

# データベースの更新

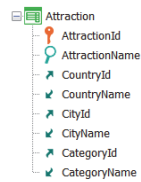
外部キーが Null を受け入れる場合の動作

*GeneXus*<sup>™</sup>

## 1. フォームを使用して更新



Insert、Update、Delete

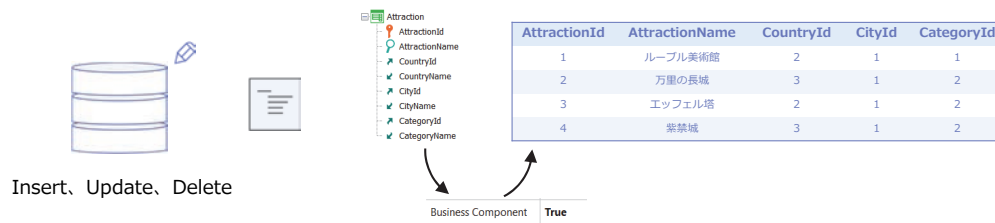


## 2. コードを使用して更新



データベースを更新する方法は2つあります。1つは画面からトランザクションオブジェクトを介してインタラクティブに更新する方法で、もう1つはコードを使用して更新する方法です。

## 1. ビジネスコンポーネント: Insert(), Update(), Delete()



## 2. プロシージャ: New, For Each, Delete

コードを使用して更新する場合、2つのオプションがあります。

1つは、トランザクションのビジネスコンポーネントを使用し、そのメソッドを介して更新する方法です。もう1つは、プロシージャ内で New コマンド、For each コマンド内で変更する項目属性に値を直接割り当て、Delete コマンドを実行する方法です。



一意性のコントロール	参照整合性コントロール	ルール/イベントの実行
ビジネスコンポーネント ✓	✓	✓
プロシージャ (New、For Each、Delete) ✓		

AttractionId	AttractionName	CountryId	CityId	CategoryId
1	ルーブル美術館	2	1	1
2	万里の長城	3	1	2
3	エッフェル塔	2	1	2
4	紫禁城	3	1	2

既に確認したとおり、どちらを選択してもレコードの一意性がコントロールされます。つまり、主キーや候補キーが重複しているレコードは、データベースに保持できません。この例では、同じ ID を持つ 2 つの観光名所を設定することはできません。また、たとえば AttractionName が候補キーであった場合、同じ名前を持つ 2 つの観光名所を設定することもできません。

ビジネスコンポーネントを使用して更新する場合と、プロシージャ内で更新コマンドを使用して直接更新する場合では、大きな違いがあります。トランザクションのルールとイベントによって適用されるロジックを実行するのはビジネスコンポーネントだけです。また、この方法の場合のみ、**プログラムを通じて**参照整合性をコントロールできます。この「プログラムを通じて」とは、どういうことでしょうか。

Name	Type	Description	Formula	Nullable
Attraction	Attraction	Attraction		
AttractionId	Id	Attraction Id		No
AttractionName	Name	Attraction Name		No
AttractionAddress	Address, GeneXus	Attraction Address		No
CountryId	Id	Country Id		No
CountryName	Name	Country Name		No
CityId	Id	City Id		No
CityName	Name	City Name		No
CategoryId	Id	Category Id		Yes
CategoryName	Name	Category Name		No

Name	Type	Description	Formula	Nullable
Category	Category	Category		
CategoryId	Id	Category Id		No
CategoryName	Name	Category Name		No

観光名所を記録する Attraction トランザクションがあり、観光名所は必ず国、都市、および何らかのカテゴリ (遺跡や美術館など) に属している場合、関連テーブルには外部キーがあります。たとえば、CategoryId 項目属性は Category テーブルの外部キーになります。

GeneXus 開発者は、トランザクションに各外部キーの参照整合性コントロールをプログラミングする必要はありません。生成されるプログラムには自動的に含まれます。これはトランザクションのロジックの一部です。

