

CS4.301 Data & Applications

Ponnurangam Kumaraguru ("PK")
#ProfGiri @ IIIT Hyderabad



pk.profgiri



/in/ponguru



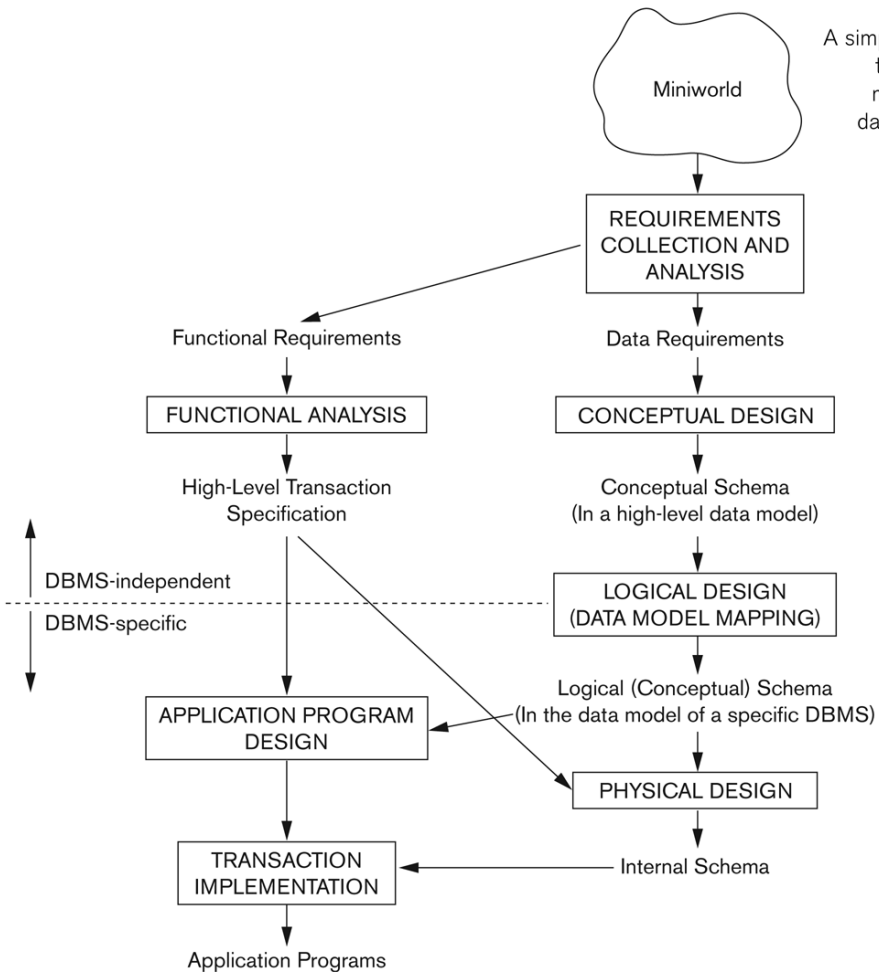
@ponguru



Ponnurangam.kumaraguru

Data Modeling Using the Entity-Relationship (ER) Model

Figure 3.1
A simplified diagram
to illustrate the
main phases of
database design.



Overview of Database Design Process

Example COMPANY Database

We need to create a database schema design based on the following (simplified) **requirements** of the COMPANY Database:

The company is organized into DEPARTMENTS. Each department has a name, number and an employee who *manages* the department. We keep track of the start date of the department manager. A department may have several locations.

Each department *controls* a number of PROJECTs. Each project has a unique name, unique number and is located at a single location.

Example COMPANY Database (Continued)

The database will store each EMPLOYEE's social security number, address, salary, gender, and birthdate.

Each employee *works for* one department but may *work on* several projects.

The DB will keep track of the number of hours per week that an employee currently works on each project.

It is required to keep track of the *direct supervisor* of each employee.

Each employee may *have* a number of DEPENDENTS.

For each dependent, the DB keeps a record of name, gender, birthdate, and relationship to the employee.

ER Model Concepts

Entities and Attributes

Entity is a basic concept for the ER model. Entities are specific things or objects in the mini-world that are represented in the database.

For example the EMPLOYEE John Smith, the Research DEPARTMENT, the ProductX PROJECT

Attributes are properties used to describe an entity.

For example an EMPLOYEE entity may have the attributes Name, SSN, Address, gender, BirthDate

A specific entity will have a value for each of its attributes.

For example a specific employee entity may have Name='John Smith', SSN='123456789', Address='731, Fondren, Houston, TX', gender='M', BirthDate='09-JAN-55'

Each attribute has a *value set* (or data type) associated with it – e.g. integer, string, date, enumerated type, ...

Types of Attributes (1)

Simple

Each entity has a single atomic value for the attribute. For example, SSN or gender.

Composite

The attribute may be composed of several components. For example:

Address (Apt#, House#, Street, City, State, ZipCode, Country), or

Name (FirstName, MiddleName, LastName).

Multi-valued

Multiple values for the attribute. For example, Color of a CAR or PreviousDegrees of a STUDENT.

