Kabul Polytechnic University
Computer Science Faculty
Information System Department

Database II

Lecture 1: Entity Relationship (ER) Model

01 September – 2024

Lectures & other related materials are available here: https://github.com/mujtabaSultani01/Database-II

Contents

- Entity Relationship (ER) – Model
- ER – Model Elements
- Entity Type Classification
- Attribute Types
- Identifiers (Keys)
- Basic ER Notations
- Relationship

Entity Relationship (ER) Model

E-R Model

- A **logical representation** of the data for an organization using:
  - **Entities** for categories of data and,
  - **Relationships** for associations between entities.

Elements of ER Model

- **Entity**
  - Person, place, object, event or concept for which data is collected and maintained. (**A table**)
- **Attribute**
  - The Property or characteristic of an entity. (**A field in a table**)
- **Relationship**
  - A logical connection between different entities.

Note that:

- **Entity Type/Class:** A set of related attributes.
- **Entity Instance:** is the member of entity type.
- **Entity Set:** is a group of entity instances

What should an Entity Be?

SHOULD BE:

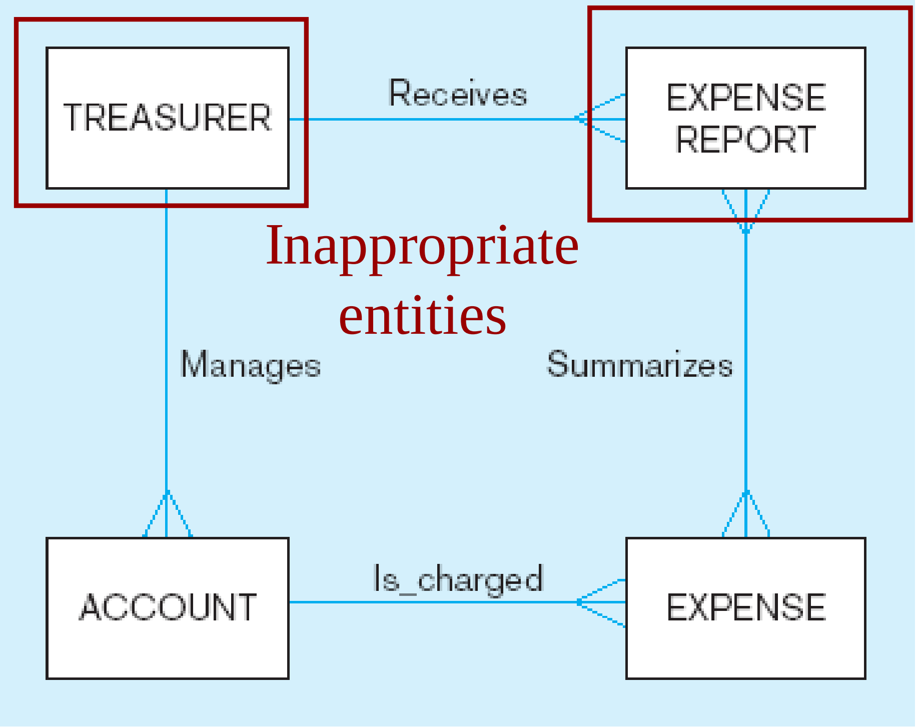
- ✓ An object that will have many instances in the database.
- ✓ An object that will be composed of multiple attributes.
- ✓ An object that we are trying to model.

SHOULD NOT BE:

- A user of the database system.
- An output of the database system (e.g., a report).

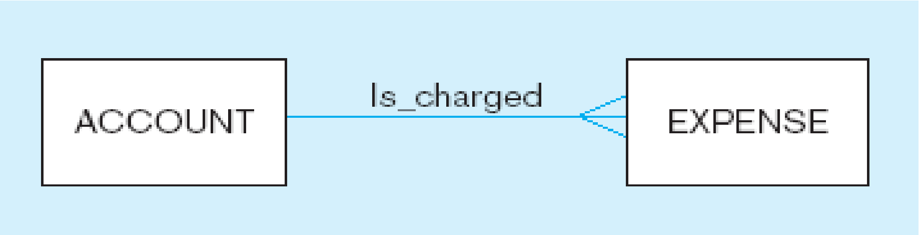
Inappropriate Entities

System user



Inappropriate entities

System output



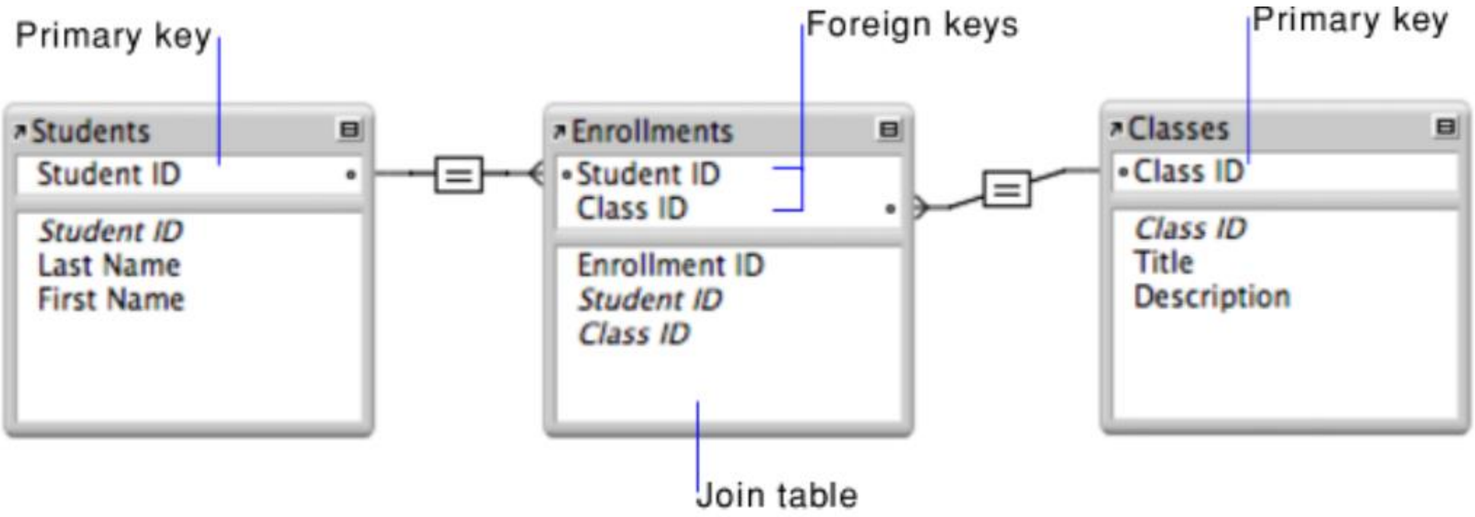
Appropriate entities

Entity Type Classification

- **Strong Entity Type:**
  - An entity that exists independently of other entity types.
  - Also called Independent Entity Type.
- **Weak Entity Type:**
  - An entity type whose existence depends on some other entity type.
  - Also called Dependent Entity Type.
  - May have no key attribute.
  - For example; RESERVATION is strong but RESERVATION_DETAILS is weak Entity Type.

