



APPLIED DATABASE MANAGEMENT SYSTEM BCSC0014



Dr. Neeraj Gupta
Assistant Professor, Dept. of CEA
neeraj.gupta@gla.ac.in



OUTLINE

- ❑ Overview of Database Design Process
- ❑ Example Database Application (COMPANY)
- ❑ ER Model Concepts
 - ❑ Entities and Attributes
 - ❑ Entity Types, Value Sets, and Key Attributes
 - ❑ Relationships and Relationship Types
 - ❑ Weak Entity Types
 - ❑ Roles and Attributes in Relationship Types
- ❑ ER Diagrams - Notation
- ❑ ER Diagram for COMPANY Schema



OVERVIEW OF DATABASE DESIGN PROCESS

Two main activities:

- Database design
- Applications design

To design the conceptual schema for a database application

Applications design focuses on the programs and interfaces that access the database

- Generally considered part of software engineering

OVERVIEW OF DATABASE DESIGN PROCESS

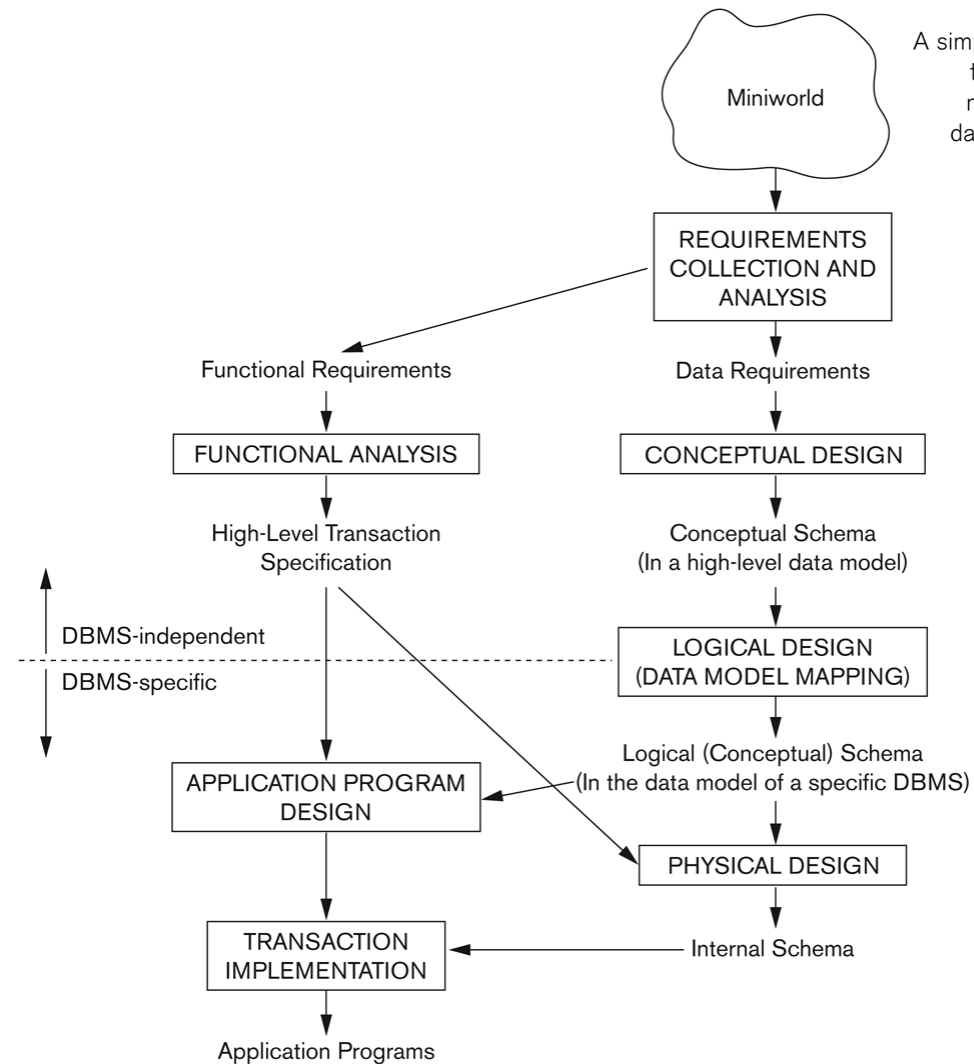


Figure 3.1
A simplified diagram
to illustrate the
main phases of
database design.

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 (CONTD.)

- We store each EMPLOYEE's social security number, address, salary, sex, and birthdate.
 - Each employee *works for* one department but may *work on* several projects.
 - We keep track of the number of hours per week that an employee currently works on each project.
 - We also keep track of the *direct supervisor* of each employee.
- Each employee may *have* a number of DEPENDENTS.
 - For each dependent, we keep track of their name, sex, birthdate, and relationship to the employee.

ER MODEL CONCEPTS

Entities and Attributes

- Entities are specific objects or things 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, Sex, 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='1 23456789', Address ='731, Fondren, Houston, TX', Sex='M', BirthDate='09-JAN-55'
- Each attribute has a *value set* (or data type) associated with it – e.g. integer, string, subrange, enumerated type, ...

TYPES OF ATTRIBUTES (1)

Simple

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

Composite

- The attribute may be composed of several components. For example:
 - Address(Apt#, House#, Street, City, State, ZipCode, Country), or
 - Name(FirstName, MiddleName, LastName).
 - Composition may form a hierarchy where some components are themselves composite.

Multi-valued

- An entity may have multiple values for that attribute. For example, Color of a CAR or PreviousDegrees of a STUDENT.
 - Denoted as {Color} or {PreviousDegrees}.

TYPES OF ATTRIBUTES (2)

In general, composite and multi-valued attributes may be nested arbitrarily to any number of levels, although this is rare.

- For example, PreviousDegrees of a STUDENT is a composite multi-valued attribute denoted by {PreviousDegrees (College, Year, Degree, Field)}
- Multiple PreviousDegrees values can exist
- Each has four subcomponent attributes:
 - College, Year, Degree, Field

EXAMPLE OF A COMPOSITE ATTRIBUTE

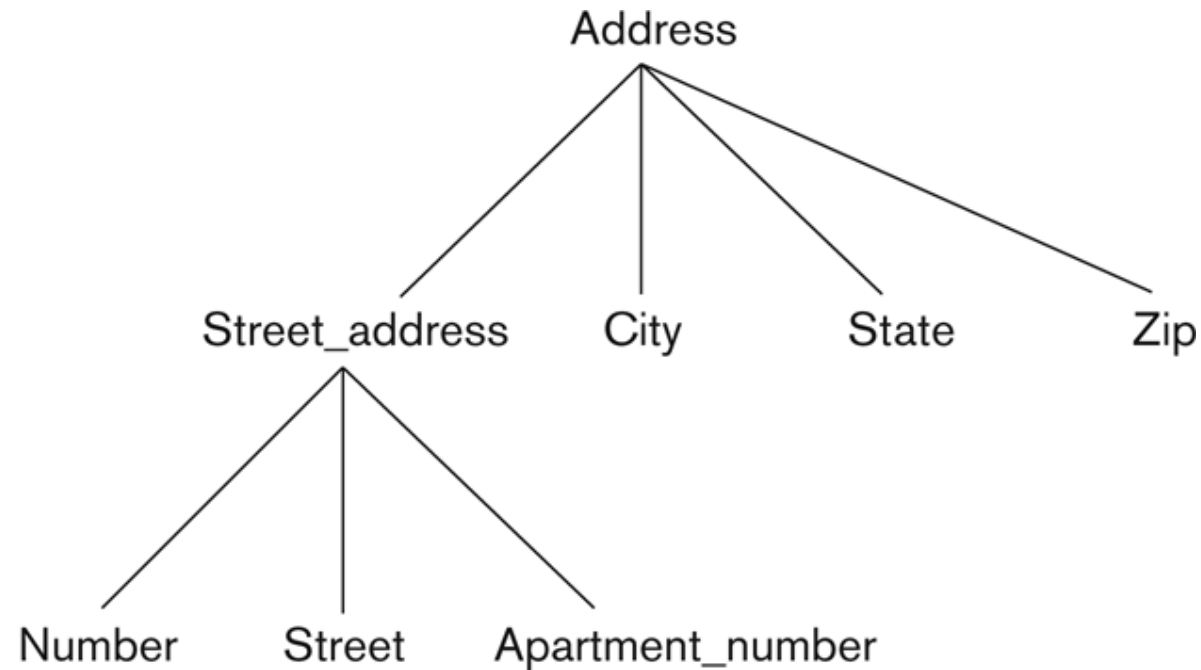


Figure 3.4

A hierarchy of composite attributes.

