

```
1 import model.Department;
2 import model.Teacher;
3 import org.hibernate.Session;
4 import org.hibernate.SessionFactory;
5 import org.hibernate.Transaction;
6 import org.hibernate.cfg.Configuration;
7
8 import java.util.ArrayList;
9
10 public class App {
11     public static void main(String[] args) {
12         //      manyToOne();
13         oneToMany();
14     }
15
16
17     public static void manyToOne() {
18         SessionFactory factory = new Configuration().
19         configure().buildSessionFactory();
20         Session session = factory.openSession();
21         Transaction transaction = session.
22         beginTransaction();
23
24         Department dept1 = new Department("IT");
25         Department dept2 = new Department("HR");
26
27         Teacher t1 = new Teacher("1000", "MHaseeb",
28         dept1);
29         Teacher t2 = new Teacher("2220", "Shahparan"
30         , dept1);
31         Teacher t3 = new Teacher("3000", "James",
32         dept1);
33         Teacher t4 = new Teacher("40000", "Joseph",
34         dept2);
35
36         session.persist(dept1);
37         session.persist(dept2);
38
39         session.persist(t1);
40         session.persist(t2);
41         session.persist(t3);
```

```
36         session.persist(t4);
37         transaction.commit();
38     }
39
40     public static void oneToMany() {
41         SessionFactory factory = new Configuration().
configure().buildSessionFactory();
42         Session session = factory.openSession();
43         Transaction t = session.beginTransaction();
44
45         Teacher t1 = new Teacher("1000", "MHaseeb");
46         Teacher t2 = new Teacher("2220", "Shahparan"
);
47         Teacher t3 = new Teacher("3000", "James");
48         Teacher t4 = new Teacher("40000", "Joseph");
49         Teacher t5 = new Teacher("200", "Ali");
50
51         ArrayList<Teacher> teacherArrayList = new
ArrayList<>();
52         teacherArrayList.add(t1);
53         teacherArrayList.add(t2);
54         teacherArrayList.add(t3);
55         teacherArrayList.add(t4);
56         teacherArrayList.add(t5);
57
58         session.persist(t1);
59         session.persist(t2);
60         session.persist(t3);
61         session.persist(t4);
62         session.persist(t5);
63
64         Department department = new Department();
65         department.setDeptName("Development");
66         department.setTeacherList(teacherArrayList);
67
68         session.persist(department);
69         t.commit();
70     }
71 }
72
```

```
1 package model;
2
3 import java.io.Serial;
4 import java.io.Serializable;
5
6 import jakarta.persistence.*;
7
8 @Entity
9 @Table
10 public class Teacher implements Serializable {
11     @Serial
12     private static final long serialVersionUID = 1L;
13     @Id
14     @GeneratedValue(strategy = GenerationType.
        IDENTITY)
15     private int teacherId;
16     private String salary;
17     private String TeacherName;
18     // @ManyToOne
19     // private Department department;
20
21     public Teacher(int teacherId, String salary,
        String teacherName) {
22         super();
23         this.teacherId = teacherId;
24         this.salary = salary;
25         TeacherName = teacherName;
26     }
27
28     public Teacher() {
29     }
30
31     public Teacher(String salary, String teacherName
        , Department department){
32         this.salary=salary;
33         this.TeacherName=teacherName;
34     //     this.department=department;
35     }
36     public Teacher(String salary, String teacherName
        ){
37         this.salary=salary;
```

```
38         this.TeacherName=teacherName;
39     }
40
41     //    public Department getDepartment(){
42     //        return department;
43     //    }
44     public int getTeacherId(){
45         return teacherId;
46     }
47     public void setTeacherId(int teacherId) {
48         this.teacherId = teacherId;
49     }
50     public String getSalary() {
51         return salary;
52     }
53     public void setSalary(String salary) {
54         this.salary = salary;
55     }
56     public String getTeacherName() {
57         return TeacherName;
58     }
59     public void setTeacherName(String teacherName) {
60         TeacherName = teacherName; }
61
62
63 }
64
```

```
1 package model;
2 import jakarta.persistence.*;
3 import java.io.Serial;
4 import java.io.Serializable;
5 import java.util.List;
6
7 @Entity
8 @Table
9 public class Department implements Serializable {
10     @Serial
11     private static final long serialVersionUID=1L;
12
13     @Id
14     @GeneratedValue(strategy = GenerationType.
IDENTITY)
15     private int deptId;
16     private String deptName;
17     @OneToMany(targetEntity= Teacher.class, cascade
= {CascadeType.ALL})
18     private List<Teacher> teacherList;
19     public Department(int deptId, String deptName){
20         super();
21         this.deptId=deptId;
22         this.deptName=deptName;
23     }
24
25     public List<Teacher> getTeacherList() {
26         return teacherList;
27     }
28     public void setTeacherList(List<Teacher>
teacherList) {
29         this.teacherList = teacherList;
30     }
31
32
33     public Department(){}
34
35     public Department(String deptName){
36         this.deptName=deptName;
37     }
38
```

```
39     public int getDeptId(){
40         return deptId;
41     }
42
43     public void setDeptId(int deptId) {
44         this.deptId = deptId;
45     }
46
47     public String getDeptName() {
48         return deptName;
49     }
50
51     public void setDeptName(String deptName) {
52         this.deptName = deptName;
53     }
54 }
55
```

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE hibernate-configuration PUBLIC
3     "-//Hibernate/Hibernate Configuration DTD 3.0
4     //EN"
5     "http://www.hibernate.org/dtd/hibernate-
6     configuration-3.0.dtd">
7 <hibernate-configuration>
8     <session-factory>
9         <!-- Drop and re-create the database on
10         startup -->
11         <property name="hibernate.hbm2ddl.auto">
12             create-drop </property>
13
14         <!-- Database connection settings -->
15         <property name="connection.driver_class">com.
16             mysql.cj.jdbc.Driver</property>
17         <property name="connection.url">jdbc:mysql://
18             localhost:3306/usersDb</property>
19         <property name="connection.username">root</
20             property>
21         <property name="connection.password">
22             SzhengSQL123</property>
23
24         <!-- MySQL DB dialect -->
25         <property name="dialect">org.hibernate.
26             dialect.MySQLDialect</property>
27         <!-- print all executed SQL on console -->
28         <property name="hibernate.show_sql" >true </
29             property>
30         <property name="hibernate.format_sql" >true
31             </property>
32
33         <!--      &lt;!&ndash; Mapping entity file -->
34         <mapping class="model.Teacher"/>
35         <mapping class="model.Department"/>
36     </session-factory>
37 </hibernate-configuration>
38

```