

Hibernate Tip: Map A View With Hibernate

Question:

How can I map a read-only database view with Hibernate?

Solution:

Database views, in general, are mapped in the same way as database tables. You just have to define an entity that maps the view with the specific name and one or more of its columns. But the normal table mapping is not read-only, and you can use the entity to change its content.

Depending on the database you use and the definition of the view, you're not allowed to perform an update on the view content. You should therefore also prevent Hibernate from updating it.

You can easily achieve this with the Hibernate-specific *@Immutable* annotation.

```
@Entity
@Immutable
public class BookView {

    ...

}
```

The *@Immutable* annotation tells Hibernate to ignore all changes on this entity, but you can use it to retrieve data from the database.

```
List<BookView> bvs = em.createQuery(
    "SELECT v FROM BookView v", BookView.class)
    .getResultList();
```

Hibernate Tip: Map A View With Hibernate

Learn more:

Views can be a nice way to pre-process the existing data in the database to improve the performance of your application. You can learn more about performance optimizations in my [Hibernate Performance Tuning Online Training](#).