

Hibernate Tip: User-defined function in a CriteriaQuery

Question:

How can I call a user-defined database function in the *WHERE* clause of my *CriteriaQuery*?

Solution:

Since JPA 2.1, you can use the *function(String name, Class type, Expression... args)* method of the *CriteriaBuilder* to call user-defined or database-specific functions.

You need to provide the name and the expected result type of the function as the first 2 parameters and you can provide one or more *Expression* which will be used as function arguments.

The following code snippet shows an example that calls the custom database function *calculate*.

The function returns a *Double* and I provide the *price* of the *Book* and a query parameter as function arguments.

```
CriteriaBuilder cb = em.getCriteriaBuilder();
CriteriaQuery cq = cb.createQuery(Book.class);
Root root = cq.from(Book.class);

// call the database function calculate
ParameterExpression doubleParam1 =
cb.parameter(Double.class);
ParameterExpression doubleParam2 =
cb.parameter(Double.class);
cq.where(cb.greaterThan(doubleParam2,
cb.function("calculate", Double.class, root.get(Book_.price),
doubleParam1)));
```

Hibernate Tip: User-defined function in a CriteriaQuery

```
TypedQuery q = em.createQuery(cq);  
q.setParameter(doubleParam1, 10.0D);  
q.setParameter(doubleParam2, 40.0D);  
List books = q.getResultList();
```

You can use this approach in the *WHERE* clause to call all functions supported by your database.

You can also use the function *function* in the *SELECT* clause of your query. But you then need to register the database function so that Hibernate knows its result type. This makes the function *function* superfluous because you can use all registered functions directly in your query.

Source Code

Get the source code from [GitHub](#)

Learn more:

You can also use the function *function* in a JPQL query as I show you in this post: [How to call custom database functions with JPA and Hibernate.](#)