

プロシージャとリスト

データベースにアクセスするためのコマンド

GeneXus™

想定される要望

- ✓ さまざまな目的でデータベース内の特定テーブルにアクセスするプロセスの定義



- ✓ 新規レコードの登録や条件に一致するレコードの更新、削除

| Id | Name | Visites |
|----|---------|---------|
| 1 | ルーブル美術館 | 8245 ← |
| 2 | 万里の長城 | 10122 |
| 3 | エッフェル塔 | 11734 ← |

- ✓ 任意のレコードをリストで出力

| 観光名所リスト | | |
|---------|---------|--------|
| 番号 | 名前 | 訪問者 |
| 1 | ルーブル美術館 | 8,245 |
| 2 | 万里の長城 | 10,122 |
| 3 | エッフェル塔 | 11,734 |

- ✓ 検索・計算・更新等のプロセス定義 (いわゆるユーザー定義の関数)



ここまでの内容では、データモデルの定義を中心に説明してきました。

しかし、業務で利用するアプリケーションを生成する場合、他にも多数の要望が想定されます。

その中の一部を挙げています。

まず、1 つはさまざまな目的でデータベース内に作成されたテーブルへのアクセスです。この点については、少し細かく見てみると、特定のテーブルに新規レコードを登録したり、ある条件に一致するレコードの更新や削除がケースとして考えられます。

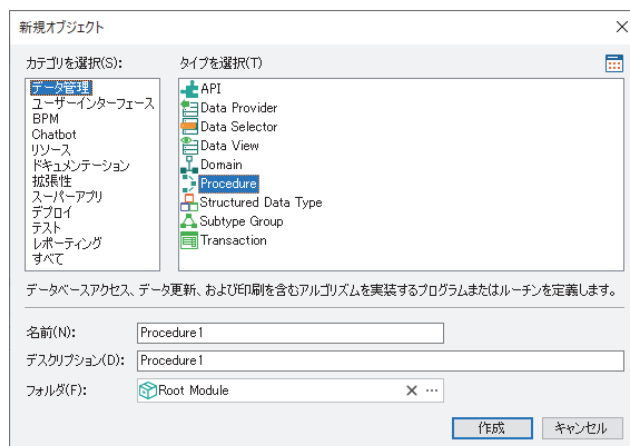
その他、レコードを参照すること考えられます。

参照した結果、これをリストとして出力することも想定されます。

別の要望としては、データの検索や、特定の計算、データの更新をプロセスとして定義すること。

言い換えれば、このプロジェクト固有の関数を定義するケースが想定されます。

Procedure オブジェクト



ここで挙げた想定される要望は、すべて Procedure オブジェクトを利用することで実装が可能です。

この章では、その中でデータベースへアクセスし、任意のレコードをリストとして出力する方法について説明します。

要望 : データを PDF 形式で出力

- ✓ 旅行代理店が扱うすべての**観光名所**のリストを、**観光名所名順**で PDF ファイルに**出力**する

固定のコンテンツ
(タイトル、画像)

可変コンテンツ
(データ)

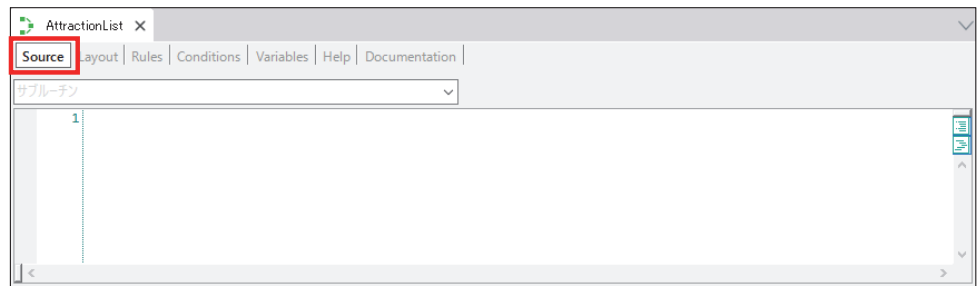
| 観光名所一覧 | | | |
|--------|------------|------|--|
| 番号 | 名前 | 国 | 写真 |
| 3 | エッフェル塔 | フランス |  |
| 4 | サグラダ・ファミリア | スペイン |  |
| 1 | ルーブル美術館 | フランス |  |
| 2 | 万里の長城 | 中国 |  |

GeneXus で、Procedure オブジェクトを利用し、印刷可能な形式で出力する場合、選択肢は PDF 形式での出力となります。

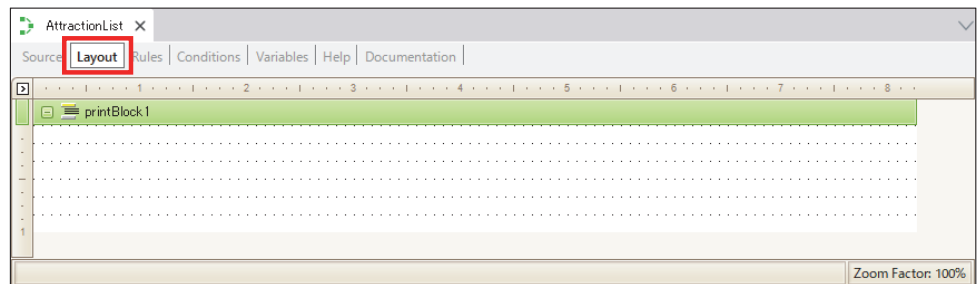
ユーザーからの要望に基づき、GeneXus でどのように PDF を出力できるように定義すればいいか説明していきます。

プロシージャー: [Source] と [Layout]

- ✓ [Source]: 実行する命令



- ✓ [Layout]: 出力のデザイン



Procedure オブジェクトを新規作成すると、初めに [Source] エlementが表示されます。

[Source] Elementでは、このオブジェクトで目的となる処理を実行するためのコーディングを行います。

今回の要望に基づく場合、PDF を出力することが目的となるため、[Source] Elementには、そのためのコードを記述します。

また、このElementで記述されたコードは、手続き型となり、記述された内容を上から順に実施するように記述します。

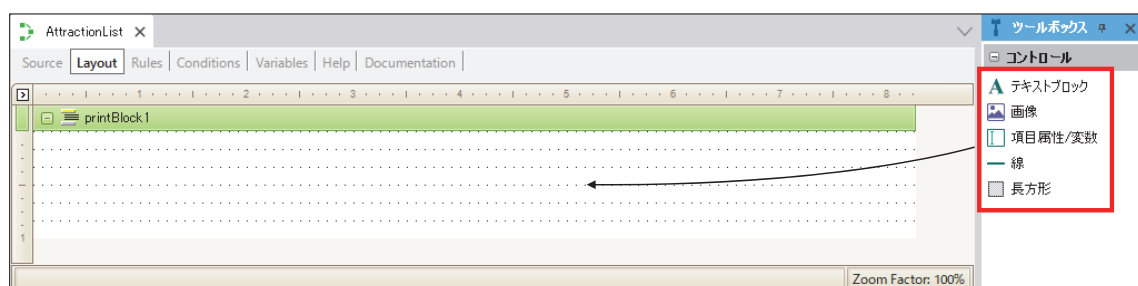
PDF の出力を行う場合、[Layout] Elementの定義も必要です。

このElementでは、出力するデザインの定義を行います。

ここに定義されたデザインに基づき、固定の内容や動的な内容、つまりデータの出力が行われます。

[Layout] エlement : 概要

- ✓ 複数の「プリントブロック」で構成される



- ✓ 既定で 1 つの「プリントブロック」が作成されている (printBlock1)

