# **One To Many**

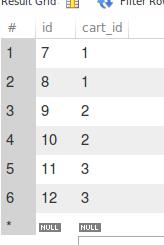
# Cart may contains many items



Cart Table



Items Table



**Create following tables**

CREATE TABLE `Cart` (

`cart\_id` int(11) unsigned NOT NULL AUTO\_INCREMENT,

PRIMARY KEY (`cart\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=5 DEFAULT CHARSET=utf8;

CREATE TABLE `Items` (

`id` int(11) unsigned NOT NULL AUTO\_INCREMENT,

`cart\_id` int(11) unsigned NOT NULL,

PRIMARY KEY (`id`),

KEY `cart\_id` (`cart\_id`),

CONSTRAINT `items\_ibfk\_1` FOREIGN KEY (`cart\_id`) REFERENCES `Cart` (`cart\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=7 DEFAULT CHARSET=utf8;

**Create pom.xml file**

<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>onetoone</groupId>

<artifactId>onetoone</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>war</packaging>

<dependencies>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>4.0.1.Final</version>

</dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-validator</artifactId>

<version>4.2.0.Final</version>

</dependency>

<dependency>

<groupId>org.hibernate.common</groupId>

<artifactId>hibernate-commons-annotations</artifactId>

<version>4.0.1.Final</version>

<classifier>tests</classifier>

</dependency>

<dependency>

<groupId>org.hibernate.javax.persistence</groupId>

<artifactId>hibernate-jpa-2.0-api</artifactId>

<version>1.0.1.Final</version>

</dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-entitymanager</artifactId>

<version>4.1.8.Final</version>

</dependency>

<dependency>

<groupId>javax.validation</groupId>

<artifactId>validation-api</artifactId>

<version>1.0.0.GA</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>1.6.4</version>

</dependency>

<dependency>

<groupId>org.jboss.logging</groupId>

<artifactId>jboss-logging</artifactId>

<version>3.1.0.CR2</version>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-log4j12</artifactId>

<version>1.6.4</version>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>5.1.10</version>

</dependency>

</dependencies>

<build>

<sourceDirectory>src</sourceDirectory>

<plugins>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.5.1</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

<plugin>

<artifactId>maven-war-plugin</artifactId>

<version>3.0.0</version>

<configuration>

<warSourceDirectory>WebContent</warSourceDirectory>

</configuration>

</plugin>

</plugins>

</build>

</project>

**Hibernate.cfg.xml**

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 3.0//EN"

"http://hibernate.org/dtd/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/manytoone</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="hibernate.current\_session\_context\_class">thread</property>

<property name="hibernate.show\_sql">true</property>

<mapping class="com.baeldung.hibernate.oneToMany.model.Cart"/>

<mapping class="com.baeldung.hibernate.oneToMany.model.Items"/>

</session-factory>

</hibernate-configuration>

**Note :**

<property name="hibernate.current\_session\_context\_class">thread</property>

Above property is useful because you can later call **sessionFactory.getCurrentSession()** which will return the session that is bound to the currently running thread. **By default it is enabled**.

**Create Entity Classes**

**Cart.java**

package com.baeldung.hibernate.oneToMany.model;

import java.util.Set;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.OneToMany;

import javax.persistence.Table;

@Entity

@Table(name = "Cart")

public class Cart {

@Id

@GeneratedValue

@Column(name = "cart\_id")

private long id;

@OneToMany(mappedBy = "cart")

private Set<Items> items;

public long getId() {

return id;

}

public void setId(long id) {

this.id = id;

}

public Set<Items> getItems() {

return items;

}

public void setItems(Set<Items> items) {

this.items = items;

}

}

**Items.java**

package com.baeldung.hibernate.oneToMany.model;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.ManyToOne;

import javax.persistence.Table;

@Entity

@Table(name = "Items")

public class Items {

@Id

@GeneratedValue

@Column(name = "id")

private long id;

@ManyToOne

@JoinColumn(name = "cart\_id", nullable = false)

private Cart cart;

public Items() {

}

public Items(Cart c) {

this.cart = c;

}

public Cart getCart() {

return cart;

}

public void setCart(Cart cart) {

this.cart = cart;

}

public long getId() {

return id;

}

public void setId(long id) {

this.id = id;

}

}

**HibernateAnnotationUtil class provide SessionFactory**

package com.baeldung.hibernate.oneToMany.config;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.Configuration;

import org.hibernate.service.ServiceRegistry;

import org.hibernate.service.ServiceRegistryBuilder;

public class HibernateAnnotationUtil {

private static SessionFactory sessionFactory;

private static ServiceRegistry serviceRegistry;

static Configuration configuration;

private static SessionFactory buildSessionFactory() {

try {

configuration = new Configuration();

configuration.configure();

serviceRegistry = new ServiceRegistryBuilder().applySettings(

configuration.getProperties()).buildServiceRegistry();

sessionFactory = configuration.buildSessionFactory(serviceRegistry);

return sessionFactory;

} catch (Throwable ex) {

System.err.println("Failed to create sessionFactory object." + ex);

throw new ExceptionInInitializerError(ex);

}

}

public static SessionFactory getSessionFactory() {

if (sessionFactory == null)

sessionFactory = buildSessionFactory();

return sessionFactory;

}

}

**Class with Main method to run**

package com.baeldung.hibernate.oneToMany.main;

import java.util.HashSet;

import java.util.Set;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.Transaction;

import com.baeldung.hibernate.oneToMany.config.HibernateAnnotationUtil;

import com.baeldung.hibernate.oneToMany.model.Cart;

import com.baeldung.hibernate.oneToMany.model.Items;

public class HibernateOneToManyAnnotationMain {

public static void main(String[] args) {

Cart cart = new Cart();

Items item1 = new Items(cart);

Items item2 = new Items(cart);

Set<Items> itemsSet = new HashSet<Items>();

itemsSet.add(item1);

itemsSet.add(item2);

cart.setItems(itemsSet);

SessionFactory sessionFactory = null;

Session session = null;

Transaction tx = null;

try {

// Get Session

sessionFactory = HibernateAnnotationUtil.getSessionFactory();

session = sessionFactory.getCurrentSession();

System.out.println("Session created");

// start transaction

tx = session.beginTransaction();

// Save the Model object

session.save(cart);

session.save(item1);

session.save(item2);

// Commit transaction

tx.commit();

System.out.println("Cart ID=" + cart.getId());

System.out.println("item1 ID=" + item1.getId() + ", Foreign Key Cart ID=" + item1.getCart().getId());

System.out.println("item2 ID=" + item2.getId() + ", Foreign Key Cart ID=" + item1.getCart().getId());

} catch (Exception e) {

System.out.println("Exception occured. " + e.getMessage());

e.printStackTrace();

} finally {

if (!sessionFactory.isClosed()) {

System.out.println("Closing SessionFactory");

sessionFactory.close();

}

}

}

}

**Summary :**

**1.In Cart class**

@OneToMany(mappedBy = "cart")

private Set<Items> items;

**2.In Items class**

@ManyToOne

@JoinColumn(name = "cart\_id", nullable = false)

private Cart cart;

Note : cartID/cart\_id of Cart is always primarykey/id so the cart\_id value from cart class will be saved to column “cart\_id” in item class.