Data Modeling Using the Entity-Relationship (ER) Model

Lecture 3

Outline

- Database Design Process
- Entity-Relationship Model
- Why Use ER Diagrams In DBMS?
- ER Model - Basic Concepts
    1. Entity - ER Model
    2. Attribute - ER Model
    3. Relationships - ER Model
- ER Diagrams - Notation
- How to Draw ER Diagram?
- Example Database Application (COMPANY)

Database Design Process

- Two main activities:
  - **Database design:** design the conceptual schema for a database application
    - Entity-Relationship Model
    - Relational Model
  - **Application design:** focuses on the programs and interfaces that access the database

Entity-Relationship Diagram (ERD)

- **Entity-Relationship (ER) Diagram:** is a model for identifying entities to be represented in the database and representation of how those entities are related. It also explains the relationship among the entities present in the database.
- They are used to model real-world objects like a person, a car, or a company and the relation between these real-world objects.
- Best used for the conceptual design of a database.
- ER Model is based on:
  - Entities and their attributes.
  - Relationships among entities.

Why Use ER Diagrams In DBMS?

- ER diagrams are used to represent the E-R model in a database, which makes them easy to be converted into relations (tables).
- ER diagrams provide the purpose of real-world modeling of objects which makes them intently useful.
- ER diagrams require no technical knowledge and no hardware support.
- These diagrams are very easy to understand and easy to create even for a naive user.
- It gives a standard solution for visualizing the data logically.

ER Model - Basic Concepts

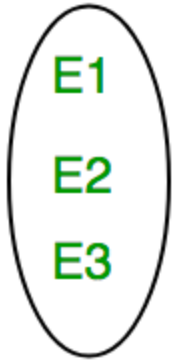
1. Entity
   - A real-world object that can be identifiable.
     - For example: in a school database, students, teachers, classes, and courses offered can be considered as entities.
   - All entities have some attributes or properties that give them their identity.
2. Attribute
   - Entities are represented by means of their properties, called attributes. All attributes have values.
     - For example: a student entity may have name, class, and age as attributes.
   - There exists a domain or range of values that can be assigned to attributes.
     - For example: a student's name cannot be a numeric value. It has to be alphabetic. A student's age cannot be negative, etc.
3. Relationship
   - The association among entities
     - For example: an employee works_at a department, a student enrolls in a course.

1. Entity - ER Model

- Entity Types:
  1. **Strong Entity:** an entity that has a key Attribute. It does not depend on other Entity in the Schema. It has a primary key, that helps in identifying it uniquely.

Entity Type

  2. **Weak Entity:** an entity for which a key attributes can't be defined.
     - For Example, A company may store the information of dependents (Parents, Children, Spouse) of an Employee. But the dependents don't have existed without the employee. So Dependent will be a Weak Entity Type and Employee will be Identifying Entity type for Dependent, which means it is Strong Entity Type.

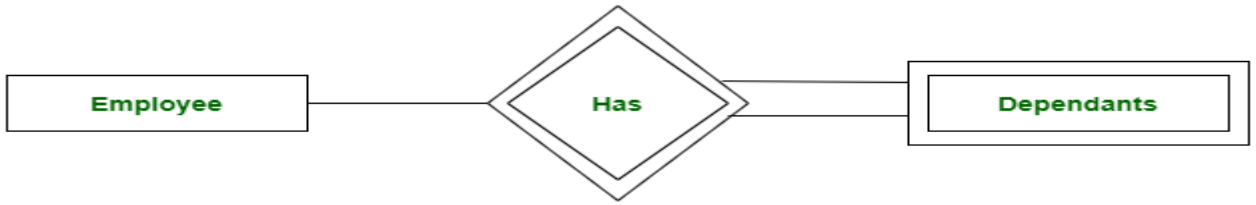


Entity Set

1. Entity - ER Model

- Representation of entities:
  1. **Strong Entity:** represented by a rectangle. Rectangles are named with the entity set they represent.

  2. **Weak Entity:** represented by a Double Rectangle. The relationship between the weak entity type and its identifying strong entity type is represented by a double diamond.



