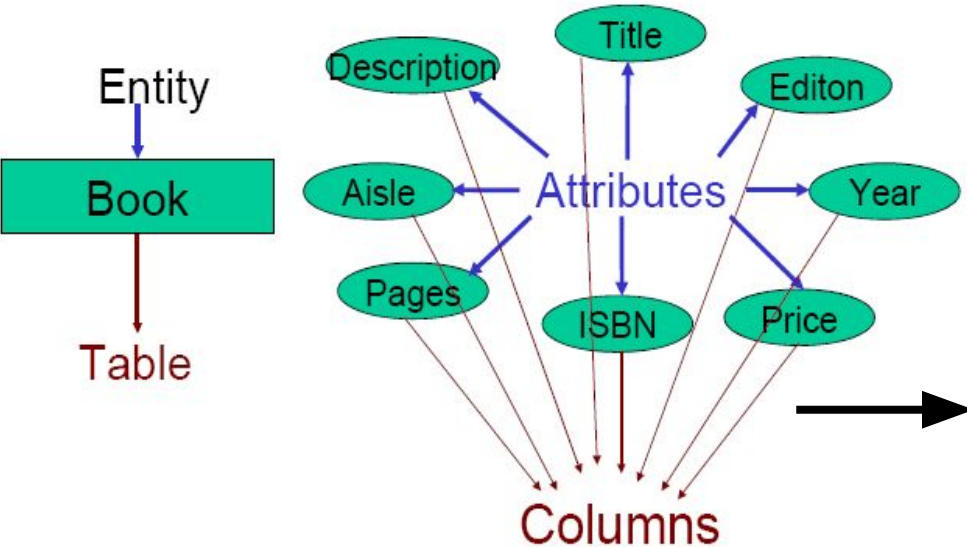


Mapping entity to a table

Mapping entity to a table (Continued)

Table: Book

Title	Edition	Year	Price	ISBN	Pages	Aisle	Description
Database Fundamentals	1	2010	24.99	978-0-98662 83-1-1	300	DB-A02	Teaches you the fundamentals of databases
Getting started with DB2 Express-C	1	2010	24.99	978-0-98662 83-5-1	280	DB-A01	Teaches you the essentials of DB2 using DB2 Express-C, the free version of DB2



# Mapping Entity - Example

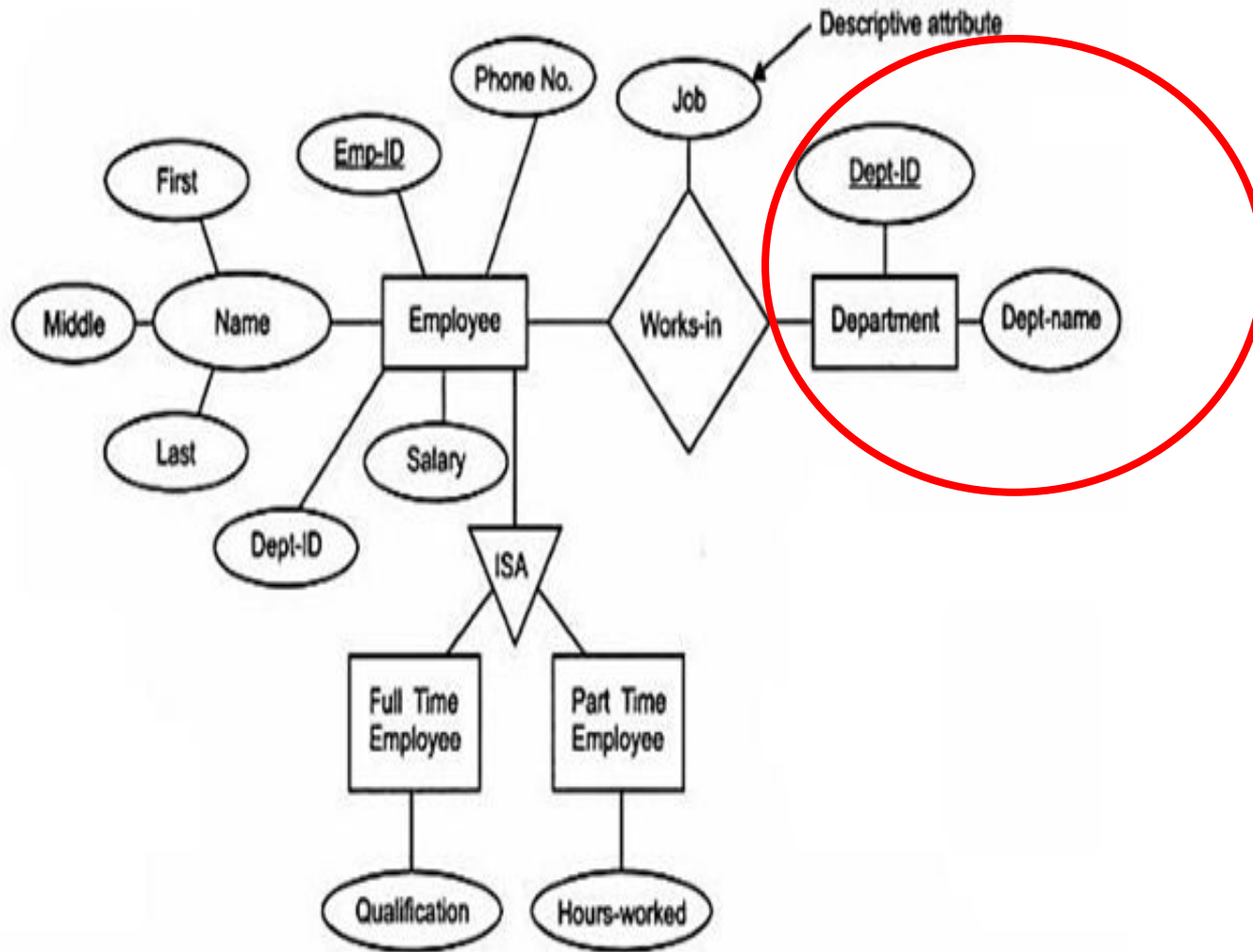


FIGURE 2.20. E-R model of employee and department entity sets.

# Mapping Entity - Example

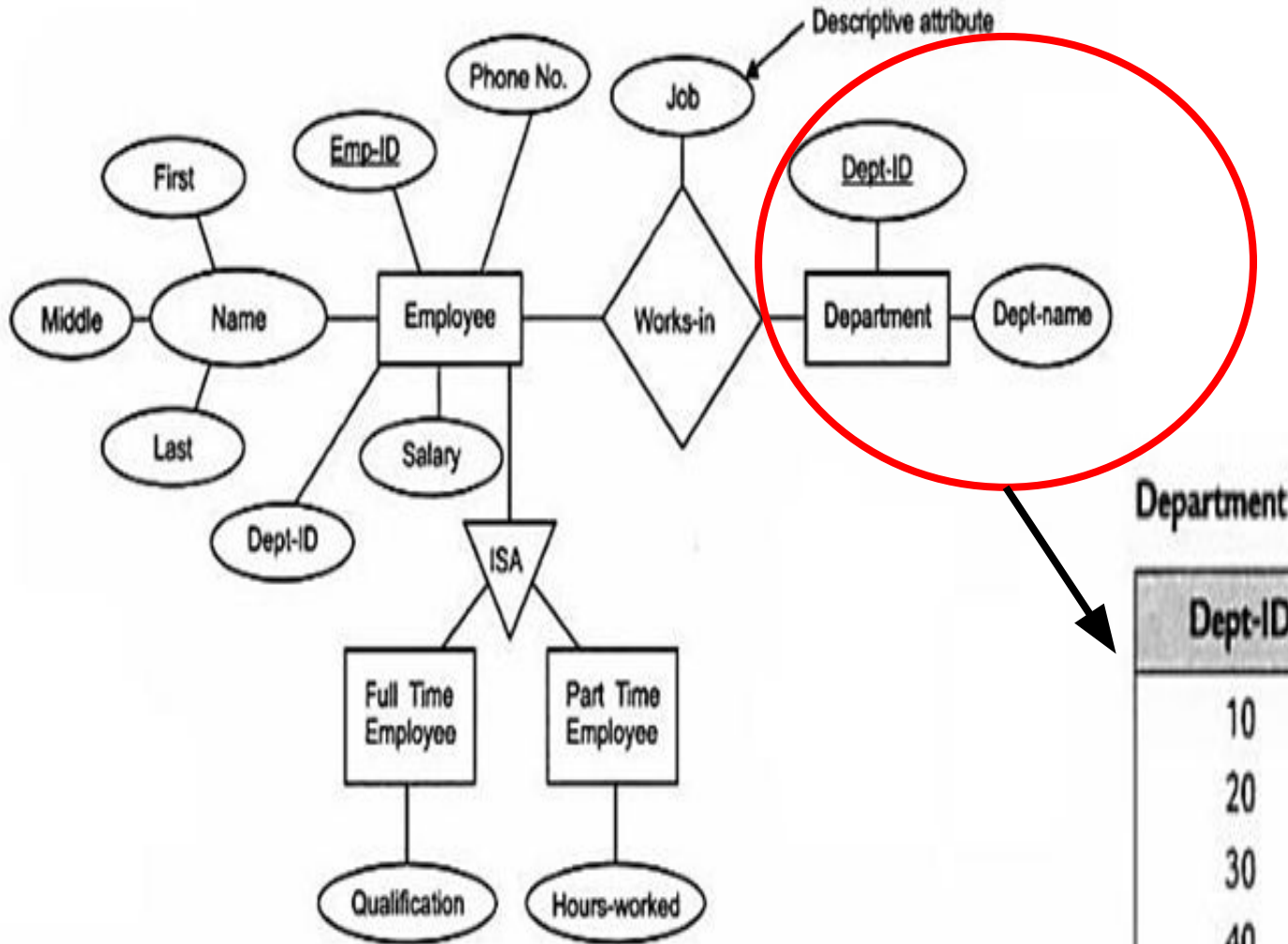


FIGURE 2.20. E-R model of employee and department entity sets.

Department

Dept-ID	Dept-name
10	Sales
20	Development
30	Testing
40	Accounts

FIGURE 2.21. The department table (Reduction of strong entity set).



# Mapping Composite Attribute - Example

- Relational model doesn't handle composite attributes

When mapping E-R composite attributes to relation schema:—

Each component attribute maps to a separate attribute in relation schema

In relation schema, simply can't refer to composite as a whole

# Mapping Composite Attribute - Example



FIGURE 2.20. E-R model of employee and department entity sets.

# Mapping Composite Attribute - Example



FIGURE 2.20. E-R model of employee and department entity sets.

# Mapping Composite Attribute - Example



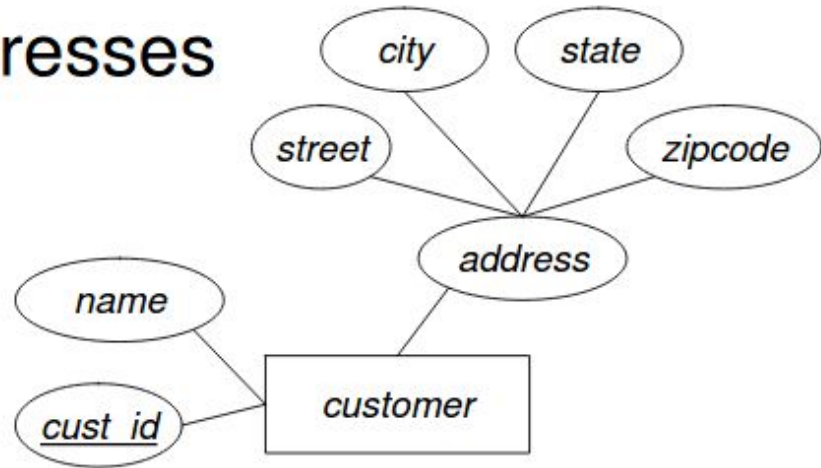
FIGURE 2.20. E-R model

EID-ID	First-name	Middle-name	Last-name	Salary	Dept-ID
A 12	Deepak	Kumar	Goyal	15,000	10
S 50	Shivi	—	Goyal	75,000	20
51 C	Anu	—	Parmar	8,000	10
67 B	Ravi	—	—	5,000	40

FIGURE 2.22. The employee table (Reduction of composite attributes).

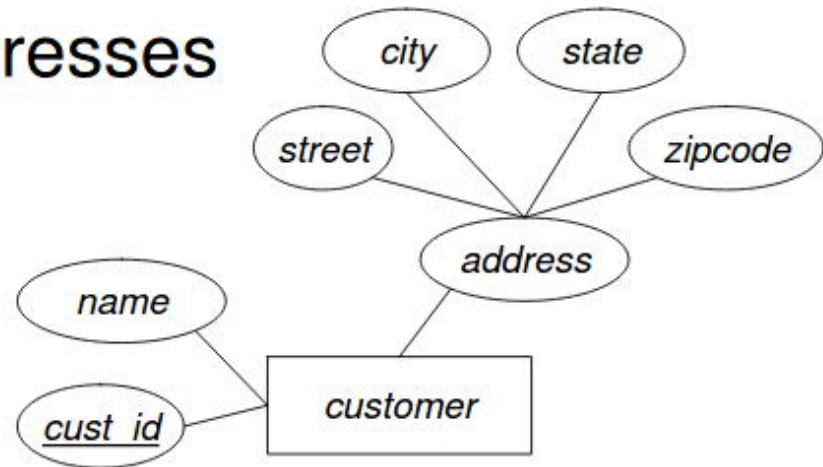
# Mapping Composite Attribute - Example

- Customers with addresses



# Mapping Composite Attribute - Example

- Customers with addresses



- Each component of address becomes a separate attribute

customer(cust\_id, name, street, city, state, zipcode)

# Mapping Multivalued Attribute

- Multivalued attributes require a separate relation schema—  
No such thing as a multivalued attribute in relational model

# Mapping Multivalued Attribute

- Multivalued attributes require a separate relation schema—  
No such thing as a multivalued attribute in relational model

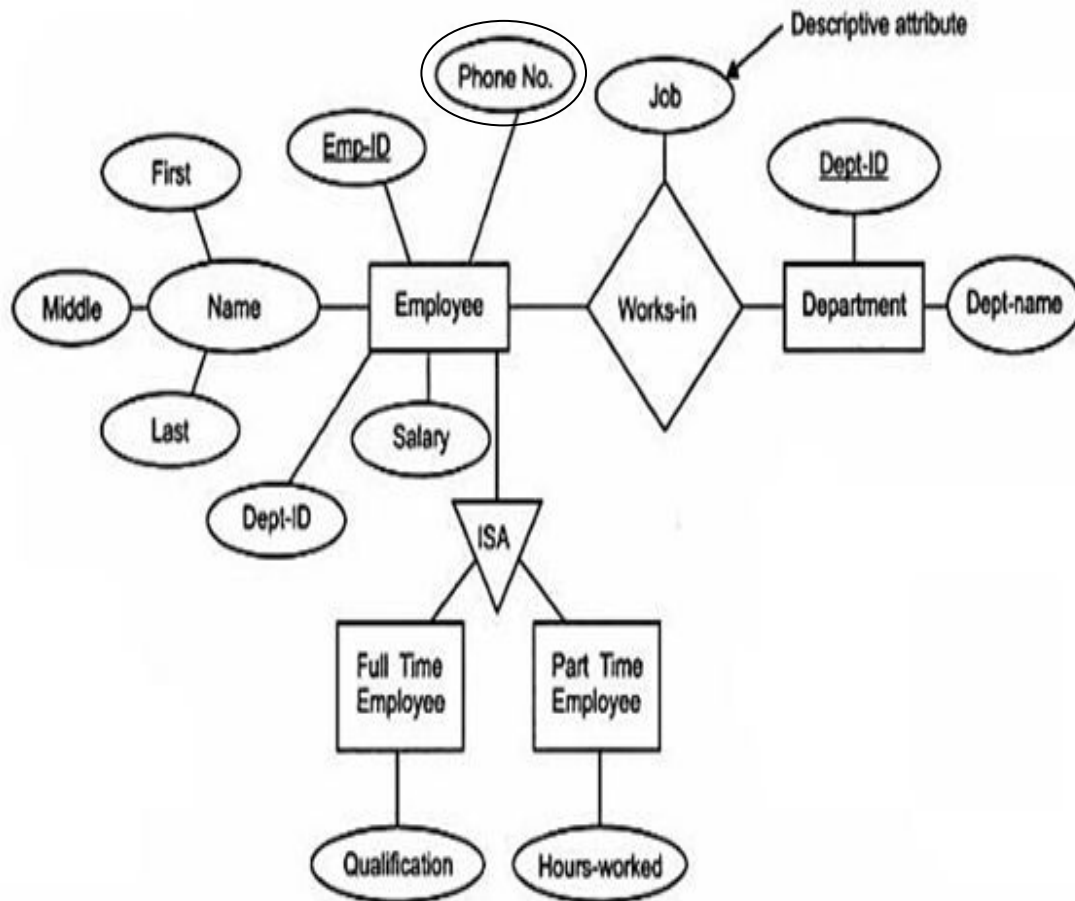


FIGURE 2.20. E-R model of employee and department entity sets.



