

Name:- Shubham Pawar

Course:- Java Full Stack Development

Ref. No. SCS/CG/2021/025

Program name :- One To One Mapping Program

1)Professor.class

```
packagecom.demo;
```

```
importjavax.persistence.CascadeType;
importjavax.persistence.Column;
importjavax.persistence.Entity;
importjavax.persistence.GeneratedValue;
importjavax.persistence.Id;
importjavax.persistence.OneToOne;
```

```
@Entity
```

```
public class Professor {
```

```
@Id
```

```
@GeneratedValue
```

```
privateint id;
```

```
@Column(name = "Professor_Name")
```

```
private String name;
```

```
@OneToOne(targetEntity = Subject.class,cascade = CascadeType.ALL,orphanRemoval = true)
```

```
private Subject subject;
```

```
public Subject getSubject() {
```

```
return subject;
```

```
}
```

```
public void setSubject(Subject subject) {
```

```
this.subject = subject;
```

```
}
```

```
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;  
}  
}
```

2)Subject.class

```
package com.demo;
```

```
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import javax.persistence.Id;
```

```
@Entity
```

```
public class Subject {
```

```
    @Id
```

```
    @GeneratedValue
```

```
    private int id;
```

```
    private String subjectName;
```

```

public int getId() {
    return id;
}

public void setId(int id) {
    this.id = id;
}

public String getSubjectName() {
    return subjectName;
}

public void setSubjectName(String subjectName) {
    this.subjectName = subjectName;
}
}

```

3) School.main.class

```

package com.demo;

import org.hibernate.Session;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;

public class School {
    public static void main(String[] args) {
        Configuration cfg = new Configuration().configure("hibernate.cfg.xml");
        Session session = cfg.buildSessionFactory().openSession();
        Transaction txn = session.beginTransaction();
        Professor p = new Professor();
    }
}

```

```

p.setName("sangharsh");
Subject s=new Subject();
s.setSubjectName("java");
p.setSubject(s);
session.save(p);
session.save(s);
txn.commit();
System.out.println("success");
session.close();
}
}

```

4)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.dialect">org.hibernate.dialect.MySQLDialect</property>

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

<property name = "hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>

<!-- Assume test is the database name -->

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

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

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

```

```
<property name="show_sql">True</property>
```

```
<!-- List of XML mapping files -->
```

```
<mapping class="com.demo.Professor"/>
```

```
<mapping class="com.demo.Subject"/>
```

```
</session-factory>
```

```
</hibernate-configuration>
```

Output

```
Hibernate: alter table Professor drop foreign key FKjf8acrisl3meneiuinu54jyjfj
Hibernate: drop table if exists hibernate_sequence
Hibernate: drop table if exists Professor
Hibernate: drop table if exists Subject
Hibernate: create table hibernate_sequence (next_val bigint) engine=InnoDB
Hibernate: insert into hibernate_sequence values ( 1 )
Hibernate: insert into hibernate_sequence values ( 1 )
Hibernate: create table Professor (id integer not null,
Professor_Name varchar(255), subject_id integer, primary key (id))
engine=InnoDB
Hibernate: create table Subject (id integer not null,
subject_Name varchar(255), primary key (id)) engine=InnoDB
Hibernate: alter table Professor add constraint FKjf8acrisl3meneiuinu54jyjfj
foreign key (subject_id) references Subject (id)
Hibernate: select next_val as id_val from hibernate_sequence for update
Hibernate: update hibernate_sequence set next_val= ? where next_val=?
Hibernate: select next_val as id_val from hibernate_sequence for update
Hibernate: update hibernate_sequence set next_val= ? where next_val=?
Hibernate: insert into Subject (subject_Name, id) values (?, ?)
Hibernate: insert into Professor (Professor_Name, subject_id, id) values
(?, ?, ?)
success
```