Strong Entity and Weak Entity

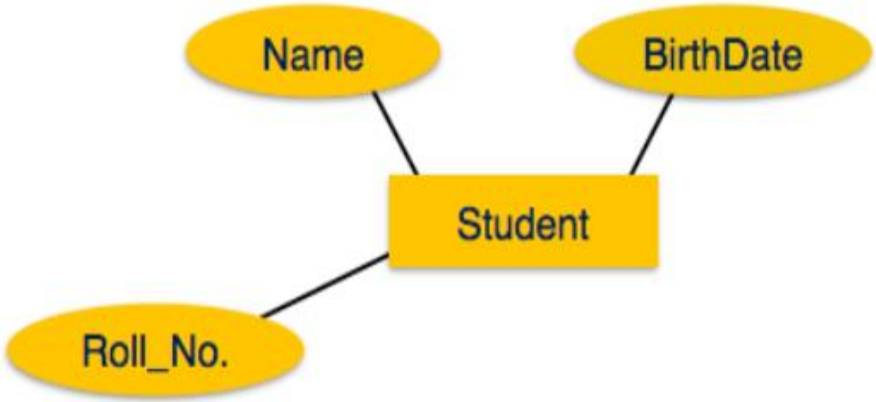
1. Entity - ER Model

Difference between Strong and Weak Entity:

Strong Entity	Weak Entity
Strong entity always has a primary key.	While a weak entity has a partial discriminator key.
Strong entity is not dependent on any other entity.	Weak entity depends on strong entity.
Strong entity is represented by a single rectangle.	Weak entity is represented by a double rectangle.
Two strong entity's relationship is represented by a single diamond.	While the relation between one strong and one weak entity is represented by a double diamond.
Strong entities have either total participation or not.	While weak entity always has total participation

2. Attribute - ER Model

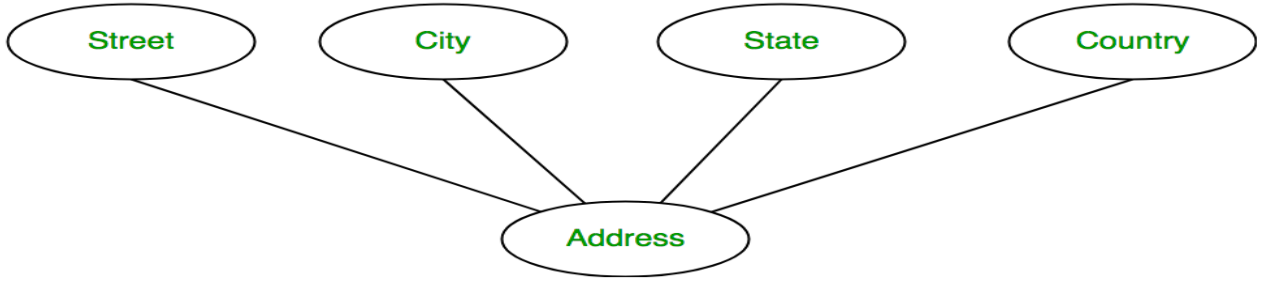
- Attributes are the properties of entities. Attributes are represented by means of ellipses.
- Every ellipse represents one attribute and is directly connected to its entity (rectangle).



2. Attribute - ER Model

- Attributes types:
  1. **Key Attribute:** uniquely identifies each entity in the entity set.

     - For example: Roll_No will be unique for each student.
     - In ER diagram, the key attribute is represented by an oval with underlying lines.
  2. **Composite Attribute:** an attribute composed of many other attributes.
     - For example: the Address attribute of the student Entity type consists of Street, City, State, and Country.
     - In ER diagram, the composite attribute is represented by an oval comprising of ovals.