Denoted as {Color} or {PreviousDegrees}.

Example of a composite attribute

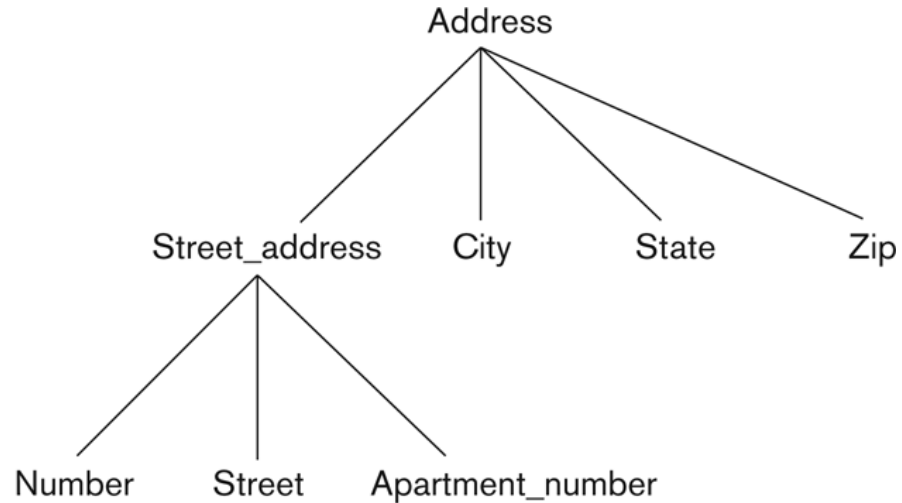


Figure 3.4

A hierarchy of composite attributes.

Value Sets (Domains) of Attributes


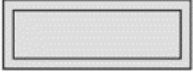
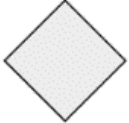




Each simple attribute is associated with a value set

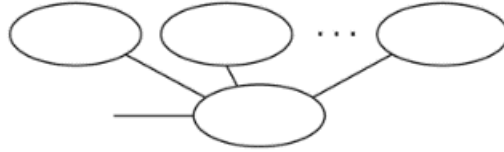
E.g., Lastname has a value which is a character string of upto 15 characters, say

Date has a value consisting of MM-DD-YYYY where each letter is an integer

A **value set** specifies the set of values associated with an attribute

Figure 3.14
Summary of the
notation for ER
diagrams.

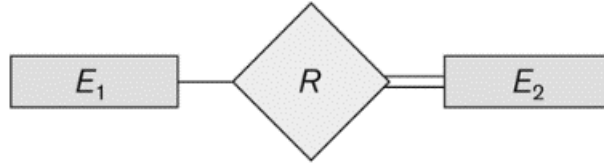
Symbol	Meaning
	Entity
	Weak Entity
	Relationship
	Identifying Relationship
	Attribute
	Key Attribute
	Multivalued Attribute



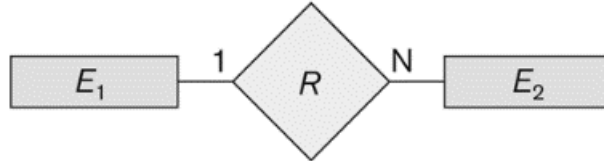
Composite Attribute



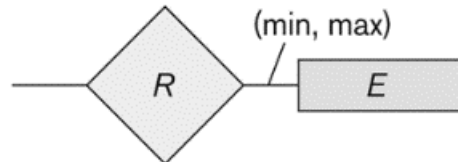
Derived Attribute



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 Type CAR with two keys and a corresponding Entity Set

(a)

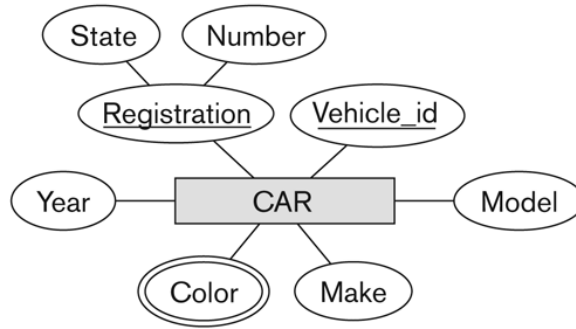


Figure 3.7

The CAR entity type with two key attributes, Registration and Vehicle_id. (a) ER diagram notation. (b) Entity set with three entities.

