ER (Entity Relationship) Diagram

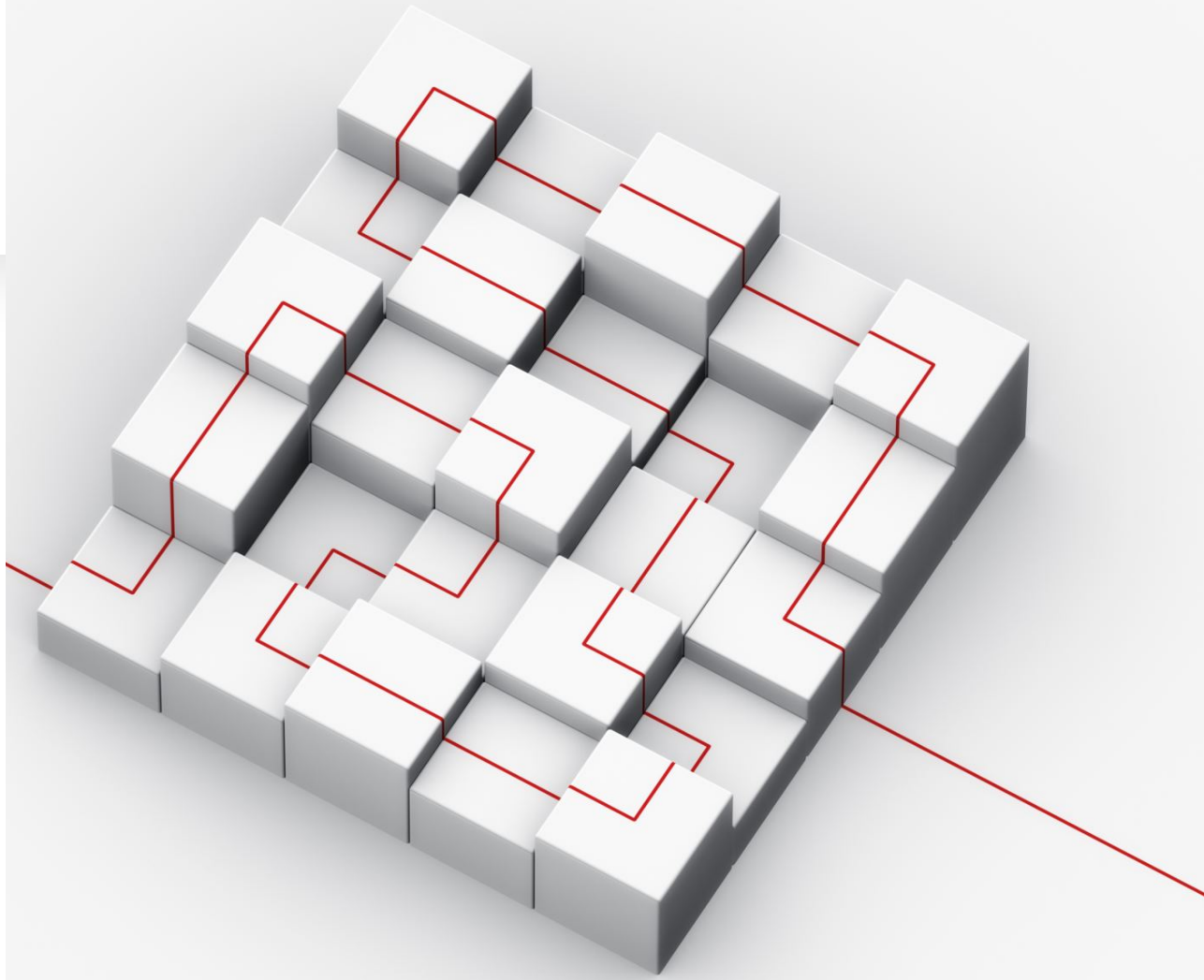
ER model stands for an Entity-Relationship model. It is a high-level data model. This model is used to define the data elements and relationship for a specified system.

It develops a conceptual design for the database. It also develops a very simple and easy to design view of data.

In ER modeling, the database structure is portrayed as a diagram called an entity-relationship diagram.

