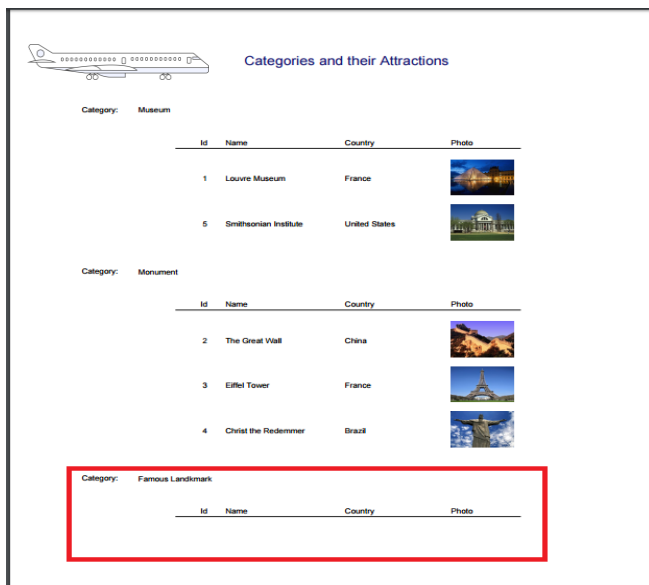


Como listar informação agrupada

For eachs aninhados



GeneXus™

Listar informação agrupada






Categories and their Attractions

Category: Museum

| Id | Name | Country | Photo |
|----|-----------------------|---------------|-----------------------------------------------------------------------------------|
| 1 | Louvre Museum | France |  |
| 5 | Smithsonian Institute | United States |  |

Category: Monument

| Id | Name | Country | Photo |
|----|---------------------|---------|-----------------------------------------------------------------------------------|
| 2 | The Great Wall | China |  |
| 3 | Eiffel Tower | France |  |
| 4 | Christ the Redemmer | Brazil |  |

Category: Famous Landmark

| Id | Name | Country | Photo |
|----|------|---------|-------|
|----|------|---------|-------|

- Esta era a lista de categorias com suas atrações.
- A categoria Famous Landmark não tem nenhuma atração no momento.
- E se agora quisermos listar não TODAS as categorias, mas somente as que tenham atrações cadastradas?

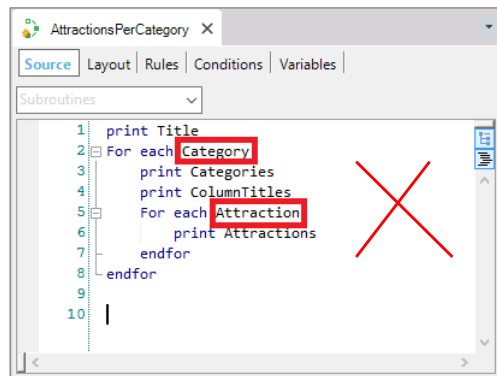
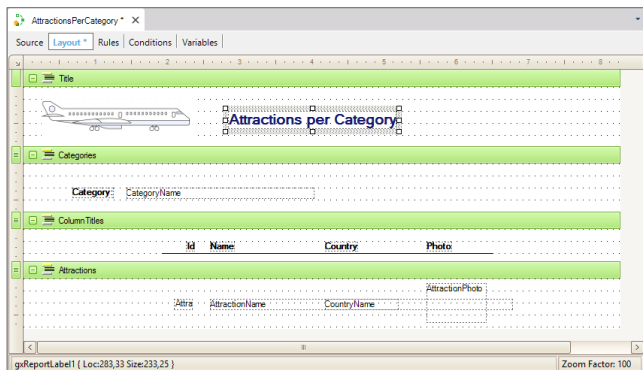
Na seção anterior, vimos uma lista solicitada pela agência de viagens, que mostrava todas as Categorias de atrações turísticas e para cada categoria, a lista de atrações que existiam registradas.

Vamos mudar para a atração "The Great Wall", sua categoria, de modo que "Famous Landmark" já não tem nenhuma atração associada.

