

# Hibernate Tip: Map generated values

## Question:

The database administrator set up a trigger to generate the value of a database column. How can map this column with Hibernate so that I can use the generated value within my application?

## Solution:

You can annotate an entity attribute with `@Generated(GenerationTime value)`, to tell Hibernate that the database will generate the value of the attribute. The `GenerationTime` Enum tells Hibernate when the database will generate the value. It can either do this `NEVER`, only on `INSERT` or `ALWAYS` (on insert and update). Hibernate will then execute an additional query to retrieve the generated value from the database.

The following code snippet shows an example for such an entity mapping.

```
@Entity
public class Author {

    @Column
    @Generated(GenerationTime.ALWAYS)
    private LocalDateTime lastUpdate;

    ...

}
```

## Hibernate Tip: Map generated values

As you can see in the following log output, Hibernate now performs an additional query for each insert and update statement to retrieve the generated value.

```
10:33:49,612 DEBUG [org.hibernate.SQL] – insert into
Author (firstName, lastName, version, id) values (?, ?, ?,
?)
```

```
10:33:49,620 DEBUG [org.hibernate.SQL] – select
author_.lastUpdate as lastUpda4_0_ from Author
author_ where author_.id=?
```

```
10:33:49,644 DEBUG [org.hibernate.SQL] – update
Author set firstName=?, lastName=?, version=? where
id=? and version=?
```

```
10:33:49,646 DEBUG [org.hibernate.SQL] – select
author_.lastUpdate as lastUpda4_0_ from Author
author_ where author_.id=?
```