ONETOMANY AND MANY TO ONE:

EX: ONE CART CAN HAVE MANY ITEMS(@OneToMany)

MANY ITEMS CAN BE IN ONE CART (@ManyToOne)

Primary class:

@Table(name = "cart")

public class Cart {

@Id

@Column

private int cid;

@Column

private String cname;

// @Exclude

// @OneToMany(cascade = CascadeType.ALL , mappedBy = "cart")

// private List<Item> items ;

}

SECONDARY CLASS:

@Table( name="item\_details")

public class Item {

@Id

@Column

private int id;

@Column

private String iname;

@Exclude

@ManyToOne(cascade = CascadeType.ALL)

@JoinColumn( name = "cid")

private Cart cart;

}

ONE WAY BINDING:

public class OneWay {

public static void main(String[] args) {

EntityManagerFactory entityManagerFactory = null;

EntityManager entityManager = null;

EntityTransaction entityTransaction = null;

Item item = new Item();

item.setId(10);

item.setIname("Redmi");

Cart cart = new Cart();

cart.setCid(20);

cart.setCname("Second");

item.setCart(cart);

try {

entityManagerFactory = Persistence.createEntityManagerFactory("mapping");

entityManager = entityManagerFactory.createEntityManager();

entityTransaction = entityManager.getTransaction();

entityTransaction.begin();

entityManager.persist(item);

entityTransaction.commit();

} catch (Exception e) {

e.printStackTrace();

}

}

}

TWO WAY :

TO WRITE THE TWO WAY FIRST UNCOMMENT THE CART CLASS COMMENTED LINES:

public class TwoWay {

public static void main(String[] args) {

EntityManagerFactory entityManagerFactory = null;

EntityManager entityManager = null;

EntityTransaction entityTransaction = null;

Item item = new Item();

item.setId(30);

item.setIname("Book");

Item item2 = new Item();

item2.setId(40);

item2.setIname("Pen");

Cart cart = new Cart();

cart.setCid(30);

cart.setCname("Third");

item.setCart(cart);

item2.setCart(cart);

ArrayList<Item> arrayList = new ArrayList<Item>();

arrayList.add(item);

arrayList.add(item2);

cart.setItems(arrayList);

try {

entityManagerFactory = Persistence.createEntityManagerFactory("mapping");

entityManager = entityManagerFactory.createEntityManager();

entityTransaction = entityManager.getTransaction();

entityTransaction.begin();

entityManager.persist(cart);

//TO ADD ITEMS TO AN ALREADY EXISTING CART

// Cart cart2 = entityManager.find(Cart.class, 30);

// Item item4 = new Item();

// item4.setId(90);

// item4.setIname("PP");

// Item item5 = new Item();

// item5.setId(100);

// item5.setIname("KK");

// item4.setCart(cart2);

// item5.setCart(cart2);

// List<Item> items = cart2.getItems();

// items.add(item5);

// items.add(item4);

// cart2.setItems(items);

entityTransaction.commit();

//TO GET ALL ITEM\_DETAILS PRESENT IN ONE CART USING CART

// Cart cart2 = entityManager.find(Cart.class, 30);

// System.out.println(cart2.getItems());

// Item item3 = entityManager.find(Item.class, 90);

// System.out.println(item3.getIname());

//TO GET ALL ITEM\_DETAILS PRESENT INSIDE ONE CART USING ITEMS

// System.out.println(item3.getCart().getItems());

} catch (Exception e) {

e.printStackTrace();

}

}

}