DEPARTMENTOF
COMPUTERSCIENCE&ENGINEERING

Discover. Learn. Empower.

Experiment - 9

Student Name: Manjot Singh
UID: 23BCS12549
Branch: BE-CSE
Section/Group: KRG-2B
Semester: 5th
Date of Performance:21/10/25
Subject Name: PBLJ
Subject Code: 23CSH-304

1. Aim:

Perform CRUD (Create, Read, Update, Delete) operations on a Student entity using Hibernate ORM with MySQL.

2. Objective:

To learn Hibernate configuration, entity mapping, and CRUD execution withMySQL.

3. Apparatus / Input Used:

Java, Hibernate, MySQL, Eclipse / IntelliJ, hibernate.cfg.xml

4. Procedure:

- Configure MySQL database and add Hibernate dependencies.
- Create hibernate.cfg.xml with DB credentials.
- Create Student.java with @Entity, @Id, @GeneratedValue annotations.
- Create HibernateUtil class for SessionFactory.
- Implement CRUD using session.save(), session.get(), session.update(),session.delete().
- Test using a main class

5. Code

Student.java:

```
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;

@Entity
public class Student {
   @Id
   @GeneratedValue(strategy = GenerationType.IDENTITY)
```

```
    private int id;
    private String name;
    private int age;

    public Student() {}

    public Student(String name, int age) {
        this.name = name;
        this.age = age;
    }

    public int getId() {
        return id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    @Override
    public String toString() {
        return "Student{id=" + id + ", name='" + name + "', age=" + age + "}";
    }
}
```

Main.java

```
import org.hibernate.Session;
import org.hibernate.Transaction;

public class Main {
    public static void main(String[] args) {
        Session session = HibernateUtil.getSessionFactory().openSession();
        Transaction tx = session.beginTransaction();

        Student s = new Student();
        s.setName("Alice");
        s.setAge(21);
        session.save(s);

        Student st = session.get(Student.class, 1);
        if (st != null) {
            st.setAge(24);
            st.setName("Alice Johnson");
            session.update(st);
            session.delete(st);
        }
        tx.commit();
        session.close();

        System.out.println("CRUD operations completed successfully.");
    }
}
```

Sample Output:

```
Insert Success!
Fetch: ID=1, Name=Alice, Age=22
Update Success: Age changed to 23
```