Data Modeling Using the Entity-Relationship Model

Database Design Process

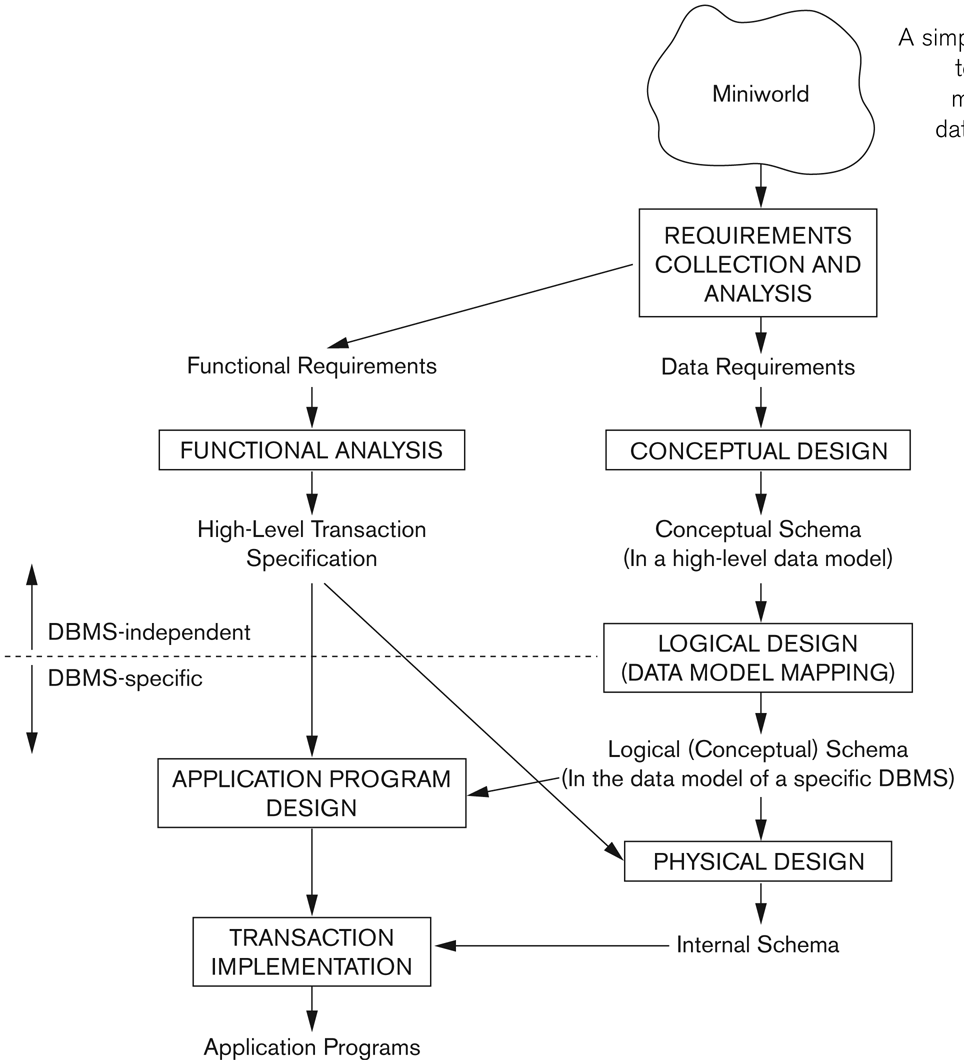
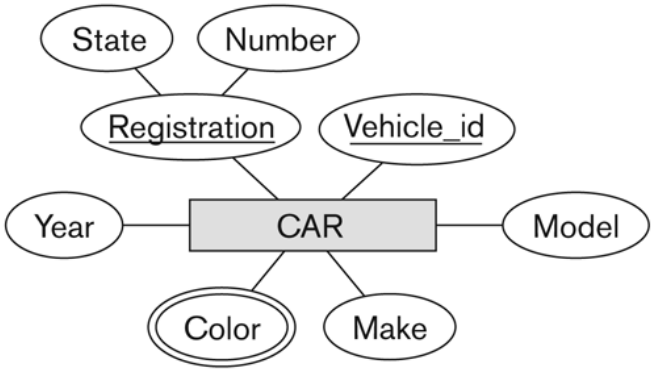


Figure 3.1
A simplified diagram to illustrate the main phases of database design.

ER Model Concepts

- Entities
  - specific objects in the mini-world.
  - E.g. EMPLOYEE John , Research DEPARTMENT

- Attributes
  - They are properties used to describe an entity.
  - Each attribute has a data type
  - E.g. integer, string, subrange, enumerated type, …



Types of Attributes

- Simple
  - Each entity has a single atomic value for the attribute. For example, SSN.
- Composite
  - The attribute is composed of several components. For example:
    - Address(House#, Street, City, State, ZipCode, Country),
    - Name(FirstName, MiddleName, LastName).
- Multi-valued
  - An entity may have multiple values for that attribute. For example:
  - PreviousDegrees of a STUDENT.

Example of a composite attribute

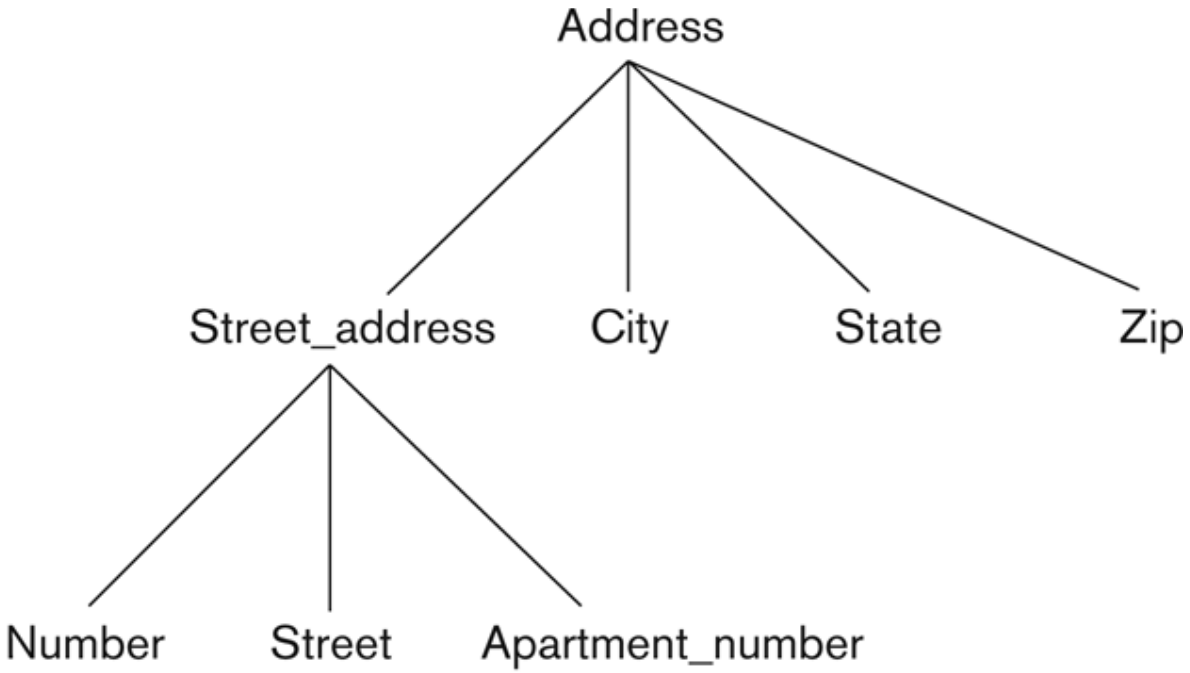


Figure 3.4
A hierarchy of composite attributes.

Entity Types and Key Attributes

- Entities with the same basic attributes are grouped into an Entity type.
  - For example, the entity type EMPLOYEE and PROJECT.

- Key attribute
  - For example, SSN of EMPLOYEE.

Entity Type CAR with two keys and a corresponding Entity Set

(a)

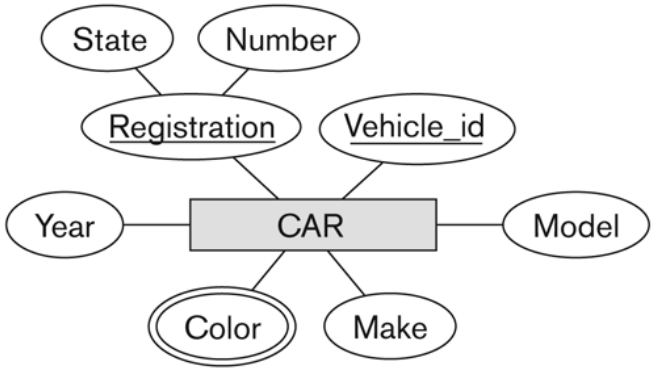


Figure 3.7
The CAR entity type with two key attributes, Registration and Vehicle_id. (a) ER diagram notation. (b) Entity set with three entities.

(b)

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

CAR_1
((ABC 123, TEXAS), TK629, Ford Mustang, convertible, 2004 {red, black})

CAR_2
((ABC 123, NEW YORK), WP9872, Nissan Maxima, 4-door, 2005, {blue})

CAR_3
((VSY 720, TEXAS), TD729, Chrysler LeBaron, 4-door, 2002, {white, blue})

⋮

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

Initial Design of Entity Types: EMPLOYEE, DEPARTMENT, PROJECT, DEPENDENT

Entity types in the COMPANY database:

- DEPARTMENT
- PROJECT
- EMPLOYEE
- DEPENDENT

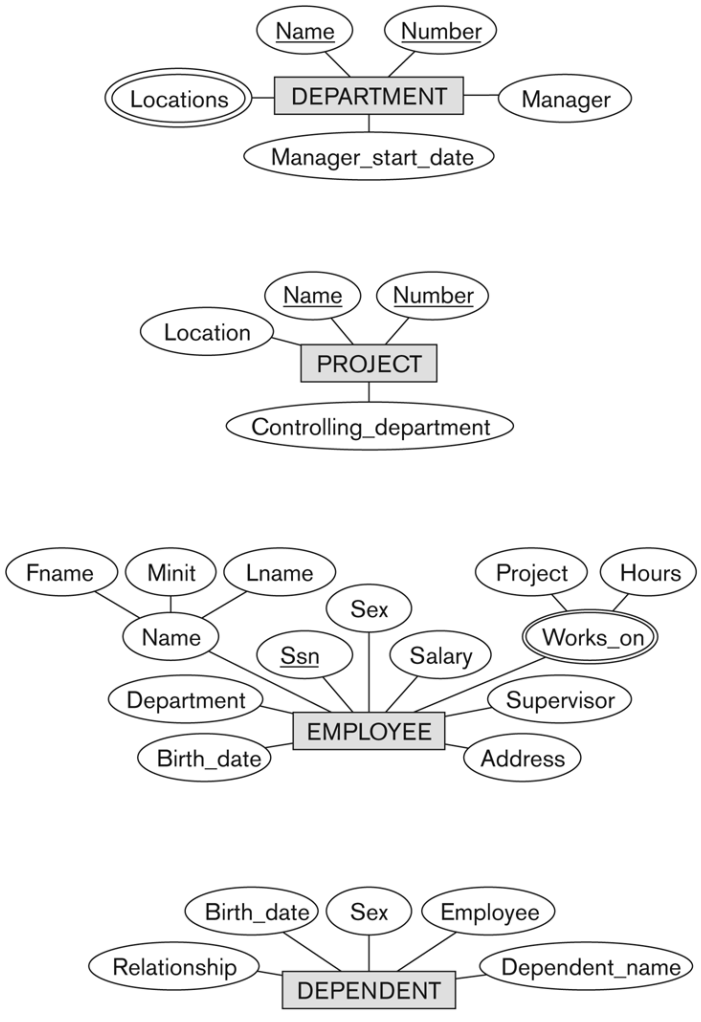


Figure 3.8
Preliminary design of entity types for the COMPANY database. Some of the shown attributes will be refined into relationships.

