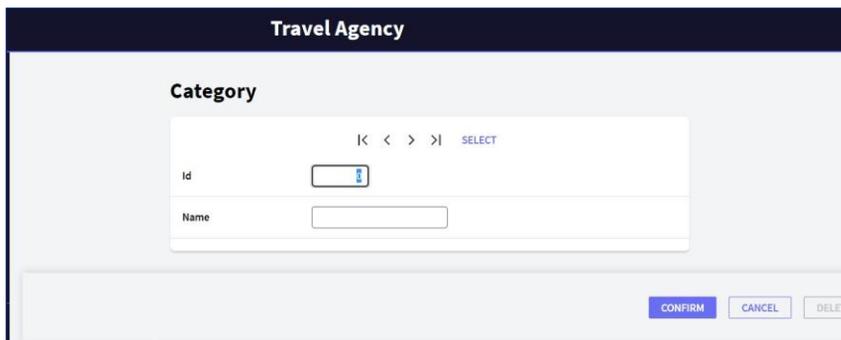


Atualização do banco de dados usando procedimentos

Introdução

GeneXus[™]

Formulário de transação



The screenshot shows a web form titled "Travel Agency" with a sub-section "Category". At the top of the form, there are navigation icons: a left arrow, a right arrow, and a double right arrow, followed by the word "SELECT". Below this, there are two input fields: "Id" with a small blue icon to its right, and "Name" with an empty text box. At the bottom right of the form, there are three buttons: "CONFIRM" (highlighted in blue), "CANCEL", and "DELETE".

Até agora, para atualizar os dados no banco de dados usamos as transações em suas duas formas de uso:

- Executando sua tela e inserindo dados de forma interativa

Transação Category como 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
```

- E através de seu Business Component associado, mediante uma variável, sem usar a tela

Adicionar uma categoria

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

Procedimentos

Mas temos outra maneira de fazer inserções, modificações e exclusões no banco de dados, válida apenas em objetos Procedimento.

Para inserção, temos o comando new:

Aqui inserimos uma nova categoria na tabela CATEGORY. Se CategoryId é autonumerado, então não lhe especificamos valor:

Adicionar uma categoria

Se CategoryId for numerado automaticamente, nenhum valor será atribuído

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

Procedimentos

Atualizar uma categoria existente

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

Para a **atualização** contamos com o próprio comando for each. Por exemplo, para mudar todas as atrações de Beijing que tinham categoria Monument, para a nova categoria Tourist Site, escrevemos:

... onde um atributo da tabela base está sendo atualizado (embora pudesse ser da tabela estendida).

Excluir uma categoria

```
For each Attraction  
Delete  
Endfor
```

E para a **exclusão**, uma vez posicionados sobre o registro a ser excluído, temos o comando Delete

Vantagens e desvantagens

Desvantagens

- Nenhuma verificação de integridade referencial é executada.
- As regras de transação não são acionadas

Vantagens

- Mais rápido

```
For each Attraction  
Delete  
Endfor
```

Esses comandos têm a desvantagem, em face à solução que vimos com business components, que eles não controlam a integridade referencial, nem disparam qualquer regra da transação.

Sua vantagem é, justamente que são mais rápidos, o que também se conhece como **“mais eficientes”**.

Se você tiver que excluir os dados de uma tabela inteira de milhões de registros, fazê-lo desta forma, com o comando Delete, irá demorar muito menos tempo. No entanto, essa exclusão não fará controles e pode deixar o banco de dados em situação inconsistente.

Não vamos estudar estes comandos neste curso.

*GeneXus*TM

training.genexus.com
wiki.genexus.com