ENTITY TYPES AND KEY ATTRIBUTES (1)

Entities with the same basic attributes are grouped or typed into an entity type.

- For example, the entity type EMPLOYEE and PROJECT.

An attribute of an entity type for which each entity must have a unique value is called a key attribute of the entity type.

- For example, SSN of EMPLOYEE.

ENTITY TYPES AND KEY ATTRIBUTES (2)

A key attribute may be composite.

- VehicleTagNumber is a key of the CAR entity type with components (Number, State).

An entity type may have more than one key.

- The CAR entity type may have two keys:
 - VehicleIdentificationNumber (popularly called VIN)
 - VehicleTagNumber (Number, State), aka license plate number.

Each key is underlined

DISPLAYING AN ENTITY TYPE

In ER diagrams, an entity type is displayed in a rectangular box

Attributes are displayed in ovals

- Each attribute is connected to its entity type
- Components of a composite attribute are connected to the oval representing the composite attribute
- Each key attribute is underlined
- Multivalued attributes displayed in double ovals

See CAR example on next slide

ENTITY TYPE CAR WITH TWO KEYS AND A CORRESPONDING ENTITY SET

(a)

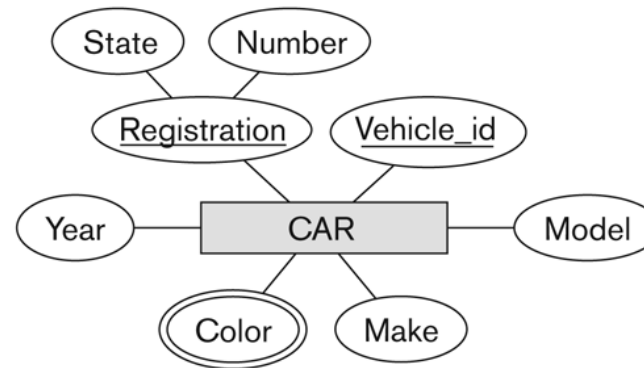


Figure 3.7

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})

⋮

ENTITY SET

Each entity type will have a collection of entities stored in the database

- Called the **entity set**

Previous slide shows three CAR entity instances in the entity set for CAR

Same name (CAR) used to refer to both the entity type and the entity set

Entity set is the current *state* of the entities of that type that are stored in the database

INITIAL DESIGN OF ENTITY TYPES FOR THE COMPANY DATABASE SCHEMA

Based on the requirements, we can identify four initial entity types in the COMPANY database:

- DEPARTMENT
- PROJECT
- EMPLOYEE
- DEPENDENT

Their initial design is shown on the following slide

The initial attributes shown are derived from the requirements description

EMPLOYEE, DEPARTMENT, PROJECT, DEPENDENT

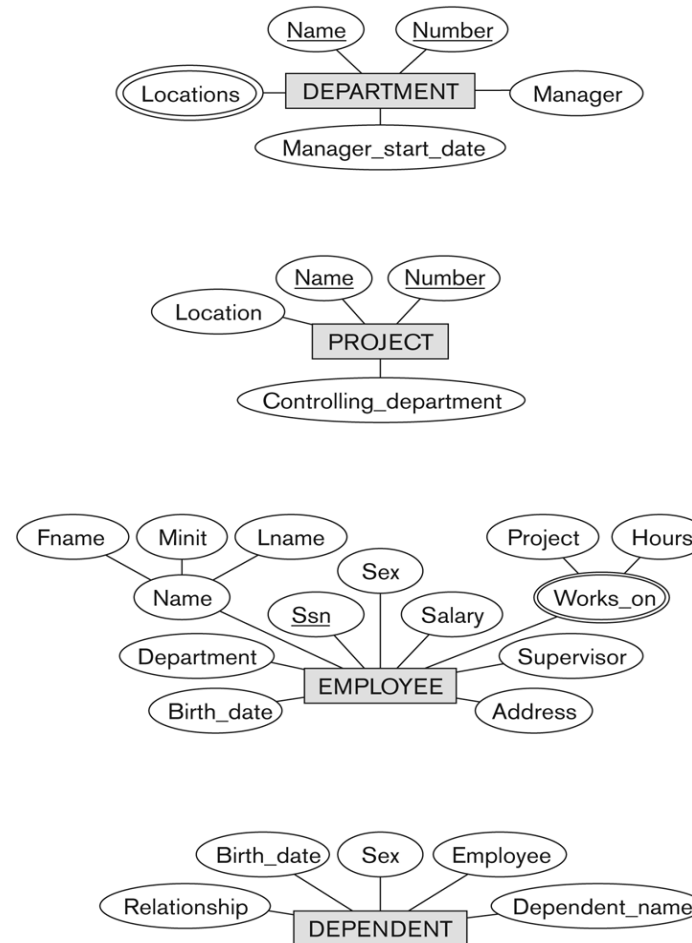


Figure 3.8

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

REFINING THE INITIAL DESIGN BY INTRODUCING **RELATIONSHIPS**

The initial design is typically not complete

Some aspects in the requirements will be represented as **relationships**

ER model has three main concepts:

- Entities (and their entity types and entity sets)
- Attributes (simple, composite, multivalued)
- Relationships (and their relationship types and relationship sets)

We introduce relationship concepts next

RELATIONSHIPS AND RELATIONSHIP TYPES (1)

A **relationship** relates two or more distinct entities with a specific meaning.

- For example, EMPLOYEE John Smith *works on* the ProductX PROJECT, or EMPLOYEE Franklin Wong *manages* the Research DEPARTMENT.

Relationships of the same type are grouped or typed into a **relationship type**.

- For example, the WORKS_ON relationship type in which EMPLOYEEs and PROJECTs participate, or the MANAGES relationship type in which EMPLOYEEs and DEPARTMENTs participate.

The degree of a relationship type is the number of participating entity types.

- Both MANAGES and WORKS_ON are *binary* relationships.