# Mapping Multivalued Attribute

- Multivalued attributes require a separate relation schema—  
No such thing as a multivalued attribute in relational model

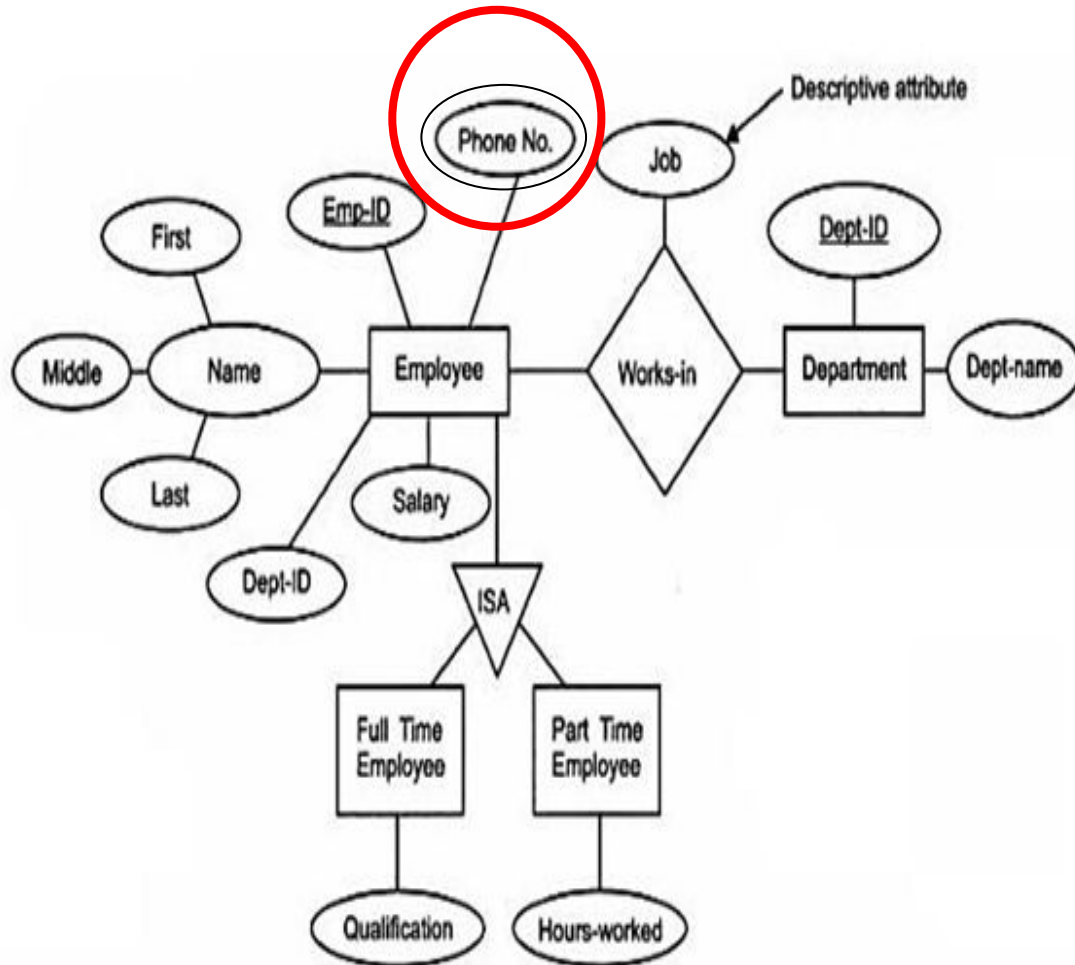
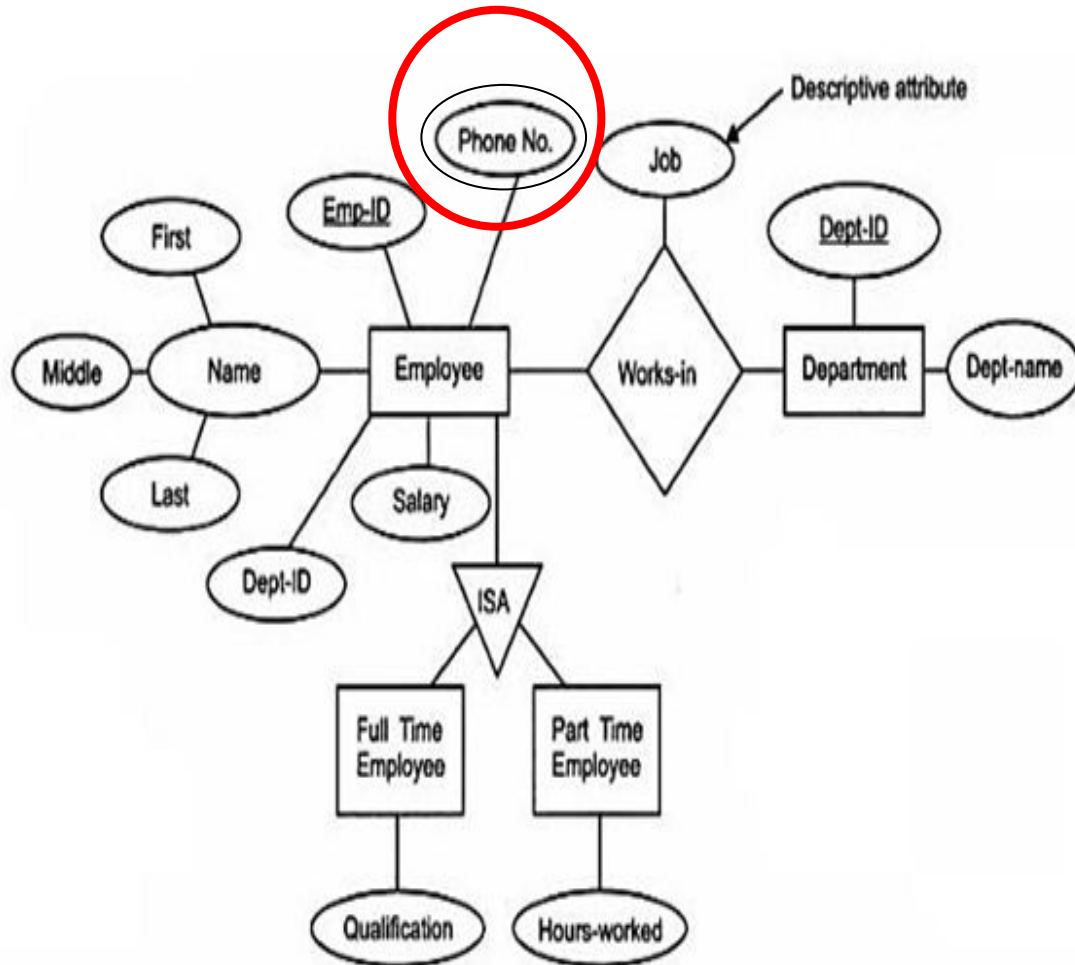


FIGURE 2.20. E-R model of employee and department entity sets.

# Mapping Multivalued Attribute

- Multivalued attributes require a separate relation schema—  
No such thing as a multivalued attribute in relational model



- For multivalued attribute M in entity-set E  
– Create a relation schema R to store M, with attribute primary key of E.

FIGURE 2.20. E-R model of employee and department entity sets.

# Mapping Multivalued Attribute

- Multivalued attributes require a separate relation schema—  
No such thing as a multivalued attribute in relational model

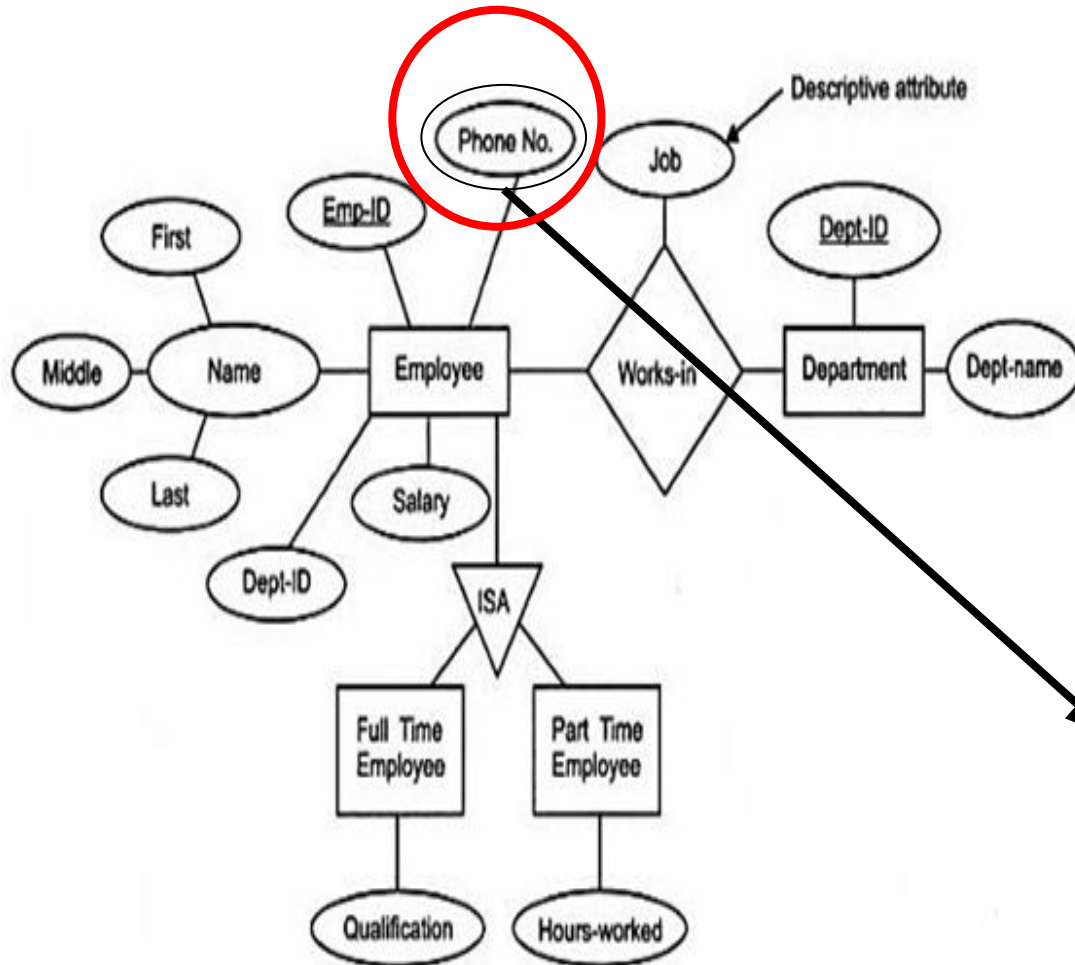


FIGURE 2.20. E-R model of employee and department entity sets.

- For multivalued attribute **M** in entity-set **E**
  - Create a relation schema **R** to store **M**, with attribute primary key of **E**.

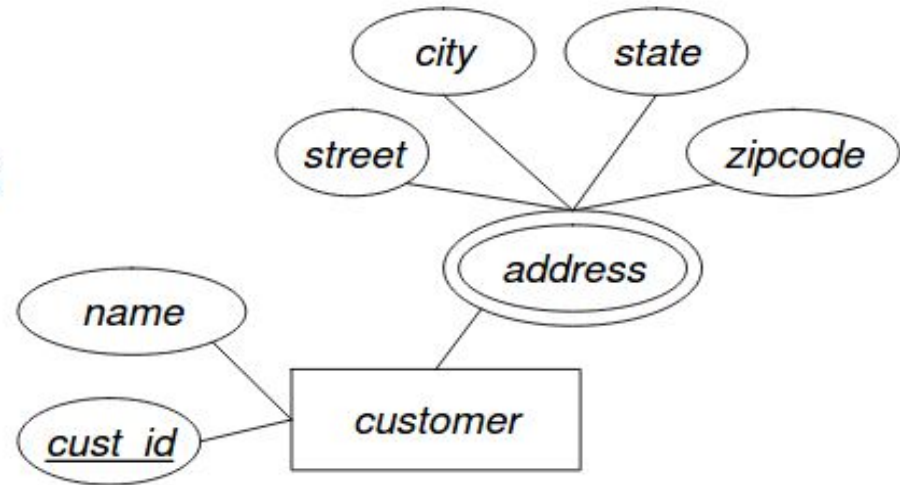
**Emp\_Phone**

Emp-ID	Phone-No.
A-12	23896
A-12	23897
51-C	38976
51-C	23551
51-C	98941
67-B	23999

FIGURE 2.23. The phone-number table (Reduction of multi-valued attributes).

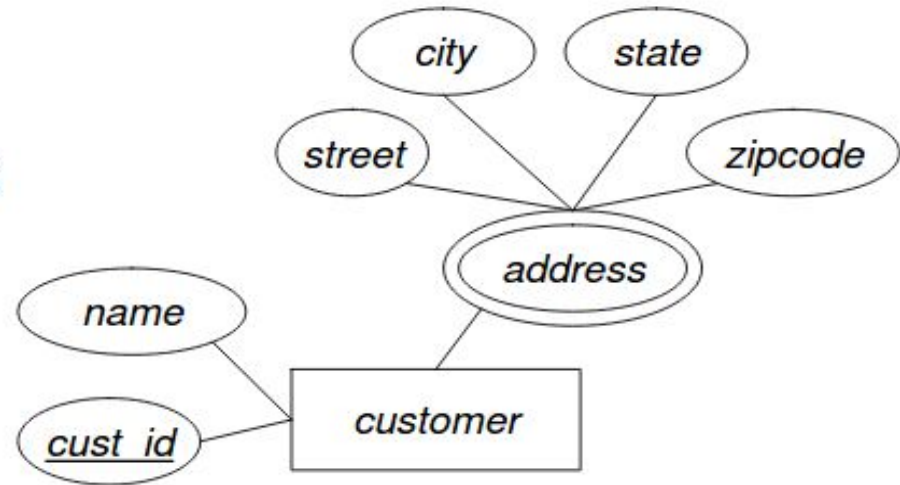
# Mapping Multivalued Attribute - Example

- Customers with multiple addresses



# Mapping Multivalued Attribute - Example

- Customers with multiple addresses



- Create separate relation to store each address  
*customer(cust id, name)*  
*cust\_addrs(cust id, street, city, state, zipcode)*

# Mapping Weak Entity

## Mapping Weak Entity Sets

A weak entity set is one which does not have any primary key associated with it.

