演習

GeneXus コース

GeneXus™ 18

2025 年 6 月

©GeneXus.All rights reserved. 本書は、GeneXus™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.の商標または登録商標です。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。

内容

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. トランザクションの定義 7

3－1. 「EMPLOYEE」トランザクション 7

3-2. 「AMUSEMENTPARK」トランザクションと「COUNTRY」トランザクション 14

3-3. 整合性維持の確認 22

3-4. 「SHOW」トランザクション 23

3-5. 「GAME」トランザクション 29

3-6. 「CATEGORY」トランザクション 33

3-7. 「EMPLOYEE」トランザクションと「AMUSEMENTPARK」トランザクション 37

3-8. 「COUNTRY」トランザクションへの都市の追加 39

3-9. 「AMUSEMENTPARK」トランザクションへの都市の追加 42

4. トランザクションに制御を追加 45

4-1. 「EMPLOYEE」トランザクションに制御を追加 45

4-2. 「COUNTRY」トランザクションに制御を追加 50

5. データを操作するためのインターフェース改善 53

5-1. WORK WITH FOR WEB パターンの適用 53

5-2. アプリケーションの挙動確認 56

5-3. パターンのカスタマイズ 59

6. 項目属性の別名参照 63

6-1. 「TECHNICIAN」トランザクション 63

6-2. 「REPAIR」トランザクション 66

6-3. 「REPAIR」トランザクションに制御を追加 71

7. 式の追加 72

7-1. 「REPAIR」トランザクションの更新 72

7-2. 実装結果の確認 75

8. 重複登録の制御 77

8-1. ユーザーインデックスの追加 77
8-2. 実装結果の確認 80
9. レポート出力 83
9-1. アミューズメントパークの一覧 83
9-2. 「ブラジル」のアミューズメントパークの一覧 89
9-3. アトラクションの一覧 91
9-4. アトラクションのあるカテゴリのみを対象とした一覧 96
9-5. ショーの一覧 98
9-6. 国名の一覧 103
9-7. パーク数が 2 以上の国のみを対象とした一覧 108
10. パラメーターの受け渡し 110
10-1. 指定数以上のパークがある国の一覧 110
10-2. アミューズメントパークを指定したショーの一覧 116
11. ビジネスコンポーネント 123
11-1. 修理価格の値上げ 123
11-2. すべての修理を削除 128
12. プロシージャーによるレコード操作 131
12-1. トランザクションの機能による初期データ登録 131
12-2. 修理価格の値上げ 133
12-3. すべての修理を削除 136
13. WEB パネルの利用 139
13-1. 国名とアミューズメントパーク数の一覧 139
13-2. ショーの一覧 143
13-3. アミューズメントパークのランキング 149
13-4. アプリケーション全体の共通部分変更 155
14. デザインシステム 158
14-1. グリッドコントロールのクラス 158

1. 課題

本コース課題は、実際のプロジェクトにおけるお客様と開発者の会話をシミュレートし、GeneXus を利用したアプリケーションの開発を行っていきます。

プロジェクト：

さまざまな国のアミューズメントパークを管理している企業から、データを保存および処理するシステムの開発を依頼されました。バックエンドの要件は以下の通りです:

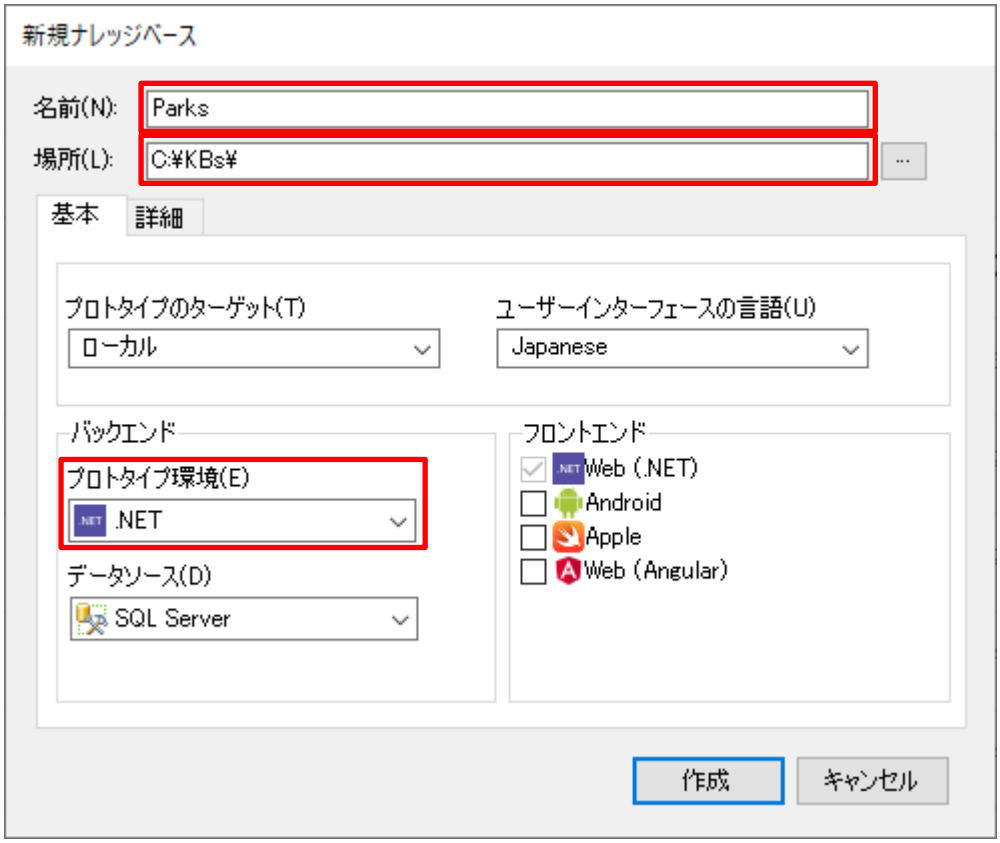
- **バックエンド**: この管理会社の社員が、インターネット接続を使用して任意の場所からデータを処理するための、Web サーバー上で実行されるアプリケーションパーツです。

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始めるにあたって、まず GeneXus を開き、「**Parks**」という名前のナレッジベースを作成します。

必須の設定事項は次の通りです。（下図赤枠とあわせて確認してください）

- 「名前」に「Parks」と指定
- 「場所」を変更する場合、「ユーザー（C:¥Users）」フォルダ以外を指定
- 「プロトタイプ環境」は「.NET」を選択



ヒント：

- 「新規ナレッジベース」ダイアログ表示方法
  1. [開始ページ]内のリンク「空のナレッジベースを作成」をクリック
  2. メニューバーの [ファイル] > [新規] > [ナレッジベース] を選択
  3. ショートカットキー「Ctrl + Shift + N」を押下

「ナレッジベースの作成中」というダイアログが消え、画面下部の [出力] ウィンドウに「Success: ナレッジベースを作成」と表示されたらナレッジベースの作成完了です。

少し時間を取り、**IDE (GeneXus の統合開発環境) に慣れてください。ウィンドウを移動したり、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])をしたり、ウィンドウの「ピン留め」を行ってください。

[**KB エクスプローラー**] (ナレッジ ベース エクスプローラー) ウィンドウ内の表示も確認してください。一部の**オブジェクト**などが既に取り込まれていることが分かります。

3. トランザクションの定義

管理会社の担当者との打ち合わせで、次のように説明されました:

> 「アミューズメントパークに関するデータを記録して、アミューズメントパークの従業員と、アミューズメントパークの来場者に提供するアトラクションやショーの両方を管理したいと考えています」

アプリケーションの構築は、現実からエンティティを識別することから始めます。
その後、識別したエンティティを**トランザクション**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でしょうか。

3－1. 「Employee」トランザクション

従業員について、どのようなデータを記録するのかを尋ねました。回答は次のとおりです:

> **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**です。

この情報を基に「**Employee**」トランザクションを作成します。

ヒント:

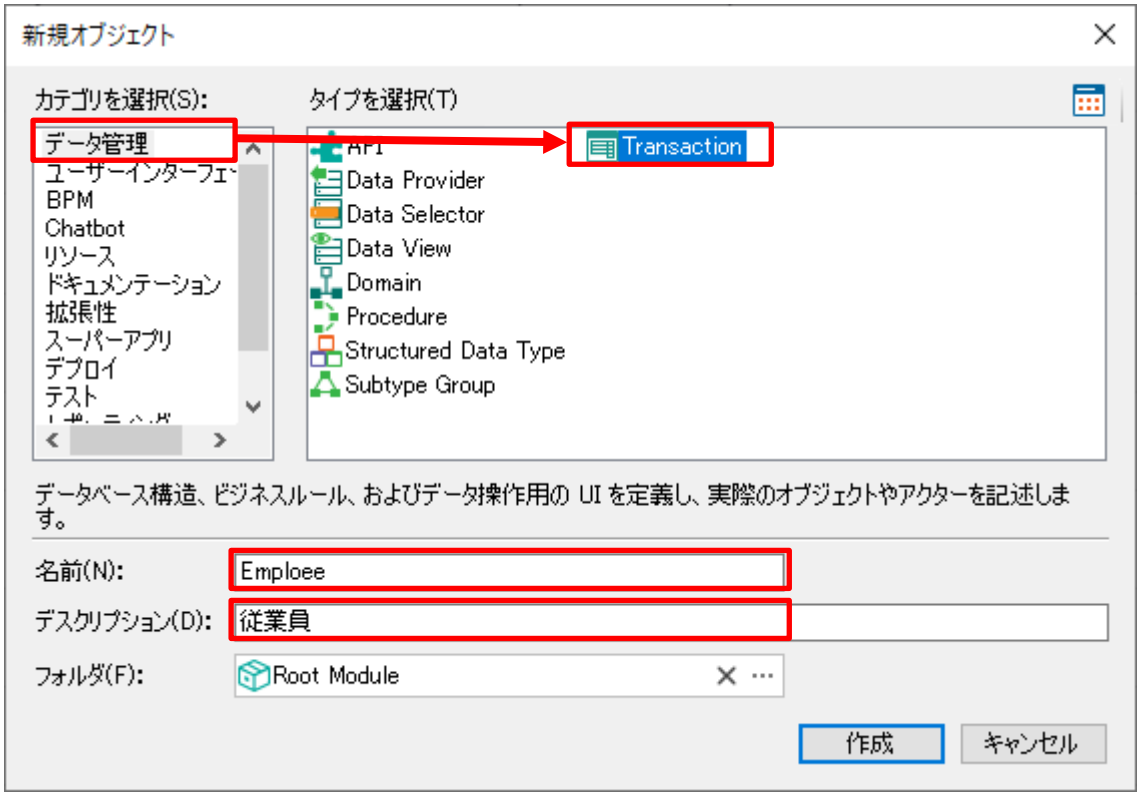
- 「新規オブジェクト」ダイアログ表示方法
  1. メニューバーの [ファイル] > [新規] > [オブジェクト]
  2. ショートカットキー「Ctrl + N」を押下
  3. ツールバーの アイコンをクリック
- 管理会社の回答にありませんが、従業員を識別するための項目属性が必要です (EmployeeId)。
- 新しい項目属性を追加するときに、ピリオド (.) を入力すると、トランザクションの名前に自動置換されます。
- **セマンティックドメイン**と呼ばれる **Address**、**Phone**、**Email** は、項目属性名の末尾に Address、Phone、Email の文字列がある場合に自動的に割り当てられます。

3-1-1. トランザクションの作成

「新規オブジェクト」ダイアログで、[カテゴリ] から「データ管理」を選択し、

[タイプを選択] 内で「Transaction」を選択し、[名前] を「Employee」、

[デスクリプション] を「従業員」とし、[作成] をクリック



3-1-2. Structure エレメントの定義

下記の内容で Structure エレメントを定義します。

名前	タイプ	デスクリプション
EmployeeId	Numeric(4.0)	従業員番号
EmployeeLastName	VarChar(20)	従業員名字
▪ EmployeeFirstName	VarChar(20)	従業員名前
▪ EmployeeAddress	Address, GeneXus	従業員住所
▪ EmployeePhone	Phone, GeneXus	従業員電話番号
▪ EmployeeEmail	Email, GeneXus	従業員メールアドレス

ヒント：

- 右の列へ移動するために [Tab] キーが利用可能
- 項目属性を新規入力するためには [Enter] キーを押下

3-1-3. 初めてのアプリケーションの実行

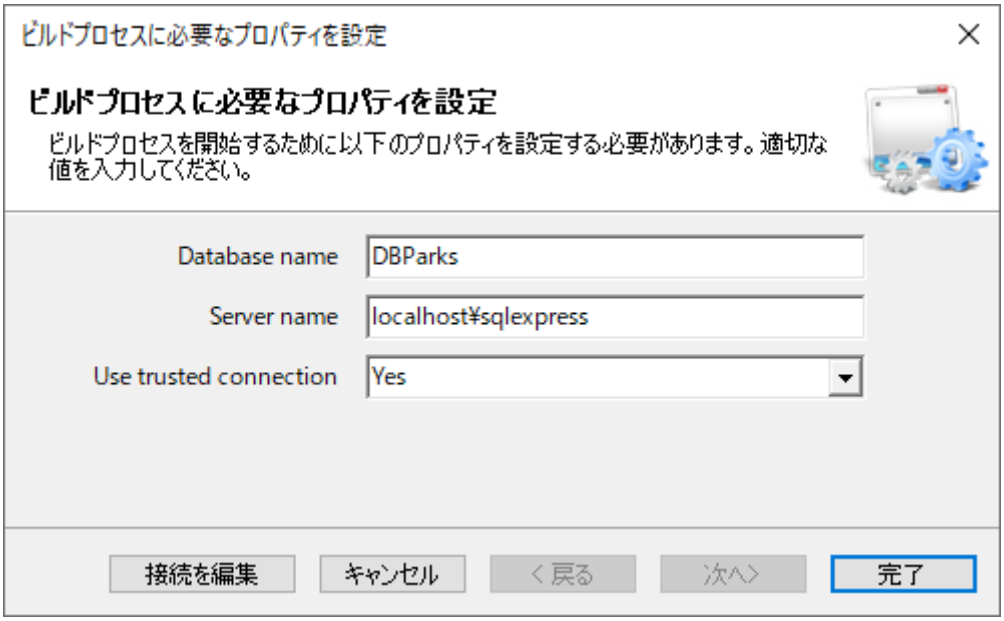
定義した「Employee」トランザクションに基づく従業員登録画面を実行します。

[F5] キーを押し、アプリケーションを**実行**します。

アプリケーションを実行するためには次の操作が必要となります。

①「**ビルドプロセスに必要なプロパティを設定**」ダイアログでの入力

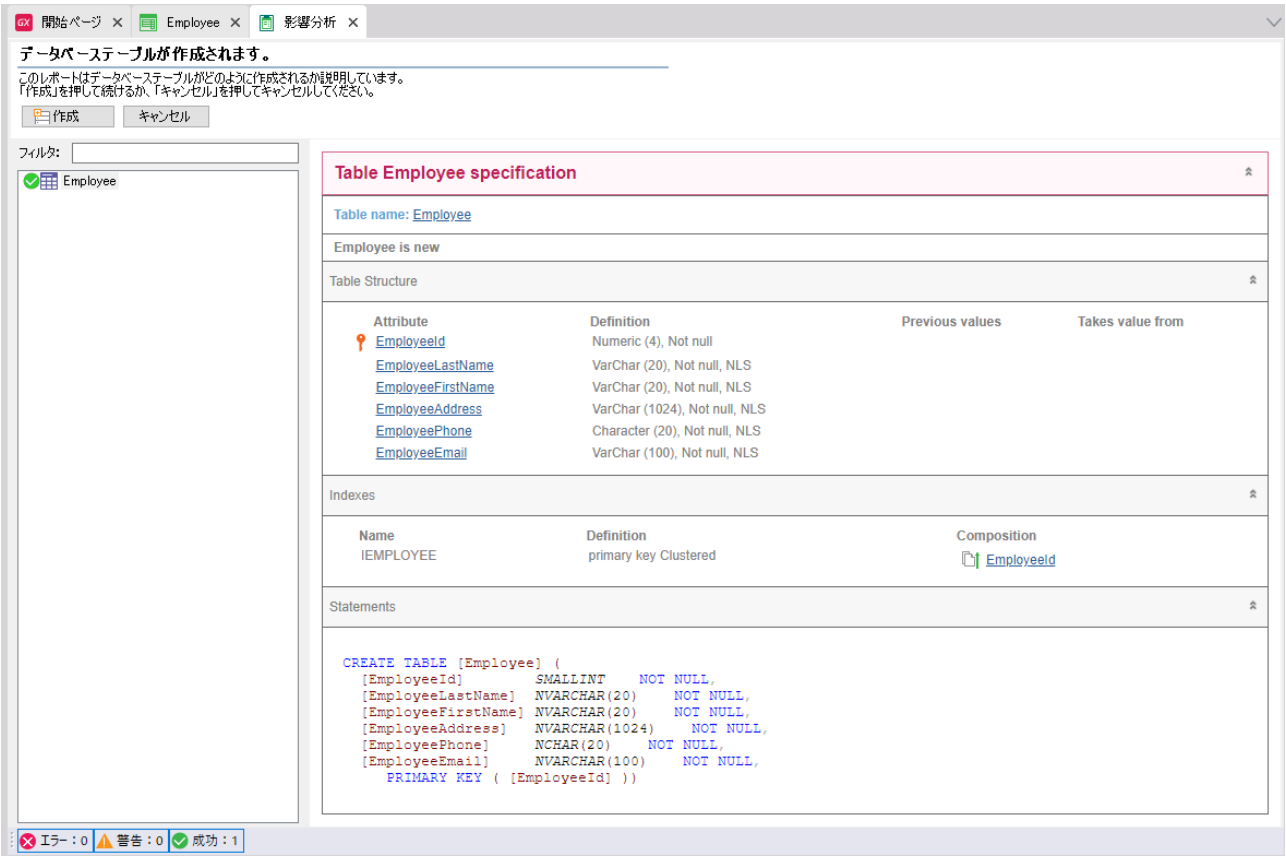
初めての実行となるため、GeneXus によって生成されるアプリケーションが利用する**データベースの名前**および、**作成先となるサーバー名**の入力が必要となります。
本課題では以下のように入力します。
Database name ：DBParks
Server name　　：localhost¥sqlexpress
※「Use trusted connection」は「Yes」のままにします。
指定したサーバーに名前が一致するデータベースがない場合、GeneXus によって自動生成が行われます。



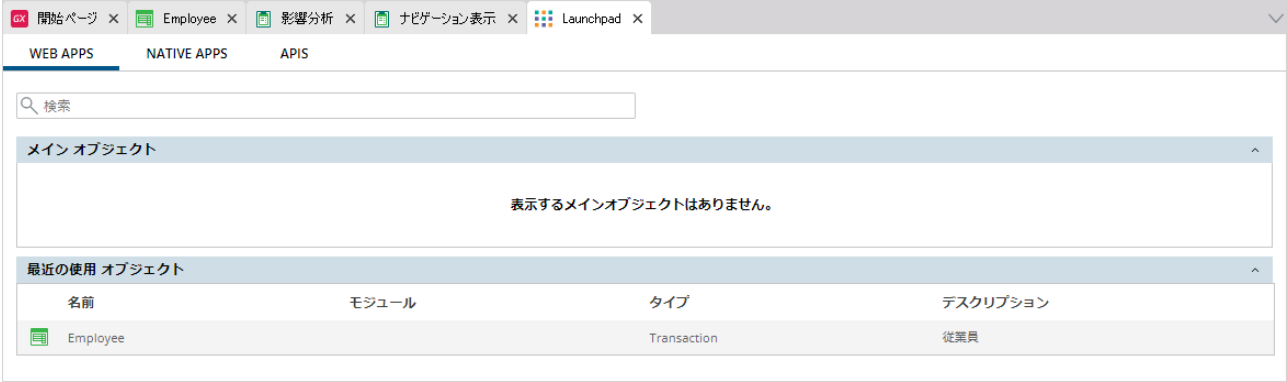
② [**影響分析**] ウィンドウでの操作

データベースを作成することと、そのデータベースに含まれる **Employee テーブル**の詳細情報表示されます。

このウィンドウで表示される [**作成**] ボタンをクリックします。



[**作成**] ボタンをクリックすると、GeneXus によってテーブルを含むデータベースの作成が実行されます。その後、アプリケーションのプログラムが作成され、生成が完了すると [**Launchpad**] ウィンドウが表示されます。



[Launchpad] ウィンドウには「**Employee**」トランザクションにより生成された画面を開くためのリンクが 1 つだけ表示されます。これをクリックします。

3-1-4. 従業員の登録/更新/削除

従業員データを管理できる画面が実行できたため、実際に利用してみます。

以下の操作を行ってください。

① 新しい従業員を以下の表（列名内「従業員」を割愛）に沿って**登録**（1 行 1 レコード）指定のない項目は未入力のままとします。

番号	名字	名前	住所	電話番号	メールアドレス
1	シルバ	ホセ	ペーニャ	11100002222	s.jose@test.com
2	ピエール	ジャン	パリ	22211110000	p.jean@test.com
3	山田	太郎	東京都	00011112222	t.yamada@test.com

② 2 番の従業員を以下の表の赤文字項目を**変更**

番号	名字	名前	住所	電話番号	メールアドレス
2	ピエール	ジャン	ベルサイユ	22211110000	p.jean@test.com

③　3 番の従業員を**削除**

ヒント：

- データを「登録」または「変更」する場合、「実行」ボタンを利用し、「削除」する場合、「削除」ボタンを利用します。
- 登録されたデータは画面上部のアイコン「|<」「<」「>」「>|」を利用し、読み込むことが可能
- 画面上部の [選択] リンクをクリックすると、リスト画面がポップアップされ、読み込むデータを選択可能
- データを読み込んだ状態で新規登録を行う場合、主キー項目を登録されていない値に変更

3-2. 「AmusementPark」トランザクションと「Country」トランザクション

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。

打ち合わせで聞いた内容を詳しく思い出すと次のような説明をしていました。

> 「さまざまな**国**のさまざまな都市にある**アミューズメントパーク**に関するデータを記録して、アミューズメントパークの従業員と、来場者に提供するアトラクションやショーを管理したいと考えています」

管理会社の社員に、アミューズメントパークのどのようなデータを記録するのかを尋ねました。回答は次のとおりです:

> **名前** (最大 40 文字)、**ウェブサイト** (最大 60 文字)、**住所**、そして**写真**です。

あわせてアミューズメントパークのある国をどのように記録したいか尋ねました。回答は次のとおりです:

> **国**は**あらかじめ登録**しておき、アミューズメントパークから**参照**したい。項目としては**名前** (最大 40 文字)です。

また、今後の実装において、データを識別する項目について指定がない場合、自動採番される数値で問題ないこと、名前の項目は最大 40 文字としたいと要望がありました。

この情報を基に「**AmusementPark**」トランザクションおよび「**Country**」トランザクションを作成します。

ヒント:

- アミューズメントパークを識別する項目（AmusementParkId）、国を識別する項目（CountryId）も必要
- 開発者によってあらかじめデータタイプ、桁数、その他設定を定義しておき、再利用するためには「ドメイン」を定義

3-2-1. ドメインの定義

初めに自動採番となる数値型を今後も繰り返し利用するため、以下の手順で「ドメイン」を定義します。

① **ドメインウィンドウ**で新規ドメインを以下の内容で定義します。

名前	タイプ	モジュール	デスクリプション
Id	Numeric(4.0)	Root Module	Id

② ドメインウィンドウで、定義した Id ドメインを**選択**し、**[F4] キー**を押下し、**[プロパティ] ウィンドウ**を表示します。

③ 表示された [プロパティ] ウィンドウで **[Autonumber]** プロパティを **[True]** に変更します。

ヒント：

- ドメインの定義方法
  1. ドメインウィンドウ：メニューバーの [表示] → [ドメイン] をクリック
  2. インライン定義：項目属性の [Type] 列で直接定義
  3. オブジェクト定義：「新規オブジェクト」ダイアログを表示し、「データ管理」カテゴリ内の「Domain」を選択
- ドメインウィンドウ内で新規ドメインを定義するためには、[Enter] キーを押下
- トランザクションの Structure 同様に [Tab] キーで右の列へ移動可能

名前のドメインは、アミューズメントパークの定義時にインライン定義を行います。

3-2-2. AmusementPark トランザクションの作成

下記の内容でトランザクションを作成し、Structure エレメントを定義します。

トランザクション定義：

名前	AmusementPark
デスクリプション	アミューズメントパーク

Structure 定義：

名前	タイプ	デスクリプション
AmusementParkId	Id	パーク番号
AmusementParkName	Name = VarChar(40)	パーク名
AmusementParkWebsite	VarChar(60)	パークウェブサイト
AmusementParkAddress	Address, GeneXus	パーク住所
AmusementParkPhoto	Image	パーク写真

ヒント：

- AmusementParkId のタイプは、直前に定義した Id ドメインが自動指定
- AmusementParkName のタイプに「Name = VarChar(40)」と入力し、フォーカスを移動すると、表示は「Name」となり、ドメインウィンドウに「Name」ドメインが新規作成されていることが確認可能

3-2-3. Country トランザクションの作成

アミューズメントパークがある国を記録するためのトランザクションを作成します。

下記の内容でトランザクションを作成し、Structure エレメントを定義します。

トランザクション定義：

名前	Country
デスクリプション	国

Structure 定義：

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

ヒント：

- CountryName のタイプは、アミューズメントパーク定義時に定義された Name ドメインが自動指定

3-2-4. AmusementPark と Country の関連付け

アミューズメントパークに国を指定できるようにするため、AmusementPark トランザクションの Structure エレメントに項目を追加します。

追加後のイメージは以下の通りです。（文字色が緑のものは定義済みのものです）

Structure 定義：

名前	タイプ	デスクリプション
AmusementParkId	Id	パーク番号
AmusementParkName	Name	パーク名
AmusementParkWebsite	VarChar(60)	パークウェブサイト
AmusementParkAddress	Address, GeneXus	パーク住所
AmusementParkPhoto	Image	パーク写真
CountryId	Id	国番号
CountryName	Name	国名