RELATIONSHIP INSTANCES OF THE WORKS_FOR N:1 RELATIONSHIP BETWEEN EMPLOYEE AND DEPARTMENT

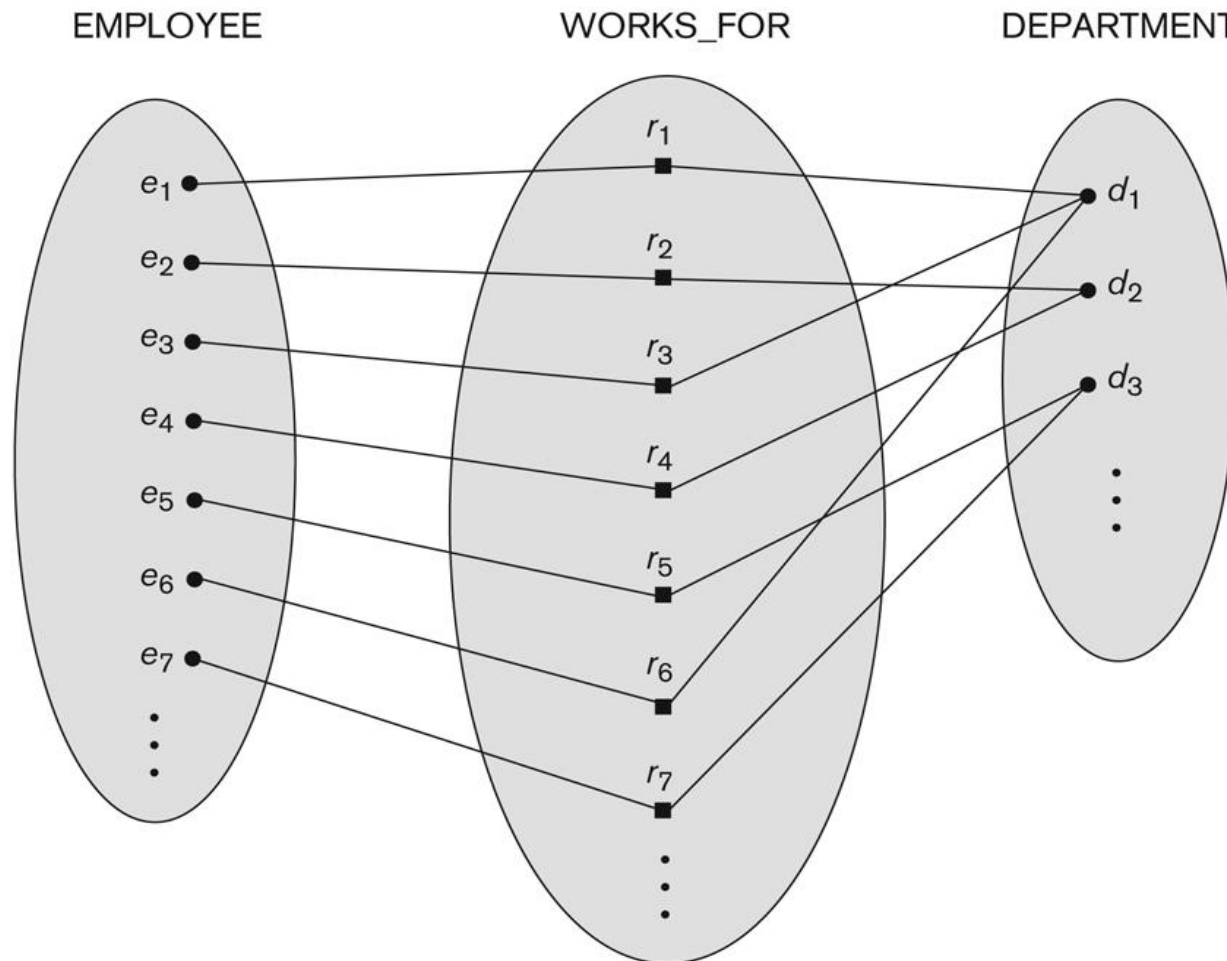


Figure 3.9

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

RELATIONSHIP INSTANCES OF THE M:N WORKS_ON RELATIONSHIP BETWEEN EMPLOYEE AND PROJECT

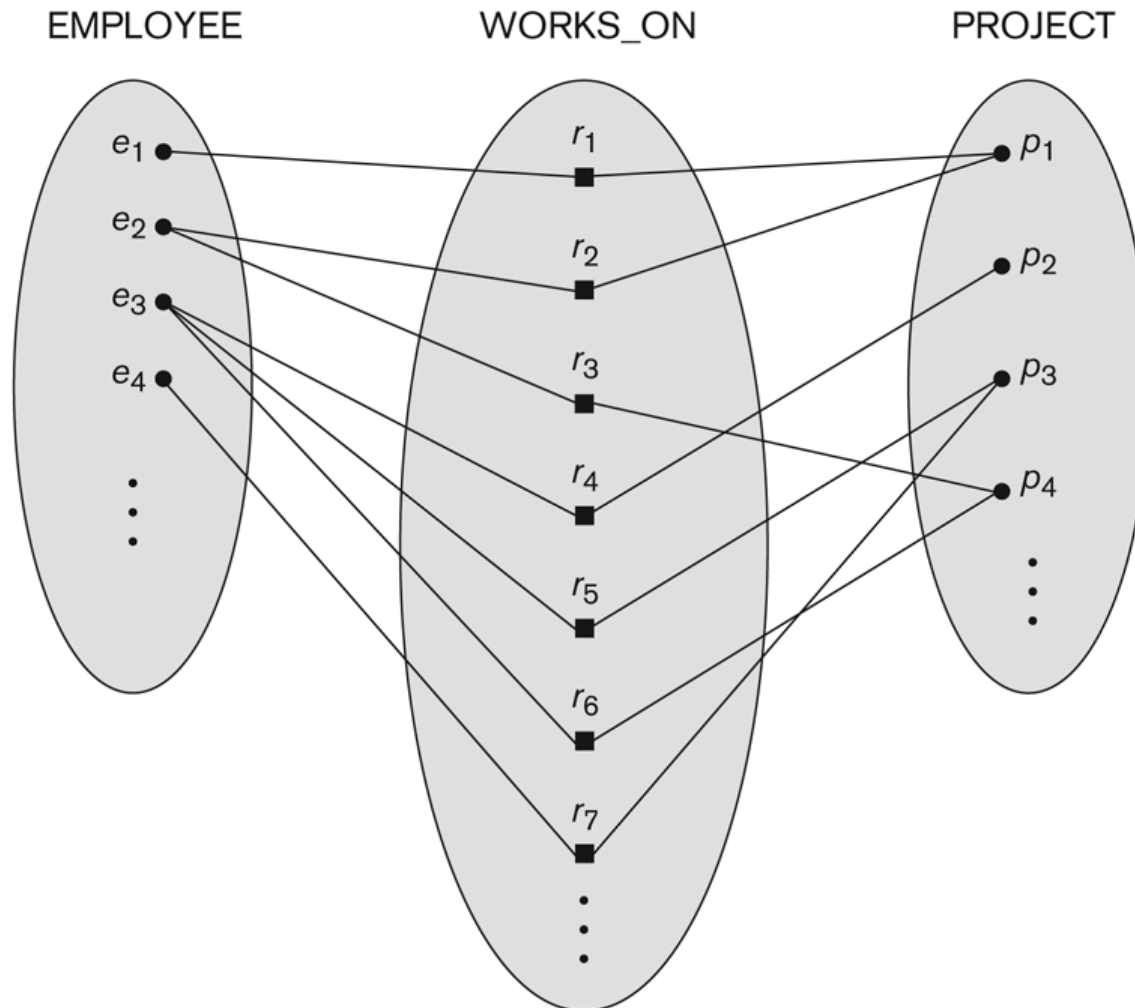


Figure 3.13
An M:N relationship,
WORKS_ON.

RELATIONSHIP TYPE VS. RELATIONSHIP SET (1)

Relationship Type:

- Is the schema description of a relationship
- Identifies the relationship name and the participating entity types
- Also identifies certain relationship constraints

Relationship Set:

- The current set of relationship instances represented in the database
- The current *state* of a relationship type

RELATIONSHIP TYPE VS. RELATIONSHIP SET (2)

Previous figures displayed the relationship sets

Each instance in the set relates individual participating entities – one from each participating entity type

In ER diagrams, we represent the *relationship type* as follows:

- Diamond-shaped box is used to display a relationship type
- Connected to the participating entity types via straight lines

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)

ER DIAGRAM — RELATIONSHIP TYPES ARE:

WORKS_FOR, MANAGES, WORKS_ON, CONTROLS, SUPERVISION, DEPENDENTS_OF

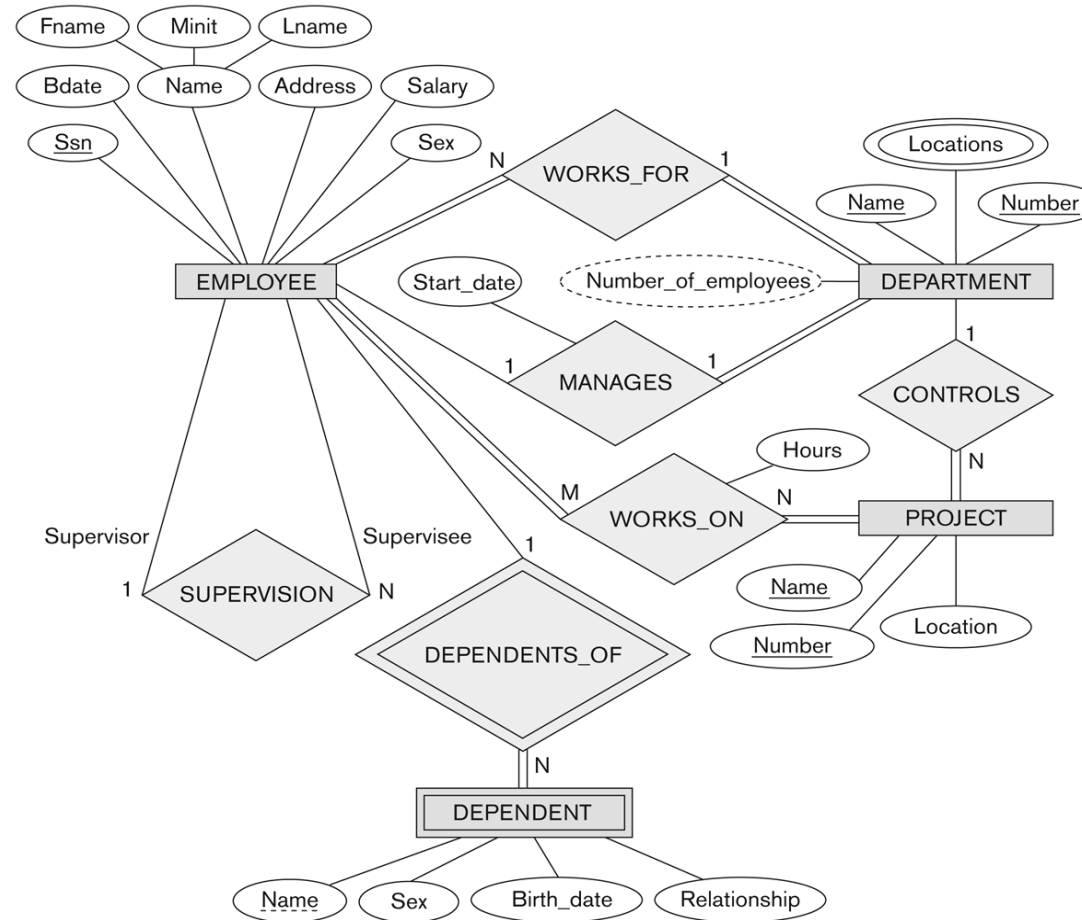


Figure 3.2
An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter.

