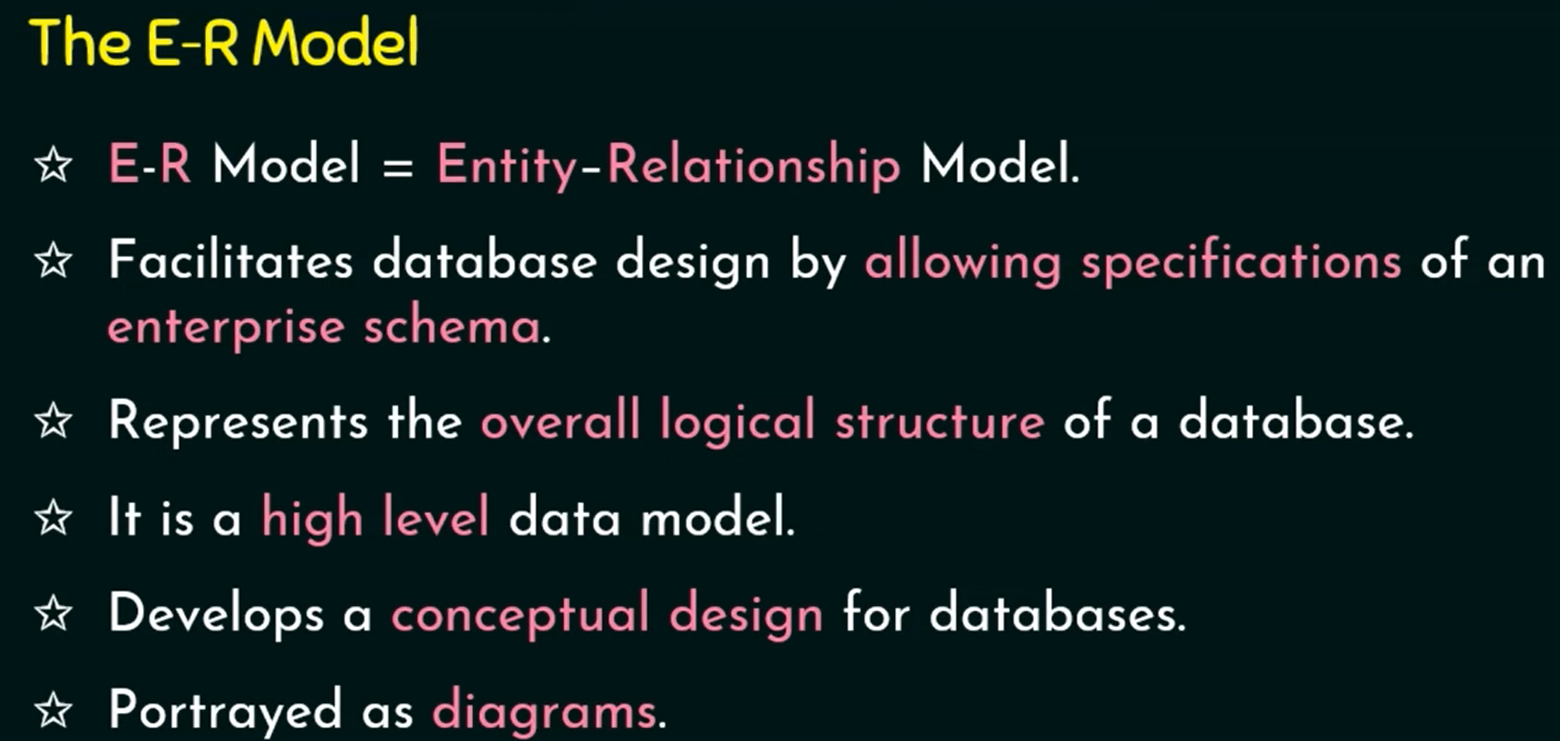
****

**1. Introduction to Entity-Relationship (ER) Model**

* **ER Model**: A high-level conceptual data model used for database design.
* **ER Diagrams**: Diagrammatic notation used to represent the ER model.
* **Unified Modeling Language (UML)**: Another modeling language related to ER concepts.

**2. Using High-Level Conceptual Data Models for Database Design**

* **Steps in Database Design**:
  1. **Requirements Collection and Analysis**:
     + Database designers interview users to understand data and functional requirements.
  2. **Conceptual Design**:
     + Create a **conceptual schema** that describes data requirements, including entity types, relationships, and constraints.
  3. **Logical Design (Data Model Mapping)**:
     + Transform the conceptual schema into a database schema using the DBMS's implementation data model.
  4. **Physical Design**:
     + Define internal storage structures, file organizations, indexes, and access paths.

