

ビジネスコンポーネントを 使用したデータベースの更新

Webフォームを使用しないトランザクションで
データを更新する方法

GeneXus™

シナリオ

- トランザクションを使用して、画面(フォーム)からデータベースのレコードを挿入、更新、削除

The image shows two screenshots of a GeneXus form titled 'カテゴリ' (Category). The left screenshot shows the '実行' (Execute) button highlighted with a red arrow and the label '挿入' (Insert). The right screenshot shows the '実行' (Execute) button highlighted with a red arrow and the label '更新' (Update), and the '削除' (Delete) button highlighted with a red arrow and the label '削除' (Delete). Both forms have fields for 'カテゴリ番号' (Category Number) and 'カテゴリ名' (Category Name). The right form also has a '終了' (End) button.

- ビジネスコンポーネント (BC) を使用方法もある

これまで、トランザクションを使用してデータベースのレコードの挿入、更新、削除を行っていました。その際には画面に表示されているボタンや UI 部品を使用していました。

ここでは、ビジネスコンポーネントの概念を適用し、ほかの GeneXus オブジェクトからコードを記述して同じ処理を行う方法を説明します。

現在の状況

- 現在「東京スカイツリー」という観光名所（id=6）が東京にあり、カテゴリは「有名なランドマーク」として登録されています。

新しい要件

- 有名なランドマークや博物館など、種類を問わず大勢の人が訪れる観光名所を表すため、新たに「人気/おすすめ」カテゴリを設ける。
- 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」カテゴリに変更する。

ここでは、次のような例を考えます。旅行代理店が、さまざまな人気の観光名所を訪れる旅行プランを提供するため、大勢の旅行客が訪れる観光名所を適切に分類する必要があります。

たとえば、東京ではランドマークに人気があるため、これまでの「有名なランドマーク」カテゴリではなく「人気/おすすめ」カテゴリに分類することにします。「人気/おすすめ」カテゴリは、これまでの分類に関わらず、訪れる人の多い観光名所を表す新しいカテゴリです。

それでは、新しいカテゴリを作成して、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」カテゴリに変更するアプリケーションを実装してみましょう。

エラールールを追加

The screenshot displays the GeneXus IDE interface with two windows open, each showing the 'Rules' tab for a specific table.

Category Window:

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

Rules list:

- 1 Error("カテゴリ名を入力して下さい")
- 2 if CategoryName.IsEmpty();
- 3

Attraction Window:

名前	タイプ	デスクリプション	式	Nu
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		
CityId	Id	都市番号		Yes
CityName	Name	都市名		
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

Rules list:

- 1 Error("観光名所名を入力して下さい")
- 2 if AttractionName.IsEmpty();
- 3

この要件を実装するため、アプリケーションでは、Category テーブルに新しいカテゴリを挿入し、都市「東京」とカテゴリ「有名なランドマーク」という条件でフィルタして Attraction テーブルの観光名所を検索し、カテゴリコードを「人気/おすすめ」に変更することが求められています。

この実装を開始する前に、名前が空欄のままの状況を制御するため、Category トランザクションと Attraction トランザクションにエラールールを追加します。エラールールが必要となる理由については、後述します。

Category トランザクションのビジネスコンポーネント

The screenshot displays the GeneXus IDE interface. At the top, the 'Category * X' window shows the 'Structure' tab with a table listing fields: 'Category' (Id, カテゴリ番号, No), 'CategoryId' (Id, カテゴリ番号, No), and 'CategoryName' (Name, カテゴリ名, No). Below this, the 'InsertCategoryUpdateAttractions * X' window shows the 'Variables' tab with a variable 'Category' of type 'Category'. A red arrow points from the 'Category' variable to the 'Category' business component in the 'Type Definition' list. The 'Properties' window for 'BusinessComponent: Category' is open, showing 'Business Component' set to 'True'.

新しいカテゴリの作成と観光名所の更新を行うには、InsertCategoryUpdateAttractions というプロシージャオブジェクトを作成します。

まず、「人気/おすすめ」カテゴリを作成します。これには Category ビジネスコンポーネントを使用します。作成するには、Category トランザクションに移動して **[Business Component]** プロパティを見つけ、値に True を割り当てます。

GeneXus により、Category トランザクションを基に、関連するトランザクションと同様の機能を持つ構造が作成されます。

この定義を使用するには、この Category ビジネス コンポーネント タイプの変数を作成する必要があります。そのため、プロシージャーの [Variables] エレメントに移動し、&category という名前で変数を作成します。GeneXus により、自動的に Category データタイプが割り当てられます。このデータタイプは、Category ビジネスコンポーネントに対応しており、Category トランザクションから作成されます。

変数のプロパティに移動して [Data Type] をクリックすると、[Business Component] というグループが見つかります。グループを開くと、トランザクション用アイコンの付いた Category ビジネスコンポーネントが見つかります。GeneXus によって、Category トランザクションに基づく新しいデータタイプが作成されています。このデータタイプは、Category トランザクションとさまざまな特性を共有しています。

