

# Hibernate Tips: Delete a child entity from a relationship

## Question:

My domain model contains a many-to-one association in which the child entity can't exist without its parent. Can Hibernate automatically delete the child entity if I remove its association to the parent?

## Solution:

Yes, the JPA specification provides the *orphanRemoval* feature for these use cases. When you activate it on the association, Hibernate removes a child entity when you remove its association to the parent entity.

Let's take a look at an example.

In most online book stores, customers can review the offered books. You can model that with a *Book* and a *Review* entity and a one-to-many association between them.

It doesn't make any sense to store a *Review* that's not associated with a *Book*. So, you should set the *orphanRemoval* attribute of the *@OneToMany* association to true. And you also need to set the *cascade* attribute to *CascadeType.PERSIST* or *CascadeType.ALL*.

```
@Entity
public class Book {

    @OneToMany(mappedBy = "book",
              orphanRemoval = true,
              cascade = CascadeType.PERSIST)
    private List<Review> reviews =
        new ArrayList<Review>();

    ...
}
```

# Hibernate Tips: Delete a child entity from a relationship

When you now remove a Review entity from the association that's modeled by the List reviews attribute, Hibernate will delete the Review entity from the database.

```
em = emf.createEntityManager();
em.getTransaction().begin();

Book b = em.find(Book.class, 1L);
b.getReviews().remove(0);

em.getTransaction().commit();
em.close();
```

As you can see in the following log messages, Hibernate performs 2 SQL SELECT statements to get the Book entity with id 1 and all associated Review entities. And when I commit the transaction, it also performs an SQL DELETE operation to remove the Review entity from the database.

```
14:12:57,197 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=?
14:12:57,201 DEBUG [org.hibernate.SQL] - select
reviews0_.fk_book as fk_book3_1_0_, reviews0_.id as
id1_1_0_, reviews0_.id as id1_1_1_, reviews0_.fk_book as
fk_book3_1_1_, reviews0_.comment as comment2_1_1_
from Review reviews0_ where reviews0_.fk_book=?
14:12:57,212 DEBUG [org.hibernate.SQL] - delete from
Review where id=?
```

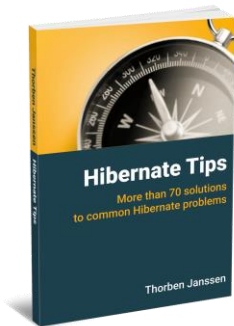
# Hibernate Tips: Delete a child entity from a relationship

## Learn more

If you want to learn more about association mappings with JPA and Hibernate, you should also take a look at the following posts:

- I wrote a broader introduction to JPA's relationship mappings in [Ultimate Guide: Association Mappings with JPA and Hibernate](#)
- And if you're already familiar with the basic association mappings, you might be interested in this post about [general best practices for mapping one-to-many and many-to-one associations](#).

## 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.