ヒント：

- CountryId、CountryName は「名前」の定義を行うと、自動で「タイプ」および「デスクリプション」に Country トランザクションでの定義内容に基づく内容を表示

3-2-5. アプリケーションの実行

定義した「Country」および「AmusementPark」トランザクションに基づく登録画面を実行します。

[F5] キーを押し、アプリケーションを**実行**します。

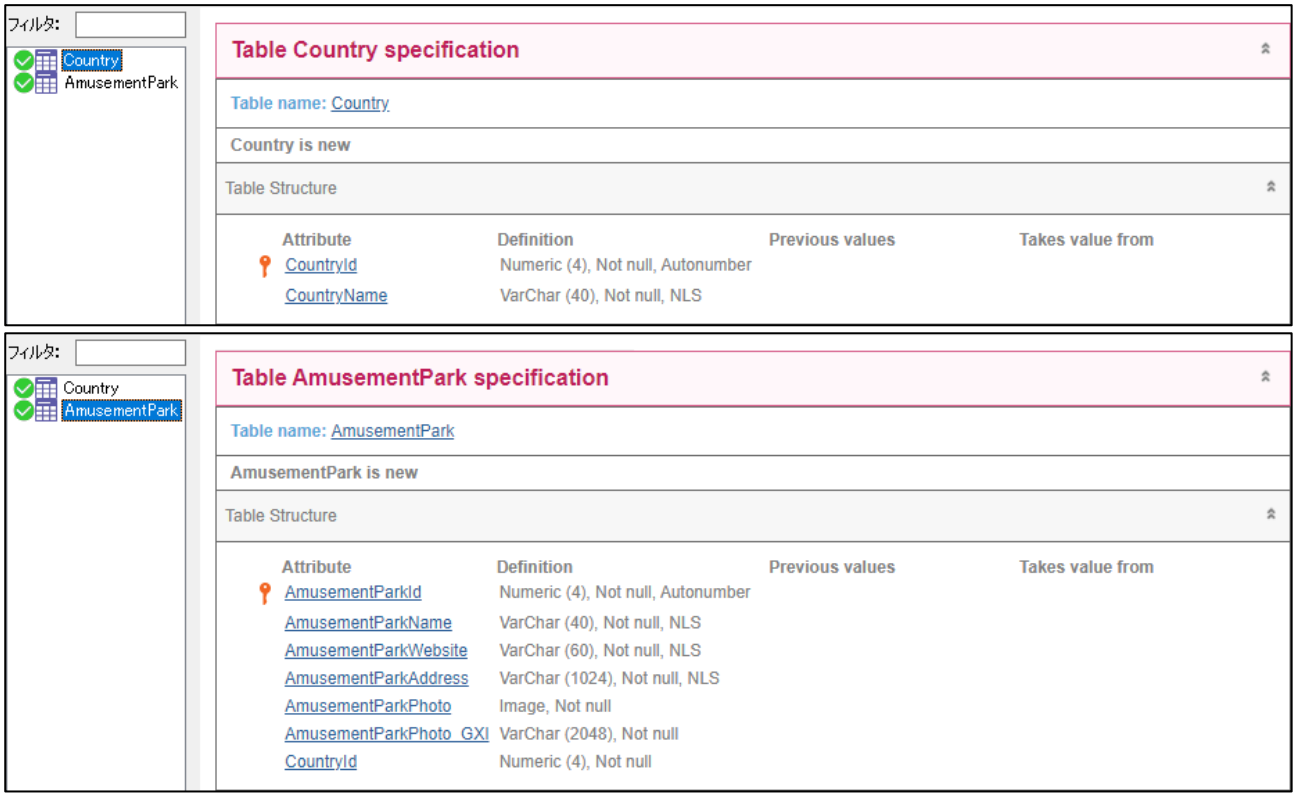
アプリケーションを実行するためには次の操作が必要となります。

①[**影響分析**] ウィンドウでの操作

データベースの再編成が必要であり、再編成内容の詳細を確認できます。

この場合、再編成の内容は 2 つのテーブルの作成です。

このウィンドウで表示される [**再編成**] ボタンをクリックします。



ヒント：

- Image タイプの項目属性をデータベース上で管理するために、GeneXus はテーブル内に列を 2 つ作成。
  1 つは項目属性名のままの列となり、もう一つは項目属性名末尾に「_GXI」を追加したもの。

3-2-6. 国およびアミューズメントパークの登録

表示された [Launchpad] ウィンドウに「Country」および「AmusementPark」トランザクションにより生成された画面を開くためのリンクが追加されています。
それぞれのリンクから登録画面を表示し、以下のデータを登録します。

以下の表は、１行が１レコードとして登録する内容となるため、１行分の入力が完了するごとに「実行」ボタンで登録してください。
指定のない項目は未入力のままとします。

①国：

国名
ブラジル
フランス
中国
アメリカ
ウルグアイ
フィリピン
日本

ヒント：

- 「国番号」は自動採番が有効なため。未入力でも最新の番号が割り当てられる

登録完了後、「選択」リンクをクリックし、下記のように自動採番され、データが登録されていることを確認します。

②アミューズメントパーク：

パーク名	パークウェブサイト	パーク住所	パーク写真	国番号
ベット・カレーロ	betocarrero.com	ペーニャ	BetoCarrero.png	1
ミニ・ムンド	minimundo.com	グラマド	MiniMundo.png	1
テラ・ボタニカ	terrabotanica.com	アンジェ	TerraBotanica.png	2

ヒント：

- 「パーク番号」は自動採番が有効なため。未入力でも最新の番号が割り当てられる
- 「パーク写真」は「 」アイコンをクリックすることで、画像ファイル選択ダイアログを表示可能
- 「国番号」は手入力をすることも「 」アイコンをクリックし、一覧から選択することも可能
- 「国名」は「国番号」を入力後、フォーカスアウト（例：Tab キー押下）することで自動読み込み

3-3. 整合性維持の確認

システム開発を進める中で、整合性維持の挙動について確認の依頼がありました。
現時点で、**AmusementPark** と **Country** は関連付けられているため、この機能を利用して依頼内容に対する確認を行います。
以下の操作を行い、実行結果を確認してください。

3-3-1. 存在しない国を利用した新規登録

アミューズメントパーク登録時に存在しない国を指定した場合どうなるかを確認します。
[Launchpad] ウィンドウから「AmusementPark」を呼び出します。
以下の内容を入力し、どのような挙動となるか確認してください。

パーク名	国番号
ビセンテナリオ公園	20

未指定の項目は空のままにします。
また、「国番号」は存在しない値のため、手入力する必要があります。

3-3-2. 存在しない国を利用した更新

すでに登録済みのアミューズメントパークを更新する場合に、存在しない国を指定した場合どうなるかを確認します。
[Launchpad] ウィンドウから「AmusementPark」を呼び出し、以下の操作を行い、どのような挙動となるか確認してください。

① 画面上部の「＞」をクリックし、「パーク番号：１」の「ベット・カレーロ」を読み込み
② 「国番号」の値を「15」に変更し、フォーカスアウト

3-3-3. 参照されているデータの削除

アミューズメントパーク登録時に参照された国を削除した場合どうなるかを確認します。

[Launchpad] ウィンドウから「Country」を呼び出し、以下の操作を行い、同のような挙動となるか確認してください。

① 画面上部の「選択」をクリックし、表示されたポップアップから「国名」が「ブラジル」のデータを選択

② 「削除」ボタンを押下

3-4. 「Show」トランザクション

続いて打ち合わせで出てきたもののうち、ショーの管理を実装します。
管理会社の社員に、ショーとしてどのようにデータを記録する必要があるか詳細を訪ねた結果、次のような回答がありました：

名前 (最大 40 文字)、**写真**です。

さらにショーについては「複数のアミューズメントパーク」で実施され、「いつ（日付、時間）」実施されるかについても記録する必要があります。
また、1 つのアミューズメントパークに複数のショーが記録可能です。

この情報を基にショーを管理する「Show」トランザクションを作成します。

ヒント：

- ショーを識別する項目（ShowId）も必要
- アミューズメントパークに複数のショーを紐づけるためには、「レベル」の追加が必要

3-4-1. Show トランザクションの作成

下記の内容でトランザクションを作成し、Structure エレメントを定義します。

トランザクション定義：

名前	Show
デスクリプション	ショー

Structure 定義：

名前	タイプ	デスクリプション
ShowId	Id	ショー番号
ShowName	Name	ショー名
ShowImage	Image	ショー画像

3-4-2. AmusementPark と Show の関連付け

アミューズメントパークに複数のショーを指定できるようにするため、AmusementPark トランザクションの Structure エレメントに項目を追加します。

追加のイメージは以下の通りです。（文字色が緑のものは定義済みのものです）

Structure 定義：

名前	タイプ	デスクリプション
AmusementParkId	Id	パーク番号
AmusementParkName	Name	パーク名
AmusementParkWebsite	VarChar(60)	パークウェブサイト
AmusementParkAddress	Address, GeneXus	パーク住所
AmusementParkPhoto	Image	パーク写真
CountryId	Id	国番号
CountryName	Name	国名
Show	Show	ショー
ShowId	Id	ショー番号
ShowName	Name	ショー番号
ShowImage	Image	ショー画像
AmusementParkShowDate	Date	ショー日付
AmusementParkShowTime	Time, GeneXus	ショー時間

ヒント：

- 新しいレベルの挿入方法（いずれもレベルを追加したい位置の一つ上の項目属性を選択した状態で実施）
  1. メニューバーの [編集] → [レベルを挿入] をクリック
  2. 右クリックオプションから [レベルを挿入] をクリック
  3. ショートカットキーの「Ctrl + L」を押下
- 「ShowId」は第 2 レベルの主キーであり、Show を参照する外部参照キーとなる

3-4-3. アプリケーションの実行

定義した「Show」および更新した「AmusementPark」トランザクションに基づく登録画面を実行します。

[F5] キーを押し、アプリケーションを実行します。

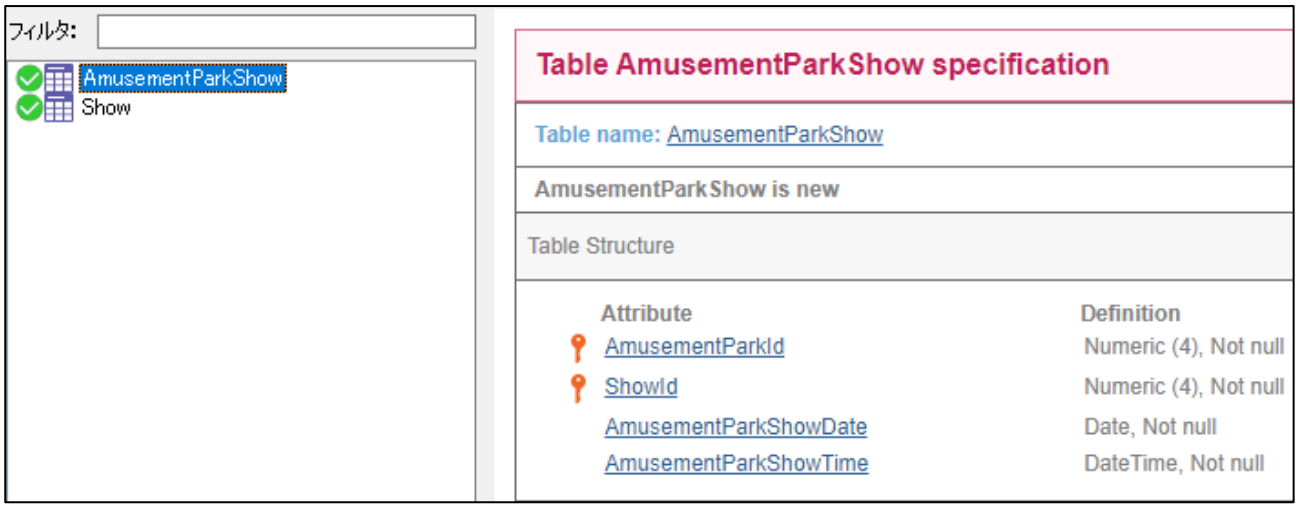
アプリケーションを実行するためには次の操作が必要となります。

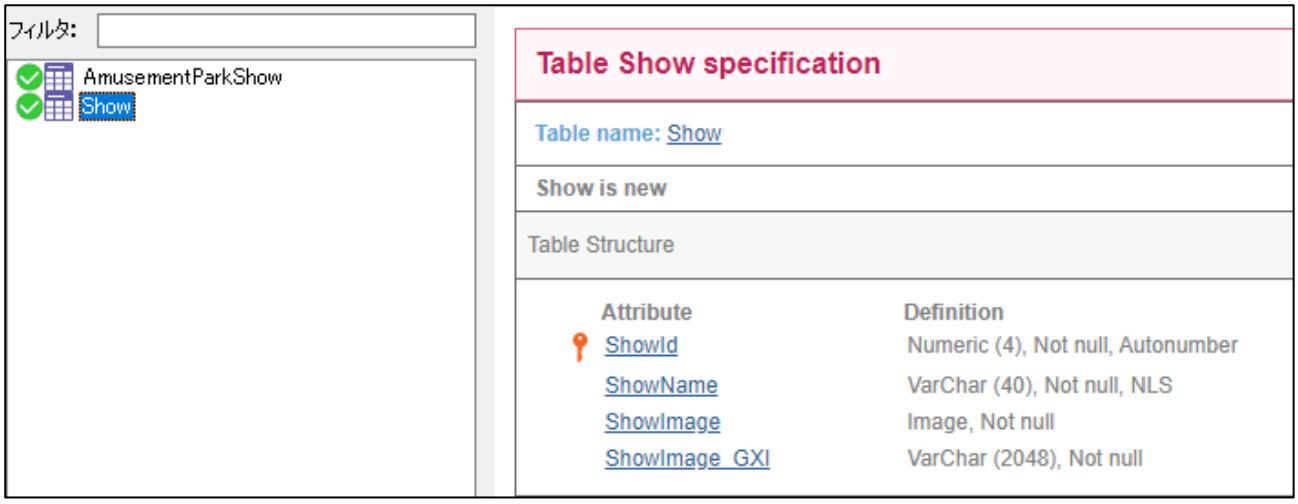
①[**影響分析**] ウィンドウでの操作

データベースの再編成が必要であり、再編成内容の詳細を確認できます。

この場合、再編成の内容は 2 つのテーブルの作成です。

このウィンドウで表示される [**再編成**] ボタンをクリックします。





ヒント：

- [AMUSEMENTPARKSHOW] テーブルは「AmusementPark」に追加した第 2 レベル「Show」に基づく
- [AMUSEMENTPARKSHOW] テーブルは第 1 レベル、第 2 レベルの主キー項目による複合キーで実装

3-4-4. ショーの登録

表示された [Launchpad] ウィンドウに「Show」トランザクションにより生成された画面を開くためのリンクが追加されました。

このリンクをクリックし、ショーの登録画面を開きます。

ショーを以下の表に沿って登録します。（指定のない項目は未入力のまま）

ショー名	ショー画像
グリーティング	Greeting.png
ステージイベント	Stage.png
パレード	Parade.png
解説ツアー	Tour.png
フラッシュモブ	FlashMob.png

3-4-5. アミューズメントパークへのショー追加

表示された [Launchpad] ウィンドウから「AmusementPark」トランザクションにより生成された画面を開きます。

以下の表に沿ってアミューズメントパークにショーを追加します。（文字色が緑のものは登録済みのものです）

パーク番号

1

ショー番号	ショー日付	ショー時間
1	現在の翌月の 1 日	10:00
2	現在の翌月の 5 日	15:00
3	現在の翌月の 5 日	12:30
5	現在の翌月の 10 日	15:00

2

ショー番号	ショー日付	ショー時間
1	現在の翌月の 1 日	11:00
4	現在の翌月の 5 日	15:00

3

ショー番号	ショー日付	ショー時間
1	現在の翌月の 1 日	13:00
2	現在の翌月の 5 日	10:00
4	現在の翌月の 5 日	15:30

ヒント：

- アミューズメントパークに対して複数のショーを一度に追加可能
- 入力可能なショーは登録済みのショーのみ
- 同じショーを異なるアミューズメントパークに追加可能
- 同じショーを１つのアミューズメントパークに複数回追加することは不可能

3-5. 「Game」トランザクション

続いてはアトラクションの管理を実装します。

このアトラクションは独立して登録を行うことができますが、特定のアミューズメントパークに所属するものとして管理する必要があります。

記録する必要があるデータについては、次のような回答がありました：

名前 (最大 40 文字)、**アミューズメントパークの名前**です。

この情報を基にアトラクションを管理する「Game」トランザクションを作成します。

ヒント：

- アトラクションを識別する項目（GameId）も必要
- アミューズメントパーク名を参照するためには外部参照キー（AmusementParkId）も必要

3-5-1. Game トランザクションの作成

下記の内容でトランザクションを作成し、Structure エレメントを定義します。

トランザクション定義：

名前	Game
デスクリプション	アトラクション

Structure 定義：

名前	タイプ	デスクリプション
GameId	Id	アトラクション番号
GameName	Name	アトラクション名
AmusementParkId	Id	パーク番号
AmusementParkName	Name	パーク名

3-5-2. アプリケーションの実行

定義した「Game」トランザクションに基づく登録画面を実行します。

[F5] キーを押し、アプリケーションを**実行**します。

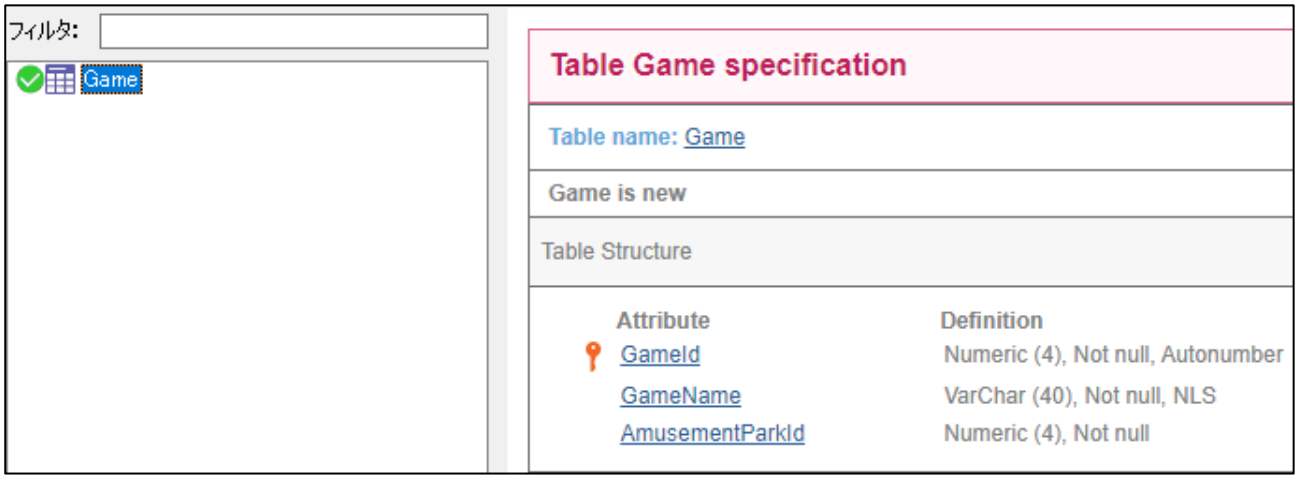
アプリケーションを実行するためには次の操作が必要となります。

①[**影響分析**] ウィンドウでの操作

データベースの再編成が必要であり、再編成内容の詳細を確認できます。

この場合、再編成の内容は 1 つのテーブルの作成です。

このウィンドウで表示される [**再編成**] ボタンをクリックします。



3-5-3．アトラクションの登録

表示された [Launchpad] ウィンドウに「Game」トランザクションにより生成された画面を開くためのリンクが追加されました。

このリンクをクリックし、アトラクションの登録画面を開きます。

アトラクションを以下の表に沿って登録します。

アトラクション名	パーク番号
フライ・エレファント	1
カルーセル	2
フリーフォール	1
ファイア・ウィップ	1
プラント・メイズ	3
ワールドシップ	2

3-6. 「Category」トランザクション

アトラクションに関する要望をよく思い出したところ、各アトラクションにはカテゴリ（子供向け、絶叫系、楽しいアトラクション等）も記録する必要があり、あらかじめ登録したカテゴリを各アトラクション登録時に関連付けることを希望しています。
ただし、カテゴリは必須ではないという要望（未指定のまま登録できる）もありました。
カテゴリについてどのようなデータを記録するか尋ねたところ、次の回答がありました：

名前 (最大 40 文字)のみです。

この情報を基に「**Category**」トランザクションの作成および「**Game**」トランザクションの更新を行います。

ヒント：

- カテゴリを識別する項目（CategoryId）も必要
- 外部参照キーを未指定で登録するためには、「Null 許容」を「Yes」に設定

3-6-1. Category トランザクションの作成

下記の内容でトランザクションを作成し、Structure エレメントを定義します。

トランザクション定義：

名前	Category
デスクリプション	カテゴリ

Structure 定義：

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

3-6-2. Game と Category の関連付け

アトラクションにカテゴリを指定できるようにするため、Game トランザクションの Structure エレメントに項目を追加します。
また、アトラクションには**画像**も管理できる必要があると改めてリクエストがあり、この追加もあわせて実施します。

追加のイメージは以下の通りです。（文字色が緑のものは定義済みのものです）

Structure 定義：

名前	タイプ	デスクリプション	Null 許容
GameId	Id	アトラクション番号	No
GameName	Name	アトラクション名	No
AmusementParkId	Id	パーク番号	No
AmusementParkName	Name	パーク名	
CategoryId	Id	カテゴリ番号	**Yes**
CategoryName	Name	カテゴリ名	
GameImage	Image	アトラクション画像	No

ヒント：

- 外部参照キー「CategoryId」を未指定で登録するために、「Null 許容」を「Yes」に設定

3-6-3. アプリケーションの実行

定義した「Category」および更新した「Game」トランザクションに基づく登録画面を実行します。

[F5] キーを押し、アプリケーションを実行します。

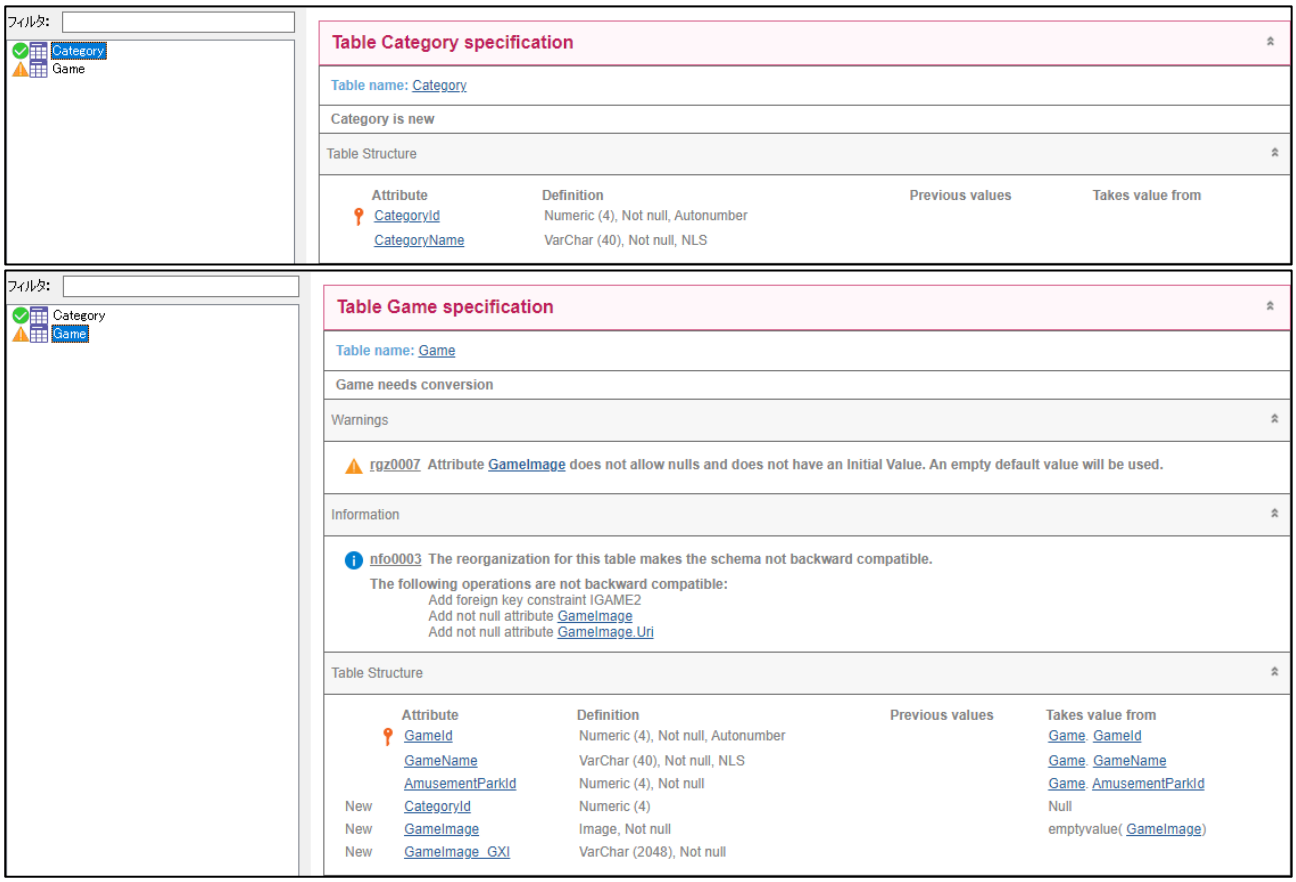
アプリケーションを実行するためには次の操作が必要となります。

①[**影響分析**] ウィンドウでの操作

データベースの再編成が必要であり、再編成内容の詳細を確認できます。

この場合、再編成の内容は 2 つのテーブルの作成です。

このウィンドウで表示される [**再編成**] ボタンをクリックします。



ヒント：

- 「GameImage」は「Null 許容」を「No」のままにし、初期値の指定がないため、GAME テーブルに既存のレコードがある場合、データタイプの既定値が自動で割り当てられた状態で列が追加されることが警告「rgz0007」として表示
- 「CategoryId」は「Null 許容」を「Yes」に変更したため、GAME テーブルに既存のレコードがあったとしても、値は未指定（Null）の状態で列を追加