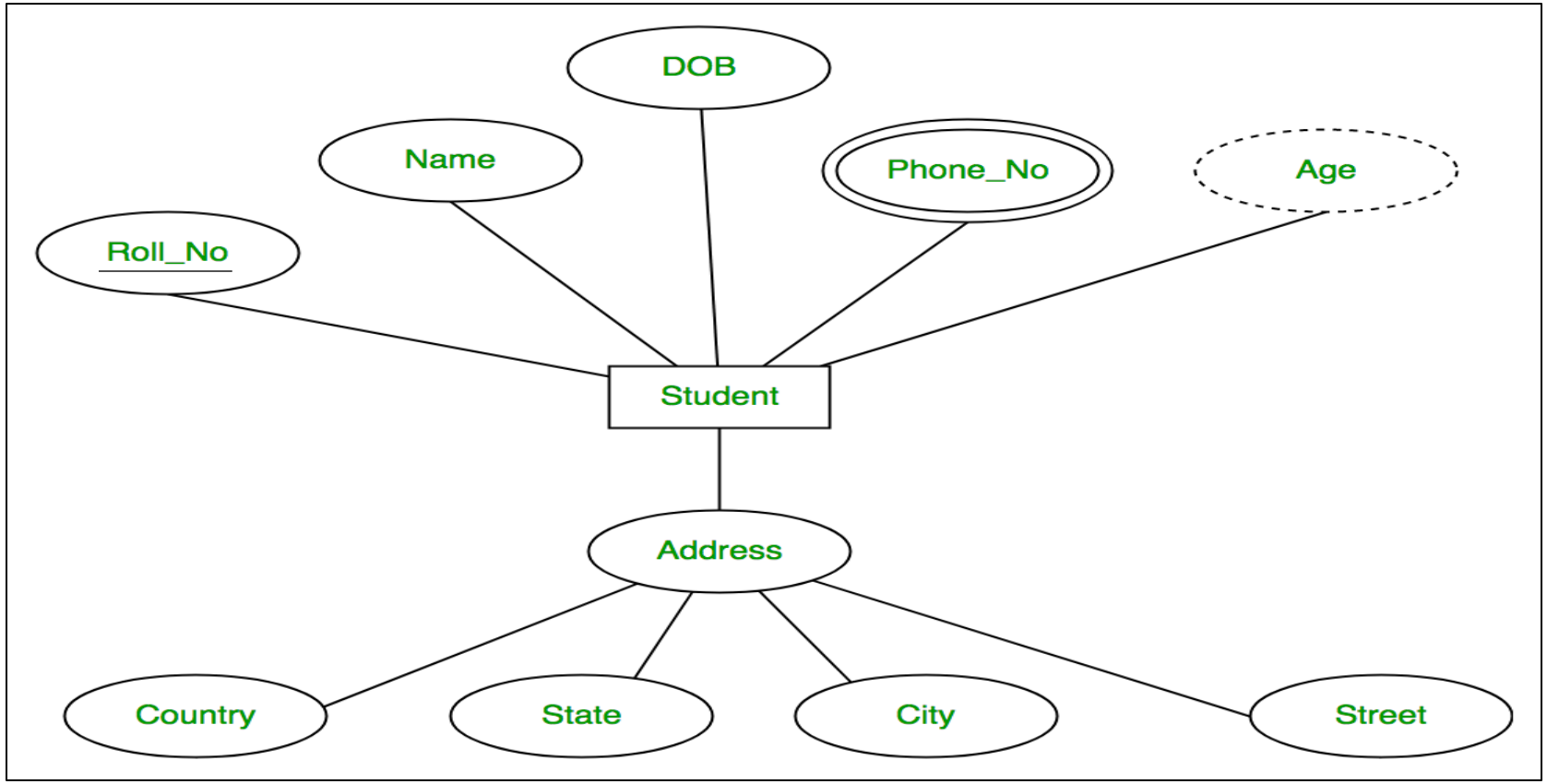
2. Attribute - ER Model

- Attributes types:
  3. **Multivalued Attribute:** An attribute consisting of more than one value for a given entity.
     - For example: Phone_No (can be more than one for a given student).
     - In ER diagram, a multivalued attribute is represented by a double oval.

  4. **Derived Attribute:** An attribute that can be derived from other attributes of the entity type.
     - For example, Age (can be derived from DOB).
     - In ER diagram, the derived attribute is represented by a dashed oval.

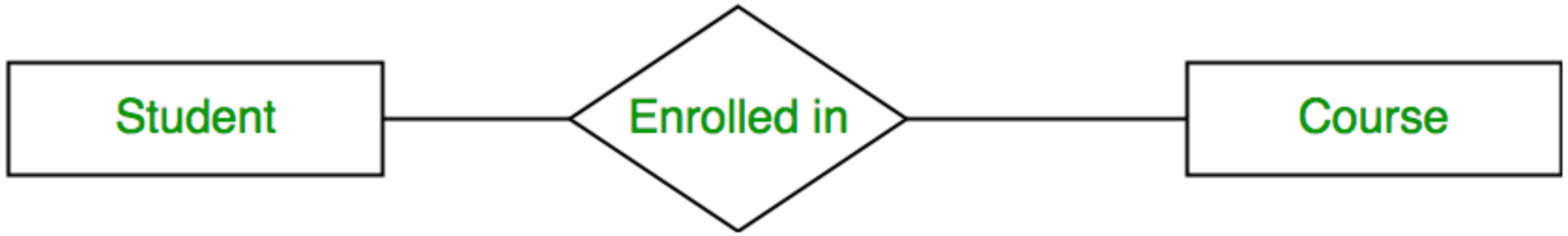
2. Attribute - ER Model

- Entity Student with its attributes can be represented as follows:



3. Relationships - ER Model

- A Relationship represents the association between entities.
  - For example, ‘Enrolled in’ is a relationship that exists between entity Student and Course.
  - In ER diagram, the relationship type is represented by a diamond and connecting the entities with lines.

