Hibernate Mapping Relation

1. Department 🡪 multiple employee’s(one-many)
2. Employee’s 🡪 department(many-one)
3. Me 🡪 My laptop(one-one)
4. Lib books 🡪 students(many-many)

@entity @entity

Public class laptop Public class student

{ {

@Id @Id

Private int Lid, Private int rollno;

Private string Lname; Private string name;

} Private Laptop Laptop;

laptop @one to one }

|  |  |
| --- | --- |
| lid | lname |
| 101 | Dell |
| 102 | Hp |

student

|  |  |  |  |
| --- | --- | --- | --- |
| roll | name | marks | lid |
| 1 | maddy | 50 | 101 |
| lid | name | Roll |  |
| 101 | Dell |  |  |
| 102 | HP |  |  |

Student

|  |  |
| --- | --- |
| roll | lid |
| 1 | 101 |
| 1 | 102 |

Student\_laptop

@entity @entity

Public class laptop Public class student

{ {

@Id @Id

Private int Lid, Private int rollno;

Private string Lname; Private string name;

@many to one Private list<Laptop> Laptop;

Private Student “stud”; @one to many }

} <mapped by “stud”>

**Many -many** (if mapped by is not use 4 tables are created otherwise 1 table is created)

Private list<student> studs; Private list<lap>laptop;

|  |
| --- |
|  |
|  | App.Java |
|  |  |
|  | package com.telusko.DemoHib; |
|  |  |
|  |  |
|  | import org.hibernate.Session; |
|  | import org.hibernate.Transaction; |
|  | import org.hibernate.SessionFactory; |
|  | import org.hibernate.cfg.Configuration; |
|  | import org.hibernate.service.ServiceRegistry; |
|  | import org.hibernate.service.ServiceRegistryBuilder; |
|  |  |
|  | /\*\* |
|  | \* Hello world! |
|  | \* |
|  | \*/ |
|  | public class App |
|  | { |
|  | public static void main( String[] args ) |
|  | { |
|  | Laptop laptop = new Laptop(); |
|  | laptop.setLid(101); |
|  | laptop.setLname("Dell"); |
|  |  |
|  | Student s = new Student(); |
|  | s.setName("Navin"); |
|  | s.setRollno(1); |
|  | s.setMarks(50); |
|  | s.getLaptop().add(laptop); |
|  |  |
|  | laptop.getStudent().add(s); |
|  |  |
|  | Configuration config = new Configuration().configure().addAnnotatedClass(Student.class).addAnnotatedClass(Laptop.class); |
|  | ServiceRegistry registry = new ServiceRegistryBuilder().applySettings(config.getProperties()).buildServiceRegistry(); |
|  | SessionFactory sf = config.buildSessionFactory(registry); |
|  | Session session = sf.openSession(); |
|  |  |
|  | session.beginTransaction(); |
|  |  |
|  | session.save(laptop); |
|  | session.save(s); |
|  |  |
|  | session.getTransaction().commit(); |
|  | } |
|  | } |
|  |  |
|  |  |
|  | -------------------------------------------------------------------------------- |
|  | Laptop.java |
|  | -------------------------------------------------------------------------------- |
|  |  |
|  | package com.telusko.DemoHib; |
|  |  |
|  | import java.util.ArrayList; |
|  | import java.util.List; |
|  |  |
|  | import javax.persistence.Entity; |
|  | import javax.persistence.Id; |
|  | import javax.persistence.ManyToMany; |
|  | import javax.persistence.ManyToOne; |
|  |  |
|  | @Entity |
|  | public class Laptop |
|  | { |
|  | @Id |
|  | private int lid; |
|  | private String lname; |
|  | @ManyToMany |
|  | private List<Student> student = new ArrayList<Student>(); |
|  |  |
|  | public int getLid() { |
|  | return lid; |
|  | } |
|  | public void setLid(int lid) { |
|  | this.lid = lid; |
|  | } |
|  | public String getLname() { |
|  | return lname; |
|  | } |
|  | public void setLname(String lname) { |
|  | this.lname = lname; |
|  | } |
|  | public List<Student> getStudent() { |
|  | return student; |
|  | } |
|  | public void setStudent(List<Student> student) { |
|  | this.student = student; |
|  | } |
|  |  |
|  |  |
|  | } |
|  |  |
|  |  |
|  | -------------------------------------------------------------------------------- |
|  | Student.java |
|  | -------------------------------------------------------------------------------- |
|  |  |
|  | package com.telusko.DemoHib; |
|  |  |
|  | import java.util.ArrayList; |
|  | import java.util.List; |
|  |  |
|  | import javax.persistence.Entity; |
|  | import javax.persistence.Id; |
|  | import javax.persistence.ManyToMany; |
|  | import javax.persistence.OneToMany; |
|  | import javax.persistence.OneToOne; |
|  |  |
|  | @Entity |
|  | public class Student |
|  | { |
|  | @Id |
|  | private int rollno; |
|  | private String name; |
|  | private int marks; |
|  | @ManyToMany(mappedBy = "student") |
|  | private List<Laptop> laptop = new ArrayList<Laptop>(); |
|  |  |
|  | public int getRollno() { |
|  | return rollno; |
|  | } |
|  | public void setRollno(int rollno) { |
|  | this.rollno = rollno; |
|  | } |
|  | public String getName() { |
|  | return name; |
|  | } |
|  | public void setName(String name) { |
|  | this.name = name; |
|  | } |
|  | public int getMarks() { |
|  | return marks; |
|  | } |
|  | public void setMarks(int marks) { |
|  | this.marks = marks; |
|  | } |
|  | public List<Laptop> getLaptop() { |
|  | return laptop; |
|  | } |
|  | public void setLaptop(List<Laptop> laptop) { |
|  | this.laptop = laptop; |
|  | } |
|  |  |
|  | @Override |
|  | public String toString() { |
|  | return "Student [rollno=" + rollno + ", name=" + name + ", marks=" + marks + "]"; |
|  | } |
|  |  |
|  |  |
|  | } |
|  |  |
|  | -------------------------------------------------------------------------------- |
|  | 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.jdbc.Driver</property> |
|  | <property name="hibernate.connection.password">1234</property> |
|  | <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/neon</property> |
|  | <property name="hibernate.connection.username">root</property> |
|  | <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property> |
|  | <property name="hbm2ddl.auto">create</property> |
|  | <property name="show\_sql">true</property> |
|  | </session-factory> |
|  | </hibernate-configuration> |