(b)

CAR
Registration (Number, State), Vehicle_id, Make, Model, Year, {Color}

CAR₁
((ABC 123, TEXAS), TK629, Ford Mustang, convertible, 2004 {red, black})

CAR₂
((ABC 123, NEW YORK), WP9872, Nissan Maxima, 4-door, 2005, {blue})

CAR₃
((VSY 720, TEXAS), TD729, Chrysler LeBaron, 4-door, 2002, {white, blue})

⋮

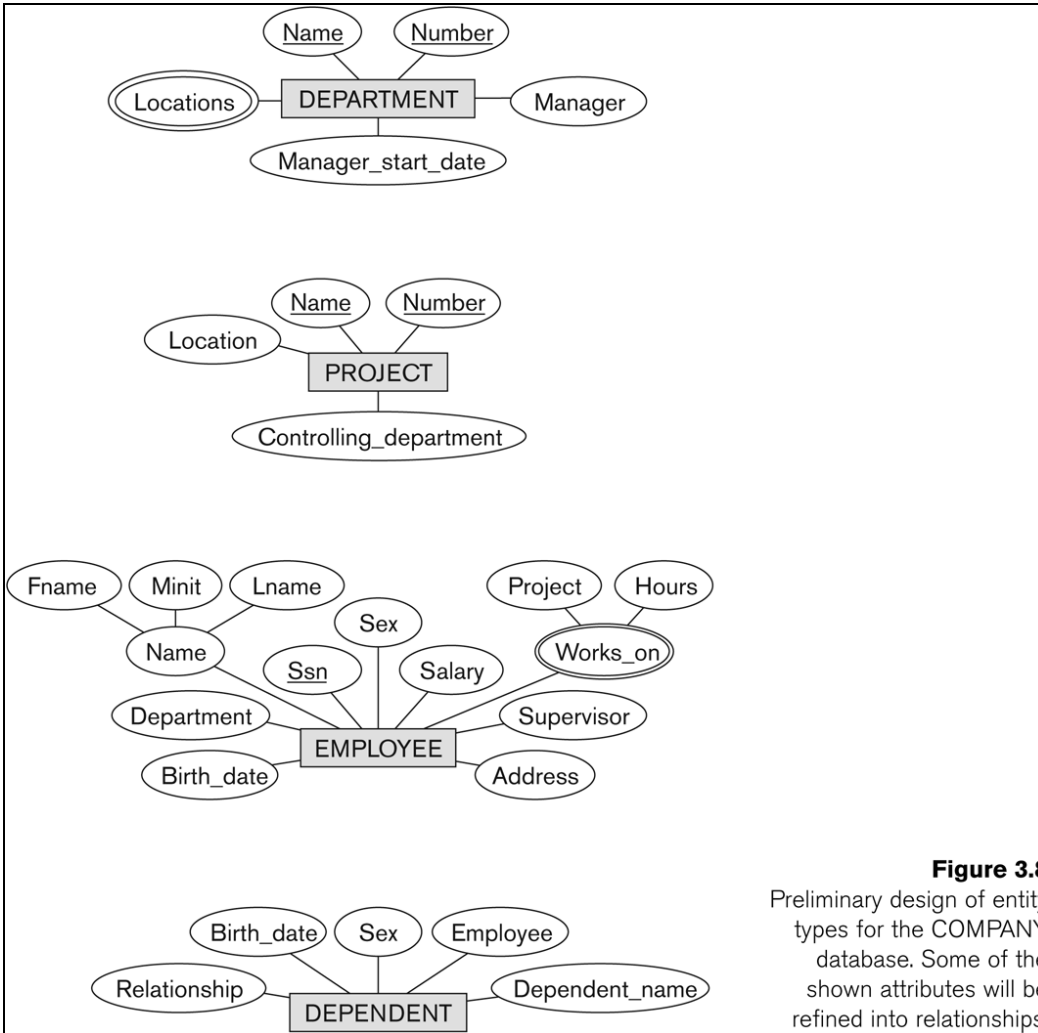


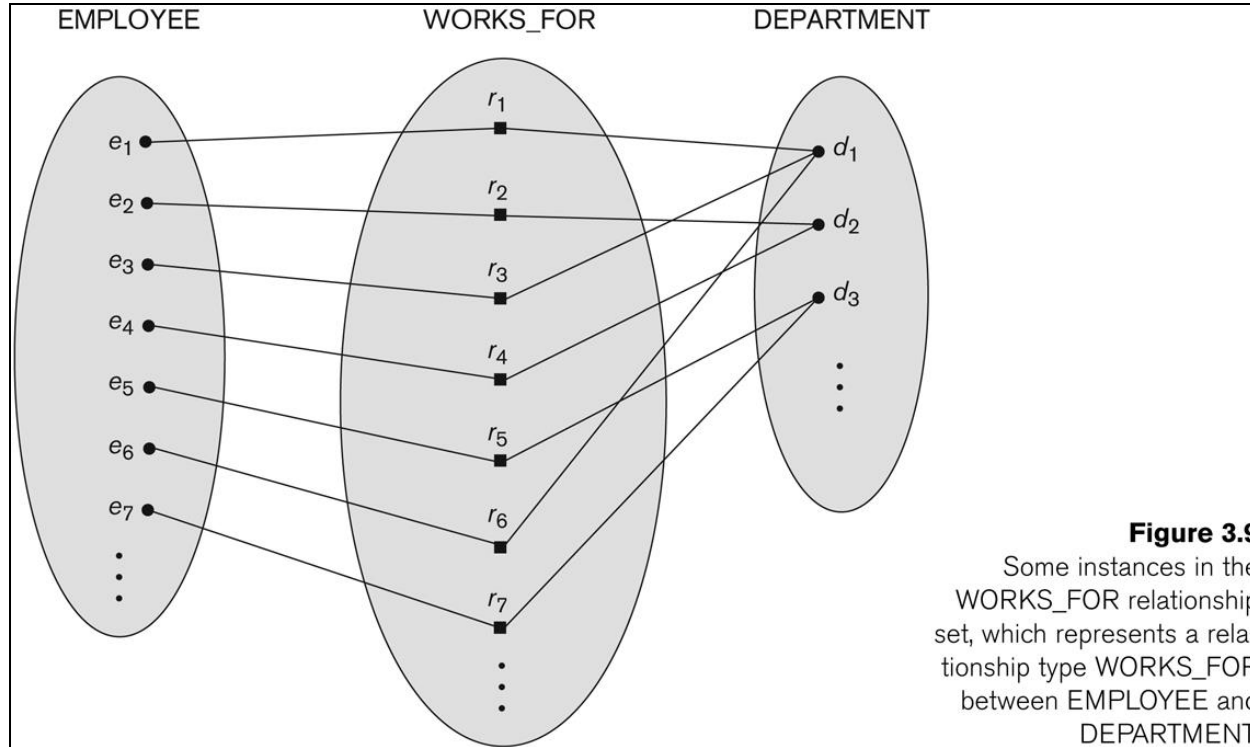
Figure 3.8

Preliminary design of entity types for the COMPANY database. Some of the shown attributes will be refined into relationships.

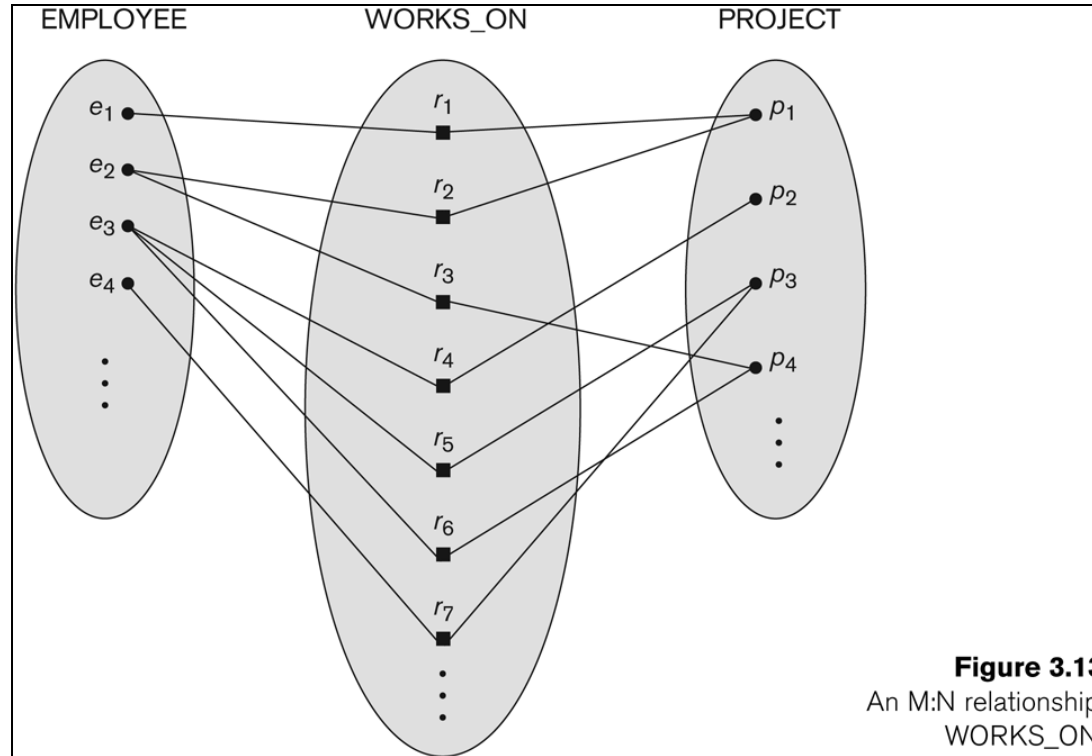
Initial Design of Entity Types:

EMPLOYEE,
DEPARTMENT, PROJECT,
DEPENDENT

Relationship instances of the WORKS_FOR N:1 relationship between EMPLOYEE and DEPARTMENT



Relationship instances of the M:N WORKS_ON relationship between EMPLOYEE and PROJECT



Refining the COMPANY database schema by introducing relationships

By examining the requirements, six relationship types are identified

All are *binary* relationships (degree 2)

Listed below with their participating entity types:

- WORKS_FOR (between EMPLOYEE, DEPARTMENT)

- MANAGES (also between EMPLOYEE, DEPARTMENT)

- CONTROLS (between DEPARTMENT, PROJECT)

- WORKS_ON (between EMPLOYEE, PROJECT)

- SUPERVISION (between EMPLOYEE (as subordinate), EMPLOYEE (as supervisor))

- DEPENDENTS_OF (between EMPLOYEE, DEPENDENT)