[Layout] Element は「プリントブロック」という単位で構成され、プリントブロックの中に出力するデザインを定義します。

デザインの定義のためには、[ツールボックス] ウィンドウから任意のコントロールをドラッグします。

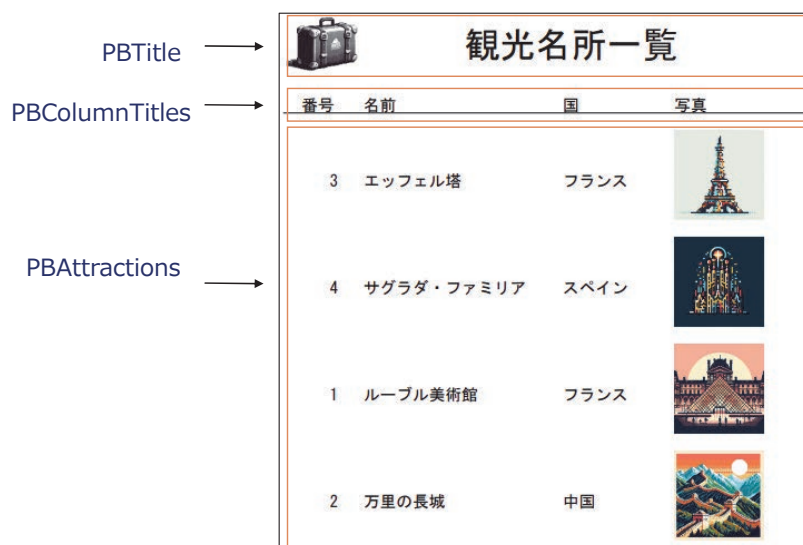
利用可能なコントロールは、タイトルなど固定の文字列を表示するための「テキストブロック」、「画像」、データや可変の値を表示するための「項目属性/変数」、縦または横にひくことができる「線」、四角形を配置することができる「長方形」が用意されています。

[Layout] Element には、新規作成時に必ず「PrintBlock1」というプリントブロックが作成されます。

このプリントブロックを必要に応じて複数デザインし、ユーザーからの要望に基づく出力内容を定義していきます。

[Layout] エlement : 要望に基づくプリントブロックの決定

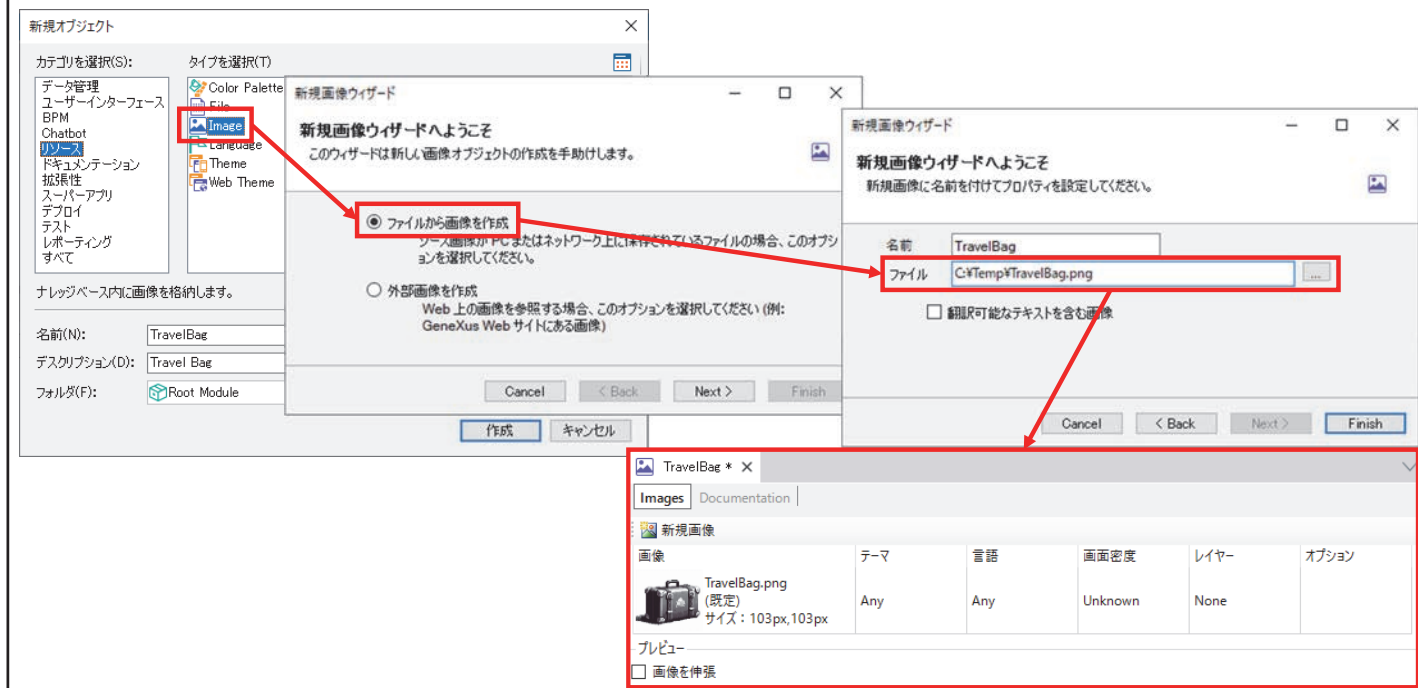
✓ プリントブロックの分割



[Layout] Elementでデザインを始める前に、ユーザーからの要望を分析し、プリントブロックの定義数を決定します。

もっとも一般的な分割としては、表示している内容のカテゴリで分割します。
例えば、PDF のタイトルまたは全体にかかわり表示する部分、表示する可変データの列のタイトル、可変データ表示領域のような形をとります。