Agora vamos executar a lista novamente. Como podemos ver, esta lista mostra TODAS as categorias inseridas, inclusive aquelas que não possuem atrações associadas.

Se isso não é o que se quer, ou seja, se o que se quer é mostrar apenas as categorias com atrações, como fazemos?

Por que não podemos utilizar a solução anterior?



| CategoryId | CategoryName |
|------------|-----------------|
| 1 | Museum |
| 2 | Monument |
| 3 | Famous Landmark |

| AttractionId | AttractionName | CategoryId | ... |
|--------------|-----------------------|------------|-----|
| 1 | Louvre Museum | 1 | |
| 5 | Smithsonian Institute | 1 | |
| 2 | The Great Wall | 2 | |
| 3 | Eiffel Tower | 2 | |
| 4 | Christ the Redemmer | 2 | |

3?

Vamos implementar isso em outro procedimento. Para fazer isso, salvamos o que tínhamos com outro nome. E vamos mudar o título do text block.

Se analisarmos o For Eachs que implementamos, vemos que a tabela base do For Each externo é Category e a tabela base do For Each aninhado é Attraction.

Mas desta forma, como se acessa primeiro a tabela de Category, a informação do registro é impressa e, logo após se passa a executar o For each aninhado, a categoria será impressa antes de saber se tem atrações relacionadas ou não.

Isso não serve para nós. Precisamos acessar as categorias **das** atrações, porque é a única maneira de ter certeza de que a categoria que será impressa tem pelo menos uma atração.

Solução

| | | Foreign Key | | Primary Key | | |
|--------------|-----------------------|-------------|------------|-------------|------------|-----------------|
| AttractionId | AttractionName | CountryId | CategoryId | ... | CategoryId | CategoryName |
| 1 | Louvre Museum | 2 | 1 | | 1 | Museum |
| 5 | Smithsonian Institute | 4 | 1 | | 2 | Monument |
| 2 | The Great Wall | 3 | 2 | | 3 | Famous Landmark |
| 3 | Eiffel Tower | 2 | 2 | | | |
| 4 | Christ the Redemmer | 1 | 2 | | | |

Category: Museum

| | | |
|---|-----------------------|---------------|
| 1 | Louvre Museum | France |
| 5 | Smithsonian Institute | United States |



Category: Monument

| | | |
|---|----------------|--------|
| 2 | The Great Wall | China |
| 3 | Eiffel Tower | France |



```

For each Attraction
  print Categories
  For each Attraction
    print Attractions
  endfor
endfor
order CategoryId
    
```

Agrupamento ou critério de corte

Controle de Corte

A ideia será agrupar as atrações da tabela Attraction por categoria, e depois percorrer esses grupos, imprimindo a categoria para cada um (para o que terá que acessar a tabela Category para recuperar seu nome); e também imprimindo cada atração do grupo.

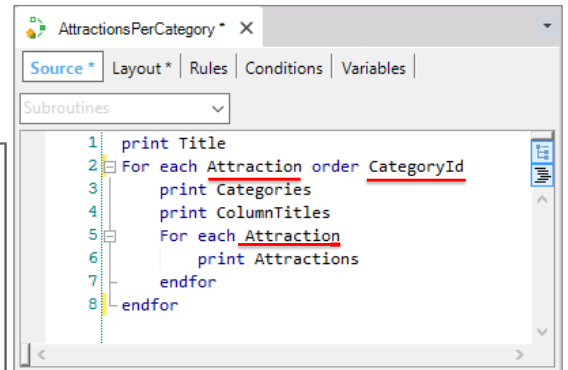
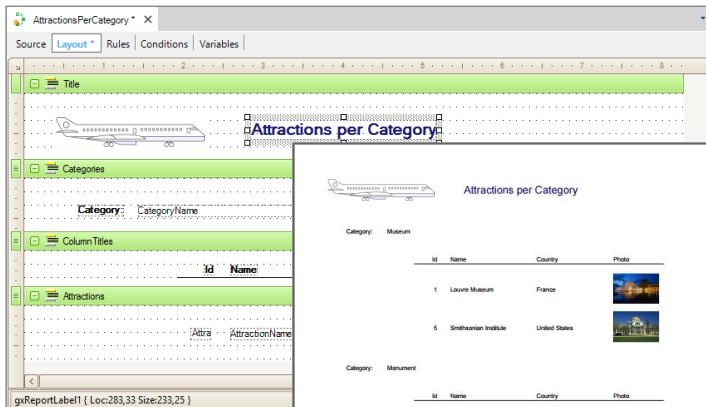
Para depois passar para o próximo grupo ... e assim por diante.

