**Experiment No. 8**

**Student Name:** GauravKumar   **UID:** 22MCC20177

**Branch:** MCA **-** CCD   **Section/Group:** 22MCD1 **/** GrpA

**Semester:** II  **Date of Performance:** 12th May 2023

**Subject Name:** Advanced Internet Programming Lab **Subject Code:** 22CAP-686

1. **Aim/Overview of the practical:**

Perform CRUD operation with the help of Hibernate.

1. **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:** And now create one file and write simple code to insert data.

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

import POJO.Employee;

import java.util.Scanner;

import org.hibernate.Session;

import org.hibernate.Transaction;

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();

}

}

}

1. **Result/Output/Writing Summary:**

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

A picture containing text, software, screenshot

Description automatically generated

1. **Learning outcomes (What I have learnt):**

1. Learn to perform CRUD operation in Hibernate.

**2.** Learn to implement Hibernate.

**Evaluation Grid:**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Parameters | Marks Obtained | Maximum Marks |
| 1. | Demonstration and Performance (Pre Lab-Quiz) |  | 5 |
| 2. | Worksheet |  | 10 |
| 3. | Post Lab Quiz |  | 5 |