ナレッジベース内へ画像を追加

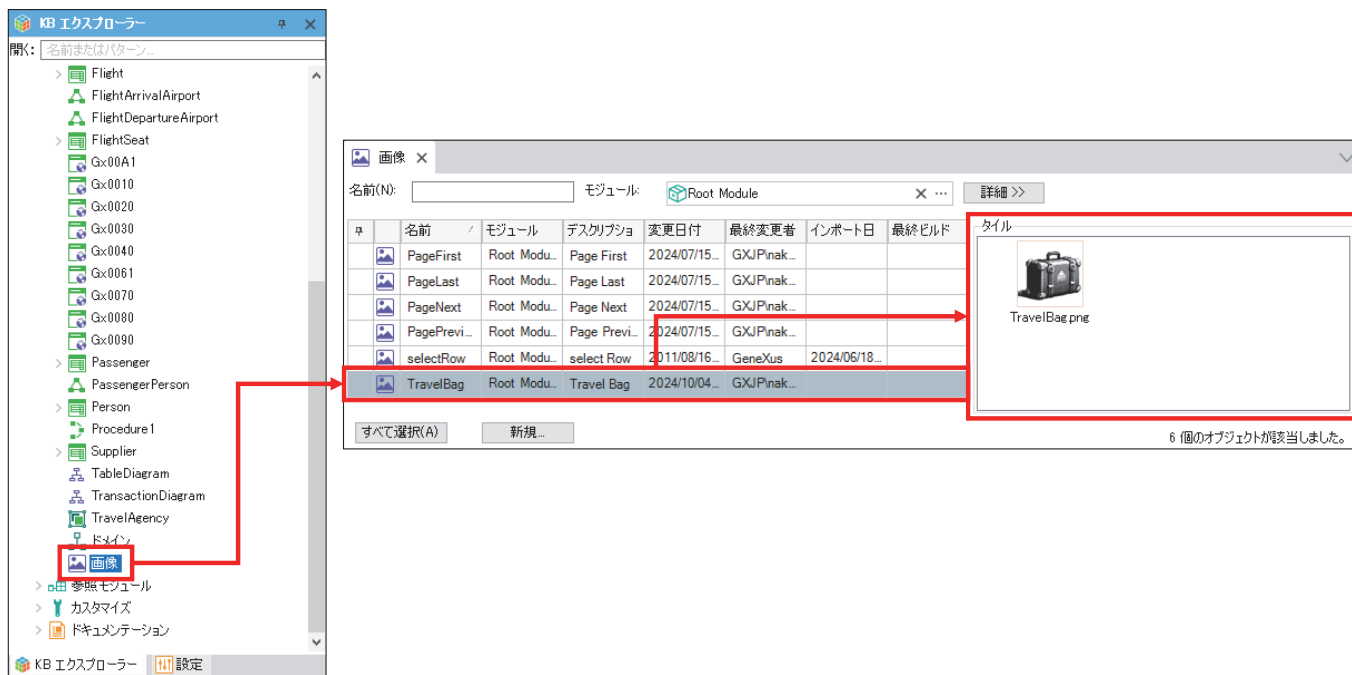


もし、ナレッジベースで定義を行うデザインで、固定の画像を利用したい場合、画像をナレッジベースに追加する必要があります。

画像もナレッジベース内の一つのオブジェクトとなるため、新規オブジェクトの作成から Image オブジェクトを作成します。
新規画像ウィザードが表示されるため、「ファイルから画像を作成」を選択し、ローカルのファイルを選択します。

Image オブジェクトが追加されたため、保存することで、ナレッジベース内の画像コントロールを配置した際に表示する画像の 1 つとして利用可能です。

ナレッジベース内の画像確認方法

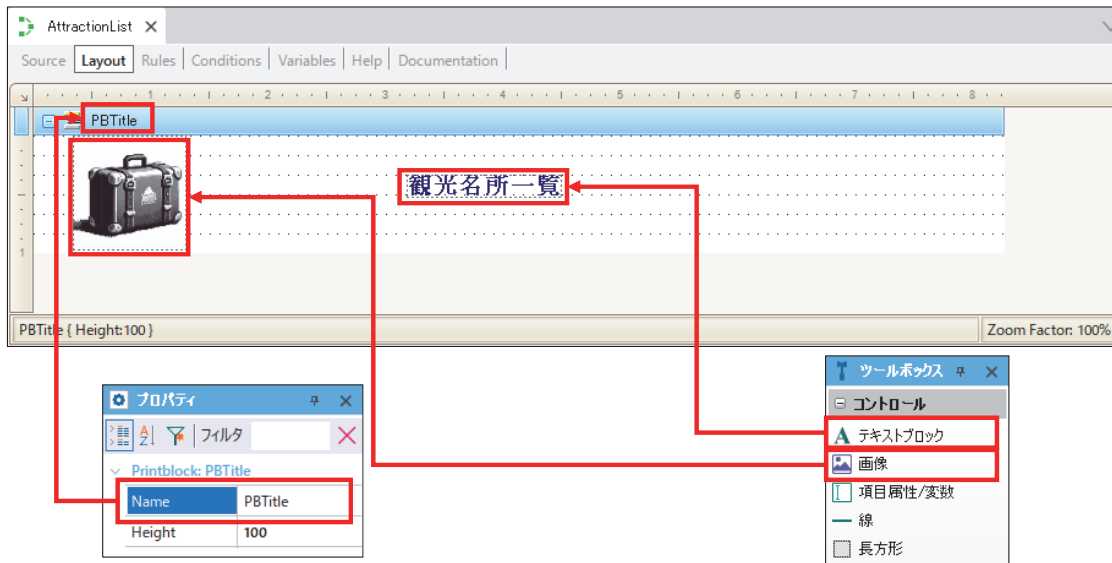


ナレッジベースで利用可能なすべての画像を確認する場合、[KB エクスプローラー] ウィンドウの [Root Module] ノード内に含まれる [画像] を開きます。

既定でナレッジベースに含まれている画像に加え、ナレッジベースに追加した画像がすべて表示されます。

[Layout] エLEMENTの定義①

✓ プリントブロック: PBTitle



プリントブロックを定義する際のポイントを記載します。

前述の通り、プリントブロックには、固有の名前を設定することができます。この名前を設定する場合、プリントブロックを選択した状態（プリントブロックのタイトルバーが水色）で、[F4] キーの押下などで、[プロパティ] ウィンドウを表示します。

[Name] プロパティがプリントブロックの名前を設定するプロパティになるため、任意の名前に変更できます。

デザインの定義として配置できるものに画像がありました。画像コントロールをツールボックスからドラッグし、配置すると、ナレッジベース内に含まれた画像の中から配置する画像を選択できます。この時に、新規画像を作成し、これを配置することもできます。