GeneXus™

Attraction

Test

trialapps3.genexus.com/idf180a9d465110c307ff42ea994a10e6/attraction.aspx?UPD,2

Name

The Great Wall

Address

Country Id

3

Country Name

China

City Id

1

City Name

Beijing

Category Id

102

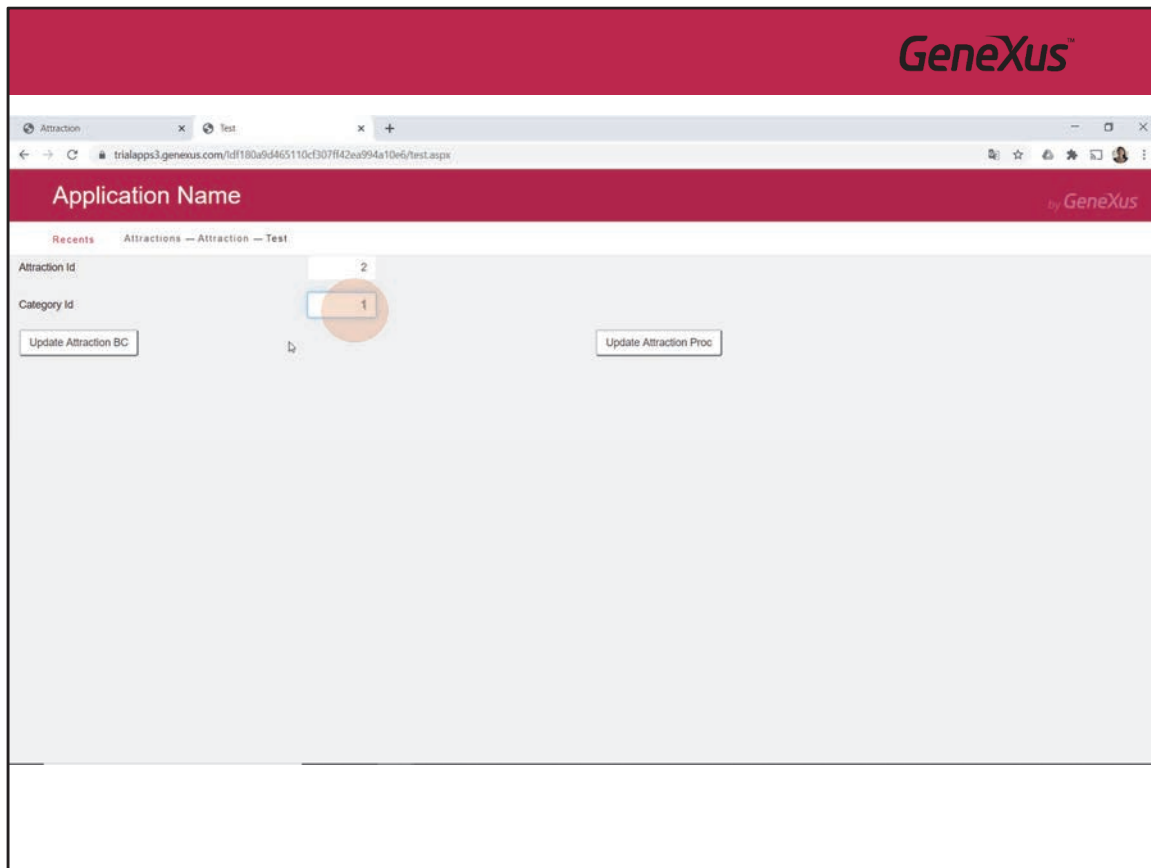
No matching "Category".

Category Name

CONFIRM

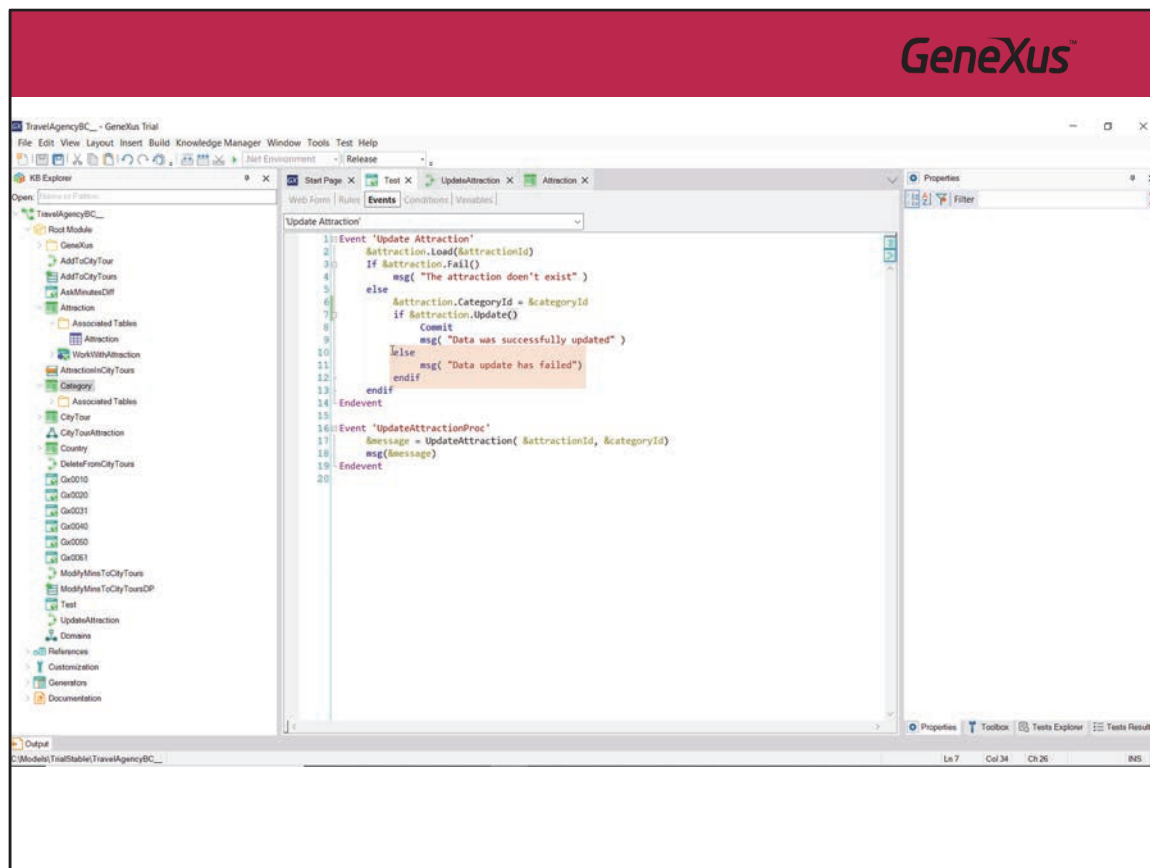
CANCEL

ユーザーが既存の観光名所を存在しないカテゴリに割り当てようとする、それは不可能であることがトランザクションで示されます。データベースで参照整合性コントロールが行われる前に、参照整合性コントロールが行われています。