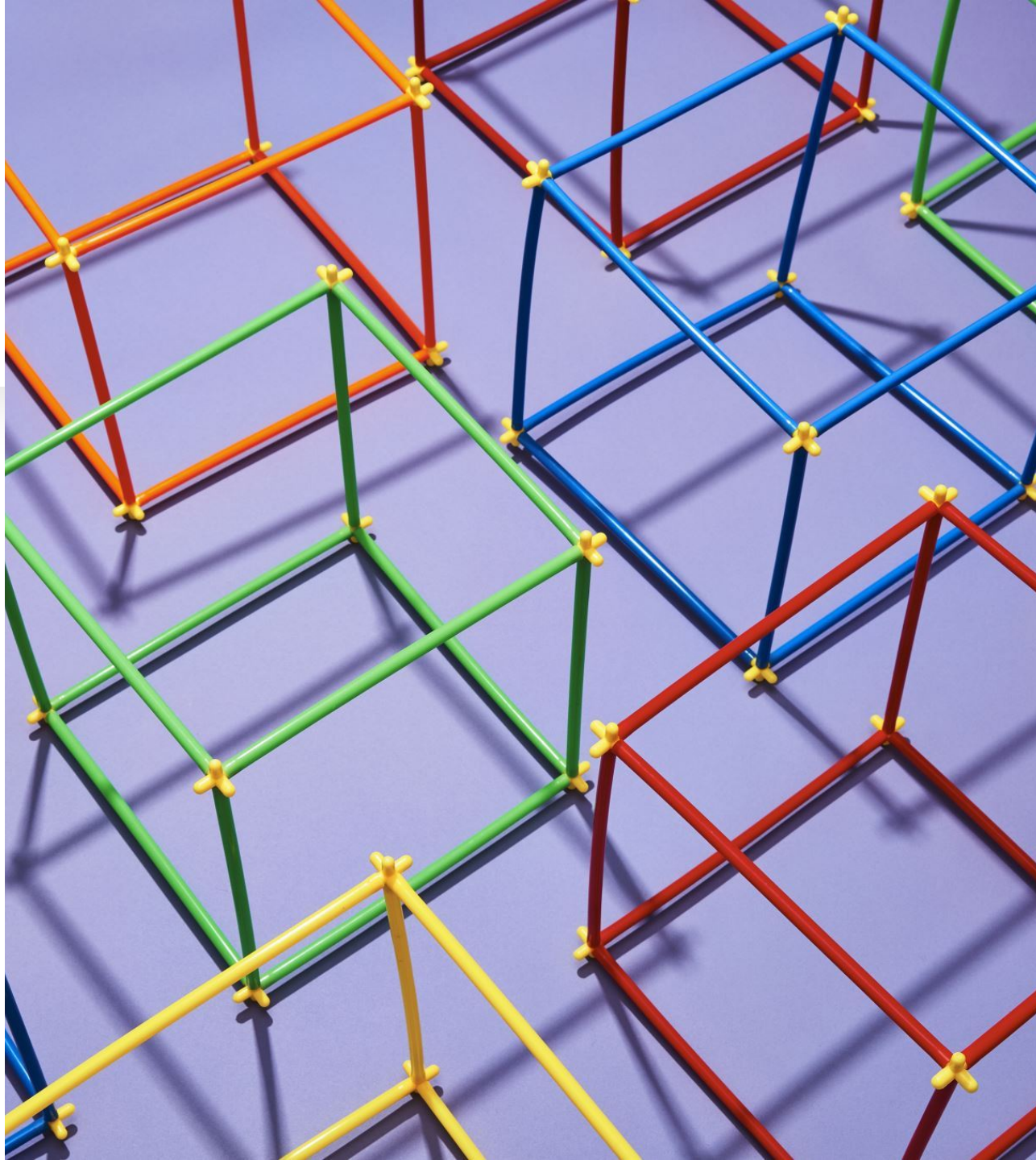
Why Use ER Diagrams In DBMS?

- ER diagrams are used to represent the E-R model in a database, which makes them easy to convert 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.