3-6-4. カテゴリの登録

表示された [Launchpad] ウィンドウに「Category」トランザクションにより生成された画面を開くためのリンクが追加されました。
このリンクをクリックし、カテゴリの登録画面を開きます。
カテゴリを以下の表に沿って登録します。

カテゴリ名
子供向け
絶叫系
楽しいアトラクション

3-6-5. アトラクションの編集

すでに登録されているアトラクションの一部にカテゴリおよび画像の追加を行います。

[Launchpad] ウィンドウから「Game」トランザクションにより生成された画面を開き、以下の表に沿ってデータを更新します。（必要な項目のみ表示）
列に値の記載がないものはデータとして未入力のままとします。

アトラクション番号	カテゴリ番号	アトラクション画像
1	1	FlyElephant.png
2	1	Carousel.png
3	2	FreeFall.png
4	2	FireWhip.png
5		PlantMaze.png
6	3	WorldShip.png

3-7. 「Employee」トランザクションと「AmusementPark」トランザクション

アプリケーションの開発を始めるときに管理会社より、アミューズメントパークと**そのアミューズメントパークの**従業員を管理したいという要望を聞いていました。
このデータの関係について改めて確認したところ、追加で以下の要望も持っていることが確認できたため、この要望もかなえる必要があります。
①従業員が勤務するアミューズメントパークは 1 つだけ
②一部の従業員は勤務するアミューズメントパークの指定は必須ではない

この情報を基に「Employee」トランザクションを更新します。

3-7-1. Employee と AmusementPar の関連付け

要望を満たすように Employee トランザクションの Structure エレメントに項目を追加します。
追加後のイメージは以下の通りです。（文字色が緑のものは定義済みのものです）

Structure 定義：

名前	タイプ	デスクリプション	Null 許容
EmployeeId	Numeric(4.0)	従業員番号	No
EmployeeLastName	VarChar(20)	従業員名字	No
EmployeeFirstName	VarChar(20)	従業員名前	No
EmployeeAddress	Address, GeneXus	従業員住所	No
EmployeePhone	Phone, GeneXus	従業員電話番号	No
EmployeeEmail	Email, GeneXus	従業員メールアドレス	No
AmusementParkId	Id	パーク番号	Yes
AmusementParkName	Name	パーク名	

3-7-2. アプリケーションの実行

更新した「Employee」トランザクションに基づく登録画面を実行します。

[F5] キーを押し、アプリケーションを**実行**します。

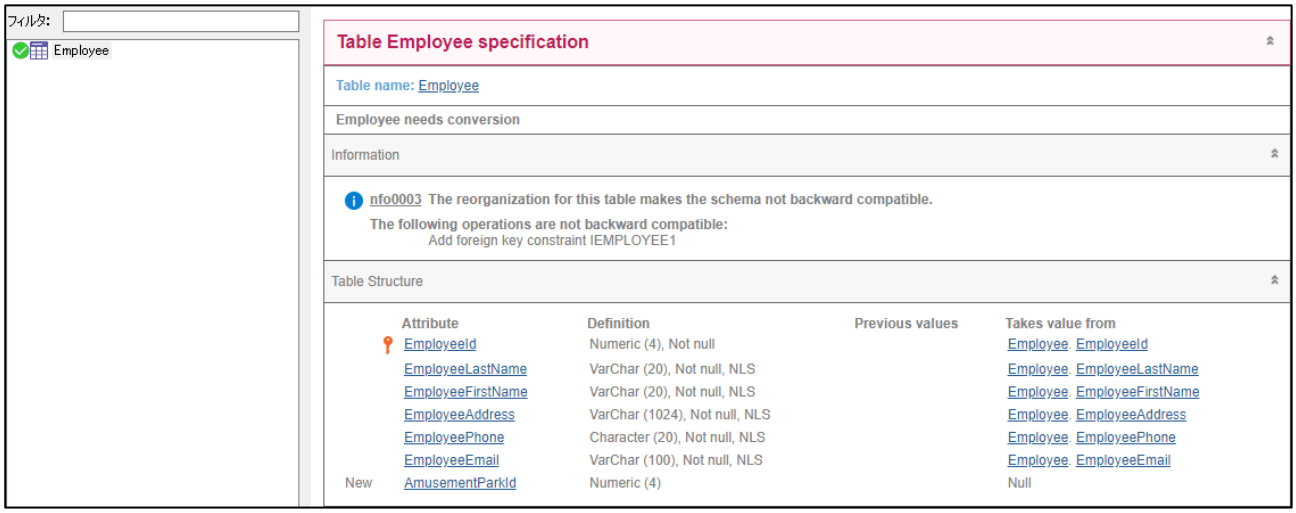
アプリケーションを実行するためには次の操作が必要となります。

①[**影響分析**] ウィンドウでの操作

データベースの再編成が必要であり、再編成内容の詳細を確認できます。

この場合、再編成の内容は 1 つのテーブルの再編成です。

このウィンドウで表示される [**再編成**] ボタンをクリックします。



3-7-3. 従業員の登録

すでに登録されている従業員の一部にアミューズメントパークの追加を行います。

[Launchpad] ウィンドウから「Employee」トランザクションにより生成された画面を開き、以下の表に沿ってデータを更新します。（必要な項目のみ表示）

列に値の記載がないものはデータとして未入力のままとします。

従業員番号	パーク番号
1	1

3-8. 「Country」トランザクションへの都市の追加

このアプリケーションでは、各国の都市に関するデータも記録する必要がありました。
管理会社の社員に、都市に必要なデータを尋ねたところ、次のような回答がありました：

> ある**国に関連**し、**一度に多数**の都市を追加し、**名前** (最大 40 文字) のみ管理します。

都市については、必ず特定の国に関連し、他の国のデータとして再利用しません。
この情報を基に「Country」トランザクションを更新します。

ヒント：

- 都市を識別する項目（CityId）も必要
- 国に複数の都市を紐づけるためには、「レベル」の追加が必要

3-8-1. Country へ City レベルの追加

国に複数の都市を登録できるようにするため、Country トランザクションの Structure エレメントに項目を追加します。

追加のイメージは以下の通りです。（文字色が緑のものは定義済みのものです）

Structure 定義：

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

ヒント：

- 新しいレベルの挿入方法（いずれもレベルを追加したい位置の一つ上の項目属性を選択した状態で実施）
  1. メニューバーの [編集] → [レベルを挿入] をクリック
  2. 右クリックオプションから [レベルを挿入] をクリック
  3. ショートカットキーの「Ctrl + L」を押下
- City レベルに項目属性を定義する際、半角ダブルクォーテーション「”」を入力するとレベルの名前に自動で置換

3-8-2. アプリケーションの実行

更新した「Country」トランザクションに基づく登録画面を実行します。

[F5] キーを押し、アプリケーションを実行します。

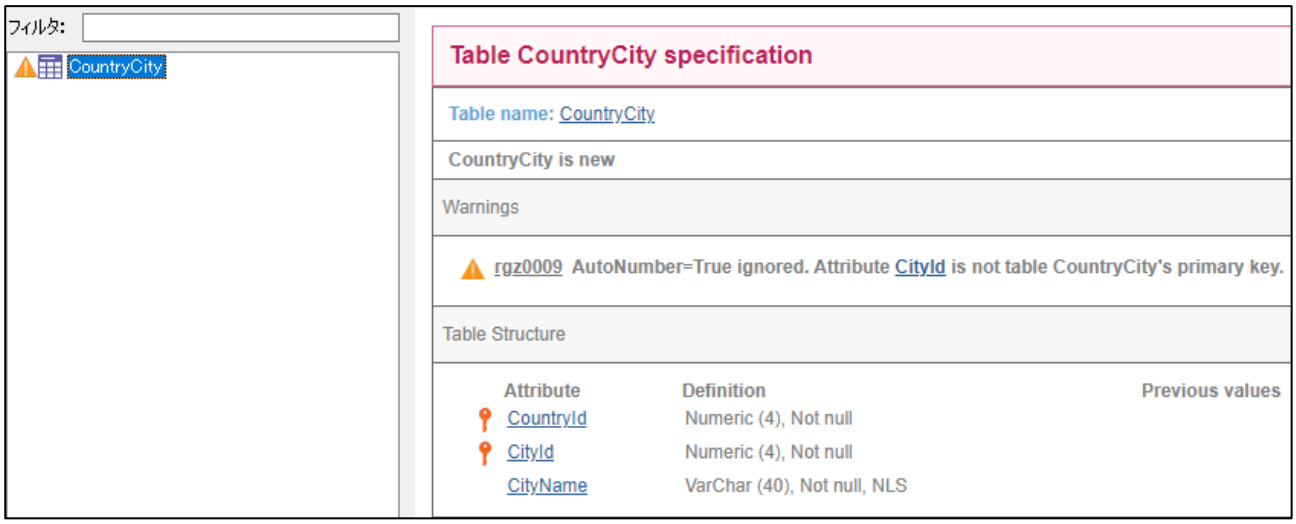
アプリケーションを実行するためには次の操作が必要となります。

①[**影響分析**] ウィンドウでの操作

データベースの再編成が必要であり、再編成内容の詳細を確認できます。

この場合、再編成の内容は 1 つのテーブルの作成です。

このウィンドウで表示される [**再編成**] ボタンをクリックします。



ヒント：

- [COUNTRYCITY] テーブルは「Country」に追加した第 2 レベル「City」に基づく
- [COUNTRYCITY] テーブルは第 1 レベル、第 2 レベルの主キー項目による複合キーで実装
- 「CityId」の AutoNumber プロパティによる設定は無視されることが警告「rgz0009」として表示
  → AutoNumber プロパティによる自動採番は単独主キーの場合のみ

3-8-3. 都市の登録

表示された [Launchpad] ウィンドウから「Country」トランザクションにより生成された画面を開きます。

以下の表に沿って国に都市を追加します。（文字色が緑のものは登録済みのものです）

国番号

1

都市番号	都市名
1	ペーニャ
2	グラマド
3	アマゾナス

2

都市番号	都市名
1	アンジェ

4

都市番号	都市名
1	ニューヨーク
2	アマゾナス
3	ニューヨーク

ヒント：

- 国に対して複数の都市を一度に追加可能
- 各国へ登録できる都市に制限はないため、同じ都市名を異なる国に追加も可能
- １つの国に同じ都市名を登録することも可能
- 都市の入力行を追加する場合、「行追加」をクリック

3-9. 「AmusementPark」トランザクションへの都市の追加

このアプリケーションでは、アミューズメントパークが存在する都市も記録する必要がありました。**ただし、タイミングによっては、管理会社がアミューズメントパークの存在する都市を把握していなかったり、アミューズメントパークが特定の都市に関連しなかったりすることがあるとのことです。**

この情報を基に「AmusementPark」トランザクションを更新します。

3-9-1. AmusementPark と City レベルの関連付け

要望を満たすように AmusementPark トランザクションの Structure エレメントに項目を追加します。

追加後のイメージは以下の通りです。（文字色が緑のものは定義済みのものです）

Structure 定義（一部省略）：

名前	タイプ	デスクリプション	Null 許容
AmusementParkId	Id	パーク番号	No
AmusementParkName	Name	パーク名	No
⋮			
CountryId	Id	国番号	No
CountryName	Name	国名	
CityId	Id	都市番号	**Yes**
CityName	Name	都市名	
Show	Show	ショー	
⋮			

ヒント：

- CountryName の下に項目属性を追加する場合、CountryName を選択した状態で Enter キー押下

3-9-1. アプリケーションの実行

更新した「AmusementPark」トランザクションに基づく登録画面を実行します。

[F5] キーを押し、アプリケーションを実行します。

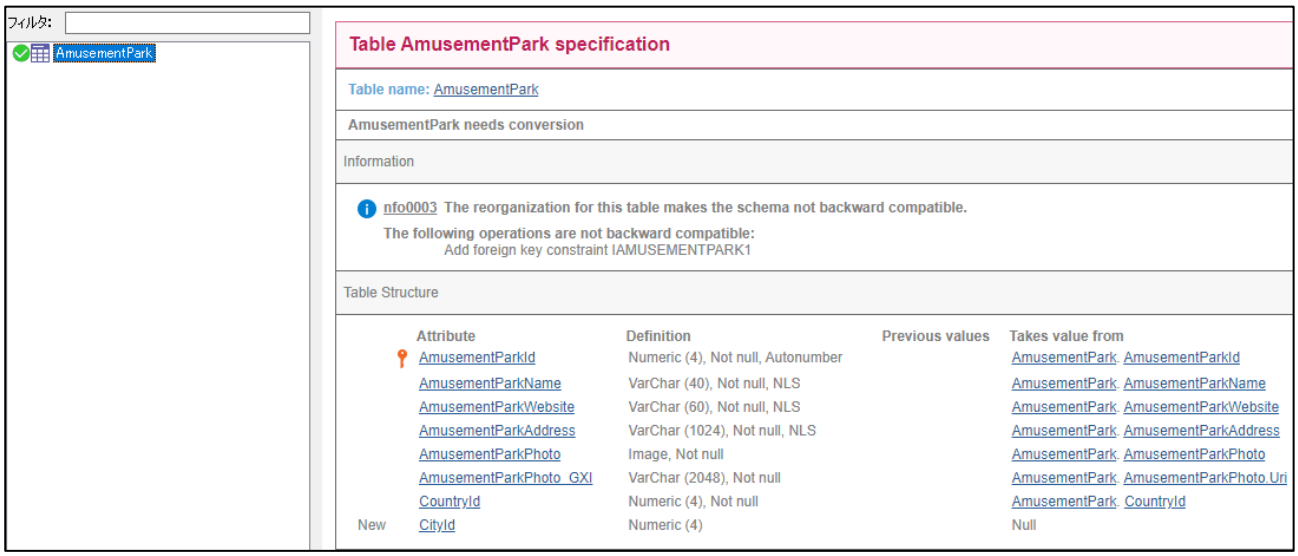
アプリケーションを実行するためには次の操作が必要となります。

①[**影響分析**] ウィンドウでの操作

データベースの再編成が必要であり、再編成内容の詳細を確認できます。

この場合、再編成の内容は 1 つのテーブルのみの再編成です。

このウィンドウで表示される [**再編成**] ボタンをクリックします。



3-9-2. アミューズメントパークの更新

表示された [Launchpad] ウィンドウから「AmusementPark」トランザクションにより生成された画面を開きます。

以下の表に沿って既存アミューズメントパークの更新、新規アミューズメントパークの登録を行います。（ショーの追加は行いません）

既存アミューズメントパークの更新（文字色が緑のものは登録済み、一部列を非表示）

パーク番号	パーク名	Web サイト URL	パーク住所	国番号	都市番号
1	ベット・カレーロ	betocarrero.com	ペーニャ	1	1
3	テラ・ボタニカ	terrabotanica.com	アンジェ	2	1

新規アミューズメントパークの登録（記載のない項目は未入力）

パーク名	パーク写真	国番号	都市番号
ヴィクトリアンガーデン	VictorianGarden.png	4	1
パルケ・テラ・マジカ	ParqueTerraMagica.png	1	

4. トランザクションに制御を追加

開発中のアプリケーションを管理会社の担当者と一緒にテストしました。
その中でデータを管理するために次のような制御が必要であると話が出ました：

- 「名字と名前を入力しないと、従業員を登録できない」
- 「従業員の電話番号が入力されていない場合は、警告を表示する」
- 「従業員がシステムに入力された日付を記録する (**EmployeeAddedDate**)。
  この項目属性の既定値は当日の日付とし、変更はできない」
- 「国に追加する都市も自動採番をしたい」

このような制御は GeneXus では「**Rules**」エレメントにおいてルールとして定義する必要があります。

では、どのようなルールを定義する必要があるでしょうか。

4-1. 「Employee」トランザクションに制御を追加

必要な制御のうち次の 3 つは Employee トランザクションで実装する必要があります。

- 「名字と名前を入力しないと、従業員を登録できない」
- 「従業員の電話番号が入力されていない場合は、警告を表示する」
- 「従業員がシステムに入力された日付を記録する (**EmployeeAddedDate**)。
  この項目属性の既定値は当日の日付とし、変更はできない」

この内容を基に「Employee」トランザクションを更新します。

ヒント：

- ルールの記述は「;」（セミコロン）区切り
- 項目属性の **IsEmpty()** メソッドを利用すると、空の値の場合、True 、入力済みの場合、False が戻り値となる
- 現在の日付を参照する場合、&Today というシステム変数が活用可能

4-1-1. 必須入力制御の追加

上図のように Employee トランザクションのウィンドウ上部より「Rules」エレメントを選択し、名字と名前それぞれを入力しなければ登録できない、つまり、「名字（EmployeeLastName）が未入力」および「名前（EmployeeFirstName）が未入力」の場合、エラーが発生するように Rules エレメントを定義します。

Rules 定義：

```
Error(”名字が未入力です。”) If EmployeeLastName.IsEmpty();
Error(”名前が未入力です。”) If EmployeeFirstName.IsEmpty();
```

ヒント：

- ルールやナレッジベース内のオブジェクト（項目属性など）の名前は「Ctrl + Space」キー押下で候補から選択可能
- １行目のルールの記入を終え、改行する場合、「If」以降が自動で改行される（上記と実行時の差異はありません）
- 全角文字列は「"」でくくられている必要がある。
  くくられていない全角文字列がある場合、以下のエラーが保存時に出力（よくあるケース：全角スペース）
  エラー src0014：エクスプレッションのパース中に不明なエラーが発生しました

4-1-2. 警告メッセージの追加

電話番号（EmployeePhone）が入力されていない（＝未入力）場合、警告メッセージを表示します。この警告メッセージを表示するように Rules エレメントを定義します。

Rules 定義（追記分のみ）：

```
Msg(”電話番号が未入力です。”) If EmployeePhone.IsEmpty();
```

ヒント：

- ルールは**宣言型**のため、定義順は重要ではない

4-1-3. Employee の Structure 定義を更新

従業員がシステムに入力された日付を記録するための項目属性を追加します。

Employee トランザクションの Structure エレメントに項目を追加します。

追加後のイメージは以下の通りです。（文字色が緑のものは定義済みのものです）

Structure 定義：

名前	タイプ	デスクリプション
EmployeeId	Numeric(4.0)	従業員番号
EmployeeLastName	VarChar(20)	従業員名字
EmployeeFirstName	VarChar(20)	従業員名前
EmployeeAddress	Address, GeneXus	従業員住所
EmployeePhone	Phone, GeneXus	従業員電話番号
EmployeeEmail	Email, GeneXus	従業員メールアドレス
AmusementParkId	Id	パーク番号
AmusementParkName	Name	パーク名
EmployeeAddedDate	Date	従業員登録日

4-1-4. 従業員登録日に関する制御の追加

従業員登録日に対して次の 2 点の制御を追加する必要があります。

①既定値として登録作業当日の日付を入力

②値の変更は許可しない

この制御を行うように Rules エレメントを定義します。

Rules 定義（追記分のみ）：

```
Default(EmployeeAddedDate,&Today);
NoAccept(EmployeeAddedDate);
```

4-1-5. アプリケーションの実行

更新した「Employee」トランザクションに基づく登録画面を実行します。

[F5] キーを押し、アプリケーションを実行します。

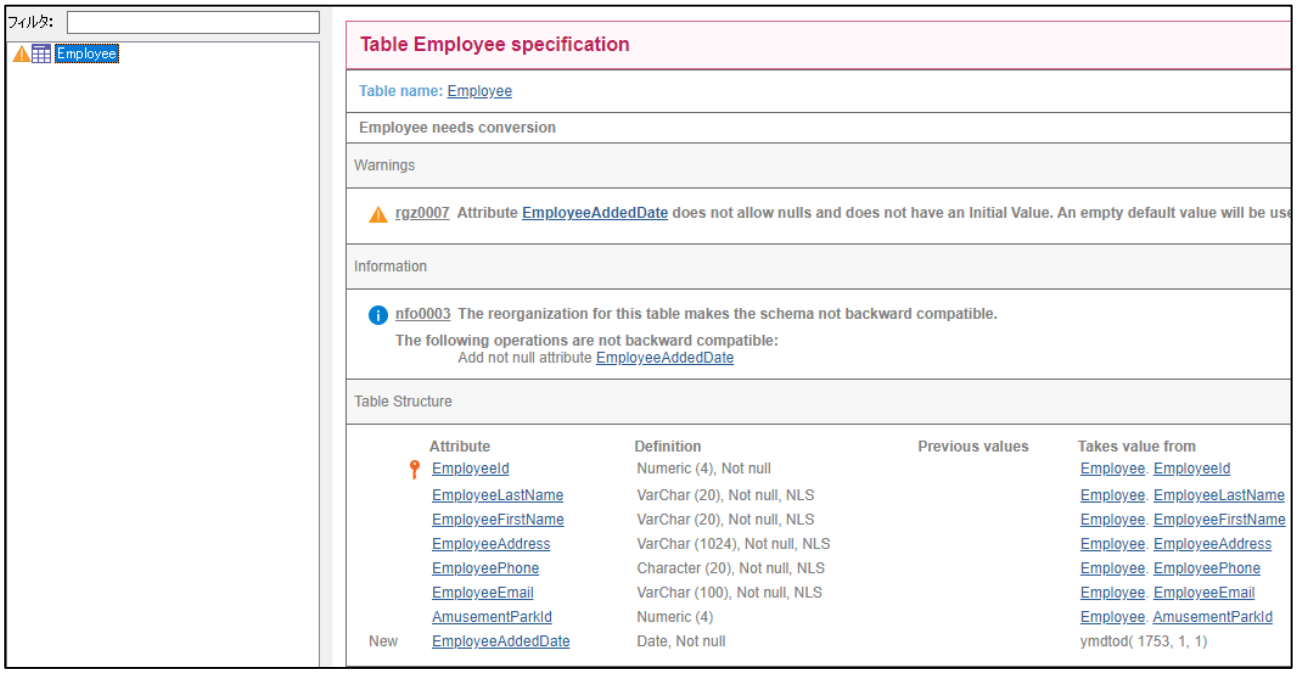
アプリケーションを実行するためには次の操作が必要となります。

①[**影響分析**] ウィンドウでの操作

データベースの再編成が必要であり、再編成内容の詳細を確認できます。

この場合、再編成の内容は 1 つのテーブルのみの再編成です。

このウィンドウで表示される [**再編成**] ボタンをクリックします。



4-1-6. 従業員の登録および更新

表示された [Launchpad] ウィンドウから「Employee」トランザクションにより生成された画面を開きます。

以下の表に沿って既存従業員の更新、新規従業員の登録を行います。

既存従業員の更新（文字色が緑のものは登録済み、セル背景色赤は値の削除）

番号	名字	名前	住所	電話番号	メールアドレス
1	シルバ	ホセ	ペーニャ		s.jose@test.com
2		ジャン	ベルサイユ	22211110000	p.jean@test.com

新規従業員の登録

番号	名字	名前	住所	電話番号	メールアドレス
3	スミス	ジョン	ロサンゼルス		j.smith@test.com
4	山田	花子	東京	00011112222	h.yamada@test.com
5	スミス		ロンドン	33322221111	w.smith@test.com

ヒント：

- 従業員番号「1」、「2」では、「従業員登録日」は未入力のまま編集不可
- 従業員番号「1」、「3」では、「従業員電話番号」が未入力のため、警告が表示
- 従業員番号「2」、「5」では、エラーが発生するため、データ操作（登録/更新）不可

4-2. 「Country」トランザクションに制御を追加

必要な制御のうち最後の 1 つは Country トランザクションで実装する必要があります。

- 「国に追加する都市も自動採番をしたい」

この内容を基に「Country」トランザクションを更新します。

4-2-1. Country の Structure 定義を更新

都市を自動採番するためには最終番号を管理するための項目属性が必要となります。

Country トランザクションの Structure エレメントに項目を追加します。

追加後のイメージは以下の通りです。（文字色が緑のものは定義済みのものです）

Structure 定義：

名前	タイプ	デスクリプション
CountryId	Id	国番号
CountryName	Name	国名
CountryLastLine	Numeric(4.0)	都市最終番号
City	City	都市
CityId	Id	都市番号
CityName	Name	都市名