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.

RECURSIVE RELATIONSHIP TYPE

An relationship type whose with the same participating entity type in **distinct roles**

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

WEAK ENTITY TYPES

An entity that does not have a key attribute

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 entity 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

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)

MANY-TO-ONE (N:1) RELATIONSHIP

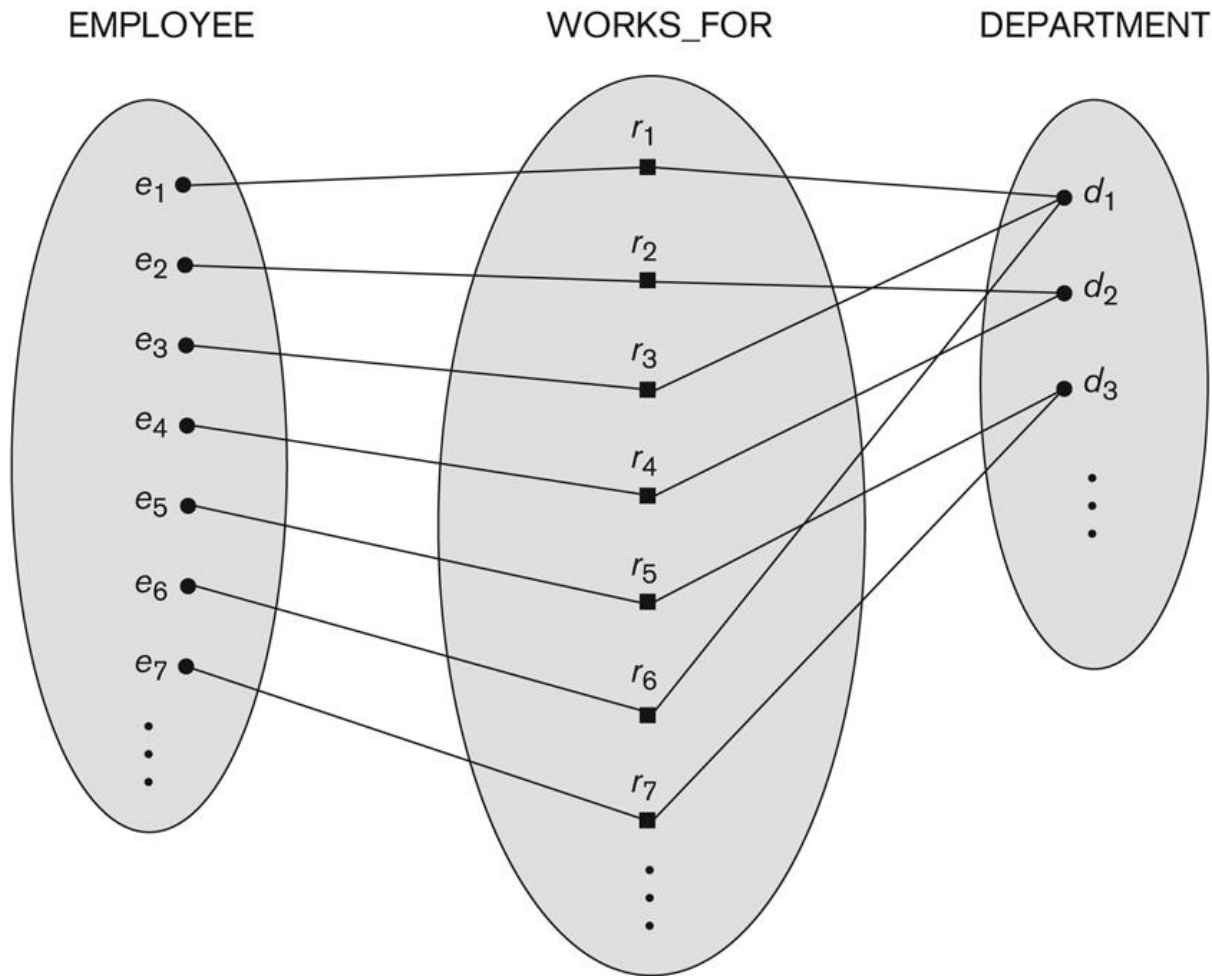


Figure 3.9

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

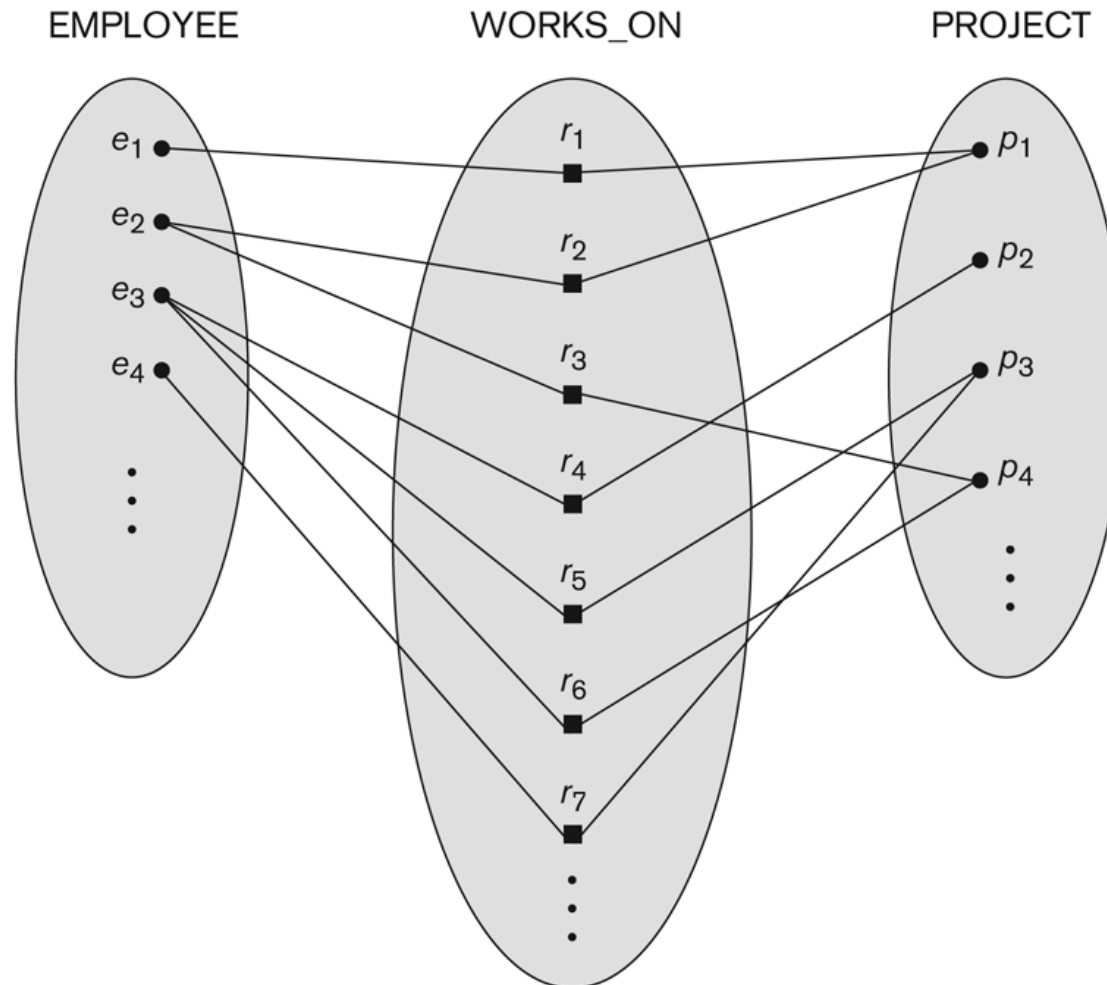


Figure 3.13
An M:N relationship,
WORKS_ON.

