

関連情報をグループ化して リスト表示する方法

ネストされた For each

GeneXusTM

関連情報をリスト表示する方法

次のようなリストを表示する必要がある場合:

タイトル

カテゴリ

列見出し

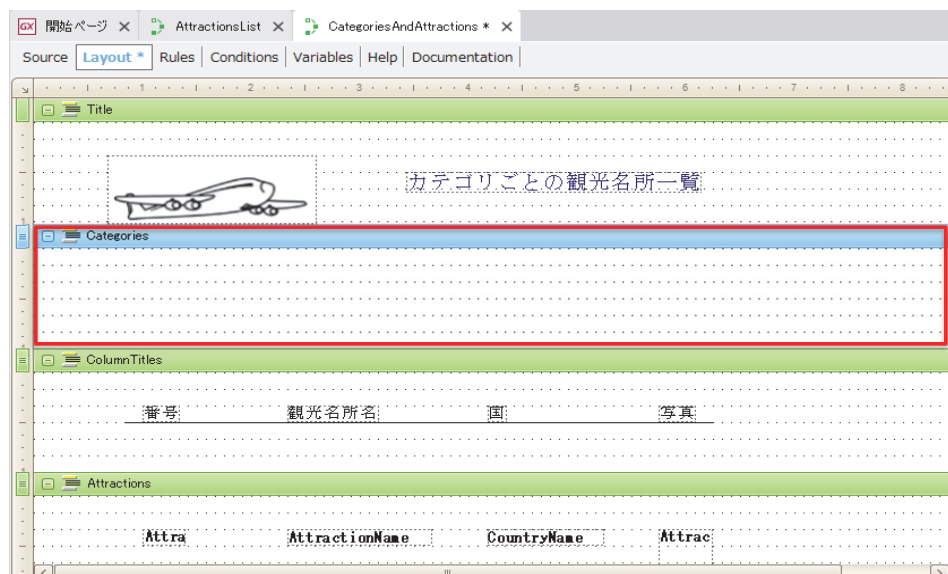
観光名所

 カテゴリごとの観光名所一覧			
カテゴリ 美術館			
番号	観光名所名	国	写真
1	ルーブル美術館	フランス	
5	スミソニアン博物館	アメリカ合衆国	
カテゴリ モニュメント			
番号	観光名所名	国	写真
3	エッフェル塔	フランス	
4	コルコバードのキリ	ブラジル	
カテゴリ 有名なランドマーク			
番号	観光名所名	国	写真
2	万里の長城	中国	

ここでは、旅行代理店からリクエストがあった想定で、観光名所のカテゴリをすべて表示し、各カテゴリの観光名所をすべて表示します。

ここまでに実装したリストとのもっとも大きな違いは、このリストでは観光名所をカテゴリごとにグループ化するという点です。

[Layout] エlement



[Layout] で、新しいプリントブロックを追加しました。

この新しいプリントブロックに「Categories」という名前を付けます。ここに「カテゴリ」というテキストブロックを挿入し、CategoryName 項目属性を設定します。その他のプリントブロックはそのままにします。

固定コンテンツのプリントブロックが 2 つあります: [Title] および [ColumnTitles]。また、データベースからデータを取得する必要がある可変コンテンツのプリントブロックが 2 つあります: [Categories] および [Attractions]。両方に項目属性が含まれます。[Categories] には、Category テーブルから取得する CategoryName が表示されます。Attractions には、Attraction 拡張テーブルに含まれるすべての項目属性が表示されます。

次に [Source] を確認します。

[Source] エlement

For each

CategoryId	CategoryName
1	美術館
2	遺跡
3	有名なランドマーク

For each

	AttractionId	AttractionName	CountryId	CategoryId	...
→	1	ルーブル美術館	2	1	
	2	万里の長城	3	3	
	3	エッフェル塔	2	2	
	4	コルコバードの キリスト像	1	2	
→	5	スミソニアン博物館	4	1	

endfor

endfor

カテゴリをナビゲートし、各カテゴリに対して観光名所を**複数ナビゲート**する必要があるため (対象のカテゴリに含まれる観光名所)、このリストは既に関連したリストとは異なります。

変更前:

単純な For each

Attraction

```
For each Attraction
  print Attractions
endfor
```



ベーステーブル

AttractionId	AttractionName	CountryId	CityId	...
1	ルーブル美術館	2	1	
2	万里の長城	3	1	
3	エッフェル塔	2	1	

