**Guided LAB 305.4.2B - Demonstration of Hibernate Query Language - HQL with Intellij**

**Lab Overview**

We will continue with the **previous LAB *(GLAB 305.4.1 - Hibernate Project Demonstration)***. In this lab, we will use HQL.

**Learning Objective:**

After finishing this lab, you will be able to use the hibernate query language.

## a) THE FROM & Select Clause:

Create a method called **findUserHql(factory,session)** inside the UserController.java class. In this example, we will use the **“From User”** & **“Select”** clause to get all of the records.

| public static void findUserHql(SessionFactory factory,Session session) {  String hqlFrom = "FROM User"; *// Example of HQL to get all records of user class*  String hqlSelect = "SELECT u FROM User u";  TypedQuery<User> query = session.createQuery(hqlFrom, User.class);  List<User> results = query.getResultList();  System.*out*.printf("%s%13s%17s%34s%n","|User Id","|Full name","|Email","|Password");  for (User u:results) {  System.*out*.printf(" %-10d %-20s %-30s %s %n", u.getId(), u.getFullName(), u.getEmail(), u.getPassword());  }  } |
| --- |

Call the **findUserHql(factory,session)** and run the **UserContorller.java** class. You will see the result on your console as shown below:

## 

## b) SELECT Clause

Use the “**Select”** clause as shown below:

| String hqlSelect = "SELECT u FROM User u"; *// Example of HQL to get all records of user class using select*  TypedQuery<User> query = session.createQuery(hqlFrom, User.class);  Change the session.createQuery string parameter to “**hqlSelect**” |
| --- |

Call the **findUserHql(factory,session)** and run the **UserContorller.java** class. You will see the result on your console as shown below:

## 

## c) WHERE Clause and Order by Clause

Create a method called “**getRecordbyId(factory,session)”** inside the UserController.java class.In this method, we will use the “**Where and Order by”** clause as shown below:

| public static void getRecordById(SessionFactory factory, Session session) {  String hql = "FROM User u WHERE u.id > 2 ORDER BY u.salary DESC";  TypedQuery<User> query = session.createQuery(hql, User.class);  List<User> results = query.getResultList();  System.*out*.printf("%s%13s%17s%34s%21s%n", "|User Id", "|Full name", "|Email", "|Password", "|Salary");  for (User u : results) {  System.*out*.printf(" %-10d %-20s %-30s %-23s %s %n", u.getId(), u.getFullName(), u.getEmail(), u.getPassword(), u.getSalary());  }  } |
| --- |

Call the **getRecordById(factory,session)** and run the **UserContorller.java** class. You will see the result on your console as shown below. Note kthat your result may differ if you added or deleted records.



### 

## 

## d) Multiple SELECT Expressions

Create a method called "**getRecords()''** inside the UserController.java class.In this method, we will use the multiple columns as shown below:

| public static void getRecords (Session session) {  TypedQuery<Object[]> query = session.createQuery(  "SELECT U.salary, U.fullName FROM User AS U", Object[].class);  List<Object[]> results = query.getResultList();  System.*out*.printf("%s%13s%n","Salary","City");  for (Object[] a : results) {  System.*out*.printf("%-16s%s%n",a[0],a[1]);  }  } |
| --- |

Call the **getRecords(session)** and run the **UserContorller.java** class. You will see the result on your console as shown below:

### 

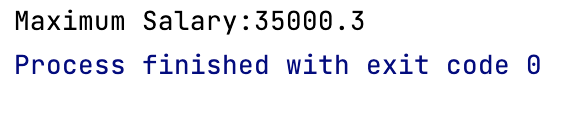
## e) Example: Aggregate Function

Create a method called "**getMaxSalary()”** inside the UserController.java class.In this method, we will use the “**max()”** function as shown below:

**Note**: We will use the **getSingleResult() method**. This method executes a SELECT query that returns a single untyped result.

| public static void getMaxSalary(Session session) {  String hql = "SELECT max(U.salary) FROM User U";  TypedQuery<Object> query = session.createQuery(hql,Object.class);  Object result = query.getSingleResult();  System.*out*.printf("%s%s","Maximum Salary:",result);  } |
| --- |

Call the **getMaxSalary(session)** and run the **UserContorller.java** class. You will see the result on your console as shown below:



Here is another aggregate function example supported by HQL:

| String hqlCount = "SELECT COUNT(\*) FROM User U";  List<Object> results = session.createQuery(hqlCount,Object.class).getResultList();  System.*out*.println("Count:"+results); |
| --- |

## f) GROUP BY Clause and Aggregate function

Create a method called "**getMaxSalaryGroupBy()”** inside the UserController.java class.In this method, we will use the **Group By** clause as shown below:

| **public void getmaxSalaryGroupBy() {  SessionFactory factory = new Configuration().configure().buildSessionFactory();  Session session = factory.openSession();  String hql = "SELECT SUM(U.salary), U.city FROM User U GROUP BY U.city";  TypedQuery query = session.createQuery(hql);  List<Object[]> result =query.getResultList();  for (Object[] o : result) {  System.out.println("Total salary " +o[0] +" | city: "+ o[1] );  } }** |
| --- |

Call the"**getMaxSalaryGroupBy(session)”** and run the **UserContorller.java** class. You will see the result on your console as shown below:

## 

# Using Named Parameters Syntax

Hibernate supports named parameters in its HQL queries. This makes writing HQL queries that accept input from the user easy, and the user does not have to defend against SQL injection attacks. Following is the syntax of using named parameters:

*You will learn about the* ***Named Parameters query*** *in upcoming lectures.*

| public static void namedQueryExample(Session session) {  String hql = "FROM User u WHERE u.id = :id";  TypedQuery<User> query = session.createQuery(hql, User.class);  query.setParameter("id", 2);  List<User> result = query.getResultList();  System.*out*.printf("%s%13s%17s%34s%21s%n", "|User Id", "|Full name", "|Email", "|Password", "|Salary");  for (User u : result) {  System.*out*.printf(" %-10d %-20s %-30s %-23s %s %n", u.getId(), u.getFullName(), u.getEmail(), u.getPassword(), u.getSalary());  }  } |
| --- |

Call the"**namedQueryExample(Session session)”** and run the **UserContorller.java** class. You will see the result on your console as shown below:

****

**Submission Instructions:**

Include the following deliverables in your submission -

* + Submit your source code or screenshot using the Start Assignment button in the top-right corner of the assignment page in Canvas.

## 