

GeneXus™ 17

演習

2022 年 7 月

2021 年 10 月 22 日 改版

Copyright © GeneXus S.A. 1988-2022. All rights reserved.

本書は、GeneXus S.A. の明示的許可なしには如何なるメディアにも複写することはできません。本書の内容は個人的使用のみを目的として提供するものです。

登録商標

GeneXus は、GeneXus S.A. の登録商標です。本書において取り上げているその他の商標はすべて、それぞれの所有財産です。

目次

1. 課題.....	4
2. 新しいプロジェクト、新しいナレッジベース.....	5
3. 最初のトランザクション.....	6
3-1.「Customer」トランザクション.....	6
3-2.関連のある「Attraction」と「Country」トランザクション.....	9
3-4.「Category」トランザクション.....	14
3-5.「Country」トランザクションへの都市の追加.....	17
3-6.「Attraction」トランザクション: 都市の追加.....	19
4. トランザクションへの動作の追加 (ルール).....	22
5. パターン: データを操作するインターフェースの改善.....	24
6. 「Flight」および「Airport」トランザクションと、サブタイプを定義する必要性.....	30
7. グローバル式.....	32
8. インデックス.....	35
9. PDF レポート.....	36
10. パラメーターの受け渡し.....	44
11. ビジネスコンポーネント.....	48
11-1.キャンペーン期間中の「人気/おすすめ」カテゴリの設置.....	48
11-2.ビジネスコンポーネントによるフライトの一斉値上げ(オプション).....	50
11-3-1.ビジネスコンポーネントコレクションのデータプロバイダーによるカテゴリの初期データ展開 (オプション) ..	53
11-3-2.トランザクションの[Data Provider]プロパティによる提携先初期データの展開.....	55
12. レコードを更新するためのプロシージャオブジェクト.....	56
12-1.「人気/おすすめ」カテゴリの設置と観光名所の更新.....	56
12-2.検討課題 BC vs レコード操作のコマンド 不適切な値による更新.....	57
12-3.検討課題 全テーブルの情報削除 (オプション).....	58
13. Web パネル.....	60
13-1.項目属性選択のみによる(ベーステーブルあり)国の一覧画面.....	60
13-2.国別の観光名所数の表示.....	61
13-3.検討課題 プログラムによる(ベーステーブルなし)国一覧画面 (オプション).....	62
13-4.すべての国、すべての観光名所.....	63
13-5 WEB パネルのタイプ.....	64
14. デザインシステム.....	67
グリッドの Infinite scrolling ページング.....	71
Appendix1 基本スタイルとユーザーコントロール (オプション).....	72
Appendix2 Progressive Web Applications (PWA) (オプション).....	76
Appendix3 Query オブジェクト (オプション).....	78
Appendix4 Web サービス (オプション).....	80

Appendix5 スマートデバイスに必要なセクション (オプション).....	82
Appendix6 GXserver (オプション).....	83
Appendix7 観光名所を予約するためのビジネス プロセス モデル (BPM) の定義 (オプション)	84

1. 課題

旅行代理店から、代理店で使用する情報の格納と管理を行うソフトウェアシステムの開発を依頼されました。次の 2 つのモジュールが含まれるシステムをイメージしてください:

- **バックエンド:** 旅行代理店の社員が情報を処理するための、Web サーバー上で実行されるサブシステムです。
- **シンプルなモバイル デバイス アプリケーション:** 旅行代理店の顧客がダウンロードして使用するアプリケーションです。申し込み可能な旅行や、各都市の主な観光名所を検索できます。

この演習では、この課題に対するソリューションの一部のみを実装します。最初にしなければいけないことは何でしょうか。

2. 新しいプロジェクト、新しいナレッジベース

アプリケーションの開発を開始するために、GeneXus を起動して、「TravelAgency」という名前のナレッジベースを作成します。

プロトタイプ環境は「.NET Framework Environment」を選択します。

留意点:

- ナレッジベースの作成には、複数の方法があります:
 - [開始ページ] から作成。
 - メニューから作成: [ファイル] → [新規] → [ナレッジベース]
 - Ctrl + Shift + N
- ナレッジベースは、「ドキュメント」というフォルダや、「Documents and Settings」内のフォルダには作成しないでください。こうしたフォルダには、Windows から特別な許可が付与されているためです。
- 数分かけて、IDE (GeneXus の統合開発環境) に慣れてください。ウィンドウの移動、興味のあるウィンドウの表示 ([表示] および [表示] → [その他のツールウィンドウ])、[KB エクスプローラー] ウィンドウ内の [Root Module 内] の確認などを行ってください。ドメイン、一部のオブジェクト、画像などが既に初期化されていることが分かります。

3. 最初のトランザクション

旅行代理店との打ち合わせで、次のような情報が得られました:

「私たちの業務は、すべての顧客データを記録し、世界のさまざまな都市への旅行を提供することです。
各都市の観光名所の登録も行っています」

アプリケーションを構築するためには、現実からエンティティを識別することから開始しなければなりません。そして、それを **トランザクションオブジェクト** で表します。どのようなトランザクションをナレッジベース (KB) に作成する必要があるのでしょうか。

3-1. 「Customer」トランザクション

旅行代理店の社員に、顧客のどのようなデータを記録しているかを尋ねました。

次のような回答だったとします:

名字と名前 (それぞれ最大 20 文字)、**住所**、**電話番号**、**E メールアドレス**

この情報を基に「Customer」トランザクションを作成します。



留意点:

- オブジェクトを作成する方法は複数あります:
 - [開始ページ] から作成
 - メニューから作成: [ファイル] → [新規] → [オブジェクト]
 - Ctrl+N で作成
- 各顧客を識別するための項目属性が必要になります (CustomerId)。
- 新しい項目属性を追加するときにピリオド (.) を入力すると、トランザクションの名前が自動で挿入されます。
- Address**、**Phone**、および **Email** は セマンティックドメイン であり、名前に Address、Phone、または Email という単語が定義されている項目属性のデータタイプにそれぞれ自動的に割り当てられます。

3-1-1. トランザクションの定義

トランザクションは次のように定義します。

名前 : Customer デスクリプション : 顧客

名前	タイプ	デスクリプション
 CustomerId	Numeric(4.0)	顧客番号
 CustomerName	VarChar(20)	顧客名前
▪ CustomerLastName	VarChar(20)	顧客名字
▪ CustomerAddress	Address ,GeneXus	顧客住所
▪ CustomerPhone	Phone ,GeneXus	顧客電話番号
▪ CustomerEmail	Email ,GeneXus	顧客メールアドレス

3-1-2. 実行結果の確認

アプリケーションをテストするために実行します。GeneXus の **[出力]** ウィンドウが有効であり、表示されていることを確認します ([表示] → [その他のツールウィンドウ] → [出力])。

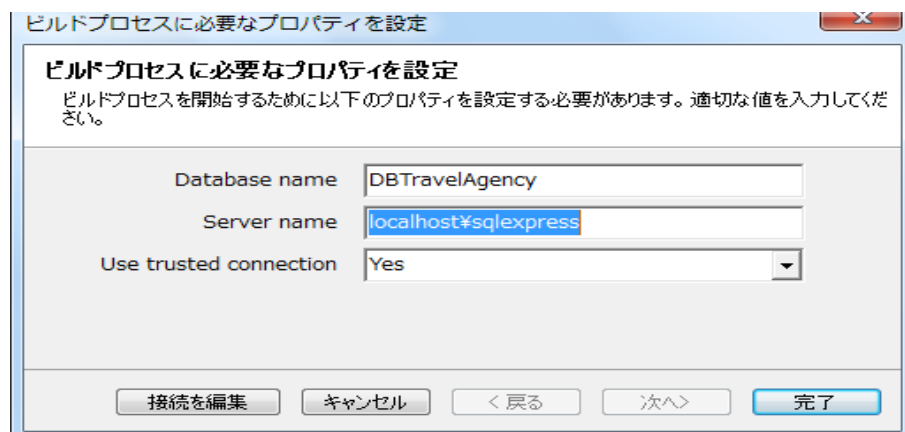
F5 キーを押し、実行環境でアプリケーションを**実行**します。

初めて実行する際に、以下の手順が必要となります。

データベースとプログラムをローカルに作成するため、次のようなウィンドウが開き、データベース、サーバー、および接続方法の詳細を入力します。指定した名前のデータベースがそのサーバーに存在しない場合は、そのデータベースが GeneXus により自動的に**作成**されます。

本課題では以下のように入力します。

Database name : DBTravelAgency
Server name : localhost%sqlexpress
Use trusted connection : Yes



次に、[影響分析 (Impact Analysis)] が表示されます。これから実行するデータベースの作成と、そのデータベースに含まれる CUSTOMER テーブルに関する詳細情報を示しています。

データベーステーブルが作成されます。
このレポートはデータベーステーブルがどのように作成されるか説明しています。
「作成」を押して続けるか、「キャンセル」を押してキャンセルしてください。

作成 キャンセル

フィルタ:

Customer

Table Customer specification

Table name: Customer

Customer is new

Table Structure

Attribute	Definition	Previous values	Takes value from
<u>CustomerId</u>	Numeric (4) Not null		
<u>CustomerName</u>	Varchar (20) Not null NLS		
<u>CustomerLastName</u>	Varchar (20) Not null NLS		
<u>CustomerAddress</u>	Varchar (1024) Not null NLS		
<u>CustomerPhone</u>	Character (20) Not null NLS		
<u>CustomerEmail</u>	Varchar (100) Not null NLS		

Indexes

Name	Definition	Composition
ICUSTOMER	primary key Clustered	<u>CustomerId</u>

Statements

```
CREATE TABLE [Customer] (  
    [CustomerId]          SMALLINT          NOT NULL,  
    [CustomerName]        NVARCHAR(20)      NOT NULL,  
    [CustomerLastName]    NVARCHAR(20)      NOT NULL,  
    [CustomerAddress]     NVARCHAR(1024)    NOT NULL,  
    [CustomerPhone]       NCHAR(20)         NOT NULL,  
    [CustomerEmail]       NVARCHAR(100)     NOT NULL,  
    PRIMARY KEY ( [CustomerId] ))
```

[作成] をクリックすると、GeneXus によりこれらのアクションを実行するプログラムが実行されます。

このプロセスの最後に、定義したオブジェクトを実行するためのメニューが、OS 既定のブラウザに表示されます。このケースの場合、Customer トランザクションが 1 つだけ表示されます。

数人の顧客をシステムに入力します。その後、入力したいいずれかの顧客のデータを変更し、1 人の顧客を削除します。

さらに、顧客を「選択リスト」から選択してみてください ([選択] ボタンを利用します)。

また、画面上の [>] ボタン を利用して、次の顧客レコードに移動できます。[選択] をクリックすると、選択画面が表示されます。ここから登録した顧客のリストを表示して、いずれかを選択することができます。

引き続き、エンティティの識別を行い、ほかのトランザクションを作成します。次の発言を思い出してください:

「私たちはすべての顧客に関するデータを記録し、世界のさまざまな国の都市への旅行を提供しています。
各都市の観光名所の登録も行っています」

3-2.関連のある「Attraction」と「Country」トランザクション

観光名所を記録するトランザクションと、観光名所が存在する国を記録する新規のトランザクションを作成します。各観光名所について、旅行代理店は次のデータを持っています。

観光名所の名前、国、画像、カテゴリ

この情報を基にトランザクションの作成を行います。まず、国情報を管理する「Country」から始めましょう。

留意点:

- トランザクション構造で項目属性の名前を指定するときにピリオド (.) を押すと、トランザクション名で初期化されます。
- Country にも、主キー項目として項目属性「CountryId」が必要です。

3-2-1. ドメインの定義

識別子の項目属性のデータタイプを定義する場合、データタイプ Numeric(4.0) を直接指定する代わりに、**Id ドメイン**を定義することにします。この Id ドメインでは AutoNumber を有効にして、このドメインに基づくすべての項目属性は、ユーザーの操作なしで自動採番されるようにします。

- 以下のドメインを「ドメイン」ウィンドウにて定義します：

名前	タイプ	デスクリプション
Id	Numeric(4.0)	Id

- 定義した Id ドメインを選択し、F4 キーで[プロパティ] ウィンドウを表示し、[Type Definition] プロパティ群配下の [Autonumber] プロパティを [True] に設定します。

補足:



- ドメインの定義には、複数の方法があります：
 - ◇ [ドメイン] ウィンドウから作成：[表示]→[ドメイン]をクリック後、[KB エクスプローラー] → [Root Module] → [ドメイン]にて、操作できます。
 - ◇ 項目属性の [Type] で直接定義(インライン定義)
 - ◇ [ファイル]→[新規]→[オブジェクト]→[共通]→[Domain]

注： GeneXus に事前定義されているすべてのドメインに注目してください。特に、「Customer」トランザクションに CustomerAddress、CustomerEmail、および CustomerPhone 項目属性を作成したときに自動的に表示される、**Address**、**Email**、および **Phone** に注目します。




3-2-2. トランザクションの定義

トランザクションは次のように定義します。

名前 : Country デスクリプション : 国

名前	タイプ	デスクリプション
 CountryId	Id	国番号
 CountryName	Name = VarChar(50)	国名

期待される結果 :

名前	タイプ
 Country	Country
 CountryId	Id
 CountryName	Name

ドメインを **CountryName** 項目属性で直接定義(インライン定義)します。データタイプに「Name = VarChar(50)」と記述し、フォーカスを移動するとタイプ列には「Name」の記述のみが残ります。

[ドメイン] ウィンドウへ次の定義が追加されていることを確認できます。

名前	タイプ	デスクリプション
Name	VarChar(50)	Name




続いて、「Attraction」トランザクションを作成しましょう。ここでは、観光名所番号、名前、ふりがな、国のみ入力します。

補足: AttractionId と入力すると、Id ドメインが自動的に特定されます。同様に、AttractionName 項目属性を入力すると、Name ドメインが自動的に特定されます。

3-2-3. トランザクションの定義






トランザクションは次のように定義します。

名前 : Attraction デスクリプション : 観光名所

名前	タイプ	デスクリプション
 AttractionId	Id	観光名所番号
 AttractionName	Name	観光名所名
AttractionFurigana	Name	観光名所ふりがな
 CountryId	Id	国番号
 CountryName	Name	国名

検討してみましょう：

- ・ CountryId に加えて、CountryName 項目属性も「Attraction」に追加したのはなぜでしょうか。

名前	タイプ	デスクリプション
 Attraction	Attraction	観光名所
 AttractionId	Id	観光名所番号
 AttractionName	Name	観光名所名
AttractionFurigana	Name	観光名所ふりがな
 CountryId	Id	国番号
 CountryName	Name	国名

名前	タイプ	デスクリプション
 Country	Country	国
 CountryId	Id	国番号
 CountryName	Name	国名

- ・ CountryId と CountryName の追加時にタイプとデスクリプションが入力済なのはなぜでしょうか。
⇒IDE の「表示」⇒「項目属性リスト」で、KB で登録済の項目属性が参照することができます。

テストを実行 (F5) すると、次の影響分析レポートが表示されます:

データベースの再編成が必要です。

このレポートにはデータベースの変更内容と、再編成プログラムでどのように扱われるかが記載されています。「再編成」を押して続けるか、「キャンセル」を押してキャンセルしてください。

再編成

キャンセル

フィルタ:

Attraction

Country

CountryName?

Table Attraction specification

Table name: [Attraction](#)

Attraction is new

Table Structure

Attribute	Definition	Pr
AttractionId	Numeric (4) Not null Autonumber	Pr
AttractionName	Varchar (50) Not null NLS	
AttractionFurigana	Varchar (50) Not null NLS	
CountryId	Numeric (4) Not null	

Indexes

Name	Definition
IATTRACTION	primary key Clustered
IATTRACTION1	duplicate

Foreign key constraints

Referenced table

[Country](#)

Statements

```
CREATE TABLE [Attraction] (  
    [AttractionId] SMALLINT NOT NULL IDENTITY
```

検討してみましょう：

GeneXus はデータベース内に Attraction テーブルを作成することを示していますが、Attraction テーブルに CountryName 項目属性が含まれていないのはなぜでしょうか。言い換えると、トランザクション構造の一部であるにもかかわらず、物理テーブルに含まれていないものがあるのはなぜでしょうか。

影響分析 (Impact Analysis)] を確認し、問題がない場合は、[再編成] をクリックします。

ブラウザが開き、各トランザクション (「Customer」、「Country」、「Attraction」) に対応する 3 つのプログラムへのリンクが含まれているメニューが表示されます。

国に次のデータを入力します: ブラジル、フランス、中国。

注: 国番号を「0」にすると、割り当て済みの最後の番号の次の番号が保存時に自動的に設定されます (自動採番)。

観光名所にも入力します (例: フランスにある「ルーブル美術館」)。

システム内の国番号を忘れた場合は、どのような国の入力方法があるでしょうか。国の「選択リスト」を使用できます。このリストは GeneXus により自動的に作成され、CountryId の横にあるアイコンで呼び出します。アイコンが表示されているのは、このトランザクションで CountryId が外部キー、つまり、ほかのテーブルを参照する役割を持っているためです。

3-3. 関連データ: 整合性の維持

「Attraction」と「Country」は関連付けられています。CountryIdをAttractionの構造の一部にすると、Country トランザクションの主キーの項目属性とまったく同じ名前であるため、GeneXus は、「Attraction」の CountryId を外部キーと認識し、データの整合性を自動的に維持します。次の操作を行い、確認してみましょう:

- 存在しない国の国番号 (例: 999) を持つ観光名所を入力してみてください。その観光名所は保存できましたか。
- 既に入力されている観光名所を開き (例: ルーブル美術館)、その国を、存在しない国に変更してください。変更内容は保存できましたか。
- 観光名所がすでに関連付けられている国 (例: フランス) を、Country トランザクションを使用して削除してみてください。削除できましたか。

結論: トランザクションに対応したプログラムにより、データの整合性が維持されます。

3-4. 「Category」 トランザクション



「Attraction」 トランザクションは、まだ完成していません。旅行代理店の社員は、それぞれの観光名所に対して**カテゴリ** (遺跡、美術館、公園など) を記録していると話していました。したがって、この情報を記録するための トランザクションの作成 と、「Attraction」 トランザクションへの カテゴリの追加 が必要です。

ただし、観光名所のカテゴリの記録は必須ではないとも言っていました。つまり、空白のままでもかまわないということです。GeneXus では参照整合性が自動的にチェックされることを学習しましたが、これはどう実現すればよいでしょうか。

3-4-1. トランザクションの定義

トランザクションは次のように定義します。








名前 : Category デスクリプション : カテゴリ

名前	タイプ	デスクリプション
 CategoryId	Id	カテゴリ番号
 CategoryName	Name	カテゴリ名

3-4-2. 「Attraction」 トランザクションへカテゴリと不足項目の追加

以下のように CategoryId、CategoryName を追加します。CategoryId は「Null 許容」列の値を「Yes」に変更します。

また、写真を管理する項目として AttractionPhoto を追加します。講師から配布された画像を使用し、AttractionPhoto に観光名所画像を登録してみましょう。

名前	タイプ	デスクリプション
 AttractionId	Id	観光名所番号
 AttractionName	Name	観光名所名
AttractionFurigana	Name	観光名所ふりがな
 CountryId	Id	国番号
 CountryName	Name	国名
 CategoryId	Id	カテゴリ番号
 CategoryName	Name	カテゴリ名
 AttractionPhoto	Image	観光名所写真

期待する結果:

名前	タイプ	DESCRIPTION	式	Null 許容
Attraction	Attraction	観光名所		
AttractionId	Id	観光名所番号		No
AttractionName	Name	観光名所名		No
AttractionFurigana	Name	観光名所ふりがな		No
CountryId	Id	国番号		No
CountryName	Name	国名		
CategoryId	Id	カテゴリ番号		Yes
CategoryName	Name	カテゴリ名		
AttractionPhoto	Image	観光名所画像		No

名前	タイプ
Category	Category
CategoryId	Id
CategoryName	Name

注意：上図の通り観光名所（カテゴリではありません）CategoryId の[Null 許容]が Yes になっていることを確認してください。

GeneXus にアプリケーションのビルドを指示し、実行環境でテストします。(F5)

データベースの再編成が必要です。

このレポートにはデータベースの変更内容と、再編成プログラムでどのように扱われるかが記載されています。「再編成」を押して続けるか、「キャンセル」を押してキャンセルしてください。

再編成 キャンセル

フィルタ:

- Category
- Attraction

Table Attraction specification

Table name: Attraction

Attraction needs conversion

Warnings

▲rqz0007Attribute AttractionPhoto does not allow nulls and does not have an Initial Value. An em

Table Structure

Attribute	Definition
AttractionId	Numeric (4) Not null Autonumber
AttractionName	Varchar (50) Not null NLS
AttractionFurigana	Varchar (50) Not null NLS
CountryId	Numeric (4) Not null
New CategoryId	Numeric (4)
New AttractionPhoto	Image Not null
New AttractionPhoto_GXI	Varchar (2048) Not null

Indexes

Name	Definition
IATTRACTION	primary key Clustered
IATTRACTION1	duplicate
New IATTRACTION2	duplicate

Foreign key constraints

Referenced table

注: Category テーブルが作成される必要があり、既存の Attraction テーブルはコンバートされる必要があります。合計 3 つ (CategoryId 用に 1 つと、AttractionPhoto 用に 2 つの) 項目が追加されます。1 つの Image タイプの項目属性には、文字列情報とバイナリのデータを格納するための 2 つのフィールドが必要になります。

再編成して実行します。

カテゴリを登録し (美術館や遺跡など)、既に入力されている観光名所を参照してデータ (カテゴリと写真) を完成させてください。

各観光名所には、カテゴリは空白のままにできることに注意してください (トランザクション構造で **[Nullable]** プロパティを **[Yes]** に設定したため)。一方で、観光名所の CategoryId として使われていない値 (例:999) を指定しても保存されません。

観光名所

<< < > >> 選択

観光名所番号

1

観光名所名

ルーブル美術館

観光名所ふりがな

るーぶるびじゅつかん

国番号

2

↑

国名

カテゴリ番号

0

↑

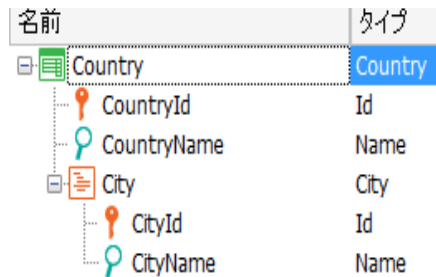
カテゴリ名

観光名所画像

更新

3-5. 「Country」 トランザクションへの都市の追加

旅行代理店が扱っている旅行先の国に加えて、それらの国の都市に関するデータも記録する必要があります。したがって、「Country」 トランザクションに第 2 レベルを追加して、都市の名前と都市番号を指定します。



留意点:

- CountryName 項目属性で右クリックし、[レベルを挿入] を選択すると、レベルを追加できます。
- 新しいレベルに名前を指定した後、ピリオドの代わりにダブルクォート「"」を入力すると、定義される項目属性がその新しいレベルの名前で初期化されます。
- 都市の識別子は、国番号に都市番号を組み合わせたものになります。つまり、国番号と都市番号の両方を指定して、ある都市を一意に特定させます。これにより、例えば、ウルグアイとアルゼンチンの両方にあるロサリオという都市に、それぞれ都市番号 1 を割り当てることが可能になります:
 - 国: 1 (ウルグアイ) – 都市: 1 (ロサリオ)
 - 国: 2 (アルゼンチン) – 都市: 1 (ロサリオ)
- アルゼンチンのロサリオに、別の番号を指定することもできます。
 - 国: 2 (アルゼンチン) – 都市: 4 (ロサリオ)

3-5-1. 「Country」 トランザクションの変更

以下のように「City」レベルとレベル配下の項目 CityId、CityName を追加します。

名前	タイプ	デスク립ション
 CountryId	Id	国番号
 CountryName	Name	国名
 City	City	都市
 CityId	Id	都市番号
 CityName	Name	都市名

実行し、結果を確認します。F5 キーを押します。

開始ページ × Country × 影響分析 ×

データベースの再編成が必要です。

このレポートにはデータベースの変更内容と、再編成プログラムでどのように扱われるかが記載されています。
「再編成」を押して続けるか、「キャンセル」を押してキャンセルしてください。

再編成 キャンセル

フィルタ: CountryCity

Table CountryCity specification

Table name: [CountryCity](#)

CountryCity is new

Warnings

! r9z0009AutoNumber=True ignored. Attribute [CityId](#) is not table CountryCity's primary key.

Table Structure

Attribute	Definition	Previous values	Takes value fr
CountryId	Numeric (4) Not null		
CityId	Numeric (4) Not null		
CityName	Varchar (50) Not null NLS		

Indexes

Name	Definition	Composition
ICOUNTRYCITY	primary key Clustered	CountryId CityId

Foreign key constraints

Referenced table	Attributes
Country	CountryId

注: 影響分析レポートで次のことが通知されます:

- **CityId** では [Autonumber] プロパティは無視されます。つまり、実行時に、ユーザーは都市番号を手動で入力する必要があります。 [Autonumber] プロパティの仕様では自動採番されるのは単純な主キーのみです。この CityId は複合キーの 2 つ目の項目であるため、自動採番の対象ではありません。
- 都市に対応する情報を格納するために、新しく **CountryCity** テーブルが作成されます。








既に登録した国の都市を入力します。

3-6. 「Attraction」 トランザクション: 都市の追加

「Attraction」 トランザクションで、観光名所が属する国の都市を追加しましょう。ただし、タイミングによっては、この観光名所の位置する都市が不明である場合や、特定の観光名所に関係ない場合もあることを旅行代理店から伝えられています。その場合、どのように トランザクションをデザインする必要があるでしょうか。

3-6-1. 「Attraction」 トランザクションの変更

以下のように CityId、 CityName を追加します。CityId は「Null 許容」列の値を「Yes」に変更します。

名前	タイプ	DESCRIPTION
 AttractionId	Id	観光名所番号
 AttractionName	Name	観光名所名
AttractionFurigana	Name	観光名所ふりがな
 CountryId	Id	国番号
 CountryName	Name	国名
 CityId	Id	都市番号
 CityName	Name	都市名
 CategoryId	Id	カテゴリ番号
 CategoryName	Name	カテゴリ名
 AttractionPhoto	Image	観光名所写真

実行し、結果を確認します。F5 キーを押します。

[影響分析(Impact Analysis)] が表示されるので [再編成] をクリックします。

期待する結果:

名前	タイプ
Attraction	Attraction
AttractionId	Id
AttractionName	Name
AttractionFurigana	Name
CountryId	Id
CountryName	Name
CityId	Id
CityName	Name
CategoryId	Id
CategoryName	Name
AttractionPhoto	Image

名前	タイプ
Country	Country
CountryId	Id
CountryName	Name
City	City
CityId	Id
CityName	Name

注意：上図の通り観光名所（カテゴリではありません）の CityID の[Null 許容]が Yes になっていることを確認してください。

3-6-2. データタイプの変更

先に進む前に、「Customer」トランザクションを開いて **CustomerId** のデータタイプを変更し、**Id** ドメインを使用するようにします（自動採番が設定されることを確認します）。

また、**CustomerName** と **CustomerLastName** のデータタイプも変更して、**Name** ドメインを利用するようにします。

再編成を実行し、データタイプが変更されたことを確認します。

4. トランザクションへの動作の追加 (ルール)

開発中のアプリケーションと一緒にテストした後、旅行代理店の社員から、アプリケーション (「Customer」トランザクション) で顧客情報を処理する際に、入力制御の動作が必要であるという話がありました。どのような動作でしょうか。

次のような動作です:

- 「名字だけ、または名前だけの顧客の入力は、システムで許可しない」
- 「電話番号が入力されておらず、入力漏れの場合は、ユーザーに警告を表示する」
- 「顧客情報をシステムに入力した日付を **CustomerAddedDate** に記録する)。
この項目属性には当日の日付を既定値にする」

留意点:

- ルールはセミコロン「;」で終了します。
- 項目属性に適用した **IsEmpty()** メソッドは、項目属性が空の場合は True を返し、それ以外の場合は False を返します。
- **&today** 変数はシステムの一部であり、当日の日付の値がロードされます。
[Rules] の画面で変数を記述する際に、「&」を入力すると、その時点で定義されているすべての変数が表示されます。そこから必要なものを選択することができます。[追加] → [変数] を使用することもできます。

4-1. 登録制限と警告メッセージ表示

「顧客名前」が未入力 (空) の場合、エラーメッセージが表示され、登録処理を完了できないようにルールを定義します。

「顧客名字」が未入力 (空) の場合、エラーメッセージが表示され、登録処理を完了できないようにルールを定義します。

「顧客電話番号」が未入力 (空) の場合、警告メッセージが表示されるようにルールを定義します。

F5 キーを押し、実行し、結果を確認します。

4-2. 動作確認

新しい顧客を入力し、名前を空白のままにしてみてください。情報を保存したり、次のフィールドに移動したりできましたか。
名字だけ入力して、同じようにしてみてください。電話番号でも同じようになりましたか。

4-3. 日付項目の追加と規定値の設定

システムへの入力日を読み取り専用にして、ユーザーが編集できないようにする場合、この動作をどのように確立するか検討してみましょう。

「Customer」トランザクションに以下の項目を追加し、新規登録時に当日の値が既定値となるようにルールを定義します。

名前	タイプ	デスクリプション
▪ CustomerAddedDate	Date	顧客追加日

F5 キーを押し、実行し、結果を確認します。**再編成**が発生します。

5. パターン: データを操作するインターフェースの改善

ここまでの試作結果を旅行代理店の社員に見せたところ、国、カテゴリ、観光名所の情報を、より使いやすい方法（問い合わせの利用、フィルタ機能、データの挿入、変更、削除など）で処理できるようにしたいという要望がありました。

これを実現するには、3つのトランザクションに Work With パターンを適用することで対応可能です。

注:

- GeneXus には、トランザクションごとに複数のオブジェクトの自動作成を実装する「Work With」パターンがあります。
- Work With for Smart Devices も利用できますが、適用するパターンは、構築中の Web アプリケーションに対応するものであることを確認してください。

5-1. パターンの適用

「Country」、「Category」、「Attraction」へ [Work With for Web] を適用します。

以下の画像のように各トランザクションの「Patterns」エレメントにて [Work With for Web] を選択し、

「保存時にこのパターンを適用」チェックボックスにチェックを入れ、保存します。



パターンを適用した結果、Category、Country、および Attraction トランザクションが開発者メニューに表示されなくなります。かわりに「WWCategory, WWCustomer, WWAtraction」といったメニュー項目が表示されます。

一歩進んだ学習:

一般に、GeneXus は引数を必要とするオブジェクトを開発者メニュー上に表示しません。パターンの適用の結果、トランザクションオブジェクトには引数(Parm ルール)が必要になり、開発者メニューから表示されなくなりました。その一方、引数が不要な自動生成された一覧画面が、メニュー上に表示されるようになりました。この「引数の概念」や「Parm ルール」は、研修の後半で学習します。

5-2. 実行結果の確認

パターンが適応された画面を使用し、次の操作を確認します。

また、必要に応じてパターンのカスタマイズを行います。

5-2-1. 国の新規登録

国の一覧画面から新しい国の入力画面を呼び出し、新規登録を実行します。

「+追加」をクリックすることで可能です。

5-2-2. 国の更新

国の一覧画面から既存の国の変更画面を呼び出し、更新 を実行します。(例：都市の追加)

該当レコード行の「更新」をクリックすることで可能です。

5-2-3. 国の削除

国の一覧画面から既存の国の削除画面を呼び出し、削除を実行します。

該当レコード行の「削除」をクリックすることで可能です。

5-2-4. 国の詳細情報表示

国の一覧画面から既存の国のデータを表示する画面を呼び出します。

該当レコード行の _____ (アンダーライン)がついたリンクになっている列をクリックすることで可能です。

5-2-5. 国のフィルタリング

国の照会画面にて、国名でフィルタリングを実行します。

「国名」欄に文字を入力することでフィルタリングが実行されます。

5-2-6. 観光名所の新規登録

観光名所の一覧画面から新しい観光名所の入力画面を呼び出し、いくつかの観光名所を新規登録します。

(例: 中国の北京にある万里の長城、フランスのパリにあるエッフェル塔 など)。

5-2-7. 表示項目のカスタマイズ

観光名所の照会画面にて、国と都市の識別子は表示する必要がないという要望がありました。

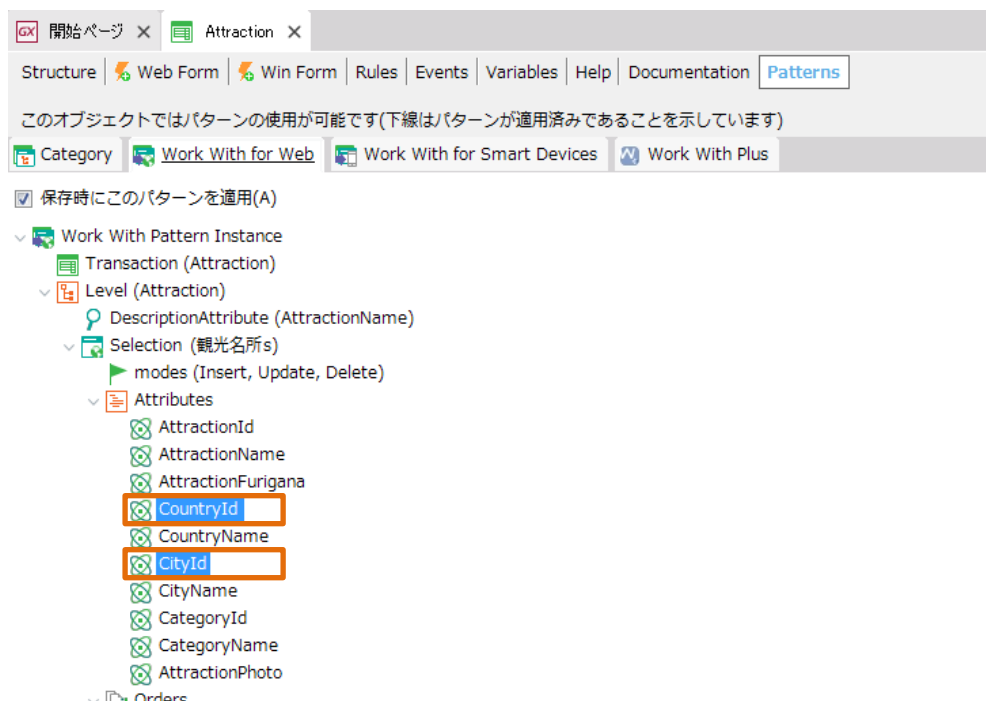
観光名所s 🔍 観光名所名 + 追加

観光名...	観光名...	観光名...	国番号	国名	都市番号	都市名	カテゴ...	カテゴ...	観光名...		
3	エッフェル塔	えつふえるとう	2	フランス	1	パリ	0			更新	削除
1	ルーブル美術館	るーぶるびじゅつかん	2	フランス	1	パリ	0			更新	削除

観光名所の Work With 画面から該当項目を表示しないようにします。

留意点:

- ・ 照会画面に表示する項目は [Selection] ノード配下の [Attributes] ノードに含まれているものです。
- ・ 項目を表示しないようにする場合、以下の 2 種類の方法があります。
 - 項目を削除：表示しない項目を選択し、キーボードの [Delete] キーを押すか、項目を右クリックし [削除] を選択。
 - 項目を非表示：表示しない項目を選択し、[Visible] プロパティを [False] に変更。



実行し、結果を確認します。F5 キーを押します。

5-2-8. 観光名所のフィルタリング

観光名所の照会画面にて、観光名所名でフィルタリングを実行します。

現行の仕様では「観光名所名」欄に文字を入力することでフィルタリングが実行されます。

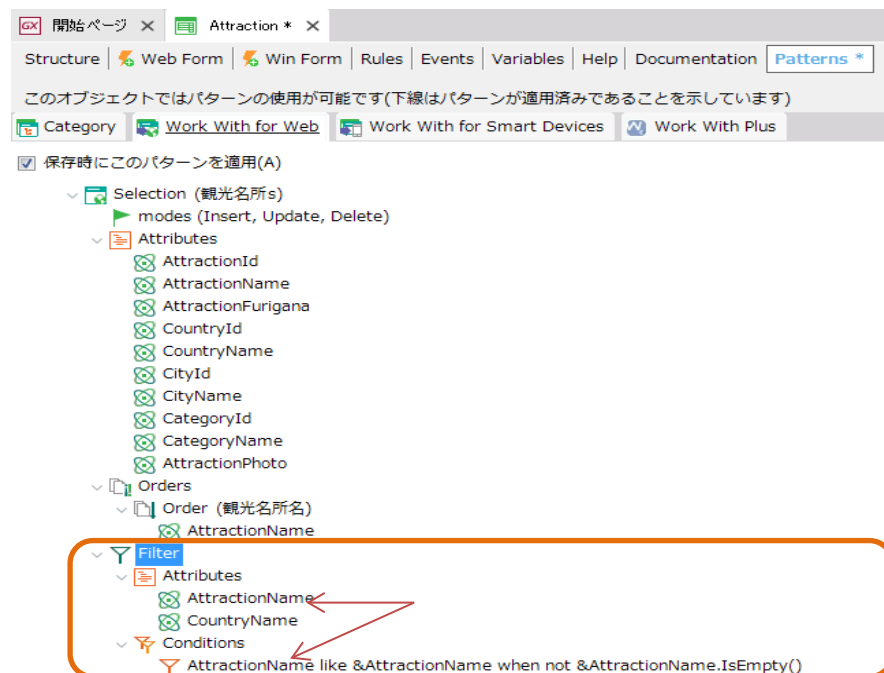
この画面で国名 (例：フランス) で抽出したいとの要望がありました。

国名でフィルタリングを行う機能はまだ実装されていないため、このトランザクションの Work With パターンをカスタマイズして、追加する必要があります。

まず、パターンの既存のフィルタがどのように指定されているかを確認します：

[Selection] ノード配下の [Filter] ノードの定義によってフィルタが指定されています。

サブノードとして持つ [Attributes] ノードにフィルタの対象となる項目を指定し、[Conditions] ノードに実行するフィルタを記載します。



フィルタを追加する場合には、以下のステップが必要となります。

1. [Attributes] ノードを右クリックし、[項目属性を選択] を選択
2. 表示されるダイアログから対象としたい項目属性を選択し、[OK] をクリック
3. 表示される確認ダイアログにて [Yes] をクリック

別解：複雑な抽出条件を設定する場合の設定例

1. [Attributes] ノードを右クリックし、[追加] を選択してから、[Filter Attribute/Variable] を選択
2. [Attributes] ノード配下に新たに追加されたノードの [Name] プロパティで対象となる項目を指定
3. [Conditions] ノードを右クリックし、[追加] を選択してから、[Condition] を選択
4. [Conditions] ノード配下に新たに追加されたノードの [Value] プロパティで実行するフィルタを記載

5-2-9. 照会画面のオーダー

観光名所一覧を、観光名所名または国名の順でユーザーから表示できるようにしたいという要望がありました。(既定では観光名所名の並び替えのみが実装されています。)

「国名」による並び替えも行えるようにカスタマイズを行います。

まず、パターンの既存のオーダーがどのように指定されているかを確認します：

[Selection] ノード配下の [Orders] ノードの定義によってオーダーが指定されています。

サブノードとして持つ [Order (<名前>)] ノードに並び替えに使用される項目がリストされます。

(<名前> 部分は [Name] プロパティの値)

複数種類の並び替えを利用するためにはこの [Orders] ノード配下に複数の [Order (<名前>)] ノードが必要となります。

Category
Work With for Web
Work With for Smart Devices
Work With Plus

☒ 保存時にこのパターンを適用(A)

Work With Pattern Instance
Transaction (Attraction)
Level (Attraction)
DescriptionAttribute (AttractionName)
Selection (観光名所s)
modes (Insert, Update, Delete)
Attributes
Orders
Order (観光名所名)
AttractionName
Filter
Attributes
AttractionName

