One to One mapping-

One object is associated with one object only.

The primary key of one table that becomes the foreign key of another table called as one to one mapping.

| Customer Table | Transection Table |
|---|---|
| Id(PK) | Transection id(pk) |
| Name | Date |
| Email | Total |
| address | |
| Transection id(fk) | |

Customer.java

```
package com.test;


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.OneToOne;
import javax.persistence.Table;


@Entity
@Table(name = "customer")
public class Customer {
```

```
@Id
@GeneratedValue(strategy=GenerationType.IDENTITY)
private int id;
@Column(name="firstName")
private String firstName;
@Column(name="mobileNumber")
private String mobileNumber;

@OneToOne(targetEntity=Transection.class,cascade=CascadeType.ALL)
private Transection transection;

public int getId() {
        return id;
}

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

public String getFirstName() {
        return firstName;
}

public void setFirstName(String firstName) {
        this.firstName = firstName;
```

```
	}

	public String getMobileNumber() {
		return mobileNumber;
	}

	public void setMobileNumber(String mobileNumber) {
		this.mobileNumber = mobileNumber;
	}

	public Transection getTransection() {
		return transection;
	}

	public void setTransection(Transection transection) {
		this.transection = transection;
	}

}
```

Transection.java

```
package com.test;

import java.util.Date;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.OneToOne;
import javax.persistence.Table;

@Entity
@Table(name = "Transection")
public class Transection {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;

    @Column(name = "date")
    private Date date;

    @Column(name = "total")
    private int total;

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

    public Date getDate() {
        return date;
    }

    public void setDate(Date date) {
        this.date = date;
    }

    public int getTotal() {
        return total;
    }

    public void setTotal(int total) {
        this.total = total;
    }
}
```

MainTest.Java

```
package com.test;

import java.util.Date;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;

public class MainTest {

	public static void main(String[] args) {

		Configuration configuration = new Configuration();
		configuration.configure("hibernate.cfg.xml");

		SessionFactory sessionFactory = configuration.buildSessionFactory();
		Session session=sessionFactory.openSession();

		Transaction t= session.beginTransaction();

		Customer customer=new Customer();
		customer.setFirstName("sohan");
		customer.setMobileNumber("7575858595");

		Transection transection= new Transection();
		transection.setDate(new Date());
		transection.setTotal(5000);
```

```
                customer.setTransection(transection);


                session.save(customer);

                t.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/sys</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.Customer"></mapping>
        <mapping
class="com.test.Transection"></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>
```