**Guided LAB 305.4.2A - Demonstration of Hibernate Query Language - HQL**

**Lab Overview**

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

**Lab Objective:**

By the end of this lab, learners will be able to use Hibernate Query Language - HQL.

# Example one: HQL to get all of the records.

## a) FROM Clause:

Create a class named **findUser\_Hql.java** under the **com.test.hib.controller** package and add the following code. In the example, we will use the **“From User”** clause.

| import java.util.List; import org.hibernate.Query; import org.hibernate.Session; import org.hibernate.SessionFactory; import org.hibernate.Transaction; import org.hibernate.cfg.Configuration; import com.test.hib.model.User; public class findUser\_Hql {  public void findUser() {  SessionFactory factory = new Configuration().configure().buildSessionFactory();  Session session = factory.openSession();  String hql = "FROM User"; // Example of HQL to get all records of user class  TypedQuery query = session.createQuery(hql);  List<User> results = query.getResultList();  for (User u : results) {  System.out.println("User Id: " +u.getId() + "|" + " Full name:" + u.getFullname() +"|"+ "Email: " + u.getEmail() +"|"+ "password" + u.getPassword());  }  } } |
| --- |

**For the Eclipse IDE:** Open the **App.java** class. You can find it under the ***com.test.project.HibernateJPABeginner package.***

**For the IntelliJ IDE:** You have to create an **App.java** class. You can create it anywhere in your project.

In the **App.java** class, create an object of the **findUser\_Hql** class and call the **findUser()** method as shown below:

**App.java class**

| **import com.test.hib.controller.findUser\_Hql; public class App  {   public static void main( String[] args )  {  findUser\_Hql u = new findUser\_Hql();  u.findUser();  } }** |
| --- |

Run the **App.java** class. You will see the result on your console as shown below:

## 

## b) SELECT Clause

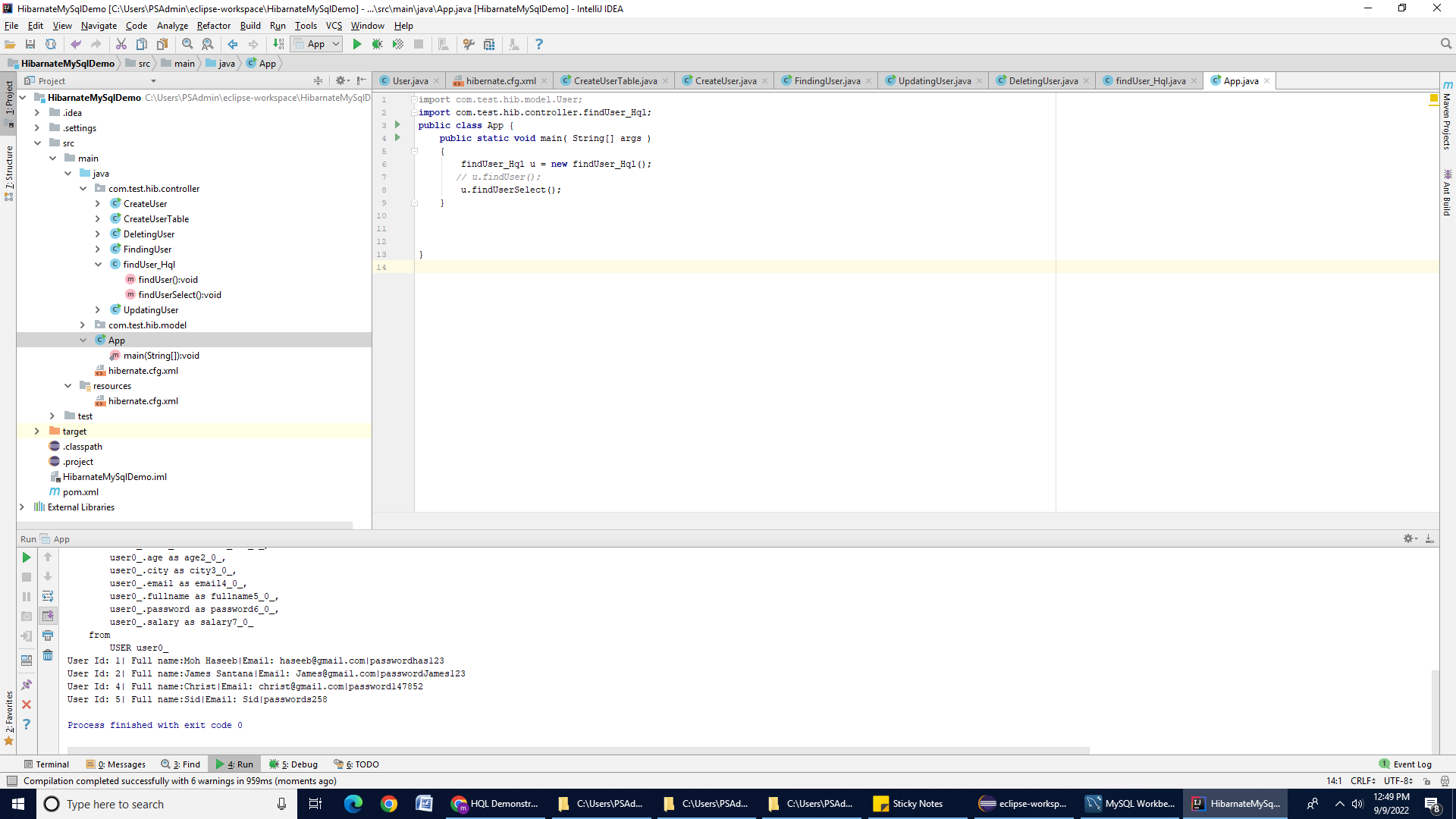
Create one more method named **findUserSelect()** in the **findUser\_Hql** class**.** In this method, we will use the “**Select”** clause as shown below:

| **public void findUserSelect()  {  SessionFactory factory = new Configuration().configure().buildSessionFactory();  Session session = factory.openSession(); /\* ------------ Example of HQL to get all the records------- \*/  String hql = "SELECT u FROM User u";   Query query = session.createQuery(hql);  List<User> list = query.getResultList();    for (User u : list) {  System.out.println("User Id: " +u.getId() + "|" + " Full name:" + u.getFullname() +"|"+ "Email: " + u.getEmail() +"|"+ "password" + u.getPassword());   }   }** |
| --- |

Call **findUserSelect()** method in the **App.java** class (main method) as shown below:

| **public class App  {   public static void main( String[] args )  {  findUser\_Hql u = new findUser\_Hql();  *// u.findUser();***  **u.findUserSelect();  } }** |
| --- |

Run the***App.java*** class. You will see the result on your console as shown below:

****

## c) `WHERE` Clause and Order by Clause

Create one more method named “**getRecordbyId()”** in **findUser\_Hql** class**.** In this method, we will use the “**Where and Order by”** clause as shown below:

| **public void getRecordbyId() {**  SessionFactory factory = new Configuration().configure().buildSessionFactory(); **Session session = factory.openSession();  String hql = "FROM User E WHERE E.id > 2 ORDER BY E.salary DESC";  TypedQuery query = session.createQuery(hql);  List<User> results = query.getResultList();  for (User u : results) {  System.out.println("User Id: " +u.getId() + "|" + " Full name:" + u.getFullname() +"|"+ "Email: " + u.getEmail() +"|"+ "password" + u.getPassword());  }**  **}** |
| --- |

Call the **getRecordbyId()** method in **App.Class** (main class) as shown below:

| **public class App  {   public static void main( String[] args )  {  findUser\_Hql u = new findUser\_Hql();  *// u.findUser();***  ***// u.findUserSelect();***  **u.getRecordbyId();  } }** |
| --- |

Run the **App.java** class. You will see the result on your console as shown below.

### 

## 

## d) Multiple SELECT Expressions

Create one more method named "**getrecords()''** in the **findUser\_Hql** class**.** In this method, we will use the multiple columns as shown below:

| public void getrecords() {  SessionFactory factory = new Configuration().configure().buildSessionFactory();  Session session = factory.openSession();  TypedQuery<Object[]> queryy = session.createQuery(  "SELECT U.salary, U.fullname FROM User AS U", Object[].class);  List<Object[]> resultss = queryy.getResultList();  for (Object[] a : resultss) {  System.out.println("Salary: " + a[0] + ", City: " + a[1]);  } } |
| --- |

Call the **getrecords()** method in **App.Class** (main class) as shown below:

| **public class App  {   public static void main( String[] args )  {  findUser\_Hql u = new findUser\_Hql();  *// u.findUser();***  ***// u.findUserSelect();***  ***// u.getRecordbyId();***  **u.getrecords();  } }** |
| --- |

Run the **App.java** class. You will see the result on your console as shown below.

### 

## e) Example: Aggregate Function

Create one more method named "**getmaxSalary()”** in the **findUser\_Hql** class**.** In this method, we will use the “**max()”** function as shown below:

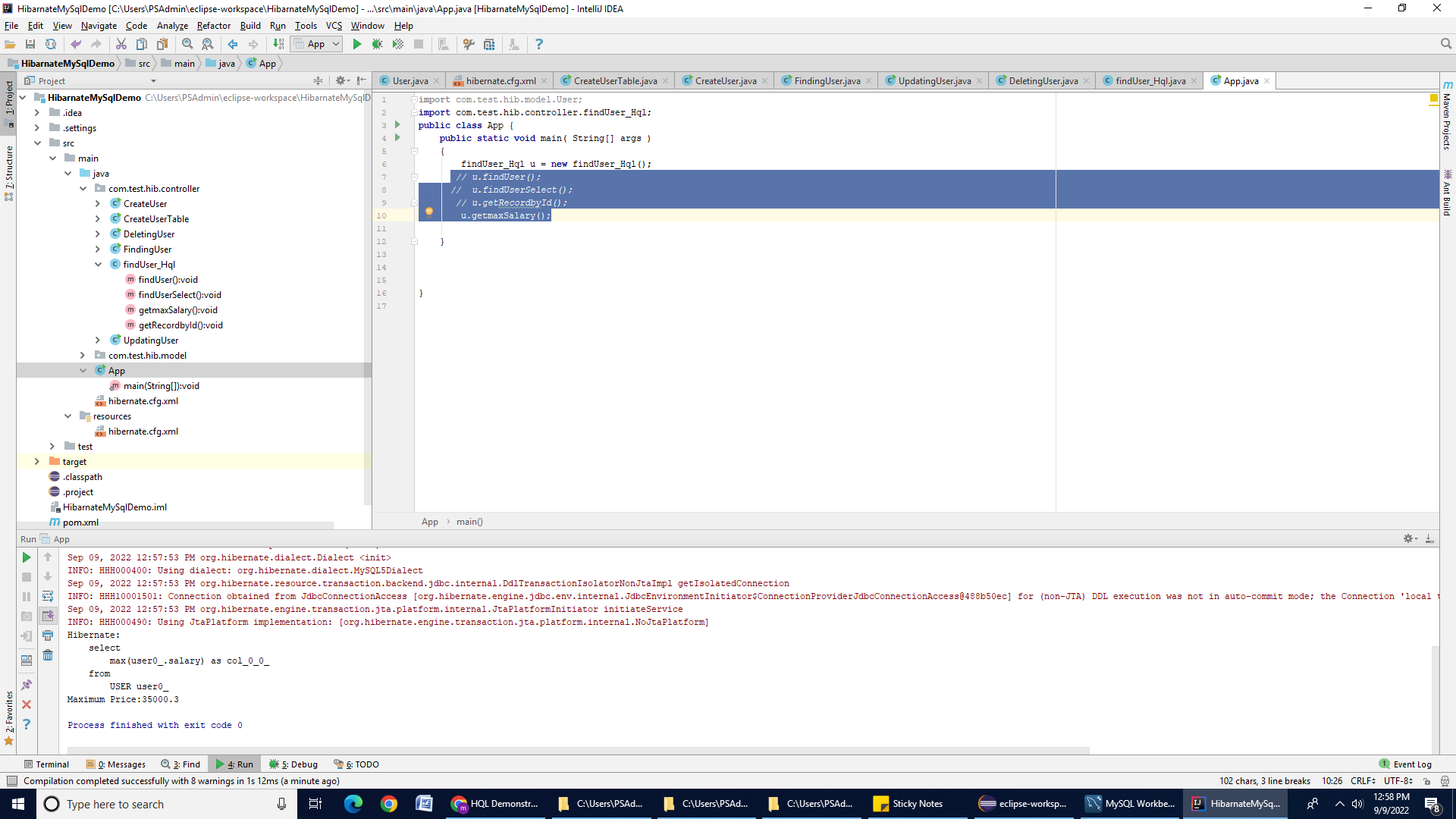
**Note**: We will use the **getSingleResult() method**. This method executes a SELECT query that returns a single untyped result.

| **public void getmaxSalary() {  SessionFactory factory = new Configuration().configure().buildSessionFactory();  Session session = factory.openSession();  String hql = "SELECT max(U.salary) FROM User U";  TypedQuery query = session.createQuery(hql);  double result =(double) query.getSingleResult();  System.out.println(result); }** |
| --- |

Call the **getmaxSalary()** method in **App.Class** (main class ) as shown below:

| **public class App  {   public static void main( String[] args )  {  findUser\_Hql u = new findUser\_Hql();**  ***// u.findUser();***  ***// u.findUserSelect();***  ***// u.getRecordbyId();***  **u.getmaxSalary();  } }** |
| --- |

Run the **App.java** class. You will see the result on your console, as shown below.

****

Here is another aggregate function example supported by HQL:

| **String hql = "SELECT COUNT(\*) FROM User U"; List results = session.createQuery(hql).getResultList();  System.out.println(results);** |
| --- |

## f) GROUP BY Clause and Aggregate function

Create one more method named "**getmaxSalaryGroupBy()”** in the **findUser\_Hql** class**.** In this method, we will use the **Group By** clause and “**max()” function** 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()** method in **App.Class** (main class ) as shown below:

| **public class App  {   public static void main( String[] args )  {  findUser\_Hql u = new findUser\_Hql();**  ***// u.findUser();***  ***// u.findUserSelect();***  ***// u.getRecordbyId();***  ***// u.getmaxSalary();***  **u.getmaxSalaryGroupBy();  } }** |
| --- |

Run the **App.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 void NamedQueryExample()  {  SessionFactory factory = new Configuration().configure().buildSessionFactory();  Session session = factory.openSession();  String hql = "FROM User u WHERE u.id = :id";  TypedQuery query = session.createQuery(hql);  query.setParameter("id",2);  List<User> result = query.getResultList();  for (User u : result) {  System.out.println("User Id: " +u.getId() + "|" + " Full name:" + u.getFullname() +"|"+ "Email: " + u.getEmail() +"|"+ "password" + u.getPassword()); **}** |
| --- |

**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.

## 