Q1 Using hibernate ,Create Student table with Student ID Name and also create a Course table

**Demonstrate the operations** 

I Many to Many Bidirectional

ii many to many unidirectional.

- MANYTOMANY Bi Direction
- 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>
cproperty name="hibernate.connection.driver class">
com.mysql.cj.jdbc.Driver
</property>
property name="hibernate.connection.url">
jdbc:mysql://localhost:3306/cdac tvm?useSSL=false
</property>
property
name="hibernate.connection.username">root
property
name="hibernate.connection.password">patil123
ty>
```

#### Student

```
package com.manytomanybidemo.pojo;

import java.util.Objects;
import java.util.Set;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.JoinTable;
import javax.persistence.ManyToMany;
import javax.persistence.Table;
```

@Entity

```
@Table(name = "student")
public class Student {
       @ld
       @Column(name = "stud_id")
       private int studentId;
       @Column(name = "stud_name")
       private String studentName;
       @ManyToMany
       @JoinTable(name = "stud_course", joinColumns = { @JoinColumn(name = "stud_id") },
       inverseJoinColumns = {@JoinColumn(name = "course_id") })
       private Set<Course> course;
       public Student() {
              super();
       }
       public int getStudentId() {
              return studentId;
       }
       public void setStudentId(int studentId) {
              this.studentId = studentId;
       }
       public String getStudentName() {
              return studentName;
       }
       public void setStudentName(String studentName) {
              this.studentName = studentName;
       }
```

```
public Set<Course> getCourse() {
       return course;
}
public void setCourse(Set<Course> course) {
       this.course = course;
}
@Override
public int hashCode() {
       return Objects.hash(course, studentId, studentName);
}
@Override
public boolean equals(Object obj) {
       if (this == obj)
               return true;
       if (obj == null)
               return false;
       if (getClass() != obj.getClass())
               return false;
       Student other = (Student) obj;
       return Objects.equals(course, other.course) && studentId == other.studentId
                       && Objects.equals(studentName, other.studentName);
}
```

}

```
package com.manytomanybidemo.pojo;
import java.util.Objects;
import java.util.Set;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.ld;
import javax.persistence.ManyToMany;
import javax.persistence.Table;
@Entity
@Table(name="course")
public class Course {
       @ld
       @Column(name="course_id")
       private int courseld;
       @Column(name="name")
       private String name;
       @ManyToMany
       private Set<Student>student;
       public Course() {
               super();
       }
       public int getCourseld() {
               return courseld;
       }
       public void setCourseId(int courseId) {
               this.courseld = courseld;
```

```
}
public String getName() {
       return name;
}
public void setName(String name) {
       this.name = name;
}
public Set<Student> getStudent() {
       return student;
}
public void setStudent(Set<Student> student) {
       this.student = student;
}
@Override
public int hashCode() {
       return Objects.hash(courseld, name);
}
@Override
public boolean equals(Object obj) {
       if (this == obj)
               return true;
       if (obj == null)
               return false;
       if (getClass() != obj.getClass())
               return false;
       Course other = (Course) obj;
       return courseld == other.courseld && Objects.equals(name, other.name);
}
```

```
}
```

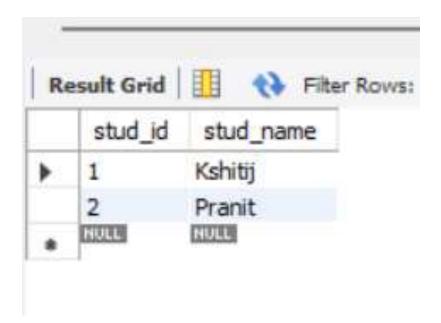
Main

```
package com.manytomanybidemo.main;
import java.util.HashSet;
import java.util.Set;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;
import com.manytomanybidemo.pojo.Course;
import com.manytomanybidemo.pojo.Student;
public class AppMain {
       public static void main(String args[]) {
               Configuration config=new Configuration();
               config.configure("hibernate.cfg.xml");
               SessionFactory sf=config.buildSessionFactory();
               Session s= sf.openSession();
```

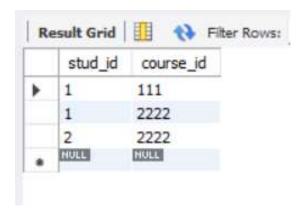
```
Transaction tr=s.beginTransaction();
Student s1 = new Student();
s1.setStudentId(1);
s1.setStudentName("Kshitij");
Student s2 = new Student();
s2.setStudentId(2);
s2.setStudentName("Pranit");
Course c1 = new Course();
c1.setCourseId(111);
c1.setName("cricket");
Course c2 = new Course();
c2.setCourseId(2222);
c2.setName("tennis");
Set<Course> cset1 = new HashSet<>();
cset1.add(c1);
cset1.add(c2);
Set<Course> cset2 = new HashSet<>();
cset2.add(c2);
s1.setCourse(cset1);
s2.setCourse(cset2);
s.save(c1);
s.save(c2);
s.save(s1);
s.save(s2);
```