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`

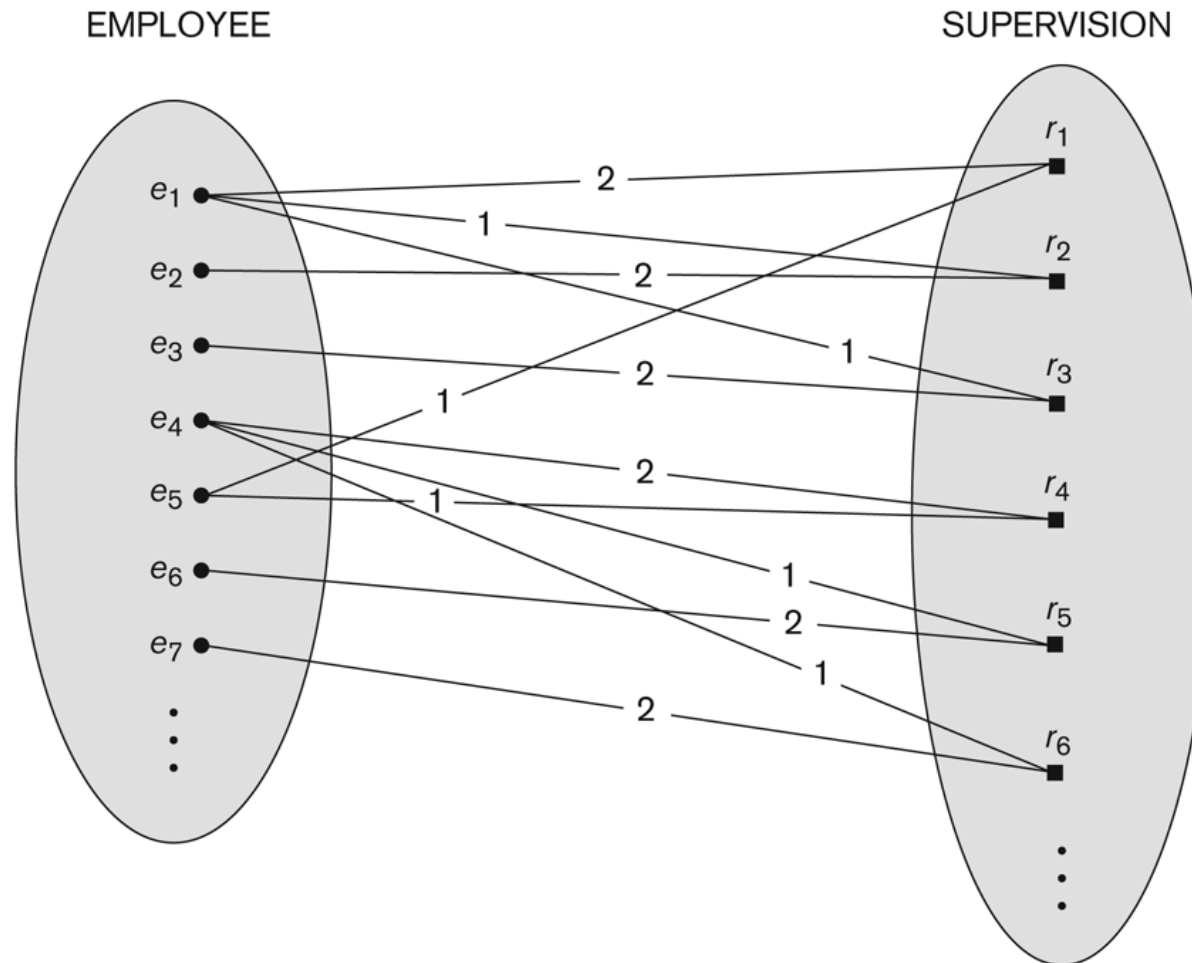


Figure 3.11
A recursive relationship SUPERVISION between EMPLOYEE in the *supervisor* role (1) and EMPLOYEE in the *subordinate* role (2).

RECURSIVE RELATIONSHIP TYPE IS: SUPERVISION (PARTICIPATION ROLE NAMES ARE SHOWN)

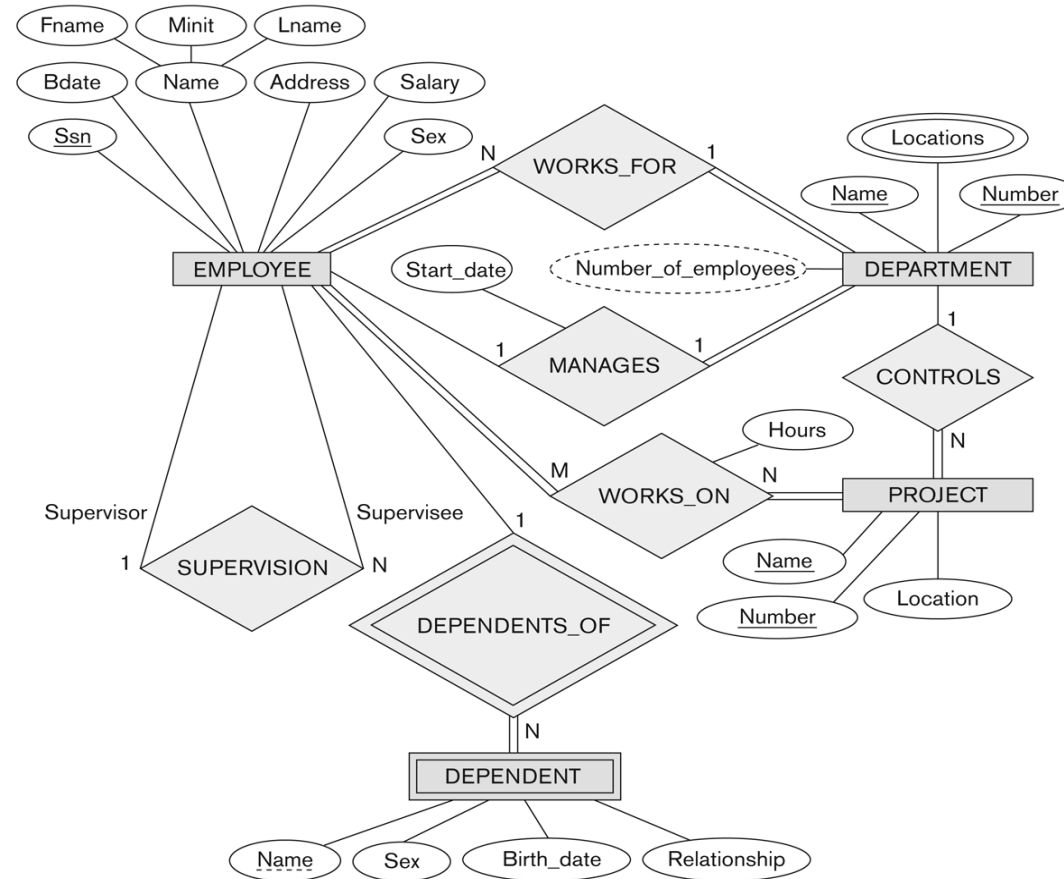


Figure 3.2

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

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
 - In 1:N relationships, they can be transferred to the entity type on the N-side of the relationship