Importance

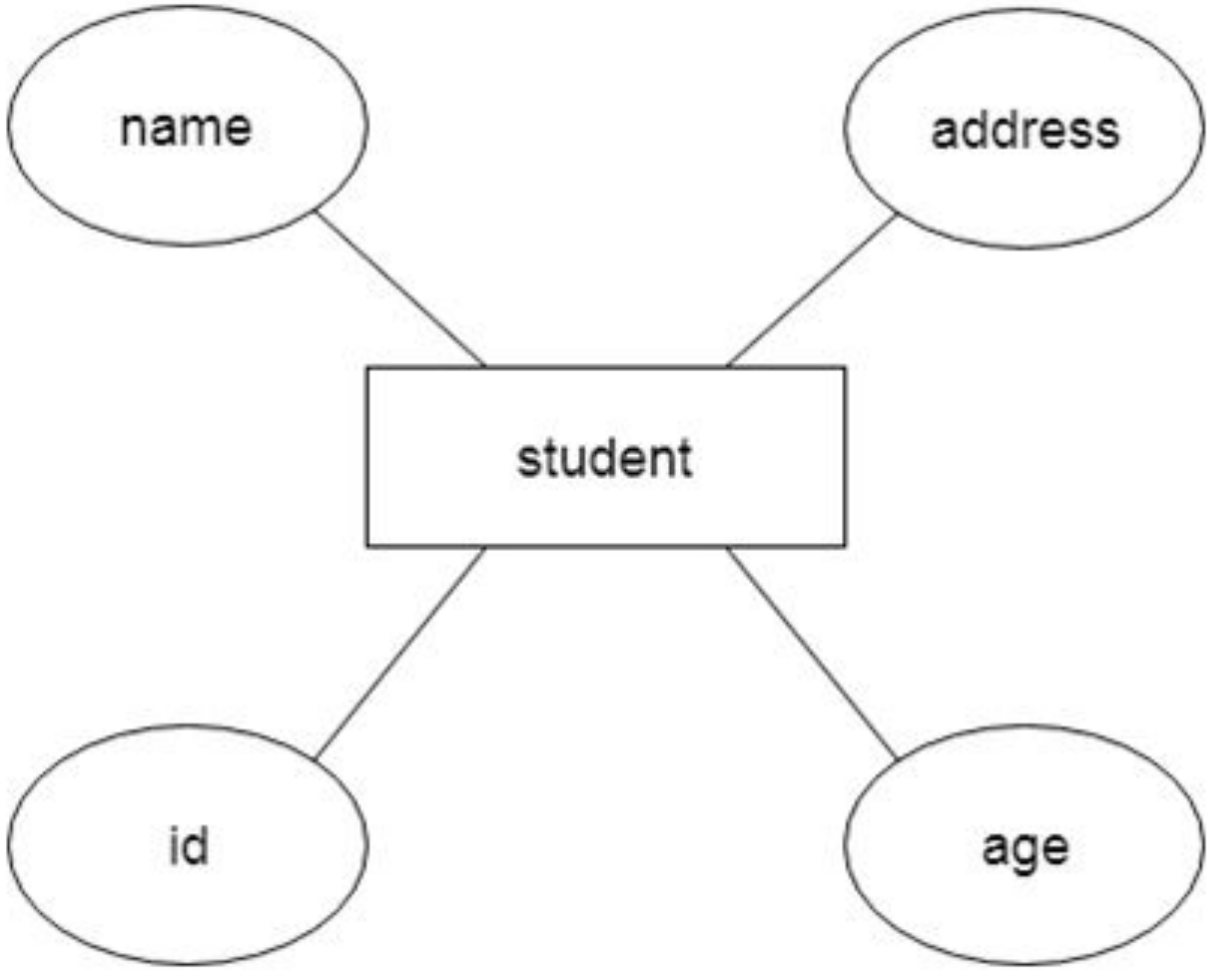
They help in planning and organizing database structure, ensuring data consistency and integrity.