4-2-2. 都市の自動採番制御の追加

都市の番号（CityId）は第 2 レベルの主キーとなり、実際のテーブルは複合主キーとなるため、AutoNumber プロパティによる自動採番は行えません。

そのため、ルールを利用した「制御の追加」で実装します。

次のように Rules エレメントを定義します。

Rules 定義：

```
Serial(CityId, CountryLastLine, 1);
```

4-2-3. アプリケーションの実行

更新した「Country」トランザクションに基づく登録画面を実行します。

[F5] キーを押し、アプリケーションを実行します。

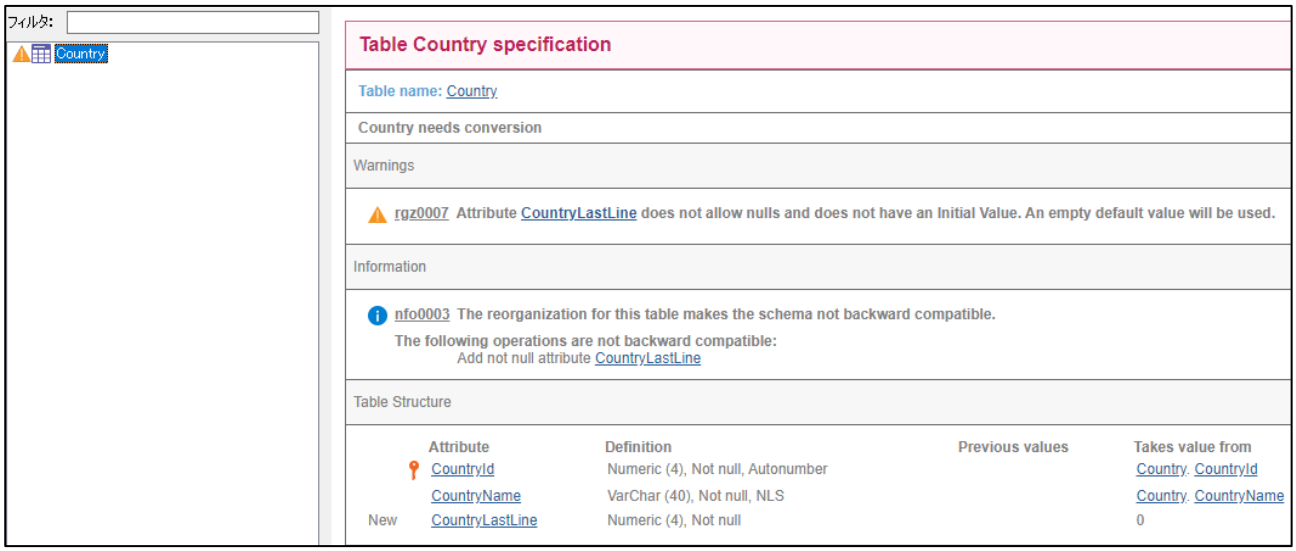
アプリケーションを実行するためには次の操作が必要となります。

①[**影響分析**] ウィンドウでの操作

データベースの再編成が必要であり、再編成内容の詳細を確認できます。

この場合、再編成の内容は 1 つのテーブルのみの再編成です。

このウィンドウで表示される [**再編成**] ボタンをクリックします。



4-2-4. 国の登録

表示された [Launchpad] ウィンドウから「Country」トランザクションにより生成された画面を開きます。

以下の表に沿って新しく国の登録を行います。
値の指定がない項目は未入力のままです。

国名
イギリス

都市名
ロンドン
マンチェスター

ヒント：

- 都市番号の自動採番が有効となり、入力せずに最新の値が自動入力されることが確認可能

5. データを操作するためのインターフェース改善

再度、管理会社の担当者にアプリケーションを確認してもらったところ、現在実装されているすべての画面（従業員、アミューズメントパーク、国、ショー、アトラクション、カテゴリ）から登録できるデータを操作するためにインターフェースの改善を行いたいと次のような要望がありました。

①フィルタリングできるデータの一覧表示

②一覧から新規登録画面の呼び出し

③特定データに対する更新や削除画面の呼び出し

これらの要望は、トランザクションに **Work With for Web** パターンを適用することで実装することが可能です。

実際に適用し、挙動を見ていきましょう

5-1. Work With for Web パターンの適用

これまでに定義してきた「**Employee**」、「**AmusementPark**」、「**Country**」、「**Show**」、「**Game**」、「**Category**」に対し、要望を満たすために Work With for Web パターンを適用します。

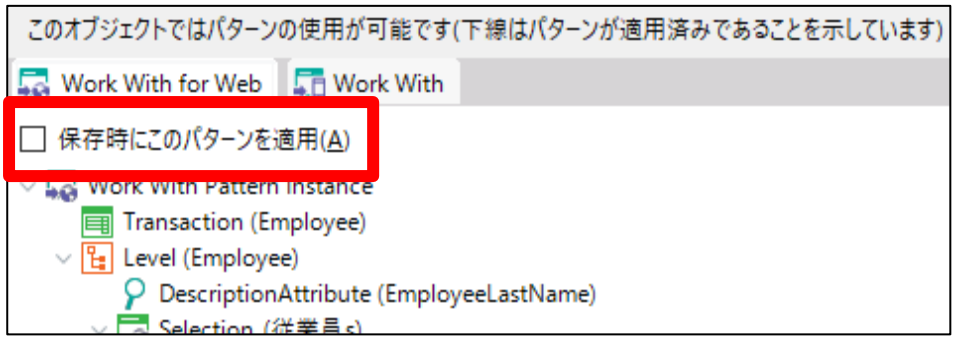
ヒント：

- GeneXus には既定で「Work With」（「for Web」がない）もありますが、これは本コース対象外の機能
- パターンの適用によってナレッジベース内にいくつかのオブジェクトが自動生成される

5-1-1. 初めてのパターン適用

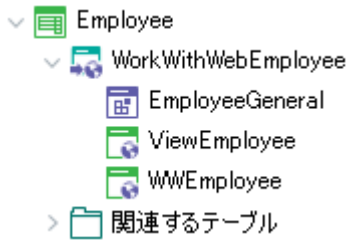
上図のように Employee トランザクションのウィンドウ上部より「Patterns」エレメントを選択します。

表示された下図の画面にて「Work With for Web」が選択されていることを確認し、「保存時にこのパターンを適用」チェックボックスをチェックし、オブジェクトを**保存**します。



ヒント：

- 保存後、[KB エクスプローラー] ウィンドウで Employee トランザクションの子ノードが下図のように追加されることが確認可能



- 作成されるオブジェクトについては後述

5-1-2. 他トランザクションへのパターンの適用

Employee トランザクションと同様に下記トランザクションへパターンを適用します。

- AmusementPark
- Country
- Show
- Game
- Category

パターンを適用するための手順は以下の通りです。

①対象のトランザクションのウィンドウ上部より「Patterns」エレメントを選択

②表示された画面で「Work With for Web」が選択されていることを確認

③「保存時にこのパターンを適用」チェックボックスをチェック

④オブジェクトを保存

ヒント：

- Employee トランザクション同様に [KB エクスプローラー] ウィンドウで子ノードが追加されることが確認可能
- 生成されるオブジェクトの数は、トランザクションの定義に依存

5-1-3. アプリケーションの実行

パターンを適用した各トランザクションの画面を実行します。

[F5] キーを押し、アプリケーションを実行します。

表示された [Launchpad] ウィンドウに今まで表示されていたトランザクション名と同名のリンクは表示されなくなり、代わりにトランザクション名の頭に「WW」が付与されたリンクが表示されることを確認します。

ヒント：

- [Launchpad] ウィンドウはあくまでも開発者用のメニューとなるため、表示されなくなる点は重要ではありません
- [Launchpad] ウィンドウはパラメーターを受け取るオブジェクトは一覧に表示しない
  （パラメーターの受け取りについては後述）

5-2. アプリケーションの挙動確認

パターンを適用したことによってインターフェースの改善が行われました。

この改善で管理会社の要望をかなえられているか確認するために実行画面で次の操作を行い、確認します。

5-2-1. 国名での検索

[Launchpad] ウィンドウから「WWCountry」を開き、国の一覧が表示されることを確認し、画面右上の入力欄に文字を入力し、「国名」で検索が行えることを確認します。

ヒント：

- 既定の検索機能は「前方一致」として実装

5-2-2. 新しい国の登録

「WWCountry」画面の「追加」ボタンをクリックします。

国の登録画面が表示されるため、以下の表に基づきデータを入力し、「実行」ボタンをクリックします。

国名
アルゼンチン **都市名** ブエノスアイレス

ヒント：

- 「実行」ボタンをクリックすると一覧画面へ自動で遷移

5-2-3. 既存の国の更新

「WWCountry」画面に表示される国番号「3」の「中国」が表示されている行の「更新」リンクをクリックします。

国の更新画面が表示されるため、以下の表に基づきデータを更新し、「実行」ボタンをクリックします。

国名
中華人民共和国

5-2-4. 既存の国の削除

「WWCountry」画面に表示される国番号「9」の「アルゼンチン」が表示されている行の「削除」リンクをクリックします。

国の削除画面が表示されるため、「削除」ボタンをクリックします。

ヒント：

- 削除用の画面となるため、すべての項目は読み取り専用で表示

5-2-5. 新しいアミューズメントパークの登録

[Launchpad] ウィンドウから「WWAmusementPark」を開き、以下の表に基づきデータを登録します。

登録するためには「追加」ボタンをクリックし、登録画面を表示します。

（記載のない項目は未入力）

パーク名	国番号
ドリーウッド	4

ヒント：

- 外部参照項目が一覧に表示されている場合、該当データを詳細表示する画面へのリンクが自動生成
- リンク先は対象データを管理するトランザクションに適用されたパターンによって生成された画面

5-3. パターンのカスタマイズ

画面を確認したところ、アミューズメントパークの一覧画面へ追加の要望がありました。これらの要望を満たすため、パターンのインスタンスをカスタマイズしていきます。次の内容に沿って実装を行います。

ヒント：

- パターンのカスタマイズはトランザクションの「Patterns」エレメントで行う
- 「Work With for Web」パターンは生成される画面を階層構造で表現し、これを編集することが可能

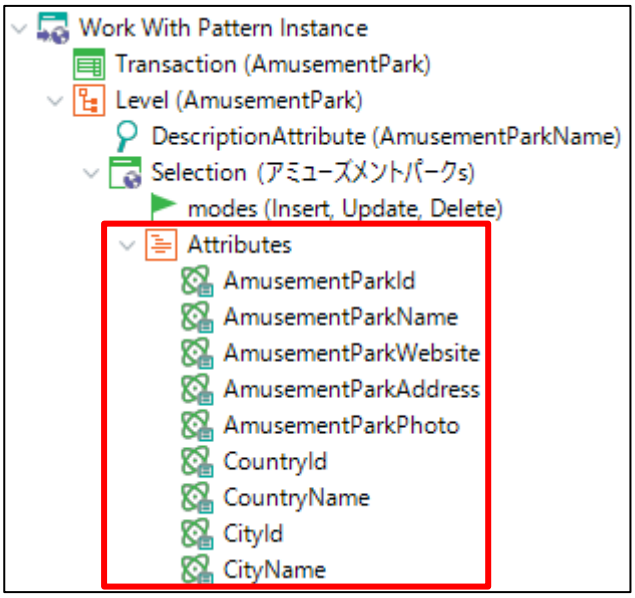
5-3-1. 一覧表示項目の削除

一覧画面の列のうち、外部参照キーとなる「国番号」、「都市番号」は表示する必要がないため、パターンインスタンスの定義から削除します。

以下のように操作を行います。

① AmusementPark トランザクションの「Patterns」エレメントを開く

② 「Selection (アミューズメントパークｓ)」ノード内の「Attributes」ノードが下図のように展開されていることを確認



③ 「Attributes」ノード内から「CountryId」と「CityId」を削除

④ オブジェクトを保存

ヒント：

- 表示列をパターンインスタンスから削除する方法は次の 2 つの方法が利用可能
  1. 対象のノードを右クリック→ [削除] をクリック
  2. 対象のノードを選択し、[Delete] キーを押下
- 実行画面に表示させない方法として「非表示」という方法も選択可能（値を内部的に利用可能）
  →対象のノードの [Visoble] プロパティを [True] に変更
  ただし、今回は内部的にも利用の予定がないため、「削除」を実施

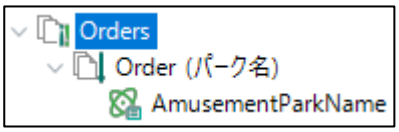
5-3-2. 「国名」を利用した並び替え

一覧画面で「パーク名」による並び替えが行われているが、「国名」による並び替えを行いたい場合もあるため、並び替えを選択できるよう定義を更新します。

以下のように操作を行います。

① AmusementPark トランザクションの「Patterns」エレメントを開く

② 「Selection (アミューズメントパーク s)」ノード内の「Orders」ノード（下図）を右クリックし、[追加] → [Order] をクリック



③ 追加された「Order ()」ノードを選択し、[F4] キーを押下し、[プロパティ] ウィンドウを表示させ、[Name] プロパティに「国名」と入力

④ 「Order (国名)」という表示に変わった追加したノードを右クリックし、[追加] → [Attribute] をクリックし、表示されたダイアログで「CountryName」を選択し、「OK」ボタンをクリック

⑤ 下図のような表示になったことを確認し、オブジェクトを保存

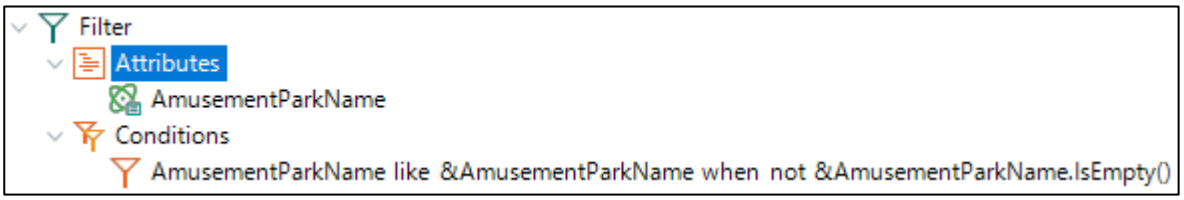
5-3-3. 「国名」を利用したフィルタリング

一覧画面で「パーク名」によるフィルタリングに加え、「国名」のフィルタリングも行いたいため、フィルタリングについての定義を更新します。

以下のように操作を行います。

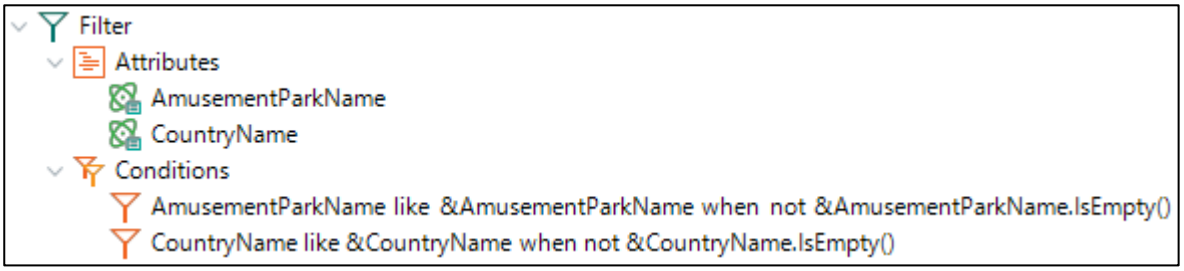
① AmusementPark トランザクションの「Patterns」エレメントを開く

② 「Selection (アミューズメントパーク s)」ノード内の「Filter」ノードに含まれる「Attributes」ノード（下図）を右クリックし、[項目属性を選択...] をクリック



③ 表示されたダイアログで「CountryName」を選択し、「OK」ボタンをクリックし、「対応するフィルター条件を追加しますか？」という確認ダイアログで「Yes」ボタンをクリック

④ 下図のような表示になったことを確認し、オブジェクトを保存



5-3-4. アプリケーションの実行

パターンのカスタマイズが完了したため、実行します。

[F5] キーを押し、アプリケーションを実行します。

表示された [Launchpad] ウィンドウから「WWAmusementPark」を開き、カスタマイズを行った次の点について確認してください。

①一覧に「国番号」、「都市番号」が表示されないこと

②「国名」による並び替えが選択可能であること

③「国名」でフィルタリングが可能であること

ヒント：

- 追加した並び替え、フィルターは「パーク名」の右に表示される「 」アイコンをクリックすることで表示可能
- フィルタリングは「 」アイコンをクリックし、表示された画面内の「国名」をクリックすることで展開

6. 項目属性の別名参照

このアプリケーションにおける要件として、管理会社の担当者が伝え忘れていた点があったと連絡がありました。この内容は次の通りです。

> 「管理するアトラクションのうち、**修理**が必要なアトラクションの状態も記録する必要があり、修理を担当する**技術者**を管理したいと考えています。この技術者は１つの修理に関して正副の２名体制で記録が必要です」

この要望をかなえるための実装を進めます。

6-1. 「Technician」トランザクション

管理会社の担当者に、技術者についてどのようなデータを記録するのかを尋ねました。

回答は次の通りです：

> **技術者管理番号** (Id)**、名字**と**名前** (それぞれ最大 20 文字)、**電話番号**、出身**国、都市**です。

この情報を基に「**Technician**」トランザクションを作成します。

6-1-1. Technician トランザクションの作成

下記の内容でトランザクションを作成し、Structure エレメントを定義します。

トランザクション定義：

名前	Technician
デスクリプション	技術者

Structure 定義：

名前	タイプ	デスクリプション
TechnicianId	Id	技術者番号
TechnicianLastName	VarChar(20)	技術者名字
TechnicianFirstName	VarChar(20)	技術者名前
TechnicianPhone	Phone, GeneXus	技術者電話番号
CountryId	Id	国番号
CountryName	Name	国名
CityId	Id	都市番号
CityName	Name	都市名

ヒント：

- 「TechnicianLastName」、「TechnicianFirstName」は定義した際にタイプが自動的に「Name」ドメインが指定されますが、要望されている桁数と一致しないため、データタイプを指定します。

6-1-2. アプリケーションの実行

定義した「Technician」トランザクションに基づく登録画面を実行します。

[F5] キーを押し、アプリケーションを**実行**します。

アプリケーションを実行するためには次の操作が必要となります。

①[**影響分析**] ウィンドウでの操作

データベースの再編成が必要であり、再編成内容の詳細を確認できます。

この場合、再編成の内容は 1 つのテーブルの作成です。

このウィンドウで表示される [**再編成**] ボタンをクリックします。

6-1-3. 技術者の登録

表示された [Launchpad] ウィンドウに「Technician」トランザクションにより生成された画面を開くためのリンクが追加されました。

このリンクをクリックし、技術者の登録画面を開きます。

技術者を以下の表（列名内「技術者」を割愛）に沿って登録します。

名字	名前	電話番号	国番号	都市番号
シルヴァ	ジョアン	44455556666	1	1
ジョンソン	ジェームズ	99955557777	4	1
デュポン	ジャン	22299994444	2	1
フェレイラ	マリア	88800002222	1	3
ブラウン	ジェニファー	77744449999	4	2

6-2. 「Repair」トランザクション

管理会社の担当者に修理が必要な状態のアトラクションについてどのようなデータを記録するのかを尋ねました。回答は次の通りです：

> **修理番号** (Id)**、修理開始日**（日付）、**修理日数** (数値 4 桁)、対象**アトラクション**、**費用** (数値整数 5 桁＋小数点以下 2 桁)、担当**技術者** (正および副) です。

また、今後の実装において、費用の項目は数値で、整数 5 桁、小数点以下 2 桁を共通の定義としたいと要望がありました。

この情報を基に「**Repair**」トランザクションを作成します。

6-2-1. ドメインの定義

費用項目の共通データタイプを準備するため、以下の手順で「ドメイン」を定義します。

ドメインウィンドウを開き、以下の内容で定義します。

名前	タイプ	モジュール	デスクリプション
Cost	Numeric(8.2)	Root Module	Cost

ヒント：

- Numeric タイプの桁数定義は「.」（ピリオド）の左に小数点など数値として入力するすべての桁数における最大桁数（文字数）を指定（整数 5 桁＋小数点 1 桁（文字）＋小数点以下 2 桁＝8）

6-2-2. サブタイプグループの定義

修理の登録において Technician トランザクションで登録したデータを 2 種類の異なる意味合いで関連付ける必要があります（担当者の正と副）。

これを実装するため、下記の内容でサブタイプグループオブジェクトを作成し、サブタイプ項目属性を定義します。

サブタイプグループ定義：

名前	SubstituteTechnician
デスクリプション	副技術者

Group Structure 定義：

サブタイプ	デスクリプション	スーパータイプ
SubstituteTechnicianId	副技術者番号	TechnicianId
▪ SubstituteTechnicianFirstName	副技術者名字	TechnicianFirstName

ヒント：

- サブタイプグループオブジェクトの作成は「新規オブジェクト」ダイアログで、[カテゴリ] から「データ管理」を選択し、[タイプを選択] 内で「Subtype Group」を選択
- 「Group Structure」内の 4 列目の「デスクリプション」は「スーパータイプ」列で指定した項目属性に基づくデスクリプションの表示となるため、読み取り専用

6-2-3. Repair トランザクションの作成

下記の内容でトランザクションを作成し、Structure エレメントを定義します。

トランザクション定義：

名前	Repair
デスクリプション	修理

Structure 定義：

名前	タイプ	デスクリプション
RepairId	Id	修理番号
RepairFromDate	Date	修理開始日
RepairDaysQuantity	Numeric(4.0)	修理日数
GameId	Id	アトラクション番号
GameName	Name	アトラクション名
RepairCost	Cost	修理費用
TechnicianId	Id	技術者番号
TechnicianFirstName	VarChar(20)	技術者名字
SubstituteTechnicianId	Id	副技術者番号
SubstituteTechnicianFirstName	VarChar(20)	副技術者名字

6-2-4. アプリケーションの実行

定義した「Repair」トランザクションに基づく登録画面を実行します。

[F5] キーを押し、アプリケーションを**実行**します。

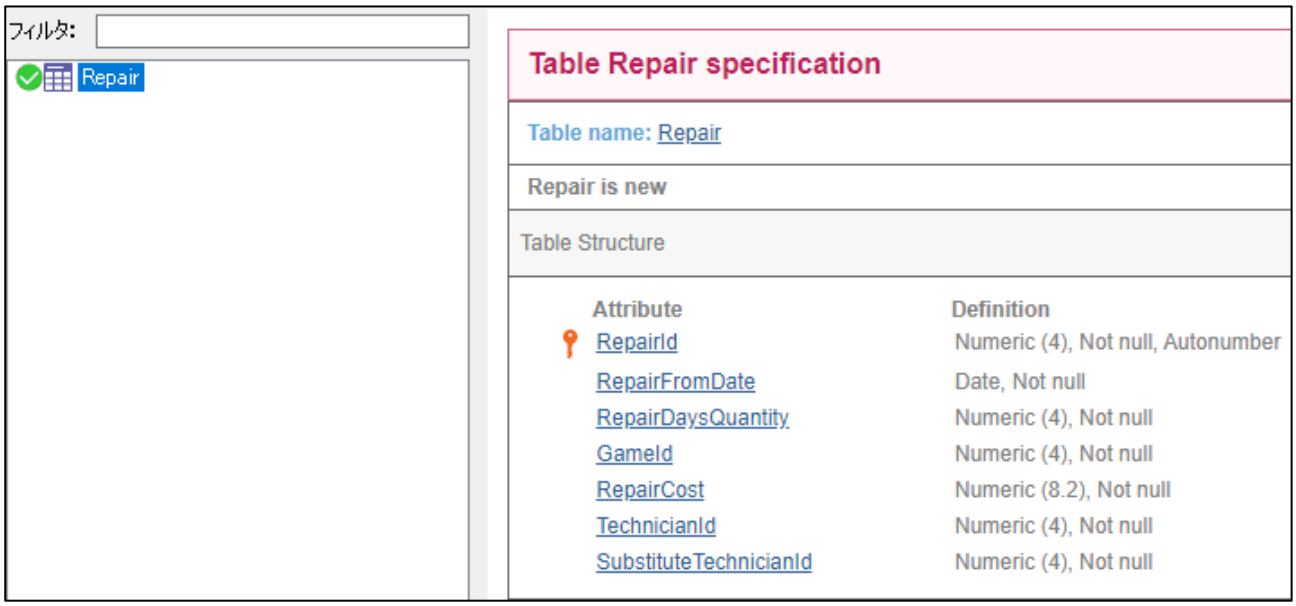
アプリケーションを実行するためには次の操作が必要となります。

①[**影響分析**] ウィンドウでの操作

データベースの再編成が必要であり、再編成内容の詳細を確認できます。

この場合、再編成の内容は 1 つのテーブルの作成です。

このウィンドウで表示される [**再編成**] ボタンをクリックします。



6-2-5. 修理の登録

表示された [Launchpad] ウィンドウに「Repair」トランザクションにより生成された画面を開くためのリンクが追加されました。

このリンクをクリックし、修理の登録画面を開きます。

修理を以下の表に沿って登録します。

修理開始日	修理日数	アトラクション番号	修理費用	技術者番号	副技術者番号
今日の日付	1	2	100	1	4
２日後の日付	15	3	400	4	1
1 日前の日付	30	5	800	3	5
今日の日付	3	1	150	1	4
５日後の日付	7	6	300	2	3

6-3. 「Repair」トランザクションに制御を追加

実行したアプリケーションを操作したところ、正技術者と副技術者に同じ技術者を指定可能であることが確認できたため、これを制御するため、「Repair」トランザクションを更新します。

6-3-1. 入力値比較による登録制御の追加

正技術者（TechnicianId）と副技術者（SubstituteTechnicianId）に指定された値が同一の場合、エラーが発生し、登録できないように Rules エレメントを定義します。

Rules 定義：

```
Error(”同じ技術者を指定することはできません。”) If TechnicianId =
SubstituteTechnicianId;
```

6-3-2. アプリケーションの実行

制御を追加した「Repair」トランザクションを実行します。

[F5] キーを押し、アプリケーションを実行します。

表示された [Launchpad] ウィンドウから「Repair」トランザクションにより生成された画面を開きます。

新規の修理を以下の表に沿って入力し、登録できるか確認します。

修理開始日	修理日数	アトラクション番号	修理費用	技術者番号	副技術者番号
今日の日付	20	4	600	2	2

7. 式の追加

出来上がったアプリケーションを確認してもらったところ、修理を記録する中で不足している項目があることに気が付いたと管理会社の担当者から連絡がありました。
不足項目としては以下の通りです。

①修理見込み日（修理開始日に修理日数を加算した日付）
②割引率（修理費用を割り引きするパーセンテージ）
③最終費用（割引率を適用した最終的な費用）

また、割引率を管理するにあたり、今後もアプリケーション内にパーセンテージの値を管理する項目が発生する可能性があるため、あらかじめ共通で 3 桁の数値が管理できるようにデータタイプも追加したいという要望がありました。

