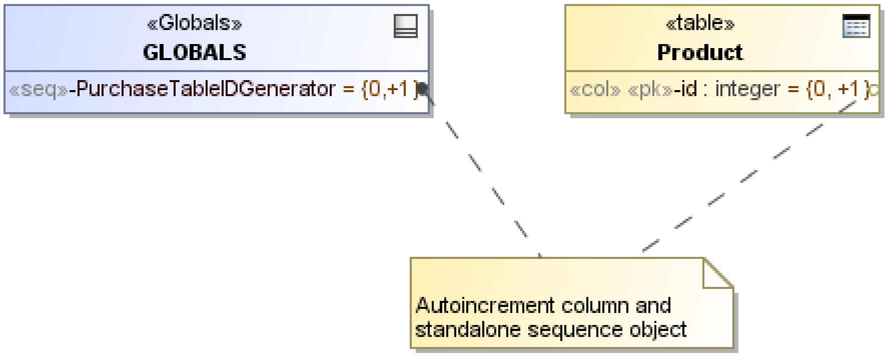
Sequences and autoincrement columns

> ⚠
> - SQL Sequence is modeled as UML Property with «Sequence» stereotype applied. For the sake of compactness, sequences are displayed with the «seq» keyword (instead of the long form - «Sequence») on the diagram.
> - Autoincrement parameters (start value, increment, etc.) data is stored as a separate model element - UML OpaqueExpression, with «IdentitySpecifier» stereotype applied. This element is set as **defaultValue** of the Property - either sequence property (when standalone sequences are modeled) or column property (when autoincrement table columns are modeled).

SQL has facilities to generate sequences of numbers (0, 1, 2, 3, ...). These sequences are often used to fill in values for identifier columns - to uniquely number the row data in the table. There are 2 separate facilities.

- Standalone sequence object. This generator is not tied to any other object. Programer must explicitly query it to get the next value from the sequence and then use the retrieved value appropriately (usually in the INSERT statement to insert value for id column). Usually there are separate sequences for each table; sometimes the same sequence is reused for several columns.
- Autoincrement columns. Column of the table can be designated as autoincrement. When row is inserted into the table, if value of such column is not explicitly provided, one is generated automatically.


«Globals»
GLOBALS
«seq»-PurchaseTableIDGenerator = {0,+1}
«table»
Product
«col» «pk»-id : integer = {0, +1}
Autoincrement column and standalone sequence object

Example of sequence and autoincrement column modeling.

Cameo Data Modeler has modeling support for both kinds of sequences.

To create a standalone sequence

Do one of the following:

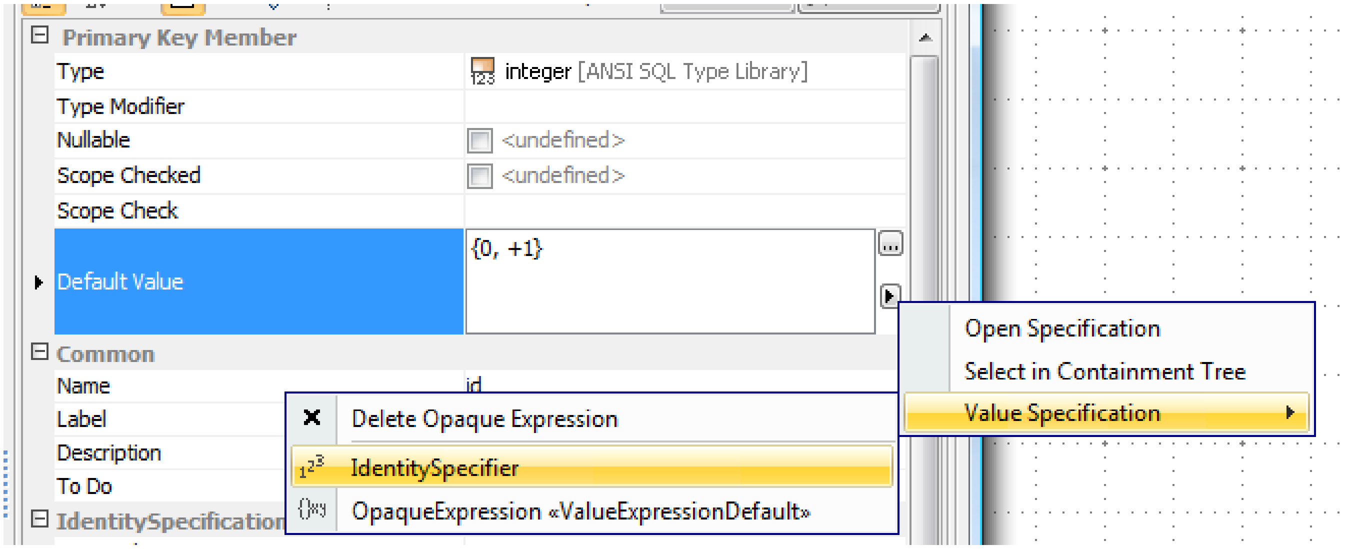
- Select the GLOBALS element shape on a diagram pane and click an appropriate smart manipulation button.
- Right-click the GLOBALS element in the Containment tree and on its shortcut menu, select **New Element** > **Sequence**.