Em suma, precisamos apenas percorrer a tabela de Attraction, primeiro agrupando-a por categoria e imprimindo a categoria; e depois imprimindo de cada grupo de categoria (navegando a mesma tabela de Attractions, é claro) suas atrações.

Observemos que a maneira de indicar ao GeneXus que se deseja agrupar por CategoryId é especificando a cláusula Order.

Este caso de for eachs aninhados que percorrem a mesma tabela é conhecido como **corte de controle**.

Listar informação agrupada: control break/corte de controle



Façamos as mudanças em nosso procedimento ...

Primeiro mudamos a transação base do for each externo, por Attraction... E adicionamos a cláusula Order, para ordenar por CategoryId, que no caso do corte de controle será também para algo mais forte: para agrupar por esse atributo.

Executemos. Vemos que efetivamente a categoria "Famous Landmark", que não tem atrações, não está realmente sendo listada.

Listar informação agrupada: control break/corte de controle

The screenshot displays the GeneXus IDE interface. On the left, a tree view shows the project structure with 'AttractionsPerCategory' selected. The main window shows a 'Procedure AttractionsPerCategory Navigation Report'.

Procedure AttractionsPerCategory Navigation Report

| | | | |
|----------------|--------------------------|---------------|------------------------|
| Name | AttractionsPerCategory | Environment | Default (C#) |
| Description | Attractions Per Category | Spec. Version | 15_0_0-105189 |
| Output Devices | File | Form Class | Graphic |
| Main | Yes | Program Name | AttractionsPerCategory |
| | | Call Protocol | HTTP |
| | | Parameters | |

Levels

For Each Attraction (Line: 6)

Order: CategoryId
Index: IATTRACTION2

Navigation Start from: FirstRecord
filters: Loop while: NotEndOfTable
Join location: Server

- Attraction (AttractionId)
- Country (CountryId)
- Category (CategoryId)

Break Attraction (Line: 14)

Order: CategoryId
Index: IATTRACTION2

Navigation Loop while: CategoryId = @CategoryId
filters:
Join location: Server

- Attraction (AttractionId)
- Country (CountryId)

At the bottom of the IDE, a status bar shows: 0 Errors, 0 Warnings, 1 Success.

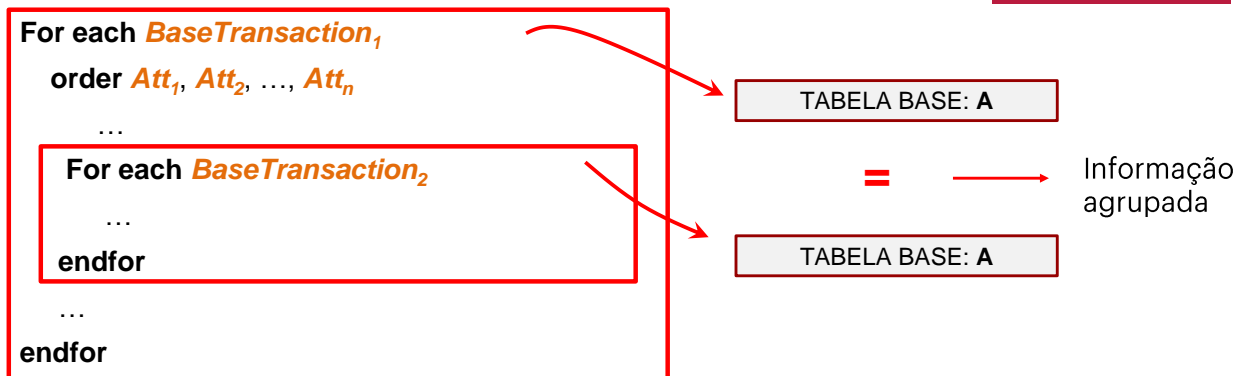
Se observarmos a lista de navegação resultante, podemos ver que nos informa de um for each na tabela de Attraction, ordenado por CategoryId... e que será cortado, Break, pelo for each aninhado sobre a mesma tabela, Attraction. Observemos que neste "corte" apenas as atrações da categoria do grupo são percorridas.

