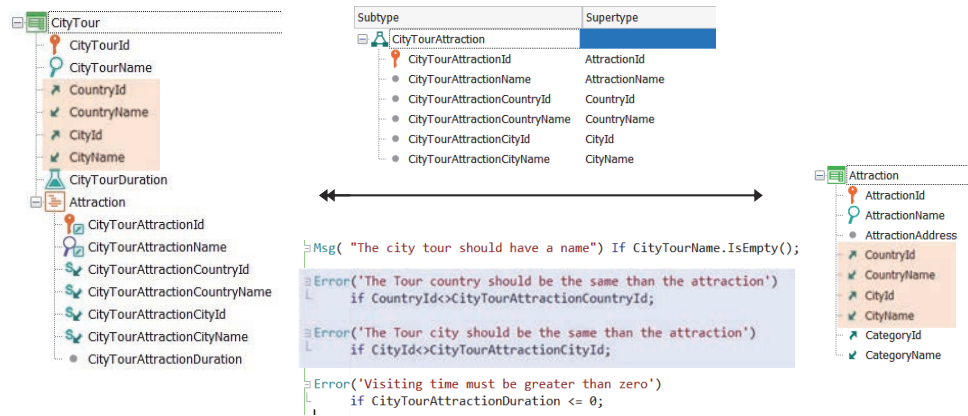


データベースの更新

2 レベルのビジネスコンポーネントの使用

GeneXus[™]

2 レベルのビジネスコンポーネント



旅行代理店の顧客に提供される、ある都市のさまざまな観光名所を訪問するツアーを表示する CityTour トランザクションが作成されているとします。

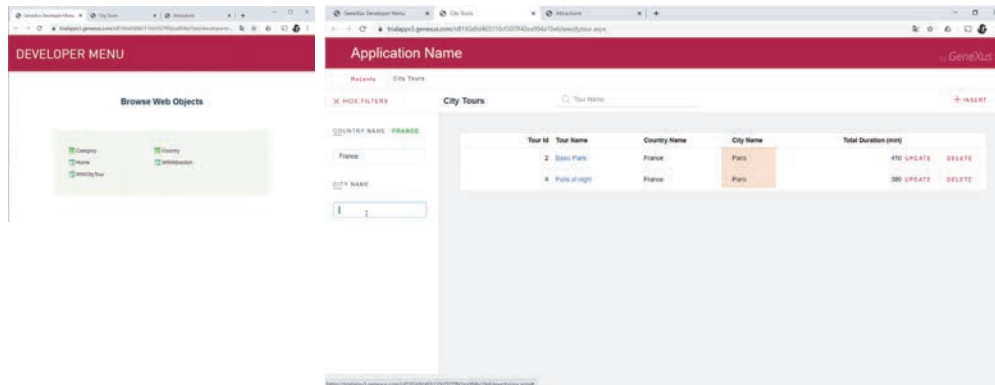
これは 2 レベルのトランザクションです。第 1 レベルでは、ツアーの名前に加えて国と都市を指定し、各観光名所の予想滞在時間を合計したツアー全体の所要時間も指定します。

サブレベルでは、そのツアーに含まれる観光名所を指定します。第 1 レベルで市内観光の国と都市を指定する必要があるため、サブタイプのグループを定義する必要があります。さらに、観光名所は国と都市を持つため、これらがツアーのものと一致することをチェックする必要があります。

CityTourAttracionId は AttractionId のサブタイプであるため、同じ項目属性のように扱われます。したがって、CityTour の第 2 レベルのテーブルの外部キーになります。

そのテーブルには、観光名所の予想滞在時間を指定するための従属項目属性 CityTourAttractionDuration も含まれます。

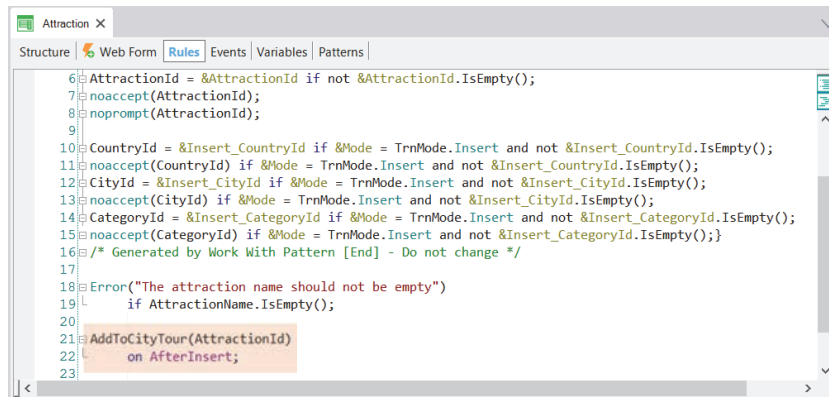
多数の行の挿入



CityTour に Work With パターンを適用し、より快適に作業できるように観光名所にも適用しました。そして、一部のデータを読み込んだ状態です。たとえば、パリには2つの市内観光しか作成されていません。1つはルーブル美術館とエッフェル塔で、もう1つはエッフェル塔のみです。

バックオフィスユーザーが観光名所について作業していて、システムに新しいもの(たとえば、ノートルダム寺院)を入力するとします。観光名所の都市(この場合はパリ)に対応するすべての市内観光に、滞在時間(固定で120分)とともに、これを自動的に追加する必要があります。この滞在時間は後で変更できます。これはどうやって実行したらよいでしょうか。

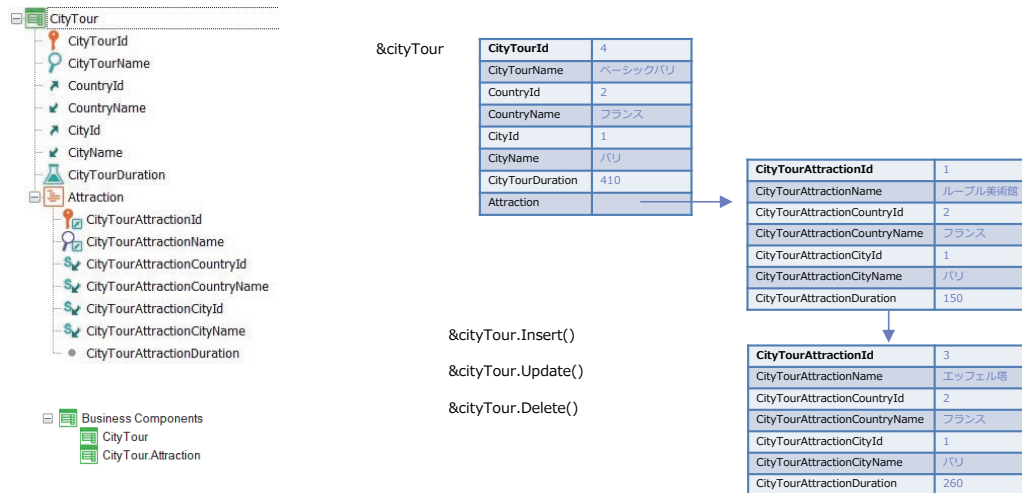
多数の行の挿入



```
Attraction X
Structure Web Form Rules Events Variables Patterns
6 AttractionId = &AttractionId if not &AttractionId.IsEmpty();
7 noaccept(AttractionId);
8 noprompt(AttractionId);
9
10 CountryId = &Insert_CountryId if &Mode = TrnMode.Insert and not &Insert_CountryId.IsEmpty();
11 noaccept(CountryId) if &Mode = TrnMode.Insert and not &Insert_CountryId.IsEmpty();
12 CityId = &Insert_CityId if &Mode = TrnMode.Insert and not &Insert_CityId.IsEmpty();
13 noaccept(CityId) if &Mode = TrnMode.Insert and not &Insert_CityId.IsEmpty();
14 CategoryId = &Insert_CategoryId if &Mode = TrnMode.Insert and not &Insert_CategoryId.IsEmpty();
15 noaccept(CategoryId) if &Mode = TrnMode.Insert and not &Insert_CategoryId.IsEmpty();
16 /* Generated by Work With Pattern [End] - Do not change */
17
18 Error("The attraction name should not be empty")
19 if AttractionName.IsEmpty();
20
21 AddToCityTour(AttractionId)
22 on AfterInsert;
23
```

簡単にいえば、このトランザクションで新しい観光名所が挿入されると、直ちに、同じ国と都市を持つすべての市内観光に、この観光名所を含む新しい行が滞在時間 120 分とともに自動的に追加する必要があります。

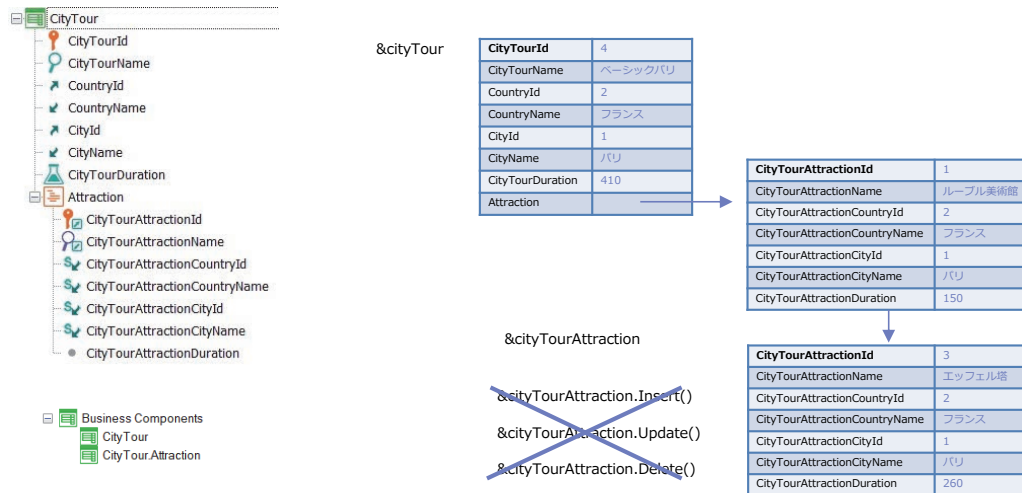
次に、on AfterInsert で (つまり、Attraction テーブルに新しい観光名所が挿入された直後に)、プロシーチャーを呼び出し、入力された観光名所の ID を渡して、同じ国と都市を持つ市内観光の行を挿入します。



