

## 構造化データタイプのロード

GeneXus オブジェクト: データプロバイダー

**GeneXus™**

データベース上に格納された値とは別に、同様のデータタイプの**コレクション (要素のリスト)** としてメモリ上に格納する場合があります。



たとえば、旅行代理店で共通の特徴を持つ顧客のグループに対して操作を実行する必要がある場合や、観光名所のグループに関する特定のデータの情報を処理するよう求められる場合があります。このようなときに、リストを一時的にメモリにロードする場合があります。

SDTCustomer

Id: ID  
Name: 名前  
Address: 住所

**&Customer: SDTCustomer**

Id: 1  
Name: 'ジョン スミス'  
Address: '5番通り'

このような要件を満たすためには、メモリ上に**コレクション (要素のリスト)** として**格納可能な構造**を作成しなければなりません。

これまでの学習で、この例の SDTCustomer のような**構造化データタイプ**を使用すると、**あるアイテム**に対応した複数のデータを格納する構造を定義できることを見てきました。

この図は、**1人**の顧客番号、名前、および住所をメモリに格納している様子を表しています。

## SDTCustomer

Id: 1  
Name: ジョン スミス  
Address: 5番通り

← 1 人の顧客

## SDTCustomers

Id: 1  
Name: ジョン スミス  
Address: 5番通り

Id: 2  
Name: スーザン ブラウン  
Address: 7番通り

Id: 3  
Name: ロバート ヒル  
Address: 81番通り

Id: 4  
Name: ピーター ジョンソン  
Address: ポール通り

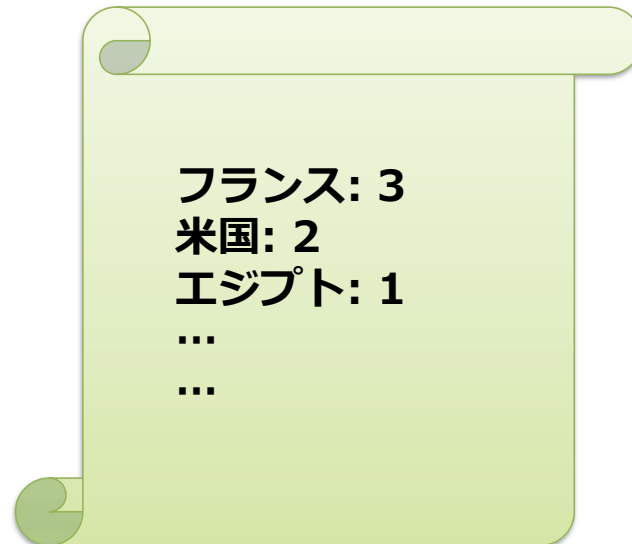
...

↑  
顧客のコレクション

顧客データが含まれている複数のアイテムを格納するには、構造化データタイプを定義し、それがコレクションであると指定する必要があります。

## 新しい要望: 国のランキング

観光名所の数に基づいた国のランキング:



ここで、旅行代理店の新しい要望を実装します。

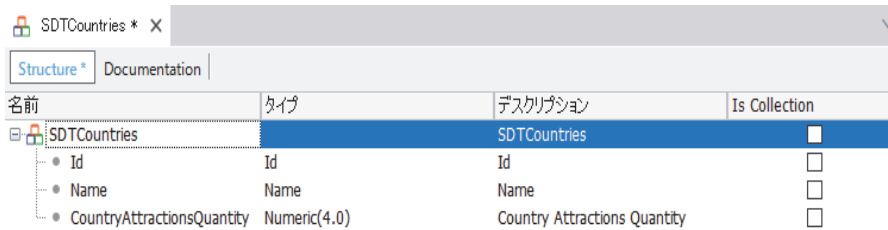
旅行代理店は、ツアーを実施している観光名所の数に基づいて、顧客に国のランキングを提示したいと考えています。

つまり、すべての国を、ツアーを実施している観光名所の数が多い順に並べ替えて表示する必要があります。

この要望を満たすためには、まず、データベースに格納されているすべての国を、それぞれの観光名所の数とともに 1 つの構造にロードします。次に、観光名所の数が多い順に並べ替えて、Web パネルまたは PDF 上にリストアップして表示します。

