

Question:

I often need to provide a value that gets calculated based on an entity attribute, like the age of a person. How can I cache this calculated attribute in the entity without storing it in the database?

Solution:

There are different ways to provide a calculated value to the user of the entity:

1. You can use a *@Formula* to provide an SQL expression which returns the value.
2. You can use field access and calculate the value in a getter method.
3. You can use a *transient* entity attribute which stores calculated value without persisting it in the database.

I already explained option 1 in a previous post and will, therefore, focus on option 2 and 3.

Calculate in a getter method

Option 2 is easy to implement. When you use field access, you can add additional getter methods to your entity. You can for example, add a *getAge()* method that calculates and returns the age of the *Author*. This approach requires you to calculate the value for each call of the

Hibernate Tip: Cache preprocessed, non-persistent attributes

This approach requires you to calculate the value for each call of the *getAge()* method and is, therefore, not a good solution for complex calculations.

```
@Entity
public class Author {

    ...

    @Column
    private LocalDate dateOfBirth;

    public int getAge() {
        return Period.between(dateOfBirth,
            LocalDate.now()).getYears();
    }
}
```

Transient entity attributes

The transient attribute approach requires a little more code but allows you to store the calculated value in an entity attribute without persisting it. The *@Transient* annotation tells Hibernate, and any other JPA implementation, to ignore the attribute when writing or reading the entity. You can, therefore, use it in your *getAge()* method to store the calculation result after it was calculated. This can be useful, if you need to perform a complex calculation that you don't want to repeat for each call.

Hibernate Tip: Cache preprocessed, non-persistent attributes

When Hibernate fetches an Author entity from the database, it adds the SQL snippet of the formula annotation to its SQL statement.

```
@Entity
public class Author {

    ...

    @Column
    private LocalDate dateOfBirth;

    @Transient
    private Integer age;

    ...

    public int getAge() {
        if (this.age == null) {
            this.age = Period.between(dateOfBirth,
LocalDate.now()).getYears();
        }

        return age;
    }
}
```

Hibernate Tip: Cache preprocessed, non-persistent attributes

Learn more:

You can also use `@ColumnTransformer` and `AttributeConverter` if you want to replace the presentation of an attribute.