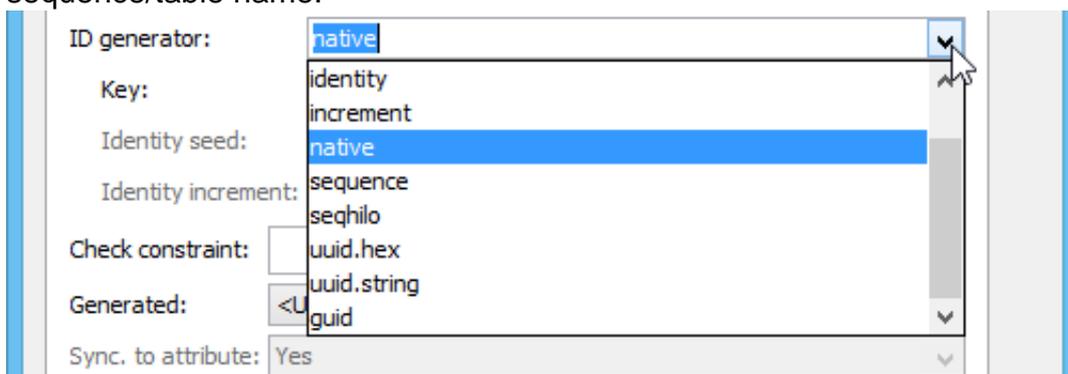

Defining ID Generator for primary key generation

In relational database, records are identified by a unique value, called the primary key. Some databases support controlling how this unique value can be generated, through the use of an ID generator. In this article you will learn how to select an ID generator for a primary key. And since ID generator is designed to serve primary key columns, it is only available to columns that are included in primary keys.

Setting ID generator for a column

ID generator is a column-specific option that can be set in the **Column Specification** window. To set ID generator:

1. Right click on the primary key column of an entity and select **Open Specification...** from the popup menu.
2. Select the ID generator from the drop-down menu of **ID Generator**. If you select sequence, native, seqhilo or hilo as ID Generator, you have to enter the key for the sequence/table name.

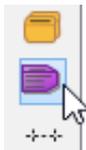


3. Click **OK** to confirm editing.

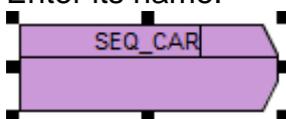
Using sequence as ID generator

You can model a sequence and use it as the ID Generator. To model a sequence:

1. Select **Sequence** from diagram toolbar.



2. Click on the diagram to create a sequence shape.
3. Enter its name.



4. Specify the details of the sequence. Right click on the shape and select **Open Specification...** from the popup menu.
5. Fill in the details of your sequence, such as **Start With**, **Increment By**, **Min Value**, **Max Value**, etc.

6. Click **OK** to confirm editing.
7. Now, you can select this sequence as an ID generator of primary key column.

Description of common ID generators

ID Generator	Description
assigned	allows the application to assign an identifier to the object before save() is called.
guid	uses a database-generated GUID string on MS SQL Server and MySQL.
hilo	uses a hi/lo algorithm to efficiently generate identifiers of type long, short or int, given a table and column as a source of hi values. The hi/lo algorithm generates identifiers that are unique only for a particular database.
identity	supports identity columns in DB2, MySQL, MS

ID Generator Description

increment

SQL Server, Sybase and HypersonicSQL. The returned identifier is of type long, short or int. generates identifiers of type long, short or int that are unique only when no other process is inserting data into the same table. Do not use in a cluster.

native

(default) picks identity, sequence or hilo depending upon the capabilities of the underlying database.

seqhilo

uses a hi/lo algorithm to efficiently generate identifiers of type long, short or int, given a named database sequence.

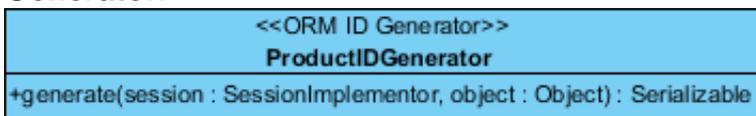
sequence

uses a sequence in DB2, PostgreSQL, Oracle. The returned identifier is of type long,short or int

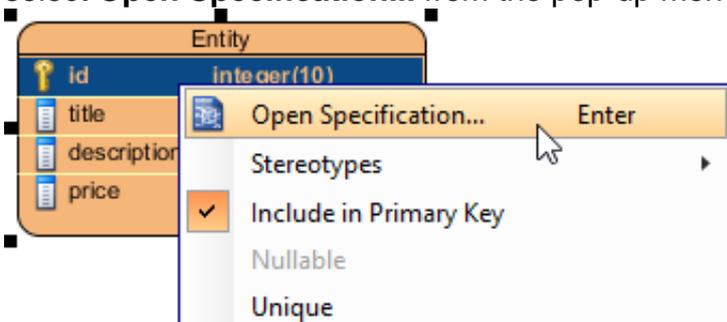
Customizing ID generator

Besides the built-in strategies for generating ID, users can implement how ID will be generated by customizing an ID generator.

