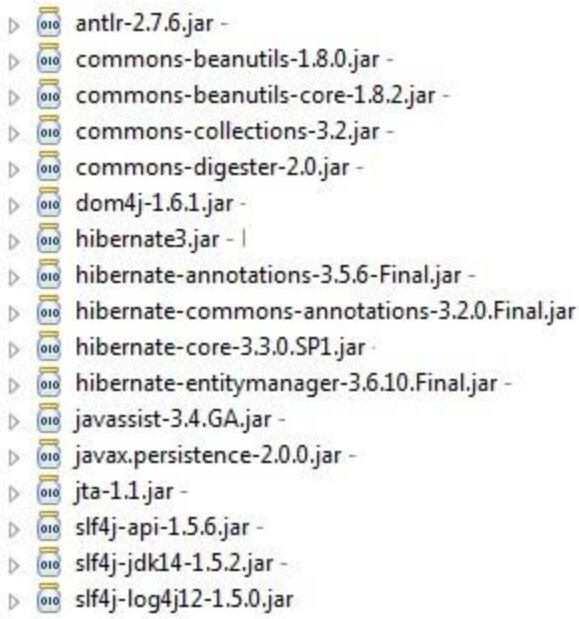
LAB 1.1

Give one Simple example for JPA integration with database. And also perform CURD operations.

Steps:

- Open IDE then create new JPA project. Name your project as JPADemo.
- Add the following jars into your classpath.



- Create new package in the name of org.capgemini and then add the following classes.

BankAccount.java

```
package org.capgemini;
import java.util.Date;

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


@Entity
public class BankAccount {

	@Id
	@GeneratedValue
```

```
	private int accountNo;

	@Column(nullable=false)
	private String accountName;
	private double depositAmt;
	private Date openDate;
	public int getAccountNo() {
		return accountNo;
	}
	public void setAccountNo(int accountNo) {
		this.accountNo = accountNo;
	}
	public String getAccountName() {
		return accountName;
	}



	public void setAccountName(String accountName) {
		this.accountName = accountName;
	}
	public double getDepositAmt() {
		return depositAmt;
	}
	public void setDepositAmt(double depositAmt) {
		this.depositAmt = depositAmt;
	}
	public Date getOpenDate() {
		return openDate;
	}
	public void setOpenDate(Date openDate) {
		this.openDate = openDate;
	}
	@Override
	public String toString() {
		return "BankAccount [accountNo=" + accountNo + ", accountName="
				+ accountName + ", depositAmt=" + depositAmt + ",
openDate="
				+ openDate + "]";
	}
}
```

TestAccount.java

```
package org.capgemini;

import java.util.Date;
import javax.persistence.EntityManager;
```

```
import javax.persistence.EntityManagerFactory;
import javax.persistence.Persistence;

public class TestAccount {
	public static void main(String[] args) {
		EntityManagerFactory
emf=Persistence.createEntityManagerFactory("JPADemo");

		EntityManager em=emf.createEntityManager();

		em.getTransaction().begin();

			BankAccount bankAccount=new BankAccount();
			//bankAccount.setAccountNo(1111);
			bankAccount.setAccountName("Tom");
			bankAccount.setDepositAmt(78000);
			bankAccount.setOpenDate(new Date());


			BankAccount bankAccount1=new BankAccount();
			//bankAccount1.setAccountNo(12345);
			bankAccount1.setAccountName("Jerry");
			bankAccount1.setDepositAmt(8000);
			bankAccount1.setOpenDate(new Date());


			BankAccount bankAccount2=new BankAccount();
			//bankAccount2.setAccountNo(1111);
			bankAccount2.setAccountName("Jack");
			bankAccount2.setDepositAmt(7000);
			bankAccount2.setOpenDate(new Date());

			//save
			em.persist(bankAccount);
			em.persist(bankAccount1);
			em.persist(bankAccount2);
			em.getTransaction().commit();

	em.getTransaction().begin();
		BankAccount account=(BankAccount)em.find(BankAccount.class, 2);
		em.remove(account);
		//account.setDepositAmt(12000);
	//em.merge(account);
	System.out.println(account);
	em.getTransaction().commit();

	}
}
```

- Open the persistence.xml file under META-INF folder, do the changes mentioned below.

persistence.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.0" xmlns="http://java.sun.com/xml/ns/persistence"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
http://java.sun.com/xml/ns/persistence/persistence_2_0.xsd">
      <persistence-unit name="JPADemo">
      <provider>org.hibernate.ejb.HibernatePersistence</provider>
      <class>org.capgemini.BankAccount</class>