Category トランザクションの挿入方法

Application Name

Recents カテゴリ

カテゴリ

カテゴリ番号 0

カテゴリ名

実行 終了

```
Error("カテゴリ名を入力して下さい")  
if CategoryName.IsEmpty();
```

カテゴリ

カテゴリ番号 0

カテゴリ名 人気/おすすめ

実行 終了

ビジネスコンポーネントでの実装を進める前に、ブラウザでのトランザクションオブジェクトの操作を確認しておきましょう。

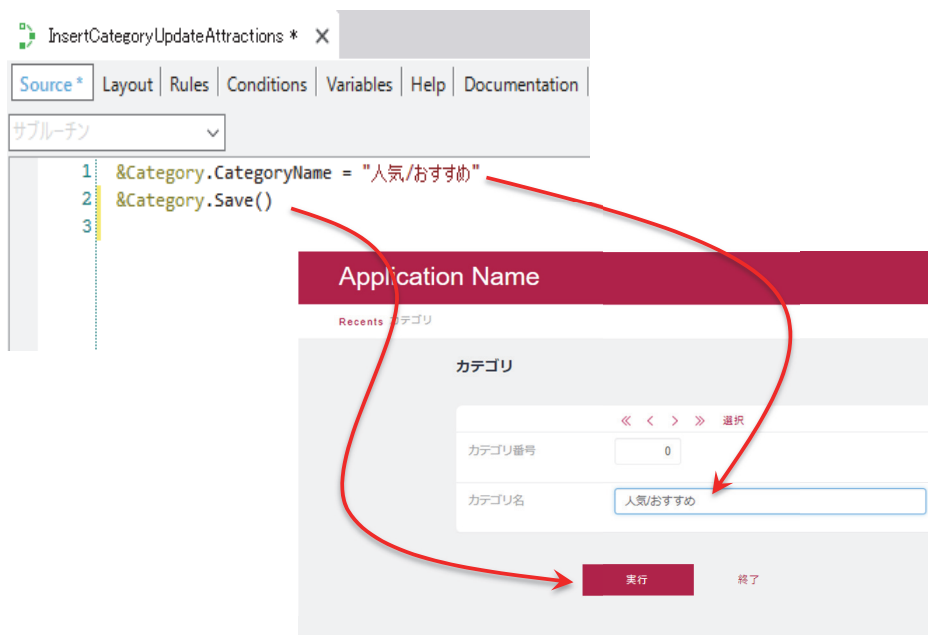
エンドユーザーがブラウザからカテゴリトランザクションを呼び出した際 (パターンを適用していない既定の状態)、新しいレコードを追加できる状態で画面が開きます。

まず、カテゴリ ID を確認します。CategoryId 項目属性は自動的に番号が割り当てられるため、値を割り当てる必要はありません。そのため、空欄のまま Tab キーを押して、このフィールドを離れます。

次に、カテゴリ名を入力します。このフィールドを空欄のままにするとどうなるでしょうか。CategoryName 項目属性に値を入力しなかった場合に制御を行うエラールールがトリガーされます。

カテゴリ名に「人気/おすすめ」と入力します。そして最後に、エンドユーザーは各項目属性に割り当てたデータをデータベースに保存させるため [実行] ボタンを押します。これにより、データ挿入が実行されたとみなされます。

Category ビジネスコンポーネントのレコードの挿入方法



ビジネスコンポーネントを使用した場合でも、手順はまったく同じです。新しいレコードを挿入するには、カテゴリの各フィールドに値を割り当てて、変更内容を確認します。

そのためには、&category 変数を [Source] に挿入します。ピリオドを押すと、利用可能なメソッド (Save、Delete、Check など) が表示されます。また、CategoryId と CategoryName が表示され、値を割り当てることができます。

カテゴリ ID は自動的に番号が割り当てられるため、値は割り当てません。既定では、挿入するものと想定されます。トランザクションの画面を使用する場合と同じように、カテゴリ名を指定します。

今回の例では、トランザクションの画面を使用している場合は、カテゴリ名を割り当てた後に [実行] ボタンを押しますが、ビジネスコンポーネントを使用している場合は、次のように Save() メソッドを使用します。

```
&category.CategoryName = "人気/おすすめ"
&category.Save()
```

&category などの各ビジネスコンポーネント変数に対して Save 処理を実行すると、メッセージのコレクションがメモリ内にロードされ、その結果として、すべての警告メッセージやエラーメッセージが生成されます。

このメッセージを表示する非常に簡単な取得および表示方法もありますが、それについてはここでは考慮しません。

観光名所 6 (東京スカイツリー) のカテゴリの変更方法

ビジネスコンポーネント

VS

トランザクション

ビジネスコンポーネント

```

1 &Category.CategoryName = "人気/おすすめ"
2 &Category.Save()
3
4 &Attraction.Load(6)
5 &Attraction.CategoryId = &Category.CategoryId
6
7 &Attraction.Save()
8

