

HIBERNATE

App.java

```
package com.elysiumm.DemoHibb;
import java.util.List;
import java.util.Random;
import org.hibernate.Query;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
import org.hibernate.service.ServiceRegistry;
import org.hibernate.service.ServiceRegistryBuilder;
public class App
{
    public static void main( String[] args) {
        Configuration config = new Configuration().configure().addAnnotatedClass(Student.class);
        ServiceRegistry registry = new
        ServiceRegistryBuilder().applySettings(config.getProperties()).buildServiceRegistry();
        SessionFactory sf = config.buildSessionFactory(registry);
        Session session = sf.openSession();
        session.beginTransaction();

        // Query q = session.createQuery("from Student");    //(Fetch all student data)
        // Query q = session.createQuery("from Student where marks > 50");    //(it show only that record
        // where marks will be greater than 50)

        /* Query q = session.createQuery("from Student where rollno=1");
        Student student = (Student) q.uniqueResult();
        List<Student> students = q.list();
        System.out.println(student);

        for(Student s : students)
        {
            System.out.println(s);
        }
        */
        Random r = new Random();
        for(int i=1;i<=50;i++)
        {
            Student s = new Student();
            s.setRollno(i);
            s.setName("Name" + i);
            s.setMarks(r.nextInt(100));
            session.save(s);
        }
    }
}
```

```
        session.getTransaction().commit();
    }
}
```

Student.java

```
package com.elysiumm.DemoHibb;
import javax.persistence.Entity;
import javax.persistence.Id;
@Entity
public class Student
{
    @Id
    private int rollno;
    private String name;
    private int marks;

    public int getRollno() {
        return rollno;
    }
    public void setRollno(int rollno) {
        this.rollno = rollno;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getMarks() {
        return marks;
    }
    public void setMarks(int marks) {
        this.marks = marks;
    }

    @Override
    public String toString() {
        return "Student [rollno=" + rollno + ", name=" + name + ", marks=" + marks + "];"
    }
}
```

Hibernate.cfg.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
    "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
    <session-factory>
        <property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
        <property name="hibernate.connection.password">Harini**2003</property>
        <property name="hibernate.connection.username">root</property>
        <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/onetooone</property>
        <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>

        <property name="show_sql">true</property>
    </session-factory>
    <property name="hbm2ddl.auto">create</property>

    <mapping class="practis.com.OneToOne.UserDetails"></mapping>
    <mapping class="practis.com.OneToOne.Vehicle"></mapping>
</hibernate-configuration>
```

Pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-
4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.elysiumm</groupId>
    <artifactId>DemoHibb</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <packaging>jar</packaging>

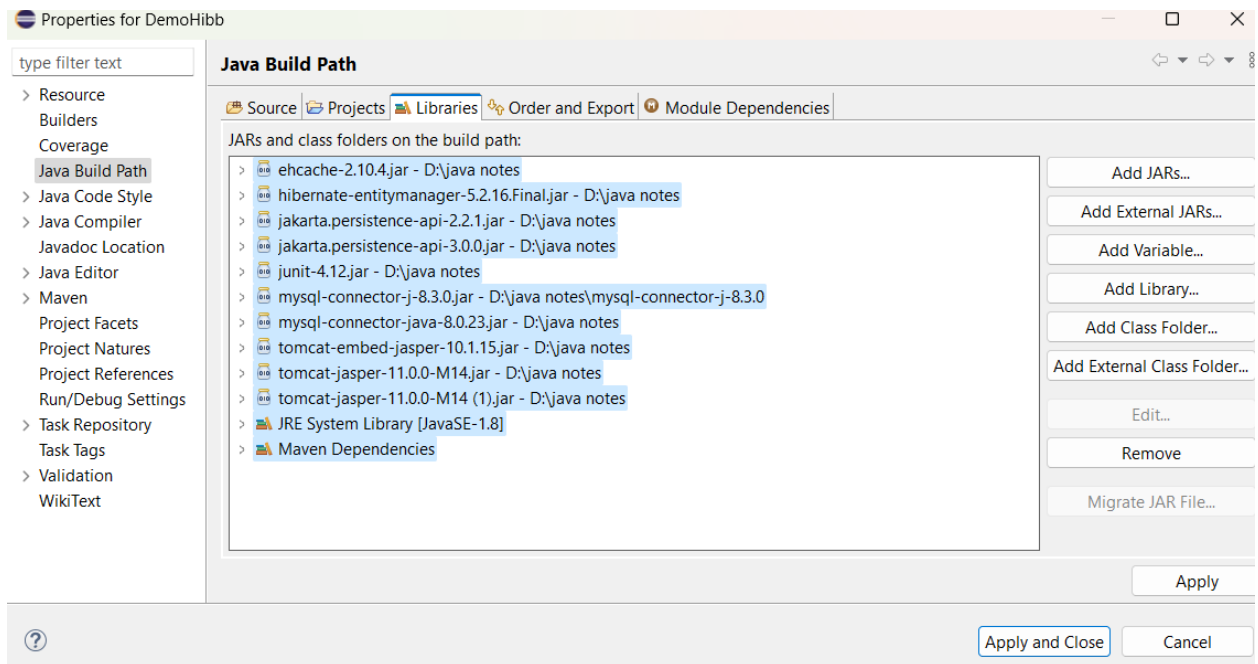
    <name>DemoHib</name>
    <url>http://maven.apache.org</url>

    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>

    <dependencies>
        <dependency>
            <groupId>org.hibernate</groupId>
            <artifactId>hibernate-core</artifactId>
            <version>4.1.6.Final</version>
```

```
</dependency>

<dependency>
  <groupId>net.sf.ehcache</groupId>
  <artifactId>ehcache</artifactId>
  <version>2.10.3</version>
</dependency>
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-ehcache</artifactId>
  <version>4.1.6.Final</version>
</dependency>
<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>5.1.47</version>
</dependency>
<dependency>
  <groupId>javax.xml.bind</groupId>
  <artifactId>jaxb-api</artifactId>
  <version>2.3.0</version>
</dependency>
<dependency>
  <groupId>junit</groupId>
  <artifactId>junit</artifactId>
  <version>3.8.1</version>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>jakarta.transaction</groupId>
  <artifactId>jakarta.transaction-api</artifactId>
  <version>2.0.0</version> <!-- Replace with the latest version -->
</dependency>
<dependency>
  <groupId>jakarta.xml.bind</groupId>
  <artifactId>jakarta.xml.bind-api</artifactId>
  <version>3.0.0</version> <!-- Replace with the latest version -->
</dependency>
<dependency>
  <groupId>com.fasterxml</groupId>
  <artifactId>classmate</artifactId>
  <version>1.5.1</version> <!-- Replace with the latest version -->
</dependency>
</dependencies>
</project>
```

