Hibernate Tip: Cache Query Results

Question:

Hibernate does not use the first- and second-level for queries. Is there any way to cache the result of a query?

Solution:

Hibernate also supports the *QueryCache*, which can store the result of a query.

You need to activate it in the *persistence.xml* file by setting the parameter *hibernate.cache.use_query_cache* to *true* and defining a *hibernate.cache.region.factory_class*.

```
<persistence
        <persistence-unit name="my-persistence-unit">
                …
                <properties>
                        <!--  configure caching -->
                        <property name =
                                        "hibernate.cache.use_query_cache"
                                value = "true"/>
                        <property name =
                                        "hibernate.cache.region.factory_class"
                                value = "org.hibernate.cache.ehcache.
                                                EhCacheRegionFactory"/>
                </properties>
        </persistence-unit>
</persistence>
```

And you also need to activate caching for the specific query of which you want to cache the results by calling the *setCacheable* of the Hibernate-specific *Query* interface with the parameter true.

```
Session s = (Session) em.getDelegate();
Query q = s.createQuery(
                        "SELECT a FROM Author a WHERE id = :id");
q.setParameter("id", 1L);
q.setCacheable(true);
log.info(q.uniqueResult());
```

Learn more:

I get into more details about Hibernate's 3 different kinds of caches in the Hibernate Performance Tuning Online Training.