Refining the initial design by introducing **relationships**

- 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)

Relationships

- A **relationship** relates two or more distinct entities with a specific meaning.
  - For example, EMPLOYEE John *works on* the ProductX PROJECT,
  - EMPLOYEE Franklin *manages* the Research DEPARTMENT.
- Relationships of the same type are grouped into a **relationship type**.
- Degree of a relationship type is the no 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

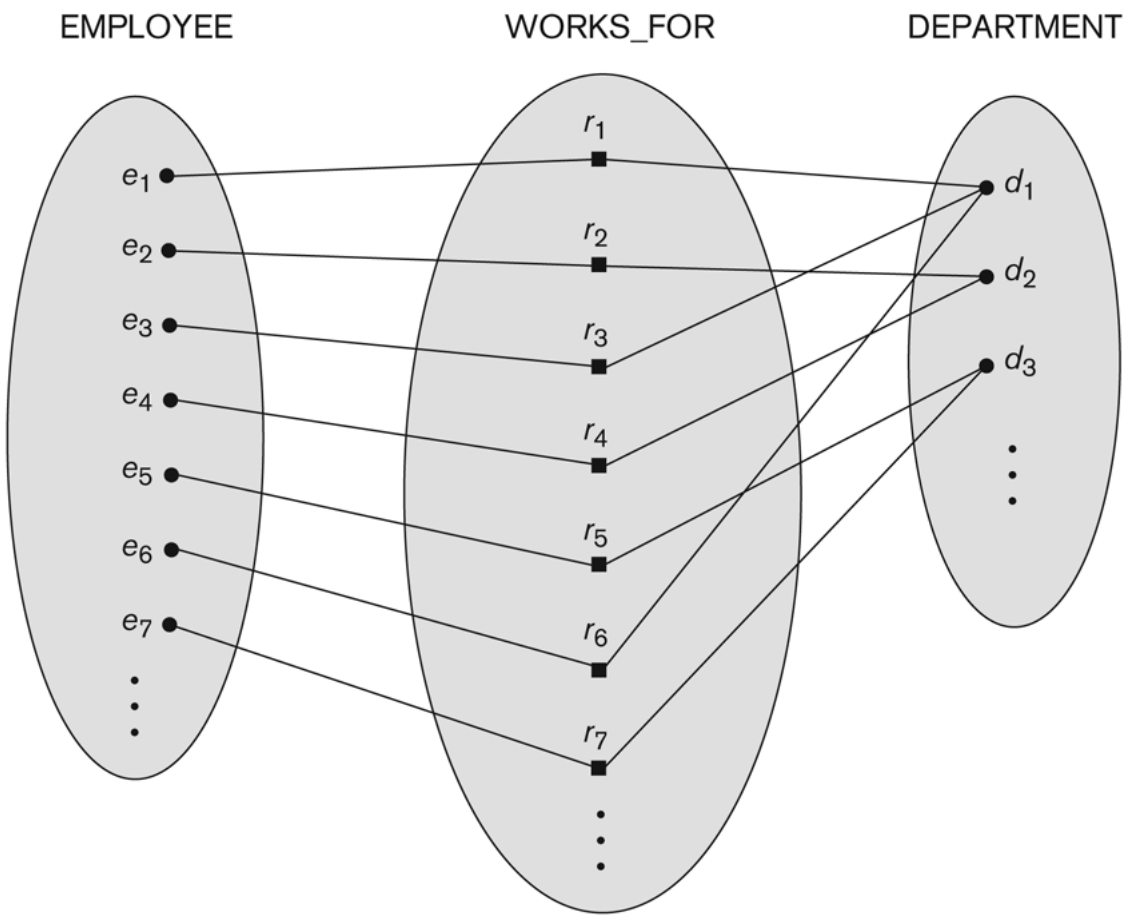


Figure 3.9
Some instances in the WORKS_FOR relationship set, which represents a relationship type WORKS_FOR between EMPLOYEE and DEPARTMENT.

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

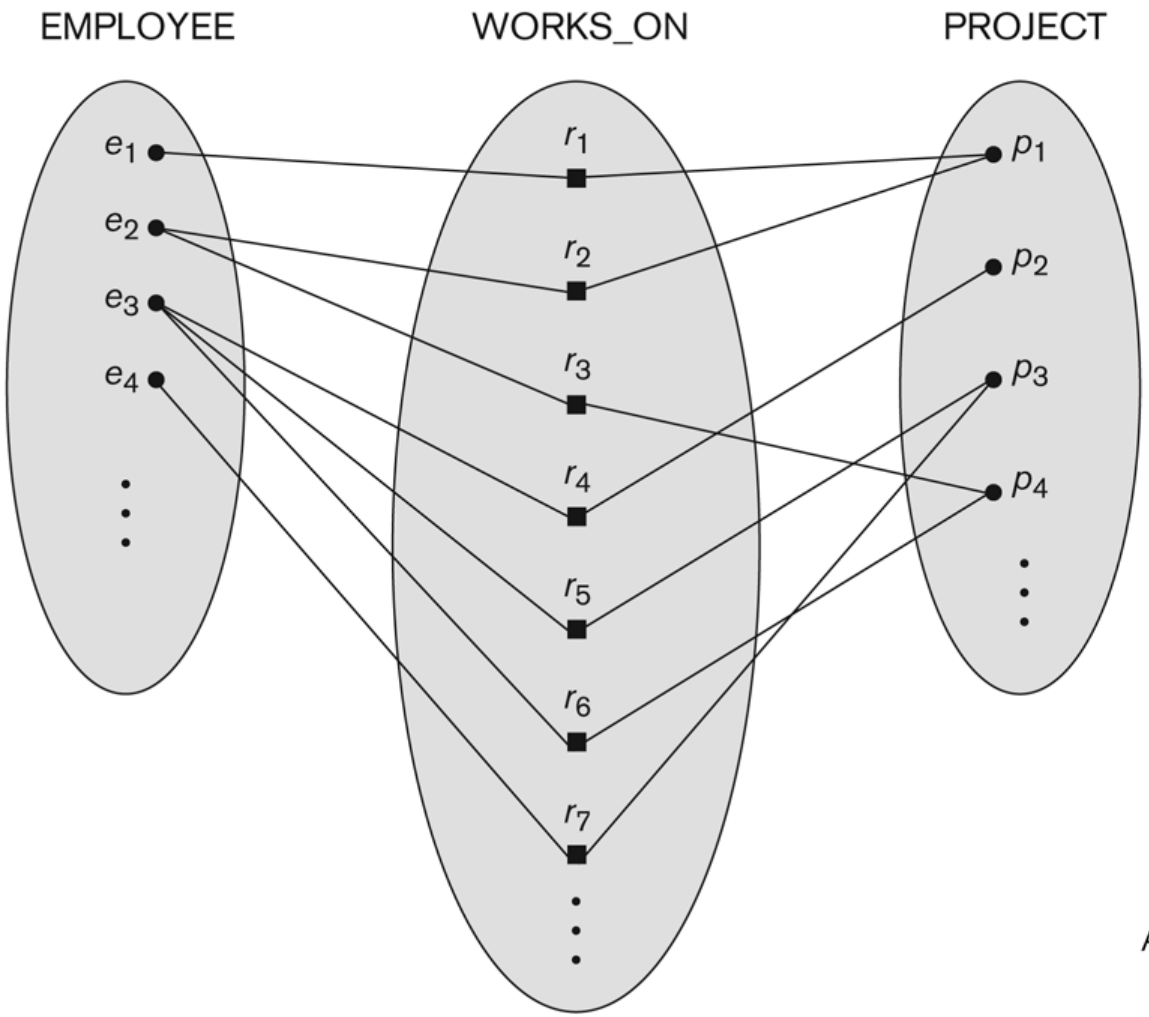
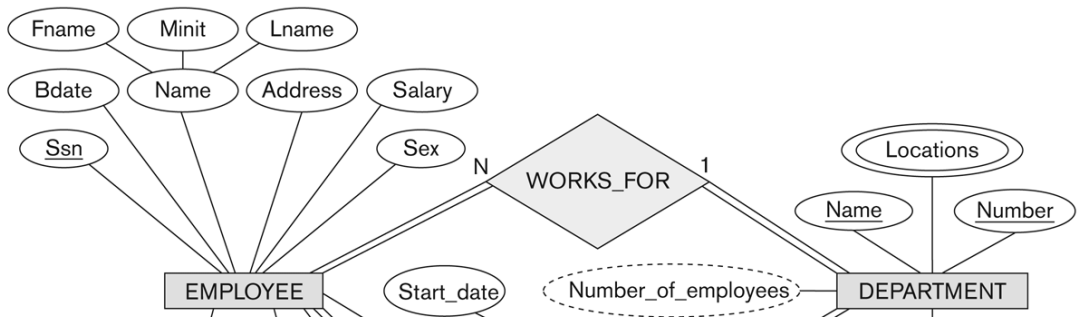


Figure 3.13
An M:N relationship, WORKS_ON.

