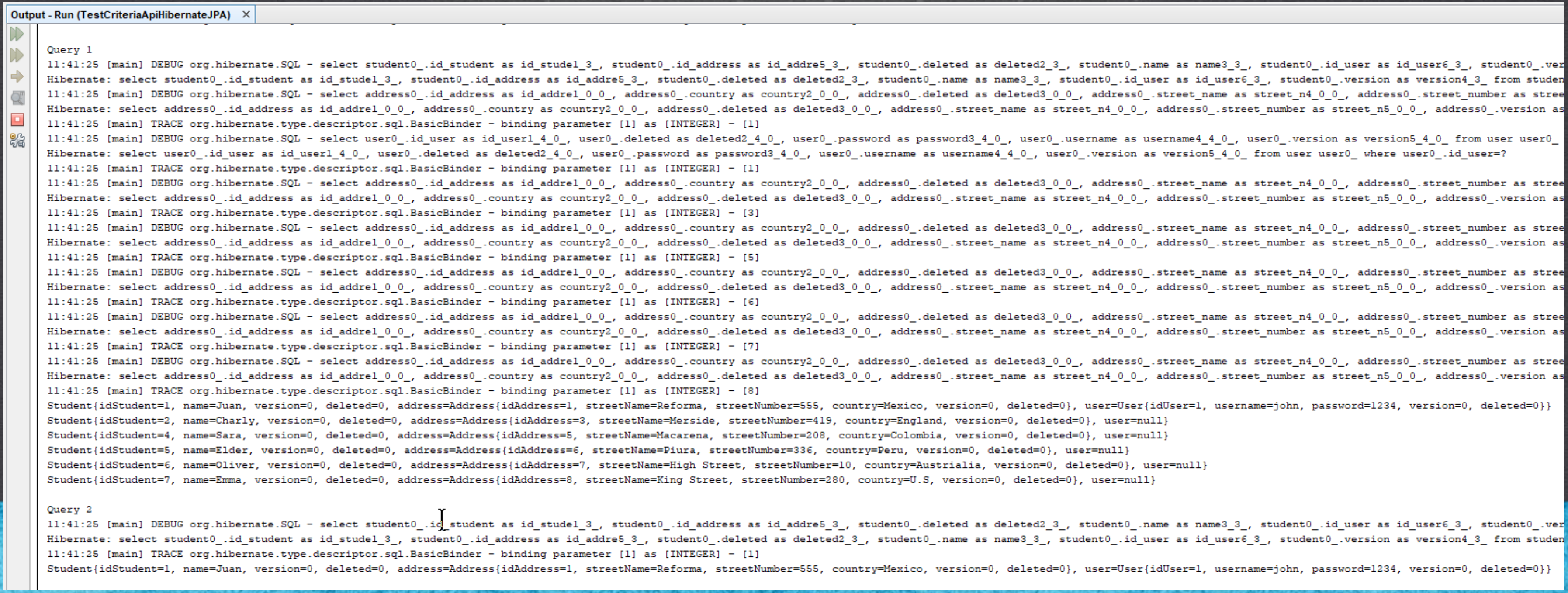
HIBERNATE & JPA COURSE

EXERCISE

CRITERIA API WITH HIBERNATE/JPA

EXERCISE OBJECTIVE

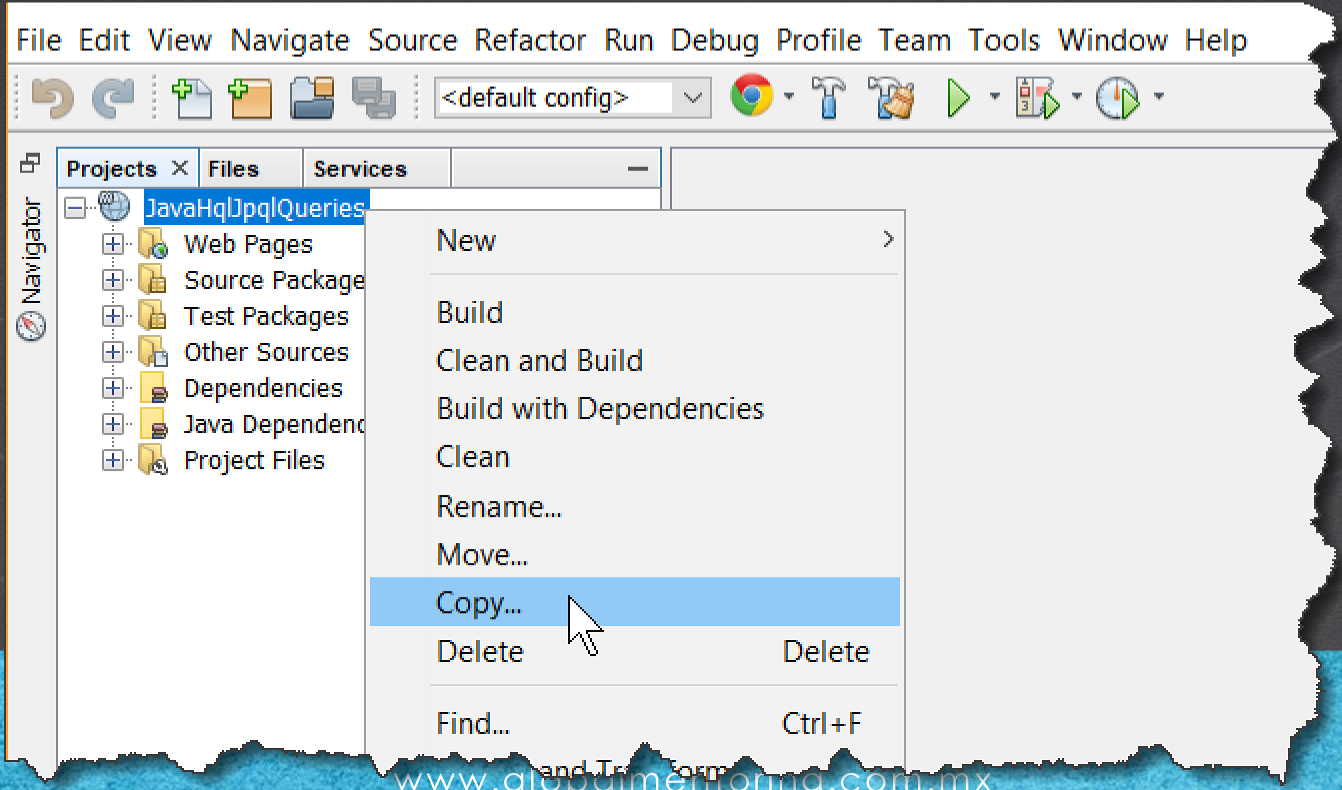
Use the Criteria API to implement this API. At the end we should observe the following:



```
Query 1
11:41:25 [main] DEBUG org.hibernate.SQL - select student0_.id_student as id_stude1_3_, student0_.id_address as id_addre5_3_, student0_.deleted as deleted2_3_, student0_.name as name3_3_, student0_.id_user as id_user6_3_, student0_.ver
Hibernate: select student0_.id_student as id_stude1_3_, student0_.id_address as id_addre5_3_, student0_.deleted as deleted2_3_, student0_.name as name3_3_, student0_.id_user as id_user6_3_, student0_.version as version4_3_ from studen
11:41:25 [main] DEBUG org.hibernate.SQL - select address0_.id_address as id_addre1_0_0_, address0_.country as country2_0_0_, address0_.deleted as deleted3_0_0_, address0_.street_name as street_n4_0_0_, address0_.street_number as stree
Hibernate: select address0_.id_address as id_addre1_0_0_, address0_.country as country2_0_0_, address0_.deleted as deleted3_0_0_, address0_.street_name as street_n4_0_0_, address0_.street_number as street_n5_0_0_, address0_.version as
11:41:25 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [1] as [INTEGER] - [1]
11:41:25 [main] DEBUG org.hibernate.SQL - select user0_.id_user as id_user1_4_0_, user0_.deleted as deleted2_4_0_, user0_.password as password3_4_0_, user0_.username as username4_4_0_, user0_.version as version5_4_0_ from user user0_
Hibernate: select user0_.id_user as id_user1_4_0_, user0_.deleted as deleted2_4_0_, user0_.password as password3_4_0_, user0_.username as username4_4_0_, user0_.version as version5_4_0_ from user user0_ where user0_.id_user=?
11:41:25 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [1] as [INTEGER] - [1]
11:41:25 [main] DEBUG org.hibernate.SQL - select address0_.id_address as id_addre1_0_0_, address0_.country as country2_0_0_, address0_.deleted as deleted3_0_0_, address0_.street_name as street_n4_0_0_, address0_.street_number as stree
Hibernate: select address0_.id_address as id_addre1_0_0_, address0_.country as country2_0_0_, address0_.deleted as deleted3_0_0_, address0_.street_name as street_n4_0_0_, address0_.street_number as street_n5_0_0_, address0_.version as
11:41:25 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [1] as [INTEGER] - [3]
11:41:25 [main] DEBUG org.hibernate.SQL - select address0_.id_address as id_addre1_0_0_, address0_.country as country2_0_0_, address0_.deleted as deleted3_0_0_, address0_.street_name as street_n4_0_0_, address0_.street_number as stree
Hibernate: select address0_.id_address as id_addre1_0_0_, address0_.country as country2_0_0_, address0_.deleted as deleted3_0_0_, address0_.street_name as street_n4_0_0_, address0_.street_number as street_n5_0_0_, address0_.version as
11:41:25 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [1] as [INTEGER] - [5]
11:41:25 [main] DEBUG org.hibernate.SQL - select address0_.id_address as id_addre1_0_0_, address0_.country as country2_0_0_, address0_.deleted as deleted3_0_0_, address0_.street_name as street_n4_0_0_, address0_.street_number as stree