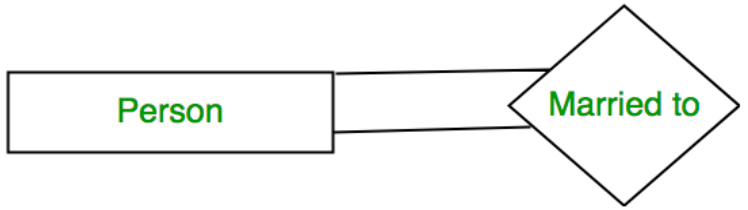


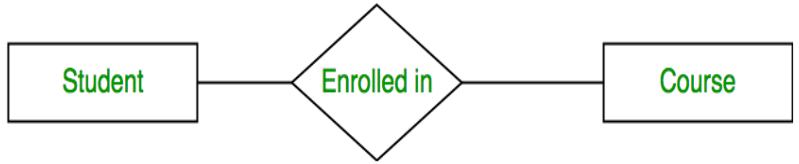
3. Relationships - ER Model

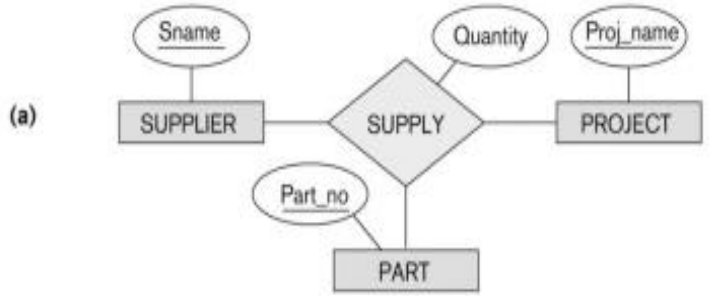
- A recursive relationship: is a relationship between the same participating entity in distinct roles.
  - Example: the SUPERVISION relationship where EMPLOYEE participates twice in two distinct roles: supervisor (or boss) role and supervisee (or subordinate) role
  - Each relationship instance relates two distinct EMPLOYEE entities: One employee in supervisor role and One employee in supervisee role

3. Relationships - ER Model

- Degree of a Relationship: the number of entities participating in a relationship.
  1. **Unary Relationship:** when there is only ONE entity set participating in a relation, the relationship is called a unary relationship. For example, one person is married to only one person.
  2. **Binary Relationship:** when there are TWO entities set participating in a relationship, the relationship is called a binary relationship. For example, a Student is enrolled in a Course.
  3. **n-ary Relationship:** when there are n entities set participating in a relation, the relationship is called an n-ary relationship.

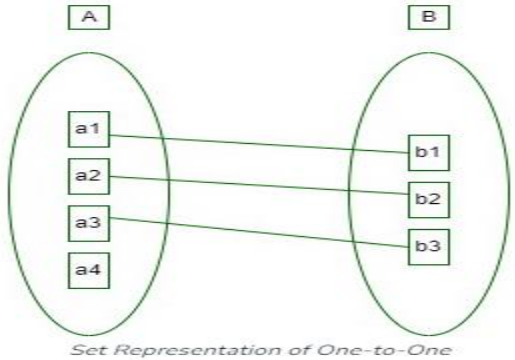






3. Relationships - ER Model

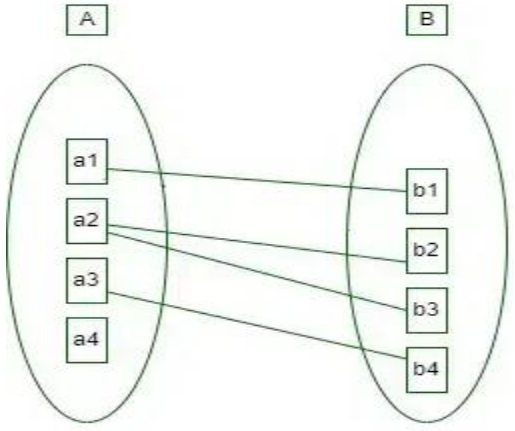
- Relationship Cardinality: The number of times an entity of an entity set participates in a relationship.
- Cardinality can be of different types:
  1. **One-to-One:** When each entity in each entity set can take part only once in the relationship.
     1. Assume that a male can marry one female and a female can marry one male.



