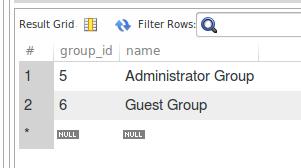
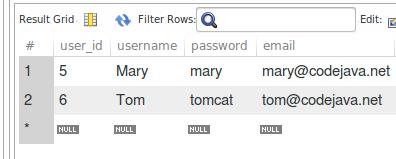


**Output** :

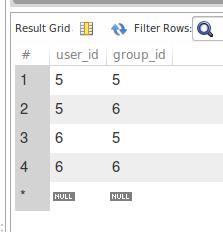
Groups



Users



User\_group



Create database manytomany;

CREATE TABLE `groups` (

`group\_id` int(11) NOT NULL AUTO\_INCREMENT,

`name` varchar(45) NOT NULL,

PRIMARY KEY (`group\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=5 DEFAULT CHARSET=latin1;

CREATE TABLE `users` (

`user\_id` int(11) NOT NULL AUTO\_INCREMENT,

`username` varchar(45) NOT NULL,

`password` varchar(45) NOT NULL,

`email` varchar(45) NOT NULL,

PRIMARY KEY (`user\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=5 DEFAULT CHARSET=latin1;

CREATE TABLE `USERS\_GROUPS` (

`user\_id` int(11) NOT NULL,

`group\_id` int(11) NOT NULL,

PRIMARY KEY (`user\_id`,`group\_id`),

KEY `fk\_user` (`user\_id`),

KEY `fk\_group` (`group\_id`),

CONSTRAINT `fk\_group` FOREIGN KEY (`group\_id`) REFERENCES `groups` (`group\_id`),

CONSTRAINT `fk\_user` FOREIGN KEY (`user\_id`) REFERENCES `users` (`user\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

**Pom file**

<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>net.codejava.hibernate</groupId>

<artifactId>HibernateMany2ManyAnnotationsExample</artifactId>

<version>1.0</version>

<name>HibernateMany2ManyAnnotationsExample</name>

<description>A sample project that demonstrates using JPA annotations to map

a many-to-many association in Hibernate</description>

<dependencies>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>4.0.1.Final</version>

</dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-validator</artifactId>

<version>4.1.0.Final</version>

</dependency>

<dependency>

<groupId>org.hibernate.common</groupId>

<artifactId>hibernate-commons-annotations</artifactId>

<version>4.0.1.Final</version>

<classifier>tests</classifier>

</dependency>

<dependency>

<groupId>org.hibernate.javax.persistence</groupId>

<artifactId>hibernate-jpa-2.0-api</artifactId>

<version>1.0.1.Final</version>

</dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-entitymanager</artifactId>

<version>4.0.1.Final</version>

</dependency>

<dependency>

<groupId>javax.validation</groupId>

<artifactId>validation-api</artifactId>

<version>1.0.0.GA</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>1.6.4</version>

</dependency>

<dependency>

<groupId>org.jboss.logging</groupId>

<artifactId>jboss-logging</artifactId>

<version>3.1.0.CR2</version>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-log4j12</artifactId>

<version>1.6.4</version>

</dependency>

<dependency>

<groupId>mysql</groupId>

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

<version>5.1.10</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.5.1</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

</plugins>

</build>

</project>

**Create User.java**

package net.codejava.hibernate;

import java.util.HashSet;

import java.util.Set;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

import javax.persistence.ManyToMany;

import javax.persistence.Table;

@Entity

@Table(name = "users")

public class User {

private long id;

private String username;

private String password;

private String email;

private Set<Group> groups = new HashSet<Group>();

public User(String username, String password, String email) {

this.username = username;

this.password = password;

this.email = email;

}

public void addGroup(Group group) {

this.groups.add(group);

}

@Id

@GeneratedValue

@Column(name = "USER\_ID")

public long getId() {

return id;

}

public void setId(long id) {

this.id = id;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

@ManyToMany(mappedBy = "users")

public Set<Group> getGroups() {

return groups;

}

public void setGroups(Set<Group> groups) {

this.groups = groups;

}

}

**Group.java**

package net.codejava.hibernate;

import java.util.HashSet;

import java.util.Set;

import javax.persistence.CascadeType;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.JoinTable;

import javax.persistence.ManyToMany;

import javax.persistence.Table;

@Entity

@Table(name = "groups")

public class Group {

private long id;

private String name;

private Set<User> users = new HashSet<User>();

public Group(String name) {

this.name = name;

}

public void addUser(User user) {

this.users.add(user);

}

@Id

@GeneratedValue

@Column(name = "GROUP\_ID")

public long getId() {

return id;

}

public void setId(long id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

@ManyToMany(cascade = CascadeType.ALL)

@JoinTable(

name = "USERS\_GROUPS",

joinColumns = @JoinColumn(name = "GROUP\_ID"),

inverseJoinColumns = @JoinColumn(name = "USER\_ID")

)

public Set<User> getUsers() {

return users;

}

public void setUsers(Set<User> users) {

this.users = users;

}

}

**Create UserManager class to insert manyToMany relationship data**

package net.codejava.hibernate;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.Configuration;

import org.hibernate.service.ServiceRegistry;

import org.hibernate.service.ServiceRegistryBuilder;

/\*\*

\* A program that demonstrates using JPA annotations to map

\* a bidirectional many-to-many association in Hibernate framework.

\*

\*/

public class UsersManager {

public static void main(String[] args) {

// loads configuration and mappings

Configuration configuration = new Configuration().configure();

ServiceRegistryBuilder registry = new ServiceRegistryBuilder();

registry.applySettings(configuration.getProperties());

ServiceRegistry serviceRegistry = registry.buildServiceRegistry();

// builds a session factory from the service registry

SessionFactory sessionFactory = configuration

.buildSessionFactory(serviceRegistry);

// obtains the session

Session session = sessionFactory.openSession();

session.beginTransaction();

Group groupAdmin = new Group("Administrator Group");

Group groupGuest = new Group("Guest Group");

User user1 = new User("Tom", "tomcat", "tom@codejava.net");

User user2 = new User("Mary", "mary", "mary@codejava.net");

groupAdmin.addUser(user1);

groupAdmin.addUser(user2);

groupGuest.addUser(user1);

groupGuest.addUser(user2);

user1.addGroup(groupAdmin);

user1.addGroup(groupGuest);

user2.addGroup(groupGuest);

user2.addGroup(groupAdmin);

session.save(groupAdmin);

session.save(groupGuest);

session.save(user1);

session.save(user2);

session.getTransaction().commit();

session.close();

}

}

**Hibernate.cfg.xml file**

<?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>

<!-- Database connection settings -->

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

<property name="connection.url">jdbc:mysql://localhost:3306/manytomany</property>

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

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

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

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

<mapping class="net.codejava.hibernate.User"/>

<mapping class="net.codejava.hibernate.Group"/>

</session-factory>

</hibernate-configuration>

**Summary :**

**1.In Users class**

@ManyToMany(mappedBy = "users")

public Set<Group> getGroups() {

return groups;

}

**2.In Group class**

@ManyToMany(cascade = CascadeType.ALL)

@JoinTable(

name = "USERS\_GROUPS",

joinColumns = @JoinColumn(name = "GROUP\_ID"),

inverseJoinColumns = @JoinColumn(name = "USER\_ID")

)

public Set<User> getUsers() {

return users;

}

//Group’s table’s GROUP\_ID goes to Users\_group’s table’s GROUP\_ID.

//User’s table’s USER\_ID goes to Users\_group’s table’s USER\_ID