

**AMITY UNIVERSITY, PATNA**

**AMITY INSTITUTE OF INFORMATION TECHNOLOGY**

**Advanced Java Lab File  
LAB - 4**

**BCA**



Name : Aryan Amar  
Program/Semester : BCA – 6 ‘A’  
Enroll. Number : A45304821047  
Submitted to : Dr. Naveen Kumar Singh

## HIBERNATE (THROUGH ANNOTATIONS)

### Problem Statement :

Hibernate program to demonstrate crud operations with the help of annotations.

### Introduction :

The objective of this project is to develop a Hibernate program that performs CRUD (Create, Read, Update, Delete) operations on a table named "emp\_details". The program will provide a menu-based interface allowing users to interact with the database. Users can insert, retrieve, update, or delete records from the "emp\_details" table based on their requirements. This program will use Hibernate annotations such as @Entity and @Table for mapping Java objects to the database table, eliminating the need for separate XML mapping files.

### Problem Description:

The program will consist of the following functionalities :

#### **1. Insert Record :**

Users will have the option to insert a new record into the "emp\_details" table. Upon selecting this option, the program will prompt the user to enter details about the employee, including the employee's name and salary. The entered information will then be added as a new record in the database.

#### **2. Retrieve a Particular Record :**

Users can retrieve information about a specific employee by providing the employee's ID. If an employee with the provided ID exists in the "emp\_details" table, the program will display the corresponding information about that employee. If the employee is not found, the program will display a message indicating that the employee with the entered ID is not found.

#### **3. Retrieve All Records :**

Users can retrieve all records from the "emp\_details" table. The program will display information about all employees stored in the database, including their names and salaries.

#### **4. Update Record :**

Users will be able to update information about a particular employee. Upon selecting this option, the program will prompt the user to enter the ID of the employee whose information they want to update. If the provided ID corresponds to an employee in the database, the program will allow the user to modify the employee's salary. If the employee is not found, the program will display a message indicating that the employee with the entered ID does not exist.

#### **5. Delete Record :**

Users can delete a record of an employee from the "emp\_details" table by providing the employee's ID. If the provided ID matches an employee in the database, the program will delete the corresponding record. If the employee is not found, the program will display a message indicating that the employee with the entered ID is not found.

**6. Exit Program :**

Users will have the option to exit the program. Upon selecting this option, the Hibernate session and factory will close, allowing users to exit the program gracefully.

**Implementation Approach :**

The program will be implemented using Hibernate, a popular object-relational mapping (ORM) framework for Java. Hibernate annotations such as @Entity and @Table will be used to map Java objects to the database table. Each CRUD operation will be implemented as a separate method or class, following a modular and object-oriented approach. The program will provide a user-friendly menu interface for interacting with the database, ensuring ease of use and clarity for users.

**Expected Input and Output:**

**1. Insert Record :**

User inputs : Employee's name and salary.

**2. Retrieve a Particular Record :**

User inputs : Employee's ID.

Output : Information about the specified employee if found, or a message indicating that the employee with the entered ID is not found.

**3. Retrieve All Records :**

Output : Information about all employees stored in the database.

**4. Update Record :**

User inputs : Employee's ID and new employee's salary.

Output : Confirmation message upon successful update, or a message indicating that the employee with the entered ID is not found.

**5. Delete Record :**

User inputs : Employee's ID.

Output : Confirmation message upon successful deletion, or a message indicating that the employee with the entered ID is not found.

**6. Exit Program :**

Output : Termination of the program.

### **Tools and Technologies used:**

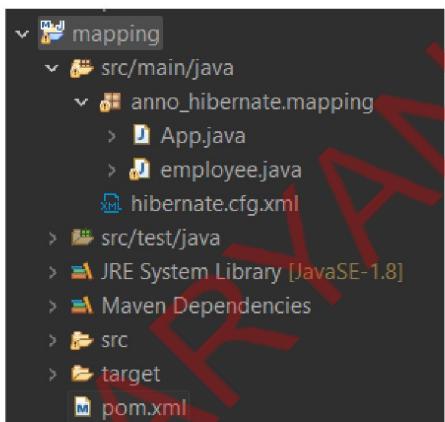
Eclipse IDE and MYSQL database.