Relationship type vs. relationship set

- 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 *state* of a relationship type



COMPANY DATABASE: RELATIONSHIPS

- Six relationship types are identified
- All are *binary* relationships(degree 2)
  - WORKS_FOR (between EMPLOYEE, DEPARTMENT)
  - MANAGES (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

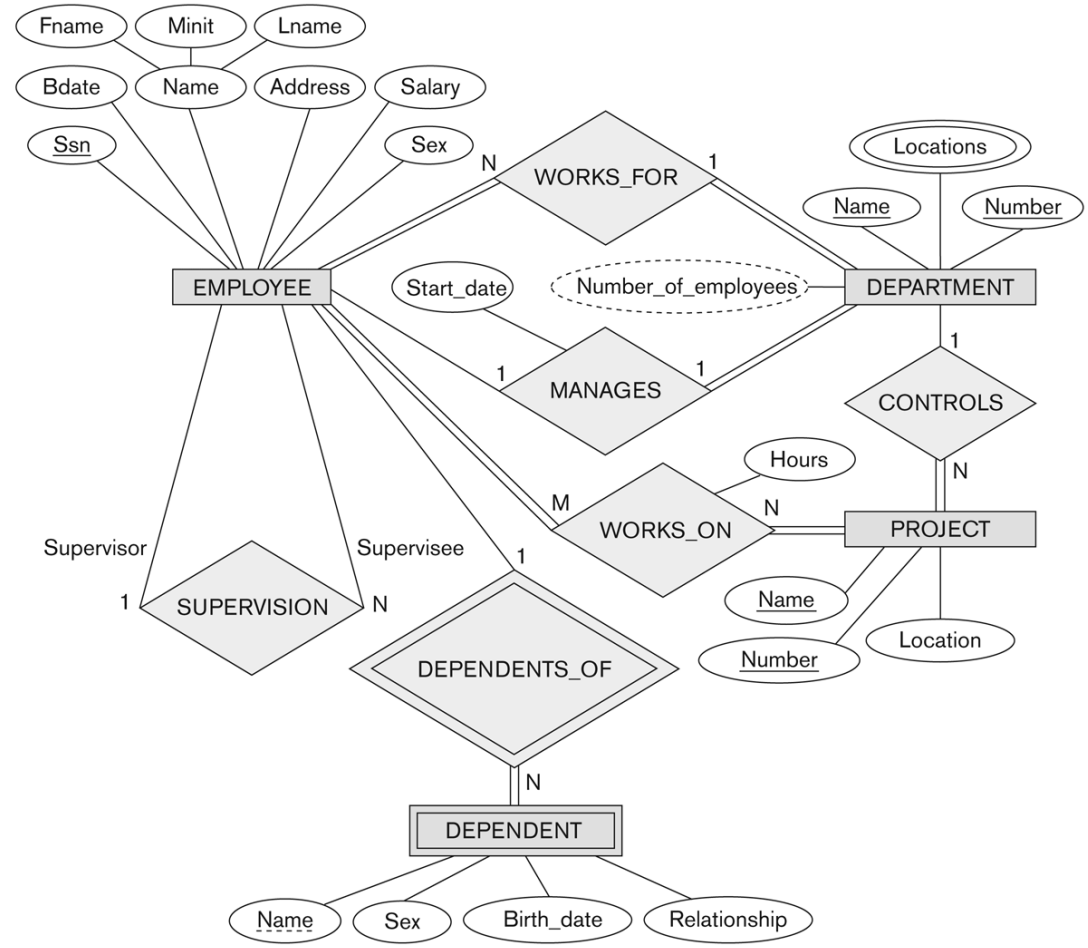


Figure 3.2
An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter.

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:
  - 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 (*partial key*), *and*
    - the specific EMPLOYEE with whom the dependent is related

Constraints on Relationships

- Constraints on Relationship Types (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

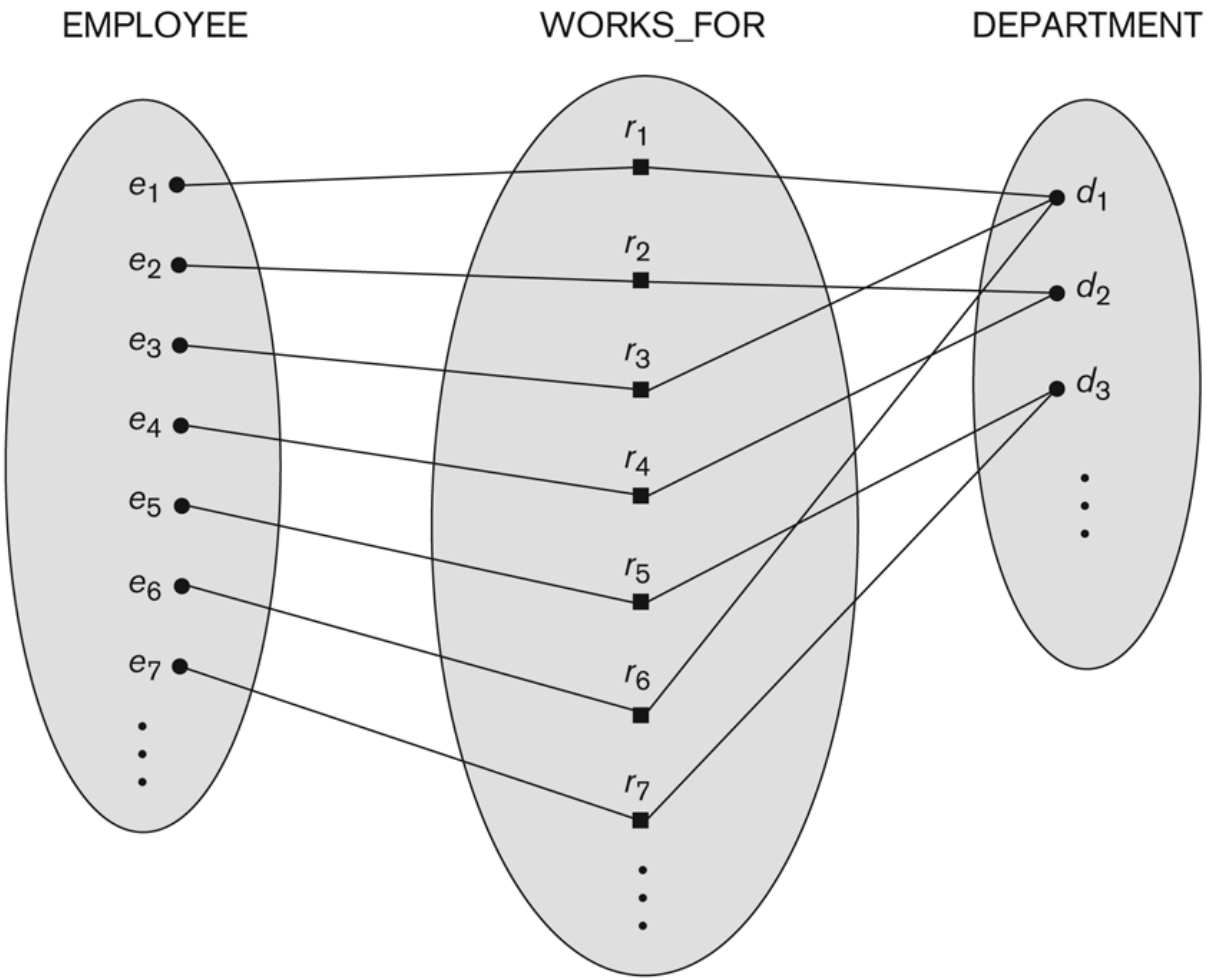


Figure 3.9
Some instances in the WORKS_FOR relationship set, which represents a relationship type WORKS_FOR between EMPLOYEE and DEPARTMENT.

Many-to-many (M:N) Relationship

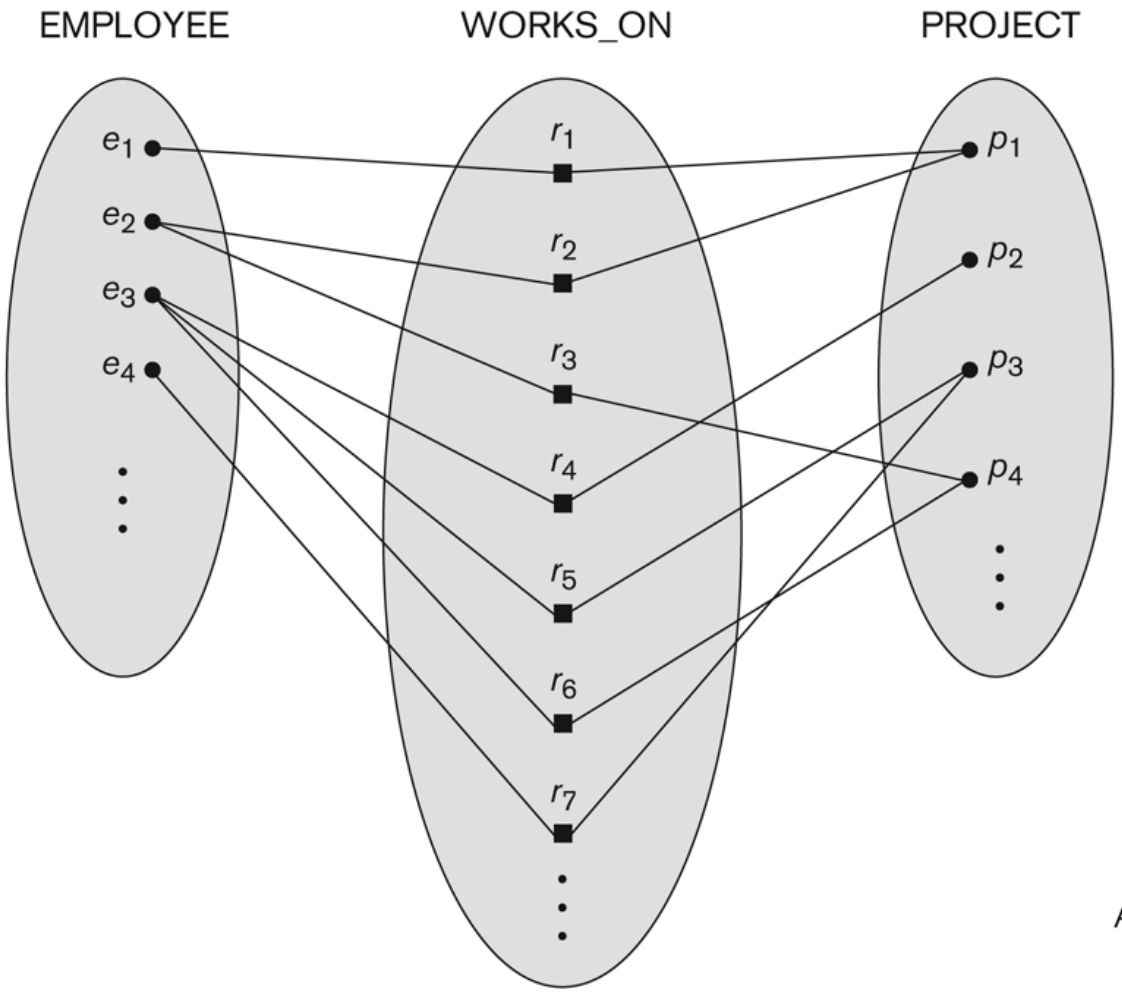


Figure 3.13
An M:N relationship, WORKS_ON.