**3. Sample Database Application: COMPANY Database**

* **Entities**:
* **Employee**:

Attributes: Name, Social Security Number (SSN), Address, Salary, Gender, Birth Date.

Represents individuals working in the company.

* **Department**:

Attributes: Department Name, Department Number, Manager.

Represents organizational units within the company.

* **Project**:

Attributes: Project Name, Project Number, Location.

Represents tasks or initiatives undertaken by the company.

* **Dependent**:

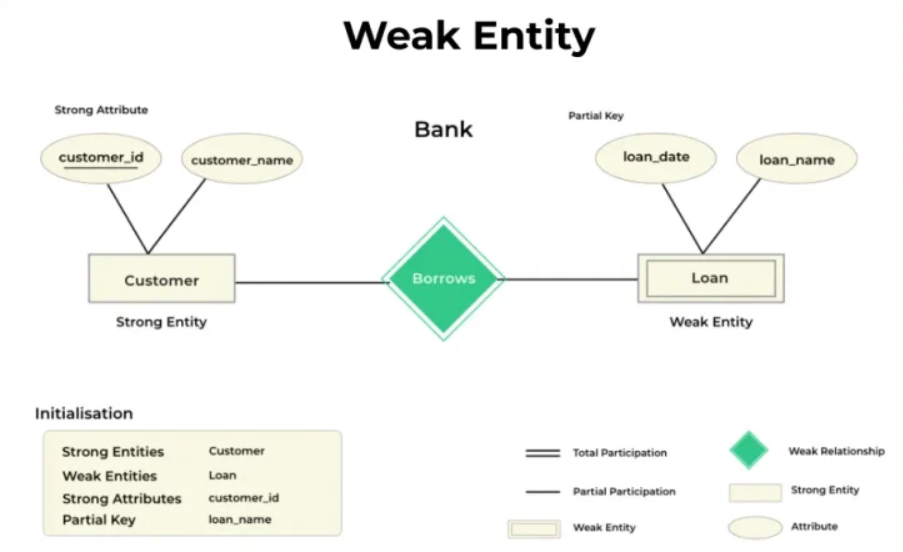
Attributes: Name, Relationship to Employee, Birth Date, Gender.

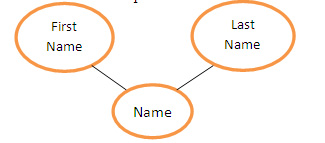
Represents individuals (e.g., family members) who depend on an employee.

*Weak Entity*: Dependent **does not have a key attribute** of its own and is identified through its relationship with an Employee (the owner entity).

* **Relationships**:
  + Departments control projects.
  + Employees work on projects and have dependents.
* **Attributes**:
  + Employee: Name, Social Security Number (SSN), Address, Salary, Gender, Birth Date.
  + Department: Name, Number, Locations
  + Projects: Name, Number, Location
  + Dependents: Name, Relationship to Employee, Gender, DOB.

**4. Entity Types, Entity Sets, Attributes, and Keys**

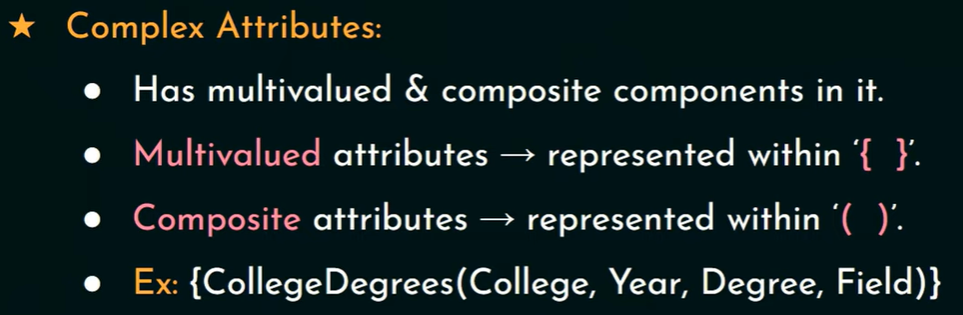
****

* **Entity type/relation:** A collection of entities (records) that share common properties or characteristics.
* **Entity instance:** A single occurrence (record/row) of an entity type (relation). We often use the single term **entity** rather than **entity instance**.
* Entity**Entity**: A real-world object with an independent existence (e.g., Employee, Department).
  + Strong Entities: Exist independently (e.g., Student, Employee).
  + Weak-EntityWeak Entities: Depend on a strong entity (e.g., Order Items depend on Order).
* **Entity Set**: A collection of entities of the same type (relation).
* **Attributes**: Properties that describe an entity.
  + **Types of Attributes**:
    1. **Simple (Atomic) Attribute**: Single component (e.g., SSN).
    2. **Composite Attribute**: Composed of multiple components (e.g., Address: Street, City, State).
    3. **Single-Valued Attribute**: Can have only one, holds a

single value (e.g., Age).