上記の記述を基に「国名」でのオーダーも実行できるようにカスタマイズを実装し、実行結果の確認を行います。

Application Name
by GeneXus

Recents 観光名所s

X HIDE FILTERS
観光名所s
観光名所名

Ordered By : 国名

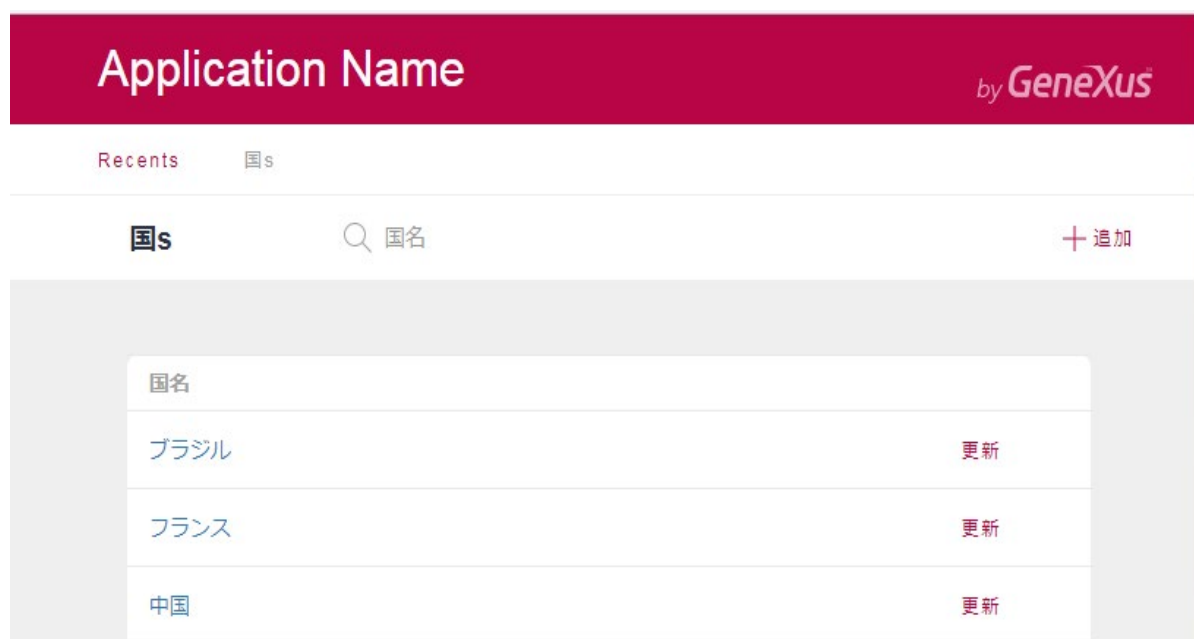
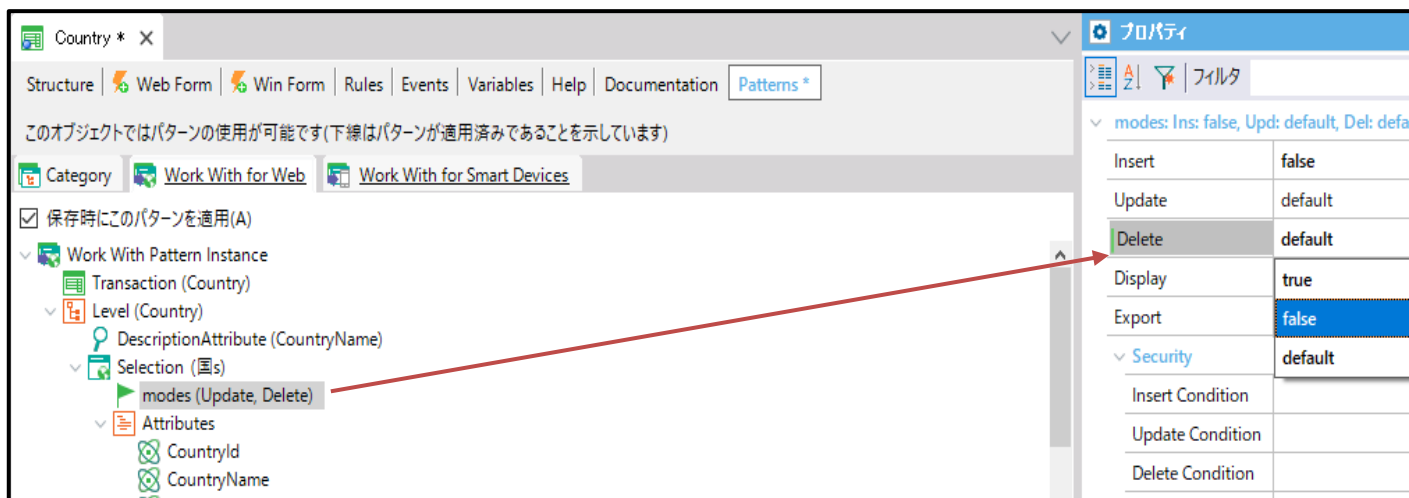
観光名所名
国名

国名

観光名所...	観光名所名	観光名所...	国名	都市名	カテゴリ...	カテゴリ名	観光名所...
1	ルーブル美術館	るーぶるびじゅつかん	フランス	パリ	0		更新 削除
3	エッフェル塔	えっふえるとう	フランス	パリ	0		更新 削除
2	万里の長城	ばんりのちようじょう	中国	北京	0		更新 削除

5-2-10. 一覧画面の削除機能の無効化

パターンを適用して生成したメイン画面でユーザーに国の削除機能を提供することは推奨されません。パターンのインスタンスを編集し、必要なプロパティを変更してください。



6. 「Flight」 および 「Airport」 トランザクションと、サブタイプを定義する必要性

次に、旅行代理店が取り扱っているフライトを記録する必要があります。各フライトはフライト番号を持ち、ある空港から別の空港へと移動します。また、各フライトには価格も設定されています。価格用に、Numeric(10.0) のドメインを作成します。







一方、フライトとは別に、空港を記録するためのトランザクションも作成します。各空港には、空港番号、名前、空港の国および都市が設定する必要があります。

各フライトに出発空港と到着空港があることをどのように定義すればよいでしょうか。







6-1. トランザクションの定義

トランザクションは次のように定義します。

名前 : Airport デスクリプション : 空港

名前	タイプ	デスクリプション
 AirportId	Id	空港番号
 AirportName	Name	空港名
 CountryId	Id	国番号
 CountryName	Name	国名
 CityId	Id	都市番号
 CityName	Name	都市名

名前 : Flight デスクリプション : フライト



名前	タイプ	デスクリプション
 FlightId	Id	フライト番号
 FlightPrice	Price = Numeric(10.0)	フライト価格
 FlightDepartureAirportId	Id	出発空港番号
 FlightDepartureAirportName	Name	出発空港名
 FlightArrivalAirportId	Id	到着空港番号
 FlightArrivalAirportName	Name	到着空港名

  マークは、次ページのサブタイプグループの定義のあとにオブジェクトを開き直すと表示されます。



6-2. サブタイプグループオブジェクトの定義

フライトに出発空港と到着空港を定義するためにサブタイプグループを 2 つ作成します。新規オブジェクトの作成ダイアログの「共通」カテゴリで、サブタイプグループオブジェクトは選択できます。

名前 : Group1 デスクリプション : 出発空港

サブタイプ	スーパータイプ
 FlightDepartureAirportId	AirportId
 FlightDepartureAirportName	AirportName

名前 : Group2 デスクリプション : 到着空港

サブタイプ	スーパータイプ
 FlightArrivalAirportId	AirportId
 FlightArrivalAirportName	AirportName




6-3. 実行結果の確認

実行し、結果を確認します。**F5** キーを押します。**再編成**が発生します。

フライトの入力時に、そのフライトに割り当てている出発空港が存在しない場合はエラーが実行されることを**確認**します。

同様に、到着空港も確認します。

IDE に戻りトランザクションオブジェクトを開き直し、次の点に確認してください。

- 上向き矢印のアイコン  は、その項目属性が外部キーであること、すなわち、別のテーブルを参照していることを示します。
- 下向き矢印のアイコン  は、その項目属性が別のテーブルから推論されていることを示します。
-  のアイコンは、その項目属性がサブタイプであることを示します。

6-4. 登録制限

出発空港が到着空港と同じフライトは入力できないようにする必要があります。

「出発空港番号」と「到着空港番号」が同じ場合、エラーメッセージが表示され、登録処理を完了できないようにルールを定義します。このエラーチェックを実装し、実行環境でテストしてください。

7. グローバル式

各フライトに割引率を記録します。この割引率を格納するための新しい項目属性を Flight トランザクションに定義します。新しい項目属性に次の名前を設定します: **FlightDiscountPercentage**。データタイプは、3桁の数値を持つ「Percentage」ドメインに設定します。

割引適用後の「フライト最終価格」を表示する必要があります。割引率の項目とは別に **FlightFinalPrice** という名前の項目属性を新しく定義し、フライトの最終的な価格を自動的に計算する**グローバル式**を設定します。

7-1. トランザクションの修正

「Flight」トランザクションに以下の項目を追加します。

名前	タイプ	デスクリプション
▪ FlightDiscountPercentage	Percentage = Numeric(3.0)	フライト割引率
▪ FlightFinalPrice	Price	フライト最終価格

7-2. グローバル式の定義

「FlightFinalPrice」項目の式列に以下の記述を追加します。

FlightPrice * (1 – FlightDiscountPercentage / 100)

名前	タイプ	デスクリプション	式	Null 許容
Flight	Flight	フライト		
FlightId	Id	フライト番号		No
FlightPrice	Price	フライト価格		No
FlightDepartureAirportId	Id	出発空港番号		No
FlightDepartureAirportName	Name	出発空港名		
FlightArrivalAirportId	Id	到着空港番号		No
FlightArrivalAirportName	Name	到着空港名		
FlightDiscountPercentage	Percentage	フライト割引率		No
FlightFinalPrice	Price	フライト最終価格	FlightPrice*(1-FlightDiscountPercentage/100)	

7-3. 実行結果の確認

実行し、結果を確認します。**F5** キーを押します。**再編成**が発生します。

[影響分析(Impact Analysis)] で、物理的に作成される項目属性と作成されない項目属性を確認し、アプリケーションの実行結果を確認してください。

座席数の表示

1 つのフライトには複数の座席を記録します。座席は数字と文字で識別され、座席位置を記録します。この座席数は、各フライト別にカウントする必要があります。

7-4. ドメインの定義


各フライトに座席を定義する前に、座席の文字は A から F までの制限された範囲から選択し、座席位置も 3 種類から選択するように要望がありました。

この要望を満たすために列挙型ドメインを使用します。

7-4-1. 以下のドメインを「ドメイン」ウィンドウにて定義します：

名前	タイプ	デスクリプション
SeatChar	Character(1)	SeatChar
Location	Character(1)	Location

7-4-2. SeatChar の [Enum Values] プロパティを設定します。

該当のプロパティを選択すると表示される  ボタンをクリックし、表示される「値エディター」ダイアログに次の内容を追加します。

名前	デスクリプション	値
A	A	A
B	B	B
C	C	C
D	D	D
E	E	E
F	F	F

7-4-3. Location の [Enum Values] プロパティを設定します。

順番 7-4-2. と同様に以下の内容を追加します。

名前	デスクリプション	値
Window	窓側	W
Middle	中間	M
Aisle	通路側	A





7-5. 「Flight」トランザクションの変更

座席が管理できるように「Flight」トランザクションに次の名前の第 2 のレベルを作成します：Seat
座席は数字と文字で識別されるので、第 2 レベルの主キーは、次の 2 つの項目属性が含まれている必要があります。

- **FlightSeatId:** 座席番号の記録用

- **FlightSeatChar:** 文字の入力用

以下のように「Seat」レベルとレベル配下の項目を追加します。

名前	タイプ	デスクリプション
 Seat	Seat	座席
 FlightSeatId	Id	座席番号
 FlightSeatChar	SeatChar	座席文字
 FlightSeatLocation	Location	座席位置

留意点:

- 特定の項目属性を主キーの一部と定義する場合、該当項目属性を右クリックし、[主キーを設定・解除]を選択します。

7-6. 作業経過の確認

実行し、結果を確認します。**F5** キーを押します。**再編成**が発生します。

フライトのフォームを確認すると、座席の文字が A から F までをコンボボックスから選択することができ、同様に各座席の位置が、「窓側」、「中間」、「通路側」をコンボボックスで指定できることが分かります。

7-7. 予約可能な座席数の表示

7-7-1. トランザクションの修正

「Flight」トランザクションの第 1 レベルに、以下の項目を追加します。

名前	タイプ	デスクリプション
▪ FlightCapacity	Numeric(4.0)	フライト座席数

7-7-2. グローバル式の定義

「FlightCapacity」項目の式列に以下の記述を追加します。

Count(FlightSeatLocation)

実行し、結果を確認します。**F5** キーを押します。

登録制限

フライトに収容可能な乗客数 (座席数) をチェックする必要があるという要望がありました。1 つのフライトの収容乗客数が 4 未満のフライトは登録できません。これらの要望を満たすために次の修正を実行します。

7-8. 登録制限

また、データの登録処理の結果、「フライト座席数」が 4 未満の場合、エラーメッセージが表示され、登録処理を完了できないようにルールを定義します。このエラーメッセージは、座席の全入力処理された後に判定を行う必要もあります。そのため、座席(第 2 レベル)に対する処理が終わったタイミングのトリガーイベント節を定義に加えます。

実行し、結果を確認します。**F5** キーを押します。

8. インデックス

[表示] → [テーブル] メニューを選択し、国や観光名称の名前が重複しないようにするためのコントロールを設定します。

8-1. 観光名所名の一意インデックスの追加

Attraction テーブルオブジェクトを開き、[Indexes] エlement を開き、新しいユーザーインデックスを追加します。
追加したインデックスの「順序」列を「重複」から「一意」に変更し、インデックスには AttractionName 項目属性を追加します。

8-2. 国名の一意インデックスの追加

Country テーブルオブジェクトを開き、[Indexes] エlement を開き、新しいユーザーインデックスを追加します。
追加したインデックスの「順序」列を「重複」から「一意」に変更し、インデックスには CountryName 項目属性を追加します。

8-3. 実行結果の確認



F5 キー押し、実行したアプリケーションで観光名所の登録画面にてすでに登録されている観光名所名を入力してみます。
同じように国の登録画面にてすでに登録されている国名を入力してみます。

9. PDF レポート

次に、アプリケーションの一部として、必要なデータが記載された PDF レポートをユーザーが表示できる機能を実装する必要がある場合を想定しましょう。たとえば、データベースに格納されている観光名所を名前順に表示するレポートが必要であると想定します。



観光名所一覧

番号	観光名所名	国	写真
3	エッフェル塔	フランス	
1	ルーブル美術館	フランス	

9-1. 観光名所一覧の作成

データベースに格納されたすべての観光名所を国名とともに一覧出力を行う PDF を作成します。

9-1-1. プロシージャの定義

帳票出力を行うためのプロシージャーを次のように定義します。

名前 : AttractionsList デスクリプション : 観光名所一覧

9-1-2. レイアウトの定義

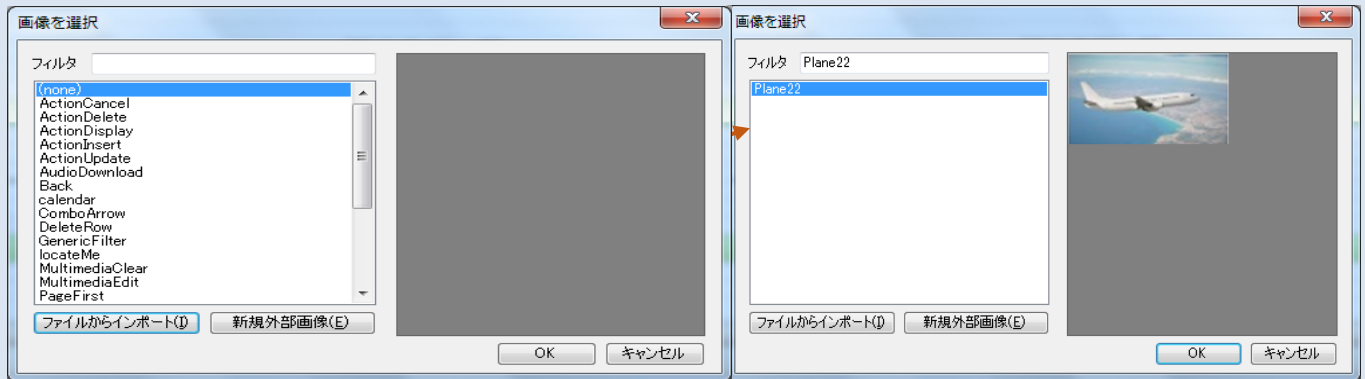
[Layout] エLEMENTでレポートのタイトルテキスト、各出力項目のタイトルテキスト、各項目のデータベースの値を表示するためのプリントブロックを定義します。

留意点:

- プリントブロックはプロシージャー内で一意の名前が必須となります。
各プリントブロックの [Name] プロパティで変更が可能です。
- 新しいプリントブロックを追加する場合、[Layout] ELEMENT内で右クリックし、[プリントブロックを挿入] を選択
- プリントブロックにコントロールを配置する場合、[ツールボックス] ウィンドウより該当コントロールをドラッグ&ドロップします。
 - テキスト → [テキストブロック]
 - 四角形の図 → [長方形]
 - 直線 → [線]
 - 項目属性や変数(可変な値の表示) → [項目属性/変数]

推奨: タイトルの左側の画像を配置するには、ツールボックスの画像コントロールを使います。

画像はナレッジベースに含まれている必要があります。このためには、[ファイルからインポート] オプションを使い、画像を探して名前を割り当てます。



9-1-3. 出力処理の定義

[Source] エレメントでレポートを出力するための処理を定義します。

留意点:

- プリントブロックを出力するためには [Print] コマンドが必要となります。
- データベースを参照し、そのデータを取得やデータに対する処理を実行するためには以下のコマンドが必要となります。

```
For Each <ベーストランザクション名>
    <処理>
Endfor
```
- データを観光名所名順で表示するためには [Order] 節を [For Each] コマンドの <ベーストランザクション名>と <処理> の間に記述し、節の後ろに対象項目名を指定します。

9-1-4. 実行結果の確認

実行し、結果を確認します。オブジェクトタブを右クリックして [これだけを実行] を選択します。

留意点:

- GeneXus で、レポートを表示するには、PDF 形式で生成する必要があります。
 このためには、Procedure オブジェクトで次のようにプロパティを変更する必要があります:
 - [Main program] = [True]
 - [Call Protocol] = [HTTP]
- また、次のルールも定義する必要があります:
 - Output_file(' <ファイル名> ', 'PDF');
- レポートを単体で実行し、結果を表示するためには、オブジェクトタブを右クリックして [これだけを実行] を選択します。

9-1-5. ナビゲーション表示の確認

プロシージャーの [ナビゲーション表示] に記載された情報を確認しましょう。

9-1-6. 表示順の変更

このレポートを国名順に出力してほしいという要望がありました。

観光名所名順ではなく、国名順で表示するように処理の定義を変更します。

実行し、結果を確認します。オブジェクトタブを右クリックして [これだけを実行] を選択します。

ナビゲーションレポートに記載された情報を確認してテストしてください。

9-1-7. データのフィルタリング

ある国内の観光名所一覧のみ(例：フランス)を出力してほしいという要望がありました。

現在登録されている国でフィルタリングされるように処理の定義を変更します。

実行し、結果を確認します。オブジェクトタブを右クリックして [これだけを実行] を選択します。

ナビゲーションレポートに記載された情報を確認してテストしてください。







留意点:

- フィルタリングを行うためには [Where] 節を [For Each] コマンドの <処理> の前に記述し、節の後ろに判定条件を指定します。
- [Order] 節と [Where] 節は必ずこの順番で定義する必要があります。

9-2. カテゴリごとの観光名所一覧の作成

すべてのカテゴリと、各カテゴリの観光名所を一覧で表示する必要があります。

この PDF には、以下のような出力を想定しています。

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	

9-2-1. プロシーチャーの定義

帳票出力を行うためのプロシーチャーを次のように定義します。

名前 : CategoriesAndAttractions デスクリプション : カテゴリごとの観光名所一覧

ヒント:

最近作成したリストの一部を再利用するにはプロシーチャー名で右クリックし、[名前を付けて保存...] オプションを選択して別名で保存します。

教科書の指定（もしくは前ページの留意点）に記述されている PDF 印刷のためのプロパティ設定とルールを定義を行います。

9-2-2. レイアウトの定義

[Layout] エLEMENTでレポートのタイトルテキスト、各出力項目のタイトルテキスト、各項目のデータベースの値を表示するためのプリントブロックを定義します。

9-2-3. 出力処理の定義

[Source] ELEMENTでレポートを出力するための処理を定義します。

留意点:

- カテゴリごとにアトラクションを複数出力するためには、異なる複数のテーブルを参照する必要があるため、ネストされた [For Each] コマンドが必要となります。

9-2-4. 実行結果の確認

実行し、結果を確認します。オブジェクトタブを右クリックして [これだけを実行] を選択します。

テストプラン:

- システムに「グルメ」「未来的建築」などの新しいカテゴリを追加します。リストを再実行します。カテゴリは一覧に追加されましたか。

9-2-5. コントロールブレイクの実装

新しいカテゴリ(例: ランドマーク)を追加した後、前項目で作成したレポートに表示されることを確認します。これは観光名所が 1 つもないカテゴリが表示されるはずで、観光名所が関連付けられていないカテゴリは、除外されるように処理の定義を変更します。


留意点:

- コントロールブレイクを実装するためには以下の条件をすべて満たす必要があります。
 - ネストされた [For Each] コマンド
 - 同じテーブルを参照
 - [Order] 節によるグルーピング項目の指定
- 観光名所に関連付けられたカテゴリだけを表示するために参照するテーブルは「カテゴリ」ではなく「観光名所」である必要があります。

実行し、結果を確認します。ナビゲーションレポートがどのように変更されたか確認しましょう。

9-3. 国ごとの観光名所数一覧の作成

データベースに格納されたすべての国の名前と、各国に関連した**観光名所の数**を表示する必要があります。
観光名所の数はデータベース上に項目としては定義されていません。どのような定義が必要でしょうか。
この PDF には、以下のような出力を想定しています。



国名	観光名所数
ブラジル	1
フランス	2
中国	1
アメリカ合衆国	1

9-3-1. プロシーチャーの定義

帳票出力を行うためのプロシーチャーを次のように定義します。

名前 : NumbersOfattractionsByCountry デスクリプション : 国ごとの観光名所数一覧

教科書 196 ページのプロパティの設定とルールの定義を行います。

9-3-2. レイアウトの定義

[Layout] エLEMENTでレポートのタイトルテキスト、各出力項目のタイトルテキスト、各項目のデータベースの値を表示するためのプリントブロックを定義します。

留意点:

- 「観光名所数」を表示するためにはあらかじめ [Variables] ELEMENTで変数を定義する必要があります。

9-3-3. 出力処理の定義

[Source] ELEMENTでレポートを出力するための処理を定義します。

留意点:

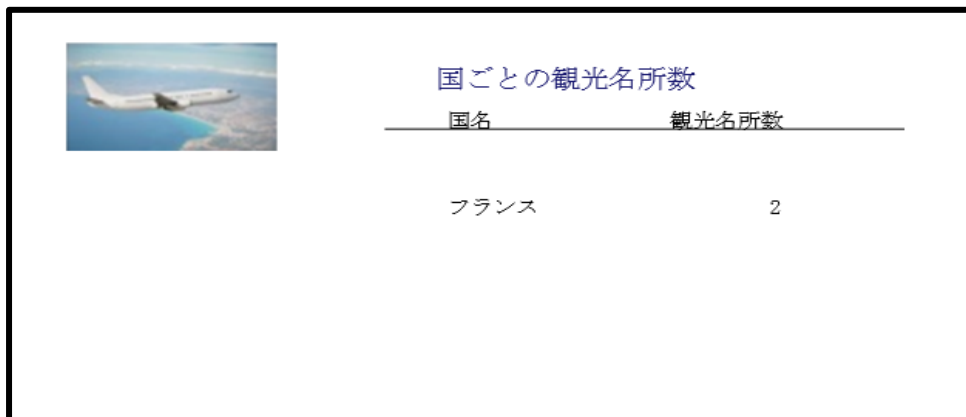
- 変数で結果を表示するためには、事前に割り当て処理を記述する必要があります。
- 観光名所の数を取得するためには [Count(<カウント項目>)] 関数を使用する必要があります。

9-3-4. 実行結果の確認

実行し、結果を確認します。オブジェクトタブを右クリックして [これだけを実行] を選択します。

9-3-5. データのフィルタリング

観光名所が特定数(例：2)以上ある国をすべて出力してほしいという要望がありました。



国名	観光名所数
フランス	2

各国が持つ観光名所の数でフィルタリングされるように処理の定義を変更します。

実行し、結果を確認します。オブジェクトタブを右クリックして [これだけを実行] を選択します。

10. パラメーターの受け渡し

一般的に、オブジェクトがパラメーターを受け取って、そのパラメーターに基づいてロジックを実行する必要がある場合があります。たとえば、データベース内のすべての観光名所のうち、特定の名前の範囲に含まれる観光名所のリストを定義するケースなどが考えられます。

本課題では、ユーザーが抽出条件を指定する画面を実装し、指定された値をパラメーターで受け渡して、PDF を出力するオブジェクト(AttractionsList を改造したもの)を呼び出せるようにします。

10-1. 呼び出し元 Web パネルの定義

次に抽出条件を指定する Web パネルを作成します（すでに K B に、このオブジェクトがある場合は、編集します）。

名前：EnterAttractionFilter デスクリプション：観光名所 PDF リスト出力画面

10-1-1. 変数宣言

[Variables] エLEMENTで、2 つの変数 FromAttractionFurigana, ToAttractionFurigana を定義します。この変数の用途は、抽出条件の入力領域と、次ページで作成するプロシーチャーの引数に使います。

10-1-2. WEB フォームとイベント

- ・ [Web Layout] エLEMENTに、入力領域として 2 つの変数を配置します。
- ・ ボタンコントロールを配置します。このとき、クリック時に呼び出されるイベント名が求められるので **callAttractionsByName** とします（任意のイベント名が使えます）。

このボタンの Caption プロパティに「ふりがなフィルタつき観光名所リスト」とします。

この時点ではまだ、呼び出し先オブジェクトが作成されていません。呼び出し構文を保存するには、すでにオブジェクトが存在する必要があります。次ページの通り、呼び出し先を作成します。

留意点:

- ・ ボタンをクリックした時に実行する処理を記述するためのイベントは以下のいずれかの方法で定義可能です。
 - [Web Layout] エLEMENTの該当ボタンコントロールをダブルクリック
 - 次の内容を直接記述する
Event <イベント名>
EndEvent
- ・ オブジェクトの呼び出しは以下のいずれかの記述で実装可能です。
 - <オブジェクト名>.Call(<項目属性/変数/定数>)
 - <オブジェクト名>(<項目属性/変数/定数>)
- ・ また、必要に応じて「,」で区切り、複数の値を受け渡すことが可能です。

10-2. 呼び出し先 PDF 作成プロシーチャーの定義

前項で作成した、観光名所リスト出力するためプロシーチャー(AttractionsList)を別名で保存し、次のように定義します。

名前: AttractionsByName デスクリプション: 名前に基づく観光名所一覧

10-2-1. 変数の宣言

[Variables] エLEMENTで、引数を格納するための変数 FuriganaFrom,FuriganaTo を宣言します。

ヒント:

- 呼び出し元オブジェクトで呼び出しを行うために使用する変数と、呼び出し先オブジェクトで受け取るパラメーターを宣言するために使用する変数は、同じ名前である必要はありませんが、データタイプは互換可能である必要があります。
- 変数のデータタイプを項目属性に基づいて定義すると、項目属性のデータタイプにリンクされ、後で項目属性のデータタイプを変更すると変数のデータタイプも変更されます。Based on プロパティで指定リンクを指定します。

10-2-2. 引数の宣言

[Rules] エLEMENTで引数を受け取るための宣言を行います。

```
parm( in: &FuriganaFrom, in: &FuriganaTo );
```

10-2-3. 出力処理の定義

[Source] エLEMENTで、抽出処理条件を指定し、保存します。

```
print Title
print ColumnTitles
for each Attraction
    where AttractionFurigana >= &FuriganaFrom
    where AttractionFurigana <= &FuriganaTo
    print Attractions
endfor
```

10-2-4. 呼び出し構文の記述

呼び出し元の WEB パネル EnterAttractionFilter に戻ります。

「ふりがなフィルタつき観光名所リスト」 ボタンを右クリックし[イベントに移動]を選択します。

次の呼び出し構文を記述します。

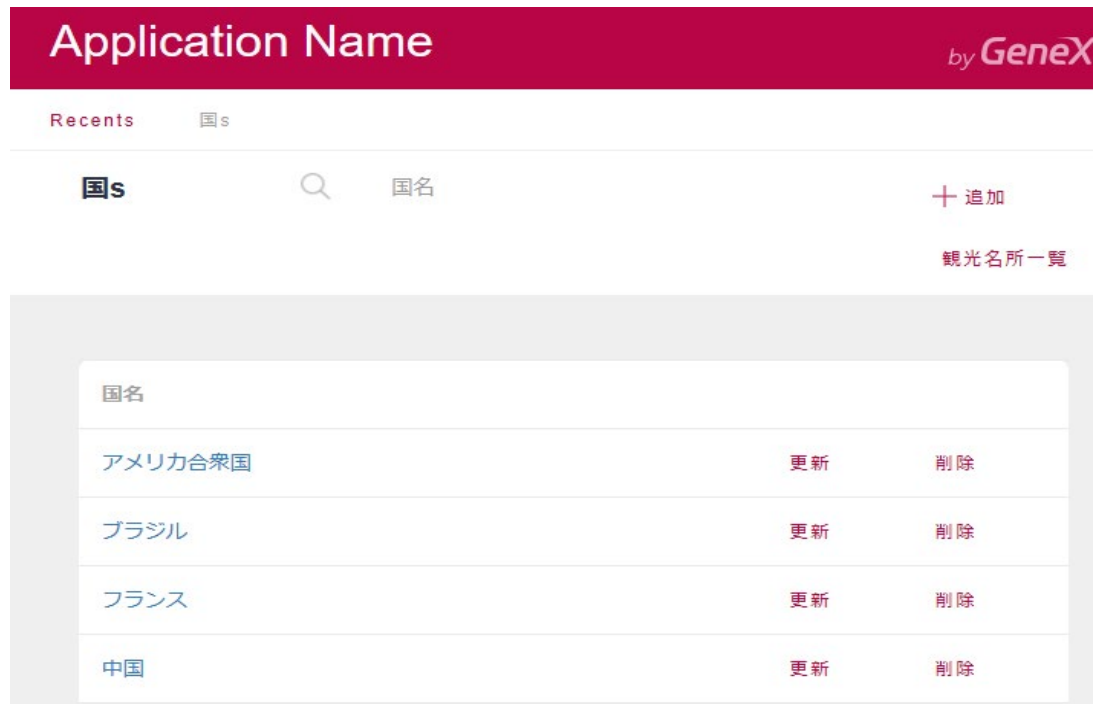
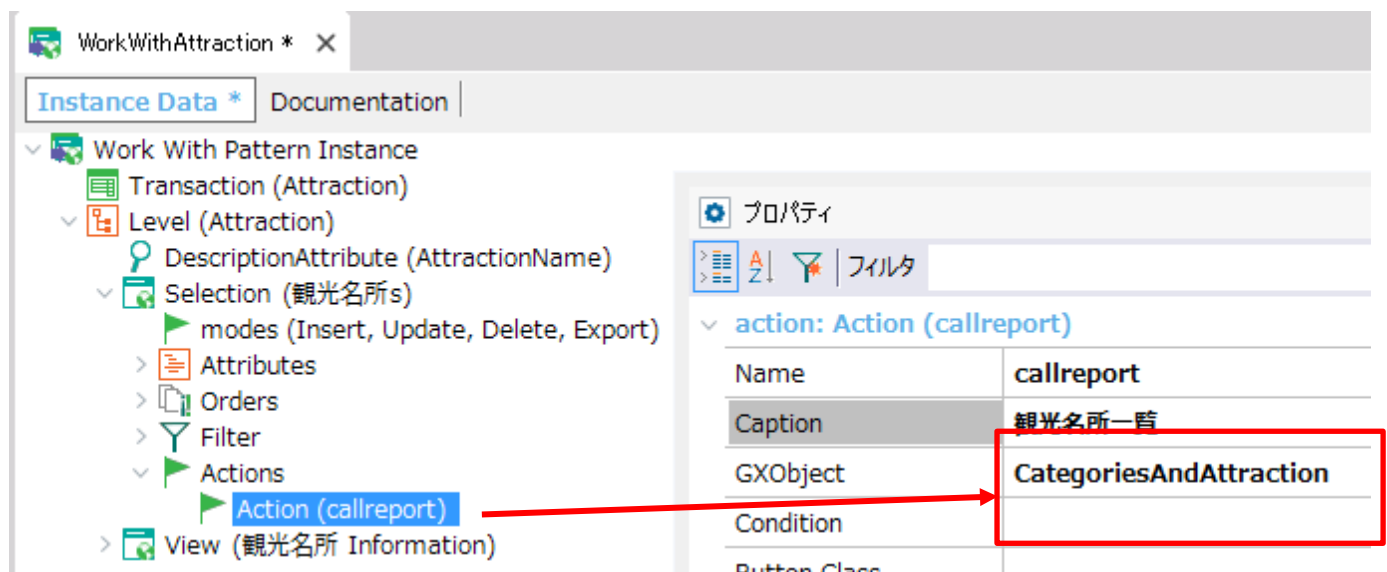
```
AttractionsByName( &FromAttractionFurigana, &ToAttractionFurigana)
```

ヒント:

- この課題では、呼び出し元を途中まで作成してから、呼び出し先を作成しました。呼び出し先を作成してから、呼び出し元を作成してもかまいません。

10-2-5. Workwith 一覧画面からの呼び出し

旅行代理店から、WWCountry のメイン画面にボタンを追加して先ほどのリストを呼び出せるようにしたいとリクエストされました。

**ソリューション :**

11. ビジネスコンポーネント

11-1.キャンペーン期間中の「人気/おすすめ」カテゴリの設置

新たに「人気/おすすめ」カテゴリレコードを新規に追加し、キャンペーン期間中、一時的に東京の「有名なランドマーク」カテゴリに属するすべての観光名所を「人気/おすすめ」カテゴリに変更します。キャンペーン終了時に、この新規に登録したカテゴリレコードを名称で検索し、削除できるように名称を必須入力にします。

11-1-1.エラーメッセージ表示と[Business compornet]プロパティの設定

観光名所トランザクションにて「観光名所名」が未入力 (空) の場合、エラーメッセージが表示され、登録処理を完了できないようにルールを定義します。パターンで生成されたルールの途中に記述せずに、[Rules]の一番下に記述します。

```
13 noaccept(CategoryID) if &Mode = TrnMode.Insert and not &Insert_
14 CategoryID = &Insert_CategoryID if &Mode = TrnMode.Insert
15 noaccept(CategoryID) if &Mode = TrnMode.Insert and not &In
16 /* Generated by Work With Pattern [End] - Do not change */
17 Error("観光名所名が空欄です")
18 if AttractionName.IsEmpty();
```

また、カテゴリトランザクションにて「カテゴリ名」が未入力 (空) の場合、エラーメッセージが表示され、登録処理を完了できないようにルールを定義します。それぞれ、ビジネスコンポーネントとして利用できるように、[Business compornet]プロパティを True にします。

11-1-2.WEB パネルの作成

次に更新処理を呼び出す Web パネルを作成します。

名前: WebpanelCampaign デスクリプション: キャンペーン開始

下図を参考にレイアウトと、プロシージャオブジェクトを呼び出す操作を記述します。

新しいカテゴリ「人気/おすすめ」を作成し、カテゴリ「有名なランドマーク」で東京のすべてのアトラクションに割り当てる

テキストブロックを配置し、Caption プロパティに表示内容を設定します。

Web Form * Rules Events Conditions Variables Help Documentation

<選択されているアクショングループはありません>

MainTable Do

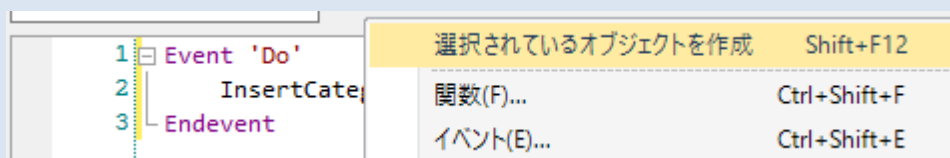
Web Form * Rules Events Conditions Variables Help Do

'Do'

```
1 Event 'Do'
2 InsertCategoryUpdateAttractions()
3 Endevent
```

アドバイス:

- 呼び出し先のプロシージャオブジェクトは、現時点で作成されていないので、呼び出し先を作成するまでこの WEB パネルは保存できません。(//でコメントアウトして、保存できます。)
- 呼び出し元の Event からこれから作成するプロシージャオブジェクトを、作成する方法の1つに「選択されているオブジェクトを作成」があります。

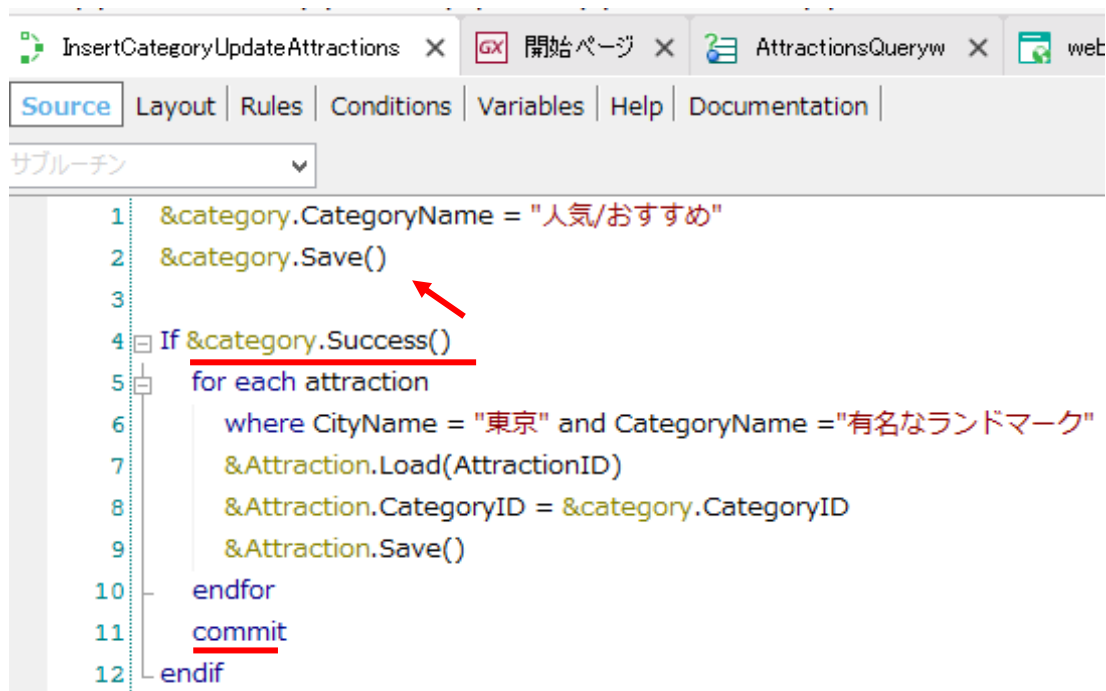


11-1-3. プロシージャオブジェクト作成

名前: InsertCategoryUpdateAttractions デスクリプション: カテゴリの追加と観光名所更新

カテゴリと観光名所のビジネスコンポーネント変数を宣言し、次のロジックを実装します。

- 「人気/おすすめ」カテゴリレコードを挿入
- 東京の「有名なランドマーク」の観光名所を検索し、挿入したカテゴリレコードの ID で更新します。



11-1-4. 動作確認

実行し、結果を確認します。**F5** キーを押します。

(キャンペーン終了時の処理は、実装不要です。教科書の記述を読み、理解を深めてください。)

11-2. ビジネスコンポーネントによるフライトの一斉値上げ(オプション)

この旅行代理店では、指定された割合で全フライトの価格を上げる必要になる場合があります。このため、ユーザーが**値上げ率**を指定し、データベース内のすべてのフライトに**適用**を指示できる**画面**を実装する必要があります。

Web Panel オブジェクトを使うと、さまざまな要件にフレキシブルに対応できる画面を実装できます。本課題では、Web Panel オブジェクトとビジネスコンポーネントの概念に基づき、手続き型のアプローチでの実装を目指します。