Hibernate: select address0_.id_address as id_addre1_0_0_, address0_.country as country2_0_0_, address0_.deleted as deleted3_0_0_, address0_.street_name as street_n4_0_0_, address0_.street_number as street_n5_0_0_, address0_.version as
11:41:25 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [1] as [INTEGER] - [6]
11:41:25 [main] DEBUG org.hibernate.SQL - select address0_.id_address as id_addre1_0_0_, address0_.country as country2_0_0_, address0_.deleted as deleted3_0_0_, address0_.street_name as street_n4_0_0_, address0_.street_number as stree
Hibernate: select address0_.id_address as id_addre1_0_0_, address0_.country as country2_0_0_, address0_.deleted as deleted3_0_0_, address0_.street_name as street_n4_0_0_, address0_.street_number as street_n5_0_0_, address0_.version as
11:41:25 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [1] as [INTEGER] - [7]
11:41:25 [main] DEBUG org.hibernate.SQL - select address0_.id_address as id_addre1_0_0_, address0_.country as country2_0_0_, address0_.deleted as deleted3_0_0_, address0_.street_name as street_n4_0_0_, address0_.street_number as stree
Hibernate: select address0_.id_address as id_addre1_0_0_, address0_.country as country2_0_0_, address0_.deleted as deleted3_0_0_, address0_.street_name as street_n4_0_0_, address0_.street_number as street_n5_0_0_, address0_.version as
11:41:25 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [1] as [INTEGER] - [8]
Student{idStudent=1, name=Juan, version=0, deleted=0, address=Address{idAddress=1, streetName=Reforma, streetNumber=555, country=Mexico, version=0, deleted=0}, user=User{idUser=1, username=john, password=1234, version=0, deleted=0}}
Student{idStudent=2, name=Charly, version=0, deleted=0, address=Address{idAddress=3, streetName=Merside, streetNumber=419, country=England, version=0, deleted=0}, user=null}
Student{idStudent=4, name=Sara, version=0, deleted=0, address=Address{idAddress=5, streetName=Macarena, streetNumber=208, country=Colombia, version=0, deleted=0}, user=null}
Student{idStudent=5, name=Elder, version=0, deleted=0, address=Address{idAddress=6, streetName=Piura, streetNumber=336, country=Peru, version=0, deleted=0}, user=null}
Student{idStudent=6, name=Oliver, version=0, deleted=0, address=Address{idAddress=7, streetName=High Street, streetNumber=10, country=Austrialia, version=0, deleted=0}, user=null}
Student{idStudent=7, name=Emma, version=0, deleted=0, address=Address{idAddress=8, streetName=King Street, streetNumber=280, country=U.S, version=0, deleted=0}, user=null}

Query 2
11:41:25 [main] DEBUG org.hibernate.SQL - select student0_.id_student as id_stude1_3_, student0_.id_address as id_addre5_3_, student0_.deleted as deleted2_3_, student0_.name as name3_3_, student0_.id_user as id_user6_3_, student0_.ver
Hibernate: select student0_.id_student as id_stude1_3_, student0_.id_address as id_addre5_3_, student0_.deleted as deleted2_3_, student0_.name as name3_3_, student0_.id_user as id_user6_3_, student0_.version as version4_3_ from studen
11:41:25 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [1] as [INTEGER] - [1]
Student{idStudent=1, name=Juan, version=0, deleted=0, address=Address{idAddress=1, streetName=Reforma, streetNumber=555, country=Mexico, version=0, deleted=0}, user=User{idUser=1, username=john, password=1234, version=0, deleted=0}}
```