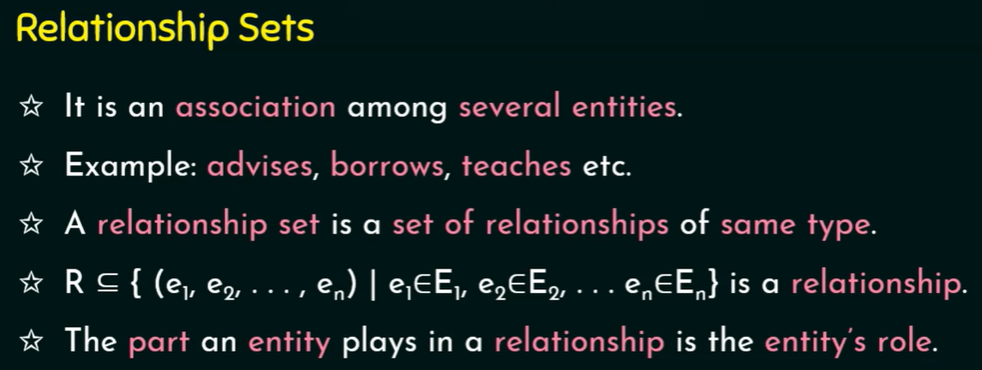
* + 1. A black and orange circle

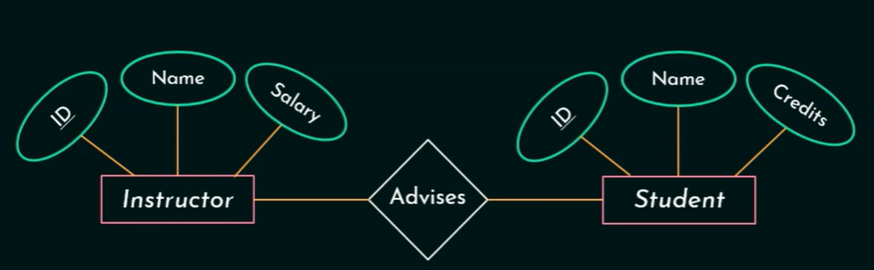
       Description automatically generated**Multi-Valued Attribute**: Can have multiple, holds multiple values (e.g., multiple phone numbers and college degrees).
    2. **Stored Attribute**: Persistently stored (e.g., Birth Date).
    3. **Derived Attribute**: Computed from other attributes (e.g., Age derived from Birth Date, GPA derived from Grades).
    4. **NULL Values**: Represents missing or unknown values.
    5. **Complex Attributes**: Combination of composite and multi-valued attributes.



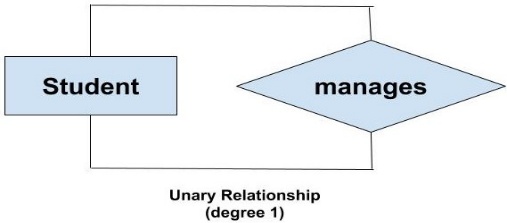
**5. Relationship Types, Relationship Sets, Roles, and Structural Constraints**

* **Relationship**: An association between entities (e.g., Employee works in Department).
* **Relationship Set**: A collection of relationships of the same type.

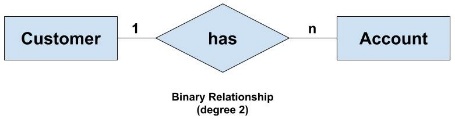




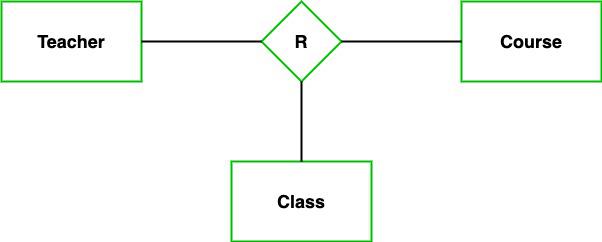
* **Degree of Relationship**:
  + **Unary**: Recursive Relationship



* + **Binary**: Between two entity types (e.g., Employee-Department).