> ⚠ Since a standalone sequence is modeled as a UML Property, it cannot be placed directly into the Schema package.

Autoincrement columns are also supported. To mark a column as autoincrement, you must switch the **Default Value** property value type from value expression to identity specifier.

To mark a column as autoincrement

1. Open the column Specification window.
2. Select the **Default Value** property.
3. Click the black-arrowed button next to the property value and select **Value Specification** > **IdentitySpecifier** as shown in the following figure.



⊟ Primary Key Member

Type	integer [ANSI SQL Type Library]
Type Modifier	
Nullable	□ <undefined>
Scope Checked	□ <undefined>
Scope Check	
Default Value	{0, +1}

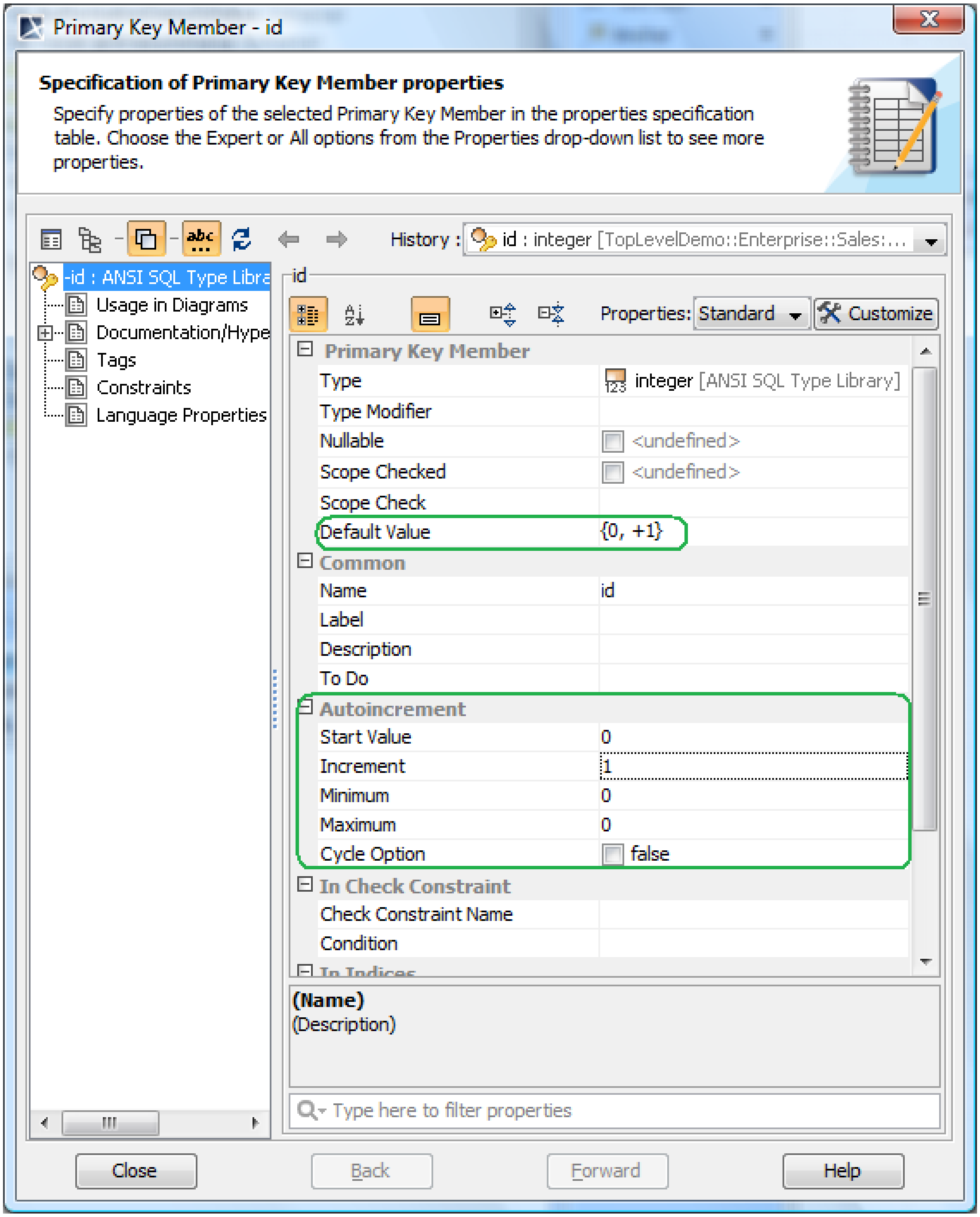
⊟ Common

Name	id
Label	
Description	
To Do	

⊟ IdentitySpecification

Marking column as autoincrement.

After the switching, the Autoincrement property group appears in the Specification window of the column allowing to specify autoincrement data (start value, increment, etc.).



Specification of Primary Key Member properties

Specify properties of the selected Primary Key Member in the properties specification table. Choose the Expert or All options from the Properties drop-down list to see more properties.

Primary Key Member	
Type	integer [ANSI SQL Type Library]
Type Modifier	
Nullable	□ <undefined>
Scope Checked	□ <undefined>
Scope Check	
Default Value	{0, +1}
Common	
Name	id
Label	
Description	
To Do	
Autoincrement	
Start Value	0
Increment	1
Minimum	0
Maximum	0
Cycle Option	□ false
In Check Constraint	
Check Constraint Name	
Condition	
In Indices	

(Name)
(Description)

Additional properties in autoincrement column’s Specification window.

Besides the standard SQL element properties and sequences, an autoincrement column has the following properties available in the **Autoincrement** property group of the Specification window.

Property name	Description
Start Value	Starting value of the sequence counter.
Increment	Delta value of the sequence counter (can be negative - to count down).
Minimum	Lower bound of the counter (if any).
Maximum	Upper bound of the counter (if any)

Cycle Option	The counter can “wrap around” when it reaches the maximum (or minimum - for downwards counters)

Additionally, sequence has an **Identity** field and column has the **Default Value** field, where textual representation of the counter options can be entered. This feature can be used for nice displaying of the counter configuration in the diagrams (the start, inc, min, max field data is normally not visible in the diagram). Some notation convention should be adopted how to map the counter data into the text representation. For example, it could be: {<start>, <inc>, <min>-<max>, <c>}. Then the counter from 0 with +1 increment, min max of 0 and 1000 and cycle option would be displayed as “{0, +1, 0-1000, C}” string. At the moment this text representation is not automatically connected to the counter field values, so synchronization has to be done by hand.