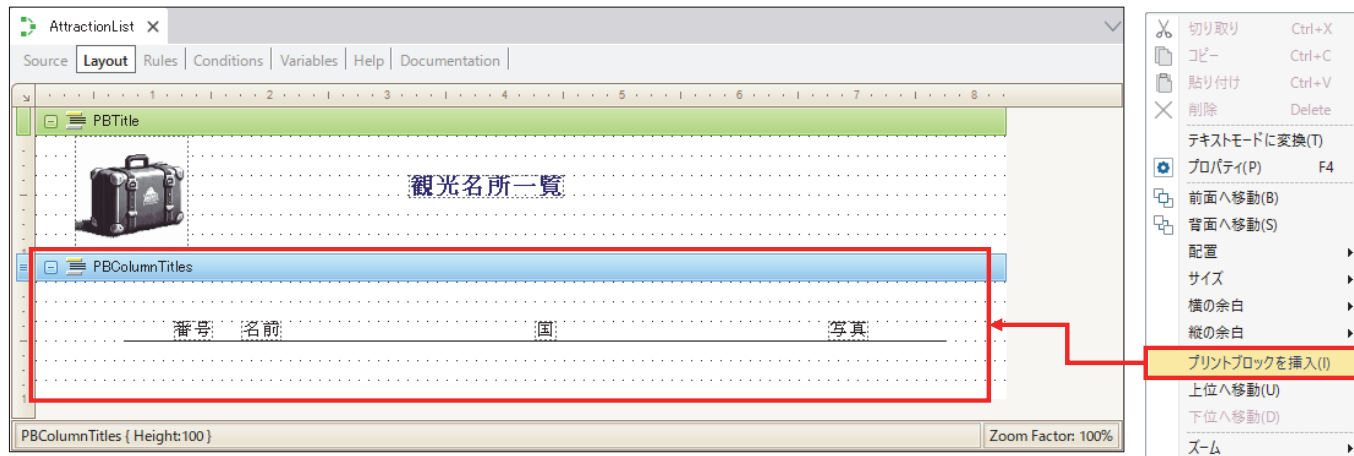
テキストブロックコントロールを配置すると、印刷されるリストのタイトルなど固定の文字列をデザインできます。

表示する文字列は、[Text] プロパティから変更でき、[Fore Color] プロパティから文字の色や、[Font] プロパティで、文字サイズや太字などの設定もできます。

※フォントを変更する場合、「MS 明朝」、「MS ゴシック」以外を利用する場合、フォントの埋め込みが必要です。

[Layout] エLEMENTの定義②

✓ プリントブロック: **PBColumnTitles**



プリントブロックが複数必要な場合、[Layout] エLEMENTの任意の場所を右クリックし、表示されるオプションから [プリントブロックを挿入] を選択します。
この操作により、[Layout] エLEMENTに新規プリントブロックが追加されます。

プリントブロックの追加される位置は、右クリックした位置に依存します。
ただし、プロシージャオブジェクトを利用した PDF 出力は、[Layout] エLEMENTでのプリントブロック表示順で出力はされません。
そのため、プリントブロックが表示されている順序は重要ではありません。

しかし、デザイン内容の可視性という観点から並び替えたい場合、右クリックし、表示されるオプションから [上位へ移動]、[下位へ移動] を利用することで、プリントブロックの表示順を変更することができます。

[Layout] エLEMENTの定義③

✓ プリントブロック: PBAttractions

The screenshot shows the GeneXus IDE interface. On the left, the 'AttractionList' window displays the layout editor for the 'PBAttractions' block. The layout includes a 'PBTitle' block with an image of a suitcase and the text '観光名所一覧', a 'PBColumnTitles' block with headers '番号', '名前', '国', and '写真', and the 'PBAttractions' block itself with fields 'AttractionName' and 'CountryName'. On the right, the '項目属性/変数を挿入' (Insert Item Property/Variable) dialog is open. It features a 'フィルタ' (Filter) section with 'パターン(P):' and 'オブジェクト(O):' fields. Below is a table titled '項目属性/変数コントロールを新規作成' (Create New Item Property/Variable Control).

| 名前 | タイプ | デスクリプション |
|--------------------------|------------------|--------------|
| Today | Date | Today |
| Time | Character(8) | Time |
| Pgmname | Character(128) | Pgmname |
| Pgmdesc | Character(256) | Pgmdesc |
| Page | Numeric(6,0) | Page |
| Line | Numeric(6,0) | Line |
| Output | Character(3) | Output |
| AirlineDiscountPercen... | Percentage | 航空会社割引率 |
| AirlineFlightMostExpe... | Id | 航空会社最高値フラ... |
| AirlineId | Id | 航空会社番号 |
| AirlineName | Name | 航空会社名 |
| AirportId | Id | 空港番号 |
| AirportName | Name | 空港名 |
| AttractionAddress | Address, GeneXus | 顧客住所 |
| AttractionCountry | VarChar(50) | 観光名所国名 |
| AttractionId | Id | 観光名所番号 |
| AttractionName | Name | 観光名所名 |

At the bottom of the dialog are buttons: '新規変数(N)...', '編集(E)...', 'OK(O)', and 'キャンセル'.

データベースに登録されたデータを出力したい場合、[ツールボックス] ウィンドウから「項目属性/変数」というコントロールを配置する必要があります。

このコントロールは、可変の値を表示するためのコントロールを配置するため、変数か、項目属性を配置することができます。

そのため、コントロールをドラッグすると、「項目属性/変数を挿入」というダイアログが表示され、このオブジェクト内で定義されている変数または、ナレッジベース内で定義されている項目属性を選択できます。

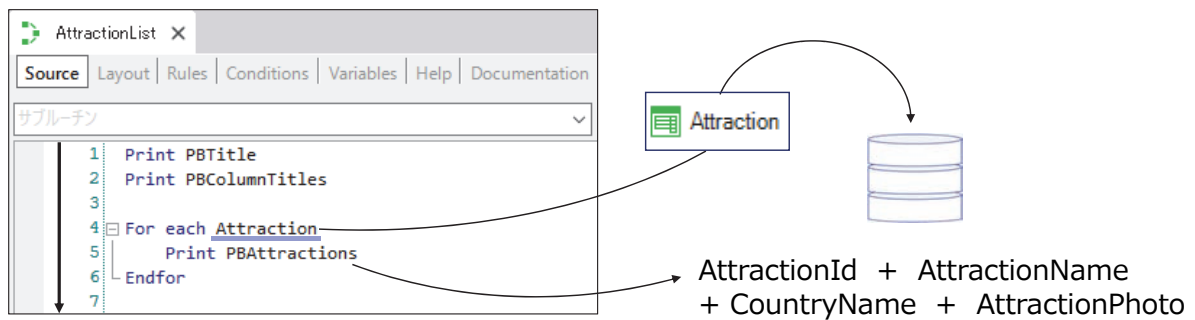
追加したいコントロールが、項目属性のみの場合、メニューバーの **[追加]** → **[項目属性]** を使用すると、「項目属性を挿入」というダイアログが表示され、ここには、項目属性のみが表示されます。

また、配置したコントロールの配置を整えたい場合、対象となる複数のコントロールを選択し、メニューバーの **[レイアウト]** → **[配置]** に含まれるオプションを利用できます。

[Source] エlementによる制御

✓ コマンド:

- **Print:** 任意のプリントブロックを出力
- **For each:** ベーステーブルとその拡張テーブルを決定し、各レコードごとの処理を記述

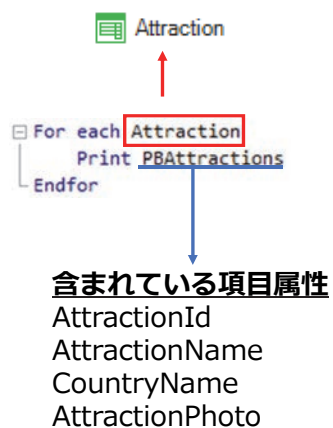


PDF を出力するため、[Source] Elementで制御を行うコードを記述します。
出力するために必要なコマンドは、Print コマンドです。
このコマンドは、続けて [Layout] Elementで定義したプリントブロック名を記述することで、デザイン内容を出力の一部とします。

そして、もし出力内容にデータベースの内容を読み込む必要がある場合、For each コマンドも必要となります。
後ほど構文について紹介しますが、このコマンドは、「For each」から始まり、「Endfor」の記述までに記載された内容を参照するテーブルの対象レコード数分繰り返し実行します。
参照するテーブルは、ベーステーブルと呼ばれ、For each コマンドの後にスペース区切りでトランザクション名を指定することで決定します。

ここまでに記載の通り、[Source] Elementは、手続き型となり、記述された内容が上から順に実行されます。

For each コマンドのベーストランザクションとベーステーブル



ベーステーブル ATTRACTION 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 | |

拡張テーブル

COUNTRYCITY
テーブル

| CountryId | CityId | CityName |
|-----------|--------|----------|
| 1 | 1 | リオデジャネイロ |
| 1 | 2 | サンパウロ |
| 2 | 1 | パリ |
| 3 | 1 | ... |

COUNTRY
テーブル

| CountryId | CountryName |
|-----------|-------------|
| 1 | ブラジル |
| 2 | フランス |
| 3 | 中国 |

For Each コマンドについてもう少し細かく説明を行います。

先ほど記載の通り、For each コマンドには、ベーステーブルを決定するため、トランザクション名を記述することができます。

このトランザクション名を記述する節を「ベーストランザクション節」と呼びます。

また、ここまでの章でベーステーブルについて説明を行いましたが、この For each コマンドでもこの概念が適用されます。

そのため、For each コマンド内の処理で、データを参照する場合は、拡張テーブルの範囲まで取得の対象とすることができます。

Procedure オブジェクトのプロパティおよびルール

The screenshot illustrates the configuration of the 'AttractionList' procedure object. On the left, the 'プロパティ' (Properties) window shows the 'Main program' property set to 'True'. In the center, a context menu is open over the '実行' (Execute) button, with '実行' selected. On the right, the 'AttractionList' window shows the 'Rules' tab with a rule 'Output_file("AttractionList", "PDF");'. Below this, the '観光名所一覧' (List of Tourist Attractions) table is displayed.

| 番号 | 名前 | 国 | 写真 |
|----|------------|------|----|
| 1 | ルーブル美術館 | フランス | |
| 2 | 万里の長城 | 中国 | |
| 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 | |

デザインとコードの定義が完了していますが、PDF を出力するための実装が不足しています。

PDF を出力するプロシージャオブジェクトとして定義するためには、次の 2 点を設定する必要があります。

1. オブジェクトのプロパティ

[Call protocol] プロパティを HTTP に変更します。

2. Output_file ルールの定義

[Rules] エlementで、Output_file ルールを定義します。

第 1 引数には、ファイル名、第 2 引数には、PDF という文字列を記述します。

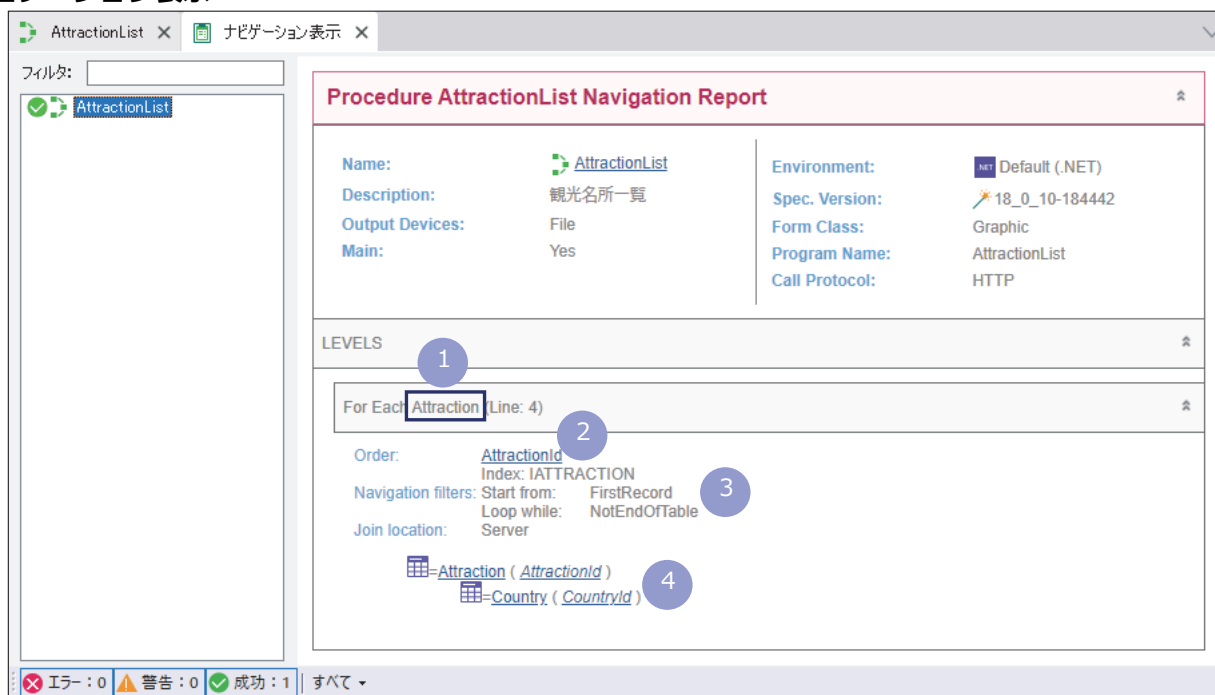
(第 1 引数のファイル名は、定義しますが、PDF 出力の場合、利用されません)

また、PDF 出力を行うプロシージャオブジェクトとして必須の設定ではありませんが、[Main program] プロパティを True に設定しておくことで、GeneXus の IDE から直接実行することができるようになります。

実行した場合、既定のブラウザ上で出力された PDF が表示されます。

この時に、GeneXus の IDE 上で表示される「ナビゲーション表示」の内容が重要となるため、次のページで解説を行います。

ナビゲーション表示



ナビゲーション表示では、プロシージャオブジェクト内で定義された For Each コマンドが対象とするテーブル（ベーステーブル）をはじめ、GeneXus が判断し、決定した内容を確認できます。
主に確認できる点を以下 4 点記載しました。