* + **Ternary**: Between three entity types (e.g., Employee-Project-Department).



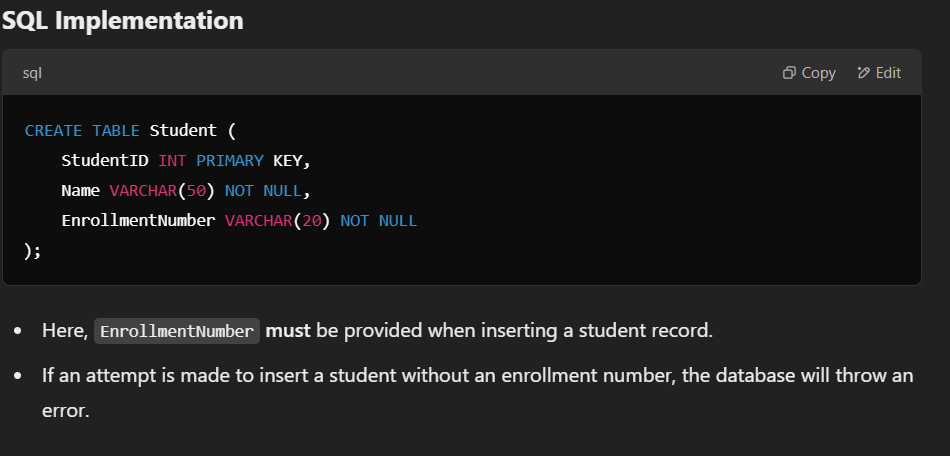
* **Roles**: The function an entity plays in a relationship (e.g., Manager and Employee in a Manages relationship).
* **Recursive Relationship**: An entity participates in a relationship with itself (e.g., Employee supervises another Employee).

**6. Optional vs. Mandatory Attributes (0:1 or 1:1 Cardinality Constraints)**

In database design, attributes can either be mandatory (must have a value) or optional (may or may not have a value). This cardinality constraint defines whether an attribute must be filled (1:1) or can be left empty (NULL) (0:1).

**🔹 Mandatory Attribute (1:1 Cardinality)**

* A **mandatory attribute** is an attribute that **must have a value** for every entity in the database.
* It **cannot** be NULL.
* Enforced using the **NOT NULL constraint** in SQL.

**Example Scenario**

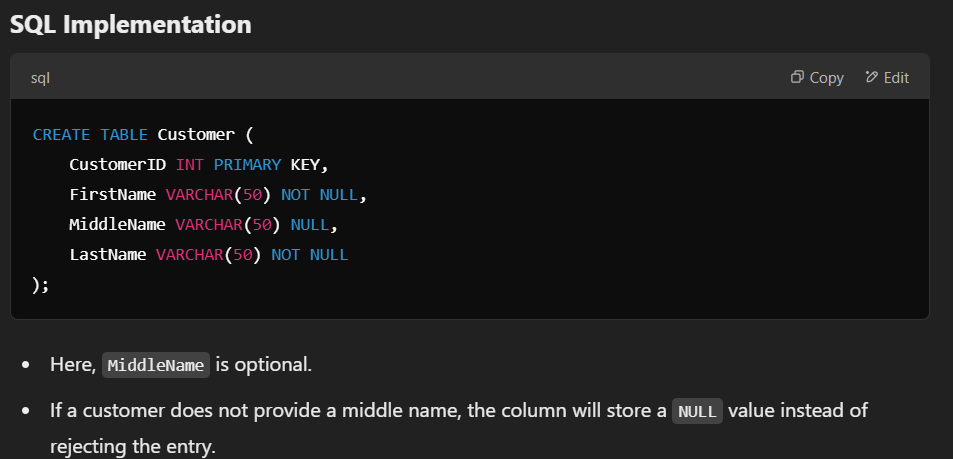
* A university database requires that every **student must have an enrollment number**.
* The **EnrollmentNumber** attribute is mandatory.