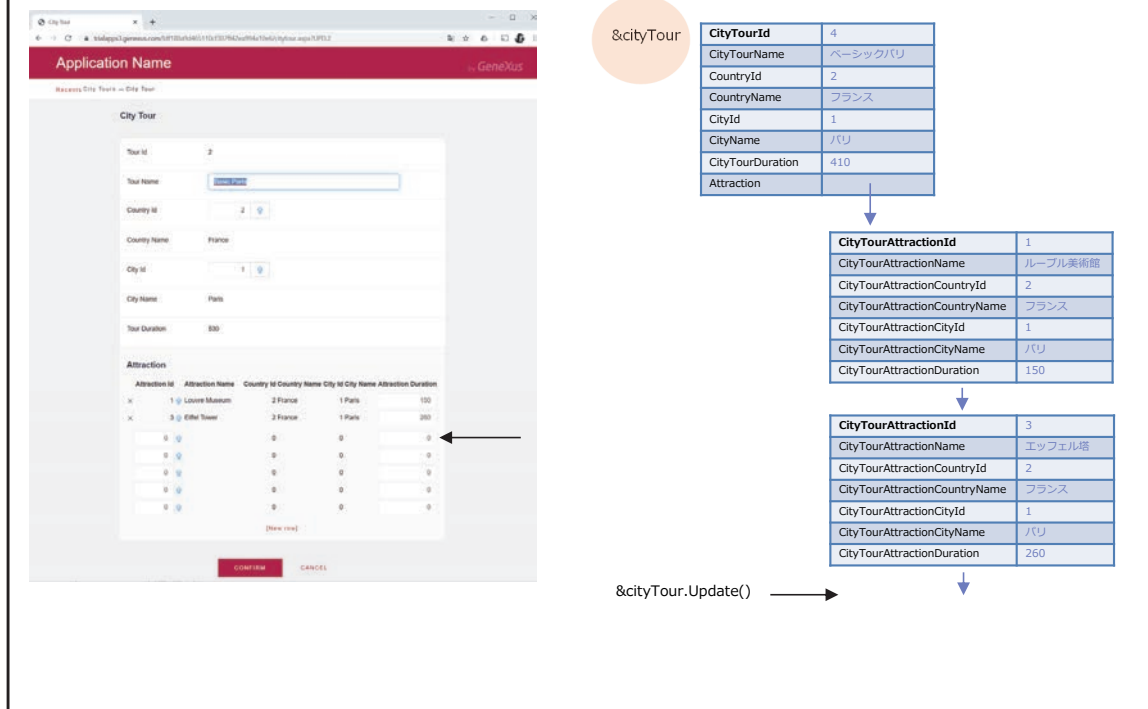
トランザクションの [Business Component] プロパティを有効にすると、1 つではなく、2 つのビジネスコンポーネントが自動的にナレッジベースに作成されます。

1 つ目は予想どおりのもので、表示されているものと同じような構造を持ちます。したがって、市内観光 4 に 2 つの行がある場合、CityTour ビジネス コンポーネント データ タイプの変数の構造は、ここに示したものと同様になります。つまり、最後のエレメント Attraction は行のコレクションです。各行がそれぞれ、構造化データアイテムに対応します。どの構造でしょうか。ビジネスコンポーネント CityTour.Attraction の構造です。これはトランザクションの各行に対応します。

つまり、トランザクション全体に対応するビジネスコンポーネントは、データベースに対して Insert、Update、Delete 操作を実行できますが、行に対して作成されたものの場合は、データ構造としてのみ使用できます。

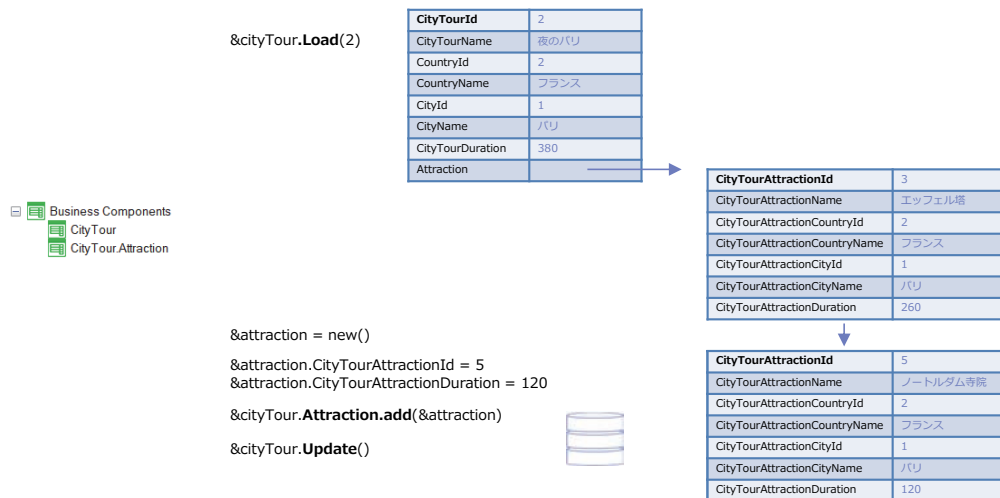


他の操作は許可されません。これに最初は戸惑うかもしれません。



CityTour トランザクション画面で新しい行を挿入したい場合、最初にヘッダーをインスタンス化してから行を挿入し、最後にグローバルな [実行] を選択する必要があります。ビジネスコンポーネントの場合も同様です。たとえば、行を追加するなど、行に対する作業を行うには、**ヘッダーとその行**に対して作業する必要があります。つまり、行を追加し、すべて (トランザクションのビジネスコンポーネント変数) に対して必要なことを実行します。

