

# データベースの更新

単一レベルのビジネスコンポーネントの使用 (復習)

*GeneXus*<sup>™</sup>

Save()

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

InsertOrUpdate()

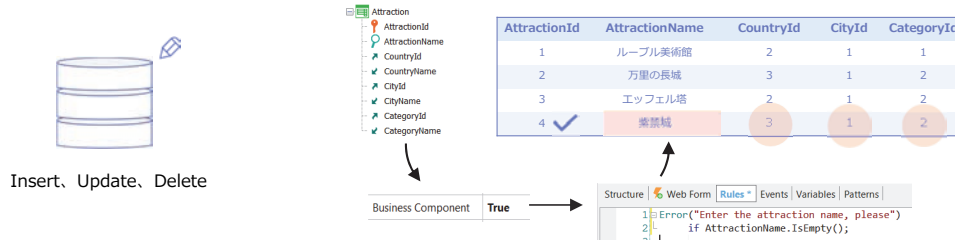


## 2. プロシージャー: New, For each, Delete

コードを使用してデータベース情報を更新する場合、次の2つの方法があります。

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 Insert、Update または Delete メソッドを使用する方法です (または Save を使用するか、Insert や Update の代わりに InsertOrUpdate を使用することもできます)。もう 1 つは、プロシージャー内のみで実行する方法です。この場合、New コマンドを使用するか、For each コマンドを使用して変更する項目属性を直接割り当てるか、For each 内で Delete コマンドを使用して自分が位置しているレコードを削除することができます。

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



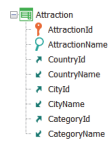
## 2. プロシージャー: New, For each, Delete

これら2つの方法の大きな違いは、最初の方法はトランザクションのロジックと強く結びついていることです。これは、重複チェックおよび参照整合性チェックを含むルールがトリガーされるためです。

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



Insert, Update, 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
5		1	1	100

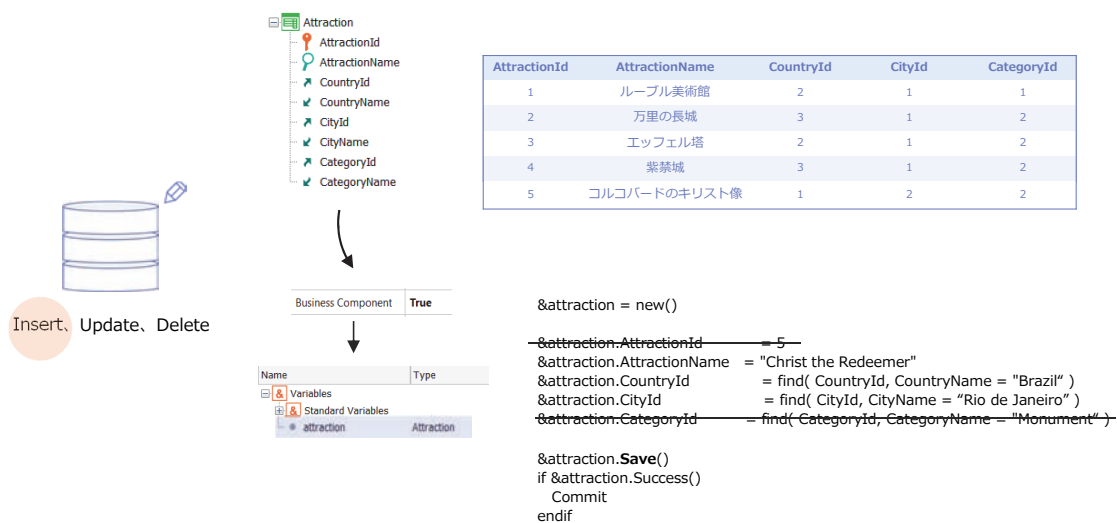


## 2. プロシージャー: New, For each, Delete

2 番目の方法は、更新がトランザクションに依存しないため、ルールはトリガーされません。実行される制御は重複チェックのみであるため、存在しないカテゴリを割り当てることもできます (プログラムはこれをチェックしません)。ただし、データベースではこれがチェックされるため、実行が中断されてユーザーのブラウザーにエラー画面が表示されます。