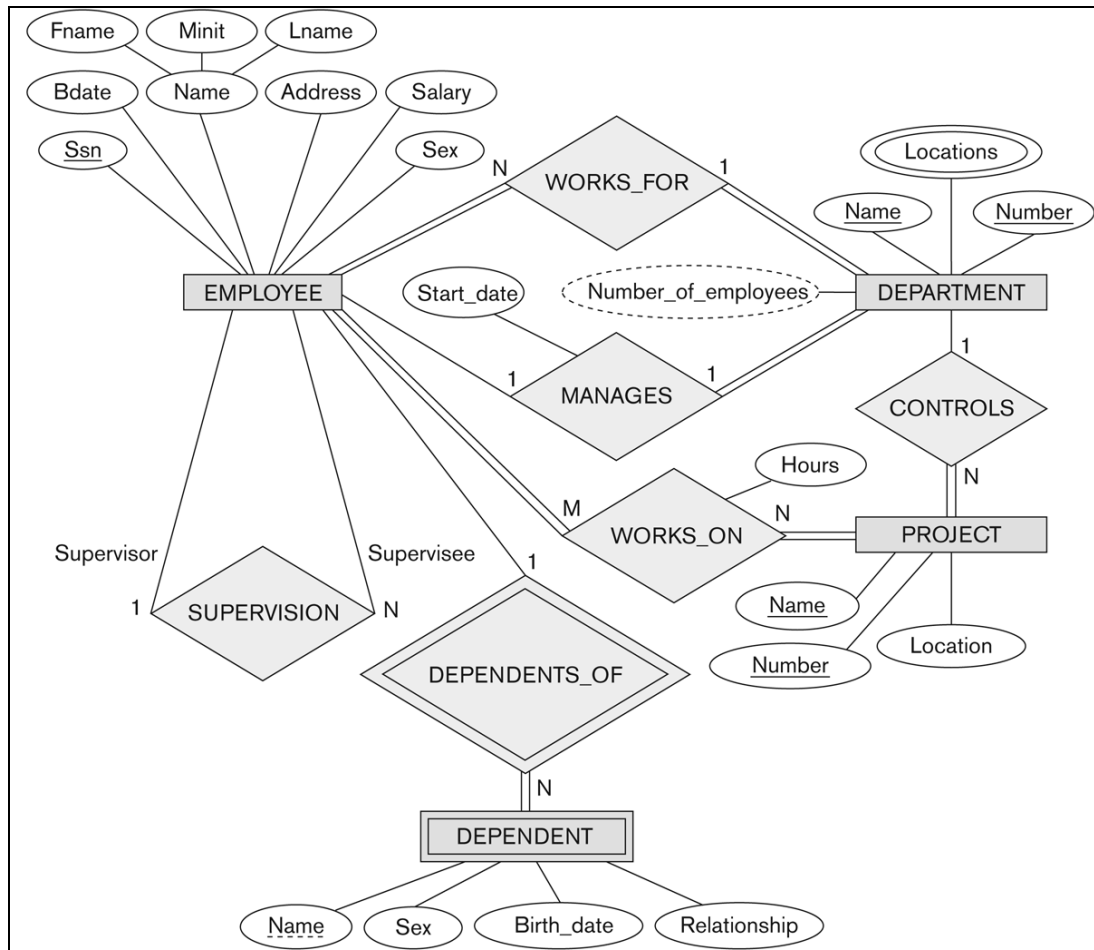


Figure 3.2

An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter.

ER DIAGRAM –
Relationship Types
are:

WORKS_FOR, MANAGES, WORKS_ON,
CONTROLS, SUPERVISION, DEPENDENTS_OF

Activity

Create requirements for Infinium

Have at least 4 Entities, and develop ER diagram for the same

Be creative in terms of attributes, relationships, and entities!

This Lecture

Discussion on Relationship Types

In the refined design, some attributes from the initial entity types are refined into relationships:

- Manager of DEPARTMENT -> MANAGES

- Works_on of EMPLOYEE -> WORKS_ON

- Department of EMPLOYEE -> WORKS_FOR

- etc

In general, more than one relationship type can exist between the same participating entity types

- MANAGES and WORKS_FOR are distinct relationship types between EMPLOYEE and DEPARTMENT

- Different meanings and different relationship instances

Constraints on Relationships

Constraints on Relationship Types

Also known as ratio constraints

Cardinality Ratio (specifies *maximum* participation)

One-to-one (1:1)

One-to-many (1:N) or Many-to-one (N:1)

Many-to-many (M:N)

Existence Dependency Constraint (specifies *minimum* participation) (also called participation constraint)

zero (optional participation, not existence-dependent)

one or more (mandatory participation, existence-dependent)

Term	Meaning	Example
Existence-dependent	One entity <i>cannot exist</i> without another	<i>Course Section</i> → <i>Course</i>
Not existence-dependent	Entity <i>can exist</i> independently	<i>Professor</i> → <i>Department</i>