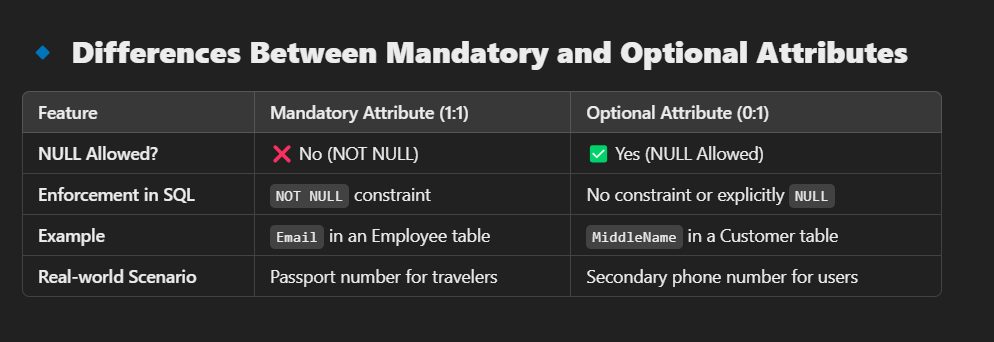
**🔹 Optional Attribute (0:1 Cardinality)**

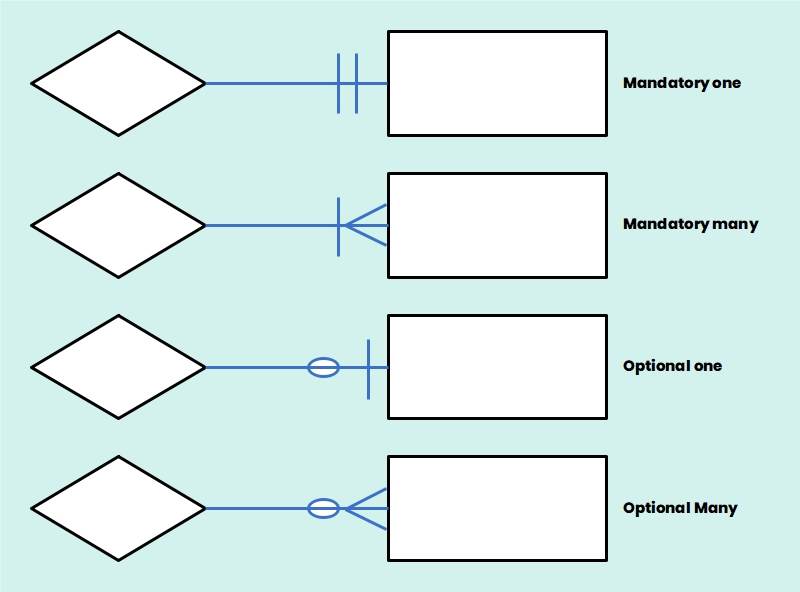
* An **optional attribute** is an attribute that **may or may not** have a value for an entity.
* It **can** be NULL.
* No constraints are applied, or the **NULL** constraint is allowed in SQL.

**Example Scenario**

* In a **customer database**, the **Middle Name** is optional.
* Some customers may have a middle name, while others may not.

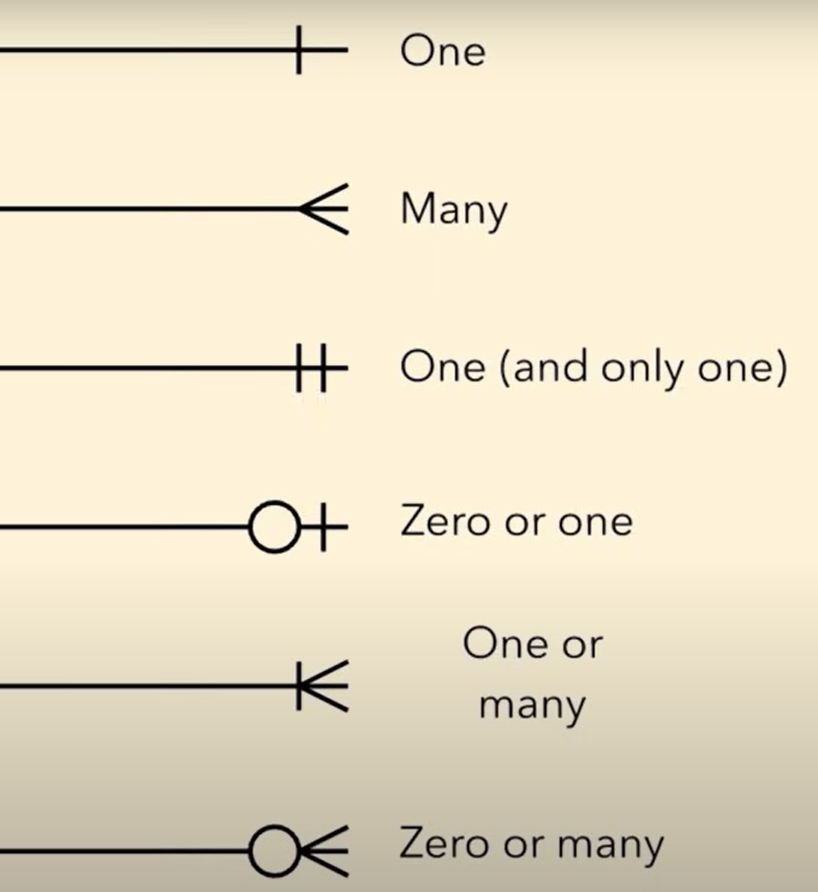
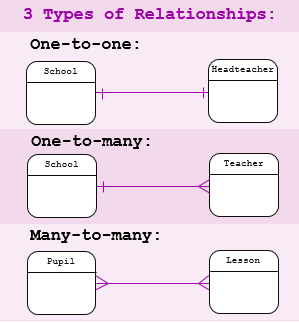






