**2. Hibernate Relationships (One-to-One, One-to-Many, Many-to-One, Many-to-Many)**

**Object Relationships in Hibernate**

**• Theory:**

**1.How Hibernate manages relationships between Java objects and database tables:-**

Hibernate manages relationships between Java objects and database tables using **ORM (Object-Relational Mapping)**. It uses **annotations** or **XML configuration** to define the relationships.

**Types of Relationships in Hibernate**

There are four main types of relationships in Hibernate:

1. **One-to-One** (1:1)
2. **One-to-Many / Many-to-One** (1:M / M:1)
3. **Many-to-Many** (M:M)
4. **Bidirectional Relationships**

**2.** **Overview of the different types of relationships: ♣ One-to-One Relationship: ♣ A single instance of an entity is related to a single instance of another entity. ♣ One-to-Many Relationship: ♣ One entity can have multiple related entities. ♣ Many-to-One Relationship: ♣ Many entities are associated with a single entity. ♣ Many-to-Many Relationship:-**

**1. One-to-One Relationship (1:1)**

A single instance of an entity is related to a single instance of another entity.

**Example: A User has only one Address.**

**2. One-to-Many Relationship (1:M)**

One entity can have multiple related entities.

**Example: A Department has multiple Employees.**

**3. Many-to-One Relationship (M:1)**

Many entities are associated with a single entity.

* This is just the **reverse** of the One-to-Many relationship.

**Example: Multiple Employees belong to one Department.**

* **Same implementation as One-to-Many**, but the focus is on @ManyToOne in the Employee entity.

**4. Many-to-Many Relationship (M:M)**

Multiple entities are related to multiple entities.

**Example**: A Student can enroll in multiple Courses, and a Course can have multiple Students.

**3.Multiple instances of an entity are associated with multiple instances of another entity:-** This describes a Many-to-Many relationship in Hibernate. In such a relationship, multiple instances of one entity are associated with multiple instances of another entity.

**1. Many-to-Many Relationship in Hibernate**

A **Many-to-Many** relationship requires a **junction table** (also called a **join table**) to store the relationships between entities.

**Example:**

* A Student can enroll in **multiple** Courses.
* A Course can have **multiple** Students.