それでは、ビジネスコンポーネントを使用する最初の方法について見ていきましょう。

## BC: Insert

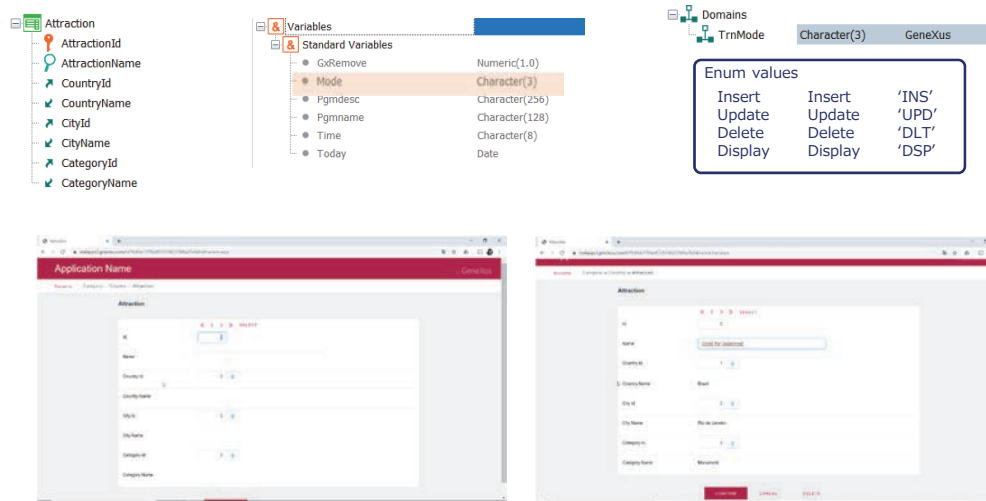


ビジネスコンポーネントを使用して新しい観光名所を挿入するには、次の手順を行うだけです。

- このタイプの変数を作成します。
- その変数に新しいメモリー領域を割り当てます (これは必須ではありませんが、前に行われた処理に関係なく、この時点で変数が完全に新規のものになるようにしておくことをお勧めします)。
- テーブルの項目属性に対応するビジネスコンポーネントのすべてのエレメントに値を割り当てます。
  - 識別子が自動採番の場合は、挿入したときにデータベースで実行されるため、値を割り当てる必要はありません。
  - テーブルの項目属性に値を割り当てない場合は、空または Null のままになります。通常はこれが問題になることはありませんが、項目属性が外部キーで、Null が許可されない場合は除きます。この場合、挿入しようとすると、参照整合性のエラーが発生します。
- 最後に、ビジネスコンポーネント構造へ値の割り当てを完了したら、あとは Insert メソッドを呼び出すだけです。
- 挿入が成功したら、必要に応じてコミットを実行します。

こうすることで、Save メソッドを使用するのと同じ処理が行われます。この場合、ビジネスコンポーネント変数が Insert モードであるため、挿入が試行されます (更新は実行されません)。