```

トランザクション

観光名所

観光名所番号 6

観光名所 東京スカイツリー

観光名所ふりがな とうきょうすかいつりー

国番号 5

国名 日本

都市番号 1

都市名 東京

カテゴリ番号 4

カテゴリ名 人気おすすめ

観光名所画像

実行 終了

Tab キー

ビジネスコンポーネントを使用して作成したカテゴリの ID に付与された値を使用する

ここまでで、「人気/おすすめ」カテゴリの作成が実装されました。次に日本のランドマークのカテゴリを変更しましょう。

変更が必要なのは、東京スカイツリー (Id = 6) のカテゴリだけだとします。トランザクションオブジェクトを画面から操作する場合は、何を行っていたでしょうか。Attraction トランザクションを開いて ID に「6」を挿入し、フィールドを離れます。GeneXus によって東京スカイツリーのすべてのデータが取得されます。対象となる値 (今回の場合はカテゴリ) を変更し、[実行] を押して保存します。

[実行] をクリックすると変更内容が反映され、該当するすべてのルールがトリガーされます。たとえば、トランザクション名を削除していた場合、更新は失敗します。また、存在しないカテゴリ ID に変更した場合、エラーが発生します。GeneXus によって、関連するテーブルの値の一貫性が制御されるため (参照整合性)、このような更新は実行できません。

こういった挙動はビジネスコンポーネントの場合でも同じです。

トランザクションに基づくため、Load コマンドを使用し、観光名所の ID を括弧内に挿入します (今回の場合は「6」)。新しいカテゴリの値 (作成した「人気/おすすめ」カテゴリの ID 値) を割り当てて、Save() を実行して完了します。

&Attraction.Load(6)

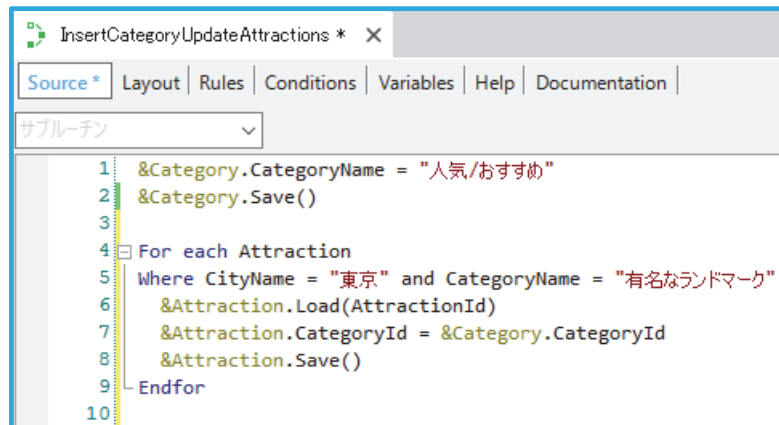
// ビジネスコンポーネントを使用して作成したカテゴリの ID 値を使用します。

&Attraction.CategoryId = &category.CategoryId

&Attraction.Save()

存在するレコードの Load を記述したため、Save メソッドは、この処理がレコードの (追加ではなく) 更新が求められていると判断します。

Attraction ビジネスコンポーネントを使用して、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変更する方法

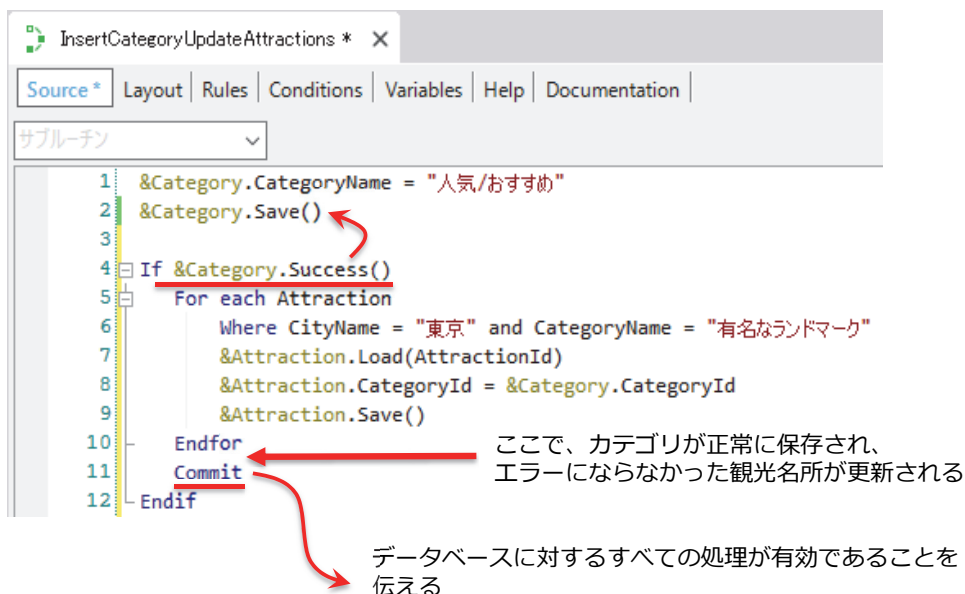


```
InsertCategoryUpdateAttractions * X
Source * Layout Rules Conditions Variables Help Documentation
サブルーチン
1  &Category.CategoryName = "人気/おすすめ"
2  &Category.Save()
3
4  For each Attraction
5      Where CityName = "東京" and CategoryName = "有名なランドマーク"
6          &Attraction.Load(AttractionId)
7          &Attraction.CategoryId = &Category.CategoryId
8          &Attraction.Save()
9  Endfor
10
```