### **Design**

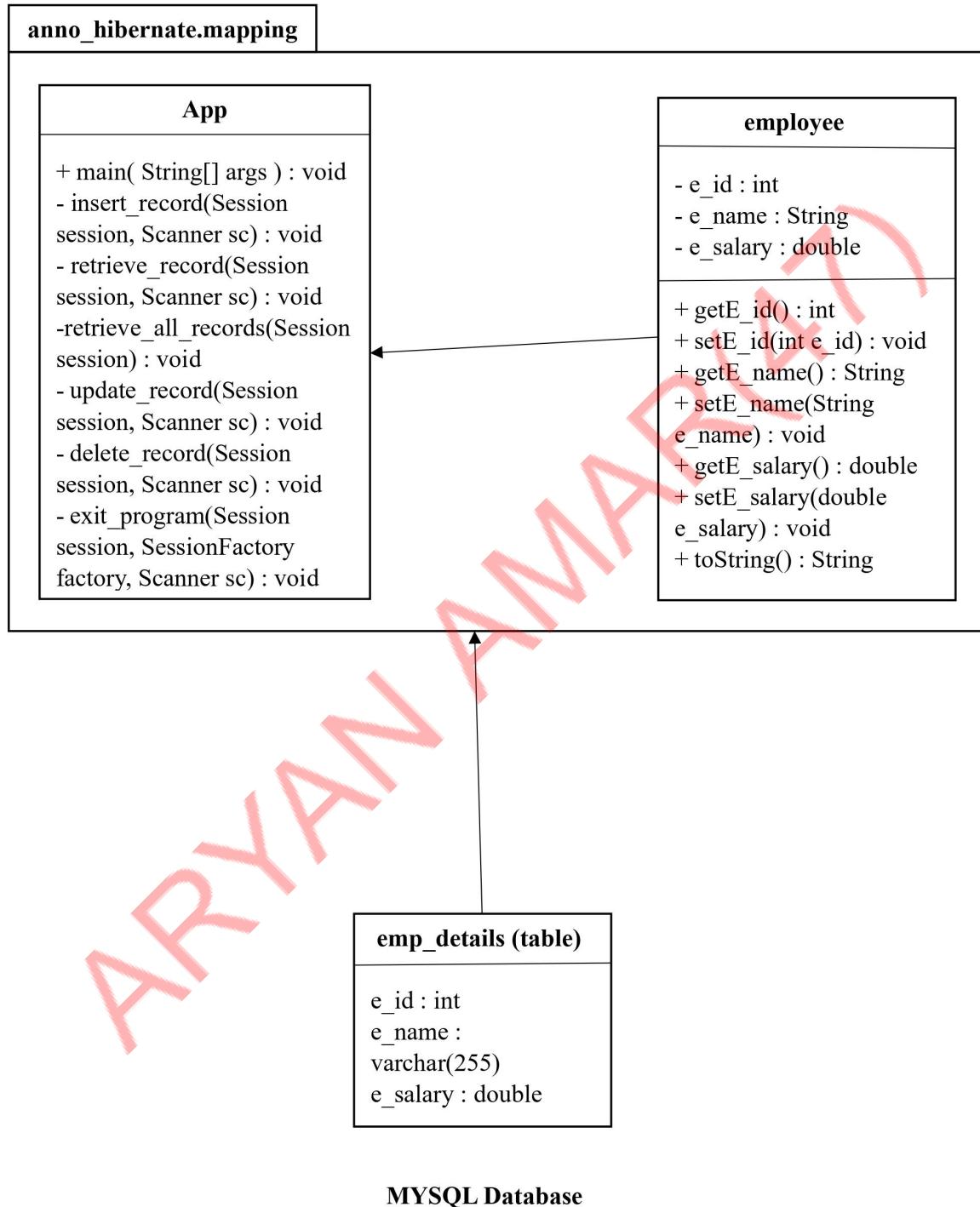
#### **Table description -**

```
mysql> desc emp_details;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| e_id  | int    | NO   | PRI | NULL    | auto_increment |
| e_name | varchar(255) | YES |     | NULL    |             |
| e_salary | double | YES |     | NULL    |             |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.02 sec)
```

#### **Folder Structure -**



**Class Diagram** – The class diagram for my project is as follows :



## CODE

### App.java

```
package anno_hibernate.mapping;

import java.util.Scanner;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;
import java.util.List;
import org.hibernate.query.Query;

public class App
{
    public static void main( String[] args )
    {
        Configuration config = new Configuration();
        config.configure();
        SessionFactory factory = config.buildSessionFactory();
        Session session = factory.openSession();
        Scanner sc = new Scanner(System.in);
        while(true)
        {
            System.out.println("\nChoose the operation you want to
perform :");
            System.out.println("1. Insert a record into the
table");
            System.out.println("2. Retrieve a particular
record from the table");
    }
}
```