Recursive Relationship Type

- 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

A Recursive Relationship Supervision`

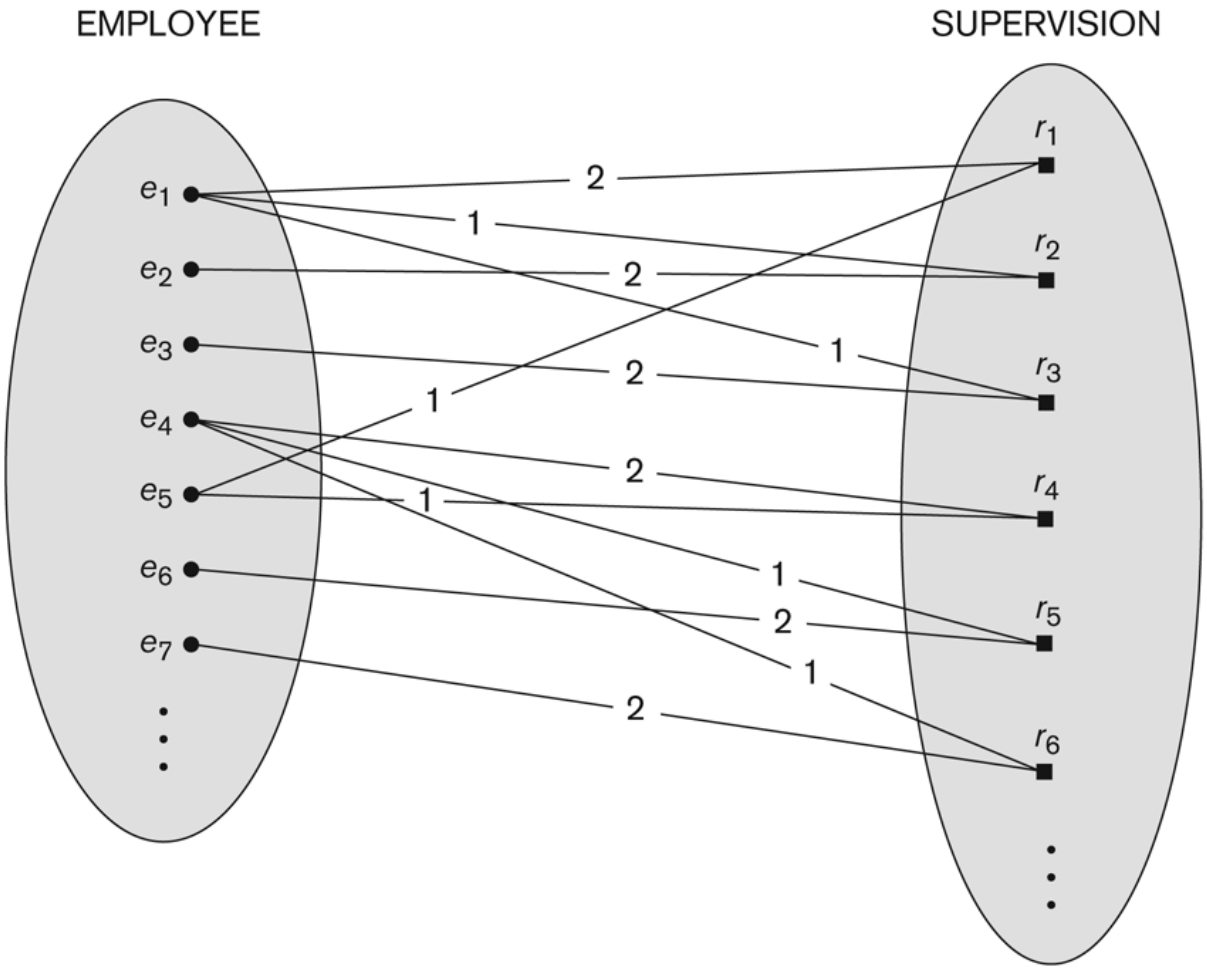


Figure 3.11
A recursive relationship SUPERVISION between EMPLOYEE in the *supervisor* role (1) and EMPLOYEE in the *subordinate* role (2).

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

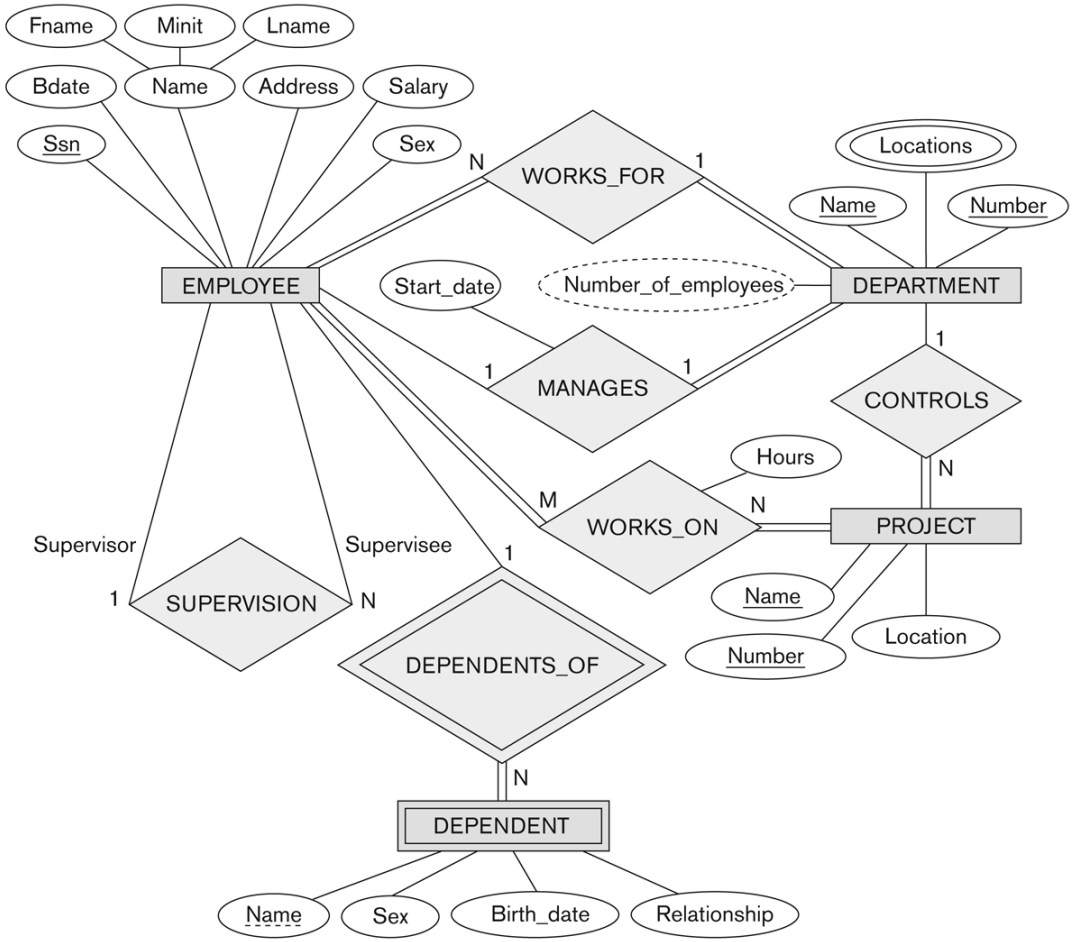


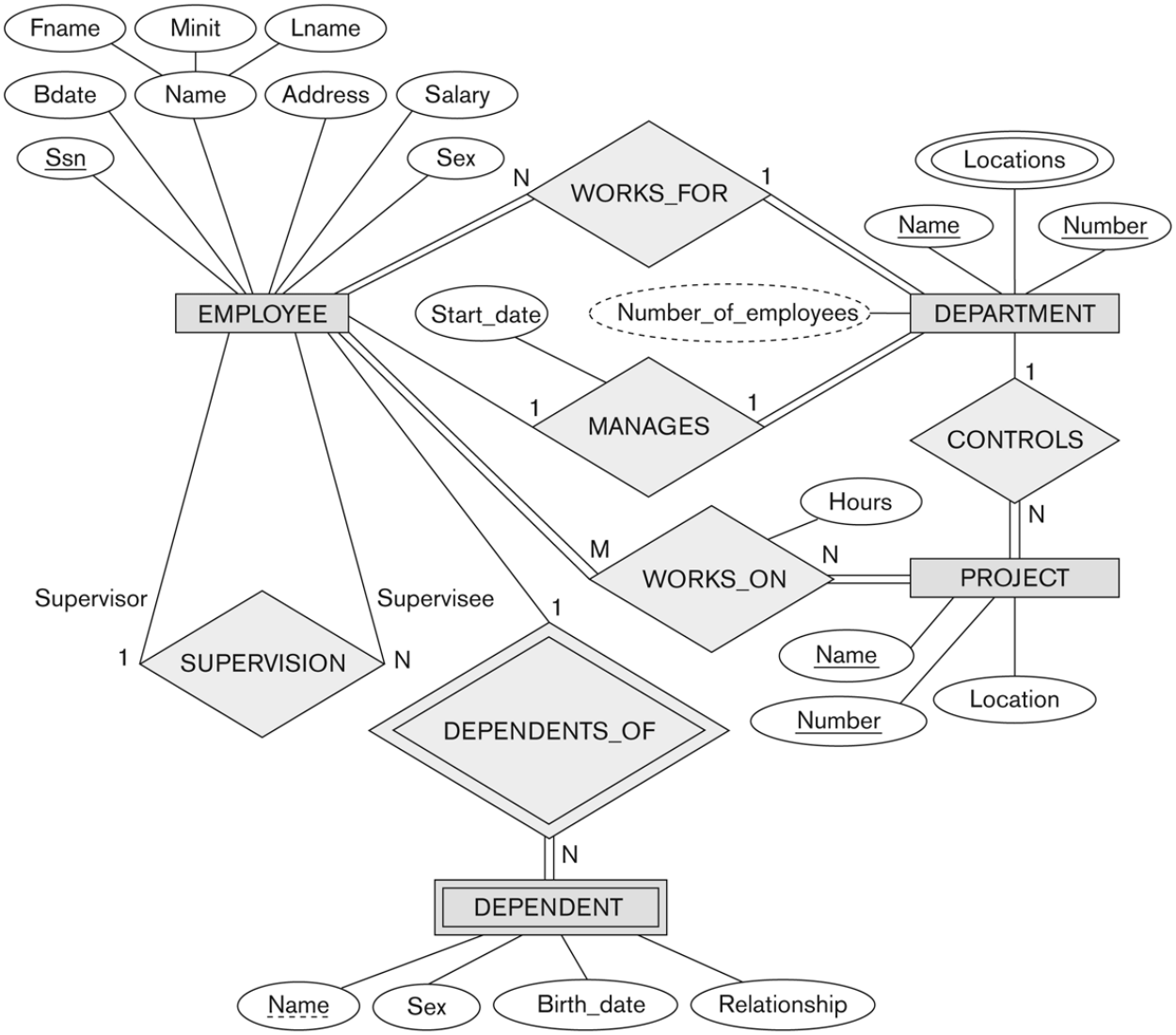
Figure 3.2
An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter.

Attributes of Relationship types

- A relationship type can have attributes:
  - For example, HoursPerWeek of WORKS_ON
  - 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:



Participation Constraint : Total vs partial.