DEPENDENCIES

```

<dependency>
<groupId>jakarta.transaction</groupId>
<artifactId>jakarta.transaction-api</artifactId>
<version>2.0.0</version> <!-- Replace with the latest version -->
</dependency>
<dependency>
<groupId>jakarta.xml.bind</groupId>
<artifactId>jakarta.xml.bind-api</artifactId>
<version>3.0.0</version> <!-- Replace with the latest version -->
</dependency>
<dependency>
<groupId>com.fasterxml</groupId>
<artifactId>classmate</artifactId>
<version>1.5.1</version> <!-- Replace with the latest version -->
</dependency>

```

MYSQL WORKBENCH SCHEMA

```

Query 1    SQL File 3*    SQL File 4*    SQL File 5* x student
Limit to 1000 rows

1 • CREATE DATABASE demo;
2 • use demo;
3 • CREATE TABLE Student (
4     rollno INT NOT NULL,
5     name VARCHAR(50),
6     marks INT,
7     PRIMARY KEY (rollno)
8 );
9 • INSERT INTO Student (rollno, name, marks) VALUES (1, 'John Doe', 85);
10 • INSERT INTO Student (rollno, name, marks) VALUES (2, 'Jane Smith', 92);
11 • INSERT INTO Student (rollno, name, marks) VALUES (3, 'Smith', 50);
12 • INSERT INTO Student (rollno, name, marks) VALUES (4, 'Jane', 70);
13 • INSERT INTO Student (rollno, name, marks) VALUES (5, 'jeya', 80);
14
15
16
17

```

UPDATED VALUES

Database Explorer:

- demo
 - Tables
 - student
 - Views
 - Stored Proc
 - Functions
- dummy
- it
- library
- onetoone
- projdb
- register
- sakila

Table: student

Columns:

- rollno int PK
- marks int
- name varchar(2)

Result Grid:

rollno	marks	name
1	26	Name1
2	6	Name2
3	36	Name3
4	73	Name4
5	3	Name5
6	63	Name6
7	53	Name7
8	90	Name8
9	46	Name9
10	65	Name10
11	41	Name11
12	81	Name12
13	47	Name13
14	40	Name14
15	28	Name15

student 2 x

Output:

#	Time	Action	Message
1	19:29:39	INSERT INTO Student (rollno, name, marks) VALUES (5, 'jeya', 80)	1 row(s) affected
2	19:42:30	SELECT * FROM demo.student LIMIT 0, 1000	1 row(s) returned
3	19:44:34	SELECT * FROM demo.student LIMIT 0, 1000	50 row(s) returned