ユーザーが観光名所の ID とカテゴリの ID を入力して、観光名所のカテゴリを変えることができる Web パネルを実装するとします。

コードを使用して行う場合は 2 つの選択肢があります。1 つは Attraction ビジネスコンポーネントを使用する方法で、もう 1 つはプロシージャーを使用する方法です。



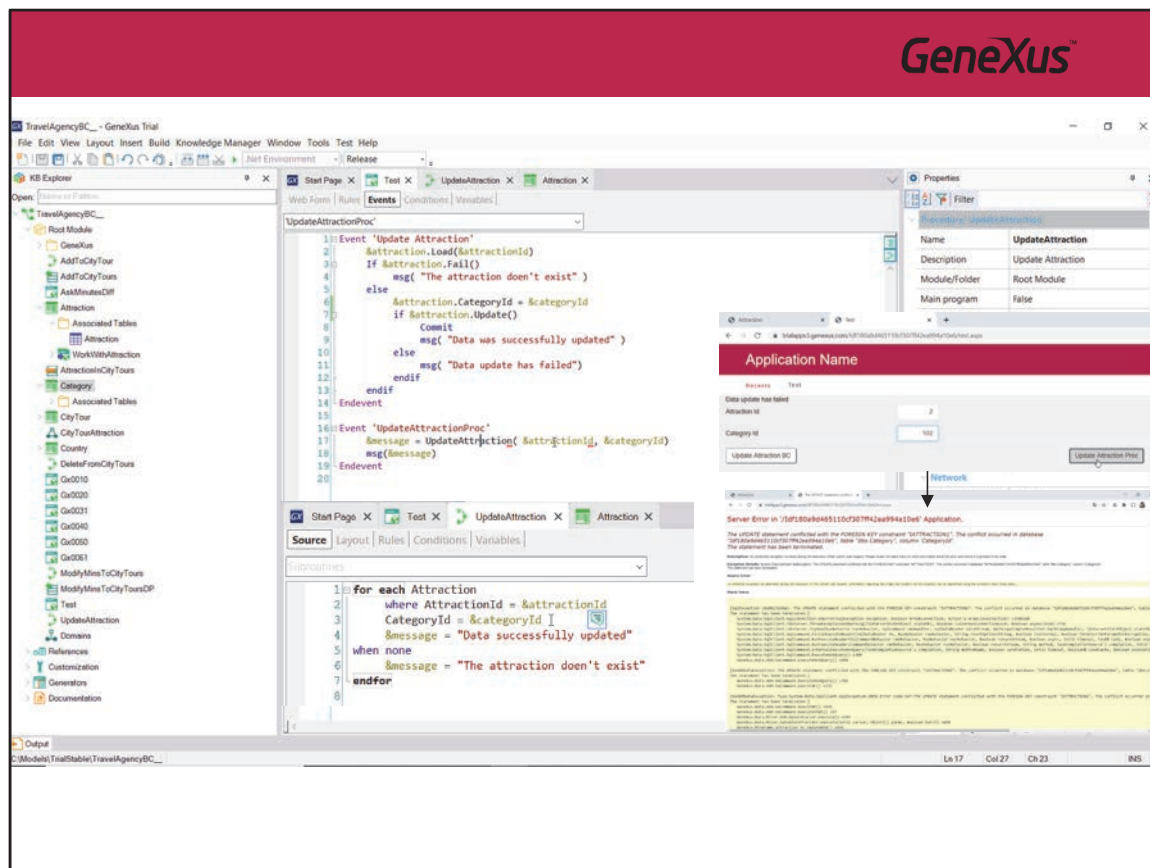
Attraction ビジネスコンポーネントを使用する方法を見てみましょう。

[Business Component] プロパティをアクティブにし、このタイプの変数を定義します。また、ボタンに関連付けられたイベントで、ユーザーが画面で入力した &AttractionId 変数からデータをロードします。

ID が存在し、データをロードできた場合、ユーザーが入力した観光名所のカテゴリを変更して更新を試みます。Update メソッドと関連付けられたコードで、そのカテゴリが存在するかどうかチェックされ、存在する場合にのみデータベースに更新の命令を送るため、この操作はトランザクションと同じように動作します。コードで該当するカテゴリが見つからなかった場合は、データベースの更新は行わず、False が返され、else が実行されます。

実行環境で確認してみましょう。

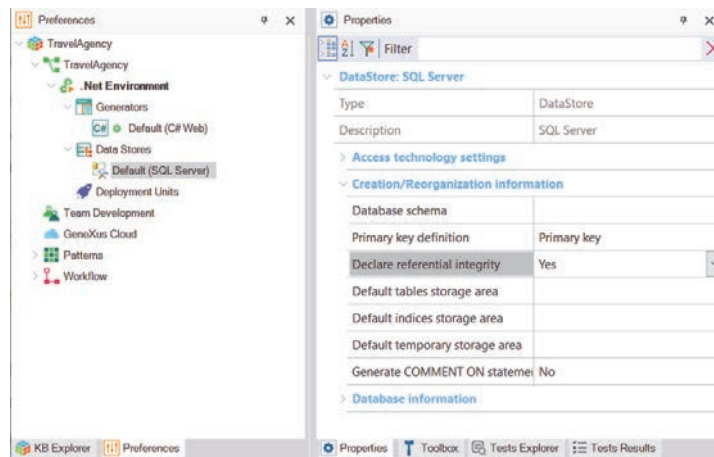




一方、プロシーチャーを使用して同じ更新操作をする場合は、ここに示すように、プロシーチャーに観光名所の ID とカテゴリの ID を送ります。そこで、観光名所とカテゴリが直接更新されるかを確認します。