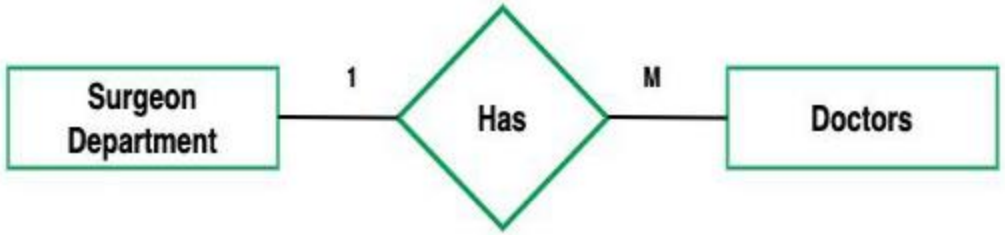
Set Representation of One-to-One

3. Relationships - ER Model

- Cardinality can be of different types:
  2. **One-to-Many:** when more than one instance of an entity is associated with a relationship, it is marked as '1:N'.
     - Assume that one department can accommodate many doctors. So the Cardinality will be 1 to M. It means one department has many Doctors.

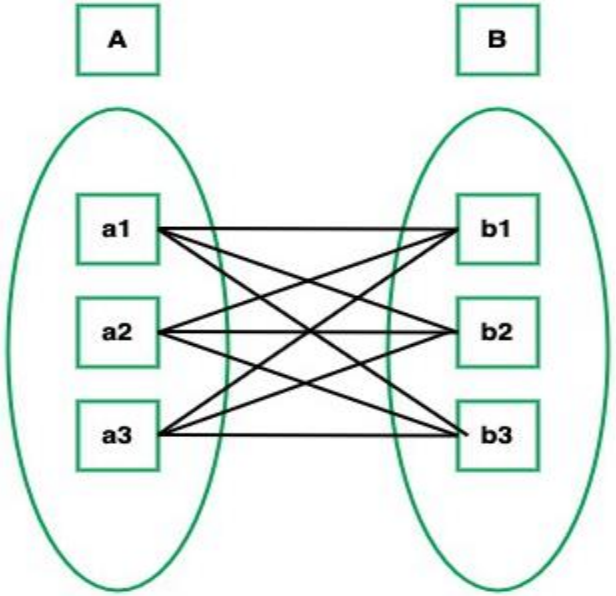


Set Representation of One-to-Many



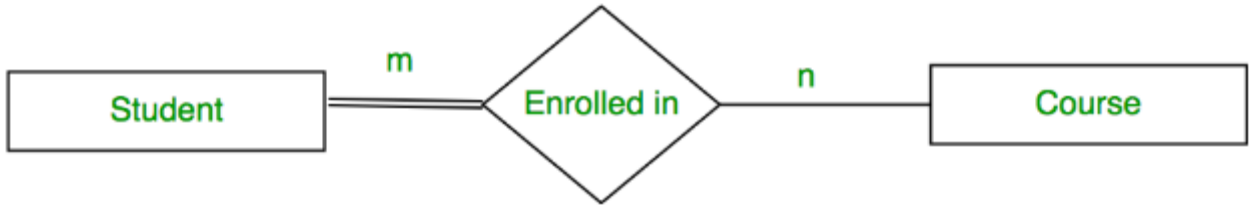
3. Relationships - ER Model

- Cardinality can be of different types:
  3. **Many-to-Many:** when entities in all entity sets can take part more than once in the relationship.
     - Assume that a student can take more than one course and one course can be taken by many students. So the relationship will be many to many.



3. Relationships - ER Model

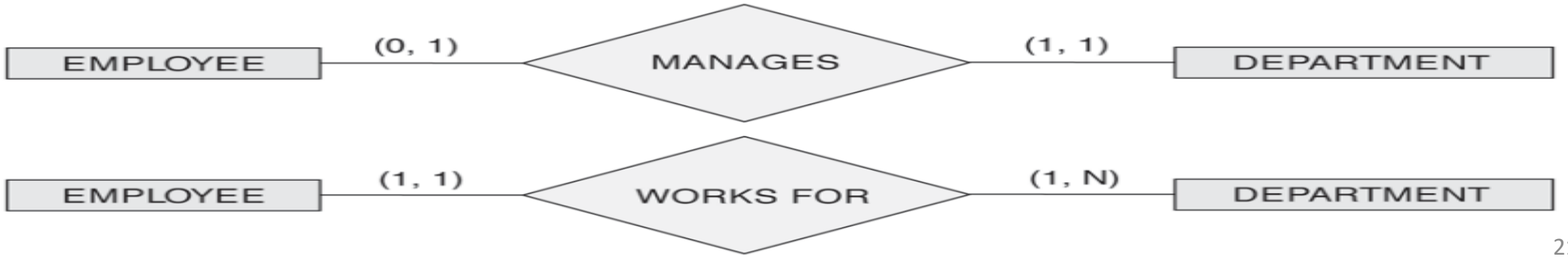
- Participation Constraint:
  1. **Total Participation:** each entity in the entity set must participate in the relationship.
     - If each student must enroll in a course, the participation of students will be total.
     - Total participation is shown by a double line in the ER diagram.
  2. **Partial Participation:** the entity in the entity set may or may NOT participate in the relationship.
     - If some courses are not enrolled by any of the students, the participation in the course will be partial.



