

Issue with Merge method

```
/**
```

```
Complaint complaint =  
Complaint.builder().id(1).text("hello").tableName("doctor").type(1).build();  
    Patient patient = Patient.builder().id(1).name("Ansh  
Makkar").email("pat@gmail.com").build();  
    entityManager.merge(patient);  
    complaint.setPatient(patient);  
    entityManager.merge(complaint);
```

```
*/
```

If you do not store the managed ``patient`` instance returned by ``entityManager.merge(patient)`` and directly use the ``patient`` instance for setting the association with the ``complaint`` entity, you may encounter unexpected behaviour depending on how your JPA provider handles detached entities and entity state transitions.

Here are a couple of possibilities:

1. ****Duplicate Entities in the Database****: If you attempt to merge the ``complaint`` entity with a detached ``patient`` entity (i.e., without first ensuring that ``patient`` is managed), and if ``patient`` with the same identifier already exists in the database, you might end up with duplicate entries for the ``patient`` entity in the database. This happens because the persistence context treats the detached ``patient`` as a new entity during the merge operation, resulting in a new row being inserted into the database.
2. ****Association with Detached Entity****: If the JPA provider allows associations between managed and detached entities without throwing an exception, the ``complaint`` entity might end up with a reference to a detached ``patient`` entity. This can lead to issues if you attempt to perform operations or fetch data using this association, as the detached ``patient`` entity is not managed and might not have the expected behaviour.

To avoid such issues and ensure consistent behaviour, it's best practice to always use managed entities when setting associations between entities in JPA. This involves storing the managed instance returned by ``entityManager.merge()`` and using it for further operations or associations.