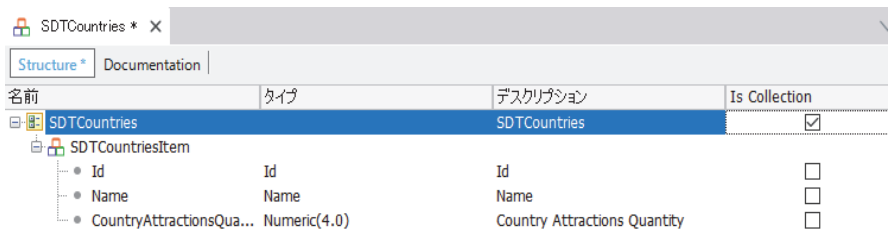
## 新しい要望: 国のランキング

### 1 つの国をメモリに保存する場合:



名前	タイプ	デスクリプション	Is Collection
SDTCountries		SDTCountries	<input type="checkbox"/>
• Id	Id	Id	<input type="checkbox"/>
• Name	Name	Name	<input type="checkbox"/>
• CountryAttractionsQuantity	Numeric(4.0)	Country Attractions Quantity	<input type="checkbox"/>

### 国のコレクションをメモリに保存する場合:



名前	タイプ	デスクリプション	Is Collection
SDTCountries		SDTCountries	<input checked="" type="checkbox"/>
• SDTCountriesItem			
• Id	Id	Id	<input type="checkbox"/>
• Name	Name	Name	<input type="checkbox"/>
• CountryAttractionsQua...	Numeric(4.0)	Country Attractions Quantity	<input type="checkbox"/>

それでは、構造化データタイプの新しいオブジェクトを作成しましょう。名前は SDTCountries にします。

各国の国番号、名前、登録されている観光名所の数が必要です。

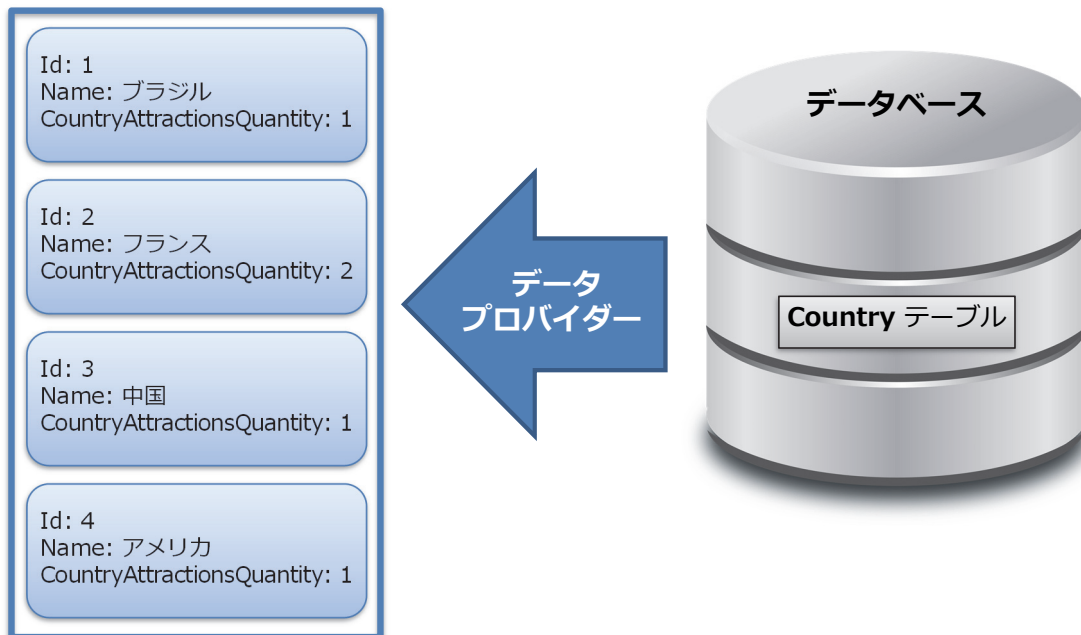
スライドの 1 つ目の定義では、単一の国に関連する情報をメモリに保存することを示しています。