Entity Type Classification

- **Associative Entity Type:**
  - An entity type that **associates the instances** of one or more entity types with one another.



Attribute Types (1)

Attribute: Property or characteristic of an entity.

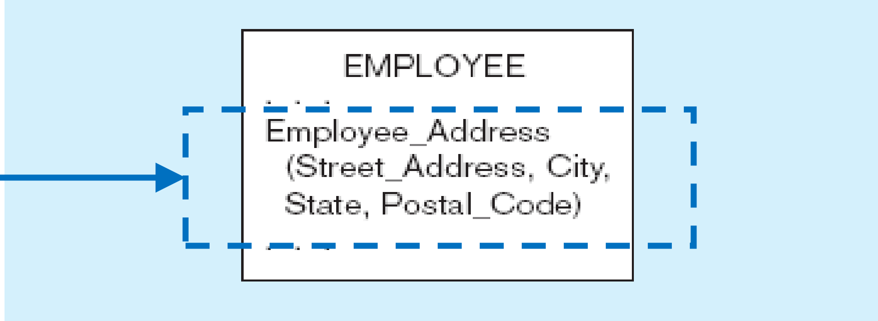
- **Composite Attribute:**
  - Can be divided into smaller subparts, which represent more basic attributes with independent meanings.
  - For example, **Address** or **Name**.
- **Simple (Atomic) Attribute:**
  - Cannot be divided into smaller components.
  - A single atomic value for the attribute.
  - For example, **first name**, **month**, **ID**.

Attribute Types (2)

- **Multi Valued Attribute:**
  - An entity may have multiple values for that attribute.
  - For example, **Skills of an employee** or **subjects**.
- **Derived Attribute:**
  - An attribute whose values can be calculated from related attribute values.
  - For example, **Age** is derived from date of **birth of an employee**.
- **Identifier Attribute:**
  - An attribute whose value separates instances of an entity type.
  - For example, **Employee_ID**.

Attribute Types (3)

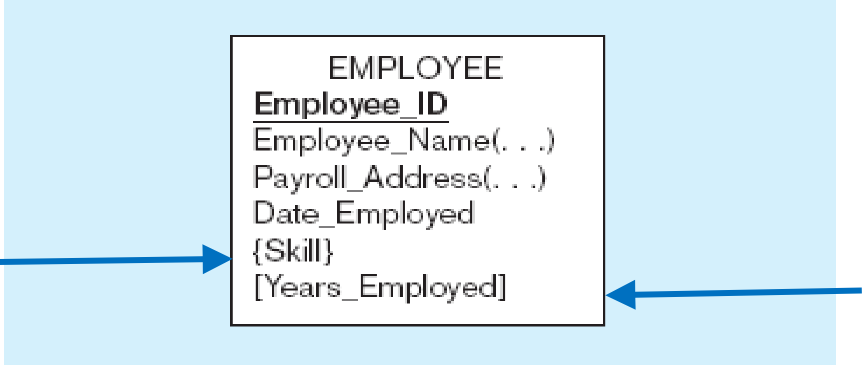
Composite Attribute:
An attribute broken into component parts.



EMPLOYEE
Employee_Address
(Street_Address, City, State, Postal_Code)

Entity with **multivalued** attribute (Skill) and **derived** attribute (Years_Employed)

Multivalued
an employee can have more than one skill.

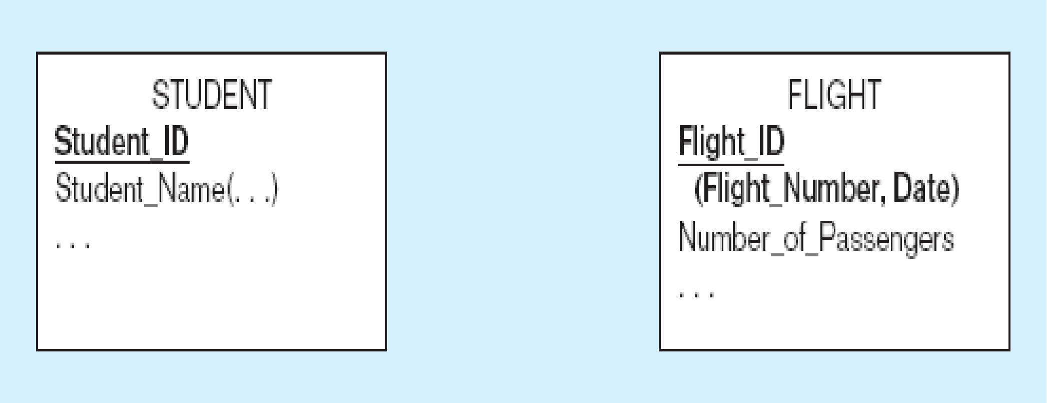


EMPLOYEE
Employee_ID
Employee_Name(. . .)
Payroll_Address(. . .)
Date_Employed
{Skill}
[Years_Employed]

Derived
from date employed and current date

Identifiers (Keys)

- **Identifier (Key):** An attribute (or combination of attributes) that uniquely identifies individual instances of an entity type.
- **Simple** versus **Composite** Identifier.



STUDENT
Student_ID
Student_Name(. . .)
. . .

FLIGHT
Flight_ID
(Flight_Number, Date)
Number_of_Passengers
. . .

(a) Simple identifier attribute

(b) Composite identifier attribute

- **Candidate Identifier:** An attribute that could be a key satisfies the requirements for being an identifier.