# Mapping Weak Entity

## Mapping Weak Entity Sets

A weak entity set is one which does not have any primary key associated with it.

## Mapping Process

- Create table for weak entity set.
- Add all its attributes to table as field.
- Add the primary key of identifying entity set.
- Declare all foreign key constraints.

# Mapping Weak Entity

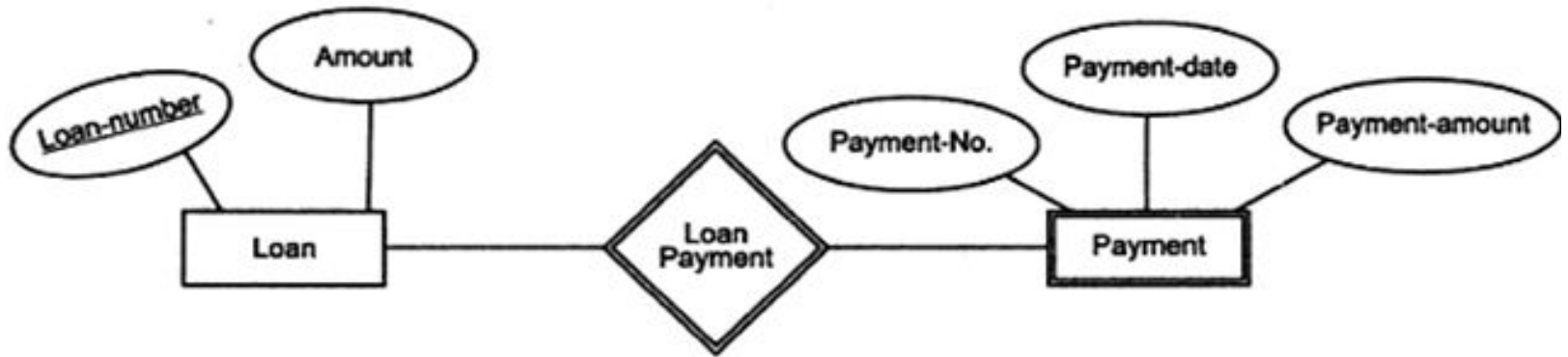


FIGURE 2.24. E-R diagram of weak entity set payment.



# Mapping Weak Entity

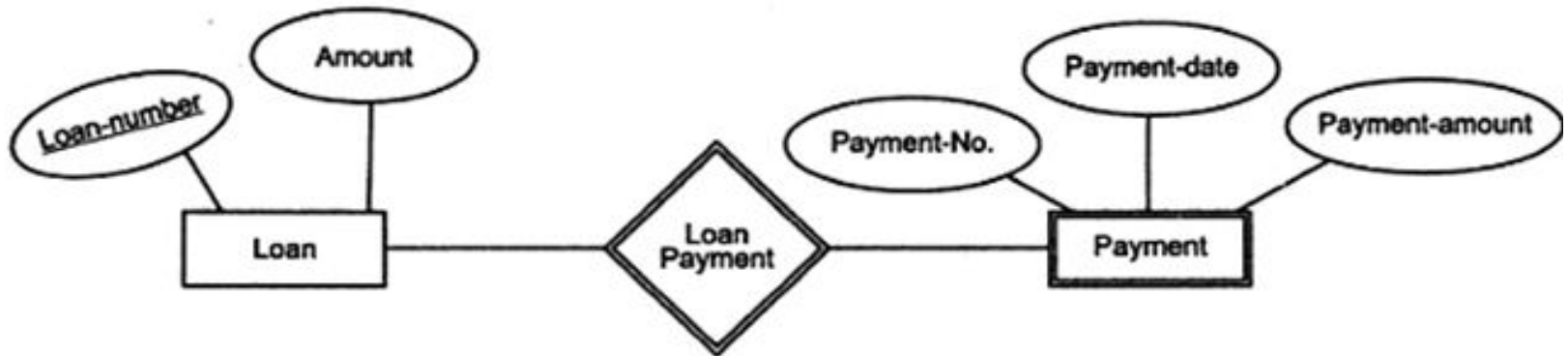


FIGURE 2.24. E-R diagram of weak entity set payment.

**Payment**

Loan-number	Payment-No.	Payment-date	Payment-amount
E-12	2	19-2-2004	6912
C-55	5	31-1-2005	5000
H-96	11	2-2-2005	2000
P-77	2	6-9-2005	2500

FIGURE 2.25. The payment table (Reduction of weak entity set).

# Mapping Weak Entity

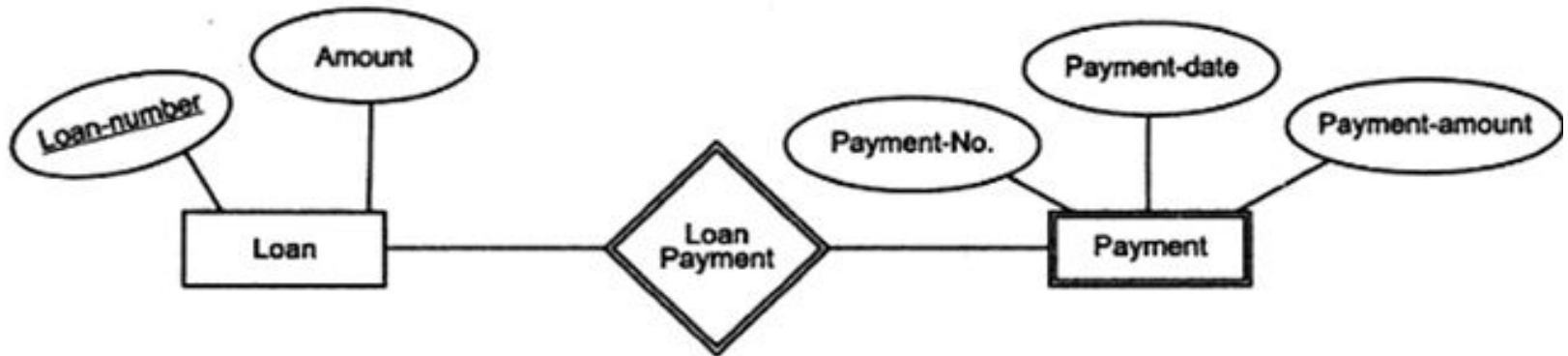


FIGURE 2.24. E-R diagram of weak entity set payment.

**foreign key**

Payment

Loan-number	Payment-No.	Payment-date	Payment-amount
E-12	2	19-2-2004	6912
C-55	5	31-1-2005	5000
H-96	11	2-2-2005	2000
P-77	2	6-9-2005	2500

FIGURE 2.25. The payment table (Reduction of weak entity set).

# Mapping Relationship

A relationship is an association among entities.

## Mapping Process

- Create table for a relationship.
- Add the primary keys of all participating Entities as fields of table with their respective data types.
- If relationship has any attribute, add each attribute as field of table.
- Declare a primary key composing all the primary keys of participating entities.
- Declare all foreign key constraints.

# Mapping Relationship

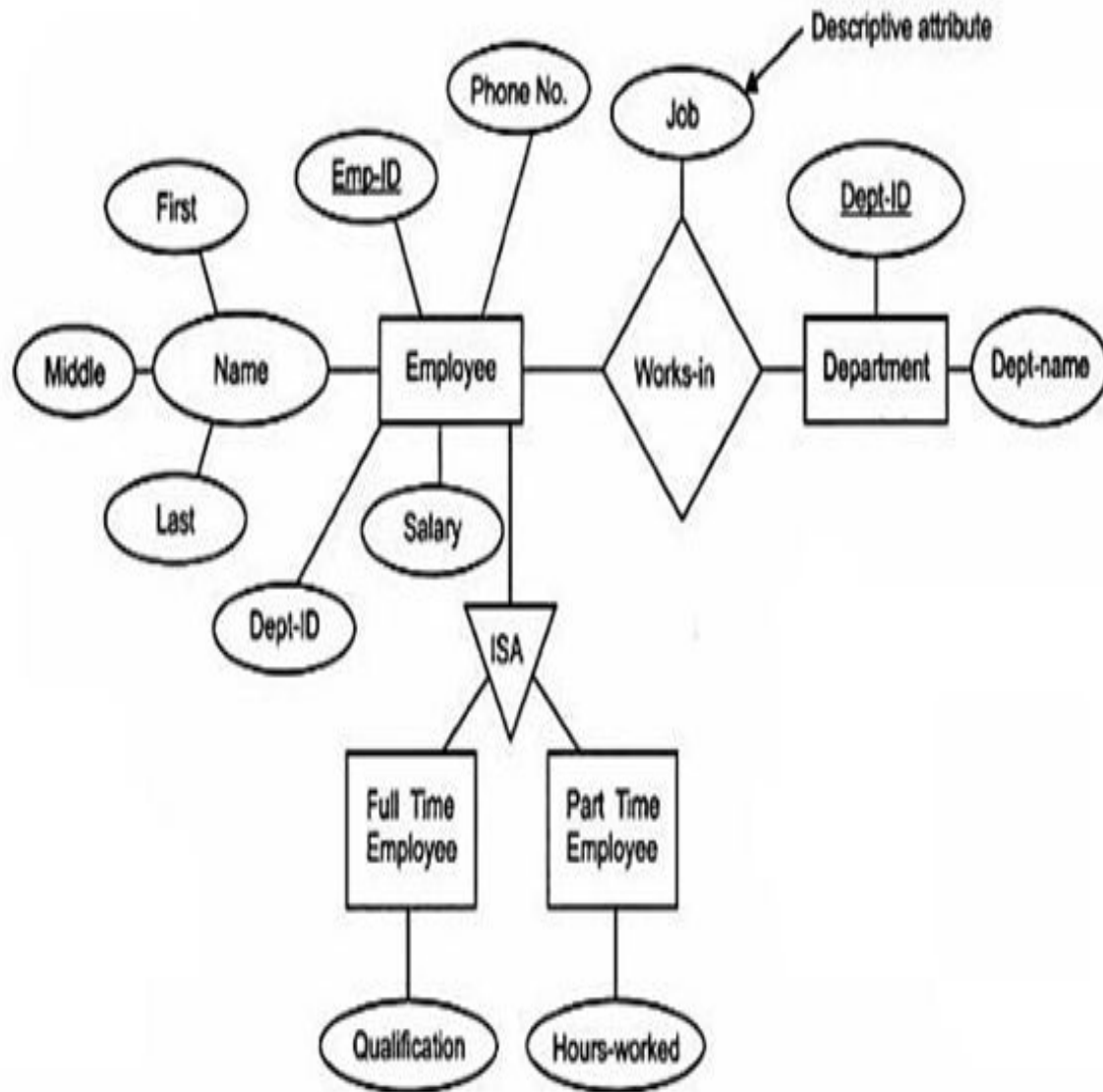


FIGURE 2.20. E-R model of employee and department entity sets.

# Mapping Relationship

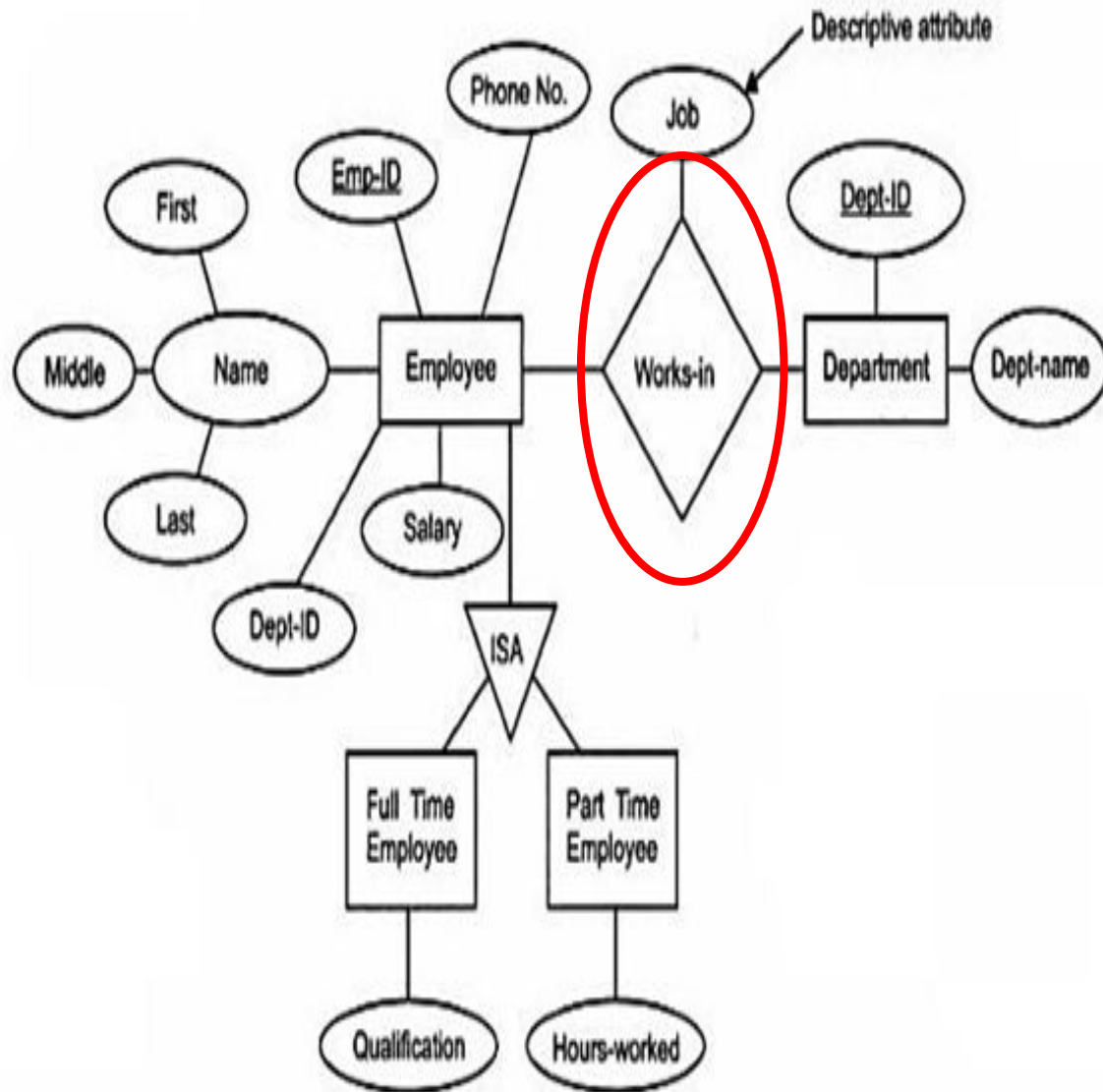


FIGURE 2.20. E-R model of employee and department entity sets.