ただし、要件で求められているのは、東京スカイツリーのカテゴリだけを変更することではなく、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のカテゴリを変更することです。そのため、該当するすべての観光名所を検索します。

For each を使用してレコードを検索し、反復処理によって発見された AttractionId をロードします。

ビジネスコンポーネントが正常に保存されたかどうかを確認する方法と、データベース処理の評価方法



各 Save() は、ビジネスルールや参照整合性に応じて、成功する場合も失敗する場合もあるため、データベース内のレコードの挿入や更新も実行されるとときと実行されない場合があります。

Save() コマンドが成功したかどうかを確認するには、Success() メソッドを使用します。True が返ってきた場合は成功しており、False が返ってきた場合は失敗しています。Save() コマンドが失敗する理由としては、エラールールがトリガーされたことで、保存が許可されなかったケースや、保存時にシステムに障害が発生し、レコードが保存されなかったケースなどが考えられます。また、参照整合性に問題があった可能性もあります。

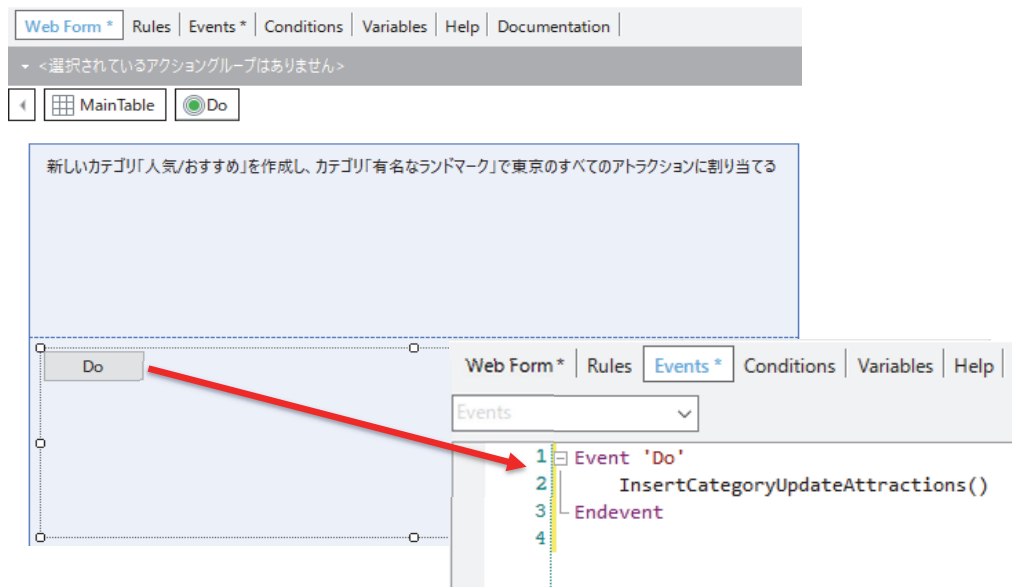
新しいカテゴリの挿入に失敗した場合、カテゴリが存在しないため、観光名所の保存も失敗します。そのため、観光名所の更新は、Success() メソッドの結果によって決まります。

次に、レコード群が適切に追加/更新できたことをデータベースに通知し、これらの変更をシステム障害で喪失しないように反映させます。そのためには、Commit コマンドを追加します。

End for が処理される段階で、カテゴリが正常に登録され、観光名所が正常に更新されています (更新を許可しないビジネスルールによってエラーは生成されないため)。次に、すべての更新内容が有効であるとデータベースに伝える必要があります。これまで、1 つのレコードを追加し、いくつかのレコードを更新しました。システムに障害が発生した場合 (停電による障害など)、Commit が実行されていない処理をすべて元に戻すことで、データベースの回復処理が行われます。

Commit は特別なコマンドで、ひとまとまりとして実行される処理ブロックの終わりをデータベースに伝えることができます (Commit できない場合、すべての処理が破棄されます)。このコマンドを実行すると、データベースは、システム障害や停電の発生時にもデータが喪失されないように、データを保存します。

Web パネルからプロシーチャーを呼び出す



プロシーチャーを実行するには、それを呼び出す別の GeneXus オブジェクトが必要です。

この処理は、Web Panel オブジェクトから実行します。Web パネルは、ニーズに応じて、データベースのデータの表示や、システムの開始画面の作成、データの入力など、さまざまな機能を実行するためのツールです。

今回の例では、作成したプロシーチャーを呼び出すボタンを Web パネルに設置します。

ボタンに関連付けられたイベントで、InsertCategoriesUpdateAttractions プロシーチャーを呼び出します。このプロシーチャーにはパラメーターは必要ありません。