存在しないカテゴリで実行すると、データベースの例外が発生し、ユーザーはこの分かりにくい画面に対処する必要があります。プログラムでは参照整合性コントロールは実行されませんが、データベースで実行されます。エラーが発生したのはそのためです。

エラーをキャプチャーして、プログラムがキャンセルされないように、より分かりやすいものにすることができます。それには Error\_handler ルールを使用します。パラメーターとして、オブジェクトに定義する必要があるサブルーチンの名前を入力します。また、データベースエラーが発生した場合は、まずそのエラーを確認するように動作をプログラミングします。この処理の詳細については、GeneXus の Wiki で確認できます。



参照整合性のステートメントをデータベースから削除して、データベースでコントロールが実行されないようにすることも可能です。これは推奨ではありませんが、必要な場合は、このプロパティがデータストアレベルにあります。



CategoryId を  
CategoryId = 102 のカテゴリ  
から選択する場合  
...



AttractionId	AttractionName	CountryId	CityId	CategoryId
1	ルーブル美術館	2	1	1
2	万里の長城	3	1	102
3	エッフェル塔	2	1	2
4	紫禁城	3	1	2

Name	Type	Description	Formula	Nullable
Attraction	Attraction	Attraction		No
AttractionId	Id	Attraction Id		No
AttractionName	Name	Attraction Name		No
AttractionAddress	Address, Generic	Attraction Address		No
CountryId	Id	Country Id		No
CountryName	Name	Country Name		No
CityId	Id	City Id		No
CityName	Name	City Name		No
CategoryId	Id	Category Id		Yes
CategoryName	Name	Category Name		No

AttractionId	AttractionName	CountryId	CityId	CategoryId
1	ルーブル美術館	2	1	1
2	万里の長城	3	1	null
3	エッフェル塔	2	1	2
4	紫禁城	3	1	2

このプロパティを変更しない場合、既定で、参照整合性はプログラムによってコントロールされる場合とされない場合がありますが、データベースでは**必ず**コントロールされます。