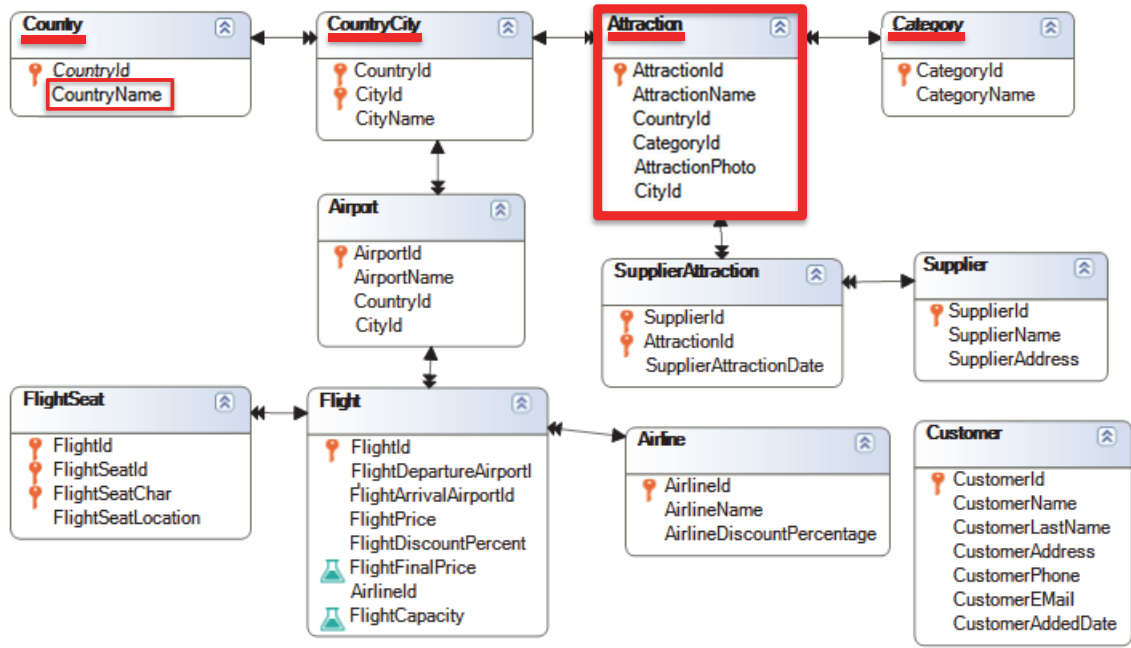
CountryId	CityId	CityName
1	1	リオデジャネイロ
1	2	サンパウロ
2	1	パリ
...

CountryId	CountyName
1	ブラジル
2	<u>フランス</u>
3	中国

拡張

以前のリストでは、観光名所をナビゲートしました。各観光名所は 1 つの国のみに関連付けられているため、観光名所に対応する国を取得できました。これは、ナビゲートしているベーステーブルの拡張テーブルに国があったためです。

変更前:
単純な For each

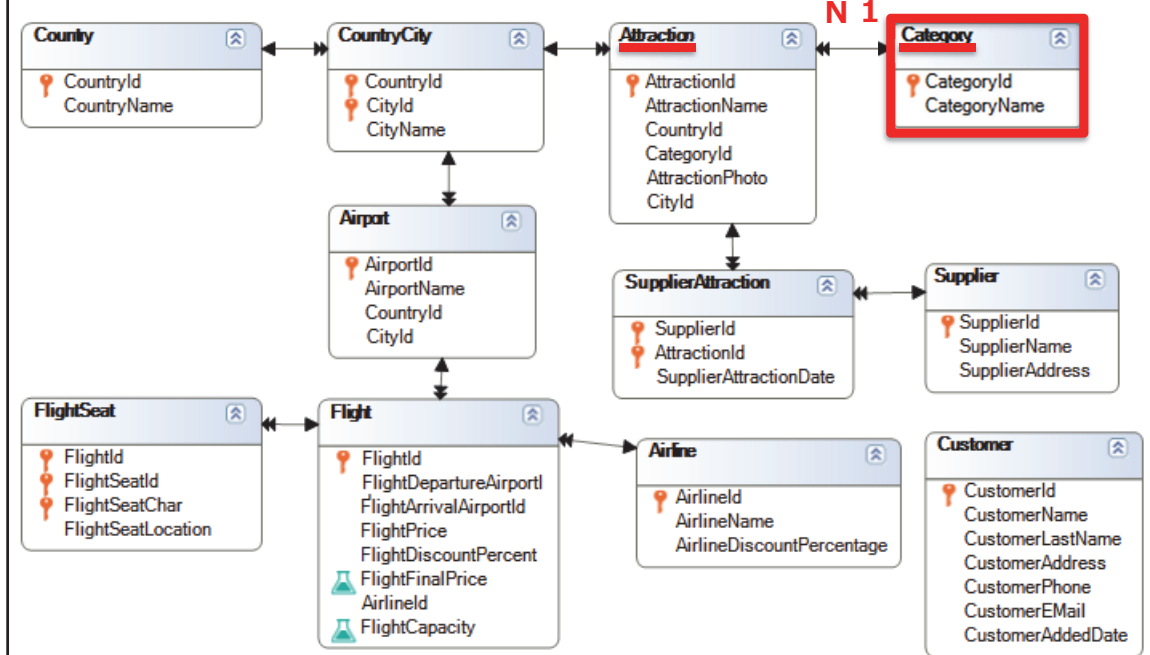


ナビゲートしているベーステーブルの拡張テーブルから必要な情報を取得できる場合は、**そのデータを For each コマンドで直接参照できます**。CountryName の場合がそうでした。