EXAMPLE ATTRIBUTE OF A RELATIONSHIP TYPE: HOURS OF WORKS_ON

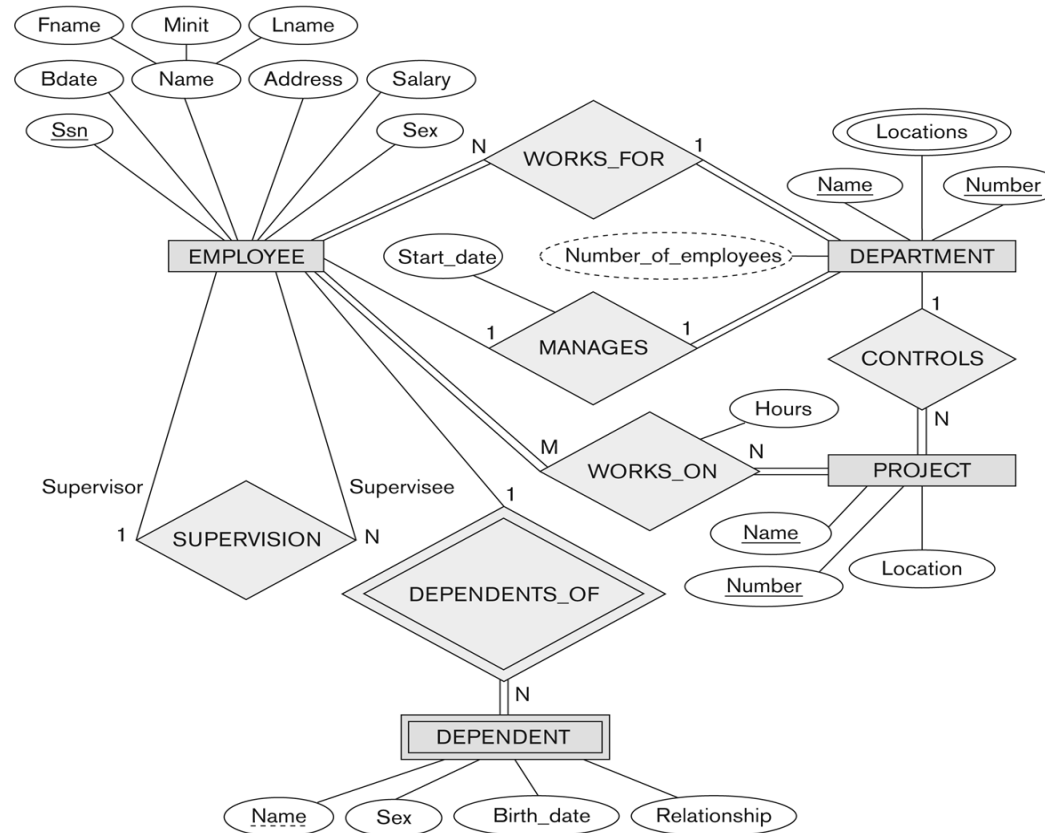


Figure 3.2

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

NOTATION FOR CONSTRAINTS ON RELATIONSHIPS

Cardinality ratio (of a binary relationship): 1:1, 1:N, N:1, or M:N

- Shown by placing appropriate numbers on the relationship edges.

Participation constraint (on each participating entity type): total (called existence dependency) or partial.

- Total shown by double line, partial by single line.

NOTE: These are easy to specify for Binary Relationship Types.

ALTERNATIVE (MIN, MAX) NOTATION FOR RELATIONSHIP STRUCTURAL CONSTRAINTS:

Specified on each participation of an entity type E in a relationship type R

Specifies that each entity e in E participates in at least *min* and at most *max* relationship instances in R

Default(no constraint): $\text{min}=0$, $\text{max}=\infty$ (signifying no limit)

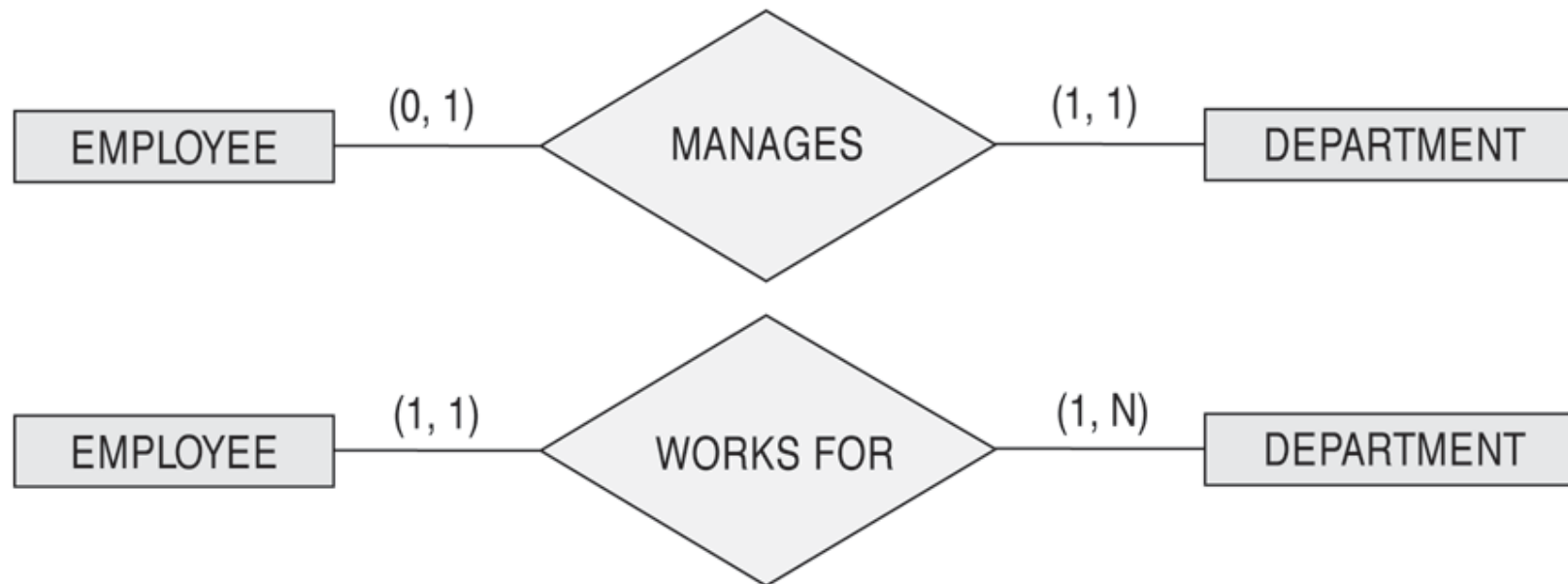
Must have $\text{min} \leq \text{max}$, $\text{min} \geq 0$, $\text{max} \geq 1$

Derived from the knowledge of mini-world constraints

Examples:

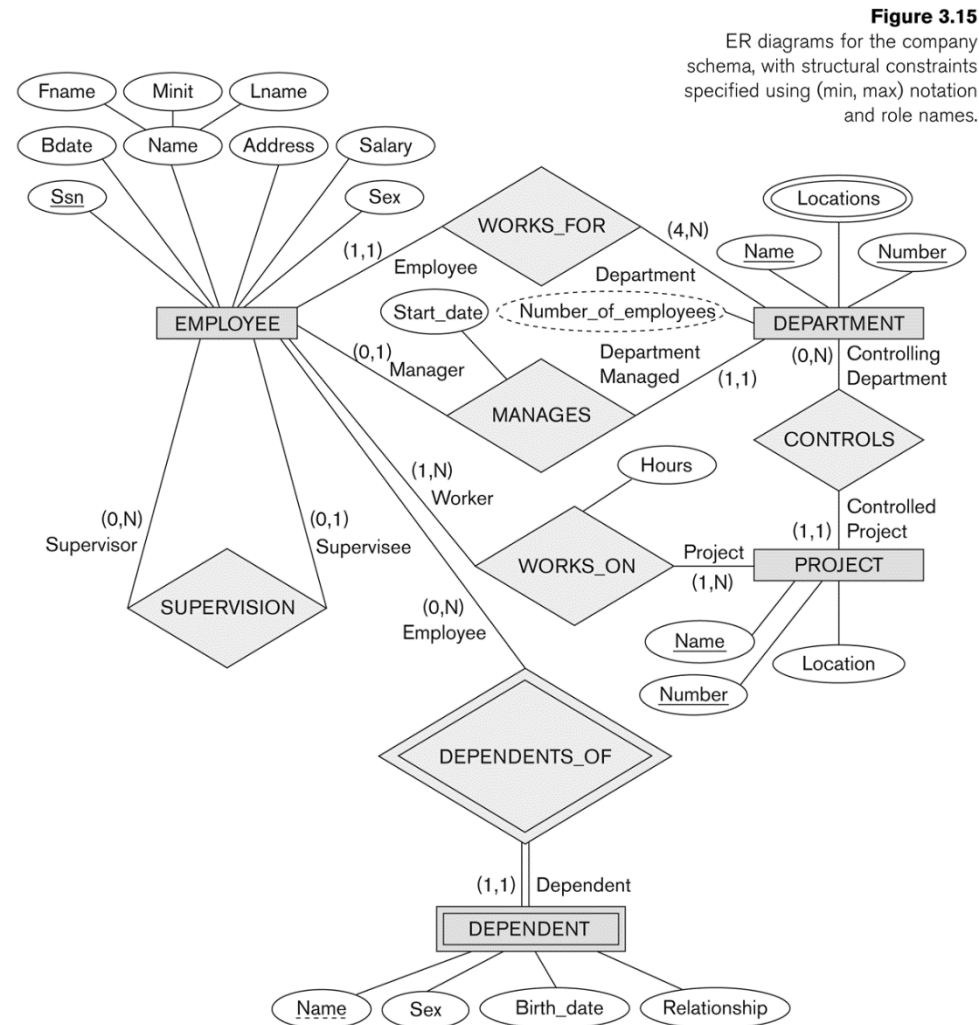
- A department has exactly one manager and an employee can manage at most one department.
 - Specify (0,1) for participation of EMPLOYEE in MANAGES
 - Specify (1,1) for participation of DEPARTMENT in MANAGES
- An employee can work for exactly one department but a department can have any number of employees.
 - Specify (1,1) for participation of EMPLOYEE in WORKS_FOR
 - Specify (0,n) for participation of DEPARTMENT in WORKS_FOR