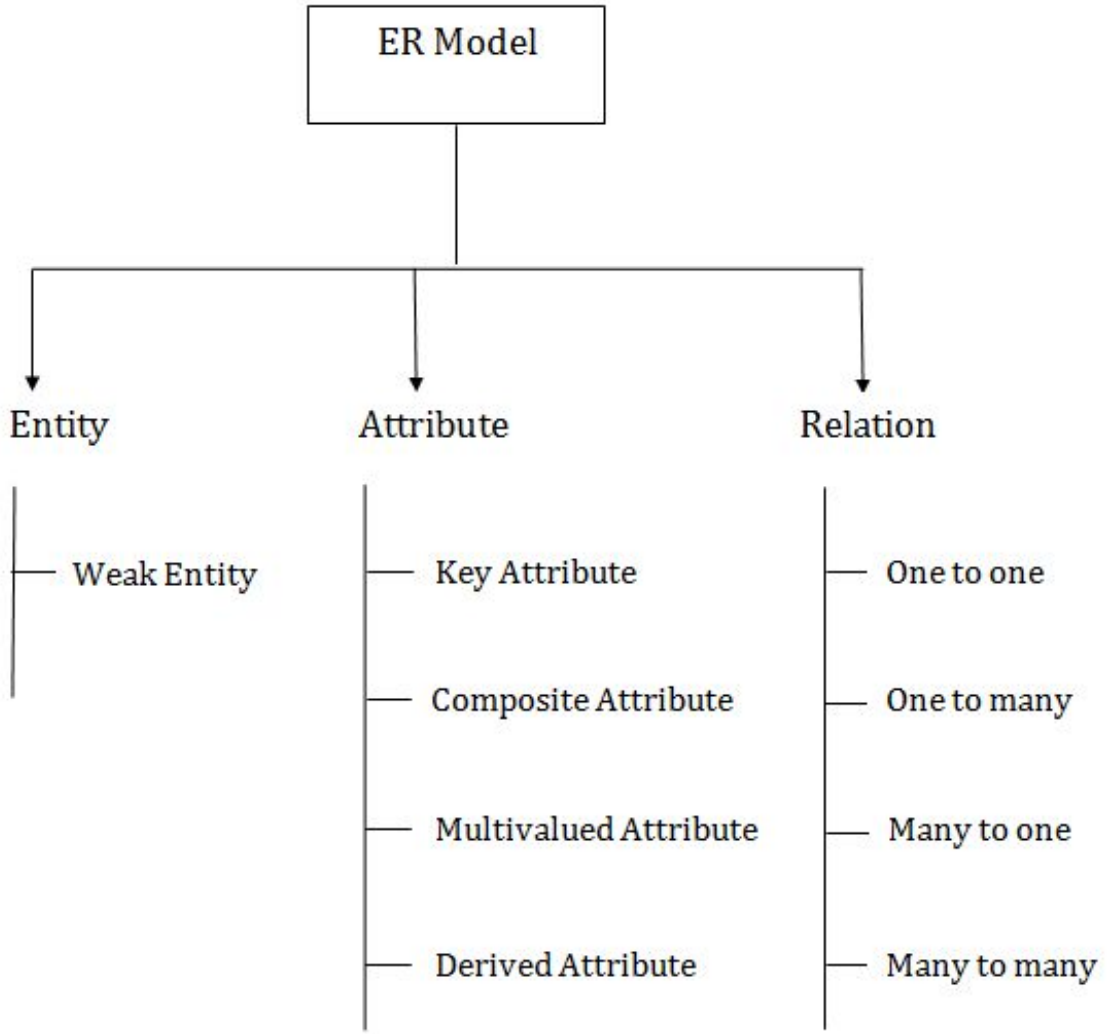
Symbols Used in ER Model

- **Rectangles:** Rectangles represent Entities in the ER Model.
- **Ellipses:** Ellipses represent Attributes in the ER Model.
- **Diamond:** Diamonds represent Relationships among Entities.
- **Lines:** Lines represent attributes to entities and entity sets with other relationship types.
- **Double Ellipse:** Double Ellipses represent Multi-Valued Attributes.
- **Double Rectangle:** Double Rectangle represents a Weak Entity.

Example Suppose we design a school database. In this database, the student will be an entity with attributes like address, name, id, age, etc. The address can be another entity with attributes like city, street name, pin code, etc and there will be a relationship between them.



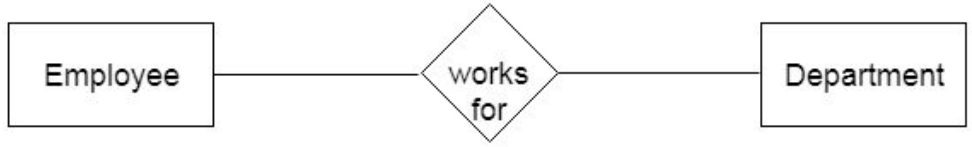
Component of ER Diagram



Entity

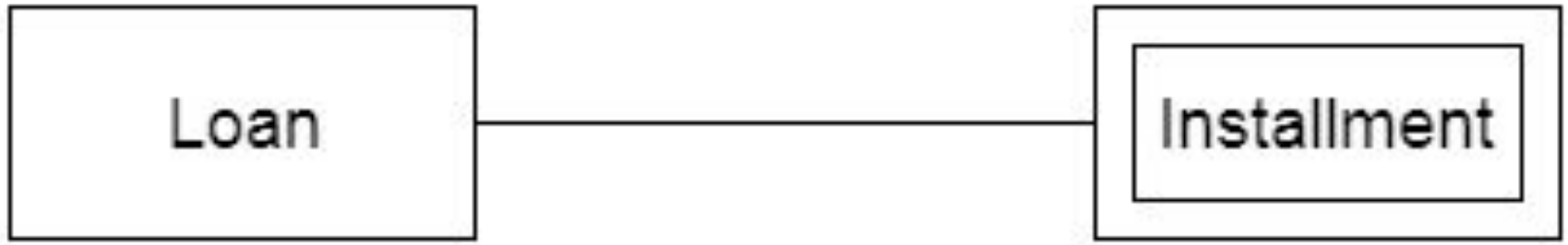
An entity may be any object, class, person or place. In the ER diagram, an entity can be represented as rectangles.

Consider an organization as an example- manager, product, employee, department etc. can be taken as an entity.



Weak Entity

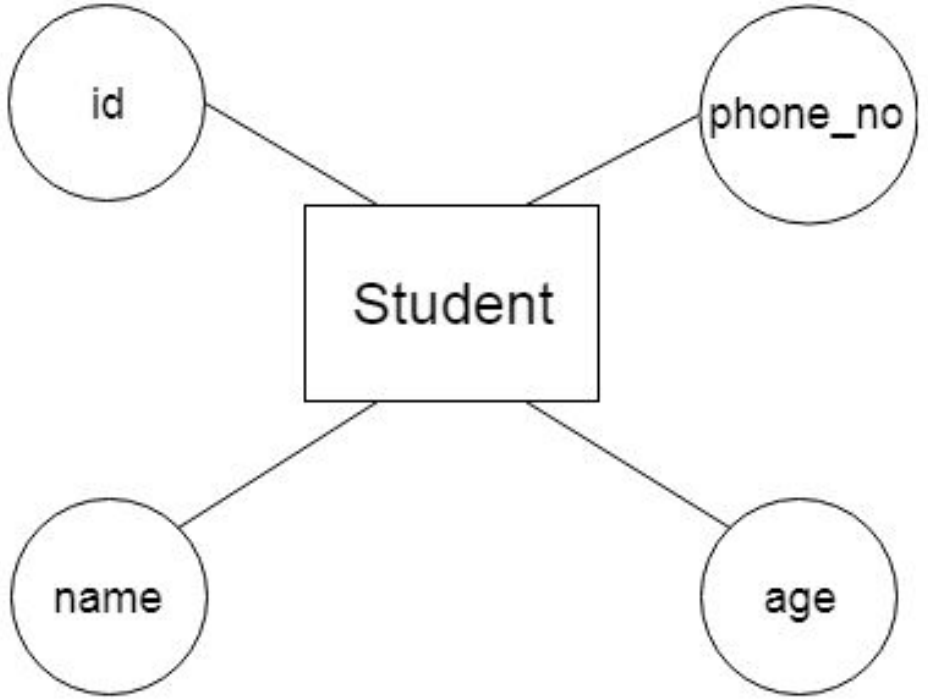
An entity that depends on another entity called a weak entity. The weak entity doesn't contain any key attribute of its own. The weak entity is represented by a double rectangle.