Total Participation and Partial Participation

3. Relationships - ER Model

- <u>(min, max) notation for relationship structural constraints:</u>
  - Specified on each participation of an entity E in a relationship R.
  - Specifies that each entity E participates in at least min and at most max relationship instances in R
  - 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

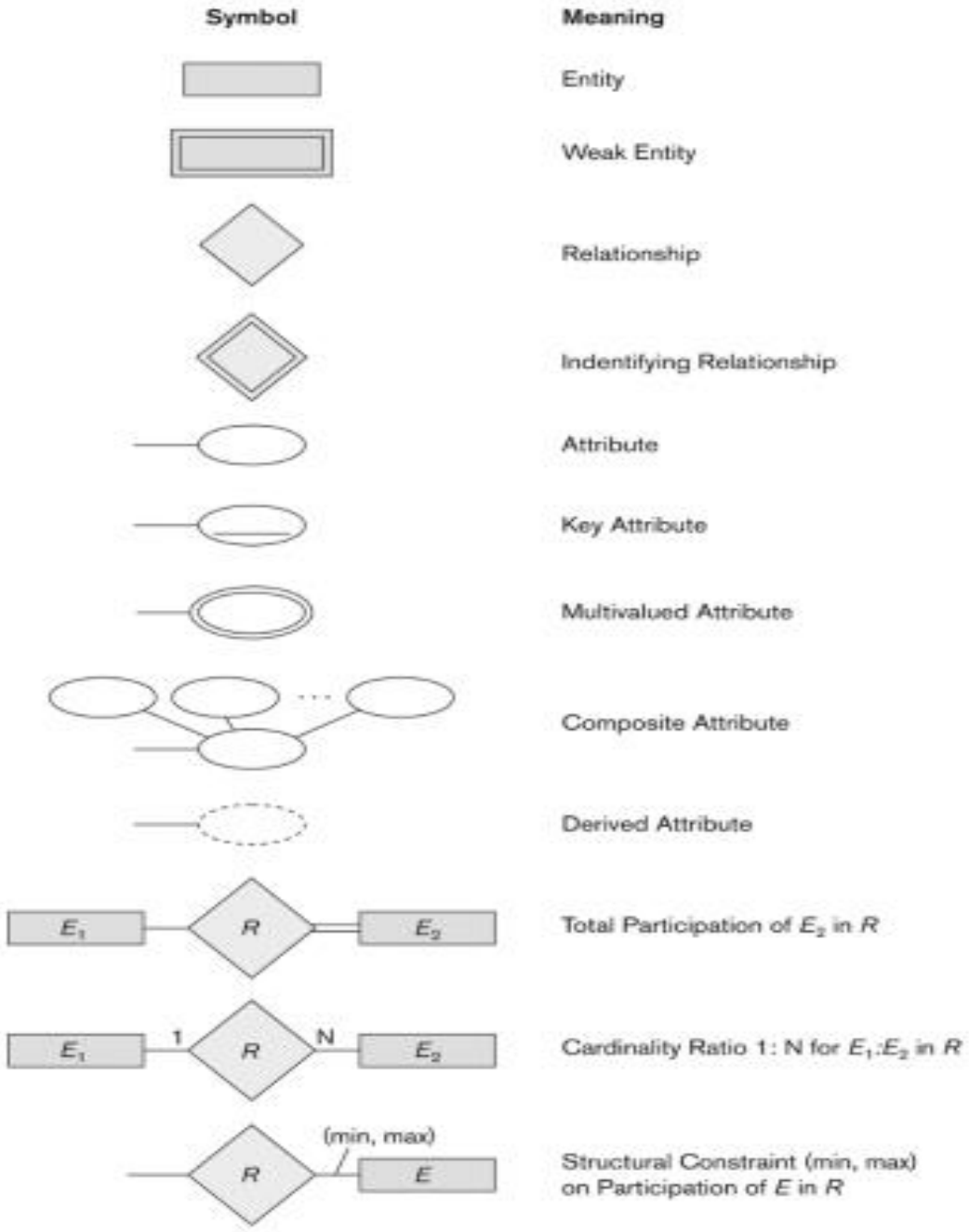


3. Relationships - ER Model

- A relationship can have attributes:
  - 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

NOTATION for ER Diagrams

Figure 3.14
Summary of the notation for ER diagrams.