# Mapping Relationship

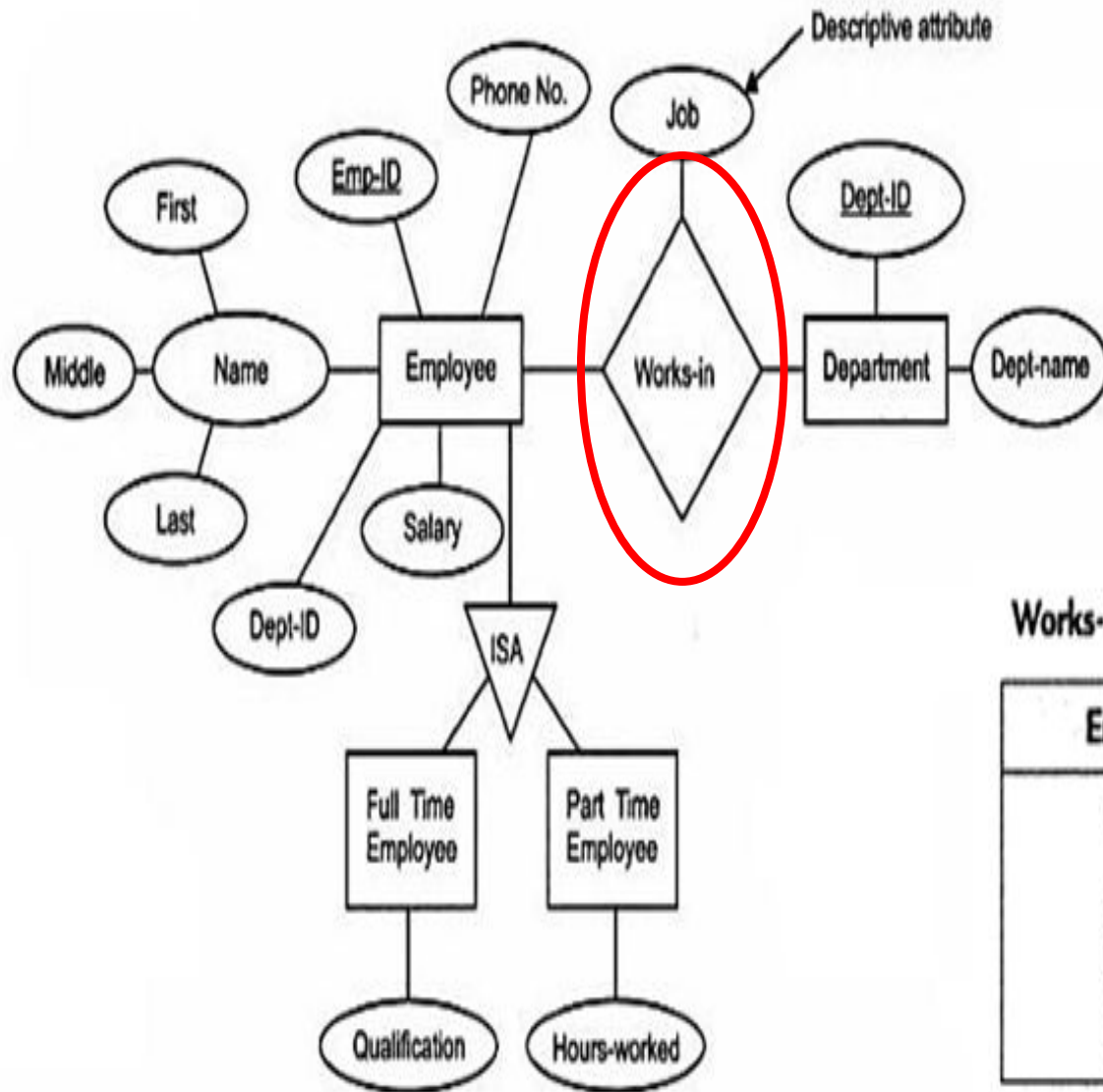


FIGURE 2.20. E-R model of employee and department entity sets.

Works-in

Emp-ID	Dept-ID	Job
S-50	20	Engineer
A-12	10	Salesman
51-C	10	Salesman
67-B	40	Accountant

FIGURE 2.26. The works-in table (Reduction of relationship sets).

# Mapping Relationship

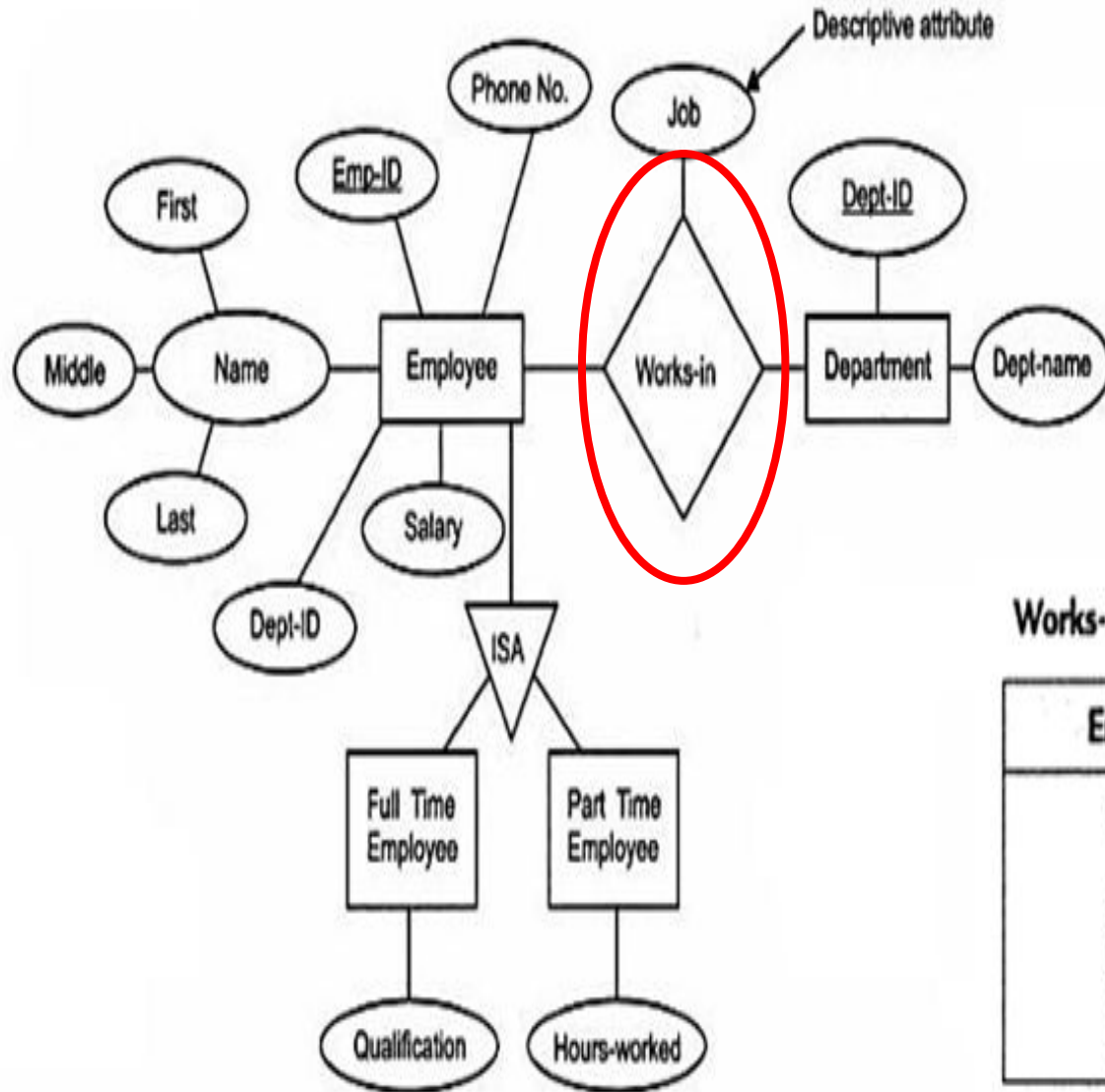


FIGURE 2.20. E-R model of employee and department entity sets.

**foreign  
key ???**

**Works-in**

Emp-ID	Dept-ID	Job
S-50	20	Engineer
A-12	10	Salesman
51-C	10	Salesman
67-B	40	Accountant

FIGURE 2.26. The works-in table (Reduction of relationship sets).



# Mapping Relationship

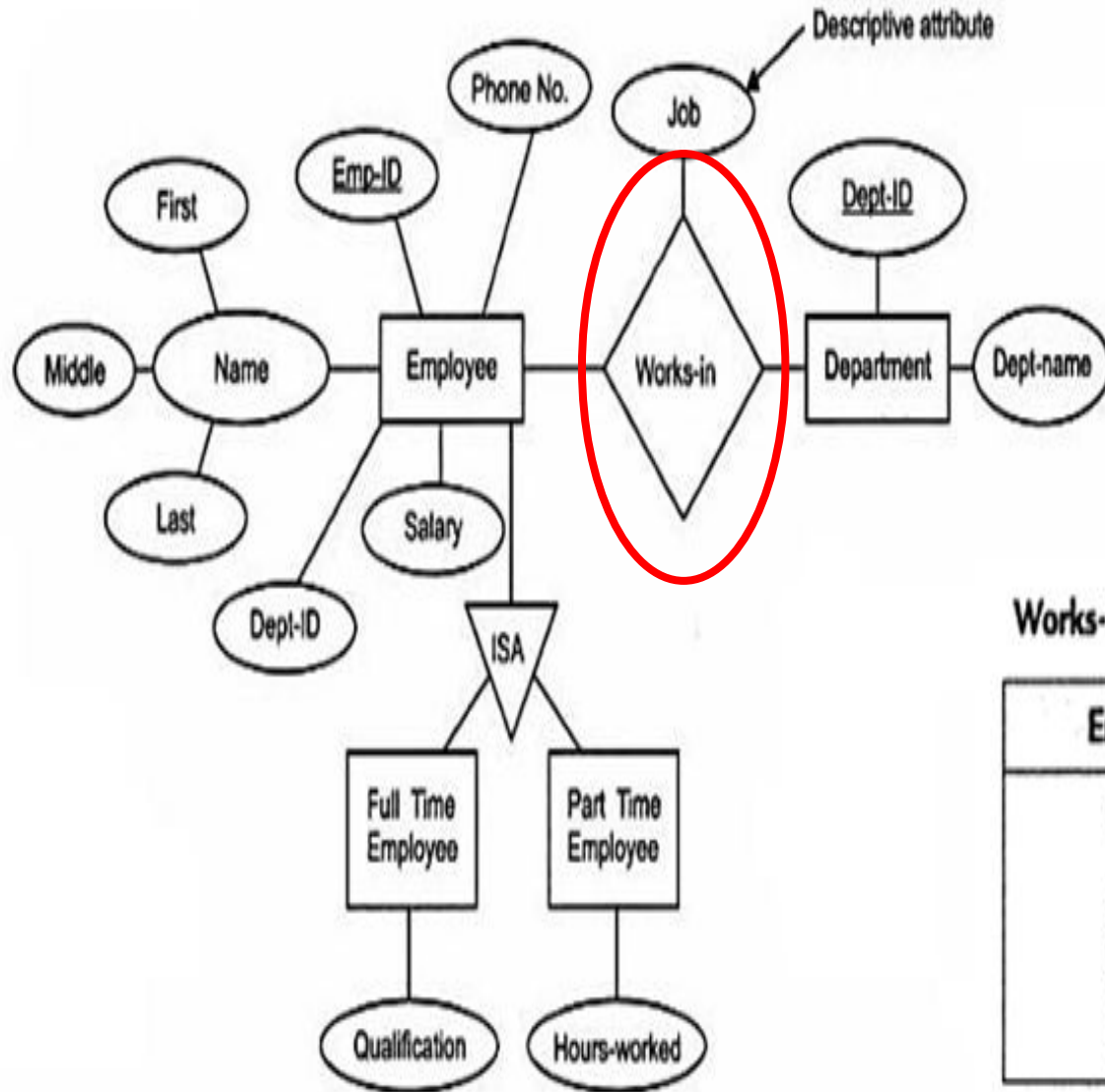


FIGURE 2.20. E-R model of employee and department entity sets.

**foreign key**

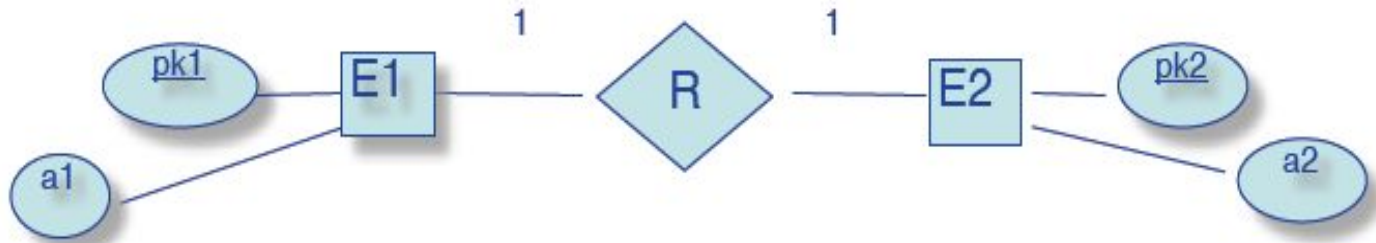
Works-in

Emp-ID	Dept-ID	Job
S-50	20	Engineer
A-12	10	Salesman
51-C	10	Salesman
67-B	40	Accountant

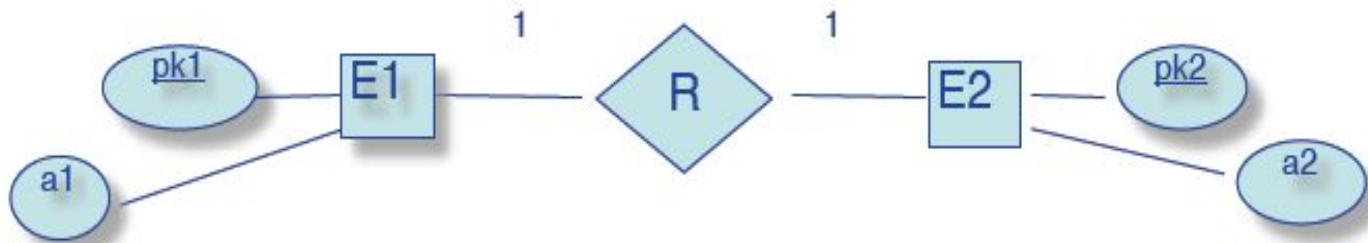
FIGURE 2.26. The works-in table (Reduction of relationship sets).