しかし、国のランキングを作成するには、複数の国を格納する必要があります。そこで [Is Collection] チェックボックスをオンにして、複数の国のデータを格納できる構造に設定します (スライドの 2 つ目の定義)。

この構造には、1 つ目の定義と同じメンバーが含まれていますが、それらは、SDTCountriesItem というサブ構造にグループ化されています。このサブ構造は、[Is Collection] チェックボックスをオンにしたときに自動的に作成されたものです。

各アイテムには 1 つの国データが格納され、コレクションにはそれらの国データのグループが格納されます。

## コレクションの構造



コレクションのデータをロードするには、GeneXus の **Data Provider** オブジェクトを使用します。

Data Provider オブジェクトは、データ構造のロードを可能にします。たとえば、データベースのデータに基づき、Data Provider オブジェクトはロードされたデータを戻り値として返します。

## 新しい要望: 国のランキング

SDT を Data Provider の [Source] にドラッグする

```

SDTCountries
{
  SDTCountriesItem
  {
    Id = /*Id value*/
    Name = /*Name value*/
    CountryAttractionsQuantity = /*Country Attractions Quantity value*/
  }
}

```

プロパティ	
Data Provider: DPRankingCountriesWithAttractionsQty	
Name	DPRankingCountriesWithAttractionsQty
Description	DPRanking Countries With Attractions Qty
Expose as Web Service	False
Generate Open API interface	Use Environment property value
Main program	False
Call protocol	Internal
Module/Folder	Root Module
Qualified Name	DPRankingCountriesWithAttractionsQty
Object Visibility	Public
Output	
Infer Structure	No
Output	SDTCountries
Collection	False
Network	

GeneXus で Data Provider オブジェクトを作成して次の名前を付けます:  
DPRankingCountriesWithAttractionsQty。

Data Provider の [Source] エLEMENTが開きます。ここで、返されるコレクションに、データをどのようにロードするかを宣言します。ロードの宣言は非常に簡単です:  
KB エクスプローラーのウィンドウに移動し、構造化データタイプ SDTCountries を検索して、それを Data Provider の [Source] にドラッグします。

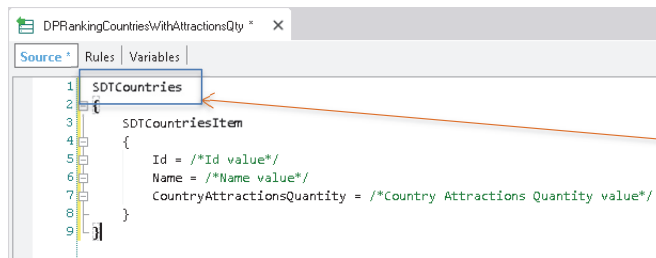
GeneXus によって数行のテキストが自動的に記述されます。

Data Provider のプロパティを開くと、GeneXus によって SDTCountries コレクションの名前が [Output] プロパティに割り当てられたことが分かります。つまり、**Data Provider は、構造化データタイプ SDTCountries のコレクション型のデータをロードして返します。**

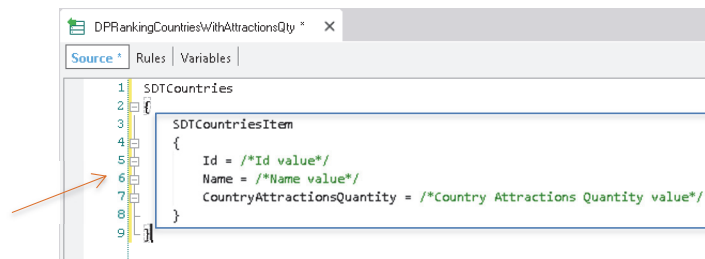
SDTCountries は既にコレクションなので、Data Provider の [Collection] プロパティの値を [True] に設定する必要はありません。この設定が必要になるのは、Data Provider の [Output] プロパティに単純な構造化データタイプを指定し、これに基づくコレクションを返す場合です。