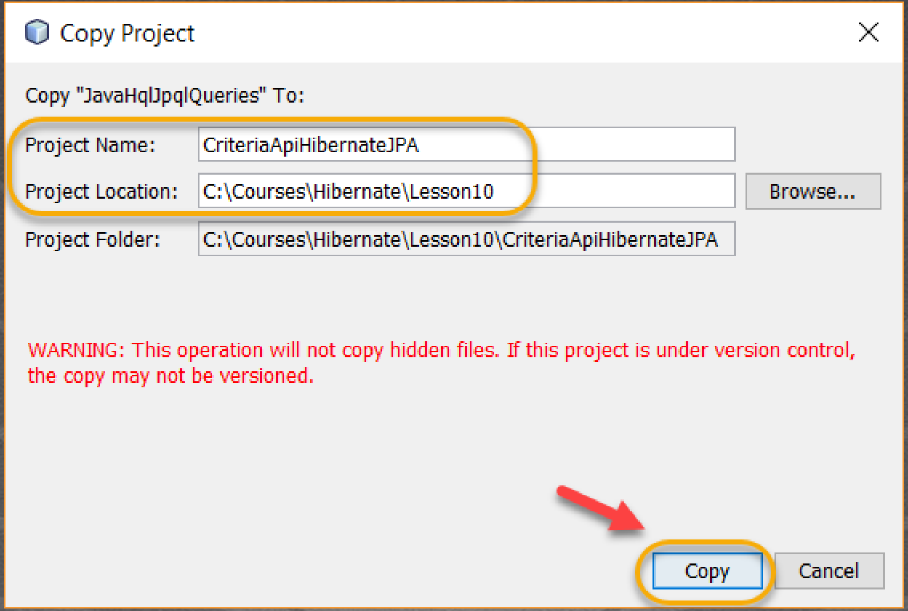
1. COPY THE PROJECT

We copy the project starting from JavaHqlJpqlQueries:



1. COPY THE PROJECT

Create the Project CriteriaApiHibernateJPA:



Copy Project

Copy "JavaHqlJpqlQueries" To:

Project Name: CriteriaApiHibernateJPA

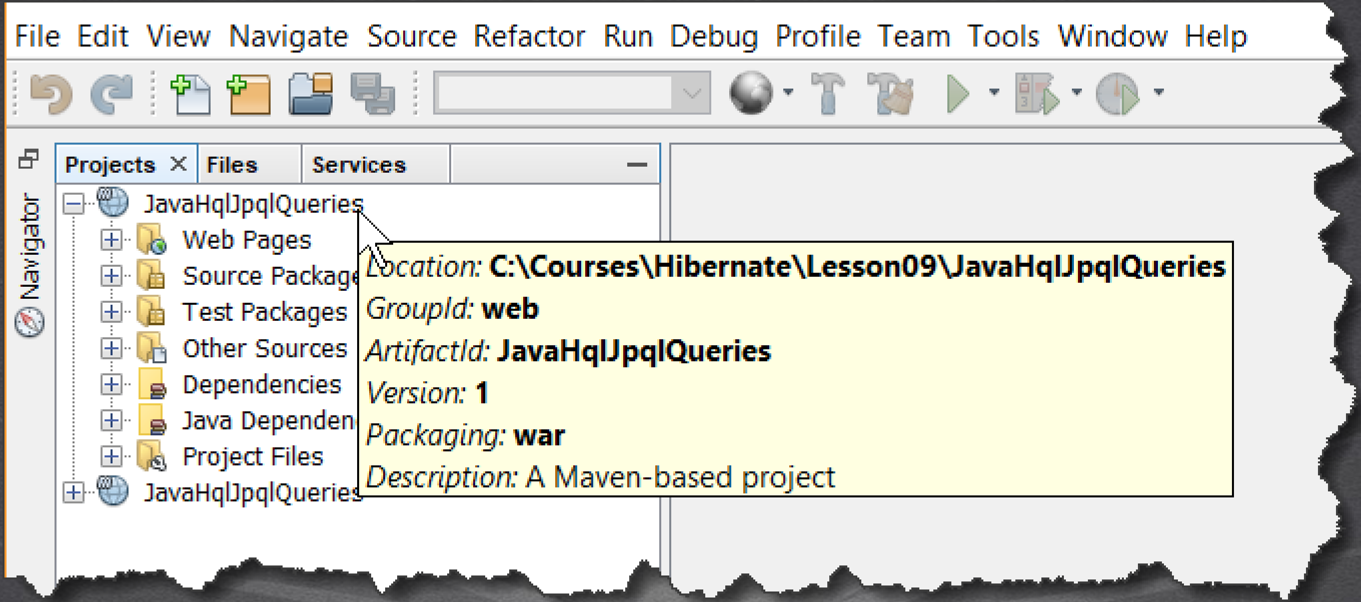
Project Location: C:\Courses\Hibernate\Lesson10

Project Folder: C:\Courses\Hibernate\Lesson10\CriteriaApiHibernateJPA

WARNING: This operation will not copy hidden files. If this project is under version control, the copy may not be versioned.

2. CLOSE THE PROJECT

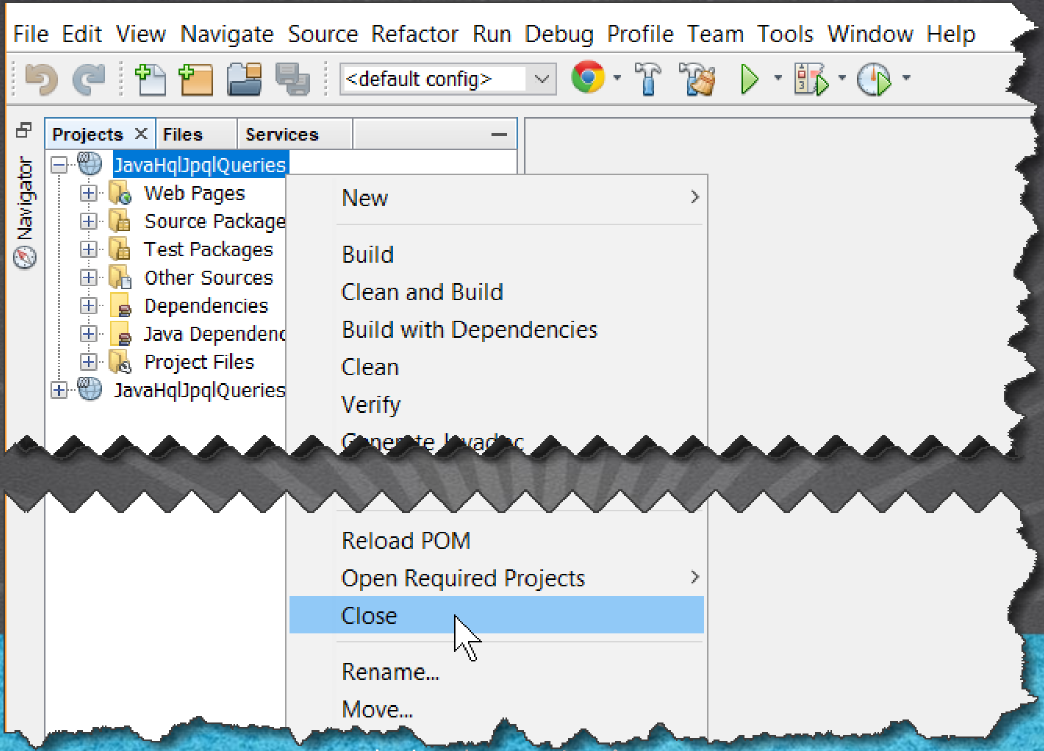
We close the project that we no longer use:



Location: **C:\Courses\Hibernate\Lesson09\JavaHqlJpqlQueries**
GroupId: **web**
ArtifactId: **JavaHqlJpqlQueries**
Version: **1**
Packaging: **war**
Description: A Maven-based project

