



## **Experiment - 9**

**Student Name:** Shivam Kumar Tiwari

**UID:** 23BCS12755

**Branch:** BE-CSE

**Section/Group:** KRG-2B

**Semester:** 5<sup>th</sup>

**Date of Performance:** 21/10/25

**Subject Name:** Project Based Learning in Java

**Subject Code:** 23CSH-304

### **Aim:**

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

### **Objective:**

To learn Hibernate configuration, entity mapping, and CRUD execution with MySQL.

### **Apparatus / Tools Used:**

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

### **Requirements:**

- Create Student entity (id, name, age)
- Configure Hibernate using hibernate.cfg.xml
- Implement SessionFactory
- Perform Insert, Read, Update, Delete

### **Procedure:**

1. Configure MySQL database and add Hibernate dependencies.
2. Create hibernate.cfg.xml with DB credentials.
3. Create Student.java with @Entity, @Id, @GeneratedValue annotations.
4. Create HibernateUtil class for SessionFactory.
5. Implement CRUD using session.save(), session.get(), session.update(),



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
session.delete();
```

6. Test using a main class.

## Code: (Student.java)

```
@Entity public class Student

{ @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY) private int
    id; private String name; private int age;

    //getters, setters, constructors

}
```

## Code: (Main CRUD Test)

```
Session session = HibernateUtil.getSessionFactory().openSession();

Transaction tx = session.beginTransaction();

Student s = new Student(); s.setName("John");
s.setAge(22); session.save(s);

Student st = session.get(Student.class,1); st.setAge(23); session.update(st);

session.delete(st); tx.commit();
session.close();
```

## Sample Output:

Insert Success!

Fetch: ID=1, Name=John, Age=22 Update Success:  
Age changed to 23 Delete Success!