## 新しい要望: 国のランキング



構造化データタイプ名

コレクションの  
項目のサブ構造

ここで、GeneXus によって [Source] に記述された内容を分析してみましょう。

SDTCountries という名前の構造化データタイプがコレクションであることが分かります。そして、コレクションの項目のサブ構造は、波括弧 { } の間にあります。

## 新しい要望: 国のランキング

The screenshot displays the GeneXus IDE interface with two tabs: 'DPRankingCountriesWithAttractionsQty \*' and 'SDTCountries \*'. The 'SDTCountries \*' tab is active, showing the 'Structure' view. Below the tabs, a table lists the structure of the data type.

名前	タイプ	デスクリプション	Is Collection
SDTCountries		SDTCountries	<input checked="" type="checkbox"/>
SDTCountriesItem			
Id	Id	Id	<input type="checkbox"/>
Name	Name	Name	<input type="checkbox"/>
CountryAttraction...	Numeric(4.0)	Country Attractions Quantity	<input type="checkbox"/>

Below the structure view, the 'Source' tab is active, showing the source code for the SDTCountries data type. The code is as follows:

```

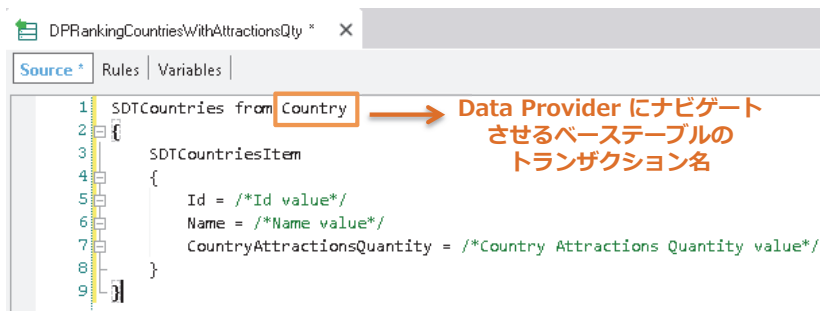
1 SDTCountries
2 {
3   SDTCountriesItem
4   {
5     Id = /*Id value*/
6     Name = /*Name value*/
7     CountryAttractionsQuantity = /*Country Attractions Quantity value*/
8   }
9 }

```

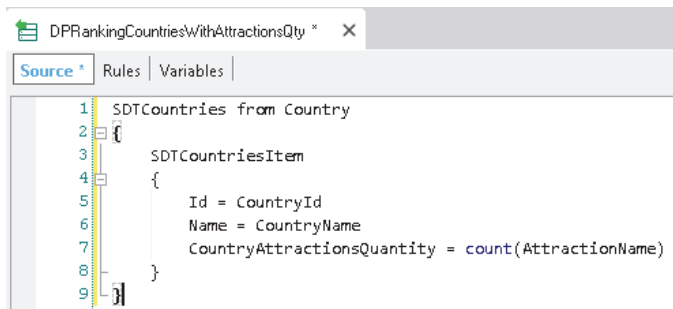
これを SDT の構造と比較します:

SDTCountries の構造がテキストとして表現されていて、Id、Name、および CountryAttractionsQuantity というメンバーは、これらの値をロードするための SDTCountriesItem サブ構造から提供されたことが分かります。

## 新しい要望: 国のランキング



## コレクションの各要素にロードする項目属性/計算を指定:



このコレクションを Country テーブルの内容に基づいてロードするため、そのテーブルをナビゲートする必要があることを Data Provider に示す必要があります。このためには From 節を使用し、その横にベーステーブルのトランザクション名を指定します。

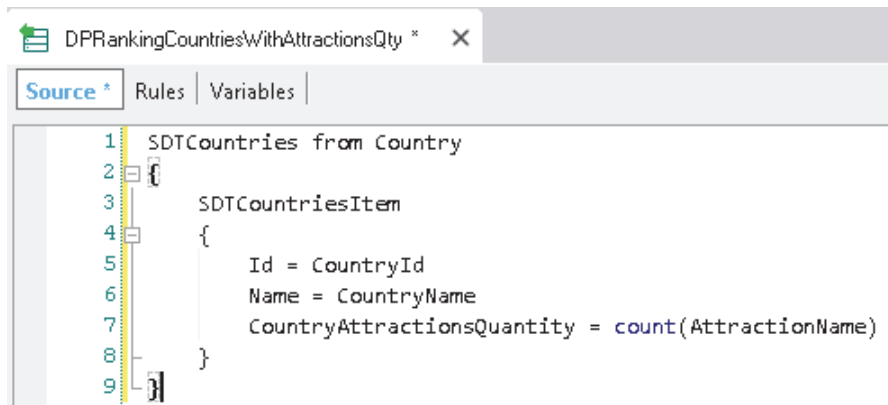
この例では、from Country です。

トランザクションにレベルが複数ある場合は、検索する特定のベーステーブルに関連付けられているレベルを指定するために、**トランザクションの名前、ピリオド、およびレベルの名前**を記述する必要があります。

次に、ID メンバーに CountryId 項目属性の値をロードします。また、Name メンバーに CountryName の値をロードし、CountryAttractionsQuantity メンバーに各国の観光名所の数を読み込むために、このメンバーに Count(AttractionName) インライン式の結果を割り当てるように指定します。

これまでに学んだ概念をおさらいしてみましょう。この定義済みのインライン式は、括弧内で指定された項目属性に基づいて、Attraction テーブルをナビゲートします。また、Data Provider と式で検索するテーブル間には共通の項目属性 (CountryId) があるため、この式は、Data Provider がナビゲートする**国の観光名所**を毎回カウントします。

## 新しい要望: 国のランキング



**Data Provider のベーステーブルは  
Country**

ここまでで行ったことは、**Data Provider がナビゲートすべきテーブルを宣言し、アクセスされたレコードごとに、国のコレクションに割り当てたい新しい値を指定すること**でした。

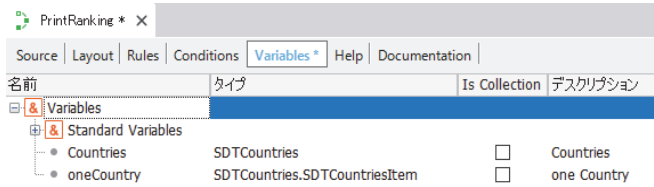
Data Provider が Country テーブルを処理することを、通常は、「**Data Provider のベーステーブルは Country である**」と言います:

最終的に、メモリ内のコレクションには、データベース内のすべての国のデータと、それぞれの国の観光名所の数が格納されます。

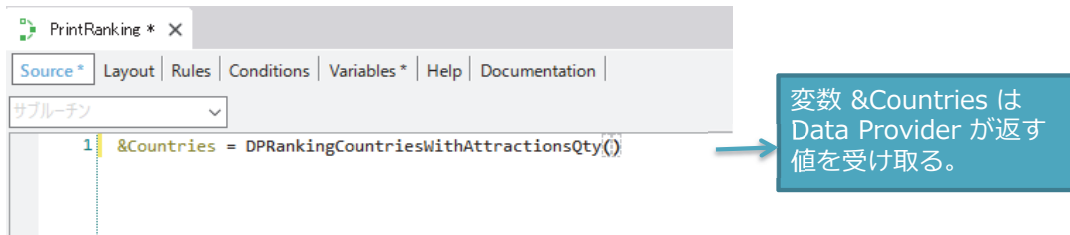
## 新しい要望: 国のランキング

1) Procedure オブジェクトを作成する

2) 変数を定義する:



3) プロシージャの [Source] で Data Provider を呼び出す:



それでは、国コレクションの中身を表示する Procedure オブジェクトを作成します。このプロシージャに「PrintRanking」という名前を付けます。

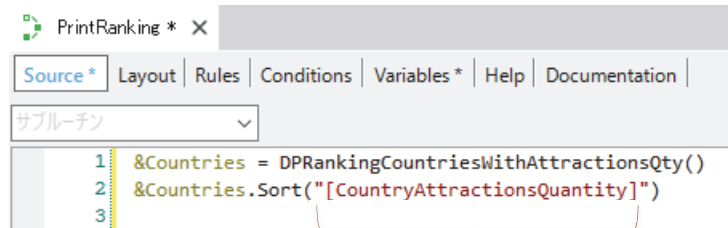
このプロシージャの [Variables] エレメントに移動して、SDTCountries データタイプの &Countries 変数を定義します。

次に、[Source] に移動して、このコレクション型の変数に値をロードするために、先ほど作成した Data Provider の戻り値をこれに割り当てます。

記述した指示に基づき、Data Provider の**呼び出し**が行われ、この Data Provider が、&Countries 変数にロードされる国のコレクションを返します。

## 新しい要望: 国のランキング

ランキングを実装するには、観光名所の数が多い順に並べ替える必要があり、**Sort** メソッドを使用する:



コレクションを並べ替えるための SDT のメンバー名

引用符の内側の「角括弧 []」により、逆の順序、つまり多いものから少ないものの順を指定します。

ここで、旅行代理店の要望を正確に思い出してみましょう: **記録されている観光名所の数が多いものから少ないものの順にすべての国を並べ替えて**ランキングを表示します。

したがって、残っている作業は、ロードしたコレクションの並べ替えです。つまり、国のコレクションの項目を表示する前に、記録されている観光名所の数が多いものから少ないものの順に並べ替えます。

これを行うために Sort メソッドがあります。この構文は次のとおりです:

```
&Countries.Sort("CountryAttractionsQuantity")
```

この場合、国のコレクションが観光名所の数の少ない国から多い国へと並べ替えられますが、ランキングを実装するには逆の順序で並び替える必要があります。

そこで、逆の順序を指定するために、引用符内に角括弧を追加します。

## 新しい要望: 国のランキング

メモリ上の格納されたコレクションの各要素を 1 つずつ処理し、出力するために、プリントブロックで For...in コマンドを記述する



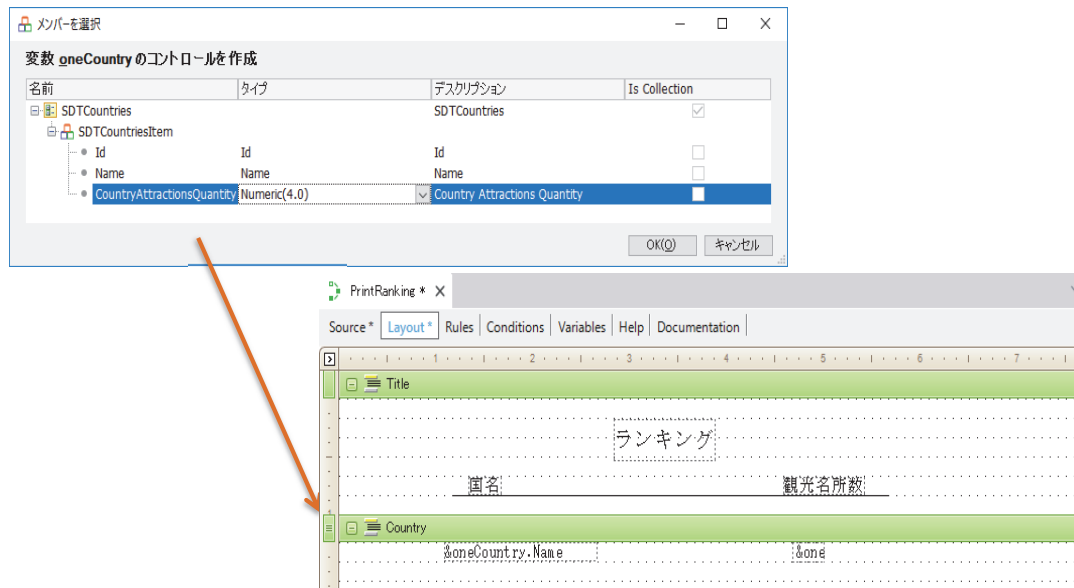
この時点で唯一残っている作業は、Data Provider で取得したコレクションを再度参照し、その内部の各項目と、格納されているデータを印刷することです。