1. 対象テーブル（ベーステーブル）

前述の通り、ベーストランザクション節の定義に基づき、For each コマンドは、ベーステーブルを決定します。
ナビゲーション表示には、ベーストランザクション節に基づき、関連するテーブルを推論し、表示します。
そのため、記載された名前がトランザクションオブジェクトと一致している場合でも、これは物理テーブル名を示しています。

2. データ読み込みの並び替え基準

テーブルへアクセスする場合、データの並び替えを行うことが可能です。
指定がない場合、既定でキー項目を対象に昇順で並び替えを行います。

3. データのフィルタリング

参照するデータを絞り込むことが可能です。
指定がない場合、最初のレコードからテーブルの最後で無い限り参照を続けます。

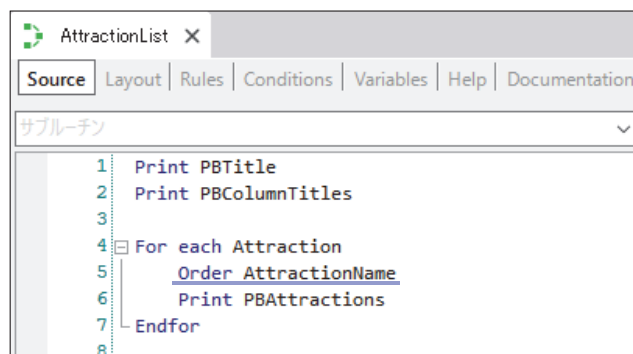
4. データ取得元テーブル

For each のベーステーブルを参照し、データを読み込みますが、拡張テーブルを参照する場合もあります。
ベーステーブル以外にアクセスを行っているテーブルについて確認することが可能です。

データの順序を変更する方法

✓ 要件:

旅行代理店が扱うすべての観光名所のリストを、**観光名所名順**で PDF ファイルに出力する



For each でナビゲートするテーブルの
拡張テーブルに含まれる項目属性で
並び替えできる

```
For each Attraction
  Order CountryName
  Print PBAttractions
Endfor
```

For each コマンドでは、Order 節を記述することで、取得するデータの並び替えも可能でした。

ベーストランザクション節に続き、スペース区切りで「Order」と入力し、さらにスペース区切りで、並び替えの基準となる項目属性名を記述します。

スライドのように改行でも問題ありません。

記述した項目属性に基づき、昇順で並び替えられるため、降順で並び替えたい場合、項目属性名を「()」で囲います。

また、並び替えの基準として利用可能な項目属性は、拡張テーブルに含まれる項目属性すべてを利用することが可能です。

データの順序を変更する方法：ナビゲーションレポート

AttractionList × ナビゲーション表示 ×

フィルタ:

AttractionList

Procedure AttractionList Navigation Report

Name: AttractionList
Description: 観光名所一覧
Output Devices: File
Main: Yes

Environment: .NET Default (.NET)
Spec. Version: 18_0_10-184442
Form Class: Graphic
Program Name: AttractionList
Call Protocol: HTTP

LEVELS

For Each Attraction (Line: 4)

Order: AttractionName
Index: UATTRACTION

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

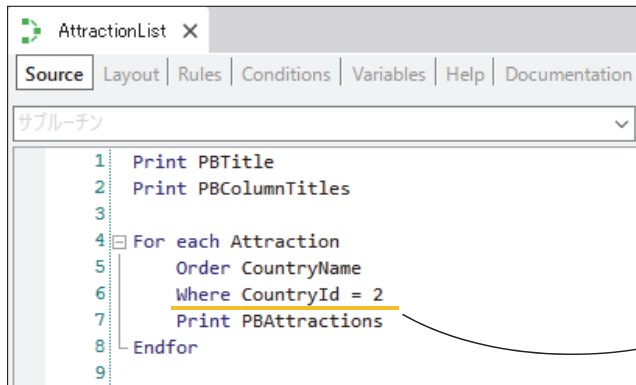
Attraction (AttractionId)
 Country (CountryId)

エラー: 0 警告: 0 成功: 1

Order 節を記述した状態で、ナビゲーション表示を確認すると、データ読み込みの並び替え基準が主キー項目から指定した項目属性に変更されていることを確認できます。
 [Source] エlementで記述した内容に基づく実装は、すべてナビゲーション表示で確認することが重要です。

フィルタを定義する方法

- ✓ 要件：国名順で表示し、フランス国内のすべての観光名所を一覧表示する



| 番号 | 名前 | 国 | 写真 |
|----|---------|------|----|
| 1 | ルーブル美術館 | フランス | |
| 3 | エッフェル塔 | フランス | |

Where CountryName = "フランス"

For each コマンドでは、Where 節を記述することで、取得するデータの絞り込みも可能でした。

Order 節に続き、スペース区切りで「Where」と入力し、さらにスペース区切りで、絞り込みのフィルタを記述します。

スライドのように改行でも問題ありません。

また、絞り込みのフィルタにおいて利用可能な項目属性は、拡張テーブルに含まれる項目属性すべてを利用することが可能です。

フィルタを定義する方法：ナビゲーションレポート

AttractionList × ナビゲーション表示 ×

フィルタ: AttractionList

Procedure AttractionList Navigation Report

Name: AttractionList
Description: 観光名所一覧
Output Devices: File
Main: Yes

Environment: Default (.NET)
Spec. Version: 18_0_10-184442
Form Class: Graphic
Program Name: AttractionList
Call Protocol: HTTP

Warnings

▲ spc0038 There is no index for order CountryName; poor performance may be noticed in group starting at line 4.

LEVELS

For Each Attraction (Line: 4)

Order: CountryName
No index

Navigation filters: Start from: CountryName = "フランス"
Loop while: CountryName = "フランス"

Join location: Server

Attraction (AttractionId)
Country (CountryId)