**7. Attributes of Relationship Types**

* **1:1 or 1:N Relationships**:
  + Attributes can be migrated to one entity type.
  + Example: For a 1:N relationship, the attribute is migrated to the N-side.
* **M:N Relationships**:
  + Attributes must be specified as relationship attributes.
  + Example: Hours worked by an Employee on a Project.

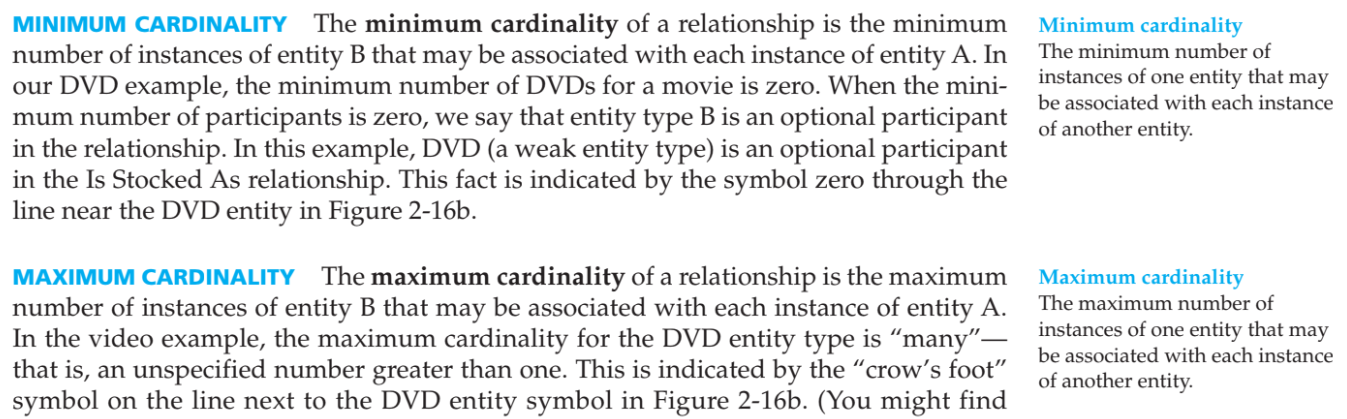


**(a) Basic relationship**



**(b) Relationship with cardinality constraints**





A close-up of a blue box

AI-generated content may be incorrect.

A black text on a white background

AI-generated content may be incorrect.

A diagram of a company

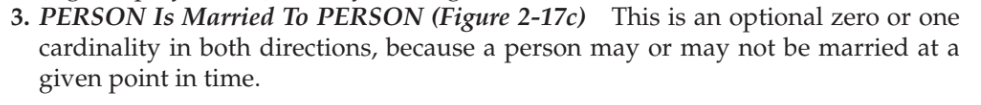
AI-generated content may be incorrect.