メモリに格納されたコレクションを 1 要素ずつ参照するために、**For <エレメント> in <コレクション>** コマンドがあります。

&OneCountry 変数を定義し、コレクションを繰り返し参照し、各要素をこの変数にロードします。

## 新しい要望: 国のランキング

## Country プリントブロックに変数を追加する



[Layout] エlementに移動し、[Country] という名前のプリントブロックに変更します。[追加] -> [変数] を選択して &OneCountry を選択し、レイアウトに配置するメンバーを選択します。

【補足】 ここで、誤って &Countries を選択すると、実行時エラーが発生します。



## 新しい要望: 国のランキング

## ランキング

国名	観光名所数
フランス	2
ブラジル	1
中国	1
アメリカ合衆国	1

ここでは、必要なプロパティを定義し、PDF 形式でリストを出力します。

プロシーチャーのプロパティに移動し、[Main program] で [True] の値を選択します。

その後、[Main object properties] で [Call protocol] を選択し、[HTTP] を選択します。

最後に、[Rules] エlementに OutputFile ルールを挿入する必要があります:  
[追加] -> [ルール] を選択し、[Output\_file] ルールを選択します。

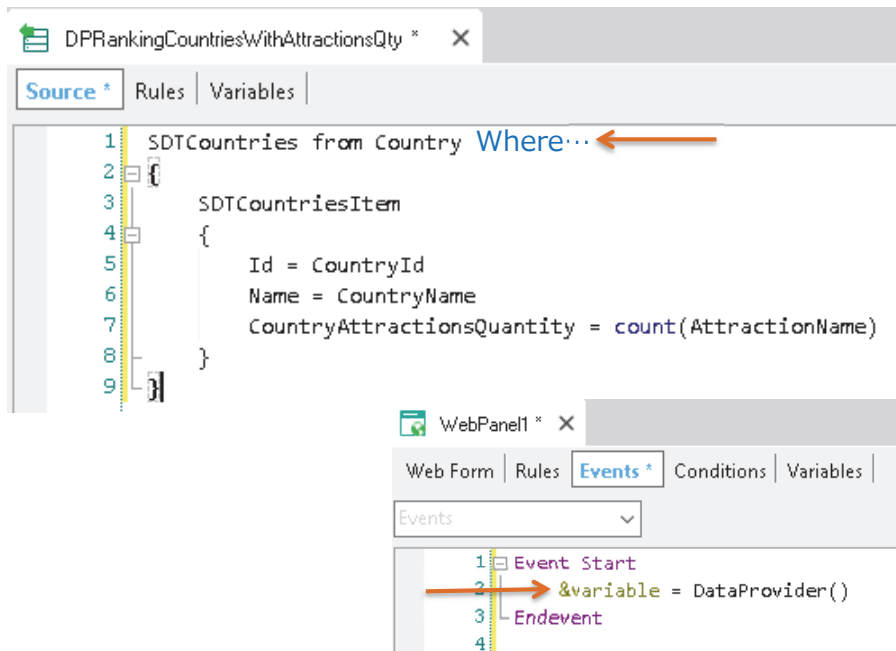
「Ranking.PDF」と名前を付けて終了します。形式は [PDF] を使用します。

これで、要望に対する開発が完了しました: プロシーチャーを実行し、ランキングを確認します。

PDF のリストには、データベースに格納されているすべての国が表示され、各国の観光名所の数と国のランキングも要望されたとおりに示されています。

このように、Data Provider は、データ構造内のデータをメモリにロードする強力な機能であり、コレクション型には特に有効です。また、ロードする内容を簡単に宣言できることも見てきました。実行するために必要なことは、すべて GeneXus が解決してくれます。

## 新しい要望: 国のランキング



【補足】Data Provider では、オプションで Where 節を指定して For each コマンドのようにフィルタを指定できます。

さらに進んだ章では、その他のデータプロバイダーの利用例を見ていきます。ほかのオブジェクト章で扱う WebPanel の [Events] において、データプロバイダーが呼び出されています。