Characteristics of Identifiers

i. Will not change in value

ii. Will not be null

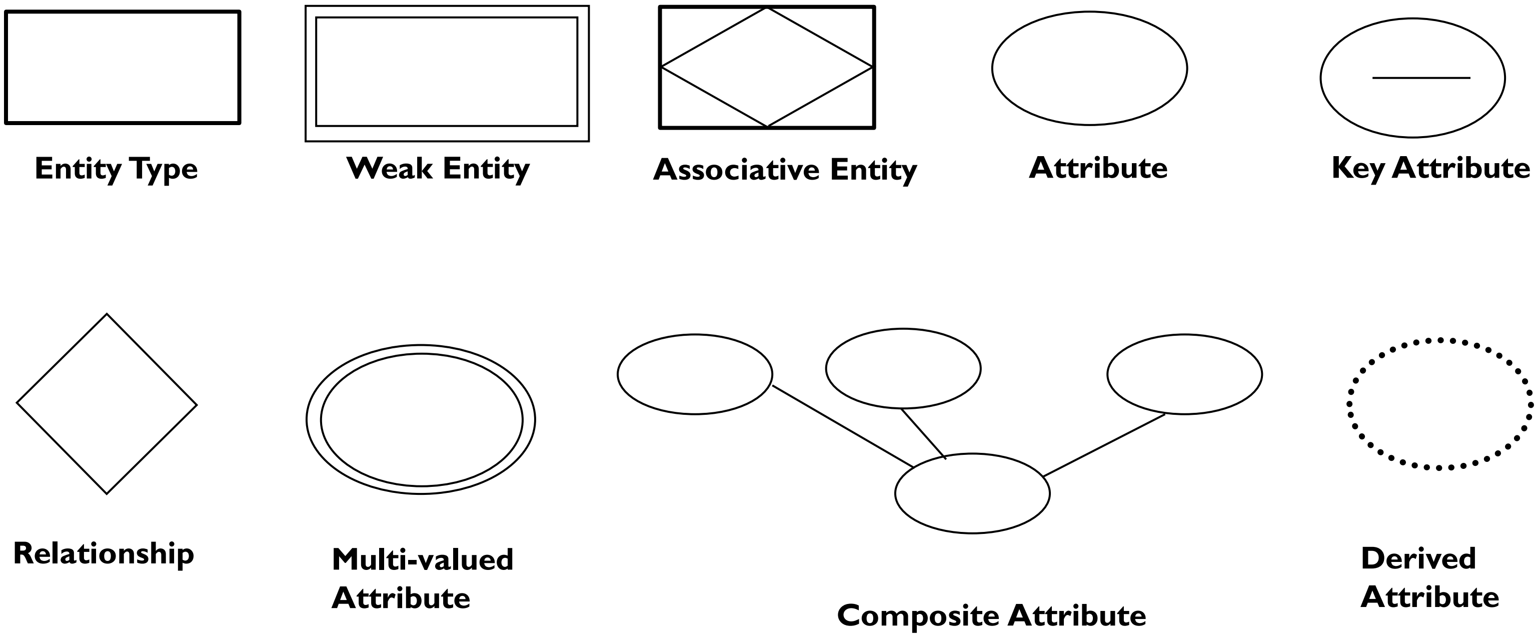
iii. No intelligent identifiers

    (e.g., containing locations or people that might change)

iv. Simple keys, composite keys

Basic ER Notations

- **ER-D** is a graphical presentation of an entity-relationship model.
- ***ER Notations:***