11-2-1. ビジネスコンポーネントの定義

「フライト」トランザクションをビジネスコンポーネントとして利用できるようにプロパティを変更します。

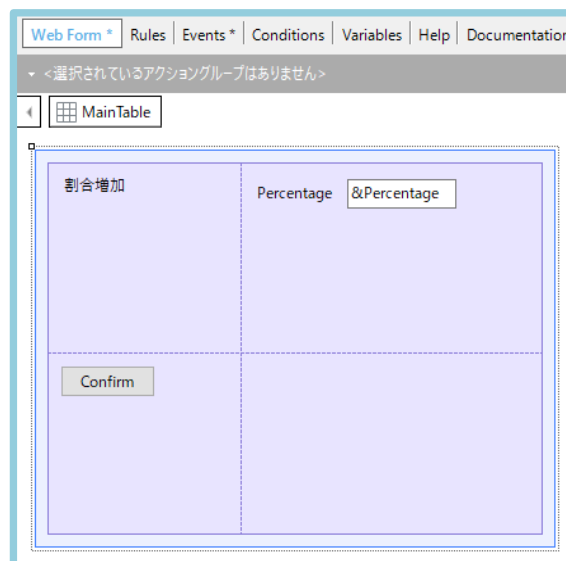
11-2-2. Web パネルの定義

値上げ処理を実行するための Web パネルを次のように定義します。

名前 : WPIIncreasePercentage デスクリプション : 割合増加

11-2-3. 画面の定義

[Web Layout] エLEMENTで「値上げ率」の入力欄と処理を実行するボタンを定義します。



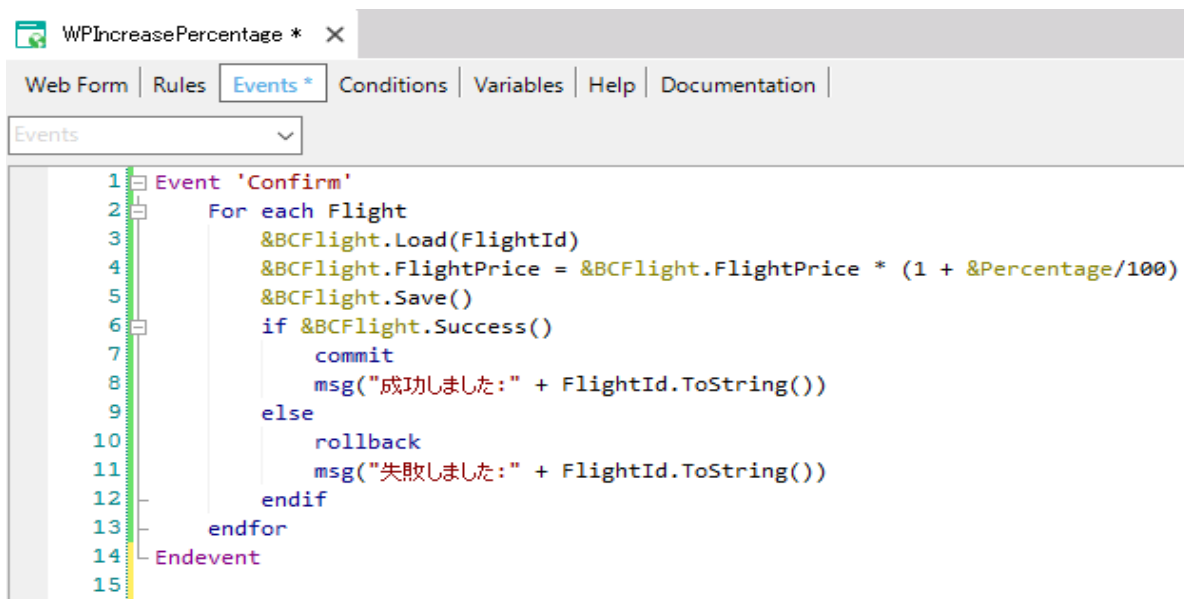
留意点:

- 入力領域として変数を利用するためには、あらかじめ [Variables] エLEMENTで「値上げ率 (Percentage 型)」の変数を宣言してから、[Web Layout]ELEMENTで配置しましょう。

11-2-4. 更新処理の定義

[Events] エLEMENTでボタンを押したときにビジネスコンポーネントを使用する更新処理を定義します。

各フライトを編集して、FlightPrice の値を変更するにはどうすればよいでしょうか。このためには、Flight トランザクションの構造全体を持ちデータベースへの保存も可能な Flight ビジネスコンポーネント型の変数が必要です。

**留意点:**

- ビジネスコンポーネントで既存のレコードを更新や削除を実行する場合、初めにデータを読み込む必要があるため、Load() メソッドを使用する必要があります。
- X の値を 20% 増加させるには、 $X = X * (1 + 20 / 100) = X * 1.20$ という式を利用します。
- 更新対象が全フライトとなるため、データベースを参照する [For Each] コマンドを使用する必要があります。
- ビジネスコンポーネントでレコード操作を実行させるためには Save() メソッドを使用する必要があります。
- レコード操作が正常完了したか確認するためには Boolean 型の戻り値を返す Success() メソッドが利用可能です。
- レコード操作が正常完了した場合、[Commit] コマンドを実行するように定義する必要があります。
- 異常が発生した場合、戻す必要がある場合には [Rollback] コマンドを記述可能です。
- 定義した処理の中で画面にメッセージを表示したい場合には Msg('<出力内容>') 関数が利用可能です。

11-2-5. 実行結果の確認

実行し、結果を確認します。**F5** キーを押します。

留意点:

- 更新が失敗した場合は、その理由を検討しましょう。
フライトトランザクションの画面で、実際に値上げを登録してみます。
- システムで、失敗理由を示すより詳しいエラーメッセージを表示するには、rollback コマンドの後ろに次のように実装します。
&message は「Messages.message」型の変数です。この型はビルドイン SDT の Messages の message サブ構造です。

```
rollback
msg("失敗しました:" + FlightID.ToString())
for &message in &BCFlight.GetMessages()
    msg(&message.Description)
endfor
```

11-3.初期データのデータベースへの一斉展開

開発したアプリケーションを本番稼動する (旅行代理店がアプリケーションを使い始める) 際に、カテゴリ、国、および観光名所に関するデータをすべてデータベース上に展開する必要があります。

本課題では、GeneXus おけるさまざまなデータ投入アプローチで、テーブルの初期化を検討します。

留意点:

- GeneXus では、次のいずれかの方法でテーブルに複数件のデータを一斉に取り込むことが可能です。
 - ビジネスコンポーネントのコレクションを返すデータプロバイダー
 - トランザクションの [Data Provider] プロパティ
 - New コマンド (テーブルに **1** 件のレコードを入力)

11-3-1. ビジネスコンポーネントコレクションのデータプロバイダーによるカテゴリの初期データ展開（オプション）

データプロバイダーの定義に基づき、ビジネスコンポーネントを通じてカテゴリデータを挿入することを検討します。

11-3-1-1. ビジネスコンポーネントの定義

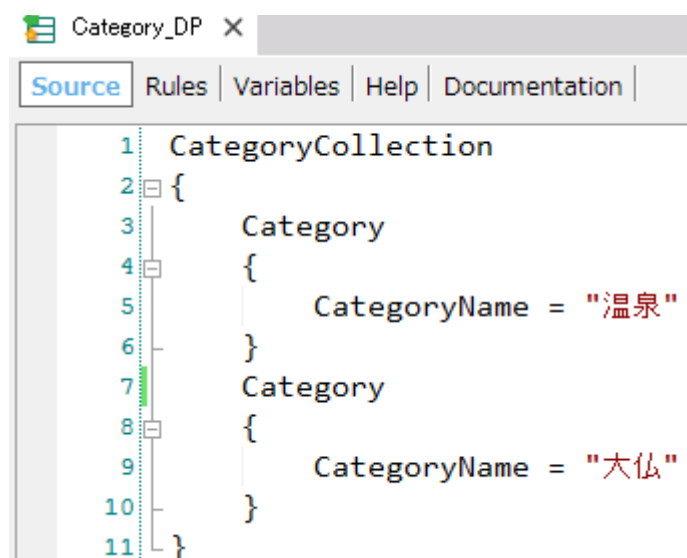
カテゴリトランザクションのビジネスコンポーネント プロパティを True に変更します。

11-3-1-2. データプロバイダーの定義

挿入する新しいカテゴリについて定義したデータプロバイダーを作成します。

名前 : Category_DP

デスクリプション : カテゴリデータ自動生成情報取得



データプロバイダーのプロパティについて:

カテゴリトランザクションは **1** 件のカテゴリを示します。しかし、データプロバイダーでカテゴリの **コレクション** を返す必要があります。したがって、**[Collection]** の値は **[True]** になります。

Output

Infer Structure	No
Output	Category
Collection	True
Collection Name	CategoryCollection

11-3-1-3. カテゴリのための Web パネルの定義とデータプロバイダーの定義

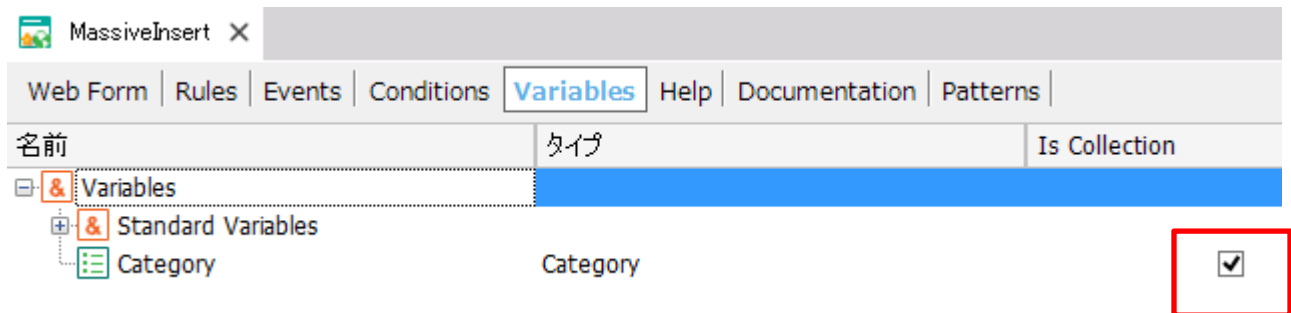
カテゴリを一斉登録するための Web パネルを次のように定義します。

名前 : MassiveInsert

デスクリプション : カテゴリデータ自動生成

11-3-1-4. 変数の宣言

データプロバイダーの値を受け取り、挿入するためのビジネスコンポーネント変数（コレクション）を定義します。

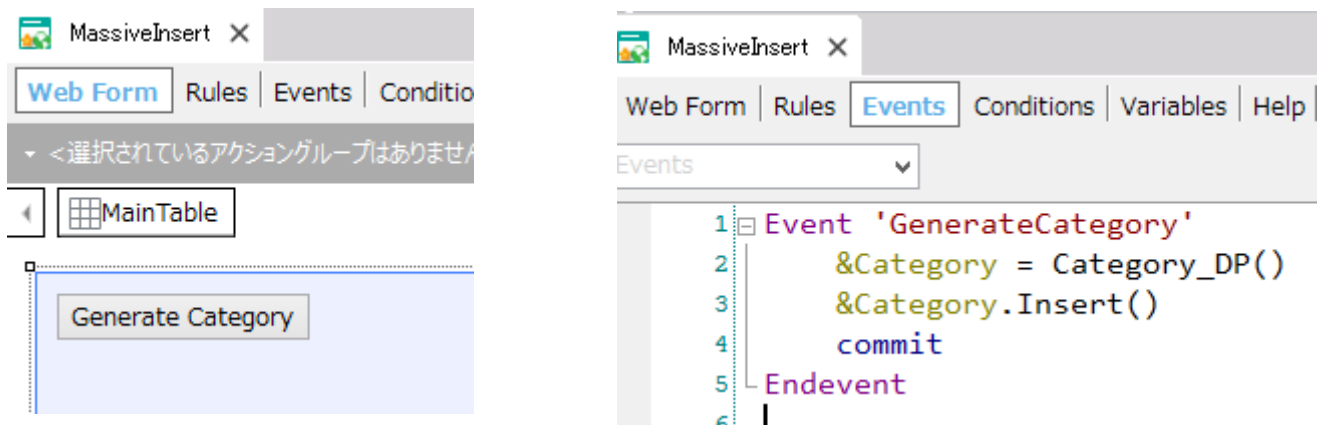


コレクションにチェックが入っていることを確認してください。

11-3-1-5. フォームとイベント定義

[Web Layout]エレメントにて、カテゴリ作成プロセスを自動的にトリガーするボタンを配置し、イベント名を「GenerateCategory」とします。ボタンを右クリックし「イベントへ移動」を選択すると、イベントの定義が楽に行えます。

最終的には次の定義が必要になります。



注: ビジネスコンポーネントのコレクション（この場合はカテゴリのコレクション）に適用された Insert メソッドは、コレクション内のすべての観光名所をデータベースに挿入可能にします。






11-3-1-6. 動作確認

- アプリケーションを実行します。
- MassiveInsert 画面を操作し、ボタンをクリックします。
- Work with カテゴリで、Category_DP で定義されたカテゴリが、正しく作成されたことを確認します

11-3-2. トランザクションの[Data Provider]プロパティによる提携先初期データの展開

11-3-2-1. トランザクションの作成

名前: Supplier デスクリプション: 提携先

名前	タイプ	デスクリプション
 SupplierID	Id	提携先番号
 SupplierName	Name	提携先名
SupplierAddress	Address	提携先住所
 Attraction	Attraction	観光名所
 AttractionId	Id	観光名所番号
 AttractionName	Name	観光名所名

- [Data Provider] プロパティを使用して**提携先の初期データ**を取り込みます。[True] の値を設定します。
- Supplier トランザクションの下に作られるデータプロバイダー「Supplier _DataProvider」を定義します

[Source](#) | [Rules](#) | [Variables](#) | [Help](#) | [Documentation](#)

```

1  SupplierCollection
2  {
3      Supplier
4      {
5          SupplierName = "テストデータ"
6          SupplierAddress = "東京都品川区五反田"
7      }
8      Supplier
9      {
10         SupplierName = "GXツアー株式会社"
11         SupplierAddress = "東京都品川区五反田"
12         Attraction
13         {
14             Attraction
15             {
16                 AttractionID = find(AttractionID,AttractionName="東京スカイツリー",1)
17             }
18         }
19     }
20 }

```

再編成を行うと、指定されたデータを持つテーブルがロードされます。

12. レコードを更新するためのプロシージャオブジェクト

12-1. 「人気/おすすめ」カテゴリの設置と観光名所の更新

キャンペーン期間中、一時的に「東京」の「有名なランドマーク」カテゴリに属するすべての観光名所を「人気/おすすめ」カテゴリに変更していましたが、旅行代理店から、キャンペーン終了時の処理の実装が求められました。

「東京」の「人気/おすすめ」カテゴリに属するすべての観光名所を「有名なランドマーク」に戻し、「人気/おすすめ」カテゴリレコードを削除することが求められています。

12-1-1. プロシージャの定義

名前 : DeleteCategoryUpdateAttractions

デスクリプション : キャンペーン終了

```
1 //観光名所の更新
2 &CategoryID = find(CategoryID,CategoryName = "有名なランドマーク",0)
3 For each Attraction
4     where CityName = "東京" and CategoryName = "人気/おすすめ"
5     CategoryID =&CategoryID
6 Endfor
7
8 //カテゴリの削除
9 For each Category
10     Where CategoryName = "人気/おすすめ"
11     Delete
12 Endfor
```

Find 式で「有名なランドマーク」の主キーを取得
必要な変数は宣言します。

Delete コマンドで、カテゴリのレコードを削除します。

12-1-2. WEB パネルのフォームとイベント定義

上のプロシージャを呼び出すための WebpanelCampaign Web パネルを編集します。

[Web Layout]エレメントにて、イベント名を「Undo」とする Undo ボタンを配置します。

ボタンを右クリックし「イベントへ移動」を選択すると、イベントの定義が楽に行えます。

Undo イベントより、DeleteCategoryUpdateAttractions プロシージャオブジェクトを呼び出します。

12-2. 検討課題 BC vs レコード操作のコマンド 不適切な値による更新

参照整合性に反する処理をした場合、なにか起きるか観察します。

プロシージャオブジェクトの場合

```
1 //観光名所の更新
2 &CategoryID = find(CategoryID,CategoryName)
3 For each Attraction
4     where CityName = "東京" and CategoryName = "人気/おすすめ"
5     CategoryID =&CategoryID
6 Endfor
7
8 //カテゴリの削除
9 For each Category
10     Where CategoryName = "人気/おすすめ"
11     Delete
12 Endfor
```

BCの場合

```
1 &category.CategoryName = "人気/おすすめ"
2 &category.Save()
3
4 If &category.Success()
5     for each attraction
6         where CityName = "東京" and CategoryName = "有名"
7         &Attraction.Load(AttractionID)
8         &Attraction.CategoryID = &category.CategoryID
9         &Attraction.Save()
10    endfor
11    commit
12 endif
```

12-3. 検討課題 全テーブルの情報削除 (オプション)

データベース上の全データを削除するにはどうすればよいでしょうか。

ソリューション

プロシージャオブジェクトを作成し、[Source] エlementに次のとおり定義します:

名前: DeleteAll デスクリプション: プロシージャ全データ削除

```
for each Flight
  for each Flight.Seat
    delete
  endfor
delete
endfor

for each Airport
  delete
endfor

for each Attraction
  delete
endfor

for each Category
  delete
endfor

for each Customer
  delete
endfor

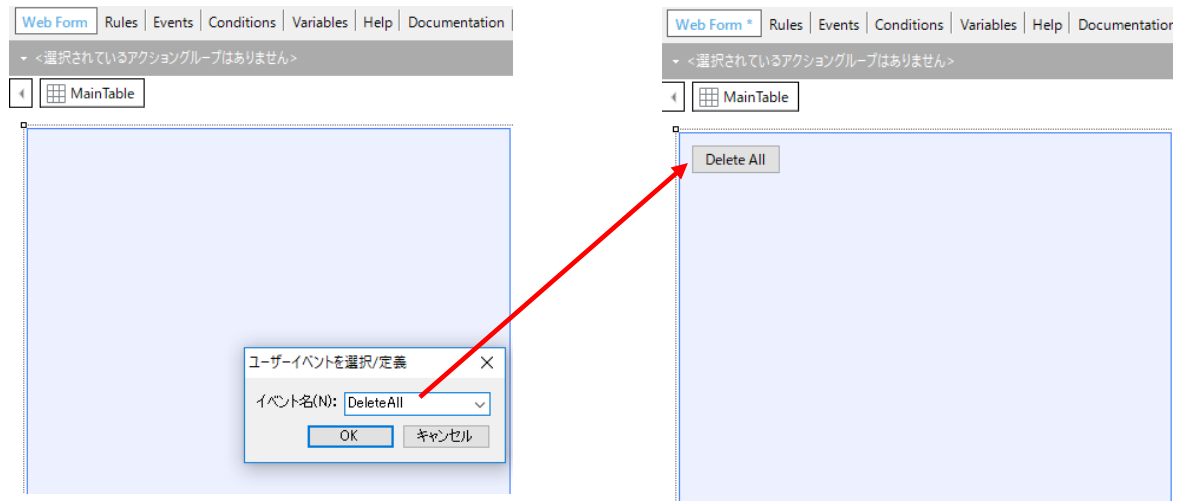
for each Country
  for each Country.City
    delete
  endfor
delete
endfor
```

プロシージャはデータの一貫性を検証しませんが、データベースは一貫性を検証することに留意してください。つまり、データベースは参照整合性を検証するため、データを削除する際は順序が重要になります。たとえば、観光名所より先に国を削除しようとしても、データベースに許可されません。これはユーザーフレンドリーではありません。

