Hibernate Tip: Map an entity to multiple tables

Question:

I'm working with a legacy database, and I need to map an entity to 2 database tables. Is there any way to do that with JPA or Hibernate?

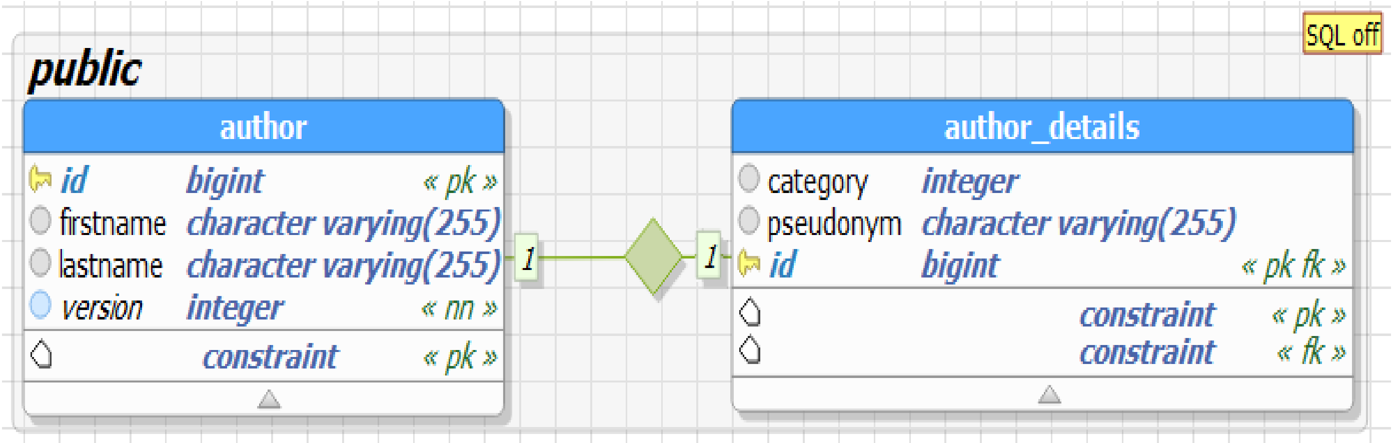
Solution:

Yes, you can map an entity to 2 database tables in 2 simple steps:

1. You need to annotate your entity with JPA's *@Table* and *@SecondaryTable* annotations and provide the names of the first and second table as the value of the *name* parameters.
2. You need to annotate each attribute which you want to map to the secondary table with a *@Column* annotation and set the name of the secondary table as the value of the *table* attribute.

Map the Author entity to 2 tables:

Let's take a look at a simple example that maps the *author* and the *author_details* table to the *Author* entity. Here are the 2 tables:



author		
id	bigint	« pk »
firstname	character varying(255)	
lastname	character varying(255)	
version	integer	« nn »
	constraint	« pk »

author_details		
category	integer	
pseudonym	character varying(255)	
id	bigint	« pk fk »
	constraint	« pk »
	constraint	« fk »

The following code maps these tables to the *Author* entity.

The *@Table* annotation defines the primary table to which the entity attributes get mapped by default. In this example, that's the case for the *id*, *version*, *firstName*, and *lastName* attributes.

The *@SecondaryTable* annotation specifies the second database table to which the entity gets mapped. It consists of the columns *id, pseudonym,* and *category.* You need to annotate the attributes that map these columns with an additional *@Column* annotation that provides the name of the secondary table.

```
@Entity
@Table(name = "author")
@SecondaryTable(name = "author_details")
public class Author {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(updatable = false, nullable = false)
    private Long id;

    @Version
    private int version;

    private String firstName;

    private String lastName;

    @Column(table = "author_details")
    private String pseudonym;

    @Column(table = "author_details")
    private Category category;

    ...
}
```

That’s all you need to do to map the 2 database tables to the *Author* entity. Every time you persist or update an *Author* entity, Hibernate writes the values of the *id*, *firstName*, *lastName,* and *version* attributes to the *author* table, and the values of the *id*, *pseudonym*, and *category* attributes to the *author_details* table.

And when you read an *Author* entity, Hibernate gets the attribute values from the same 2 tables.