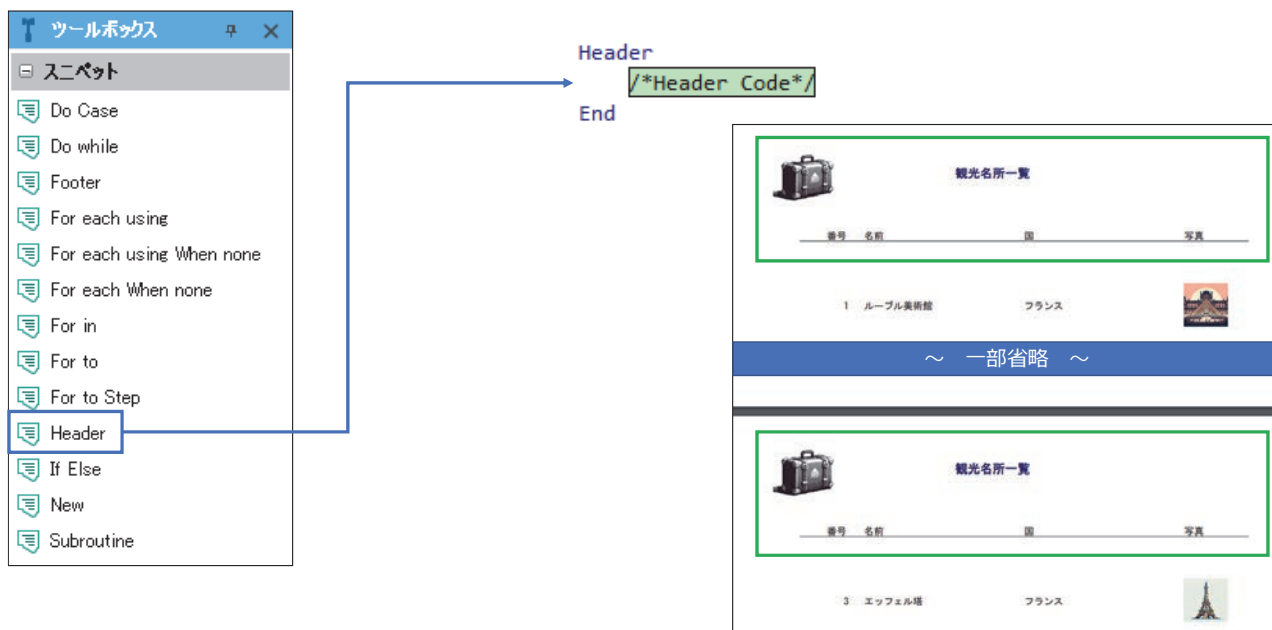
エラー: 0 警告: 1 成功: 0

Where 節を追加した結果、Navigation filters に表示される開始レコードおよび反復処理を実行する条件が変化したことを確認できます。

Where 節に指定した条件と一致するレコードだけを対象とするように変化し、テーブル全体を参照しない実装となるため、パフォーマンスの向上が見込めます。

このような絞り込みを実装する場合、Order 節で並び替えを行っている項目属性と、Where 節で絞り込みの条件として利用している項目属性が同一となる必要があります。そうでない場合、For each コマンドによるテーブル参照は、すべてのレコードを参照し、条件に一致するデータかどうか確認を行うような挙動になります。

スニペット



GeneXus の IDE は、状況に依存し、表示が切り替わる部分が多数あります。
 この章でも、フォーカスのあるコントロールやオブジェクトによって、[プロパティ] ウィンドウに表示される内容は異なるものが表示されました。
 [ツールボックス] ウィンドウも同様です。
 [Layout] エlementを表示している場合、デザインするために配置可能なコントロールが表示されました。
 [Source] エlementを表示している場合、このウィンドウには何が表示されるのでしょうか。

この場合、「スニペット」のリストが表示されます。
 スニペットに表示されるものは、[Source] エlementヘドラッグすることで、いくつかの節や構造を含み数行のコードとしてコマンドなどが展開されます。
 この展開される内容が判断できるような短い文章でスニペットは表示されています。

例えば、Header をドラッグした場合、Header コマンドが展開されます。
 このコマンドでは、Header から End までの間にプリントブロックを出力するように Print コマンドの記述を行うことで、出力する PDF に改ページが発生した場合も常に表示するプリントブロックの指定が可能です。

表示されるスニペットから展開されるコードについてこのコースでは一部のみを対象としますが、スニペットからの展開は行いません。
 対象外のコードについては、GeneXus Wiki 等をご参照ください。

For each 構文

For each <ベーストランザクション名>

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>

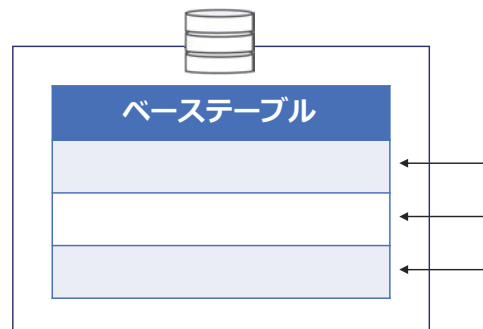
<メインのコード>

Endfor

この章で扱った For each コマンドは、他の章で構文の発展を行います。
そのため、この章で説明した内容をまとめます。

For each 構文 : ベーストランザクション節 (基本)

```
For each <ベーストランザクション名>  
  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
```



For each コマンドは、ベーストランザクション節を利用し、トランザクション名を指定することが出来ました。
指定したトランザクション名に基づき、GeneXus が関連するテーブルを推論し、参照対象のテーブルつまりベーステーブルを決定します。

このテーブルのレコード 1 件に対し、コマンド内に記載された処理が 1 回実行されます。

For each 構文 : ベーストランザクション節（レベルの指定）

- 第 1 レベルの指定 : <トランザクション名>

For each Flight

...

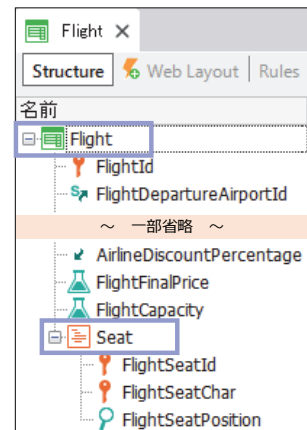
Endfor

- 第 2 レベル以下の指定 : <トランザクション名>.<レベル名>

For each Flight.Seat

...

Endfor



もし、ベーストランザクション節に指定したトランザクションが、第 2 レベルなど複数のレベルが定義されていた場合、ベーステーブルの決定はどのように行われるでしょうか。

はじめに、ベーストランザクション節に記述できる名前は、正確にはトランザクション名 + レベル名でした。

そのため、もし、トランザクション名だけを指定した場合、最も親となるレベル、つまり第 1 レベルに基づくテーブルがベーステーブルとして推論されます。

トランザクション名に続けて「.」（ピリオド）を入力した場合、レベル名を続けて指定することができます。

この場合、指定されたレベルに基づくテーブルがベーステーブルとして推論されます。

For each 構文: Order 節

For each <ベーストランザクション名>

Order <項目属性₁>, <項目属性₂>, …, <項目属性_n>

Where <条件₁>

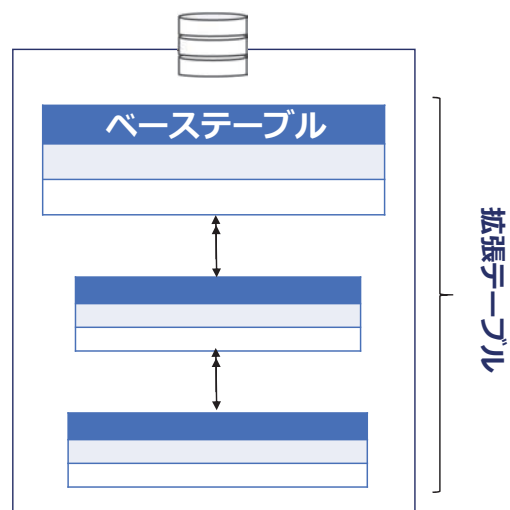
Where <条件₂>

...