Cardinality Ratio: 1:N, 1:1, N:M

Figure 3.2

Steps to draw an ER diagram

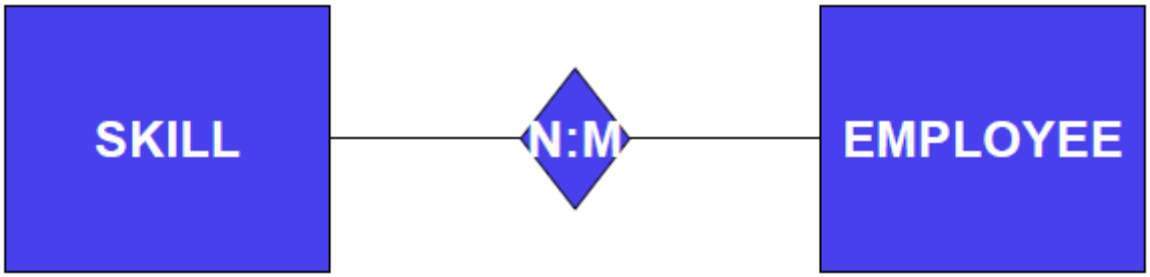
- Get problem description
- Define Entities
- Add Attributes
- Specify Key, multiple, composite attributes
- Add Relations
- Specify Cardinality, total/partial relations
- Iterate

Steps to draw an ER diagram

- Essential to further design, but often given little care:
- Is an entity a weak entity (key?)
- Multivalued attributes
- Derived attribute
- Total/partial participation
- Cardinality ratio

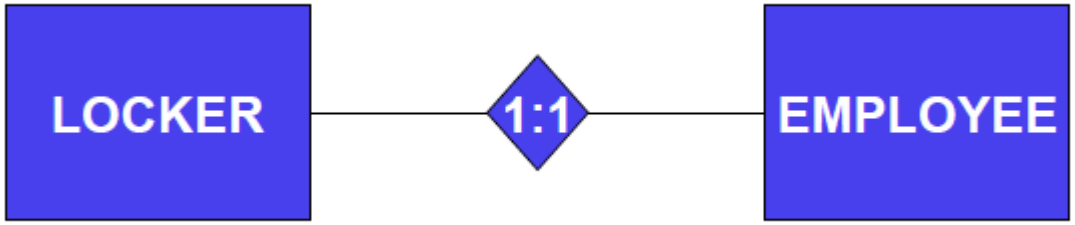
Examples

- An employee may have several skills; and
- A particular skill may be held by several employees



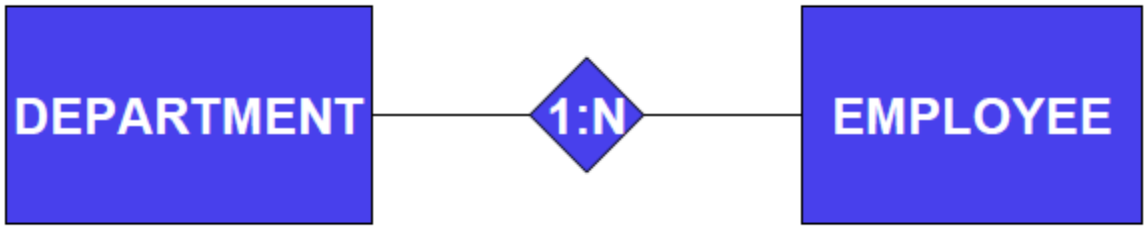
Example

- An employee may have no more than one locker; and
- A locker may only be accessible by one employee



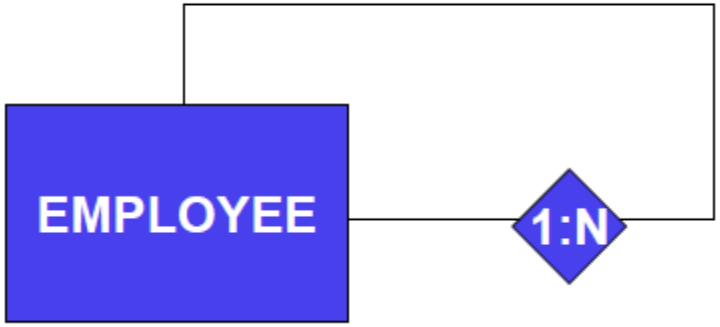
Example

- An employee may only work for one department; and
- A department has several employees

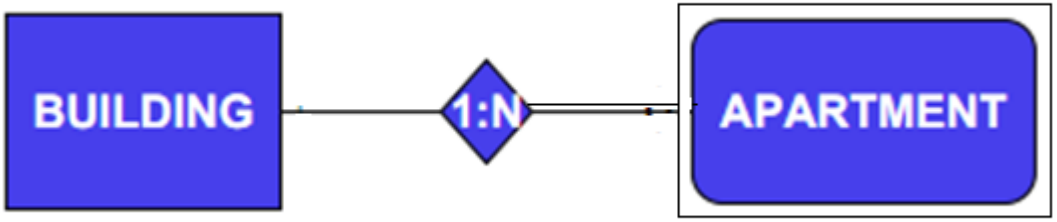


Example

- An employee can have many colleagues



- A building has many apartments
- An apartment exists in only one building



Problem 2

Draw E\R model for the university database with the following requirements

a) The university keeps track of each student's name, rollno, SSN, current address and phone, permanent address and phone, birthdate, class (freshman, sophomore, ..., graduate), major department, minor department (if any), and degree program (B.A., B.S., ..., Ph.D.). Some user applications need to refer to the city, state, and zip of the student's permanent address, and to the student's last name. Both SSN and Rollno have unique values for each student.

b) Each department is described by a name, code, office number, office phone, and college. Both name and code have unique values for each department.

c) Each course has a course name, description, course number, number of semester hours, level, and offering department. The value of course number is unique for each course.

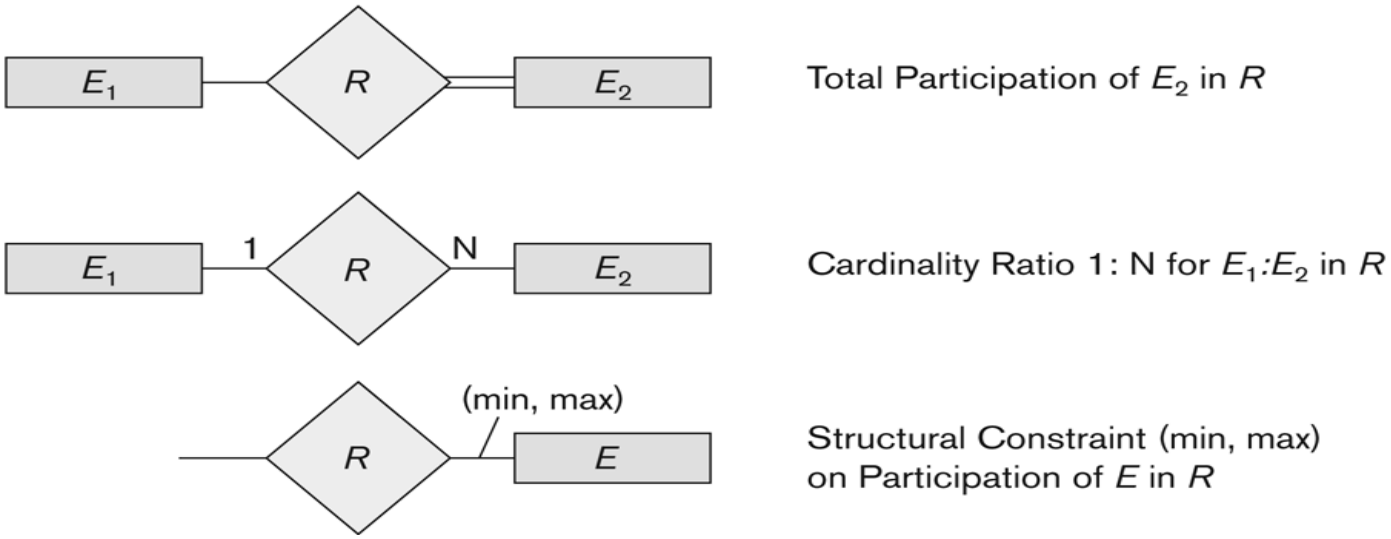
d) Each section has an instructor, semester, year, course, and section number. The section number distinguishes different sections of the same course that are taught during the same semester/year; its values are 1, 2, 3, ...; up to the number of sections taught during each semester.

e) A grade report has a student rollno,

f) section, letter grade, and numeric grade (0, 1, 2, 3, 4 for F, D, C, B, A, respectively).

Alternative (min, max) notation for relationship Structural constraints:

- Specifies that each entity e in E participates in at least *min* and at most *max* relationship instances in R
- Default(no constraint): min=0, max=n
- Must have min ≤ max, min ≥ 0, max ≥ 1
- Derived from the knowledge of mini-world constraints



(MIN , MAX) NOTATION FOR RELATIONSHIP CONSTRAINTS

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

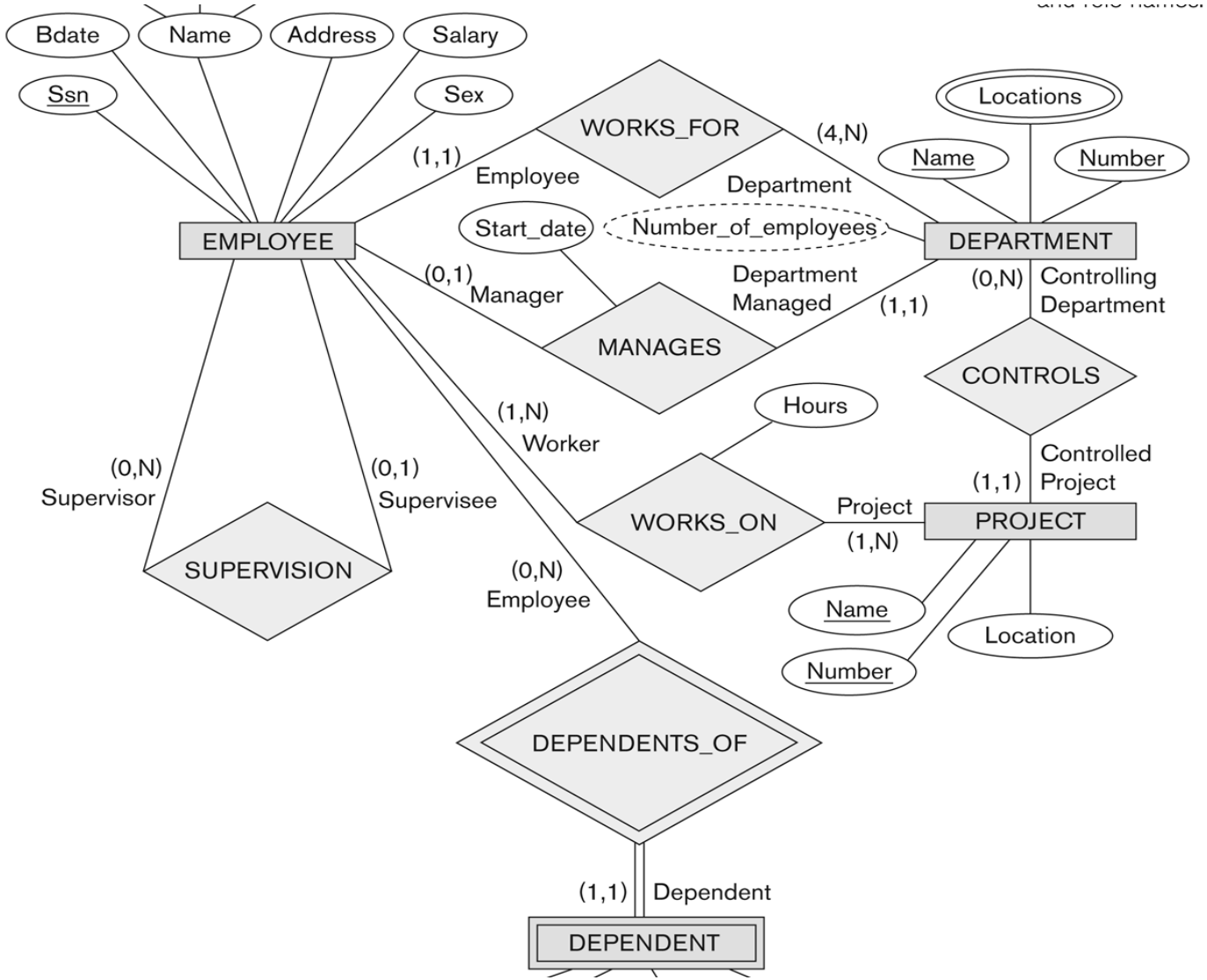
Read the min,max numbers next to the entity type and looking **away from** the entity type

The (min,max) notation for relationship constraints

An employee can work for exactly one department but a department can have any number of employees.