ただし、存在しない外部キーの値が許可されるケースがあることは確認しました。厳密に言えば、値ではありません。

外部キーが null であることが許容されるケースです。外部キーに null を許容することを、データベースにどのように伝えたのでしょうか。

トランザクション構造内に [Null 許容] 列があり、既定で No に設定されますが、これを Yes に変えることができます。

GeneXus™

Attraction

Test

trialapps3.genexus.com/ldf180a9d465110cf307ff42ea994a10e6/attraction.aspx?UPD,2

Name

The Great Wall

Address

Country Id

3

Country Name

China

City Id

1

City Name

Beijing

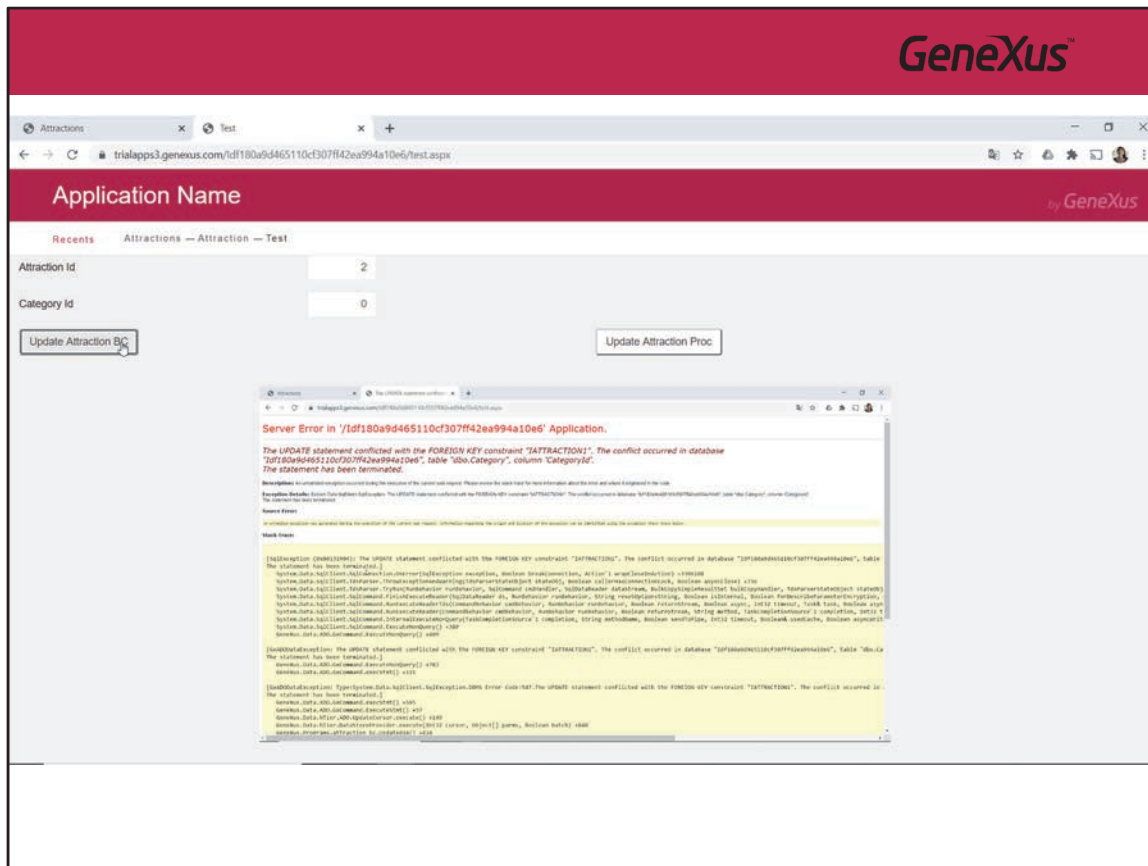
Category Id

Category Name

CONFIRM

CANCEL

これは、トランザクションが実行されたときにカテゴリ ID が空のままの場合、プログラムでもデータベースでも参照整合性コントロールが行われないことを意味します。



しかし、ビジネスコンポーネントを介して同じ処理を行うと、以前見たように、参照整合性エラーによるデータベースの例外がスローされます。何が起きたのでしょうか。

トランザクションは使用できたのに、そのビジネスコンポーネントは使用できなかったのはなぜでしょうか。

空の値    ≠    null

数値  
CategoryId    0

数値  
CategoryId    null

空の値と null 値はまったく別のものです。

空の値は値です。データタイプによっては、このタイプの値が受け入れられます。タイプが数値である場合、空の値はゼロになります。データベースでは、データタイプに応じて空の値が指定されています。しかし、厳密に言えば、null 値は**値ではありません**。実際には**値がない**ということです。CategoryId が null であることは、その値が指定されておらず、未知であることと同じです。空であるのとは異なります。

空の値      ≠      null



AttractionId = 2  
→ CategoryId = 0 の場合



AttractionId	AttractionName	CountryId	CityId	CategoryId
1	ルーブル美術館	2	1	1
2	万里の長城	3	1	2
3	エッフェル塔	2	1	2
4	紫禁城	3	1	2



AttractionId = 2  
→ CategoryId = null の場合

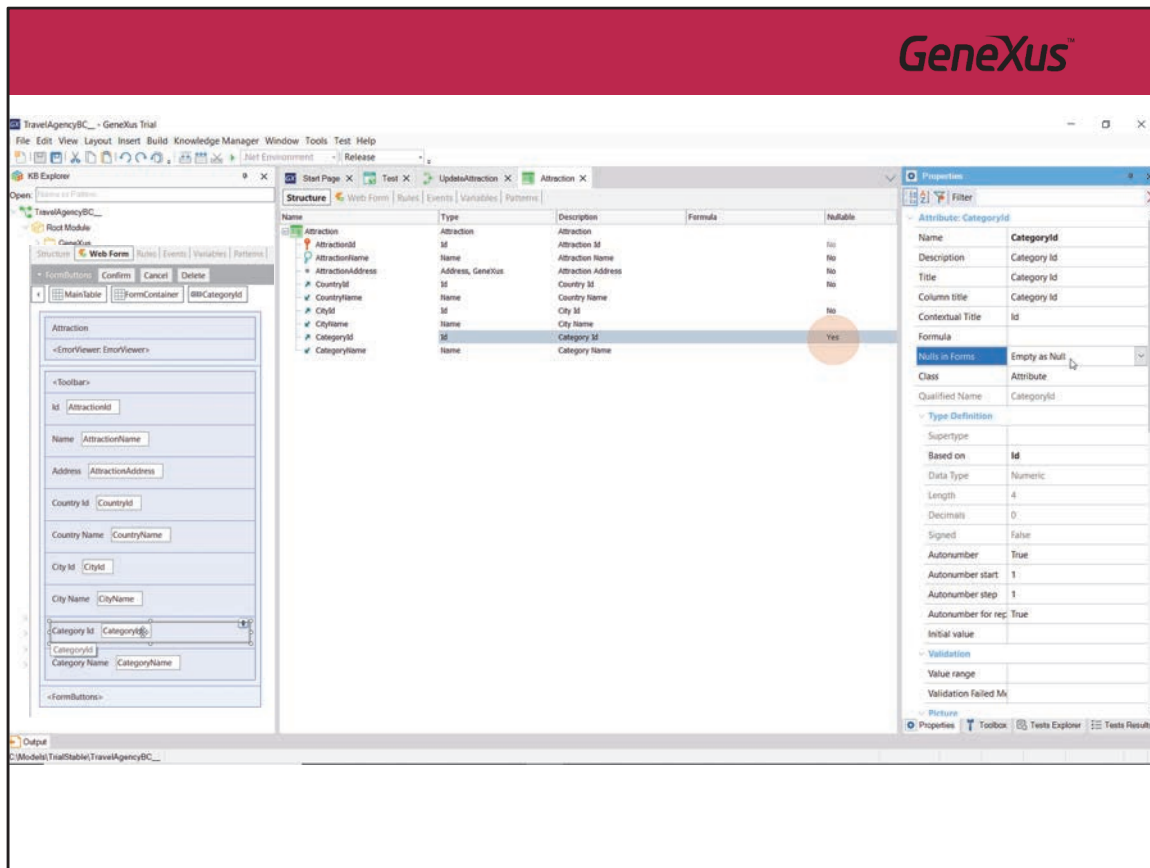


AttractionId	AttractionName	CountryId	CityId	CategoryId
1	ルーブル美術館	2	1	1
2	万里の長城	3	1	null
3	エッフェル塔	2	1	2
4	紫禁城	3	1	2