Where <条件_n>

<メインのコード>

Endfor



Order 節を利用し、For each コマンドで参照するテーブルの並び順を指定することが出来ました。

Order 節には、For each コマンドのベーステーブルまたはその拡張テーブルに含まれる項目属性を指定できます。

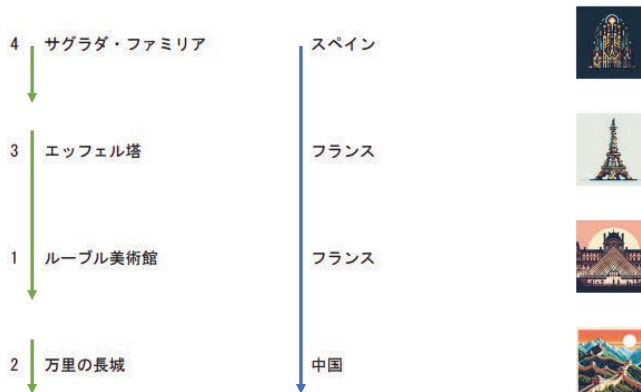
また、この For each 構文の通り、項目属性は、単一だけでなく、カンマ区切りで、複数の項目属性で順序を指定できます。

Order に複数の項目属性を指定した例

```
Print PBTitle
Print PBColumnTitles

For each Attraction
  Order CountryName, AttractionName
  Print PBAttractions
Endfor
```

| PBAttractions | | | |
|---------------|----------------|-------------|-----------------|
| Attr | AttractionName | CountryName | AttractionImage |
| | | | |
| | | | |
| | | | |
| | | | |



Order 節にカンマ区切りで複数の項目属性を指定した場合、左の項目属性から順に並び替えが行われます。

例えば、2 つの項目属性を対象にした場合、1 つ目の項目属性に同じ値が登録されたレコードが多数ある場合、2 つ目の項目属性は、この同じ値の中で並び替えを行う挙動を取ります。

For each 構文: Where 節

For each <ベーストランザクション名>

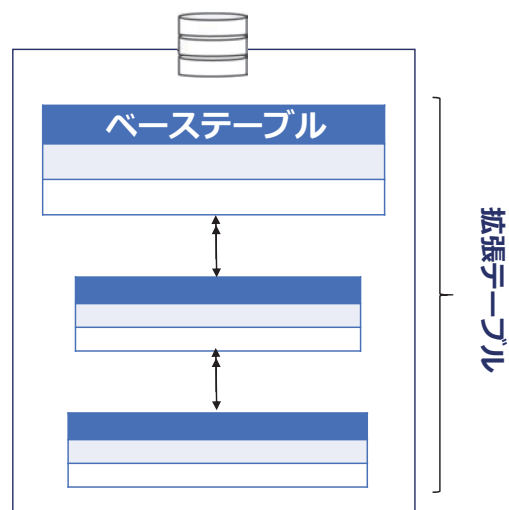
Order <項目属性₁>, <項目属性₂>, …, <項目属性_n>

Where <条件₁>

<条件_a> and <条件_b>
<条件_a> or <条件_b>

<メインのコード>

Endfor



Where 節を利用し、For each コマンドで参照するテーブルのフィルタを指定することが出来ました。

Where 節には、参照したいレコードの条件を指定します。

1 つの条件の中で、and や or で複数の条件を結合することができます。

条件a **and** 条件b: 両方の条件を同時に満たす必要があります。

条件a **or** 条件b: 一方の条件が満たされれば、該当するレコードとして選択されます。

For each 構文: Where 節

For each <ベーストランザクション名>

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>

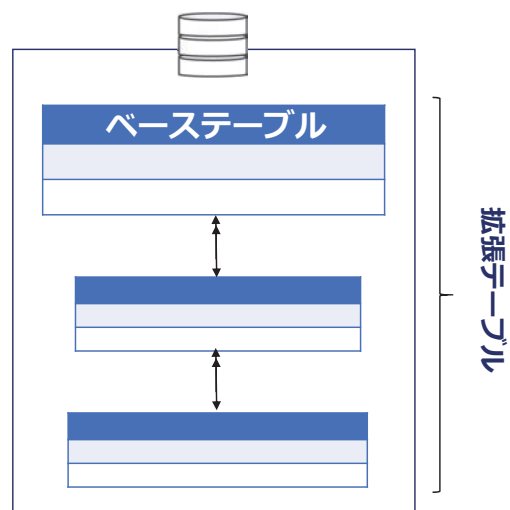
) and

) and

) and

<メインのコード>

Endfor



複数の **where** 節を記述することもできました。

この場合、1 つの Where 節で、複数の条件を **and** で結合した場合と同じ結果になります。

残念ながら複数の Where 節を記述する方法で、or 結合した場合と同じ結果にする方法はありません。

For each 構文: メインのコード

For each <ベーストランザクション名>

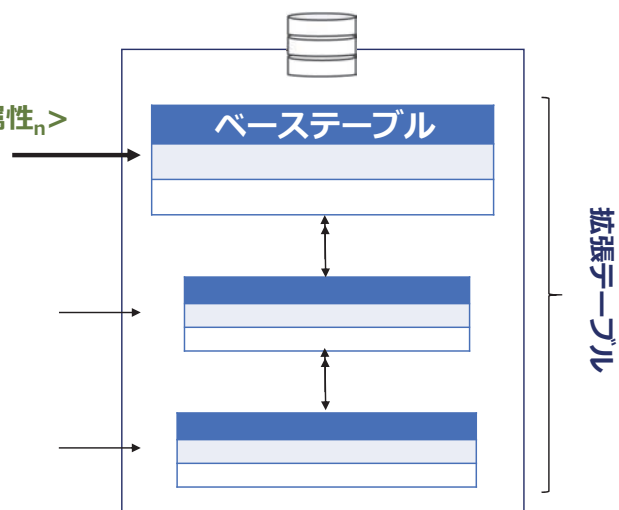
Order <項目属性₁>, <項目属性₂>, ..., <項目属性_n>Where <条件₁>Where <条件₂>

...

Where <条件_n>

<メインのコード>

Endfor



For each コマンド内のメインのコードには、必要な処理を実行したい順番に記述します。ベーステーブル内の該当レコード 1 行ずつを対象に処理を行う必要な操作を記述します。この時、GeneXus が必要と判断した場合、拡張テーブルのデータも参照します。

この章では、Print コマンドを利用し、プリントブロックの出力処理を記述しました。

*GeneXus*TM

training.genexus.com
wiki.genexus.com