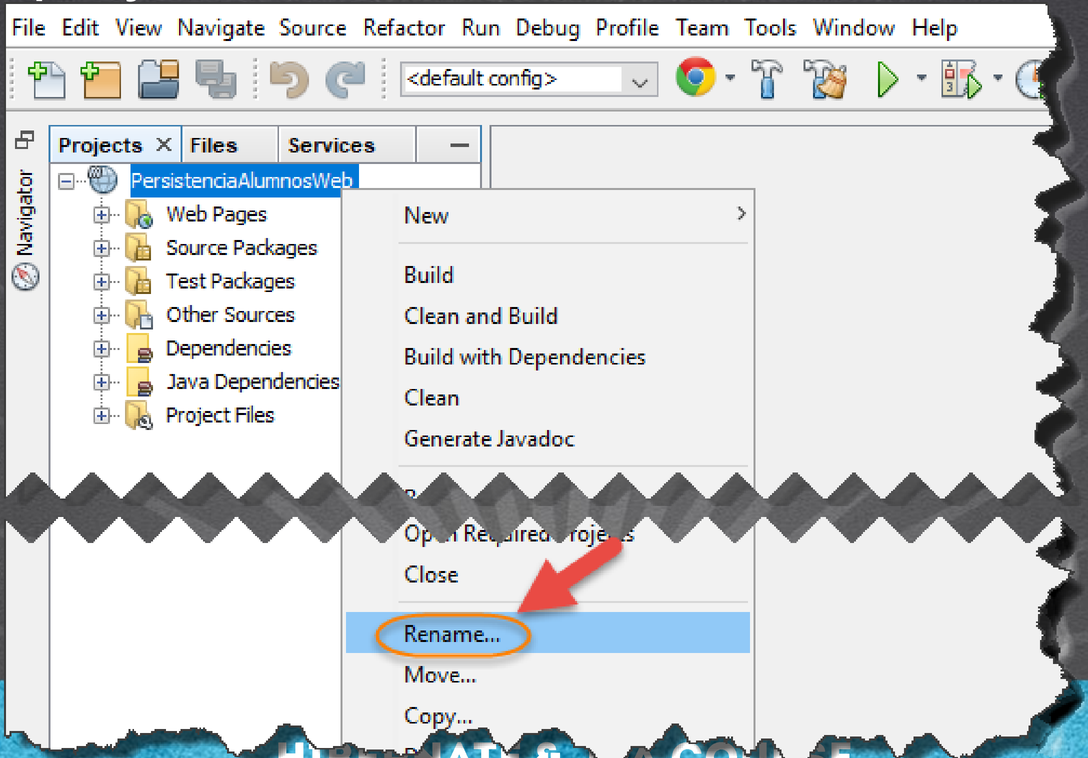
2. CLOSE THE PROJECT

We close the project that we no longer use:



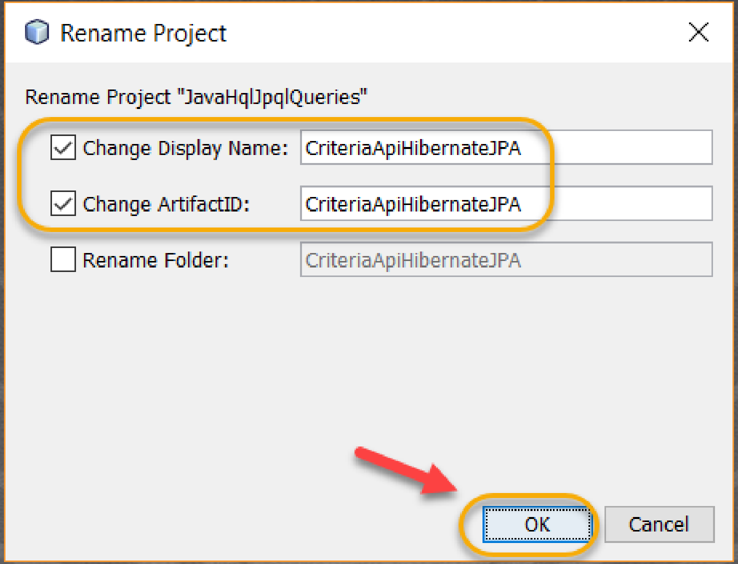
3. RENAME THE PROJECT

Rename the project:



3. RENAME THE PROJECT

Rename the Project:



Rename Project "JavaHqlJpqlQueries"

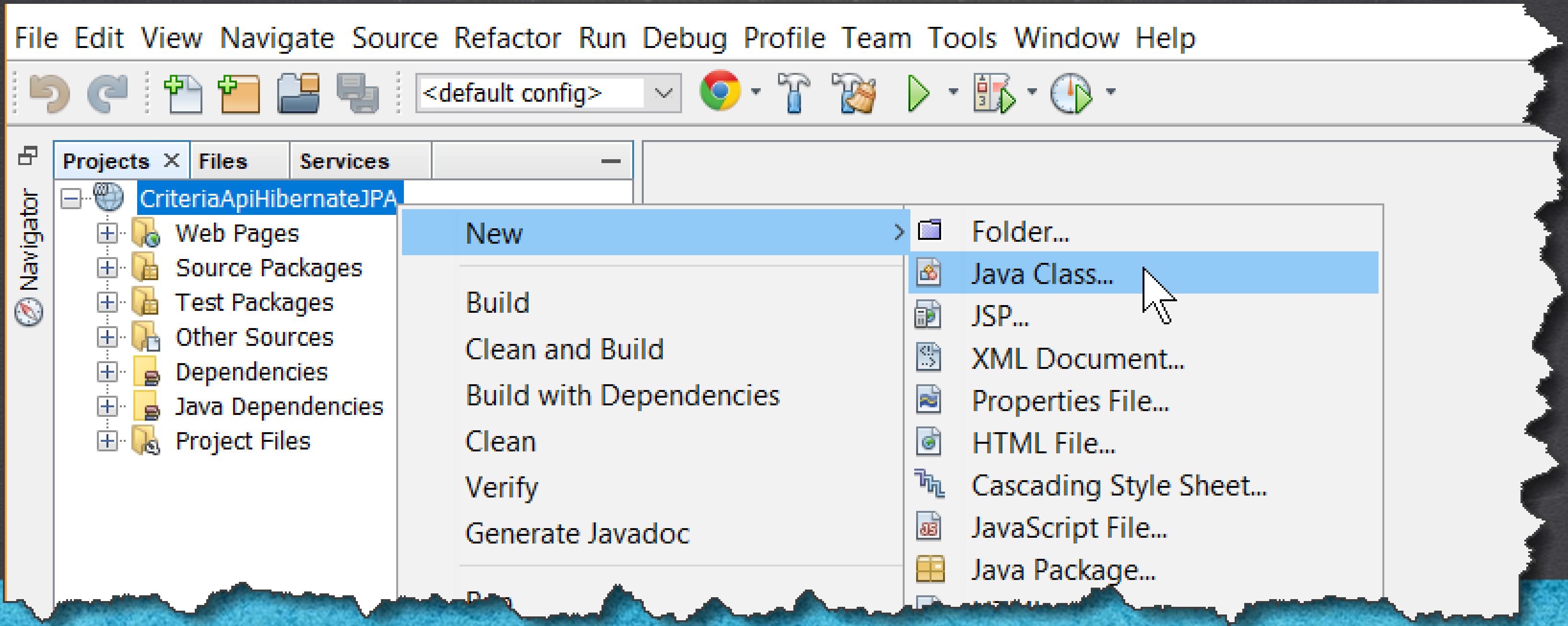
☑ Change Display Name: CriteriaApiHibernateJPA