この要望をかなえるための実装を進めます。

7-1. 「Repair」トランザクションの更新

修理に追加が必要な項目とその内容について確認し、以下の通り回答がありました。

①修理見込み日（日付）
　→修理開始日（RepairFromDate）に修理日数（RepairDaysQuantity）を加算

②割引率（数値 3 桁）
　→利用者による手入力

③最終費用（整数 5 桁＋小数点以下 2 桁）
　→修理費用（RepairCost）から「割引率」に基づく費用を減算

この内容を基に「Repair」トランザクションを更新します。

7-1-1. Repair の Structure 定義を更新

Repair トランザクションの Structure エレメントに項目を追加します。

追加後のイメージは以下の通りです。（文字色が緑のものは定義済みのものです）

Structure 定義：

名前	タイプ	デスクリプション
RepairId	Id	修理番号
RepairFromDate	Date	修理開始日
RepairDaysQuantity	Numeric(4.0)	修理日数
RepairToDate	Date	修理見込み日
GameId	Id	アトラクション番号
GameName	Name	アトラクション名
RepairCost	Cost	修理費用
RepairDiscountPercentage	Percentage = Numeric(3.0)	修理割引率
RepairFinalCost	Cost	修理最終費用
TechnicianId	Id	技術者番号
TechnicianFirstName	VarChar(20)	技術者名字
SubstituteTechnicianId	Id	副技術者番号
SubstituteTechnicianFirstName	VarChar(20)	副技術者名字

7-1-2. グローバル式の定義

追加した項目のうち、2 つは自動計算する必要があるため、Structure 上の「式」列に計算式を記述する必要があります。

以下の表に基づき、対象の項目属性の式を追加します。

名前	式
RepairToDate	`RepairFromDate.AddDays(RepairDaysQuantity)`
RepairFinalCost	`RepairCost * (1 - RepairDiscountPercentage / 100)`

ヒント：

- 式を編集する場合、対象の項目属性の「式」列を選択した際に表示される「...」アイコンをクリックすることで、「式エディター」ダイアログが表示され、「Rules」エレメントのように「Ctrl + Space」キーで入力候補が利用可能
- AddDays メソッドは指定された数値分日付を加算
- 式が定義された項目属性はアイコンが「」に変化

7-2. 実装結果の確認

定義した内容に沿ってアプリケーションを実行した際に何が起きるか確認します。

7-2-1. アプリケーションの実行

更新した「Repair」トランザクションに基づく登録画面を実行します。

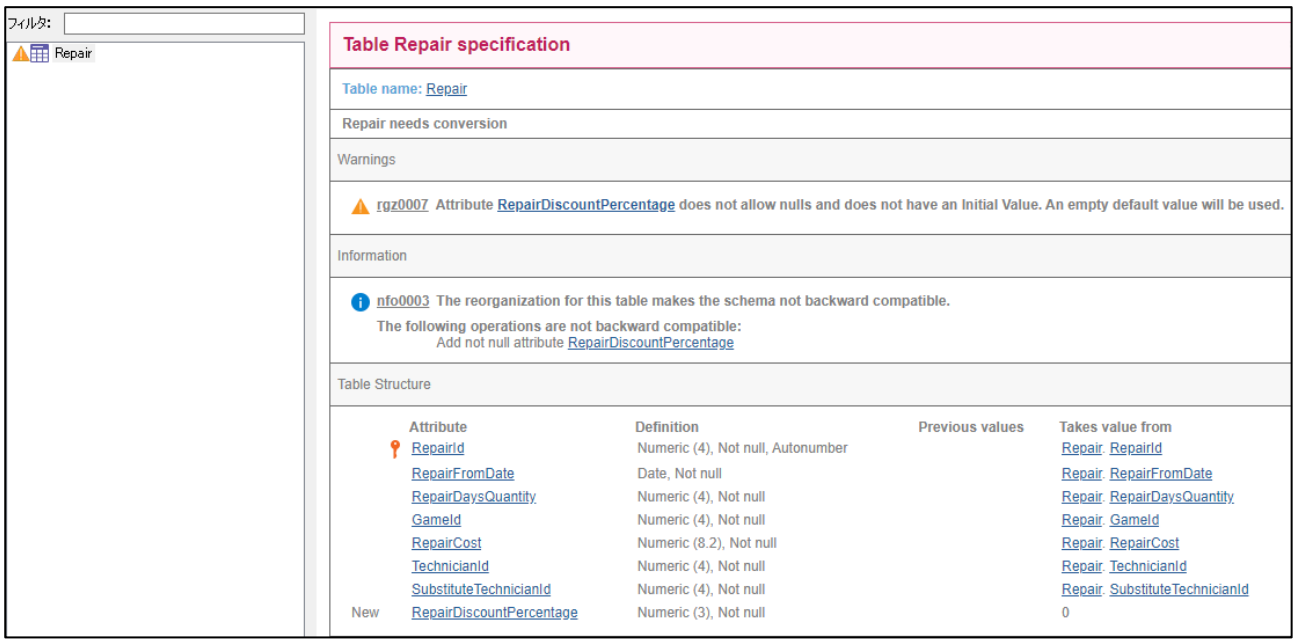
[F5] キーを押し、アプリケーションを**実行**します。

アプリケーションを実行するためには次の操作が必要となります。

①[**影響分析**] ウィンドウでの操作

データベースの再編成が必要であり、再編成内容の詳細を確認できます。

この場合、再編成の内容は 1 つのテーブルの再編成です。



ヒント：

- 割引率を入力するための「RepairDiscountPercentage」は物理項目として作成
- グローバル式が定義された「RepairToDate」、「RepairFinalCost」は論理項目となるため、作成されない

7-2-2. 修理の更新

表示された [Launchpad] ウィンドウから「Repair」トランザクションにより生成された画面を開きます。

既存の修理を以下の表に沿って更新し、修理見込み日、修理最終費用の値を確認します。更新不要の項目は表に含めていません。（文字色が緑のものは定義済みのものです）

修理番号	修理割引率
1	10
2	20
3	15
4	20
5	50

8. 重複登録の制御

アプリケーションをテストする中で一部のデータに重複する名前が指定できることを管理会社の担当者より指摘されました。

重複する名前でデータを登録したくない画面は国とアミューズメントパークです。

この要望をかなえるための実装を進めます。

8-1. ユーザーインデックスの追加

重複したデータ登録をさせないためには、GeneXus では「一意」タイプのユーザーインデックスを利用することで簡単に実装できます。

トランザクションオブジェクトに基づき自動で作成されたテーブルオブジェクトを開き、この要望を実装します。

ヒント：

- 自動で作成される**テーブルオブジェクト**は**トランザクションオブジェクト**と**同名**になるため、本資料では**明確な区別**を行うため、**すべて大文字**で標記しています。

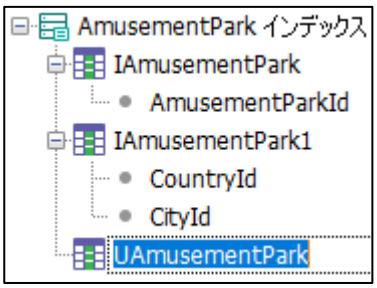
8-1-1. AMUSEMENTPARK テーブルの更新

AMUSEMENTPARK テーブルにユーザーインデックスを追加するため、以下の操作を行います。

① **[KB エクスプローラー]** ウィンドウで **AmusementPark** トランザクションの子ノード「**関連するテーブル**」を展開（下図の通り）し、**AMUSEMENTPARK** テーブルオブジェクトを開く

② AMUSEMENTPARK テーブルオブジェクトの「**Indexes**」エレメントを開く

③ 「Indexes」エレメントが開いたところで **[Enter]** キーを押下し**ユーザーインデックスノード**を追加（下図の状態）



④ ユーザーインデックス「UAmusementPark」の編集状態を **[Enter]** キーで確定

⑤ **[Enter]** キーを押下し、「UAmusementPark」の**子ノード**を追加し、項目属性「**AmusementParkName**」を指定

⑥ 「UAmusementPark」の **2 列目**（列のタイトルは「順序」）の値を「重複」から「**一意**」へ変更

⑦ AMUSEMENTPARK テーブルオブジェクトを**保存**

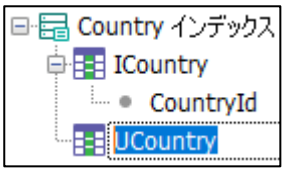
8-1-2. COUNTRY テーブルの更新

COUNTRY テーブルにユーザーインデックスを追加するため、以下の操作を行います。

① **[KB エクスプローラー]** ウィンドウで **Country** トランザクションの子ノード「**関連するテーブル**」を展開（下図の通り）し、**COUNTRY** テーブルオブジェクトを開く

② COUNTRY テーブルオブジェクトの「**Indexes**」エレメントを開く

③ 「Indexes」エレメントが開いたところで **[Enter]** キーを押下し**ユーザーインデックスノード**を追加（下図の状態）



④ ユーザーインデックス「UCountry」の編集状態を **[Enter]** キーで確定

⑤ **[Enter]** キーを押下し、「UCountry」の**子ノード**を追加し、項目属性「**CountryName**」を指定

⑥ 「UCountry」の **2 列目**（列のタイトルは「順序」）の値を「重複」から「**一意**」へ変更

⑦ COUNTRY テーブルオブジェクトを**保存**

8-2. 実装結果の確認

定義した内容に沿ってアプリケーションを実行した際に何が起きるか確認します。

8-2-1. アプリケーションの実行

ユーザーインデックスが追加された国、アミューズメントパークを実行します。

F5 キーを押し、アプリケーションを実行します。

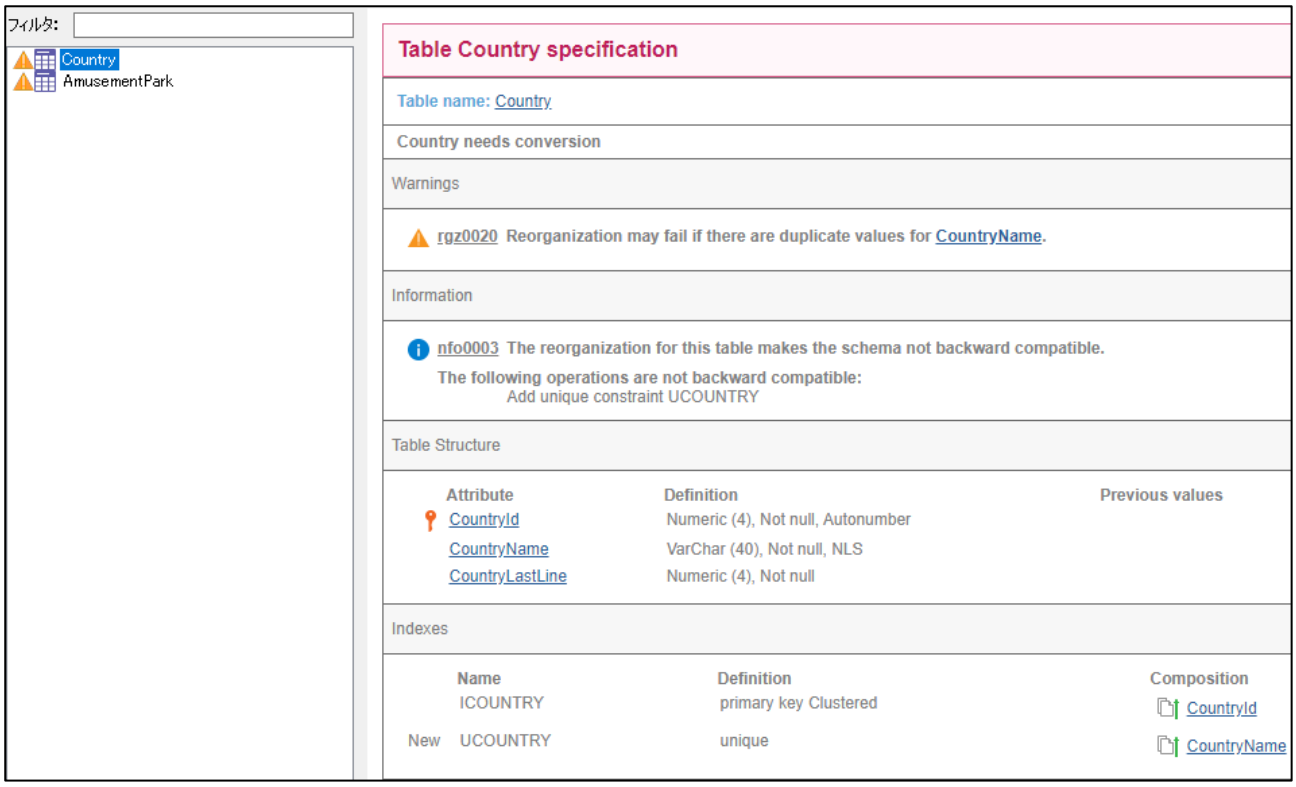
アプリケーションを実行するためには次の操作が必要となります。

①[**影響分析**] ウィンドウでの操作

データベースの再編成が必要であり、再編成内容の詳細を確認できます。

この場合、再編成の内容は 2 つのテーブルの再編成です。

このウィンドウで表示される [**再編成**] ボタンをクリックします。



フィルタ:

- Country
- AmusementPark

Table AmusementPark specification

Table name: AmusementPark

AmusementPark needs conversion

Warnings

rgz0020 Reorganization may fail if there are duplicate values for AmusementParkName.

Information

nfo0003 The reorganization for this table makes the schema not backward compatible.
The following operations are not backward compatible:
Add unique constraint UAMUSEMENTPARK

Table Structure

Attribute	Definition	Previous values
AmusementParkId	Numeric (4), Not null, Autonumber	
AmusementParkName	VarChar (40), Not null, NLS	
AmusementParkWebsite	VarChar (60), Not null, NLS	
AmusementParkAddress	VarChar (1024), Not null, NLS	
AmusementParkPhoto	Image, Not null	
AmusementParkPhoto_GXI	VarChar (2048), Not null	
CountryId	Numeric (4), Not null	
CityId	Numeric (4)	

Indexes

	Name	Definition	Composition
	IAMUSEMENTPARK	primary key Clustered	AmusementParkId
	IAMUSEMENTPARK1	duplicate	CountryId CityId
New	UAMUSEMENTPARK	unique	AmusementParkName

ヒント：

- 「Indexes」という表示領域に定義したユーザーインデックスは表示
- 重複を制御する定義を追加したため、既存のレコードで重複したデータがある場合、再編成が失敗することを警告「rgz0020」として表示

8-2-2. アミューズメントパークの登録

表示された [Launchpad] ウィンドウから「WWAmusementPark」を開き、「追加」ボタンをクリックし以下の表に基づきデータを入力し、登録できるか確認します。
（記載のない項目は未入力）

パーク名	国番号
テラ・ボタニカ	4

8-2-3. 国の登録

表示された [Launchpad] ウィンドウから「WWCountry」を開き、「追加」ボタンをクリックし以下の表に基づきデータを入力し、登録できるか確認します。
（都市の追加は行いません）

国名
アメリカ

9. レポート出力

必要な情報をまとめたレポートを PDF 形式で出力する機能の追加要望が新たに発生しました。要望を受けたレポートについては以下の通りです：

①パーク名の昇順で表示されるアミューズメントパークの一覧
②カテゴリごとのアトラクションの一覧
③特定のアミューズメントパークで実施される特定日以降のショーの一覧
④アミューズメントパークの数とともに表示する国名の一覧

これらのレポートを出力するための実装を進めます。

9-1. アミューズメントパークの一覧

パーク名の昇順で表示されるアミューズメントパークの一覧を実装します。

この一覧では以下の内容を表示してほしいとリクエストがありました。

- **ヘッダー部：**
  - 画像
  - タイトル
  - 一覧表示する項目のタイトル
- **表示項目部：**
  - パーク番号
  - パーク名
  - パーク写真

この情報を基に「**AmusementParkList**」プロシージャーを作成します。

9-1-1. 画像の取り込み

このレポートのヘッダー部に表示する画像をナレッジベース内で管理するため、イメージオブジェクトを作成するため、以下の操作を行います。

① 「**新規オブジェクト**」ダイアログで、[カテゴリ] から「**リソース**」を選択し、[タイプを選択] 内で「**Image**」を選択し、[名前] を「**ParkImage**」、[デスクリプション] を「Park Image」（既定値のまま）とし、[**作成**] をクリック

② 「**新規画像ウィザード**」ダイアログで「**ファイルから画像を作成**」（既定値）を選択し、[**Next**] をクリック

③ 「**画像を選択**」エクスプローラーで本資料同梱画像のうち「**ParkImage.png**」を選択し、[**開く**] をクリック

④ 「新規画像ウィザード」ダイアログの「名前」と「ファイル」に値が入力されていることを確認し、[**Finish**] をクリック

⑤ 表示された画像オブジェクトを**保存**

9-1-2. プロシージャーオブジェクトの作成

PDF 形式でレポートを出力するため、下記の内容でプロシージャーオブジェクトを作成し、一部のプロパティを設定します。

プロシージャー定義：

名前	AmusementParkList
デスクリプション	アミューズメントパークの一覧

プロパティ設定：

プロパティ	**値**
Main program	True
Call protocol	HTTP

ヒント：

- プロシージャーオブジェクトの作成は「新規オブジェクト」ダイアログで、[カテゴリ] から「データ管理」を選択し、[タイプを選択] 内で「Procedure」を選択
- プロパティを設定するための「プロパティ」ウィンドウは、オブジェクトウィンドウのタブ（下図）を右クリックし、[プロパティ] をクリック

9-1-3. Rules エレメントの定義

レポートを出力するプロシージャーオブジェクトとして定義するため、以下のルールを Rules エレメントで定義します。

Rules 定義：

```
Output_file("AmusementParkList","PDF");
```

9-1-4. Layout エレメントの定義

出力する内容をデザインするため、Layout エレメントで以下の操作を行います。

ヘッダー部

① 既定で作成されているプリントブロック「**printBlock1**」の [**Name**] プロパティを [**PBHeader**] に変更

② ツールボックスより「**画像**」コントロールを「PBHeader」プリントブロックへドラッグアンドドロップ

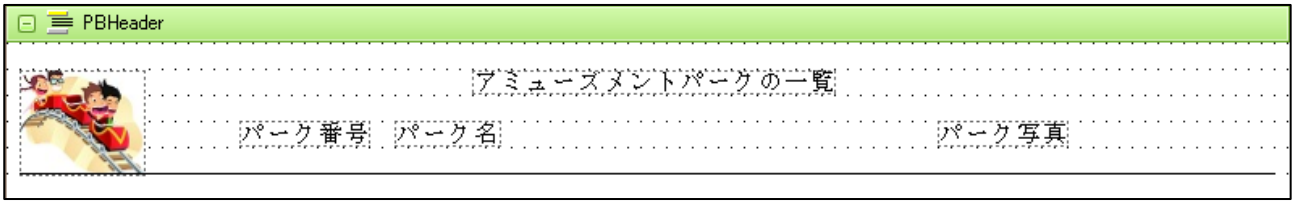
③ 表示された「オブジェクトを選択」ダイアログで **ParkImage** を選択し、[OK] をクリック

④ 一覧のタイトルを表示するため、ツールボックスより「**テキストブロック**」コントロールを「PBHeader」プリントブロックへドラッグアンドドロップし、コントロールをダブルクリックし、テキストを「**アミューズメントパークの一覧**」に変更

⑤ 一覧表示する項目のタイトルを表示するため、ツールボックスより「**テキストブロック**」コントロールを「PBHeader」プリントブロックへ **3 つ**ドラッグアンドドロップし、テキストがそれぞれ「**パーク番号**」、「**パーク名**」、「**パーク写真**」となるように変更

⑥ ヘッダー部とデータの表示範囲の境界をわかりやすくするため、ツールボックスより「**線**」コントロールを「PBHeader」プリントブロックへドラッグアンドドロップ

⑦ 配置したコントロールを下図のように位置を調整（完全一致の必要はありません）



ヒント：

- プロパティを操作するためには、[プロパティ] ウィンドウを表示する必要があり、以下の 2 つの方法が利用可能
  1. 対象のプリントブロックを右クリックし、[プロパティ] をクリック
  2. 対象のプリントブロック選択状態（プリントブロックのタイトル部分の色が水色）で [F4] キー
- [ツールボックス] ウィンドウは既定で画面右側のウィンドウ群に統合され、表示されていますが、見つからない場合、メニューバーの [表示] > [その他のツールウィンドウ] > [ツールボックス] のクリックで表示可能
- 「テキストブロック」コントロールのテキストを直接変更した場合、「テキストブロック」コントロール以外をクリックすることで変更が確定
- 「テキストブロック」コントロールのテキストは、テキストブロックの [Text] プロパティからも変更可能
- プリントブロックに配置されたコントロールはドラッグアンドドロップで任意の場所へ移動可能
- プリントブロックに配置されたコントロールはコントロール選択時に表示される「□」をドラッグすることでサイズの変更が可能
- プリントブロックの高さはプリントブロックの境界にマウスがある場合（カーソルが変化）、ドラッグすることで変更可能であり、同様に [Height] プロパティを利用して数値で指定も可能

表示項目部

① 表示項目部を定義するため、「PBHeader」を**右クリック**し、「**プリントブロックを挿入**」をクリックし、追加されたプリントブロックの [**Name**] プロパティを [**PBAmusementPark**] に変更

② ツールボックスより「**項目属性/変数**」コントロールを「**PBAmusementPark**」プリントブロックへドラッグアンドドロップ

③ 表示された「項目属性／変数を挿入」ダイアログで **AmusementParkId** を選択し、[OK] をクリック

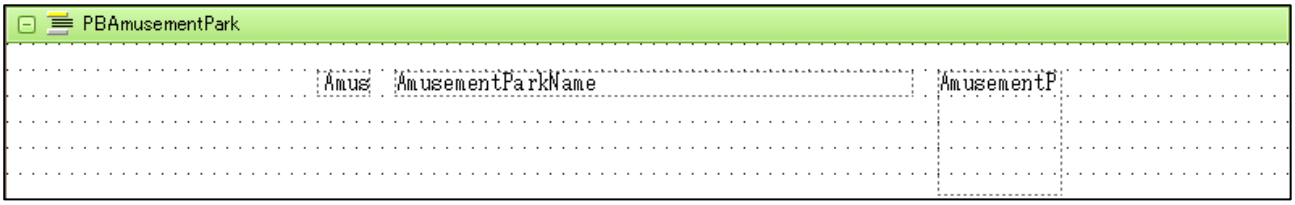
④ ツールボックスより「**項目属性/変数**」コントロールを「**PBAmusementPark**」プリントブロックへドラッグアンドドロップ

⑤ 表示された「項目属性／変数を挿入」ダイアログで **AmusementParkName** を選択し、[OK] をクリック

⑥ ツールボックスより「**項目属性/変数**」コントロールを「**PBAmusementPark**」プリントブロックへドラッグアンドドロップ

⑦ 表示された「項目属性／変数を挿入」ダイアログで **AmusementParkPhoto** を選択し、[OK] をクリック

⑧ 配置したコントロールを下図のように位置を調整（完全一致の必要はありません）



ヒント：

- 「プリントブロックを挿入」を実施した際に既存のプリントブロックよりも上にプリントブロックが追加された場合も今後の実装に影響はありません
- デザイン定義の可視性を優先するため、プリントブロックの表示順を変更したい場合、対象のプリントブロックを右クリックし、「上位へ移動」または「下位へ移動」をクリックすることで変更可能
- 「項目属性／変数を挿入」ダイアログでは複数選択は不可

9-1-5. Source エレメントの定義

出力する処理を定義するため、Source エレメントで以下の定義をします。

Source 定義：

```
Print PBHeader
For each AmusementPark
      Order AmusementParkName
      Print PBAmusementPark
Endfor
```

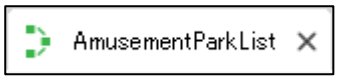
ヒント：

- プリントブロックや項目属性は「Ctrl + Space」キー押下で候補から選択可能
- Order 節を記述した場合、指定した項目属性で昇順の並び替えが実施

9-1-6. レポートの実行

作成したプロシージャーオブジェクトを実行し、PDF 形式のレポートを確認します。

「**AmusementParkList**」のオブジェクトウィンドウの**タブ（下図）**を右クリックし、「**実行**」をクリック



ヒント：

- 実行時に表示が更新された「ナビゲーション表示」ウィンドウでは、このデータ出力を実現するために内部的にどのようなデータ抽出が行われているか確認可能

9-2. 「ブラジル」のアミューズメントパークの一覧

作成したアミューズメントパークの一覧を管理会社の担当者が確認したところ、同様のレイアウトで「**ブラジル**」だけを対象としたアミューズメントパークの一覧も必要とリクエストがありました。

この情報を基に「**AmusementParkListBrazil**」プロシージャーを作成します。

9-2-1. オブジェクトの複製

ここで必要となるオブジェクトは「**AmusementParkList**」とほぼ同一となるため、作業を短縮するため、オブジェクトを複製し、作業を開始します。

「**AmusementParkList**」のオブジェクトウィンドウのタブを右クリックし、[**名前を付けて保存...**] をクリックし、表示された「新規オブジェクト」ダイアログで以下のように変更し、作成します。

プロシージャー定義：

名前	AmusementParkListBrazil
デスクリプション	アミューズメントパークの一覧（ブラジル）

9-2-2. Source エレメントの更新

Source エレメントの定義を「ブラジル」のみが対象となるように更新します。

（文字色が緑のものは定義済みのものです）

Source 定義：

```
Print PBHeader
For each AmusementPark
      Order AmusementParkName
      Where CountryName = ”ブラジル”
      Print PBAmusementPark
Endfor
```