Conceituação

- Agrupar a informação por algum critério

✓ **Mesma tabela base**

Corte de controle



Vamos conceituar, então, como um corte de controle é definido.

Se trata de For eachs aninhados... Com a mesma tabela base para todos os For Eachs...

Conceituação

- Agrupar a informação por algum critério

✓ Critério de corte:
cláusula order

For each *BaseTransaction*₁

order *Att*₁, *Att*₂, ..., *Att*_n

...

For each *BaseTransaction*₂

...

endfor

...

endfor

Break criteria

TABELA BASE : A

Agrupada por...



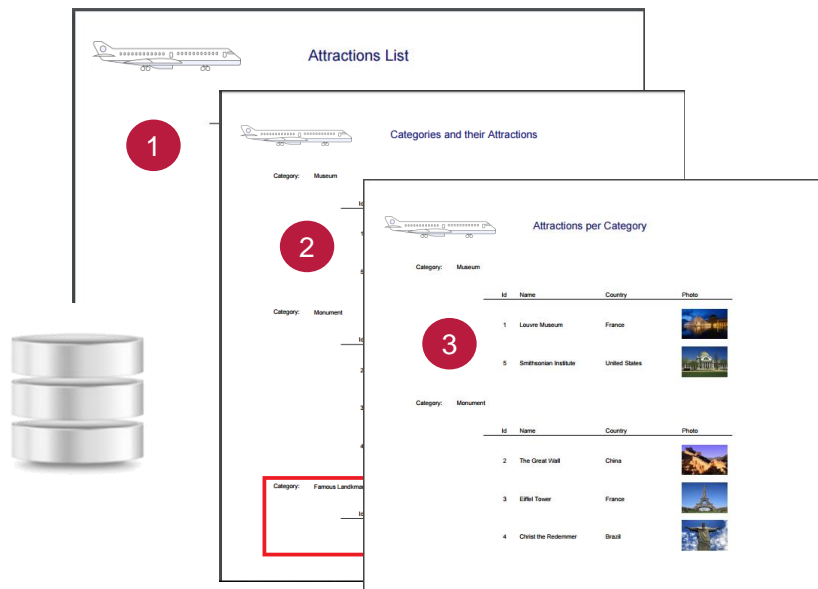
E a cláusula Order para estabelecer o critério de corte.

Resumo

Resumindo

- Listagens
- ✓ Layout
- ✓ Source

Comando
For each



Com estes vídeos, vimos as facilidades que nos proporciona GeneXus para fazer listagens simples (que navegam uma única tabela), ou mais complexas, navegando informações de várias tabelas relacionadas (join), ou da mesma tabela, mas agrupadas por algum critério (corte de controle).

O comando que usamos em todos os casos para acessar a base de dados é o For each.

Resumindo

- Listagens

- ✓ Layout
- ✓ Source



| Id | Name | Country | Photo |
|----|----------------|---------|-----------------------------------------------------------------------------------|
| 1 | Louvre Museum | France |  |
| 2 | The Great Wall | China |  |
| 3 | Eiffel Tower | France |  |

For each
command



1

For each *Transação Base*
order *Att₁, Att₂, ..., Att_n*
where *condition₁*
where *condition₂*
...
where *condition_n*
MainCode
endfor

For each simples

No primeiro caso, usamos um for each simples, no qual, através de sua Transação Base se infere a tabela a ser navegada.