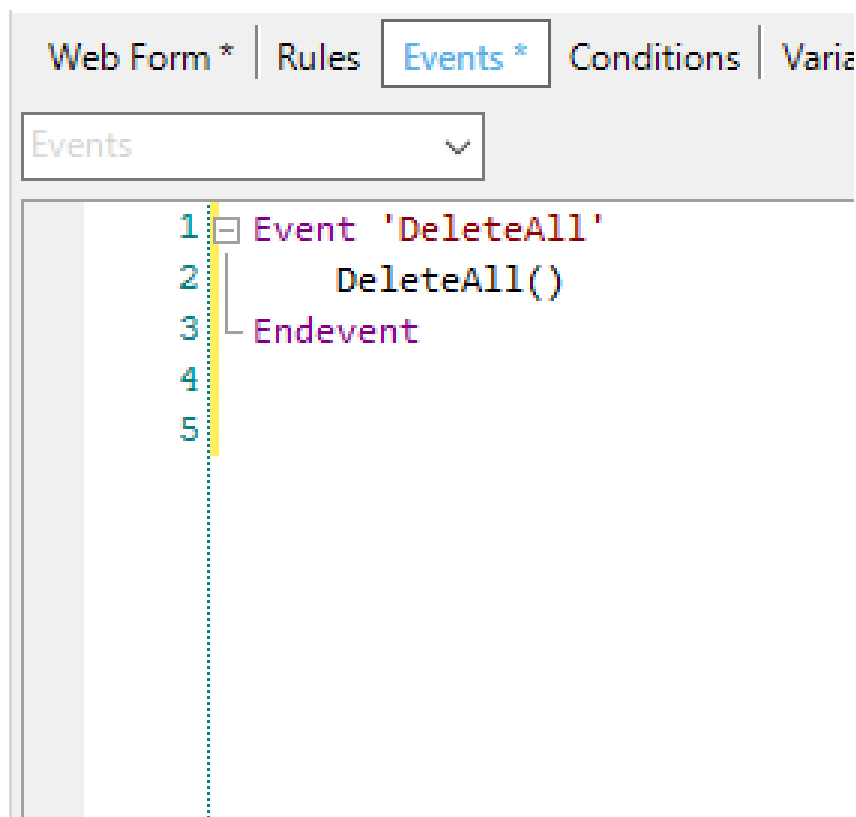
Web パネルを作成し、「ユーザーイベント」に関連付けて、プロシーチャーを呼び出します。

名前 : WPDeleteAll デスクリプション : 全データ削除

ボタンコントロールを複数配置する場合、どのボタンも Enter イベントに紐づけるとどのボタンも、Enter イベントを処理することになるので、すでに Enter イベントが使われている場合は、別名の新しいユーザーイベントを定義するようにします。



次に、ボタンにイベントに関連付けたら、ボタンを右クリックして [イベントへ移動] を選択し、イベント内にプロシーチャーの呼び出しを定義します。



13. Web パネル

13-1. 項目属性選択のみによる(ベーステーブルあり)国の一覧画面

旅行代理店から、国を国名でフィルタできる対話形式の画面をリクエストされました。国番号と国名を表示する必要があります。

13-1-1. Web パネルの定義

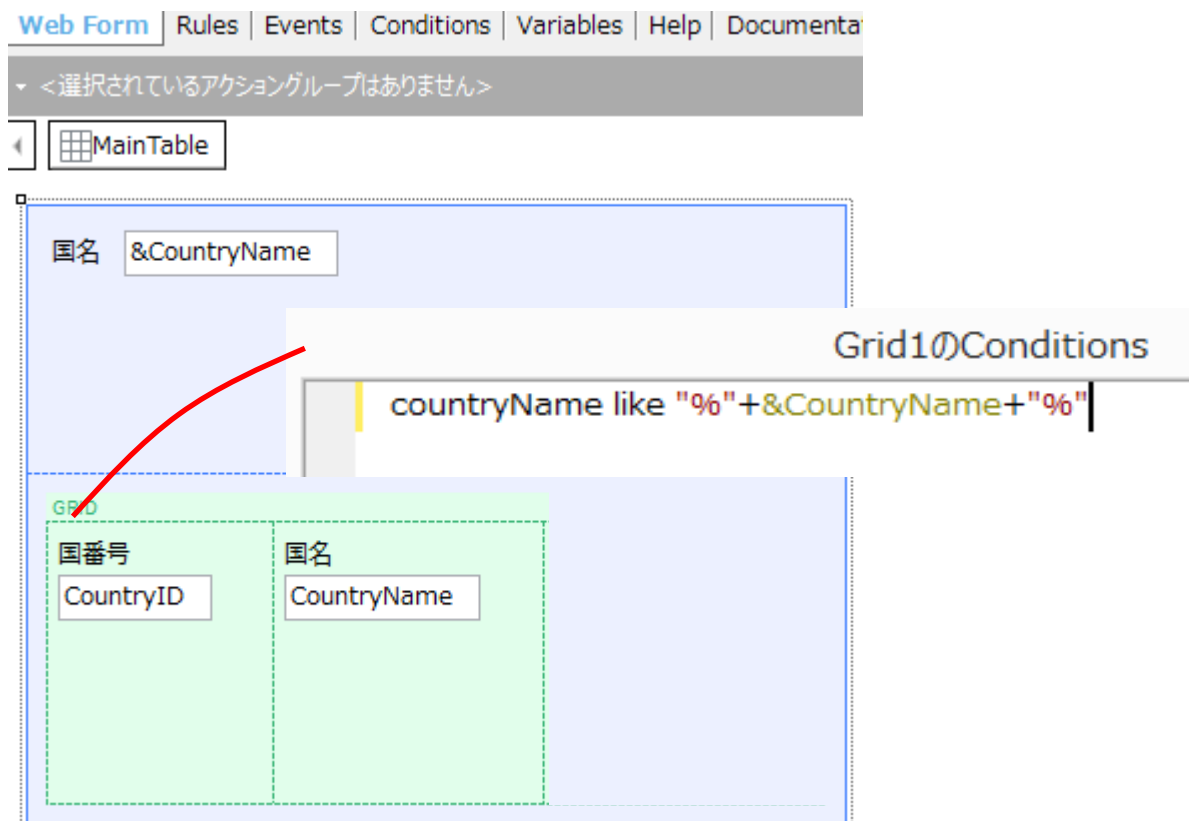
データの一覧表示を行うための Web パネルを次のように定義します。

名前: WPCountryList デスクリプション: 国リスト

13-1-2. 入力領域、グリッドの配置

- 変数 (&CountryName) を定義します。

ベーストランザクションと、グリッドレベルに必要な条件を宣言します。あいまい検索には like 演算子を使います。



13-1-3. 実行結果の確認

実行し、結果を確認します。F5 キーを押します。

13-2. 国別の観光名所数の表示

旅行代理店とのレビューの結果、さらに「国別の観光名所数」を表示したいというリクエストがありました。国別の観光名所数はデータベース上に項目としては定義されていません。どのような実装が必要でしょうか。

さきほど作成した WEB パネルの Grid コントロールには、表示する列として項目属性だけではなく、変数を指定することが可能です。したがってデータベース上にない項目は変数を使って表示させます。

13-2-1. 変数の宣言

AttractionQty 変数を宣言し、グリッド上に配置します。

13-2-2. グリッド出力時の定義

[Events] エlementでグリッドの各行に表示する観光名所数の取得を定義します。

留意点:

- [Grid] コントロールの各行が出力されるたびに実行する処理がある場合、[Events] Elementで以下のイベントを記述する必要があります。

Event Load

<処理>

Endevent

13-2-3. 実行結果の確認

実行し、結果を確認します。F5 キーを押します。

13-3. 検討課題 プログラムによる(ベーステーブルなし)国一覧画面 (オプション)

学習目的のため「ベーステーブルなし」で国の一覧画面を作成します。ここでは、国名の一覧の表示のみを行います。

13-3-1. Web パネルの定義

データの一覧表示を行うための Web パネルを次のように定義します。

名前: WPCountryListWithoutBasetable デスクリプション: 国リスト (ベーステーブルなし)

13-3-2. 変数の宣言とフォームの配置

ベーステーブルなしの WEB パネルでは、変数を使いデータソース (DB, CSV, エクセル) の値を表示します。

表示したい値の変数を宣言します。

13-3-3. グリッド出力時の定義

[Events] エLEMENTでグリッドの各行に表示する旅行数の取得を定義します。

The screenshot displays the GeneXus IDE interface. On the left, the 'Web Form' tab is active, showing a 'MainTable' grid. Inside the grid, a label '国名' (Country Name) is associated with the variable '&CountryName'. On the right, the 'Load' event script is visible, showing a 'For each' loop that iterates over the data source, assigning the value of '&CountryName' to the variable 'countryName', and then loading the data into the grid.

```
1 EventLoad
2   For each
3     &CountryName =countryName
4     load
5   endfor
6 Endevent
```

13-3-4. 動作確認

この課題では、変数のみ配置された Web パネルでの Load イベントにおける Load コマンドの挙動を深く理解することが求められています。Load イベントは何度発生し、Load コマンドは、どういう役割を果たしましたか？

13-4.すべての国、すべての観光名所

旅行代理店から、すべての国と、それぞれの観光名所（名前、写真、都市、およびカテゴリ）を表示する画面(WPMultiGrid)をリクエストされました。

Application Name							
by GeneXus							
Recents							
ブラジル				フランス			
観光名所画像	観光名所名	カテゴリ名	都市名	観光名所画像	観光名所名	カテゴリ名	都市名
	コルコバードのキリスト像	モニュメント	リオデジャネイロ		ルーブル美術館	美術館	パリ
					エッフェル塔	モニュメント	パリ
					凱旋門	モニュメント	パリ
中国				アメリカ合衆国			
観光名所画像	観光名所名	カテゴリ名	都市名	観光名所画像	観光名所名	カテゴリ名	都市名
	万里の長城	モニュメント	北京		スミソニアン博物館	美術館	ワシントン
日本							
観光名所画像	観光名所名	カテゴリ名	都市名				
	スカイツリー	有名なランドマーク	東京				

ヒント 1: フリー スタイル グリッドを使用すると、さらに柔軟な方法でコントロールを調整できます。[Columns] プロパティを設定することで、表示する列の数を指定できます。

ヒント 2: 国名にはラベル（国名）がありません。[Label Position]プロパティを[None]にします。

13-5 WEB パネルのタイプ

13-5-1. オブジェクトの特定

KB エクスプローラーを開き、フィルタで「RWD Master Page」を検索します。これは、ナレッジベースにあるすべての GeneXus オブジェクトに関連付けられているマスターページです。

13-5-2. タイトルの表示

これには次の 3 つのセクションがあります：

- ・アプリケーション名とロゴを含むヘッダー。
- ・最近使用したリンクのメニューを提供する Web Component オブジェクト
- ・アプリケーションのオブジェクトを表示するコンテンツプレースホルダー

ヘッダーで ApplicationHeader コントロールを選択し、[Caption] プロパティをアプリケーション名 (旅行代理店システム) に変更します。

The screenshot shows the GeneXus IDE interface. On the left, the 'Web Layout' editor displays a design with a header section containing the text '旅行代理店システム' (Travel Agency System) and a logo icon. Below the header are sections labeled '<Component>' and '<ContentPlaceholder>'. On the right, the 'プロパティ' (Properties) window is open for the selected 'textblock: ApplicationHeader' control. The 'Caption' property is highlighted with a red box and set to '旅行代理店システム'. Other visible properties include 'Control Name' (ApplicationHeader), 'On Click Event', 'Return On Click' (False), 'Class' (TextBlockHeader), 'Format' (Text), and 'Tooltip Text'.

textblock: ApplicationHeader	
Control Name	ApplicationHeader
Caption	旅行代理店システム
On Click Event	
Return On Click	False
Appearance	
Class	TextBlockHeader
Format	Text
Tooltip Text	
Cell information	

13-5-3. ロゴの変更

Image1 コントロールを選択し、[Image] プロパティで企業のロゴとして使用する画像を選択します (ISO.png)。[Class] プロパティが PoweredBy になっていることを確認します。

Web Layout * Rules Events Conditions Variables Help Documentation

<選択されているアクショングループはありません>

MainTable Header Image1

旅行代理店システム

<Component>

WRecentLinks

<ContentPlaceHolder>

プロパティ

全般 クラス

フィルタ

image: Image1

Control Name	Image1
Image	ISO
Low Resolution Image	(none)
Return On Click	False
On Click Event	

Appearance

Class	PoweredBy
-------	-----------

ヘッダーコントロールを選択し、[Class] プロパティが ContainerFluid と HeaderContainer になっていることを確認します。

MainTable Header ApplicationHeader

旅行代理店システム

<Component>

<ContentPlaceHolder>

プロパティ

全般 クラス

フィルタ

レスポンステーブル: Header

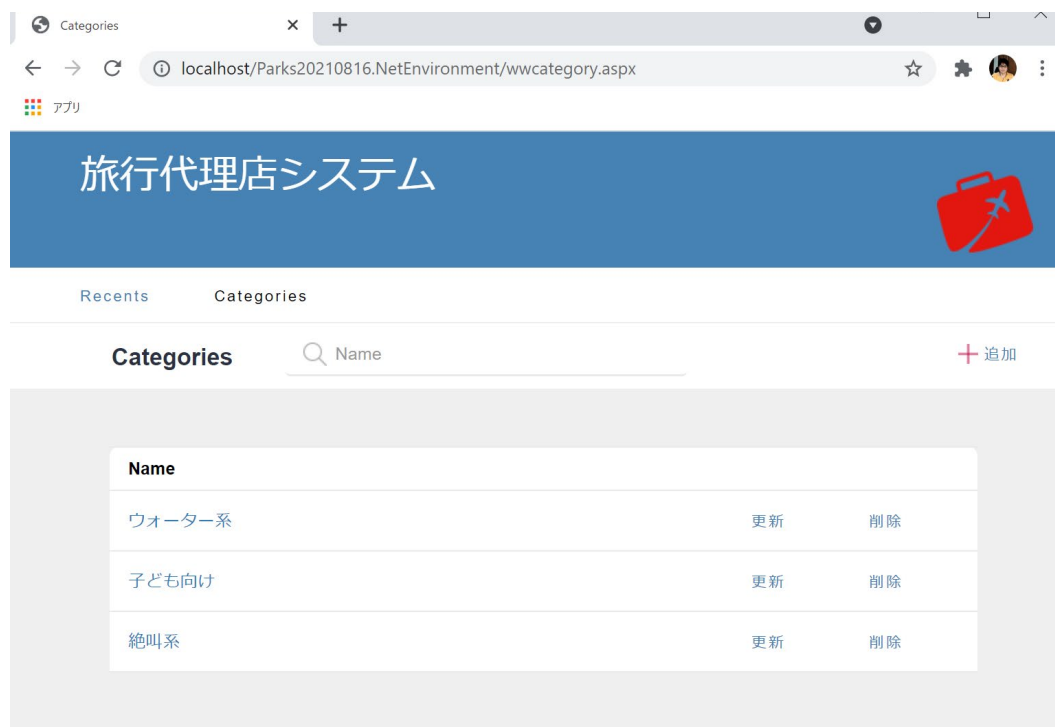
Control Name	Header
Responsive Sizes	[{"scale": "sm", "rows": [{"width": 83}, {"width": 17}]]
Row Heights	
Tooltip Text	
Header	

Appearance

Class	ContainerFluid HeaderContainer
-------	--------------------------------

13-5-3. ベースカラーの変更

次に、アプリケーションの色を編集します。KB エクスプローラーで、[カスタマイズ] ノード-> [テーマ] -> [Carmine] に移動し、ダブルクリックして [Table] 内の HeaderContainer を選択します。[Background Color] プロパティに、ヘッダーのベースカラーが選択されていることに注目してください。これを好みの色（SteelBule 等）に変更し、アプリケーションのデザインをもっと良くしていきましょう。



14. デザインシステム

これまでに実装した国番号と国名、観光名所数を表示する国に関する対話形式の画面（WPCountryList）のブラッシュアップをリクエストされました。この Web パネルは、Work With for Web のグリッドに表示されるものと同じ形式（見た目）である必要があります。

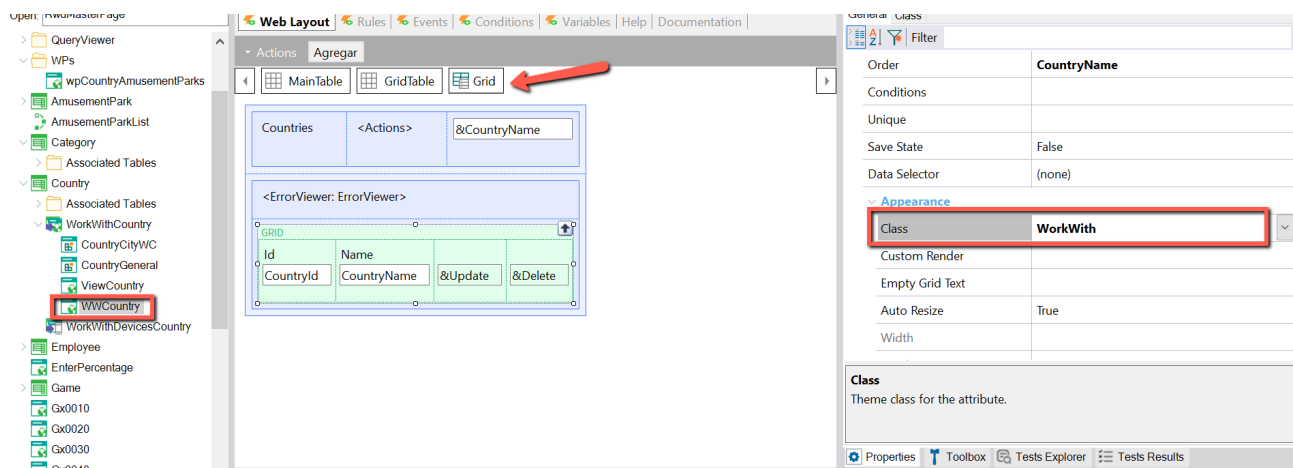
画面が小さいときの例：



（幅が大きいときは観光名所数が表示されます）

14-1. Workwith パターンの実装内容の分析に基づく、手動で作成した WEB パネルの変更

お手本として **WorkWith** に関連付けられている **WW** Web パネルで、グリッドの [Class] プロパティが **WorkWith** になっていることを確認します。



このお手本にそって実装するため、WPCountryList のグリッドでも [Class] プロパティを[WorkWith]にして Web パネルのグリッドコントロールに指定し、形式を適用します。

The screenshot shows the GeneXus IDE interface. On the left, the 'Web Layout *' tab is active, displaying a design with a 'MainTable' and a 'Grid1' control. The 'Grid1' control is highlighted with a green border and contains three columns: '国番号' (CountryID), '国名' (CountryName), and 'Attraction Qty' (&AttractionQty). Above the grid, there is a label '国名' (&CountryName). On the right, the 'プロパティ' (Properties) panel is open, showing the 'クラス' (Class) tab. The 'Grid1' control is selected, and its properties are listed in a table.

