Student Name: Rishav Kumar **UID: 22MCC20039**
Section/Group: 22MCD-1 (A) **Semester: 2nd**
Branch: MCA-CC &DevOps **Date of Submission: 12-05-2023**
Subject: Advanced Internet Programming Lab **Subject Code: 22CAP-686**

Experiment No. 8

Write the database operations as Insert, delete, update, search operation in Hibernate

1. **Aim/Overview of the practical:**
   Perform CRUD operation with the help of Hibernate.

   Task to be done:
   Perform CRUD operation with the help of Hibernate.

2. **Algorithm/Flowchart :**
   Step 1: Create one java application with name Exp1.
   Step 2: Now right click on source package>>new>>others and then select hibernate and select hibernate configuration wizard and select database and click on finish.
   Step 3: Right click on default package>>new>>others and then select hibernate and select hibernate reverse engineering wizard. Then select available table employee and click on add.
   Step 4: Right click on source package>>new>>java package (with name POJO).
   Step 5: Now right click on POJO>>new>>other and then click on hibernate and select hibernate mapping files and POJO's from database.
   Step 6: Create one more package with name connection. right click on connection>>new>>other and then click on hibernate and select HibernateUtil.java.
   Step 7: Now create one java application to perform CRUD operations.

3. **Code for experiment/practical:**

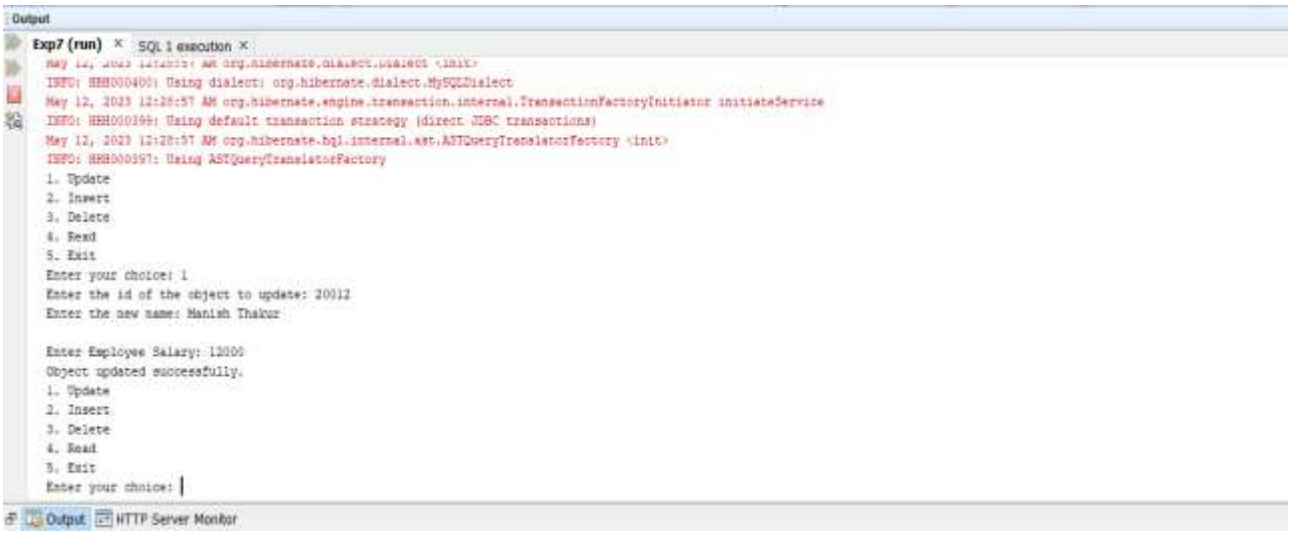
```
package connection;

import POJO.Employee;
import java.util.Scanner;
import org.hibernate.Session;
import org.hibernate.Transaction;

/**
 *
 * author Anshul
```

```
 */
public class EmployeeDB {

   public static void insert(Session session,Transaction tx,int Id,String Name,Integer Salary){
      Employee e=new Employee();
      e.setEmpId(Id);
      e.setEmpName(Name);
      e.setEmpSalary(Salary);
      tx=session.beginTransaction();
      session.save(e);
      tx.commit();
      System.out.println("Object inserted successfully.");
   }

   public static void update(Session session,Transaction tx, int updateId,String newName,Integer Salary){
      tx=session.beginTransaction();
      Employee e =(Employee) session.get(Employee.class, updateId);

      if (e != null) {
         e.setEmpName(newName);
         e.setEmpSalary(Salary);
         session.update(e);
         tx.commit();
         System.out.println("Object updated successfully.");
      }
      else {
         System.out.println("Object not found.");
      }
   }

   public static void delete(Session session, Transaction tx, int id){
      tx=session.beginTransaction();
      Employee e=(Employee) session.get(Employee.class, id);
      if(e != null) {
        session.delete(e);
        tx.commit();
        System.out.println("Object deleted successfully.");
   }   else {
        System.out.println("Object not found.");
   }
   }

   public static void read(Session session,int id){
      Employee e=(Employee)session.get(Employee.class,id);
      System.out.print("Employee Id    :" + e.getEmpId() + "\n");
      System.out.print("Employee Name  :" + e.getEmpName() + "\n");
      System.out.print("Employee Salary:" + e.getEmpSalary() + "\n");
   }
   public static void main(String[] args){