Resumido

- Listagens

- ✓ Layout
- ✓ Source

Comando For each



Categories and their Attractions

| Category: Museum | | | |
|------------------|-----------------------|---------------|-------|
| ID | Name | Country | Photo |
| 1 | Louvre Museum | France | |
| 5 | Smithsonian Institute | United States | |

| Category: Monument | | | |
|--------------------|---------------------|---------|-------|
| ID | Name | Country | Photo |
| 2 | The Great Wall | China | |
| 3 | Eiffel Tower | France | |
| 4 | Christ the Redeemer | Brazil | |

| Category: Famous Landmark | | | |
|---------------------------|------|---------|-------|
| ID | Name | Country | Photo |

2

```

For each Transação Base
order Att1, Att2, ..., Attn
where condition1
where condition2
...
where conditionn
...
For each Transação Base
order Att1, Att2, ..., Attn
where condition1
where condition2
...
where conditionn
...
endfor
...
endfor

```

1
≠
N

For eachs aninhados

Join

No segundo, temos um par de for eachs aninhados, em que a partir de transações base diferentes, é descoberta uma relação de um a muitos entre a informação de cada for each.

Resumindo

- Listagens

- ✓ Layout
- ✓ Source

For each
command



Attractions per Category

| Category: Museum | | | |
|------------------|-------------------------|---------------|-------|
| ID | Name | Country | Photo |
| 1 | Louvre Museum | France | |
| 5 | Smithsonian Institution | United States | |

| Category: Monument | | | |
|--------------------|---------------------|---------|-------|
| ID | Name | Country | Photo |
| 2 | The Great Wall | China | |
| 3 | Eiffel Tower | France | |
| 4 | Christ the Redeemer | Brazil | |

3

```
For each Transação Base
order Att1, Att2, ..., Attn
where condition1
where condition2
...
where conditionn
```

```
For each Transação Base
order Att1, Att2, ..., Attn
where condition1
where condition2
...
where conditionn
```

```
endfor
```

For eachs aninhados

Controle de corte

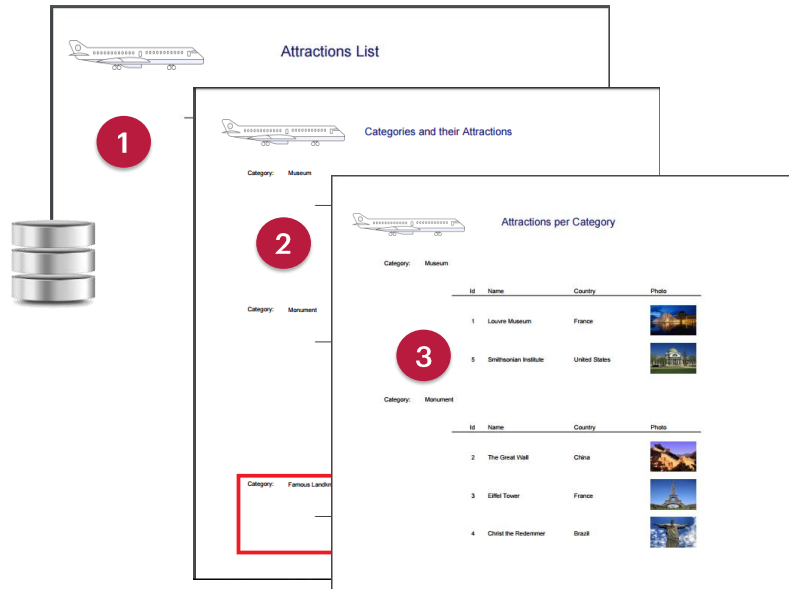
Enquanto que no terceiro, também temos um par de for eachs aninhados, mas cujas transações base **coincidem**. GeneXus entende que se deseja "cortar" ou "agrupar" a informação da tabela a ser percorrida, pelo atributo ou conjunto de atributos que são especificados na cláusula order do for each exterior.

E mais...

- Listagens
- ✓ Layout
- ✓ Source

comando
For each

- Painéis
- Objeto Query



Mais tarde, veremos outros mecanismos para fazer consultas à base de dados e obter informações de forma flexível e vistosa.

GeneXus™

training.genexus.com
wiki.genexus.com
training.genexus.com/certifications