

# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CHANDIGARH  
UNIVERSITY

Discover. Learn. Empower.

## Experiment - 9

**Student Name:** Hemant Narain Jha  
**Branch:** BE-CSE  
**Semester:** 5<sup>th</sup>  
**Subject Name:** PBLJ

**UID:** 23BCS10022  
**Section/Group:** KRG-2B  
**Date of Performance:** 21/10/25  
**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 / Input 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

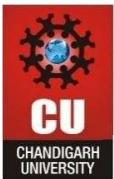
### **Procedure:**

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(),

### **Code**

#### **Student.java**

```
@Entity  
public class Student
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CHANDIGARH  
UNIVERSITY

Discover. Learn. Empower.

```
{  
@Id  
@GeneratedValue(strategy = GenerationType.IDENTITY)  
private int id;  
private String name;  
private int age;  
//getters, setters, constructors  
}
```

## Main.java

```
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!
```