# 1-1 Relationship types

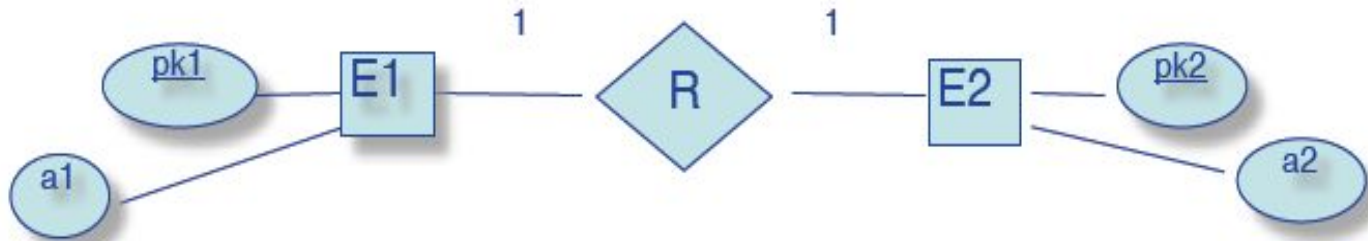


# 1-1 Relationship types



	E1	R	E2												
Alt 3:	<table><tr><td><u>pk1</u></td><td>a1</td></tr><tr><td></td><td></td></tr></table>	<u>pk1</u>	a1			<table><tr><td><u>f k1</u></td><td><u>f k2</u></td></tr><tr><td></td><td></td></tr></table>	<u>f k1</u>	<u>f k2</u>			<table><tr><td><u>pk2</u></td><td>a2</td></tr><tr><td></td><td></td></tr></table>	<u>pk2</u>	a2		
<u>pk1</u>	a1														
<u>f k1</u>	<u>f k2</u>														
<u>pk2</u>	a2														

# 1-1 Relationship types



Alt 1:

<u>pk1</u>	a1

<u>pk2</u>	a2	f k1

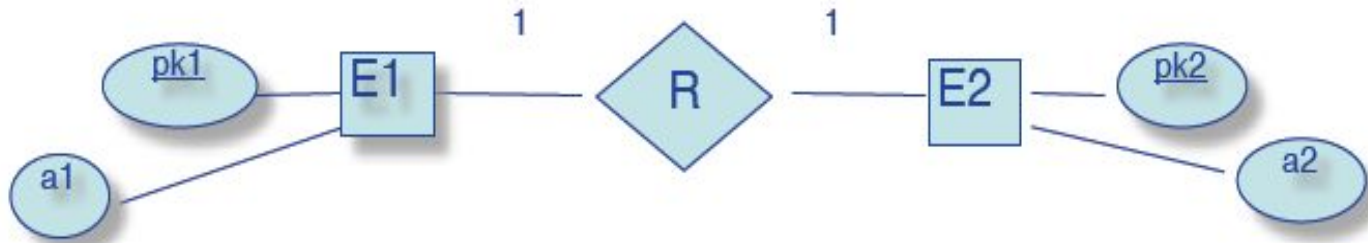
Alt 3:

<u>pk1</u>	a1

<u>f k1</u>	<u>f k2</u>

<u>pk2</u>	a2

# 1-1 Relationship types



Alt 1:

<u>pk1</u>	a1

<u>pk2</u>	a2	f k1

Alt 2:

<u>pk1</u>	a1	f k2

<u>pk2</u>	a2

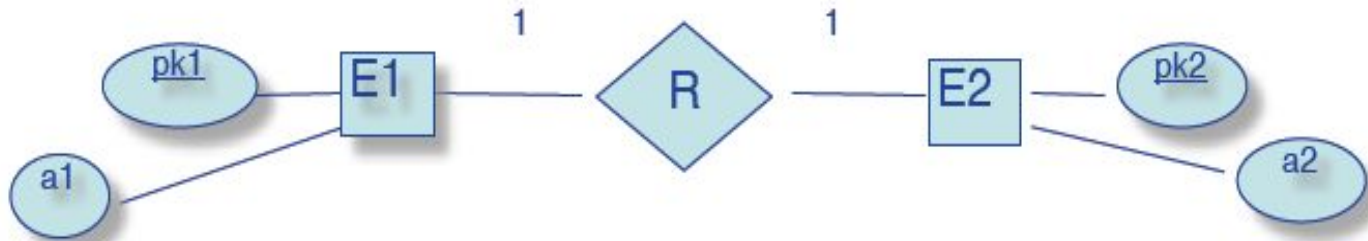
Alt 3:

<u>pk1</u>	a1

<u>f k1</u>	<u>f k2</u>

<u>pk2</u>	a2

# 1-1 Relationship types



Alt 1:

<u>pk1</u>	a1

<u>pk2</u>	a2	f k1

Alt 2:

<u>pk1</u>	a1	f k2

<u>pk2</u>	a2

Alt 3:

<u>pk1</u>	a1

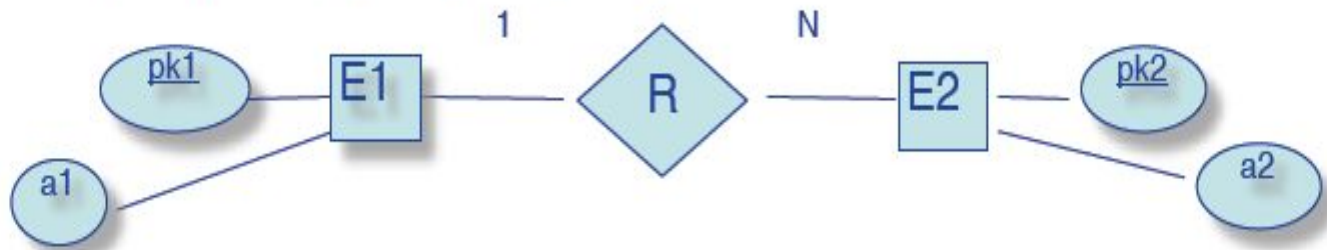
<u>f k1</u>	<u>f k2</u>

<u>pk2</u>	a2

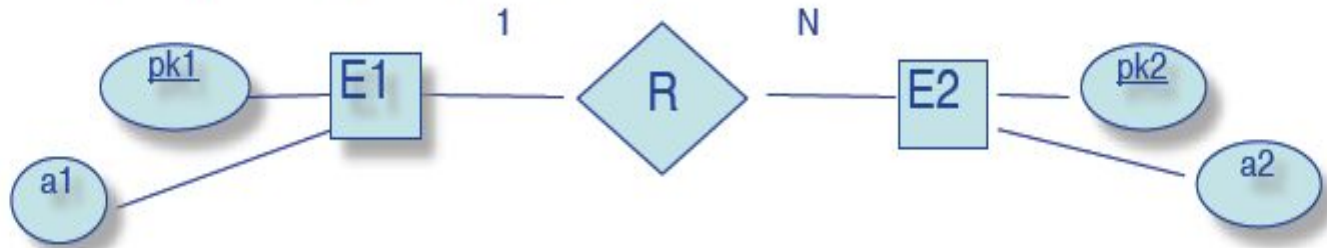
Alt 4:

<u>pk1</u>	a1	pk2	a2

# 1-N Relationship types



# 1-N Relationship types



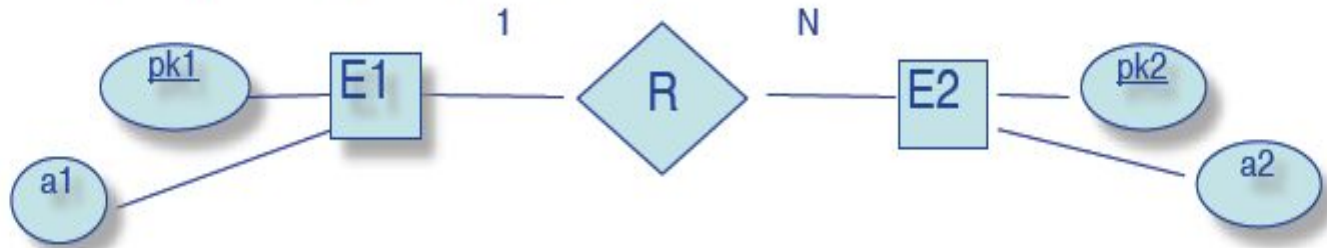
Alt 2:

<u>pk1</u>	a1

f k1	<u>f k2</u>

<u>pk2</u>	a2

# 1-N Relationship types



Alt 1:

<u>pk1</u>	a1

<u>pk2</u>	a2	f k1

Alt 2:

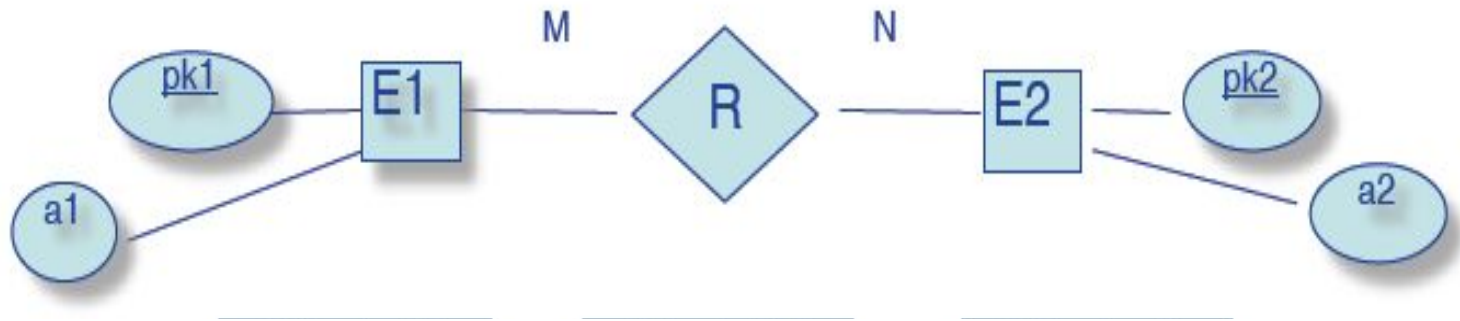
<u>pk1</u>	a1

f k1	<u>f k2</u>

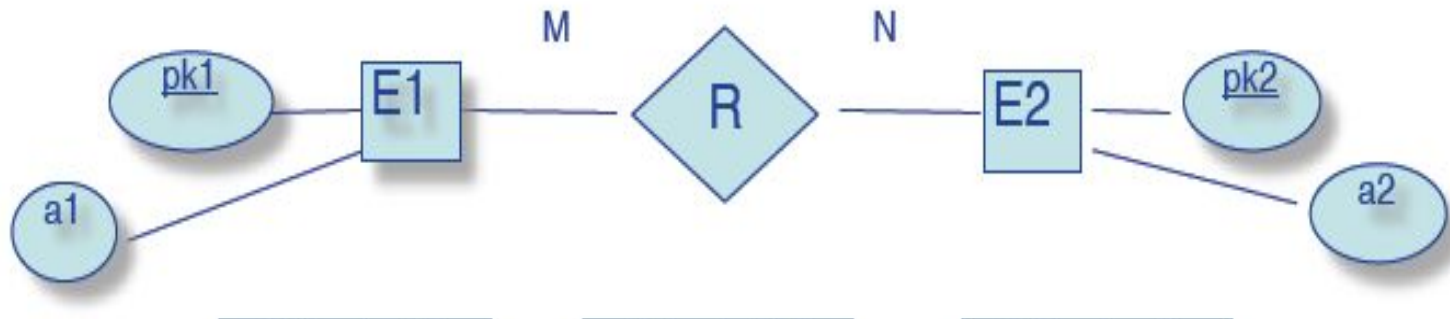
<u>pk2</u>	a2



# M-N Relationship types



# M-N Relationship types

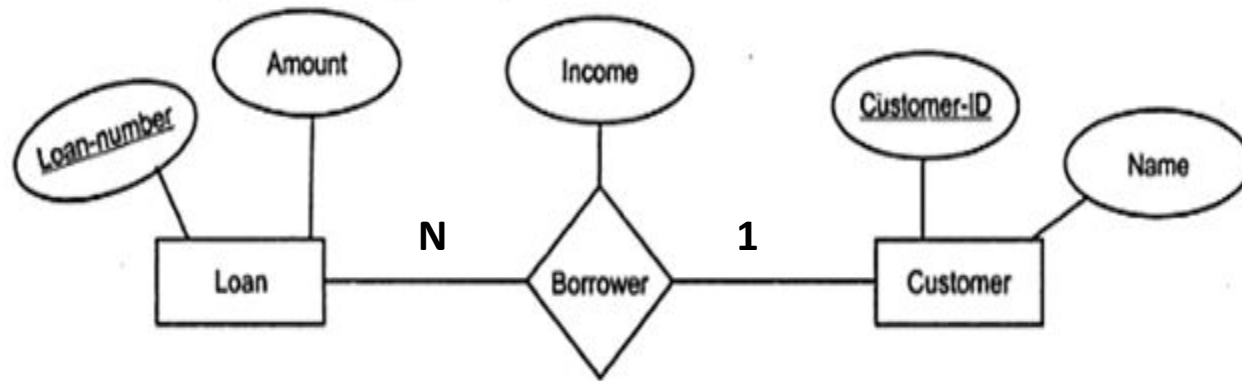


<u>pk1</u>	a1

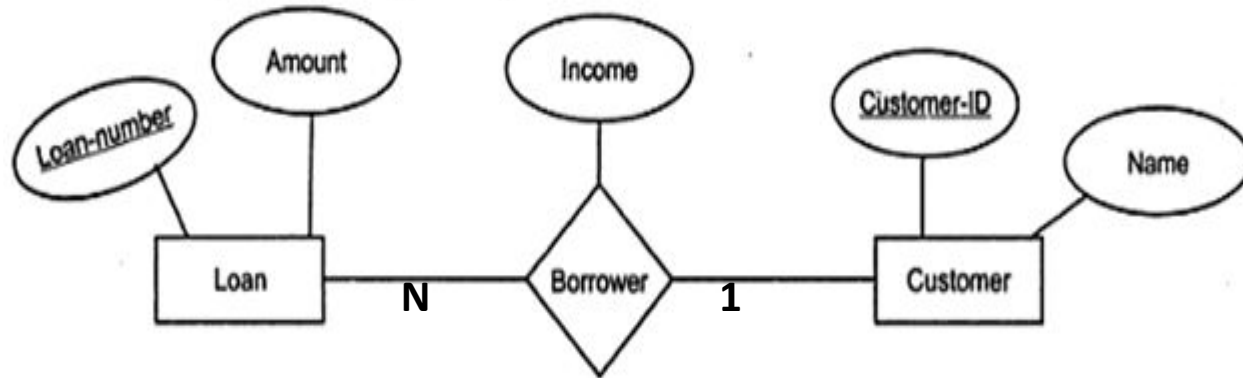
<u>f k1</u>	<u>f k2</u>

<u>pk2</u>	a2

# Example



## Example



### **Solution 1:**

**Loan{Loan-number, amount}**

**Borrower{Loan-number, Customer-ID, Income}**

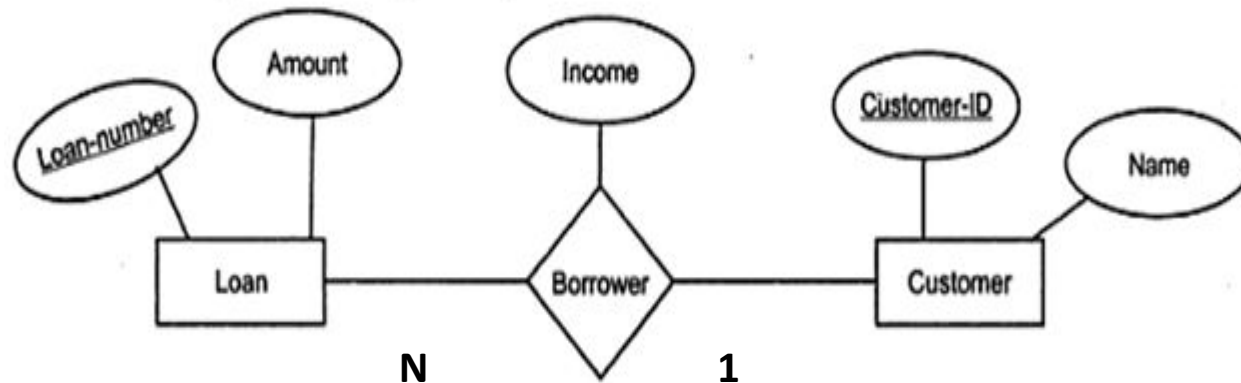
**Customer{Customer-ID, Name}**

**Primary Key: Loan-number in Loan Relation**

**Primary Key: Customer-ID in Customer Relation**

**Foreign Key: Loan-number, Customer-ID in Borrower Relation**

## Example



### **Solution 1:**

Loan{Loan-number, amount}

Borrower{Loan-number, Customer-ID, Income}

Customer{Customer-ID, Name}

Primary Key: Loan-number in Loan Relation

Primary Key: Customer-ID in Customer Relation

Foreign Key: Loan-number, Customer-ID in Borrower Relation

### **Solution 2:**

Loan{Loan-number, amount, Customer-ID, Income}

Customer{Customer-ID, Name}

Primary Key: Loan-number in Loan Relation

Primary Key: Customer-ID in Customer Relation

Foreign Key: Customer-ID in Loan Relation

# Mapping Hierarchical Entities

ER specialization or  
generalization comes in  
the form of hierarchical  
entity sets

