## ****3. Hibernate Annotation Configuration – Detailed Walkthrough****

### ****1. Overview of Hibernate Annotations****

Hibernate allows Java developers to use **annotations** (as per **JSR 220: Java Persistence API**) to define mapping information directly within Java classes. This approach removes the need for .hbm.xml files and places the configuration closer to the business logic. It's more readable, less error-prone, and easier to refactor as the application grows.

### ****2. ORM Using Annotations in the Entity Class (****Employee.java****)****

In annotation-based configuration, the mapping metadata is added directly into the class using annotations like @Entity, @Table, @Id, and @Column. Below is an example of how the Employee class is structured:

java

CopyEdit

import jakarta.persistence.\*;

@Entity@Table(name = "employee")public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

@Column(name = "name")

private String name;

@Column(name = "salary")

private double salary;

// Getters and Setters

}

This class maps the Employee Java object to the employee table in the database.

### ****3. Annotations Explained****

**@Entity**  
Marks the class as an entity to be managed by Hibernate. It tells Hibernate that this class should be persisted to a database table.

**@Table(name = "employee")**  
Specifies the table name in the database that corresponds to this entity.

**@Id**  
Denotes the primary key field of the entity.

**@GeneratedValue(strategy = GenerationType.IDENTITY)**  
Instructs Hibernate to auto-generate the primary key using the database's identity column.

**@Column(name = "column\_name")**  
Maps the class field to a specific column in the database table.

### ****4. Hibernate Configuration File –**** hibernate.cfg.xml

The hibernate.cfg.xml file is still needed for database connection and session factory configuration. It also registers the annotated class.

xml

CopyEdit

<?xml version="1.0" encoding="UTF-8"?><!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 3.0//EN"

"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd"><hibernate-configuration>

<session-factory>

<!-- JDBC connection configuration -->

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

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

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

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

<!-- SQL dialect -->

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

<!-- Schema update and logging -->

<property name="hibernate.hbm2ddl.auto">update</property>

<property name="hibernate.show\_sql">true</property>

<!-- Register annotated entity class -->

<mapping class="com.example.Employee"/>

</session-factory></hibernate-configuration>

### ****5. Configuration Tags Explained****

**hibernate.connection.driver\_class**  
Indicates which JDBC driver should be used for the database connection.

**hibernate.connection.url**  
Provides the database connection URL.

**hibernate.connection.username / password**  
Credentials used to connect to the database.

**hibernate.dialect**  
Defines the SQL dialect Hibernate will use to generate SQL statements appropriate for the database.

**hibernate.hbm2ddl.auto**  
Controls schema management (e.g., update, create, validate).

**hibernate.show\_sql**  
Enables printing of SQL queries in the console for debugging purposes.

**<mapping class="..."/>**  
Registers the annotated Java class with Hibernate.

### ****6. Benefits of Using Annotations****

Keeps the mapping logic **close to the entity class**, making the code more cohesive.

Eliminates the need to create and maintain separate .hbm.xml files.

Improves **readability and maintainability** of the code.

Compatible with modern IDEs, JPA providers, and frameworks.

Simplifies the use of **advanced JPA features** like inheritance, lazy loading, and entity relationships.