      Scanner scanner = new Scanner(System.in);
```

```
Transaction tx = null;
Session session = null;
try {
    session = Controller.getSessionFactory().openSession();
    while (true) {
        System.out.println("1. Update");
        System.out.println("2. Insert");
        System.out.println("3. Delete");
        System.out.println("4. Read");
        System.out.println("5. Exit");
        System.out.print("Enter your choice: ");
        int choice = scanner.nextInt();
        switch (choice) {
            case 1:
                System.out.print("Enter the id of the object to update: ");
                int updateId = scanner.nextInt();
                scanner.nextLine();
                System.out.print("Enter the new name: ");
                String newName = scanner.next();
                System.out.print("Enter Employee Salary: ");
                int salary=Integer.parseInt(scanner.next());
                update(session, tx, updateId, newName,salary);
                break;

            case 2:
                System.out.println("Enter the details: ");
                System.out.println("Enter Employee Id: ");
                int Id = scanner.nextInt();
                System.out.println("Enter Employee Name: ");
                String name = scanner.next();
                System.out.print("Enter Employee Salary: ");
                int sal=Integer.parseInt(scanner.next());
                insert(session, tx,Id,name,sal);
                break;

            case 3:
                System.out.print("Enter the id of the object to delete: ");
                int deleteId = scanner.nextInt();
                delete(session, tx, deleteId);
                break;

            case 4:
                System.out.print("Search details:/n");
                System.out.println("Enter Employee Id: ");
                int id = scanner.nextInt();
                read(session,id);
                break;

            case 5:
                System.out.println("Exiting program.");
```

```
                return;

            default:
                System.out.println("Invalid choice. Please try again.");
                break;
        }
    }
} catch (Exception e) {
    e.printStackTrace();
} finally {
    session.close();
}
}

}
```

4. Result/Output/Writing Summary:



```
Output
Exp7 (run)   SQL 1 execution
Enter the new name: manian thakur

Enter Employee Salary: 12000
Object updated successfully.
1. Update
2. Insert
3. Delete
4. Read
5. Exit
Enter your choice: 2
Enter the details:
Enter Employee Id:
20008
Enter Employee Name:
Pratham Dua
Enter Employee Salary: 12644
Object inserted successfully.
1. Update
2. Insert
3. Delete
4. Read
5. Exit
Enter your choice:
```

```
Output
Exp7 (run)   SQL 1 execution
Enter employee id:
20008
Enter Employee Name:
Pratham Dua
Enter Employee Salary: 12644
Object inserted successfully.
1. Update
2. Insert
3. Delete
4. Read
5. Exit
Enter your choice: 4
Search details:/nEnter Employee Id:
20011
Employee Id    :20011
Employee Name  :Anshul Gupta
Employee Salary:15000
1. Update
2. Insert
3. Delete
4. Read
5. Exit
Enter your choice:
Output   HTTP Server Monitor
```

Learning outcomes:

1. Learn to perform CRUD operation in Hibernate.

2. Learn to implement Hibernate.