# Mapping Hierarchical Entities

ER specialization or generalization comes in the form of hierarchical entity sets

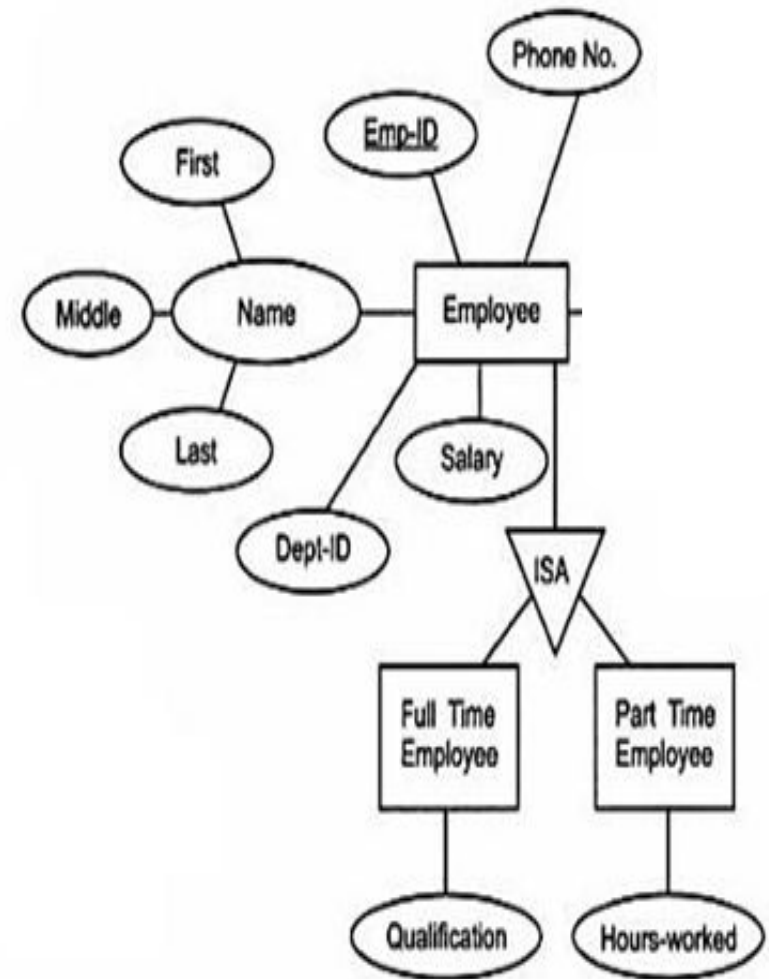


FIGURE 2.20. E-R model of employee and depa

# Mapping Hierarchical Entities

ER specialization or generalization comes in the form of hierarchical entity sets

## Mapping Process

- Create tables for all higher-level entities.
- Create tables for lower-level entities.
- Add primary keys of higher-level entities in the table of lower-level entities.
- In lower-level tables, add all other attributes of lower-level entities.
- Declare primary key of higher-level table and the primary key for lower-level table.
- Declare foreign key constraints.

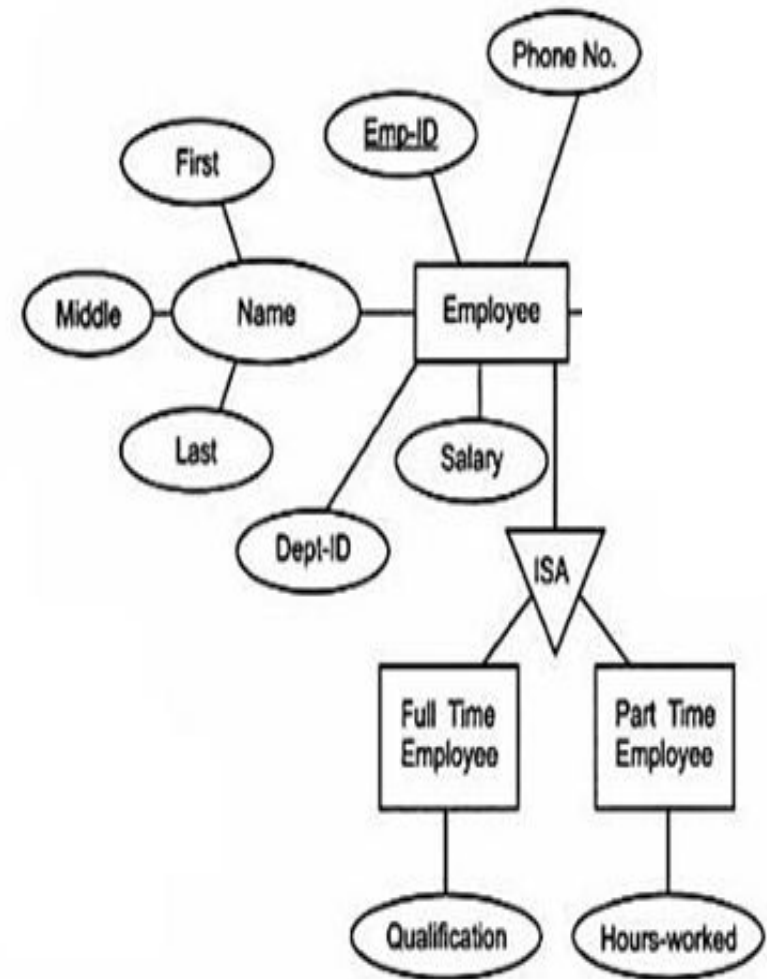


FIGURE 2.20. E-R model of employee and depa



# Mapping Hierarchical Entities

ER specialization or generalization comes in the form of hierarchical entity sets

## Mapping Process

- Create tables for all higher-level entities.
- Create tables for lower-level entities.
- Add primary keys of higher-level entities in the table of lower-level entities.
- In lower-level tables, add all other attributes of lower-level entities.
- Declare primary key of higher-level table and the primary key for lower-level table.
- Declare foreign key constraints.

### Solution:

Employee (Emp-ID, Fname, Name, Name, Salary, Phone-no)

FullTime(Emp-ID, Qualification)

PartTime(Emp-ID, Hours-worked)

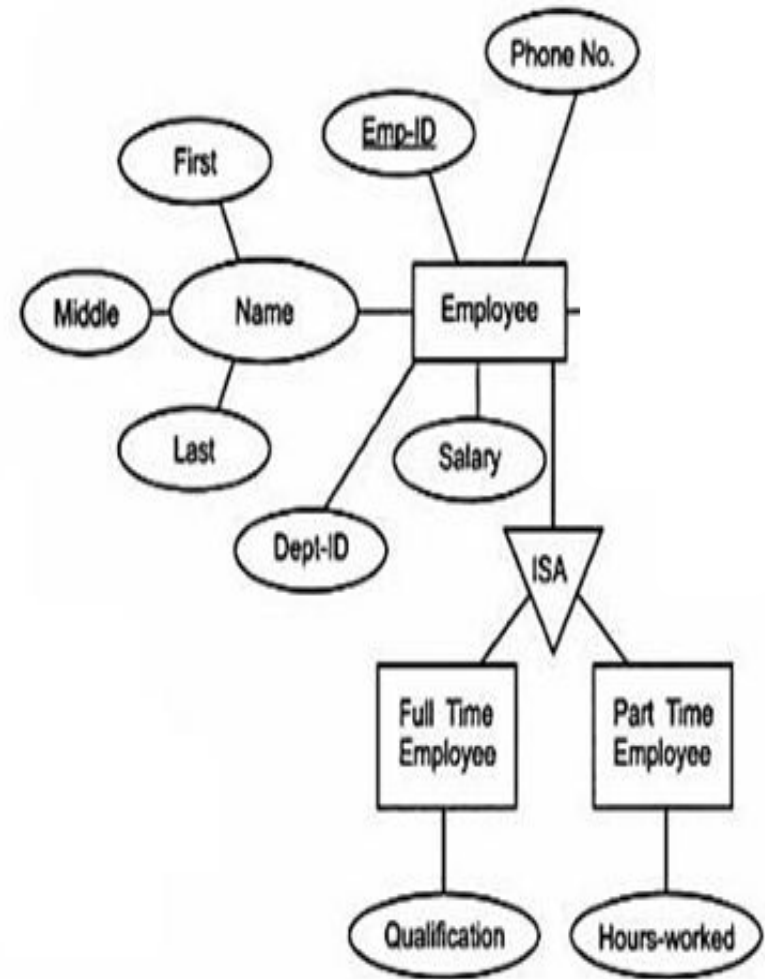
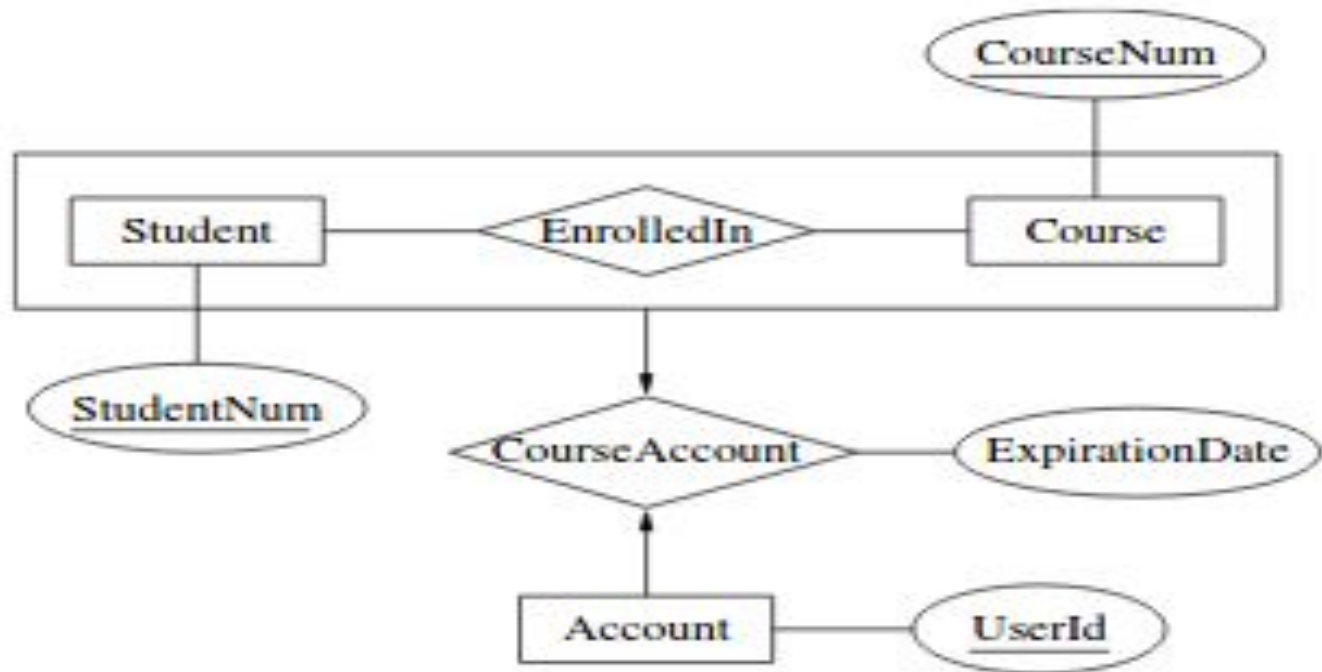


FIGURE 2.20. E-R model of employee and depa

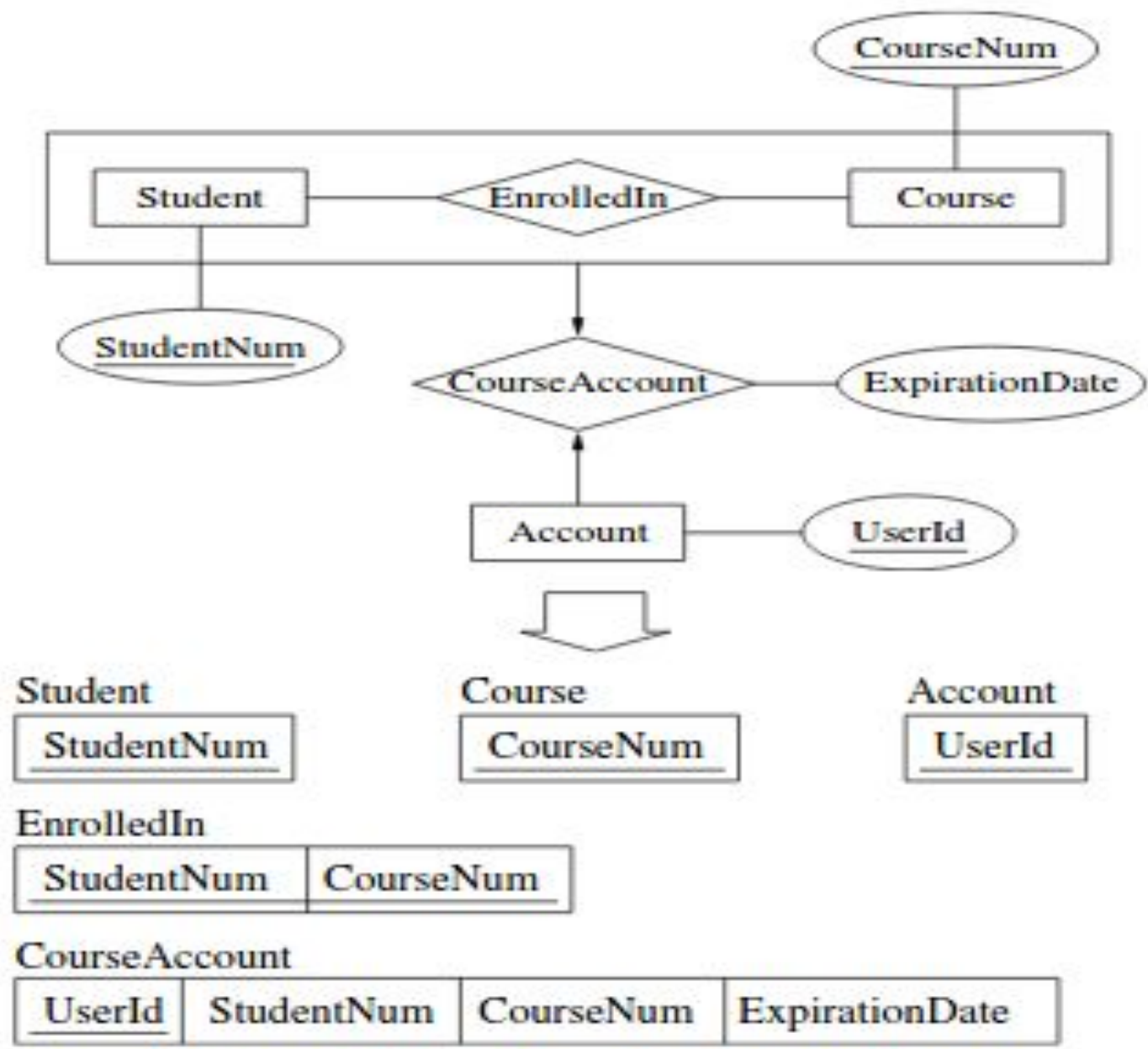
# Mapping Aggregation



## Mapping Process

- Create tables for all higher-level entities.
- Create tables for all relationships
- Declare primary key and foreign key constraints