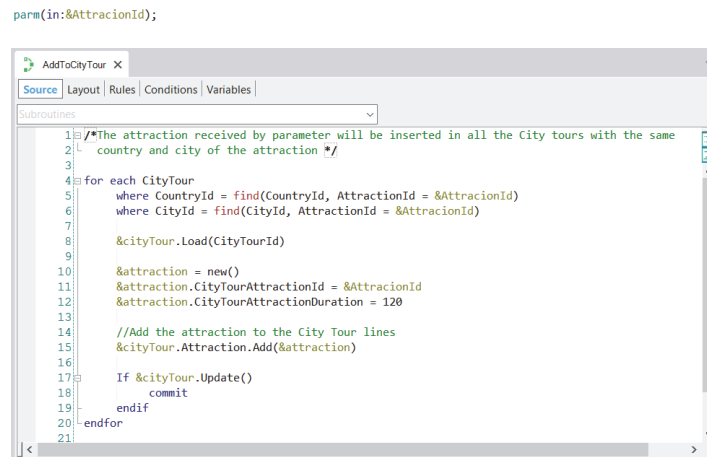
1 つの行の挿入



したがって、市内観光 2 の新しい観光名所を挿入するには、CityTour トランザクションのビジネス コンポーネント データ タイプの &cityTour 変数が必要です。**Load** 操作を適用すると、データベースからデータを読み込んだ変数を取得できます。この場合はヘッダーとその 1 つの行です。新しい行を追加する必要があるため、次に、行に対応するビジネス コンポーネント データ タイプの &attraction 変数を定義します。エレメントの値をロードしたら、後はこの構造を行のコレクションに追加するだけです。これは Attraction コレクションに適用される **Add メソッド** で実行します。

&cityTour 変数に期待する情報が入ったら、**Update 操作**を実行します。ID 2 の CityTour は既に存在するため、行を追加して変更するだけです。この操作により、実際にデータベースに行が挿入されます。

多数の行の挿入

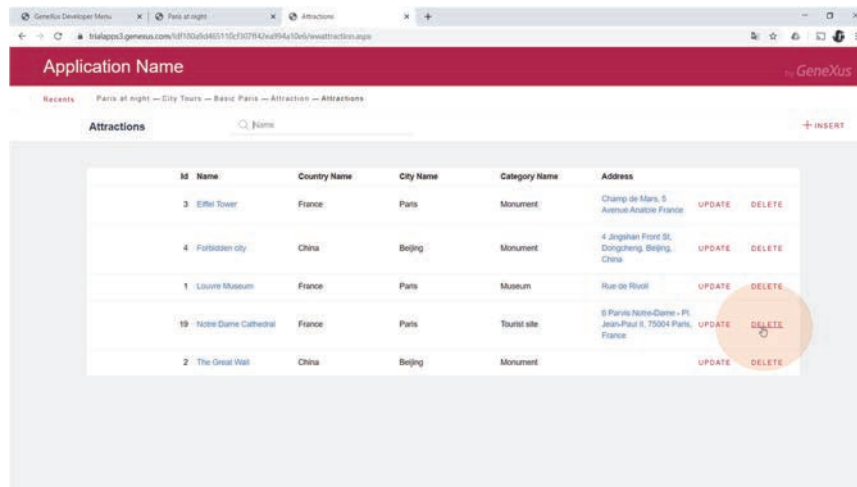


実装されているプロシージャーを見ると、すべてが理解できます。
観光名所の ID は &AttractionId 変数で受け取ります。

受け取った観光名所の国と都市を持つすべての市内観光について、For Each コマンドで処理を繰り返します。ここで、Attraction テーブルに観光名所を挿入した後、プロシージャーを呼び出さなければならない理由がはっきり分かります。そうしない場合、このフィルタを使用するための国と都市を見つけることができません。

&cityTour は CityTour ビジネス コンポーネント データ タイプの変数です。For Each コマンドで見つかった市内観光の ID からロードされます。トランザクションで行を追加するだけです。
そのためには、行に対応するビジネス コンポーネント データ タイプの変数を定義します。この変数のために新しいメモリー領域を作成し、格納できるエレメントを割り当てます。つまり、観光名所の ID と滞在時間です。
そして、この行を CityTour の行のコレクションに追加するだけです。
最後に Update 操作を実行すると、成功の場合は True が返されます。その場合は Commit を選択します。
これで、次の繰り返しに進みます。

多数の行の削除



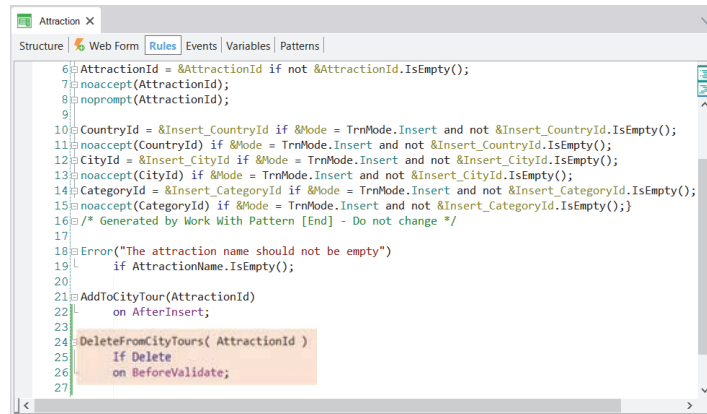
Id	Name	Country Name	City Name	Category Name	Address	UPDATE	DELETE
3	Eiffel Tower	France	Paris	Monument	Champ de Mars, 5 Avenue Alexandre Dumas, France	UPDATE	DELETE
4	Forbidden city	China	Beijing	Monument	4 Jingshan Front St, Dongcheng, Beijing, China	UPDATE	DELETE
1	Louvre Museum	France	Paris	Museum	Rue de Rivoli	UPDATE	DELETE
19	Notre Dame Cathedral	France	Paris	Tourist site	6 Parvis Notre-Dame - Pl. Jean-Paul II, 75004 Paris, France	UPDATE	DELETE
2	The Great Wall	China	Beijing	Monument		UPDATE	DELETE