```
tr.commit();
System.out.println("Object Save");
sf.close();
}
```

}







- ManyToMany Uni
- Student

package com.manytomanyunidemo.pojo;

```
import java.util.Objects;
import java.util.Set;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.ManyToMany;
import javax.persistence.Table;

@Entity
@Table(name="course")
public class Course {
    @Id
    @Column(name="course_id")
    private int courseld;
    @Column(name="name")
    private String name;
```

```
public Course() {
       super();
}
public int getCourseId() {
       return courseld;
}
public void setCourseId(int courseId) {
       this.courseld = courseld;
}
public String getName() {
       return name;
}
public void setName(String name) {
       this.name = name;
}
@Override
public int hashCode() {
       return Objects.hash(courseld, name);
}
@Override
public boolean equals(Object obj) {
       if (this == obj)
               return true;
       if (obj == null)
               return false;
       if (getClass() != obj.getClass())
               return false;
       Course other = (Course) obj;
       return courseld == other.courseld && Objects.equals(name, other.name);
}
```

# COURSE

```
package com.manytomanyunidemo.pojo;
import java.util.Objects;
import java.util.Set;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.ld;
import javax.persistence.JoinColumn;
import javax.persistence.JoinTable;
import javax.persistence.ManyToMany;
import javax.persistence.Table;
@Entity
@Table(name = "student")
public class Student {
   @ld
   @Column(name = "stud_id")
   private int studentId;
   @Column(name = "stud_name")
   private String studentName;
   @ManyToMany
   @JoinTable(name = "stud_course", joinColumns = { @JoinColumn(name = "stud_id") },
   inverseJoinColumns = {@JoinColumn(name = "course_id") })
   private Set<Course> course;
```

```
public Student() {
       super();
}
public int getStudentId() {
       return studentId;
}
public void setStudentId(int studentId) {
       this.studentId = studentId;
}
public String getStudentName() {
       return studentName;
}
public void setStudentName(String studentName) {
       this.studentName = studentName;
}
public Set<Course> getCourse() {
       return course;
}
public void setCourse(Set<Course> course) {
       this.course = course;
}
@Override
public int hashCode() {
       return Objects.hash(course, studentId, studentName);
```

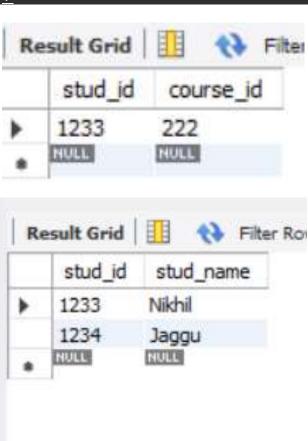
```
@Override
public boolean equals(Object obj) {
    if (this == obj)
        return true;
    if (obj == null)
        return false;
    if (getClass() != obj.getClass())
        return false;
    Student other = (Student) obj;
    return Objects.equals(course, other.course) && studentid == other.studentid
        && Objects.equals(studentName, other.studentName);
}
```

## MAIN

```
package com.manytomanyuniidemo.main;
import java.util.HashSet;
import java.util.Set;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;
import com.manytomanyunidemo.pojo.Course;
```

```
import com.manytomanyunidemo.pojo.Student;
public class AppMain {
public static void main(String args[]) {
Configuration config=new Configuration();
config.configure("hibernate.cfg.xml");
SessionFactory sf=config.buildSessionFactory();
Session s = sf.openSession();
Transaction tr=s.beginTransaction();
Student s1=new Student();
s1.setStudentId(1233);
s1.setStudentName("Nikhil");
Student s2=new Student();
s2.setStudentId(1234);
s2.setStudentName("Jaggu");
Course c1=new Course();
c1.setCourseId(111);
c1.setName("CDAC");
Course c2=new Course();
c2.setCourseId(222);
c2.setName("DBDA");
Set<Course> cset1=new HashSet<>();
cset1.add(c1);
cset1.add(c2);
```