Entity Type

Weak Entity

Associative Entity

Attribute

Key Attribute

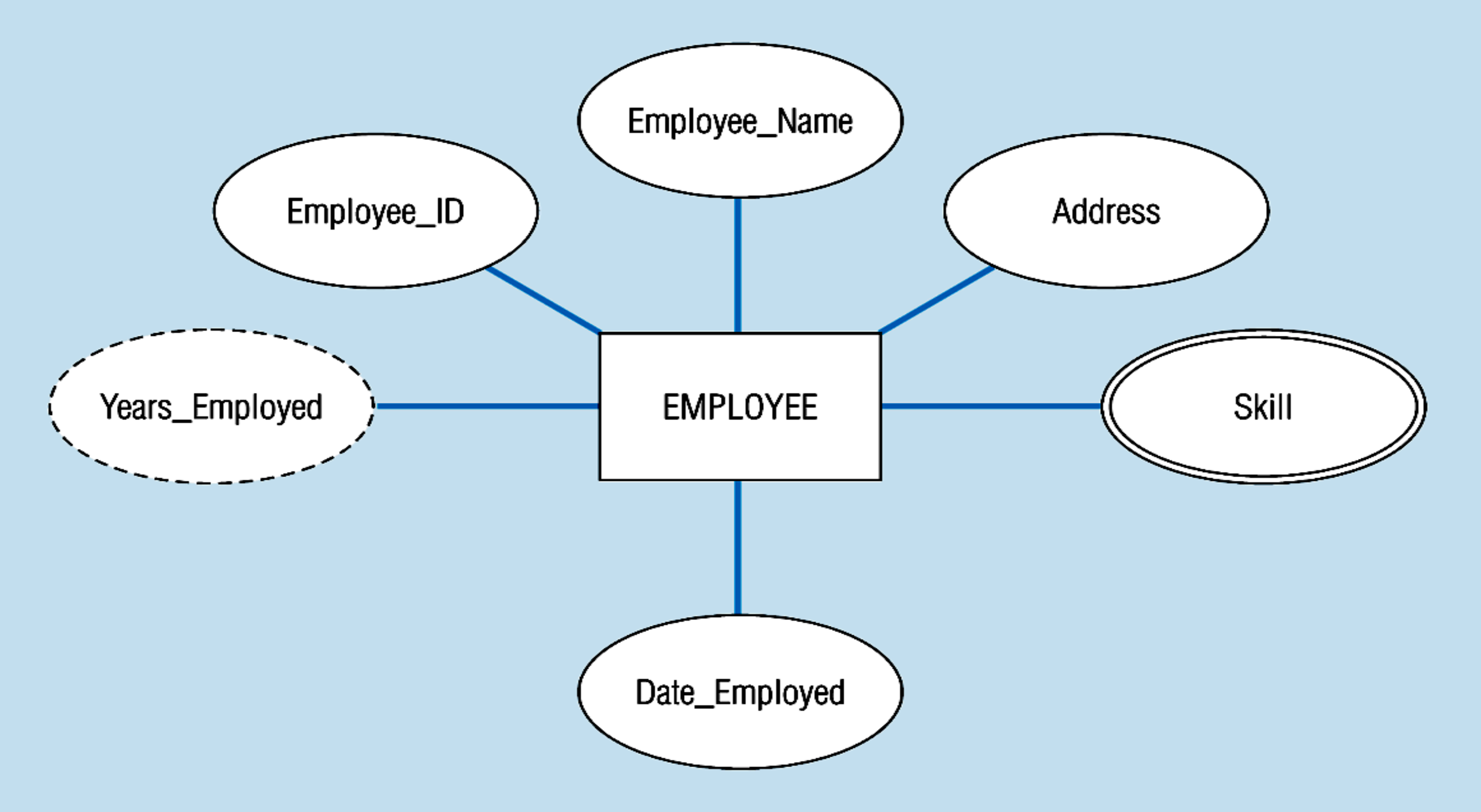
Relationship

Multi-valued Attribute

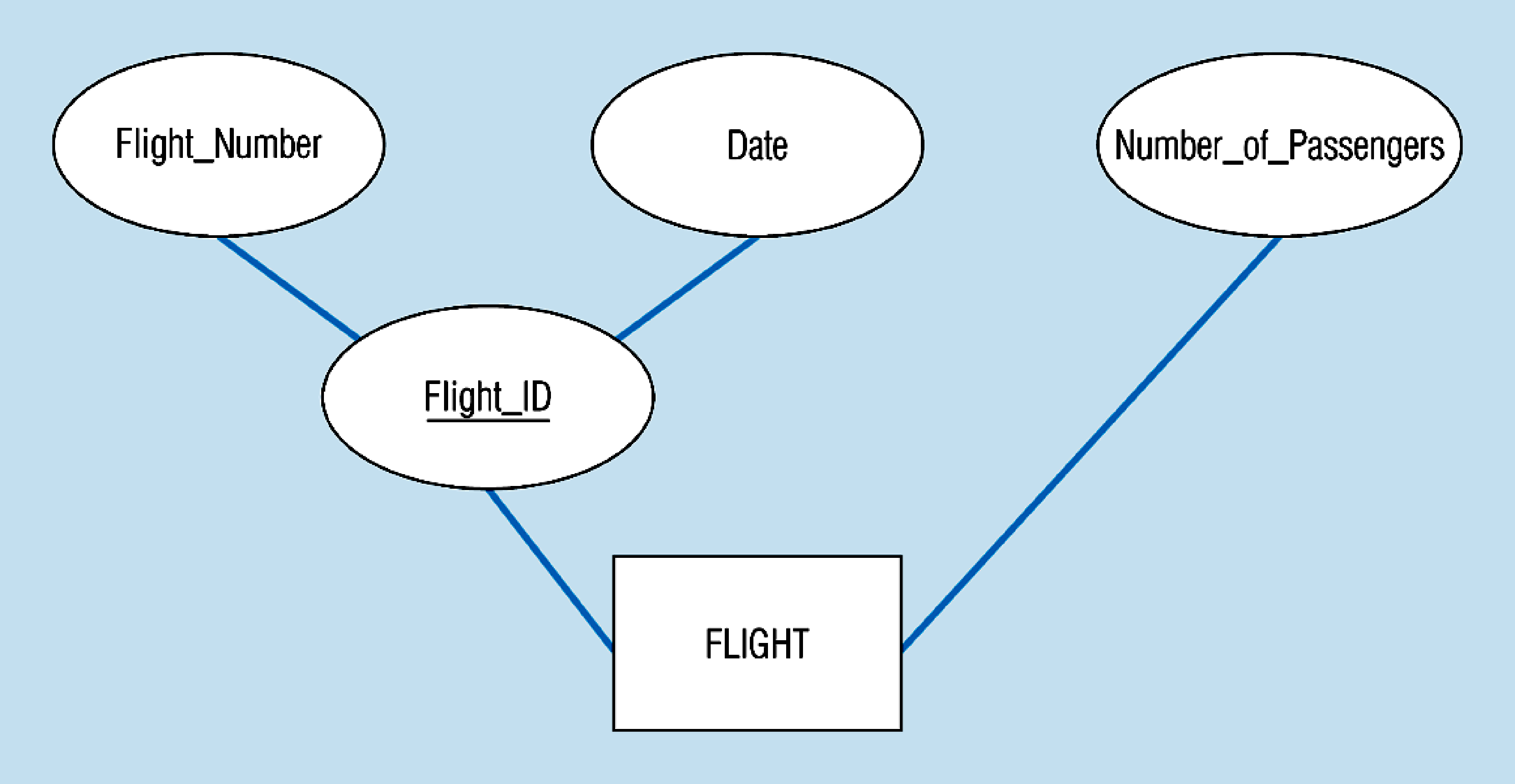
Composite Attribute

Derived Attribute

Example1: Entity Type & Attributes



Example2: Entity Type & Attributes

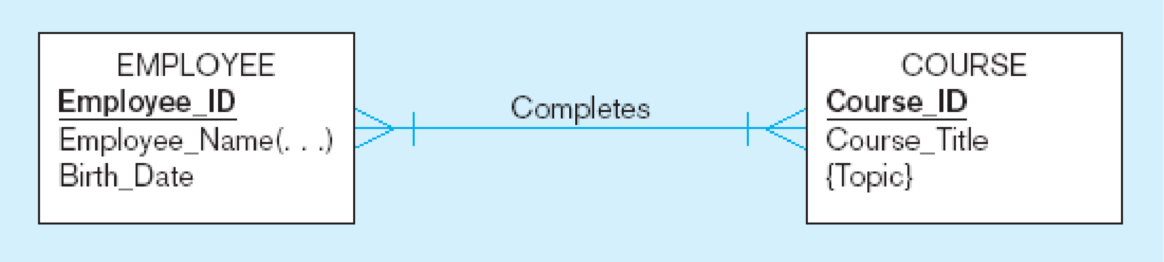


Relationship

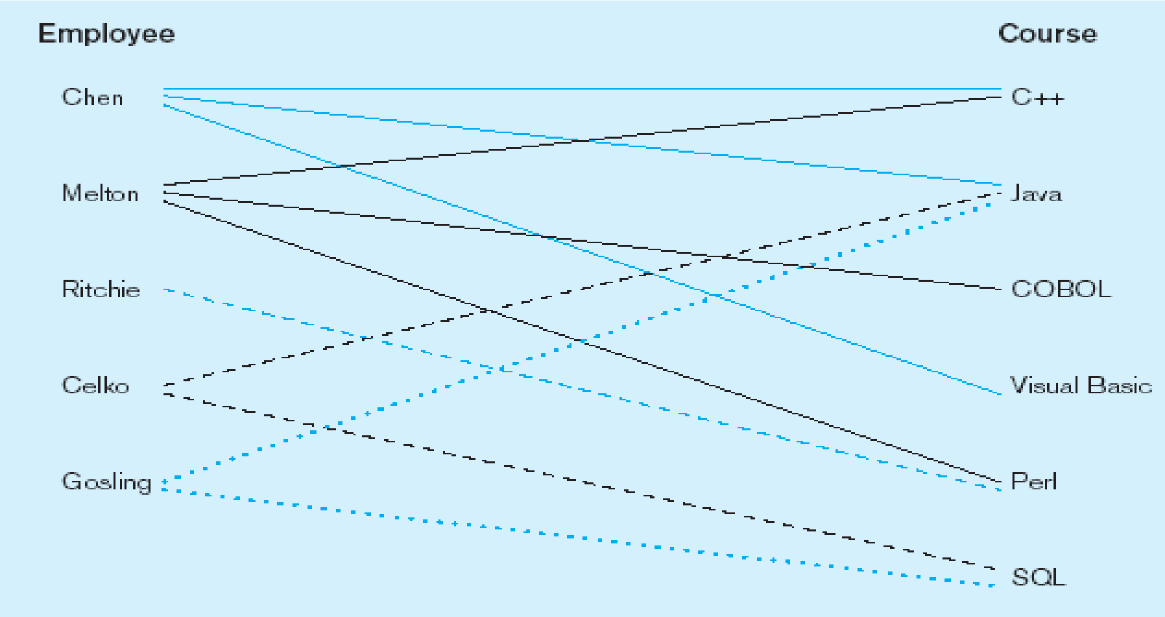
- An association between two or more entities types with a specific meaning.
  - For example,The **EMPLOYEE Khan** works on the **Education PROJECT**.
  - A relationship has a **name** and **degree** for confirmation.
  - Relationship is represented by:
    - **Diamond-shaped box** is used to display a relationship type.
    - Connected to the participating entity types via **straight lines.**
  - The entities having relationship are called **participating entities**.

Relationship

a) Relationship type



b) Relationship instances



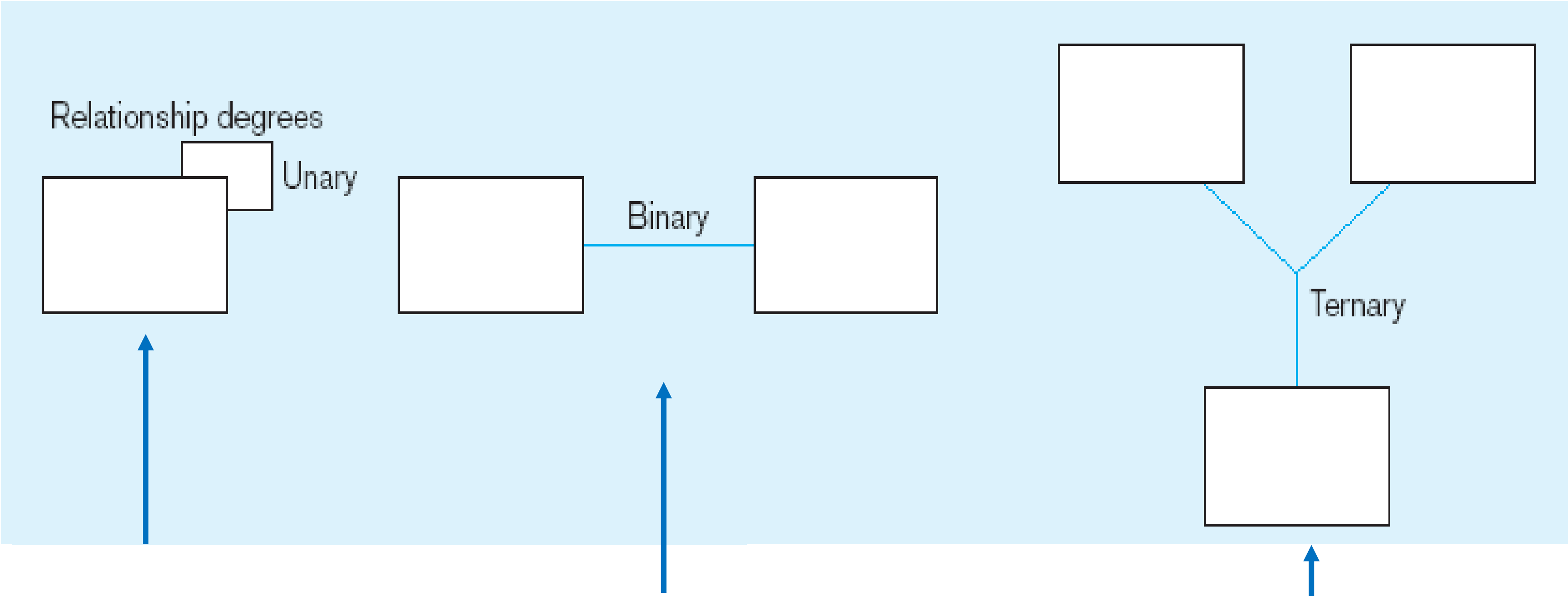
Degree of Relationship (1)

- The number of entity types that participate in a relationship.

Types of Relationships:

- ✓ **Unary Relationship**
- ✓ **Binary Relationship**
- ✓ **Ternary Relationship**

Degree of Relationship (2)



A relationship between instances of single entity type. Also called recursive relationship.