☑ Change ArtifactID: CriteriaApiHibernateJPA

□ Rename Folder: CriteriaApiHibernateJPA

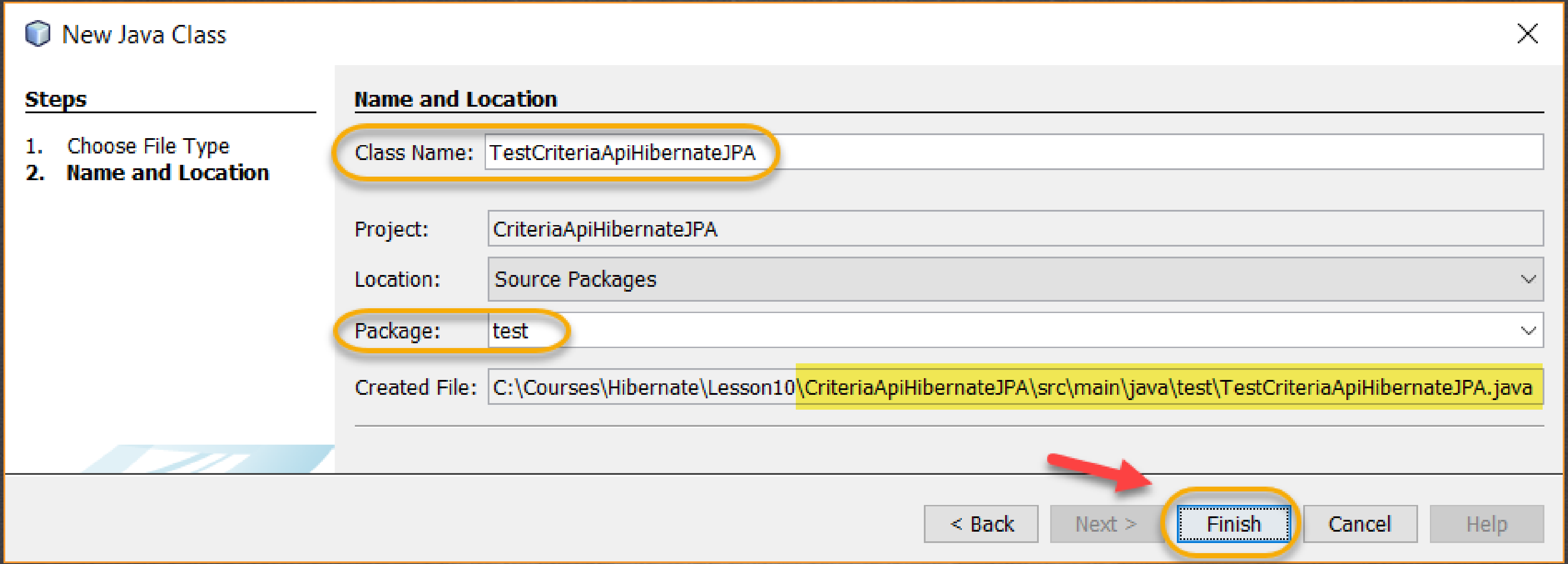
4. CREATE A CLASS

Create a class TestCriteriaApiHibernateJPA.java:



4. CREATE A CLASS

Create a class TestCriteriaApiHibernateJPA.java:



New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name: TestCriteriaApiHibernateJPA

Project: CriteriaApiHibernateJPA

Location: Source Packages

Package: test

Created File: C:\Courses\Hibernate\Lesson10\CriteriaApiHibernateJPA\src\main\java\test\TestCriteriaApiHibernateJPA.java

5. MODIFY THE CODE

TestCriteriaApiHibernateJPA.java:

Click to download

```
package test;

import java.util.*;
import javax.persistence.*;
import javax.persistence.criteria.*;
import model.*;

public class TestCriteriaApiHibernateJPA {

    public static void main(String[] args) {
        EntityManagerFactory fabrica = Persistence.createEntityManagerFactory("HibernateJpaPU");
        EntityManager em = fabrica.createEntityManager();

        //Help variables
        CriteriaBuilder cb = em.getCriteriaBuilder();
        List<Student> students = null;
        Student student = null;
```

5. MODIFY THE CODE

TestCriteriaApiHibernateJPA.java:

Click to download

```
// Query 1
// Select all Students
//JPQL equivalent:SELECT a FROM Student a
System.out.println("\nQuery 1");
//The criteria query object is created
CriteriaQuery<Student> q1 = cb.createQuery(Student.class);
//Set the root of the query
q1.from(Student.class);
//The query is executed
students = em.createQuery(q1).getResultList();
//We print the students
printStudents(students);

// Query2
// Select the Student with id = 1
System.out.println("\nQuery 2");
CriteriaQuery<Student> q2 = cb.createQuery(Student.class);
Root<Student> c2 = q2.from(Student.class);
ParameterExpression<Integer> pId = cb.parameter(Integer.class);
q2.select(c2).where(cb.equal(c2.get("idStudent"), pId));
//execute the query
TypedQuery<Student> query = em.createQuery(q2);
//set the value of the parameter
query.setParameter(pId, 1);
student = query.getSingleResult();
System.out.println(student);
```

5. MODIFY THE CODE

TestCriteriaApiHibernateJPA.java:

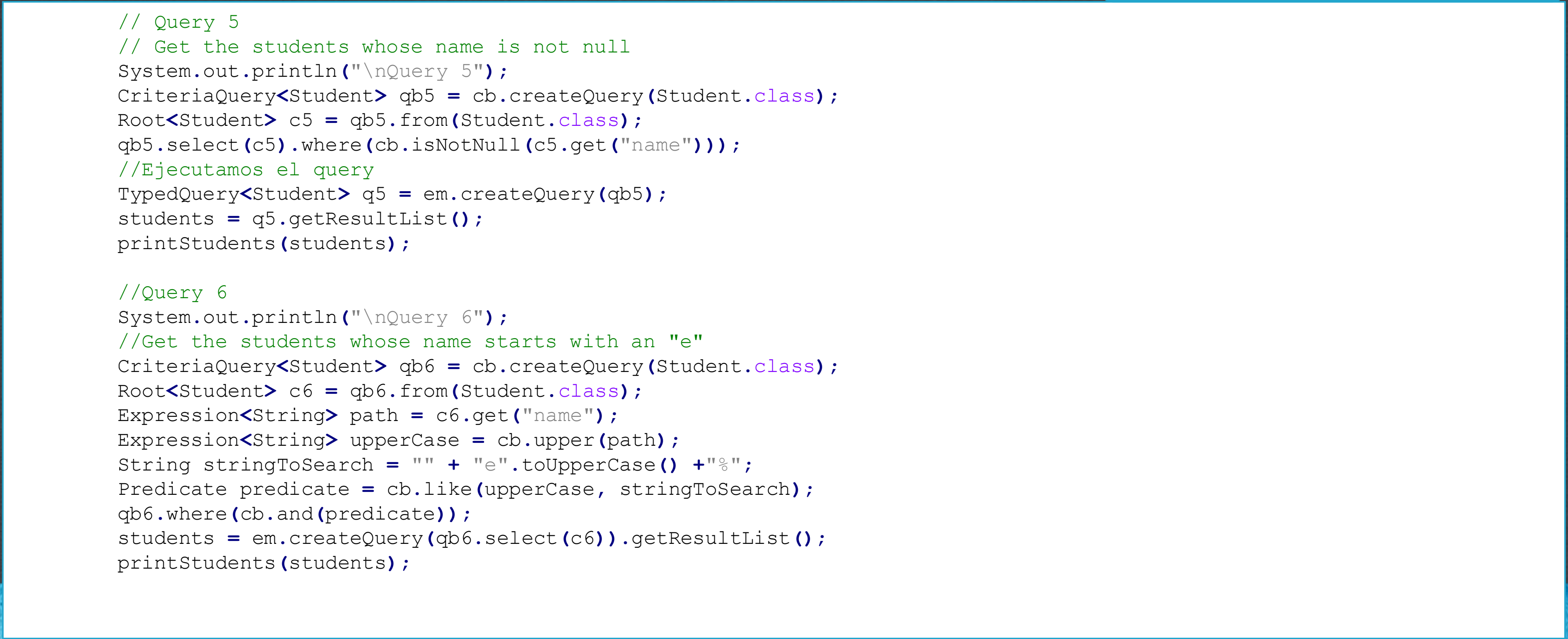
Click to download

```java
// Query 3
// Select the student with name
System.out.println("\nQuery 3");
CriteriaQuery<Student> q3 = cb.createQuery(Student.class);
Root<Student> c3 = q3.from(Student.class);
ParameterExpression<String> pNombre = cb.parameter(String.class);
q3.select(c3).where(cb.equal(c3.get("name"), pNombre));
//execute the query
TypedQuery<Student> query3 = em.createQuery(q3);
//set the value of the parameter
query3.setParameter(pNombre, "Charly");
students = query3.getResultList();
printStudents(students);

// Query 4
// Select students restricting by the idStudent
System.out.println("\nQuery 4");
CriteriaQuery<Student> qb4 = cb.createQuery(Student.class);
Root<Student> c4 = qb4.from(Student.class);
qb4.where(c4.get("idStudent").in(cb.parameter(Collection.class)));

TypedQuery<Student> q4 = em.createQuery(qb4);
Integer[] idStudents = {1,2};//place valid id's
for (ParameterExpression parameter : qb4.getParameters()) {
    q4.setParameter(parameter, Arrays.asList( idStudents ));
}
students = q4.getResultList();
printStudents(students);
```

5. MODIFY THE CODE

TestCriteriaApiHibernateJPA.java:

Click to download

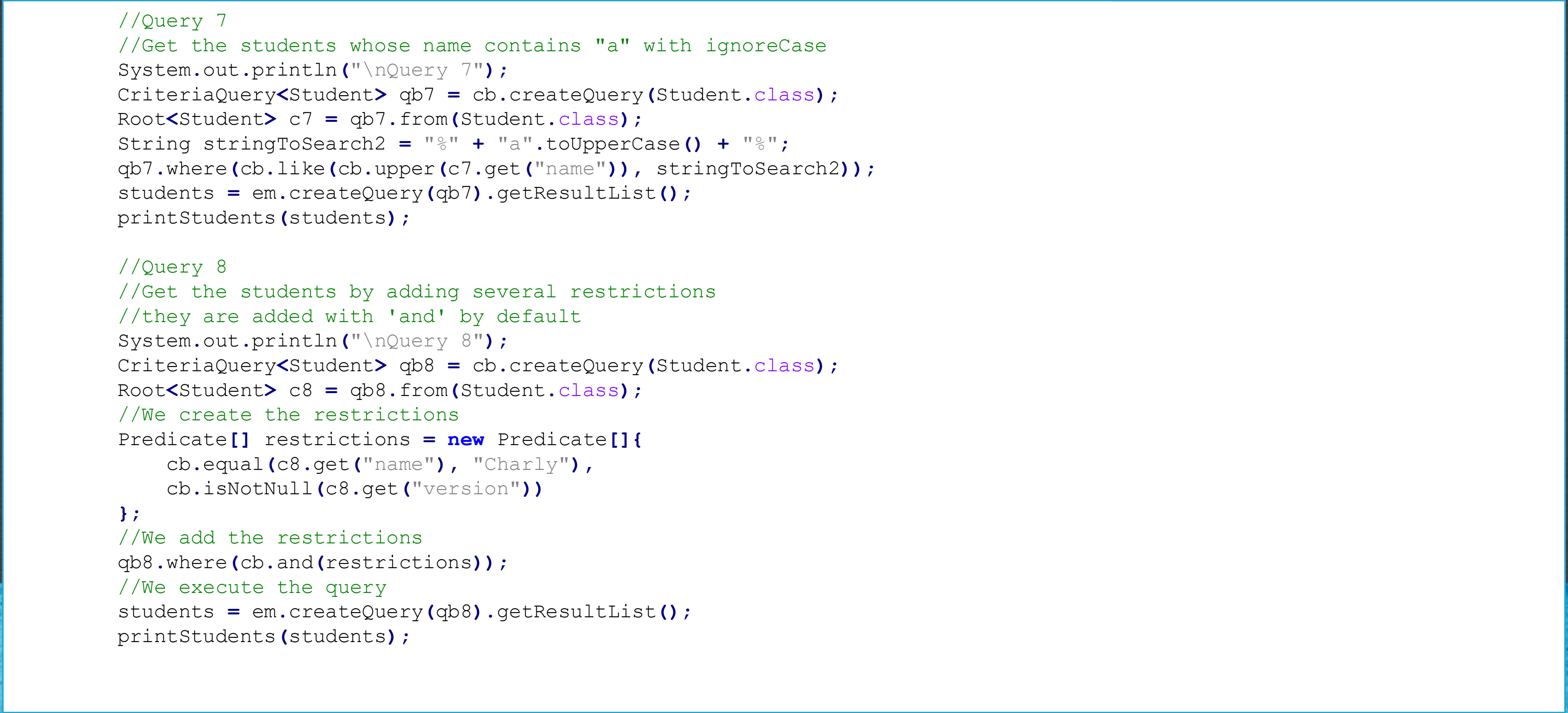


```
// Query 5
// Get the students whose name is not null
System.out.println("\nQuery 5");
CriteriaQuery<Student> qb5 = cb.createQuery(Student.class);
Root<Student> c5 = qb5.from(Student.class);
qb5.select(c5).where(cb.isNotNull(c5.get("name")));
//Ejecutamos el query
TypedQuery<Student> q5 = em.createQuery(qb5);
students = q5.getResultList();
printStudents(students);

//Query 6
System.out.println("\nQuery 6");
//Get the students whose name starts with an "e"
CriteriaQuery<Student> qb6 = cb.createQuery(Student.class);
Root<Student> c6 = qb6.from(Student.class);
Expression<String> path = c6.get("name");
Expression<String> upperCase = cb.upper(path);
String stringToSearch = "" + "e".toUpperCase() +"%";
Predicate predicate = cb.like(upperCase, stringToSearch);
qb6.where(cb.and(predicate));
students = em.createQuery(qb6.select(c6)).getResultList();
printStudents(students);
```