ヒント：

- 条件は、国名が「ブラジル」と一致するレコードと指定
- もし、該当するレコードがない場合、データの一覧として、表示されるものはない

9-2-3. レポートの実行

作成したプロシージャーオブジェクトを実行し、PDF 形式のレポートを確認します。

「AmusementParkListBrazil」のオブジェクトウィンドウのタブを右クリックし、「実行」をクリック

ヒント：

- 実行時に表示が更新された「ナビゲーション表示」ウィンドウでは、データのフィルタリングが行われていること、フィルタリングに利用された集計関数の抽出に関する情報も確認可能

9-3. アトラクションの一覧

カテゴリごとに表示されるアトラクションの一覧を実装します。

この一覧では以下の内容を表示してほしいとリクエストがありました。

- **ヘッダー部：**
  - 画像
  - タイトル
- **カテゴリ表示部**
  - カテゴリ名（「カテゴリ：」という文字列に続いて表示）
  - 一覧表示するアトラクション項目のタイトル
- **アトラクション表示部：**
  - アトラクション名
  - パーク名
  - アトラクション写真

この情報を基に「**GameList**」プロシージャーを作成します。

9-3-1. 画像の取り込み

このレポートのヘッダー部に表示する画像をナレッジベース内で管理するため、イメージオブジェクトを作成するため、以下の操作を行います。

① 「新規オブジェクト」ダイアログで、[カテゴリ] から「リソース」を選択し、[タイプを選択] 内で「Image」を選択し、[名前] を「**AttractionsImage**」、[デスクリプション] を「Attractions Image」（既定値のまま）とし、[作成] をクリック

② 「新規画像ウィザード」ダイアログで「ファイルから画像を作成」（既定値）を選択し、[Next] をクリック

③ 「画像を選択」エクスプローラーで本資料同梱画像のうち「**Attractions.png**」を選択し、[開く] をクリック

④ 「新規画像ウィザード」ダイアログの「名前」と「ファイル」に値が入力されていることを確認し、[Finish] をクリック

⑤ 表示された画像オブジェクトを**保存**

9-3-2. プロシージャーオブジェクトの作成

PDF 形式でレポートを出力するため、下記の内容でプロシージャーオブジェクトを作成し、一部のプロパティを設定します。

プロシージャー定義：

名前	GameList
デスクリプション	アトラクションの一覧

プロパティ設定：

プロパティ	**値**
Main program	True
Call protocol	HTTP

9-3-3. Rules エレメントの定義

レポートを出力するプロシージャーオブジェクトとして定義するため、以下のルールを Rules エレメントで定義します。

Rules 定義：

```
Output_file("GameList","PDF");
```

9-3-4. Layout エレメントの定義

出力する内容をデザインするため、Layout エレメントで以下の操作を行います。

ヘッダー部

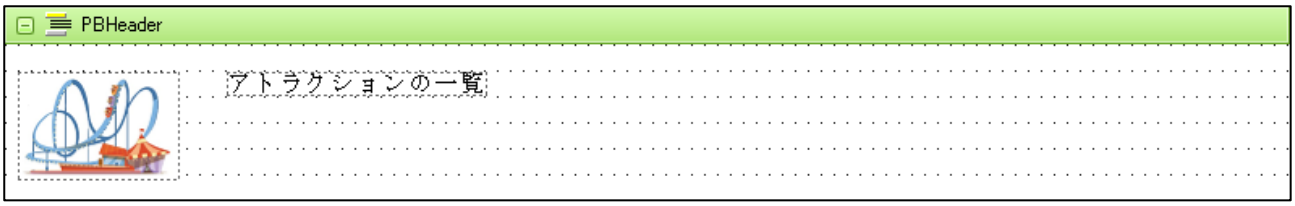
① 既定で作成されているプリントブロック「**printBlock1**」の [**Name**] プロパティを [**PBHeader**] に変更

② ツールボックスより「**画像**」コントロールを「PBHeader」プリントブロックへドラッグアンドドロップ

③ 表示された「オブジェクトを選択」ダイアログで **AttractionsImage** を選択し、[OK] をクリック

④ 一覧のタイトルを表示するため、ツールボックスより「**テキストブロック**」コントロールを「PBHeader」プリントブロックへドラッグアンドドロップし、コントロールをダブルクリックし、テキストを「**アトラクションの一覧**」に変更

⑤ 配置したコントロールを下図のように位置を調整（完全一致の必要はありません）



カテゴリ表示部

① 表示項目部を定義するため、「PBHeader」を**右クリック**し、「**プリントブロックを挿入**」をクリックし、追加されたプリントブロックの [**Name**] プロパティを [**PBCategory**] に変更

② ツールボックスより「**テキストブロック**」コントロールを「**PBCategory**」プリントブロックへドラッグアンドドロップし、コントロールをダブルクリックし、テキストを「**カテゴリ：**」に変更

③ アトラクションの一覧が始まったことがわかるように、ツールボックスより「**テキストブロック**」コントロールを「**PBCategory**」プリントブロックへドラッグアンドドロップし、コントロールをダブルクリックし、テキストを「**アトラクション**」に変更

④ ツールボックスより「**項目属性/変数**」コントロールを「**PBCategory**」プリントブロックへドラッグアンドドロップ

⑤ 表示された「項目属性／変数を挿入」ダイアログで **CategoryName** を選択し、[OK] をクリック

⑥ 一覧表示するアトラクション項目のタイトルを表示するため、ツールボックスより「**テキストブロック**」コントロールを「**PBCategory**」プリントブロックへ **3 つ**ドラッグアンドドロップし、テキストがそれぞれ「**名前**」、「**パーク名**」、「**写真**」となるように変更

⑦ ヘッダー部とデータの表示範囲の境界をわかりやすくするため、ツールボックスより「**線**」コントロールを「PBHeader」プリントブロックへドラッグアンドドロップ

⑧ 配置したコントロールを下図のように位置を調整（完全一致の必要はありません）

アトラクション表示部

① アトラクション表示部を定義するため、「PBCategory」を**右クリック**し、「**プリントブロックを挿入**」をクリックし、追加されたプリントブロックの [**Name**] プロパティを [**PBAttraction**] に変更

② 項目属性「**GameName**」、「**AmusementParkName**」、「**GameImage**」を配置するため、ツールボックスより「**項目属性/変数**」コントロールを「**PBAttraction**」プリントブロックへドラッグアンドドロップし、「項目属性／変数を挿入」ダイアログで対象の項目属性を選択し、[OK] をクリックする操作を **3 度**実施

③ 配置したコントロールを下図のように位置を調整（完全一致の必要はありません）

9-3-5. Source エレメントの定義

出力する処理を定義するため、Source エレメントで以下の定義をします。

Source 定義：

```
Print PBHeader
For each Category
      Print PBCategory
      For each Game
            Print PBAttraction
      Endfor
Endfor
```

9-3-6. レポートの実行

作成したプロシージャーオブジェクトを実行し、PDF 形式のレポートを確認します。

「GameList」のオブジェクトウィンドウのタブを右クリックし、「実行」をクリック

ヒント：

- 実行時に表示が更新された「ナビゲーション表示」ウィンドウでは、このデータ出力を実現するために For Each がネストされて実行されていることが確認可能

9-4. アトラクションのあるカテゴリのみを対象とした一覧

管理会社の担当者が新しいカテゴリ「**ウォーター系**」を追加（この時点で関連するアトラクションなし）したところ作成したアトラクションの一覧では、アトラクションの無いカテゴリも表示されることが確認できました。
アトラクションの無いカテゴリは表示しないカテゴリごとのアトラクション一覧も必要とリクエストがありました。

この情報を基に「**GameNoZeroList**」プロシージャーを作成します。

ヒント：

- 実際に新規カテゴリを追加し、「GameList」を実行することで、アトラクションの無いカテゴリが出力されることを確認可能

9-4-1. オブジェクトの複製

ここで必要となるオブジェクトは「**GameList**」とほぼ同一となるため、作業を短縮するため、オブジェクトを複製し、作業を開始します。

「GameList」のオブジェクトウィンドウのタブを右クリックし、[名前を付けて保存...]をクリックし、表示された「新規オブジェクト」ダイアログで以下のように変更し、作成します。

プロシージャー定義：

名前	GameNoZeroList
デスクリプション	カテゴリごとのアトラクション一覧

9-4-2. Source エレメントの更新

Source エレメントの定義をアトラクションの登録があるカテゴリのみが出力されるように更新します。

（文字色が緑のものは定義済みのもの、赤は以前の定義から変更したものです）

Source 定義：

```
Print PBHeader
For each Game
      Order CategoryId
      Print PBCategory
      For each Game
            Print PBAttraction
      Endfor
Endfor
```

ヒント：

- ネストされた For Each 双方のベーステーブルを同じにし、コントロールブレークを適用
- Order 節を追加し、データをグルーピングする項目属性を指定

9-4-3. レポートの実行

作成したプロシージャーオブジェクトを実行し、PDF 形式のレポートを確認します。

「GameNoZeroList」のオブジェクトウィンドウのタブを右クリックし、「実行」をクリック

ヒント：

- 実行時に表示が更新された「ナビゲーション表示」ウィンドウでは、内側の Forr Each が Break として実装されたことが確認可能

9-5. ショーの一覧

パーク番号「1」において、「今日」の日付以降に実施されるショーの一覧を実装します。

この一覧では以下の内容を表示してほしいとリクエストがありました。

・ヘッダー部：

・画像

・タイトル

・対象のパーク名

・今日の日付

・一覧表示する項目のタイトル

・表示項目部：

・ショー名

・ショー画像

・ショー日付

・ショー時間

この情報を基に「**ShowList**」プロシージャーを作成します。

9-5-1. 画像の取り込み

このレポートのヘッダー部に表示する画像をナレッジベース内で管理するため、イメージオブジェクトを作成するため、以下の操作を行います。

⑥ 「新規オブジェクト」ダイアログで、[カテゴリ] から「リソース」を選択し、[タイプを選択] 内で「Image」を選択し、[名前] を「**ShowListImage**」、[デスクリプション] を「Show List Image」（既定値のまま）とし、[作成] をクリック

⑦ 「新規画像ウィザード」ダイアログで「ファイルから画像を作成」（既定値）を選択し、[Next] をクリック

⑧ 「画像を選択」エクスプローラーで本資料同梱画像のうち「**Shows.png**」を選択し、[開く] をクリック

⑨ 「新規画像ウィザード」ダイアログの「名前」と「ファイル」に値が入力されていることを確認し、[Finish] をクリック

⑩ 表示された画像オブジェクトを**保存**

9-5-2. プロシージャーオブジェクトの作成

PDF 形式でレポートを出力するため、下記の内容でプロシージャーオブジェクトを作成し、一部のプロパティを設定します。

プロシージャー定義：

名前	ShowList
デスクリプション	ショーの一覧

プロパティ設定：

プロパティ	**値**
Main program	True
Call protocol	HTTP

9-5-3. Rules エレメントの定義

レポートを出力するプロシージャーオブジェクトとして定義するため、以下のルールを Rules エレメントで定義します。

Rules 定義：

```
Output_file("ShowList","PDF");
```

9-5-4. Layout エレメントの定義

出力する内容をデザインするため、Layout エレメントで以下の操作を行います。

ヘッダー部

① 既定で作成されているプリントブロック「**printBlock1**」の [**Name**] プロパティを [**PBHeader**] に変更

② ツールボックスより「**画像**」コントロールを「PBHeader」プリントブロックへドラッグアンドドロップ

③ 表示された「オブジェクトを選択」ダイアログで **ShowListImage** を選択し、[OK] をクリック

④ 一覧のタイトルを表示するため、ツールボックスより「**テキストブロック**」コントロールを「PBHeader」プリントブロックへドラッグアンドドロップし、コントロールをダブルクリックし、テキストを「**ショーの一覧**」に変更

⑤ ヘッダー部に表示する対象パーク名、表示開始日の説明用テキストを表示するため、ツールボックスより**テキストブロック**」コントロールを「PBHeader」プリントブロックへ **2 つ**ドラッグアンドドロップし、テキストがそれぞれ「**パーク名：**」、「**ショー表示範囲（開始）：**」となるように変更

⑥ ツールボックスより「**項目属性/変数**」コントロールを「PBHeader」プリントブロックへドラッグアンドドロップ

⑦ 表示された「項目属性／変数を挿入」ダイアログで **AmusementParkName** を選択し、[OK] をクリック

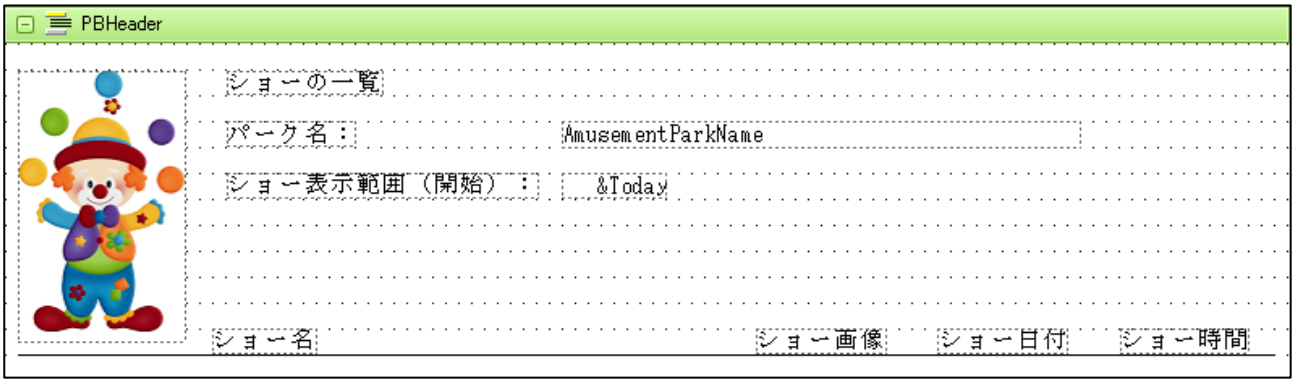
⑧ ツールボックスより「**項目属性/変数**」コントロールを「PBHeader」プリントブロックへドラッグアンドドロップ

⑨ 表示された「項目属性／変数を挿入」ダイアログで **Today** を選択し、[OK] をクリック

⑩ 一覧表示する項目のタイトルを表示するため、ツールボックスより「**テキストブロック**」コントロールを「PBHeader」プリントブロックへ **4 つ**ドラッグアンドドロップし、テキストがそれぞれ「**ショー名**」、「**ショー画像**」、「**ショー日付**」、「**ショー時間**」となるように変更

⑪ ヘッダー部とデータの表示範囲の境界をわかりやすくするため、ツールボックスより「**線**」コントロールを「PBHeader」プリントブロックへドラッグアンドドロップ

⑫ 配置したコントロールを下図のように位置を調整（完全一致の必要はありません）



ヒント：

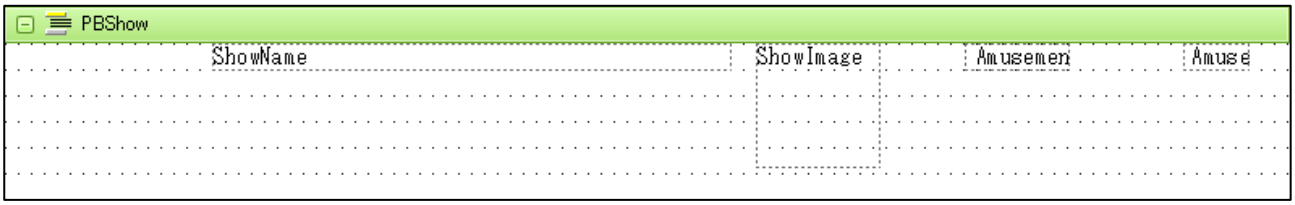
- プリントブロックの高さはプリントブロックの境界にマウスがある場合（カーソルが変化）、ドラッグすることで変更可能であり、同様に [Height] プロパティを利用して数値で指定も可能
- 「項目属性／変数を挿入」ダイアログの 1 列目に表示されるアイコンは、変数と項目属性を区別するために利用可能変数は「&」、項目属性は「」で表示

表示項目部

① 表示項目部を定義するため、「PBHeader」を**右クリック**し、「**プリントブロックを挿入**」をクリックし、追加されたプリントブロックの [**Name**] プロパティを [**PBShow**] に変更

② 項目属性「**ShowName**」、「**ShowImage**」、「**AmusementParkShowDate**」、「**AmusementParkShowTime**」を配置するため、ツールボックスより「**項目属性/変数**」コントロールを「**PBShow**」プリントブロックへドラッグアンドドロップし、「項目属性／変数を挿入」ダイアログで対象の項目属性を選択し、[OK] をクリックする操作を **4 度**実施

③ 配置したコントロールを下図のように位置を調整（完全一致の必要はありません）



9-5-5. Source エレメントの定義

出力する処理を定義するため、Source エレメントで以下の定義をします。

Source 定義：

```
For each AmusementPark
      Where AmusementParkId = 1
      Print PBHeader
      For each AmusementPark.Show
            Where AmusementParkShowDate >= &Today
            Print PBShow
      Endfor
Endfor
```

9-5-6．レポートの実行

作成したプロシージャーオブジェクトを実行し、PDF 形式のレポートを確認します。

「ShowList」のオブジェクトウィンドウのタブを右クリックし、「実行」をクリック

9-6．国名の一覧

その国にあるアミューズメントパークの数とともに国名を表示する一覧を実装します。

この一覧では以下の内容を表示してほしいとリクエストがありました。

- **ヘッダー部：**
  - 画像
  - タイトル
  - 一覧表示する項目のタイトル
- **表示項目部：**
  - 国名
  - パーク数

この情報を基に「**CountryList**」プロシージャーを作成します。

9-6-1. 画像の取り込み

このレポートのヘッダー部に表示する画像をナレッジベース内で管理するため、イメージオブジェクトを作成するため、以下の操作を行います。

① 「新規オブジェクト」ダイアログで、[カテゴリ] から「リソース」を選択し、[タイプを選択] 内で「Image」を選択し、[名前] を「**CountryParkImage**」、[デスクリプション] を「Country Park Image」（既定値のまま）とし、[作成] をクリック

② 「新規画像ウィザード」ダイアログで「ファイルから画像を作成」（既定値）を選択し、[Next] をクリック

③ 「画像を選択」エクスプローラーで本資料同梱画像のうち「**CountryParks.png**」を選択し、[開く] をクリック

④ 「新規画像ウィザード」ダイアログの「名前」と「ファイル」に値が入力されていることを確認し、[Finish] をクリック

⑤ 表示された画像オブジェクトを**保存**

9-6-2. プロシージャーオブジェクトの作成

PDF 形式でレポートを出力するため、下記の内容でプロシージャーオブジェクトを作成し、一部のプロパティを設定します。

プロシージャー定義：

名前	CountryList
デスクリプション	国の一覧

プロパティ設定：

プロパティ	**値**
Main program	True
Call protocol	HTTP

9-6-3. Rules エレメントの定義

レポートを出力するプロシージャーオブジェクトとして定義するため、以下のルールを Rules エレメントで定義します。

Rules 定義：

```
Output_file("CountryList","PDF");
```

9-6-4. Layout エレメントの定義

出力する内容をデザインするため、Layout エレメントで以下の操作を行います。

ヘッダー部

① 既定で作成されているプリントブロック「**printBlock1**」の [**Name**] プロパティを [**PBHeader**] に変更

② ツールボックスより「**画像**」コントロールを「PBHeader」プリントブロックへドラッグアンドドロップ

③ 表示された「オブジェクトを選択」ダイアログで **CountryParkImage** を選択し、[OK] をクリック

④ 一覧のタイトルを表示するため、ツールボックスより「**テキストブロック**」コントロールを「PBHeader」プリントブロックへドラッグアンドドロップし、コントロールをダブルクリックし、テキストを「**国の一覧**」に変更

⑤ 一覧表示する項目のタイトルを表示するため、ツールボックスより「**テキストブロック**」コントロールを「PBHeader」プリントブロックへ **2 つ**ドラッグアンドドロップし、テキストがそれぞれ「**国名**」、「**パーク数**」となるように変更

⑥ ヘッダー部とデータの表示範囲の境界をわかりやすくするため、ツールボックスより「**線**」コントロールを「PBHeader」プリントブロックへドラッグアンドドロップ

⑦ 配置したコントロールを下図のように位置を調整（完全一致の必要はありません）

表示項目部

① 表示項目部を定義するため、「PBHeader」を**右クリック**し、「**プリントブロックを挿入**」をクリックし、追加されたプリントブロックの [**Name**] プロパティを [**PBCountry**] に変更

② ツールボックスより「**項目属性/変数**」コントロールを「**PBCountry**」プリントブロックへドラッグアンドドロップ

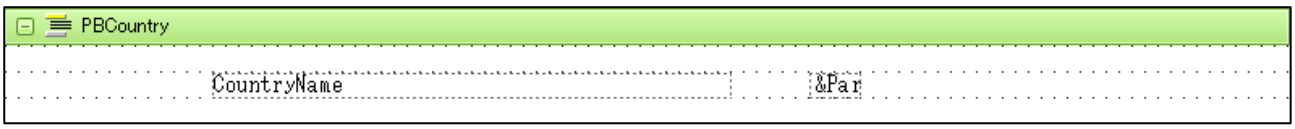
③ 表示された「項目属性／変数を挿入」ダイアログで **CountryName** を選択し、[OK] をクリック

④ ツールボックスより「**項目属性/変数**」コントロールを「**PBCountry**」プリントブロックへドラッグアンドドロップ

⑤ 表示された「項目属性／変数を挿入」ダイアログで [**新規変数**] をクリック

⑥ 表示された「新規変数」ダイアログで [**Name**] プロパティに [**ParkQty**] と入力し、[OK] をクリック

⑦ 配置したコントロールを下図のように位置を調整（完全一致の必要はありません）



ヒント：

- 「パーク数」はデータとして記録していないため。変数で表示
- 「パーク数」を表示するための変数は 「Variables」エレメントで定義することも可能
- 変数のデータタイプは既定の「Numeric(4.0)」で問題ないため、指定を割愛

9-6-5. Source エレメントの定義

出力する処理を定義するため、Source エレメントで以下の定義をします。

Source 定義：

```
Print PBHeader
For each Country
      &ParkQty = Count(AmusementParkName)
      Print PBCountry
Endfor
```

ヒント：

- 「パーク数」を表示するためにはローカル式の集計関数「Count」を利用し、読み込んだ国と関連するアミューズメントパークの数をカウントし、変数へ代入

9-6-6. レポートの実行

作成したプロシージャーオブジェクトを実行し、PDF 形式のレポートを確認します。

「CountryList」のオブジェクトウィンドウのタブを右クリックし、「実行」をクリック

9-7. パーク数が 2 以上の国のみを対象とした一覧

作成した国の一覧を管理会社の担当者が確認したところ、同様のレイアウトで「**パーク数が 2 以上の国**」だけを対象とした国の一覧も必要とリクエストがありました。

この情報を基に「**CountryListThan2**」プロシージャーを作成します。

9-7-1. オブジェクトの複製

ここで必要となるオブジェクトは「**CountryList**」とほぼ同一となるため、作業を短縮するため、オブジェクトを複製し、作業を開始します。

「CountryList」のオブジェクトウィンドウのタブを右クリックし、[名前を付けて保存...] をクリックし、表示された「新規オブジェクト」ダイアログで以下のように変更し、作成します。

プロシージャー定義：

名前	CountryListThan2
デスクリプション	国の一覧（パーク数 2 以上）

9-7-2. Source エレメントの更新

Source エレメントの定義を「パーク数が 2 以上の国」のみが対象となるように更新します。（文字色が緑のものは定義済みのものです）

Source 定義：

```
Print PBHeader
For each Country
      Where Count(AmusementParkName) >= 2
      &CountryQty = Count(AmusementParkName)
      Print PBCountry
Endfor
```

ヒント：

- **条件**として利用する「パーク数」と、**表示**用の「パーク数」はそれぞれでローカル式を実施する必要がある

9-7-3. レポートの実行

作成したプロシージャーオブジェクトを実行し、PDF 形式のレポートを確認します。

「CountryListThan2」のオブジェクトウィンドウのタブを右クリックし、「実行」をクリック

10. パラメーターの受け渡し

レポート出力の機能を実装する中で固定値の条件で表示したレポートがありました。

これらのレポートについてユーザーから実行時に指定して表示できるようにしたいとリクエストがあったことが、管理会社の担当者から連絡がありました。

該当のレポートと要望については以下の通りです。

① パーク数 2 以上の国名の一覧（CountryListThan2）

表示する国を決定するパーク数を自由に指定できるようにしたい

② ショーの一覧

表示対象となる「アミューズメントパーク」、「ショーの日付」を自由に指定できるようにしたい

このリクエストを実装するためには、条件を指定する画面を用意し、画面で指定された内容をパラメーターとして受け渡す必要があります。

この機能の実装を進めます。

10-1. 指定数以上のパークがある国の一覧

画面上でパーク数を指定し、その数以上の国が表示されるレポートを実装します。

この一覧に表示したい内容は「パーク数 2 以上の国名の一覧（CountryListThan2）」と同じであり、一覧を表示する前に対象を決定するパーク数を入力できる画面が必要です。

この画面では、データ登録は不要なため、Web パネルオブジェクトを利用します。

この情報を基に「**ParksPerCountry**」プロシージャー、「**ParksFilter**」Web パネルを作成します。

10-1-1. プロシージャーオブジェクトの複製

ここで必要となるオブジェクトは「**CountryListThan2**」をベースとしたいため、オブジェクトを複製し、作業を開始します。

「CountryListThan2」のオブジェクトウィンドウのタブを右クリックし、[名前を付けて保存...] をクリックし、表示された「新規オブジェクト」ダイアログで以下のように変更し、作成します。

プロシージャー定義：

名前	ParksPerCountry
デスクリプション	国の一覧（パーク数指定）

10-1-2. ParksPerCountry の変数追加

パーク数の指定値を受け取るため、変数の追加が必要です。

Variables エレメントで以下の変数を新たに定義します。

Variables 定義：

名前	タイプ
MinParkQty	Numeric(4.0)