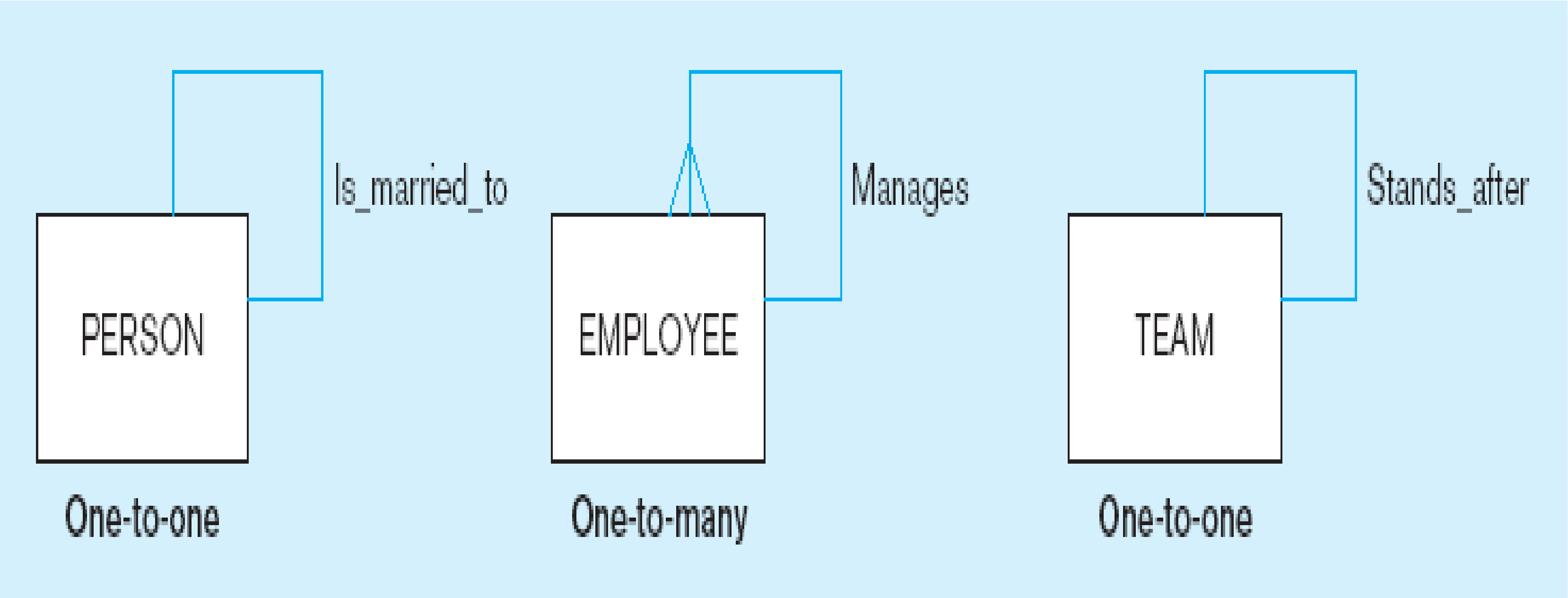
A relationship between the instances of two entity types.

A relationship between the instances of three entity types.

