Hibernate Tip: Increase the Version of the Parent Entity

Question:

I'm using optimistic locking to avoid concurrent modifications of the same entity. But Hibernate's versioning mechanism ignores changes on the one-to-many association. How can I tell Hibernate to increase the version number when I add or remove a child entity?

Solution:

The version check defined by section 3.4.2. of the JPA specification explicitly excludes relationships that are not owned by the entity.

> *"The version attribute is updated by the persistence provider runtime when the object is written to the database. All non-relationship fields and properties and all relationships owned by the entity are included in version checks."*

By default, the child or to-many site of the relationship owns the one-to-many association. Hibernate, therefore, doesn't increment the version number of the of parent entity when you add or remove a child entity.

But you can use *LockModeType.OPTIMISTIC_FORCE_INCREMENT* to trigger the version update programmatically.

You can choose between 2 options to set the *LockModeType* for a specific entity. You either call the *lock* method on the *EntityManager* or you perform a JPQL query and provide the *LockModeType.OPTIMISTIC_FORCE_INCREMENT* to the *setLockMode* method.

Let's take a look at the first option.

Increase the Version Number With the EntityManager.lock Method

The *EntityManager.lock* method locks a managed entity. So, I first call the *find* method to load the parent entity by its primary key and to get a managed entity. You can skip this step if your parent entity is already managed. And then I call the *lock* method with the parent entity and the *LockModeType.OPTIMISTIC_FORCE_INCREMENT*.

```
Book parent = em.find(Book.class, 1L);
em.lock(parent,
LockModeType.OPTIMISTIC_FORCE_INCREMENT);
```

When you activate the logging of the executed SQL statements, you can see that Hibernate performs an SQL SELECT statement to read the *Book* entity before it performs an SQL UPDATE statement to increase its version number.

```
08:48:41,902 DEBUG [org.hibernate.SQL] - select book0_.id
as id1_0_0_, book0_.title as title2_0_0_, book0_.version as
version3_0_0_ from Book book0_ where book0_.id=?
08:48:41,919 TRACE
[org.hibernate.type.descriptor.sql.BasicBinder] - binding
parameter [1] as [BIGINT] - [1]
08:48:41,939 TRACE
[org.hibernate.type.descriptor.sql.BasicExtractor] - extracted
value ([title2_0_0_] : [VARCHAR]) - [Hibernate Tips]
08:48:41,940 TRACE
[org.hibernate.type.descriptor.sql.BasicExtractor] - extracted
value ([version3_0_0_] : [INTEGER]) - [0]
08:48:42,003 DEBUG [org.hibernate.SQL] - update Book set
version=? where id=? and version=?
08:48:42,005 TRACE
[org.hibernate.type.descriptor.sql.BasicBinder] - binding
parameter [1] as [INTEGER] - [1]
```

```
08:48:42,006 TRACE
[org.hibernate.type.descriptor.sql.BasicBinder] - binding
parameter [2] as [BIGINT] - [1]
08:48:42,007 TRACE
[org.hibernate.type.descriptor.sql.BasicBinder] - binding
parameter [3] as [INTEGER] - [0]
```

Increase the Version With a JPQL Query

You can do the same with a JPQL or Criteria query. The *Query* and *TypedQuery* interface provide the *setLockMode* method.

You can use this method to lock the selected entities. In this case, I don't set a database lock but activate the *LockModeType.OPTIMISTIC_FORCE_INCREMENT* to increase the version number of the selected entity.

```
TypedQuery q = em.createQuery("SELECT b FROM Book b
WHERE b.id = 1", Book.class);
q.setLockMode(LockModeType.OPTIMISTIC_FORCE_INCR
EMENT);
Book b = q.getSingleResult();
```

As in the previous example, you can see in the log output that Hibernate selects the *Book* entity before it increments its version number.

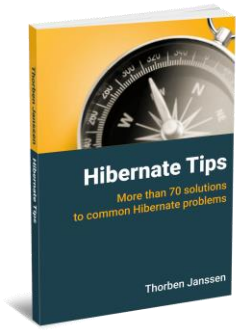
```
08:51:31,314 DEBUG [org.hibernate.SQL] - select book0_.id
as id1_0_, book0_.title as title2_0_, book0_.version as
version3_0_ from Book book0_ where book0_.id=1
08:51:31,327 TRACE
[org.hibernate.type.descriptor.sql.BasicExtractor] - extracted
value ([id1_0_] : [BIGINT]) - [1]
08:51:31,343 TRACE
[org.hibernate.type.descriptor.sql.BasicExtractor] - extracted
value ([title2_0_] : [VARCHAR]) - [Hibernate Tips]
```

```
08:51:31,347 TRACE
[org.hibernate.type.descriptor.sql.BasicExtractor] - extracted
value ([version3_0_] : [INTEGER]) - [0]
08:51:31,395 DEBUG [org.hibernate.SQL] - update Book set
version=? where id=? and version=?
08:51:31,397 TRACE
[org.hibernate.type.descriptor.sql.BasicBinder] - binding
parameter [1] as [INTEGER] - [1]
08:51:31,398 TRACE
[org.hibernate.type.descriptor.sql.BasicBinder] - binding
parameter [3] as [INTEGER] - [0]
```

Learn more

You can learn more about optimistic and pessimistic locking and their performance impact in my Hibernate Performance Tuning Online Training.

Hibernate Tips Book



Get more recipes like this one in my book Hibernate Tips: More than 70 solutions to common Hibernate problems.

It gives you more than 70 ready-to-use recipes for topics like basic and advanced mappings, logging, Java 8 support, caching and statically and dynamically defined queries.