Attribute

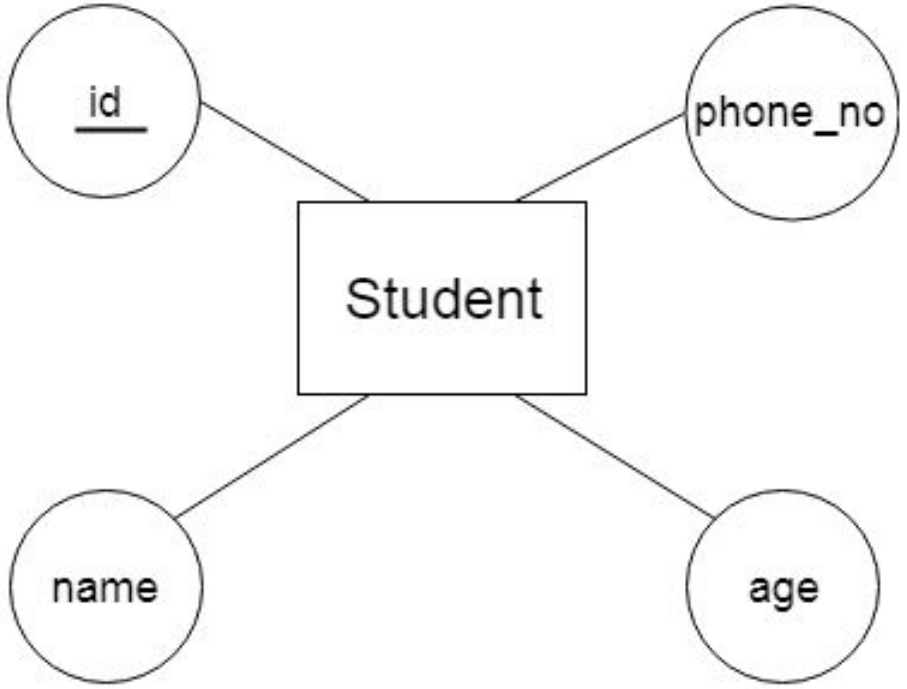
The attribute is used to describe the property of an entity. Eclipse is used to represent an attribute.

For example, id, age, contact number, name, etc. can be attributes of a student.



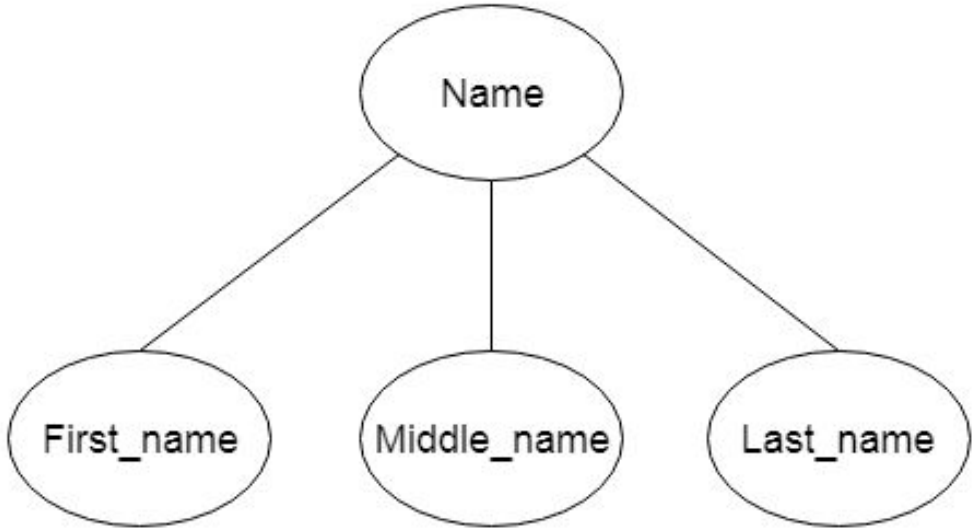
Key Attribute

The key attribute is used to represent the main characteristics of an entity. It represents a primary key. The key attribute is represented by an ellipse with the text underlined.