APRIL AMAR(47)

## CODE

```
System.out.println("3. Retrieve all the records  
from the table");  
  
System.out.println("4. Update some record from  
the table ");  
  
System.out.println("5. Delete some record from  
the table ");  
  
System.out.println("6. Exit");  
  
System.out.println("\nPlease enter your choice  
:");  
  
int choice = sc.nextInt();  
  
switch(choice)  
{  
    case 1 : insert_record(session, sc);  
    break;  
    case 2 : retrieve_record(session, sc);  
    break;  
    case 3 : retrieve_all_records(session);  
    break;  
    case 4 : update_record(session, sc);  
    break;  
    case 5 : delete_record(session, sc);  
    break;  
    case 6 : exit_program(session, factory, sc);  
    break;  
    default : System.out.println("\nInvalid choice.  
Please try again.\n");  
}  
  
}
```

## CODE

```
private static void insert_record(Session session, Scanner sc)
{
    Transaction transaction =
    session.beginTransaction();
    System.out.println("\nPerforming INSERT
operation.....");
    System.out.println("\nPlease enter the name of
the employee -");
    String name = sc.next();
    System.out.println("Please enter the salary of
the employee -");
    double salary = sc.nextDouble();
    System.out.println();
    employee emp_insert = new employee();
    emp_insert.setE_name(name);
    emp_insert.setE_salary(salary);
    session.save(emp_insert);
    transaction.commit();
}
private static void retrieve_record(Session session, Scanner sc)
{
    Transaction transaction = session.beginTransaction();
    System.out.println("\nRetrieving a particular
employee from the table based on his/her id.....");
    System.out.println("\nEnter the id of the employee
who's information you want to retrieve - ");
    int retrieved_id = sc.nextInt();
    System.out.println();
    employee E = session.get(employee.class,
retrieved_id);
```

## CODE

```
if (E != null)
{
    System.out.println("Employee - " + E);
}
else
{
    System.out.println("Employee with ID " + retrieved_id
+ " doesn't exist.\n");
}
transaction.commit();
}

private static void retrieve_all_records(Session session)
{
    Transaction transaction = session.beginTransaction();

    System.out.println("\nRetrieving everything from the
table.....\n");

    Query<employee> query = session.createQuery("FROM
employee", employee.class);

    List<employee> employees_list = query.list();
    for (employee emp : employees_list)
    {
        System.out.println("Employee - " + emp);
    }
    transaction.commit();
}

private static void update_record(Session session, Scanner
sc)
{
    Transaction transaction = session.beginTransaction();
```

## CODE

```
System.out.println("\nUpdating the department of the
employee.....");

System.out.println("\nEnter the id of the employee
who's salary you want to update -");

int id = sc.nextInt();

employee emp_update = session.get(employee.class, id);

if(emp_update != null)

{

    System.out.println("\nEnter the new salary of the
employee -");

    double new_sal = sc.nextDouble();

    System.out.println();

    emp_update.setE_salary(new_sal);

    session.saveOrUpdate(emp_update);

    System.out.println("Salary - " + emp_update + "
updated successfully.\n");

}

else

{

    System.out.println("\nEmployee with ID " + id + "
doesn't exist.\n");

}

transaction.commit();

}

private static void delete_record(Session session, Scanner
sc)

{

    Transaction transaction = session.beginTransaction();

    System.out.println("\nDeleting some record from the
table.....");
```

## CODE

```
System.out.println("\nEnter the id of the employee  
whose information you want to delete -");  
  
int id = sc.nextInt();  
  
System.out.println();  
  
employee emp_delete = session.get(employee.class, id);  
  
if (emp_delete != null)  
{  
  
    session.delete(emp_delete);  
  
    System.out.println("Record - " + emp_delete + "  
deleted successfully.\n");  
  
}  
  
else  
  
{  
  
    System.out.println("\nEmployee with ID " + id + "  
is not found.\n");  
  
}  
  
transaction.commit();  
  
}  
  
private static void exit_program(Session session,  
SessionFactory factory, Scanner sc)  
  
{  
  
    System.out.println("\nExiting...\n");  
  
    sc.close();  
  
    session.close();  
  
    factory.close();  
  
    System.exit(0);  
  
}  
}
```

APYAMAR(+)T

## CODE

### employee.java

```
package anno_hibernate.mapping;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

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

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int e_id;
    private String e_name;
    private double e_salary;
    public employee() {
        super();
        // TODO Auto-generated constructor stub
    }
    public employee(int e_id, String e_name, double e_salary) {
        super();
        this.e_id = e_id;
        this.e_name = e_name;
        this.e_salary = e_salary;
    }
    public int getE_id() {
        return e_id;
    }
}
```

