4 options to generate primary keys

The JPA specification supports 4 different primary key generation strategies which generate the primary key values programmatically or use database features, like auto-incremented columns or sequences.

GenerationType.AUTO

The *GenerationType.AUTO* is the default generation type and lets the persistence provider choose the generation strategy.

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

GenerationType.IDENTITY

The *GenerationType.IDENTITY* is easiest to use but not the best one from a performance point of view. It relies on an auto-incremented database column and lets the database generate a new value with each insert operation.

It requires Hibernate to perform the insert statement immediately and prevents it from using other performance optimization techniques.

```
@Id
@GeneratedValue(strategy = GenerationType.IDENTITY)
@Column(name = "id", updatable = false, nullable = false)
private Long id;
```

GenerationType.SEQUENCE

The *GenerationType.SEQUENCE* is my preferred way to generate primary key values and uses a database sequence to generate unique values.

You can use the *@SequenceGenerator* annotation to define the name and schema of the database sequence, the allocation size and the incremental size of the sequence.

```
@Id
@GeneratedValue(strategy = GenerationType.SEQUENCE,
                generator = "book_generator")
@SequenceGenerator(name="book_generator",
                   sequenceName = "book_seq",
                   initialValue=100, allocationSize=50)
@Column(name = "id", updatable = false, nullable = false)
private Long id;
```

GenerationType.TABLE

The *GenerationType.TABLE* simulates a sequence by storing and updating its current value in a database table which requires the use of pessimistic.

You should use *GenerationType.SEQUENCE*, if it's supported by your database.

The *@TableGenerator* annotation is similar to the *@SequenceGenerator* annotation and you can use it to define database table Hibernate shall use for the generator.

```
@Id
@GeneratedValue(strategy = GenerationType.TABLE,
                generator = "book_generator")
@TableGenerator(name="book_generator",
                table="id_generator",
                schema="bookstore")
@Column(name = "id", updatable = false, nullable = false)
private Long id;
```