Hibernate Tip: Map native queries to entities

Question:

My query is too complex for JPQL, and I have to use a native query. Is there a way to map the result of the query to managed entities?

Solution:

If your query returns all columns that are mapped by an entity, you can tell Hibernate to map the result to a managed entity. Afterwards, you can use the entity in the same way as any other entity.

There are 2 options to define the mapping:

1. You can use an implicit mapping if your query result uses the same column names as your entity mapping.
2. You can create your own mapping if the column names do not match the entity mapping

Implicit Mapping

The implicit mapping is easier to use and the better approach for most use cases. You only need to provide the class of the entity as the second parameter to the *createNativeQuery* method.

```
Book b = (Book) em.createNativeQuery(
                                "SELECT * FROM book b WHERE id = 1",
                                Book.class).getSingleResult();
```

Explicit Mapping

If the column names of your query result do not match the column names of your entity mapping, you have to define the mapping yourself.

You can do this with a *@SqlResultSetMapping* which specifies the mapping for each entity attribute.

```
@SqlResultSetMapping(
        name = "BookMapping",
        entities = @EntityResult(
                entityClass = Book.class,
                fields = {
                        @FieldResult(name = "id", column = "id"),
                        @FieldResult(name = "version",
                                     column = "version"),
                        @FieldResult(name = "title", column = "title"),
                        @FieldResult(name = "publishingDate",
                                     column = "publishingDate"),
                        @FieldResult(name = "publisher",
                                     column = "publisherid")}))
```

As you can see in the code snippet, the *@SqlResultSetMapping* requires a name and an *@EntityResult* annotation which defines the mapping to the entity.

You, therefore, need to specify the class of the entity and a set of *@FieldResult* annotation which define the mapping between the result set column and the entity attribute.

You can then use this mapping by providing its name as the 2nd parameter to the *createNativeQuery* method.

```
em.createNativeQuery(
                        "SELECT * FROM book b WHERE id = 1",
                        "BookMapping").getSingleResult();
```

Learn more

@SqlResultSetMappings are a powerful feature which allows you to define complex mappings for native query results. You can also use it to map the query result to multiple entities or POJOs.