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

1. **Introduction to Hibernate with Annotations**

Hibernate provides support for Java annotations (JSR 220) to map Java classes to database tables. This approach eliminates the need for separate XML mapping files (\*.hbm.xml) by placing metadata directly on the Java classes. Annotations are simple, concise, and easier to maintain than XML configuration.

2. **Object to Relational Mapping in Persistence Class (Employee.java)**

The Employee class in Hibernate annotation-based mapping includes metadata using annotations like @Entity, @Table, @Id, @Column, etc., which define how the class maps to a database table.

Here is an example of a simple Employee class:

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;

}

3. **Explanation of Annotations**

**@Entity**

* Declares that the class is a Hibernate entity.
* Required for Hibernate to recognize and manage the class.
* Maps the class to a table in the database.

4. **Hibernate Configuration (hibernate.cfg.xml)**

The hibernate.cfg.xml file provides database connectivity information and tells Hibernate which classes are annotated entities.

<?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>

<!-- Database connection settings -->

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

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

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

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

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

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

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

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

</session-factory>

</hibernate-configuration>

5. **Explanation of Configuration Tags**

#### hibernate.dialect

* Tells Hibernate which SQL dialect to use based on the database.
* Specifies the JDBC **driver class** used to connect to the database.
* Specifies the **JDBC connection string**.
* Specifies the **credentials** for the database.
* Specifies the **fully qualified class name** of the annotated entity.

6. **Advantages of Annotation-Based Configuration**

 No need to maintain separate XML mapping files.

 Easier to read and maintain as everything is defined in one place.

 Works well with modern Java development and tools.

 Supports advanced JPA features natively.