変更後:
ネストされた For each

ネストされた
For each の
ベーステーブル

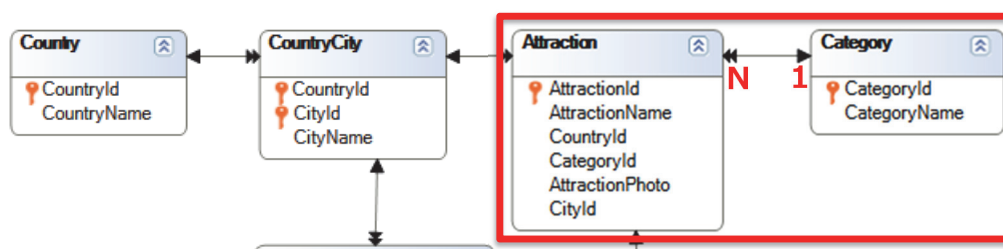
外側の For each の
ベーステーブル



一方、あるテーブル (今回は Category テーブル) のナビゲート中に、処理中の行から複数件の関連するテーブルのレコード群 (今回は Attraction テーブル。**ナビゲート中の Category テーブルの拡張テーブルに属さない、Attraction テーブルにある関連する複数件のレコード=グループ**) を処理するには、**1 つ目の For each コマンドの内側に、もう 1 つ For each コマンドを記述する必要があります。**

つまり、For each コマンド内に別の For each コマンドを**ネスト**することになります。

ソースの設計



プロシーチャーの [Source] に戻り、1 つ目のカテゴリをナビゲートして表示する For each コマンドを記述してみましょう。

For each コマンドの横に「Category」と入力します。これまで見てきた通り、これは **ベーストランザクション** になります。つまり、ナビゲートする情報のトランザクションレベルの名前です。

For each コマンドでアクセスする各カテゴリで、最初に何を行うのでしょうか。出力です。そのため、For each コマンド内で次のように入力します：
print Categories。

Categories プリントブロックには CategoryName 項目属性のみが含まれており、GeneXus は For each コマンドの **ベーステーブル** が Category であることを推論します。CategoryName は、このベーステーブルの拡張テーブルに (この例ではベーステーブル自身) に含まれているため、正しい順序でデータを取得することが可能です。このようにならない場合、GeneXus でエラーが発生します。

カテゴリの出力後、**そのカテゴリの観光名所のグループ**をナビゲートします。そのため、2 つ目の For each コマンドを入力し、ナビゲートするカテゴリの N 個の観光名所をすべて参照する必要があります。

ただし、そのカテゴリの観光名所のグループをナビゲートする直前に、表示される観光名所のタイトルを出力する必要があるため、**print ColumnTitles** の命令を入力します。

以上の理由から、ここで、1 つ目の For each コマンドの本文に、2 つ目の For each コマンドを入力します。

次に、「Attraction」と入力します。Attraction はここでナビゲートしたいテーブルに関連しているトランザクションの名前です。For each コマンド内に「print Attractions」と入力します。

「**endfor**」と入力してこのナビゲーションをクローズし、もう一度「**endfor**」と入力して最初のナビゲーションをクローズします。

GeneXus は、各カテゴリで表示すべき観光名所を指示されなかったにも関わらず、どのようにしてそれを認識できたのでしょうか。

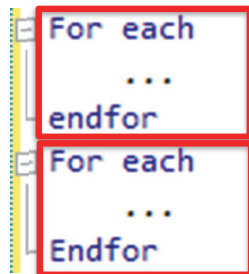
For each コマンドを確認してみましょう。For each コマンドはテーブル内の N 個のレコードを参照し、各レコードに対して、For each コマンド内に記述された一連の命令を実行します。この 1 つ目の For each コマンドの「本文」で、毎回 1 つのカテゴリが処理されます。これを「毎ループ、あるカテゴリが**インスタンス化**される」ともいえます。対象となるのは**ある 1 つのカテゴリ**です。本文の命令の実行が完了してから次のカテゴリに移ります。

そのため、ネストされた For each コマンドの実行を開始する前に、GeneXus では、その時点で対象となるカテゴリを既に認識しています。

このような理由から、**Where** 節なしで、2 つ目の For each コマンドを記述しました。この For each コマンドでは「観光名所のカテゴリが、1 つ目の For each コマンドの現在のカテゴリと一致すること」という条件を記述する必要はありません。

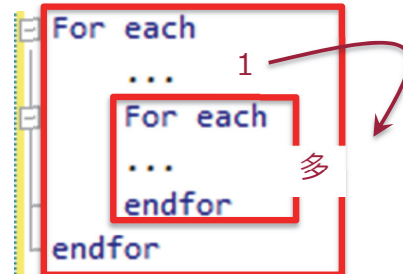
ネストされた For each の暗示的フィルタ

並列の For each



独立した
ナビゲーション

ネストされた For each



関連する
ナビゲーション?