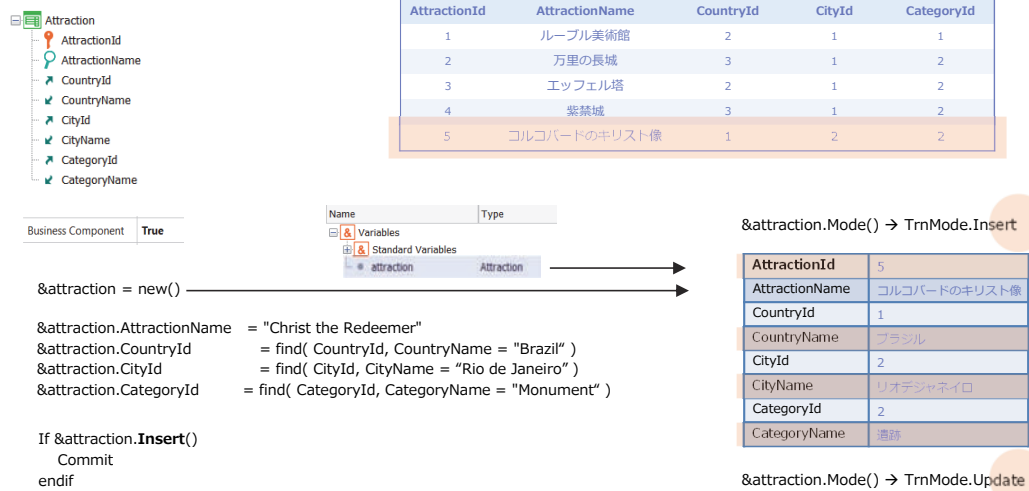
## Mode



各トランザクションの標準変数の1つに Mode があります。この変数には常に、トランザクションを実行しているモードが含まれます。この値を処理するために、GeneXus モジュールには TrnMode 列挙型ドメインが含まれており、Insert、Update、Delete、Display の4つの値を取ることができます。Display は情報が表示されるが何も実行されないことを示します。

トランザクションを開くときに、パラメーターとしてモードおよび識別子を受け取るように指定しない場合は、Insert モードで開きます。そのため、フィールドが空になります。識別子のフィールドから移動すると、指定した値(この場合は0)を使用してレコードが検索されますが、レコードは見つからないため、トランザクションは Insert モードのままになります。If Insert で条件付けられているルールが存在する場合は、それがトリガーされます。保存後、データが挿入されたことが通知されます。また、再度フォームが空になっています。これはトランザクションが再度 Insert モードのままになっていることを意味します。ただし、ビジネスコンポーネントを使用して挿入する場合は、Update モードのままになります。識別子としてデータベースの既存の値を選択するとします。たとえば、挿入されたばかりの5を選択して、フィールドを離れると、トランザクションにより値が画面のフィールドにロードされ、自動的に Update モードのままになります。カテゴリを削除するなど何かを変更して、空のままにすることができます(その外部キーが Null を受け入れる場合)。確認後に、レコードの更新が成功したことが通知されますが、Update モードで同じレコードに位置したままになります。この場合、ビジネスコンポーネントは引き続き同じように動作します。

## Mode()



ビジネスコンポーネントを使用する場合の処理について確認します。

`Mode()` メソッドでモードを確認できます。これは読み取り専用です。

`&attraction` 変数を定義して、ほかの処理を実行する前にモードをチェックすると、Insert モードであることがわかります。

`New` を使用して再度開始するたびにこのモードになります。

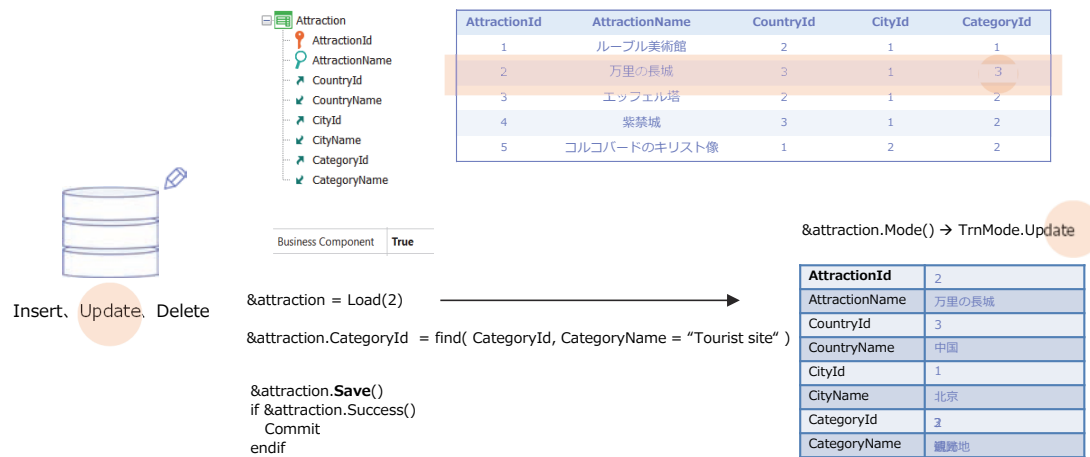
関連付けられたテーブルの項目属性に指定したい値を入力する必要があります (空のままにしたいもののみに入力します)。このとき、ビジネスコンポーネント変数はずっと Insert モードのままになります。

Insert メソッドを実行するとどうなるでしょうか。

データベースへの挿入が試行され、対応するルールが実行されます。そのプロセスが成功すると、ビジネスコンポーネント変数のすべてのエレメントが対応する値とともに読み込まれます。AttractionId は自動採番されるため、取得される値はデータベースで割り当てられたものです。トランザクションで推論される項目属性もここで読み込まれ、クエリが実行される場合もあります。