5. MODIFY THE CODE

TestCriteriaApiHibernateJPA.java:

Click to download



```java
//Query 7
//Get the students whose name contains "a" with ignoreCase
System.out.println("\nQuery 7");
CriteriaQuery<Student> qb7 = cb.createQuery(Student.class);
Root<Student> c7 = qb7.from(Student.class);
String stringToSearch2 = "%" + "a".toUpperCase() + "%";
qb7.where(cb.like(cb.upper(c7.get("name")), stringToSearch2));
students = em.createQuery(qb7).getResultList();
printStudents(students);

//Query 8
//Get the students by adding several restrictions
//they are added with 'and' by default
System.out.println("\nQuery 8");
CriteriaQuery<Student> qb8 = cb.createQuery(Student.class);
Root<Student> c8 = qb8.from(Student.class);
//We create the restrictions
Predicate[] restrictions = new Predicate[]{
    cb.equal(c8.get("name"), "Charly"),
    cb.isNotNull(c8.get("version"))
};
//We add the restrictions
qb8.where(cb.and(restrictions));
//We execute the query
students = em.createQuery(qb8).getResultList();
printStudents(students);
```

5. MODIFY THE CODE

TestCriteriaApiHibernateJPA.java:

Click to download

```
//Query 9
//Get the students by adding several restrictions
//they are added with 'or'
System.out.println("\nQuery 9");
CriteriaQuery<Student> qb9 = cb.createQuery(Student.class);
Root<Student> c9 = qb9.from(Student.class);
//We create the restrictions
Predicate[] restrictions2 = new Predicate[]{
    cb.equal(c9.get("name"), "Charly"),
    cb.isNotNull(c9.get("version"))
};
//We add the restrictions
qb9.where(cb.or(restrictions2));
//We execute the query
students = em.createQuery(qb9).getResultList();
printStudents(students);
```

5. MODIFY THE CODE

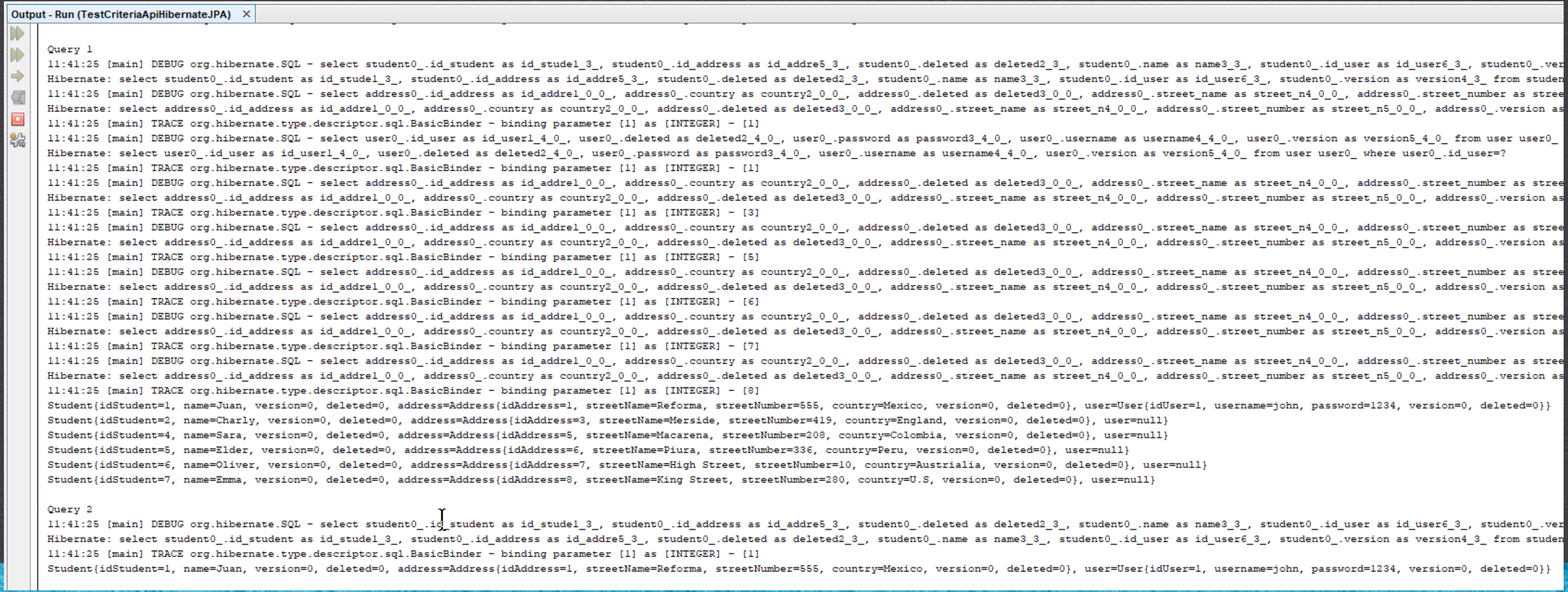
TestCriteriaApiHibernateJPA.java:

Click to download

```
        //Query 10
        //Get the students whose name is not null
        //adding ordering by name asc and version desc
        System.out.println("\nQuery 10");
        CriteriaQuery<Student> qb10 = cb.createQuery(Student.class);
        Root<Student> c10 = qb10.from(Student.class);
        //We create the restrictions
        Predicate[] restrictions3 = new Predicate[]{
            cb.equal(c10.get("name"), "Charly"),
            cb.isNotNull(c10.get("version"))
        };
        //We add the restrictions
        qb10.where(cb.or(restrictions3));
        //We add order
        qb10.orderBy(cb.asc(c10.get("name")), cb.desc(c10.get("version")));
        //We execute the query
        students = em.createQuery(qb10).getResultList();
        printStudents(students);

    }

    private static void printStudents(List<Student> students) {
        for (Student s : students) {
            System.out.println(s);
        }
    }
}
```