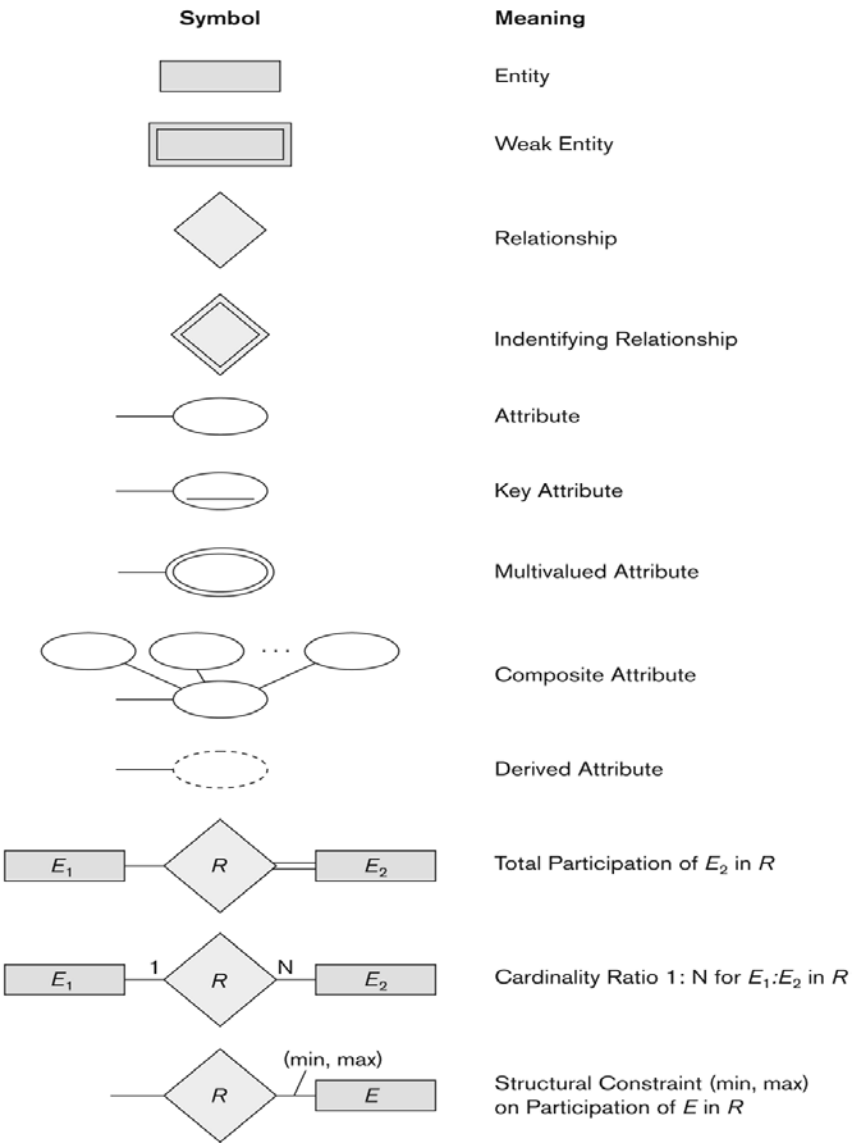
- Specify (1,1) for participation of EMPLOYEE in WORKS_FOR
- Specify (1,n) for participation of DEPARTMENT in WORKS_FOR

MIN, MAX - NOTATION



NOTATION FOR ER DIAGRAMS

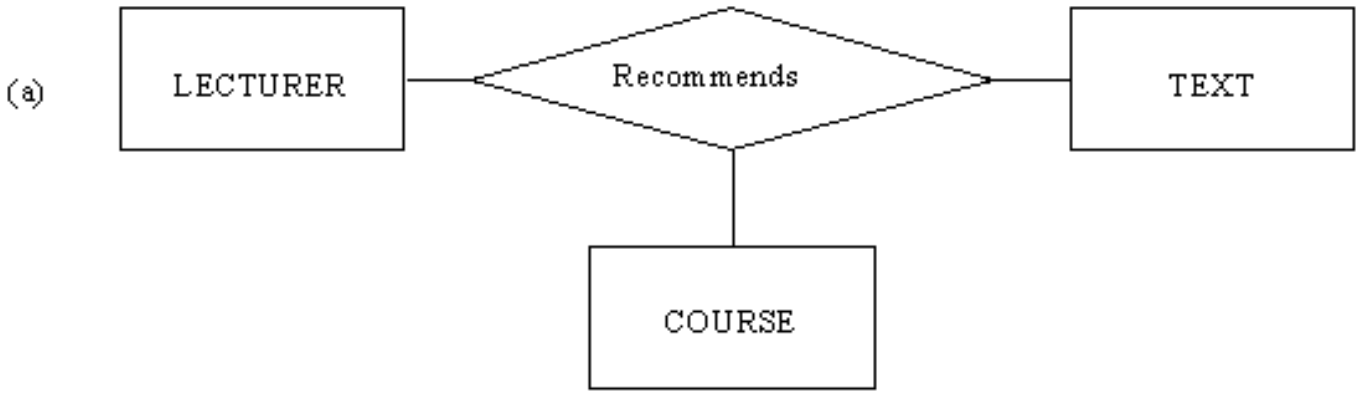
Figure 3.14
Summary of the notation for ER diagrams.



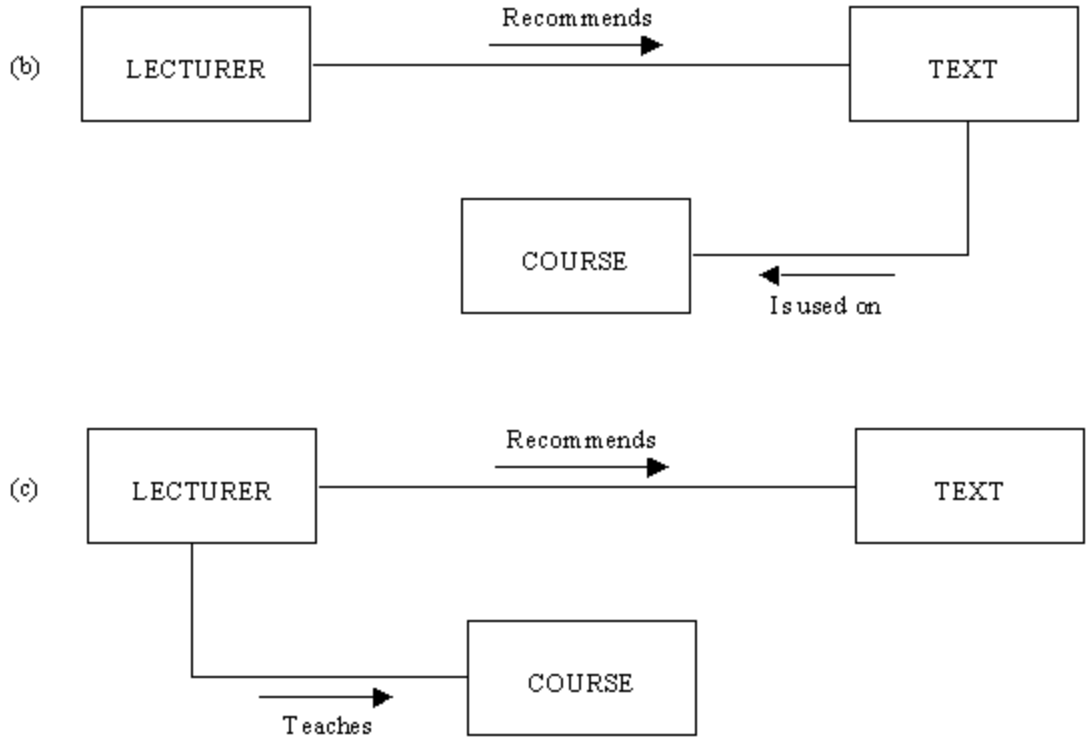
Relationships of Higher Degree

- Relationship types of degree 2 are called binary
- Relationship types of degree 3 are called ternary and of degree n are called n-ary
- Constraints are harder to specify for higher-degree relationships (n > 2) than for binary relationships