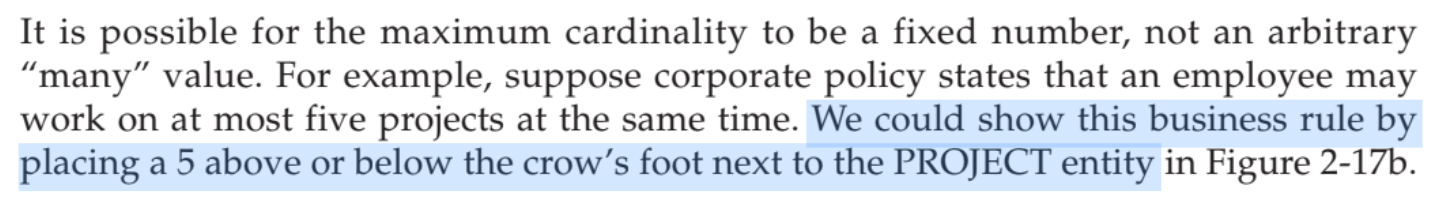
A close up of a text

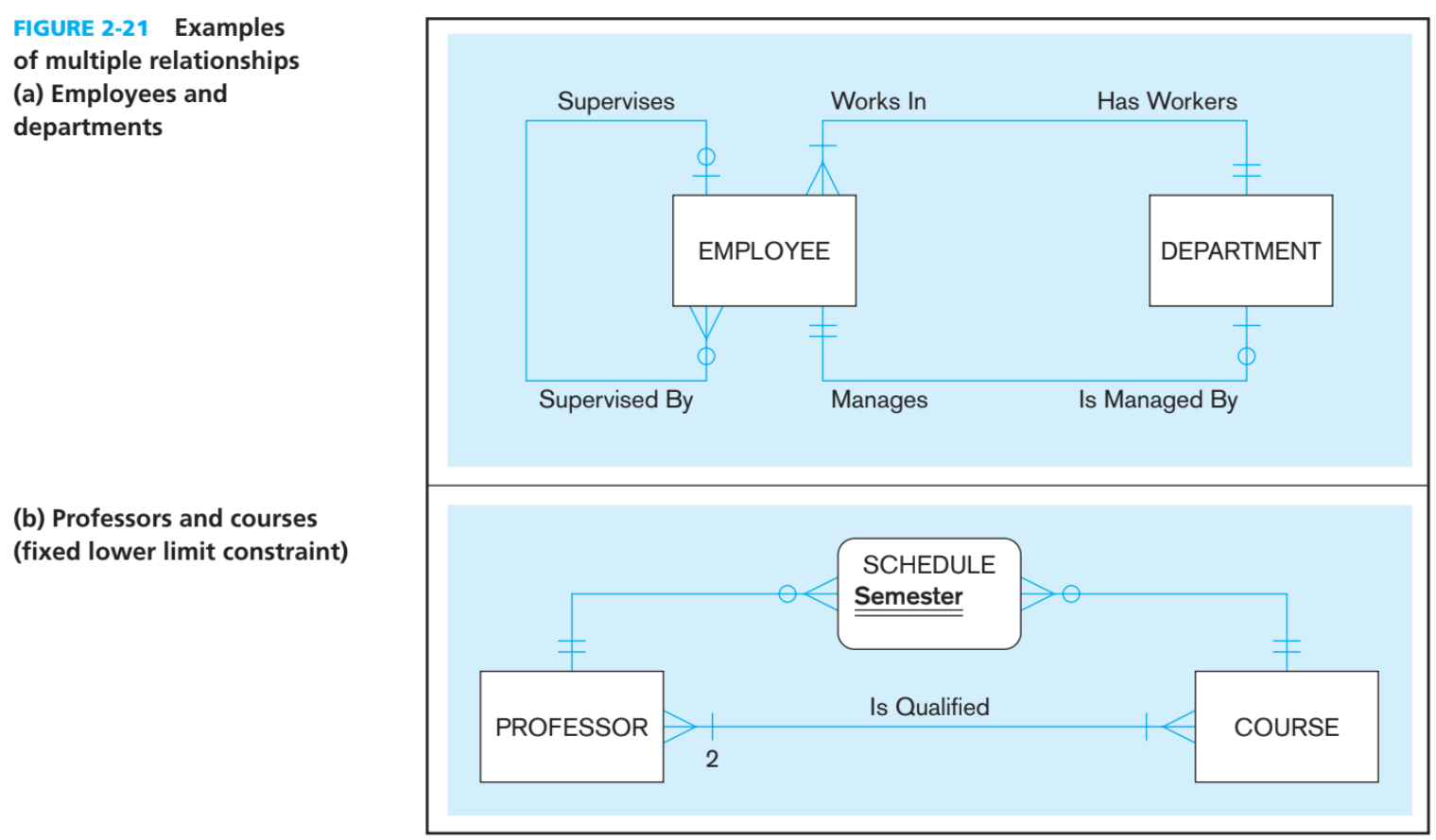
AI-generated content may be incorrect.