How to Draw ER Diagram?

1. Identifying all the Entities, and place them in a Rectangle, and labeling them accordingly.
2. Identify the relationship between them and pace them accordingly using the Diamond, and make sure that, Relationships are not connected to each other.
3. Attach attributes to the entities properly.
4. Remove redundant entities and relationships.
5. Add proper colors to highlight the data present in the database.

Example COMPANY Database

- We need to create a database schema design based on the following 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.
  - The database will store each EMPLOYEE's social security number, address, salary, sex, 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, sex, birthdate, and relationship to the employee.

Initial Conceptual Design of Entity Types for the COMPANY Database Schema

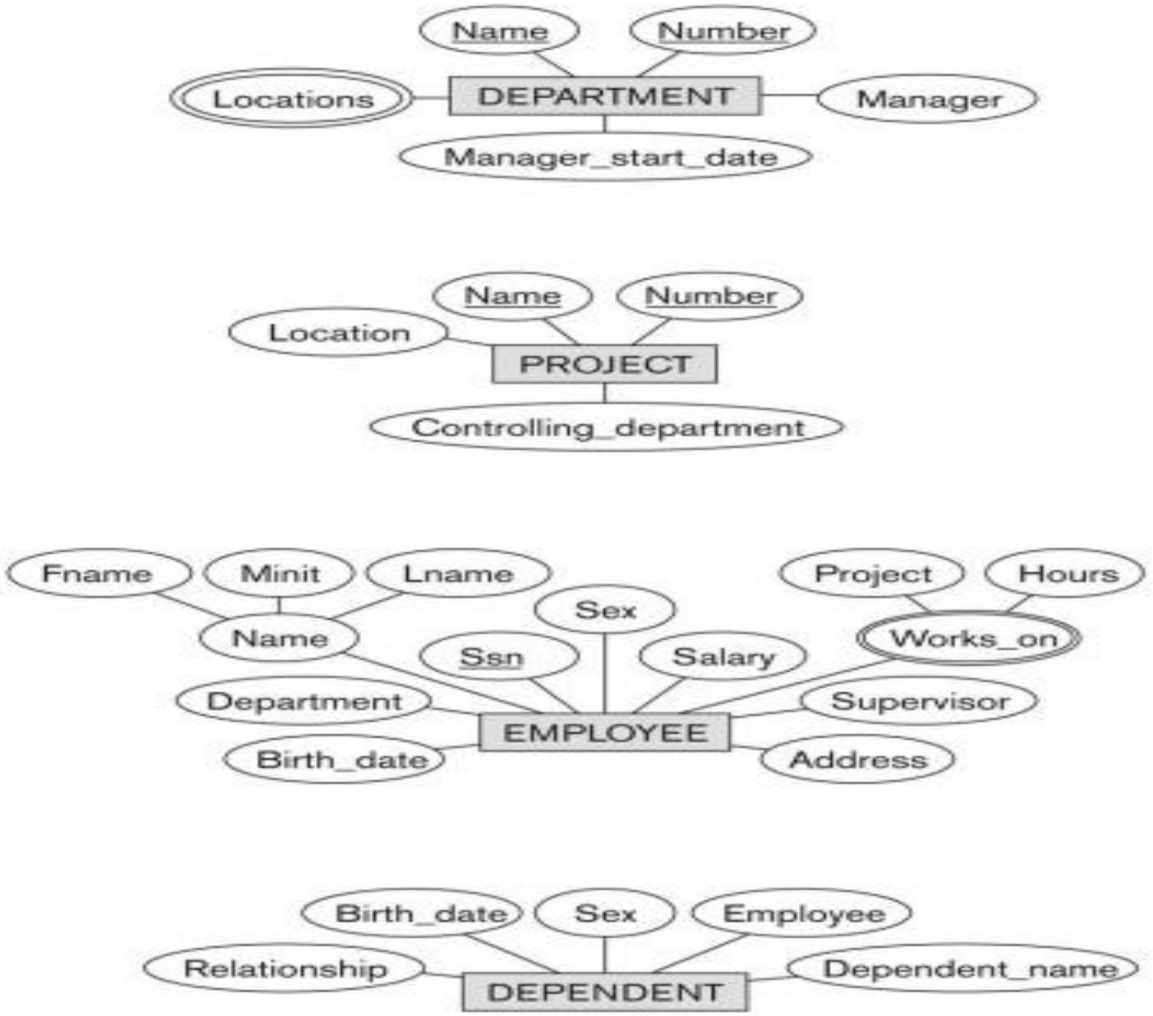


Figure 3.8
Preliminary design of entity types for the COMPANY database. Some of the shown attributes will be refined into relationships.

