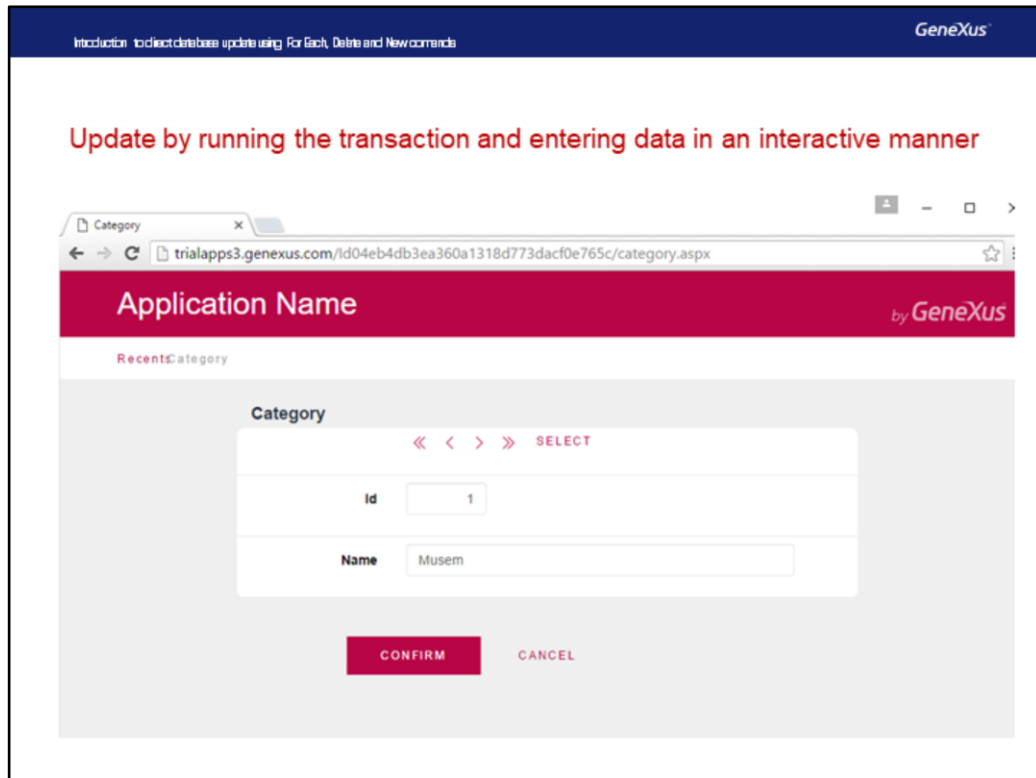


Direct database update using For Each, New and Delete commands

Introduction

GeneXus 16



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

- By running their screen and entering data in an interactive manner.

Update through the Business Component associated with the transaction

The screenshot displays the GeneXus IDE interface. At the top, a tab labeled 'Category' is active. Below it, the 'Structure' tab is selected, showing a table with columns: Name, Type, Description, Formula, and N. The table contains two rows: 'CategoryId' (Type: Id, Description: Category Id, Formula: No, N: No) and 'CategoryName' (Type: Name, Description: Category Name, Formula: No, N: No). To the right, the 'Properties' panel is visible, showing the 'BusinessComponent: Category' properties. The 'Business Component' property is highlighted with a red box and set to 'True'. Below the 'Structure' tab, the 'Variables' tab is selected, showing a table with columns: Name, Type. The table contains one row: 'category' (Type: Category). To the right of the 'Variables' tab, the following code is displayed:

```
&category.CategoryName = "Tourist site"
&category.Save()
```

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

To perform the insertion, we can use the New command

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

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

Only valid in procedure objects!

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

To perform the insertion, we use the New command.

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

To edit data, we use the For Each command

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

Only valid in procedure objects!

For the **update** operation, the For Each command can be used. For example, to change all of Beijing's attractions with the Monument category to the Tourist Site category, we type a For Each command and assign the new values to the attributes we want to update.

Here we're updating an attribute of the base table, even though it could belong to the extended table.

To delete data, we use the Delete command

```
For each Attraction  
  Delete  
Endfor
```

Only valid in procedure objects!

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

Advantages and disadvantages of using these commands in procedures

- Disadvantages:
 - No referential integrity controls are made; therefore, data can be left inconsistent.
 - They don't trigger any transaction rules
- Advantages:
 - They are faster (more "performant")

Compared to the solution that we saw with Business Components, these commands have the disadvantage that they don't control referential integrity, and don't trigger any transaction rules.

Their advantage is, precisely, that they are faster; for this reason, they are considered as "more performant".

If you need to delete the data from an entire table with millions of records, doing it in this way, with the Delete command, will take considerably less time. However, this deletion will not make any controls and the database can be left in an inconsistent state.

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.

GeneXus™

The power of doing.

Videos

Documentation

Certifications

training.genexus.com

wiki.genexus.com

training.genexus.com/certifications