      <properties>

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

<property name="hibernate.dialect"
value="org.hibernate.dialect.MySQL5Dialect"/>

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

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

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

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

<property name="hibernate.show_sql" value="true"/>
</properties>
</persistence-unit>
</persistence>
```

Learning:

- From the above example we learnt how to integrate database with JPA Object Oriented Mapping Tool.
- Introduce the following annotations also, @GeneratedValue, @Basic, @Transient, @Temporal, @Column, @Table.
- The Class should be similar like the following class.

BankAccount.java

```
package org.capgemini;
import java.util.Date;
```

```
import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;
import javax.persistence.Transient;

@Entity
@Table(name="Bank_Account_Table")
public class BankAccount {

	@Id
	@GeneratedValue
	private int accountNo;

	@Transient
	private String accPassword;

	@Column(name="AccName", nullable=false)
	private String accountName;

	@Basic
	private double depositAmt;

	@Temporal(TemporalType.DATE)
	private Date openDate;

	public String getAccPassword() {
		return accPassword;
	}
	public void setAccPassword(String accPassword) {
		this.accPassword = accPassword;
	}
	public int getAccountNo() {
		return accountNo;
	}
	public void setAccountNo(int accountNo) {
		this.accountNo = accountNo;
	}
	public String getAccountName() {
		return accountName;
	}
	public void setAccountName(String accountName) {
		this.accountName = accountName;
	}
```

```
	public double getDepositAmt() {
		return depositAmt;
	}
	public void setDepositAmt(double depositAmt) {
		this.depositAmt = depositAmt;
	}
	public Date getOpenDate() {
		return openDate;
	}
	public void setOpenDate(Date openDate) {
		this.openDate = openDate;
	}
	@Override
	public String toString() {
		return "BankAccount [accountNo=" + accountNo + ", accountName="
				+ accountName + ", depositAmt=" + depositAmt + ",
openDate="
				+ openDate + "]";
	}

}
```

LAB 1.2

Give one demo for how to map one class into two different tables in JPA.

Steps:

- Steps to realize / do the assignments

```
package org.capgemini.oneClass;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.SecondaryTable;
import javax.persistence.Table;

@Entity
@Table(name="cap_Customer")
@SecondaryTable(name="cap_customerDetails")
public class Customer {

	@Id
	@GeneratedValue
	private int custId;
	private String custName;

	@Column(table="cap_customerDetails")
	private String doorNo;

	@Column(table="cap_customerDetails")
	private String streetName;

	@Column(table="cap_customerDetails")
	private String city;


	public int getCustId() {
		return custId;
	}
	public void setCustId(int custId) {
		this.custId = custId;
	}
	public String getCustName() {
		return custName;
	}
	public void setCustName(String custName) {
```

```
		this.custName = custName;
	}
	public String getDoorNo() {
		return doorNo;
	}
	public void setDoorNo(String doorNo) {
		this.doorNo = doorNo;
	}
	public String getStreetName() {
		return streetName;
	}
	public void setStreetName(String streetName) {
		this.streetName = streetName;
	}
	public String getCity() {
		return city;
	}
	public void setCity(String city) {
		this.city = city;
	}

}
```

```
package org.capgemini.oneClass;

import javax.persistence.EntityManager;

import javax.persistence.EntityManagerFactory;

import javax.persistence.Persistence;

public class TestClass {

	public static void main(String[] args) {

		EntityManagerFactory emf=Persistence.createEntityManagerFactory("JPADemo");

		EntityManager em=emf.createEntityManager();

			Customer customer=new Customer();

			customer.setCustName("Tom");

			customer.setDoorNo("123/45");

			customer.setStreetName("North Street");
```

```
                customer.setCity("Chennai");

                em.getTransaction().begin();

                em.persist(customer);//Persist

                em.getTransaction().commit();

        }

}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.0"
xmlns="http://java.sun.com/xml/ns/persistence"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
http://java.sun.com/xml/ns/persistence/persistence_2_0.xsd">
      <persistence-unit name="JPADemo">
      <provider>org.hibernate.ejb.HibernatePersistence</provider>
      <class>org.capgemini.oneClass.Customer</class>

<properties>
<property name="hibernate.connection.url"
value="jdbc:mysql://localhost:3306/capgemini"/>
<property name="hibernate.dialect"
value="org.hibernate.dialect.MySQL5Dialect"/>
<property name="hibernate.connection.driver_class"
value="com.mysql.jdbc.Driver"/>
<property name="hibernate.connection.password" value="admin"/>
<property name="hibernate.connection.username" value="root"/>
<property name="hibernate.hbm2ddl.auto" value="create"/>
<property name="hibernate.show_sql" value="true"/>
</properties>
</persistence-unit>
</persistence>
```

Learning:

- Leaning from this particular assignment (basically, purpose of this assignment)