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

**Mapping Relationships in Hibernate:**

**• Theory:**

1. **How to map relationships in Hibernate using annotations like @OneToOne, @OneToMany, @ManyToOne, and @ManyToMany:-**
2. **The concept of owning and inverse sides in relationships:-**

hen defining relationships in Hibernate, there are two sides:

1. Owning Side (the side responsible for the relationship and foreign key mapping).
2. Inverse Side (the side that is mapped using mappedBy and does not own the foreign key).

Hibernate requires that one side of a bidirectional relationship be the owning side, and the other be the inverse side, to avoid redundant foreign key columns.

1. **Cascade types and how they affect related entities:-**

**Cascade Type:-** CascadeType.PERSIST , CascadeType.MERGE , CascadeType.REMOVE , CascadeType.REFRESH , CascadeType.DETACH , CascadeType.ALL

* + - Cascade types in Hibernate define how operations (like persist, merge, delete, etc.) are **propagated** from a parent entity to its related entities. These are used with **@OneToOne, @OneToMany, ManyToOne, and @ManyToMany** relationships.