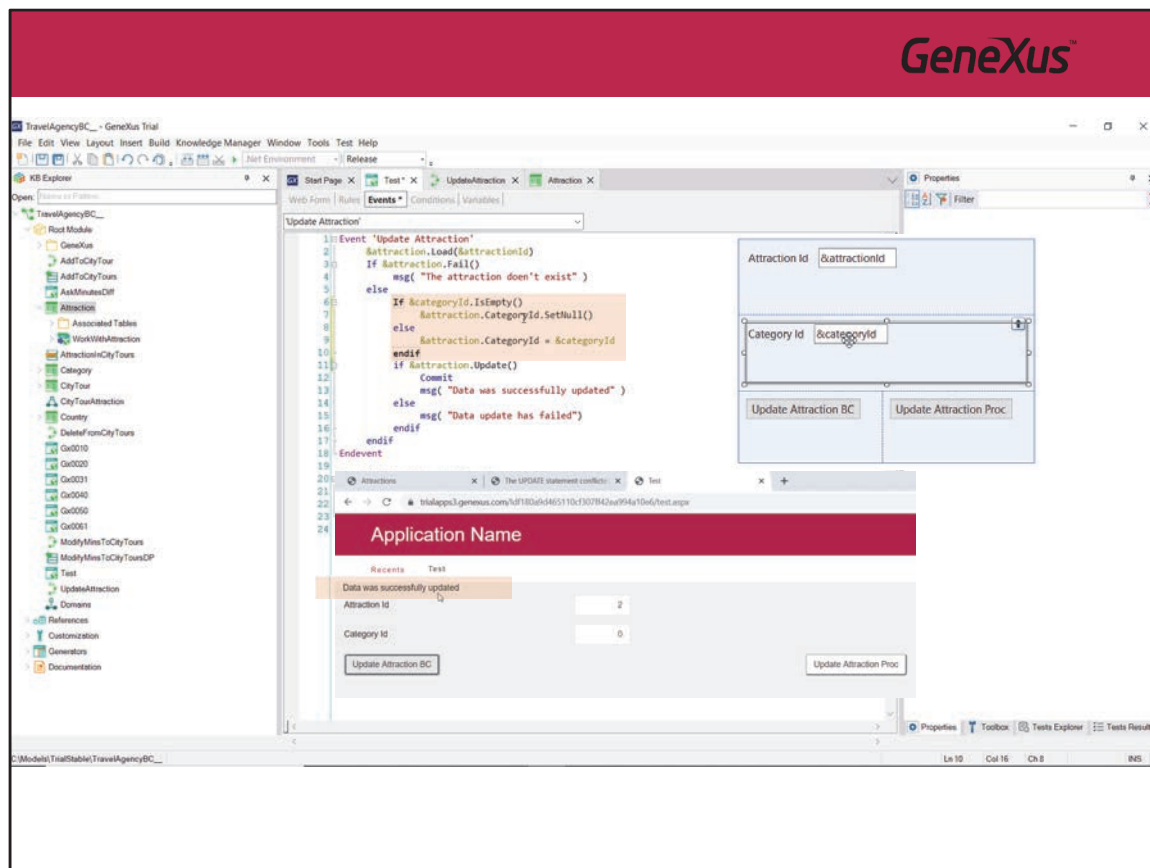
データベース内において、空の値と null 値はまったく別のものです。プログラムを実装し、データベース内で観光名所 2 のカテゴリを 0 に変更する命令を送信すると、参照整合性エラーによるデータベースの例外を示す画面が表示されるだけです。それに対して、値を null に変更すると、トランザクションを使用して同じ処理をした場合と同様に、何の問題も発生しません。

なぜでしょうか。トランザクションではデータベースに null への変更が指示されるのに、ビジネスコンポーネントでは空の値 (ゼロ) が送られるのはなぜでしょうか。



外部キー項目属性 CategoryId のプロパティには [Nulls in Forms] があり、これが [Empty as Null] に設定されています。つまり、このケースのように、項目属性が null 値を受け入れる場合、フォーム内で値が空のままになっていると、データベースに送られたときに空ではなく null と見なされます。