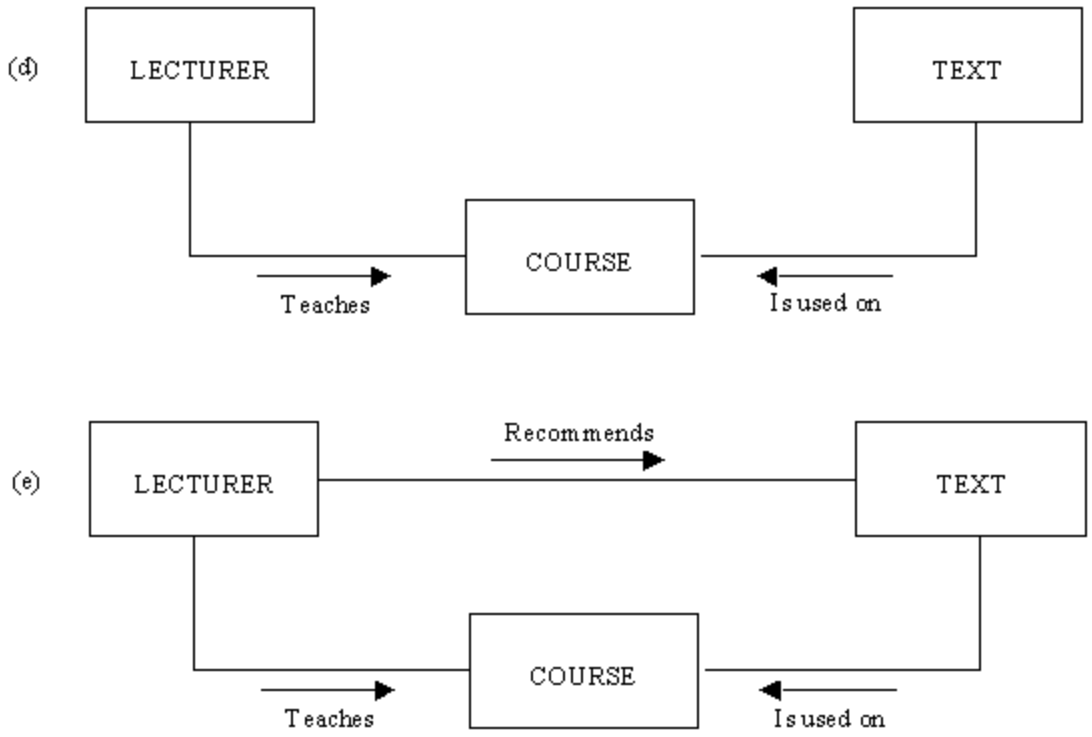
Example of a ternary relationship



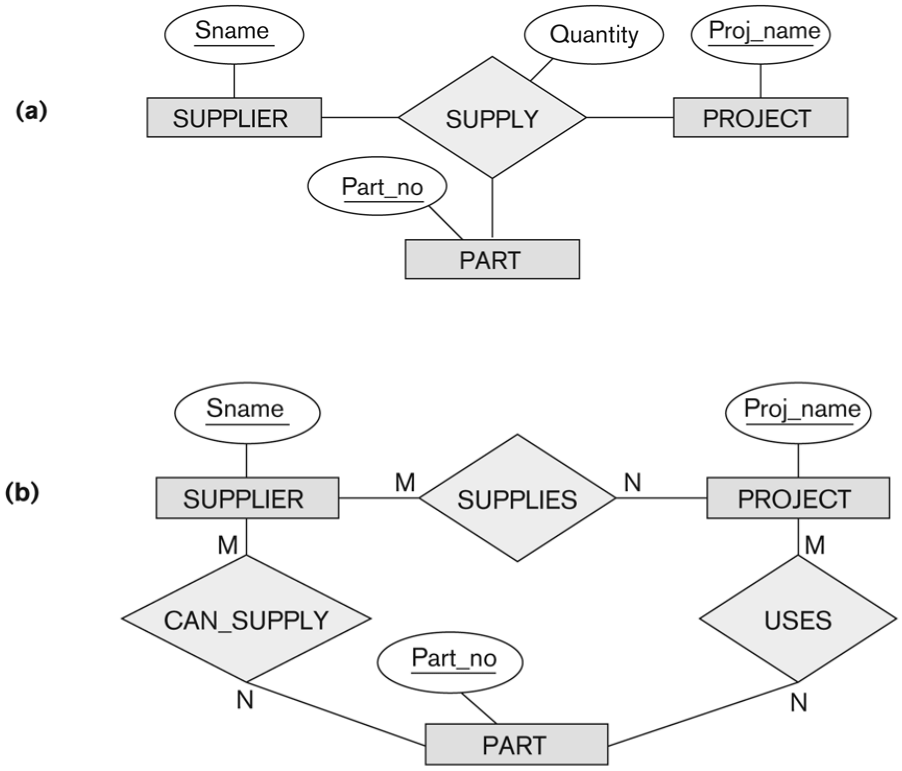
EXAMPLE OF A TERNARY RELATIONSHIP



EXAMPLE OF A TERNARY RELATIONSHIP

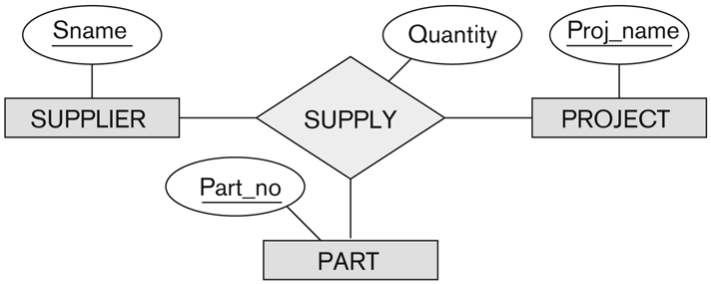


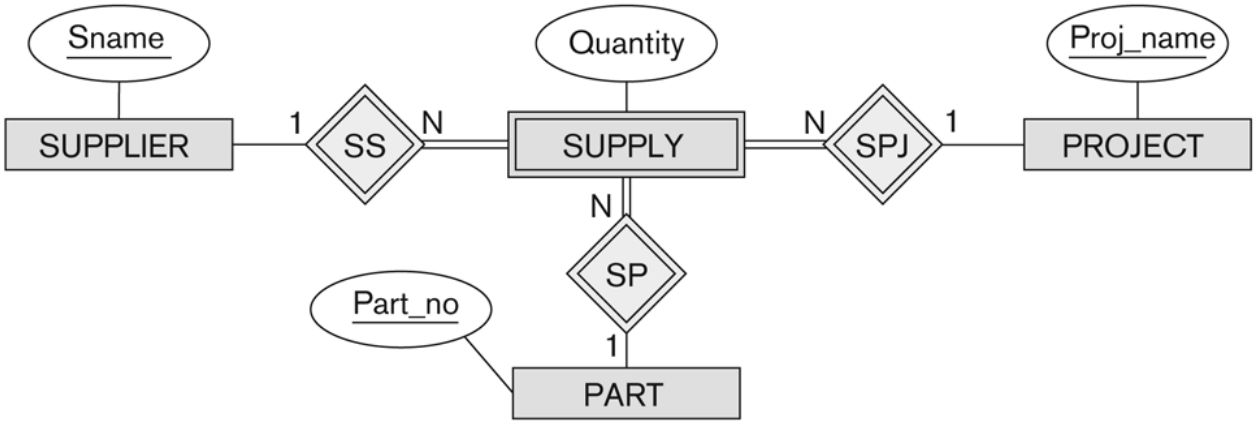
Example 2 of a ternary relationship



N-ARY RELATIONSHIPS (N > 2)

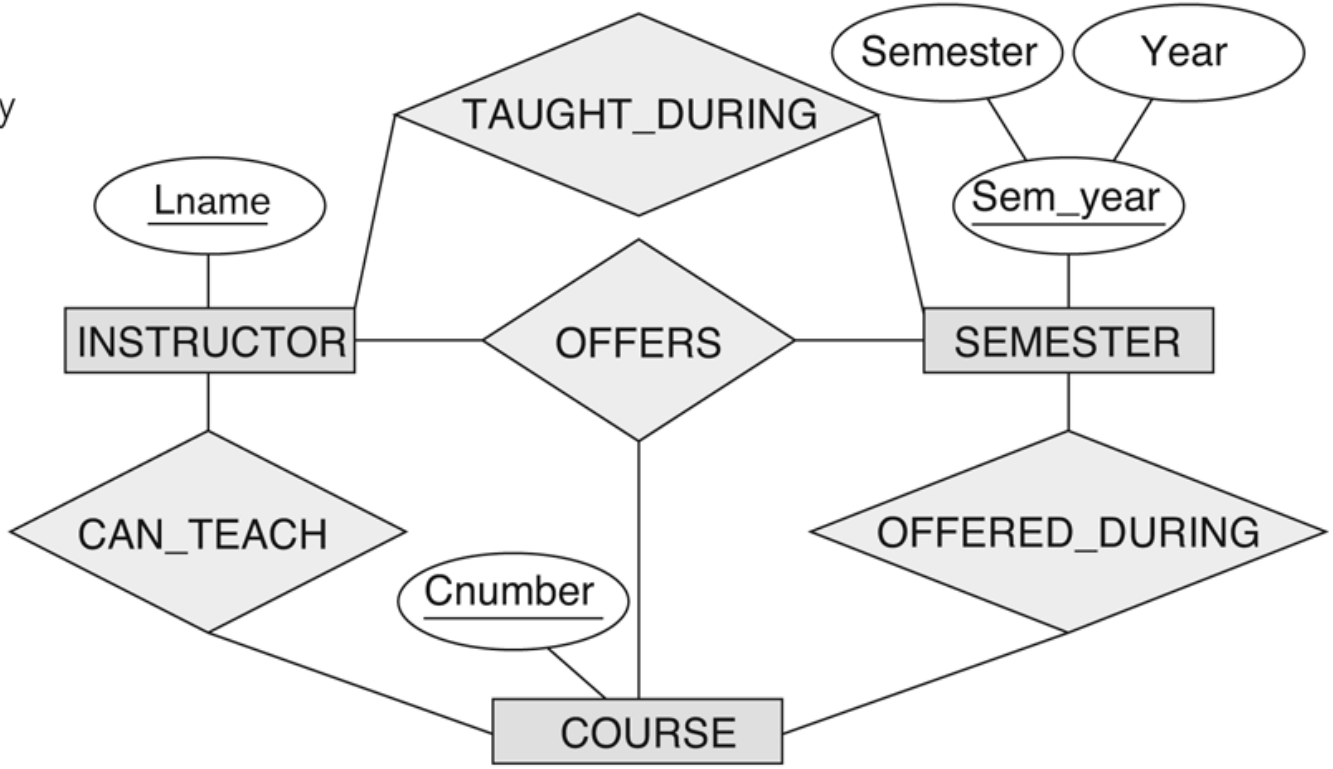
- Three binary relationships represents different information than a single ternary relationship
- In some cases, a ternary relationship can be represented as a weak





Example 3 of a ternary relationship

Figure 3.18
Another example of ternary versus binary relationship types.



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 can be derived from the ternary relationship OFFERS (based on the meaning of the relationships)

Displaying constraints on higher-degree relationships

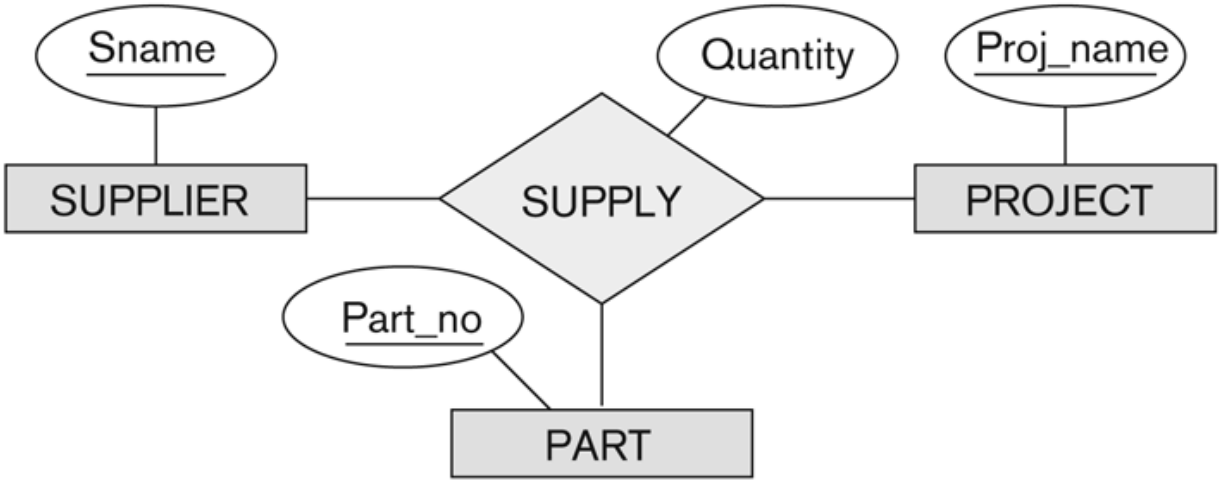
- Displaying a 1, M, or N indicates
  - 1 indicates that an entity can participate in at most one relationship instance *that has a particular combination of the other participating entities*
  - M or N indicates no constraint

- (min, max) constraints can be displayed on the edges – however, they do not fully describe the constraints

- In general, both (min, max) and 1, M, or N are needed to describe fully the constraints

Cardinality for ternary relationship

Constraint: For a particular project-part combination, only one supplier will be used (only one supplier supplies a particular part to a particular project).