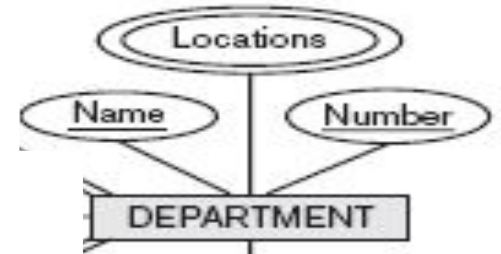
# Mapping Aggregation



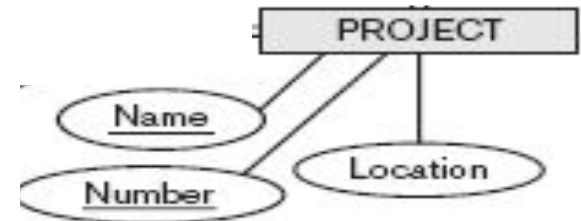
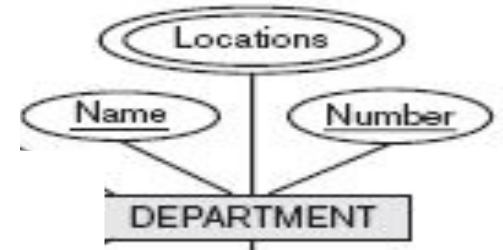
# Draw ER diagram for COMPANY DATABASE

- The company is organized into departments. Each department has a unique name, a unique number, and a particular employee who manages the department. We keep track of the start date when that employee began managing the department. A department may have several locations.
- A department controls a number of projects, each of which has a unique name, a unique number, and a single location.
- We store each employee's name, Social Security number, address, salary, sex (gender), and birth date. An employee is assigned to one department, but may work on several projects, which are not necessarily controlled by the same department.
- We keep track of the current number of hours per week that an employee works on each project. We also keep track of the direct supervisor of each employee (who is another employee).
- We want to keep track of the dependents of each employee for insurance purposes. We keep each dependent's first name, sex, birth date, and relationship to the employee.