Many-to-one (N:1) Relationship

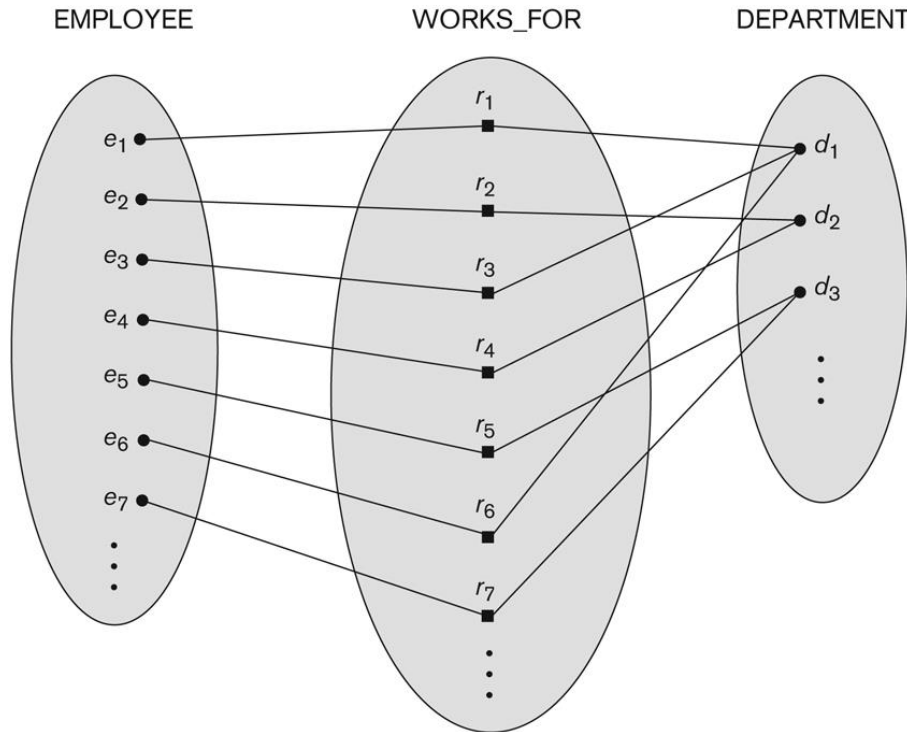


Figure 3.9

Some instances in the WORKS_FOR relationship set, which represents a relationship type WORKS_FOR between EMPLOYEE and DEPARTMENT.

Many-to-many (M:N) Relationship

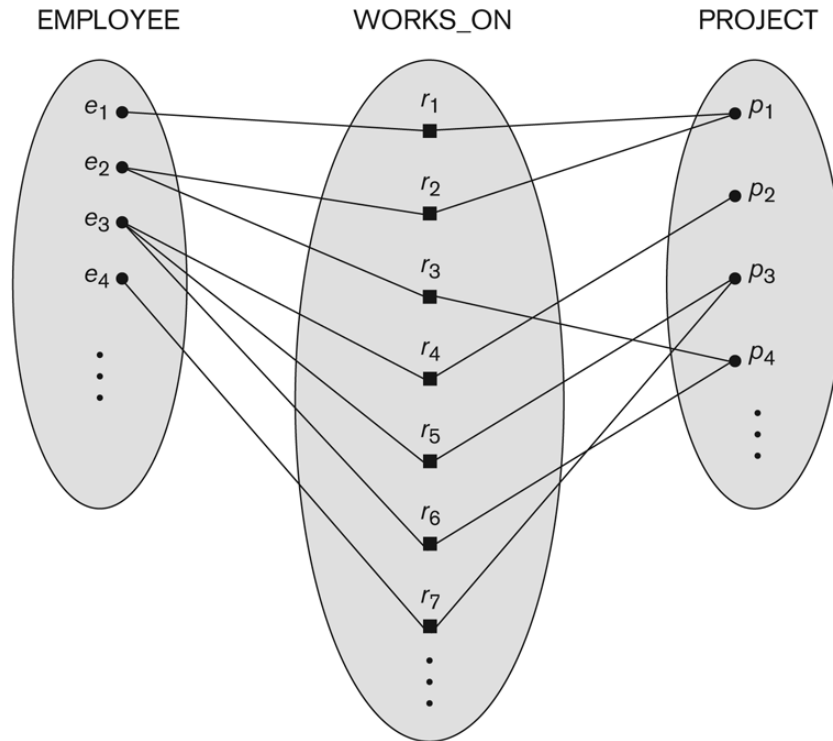


Figure 3.13
An M:N relationship,
WORKS_ON.

Recursive Relationship Type

A relationship type between the same participating entity type in **distinct roles**

Also called a **self-referencing** relationship type.

Example: the SUPERVISION relationship

EMPLOYEE participates twice in two distinct roles:

- supervisor (or boss) role

- supervisee (or subordinate) role

Each relationship instance relates two distinct EMPLOYEE entities:

- One employee in *supervisor* role

- One employee in *supervisee* role

Displaying a recursive relationship

In a recursive relationship type.

Both participations are same entity type in different roles.