## CODE

```
}

public void setE_id() {
    this.e_id = e_id;
}

public String getE_name() {
    return e_name;
}

public void setE_name(String e_name) {
    this.e_name = e_name;
}

public double getE_salary() {
    return e_salary;
}

public void setE_salary(double e_salary) {
    this.e_salary = e_salary;
}

@Override
public String toString() {
    return "employee [e_id=" + e_id + ", e_name=" + e_name +",
    e_salary=" + e_salary + "]";
}
}
```

### hibernate.cfg.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
        "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
```

## CODE

```
"http://www.hibernate.org/dtd/hibernate-configuration-  
3.0.dtd">  
  
<hibernate-configuration>  
<session-factory>  
  
    <property  
    name="connection.driver_class">com.mysql.cj.jdbc.Driver</propert  
y>  
    <property  
    name="connection.url">jdbc:mysql://localhost:3306/multiple_db</p  
roperty>  
    <property name="connection.username">root</property>  
    <property name="connection.password">aryan123</property>  
    <property  
    name="dialect">org.hibernate.dialect.MySQLDialect</property>  
    <property  
    name="current_session_context_class">thread</property>  
    <property  
    name="cache.provider_class">org.hibernate.cache.internal.NoCache  
Provider</property>  
    <property name="show_sql">true</property>  
    <property name="hbm2ddl.auto">update</property>  
    <mapping class="anno_hibernate.mapping.employee"/>  
  
</session-factory>  
</hibernate-configuration>
```

## CODE

### pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  http://maven.apache.org/xsd/maven-4.0.0.xsd">

  <modelVersion>4.0.0</modelVersion>

  <groupId>anno_hibernate</groupId>
  <artifactId>mapping</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>jar</packaging>

  <name>mapping</name>
  <url>http://maven.apache.org</url>

  <properties>
    <project.build.sourceEncoding>UTF-
8</project.build.sourceEncoding>
  </properties>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
    <!-- https://mvnrepository.com/artifact/mysql/mysql-
connector-java -->
```

## **CODE**

```
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.33</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-
core -->

<dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-core</artifactId>
    <version>5.4.5.Final</version>
</dependency>
</dependencies>
</project>
```

# INPUT/OUTPUT

## Hibernate auto configuration

```
Problems Servers Terminal Data Source Explorer Properties Console X Progress
<terminated> App (3) [Java Application] C:\Users\aryan.p2\pool\plugins\org.eclipse.jst.openjdk.hotspot.jre.full.win32.x86_64_21.0.1.v20231028-0937\jre\bin\javaw.exe (Feb 17, 2024, 5:09:24 PM) [pid: 13252]
Feb 17, 2024 5:06:59 PM org.hibernate.Version logVersion
INFO: HHH000412: Hibernate Core (5.4.5.Final)
Feb 17, 2024 5:07:00 PM org.hibernate.annotations.common.reflection.java.JavaReflectionManager <clinit>
INFO: HCANN000001: Hibernate Commons Annotations (5.1.0.Final)
Feb 17, 2024 5:07:00 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl configure
WARN: HHH10001002: Using Hibernate built-in connection pool (not for production use!)
Feb 17, 2024 5:07:00 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001005: using driver [com.mysql.cj.jdbc.Driver] at URL [jdbc:mysql://localhost:3306/multiple_db]
Feb 17, 2024 5:07:00 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001001: Connection properties: {password=****, user=root}
Feb 17, 2024 5:07:00 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001003: Autocommit mode: false
Feb 17, 2024 5:07:00 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PooledConnections <init>
INFO: HHH000115: Hibernate connection pool size: 20 (min=1)
Feb 17, 2024 5:07:01 PM org.hibernate.dialect.Dialect <init>
INFO: HHH000400: Using dialect: org.hibernate.dialect.MySQLDialect
Feb 17, 2024 5:07:02 PM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl getIsolatedConnect
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator$C
```

## Inserting a couple of rows in the table

```
Problems Servers Terminal Data Source Explorer Properties Console X Progress
<terminated> App (3) [Java Application] C:\Users\aryan.p2\pool\plugins\org.eclipse.jst.openjdk.hotspot.jre.full.win32.x86_64_21.0.1.v20231028
Choose the operation you want to perform :
1. Insert a record into the table
2. Retrieve a particular record from the table
3. Retrieve all the records from the table
4. Update some record from the table
5. Delete some record from the table
6. Exit

Please enter your choice :
1

Performing INSERT operation.....
Please enter the name of the employee -
Aryan_Amar
Please enter the salary of the employee -
23456.85

Hibernate: insert into emp_details (e_name, e_salary) values (?, ?)
```