クラス	
Control Name	Grid1
Collection	
Base Trn	
Order	
Conditions	CountryName like "%" + &C
Unique	
Save State	False
Data Selector	(none)
Appearance	
Class	WorkWith
Custom Render	
Empty Grid Text	

デザインを揃えるため、WorkWith のグリッドの各列には、関連するクラスを指定することに留意してください：

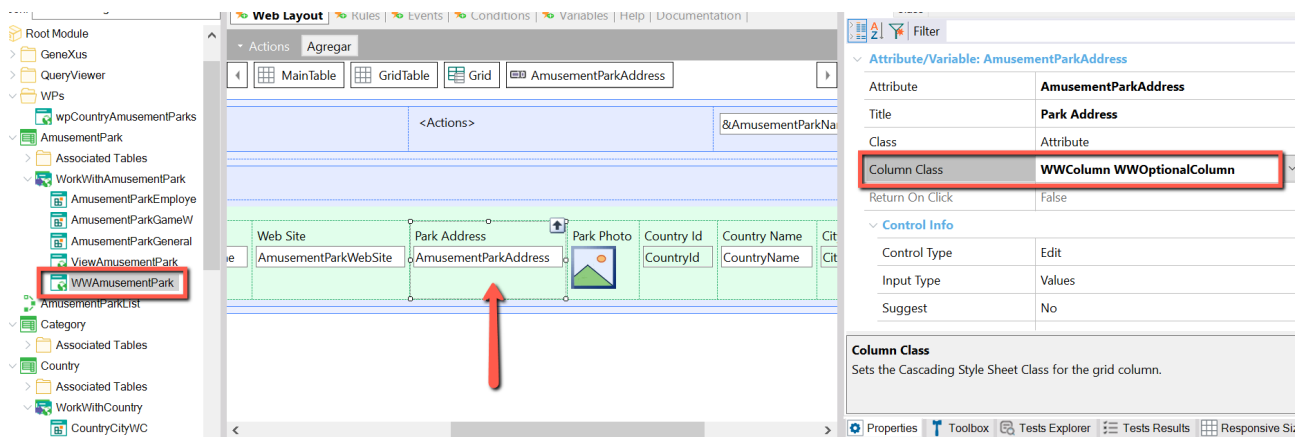
WWColumn は、画面のサイズに関係なく常に表示する、重要な項目属性に使用します。

WWColumn WWOptionalColumn は、画面が小さくなると非表示になる項目属性に使用します。

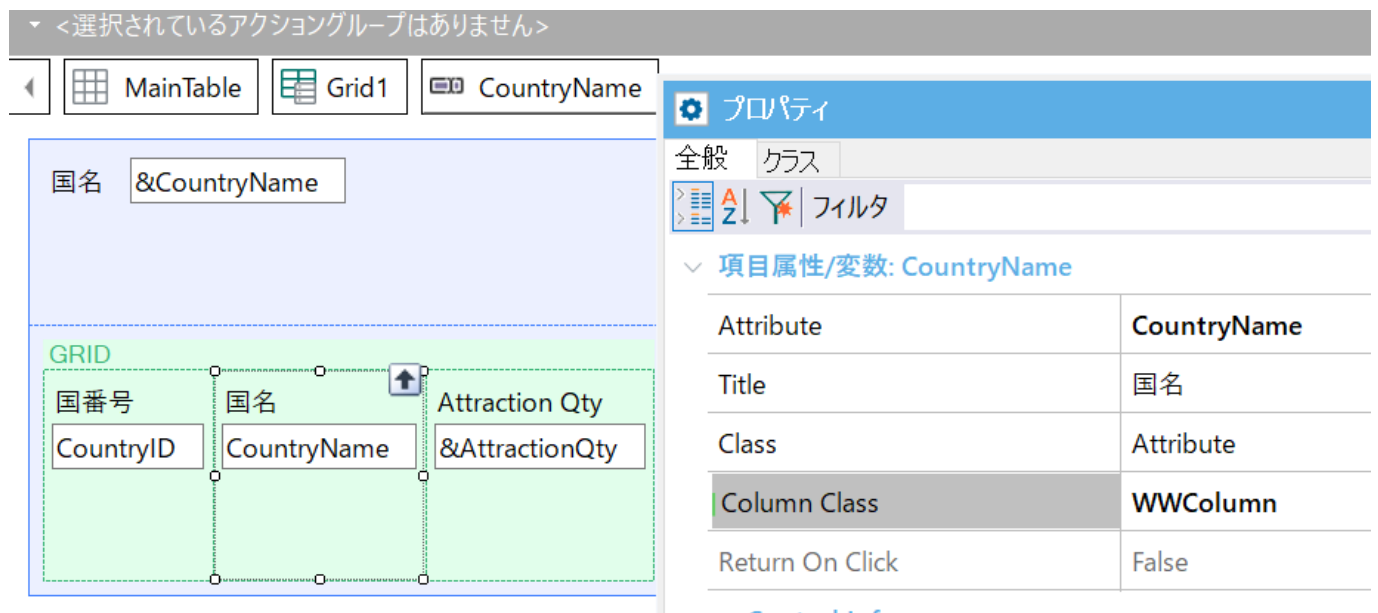
Workwith パターンで実装したオブジェクトの分析例：

The screenshot shows the GeneXus IDE interface. On the left, the 'Web Layout' tab is active, displaying a design with a 'MainTable' and a 'GridTable' control. The 'GridTable' control is highlighted with a green border and contains five columns: 'AmusementParkName', 'Web Site', 'Park Address', 'AmusementParkAddress', and 'Park Photo'. A red arrow points to the 'AmusementParkName' column. On the right, the 'プロパティ' (Properties) panel is open, showing the 'Attribute/Variable: AmusementParkName' tab. The 'Column Class' property is highlighted with a red box and set to 'WWColumn'.

Attribute/Variable: AmusementParkName	
Attribute	AmusementParkName
Title	Park Name
Class	DescriptionAttribute
Column Class	WWColumn
Return On Click	False
Control Info	
Control Type	Edit
Input Type	Values
Suggest	No
Column Class	
Sets the Cascading Style Sheet Class for the grid column.	

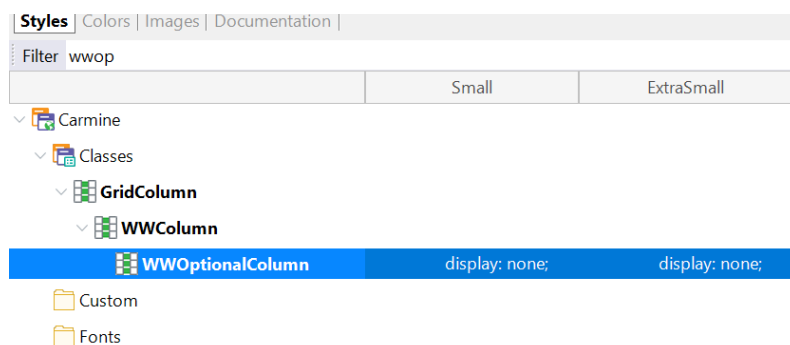


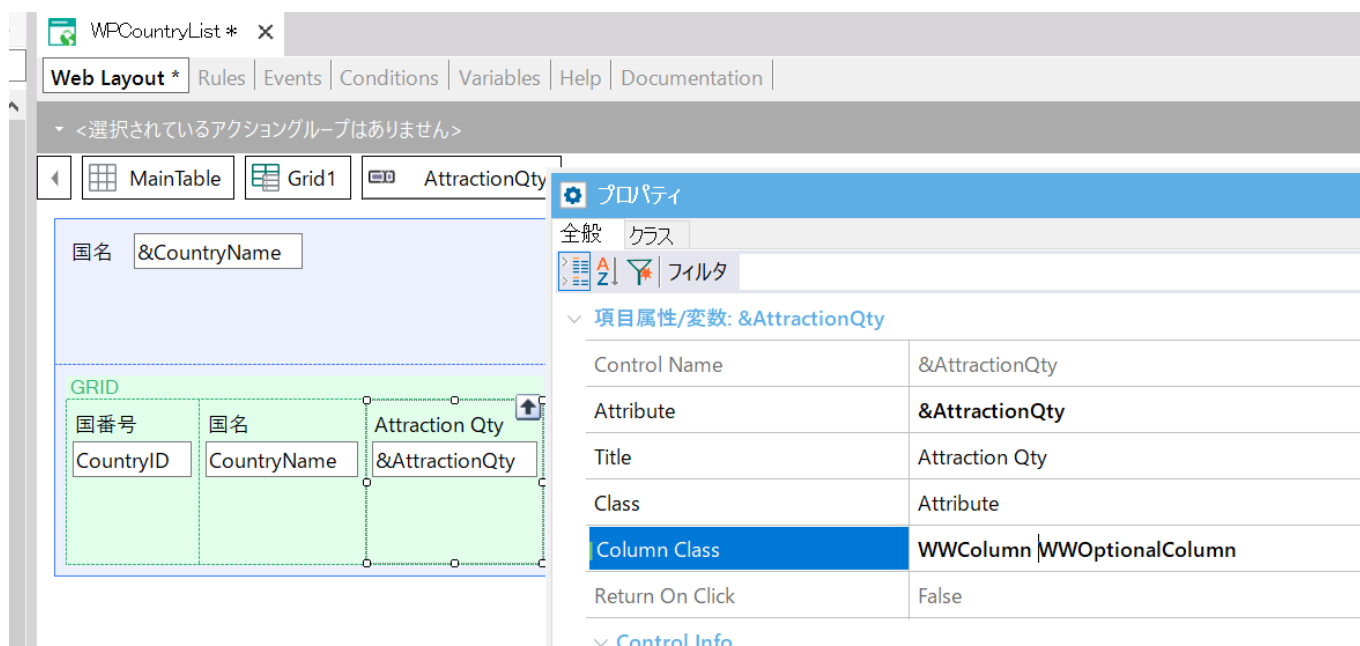
WPCountryList のグリッドで、この動作を実現するには、国番号、国名で、[Column Class] プロパティに **WWColumn** と指定します。



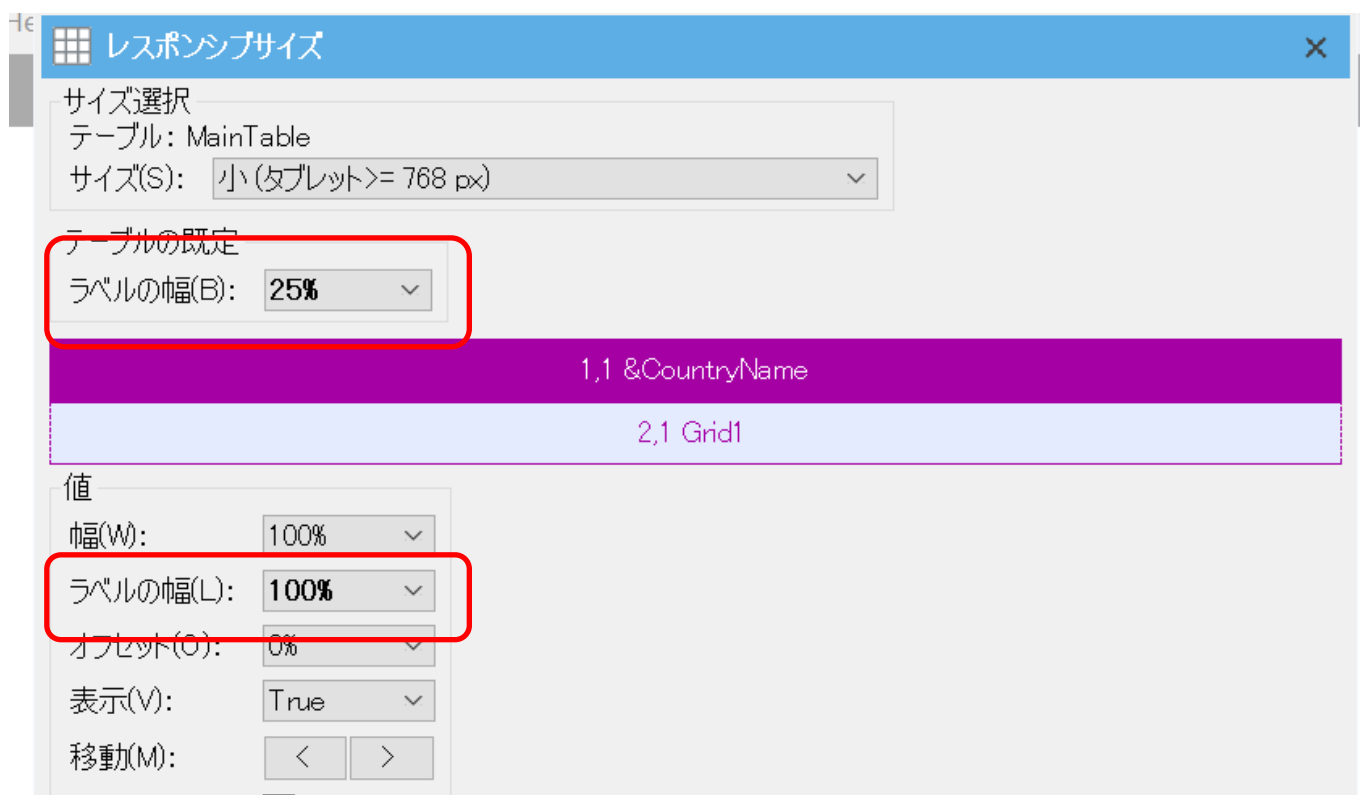
観光名所数の列を画面が狭いときに、非表示にするにはどうすればいいでしょうか。

クラスの **WWOptionalColumn** を確認します。小 (Small) および極小 (ExtraSmall) の列を見ると、これらの画面サイズの場合、そのクラスに割り当てられている項目属性は表示されないことが分かります。これを指定してください。





最後に、フィルタ条件の入力領域のラベルを上部に表示するため、レスポンシブサイズのボックスを開きます。表示されない場合は、Web パネルでメインテーブルを選択し、[Responsive Sizes] プロパティをクリックします。変数を選択し、レスポンシブサイズとして小 (Small) を選択し、[Label Width] を下図のように指定します。サイズを極小 (Extra Small) に変更して、[Label Width] を同様に設定します。



14-2. 動作確認

アプリケーションを実行し、動作をテストします。

グリッドの Infinite scrolling ページング

前の画面 (WPCountryList) で国を見やすく表示するには、グリッドのページングが必要です。

そのために、まず国のテストレコードを増やします。

Web パネルのグリッドのプロパティで、[Rows] を 5 に変更し、[Paging] が **One page at a time** になっていることを確認します。

Height	
Rows	5
Header	
Tooltip Text	
▼ PagingMode	
Paging	One page at a time

実行して、1 ページにつき 5 個のレコードが表示されることを確認します。また、グリッド上の矢印のナビゲーションを使用して、国の表示を移動できることを確認します。

データを複数のページに分けて表示するのではなく、すべてのコンテンツを単一の画面に含めて、スクロールしてレコードにアクセスするにはどうしたらいいでしょうか。Web パネルに戻り、[Paging] プロパティを **Infinite scrolling** に変更します。

アプリケーションを実行し、レコードをスクロールできることを確認してください。

Appendix1 基本スタイルとユーザーコントロール (オプション)

次のようなデザインで観光名所を一覧表示する Web パネルが必要だとします。



ソリューション:

顧客が希望するデザインのガイドラインに沿って観光名所のデータを表示できるように、ユーザー コントロール タイプのオブジェクトを作成します。

留意点:

生成するユーザーコントロールのスタイルを確立するには、ベースライブラリが必要です。

基本スタイルとは、デザイナーや使用するデザインシステムから提供された CSS、JS、アセットなどがすべて含まれる ZIP ファイルのことです。

基本スタイルを作成する:

ライブラリがない場合、[セマンティック UI のページ](https://semantic-ui.com/introduction/getting-started.html)¹の [Download Zip] ボタンをクリックして、必要なファイルをダウンロードしてください。

¹ <https://semantic-ui.com/introduction/getting-started.html>

Simpler Setup

If you are using Semantic UI as a dependency and just want to use our default theme, use our lighter `semantic-ui-css` or `semantic-ui-less` repo. If you just need the files you can download them as a zip from GitHub.

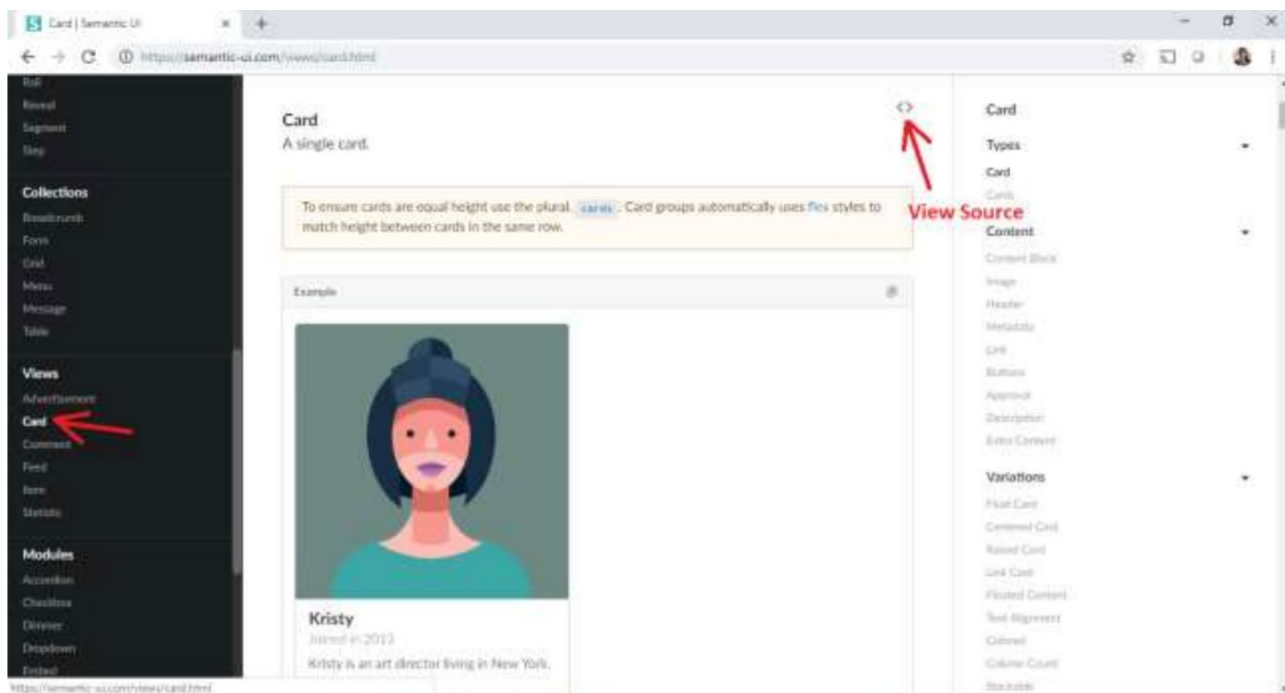
[Download Zip](#)[View all Semantic-Org Repos](#)

.zip ファイルを **gxlibrary** という拡張子で保存します。

SemanticUI_gxlibrary というファイルタイプのオブジェクトを作成し、**ダウンロードしたファイル**をインポートします。

ユーザーコントロールを作成する:

セマンティック UI フレームワークの「カード」を利用してユーザーコントロールを作成します。セマンティック UI のサイトで「Card」を検索すると、さまざまなタイプのコントロールがあることが分かります。たとえば、一番シンプルなカードを選択して HTML コードを表示し、それをコピーすることができます:



ここでは、開発スピードを上げるため、使用するコードを提供します。

ユーザー コントロール タイプの新しいオブジェクト CardSemantic を作成します。

[Screen Template] エlementに次のコードをコピーします:

```
<div class="ui link cards">
  <div class="card">
    <div class="image">
      
    </div>
    <div class="content">
      <div class="header">{{Name}}</div>
      <div class="meta">
        <a>{{AddedDate}}</a>
      </div>
      <div class="description">
        {{description}}
      </div>
    </div>
    <div class="extra content">
      <span class="right floated">
        {{Country}}
      </span>
      <span>
        <i class="user icon"></i>
        {{AttractionID}}
      </span>
    </div>
  </div>
</div>
```

表示したい内容に応じて、プロパティがパラメーター化されています。

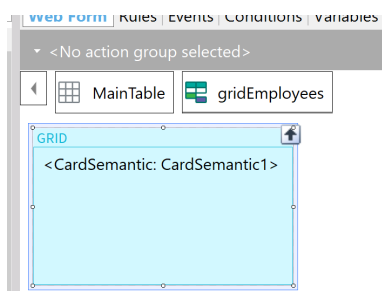
ここでは、次のように指定します:

Base Control Type	None
Base CSS	SemanticUI
Qualified Name	
Object Visibility	SemanticUI

User Control オブジェクトの [Base CSS] プロパティが SemanticUI に設定されていることを確認します。

AttractionDetail という名前の Web パネルを作成します。

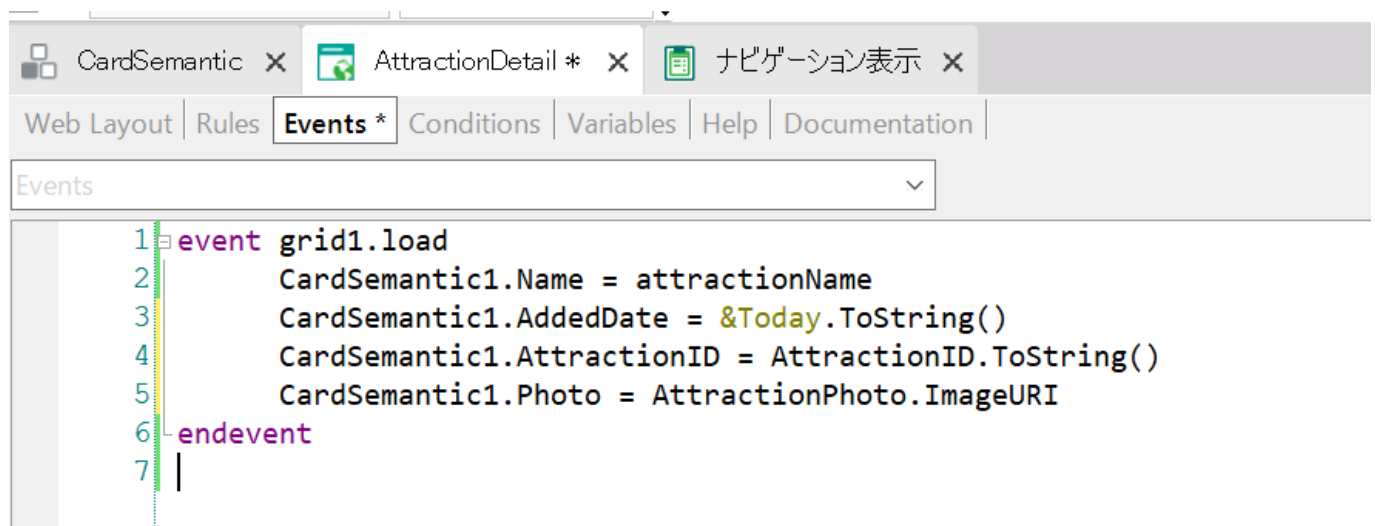
フリー スタイル グリッドを追加し、作成したユーザーコントロールを挿入します。



▼ Columns	4
Extra Small	4
Small	4
Medium	4
Large	4
Fmmtv Grid Text	

フリースタイルグリッドで、4列表示を指定します。

これらの項目属性に基づいて、ユーザーコントロールの各プロパティを初期化します。



```
1 event grid1.load
2     CardSemantic1.Name = attractionName
3     CardSemantic1.AddedDate = &Today.ToString()
4     CardSemantic1.AttractionID = AttractionID.ToString()
5     CardSemantic1.Photo = AttractionPhoto.ImageURI
6 endevent
7 |
```