For example, SUPERVISION relationships between EMPLOYEE (in role of supervisor or boss) and (another) EMPLOYEE (in role of subordinate or worker).

In following figure, first role participation labeled with 1 and second role participation labeled with 2.

In ER diagram, need to display role names to distinguish participations.

A Recursive Relationship Supervision`

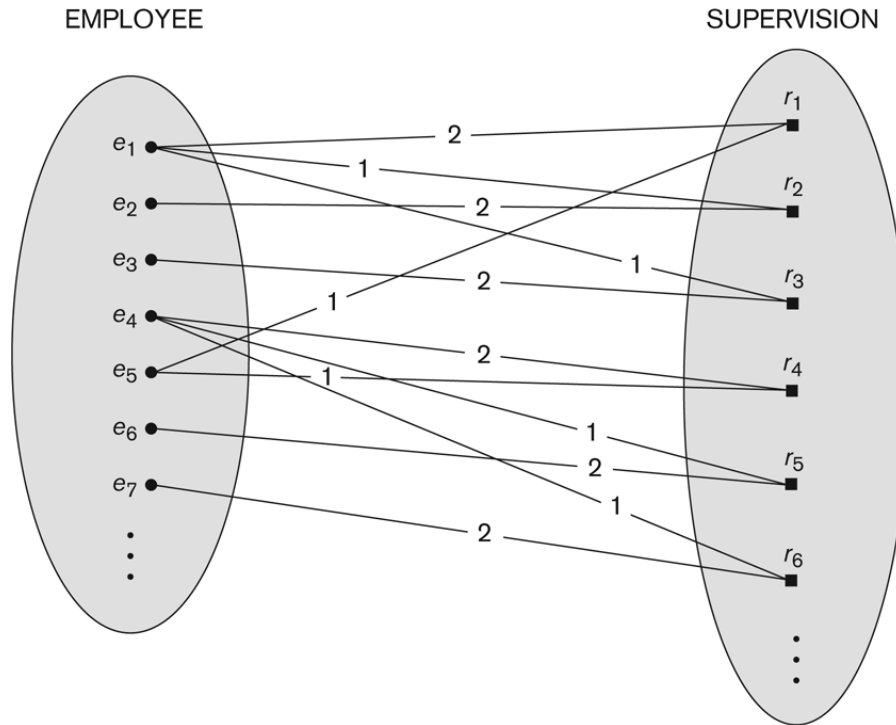
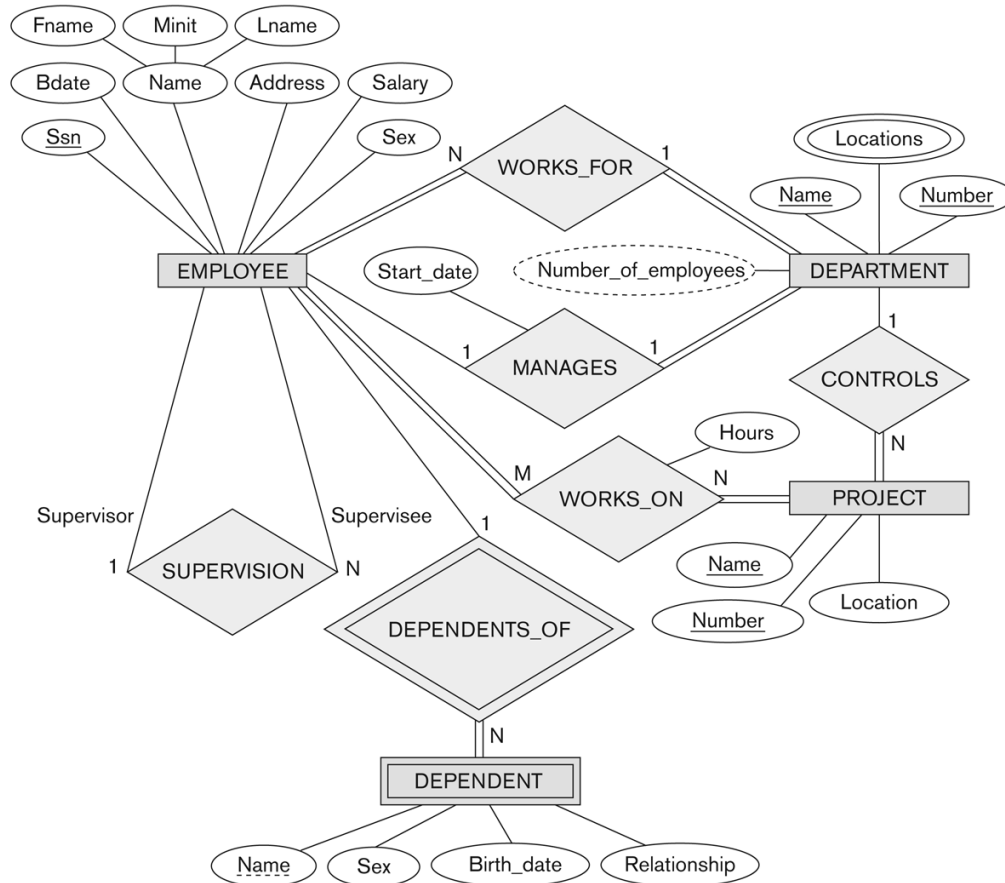


Figure 3.11

A recursive relationship SUPERVISION between EMPLOYEE in the *supervisor* role (1) and EMPLOYEE in the *subordinate* role (2).