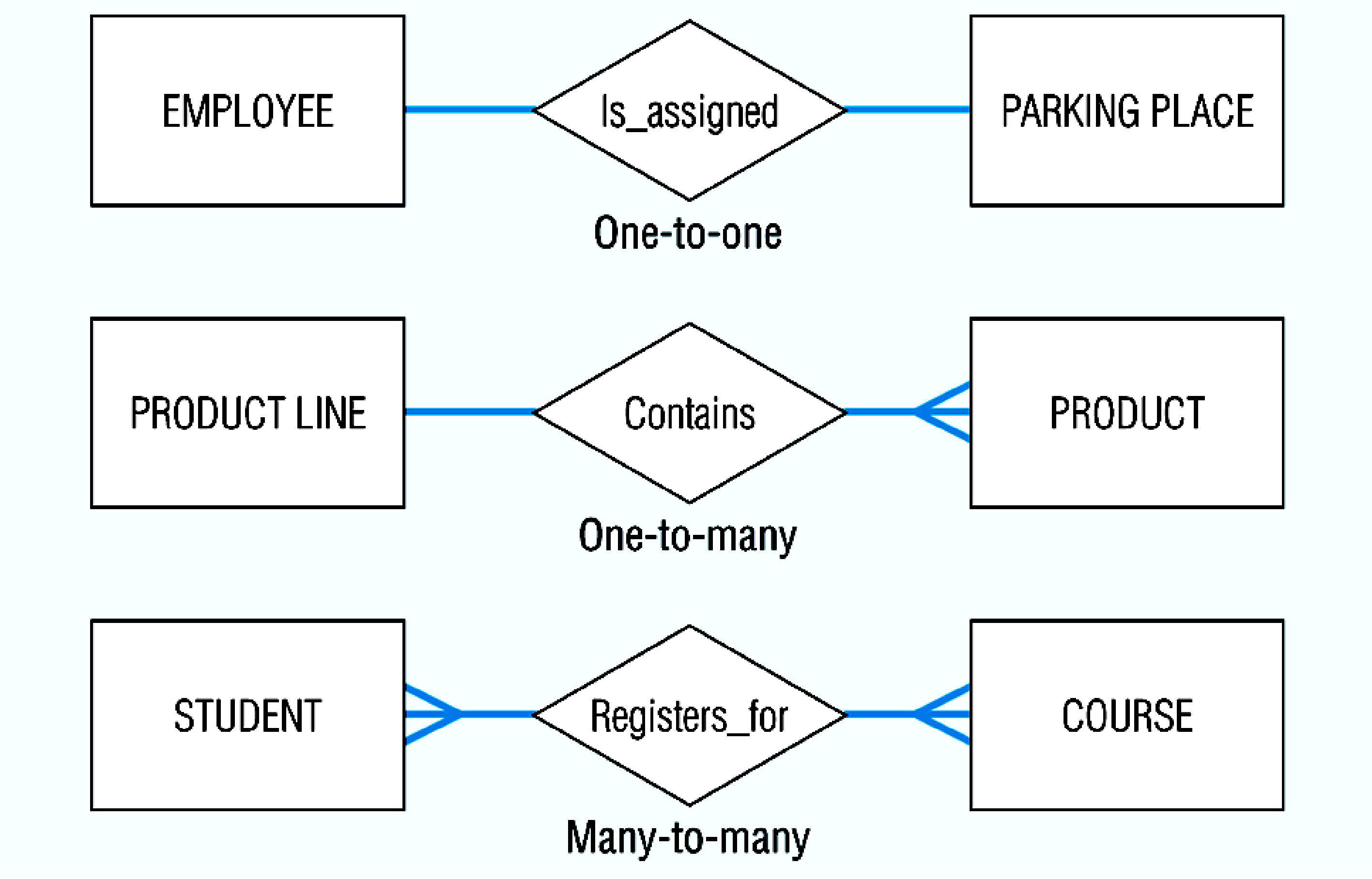
Cardinality of Relationship

- The maximum number of relations between entities.
  - ✓**One – to – One (1:1)**
    - Both entities of the relationship can have one related instance. (teacher - department)
  - ✓**One – to – Many (1:N)**
    - An entity can have many related instances, but the other entity will have a maximum of one related instance. (teacher - courses)
  - ✓**Many – to – Many (M:N)**
    - Both entities of the relationship can have many related instances. (student - courses)

