* Hibernate One to One Mapping is create using @onetoonemapping .
* Data is persisted across tables using a foreign key.
* Mapping is created with @OneToOne annotation and using @joincolumn with unique=true on owner side
* On the inverse end mappedBy is used to refer to attribute on owner side

|  |
| --- |
| Implementation Details |

import javax.persistence.CascadeType;

import javax.persistence.Entity;

import javax.persistence.FetchType;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.OneToOne;

@Entity

public class **Customer** {

@Id

@GeneratedValue(strategy=GenerationType.AUTO)

private Long id;

private String name;

public String getName()

{

return name;

}

public void setName(String name) {

this.name = name;

}

**@OneToOne(cascade={CascadeType.PERSIST},fetch=FetchType.LAZY)**

**@JoinColumn(name="passport\_id", unique=true)**

private Passport passport;

public Customer() {}

public Customer(String name, Passport passport) {

this.name = name;

this.passport = passport;

}

public Passport getPassport() {

return passport;

}

@Override

public String toString() {

StringBuilder builder = new StringBuilder();

builder.append("Customer [name=");

builder.append(name);

builder.append("]");

return builder.toString();

}

}

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.OneToOne;

@Entity

public class **Passport** {

@Id

@GeneratedValue(strategy=GenerationType.AUTO)

private Long id;

@Column(name="passport\_number")

private String passportNumber;

**@OneToOne(mappedBy="passport")**

**private Customer customer;**

public Passport() {}

public Passport(String passportNumber) {

this.passportNumber = passportNumber;

}

public Customer getCustomer() {

return customer;

}

}

EntityManager em = emf.createEntityManager();

em.getTransaction().begin();

Passport passport = **new** Passport("234234234234");

Customer customer = **new** Customer("Morgan Scott", passport);

em.persist(customer);

em.getTransaction().commit();

em.close();

|  |
| --- |
| Query |

EntityManager em = emf.createEntityManager();

em.getTransaction().begin();

Customer customer = em.find(Customer.**class**, 2L);

System.***out***.println(customer);

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Using JPQL\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

/\* JPQL \*/

Query query = em.createQuery("SELECT c FROM Customer c");

List<Customer> customerList = query.getResultList();

**for**(Customer cust : customerList)

{

System.***out***.println(cust);

}

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Using Native Query\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

em.getTransaction().commit();

em.close();

|  |
| --- |
| Pointer for Better Performance |