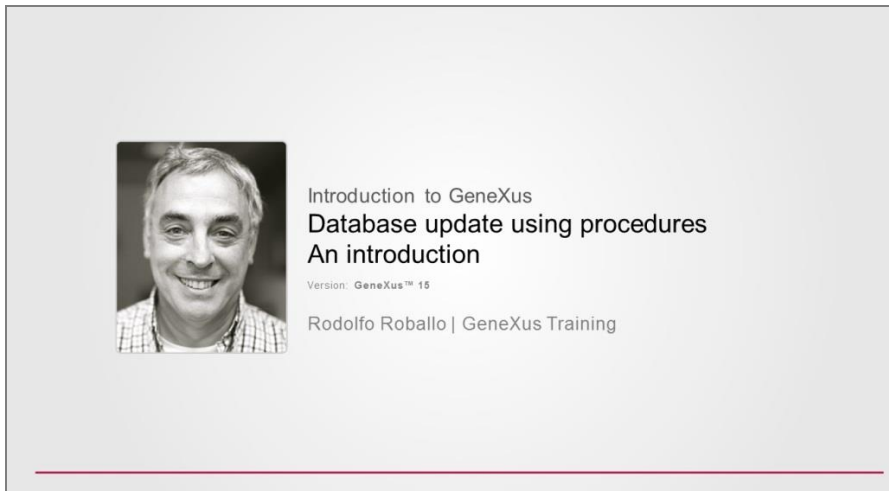


## Introduction to updating the database with For Each, Delete and New commands

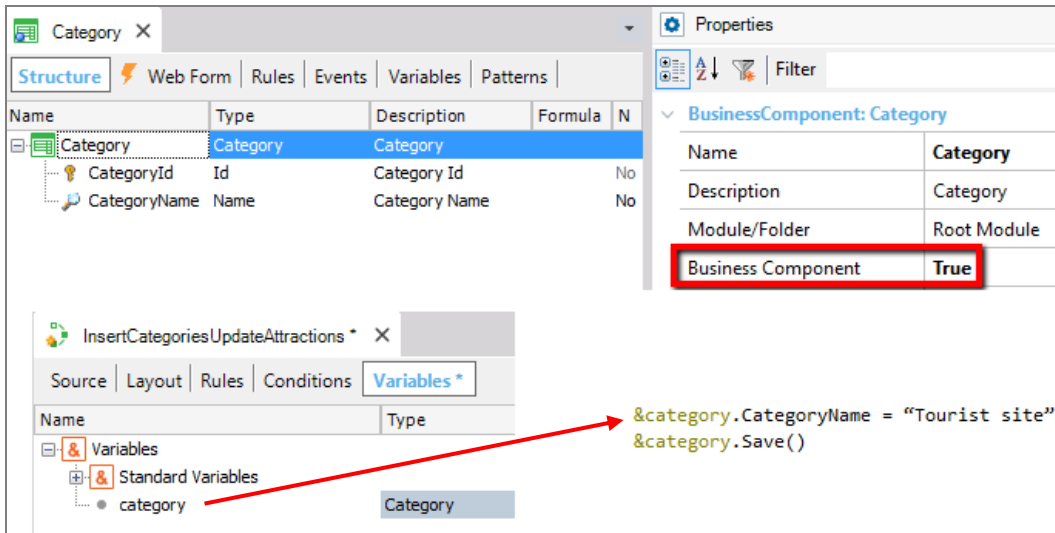


So far, to update the database data we have used transactions in the two ways available to use them:

- Running their screen and entering data in an interactive manner

A screenshot of a web browser window showing a web application. The browser's address bar contains the URL 'trialapps3.genexus.com/Id04eb4db3ea360a1318d773dacf0e765c/category.aspx'. The application has a dark red header with 'Application Name' on the left and 'by GeneXus' on the right. Below the header, the text 'Recents: category' is visible. The main content area is titled 'Category' and contains a form with a 'SELECT' button and navigation arrows. The form has two input fields: 'Id' with the value '1' and 'Name' with the value 'Musem'. At the bottom of the form are two buttons: 'CONFIRM' and 'CANCEL'.

- Through their associated Business Component, through a variable, without using their screen.



In addition, there's another way to make insertions, changes and deletions to the database, which is only valid in Procedure objects.

To insert data, we use the New command:

```
New
  CategoryId = 5
  CategoryName = "Tourist Site"
Endnew
```

Here we insert a new category in the CATEGORY table. If CategoryId is autonumbered, we don't enter a value for it:

```
New
  CategoryName = "Tourist Site"
Endnew
```

To **update** the database, we use the For Each command itself. For example, to update all the attractions in Beijing that have the Monument category and give them the new Tourist Site category, we type:

```
For each Attraction
Where CityName = "Beijing" and CategoryName = "Monument"
  CategoryId = find( CategoryId, CategoryName = "Tourist site")
Endfor
```

... an attribute of the base table is being updated (even though it could also belong to the extended table).

For **deletions**, once we're positioned over the record to be deleted, we use the Delete command.

```
For each Attraction
Delete
Endfor
```

When compared to the solution that uses Business Components, one disadvantage of these commands is that they don't check for referential integrity, or trigger any of the rules stated in the transaction.

Their advantage is, precisely, that they are faster and that's why they are also known as "more performant."

If you need to delete data from a table that has millions of records, doing it in this way, using the Delete command, will take far less time. However, this deletion will not make any controls and inconsistencies may be introduced in the database.

We will not study these commands in this course. If you want to learn more about them, we suggest reading the materials corresponding to the GeneXus Analyst Course.

