**📦 1. Entity and Table Mapping**

| **Annotation** | **Purpose** |
| --- | --- |
| @Entity | Marks the class as a Hibernate entity |
| @Table(name = "table\_name") | Maps the class to a specific database table |

**✅ Example:**

@Entity

@Table(name = "users")

public class User {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

}

**🗂️ 2. Primary Key**

| **Annotation** | **Purpose** |
| --- | --- |
| @Id | Marks the primary key |
| @GeneratedValue | Specifies auto-generation of ID |
| @SequenceGenerator | Sequence-based key generation (for Oracle/Postgres) |

**✅ Example:**

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

**🧬 3. Column Mapping**

| **Annotation** | **Purpose** |
| --- | --- |
| @Column(name = "col\_name") | Maps a field to a DB column |
| @Transient | Exclude from persistence |
| @Lob | Used for large objects (e.g., text, images) |
| @Temporal | Used for date/time fields (for Date, Calendar) |

**✅ Example:**

@Column(name = "email", nullable = false, unique = true)

private String email;

@Transient

private String temporaryToken;

**🔗 4. Relationships**

| **Annotation** | **Purpose** |
| --- | --- |
| @OneToOne | One-to-one relationship |
| @OneToMany | One-to-many |
| @ManyToOne | Many-to-one |
| @ManyToMany | Many-to-many |
| @JoinColumn | Foreign key column |
| @JoinTable | Join table for many-to-many |

**✅ Example: @ManyToOne + @JoinColumn**

@ManyToOne

@JoinColumn(name = "department\_id")

private Department department;

**✅ Example: @OneToMany**

@OneToMany(mappedBy = "department", cascade = CascadeType.ALL)

private List<Employee> employees;

**🧰 5. Validation (Hibernate Validator)**

| **Annotation** | **Purpose** |
| --- | --- |
| @NotNull | Field cannot be null |
| @Size(min=, max=) | String length |
| @Email | Valid email |
| @Min, @Max | Numeric limits |

**✅ Example:**

@NotNull

@Size(min = 3, max = 50)

private String name;

**🏷️ 6. Customizing Fetching and Cascading**

| **Annotation** | **Purpose** |
| --- | --- |
| fetch = FetchType.LAZY | Load only when needed |
| cascade = CascadeType.ALL | Apply changes to child entities |

**✅ Example:**

@OneToMany(fetch = FetchType.LAZY, cascade = CascadeType.ALL)

private List<Order> orders;

**✅ BONUS: Complete Example**

@Entity

@Table(name = "products")

public class Product {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

@Column(nullable = false)

private String name;

@ManyToOne

@JoinColumn(name = "category\_id")

private Category category;

@Temporal(TemporalType.TIMESTAMP)

private Date createdAt;

}