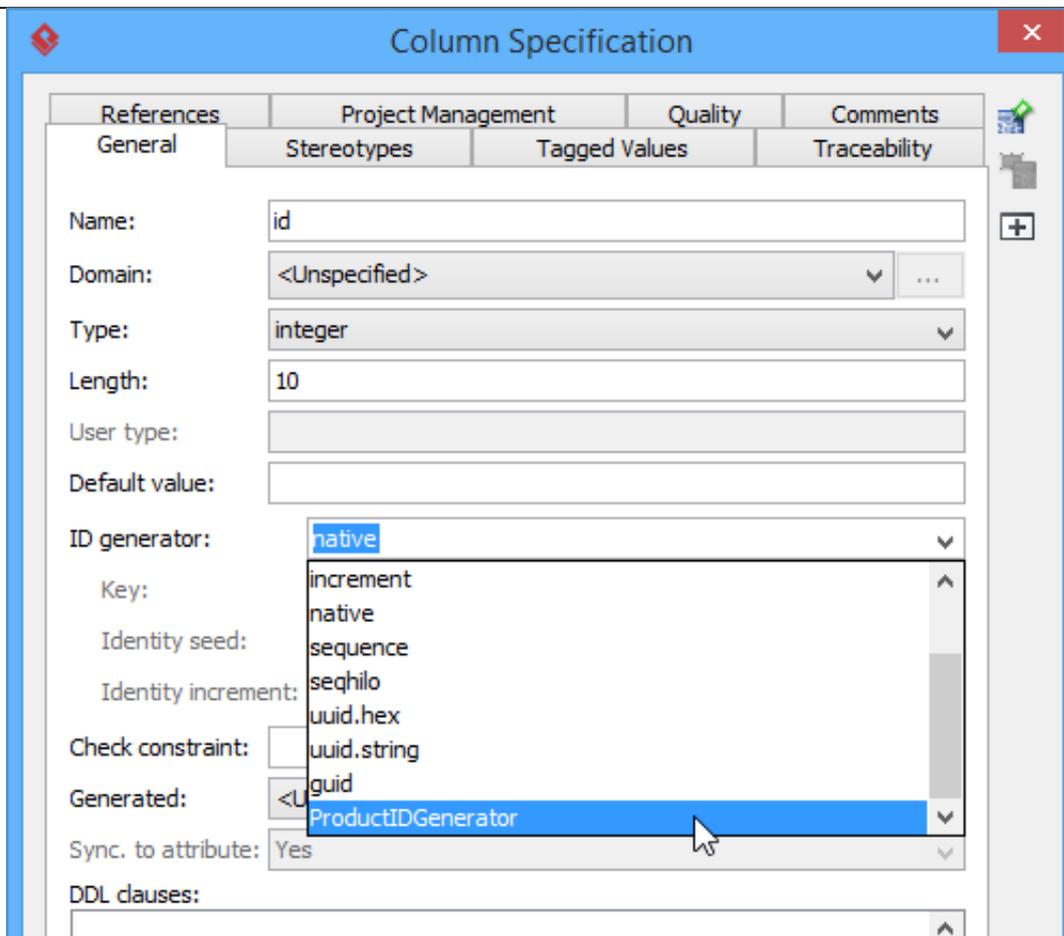
1. In Class Diagram, create the ID generator class and stereotype it as **ORM ID Generator**.



2. Right click on the primary key column that you want to select an ID generator for and select **Open Specification...** from the pop-up menu.



3. In the **Column Specification** window, select the class in the **ID Generator**.



4. Click **OK** to confirm.
5. After generated the ORM code, look for the ID generator class and implement the **generate** method by returning an Integer or Long.

```

/**   * Licensee: VP
    Developmen
t   * License Type: Pu
rchased */ import java
.io.Serializable; import org.hibernate
.engine.SessionImplemento
r; import org.hibernate
.id.IdentifierGenerator;    public class ProductIDGenerator imple
ments IdentifierGenerator {    public Serializable generate(Ses
sionImplementor session, Object object) {    //TODO: Implem
ent Method        throw new UnsupportedOperationException();
    }    } //ORM Hash:fae9faed19486e5f2b85c9d2d0d52cd9

```