変更前

F5

観光名所s

Q 観光名所名

観光名所番号	観光名所名	観光名所ふりがな	国番号	国名	都市名	カテゴリ名	観光名所画像		
3	エッフェル塔	えっふえるとう	2	フランス	パリ	モニュメント		更新	削除
4	コロンバードのキリスト像	こるこばーどのかりすとぞう	1	ブラジル	リオデジャネイロ	モニュメント		更新	削除
5	スミソニアン博物館	すみそにあんはくぶつかん	4	アメリカ合衆国	ワシントン	美術館		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	パリ	美術館		更新	削除
6	東京スカイツリー	とうきょうスカイツリー	5	日本	東京	有名なランドマーク		更新	削除
2	万里の長城	ばんりのちょうじょう	3	中国	北京	モニュメント		更新	削除

アプリケーションを実行すると、日本の観光名所のカテゴリは「有名なランドマーク」になっています。

詳細情報を確認すると、どちらも東京にあることが分かります。

変更後

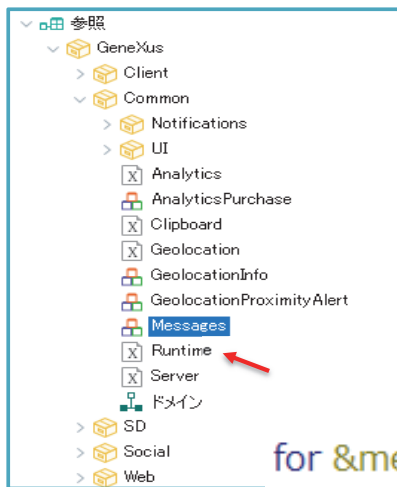
F5

Application Name	観光名所s						
Recents Web Panel2	Q 観光名所名						
新しいカテゴリ「人気/おすすめ」を作成し、カテ Do	観光名所番号	観光名所名	観光名所ふりがな	国番号	国名	都市名	カテゴリ名
	3	エッフェル塔	えっふえるとう	2	フランス	パリ	モニュメント
	4	コルコバードのキリスト像	こるこばーどのきりすとぞう	1	ブラジル	リオデジャネイロ	モニュメント
	5	スミソニアン博物館	すみそにあんはくぶつかん	4	アメリカ合衆国	ワシントン	美術館
	1	ルーブル美術館	るーぶるびじゅつかん	2	フランス	パリ	美術館
	6	東京スカイツリー	とうきょうスカイツリー	5	日本	東京	人気/おすすめ
	2	万里の長城	ばんりのちやうじょう	3	中国	北京	モニュメント

CategoriesAndAttractions Web パネルを開いて [実行] ボタンを押すと、観光名所が返されます。「人気/おすすめ」カテゴリが作成されており、東京の「有名なランドマーク」カテゴリに属していた 2 つの観光名所が「人気/おすすめ」カテゴリになっていることが分かります。

カテゴリのケースでは、GeneXus によって、「挿入」を実行すると推論され、2 つ目のケースで観光名所のカテゴリの値を変更したときは、「更新」であることが推論されました。これは、トランザクションの場合と同じです。プロシージャー内で Save() メソッドを使用したのはそのためです。Save() メソッドは、実行する処理に関わらずデータを保存できます。

Errorルールで定義したエラーメッセージはどこに表示されるのか？



画面ではなく B C で名前を空白にした場合、エラーメッセージはどこに表示されるか？

Messages [読み取り専用] X			
Structure		Documentation	
名前	タイプ	Is Collection	デスクリプション
Messages			Messages
Message			
• Id	VarChar(128)	<input type="checkbox"/>	Id
• Type	MessageTypes, GeneXus	<input type="checkbox"/>	Type
• Description	VarChar(256)	<input type="checkbox"/>	Description

```
for &message in &Attraction.GetMessages()
  msg(&message.Description)
endfor
```

ビジネスコンポーネントを使用した場合の特徴として、ビジネスコンポーネントを作成する際に使用したトランザクション内に定義されているルールもトリガーされます。そのため、今回の例でカテゴリ名に値を割り当てなかった場合、Category トランザクションのフォームを使用した場合と同様に、この処理の制御を行うエラールールがトリガーされます。このエラールールは、割り当てが実行されただけではトリガーされませんが、ビジネスコンポーネントが保存 (Save メソッド) を指示されたときはトリガーされます。

コードを通じてビジネスコンポーネントを実行しているため、必ずしもすべてのルールがトリガーされるとは限りません。たとえば、オブジェクトを呼び出すルールはトリガーされません (別のトランザクションを呼び出すルールや Web パネルを呼び出すルールなど)。また、Parm ルールは、トランザクション内に定義されていたとしても、トリガーされません。

カテゴリ名を指定しなかった場合、レコードは挿入されませんが、フォームを使用していない状況では、エラーメッセージはどこに表示されるのでしょうか？ GeneXus では、アイテムのコレクションである Messages という名前の事前定義済みの SDT を各ナレッジベースに作成し、これにメッセージを記録します。

Insert() メソッドと Update() メソッドによる Save() の限定化

```

&Category.CategoryName = "人気/おすすめ"
&Category.Insert()

If &Category.Success()
  For each Attraction
    Where CityName = "東京" and CategoryName = "有名なランドマーク"
    &Attraction.Load(AttractionId)
    &Attraction.CategoryId = &Category.CategoryId
    &Attraction.Update()
  Endfor
  Commit
Endif

```

Red arrows in the original image point to `&Category.Insert()`, `&Attraction.Update()`, and `&Attraction.CategoryId = &Category.CategoryId`.