10-1-3. ParksPerCountry の Rules エレメント更新

Rules エレメントの定義にパラメーターを受け取るルールを追記します。

以下に追記分のみを記載しています。

Rules 定義：

```
Parm(IN:&MinParkQty);
```

10-1-4. ParksPerCountry の Source エレメント更新

Source エレメントの定義を、パラメーターで受け取った値より多いパーク数の国のみを出力するように更新します。

（文字色が緑のものは定義済みのもの、赤は以前の定義から変更したものです）

Source 定義：

```
Print PBHeader
For each Country
      Where Count(AmusementParkName) >= &MinParkQty
      &ParkQty = Count(AmusementParkName)
      Print PBCountry
Endfor
```

10-1-5. Web パネルオブジェクトの作成

パーク数を入力するため、下記の内容で Web パネルオブジェクトを作成します。

Web パネル定義：

名前	ParksFilter
デスクリプション	パーク数の指定

ヒント：

- Web パネルオブジェクトの作成は「新規オブジェクト」ダイアログで、[カテゴリ] から「ユーザーインターフェース」を選択し、[タイプを選択] 内で「Web Panel」を選択

10-1-6. ParksFilter の変数定義

パーク数を入力するため、変数の定義が必要です。

Variables エレメントで以下の変数を定義します。

Variables 定義：

名前	タイプ	デスクリプション
MinParkQty	Numeric(4.0)	パーク数

ヒント：

- 定義した変数は画面上への配置が予定されているため、デスクリプション列も入力

10-1-7. ParksFilter の Web Layout エレメント定義

パーク数入力画面の UI をデザインするため、Web Layout エレメントで以下の操作を行います。

① パーク数入力欄を表示するため、ツールボックスより「**項目属性/変数**」コントロールを**紫色の枠（MainTable）内**へドラッグアンドドロップ

② 表示された「項目属性／変数を挿入」ダイアログで **MinParkQty** を選択し、[OK]をクリック

③ **ParksPerCountry**（国の一覧（パーク数指定））を呼び出すボタンを作成するため、ツールボックスより「**ボタン**」コントロールを**紫色の枠（MainTable）内下部**へドラッグアンドドロップ

④ 表示された「ユーザーイベントを選択/定義」ダイアログで「**イベント名**」に「**CallParksPerCountry**」と入力し、[OK] をクリック

⑤ 配置した**ボタン**コントロールの [**Caption**] プロパティを [**表示**] に変更

⑥ 配置したコントロールは下図のようになっていることを確認

⑦ **ボタン**コントロールをダブルクリック（Events エレメントにイベントが作成）

ヒント：

- Web Layout にあらかじめ配置されている紫色の枠（MainTable）の実態は、「テーブル」コントロール
- テーブルコントロール内へ他のコントロールを配置する場合、表示されたセルの上下左右にセルを追加し、配置が可能（何も配置されていないテーブルの場合、中央に配置可能）

10-1-8. ParksFilter の Events エレメント定義

画面で入力したパーク数をパラメーターとし、ParksPerCountry（国の一覧（パーク数指定））を呼び出すため、Events エレメントで「**CallParksPerCountry**」イベントの内容を以下のように記述します。（文字色が緑のものは定義済みのものです）

Events 定義：

```
Event 'CallParksPerCountry'
      ParksPerCountry(&MinParkQty)
Endevent
```

10-1-9. アプリケーションの実行

作成した「ParksPerCountry」プロシージャー、「ParksFilter」Web パネルを実行します。

[F5] キーを押し、アプリケーションを**実行**します。

表示された [Launchpad] ウィンドウから「ParksFilter」Web パネルにより生成された画面を開きます。

好きな値を指定し、「表示」ボタンをクリックし、結果を確認します。

10-2. アミューズメントパークを指定したショーの一覧

画面上でアミューズメントパークとショー日付を指定し、その条件に一致するショーが表示されるレポートを実装します。

この一覧に表示したい内容は「ShowList」と同じであり、一覧を表示する前に対象を決定する「アミューズメントパーク」と「ショー日付」を入力できる画面が必要です。

この情報を基に「**ShowsInSelectedPark**」プロシージャー、「**ParkAndDate**」Web パネルを作成します。

10-2-1. プロシージャーオブジェクトの複製

個々で必要となるオブジェクトは「**ShowList**」をベースとしたいため、オブジェクトを複製し、作業を開始します。

「ShowList」のオブジェクトウィンドウのタブを右クリックし、[名前を付けて保存...] をクリックし、表示された「新規オブジェクト」ダイアログで以下のように変更し、作成します。

プロシージャー定義：

名前	ShowsInSelectedPark
デスクリプション	ショーの一覧（パーク指定）

10-2-2. ShowsInSelectedPark の変数定義

表示対象となるアミューズメントパークの番号、ショーの日付を受け取るため、変数の定義が必要です。

Variables エレメントで以下の変数を定義します。

Variables 定義：

名前	タイプ
ParkId	Id
FromDate	Date

10-2-3. ShowsInSelectedPark の Rules エレメント更新

Rules エレメントの定義にパラメーターを受け取るルールを追記します。

以下に追記分のみを記載しています。

Rules 定義：

```
Parm(IN:&ParkId,IN:&FromDate);
```

10-2-4. ShowsInSelectedPark の Layout エレメント更新

ショー表示範囲の開始日は今日の日付ではなく、パラメーターで受け取った値となるため、PBHeader に配置された「&Today」変数を削除し、同じ場所へ「&FromDate」を配置します。変更した結果のレイアウトは下図を想定しています（完全一致の必要はありません）

10-2-5. ShowsInSelectedPark の Source エレメント更新

Source エレメントの定義を、パラメーターで受け取ったパーク番号と一致し、日付より後の日付に開催されるショーのみを出力するように更新します。

（文字色が緑のものは定義済みのもの、赤は以前の定義から変更したものです）

Source 定義：

```
For each AmusementPark
      Where AmusementParkId = &ParkId
      Print PBHeader
      For each AmusementPark.Show
            Where AmusementParkShowDate >= &FromDate
            Print PBShow
      Endfor
Endfor
```

10-2-6. Web パネルオブジェクトの作成

アミューズメントパーク、ショー日付を入力するため、下記の内容で Web パネルオブジェクトを作成します。

Web パネル定義：

名前	ShowsFilter
デスクリプション	パークと日付の指定

10-2-7. ShowsFilter の変数定義

ユーザーは入力する際にはパーク名を入力し、パラメーターとする場合に内部的にパーク番号となることを望んでいると管理会社の担当者より連絡があったため、パーク番号、パーク名、ショー日付の変数の定義が必要です。

Variables エレメントで以下の変数を定義します。

Variables 定義：

名前	タイプ	デスクリプション
AmusementParkId	Attribute:AmusementParkId	パーク番号
AmusementParkName	Attribute:AmusementParkName	パーク名
AmusementParkShowDate	Attribute:AmusementParkShowDate	ショー日付

ヒント：

- 項目属性名と同名で変数を定義した場合、データタイプが自動的に項目属性にリンクし、項目属性のデータタイプが変更されると、変数のデータタイプも変更される
- 任意名称の変数のデータタイプを項目属性に紐づける場合には、[Based on] プロパティを利用

10-2-8. ShowsFilter の Web Layout エレメント定義

パーク名、ショー日付を入力する画面の UI をデザインするため、Web Layout エレメントで以下の操作を行います。

① パーク名入力欄を表示するため、ツールボックスより「**項目属性/変数**」コントロールを**紫色の枠（MainTable）内**へドラッグアンドドロップ

② 表示された「項目属性／変数を挿入」ダイアログで **AmusementParkName** を選択し、[OK]をクリック

③ ショー日付入力欄を表示するため、ツールボックスより「**項目属性/変数**」コントロールを**紫色の枠（MainTable）内下部**へドラッグアンドドロップ

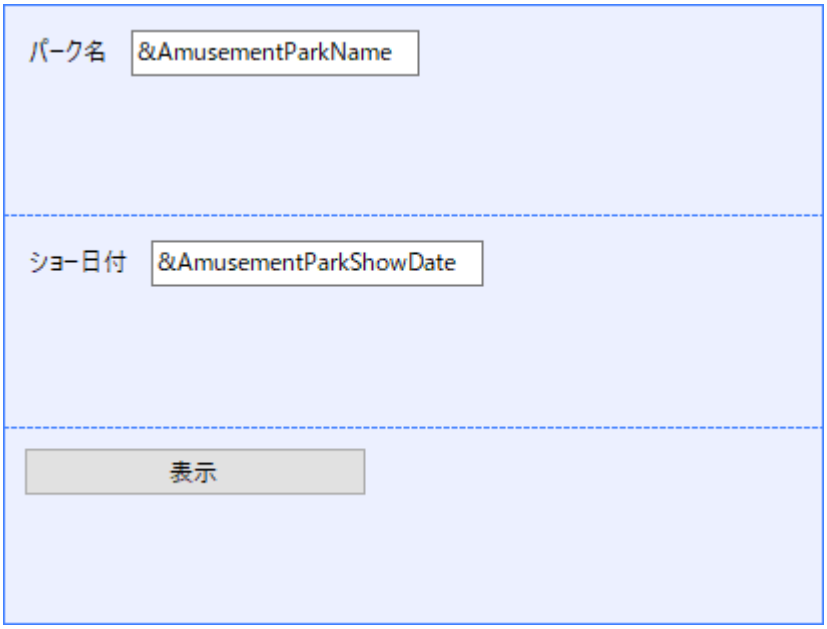
④ 表示された「項目属性／変数を挿入」ダイアログで **AmusementParkShowDate** を選択し、[OK]をクリック

⑤ **ShowsInSelectedPark**（ショーの一覧（パーク指定））を呼び出すボタンを作成するため、ツールボックスより「**ボタン**」コントロールを**紫色の枠（MainTable）内下部**へドラッグアンドドロップ

⑥ 表示された「ユーザーイベントを選択/定義」ダイアログで「**イベント名**」に「**CallShowsInSelectedPark**」と入力し、[OK] をクリック

⑦ 配置した**ボタン**コントロールの [**Caption**] プロパティを [**表示**] に変更

⑧ 配置したコントロールは下図のようになっていることを確認



⑨ **ボタン**コントロールをダブルクリック（Events エレメントにイベントが作成）

10-2-9. ShowsFilter の Events エレメント定義

画面で入力したパーク名からパーク番号を取得し、ショー日付とともにパラメーターとし、ShowsInSelectedPark（ショーの一覧（パーク指定））を呼び出すため、Events エレメントで「**CallShowsInSelectedPark**」イベントの内容を以下のように記述します。（文字色が緑のものは定義済みのものです）

Events 定義：

```
Event 'CallShowsInSelectedPark'
      &AmusementParkId = Find(AmusementParkId,AmusementParkName = &AmusementParkName,0)
      ShowsInSelectedPark(&AmusementParkId,&AmusementParkShowDate)
Endevent
```

ヒント：

- 呼び出し元オブジェクトでパラメーターとして定義された変数名と、呼び出されるオブジェクトでパラメーターを受け取るために定義された変数名は同じである必要はありませんが、データタイプには互換性のある定義が必要

10-2-10. アプリケーションの実行

作成した「ShowsInSelectedPark」プロシージャー、「ShowsFilter」Web パネルを実行します。

[F5] キーを押し、アプリケーションを**実行**します。

表示された [Launchpad] ウィンドウから「ShowsFilter」Web パネルにより生成された画面を開きます。

パーク名、ショー日付を指定し、「表示」ボタンをクリックし、結果を確認します。

11. ビジネスコンポーネント

管理会社の担当者から、このアプリケーションには、多くのレコードに対し、同一の操作が必要となるケースがあると追加の要望について連絡がありました。

現時点で判明しているケースとしては以下の 2 パターンとのことです。

① 記録済みのすべての「修理」に対し、一定の割合で価格を引き上げて更新

② 記録済みのすべての「修理」をすべて削除

どちらの操作も画面を利用し、1 件ずつ対応する場合、件数に依存しますが、非常に時間がかかってしまうため、内部的な処理で対応できるよう**ビジネスコンポーネント**を使用してこれらの要望をかなえるよう実装を進めます。

11-1. 修理価格の値上げ

画面上で値上げ率を指定し、すべての修理の価格を値上げする機能を実装します。

この場合、値上げ率を入力できる画面の作成と、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定義する必要があります。

この情報を基に「**IncreasePercentageBC**」Web パネルを作成します。

11-1-1. 「Repair」トランザクションのプロパティ更新

Repair 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するため、オブジェクトのプロパティを設定します。

以下の操作を行います。

① [KB エクスプローラー] ウィンドウで「Repair」をダブルクリック
② 表示された Repair トランザクションのオブジェクトウィンドウのタブを右クリックし、[プロパティ] をクリック
③ 表示された [プロパティ] ウィンドウで [Business Component] プロパティを [True] に変更
④ オブジェクトを保存

11-1-2. Web パネルオブジェクトの作成

値上げ率を入力し、データの更新処理を行うため、下記の内容で Web パネルオブジェクトを作成します。

Web パネル定義：

名前	IncreasePercentageBC
デスクリプション	修理価格値上げ（BC）

11-1-3. IncreasePercentageBC の変数定義

値上げ率を入力する変数と、ビジネスコンポーネントを利用するための変数定義が必要となるため、Variables エレメントで以下の変数を定義します。

Variables 定義：

名前	タイプ	デスクリプション
Percentage	Percentage	値上げ率
BCRepair	Repair	BCRepair
Messages	Messages, GeneXus.Common	Messages
MessageItem	Messages.Message, GeneXus.Common	Message Item

ヒント：

- トランザクションの [Business Component] プロパティを [True] にした場合、ナレッジベース内にトランザクション名と同名のビジネスコンポーネント型データタイプが追加
- ビジネスコンポーネントのエラーメッセージを取得するため、Messages 型および Messages.Message 型の変数も追加

11-1-4. IncreasePercentageBC の Web Layout エレメント定義

値上げ率を入力する UI をデザインするため、Web Layout エレメントで以下の操作を行います。

① 値上げ率入力欄を表示するため、ツールボックスより「**項目属性/変数**」コントロールを**紫色の枠（MainTable）内**へドラッグアンドドロップ

② 表示された「項目属性／変数を挿入」ダイアログで **Percentage** を選択し、[OK]をクリック

③ ビジネスコンポーネントを利用した処理を実行するボタンを作成するため、ツールボックスより「**ボタン**」コントロールを**紫色の枠（MainTable）内下部**へドラッグアンドドロップ

④ 表示された「ユーザーイベントを選択/定義」ダイアログで「**イベント名**」に「**DoIncrease**」と入力し、[OK] をクリック

⑤ 配置した**ボタン**コントロールの [**Caption**] プロパティを [**実行**] に変更

⑥ 配置したコントロールは下図のようになっていることを確認

⑦ **ボタン**コントロールをダブルクリック（Events エレメントにイベントが作成）

11-1-5. IncreasePercentageBC の Events エレメント定義

画面で入力した値上げ率を利用し、すべての修理費用を値上げする処理を実装するため、Events エレメントで「**DoIncrease**」イベントの内容を以下のように記述します。（文字色が緑のものは定義済みのものです）

Events 定義：

```
Event 'DoIncrease'
      For each Repair
            &BCRepair.Load(RepairId)
            &BCRepair.RepairCost = &BCRepair.RepairCost * (1 + &Percentage / 100)
            If &BCRepair.Update()
                  Commit
                  Msg(RepairId.ToString() + ”の更新が成功しました。”)
            Else
                  Rollback
                  Msg(RepairId.ToString() + ”の更新が失敗しました。”)
                  &Messages = &BCRepair.GetMessages()
                  For &MessageItem in &Messages
                        Msg(&MessageItem.Description)
                  Endfor
            Endif
      Endfor
Endevent
```

ヒント：

- Web パネルオブジェクトの Events エレメントで Msg 関数（Events に記載しているので、ルールではない）を利用することで、処理が完了した際に画面上へメッセージを表示可能
- For In コマンドを利用することで、In の右辺に指定したコレクション内のアイテム 1 つを左辺に代入し、処理を実施することが可能。処理はコレクション内最後のアイテムに到達するまで実施
- Messages 型は SDT のコレクション型データタイプ

11-1-6. アプリケーションの実行

作成した「IncreasePercentageBC」Web パネルを実行します。

[F5] キーを押し、アプリケーションを**実行**します。

表示された [Launchpad] ウィンドウから「IncreasePercentageBC」Web パネルにより生成された画面を開きます。

値上げ率を指定し、「実行」ボタンをクリックします。

[Launchpad] ウィンドウへ戻り、Repair トランザクションのリンクをクリックし、データを表示し、処理結果を確認します。

11-2. すべての修理を削除

画面上でボタンをクリックすることで、すべての修理を削除する機能を実装します。

この処理はビジネスコンポーネントを利用することで実装できます。

この情報を基に「**DeleteAllRepairBC**」Web パネルを作成します。

11-2-1. Web パネルオブジェクトの作成

ボタンをクリックすることですべての修理を削除するため、下記の内容で Web パネルオブジェクトを作成します。

Web パネル定義：

名前	DeleteAllRepairBC
デスクリプション	修理削除（BC）

11-2-2. 変数定義

ビジネスコンポーネントを利用するための変数定義が必要となるため、Variables エレメントで以下の変数を定義します。

Variables 定義：

名前	タイプ	デスクリプション
BCRepair	Repair	BCRepair
Messages	Messages, GeneXus.Common	Messages
MessageItem	Messages.Message, GeneXus.Common	Message Item

11-2-3. Web Layout エレメント定義

クリックするボタンを配置するように UI をデザインします。

Web Layout エレメントで以下の操作を行います。

① ビジネスコンポーネントを利用した処理を実行するボタンを作成するため、ツールボックスより「**ボタン**」コントロールを**紫色の枠（MainTable）内**へドラッグアンドドロップ

② 表示された「ユーザーイベントを選択/定義」ダイアログで「**イベント名**」に「**DoDelete**」と入力し、[OK] をクリック

③ 配置した**ボタン**コントロールの [**Caption**] プロパティを [**全件削除**] に変更

④ **ボタン**コントロールをダブルクリック（Events エレメントにイベントが作成）

11-2-4. Events エレメント定義

すべての修理を削除する処理を実装するため、Events エレメントで「**DoDelete**」イベントの内容を以下のように記述します。（文字色が緑のものは定義済みのものです）

Events 定義：

```
Event 'DoDelete'
      For each Repair
            &BCRepair.Load(RepairId)
            &BCRepair.Delete()
            If &BCRepair.Success()
                  Commit
                  Msg(RepairId.ToString() + ”の削除が成功しました。”)
            Else
                  Rollback
                  Msg(RepairId.ToString() + ”の削除が失敗しました。”)
                  &Messages = &BCRepair.GetMessages()
                  For &MessageItem in &Messages
                        Msg(&MessageItem.Description)
                  Endfor
            Endif
      Endfor
Endevent
```

11-2-5. アプリケーションの実行

作成した「DeleteAllRepairBC」Web パネルを実行します。

[F5] キーを押し、アプリケーションを**実行**します。

表示された [Launchpad] ウィンドウから「DeleteAllRepairBC」Web パネルにより生成された画面を開きます。

「全件削除」ボタンをクリックします。

[Launchpad] ウィンドウへ戻り、Repair トランザクションのリンクをクリックし、データを表示し、処理結果を確認します。

12. プロシージャーによるレコード操作

直前の課題でビジネスコンポーネントを利用し、実装した処理は、プロシージャーオブジェクトの機能を利用した場合も実装することができます。

そのため、既にユーザーの要望をかなえる機能を実装済みですが、同様の機能をプロシージャーオブジェクトで実装する方法を確認します。
また、確認の前に別の機能を利用し、修理のデータを作成します。

12-1. トランザクションの機能による初期データ登録

トランザクションオブジェクトが持つ機能を利用することで、GeneXus 上でアプリケーションのビルド時に指定したデータをテーブルに登録することができます。

この課題において必要なデータは、この機能を利用し、準備します。

12-1-1. 「Repair」トランザクションのプロパティ更新

Repair トランザクションによるデータ入力機能を利用できるようにするため、オブジェクトのプロパティを設定します。

以下の操作を行います。

① [KB エクスプローラー] ウィンドウで「Repair」をダブルクリック
② 表示された Repair トランザクションのオブジェクトウィンドウのタブを右クリックし、[プロパティ] をクリック
③ 表示された [プロパティ] ウィンドウで [Data Provider] プロパティを [True] に変更
④ オブジェクトを保存

12-1-2. データプロバイダーオブジェクトの定義

[Data Provider] プロパティを [True] に変更したことで作成される

「Repair_DataProvider」データプロバイダーオブジェクトの定義を完成させます。

以下の操作を行います。

① [KB エクスプローラー] ウィンドウで Repair トランザクションの子ノード Repair_DataProvider データプロバイダーオブジェクト（下図）をダブルクリック

② 別添ファイル「**12-1-2_Repair_DataProvider.pdf**」の内容を Source エレメントへ貼り付け

③ オブジェクトを保存

ヒント：

- GeneXus におけるデータプロバイダーオブジェクトは、構造化データタイプを出力可能なオブジェクトタイプ
- RepairId は自動採番が有効なため、データプロバイダー内で指定は不要

12-1-3. アプリケーションの実行

データの追加を実施するため、「Repair」トランザクションを実行します。

[F5] キーを押し、アプリケーションを実行します。

表示された [Launchpad] ウィンドウから「Repair」トランザクションにより生成された画面を開きます。

修理情報が登録されていることを確認します。

12-2. 修理価格の値上げ

ビジネスコンポーネントを利用し、既に実装している値上げ率を指定し、すべての修理の価格を値上げする機能をプロシージャーオブジェクトで実装します。

前提については、既にビジネスコンポーネントで実施したものと同一です。

この情報を基に「**IncreaseRepairCost**」プロシージャー、

「**IncreasePercentageP**」Web パネルを作成します。

12-2-1. プロシージャーオブジェクトの作成

データを更新する処理を実装するため、下記の内容でプロシージャーオブジェクトを作成します。

プロシージャー定義：

名前	IncreaseRepairCost
デスクリプション	値上げ処理

12-2-2. IncreaseRepairCost の変数定義

値上げ率を入力する Web パネルからプロシージャーにパラメーターとして値を受け取る必要があるため、変数の定義が必要です。

Variables エレメントで以下の変数を定義します。

Variables 定義：

名前	**タイプ**
IncreasePercentage	Percentage

12-2-3. IncreaseRepairCost の Rules エレメント定義

Rules エレメントの定義にパラメーターを受け取るルールを以下のように定義します。

Rules 定義：

```
Parm(IN:&IncreasePercentage);
```

12-2-4. IncreaseRepairCost の Source エレメント定義

Source エレメントにパラメーターで受け取った値を利用し、すべての修理の費用を値上げする処理を定義します。

Source 定義：

```
For each Repair
      RepairCost = RepairCost * (1 + &IncreasePercentage / 100)
Endfor
```

ヒント：

- データの更新は割り当てのみで成立
- 処理の成功/失敗を判断するメソッドは無し
- Commit はオブジェクト内の処理がすべて実行された後に自動で実行されるため、明記の必要無し

12-2-5. Web パネルオブジェクトの複製

ここで必要となるインターフェースは「**IncreasePercentageBC**」をベースとしたいため、オブジェクトを複製し、作成を開始します。

「IncreasePercentageBC」のオブジェクトウィンドウのタブを右クリックし、[名前を付けて保存...] をクリックし、表示された「新規オブジェクト」ダイアログで以下のように変更し、作成します。

Web パネル定義：

名前	IncreasePercentageP
デスクリプション	修理価格値上げ（プロシージャー）

12-2-6. IncreasePercentageP の Events エレメント修正

画面で入力した値上げ率をパラメーターとして IncreaseRepairCost を呼び出すように Events エレメントで「**DoIncrease**」イベントの内容を以下のように修正します。（文字色が緑のものは定義済みのもの、赤は以前の定義から変更したものです）

Events 定義：

```
Event 'DoIncrease'
      IncreaseRepairCost(&Percentage)
      Msg("処理が完了しました")
Endevent
```

12-2-7. アプリケーションの実行

作成した「IncreasePercentageP」Web パネルを実行します。

[F5] キーを押し、アプリケーションを**実行**します。

表示された [Launchpad] ウィンドウから「IncreasePercentageP」Web パネルにより生成された画面を開きます。

値上げ率を指定し、「実行」ボタンをクリックします。

[Launchpad] ウィンドウへ戻り、Repair トランザクションのリンクをクリックし、データを表示し、処理結果を確認します。

12-3. すべての修理を削除

ビジネスコンポーネントを利用し、既に実装しているすべての修理を削除する機能をプロシージャーオブジェクトで実装します。

前提については、既にビジネスコンポーネントで実施したものと同一です。

この情報を基に「**RepairDelete**」プロシージャー、「**DeleteAllRepairP**」Web パネルを作成します。

12-3-1. プロシージャーオブジェクトの作成

データを削除する処理を実装するため、下記の内容でプロシージャーオブジェクトを作成します。

プロシージャー定義：

名前	RepairDelete
デスクリプション	削除処理

12-3-2. RepairDelete の Source エレメント定義

Source エレメントにすべての修理を削除する処理を定義します。

Source 定義：

```
For each Repair
      Delete
Endfor
```

ヒント：

- データの削除は Delete コマンドの記述で実装
- すべてのデータ削除のため、パラメーター受け取りはなく、対象データのフィルタリングも実装不要

12-3-3. Web パネルオブジェクトの複製

ここで必要となるインターフェースは「**DeleteAllRepairBC**」をベースとしたいため、オブジェクトを複製し、作成を開始します。

「DeleteAllRepairBC」のオブジェクトウィンドウのタブを右クリックし、[名前を付けて保存...] をクリックし、表示された「新規オブジェクト」ダイアログで以下のように変更し、作成します。