5. EXECUTE THE PROJECT

We execute each of the queries of the project:



```
Query 1
11:41:25 [main] DEBUG org.hibernate.SQL - select student0_.id_student as id_stude1_3_, student0_.id_address as id_addre5_3_, student0_.deleted as deleted2_3_, student0_.name as name3_3_, student0_.id_user as id_user6_3_, student0_.ver
Hibernate: select student0_.id_student as id_stude1_3_, student0_.id_address as id_addre5_3_, student0_.deleted as deleted2_3_, student0_.name as name3_3_, student0_.id_user as id_user6_3_, student0_.version as version4_3_ from studen
11:41:25 [main] DEBUG org.hibernate.SQL - select address0_.id_address as id_addre1_0_0_, address0_.country as country2_0_0_, address0_.deleted as deleted3_0_0_, address0_.street_name as street_n4_0_0_, address0_.street_number as stree
Hibernate: select address0_.id_address as id_addre1_0_0_, address0_.country as country2_0_0_, address0_.deleted as deleted3_0_0_, address0_.street_name as street_n4_0_0_, address0_.street_number as street_n5_0_0_, address0_.version as
11:41:25 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [1] as [INTEGER] - [1]
11:41:25 [main] DEBUG org.hibernate.SQL - select user0_.id_user as id_user1_4_0_, user0_.deleted as deleted2_4_0_, user0_.password as password3_4_0_, user0_.username as username4_4_0_, user0_.version as version5_4_0_ from user user0_
Hibernate: select user0_.id_user as id_user1_4_0_, user0_.deleted as deleted2_4_0_, user0_.password as password3_4_0_, user0_.username as username4_4_0_, user0_.version as version5_4_0_ from user user0_ where user0_.id_user=?
11:41:25 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [1] as [INTEGER] - [1]
11:41:25 [main] DEBUG org.hibernate.SQL - select address0_.id_address as id_addre1_0_0_, address0_.country as country2_0_0_, address0_.deleted as deleted3_0_0_, address0_.street_name as street_n4_0_0_, address0_.street_number as stree
Hibernate: select address0_.id_address as id_addre1_0_0_, address0_.country as country2_0_0_, address0_.deleted as deleted3_0_0_, address0_.street_name as street_n4_0_0_, address0_.street_number as street_n5_0_0_, address0_.version as
11:41:25 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [1] as [INTEGER] - [3]
11:41:25 [main] DEBUG org.hibernate.SQL - select address0_.id_address as id_addre1_0_0_, address0_.country as country2_0_0_, address0_.deleted as deleted3_0_0_, address0_.street_name as street_n4_0_0_, address0_.street_number as stree
Hibernate: select address0_.id_address as id_addre1_0_0_, address0_.country as country2_0_0_, address0_.deleted as deleted3_0_0_, address0_.street_name as street_n4_0_0_, address0_.street_number as street_n5_0_0_, address0_.version as
11:41:25 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [1] as [INTEGER] - [5]
11:41:25 [main] DEBUG org.hibernate.SQL - select address0_.id_address as id_addre1_0_0_, address0_.country as country2_0_0_, address0_.deleted as deleted3_0_0_, address0_.street_name as street_n4_0_0_, address0_.street_number as stree
Hibernate: select address0_.id_address as id_addre1_0_0_, address0_.country as country2_0_0_, address0_.deleted as deleted3_0_0_, address0_.street_name as street_n4_0_0_, address0_.street_number as street_n5_0_0_, address0_.version as
11:41:25 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [1] as [INTEGER] - [6]
11:41:25 [main] DEBUG org.hibernate.SQL - select address0_.id_address as id_addre1_0_0_, address0_.country as country2_0_0_, address0_.deleted as deleted3_0_0_, address0_.street_name as street_n4_0_0_, address0_.street_number as stree
Hibernate: select address0_.id_address as id_addre1_0_0_, address0_.country as country2_0_0_, address0_.deleted as deleted3_0_0_, address0_.street_name as street_n4_0_0_, address0_.street_number as street_n5_0_0_, address0_.version as
11:41:25 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [1] as [INTEGER] - [7]
11:41:25 [main] DEBUG org.hibernate.SQL - select address0_.id_address as id_addre1_0_0_, address0_.country as country2_0_0_, address0_.deleted as deleted3_0_0_, address0_.street_name as street_n4_0_0_, address0_.street_number as stree
Hibernate: select address0_.id_address as id_addre1_0_0_, address0_.country as country2_0_0_, address0_.deleted as deleted3_0_0_, address0_.street_name as street_n4_0_0_, address0_.street_number as street_n5_0_0_, address0_.version as
11:41:25 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [1] as [INTEGER] - [8]
Student{idStudent=1, name=Juan, version=0, deleted=0, address=Address{idAddress=1, streetName=Reforma, streetNumber=555, country=Mexico, version=0, deleted=0}, user=User{idUser=1, username=john, password=1234, version=0, deleted=0}}
Student{idStudent=2, name=Charly, version=0, deleted=0, address=Address{idAddress=3, streetName=Merside, streetNumber=419, country=England, version=0, deleted=0}, user=null}
Student{idStudent=4, name=Sara, version=0, deleted=0, address=Address{idAddress=5, streetName=Macarena, streetNumber=208, country=Colombia, version=0, deleted=0}, user=null}
Student{idStudent=5, name=Elder, version=0, deleted=0, address=Address{idAddress=6, streetName=Piura, streetNumber=336, country=Peru, version=0, deleted=0}, user=null}
Student{idStudent=6, name=Oliver, version=0, deleted=0, address=Address{idAddress=7, streetName=High Street, streetNumber=10, country=Austrialia, version=0, deleted=0}, user=null}
Student{idStudent=7, name=Emma, version=0, deleted=0, address=Address{idAddress=8, streetName=King Street, streetNumber=280, country=U.S, version=0, deleted=0}, user=null}

Query 2
11:41:25 [main] DEBUG org.hibernate.SQL - select student0_.id_student as id_stude1_3_, student0_.id_address as id_addre5_3_, student0_.deleted as deleted2_3_, student0_.name as name3_3_, student0_.id_user as id_user6_3_, student0_.ver
Hibernate: select student0_.id_student as id_stude1_3_, student0_.id_address as id_addre5_3_, student0_.deleted as deleted2_3_, student0_.name as name3_3_, student0_.id_user as id_user6_3_, student0_.version as version4_3_ from studen
11:41:25 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [1] as [INTEGER] - [1]
Student{idStudent=1, name=Juan, version=0, deleted=0, address=Address{idAddress=1, streetName=Reforma, streetNumber=555, country=Mexico, version=0, deleted=0}, user=User{idUser=1, username=john, password=1234, version=0, deleted=0}}
```

EXERCISE CONCLUSION

- With this exercise we have executed several of the queries with the Criteria API of Hibernate / JPA.

- With this we can already compare and decide if we use the HQL / JPQL language or the Criteria API.

- Each one has its advantages and disadvantages, but everything will depend on what we need in our application to know if we use another solution.

- In general we will use HQL / JPQL when the queries are more static and we will use the Criteria API when the queries tend to be more dynamic, in this way we will avoid less chain concatenation to create our queries.

ONLINE COURSE

HIBERNATE & JPA

By: Eng. Ubaldo Acosta

Global Mentoring

HIBERNATE & JPA COURSE

www.globalmentoring.com.mx