今度は、ビジネスコンポーネントを使用して行を削除する方法を見てみましょう。

Work with Attractions には、以前に挿入したノートルダム寺院が含まれています。市内観光に移動すると、これを含むものはパリに2つあります。ここでは、観光名所を削除できるようにします。通常、これは参照整合性チェックにより許可されません。この観光名所を含む市内観光をなくすためには、削除を許可する必要があります。つまり、[削除] をクリックしたときに、まずこの観光名所がある市内観光のすべて行を削除し、次に観光名所を確実に削除する必要があります。

実行すると、この市内観光から削除され、前に取り上げたものからも削除されていることを確認できます。これをどう実装すればいいでしょうか。

多数の行の削除

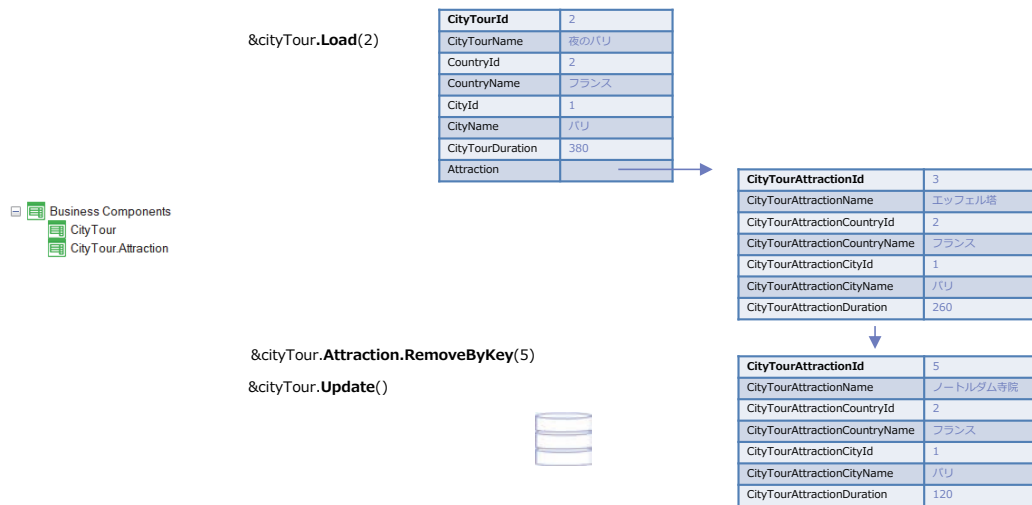


Attraction トランザクションから Delete モードで実行します。データの検証が行われる前に観光名所を削除する必要があります。つまり、参照整合性チェックが行われる前に、プロシーチャーを呼び出して、観光名所の ID を渡します。プロシーチャーは、その観光名所を含む市内観光のすべての行を削除します。

これはどのように実行すればよいでしょうか。

トランザクションを使用して行を削除するには、まず、市内観光をロードします (Update モードになります)。そして、行を削除し、最後に [実行] をクリックしてデータベースで実際に削除を実行します。この場合も同じようになります。

1 つの行の削除

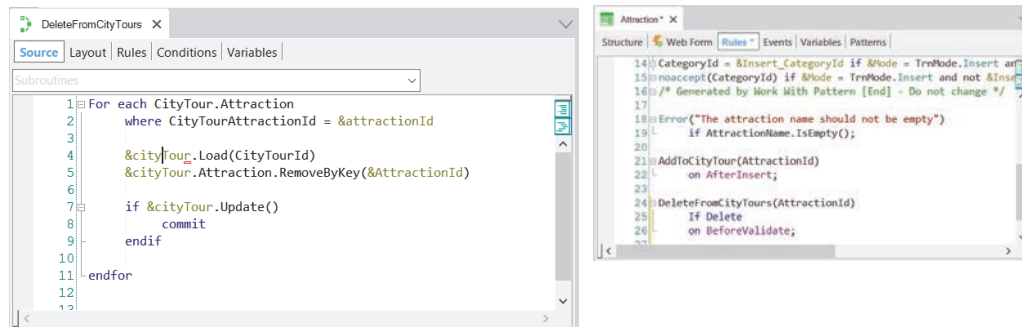


市内観光 2 からノートルダム寺院を削除するには、CityTour ビジネス コンポーネント データ タイプ の変数に **Load メソッド** を使用して構造をロードし、Attraction コレクションから行を削除します。どのようにすればよいでしょうか。

RemoveByKey メソッド を使用します。名前のとおり、このメソッドはコレクションを検索して、指定されたキー値に一致するアイテムを見つけます。この場合は、ID 5 のノートルダム寺院です。この削除はメモリー内でのみ実行されます。これをデータベースにも適用する必要があります。そのためには市内観光を更新する必要がありますため、Update メソッドを使用します。もちろん、**Save()** も使用してもかまいません。これらはすべてトランザクションの [実行] ボタンをクリックすることと同じことです。