THE (MIN,MAX) NOTATION FOR RELATIONSHIP CONSTRAINTS



Read the min,max numbers next to the entity type and looking **away from** the entity type

COMPANY ER SCHEMA DIAGRAM USING (MIN, MAX) NOTATION



ALTERNATIVE DIAGRAMMATIC NOTATION

ER diagrams is one popular example for displaying database schemas








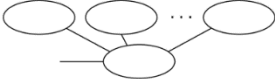

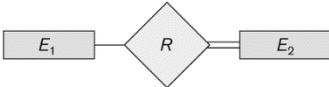


Many other notations exist in the literature and in various database design and modeling tools

Appendix A illustrates some of the alternative notations that have been used

UML class diagrams is representative of another way of displaying ER concepts that is used in several commercial design tools

SUMMARY OF NOTATION FOR ER DIAGRAMS

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

EXAMPLE OF A TERNARY RELATIONSHIP

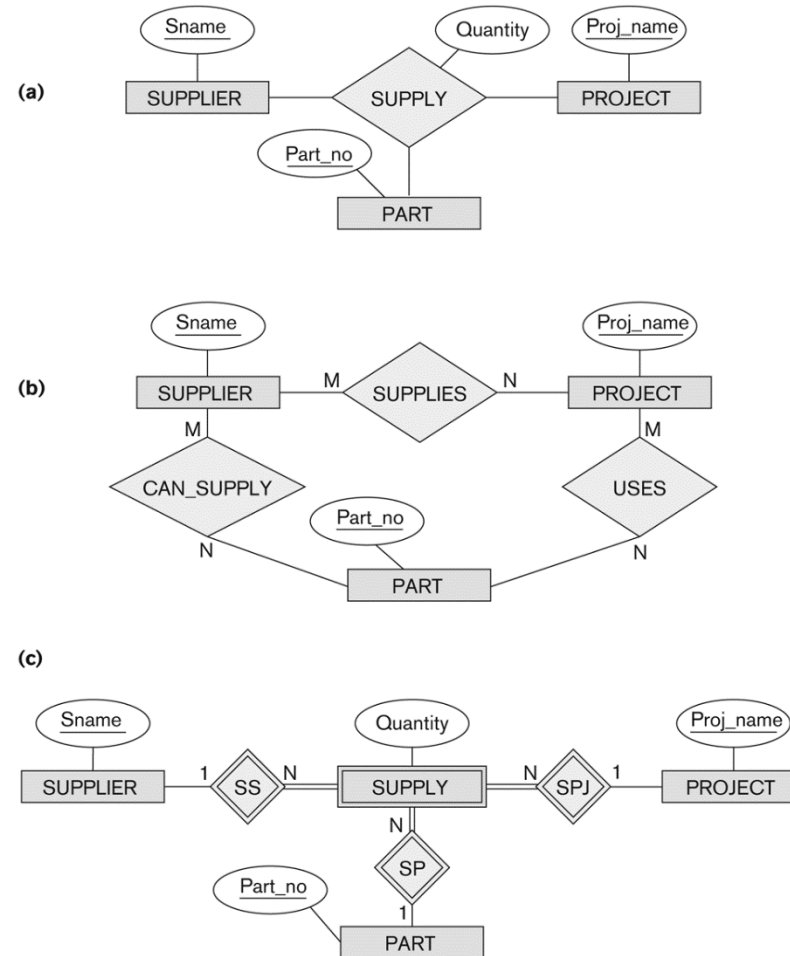


Figure 3.17
 Ternary relationship types. (a) The SUPPLY relationship. (b) Three binary relationships not equivalent to SUPPLY. (c) SUPPLY represented as a weak entity type.

DISCUSSION OF N-ARY RELATIONSHIPS ($N > 2$)

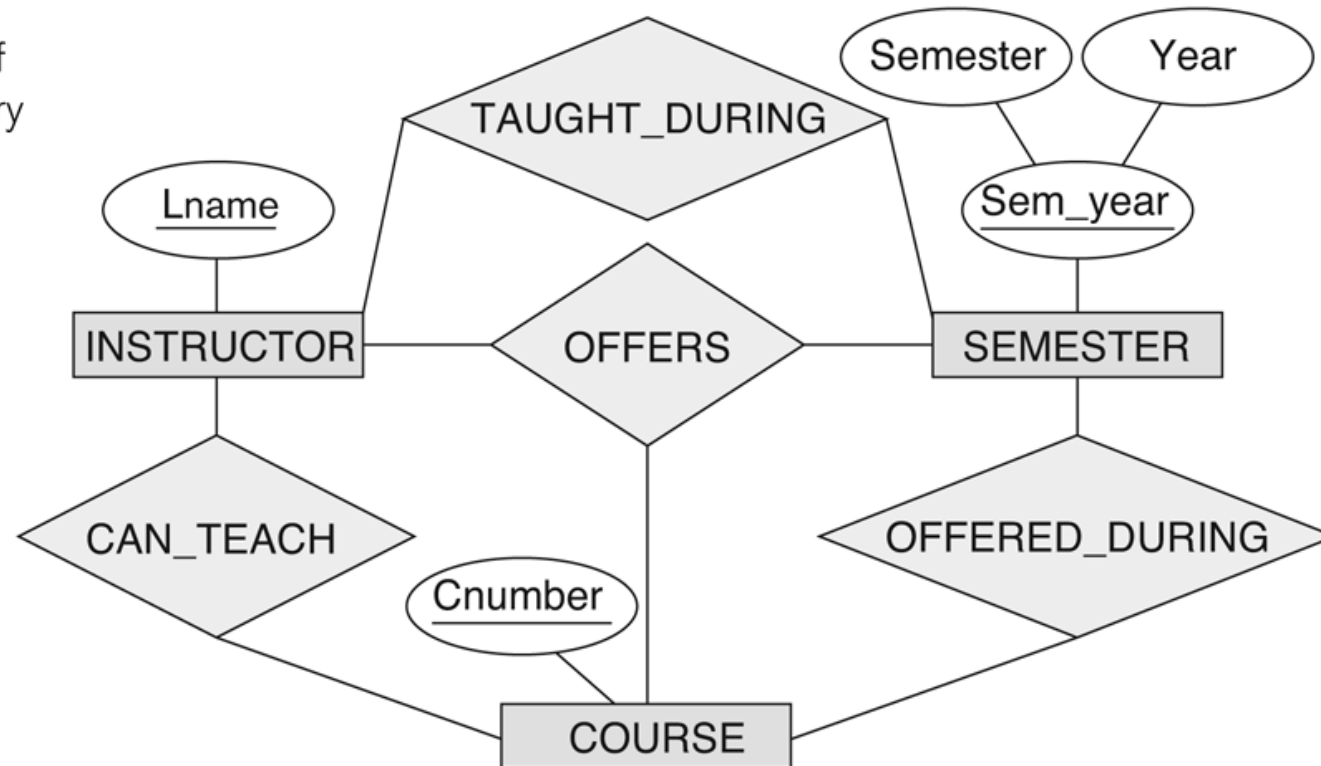
If a particular binary relationship can be derived from a higher-degree relationship at all times, then it is redundant

For example, the TAUGHT_DURING binary relationship in Figure 3.18 (see next slide) can be derived from the ternary relationship OFFERS (based on the meaning of the relationships)

ANOTHER EXAMPLE OF A TERNARY RELATIONSHIP

Figure 3.18

Another example of ternary versus binary relationship types.



DATA MODELING TOOLS

A number of popular tools that cover conceptual modeling and mapping into relational schema design.

- Examples: ERWin, S- Designer (Enterprise Application Suite), ER- Studio, etc.