Recursive
Relationship
Type is:
SUPERVISION
(participation
role names are
shown)

Figure 3.2

An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter.

Weak Entity Types

An entity that does not have a key attribute and that is identification-dependent on another entity type.

A weak entity must participate in an identifying relationship type with an owner or identifying entity type

Entities are identified by the combination of:

- A partial key of the weak entity type

- The particular entity they are related to in the identifying relationship type

Example:

A DEPENDENT entity is identified by the dependent's first name, *and* the specific EMPLOYEE with whom the dependent is related

Name of DEPENDENT is the *partial key*

DEPENDENT is a *weak entity type*

EMPLOYEE is its identifying entity type via the identifying relationship type DEPENDENT_OF

Attributes of Relationship types

A relationship type can have attributes:

For example, HoursPerWeek of WORKS_ON

Its value for each relationship instance describes the number of hours per week that an EMPLOYEE works on a PROJECT.

A value of HoursPerWeek depends on a particular (employee, project) combination

Most relationship attributes are used with M:N relationships

Example Attribute of a Relationship Type: Hours of WORKS_ON

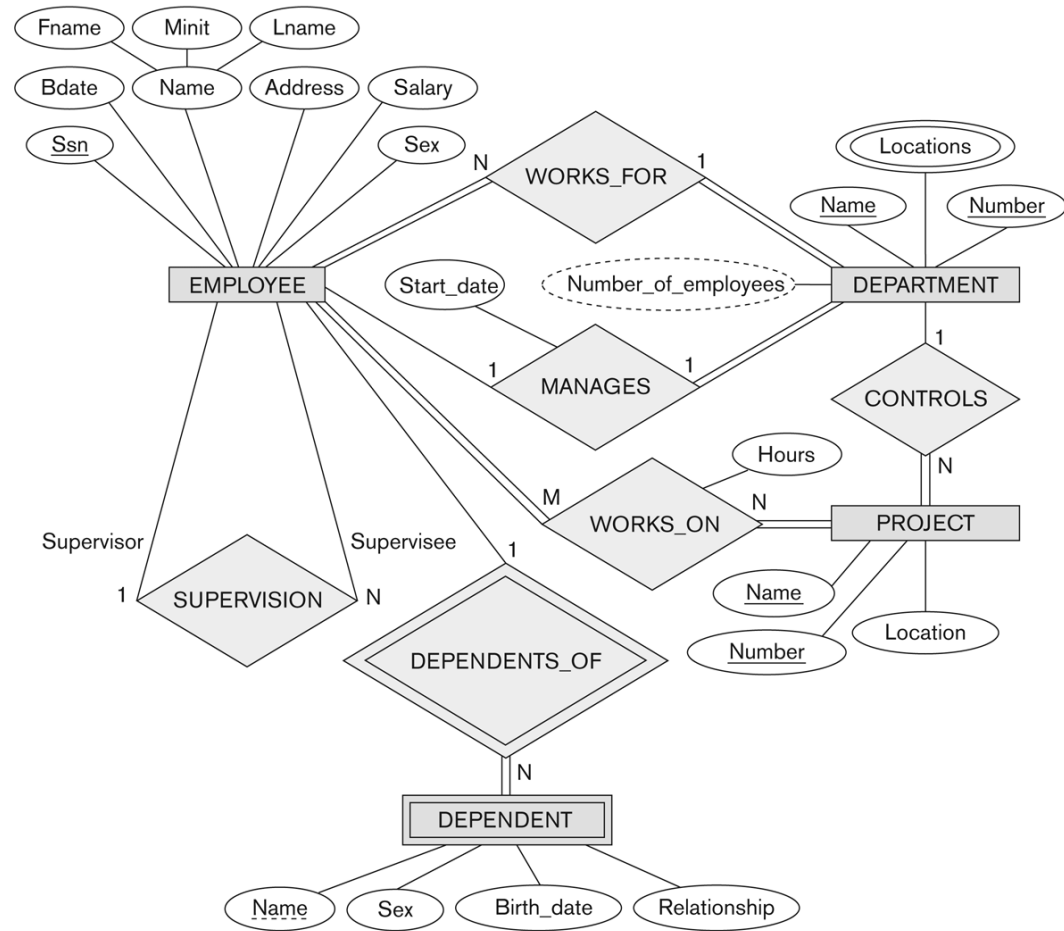


Figure 3.2

An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter.

Bibliography / Acknowledgements

Instructor materials from Elmasri & Navathe 7e

 pk.profgiri

 Ponnurangam.kumaraguru

 /in/ponguru

 ponguru

 pk.guru@iiit.ac.in

Thank you
for attending
the class!!!