多数の行の削除

```
parm( in: &AttractionId );
```



プロシージャが GeneXus にどのように実装されるか見てみましょう。
 &AttractionId 変数は削除する観光名所の ID を受け取ります。For Each コマンドを使用して市内観光の行を検索し、この観光名所 ID に一致するものを見つけます。見つかった場合は、トランザクションに対応するビジネスコンポーネントデータタイプの市内観光の変数で、その行に対応する CityTourId をロードします。そして、その市内観光の観光名所のコレクションに **RemoveByKey メソッド** を適用し、パラメーターで受け取った観光名所 ID の観光名所を削除します。次に Update を実行し、成功した場合は Commit を実行します。

このようにして、データにその観光名所が含まれるすべての市内観光の行を確実に削除できます。このプロシージャの実行が完了すると、コントロールがこちらに戻り、対応する観光名所の検証が実行されます。しかし、関連するデータがないため、削除しても参照整合性エラーは発生しません。

1 つの行の更新

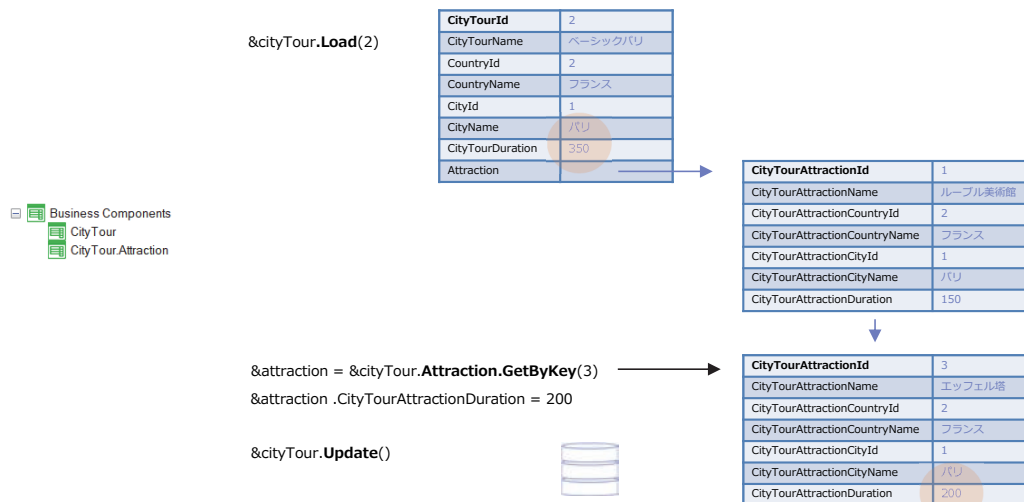
The image displays two screenshots of a web application interface. The left screenshot shows the 'City Tour' form with fields for Tour ID, Tour Name, City Name, City ID, City Name, and Tour Duration. The right screenshot shows the 'Attraction' form with fields for Attraction ID, Attraction Name, County, City, City Name, and Attraction Duration. Both forms have a 'Save' button at the bottom.

最後に、ビジネスコンポーネントを使用してトランザクションの行を更新する方法を学習します。

トランザクションを使用するこのケースでは TourId 2 を編集します。エッフェル塔の滞在時間を 260 分から 200 分に変更するとします。変更したら、後は実行するだけで更新が行われるようにします。

このような処理をビジネスコンポーネントを使用して行う必要があります。つまり、ロードして行を変更し、更新します。

1 つの行の更新



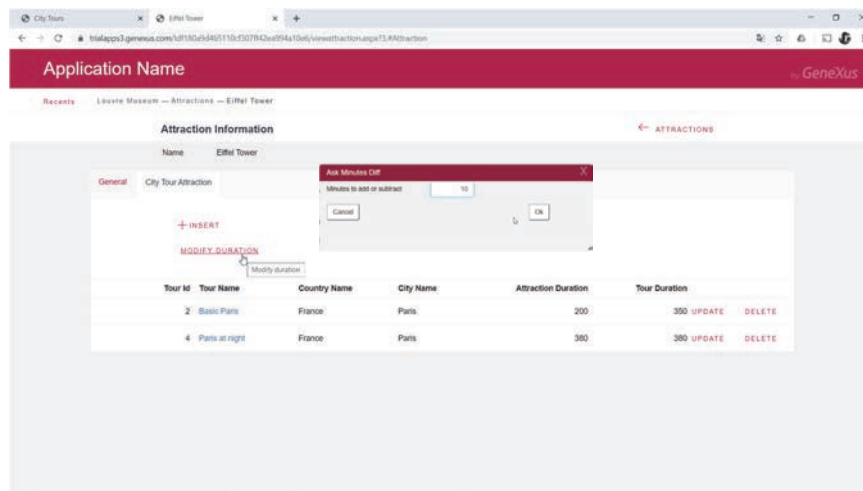
それでは、ID が 2 の市内観光の `&cityTour` 変数をロードします。

観光名所 3 に対応するコレクション内のアイテムにアクセスする必要があります。そのために **GetByKey メソッド** を使用します。Attraction コレクション (つまり、トランザクションの行) に適用すると、メモリー位置の直接参照が返されます。したがって、行に対応するビジネス コンポーネント データ タイプの `&attraction` 変数に割り当てられます。このように、`&attraction` 変数は行のコピーではなく、**その行そのもの** になります。