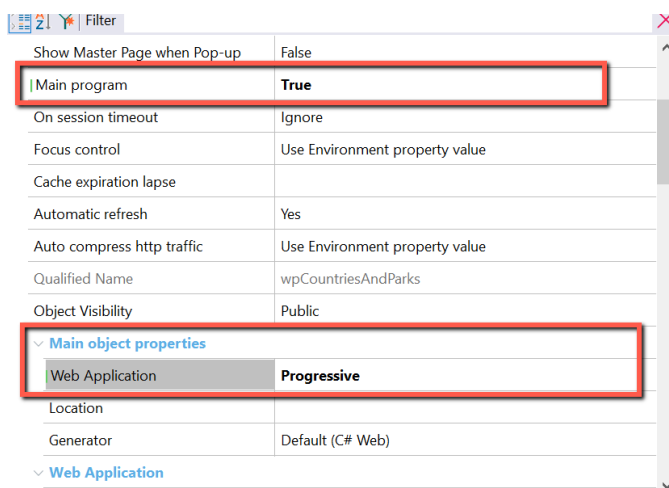
実行します。

Appendix2 Progressive Web Applications (PWA) (オプション)

準備:

LogoForPWA.xpz をインポートし、LogoForPWA 画像を KB に認識させます。

「13-4. すべての国、すべての観光名所」で作成した WPMultiGrid パネルのプロパティで、[**Main Program**] を **True** に変更します。[**Main object properties**] という新しいプロパティグループがアクティブになっていることに注目してください。このブロックで、[**Web Application**] プロパティを **Progressive** に変更します。

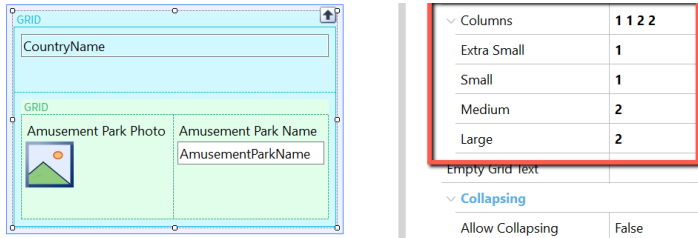


このプロパティの値を Default から Progressive に変更すると、[Web Application] という新しいプロパティグループが有効になります。これを次のように設定します:

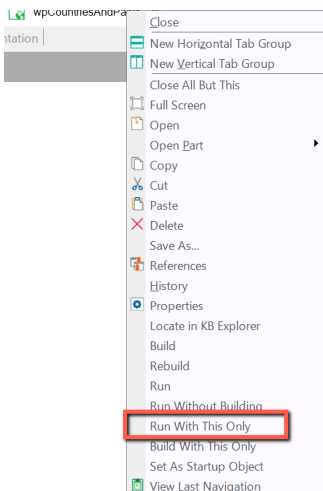
Web Application

Web Application Name	観光名所検索アプリ
Web Application Short Name	AP
Web Application Description	観光名所検索できます
Primary Text Direction	Auto
Display	Fullscreen
Background Color	<input type="color" value="#FFF"/> #FFF
Theme Color	<input type="color" value="#4682B4"/> #4682B4
Prefer Related Applications	False
Web Application Icon	LogoForPWA ...
Android Alternative App	

フリー スタイル グリッドのプロパティで、列のサイズを次のように変更します:



Web パネルのタブを右クリックして **[実行]** を選択します:



ブラウザーで、アプリケーションをインストールするためのオプションをクリックし、デスクトップに作成されたショートカットに移動します。



ダブルクリックしてアプリケーションを開きます。色、アイコン、画面サイズが設定内容と一致していることを確認します。

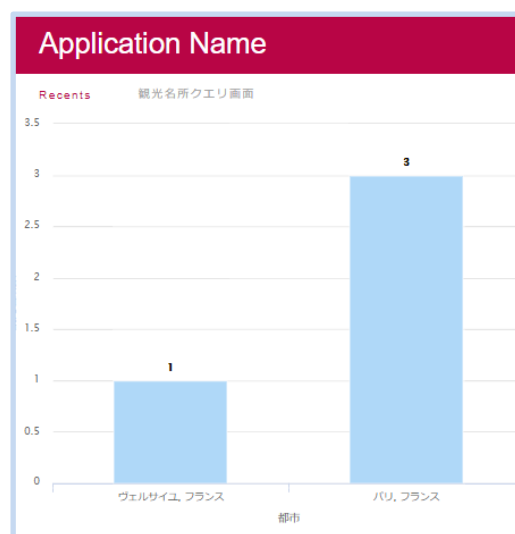


Appendix3 Query オブジェクト（オプション）

特定国内の都市のみを都市名順で取得し、各都市の観光名所の数を棒グラフで表示したいという要望があります。

これを実装するために Query オブジェクトを定義します。

Web パネルを定義し、QueryViewer コントロールを使用して、このクエリをグラフとして表示してください。



A3-1. クエリの定義

国ごとの都市を取得し、都市ごとの観光名所の数を取得できるクエリを次のように定義します。

名前: AttractionsQuery デスクリプション: 観光名所クエリ

A3-2. 取得項目の定義

[Structure] エlementに国名、都市名、観光名所の数を取得し、特定の国（例：フランス）でフィルタリングできるように定義をします。

留意点:

- [Attributes] ノードに取得する項目や関数を定義する必要があります。
- [Filters] ノードにフィルタリングの条件を定義する必要があります。

A3-3. Web パネルの定義

クエリにより取得した結果を棒グラフ形式で表示するための Web パネルを次のように定義します。

名前: WebQuery デスクリプション: 観光名所クエリ画面

A3-4. 画面の定義

[Web Layout] Elementで Query オブジェクトを表示するためのコントロールを定義します。

留意点:

- Query オブジェクトを表示するためには [QueryViewer] コントロールを配置する必要があります。
- コントロール上に表示するオブジェクトを指定するためには [Object] プロパティを変更する必要があります。
- 表示形式を指定するためには [Type] プロパティを変更する必要があります。
グラフ形式で表示するためには [Type] プロパティが [Chart] となる必要があり、
グラフの種類を指定するためには [Chart Type] プロパティを変更する必要があります。

A3-5. 実行結果の確認

実行し、結果を確認します。F5 キーを押します。

Appendix4 Web サービス (オプション)

旅行代理店から、すべての国のリストを顧客に表示し、各国の首都、通貨、および国旗も表示できる新しい PDF 出力機能の要望がありました。ただし、首都、通貨、国旗はデータベース上に登録していない情報です。どのような方法があるでしょうか。

国情報			
名前	首都	通貨	国旗
ブラジル	Brasilia	Brazil Real	
フランス	Paris	Euro	
中国	Beijing	Yuan Ren minbi	
アメリカ合衆国	Washington	Dollars	
日本	Tokyo	Yen	

既存の Web サービスを利用し、これらの情報を取得します。

今回、利用する Web サービスは CountryInfoService という名前であり、以下の URL で公開されています。

<http://www.oorsprong.org/websamples.countryinfo/CountryInfoService.wso>

A4-1. Web サービスのインポート

外部 Web サービスをこのアプリケーションから利用できるようにインポートを行います。

留意点:

- Web サービスのインポートは、[ツール] → [アプリケーションの統合] → [WSDL インポート] と選択し、表示されるウィザードで実行します。
- Web サービスを取り込むと外部オブジェクトと必要に応じて構造化データが生成されます

A4-2. トランザクションの修正

利用する Web サービスでは国の ISO コードを受け取ることで該当する国の情報を出力するため、「Country」トランザクションに以下の項目を追加します。

名前	タイプ	デスクリプション
▪ CountryISO	VarChar(2)	国 ISO コード

A4-3. プロシーチャーの定義

帳票出力を行うためのプロシーチャーを次のように定義します。

名前 : CountriesInfo デスクリプション : 国情報

帳票出力に必要なプロパティの設定とルールの定義を行います。

A4-4. レイアウトの定義

[Layout] エLEMENTでレポートのタイトルテキスト、各出力項目のタイトルテキスト、各値 (データベースから取得した値と Web サービスで取得した値) を表示するそれぞれのプリントブロックを定義します。

留意点:

- Web サービスで取得した値を表示するためには変数を利用します。

A4-5. 出力処理の定義

[Source] エLEMENTでレポートを出力するための処理を定義します。外部オブジェクトタイプの変数を定義すると、Web サービスが提供するメソッドを実行できるようになることに注意してください。

留意点:

- Web サービスの外部オブジェクトを利用するためには、これをデータタイプとする変数が必要となります。

A4-6. 実行結果の確認

実行し、結果を確認します。まず「国 ISO コード」を登録する必要があるため、**F5** キーを押します。**再編成**が発生します。

いくつかの国に ISO コードを入力したらプロシーチャーオブジェクトを実行して結果を確認します。オブジェクトタブを右クリックして [これだけを実行] を選択します。

Appendix5 スマートデバイスに必要なセクション (オプション)

旅行代理店は、エンドユーザーによって使用されるスマートデバイス向けの小規模なアプリケーションの提供も考えているとします。

その目的は、スマートデバイスを使用することにより、旅行代理店が旅行を提供するすべての都市や、利用可能なフライト、主要な観光名所について誰でも調べることができるようにすることです。

まずここでは国の一覧表示と管理をスマートデバイス向けに実装します。

A5-1. パターンの適用

「Country」へ [Work With for Smart Devices] を適用します。

項番 5 で実施したようにトランザクションの「Patterns」エレメントにて [Work With for Smart Devices] を選択し、「保存時にこのパターンを適用」チェックボックスにチェックを入れ、保存します。

A5-2. Menu の定義

スマートデバイス向けアプリケーションの起動画面となる Menu を次のように定義します。

名前: Menu デスクリプション: メニュー

A5-3. アクションの追加

[ダッシュボード] エレメントの [Items] ノードに機能呼び出すアクションを追加します。

留意点:

- アクションを追加するためには [Items] ノードを右クリックし、[追加] を選択し、[Action] を選択。表示されたダイアログで [Work With for Smart Devices] タイプのオブジェクトを選択します。

A5-4. 実行結果の確認

Menu を実行し、結果を確認します。

留意点:

- スマートデバイス向けアプリケーションの機能を追加したため、Android エミュレーターが起動されますが、今までと同じように Web アプリケーションも実行可能です。なぜでしょうか。

Appendix6 GXserver (オプション)

ここまで作成したアプリケーションを複数メンバーで開発することになりました。複数のメンバーで 1 つのナレッジベースを共有するためにはどのような方法があるでしょうか。

GXserver を利用します。

A6-1. ナレッジベースの送信

ナレッジベースを共有するために GXserver へナレッジベースの送信を行います。

ここでは GeneXus 社が一般公開している次の GXserver を利用します：

<https://sandbox.genexusserver.com/v17>

留意点:

- サーバーへの送信を行うためには、[ファイル] → [GXserver にナレッジベースを送信] を選択し、表示されたダイアログの URL に GXserver の URL を入力し、「ユーザー名」、「パスワード」にご自身の GeneXus Account を入力してください。もし、アカウントを保有していない場合、上記 URL へブラウザよりアクセスいただき、「Create one now」のリンクより新規アカウントの作成を行います。

A6-2. 変更のコミット

ナレッジベースに新規オブジェクトの作成や、既存オブジェクトの変更、削除などを行います。この変更結果を共有するためにサーバーへコミットします。

留意点:

- GXserver で共有されているナレッジベースの場合、[ナレッジマネージャ] → [チーム開発] と選択することでサーバーとのやり取りを行うためのウィンドウを表示することが可能です。
- コミットを行う場合、[チーム開発] ウィンドウ内の [サーバーにコミット] タブを使用する必要があります。

A6-3. Web コンソールでの確認

GXserver は Web アプリケーションとして展開されるため、ブラウザで URL を指定することで Web コンソールを利用することが可能です。この Web コンソールからナレッジベースの最新状況を確認します。

Appendix7 観光名所を予約するためのビジネス プロセス モデル (BPM) の定義 (オプション)

観光名所の予約プロセスは、1 つのビジネスプロセスと考えることができる一連のタスクで構成されています。

このプロセスが開始されるのは、旅行代理店を訪れた人が特定の観光名所のパッケージツアーを予約するときです。旅行代理店で最初にしなければならないことは、そのツアーの予約の入力です。

予約を入力した後は、ツアーの申し込み者が旅行代理店の顧客かどうかを確認し、顧客でない場合は、顧客として入力して、その予約と関連付ける必要があります。

これらの手順を完了した後は、旅行をする人数分の空きがあるかを確認する必要があります。







座席を確保できる場合は、予約の確定を行い、このビジネスプロセスは終了します。確保できない場合は、顧客に別のツアーを提供するため、再度ツアーの予約から始める必要があります。

この要望を満たす機能の実装を行います。

A7-1. トランザクションの定義

予約を管理するための画面とテーブルが必要となるので、トランザクションを次のように定義します。CustomerId は「Null 許容」列の値を「Yes」に変更します。(予約時に顧客 ID が存在しない場合のため)

名前: Reservation デスクリプション: 予約

名前	タイプ	デスクリプション
 ReservationId	Id	予約番号
 ReservationDate	Date	予約日
 ReservationQty	Numeric(4.0)	予約人数
 CustomerId	Id	顧客番号
 CustomerName	Name	顧客名前
 ReservationAvailable	Boolean	予約可能

A7-2. ビジネスプロセスの定義

ビジネスプロセスダイアグラムダイアグラムタイプのオブジェクトを次のように定義します。

名前: AttractionReservation 名前: ツアー予約プロセス

A7-3. ダイアグラムの定義

要望通りのプロセスを実行できるように [Diagram] エlementを定義します。

留意点:

- プロセス内のタスクを定義するためには以下の 2 つの方法があります。
 - 開始や終了、条件分岐などの記号や空のタスクを配置する場合、[ツールボックス] ウィンドウから該当の記号をドラッグ&ドロップします。
 - 定義済みのオブジェクトをタスクとして配置する場合、[KB エクスプローラー] ウィンドウの [フォルダ表示] セクションから該当のオブジェクトをドラッグ&ドロップします。
- 記号と記号をコネクタで接続するためには接続元となる記号にカーソルを置き、記号の太くなった枠からドラッグを開始し、接続先の記号に接続したところでドロップします。

A7-3-1. 開始イベントの配置

[ツールボックス] ウィンドウから [開始イベント/トリガーなし] 記号をドラッグ&ドロップします。

A7-3-2. 予約タスクの配置

[KB エクスプローラー] ウィンドウから「Reservation」トランザクションをドラッグ&ドロップし、開始イベントから予約タスクへとプロセスが進むようにコネクタで接続します。

A7-3-3. 条件分岐機能の配置

予約タスクの入力で顧客が指定されたかどうかを確認し、プロセスを分岐するための記号を配置します。

[ツールボックス] ウィンドウから [排他的なゲートウェイ] をドラッグ&ドロップします。

予約タスクから条件分岐タスクへとプロセスが進むようにコネクタで接続します。

A7-3-4. 顧客タスクの配置

[KB エクスプローラー] ウィンドウから「Customer」トランザクションをドラッグ&ドロップし、条件分岐タスクから顧客タスクへと特定条件を満たしたとき、プロセスが進むようにコネクタで接続します。
この場合の条件は「Reservation.CustomerId = 0」(予約の顧客番号が 0 (空) であれば) となります。

留意点:

- 条件付きコネクタ (緑色のコネクタ) にはそのコネクタをたどる条件を以下の 2 つの方法で指定することが可能です。
 - 条件付きコネクタの [Condition Rule] プロパティで指定
 - 条件付きコネクタをダブルクリックし、表示される条件エディターで指定

A7-3-5. プロシーチャーの定義

顧客タスクで追加された顧客を予約タスクに紐付ける更新処理をプロシーチャーオブジェクトで定義します。

プロシーチャーは次のように定義します。

名前: AssignCustomerToReservation デスクリプション: 顧客の予約関連付け

このオブジェクトはプロセス内の顧客番号と予約番号をパラメーターとして受け取り、受け取った予約番号でフィルタリングした「Reservation」テーブルの対象レコードに受け取った顧客番号を割り当てます。

A7-3-6. 顧客と予約の関連付けタスクの配置

[KB エクスプローラー] ウィンドウから 「AssignCustomerToReservation」 プロシーチャーをドラッグ&ドロップし、顧客タスクから関連付けタスクへとプロセスが進むようにコネクタで接続します。

A7-3-7. 予約確定タスクの配置

最初のタスクで登録された予約を確定するためのタスクを配置します。このタスクは「Reservation」トランザクションを再度使用します。このトランザクションが持つ [ReservationAvailable] 項目属性の値によって予約が確定できたかどうかを判断します。

[KB エクスプローラー] ウィンドウから「Reservation」トランザクションをドラッグ&ドロップし、項番 A7-3-2 で配置したタスクと区別するためにタスクの [Name] プロパティを [予約確定] に変更します。

条件分岐タスクから予約確定タスクへとどの条件も満たさない場合、プロセスが進むようにコネクタで接続します。

また、顧客と予約の関連付けタスクからも予約確定タスクへとプロセスが進むようにコネクタで接続します。

留意点:

- 条件付きコネクタ (緑色のコネクタ) をどの条件付きコネクタも満たさない場合にプロセスを進めるコネクタとするためには、コネクタの [Condition Type] プロパティを [Default] に変更する必要があります。

A7-3-8. 条件分岐機能の配置

予約確定タスクで確定できたかどうかを確認し、プロセスを分岐するための記号を配置します。

再び [ツールボックス] ウィンドウから [排他的なゲートウェイ] をドラッグ&ドロップします。

予約確定タスクから条件分岐タスクへとプロセスが進むようにコネクタで接続します。

A7-3-9. 終了タスクの配置

[ツールボックス] ウィンドウから [終了イベント/トリガーなし] 記号をドラッグ&ドロップします。

条件分岐タスクから終了イベントへとどの条件も満たさない場合、プロセスが進むようにコネクタで接続します。

A7-3-10. 予約が確定できない場合のコネクタ

予約確定タスクで予約が確定できなかった場合 (ReservationAvailable = False)、予約タスクへプロセスが進むようにコネクタの接続と条件の定義を行います。

A7-4. 実行結果の確認

実行し、結果を確認します。ビジネスプロセスダイアグラムダイアグラムタイプのオブジェクトタブを右クリックして [実行] を選択します。再編成が発生します。プロセスは複数のルートへ分岐するため、すべてのルートを確認します。

留意点:

- 実行したアプリケーションは GXflow クライアントによる実行となります。
- タスクを実行する場合、以下の 2 つの方法があります。
 - 表示されるタスクをダブルクリック
 - 表示されるタスクをクリック (選択) し、「実行」アクションをクリック
- オブジェクトをタスクとした場合、タスクを実行するとダイアログが表示され、ダイアログ内で処理を行う必要があります。
- タスクが完了し、先に進むためにはタスクをクリック (選択) し、「送信」アクションをクリックする必要があります。
- 新規プロセスを開始するためには「新規」アクションをクリックし、開始するプロセスを選択し、「OK」をクリックします。