POSITIVES:

- Serves as documentation of application requirements, easy user interface - mostly graphics editor support

NEGATIVES:

- Most tools lack a proper distinct notation for relationships with relationship attributes
- Mostly represent a relational design in a diagrammatic form rather than a conceptual ER-based design

Some of the Currently Available Automated Database Design Tools

COMPANY	TOOL	FUNCTIONALITY
Embarcadero Technologies	ER Studio	Database Modeling in ER and IDEF1X
	DB Artisan	Database administration, space and security management
Oracle	Developer 2000/Designer 2000	Database modeling, application development
Popkin Software	System Architect 2001	Data modeling, object modeling, process modeling, structured analysis/design
Platinum (Computer Associates)	Enterprise Modeling Suite: Erwin, BPWin, Paradigm Plus	Data, process, and business component modeling
Persistence Inc.	Pwertier	Mapping from O-O to relational model
Rational (IBM)	Rational Rose	UML Modeling & application generation in C++/JAVA
Resolution Ltd.	Xcase	Conceptual modeling up to code maintenance
Sybase	Enterprise Application Suite	Data modeling, business logic modeling
Visio	Visio Enterprise	Data modeling, design/reengineering Visual Basic/C++

EXTENDED ENTITY-RELATIONSHIP (EER) MODEL

The entity relationship model in its original form did not support the specialization and generalization abstractions

Next chapter illustrates how the ER model can be extended with

- Type-subtype and set-subset relationships
- Specialization/Generalization Hierarchies
- Notation to display them in EER diagrams

SUMMARY

ER Model Concepts: Entities, attributes, relationships

Constraints in the ER model

Using ER in step-by-step conceptual schema design for the COMPANY database

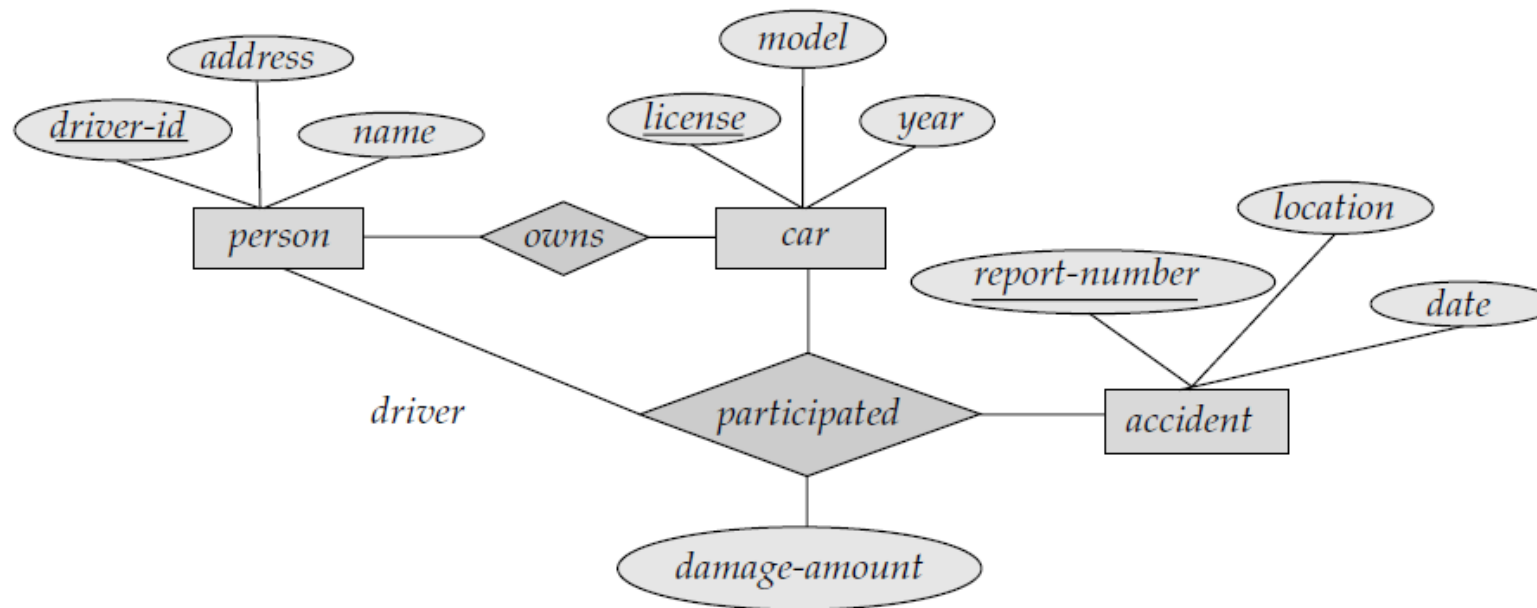
ER Diagrams - Notation

QUESTION 1

Construct an E-R diagram for a car-insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents.

QUESTION 1

Construct an E-R diagram for a car-insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents.

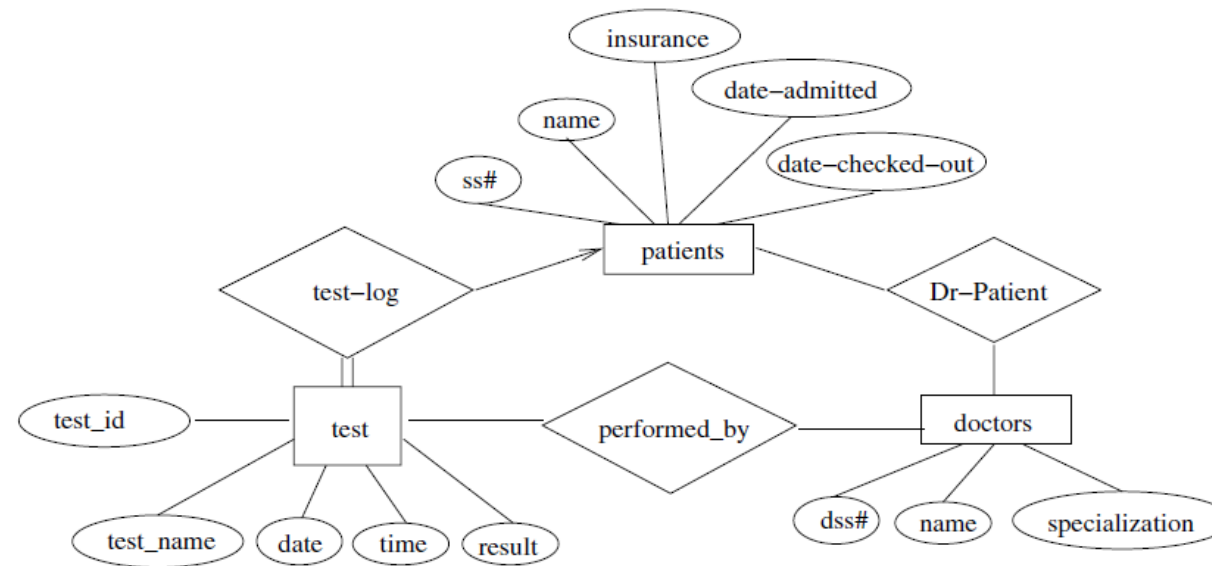


QUESTION 2

Construct an E-R diagram for a hospital with a set of patients and a set of medical doctors. Associate with each patient a log of the various tests and examinations conducted.

QUESTION 2

Construct an E-R diagram for a hospital with a set of patients and a set of medical doctors. Associate with each patient a log of the various tests and examinations conducted.



QUESTION 3

A university registrar's office maintains data about the following entities:

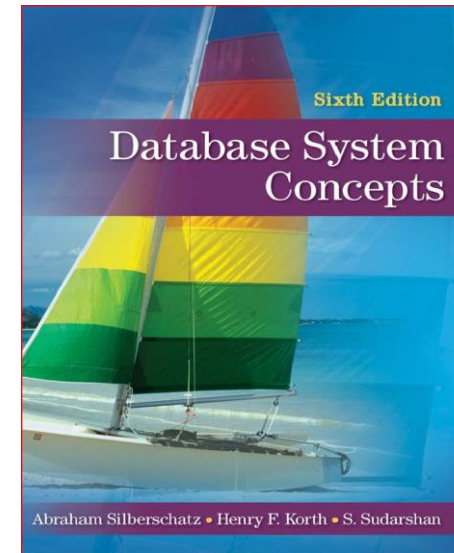
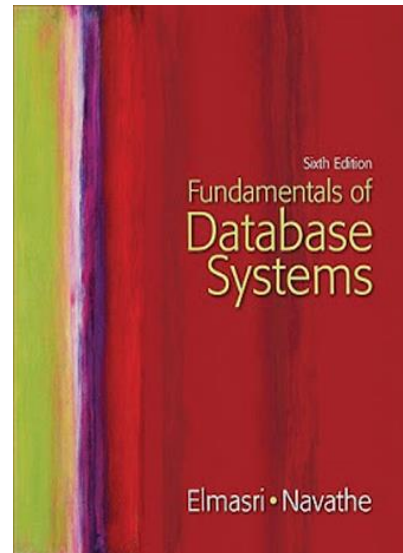
- (a) courses, including number, title, credits, syllabus, and prerequisites;
- (b) course offerings, including course number, year, semester, section number, instructor(s), timings, and classroom;
- (c) students, including student-id, name, and program; and
- (d) instructors, including identification number, name, department, and title.

Further, the enrollment of students in courses and grades awarded to students in each course they are enrolled for must be appropriately modeled.

Construct an E-R diagram for the registrar's office.

Document all assumptions that you make about the mapping constraints.

REFERENCE BOOKS





Keep Learning
Keep Growing