## INPUT/OUTPUT

```
Problems Servers Terminal Data Source Explorer Properties Console X Progress
<terminated> App (3) [Java Application] C:\Users\aryan\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.1.v2023102
Choose the operation you want to perform :
1. Insert a record into the table
2. Retrieve a particular record from the table
3. Retrieve all the records from the table
4. Update some record from the table
5. Delete some record from the table
6. Exit

Please enter your choice :
1

Performing INSERT operation.....
Please enter the name of the employee -
Shubhu_Singh
Please enter the salary of the employee -
67345.4467

Hibernate: insert into emp_details (e_name, e_salary) values (?, ?)
```

### Retrieving a particular record from the table

```
Problems Servers Terminal Data Source Explorer Properties Console X Progress
<terminated> App (3) [Java Application] C:\Users\aryan\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.1.v20231028-0937\jre\b
Choose the operation you want to perform :
1. Insert a record into the table
2. Retrieve a particular record from the table
3. Retrieve all the records from the table
4. Update some record from the table
5. Delete some record from the table
6. Exit

Please enter your choice :
2

Retrieving a particular employee from the table based on his/her id.....
Enter the id of the employee who's information you want to retrieve -
3

Employee - employee [e_id=3, e_name=Aryan_Amar, e_salary=23456.85]
```

## INPUT/OUTPUT

### Retrieving all the records from the table

```
Problems Servers Terminal Data Source Explorer Properties Console X Progress
<terminated> App (3) [Java Application] C:\Users\aryan\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.1.v20231028-0937\jre\bin\javaw.exe

Choose the operation you want to perform :
1. Insert a record into the table
2. Retrieve a particular record from the table
3. Retrieve all the records from the table
4. Update some record from the table
5. Delete some record from the table
6. Exit

Please enter your choice :
3

Retrieving everything from the table......

Hibernate: select employee0_.e_id as e_id1_0_, employee0_.e_name as e_name2_0_
Employee - employee [e_id=3, e_name=Aryan_Amar, e_salary=23456.85]
Employee - employee [e_id=4, e_name=Shubhu_Singh, e_salary=67345.4467]
```

### Updating some record from the table

```
Problems Servers Terminal Data Source Explorer Properties Console X Progress
<terminated> App (3) [Java Application] C:\Users\aryan\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.1.v20231028-0937\jre\bin\javaw.exe (Feb 17, 2024, 5:0

Choose the operation you want to perform :
1. Insert a record into the table
2. Retrieve a particular record from the table
3. Retrieve all the records from the table
4. Update some record from the table
5. Delete some record from the table
6. Exit

Please enter your choice :
4

Updating the department of the employee......

Enter the id of the employee who's salary you want to update -
3

Enter the new salary of the employee -
98764.7648

Salary - employee [e_id=3, e_name=Aryan_Amar, e_salary=98764.7648] updated successfully.

Hibernate: update emp_details set e_name=?, e_salary=? where e_id=?
```

## INPUT/OUTPUT

### Deleting some record from the table

```
Problems Servers Terminal Data Source Explorer Properties Console Progress
terminated> App (3) [Java Application] C:\Users\aryan\p2\pool\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64_21.0.1.v20231028-0937\jre\bin\javaw.exe (Feb 17, 2024, 5:06:58
Choose the operation you want to perform :
1. Insert a record into the table
2. Retrieve a particular record from the table
3. Retrieve all the records from the table
4. Update some record from the table
5. Delete some record from the table
6. Exit

Please enter your choice :
5

Deleting some record from the table......

Enter the id of the employee whose information you want to delete -
4

Record - employee [e_id=4, e_name=Shubhu_Singh, e_salary=67345.4467] deleted successfully.

Hibernate: delete from emp_details where e_id=?
```

### Exiting the program

```
Choose the operation you want to perform :
1. Insert a record into the table
2. Retrieve a particular record from the table
3. Retrieve all the records from the table
4. Update some record from the table
5. Delete some record from the table
6. Exit

Please enter your choice :
6

Exiting...

Feb 17, 2024 5:09:24 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnection
INFO: HHH10001008: Cleaning up connection pool [jdbc:mysql://localhost:3306/multiple_db]
```