**Hibernate Annotation Config implementation walk through**

Hibernate allows us to **map Java classes to database tables** using annotations instead of XML files.

**Object to Relational Database Mapping in Employee Class**

In Hibernate, the Employee class represents a database table through **object-relational mapping (ORM)**. The Employee class is a **persistence class** that is annotated to define how its fields map to the database.

import javax.persistence.\*;

@Entity

@Table(name = "EMPLOYEE")

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name = "id")

private int id;

@Column(name = "first\_name")

private String firstName;

@Column(name = "last\_name")

private String lastName;

@Column(name = "salary")

private int salary;

// Getters and Setters

}

**Mapping Process:**

* The class Employee is mapped to a database table EMPLOYEE.
* Each class field (like firstName) is mapped to a column (like first\_name) using annotations.
* Primary key (id) is identified and its generation strategy is defined.

### Annotations Used for Hibernate Operations

### @Entity

* Marks the class as a Hibernate entity (i.e., a persistent Java object).
* Hibernate will map this class to a database table.

#### @Table(name = "EMPLOYEE")

* Specifies the name of the table to map to.
* Optional: If not provided, Hibernate assumes table name is same as class name.

**@Id**

* Marks the field as a primary key.

#### @GeneratedValue(strategy = GenerationType.IDENTITY)

* Specifies the generation strategy for the primary key.
* Common strategies: AUTO, IDENTITY, SEQUENCE, TABLE.
* IDENTITY: Relies on auto-increment of DB.

#### @Column(name = "column\_name")

* Maps a class field to a specific column in the table.

**Hibernate Configuration - hibernate.cfg.xml**

This XML file configures the database connection and Hibernate behavior.

<hibernate-configuration>

<session-factory>

<property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>

<property name="hibernate.connection.driver\_class">com.mysql.jdbc.Driver</property>

<property name="hibernate.connection.url">jdbc:mysql://localhost:3306/test</property>

<property name="hibernate.connection.username">root</property>

<property name="hibernate.connection.password">password</property>

</session-factory>

</hibernate-configuration>

**hibernate.dialect :** Tells Hibernate how to generate SQL for your DB (e.g., MySQLDialect, OracleDialect).

**hibernate.connection.driver\_class :** JDBC driver class for the DB (e.g., MySQL: com.mysql.cj.jdbc.Driver).

**hibernate.connection.url :** JDBC URL for the DB (e.g.,jdbc:mysql://localhost:3306/mydb).

**hibernate.connection.username :** DB username for authentication.

**hibernate.connection.password :** DB password for authentication.