Composite Attribute

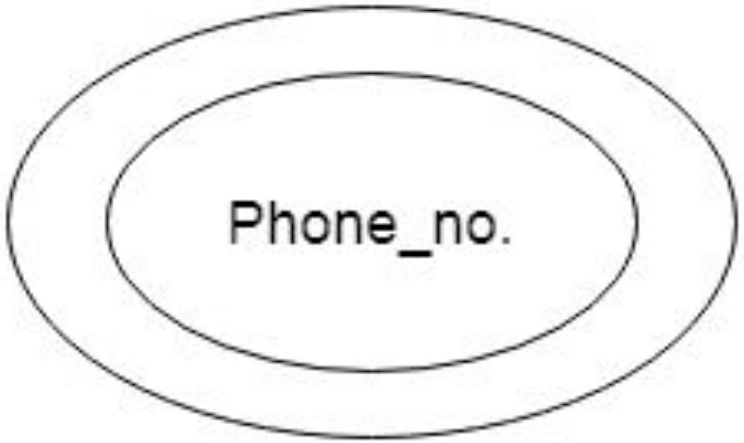
An attribute that composed of many other attributes is known as a composite attribute. The composite attribute is represented by an ellipse, and those ellipses are connected with an ellipse.



Multivalued Attribute

An attribute can have more than one value. These attributes are known as a multivalued attribute. The double oval is used to represent multivalued attribute.

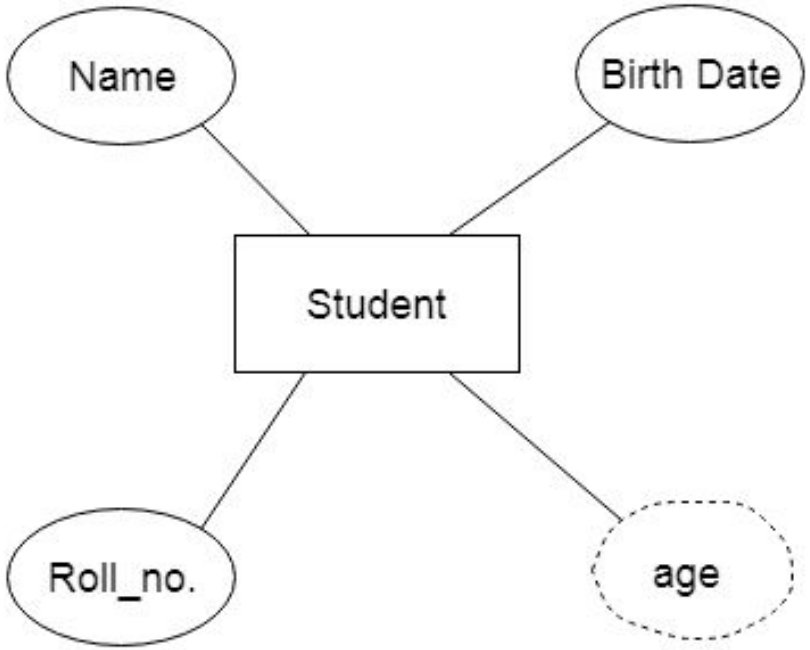
For example, a student can have more than one phone number.



Derived Attribute

An attribute that can be derived from other attribute is known as a derived attribute. It can be represented by a dashed ellipse.

For example, A person's age changes over time and can be derived from another attribute like Date of birth.



Relationship

A relationship is used to describe the relation between entities. Diamond or rhombus is used to represent the relationship.

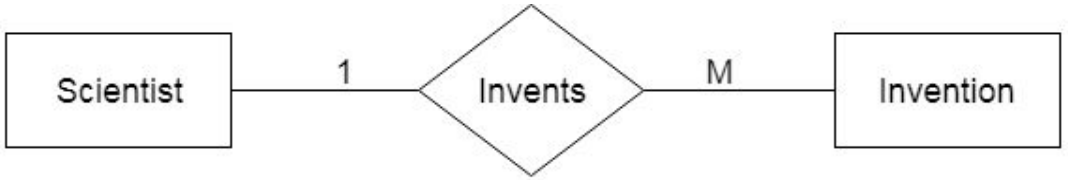
Types of relationship are as follows:

a. One-to-One Relationship

When only one instance of an entity is associated with the relationship, then it is known as one to one relationship.

For example, A female can marry to one male, and a male can marry to one female.

Types of relationship are as follows:



b. One-to-many relationship

When only one instance of the entity on the left, and more than one instance of an entity on the right associates with the relationship then this is known as a one-to-many relationship.

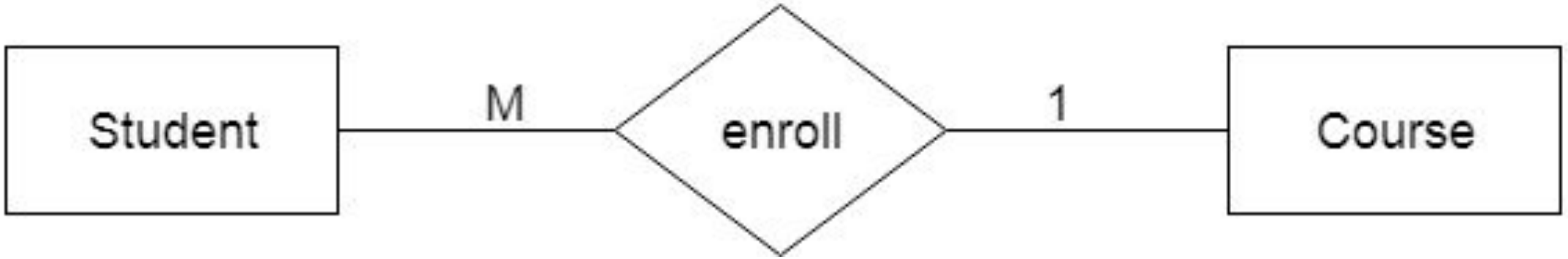
For example, Scientist can invent many inventions, but the invention is done by the only specific scientist.

