**Many to Many Mapping-**

Many to many mapping means that multiple rows in a table can be mapped to multiple rows in another table.

|  |  |
| --- | --- |
| User Table | Nominee Table |
| Id(PK) | Id(PK) |
| Name | Name |
| Email |  |

|  |
| --- |
| User\_Nominee table |
| User\_id(FK) |
| Nominee\_id(FK) |

**User.Java**

**package** com.test;

**import** java.util.\*;

**import** javax.persistence.CascadeType;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.FetchType;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.GenerationType;

**import** javax.persistence.Id;

**import** javax.persistence.ManyToMany;

**import** javax.persistence.Table;

@Entity

@Table(name = "user")

**public** **class** User {

@Id

@GeneratedValue(strategy = GenerationType.***IDENTITY***)

@Column(name = "user\_id")

**private** **int** id;

@Column(name = "name")

**private** String name;

@Column(name = "email")

**private** String email;

@ManyToMany(cascade = CascadeType.***ALL***, fetch = FetchType.***EAGER***)

**private** List<Nominee> nomineeList = **new** ArrayList<Nominee>();

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getEmail() {

**return** email;

}

**public** **void** setEmail(String email) {

**this**.email = email;

}

**public** List<Nominee> getNomineeList() {

**return** nomineeList;

}

**public** **void** setNomineeList(List<Nominee> nomineeList) {

**this**.nomineeList = nomineeList;

}

}

**Nominee.Java**

**package** com.test;

**import** java.util.ArrayList;

**import** java.util.List;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.GenerationType;

**import** javax.persistence.Id;

**import** javax.persistence.ManyToMany;

**import** javax.persistence.Table;

@Entity

@Table(name = "nominee")

**public** **class** Nominee {

@Id

@GeneratedValue(strategy = GenerationType.***IDENTITY***)

@Column(name = "nominee\_id")

**private** **int** id;

@Column(name = "name")

**private** String name;

@ManyToMany(mappedBy = "nomineeList")

**private** List<User> userList = **new** ArrayList<User>();

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** List<User> getUserList() {

**return** userList;

}

**public** **void** setUserList(List<User> userList) {

**this**.userList = userList;

}

}

**Test.Java**

**package** com.test;

**import** org.hibernate.Session;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.Transaction;

**import** org.hibernate.cfg.Configuration;

**public** **class** Test {

**public** **static** **void** main(String[] args) {

Configuration configuration = **new** Configuration();

configuration.configure("hibernate.cfg.xml"); //load the file

SessionFactory sessionFactory = configuration.~~buildSessionFactory~~();

Session session=sessionFactory.openSession();

Transaction t= session.beginTransaction();

User user1= **new** User();

user1.setName("mohit");

user1.setEmail("mohit@gmail.com");

User user2= **new** User();

user2.setName("ajay");

user2.setEmail("ajay@gmail.com");

Nominee nominee1= **new** Nominee();

nominee1.setName("Sagar");

Nominee nominee2= **new** Nominee();

nominee2.setName("Santosh");

Nominee nominee3= **new** Nominee();

nominee3.setName("madan");

user1.getNomineeList().add(nominee1);

user1.getNomineeList().add(nominee2);

user1.getNomineeList().add(nominee3);

nominee1.getUserList().add(user1);

nominee2.getUserList().add(user1);

nominee3.getUserList().add(user1);

user2.getNomineeList().add(nominee2);

user2.getNomineeList().add(nominee3);

nominee2.getUserList().add(user2);

nominee3.getUserList().add(user2);

session.persist(user1);

session.persist(user2);

t.commit();

session.close();

sessionFactory.close();

}

}

**Hibernate.cfg.xml**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE hibernate-configuration SYSTEM

"classpath://org/hibernate/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<property name=*"hibernate.connection.driver\_class"*>com.mysql.jdbc.Driver</property>

<property name=*"hibernate.connection.url"*>jdbc:mysql://localhost:3306/sample</property>

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

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

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

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

<property name=*"show\_sql"*>true</property>

<mapping class=*"com.test.User"*></mapping>

<mapping class=*"com.test.Nominee"*></mapping>

</session-factory>

</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>map</groupId>

<artifactId>ManytoManyMapping</artifactId>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>4.3.5.Final</version>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.21</version>

</dependency>

</dependencies>

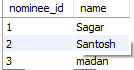
</project>

**Output**

User table



Nominee table



User\_Nominee table