これで、後は変数の `CityTourAttractionDuration` エレメントを変更するだけです。これは `&cityTour` 変数に直接影響します。

これはデータベースにも適用される必要があるため、ビジネスコンポーネントの **Update** 操作を実行します。その結果、データベースで行に対応するレコードが更新されるだけでなく、`&cityTour` 変数も更新されます。また、行の滞在時間を合計するためのトランザクションのヘッダーの式に対応する `CityTourDuration` エレメントも **変数でリフレッシュ** されます。

多数の行の更新



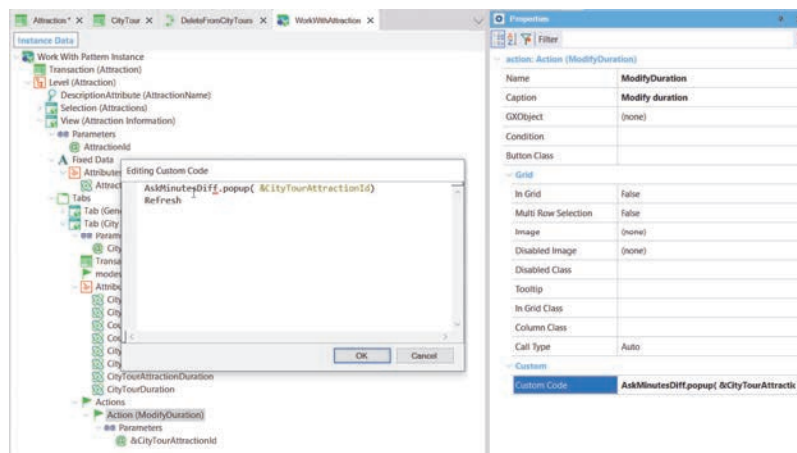
次に、ビジネスコンポーネントによる行の変更をアプリケーションに実装する方法を見てみましょう。

Work With Attractions で、たとえば、エッフェル塔などの特定の観光名所を編集するために、その観光名所が含まれるすべての市内観光が表示されるタブを選択します。そうすると、他のデータとともに、それぞれの市内観光の観光名所の滞在時間が分単位で表示されます。

ここから時間を分単位で加算したり減算したりして、滞在時間を変更できるようにします。たとえば、それぞれ 10 分追加したいとします。ここに示す画面が表示されるので、10 分の追加を指定すると、加算が実行されます。

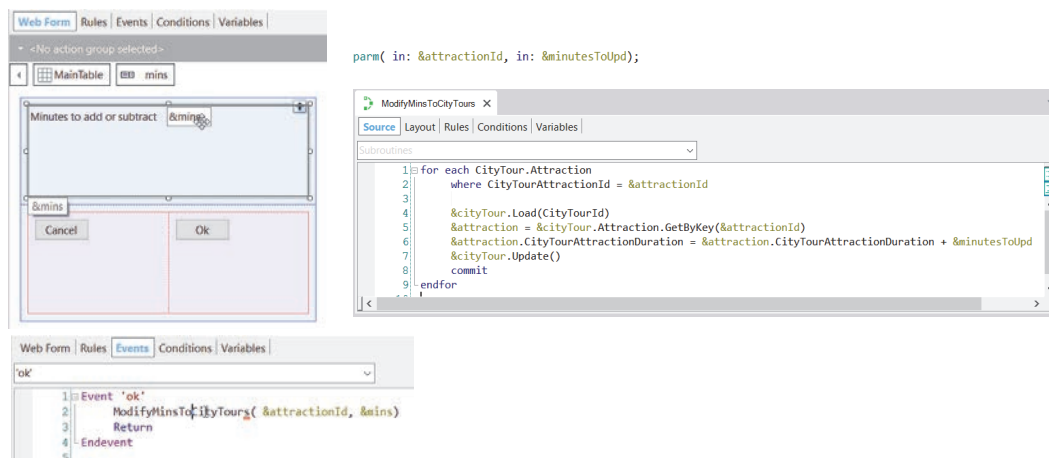
このバックグラウンドでは何が実行されたのでしょうか。これら 2 つの市内観光それぞれにアクセスし、最初のものから始めて、この観光名所に対応する行にアクセスし、予想滞在時間に 10 分加算しました。もう 1 つに対しても同じことを実行しました。つまり、2 つの市内観光それぞれの 1 つの行を変更する必要がありました。

多数の行の更新



GeneXus ではこれをどう実装するか示すために、Work with Attraction で、観光名所の市内観光に対応するタブに ModifyDuration アクションを追加しました。コードでは、AskMinutesDiff という名前の Web パネルを呼び出して観光名所の ID を渡します。この Web パネルの実行が終了したら、再表示を実行して、更新されたデータをグリッドに表示します。

多数の行の更新



AskMinutesDiff Web パネルを見てみましょう。**変数**で、時間を変更する分単位の値の入力をユーザーに求めています。これは、対応するデータを変更するプロシージャーを呼び出します。プロシージャーは観光名所の ID と時間 (分単位) を受け取ります。これに該当する市内観光の観光名所に対して、この時間を加算または減算します。