しかし、フォームを持たないビジネスコンポーネントを使用して更新を行う場合は異なります。

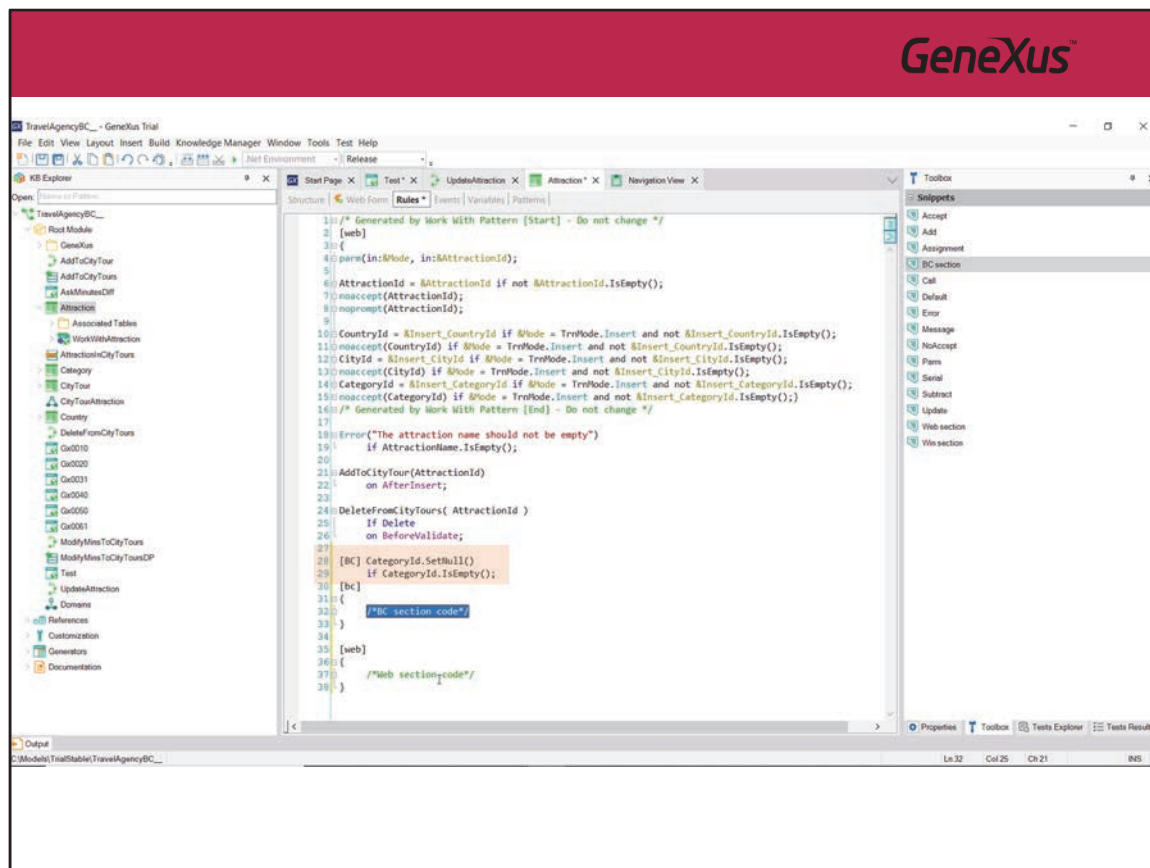
&CategoryId 変数が空の場合、このエレメントは null 値ではなく空のままになります。更新しようとしたときに参照整合性エラーになるのはそのためです。この変数が空の場合は null 値を指定する必要があります。

これを行うには2つの方法があります。1つはローカルで行う方法です。もう1つは項目属性プロパティを取得するのと同じマッピングを行う方法ですが、この場合はビジネスコンポーネントでも有効になります。

最初の方法では、CategoryId 変数が空の場合、ビジネスコンポーネントのエレメントを null に設定すると指定するだけです。SetNull メソッドは、このケースのように、null 値を受け入れる項目属性と、それに対応するビジネスコンポーネントの変数の項目属性に適用されます。

これを試すと、可能であることが分かります。

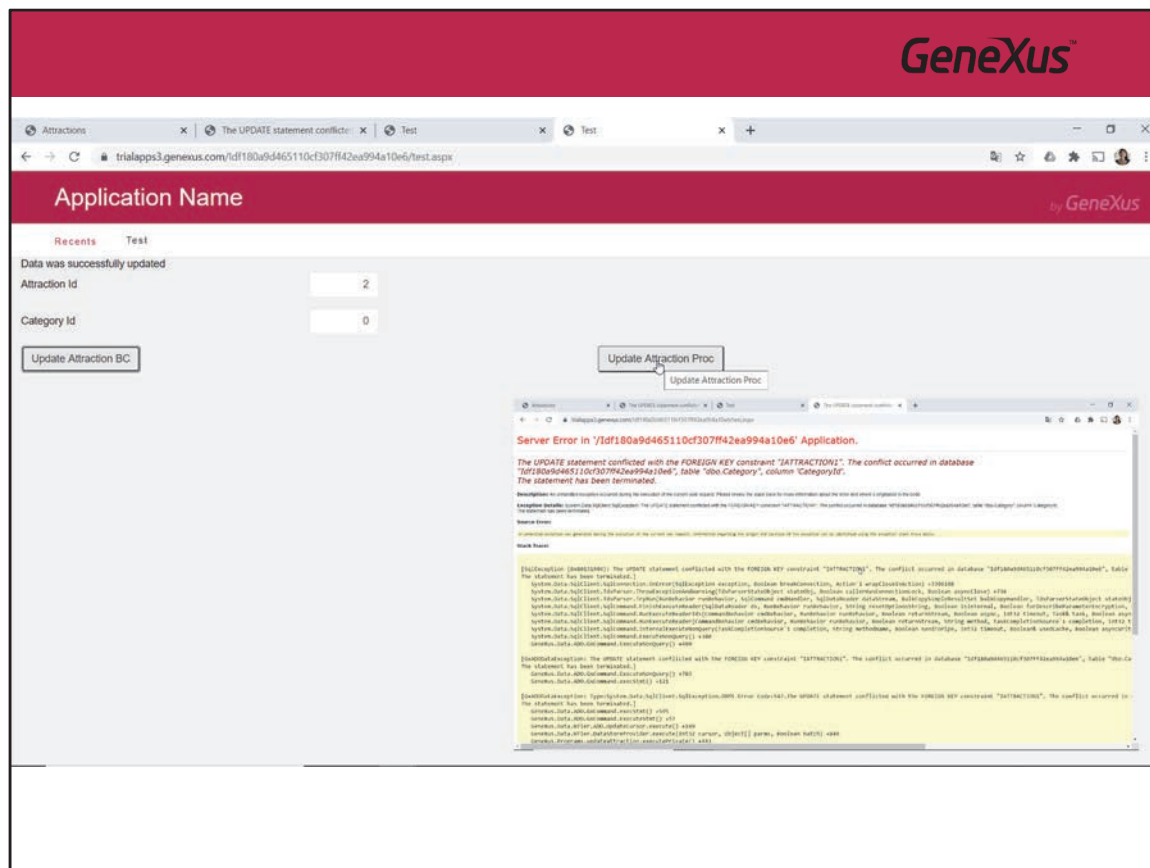
この代替方法の欠点は、他の場所のビジネスコンポーネント変数でこの外部キーを null のままにしたい場合、同じ処理をする必要があることです。もう1つのより一般的なオプションは、この SetNull をルールとしてトランザクションに組み込むことです。この実装を確認するために、1つ目の方法で追加したコードを削除します。

