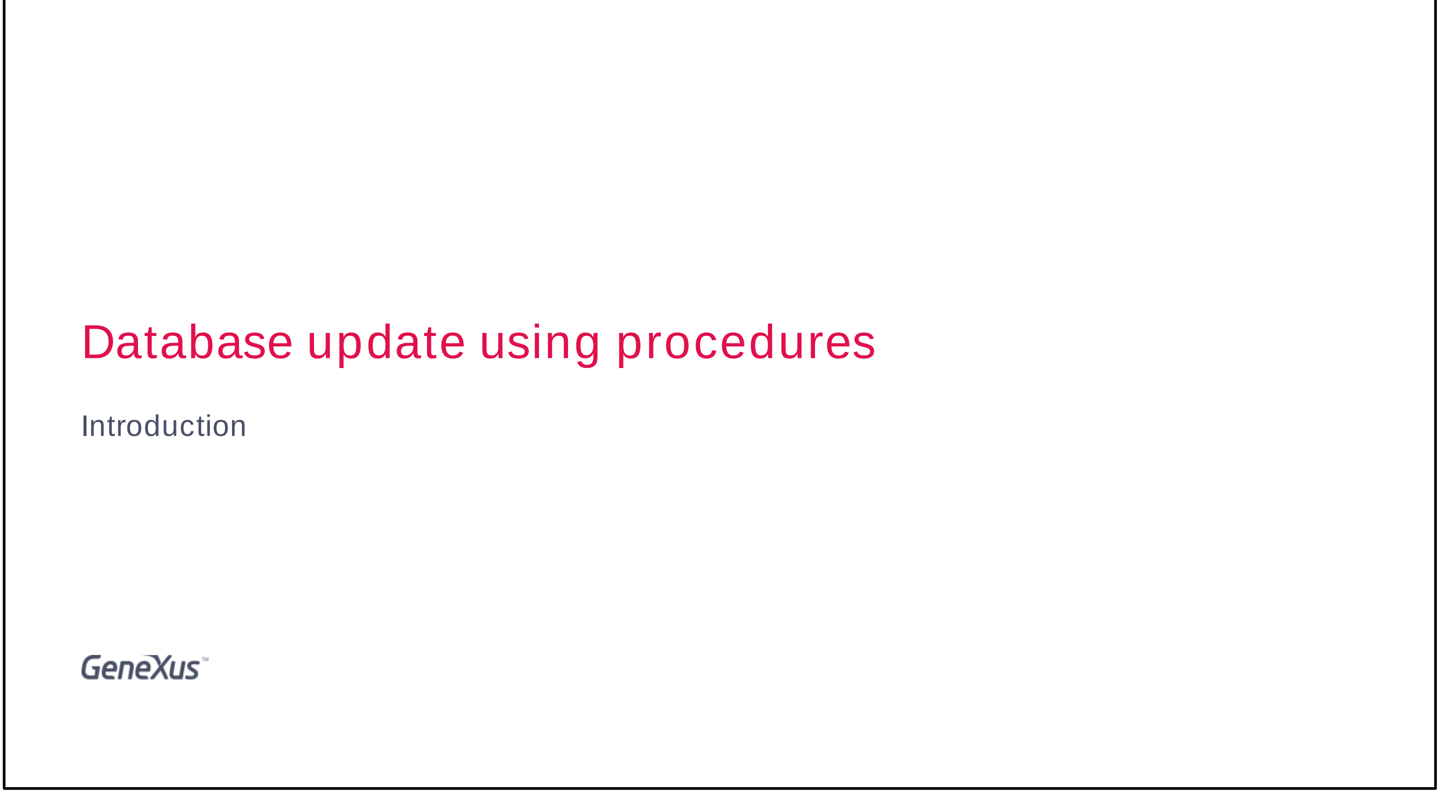
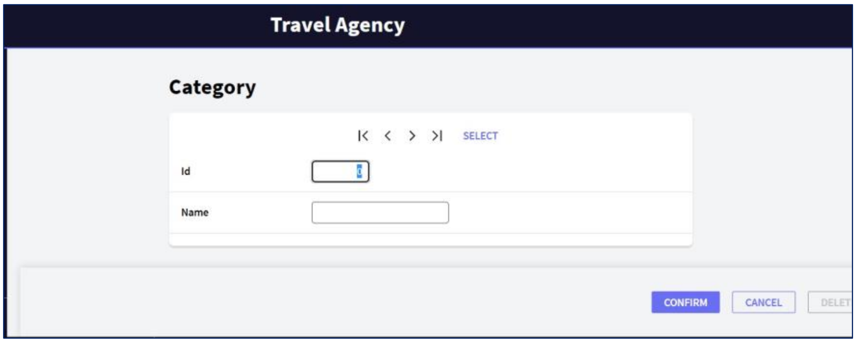