Insert を挿入して失敗した場合は Update を実行: `&attraction.InsertOrUpdate()`

InsertOrUpdate() – Insert を試行し、不可能な場合は Update を試行する

Save() メソッドでは、データの挿入か更新かに関わらず、処理を保存できます。

ただし、Insert() メソッドと Update() メソッドを使用して、実行する処理を具体的に指定できます。

コードを更新して、カテゴリに関しては Insert メソッドを、観光名所に関しては Update メソッドを使用するようにします。

なお、観光名所については次のように直接記入すると、Load は必要ありません:

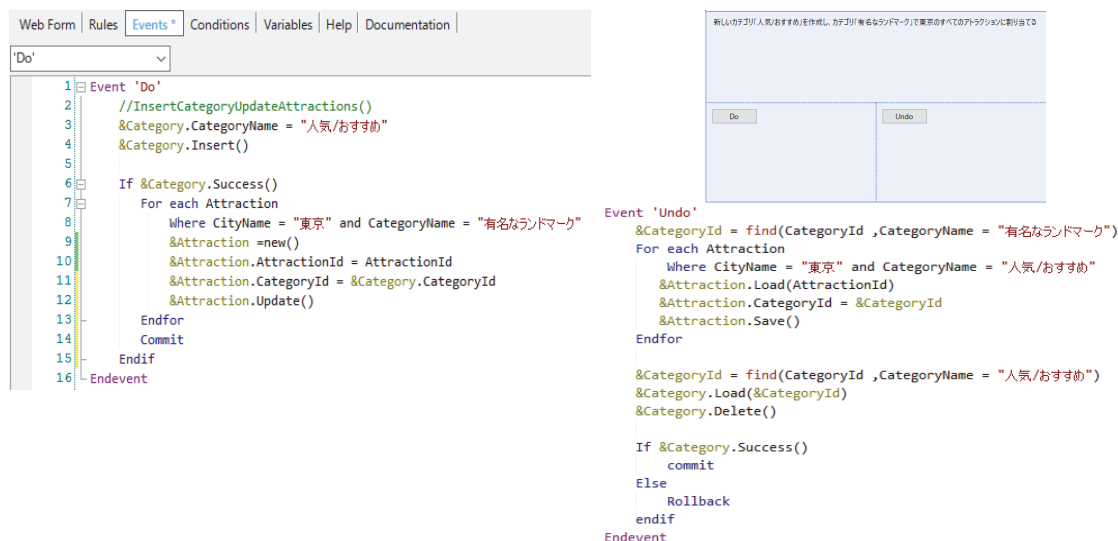
```
&Attraction.AttractionId = AttractionId
```

観光名所に関して Update を実行することを明確にしたため、レコードの更新を希望していることが GeneXus に伝わります。

Insert() および Update() によって、挿入と更新のどちらを希望しているかを指定する以外に、InsertOrUpdate() を使用して、項目属性に新しい値を割り当てたときに Insert が試行されるようにもできます。挿入が不可能な場合 (たとえば、そのキーを持つレコードが既に存在する場合など)、Update が試行されます。

Web パネルから DB に対してデータの挿入や更新を行う

例：「人気/おすすめ」に変更した観光名所のカテゴリを戻す



既に説明したとおり、同一オブジェクト (今回の例ではプロシージャ) 内の 2 つのテーブルを対象に、レコードの挿入や更新を行うことができます。このコードは、Web パネルのイベント内に直接配置することができます。これは、どのオブジェクトからでもビジネスコンポーネントを通じたデータベースの更新を実行できるためです (ただし、トランザクションの場合には制限があります)。

ビジネスコンポーネントは対象の ID を選択して、メソッド(Delete) で、削除できます。また、トランザクション同様、Delete() コードが実行されたときに参照整合性をチェックします。したがって、処理を実行する前に、観光名所のカテゴリを変更し、以前の「有名なランドマーク」カテゴリを復元する必要があります。

