**Many to one Mapping-**

Many to one mapping means that multiple rows in a table can be mapped to Single row in another table.

|  |  |
| --- | --- |
| Employee Table | address Table |
| Id | Addresseid |
| Name | City |
| Email | Country |
| addressid | State |
|  | Pincode |

**Employee. Java**

**package** com.demo;

**import** javax.persistence.CascadeType;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.GenerationType;

**import** javax.persistence.Id;

**import** javax.persistence.ManyToOne;

**import** javax.persistence.Table;

@Entity

@Table(name = "employee")

**public** **class** Employee {

@Id

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

@Column(name = "id")

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

@Column(name = "name")

**private** String name;

@Column(name = "email")

**private** String email;

@ManyToOne(cascade = CascadeType.***ALL***)

**private** Address address;

**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** Address getAddress() {

**return** address;

}

**public** **void** setAddress(Address address) {

**this**.address = address;

}

@Override

**public** String toString() {

**return** "Employee [id=" + id + ", name=" + name + ", email=" + email + ", address=" + address + "]";

}

}

**Address.Java**

**package** com.demo;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.GenerationType;

**import** javax.persistence.Id;

**import** javax.persistence.Table;

@Entity

@Table(name = "address")

**public** **class** Address {

@Id

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

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

@Column(name = "city")

**private** String city;

@Column(name = "state")

**private** String state;

@Column(name = "country")

**private** String country;

@Column(name = "pincode")

**private** String pinCode;

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

**return** id;

}

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

**this**.id = id;

}

**public** String getCity() {

**return** city;

}

**public** **void** setCity(String city) {

**this**.city = city;

}

**public** String getState() {

**return** state;

}

**public** **void** setState(String state) {

**this**.state = state;

}

**public** String getCountry() {

**return** country;

}

**public** **void** setCountry(String country) {

**this**.country = country;

}

**public** String getPinCode() {

**return** pinCode;

}

**public** **void** setPinCode(String pinCode) {

**this**.pinCode = pinCode;

}

@Override

**public** String toString() {

**return** "Address [id=" + id + ", city=" + city + ", state=" + state + ", country=" + country + ", pinCode="

+ pinCode + "]";

}

}

**Test.Java**

**package** com.demo;

**import** java.util.List;

**import** org.hibernate.Query;

**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) {

Employee e1 = **new** Employee();

e1.setName("ram");

e1.setEmail("ram@gmail.com");

Employee e2 = **new** Employee();

e2.setName("sachin");

e2.setEmail("sachin@gmail.com");

Address address1 = **new** Address();

address1.setCity("Pune");

address1.setState("Maharashtra");

address1.setCountry("India");

address1.setPinCode("411035");

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

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

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

Session session = sessionFactory.openSession();

Transaction transaction = session.beginTransaction();

e1.setAddress(address1);

e2.setAddress(address1);

session.save(e1);

session.save(e2);

transaction.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/demo</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.Address"*></mapping>

<mapping class=*"com.test.Employee"*></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>OneToOneMapping</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-**

Address table



Employee table