さらに、ビジネスコンポーネント変数のモードが変更されて Update になります。

## BC: Update



ID 2 の観光名所のカテゴリを「観光地」(Tourist site) に変更したい場合、次の手順を行います。

- ビジネスコンポーネント変数 &attraction に、Attraction テーブルの主キー 2 を持つレコードの値をロードします (Load メソッドを使用します。この場合、レコードが存在するときは変数が自動的に Update モードのままになります)。
- 変更対象のビジネスコンポーネントのすべてのエレメントに値を割り当てます (この場合、ほかの値はそのままにしたいため CategoryId のみです)。
- 最後に、Update メソッドを呼び出して、データベースで更新を実行します。
- 操作が正しく実行されたらコミットします。

Update メソッドが実行されると、トランザクションで推論される CategoryName エレメントは正しい値を持ち変数に残ります。変数は Update モードのままになります。

ここでも、Save メソッドを使用するのと同じことになります。この場合、Load の後でビジネスコンポーネント変数が Update モードのままになっているため、Save は Insert ではなく Update を実行しようとしています。



## BC: Update



Insert, Update, Delete

Attraction

- AttractionId
- AttractionName
- CountryId
- CountryName
- CityId
- CityName
- CategoryId
- CategoryName

AttractionId	AttractionName	CountryId	CityId	CategoryId
1	ルーブル美術館	2	1	1
2	万里の長城	3	1	3
3	エッフェル塔	2	1	2
4	紫禁城	3	1	3
5	コルコバードのキリスト像	1	2	2

```

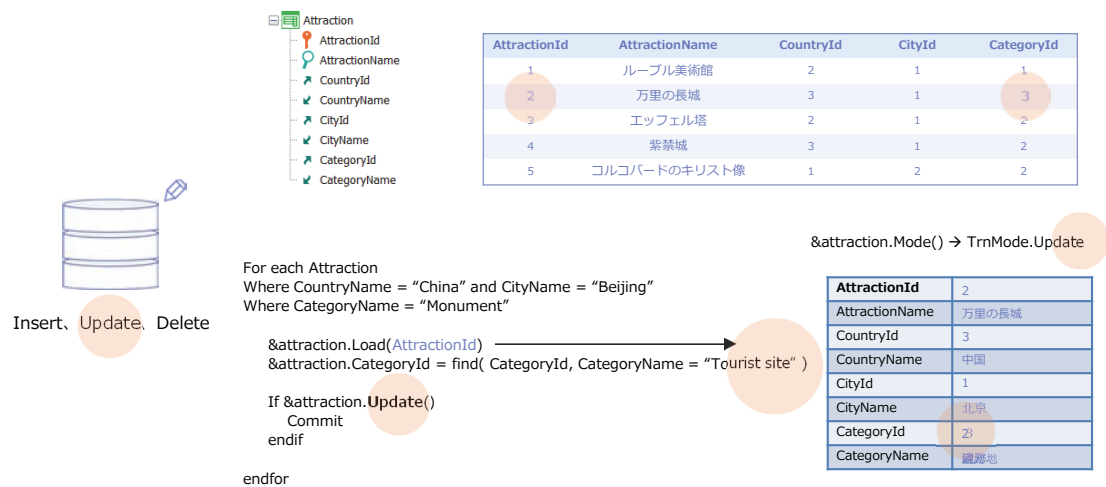
For each Attraction
Where CountryName = "China" and CityName = "Beijing"
Where CategoryName = "Monument"

  &attraction.Load(&AttractionId)
  &attraction.CategoryId = find( CategoryId, CategoryName = "Tourist site" )

  If &attraction.Update()
    Commit
  endif
endfor
  