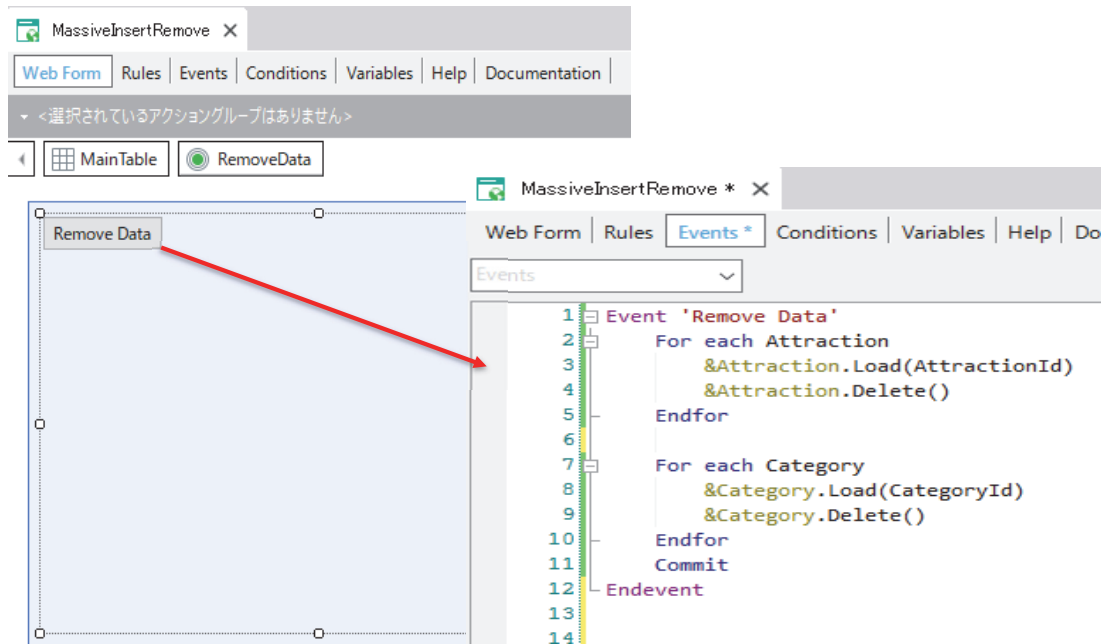
一般的に、ID の数値を記憶していなくても、名前で覚えていることがあるため、ID を覚えようとするよりも、名前を使用してデータベースを検索する方が安全です。したがって、今回の例では、「有名なランドマーク」カテゴリの ID 値を回復する場合、Find 式を使用します。該当するレコードがあると、指定した項目属性の値を返します。

東京の「人気/おすすめ」カテゴリに属する各観光名所を対象に、Attraction ビジネスコンポーネント変数ですべてのデータをロードして、CategoryId のデータだけを「有名なランドマーク」に対応するものに變更して更新します。

これで「人気/おすすめ」カテゴリを削除できるようになりました。最後に Commit を使用して削除内容を確認し、データベース上で確実に削除が実行されるようにします。これは、システム障害が発生して復元時にデータベースによってロールバックが実行される場合に、以前に削除したレコードが復元されることを防ぐためです。

任意の観光名所の更新に失敗した場合、&category の削除も失敗します。したがって、カテゴリの削除が成功しているかどうかを直接確認してから Commit を実行します。成功していない場合は、データベースに対する更新を確認しません。ロールバックを実行して、実行内容をすべて元に戻すことができます。

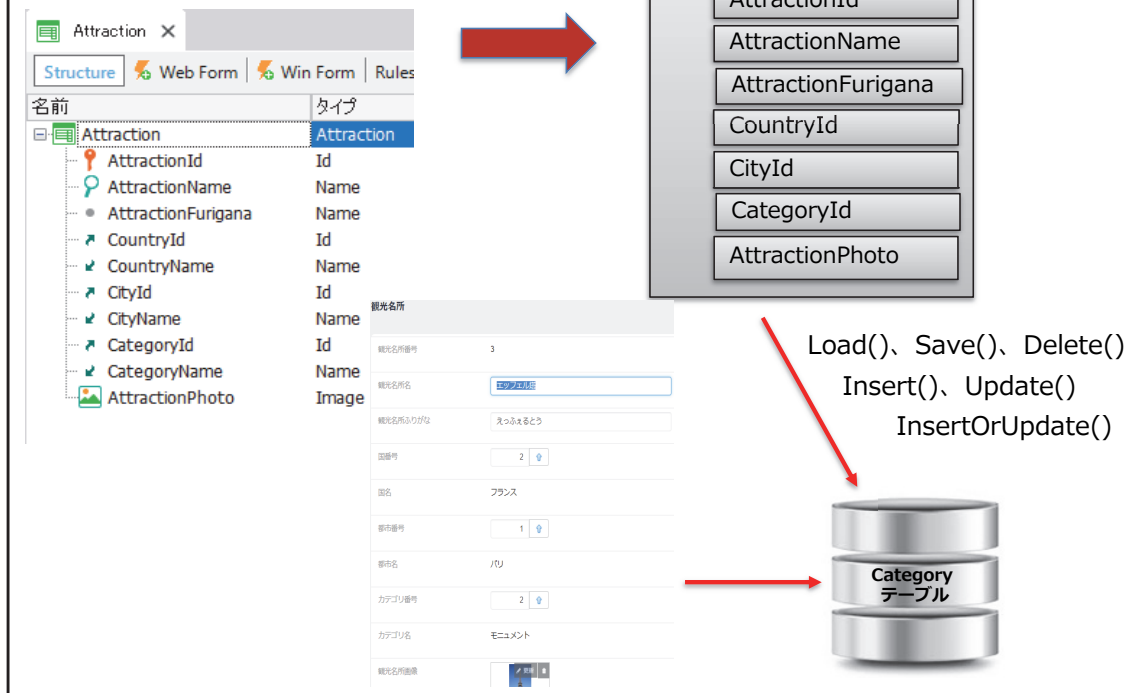
一括削除



既存の観光名所のカテゴリを変更するのではなく、すべて削除したい場合は、Attraction テーブルから該当のレコードをすべて削除します。そのためには、For each と Delete を連携させます。この処理を表示するには、Web パネル「MassiveInsertRemove」を作成して、[データを削除] ボタンを設置します。

