*Student.java *Operation.java *Main.java

```java
package com.tech;

import javax.persistence.Entity;
@Data
@Entity
public class Student {
    @Id
    private int S_id;
    private String S_name;
    private String address;
    private long mono;

}

```

*Student.java *Operation.java *Main.java *HibernateUtil.java Operation.java Main.java Main1.java

```java
package com.tech;

import java.util.HashMap;

public class HibernateUtil {

    private static StandardServiceRegistry registory;
    private static SessionFactory sf;

    public static SessionFactory getSessionFactory() {
        Map<String, Object> setting = new HashMap<>();
        setting.put(Environment.DRIVER, "com.mysql.jdbc.Driver");
        setting.put(Environment.URL, "jdbc:mysql://localhost:3306/hbmstudent");
        setting.put(Environment.USER, "root");
        setting.put(Environment.PASS, "root");
        setting.put(Environment.HBM2DDL_AUTO, "update");
        setting.put(Environment.DIALECT, "org.hibernate.dialect.MySQL55Dialect");

        setting.put(Environment.SHOW_SQL, "true");

        registory = new StandardServiceRegistryBuilder().applySettings(setting).build();
        MetadataSources ms = new MetadataSources(registory).addAnnotatedClass(Student.class);
        Metadata md = ms.getMetadataBuilder().build();
        sf = md.getSessionFactoryBuilder().build();
        return sf;
    }
}

```

```java
package com.tech;

import java.util.List;

public class Operation {
    Scanner sc = new Scanner(System.in);
    Student s = new Student();
    SessionFactory sf = HibernateUtil.getSessionFactory();
    Session session = sf.openSession();

    public void create() {
        Transaction t = session.beginTransaction();
        System.out.println("Enter id");
        s.setS_id(sc.nextInt());
        System.out.println("Enter name");
        s.setS_name(sc.next());
        System.out.println("Enter address");
        s.setAddress(sc.next());
        System.out.println("Enter mono");
        s.setMono(sc.nextLong());

        session.save(s);
        t.commit();
        session.close();
        sf.close();
        System.out.println("save data successfully");

    }

```

```
public void getData(int id)

{
    if (id == 1) {

        System.out.println("Enter id you want to get data");
        s.setS_id(sc.nextInt());
        Student stu = session.get(Student.class, s.getS_id());
        System.out.println(stu);
    } else {
        List<Student> list = session.createQuery("from Student").list();
        System.out.println(list);
        for (Student s : list) {
            System.out.println(s.getS_id());
            System.out.println(s.getS_name());
            System.out.println(s.getAddress());
            System.out.println(s.getMono());
            System.out.println();
        }
    }

    session.close();
    sf.close();

    System.out.println("get data successfully");

}

  public void deleteData() {

      System.out.println("Enter id you want to delete ");
      s.setS_id(sc.nextInt());
      Student id = session.get(Student.class, s.getS_id());
      session.delete(id);
      session.beginTransaction().commit();
      session.close();
      sf.close();
      System.out.println("delete data successfully");

  }

}
```

```java
package com.tech;

import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        while (true) {
            Operation op = new Operation();
            Scanner sc = new Scanner(System.in);
            System.out.println("enter your choice");
            System.out.println("1) Create \n2)Read \n3)Delete ");
            int x = sc.nextInt();
            switch (x) {
            case 1:
                op.create();
                break;
            case 2:
                int count = 0;
                System.out.println("1)get single data \n2)get all data");
                int y = sc.nextInt();
                switch (y) {
                case 1:
                    op.getData(1);
                    break;
                case 2:
                    op.getData(0);
                    break;
                }
                    break;
                }
            case 3:
                op.deleteData();
                break;

            default:
                break;
            }
        }
    }
```