Problem 1: ER for Notown Records

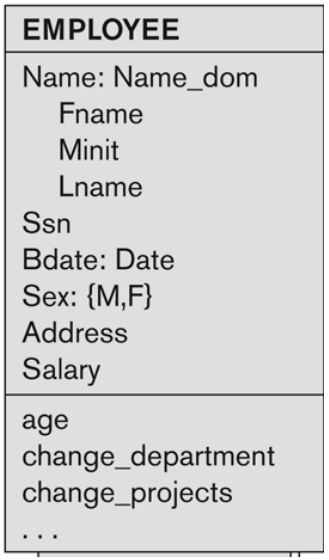
1. Each musician has an SSN, name, address, phone. Poor musicians often share the same address, and no address has more than one phone.
2. Each instrument that is used in songs recorded at Notown has a name (e.g., guitar, flute) and a musical key (e.g., C, B-flat).
3. Each album has a title, a copyright date, a format (e.g., CD or MC), and an album identifier.
4. Each song recorded at Notown has a title and an author.
5. Each musician may play several instruments, and an instrument may be played by several musicians.
6. Each album has a number of songs on it, but no song may appear on more than one album.
7. Each song is performed by one or more musicians and *we keep track of the instrument used by each musician*. A musician may perform a number of songs.
8. Each album has exactly one musician who acts as its producer. A musician may produce several albums, of course.

Alternative diagrammatic notation

- ER diagrams is one popular example for displaying database schemas
- UML class diagrams is representative of another way of displaying ER concepts

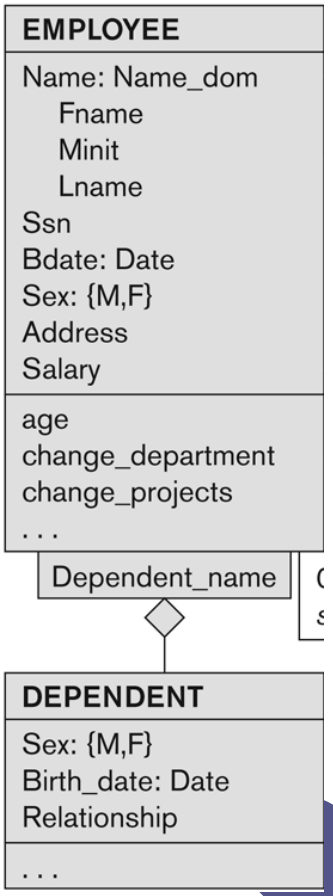
UML CLASS DIAGRAMS

- Represent classes (similar to entity types) as large boxes with three sections:
  - Top section includes entity type (class) name
  - Second section includes attributes
  - Third section includes class operations (operations are not specified in ER model)
- Composite attribute is modeled as a **structured domain**
  - Name of EMPLOYEE
- Multivalued attribute is modeled as a separate class
  - LOCATION class

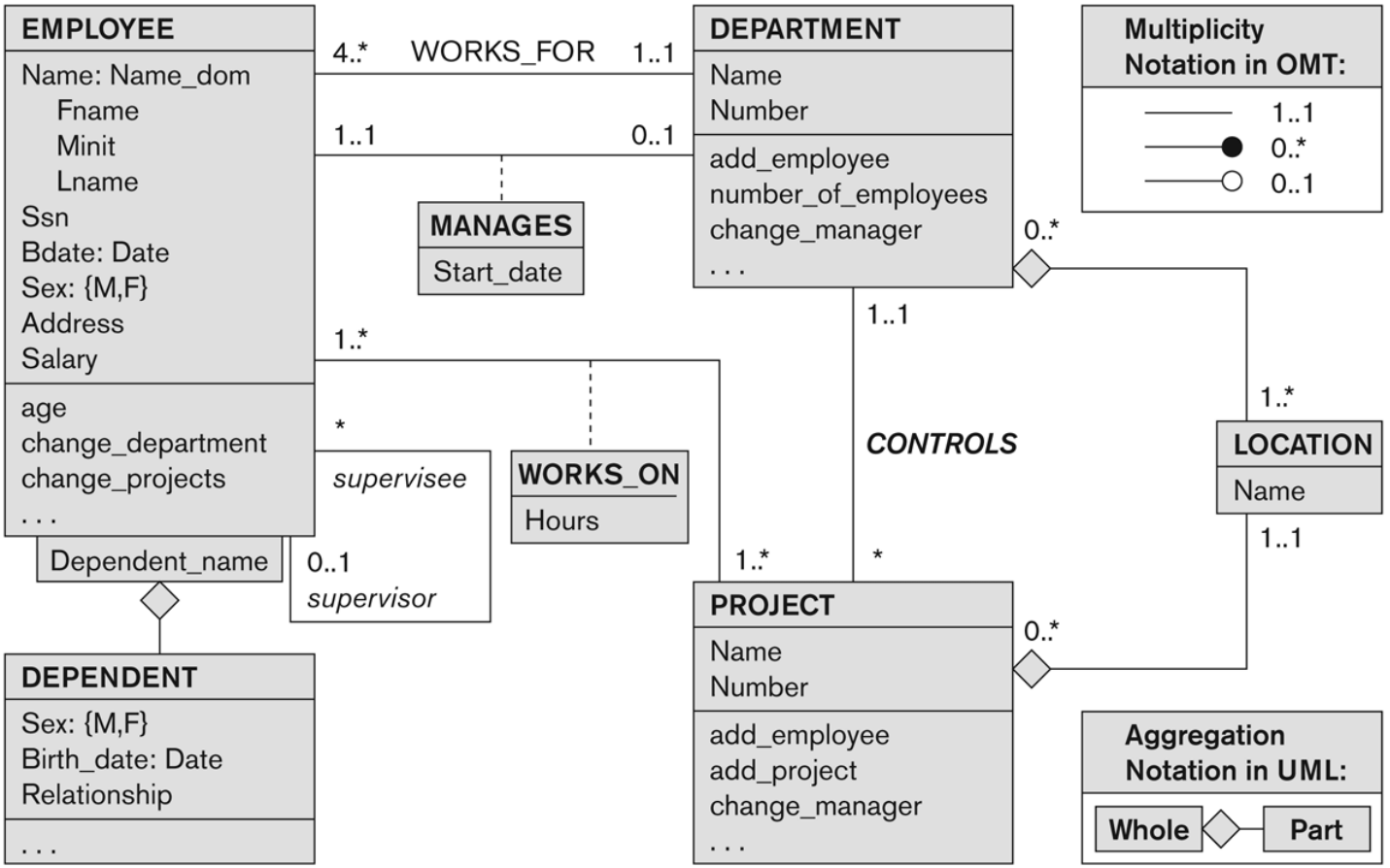


UML CLASS DIAGRAMS

- Relationship types are called **associations** in UML
- Relationship instances are called **links**
- Relationships are represented as lines connecting the classes
- The (min, max) notation is used to specify relationship constraints (**multiplicities)**
  - multiplicities are placed *on the opposite ends of the relationship*
- A recursive relationship is called a **reflexive association** in UML
- Weak entities can be modeled using the construct called **qualified association**



UML CLASS DIAGRAM FOR COMPANY DATABASE SCHEMA



Qualified association for weal entity

Other alternative diagrammatic notations

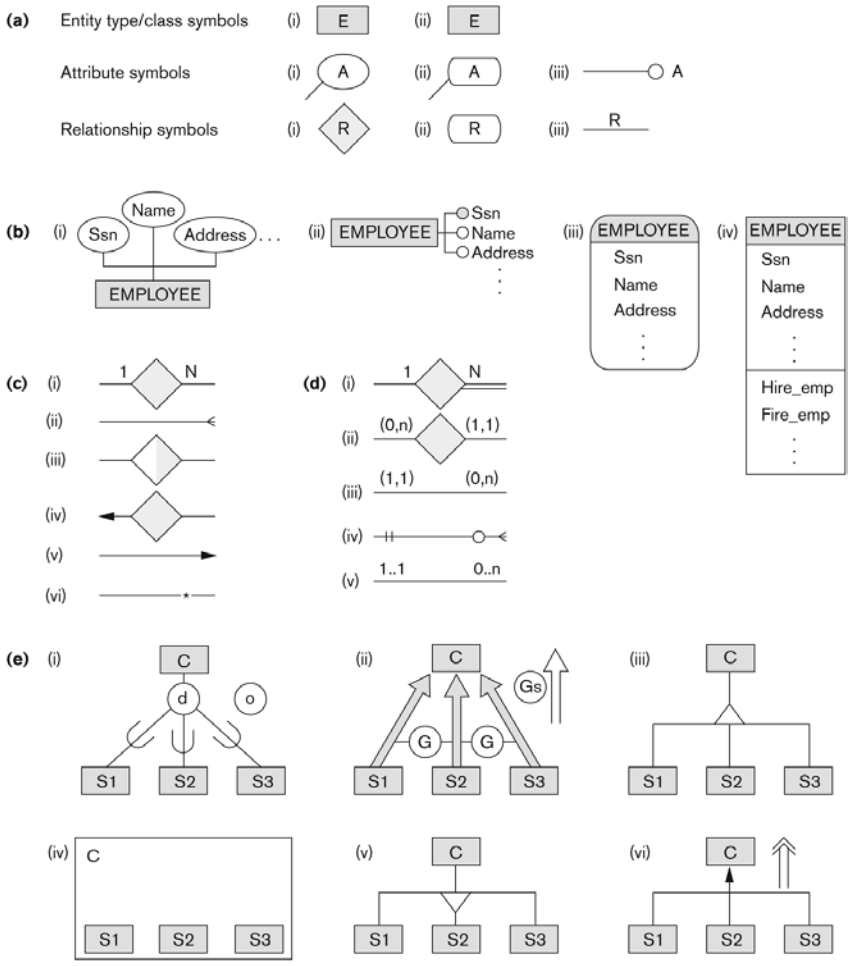


Figure A.1
Alternative notations. (a) Symbols for entity type/class, attribute, and relationship. (b) Displaying attributes. (c) Displaying cardinality ratios. (d) Various (min, max) notations. (e) Notations for displaying specialization/generalization.

Chapter 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
- Alternative Notations – UML class diagrams, others