この場合、削除を実行する順序が重要となります。ビジネスコンポーネントによって参照整合性が制御されているため、カテゴリを最初に削除することはできません。

まとめ

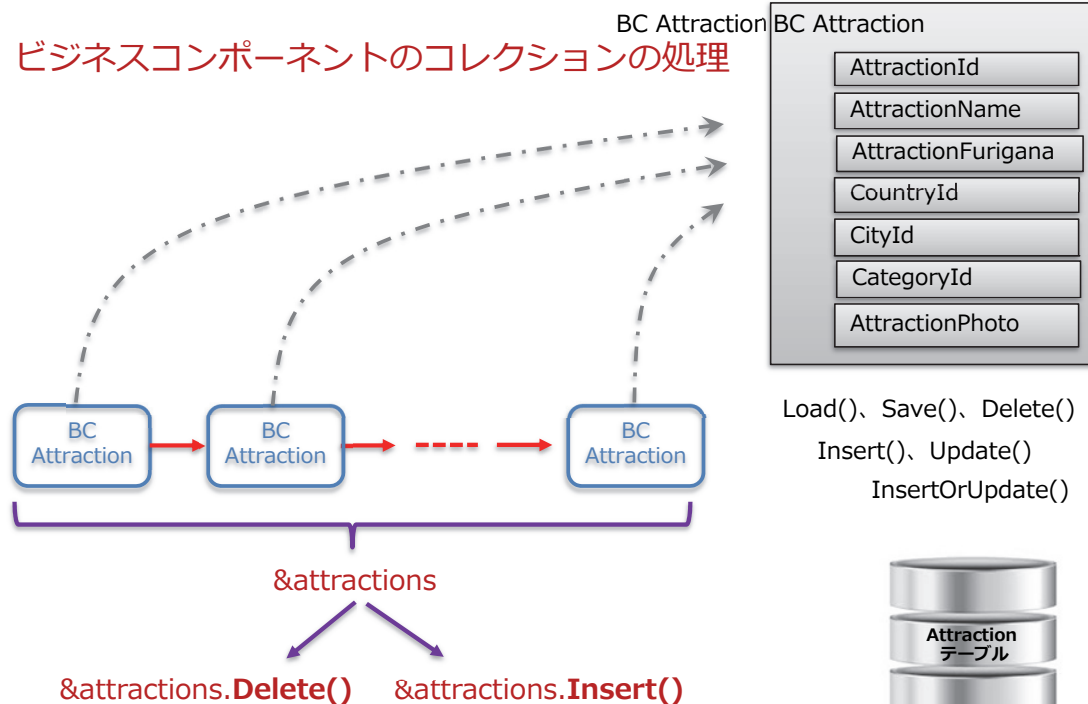


まとめると、ビジネスコンポーネントは、トランザクションをベースとする特殊なデータタイプで、トランザクションのミラーとなります。トランザクションで実行できることは、画面に関すること以外すべて実行できます。

そのため、トランザクションのビジネスコンポーネント (ミラー) では、SDT のようにデータタイプの変数を定義し、トランザクションと同じ構造を持たせることができます。Load、Save、Delete (あるいは Insert、Update、Delete) などのメソッドを通じて、トランザクションの対話形式の画面と同じ処理を実行できます。これにより、トランザクションが実行されているかのようにビジネスルールをトリガーできます。ビジネスルールとは何でしょうか? 画面を通じて対話形式で操作したり、別の画面を呼び出すルールとは関係のないルールです。そのため、ビジネスコンポーネントは、トランザクションを「非会話形式で」実行する方法とも言われます。

Attraction トランザクションのような単一レベルのトランザクションの場合、ビジネスコンポーネントの構造には、関連テーブル内に物理的に存在する項目属性ごとに 1 つの要素が含まれるだけでなく、外部キーを通じて推論されるトランザクションの構造に存在する CategoryName や CountryName、CityName などの項目属性も含まれます。これらはトランザクションの場合と同じ機能を持ち、Load の実行後に値を推論できます。

ビジネスコンポーネントのコレクションの処理



Insert、Update、Delete の処理は、ビジネスコンポーネントのコレクションに対して一括して実行することもできます。ただし、Load は一括して実行することはできません。たとえば、Business Component Attraction タイプの観光名所をすべて &attractions 変数にロードするようなビジネスコンポーネントのコレクション変数を設定した場合、&attractions.Delete() を実行することで、すべてのアイテムを削除できます。コレクション全体を検索して、アイテムごとに個別に削除する必要はありません。

Insert を使用してレコードを一括追加する例については、後述します。