どのようにして GeneXus は開発者の記述なしでフィルタを判断したのでしょうか？
これは、For each コマンドをどう記述するかによって決まります。

2 つの For each コマンドが順に記述されている場合 (スライド左側)、これらのコマンドは**相互に独立**しています。

一方、1 つの For each コマンドの**中**にもう 1 つ For each コマンドが記述されている場合 (スライド右側)、1 つ目のナビゲーションの各レコードに対して、2 つ目のナビゲーションのレコードセット (結果集合) を参照できます。

ネストされた For each コマンドを記述すると、GeneXus は、各 For each コマンドに対してナビゲートされるベーステーブルを決定し、**これらのテーブル間の関連を確認**します。

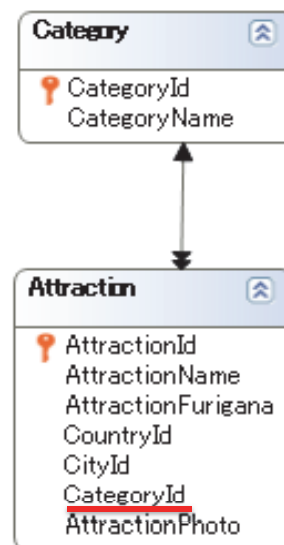
ネストされた For each の暗示的フィルタ

主キー

CategoryId	CategoryName
1	美術館
2	遺跡
3	有名なランドマーク

外部キー

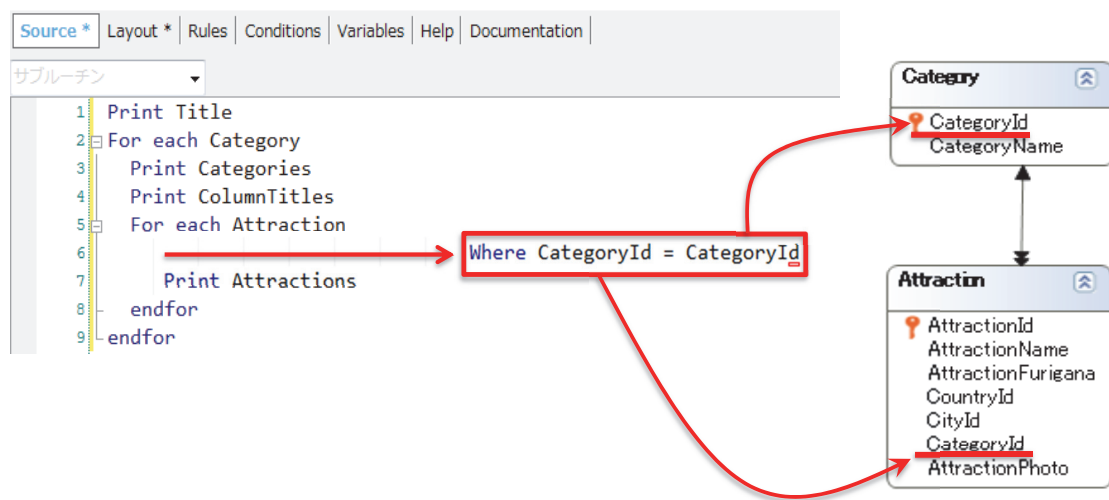
AttractionId	AttractionName	CountryId	CategoryId	...
1	ルーブル美術館	2	1	
2	万里の長城	3	3	
3	エッフェル塔	2	2	
4	コルコバードのキリスト像	1	2	
5	スミソニアン博物館	4	1	



この例の場合、外側の For each コマンドのベーステーブルは Category であり、内側の For each コマンドのベーステーブルは Attraction です。GeneXus は両方のテーブルに共通の項目属性があることを認識します: この共通の項目属性は、**Category の主キーであり、Attraction の外部キーである CategoryId です。**

このように、CategoryId 項目属性は Attraction テーブルと Category テーブルを関連付け、図に示すように 1 対 N の関連を確立しています。つまり、各カテゴリには、関連する観光名所が多数あります。

ネストされた For each の暗示的フィルタ



そのため、外側の For each コマンドでナビゲートする各カテゴリに対して、GeneXus は Attraction テーブルをナビゲートする For each コマンドを実行し、**CategoryId の値が現在参照しているカテゴリの CategoryId と一致する観光名所のみ表示するようフィルタされます。**

これは、内側の For each コマンドに **Where CategoryId=CategoryId** と記述した場合と同じです。GeneXus が検出して適用するため、このように記述する必要はありません。

ナビゲーションリスト

フィルタ:

CategoriesAndAttractions

Procedure CategoriesAndAttractions Navigation Report

Name	Description	Environment	C#	Default (C#)
CategoriesAndAttractions	カテゴリごとの観光名所一覧	Spec. Version	15_0_6-116888	
Output Devices	File	Form Class	Graphic	
Main	Yes	Program Name	CategoriesAndAttractions	
		Call Protocol	HTTP	
		Parameters		

Levels

For Each Category (Line: 7)

Order: CategoryId
Index: ICATEGORY

Navigation Start from: FirstRecord
filters: Loop while: NotEndOfTable
=Category (CategoryId)

For Each Attraction (Line: 20)

Order: CategoryId
Index: IATTRACTION2

Navigation Start from: CategoryId = @CategoryId
filters: Loop while: CategoryId = @CategoryId
Join location: Server
=Attraction (AttractionId)
=Country (CountryId)

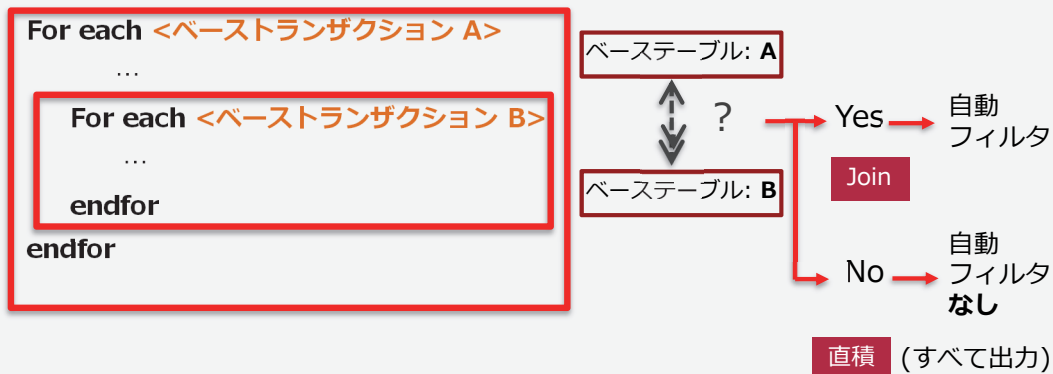
エラー: 0 警告: 0 成功: 1 | すべて

このプロシージャのナビゲーションリストを開くと、2 つの For each コマンドの情報が表示されることを確認できます: 外側の For each のベーステーブルは Category で、ネストされた For each のベーステーブルは Attraction です。また、主キー CategoryId の順序でカテゴリが取得されることも確認できます。観光名所も CategoryId の順序で取得されます。なお、この Attraction テーブルでは CategoryId は外部キーです。この外部キーが、観光名所とカテゴリを関連付けする項目属性であるため、[ナビゲーション表示] 内の [Navigation Filters] で特定のカテゴリの観光名所だけが取得されていることを確認することができます。

情報を取得して簡単にレポートに表示できることを確認しましたが、プロシージャではほかにもさまざまな処理を行えます。詳細については、後で詳しく説明します。

おさらい

- 関連情報



おさらい: ネストされた For each コマンドを記述すると、GeneXus は各 For each コマンドでナビゲートするベーステーブルを決定し、これらのベーステーブル間の関連を確認します。

両方のベーステーブルに関連がある場合、先ほどのリストで確認したように、ネストされた For each コマンドで参照されるレコードに対して自動フィルタが適用されます。ネストされた For each コマンドで、関連の基準によって情報がフィルタされるケースは、**Join** と呼ばれます。

一方、関連がない場合、フィルタは適用されません。外側の For each コマンドの各レコードに対して、ネストされた For each コマンドのすべてのレコードが出力されます。暗示的に関連が検出されないケースにおける、この種のネストされた For each コマンドは、**直積**と呼ばれます。もちろん、開発者は、For each コマンドに Where 節を直接入力することでいつでもフィルタ条件を明示的に追加できます。

これらのケースは、テーブルが異なる場合を想定したものです。次のセクションでは、外側の For each コマンドとネストされた For each コマンドのテーブルが同じである場合に何が起こるかを確認します。

グループ化された情報を リスト表示する方法

グループ化された情報のリスト表示



カテゴリごとの観光名所一覧

カテゴリ 美術館

番号	観光名所名	国	写真
1	ルーブル美術館	フランス	
5	スミソニアン博物館	アメリカ合衆国	

カテゴリ モニュメント

番号	観光名所名	国	写真
2	万里の長城	中国	
3	エッフェル塔	フランス	
4	コルコバードのキリ	ブラジル	

カテゴリ 有名なランドマーク

番号	観光名所名	国	写真
----	-------	---	----

これは既出のカテゴリとその観光名所のリストである。

現時点で、[有名なランドマーク] カテゴリには観光名所が表示されていない。

カテゴリをすべて表示するのではなく、観光名所があるカテゴリのみを表示するにはどうすればよいか。

前のセクションでは、旅行代理店がリクエストするリストで、観光名所のすべてのカテゴリを表示しました。各カテゴリに対して、入力されている観光名所のリストを表示しました。

ここでは観光名所「万里の長城」のカテゴリを変更して、「有名なランドマーク」に関連付けられた観光名所がない状態にします。

その後、もう一度リスト表示を実行します。

ここに示すように、このリストでは登録されているカテゴリがすべて表示され、観光名所が関連付けられていないカテゴリも表示されています。

観光名所が関連付けられているカテゴリのみを表示するにはどうすればよいでしょうか。その方法を説明します。

前の方法が適していない理由

The screenshot shows the GeneXus IDE interface. On the left, the 'Layout' tab is active, displaying a design for a page titled 'カテゴリーごとの観光名所一覧' (List of tourist spots by category). The design includes a table with columns: 'CategoryId', 'CategoryName', 'AttractionId', 'AttractionName', 'CountryName', and 'AttractionImage'. Below the design, two data tables are shown. The first table, 'Categories', has columns 'CategoryId' and 'CategoryName' with rows: (1, '美術館'), (2, '遺跡'), and (3, '有名なランドマーク'). The second table, 'Attractions', has columns 'AttractionId', 'AttractionName', 'CategoryId', and 'AttractionImage' with rows: (1, 'ルーブル美術館', 1), (5, 'スミソニアン博物館', 1), (2, '万里の長城', 2), (3, 'エッフェル塔', 2), and (4, 'コルコバードのキリスト像', 2). A red circle highlights the '3' in the 'CategoryId' column of the 'Attractions' table, and a red arrow points from this circle to the text '「3」がない' (There is no '3'). On the right, the 'Source' tab is active, showing the following code:

```

1 Print Title
2 For each Category
3   Print Categories
4   Print ColumnTitles
5   For each Attraction
6     Print Attractions
7   endfor
8 endfor
9

```

別のプロシージャでこれを実装します。そのため、このオブジェクトを別名で保存します。また、テキストブロックのタイトルを変更します。

これまでに実装していた For each コマンドを確認すると、外側の For each コマンドのベーステーブルが Category で、ネストされた For each コマンドのベーステーブルが Attraction であることが分かります。

この場合、Category テーブルにアクセスし、レコードのデータが出力され、ネストされた For each コマンドが実行されます。結果的に、カテゴリーに観光名所が関連付けられているかどうかを確認する前にカテゴリーが出力されます。

この処理を変更する必要があります。**Attraction テーブルの CategoryId を利用し**、Attraction テーブルからカテゴリーにアクセスする必要があります。観光名所が 1 つ以上関連付けられているカテゴリーの出力を確認するには、この方法しかありません。

解決方法

外部キー					主キー	
AttractionId	AttractionName	CountryId	CategoryId	...	CategoryId	CategoryName
1	ルーブル美術館	2	1		1	美術館
5	スミソニアン博物館	4	1		2	遺跡
2	万里の長城	3	2		3	有名なランドマーク
3	エッフェル塔	2	2			
4	コルコバードのキリスト像	1	2			

カテゴリ			
番号	観光名所名	国	写真
1	ルーブル美術館	フランス	
5	スミソニアン博物館	アメリカ合衆国	

カテゴリ			
番号	観光名所名	国	写真
2	万里の長城	中国	
3	エッフェル塔	フランス	
4	コルコバードのキリ	ブラジル	


```

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

グループ化
または
「ブレイク」
の基準

コントロールブレイク

Attraction テーブルの観光名所をカテゴリごとにグループ化し、そのグループを参照して各カテゴリを出力します。カテゴリ名を取得するには Category テーブルにアクセスする必要があります。グループの各観光名所も出力します。

1 つのグループの処理が終わったら次のグループに移ります。

つまり、必要なのは、Attraction テーブルのみを参照して、最初にカテゴリごとにグループ化してカテゴリを出力し、その後、(同じ Attraction テーブルをナビゲートして) 各カテゴリグループの観光名所を出力することです。

GeneXus では、order 節を指定することで CategoryId でグループ化するよう指示できます。

この、同じテーブルを参照するネストされた For each のケースは、**コントロールブレイク**と呼ばれています。

グループ化された情報のリスト表示: コントロールブレイク

The screenshot shows the GeneXus IDE interface. On the left, a report layout is visible with a title 'カテゴリごとの観光名所一覧' and a table of attractions grouped by category. The table has columns for '番号' (Number), '観光名所名' (Attraction Name), '国' (Country), and '写真' (Photo). The categories are '美術館' (Museum) and 'モニュメント' (Monument). The code on the right is as follows:

```

1 Print Title
2 For each Attraction order CategoryId
3   Print Categories
4   Print ColumnTitles
5   For each Attraction
6     Print Attractions
7   endfor
8 endfor

```

それでは、プロシージャーの変更を開始します。

最初に外側の For each のトランザクションを変更して Attraction を使用するようになります。

次に、CategoryId で並べ替えるように order 節を追加します。コントロールブレイクの場合、order 節に指定した項目属性で**グループ化**されるという点がより重要です。

これを実行します。観光名所が 1 つもない「有名なランドマーク」カテゴリは表示されないことに注目してください。

グループ化された情報のリスト表示: コントロールブレーク

The screenshot displays the 'Procedure CategoriesAndAttractions Navigation Report' in the GeneXus IDE. The report is organized into sections: 'Levels', 'For Each Attraction (Line: 7)', and 'Break Attraction (Line: 16)'. The 'For Each Attraction' section shows the following details:

- Name:** CategoriesAndAttractions
- Description:** カテゴリごとの観光名所一覧
- Status:** No Generation Required
- Output Devices:** File
- Main:** Yes
- Environment:** C# Default (C#)
- Spec. Version:** 15_0_6-116888
- Form Class:** Graphic
- Program Name:** CategoriesAndAttractions
- Call Protocol:** HTTP
- Parameters:**

The 'Levels' section shows the following structure:

- For Each Attraction (Line: 7)**
 - Order: CategoryId
 - Index: IATTRACTION2
 - Navigation: Start from: FirstRecord
 - filters: Loop while: NotEndOfTable
 - Join location: Server
 - Table: Attraction (AttractionId)
 - Table: Country (CountryId)
 - Table: Category (CategoryId)
- Break Attraction (Line: 16)**
 - Order: CategoryId
 - Index: IATTRACTION2
 - Navigation: Loop while: CategoryId = @CategoryId
 - filters:
 - Join location: Server
 - Table: Attraction (AttractionId)
 - Table: Country (CountryId)

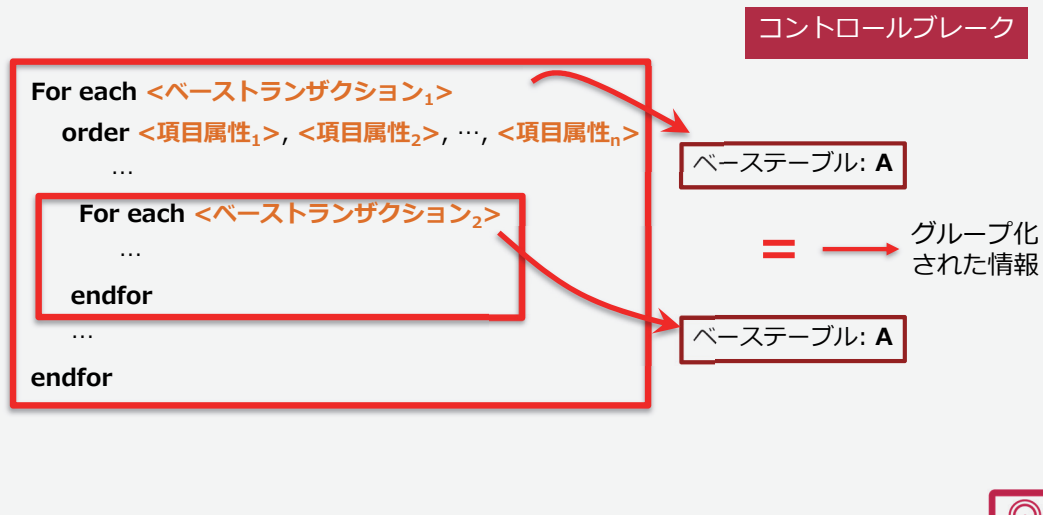
The status bar at the bottom indicates: エラー: 0, 警告: 0, 成功: 1.

この結果、ナビゲーションリストでスライドの赤い下線で示した内容が確認できます。

- Attraction テーブルに対する For each
- CategoryId で並べ替え
- (CategoryId で) ブレイク処理
- ネストされた For each も Attraction テーブルがベーステーブル
- ネストされた For each では、そのグループのカテゴリに含まれる観光名所のみ参照

概念化

- 特定の基準で情報をブレイクする
- ✓ **すべてで同じベーステーブル**



ここではコントロールブレイクの定義を概念化します。

1. ネストされた For each
 2. 各 For each は、同じベーステーブル
- これらは、コントロールブレイクの定義と結びつきます。

概念化

- 特定の基準で情報をブレイクする

✓ ブレイクの基準:
order 節

For each <ベーストランザクション₁>

order <項目属性₁>, <項目属性₂>, ..., <項目属性_n>

...

For each <ベーストランザクション₂>

...

endfor

...

endfor

ブレイクの基準

ベーステーブル: A

これらの項目で
グループ化



- order 節でブレイクの基準を指定します。

まとめ

まとめ

- リスト表示

- ✓ レイアウト
- ✓ ソース

**For each
コマンド**



カテゴリごとの観光名所一覧				
番号	カテゴリ	観光名所名	国	写真
1	ルーブル美術館	フランス		
5				
4				
3				

カテゴリごとの観光名所一覧				
番号	カテゴリ	観光名所名	国	写真
1	ルーブル美術館	フランス		
5	スミソニアン博物館	アメリカ合衆国		
3				
4	モニュメント			

カテゴリごとの観光名所一覧				
番号	カテゴリ	観光名所名	国	写真
2	万里の長城	中国		
3	エッフェル塔	フランス		

このセクションでは、GeneXus が、各ケースの判断をどのように進めているのか説明しました。

- 1: 1 つのテーブルをナビゲートする単純なケース
 - 2: 関連のある複数のテーブルをナビゲートするケース (Join)
 - 3: 同じテーブルを特定の基準でグループ化するケース (コントロールブレイク)
- 説明のみ: 関連のない複数のテーブルをナビゲートするケース (直積)

これらすべてのケースでデータベースにアクセスするために使用するコマンドが、**For each** です。

まとめ

- リスト表示

- ✓ レイアウト
- ✓ ソース

観光名所一覧



番号	観光名所名	国	写真	カテゴリ
5	スミソニアン博物館	アメリカ合衆国		美術館
4	コロンボードのキリ	ブラジル		モニュメント
3	エッフェル塔	フランス		モニュメント

For each
コマンド



1

```
For each <ベーストランザクション>
order <項目属性1>, <項目属性2>, ..., <項目属性n>
where <条件1>
where <条件2>
...
where <条件n>
<メインのコード>
endfor
```

最初の例では、単純な For each を使用して、ベーストランザクションでナビゲートするテーブルを推論しました。

まとめ

- リスト表示

- ✓ レイアウト
- ✓ ソース

For each
コマンド



カテゴリごとの観光名所

カテゴリ	観光名所	国	写真
美術館	ルーブル美術館	フランス	
	スミソニアン博物館	アメリカ合衆国	
モニュメント	万里の長城	中国	
	エッフェル塔	フランス	
	コロンバードのキリ	ブラジル	
有名なランドマーク			

```
For each <ベーストランザクション>
order <項目属性1>, <項目属性2>, ..., <項目属性n>
where <条件1>
where <条件2>
...
where <条件n>
...
```

```
For each <ベーストランザクション>
order <項目属性1>, <項目属性2>, ..., <項目属性n>
where <条件1>
where <条件2>
...
where <条件n>
...
```

endfor

endfor

ネストされた for each

Join

2 つ目の例では、ネストされた For each のペアで、異なるベーストランザクションの各 For each の情報から 1 対多の関係を検出しました。

まとめ

- リスト表示

- ✓ レイアウト
- ✓ ソース

For each
コマンド



カテゴリごとの観光名所一覧

番号	観光名所名	国	写真
3	ルーブル美術館	フランス	
5	スミソニアン博物館	アメリカ合衆国	

番号	観光名所名	国	写真
2	万里の長城	中国	
3	エッフェル塔	フランス	

For each <ベーストランザクション>
order <項目属性₁>, <項目属性₂>, ..., <項目属性_n>

where <条件₁>

where <条件₂>

...

where <条件_n>

...

For each <ベーストランザクション>
order <項目属性₁>, <項目属性₂>, ..., <項目属性_n>

where <条件₁>

where <条件₂>

...

where <条件_n>

...

endfor

...

endfor

ネストされた For each

コントロールブレイク

3 つ目の例でもネストされた For each を使用しましたが、この例では、各 For each のベーストランザクションが**同じです**。その後 GeneXus が、外側の For each の order 節で指定した項目属性または項目属性の組み合わせによって、参照**対象**テーブルの情報を「ブレイク処理」または「グループ化」します。

その他

- リスト表示

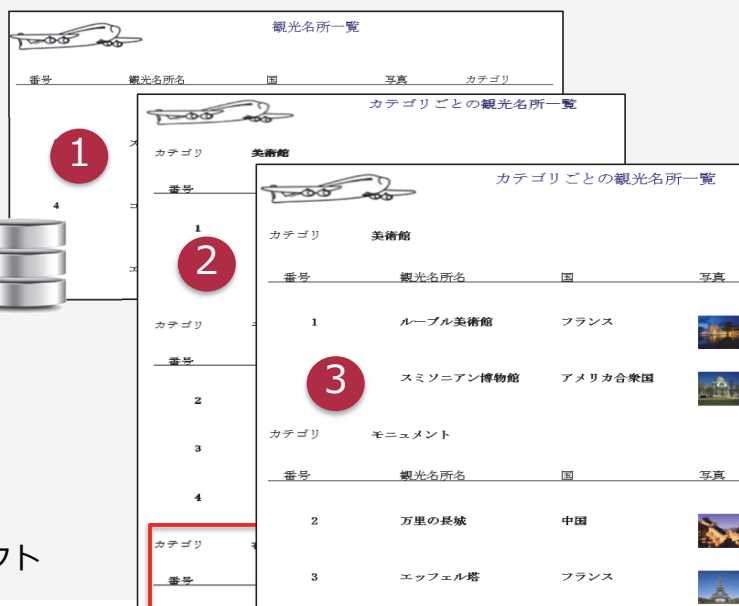
- ✓ レイアウト
- ✓ ソース

For each
コマンド



- パネル

- Query オブジェクト



この後は、柔軟かつ魅力的な方法で情報を取得するための、データベースに対する問い合わせを実装する別のメカニズムを確認します。