```
Set < Course > cset2 = new HashSet <> ();
cset2.add(c2);
s1.setCourse(cset1);
s1.setCourse(cset2);
s.save(c1);
s.save(c2);
s.save(s2);
s.save(s2);
tr.commit();
System.out.println("Object Save");
sf.close();
}
```



Q2 If the two entity types are 'Customer' and 'Account,' each 'Customer' can have many 'Accounts,' but each 'Account' can only be owned by one 'Customer.

Using this concept Demonstrate One to many bi Directional concept in Hibernate

- OneToManyBi Directional
- 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.cj.jdbc.Driver
</property>
property name="hibernate.connection.url">
jdbc:mysql://localhost:3306/company?useSSL=false
</property>
property
name="hibernate.connection.username">root
```

### Customer

```
package com.onetomanydemo.pojo;
import java.util.Set;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.OneToMany;
import javax.persistence.Table;
@Entity
```

```
@Table(name="customer")
aId
@Column(name="id")
private int CustomerId;
@Column(name="Customer name")
private String CustomerName;
@OneToMany(mappedBy = "customer")
private Set<Account> account;
public Customer() {
public int getCustomerId() {
return CustomerId;
public void setCustomerId(int customerId) {
CustomerId = customerId;
public String getCustomerName() {
return CustomerName;
public void setCustomerName(String customerName) {
CustomerName = customerName;
```

```
public Set<Account> getAccount() {

return account;
}

public void setAccount(Set<Account> account) {

this.account = account;
}
}
```

### Account

```
package com.onetomanydemo.pojo;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.Table;
@Entity
@Table(name="account")
public class Account {
@Id
@Column(name = "id")
```

```
private int AccountId;
@Column(name = "account no")
private String AccountNum;
@ManyToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "cust id" , referencedColumnName =
"id")
private Customer customer;
public Account() {
super();
public int getAccountId() {
return AccountId;
public void setAccountId(int accountId) {
AccountId = accountId;
public String getAccountNum() {
return AccountNum;
public void setAccountNum(String accountNum) {
AccountNum = accountNum;
public Customer getCustomer() {
return customer;
```

```
public void setCustomer(Customer customer) {
this.customer = customer;
}
```

### APP Main

```
package com.onetomanydemo.main;
import java.util.HashSet;
import java.util.Set;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;
import com.onetomanydemo.pojo.Account;
import com.onetomanydemo.pojo.Customer;
public static void main(String[] args) {
Configuration config = new Configuration();
config.configure("hibernate.cfg.xml");
SessionFactory sf = config.buildSessionFactory();
```

```
Session s = sf.openSession();
Transaction tr = s.beginTransaction();
Customer c1 = new Customer();
c1.setCustomerId(1233);
c1.setCustomerName("Sandip");
Account a1 = new Account();
a1.setAccountId(4566);
a1.setAccountNum("ACCSBI400064");
a1.setCustomer(c1);
Account a2 = new Account();
a2.setAccountId(4567);
a2.setAccountNum("ACCHDFC400065");
a2.setCustomer(c1);
Set<Account> aset = new HashSet();
aset.add(a1);
aset.add(a2);
c1.setAccount(aset);
s.save(c1);
s.save(a1);
s.save(a2);
tr.commit();
System.out.println("objects saved in database");
sf.close();
```

```
}
}
```

```
Hibernate: drop table if exists account
Hibernate: drop table if exists customer
Hibernate: drop table if exists customer
Hibernate: drop table if exists customer
Hibernate: create table account (id integer not null, account no warchar(255), cust id integer, primary key (id))
Hibernate: create table customer (id integer not null, Customer name varchar(255), primary key (id))
Hibernate: alter table account add constraint FE d0fnbfyHoclhnhenfrgfhjok foreign key (cust_id) references customer (id)
Hibernate: insert into customer (Customer name, id) values (7, 7)
Hibernate: insert into account (account no, cust_id, id) values (7, 7, 7)
Hibernate: insert into account (account no, cust_id, id) values (7, 7, 7)
chjects saved in database
```

id	account_no	cust id
4566	ACCSBI400064	1233
4567	ACCHDFC400065	1233