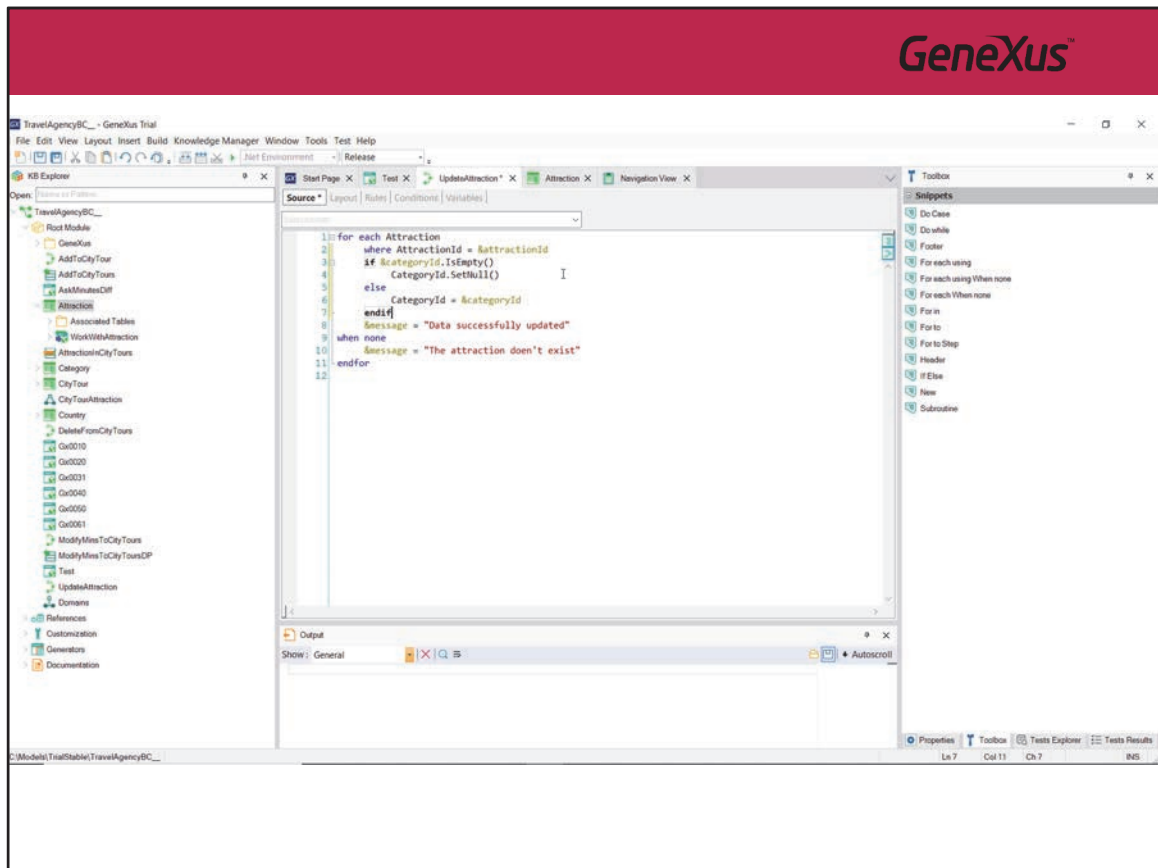


トランザクションに移動して、値が空の場合は項目属性を null に設定するように指定します。トランザクションの画面を使用するときには、[Nulls in Forms] プロパティのおかげで、既定でこの処理が行われています。そのため、厳密に言えば、このルールを対象とするビジネスコンポーネントに対してのみ実行するように指定できます。

[WEB] や [BC] という修飾子を設定することで、Web トランザクション内、または**ビジネスコンポーネント内**でのみ、ルールを**実行**するように指定できます。Web トランザクションにのみ適用するルールのセット、またはビジネスコンポーネントにのみ適用するルールのセットを記述するには、このようにします。

試してみましょう。これがもう 1 つの解決策であることが確認できます。





この解決策は、ローカルで行う方法だけです。試してみましょう。

実行すると、適切に処理されることが分かります。