**Learning to write queries in HQL**

**1.1 Introduction to HQL**

Previously, you got acquainted with Hibernate, and now I will introduce you to **HQL** , aka **Hibernate Query Language** . In fact, this is SQL converted for writing queries in Hibernate. It has several key differences.

1. Using **the class name** instead of the table name.
2. Using **the class field name** instead of the table column name.
3. **Optional** use of select.

Let's ask Hibernate to return to us all the users that it has in the database. Here's what that request would look like:

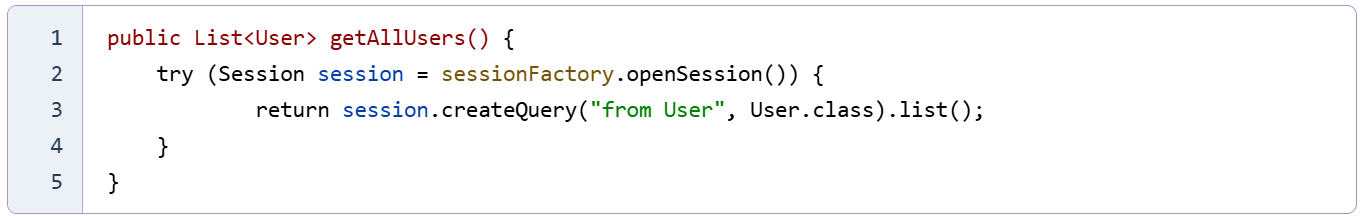


That's all, for comparison, we present a similar query in SQL:



Here User is the name of the class, and useris the name of the table.

The complete Java code will look like this:



Otherwise, HQL is very similar to SQL - it also has operators:

* WHERE
* ORDER BY
* GROUP BY
* HAVING

**1.2 Example of working with HQL**

Perhaps the previous example is a bit confusing due to the same table and field names. Let's come up with a special example where this would be easier.

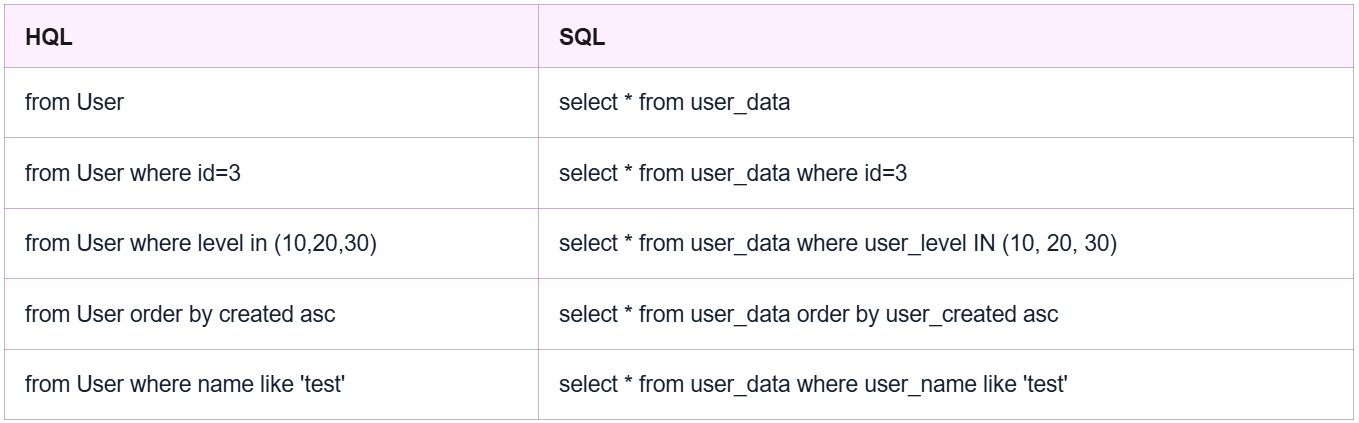
Let's say we have a **user\_data** table that contains the following fields:

* **id** INT
* **username** VARCHAR(100)
* **user\_level** INT
* **user\_created** DATE

We will create a Java class that will map to this table:



Now let's write some examples:

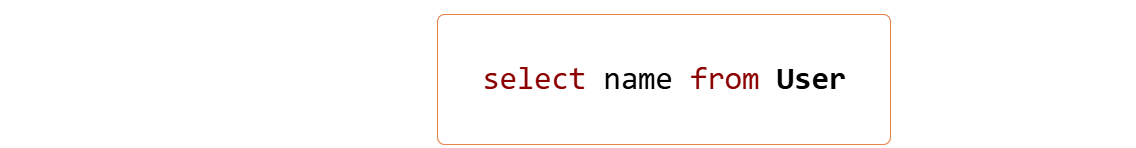


The queries are very similar, and reading HQL queries when you are familiar with class names and their fields is just as easy as reading SQL queries. It may be a little more difficult to write, but then again, very complex queries are rarely written in HQL.

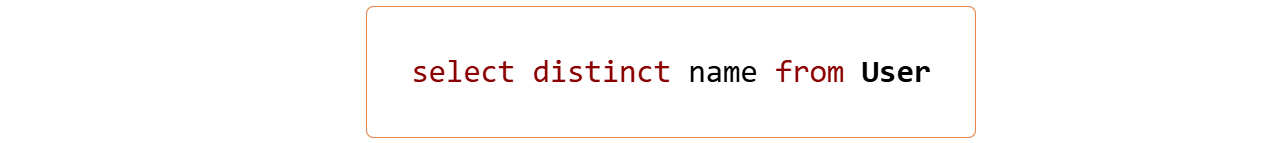
**1.3 Using select**

In HQL, you can use selectwhen the data type of the result does not match the type specified in from.

For example, we want to get the names of all users that are in our **user\_data** table , then we need to write the following query:



Also, if there are duplicates among the names, then you can use the operator DISTINCT:



Aliases work the same as in SQL:



Well, completely in the form of Java code, this request will look like this:

