

## 関連情報をリスト表示する方法

ネストされた For each コマンド

GeneXus™

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

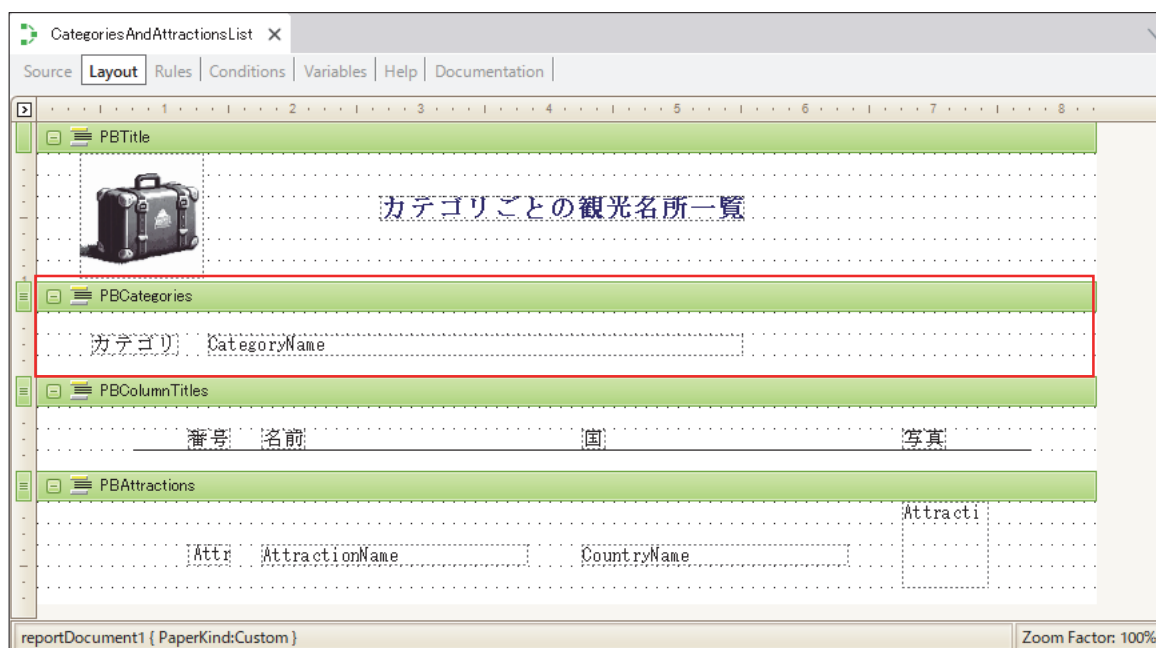
列見出し

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	名前	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	
カテゴリ: 遺跡			
番号	名前	国	写真
2	万里の長城	中国	
カテゴリ: モニュメント			
番号	名前	国	写真
3	エッフェル塔	フランス	
4	サグラダ・ファミリア	スペイン	

もし、登録されているデータのうち、ある外部参照テーブルのデータを基準に、該当するデータをすべて表示してほしいという要望があった場合、対応できるでしょうか。今までの一覧と異なる点としては、データのグルーピングが行われ、すべての登録されたレコードを表示している点です。

## [Layout] エlement



データのグルーピングが必要となる場合、グルーピング**する**データを出力するためのプリントブロックと、グルーピング**される**データを出力するためのプリントブロックは、異なる必要があります。

また、ここまでの章で触れている通り、[Layout] Elementにおいて、プリントブロックの定義順は、出力順と一致しません。そのため、定義する位置は、出力を想定した位置と同じ必要はありません。

## 必要な処理の想定

## CATEGORY の参照

CategoryId	CategoryName
1	美術館
2	遺跡
3	モニュメント

## ATTRACTION の参照

AttractionId	AttractionName	CountryId	CategoryId	...
1	ルーブル美術館	2	1	
2	万里の長城	3	2	
3	エッフェル塔	2	3	
4	サグラダ・ファミリア	9	3	
5	スミソニアン博物館	10	1	

## ATTRACTION の参照終了

## CATEGORY の参照終了

外部参照となるテーブルのデータでグルーピングが必要なケースの処理について想定します。

For each コマンドでデータを参照するベーステーブルは、1 つしか指定することができず、ベーステーブルのレコード 1 行に対して、処理が 1 回実施されます。

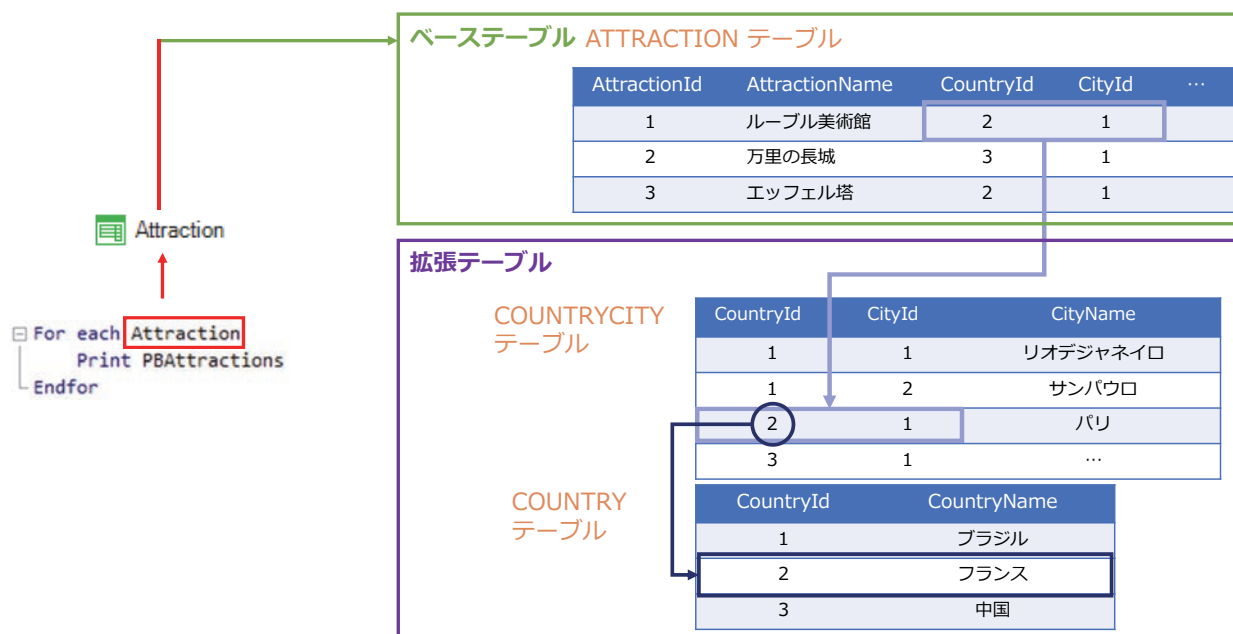
そのため、これまでに紹介した実装ではこのケースを対応できません。

想定される処理に基づき、より詳しく検討すると、外部参照となるテーブルのデータを 1 行参照し、これに関連するグルーピングしたいテーブルのデータから該当レコードのみを対象に反復処理を行うように参照する必要があります。

つまり、2 種類の反復処理が必要となります。

このような実装方法について、これまでに紹介したケースと比較して説明します。

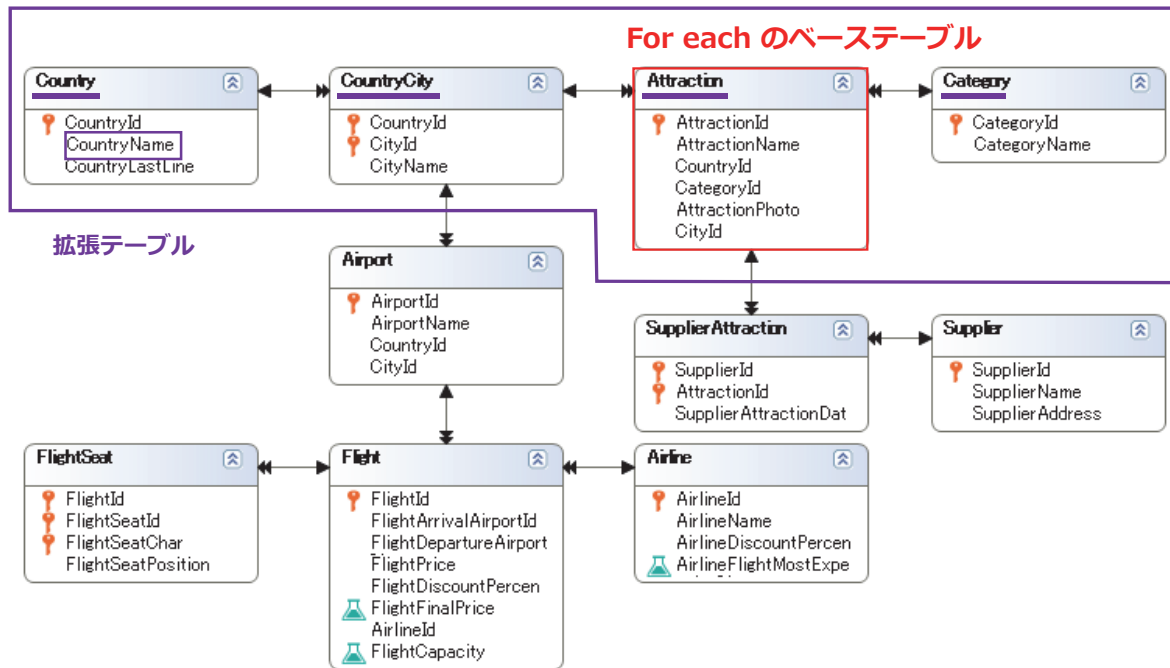
## 観光名所一覧：単純な For each コマンド



ここまでに紹介した実装の場合、1 つのベーステーブルを参照し、そのベーステーブルに含まれるすべてのデータを出力するケースを紹介しました。

このケースでも、別のテーブルを参照する必要がありましたが、あくまでもベーステーブルに対する拡張テーブルの参照でした。  
つまり、ベーステーブルの特定のデータに基づき、1 行のデータが特定できる参照です。

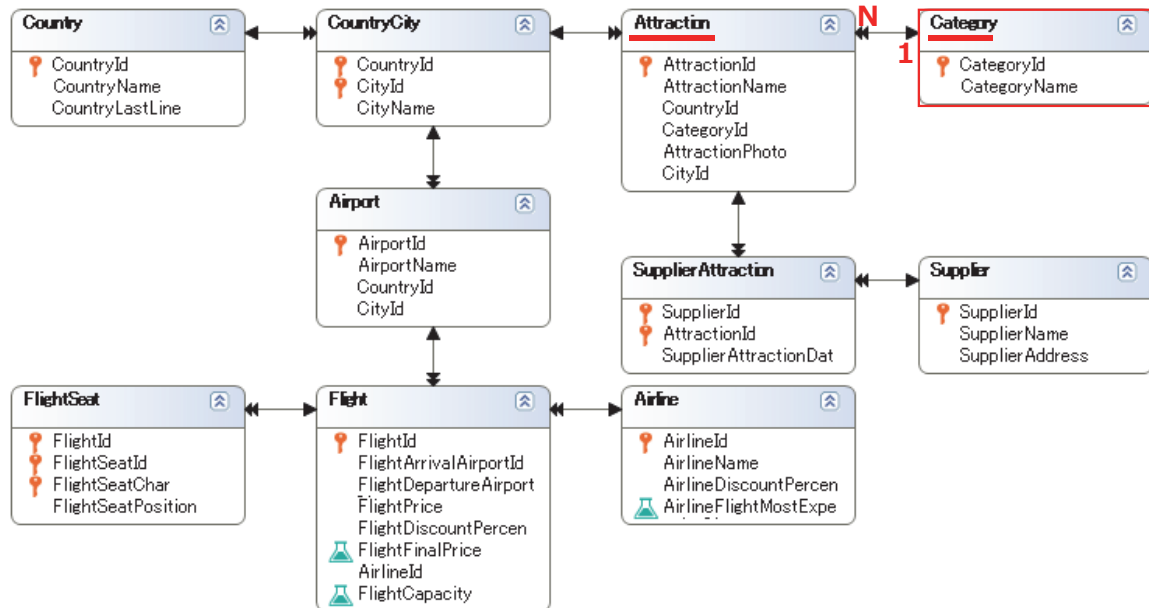
## 観光名所一覧：拡張テーブルの範囲



ベーステーブルとその拡張テーブルを確認する場合、ダイアグラムオブジェクトで、テーブルオブジェクトを配置することで確認が容易でした。  
 ここまでの説明の通り、特定のテーブルがベーステーブルとして決定した場合、拡張テーブルは、ベーステーブルを N としたとき、直接的または間接的に 1 の関係となるすべてのテーブルと、ベーステーブル自身でした。

ここまでの章で実装した For each コマンドも特定のベーステーブルと、その拡張テーブルを参照していました。

## テーブル間の関係：グルーピングするテーブル



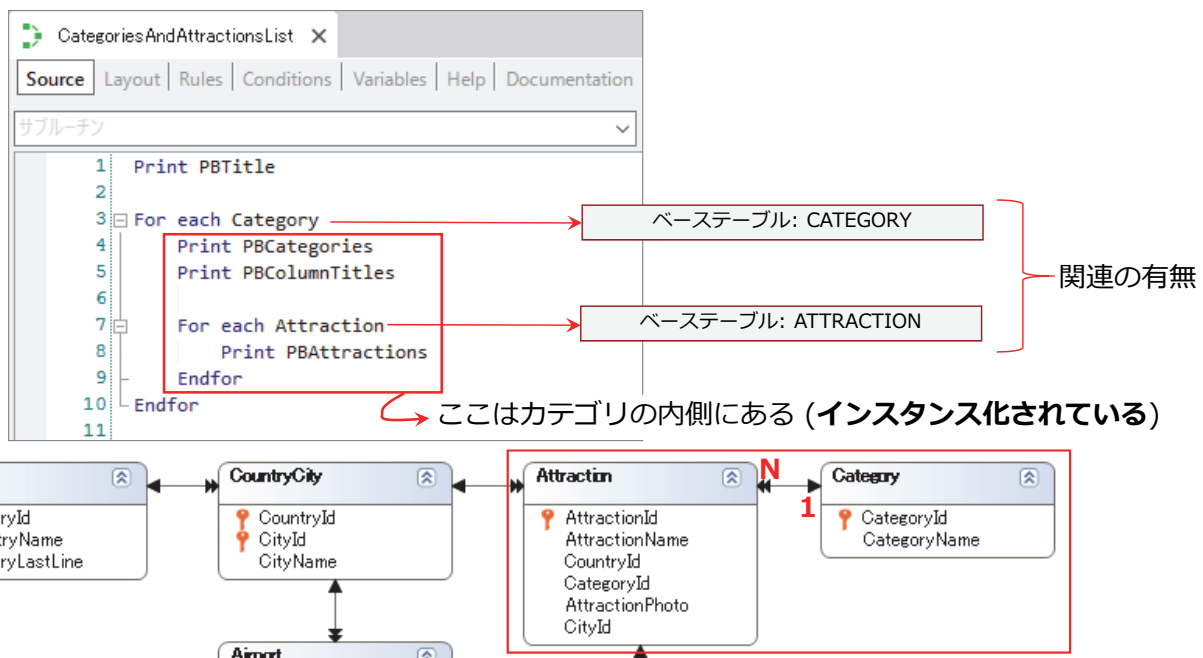
「必要な処理の想定」のページで 2 種類の反復処理が必要と記述しましたが、GeneXus において、特定のテーブルを参照し、反復処理を行うためには、For each コマンドを利用する必要があります。

そして、For each コマンドで、あるテーブルを参照中に、特定の行に関連した別テーブルの複数件のレコード群を対象とする場合、異なる For each コマンドが必要となります。

既に参照中のテーブルに関連し、別のテーブルを参照する必要があるため、新たに記述する For each コマンドは、既存の For each コマンドのメインのコードとして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 コマンドを入れ子にするネストされた記述になります。

## [Source] エLEMENTの実装



[Source] エLEMENTで、実際にネストされた For each コマンドを記述します。  
 1 つ目の For each コマンドは、グルーピングするデータを記録しているテーブルがベーステーブルとなるようにベーストランザクション節を指定します。  
 この For each コマンドのメインコードとして、まずはこのテーブルに登録されたデータを出力する Print コマンドを記述します。  
 続いて、実際にグルーピングされた N 件のデータを表示するため、ネストされた For each コマンドを記述します。  
 ネストされた For each コマンドにもベーストランザクション節を記述し、異なるベーステーブルを参照するように定義します。

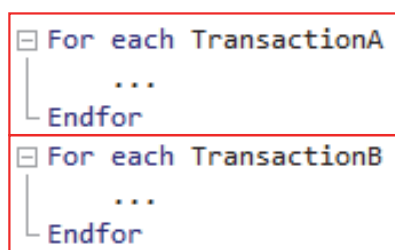
この実装によって、外側の For each コマンドでグルーピングされ、ネストされた For each コマンドは一部のデータのみを対象に処理を行います。  
 しかし、この 2 つの For each コマンドには、明確な条件を指定していません。  
 なぜ、このような挙動となるのでしょうか。

これは、外側の For each コマンドの反復処理において、対象となるレコードが確定し、処理対象のレコードがインスタンス化されています。  
 そのため、外側の For each コマンド内に記述された処理は、ベーステーブルの特定のレコードに対する処理が行われます。  
 この結果、ネストされた For each コマンドは、外側の For each コマンドのレコードを自動で認識し、条件とします。



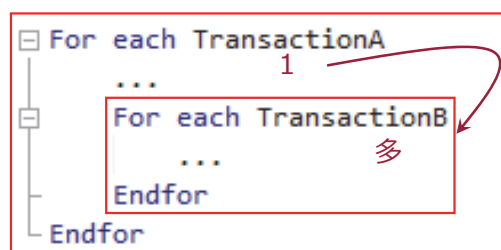
## ネストされた For each コマンドの暗示的フィルタ①

### 並列の For each コマンド



独立したナビゲーション

### ネストされた For each コマンド



ナビゲーションの関連を確認

ネストされた For each コマンドがなぜフィルタを明記せずに、動作するか説明していきます。

はじめに、この挙動については、For each コマンドの記述方法によって決定する挙動があります。

For each コマンドを順番に、つまり並列して記述した場合（スライド左側）、各 For each コマンドは独立し、影響は受けません。

For each コマンドの中に、別の For each コマンドを記述、つまりネストされた記述を行った場合（スライド右側）、1 つ目の反復処理において、各レコードごとに、2 つ目の反復処理が行われます。

この時、GeneXus は、外側の For each コマンドと、ネストされた For each コマンドのベーステーブル間に関連性の有無を確認し、実行する挙動を決定します。

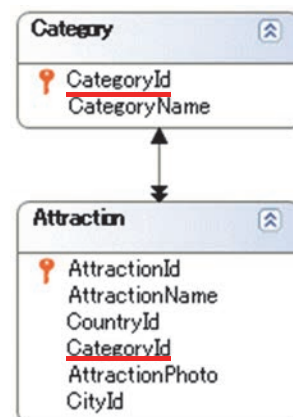
## ネストされた For each コマンドの暗示的フィルタ②

主キー

CategoryId	CategoryName
1	美術館
2	遺跡
3	モニュメント

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	2	
3	エッフェル塔	2	3	
4	サグラダ・ファミリア	9	3	
5	スミソニアン博物館	10	1	



外側の For each コマンドと、ネストされた For each コマンドのベーステーブルが決定された後、GeneXus はこれらのテーブルの共通の項目属性があるか確認します。

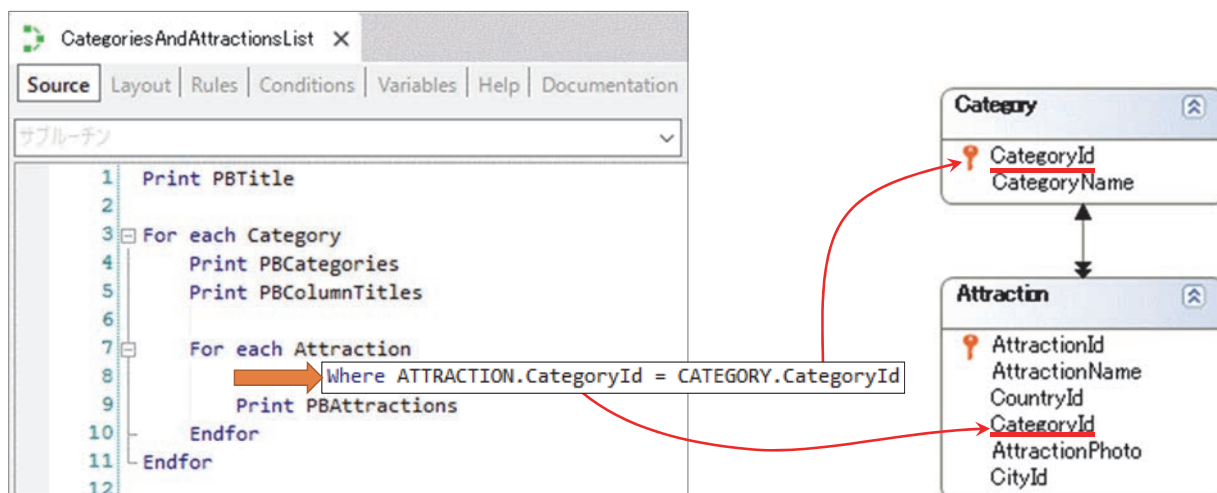
確認の結果、共通の項目属性がある場合、続けてこれらのベーステーブル間の関係性を確認します。

関係性については、考えられるケースは次の通りです。

- ・外側の For each が 1 となるとき、ネストされた For each が N となる 1 対 N
- ・外側の For each が N となるとき、ネストされた For each が 1 となる N 対 1
- ・2 つのベーステーブルに一切の関係性がない

本コースでは、1 対 N のケースを対象に説明を続けます。

### ネストされた For each コマンドの暗示的フィルタ③



外側の For each コマンドが対象とするベーステーブルと、ネストされた内側の For each コマンドが対象とするベーステーブルの関係性が 1 対 N の場合、共通の項目属性を対象に暗示的フィルタが適用されます。

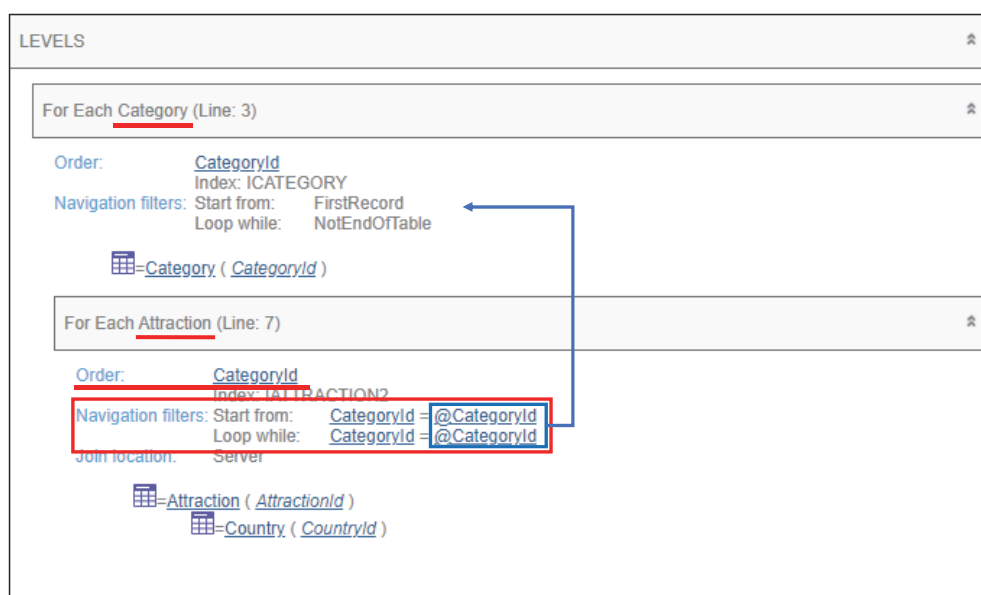
つまり、外側の For each コマンドで参照しているレコードを外部参照しているレコードのみが、ネストされた内側の For each コマンドで参照の対象となります。この場合、ネストされた For each コマンドに、Where 節を明記する必要はありませんが、実態としては、下記のような条件が記載されている状態と同様です。

Where <ネストされた内側のベーステーブル>.<共通項目属性> =  
<外側のベーステーブル>.<共通項目属性>

ただし、条件の記述にベーステーブルの指定は記述できず、記載可能な名前は、項目属性名のみです。

この場合、イコールの左右に同じ項目属性名を記載することとなり、明記すると意図した結果は得られないため、GeneXus によって自動で検出し、実装するフィルタに一任します。

## ナビゲーション表示



ネストされた For each コマンドを含むプロシージャオブジェクトのナビゲーション表示を確認します。

ナビゲーション表示でも、For each コマンドがネストして定義されていることがわかるように表示されます。

この場合、単純な For each コマンドの定義と異なる点として、ネストされた For each の表示に注目します。

Order 節は定義していませんが、自動で外側の For each コマンドと共通の項目属性が利用されるようになります。

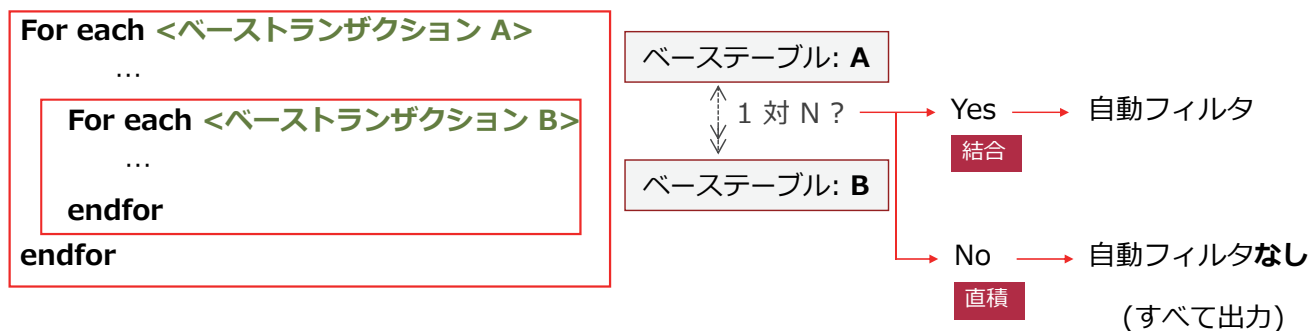
そして、Navigation filters においても前述の通り自動で条件が設定されています。

イコールの左辺は、ネストされた For each コマンドで参照するベーステーブルの項目属性を意図し、右辺は、同名の項目属性が指定されていますが、接頭辞として、「@」があります。

これは、外側の For each コマンドで参照中のレコードの値を参照することを意図しています。

つまり、外側の For each コマンドで参照した共通項目属性の値と一致するレコードの間、反復処理を行うということが明示されています。

## まとめと補足



この章で説明をしてきた内容についてまとめと、一部追加の情報を補足として説明します。

ネストされた For each コマンドを記述すると、GeneXus は各 For each コマンドで参照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 1 対 N の関連がある場合、説明の通り、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、関連に基づき、情報がフィルタされるケースは、結合と呼ばれます。

一方、一切の関連がない場合、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。ネストされた For each コマンドで、関連が検出されず、フィルタされないケースは、直積と呼ばれます。

## 1 対 N の関係性

### 結合

```
For each Country.City
  Print PBCountryCity
```

↑ 1  
↓ N

直接の 1 対 N

```
For each Attraction
  Print PBAttraction
Endfor
```

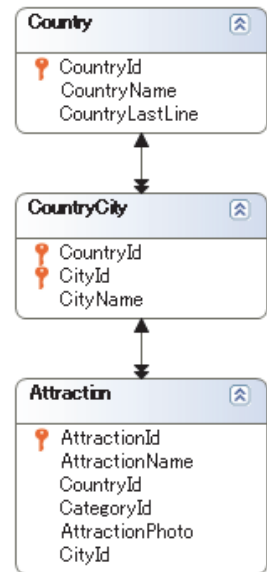
```
Endfor
```

間接の 1 対 N

```
For each Country
  Print PBCountry
```

↑ 1  
↓ N

```
For each Attraction
  Print PBAttraction
Endfor
```



最後に、ネストされた For each において、2 種類の 1 対 N があることを明確にしておきます。

1 つは、直接の 1 対 N です。外側の For each コマンドとネストされた For each コマンドのベーステーブルは、直接の関連性があり、1 対 N の関係がある場合です。

もう 1 つは、間接の 1 対 N です。外側の For each コマンドとネストされた For each コマンドのベーステーブルは、直接 1 対 N の関係はありません。

しかし、外側のベーステーブルと 1 対 N となるテーブルがあり、このテーブルが、ネストされたベーステーブルと 1 対 N の関係である場合、間接的に 1 対 N の関係がある場合です。

このケースは、ネストされたベーステーブル側から見た場合、外側のベーステーブルは、拡張テーブルの一部に含まれます。

*GeneXus*<sup>TM</sup>

[training.genexus.com](http://training.genexus.com)  
[wiki.genexus.com](http://wiki.genexus.com)