

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>

<property 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>

<property
name="hibernate.connection.password">patil123</proper
ty>
```

```

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

<property
name="hibernate.hbm2ddl.auto">create</property>

<property name="hibernate.show_sql">true</property>

<mapping class="com.manytomanybidemo.pojo.Course"/>

<mapping class="com.manytomanybidemo.pojo.Student"/>

</session-factory>

</hibernate-configuration>

```

- 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 {

    @Id

    @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);
}
}

```

- Course

```
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.ManyToMany;
```

```
import javax.persistence.Table;
```

```
@Entity
```

```
@Table(name="course")
```

```
public class Course {
```

```
    @Id
```

```
    @Column(name="course_id")
```

```
    private int courseId;
```

```
    @Column(name="name")
```

```
    private String name;
```

```
    @ManyToMany
```

```
    private Set<Student> student;
```

```
    public Course() {
```

```
        super();
```

```
    }
```

```
    public int getCourseId() {
```

```
        return courseId;
```

```
    }
```

```
    public void setCourseId(int courseId) {
```

```
        this.courseId = courseId;
```

```

    }

    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(courseId, 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 courseId == other.courseId && 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();
```

```
}
```

```
}
```

Result Grid			Filter Rows:
	stud_id	stud_name	
▶	1	Kshitij	
	2	Pranit	
✱	NULL	NULL	

Result Grid			Filter
	course_id	name	
▶	111	cricket	
	2222	tennis	
✱	NULL	NULL	

Result Grid			Filter Rows:
	stud_id	course_id	
▶	1	111	
	1	2222	
	2	2222	
●	NULL	NULL	

- **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 courseId;
```

```
    @Column(name="name")
```

```
    private String name;
```

```

public Course() {
    super();
}

public int getCourseId() {
    return courseId;
}

public void setCourseId(int courseId) {
    this.courseId = courseId;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

@Override
public int hashCode() {
    return Objects.hash(courseId, 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 courseId == other.courseId && 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.Id;
```

```
import javax.persistence.JoinColumn;
```

```
import javax.persistence.JoinTable;
```

```
import javax.persistence.ManyToMany;
```

```
import javax.persistence.Table;
```

```
@Entity
```

```
@Table(name = "student")
```

```
public class Student {
```

```
    @Id
```

```
    @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(s1);

s.save(s2);

tr.commit();

System.out.println("Object Save");

sf.close();

}

}

```

Result Grid			Filter
	stud_id	course_id	
▶	1233	222	
•	NULL	NULL	

Result Grid			Filter Row
	stud_id	stud_name	
▶	1233	Nikhil	
	1234	Jaggu	
•	NULL	NULL	

Result Grid			Filter Row
	stud_id	stud_name	

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</property>
```

```

<property
name="hibernate.connection.password">patil123</proper
ty>

<property name="hibernate.dialect">

org.hibernate.dialect.MySQLDialect

</property>

<property
name="hibernate.hbm2ddl.auto">create</property>

<property name="hibernate.show_sql">true</property>

<mapping class="com.onetomanydemo.pojo.Customer"/>

<mapping class="com.onetomanydemo.pojo.Account"/>

</session-factory>

</hibernate-configuration>

```

- 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")

public class Customer {

    @Id

    @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 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();

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();
```

```

}
}

```

```

2020-08-28 10:00:00: Successful: alter table account drop foreign key FK_d0fhbryl8oclhah4nfrghjok
2020-08-28 10:00:00: org.hibernate.tool.schemaexport.jdbc:
ERROR: can't drop 'FK_d0fhbryl8oclhah4nfrghjok': check that column/key exists
Hibernate: drop table if exists account
Hibernate: drop table if exists customer
Hibernate: create table account (id integer not null, account_no varchar(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 FK_d0fhbryl8oclhah4nfrghjok foreign key (cust_id) references customer (id)
2020-08-28 10:00:00: org.hibernate.tool.schemaexport.jdbc:
ERROR: Successful: schema export complete
Hibernate: insert into customer (Customer_name, id) values (?, ?)
Hibernate: insert into account (account_no, cust_id, id) values (?, ?, ?)
Hibernate: insert into account (account_no, cust_id, id) values (?, ?, ?)
objects saved in database
2020-08-28 10:00:00: org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionsImpl: stop
2020-08-28 10:00:00: Closing up connection pool [dataSource://localhost:3306/company/oredb001-0100]

```

Result Grid			
	id	account_no	cust_id
▶	4566	ACCSBI400064	1233
	4567	ACCHDFC400065	1233
•	NULL	NULL	NULL

Result Grid		
	id	Customer_name
▶	1233	Sandip
•	NULL	NULL