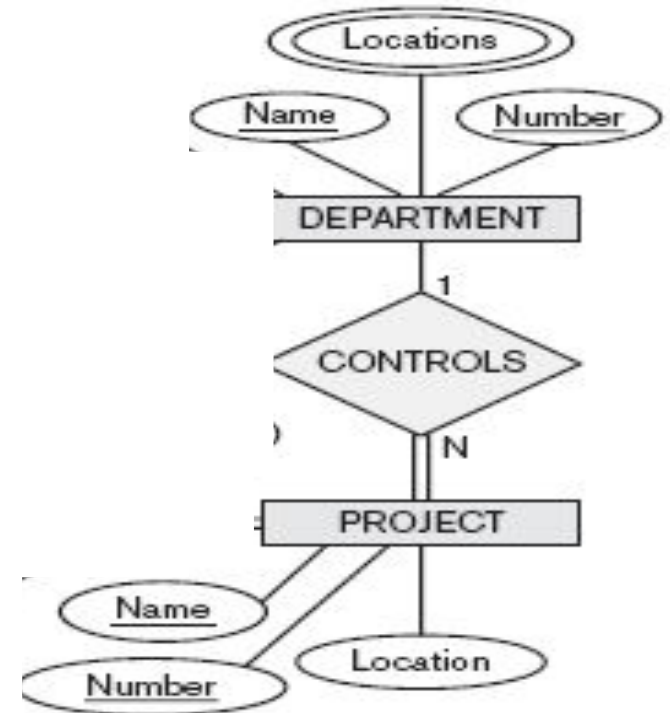
# Draw ER diagram for COMPANY DATABASE



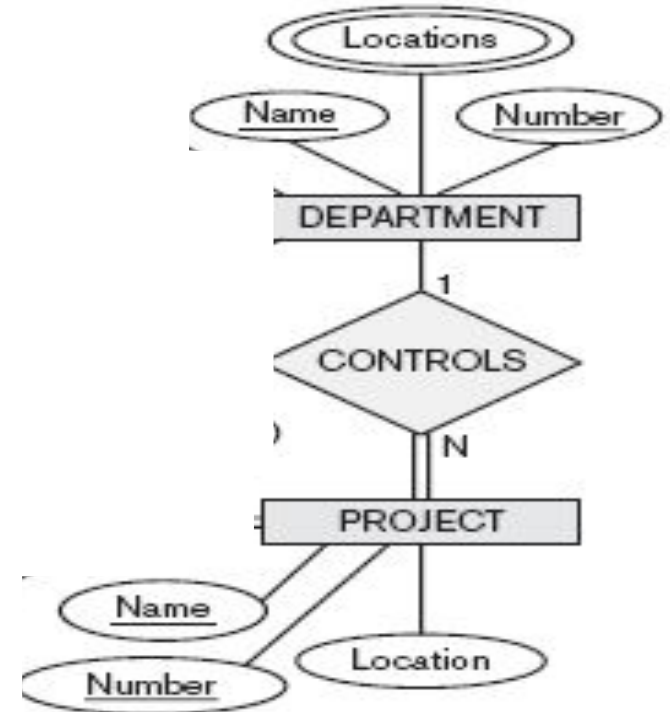
# Draw ER diagram for COMPANY DATABASE



# Draw ER diagram for COMPANY DATABASE

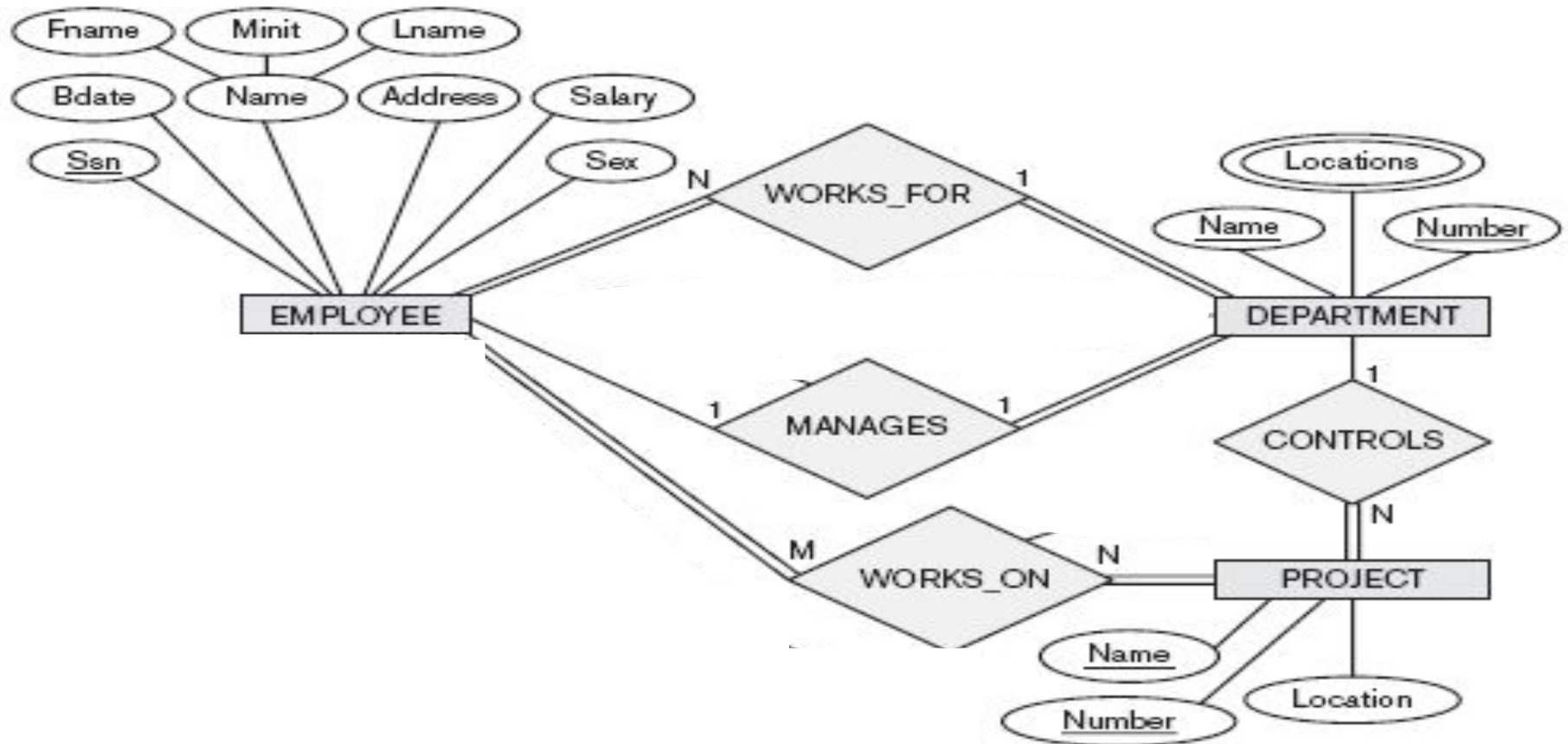


# Draw ER diagram for COMPANY DATABASE

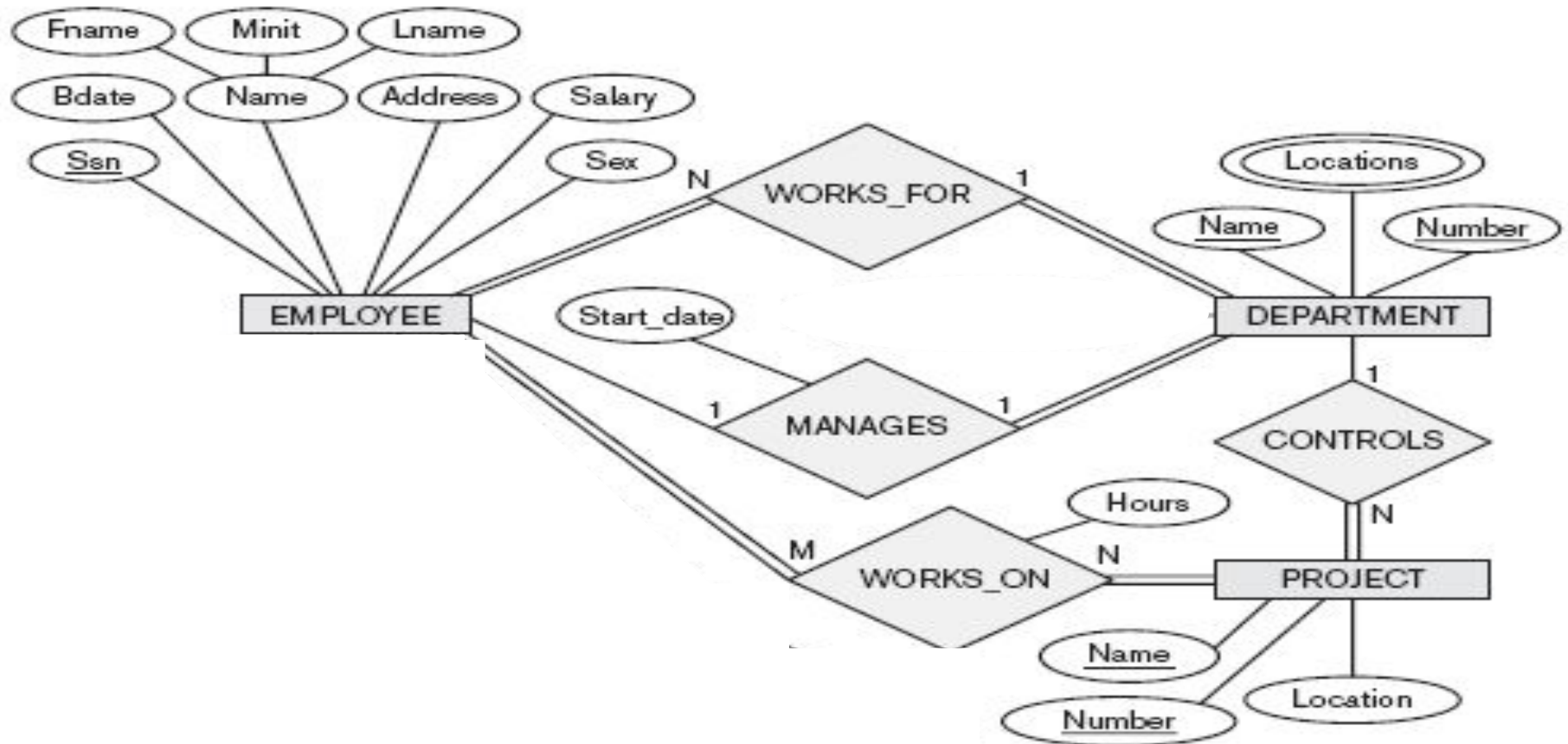




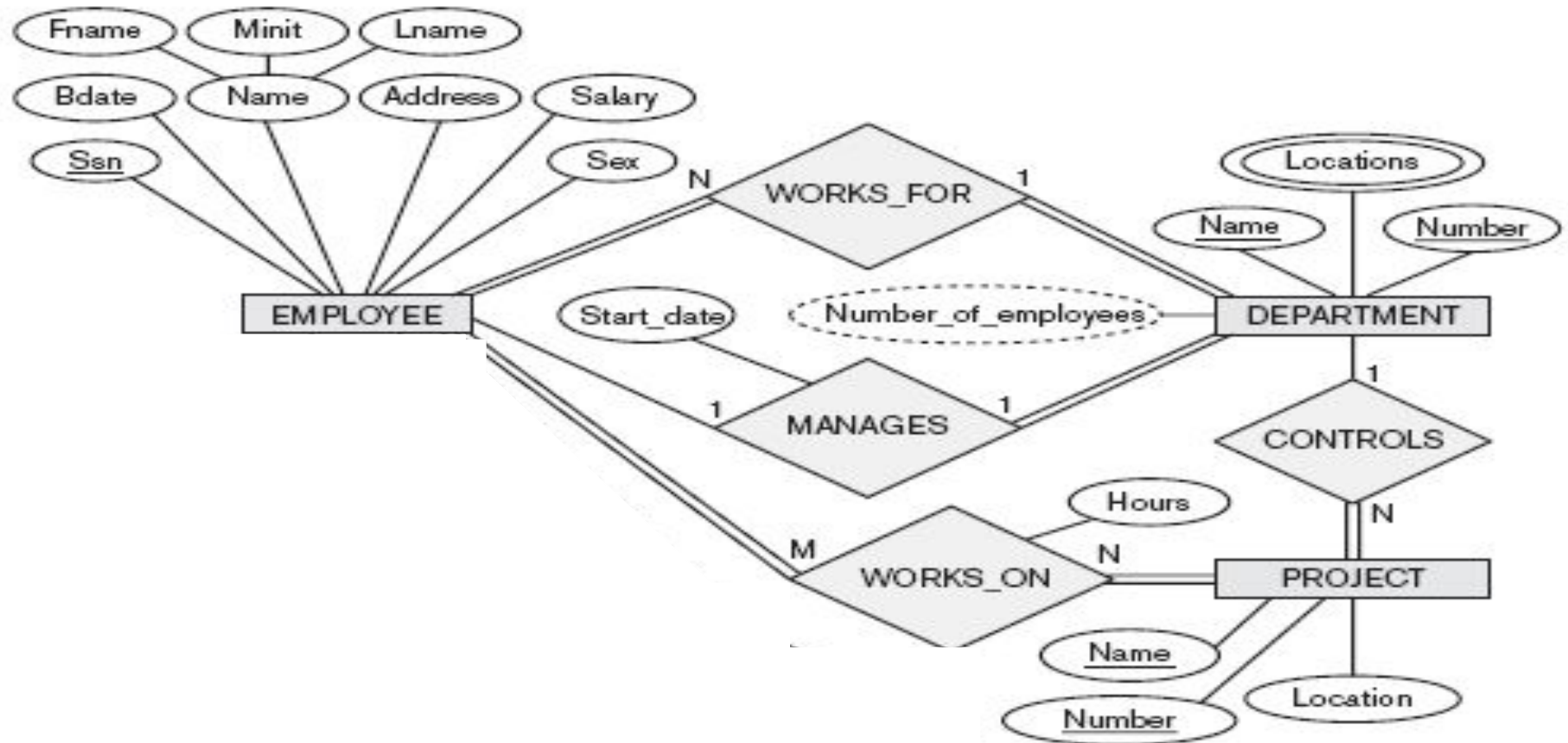
# Draw ER diagram for COMPANY DATABASE



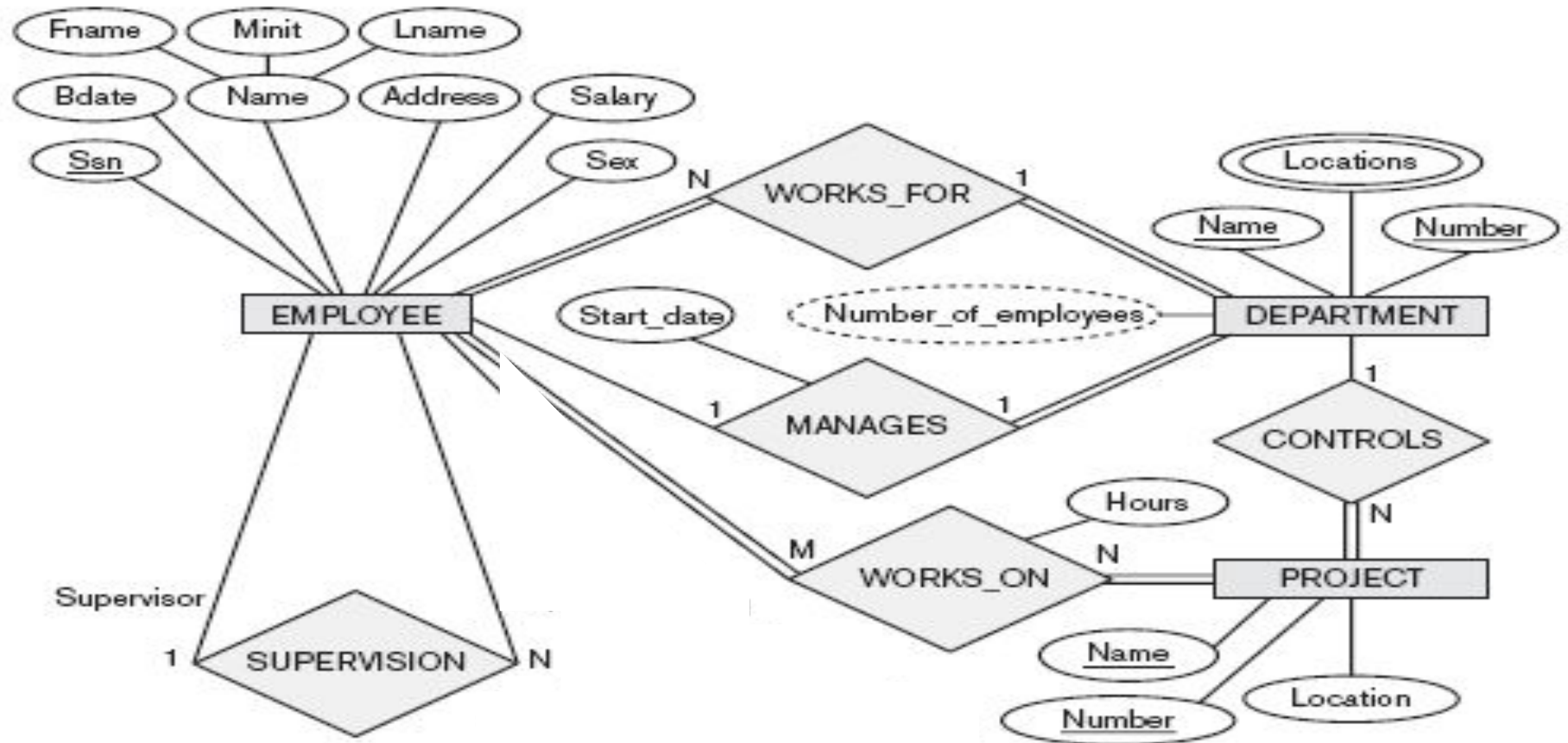
# Draw ER diagram for COMPANY DATABASE



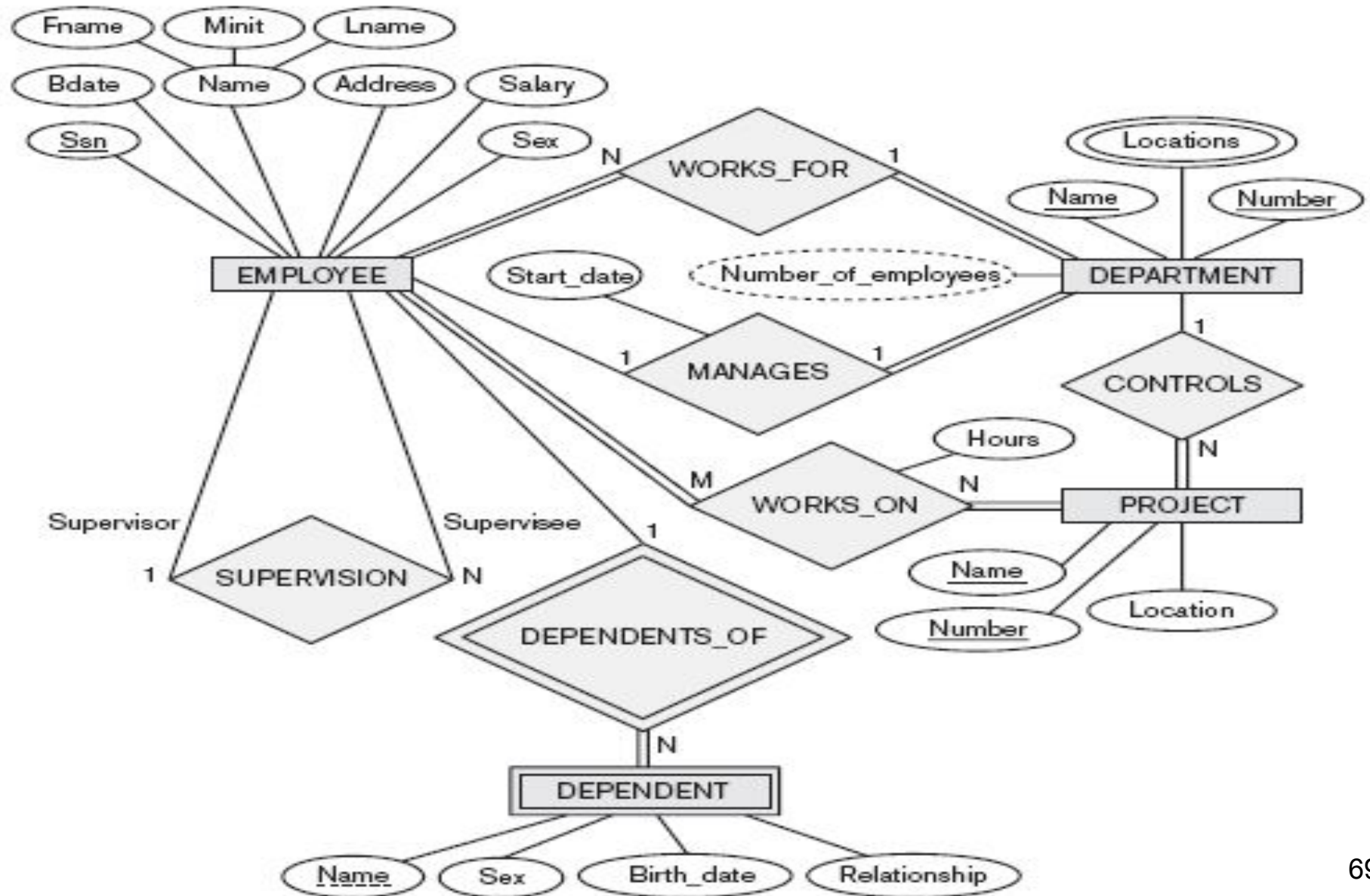
# Draw ER diagram for COMPANY DATABASE



# Draw ER diagram for COMPANY DATABASE

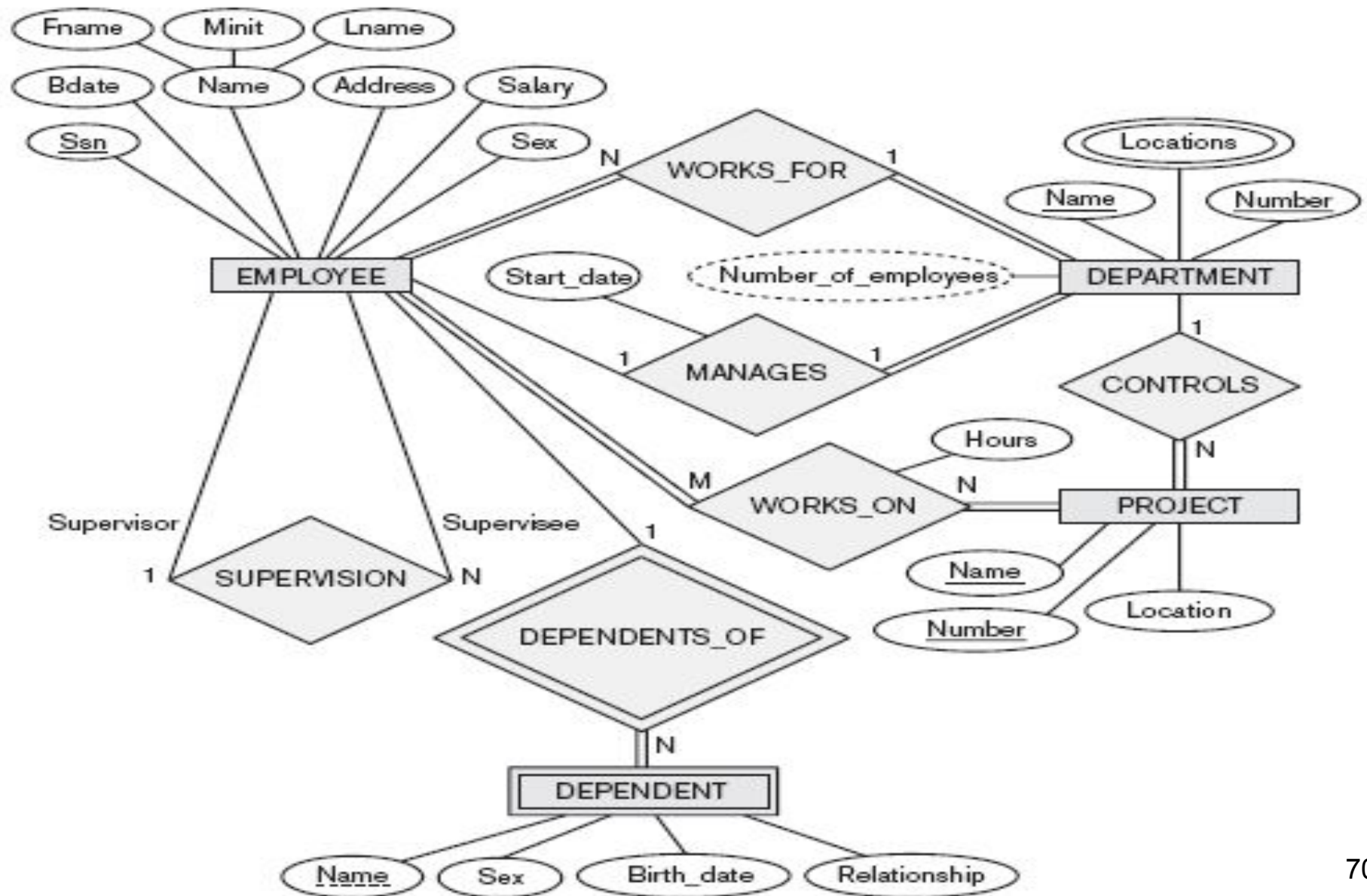


# Draw ER diagram for COMPANY DATABASE





# Mapping ER diagram for COMPANY DATABASE to Relational Model



# Mapping ER diagram for COMPANY DATABASE to Relational Model

## EMPLOYEE

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

# Mapping ER diagram for COMPANY DATABASE to Relational Model

## EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn
			▲ ▲ ▲ ▲					┆



# Mapping ER diagram for COMPANY DATABASE to Relational Model

## EMPLOYEE

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

# Mapping ER diagram for COMPANY DATABASE to Relational Model

## EMPLOYEE

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

## DEPARTMENT

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

# Mapping ER diagram for COMPANY DATABASE to Relational Model

## EMPLOYEE

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

## DEPARTMENT

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

## DEPT\_LOCATIONS

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

# Mapping ER diagram for COMPANY DATABASE to Relational Model

## 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>
----------------	------------------

# Mapping ER diagram for COMPANY DATABASE to Relational Model

## 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>	Plocation
-------	----------------	-----------

# Mapping ER diagram for COMPANY DATABASE to Relational Model

## 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>	Plocation	Dnum
-------	----------------	-----------	------

# Mapping ER diagram for COMPANY DATABASE to Relational Model

## 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>	Plocation	Dnum
-------	----------------	-----------	------

# Mapping ER diagram for COMPANY DATABASE to Relational Model

## 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>	Plocation	Dnum
-------	----------------	-----------	------

## WORKS\_ON

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



# Mapping ER diagram for COMPANY DATABASE to Relational Model

## 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>	Plocation	Dnum
-------	----------------	-----------	------

## WORKS\_ON

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

## DEPENDENT

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

# Mapping ER diagram for COMPANY DATABASE to Relational Model

## 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>	Plocation	Dnum
-------	----------------	-----------	------

## WORKS\_ON

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

## DEPENDENT

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

# Mapping ER diagram for COMPANY DATABASE to Relational Model