Types of relationship are as follows:

c. Many-to-one relationship

When more than one instance of the entity on the left, and only one instance of an entity on the right associates with the relationship then it is known as a many-to-one relationship.

For example, Student enrolls for only one course, but a course can have many students.



Types of relationship are as follows:

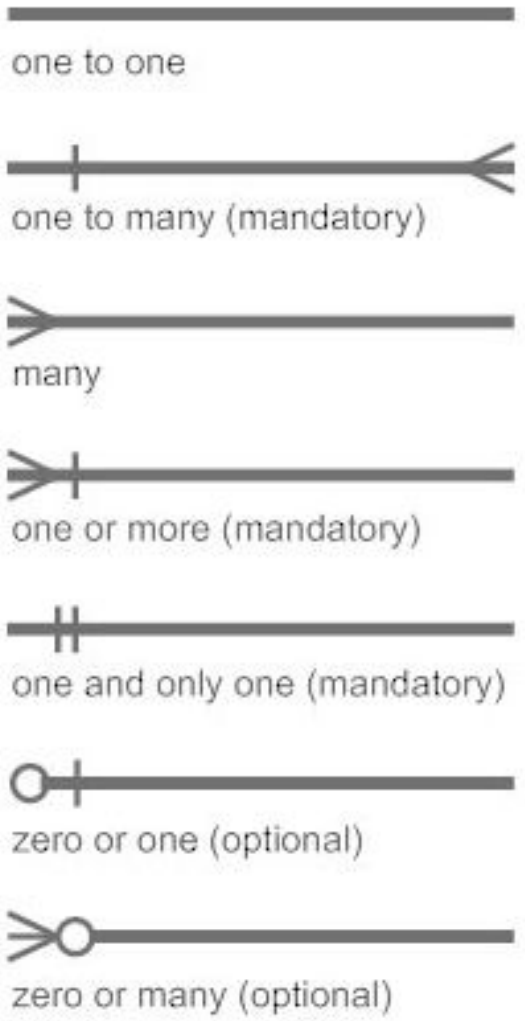
d. Many-to-many relationship

When more than one instance of the entity on the left, and more than one instance of an entity on the right associates with the relationship then it is known as a many-to-many relationship.

For example, Employee can assign by many projects and project can have many employees.