Web パネル定義：

名前	DeleteAllRepairP
デスクリプション	修理削除（プロシージャー）

12-3-4. DeleteAllRepairP の Events エレメント修正

RepairDelete を呼び出すように Events エレメントで「**DoDelete**」イベントの内容を以下のように修正します。（文字色が緑のものは定義済みのもの、赤は以前の定義から変更したものです）

Events 定義：

```
Event 'DoDelete'
      RepairDelete()
      Msg("処理が完了しました")
Endevent
```

12-3-5. アプリケーションの実行

作成した「DeleteAllRepairP」Web パネルを実行します。

[F5] キーを押し、アプリケーションを**実行**します。

表示された [Launchpad] ウィンドウから「DeleteAllRepairP」Web パネルにより生成された画面を開きます。

「全件削除」ボタンをクリックします。

[Launchpad] ウィンドウへ戻り、Repair トランザクションのリンクをクリックし、データを表示し、処理結果を確認します。

13. Web パネルの利用

管理会社の担当者より PDF 形式のレポートではなく、Web アプリケーション上に動的なレポートとして以下の画面が必要と連絡がありました。

① 国名とアミューズメントパーク数の一覧

② ショーの一覧

③ アミューズメントパークのショーの数に基づくランキング

また、どの画面を開いた場合にも共通で表示される領域があることにユーザーは気が付き、このエリアにアプリケーションの名前を表示してほしいというリクエストもあるとのことでした。

Web パネルオブジェクトを使用し、これらの要望をかなえるよう実装を進めます。

13-1. 国名とアミューズメントパーク数の一覧

ユーザーが要望した画面の内容について詳細を確認したところ、要望としては次の通りです。

- 一覧には、**国名**と各国の**アミューズメントパークの数**を表示
- フィルターとして、**国名の前方一致**を保持
- フィルターが**未入力**の場合、**全件**を表示

この情報を基に「**WPCountryList**」Web パネルを作成します。

13-1-1. Web パネルオブジェクトの作成

国の一覧を表示するため、下記の内容で Web パネルオブジェクトを作成します。

Web パネル定義：

名前	WPCountryList
デスクリプション	国の一覧

13-1-2. 変数定義

アミューズメントパークの数、フィルターとなる国名のため、変数の定義が必要です。

Variables エレメントで以下の変数を定義します。

Variables 定義：

名前	タイプ	デスクリプション
ParkNum	Numeric(4.0)	パーク数
CountryName	Attribute:CountryName	国名

13-1-3. Web Layout エレメント定義

国名によるフィルター項目と、パーク数とともにすべての国名を一覧表示するグリッドをデザインします。

Web Layout エレメントで以下の操作を行います。

① フィルターとなる国名入力欄を表示するため、ツールボックスより「**項目属性/変数**」コントロールを**紫色の枠（MainTable）内**へドラッグアンドドロップ

② 表示された「項目属性／変数を挿入」ダイアログで **CountryName（変数）**を選択し、[OK]をクリック

③ データの一覧を実装するため、ツールボックスより「**グリッド**」コントロールを**紫色の枠（MainTable）内下部**へドラッグアンドドロップ

④ 表示された「項目属性／変数を挿入」ダイアログで **CountryName（項目属性）**、**ParkNum** を選択し、[OK] をクリック

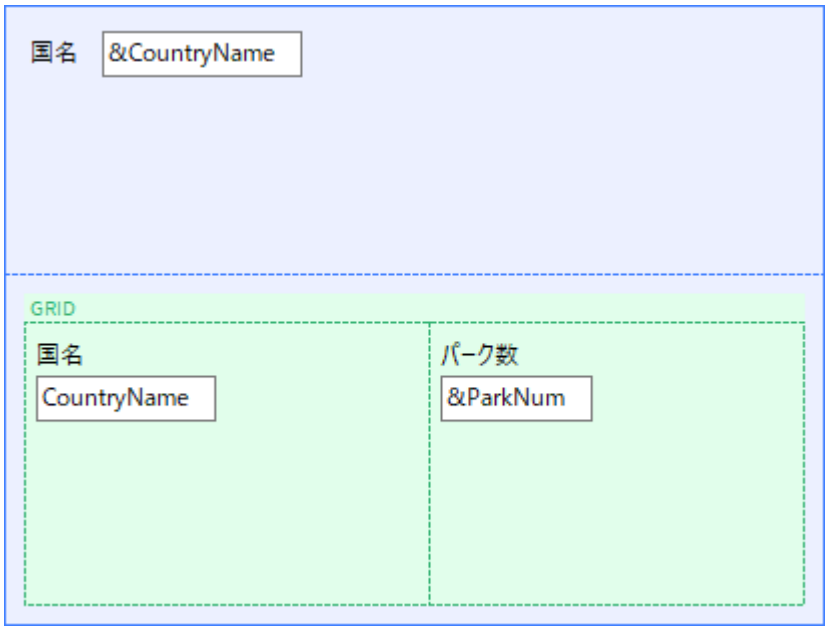
⑤ 配置したグリッドコントロールを**選択**し、**[F4] キー**を押下

⑥ 表示された **[プロパティ] ウィンドウ**で **[Base Trn]** プロパティを **[Country]** と設定

⑦ **[Conditions]** プロパティに以下の条件を設定

```
CountryName Like &CountryName when not &CountryName.IsEmpty()
```

⑧ 配置したコントロールは下図のようになっていることを確認



ヒント：

- 「項目属性／変数を挿入」ダイアログの 1 列目に表示されるアイコンは、変数と項目属性を区別するために利用可能変数は「&」、項目属性は「」で表示
- 「項目属性／変数を挿入」ダイアログで、[Ctrl] キーや [Shift] キーを押下しながら選択することで、複数選択可能
- グリッドコントロールに配置された列はドラッグアンドドロップで移動（並び替え）可能
- [Conditions] プロパティを編集する場合、プロパティを選択した際に表示される「…」アイコンをクリックすることで、Condition 定義ダイアログが表示され、「Rules」エレメントのように「Ctrl + Space」キーで入力候補が利用可能
- 条件において「Like」演算子を利用することで、前方一致を実装可能
- 「not」論理演算子を利用することで、Boolean 型の結果を否定することが可能

13-1-4. Events エレメント定義

データの一覧に含まれるパーク数は、変数のため、値を代入しなければ何も表示されない。そのため、表示する値は、Load イベントを利用し、ローカル式で代入を行う必要があります。

Events エレメントで「**Load**」イベントを以下のように記述します。

Events 定義：

```
Event Load
      &ParkNum = Count(AmusementParkName)
Endevent
```

ヒント：

- ベーステーブルを持つグリッドに対する Load イベントとなるため、グリッドへ行を表示する直前にイベントは毎行実行される

13-1-5. アプリケーションの実行

作成した「WPCountryList」Web パネルを実行します。

F5 キーを押し、アプリケーションを**実行**します。

表示された [Launchpad] ウィンドウから「WPCountryList」Web パネルにより生成された画面を開きます。

一覧にすべての国と各国のパーク数が表示されることの確認と、国名のフィルターによって前方一致で国名をフィルタリングできることを確認します。

13-2. ショーの一覧

ユーザーが要望した画面の内容について詳細を確認したところ、要望としては次の通りです。

- 一覧には、**ショー名**、**ショー画像**、**パーク名**を表示
- フィルターとして、**パーク名**を**選択**
- フィルターが**未選択**の場合、**全件**を表示
- **ショーの名前順**に並び替え
- 選択したショーの**詳細（パーク名、パーク住所、ショー名、ショー画像、ショー日付、ショー時間）**を**ポップアップ**で表示

この情報を基に「**OneShowDetail**」Web パネルオブジェクト、「**WPShowList**」Web パネルを作成します。

13-2-1. OneShowDetail Web パネルオブジェクトの作成

はじめに、ショーの詳細を表示する画面を実装します。

下記の内容で Web パネルオブジェクトを作成します。

Web パネル定義：

名前	OneShowDetail
デスクリプション	ショー詳細

13-2-2. OneShowDetail の Rules エレメント定義

Rules エレメントの定義にパラメーターを受け取るルールを以下のように定義します。

Rules 定義：

```
Parm(IN:AmusementParkId, IN:ShowId);
```

ヒント：

- 詳細情報として表示する内容は特定のアミューズメントパークに紐づけられたショー（つまり AMUSEMENTPARKSHOW テーブルの内容）のため、主キーとなる 2 項目の値を取得
- パラメーターを「項目属性」で受け取った場合、値はこの項目属性に対して等価フィルターとして利用

13-2-3. OneShowDetail の Web Layout エレメント定義

データを表示するため、項目属性を配置し、UI をデザインします。

Web Layout エレメントで以下の操作を行います。

① 値上げ率入力欄を表示するため、ツールボックスより「**項目属性/変数**」コントロールを**紫色の枠（MainTable）内**へドラッグアンドドロップ

② 表示された「項目属性／変数を挿入」ダイアログで **AmusementParkName**、**AmusementParkAddress**、**ShowName**、**ShowImage**、**AmusementParkShowDate**、**AmusementParkShowTime** を選択し、[OK] をクリック

③ オブジェクトを**保存**

13-2-4. WPShowList Web パネルオブジェクトの作成

ショーの一覧を表示する画面を実装します。

下記の内容で Web パネルオブジェクトを作成します。

Web パネル定義：

名前	WPShowList
デスクリプション	ショー一覧

13-2-5. WPShowList の変数定義

フィルターとなるパーク名および、一覧から詳細をポップアップするためのリンク実装文字列を変数で実装します。

Variables エレメントで以下の変数を定義します。

Variables 定義：

名前	タイプ	デスクリプション
AmusementParkId	Attribute:AmusementParkId	パーク番号
ShowDetail	VarChar(40)	詳細画面

ヒント：

- フィルターはパーク名の選択を希望していますが、フィルターとして利用する場合、パーク番号を扱うほうが容易なため、パーク番号に基づく変数を定義（後ほどパーク名が選択できるように修正）

13-2-6. WPShowList の Web Layout エレメント定義

パーク名によるフィルター項目と、ショーの一覧を表示するグリッドをデザインします。

Web Layout エレメントで以下の操作を行います。

① フィルターとなるパーク名入力欄を表示するため、ツールボックスより「**項目属性/変数**」コントロールを**紫色の枠（MainTable）内**へドラッグアンドドロップ

② 表示された「項目属性／変数を挿入」ダイアログで **AmusementParkId（変数）**を選択し、[OK] をクリック

③ 配置した変数コントロールを**選択**し、**[F4] キー**を押下

④ 表示された **[プロパティ] ウィンドウ**で **[Control Type]** プロパティを **[Dynamic Combo Box]** と設定

⑤ **[Item Descriptions]** プロパティで **[AmusementParkName]** を選択

⑥ **[Empty Item]** プロパティを **[True]** に設定

⑦ データの一覧を実装するため、ツールボックスより「**グリッド**」コントロールを**紫色の枠（MainTable）内下部**へドラッグアンドドロップ

⑧ 表示された「項目属性／変数を挿入」ダイアログで **AmusementParkId（項目属性）**、**ShowId**、**ShowName**、**ShowImage**、**AmusementParkName**、**ShowDetail（変数）**を選択し、[OK] をクリック

⑨ 配置したグリッドコントロールを**選択**し、**[F4] キー**を押下

⑩ 表示された **[プロパティ] ウィンドウ**で **[Base Trn]** プロパティを **[AmusementPark.Show]** と設定

⑪ **[Order]** プロパティを **[ShowName]** と設定

⑫ **[Conditions]** プロパティに以下の条件を設定

```
AmusementParkId = &AmusementParkId when not &AmusementParkId.IsEmpty()
```

⑬ グリッド内に配置した AmusementParkId 項目属性を**選択**し、**[F4] キー**を押下

⑭ **[Visible]** プロパティを **[False]** に設定

⑮ グリッド内に配置した ShowId 項目属性を**選択**し、**[F4] キー**を押下

⑯ **[Visible]** プロパティを **[False]** に設定

⑰ グリッド内に配置した ShowDetail 変数を**選択**し、**[F4] キー**を押下

⑱ **[Read Only]** プロパティを **[True]** に設定

⑲ 配置したコントロールは下図のようになっていることを確認

ヒント：

- フィルター項目は Dynamic Combo Box タイプにすることでデータベースに記録されたデータを選択肢として表示可能
- パーク番号、ショー番号は内部的に利用するため、非表示として配置
- ShowDetail 変数はリンクとして利用したいため、読み取り専用に設定

13-2-7. WPShowList の Events エレメント定義

詳細画面をポップアップして表示するために必要な処理を実装するため、以下のように Events エレメントを定義します。

Events 定義：

```
Event Start
      &ShowDetail = ”表示”
Endevent

Event &ShowDetail.Click
      OneShowDetail.Popup(AmusementParkId,ShowId)
Endevent
```

ヒント：

- Start イベントは画面を表示した際に 1 度だけ実行
- ShowDetail 変数の値はオブジェクトの利用中に変更しないため、Start イベントで代入
- 特定のコントロールをクリック時に実行されるイベントを定義する場合、「<コントロール名>.Click」を利用
- 詳細画面はポップアップで表示する必要があるため、「<オブジェクト名>.Popup()」を利用

13-2-8. アプリケーションの実行

作成した「WPShowList」Web パネルを実行します。

[F5] キーを押し、アプリケーションを**実行**します。

表示された [Launchpad] ウィンドウから「WPShowList」Web パネルにより生成された画面を開きます。

一覧にアミューズメントパークに紐づけられたすべてのショーが表示されることの確認と、アミューズメントパークのフィルターによって選択したアミューズメントパークのみのショーが表示されること、「表示」リンクのクリックで選択した行の詳細情報が表示できることを確認します。

13-3. アミューズメントパークのランキング

ユーザーが要望した画面の内容について詳細を確認したところ、要望としては次の通りです。

- 一覧には、**パーク名**とパークに紐づけられた**ショーの数**を表示
- **ショーの数**が**多い順**に並び替え
- **ショーの数**は、この機能でのみ必要な値

この情報を基に「**WPParkList**」Web パネルを作成します。

13-3-1. SDT オブジェクトの作成

ユーザーが要望した内容からショーの数はこの機能のみで利用するため、項目属性として定義する必要はありません。
この場合、集計結果を表示するためには、ローカル式が必要となりますが、グリッドコントロールに配置された変数へローカル式の結果を代入した場合、並び替えることができません。

そのため、このケースでは構造化データタイプを利用し、ロードした値の並べ替え機能を利用して要望を実現します。
そのために、下記の内容で SDT オブジェクトを作成します。

SDT 定義：

名前	SDTPark
デスクリプション	SDTPark

ヒント：

- SDT オブジェクトの作成は「新規オブジェクト」ダイアログで、[カテゴリ] から「データ管理」を選択し、[タイプを選択] 内で「Structured Data Type」を選択

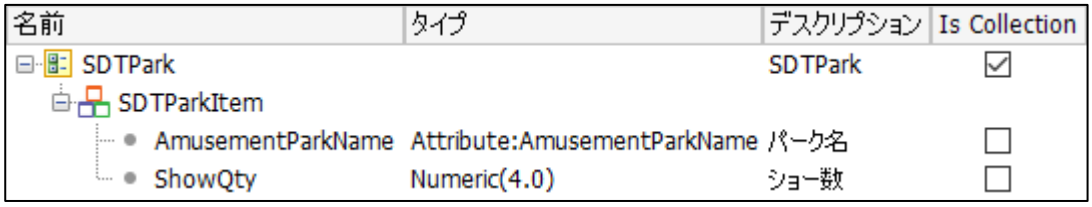
13-3-2. SDTPark の Structure エレメント定義

パーク名、ショーの数をセットにし、複数データを格納できる構造化データタイプを定義するため、Structure エレメントで以下のように定義します。

Structure 定義：

名前	タイプ	デスクリプション
AmusementParkName	Attribute:AmusementParkName	パーク名
ShowQty	Numeric(4.0)	ショー数

また、SDTPark（最上位のノード）の「Is Collection」列にチェックを入れ、下図のような表示となるように定義します。



13-3-3. データプロバイダーオブジェクトの作成

構造化データタイプのコレクションを読み込む必要があるため、GeneXus で構造化データタイプを出力することに適したデータプロバイダーオブジェクトを作成します。

データプロバイダー定義：

名前	DPPark
デスクリプション	パーク読み込み

ヒント：

- データプロバイダーオブジェクトの作成は「新規オブジェクト」ダイアログで、[カテゴリ] から「データ管理」を選択し、[タイプを選択] 内で「Data Provider」を選択

13-3-4. DPPark の Source エレメント定義

出力内容を指定するため、Source エレメントで以下の操作を行います。

① [KB エクスプローラー] ウィンドウから「SDTPark」を Source エレメントへドラッグアンドドロップ

② 自動定義された内容を以下のように修正（文字色が緑のものは自動定義された行、赤は定義済みから変更した行です）

```
SDTPark From AmusementPark
{
	SDTParkItem
	{
		AmusementParkName
		ShowQty = Count(ShowName)
	}
}
```

ヒント：

- SDT オブジェクトを Source エレメントにドラッグアンドドロップすることで、構造の自動定義が行われるだけでなく、データプロバイダーの [Output] プロパティも設定される
- データプロバイダーにおいてイコールの左辺が項目属性と一致し、割り当てる値（右辺）も同名の項目属性となる場合、割り当ての記述を割愛することが可能

13-3-5. Web パネルオブジェクトの作成

ショーの数が多いものから表示されるアミューズメントパークのランキングを表示する画面を実装します。

下記の内容で Web パネルオブジェクトを作成します。

Web パネル定義：

名前	WPParkList
デスクリプション	パーク一覧

13-3-6. WPParkList の変数定義

データを表示するための SDT 型変数の定義が必要となるため、Variables エレメントで以下の変数を定義します。

Variables 定義：

名前	タイプ	デスクリプション
SDTPark	SDTPark	SDTPark

13-3-7. WPParkList の Web Layout エレメント定義

パークの一覧を表示するグリッドをデザインします。

Web Layout エレメントで以下の操作を行います。

① SDT 型変数に基づき、グリッドコントロールが生成されるため、ツールボックスより「**項目属性/変数**」コントロールを**紫色の枠（MainTable）内**へドラッグアンドドロップ

② 表示された「項目属性／変数を挿入」ダイアログで **SDTPark** を選択し、[OK] をクリック

③ 表示された「**メンバーを選択**」ダイアログで上部にあるチェックボックス「**フリースタイルグリッド**」をチェックし、「 [OK] をクリック

④ 配置された**グリッド内**の 2 列「パーク名」、「ショー数」**両方を選択**した状態で、**[F4] キー**を押下

⑤ 表示された [プロパティ] ウィンドウで **[Read Only] プロパティ**を **[True]** に設定

⑥ 配置したコントロールは下図のようになっていることを確認

13-3-8. WPParkList の Events エレメント定義

SDT 型変数にデータプロバイダーの戻り値を代入する処理を実装するため、以下のように Events エレメントを定義します。

Events 定義：

```
Event Start
      &SDTPark = DPPark()
      &SDTPark.Sort("[ShowQty]")
Endevent
```

ヒント：

- データの代入は画面表示時に 1 度だけ実施で十分なため、Start イベントで実装
- Sort メソッドを利用することで、SDT 型コレクション内のアイテムの並びを変更可能
- Sort メソッドでは、「(　)」内に文字列で SDT オブジェクトにおける定義のアイテム名を記述することで、並び替え基準を指定
- アイテム名を「[　]」でくくることで、降順の指定（未指定の場合、昇順）

13-3-9. アプリケーションの実行

作成した「WPParkList」Web パネルを実行します。

[F5] キーを押し、アプリケーションを**実行**します。

表示された [Launchpad] ウィンドウから「WPParkList」Web パネルにより生成された画面を開きます。

一覧にパーク名とともにパークに紐づけられたショーの数が表示され、ショーの数が多いものから順に表示されていることを確認します。

13-4. アプリケーション全体の共通部分変更

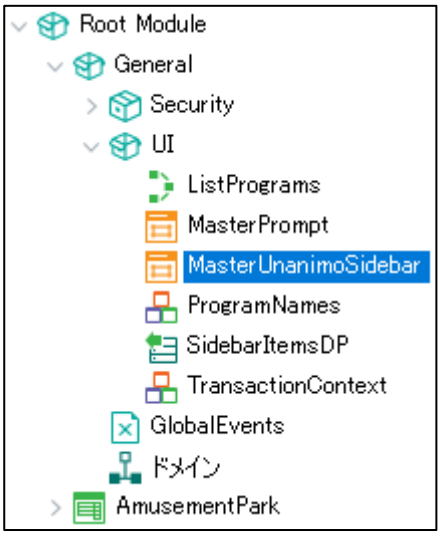
アプリケーションにおいてどの画面を開いた場合も共通で画面上部に「Application Name」という文字列が表示されているため、この文字列を「アミューズメントパーク管理」へ変更してほしいと要望を受けました。

この要望を「MasterUnanimoSidebar」Web マスターパネルに実装します。

13-4-1. MasterUnanimoSidebar の表示

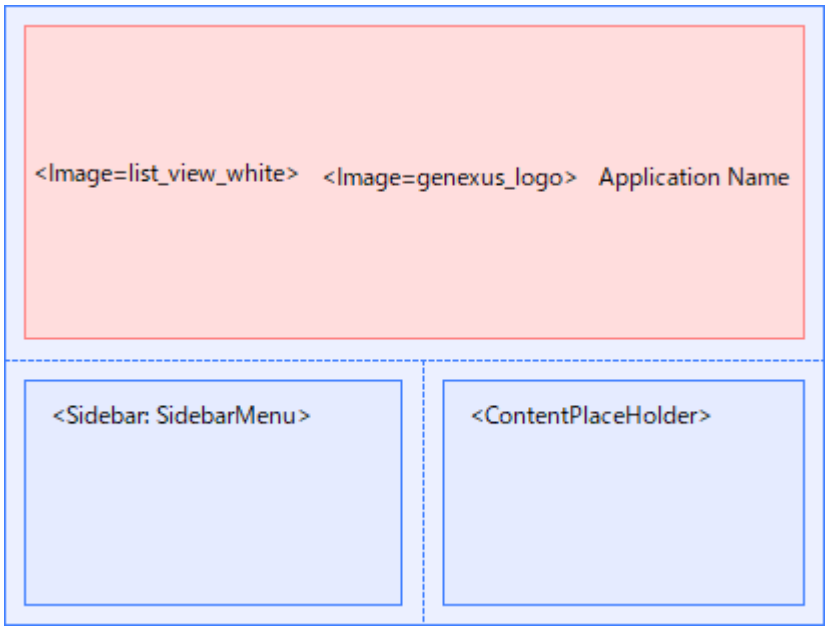
このナレッジベースでは、既定の Web マスターパネル「MasterUnanimoSidebar」が利用されています。

このオブジェクトを修正するため、[KB エクスプローラー] ウィンドウより下図の通り General モジュール内の UI モジュールを展開し、格納された「MasterUnanimoSidebar」を開きます。



13-4-2. Web Layout の編集

MasterUnanimoSidebar の Web Layout エレメントは下図のように実装されています。



このうち、ピンクの枠で表示されている「Flex」というコントロール内に、

「Application Name」という文字列を表示させるためのテキストブロックコントロールが配置されています。

アプリケーション名を指定するため、以下の操作を行います。

① **「Application Name」と表示しているテキストブロックコントロール**を**選択**し、**[F4] キー**を押下

② 表示された **[プロパティ] ウィンドウ**で **[Caption]** プロパティを **[アミューズメントパーク管理]** と変更

③ オブジェクトを**保存**

13-4-3. アプリケーションの実行

「MasterUnanimoSidebar」Web マスターパネルの変更結果を確認します。

[F5] キーを押し、アプリケーションを**実行**します。

表示された [Launchpad] ウィンドウから任意のリンク（例：DeleteAllRepairBC）を開きます。

画面上部に表示される文字列が「アミューズメントパーク管理」となっていることを確認します。

14. デザインシステム

管理会社の担当者が出来上がった機能を確認していたところ、いくつかの画面間でデザインが異なることが確認できました。

そのため、デザインの統一に関するリクエストが発生しました。

この課題ではこの統一しなければならないデザインのうち、「国の一覧（WPCountryList）」のグリッドのデザインを Work With for Web パターンを適用した画面と統一する実装のみを行います。

「**WWCountry**」Web パネルの定義を参考に「**WPCountryList**」のカスタマイズを実施します。

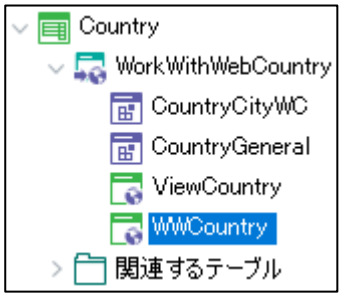
14-1. グリッドコントロールのクラス

「WWCountry」におけるグリッドコントロールのクラス設定を確認し、同様の設定を「WPCountryList」のグリッドコントロールへ適用します。

14-1-1. WWCountry を開く

「WWCountry」は、Country トランザクションへ Work With for Web パターンを適用した結果、自動で生成された国の一覧を表示する Web パネルオブジェクトです。

これを開く場合、[KB エクスプローラー] ウィンドウで下図のように Country トランザクションの子ノードを展開し、該当オブジェクトを開きます。



14-1-2. WWCountry のグリッドコントロール

表示された WWCountry の Web Layout エレメントにおいて以下の操作を行い、設定されているクラスを確認します。

① Web Layout 内の**グリッド**コントロールを**選択**し、**[F4] キー**を押下

② 表示された **[プロパティ] ウィンドウ**で **[Class]** プロパティを参照

値が「ww__grid」となっていることが確認できました。

14-1-3. WPCountryList のグリッドコントロール

確認したクラスと同クラスを WPCountryList のグリッドコントロールにも適用を行います。

Web Layout エレメントにおいて以下の操作を行います。

① Web Layout 内の**グリッド**コントロールを**選択**し、**[F4] キー**を押下

② 表示された **[プロパティ] ウィンドウ**で **[Class]** プロパティを **[ww__grid]** に変更