

Hibernate Tip: Remove entities from the persistence context

Question:

How do you remove entities from the persistence context before performing bulk update operations with JPQL or SQL?

Solution:

As I explained in [one of my previous posts](#) and in my [Hibernate Performance Tuning training](#): When you need to update or delete a huge number of entities, it is often better to use a JPQL or SQL statement instead of updating one entity after the other. That reduces a number of required queries from *way too many* (at least one for each updated entity) to 1.

But the JPQL or SQL statement also has a disadvantage. Hibernate doesn't know which database records you're updating and can't replicate the changes to the entities stored in the persistence context or 1st level cache. You, therefore, need to do that yourself and remove all entities from the persistence context before you execute the query.

That's when I always get the same question: How do you remove an entity from the persistence context?

You can do that in 2 steps:

1. Making sure that Hibernate persisted all pending changes in the database
2. Removing the entities from the persistence context

Step 1 is crucial because Hibernate delays the execution of all write operations as long as possible. Your persistence context might contain new entities which were not inserted into the database or dirty entities that require the execution of an SQL update statement. When you remove these entities from the persistence context, Hibernate will not perform the required SQL statements, and you will lose these pending changes. You can prevent that by calling the *flush* method on the *EntityManager*.

Hibernate Tip: Remove entities from the persistence context

After you've done that, you can remove a specific entity from the persistence context by calling the *detach* method or you can call the *clear* method to clear the persistence context completely.

```
// Write all pending changes to the DB
em.flush();

// Remove myAuthor from the persistence context
em.detach(myAuthor);

// or

// Remove all entities from the persistence context
em.clear();

// Perform the bulk update
Query query = em.createQuery("UPDATE Book b SET
b.price = b.price*1.1");
query.executeUpdate();
```

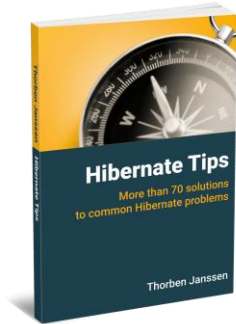
Learn more

You can learn more about bulk update operations in the following posts:

- [How to use native queries to perform bulk updates](#)
- [Criteria Update/Delete – The easy way to implement bulk operations with JPA](#)

Hibernate Tip: Remove entities from the persistence context

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.