このプロシージャーのスケジュールはどう設定すればいいでしょうか。

ここでは、2 つの変数で情報を受け取っていることが分かります。そして For Each コマンドで市内観光の観光名所を参照し、パラメーターが受け取る観光名所でフィルタします。見つかった各レコードに対して、&cityTour ビジネスコンポーネント変数で該当する市内観光をロードします。CityTour.Attraction タイプの &attraction 変数 (つまり、行に対応するビジネスコンポーネント) で、&AttractionId キーに対応する Attraction コレクションのアイテムを取得します。次に、滞在時間を変更します。アイテムの CityTourAttractionDuration エレメントで、パラメーターが受け取った時間 (分単位) を値に加算します。次に Update を実行してコミットします。

まとめ

1 つの行の挿入

```
&BC.Load(PKAttribute)
&lineBC = new()
&lineBC.PKLineAtt = ...
&lineBC.LineAtt2 = ...
...
&lineBC.LineAttn = ...
&BC.Lines.Add(&lineBC)
&BC.Update()
```

1 つの行の削除

```
&BC.Load(PKAttribute)
&BC.Lines.RemoveByKey(PKLineAtt)
&BC.Update()
```

1 つの行の更新

```
&BC.Load(PKAttribute)
&lineBC = &BC.Lines.GetByKey(PKLineAtt)
&lineBC.LineAtt2 = ...
&lineBC.LineAttn = ...
&BC.Update()
```

ここまでに見てきた、2 レベルのビジネスコンポーネントで 1 つの行を挿入、削除、変更する方法をまとめます。

ヘッダーが既に存在することが前提のため、どの場合も使用する操作は Update です。

最初の手順で、常に作業対象のビジネスコンポーネント変数をロードします。行を挿入する場合、行のビジネスコンポーネントタイプの変数が作成され、新しいメモリー領域が予約され、値を指定したいテーブルの項目属性に対応する行のすべてのエレメントに値が割り当てられ、割り当て対象でないものは Null または空になります。そして、行のコレクションに**行が追加**されます。この追加は参照にも使用されます。したがって、各行に対して作業する前に、慎重に検討してから、new() を使用して新しいメモリー領域を作成する必要があります。

削除には **RemoveByKey メソッド** を使用します。これはパラメーターを介して削除する行の ID を受け取ります。

最後に、項目属性や行の項目属性を変更するには、行のコレクションの **GetByKey メソッド** を使用します。これも、取得したい行の ID をパラメーターを介して受け取ります。このメソッドは行に対応するビジネスコンポーネントの参照を返すため、そのタイプの変数にメソッドを割り当てる必要があります。変数を変更するときには、コレクションのそのアイテムが既に変更されています。

新しいヘッダーと行の挿入

```
&cityTour = new()
```

```
&cityTour.CityTourId = 2  
&cityTour.CityTourName = '夜のパリ'  
&cityTour.CountryId = 2  
&cityTour.CityId = 1
```

CityTourId	2
CityTourName	夜のパリ
CountryId	2
CountryName	フランス
CityId	1
CityName	パリ
CityTourDuration	380
Attraction	

```
&attraction = new()  
&attraction.CityTourAttractionId = find(AttractionId, AttractionName = "エッフェル塔")  
&attraction.CityTourAttractionDuration = 260  
&cityTour.Attraction.add(&attraction)
```

```
&attraction = new()  
&attraction.CityTourAttractionId = find(AttractionId, AttractionName = "ノートルダム寺院")  
&attraction.CityTourAttractionDuration = 120  
&cityTour.Attraction.add(&attraction)
```

```
&cityTour.Insert()
```



Business Components
City Tour
City Tour Attraction

CityTourAttractionId	3
CityTourAttractionName	エッフェル塔
CityTourAttractionCountryId	2
CityTourAttractionCountryName	フランス
CityTourAttractionCityId	1
CityTourAttractionCityName	パリ
CityTourAttractionDuration	260



CityTourAttractionId	5
CityTourAttractionName	ノートルダム寺院
CityTourAttractionCountryId	2
CityTourAttractionCountryName	フランス
CityTourAttractionCityId	1
CityTourAttractionCityName	パリ
CityTourAttractionDuration	120

たとえば、市内観光 2 (存在しない場合) など、新しいヘッダーとその行を挿入するには、まず、ビジネスコンポーネント変数を定義して、新しいメモリー領域を割り当てます。

そして、推論されたものではないヘッダー項目属性または式に対応するすべてのエレメントに値を割り当てます。

次に、行のビジネスコンポーネントタイプの &attraction 変数の新しいメモリー領域を作成します。推論されたものではない項目属性の値を割り当て、グローバルビジネスコンポーネントの**行のコレクション**に変数を**追加**します。

同じ手順で第 2 の行を入力します。ここでは新しいメモリー領域の予約は**必須です**。予約しない場合、割り当ての際に前の行が上書きされます。ビジネスコンポーネントの観光名所のコレクションに新しい変数を追加し、最後に **Insert 操作** を実行して、データベースにヘッダーと 2 つの行を挿入します。

エラーがない場合は、変数にすべてのデータがロードされます。つまり、入力したもの、推論されたデータおよび式です。

ビジネスコンポーネントによる挿入、更新、削除の方法はこれだけでしょうか。

この章では、構造内の行のコレクションを直接操作して、ビジネスコンポーネントの行を挿入、更新、削除する方法を見てきました。ほかにも方法があるでしょうか。残念ながらありません。なぜなら、データプロバイダーは、構造化データを返すために使用されるからです。