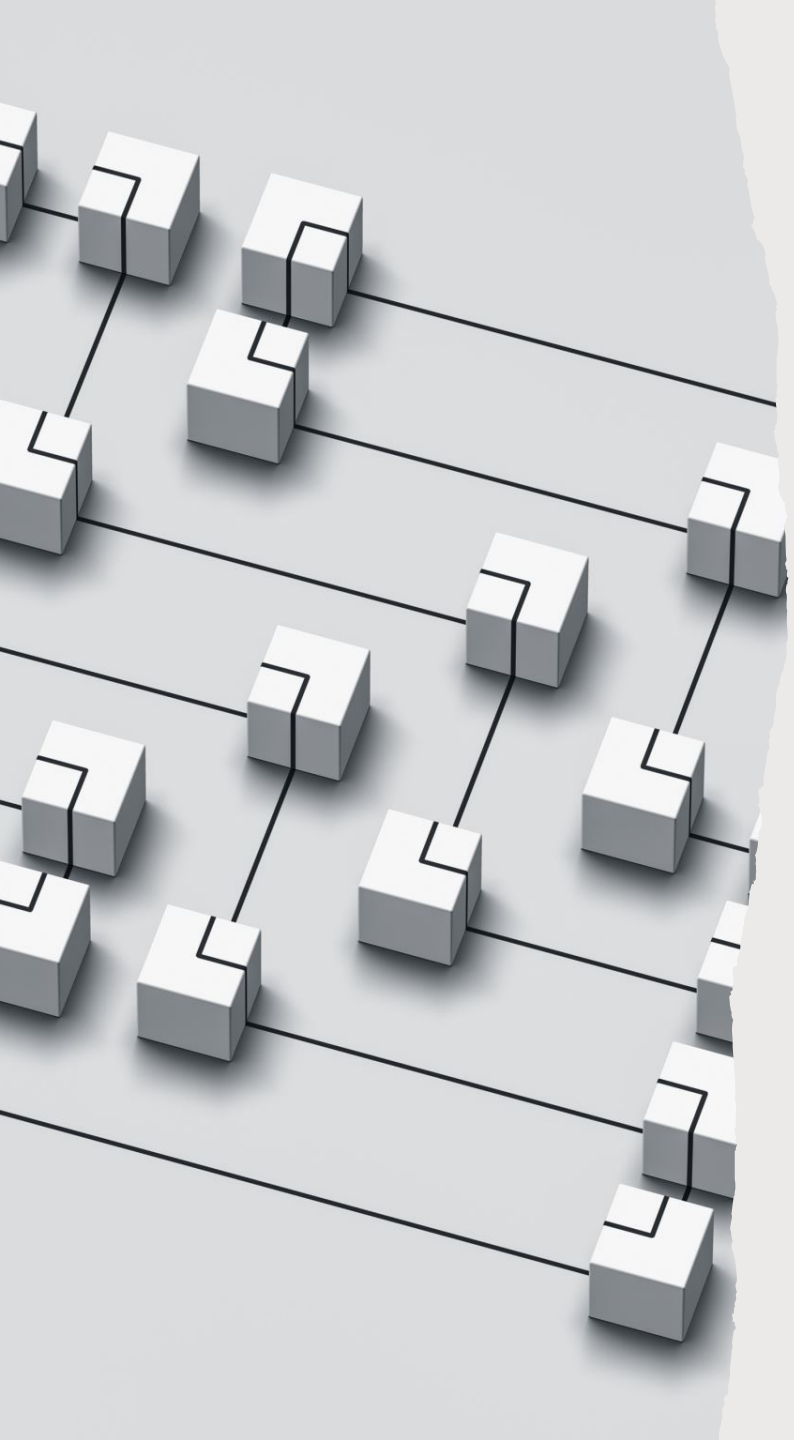
Notation of ER diagram

Database can be represented using the notations. In ER diagram, many notations are used to express the cardinality. These notations are as follows:



ER Diagram Notations

- **Standard Symbols**: Rectangles for entities, ovals for attributes, diamonds for relationships, lines for linking them.
- **Crow’s Foot Notation**: Uses symbols like crow's feet to represent cardinality and relationships.
- **Chen’s Notation**: Uses rectangles, ovals, and diamonds with labels to depict entities, attributes, and relationships.



Designing an ER Diagram

Steps in ER Diagram Design:

- Identify Entities: Determine major entities in the system.
- Identify Relationships: Define how entities interact.
- Define Attributes: Specify attributes for each entity and relationship.
- Determine Keys: Identify primary and foreign keys.

Example Process: A simple scenario (e.g., designing an ER diagram for a university database).

Primary Keys and Foreign Keys

Definition of Primary Key: A unique identifier for each entity instance (e.g., Student ID).

Definition of Foreign Key: An attribute that creates a link between two tables (e.g., Course ID in Enrollments table).

Role in ER Diagrams: Ensure data integrity and establish relationships between tables.

Examples of ER Diagrams

- **Simple ER Diagram Example**: Student, Course, Enrollment.
- **Medium Complexity Example**: Library System with Books, Members, rent.
- **Complex Example**: E-commerce System with Customers, Orders, Products, Payments.

Tools for Creating ER Diagrams

Software Tools:

- MySQL Workbench
- Lucidchart
- ER/Studio

Online Tools:

- draw.io
- Creately

ER Diagrams in Real-World Applications

Usage in Various Industries: IT, healthcare, finance, education.

Benefits of Using ER Diagrams: Improved database design, data integrity, and communication between stakeholders.

Examples from Real Projects: Highlight a few real-world examples where ER diagrams were critical.

Common Mistakes in ER Diagram Design

Overlooking	Overlooking Relationships: Failing to identify all necessary relationships.
Misidentifying	Misidentifying Primary Keys: Incorrectly choosing attributes as primary keys.
Ignoring	Ignoring Attributes: Not including all relevant attributes for entities and relationships.

Advanced ER Diagram Concepts

- **Generalization and Specialization**:
- Generalization: Combining similar entities into a single general entity (e.g., Employee and Customer into Person).
- Specialization: Dividing a general entity into more specific entities (e.g., Vehicle into Car and Truck).
- **Aggregation**: Representing a relationship as an entity to simplify complex relationships.
- **Composition**: Stronger form of aggregation indicating ownership (e.g., House and Room).

Generalization and Specialization

- **Definitions**: Generalization is the process of abstracting common features; specialization is the process of defining sub-entities.
- **Examples**: Generalization (Employee and Customer -> Person), Specialization (Vehicle -> Car, Truck).
- **Use Cases**: When to use these concepts in database design.

Aggregation

Definition: A higher-level abstraction that encapsulates a relationship set.

Example: Projects composed of tasks, employees, and deadlines.

When to Use Aggregation: To simplify ER diagrams when relationships involve multiple entities.

Composition

- **Definition**: A form of aggregation with a strong ownership between entities.
- **Example**: House (whole) and Room (part).
- **Differentiating from Aggregation**: Composition implies a lifecycle dependency between the part and the whole.