

# **EXPERIMENT -2**

## **Hibernate CRUD Operation**

### **App.java**

```
package com.example;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
public class App {

    private static final SessionFactory factory = new Configuration()
        .configure("hibernate.cfg.xml")
        .buildSessionFactory();

    public static void main(String[] args) {
        try {
            createStudent();
            readStudent();
            updateStudent();
            readStudent();
            deleteStudent();
            readStudent();
        } finally {
            factory.close();
        }
    }

    public static void createStudent() {
        try (Session session = factory.openSession()) {
            session.beginTransaction();
            Student s = new Student(11, "Sai Prasad", 95);
            session.persist(s);
        }
    }
}
```

```
        session.getTransaction().commit();
        System.out.println("Student Created");
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public static void readStudent() {
    try (Session session = factory.openSession()) {
        Student s = session.get(Student.class, 11);
        if (s != null) {
            System.out.println("Student Found: " + s.getId() + " " + s.getName() + " " +
s.getMarks());
        } else {
            System.out.println("Student Not Found");
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public static void updateStudent() {
    try (Session session = factory.openSession()) {
        session.beginTransaction();
        Student s = session.get(Student.class, 11);
        if (s != null) {
            s.setMarks(100);
            s.setName("Sai Prasad Updated");
            session.merge(s);
            System.out.println("Student Updated");
        } else {
            System.out.println("Student Not Found for Update");
        }
    }
}
```

```

        session.getTransaction().commit();

    } catch (Exception e) {
        e.printStackTrace();
    }
}

public static void deleteStudent() {
    try (Session session = factory.openSession()) {
        session.getTransaction().begin();
        Student s = session.get(Student.class, 11);
        if (s != null) {
            session.remove(s);
            System.out.println("Student Deleted");
        } else {
            System.out.println("Student Not Found for Delete");
        }
        session.getTransaction().commit();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

## Hibernate.cfg.xml code

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
"-//Hibernate/Hibernate Configuration DTD 3.0//EN"
"https://hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>
    <session-factory>
        <property
name="hibernate.connection.driver_class">com.mysql.cj.jdbc.Driver</property>

```

```

<!-- Updated JDBC URL with 'allowPublicKeyRetrieval=true' and 'useSSL=false'
-->

<property
name="hibernate.connection.url">jdbc:mysql://localhost:3306/experiment?allowPubli
cKeyRetrieval=true&useSSL=false&serverTimezone=UTC</property>

<property name="hibernate.connection.username">root</property>
<property name="hibernate.connection.password">1313</property>

<property
name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>

<property name="hibernate.hbm2ddl.auto">update</property>
<property name="hibernate.show_sql">true</property>
<mapping class="com.example.Student"/>

</session-factory>

</hibernate-configuration>

```

## OUTPUT:

```

<!DOCTYPE hibernate-configuration PUBLIC
      "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
      "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
    <session-factory>
        <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/experiment?allowPubli
cKeyRetrieval=true&useSSL=false&serverTimezone=UTC</property>
        <property name="hibernate.connection.username">root</property>
        <property name="hibernate.connection.password">1313</property>
        <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
        <property name="hibernate.hbm2ddl.auto">update</property>
        <property name="hibernate.show_sql">true</property>
        <mapping class="com.example.Student"/>
    </session-factory>
</hibernate-configuration>

```

INFO: HHH000489: No JTA platform available (set 'hibernate.transaction.jta.platform' to enable JTA platform integration)  
Dec 24, 2025 10:26:22 AM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl getIsolatedConnection  
INFO: HHH0001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator\$ConnectionProvider@468e8565] for (non-JTA) DDL execution was not in auto-commit mode; the Connection 'local transaction' will be committed and the Connection will be set into auto-commit mode.

```

== Student Management ==
1. Add Student
2. View Student
3. Update Student
4. Delete Student
5. Exit
Choose an option: 

```