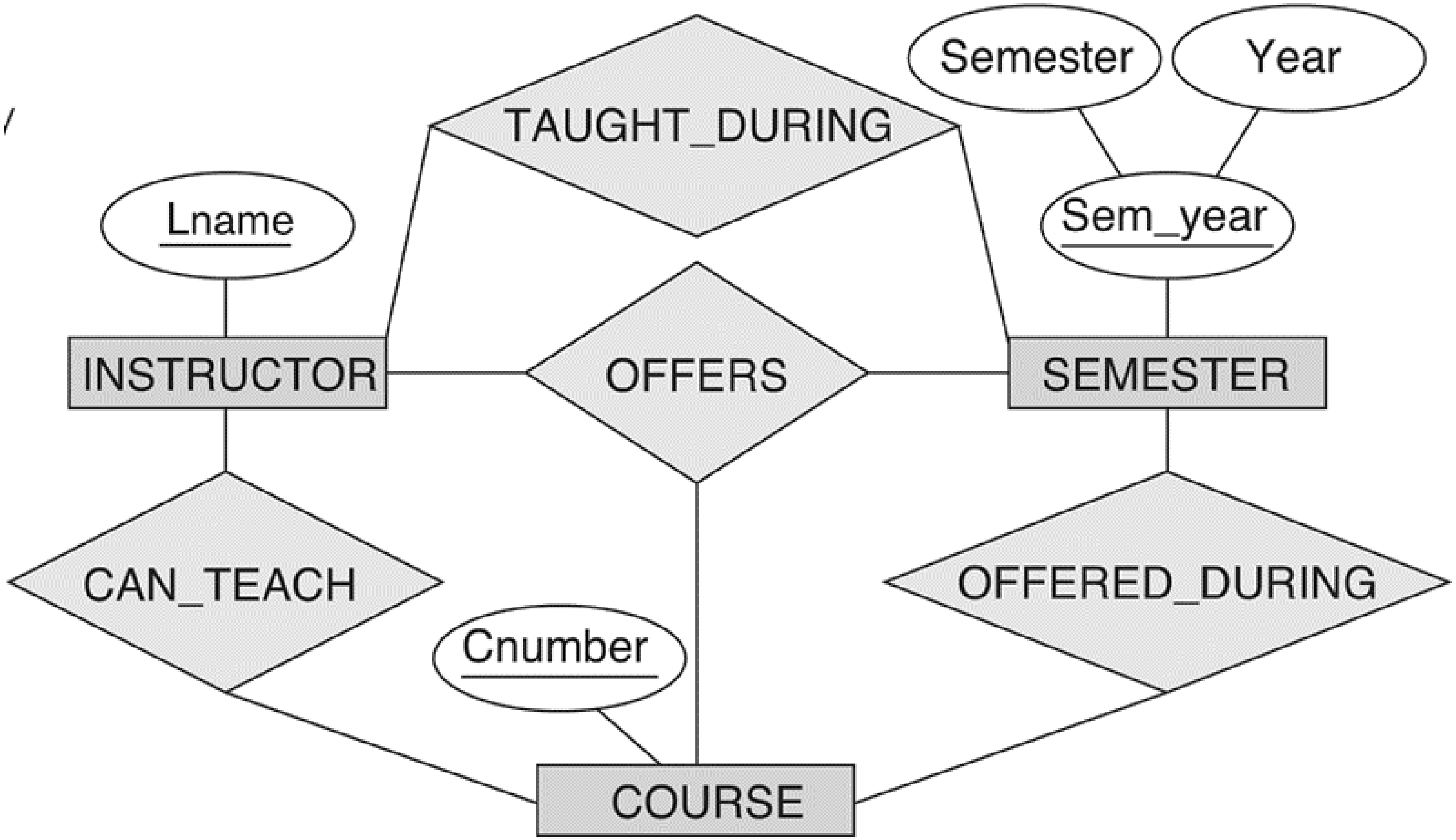
Example 1: Unary Relationship



Example 2: Binary Relationship



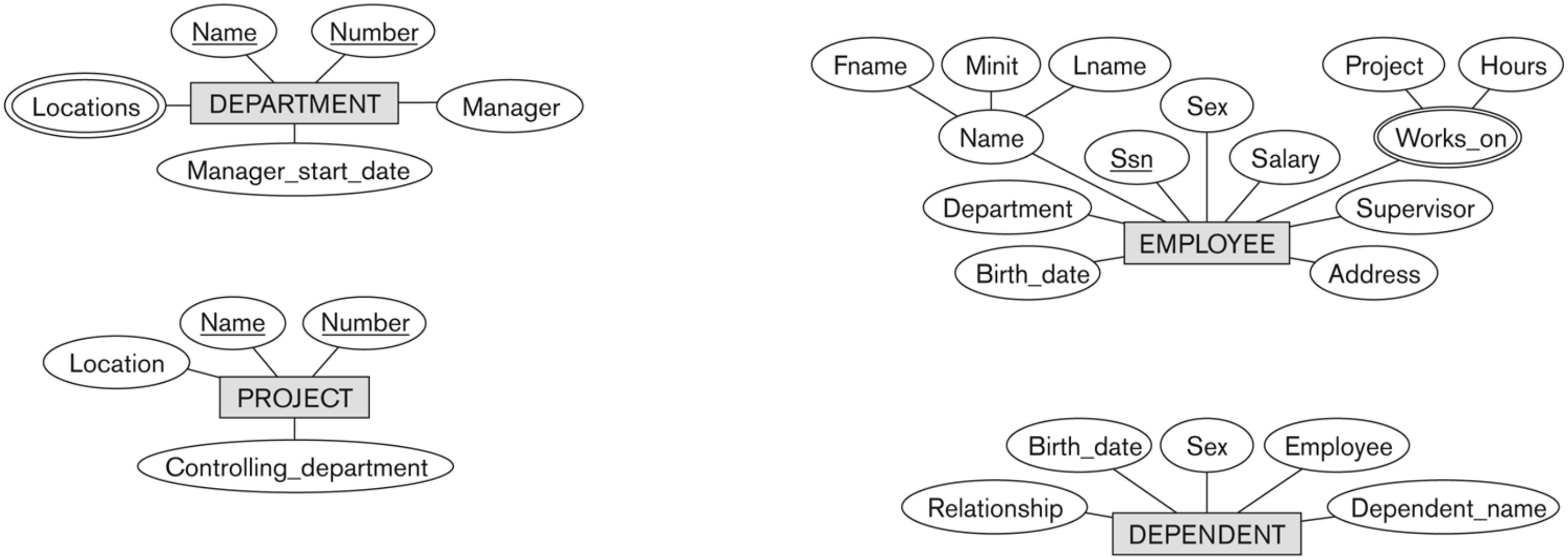
Example 3: Ternary Relationship



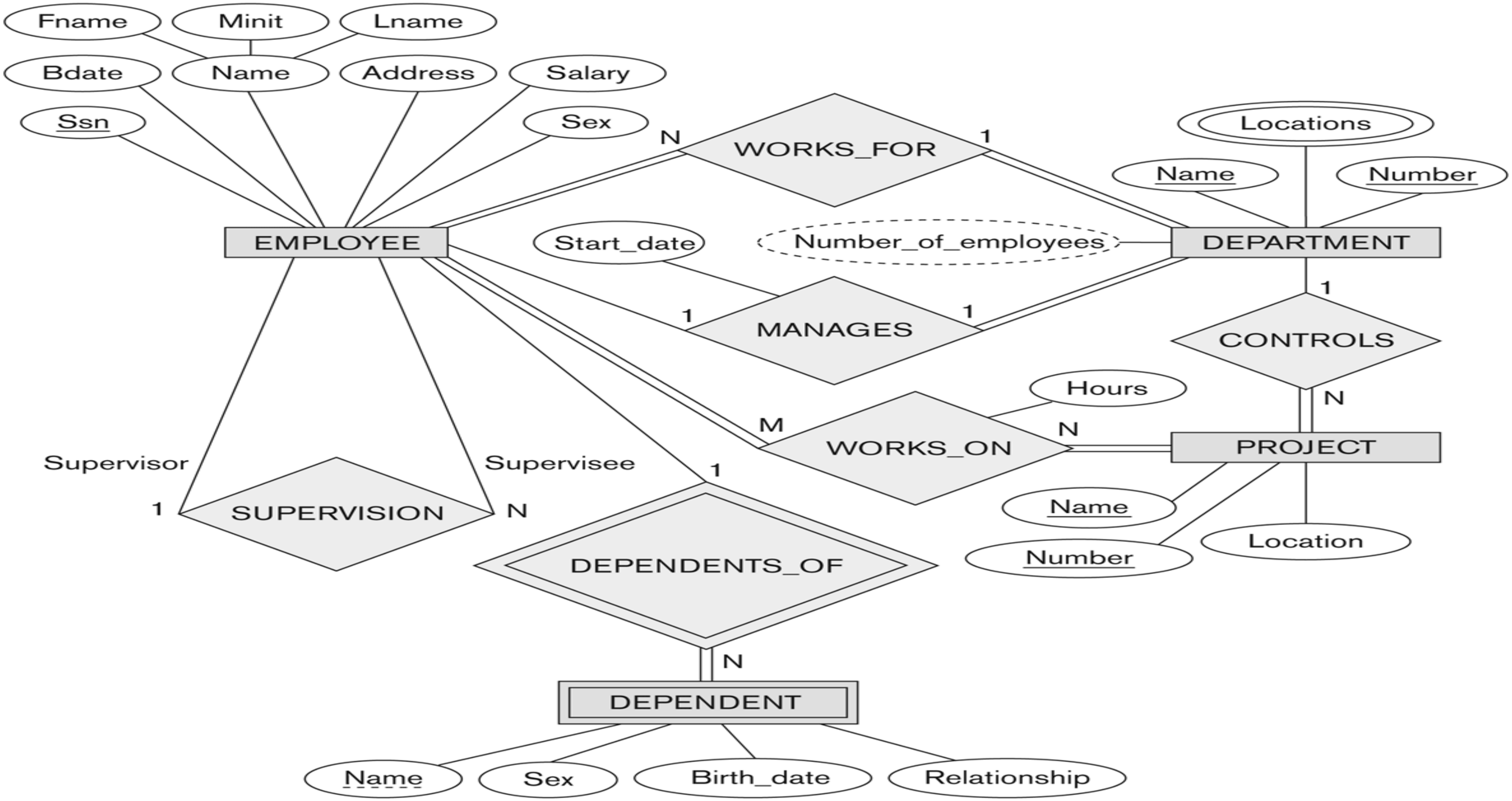
A look at the Analysis & Design Phases of Database Development

- Example of Requirements collecting of the Company:
  - The company has DEPARTMENTs.
    - Each department has a name, number & an employee who manages the department.
    - We keep the record of the starting date of manager.
    - A department may have some locations.
  - Each department controls a number of PROJECTs. Each project has a name, number and location.
  - We store each EMPLOYEE's ID, number, address, salary, gender and birthdate.
    - Each employee works for one department but may work on many projects.
    - We keep track of the number of hours per week that an employee works on each project.
    - We also keep record of the supervisor of each employee.
  - Each employee may have a number of DEPENDENTs.
    - Each dependent has name, sex, birthdate and relationship to the employee.

Conceptual Modeling



ER Design/Modeling



Home Work

- Create groups until next week (Maximum of three students).

References

- Database Systems: A practical approach to design, implementation, and management.

Questions ...?