Database update using procedures

Introduction

GeneXus™

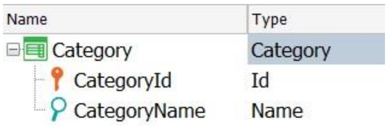
Transaction form



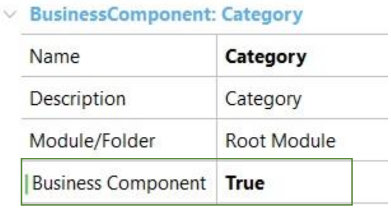
So far, to update the database data we have used transactions in the two ways available to use them:

- Running their screen and entering data in an interactive manner

Category transaction as Business Component

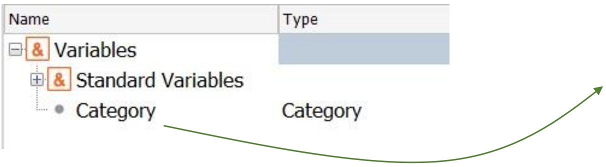


Name	Type
Category	Category
CategoryId	Id
CategoryName	Name



BusinessComponent: Category

Name	**Category**
Description	Category
Module/Folder	Root Module
Business Component	**True**

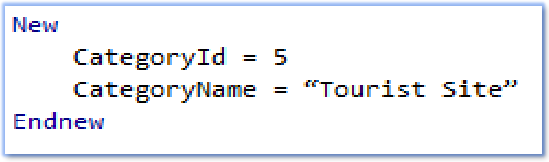


Name	Type
Variables	
Standard Variables	
Category	Category

```
&Category.CategoryName = "Tourist site"
&Category.Insert()
Commit
```

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

Add a new category



```
New
    CategoryId = 5
    CategoryName = “Tourist Site”
Endnew
```

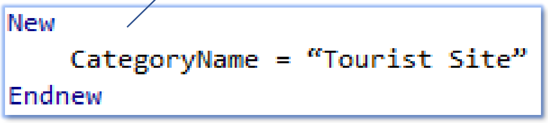
In addition, there's another way to make insertions, changes and deletions to the database, which is only valid in Procedure objects.

To insert data, we use the New command:

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

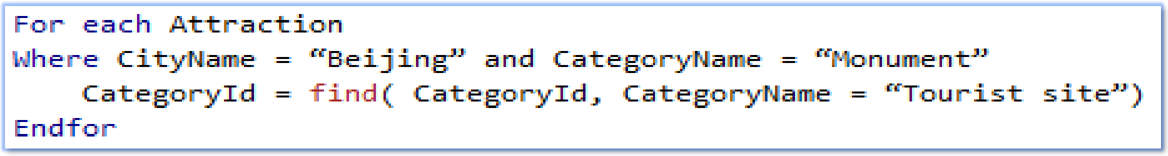
Add a new category

If CategoryId is autonumbered
then no value is assigned



```
New
    CategoryName = “Tourist Site”
Endnew
```

Update a category

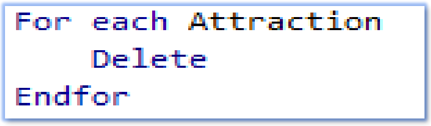


```
For each Attraction
Where CityName = “Beijing” and CategoryName = “Monument”
    CategoryId = find( CategoryId, CategoryName = “Tourist site”)
Endfor
```

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:

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

Delete a category



```
For each Attraction
    Delete
Endfor
```

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

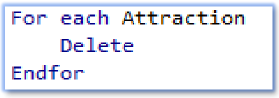
Advantages and disadvantages

Disadvantages

- No referential integrity checks are performed
- Transaction rules are not triggered.

Advantages

- Faster.



```
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.

GeneXus™

training.genexus.com
wiki.genexus.com