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

Author a = new Author();
a.setFirstName("Thorben");
a.setLastName("Janssen");
a.setCategory(Category.NON_FICTION);
a.setPseudonym("Thorben Janssen");
em.persist(a);

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

As you can see in the log output, Hibernate uses the name and value of the primary key column of the primary table also as the name and value of the primary key column of the secondary table.

```
09:12:40,154 DEBUG [org.hibernate.SQL] -
    select
        nextval ('hibernate_sequence')
09:12:40,204 DEBUG [org.hibernate.SQL] -
    insert
    into
        author
        (firstName, lastName, version, id)
```

```
values
        (?, ?, ?, ?)
09:12:40,218 TRACE
[org.hibernate.type.descriptor.sql.BasicBinder] - binding
parameter [1] as [VARCHAR] - [Thorben]
09:12:40,218 TRACE
[org.hibernate.type.descriptor.sql.BasicBinder] - binding
parameter [2] as [VARCHAR] - [Janssen]
09:12:40,219 TRACE
[org.hibernate.type.descriptor.sql.BasicBinder] - binding
parameter [3] as [INTEGER] - [0]
09:12:40,222 TRACE
[org.hibernate.type.descriptor.sql.BasicBinder] - binding
parameter [4] as [BIGINT] - [1]
09:12:40,225 DEBUG [org.hibernate.SQL] -
    insert
    into
        author_details
        (category, pseudonym, id)
    values
        (?, ?, ?)
09:12:40,225 TRACE
[org.hibernate.type.descriptor.sql.BasicBinder] - binding
parameter [2] as [VARCHAR] - [Thorben Janssen]
09:12:40,226 TRACE
[org.hibernate.type.descriptor.sql.BasicBinder] - binding
parameter [3] as [BIGINT] - [1]
```

Customize the primary key columns:

The previous example didn’t specify the name of the primary key column in the secondary table. By default, Hibernate uses an identical mapping to map the primary key attribute to both tables. If you’re working with a legacy database, you might need to adapt this for the secondary table.

You can do that with the *pkJoinColumns* attribute of the *@SecondaryTable* annotation. It allows you to customize the mapping with one or more *@PrimaryKeyJoinColumn* annotations. Its *name* attribute specifies the name of the primary key column of the secondary table and the *referencedColumnName* attribute defines the name of the primary key column of the primary table.

```
@Entity
@Table(name = "author")
@SecondaryTable(
        name = "author_details",
        pkJoinColumns =
                @PrimaryKeyJoinColumn(
                        name = "authorId",
                        referencedColumnName = "id"))
public class Author { ... }
```

When you now persist a new *Author* entity, Hibernate uses the *authorId* column as the primary key column of the *author_details* table.

```
09:13:34,254 DEBUG [org.hibernate.SQL] -
    select
        nextval ('hibernate_sequence')
09:13:34,315 DEBUG [org.hibernate.SQL] -
    insert
    into
        author
        (firstName, lastName, version, id)
    values
        (?, ?, ?, ?)
09:13:34,321 TRACE
[org.hibernate.type.descriptor.sql.BasicBinder] - binding
parameter [1] as [VARCHAR] - [Thorben]
09:13:34,323 TRACE
[org.hibernate.type.descriptor.sql.BasicBinder] - binding
parameter [2] as [VARCHAR] - [Janssen]
09:13:34,324 TRACE
[org.hibernate.type.descriptor.sql.BasicBinder] - binding
parameter [3] as [INTEGER] - [0]
```

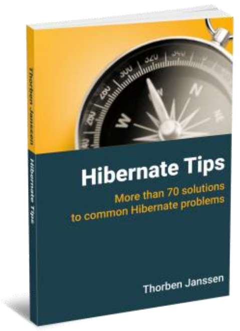
```
09:13:34,327 TRACE
[org.hibernate.type.descriptor.sql.BasicBinder] - binding
parameter [4] as [BIGINT] - [1]
09:13:34,330 DEBUG [org.hibernate.SQL] -
    insert
    into
        author_details
        (category, pseudonym, authorId)
    values
        (?, ?, ?)
09:13:34,331 TRACE
[org.hibernate.type.descriptor.sql.BasicBinder] - binding
parameter [2] as [VARCHAR] - [Thorben Janssen]
09:13:34,331 TRACE
[org.hibernate.type.descriptor.sql.BasicBinder] - binding
parameter [3] as [BIGINT] - [1]
```

Learn more

If you enjoyed this post, you might also be interested in the following posts about entity mappings:

- Hibernate Tips: How to Map Multiple Entities to the Same Table
- Hibernate Tips: How to escape table and column names
- Hibernate Tips: How to define schema and table names

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.