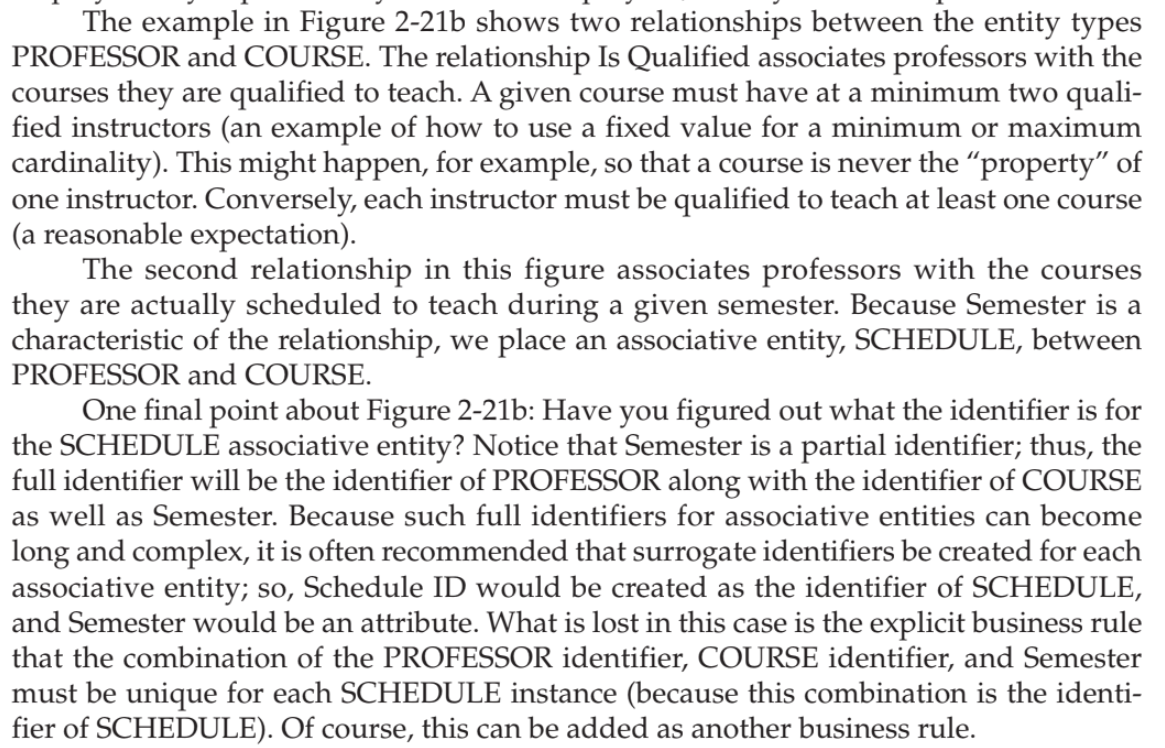
A diagram of a person and an option

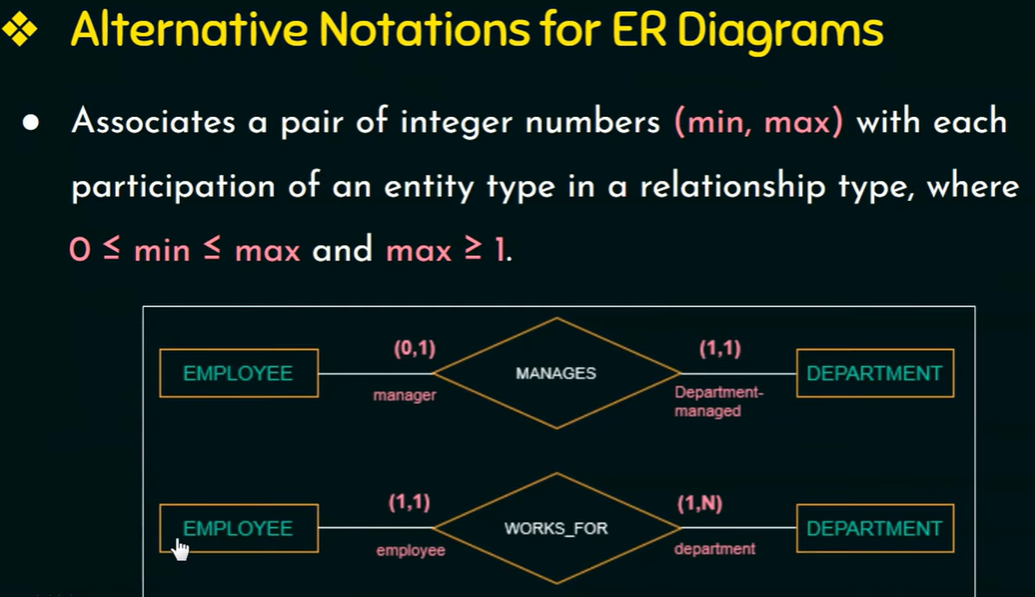
AI-generated content may be incorrect.











0 = Partial Participation

Any value greater than 0 = Total Participation