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 for the COMPANY Database

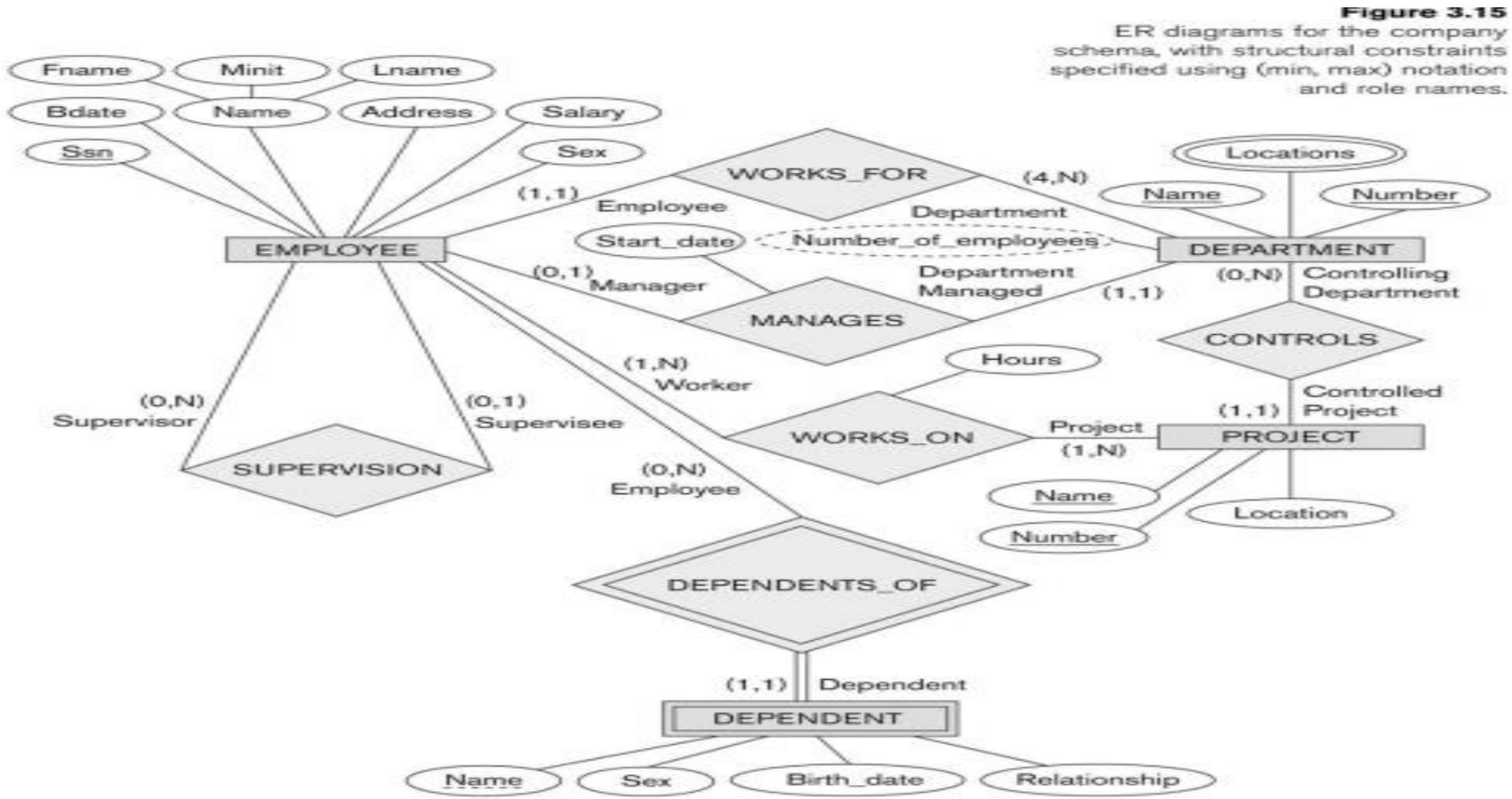


Figure 3.15
ER diagrams for the company schema, with structural constraints specified using (min, max) notation and role names.