```

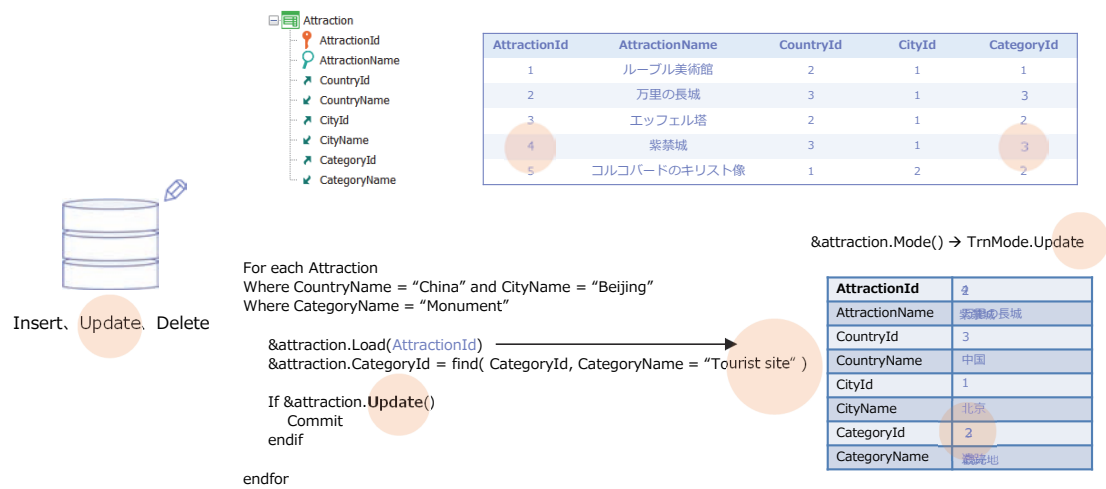
北京の遺跡であるすべての観光名所のカテゴリを変更して「観光地」(Tourist Site) カテゴリを割り当てたい場合、前のコードを For Each コマンド内に配置し、必要なレコードのみを選択して、これらのレコードの各 AttractionId に対して Load を実行します。

## BC: Update



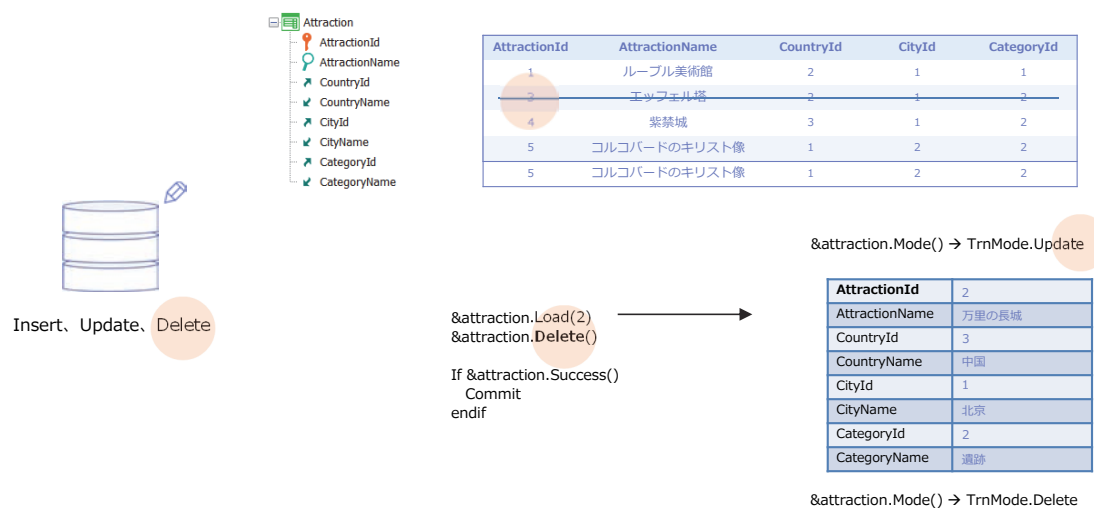
このように、最初に ID 2 の観光名所の Load を実行します。変数は Update モードのままになり、CategoryId が変更されます。Update メソッドを実行すると、テーブルのレコードが更新され、変数の CategoryName は対応したものになります。

## BC: Update




条件を満たす次のレコード (ID 4) の場合も同様です。

## BC: Delete



では、ID 2 の観光名所を削除したい場合はどうでしょうか。まず、これをビジネスコンポーネント変数にロードする必要があります。ロード後もこれは Update モードのままです。次に Delete メソッドを使用してテーブルから削除します。削除後も変数にデータがロードされたままですが、モードは Delete になります。そして、コミットを実行するために処理が成功したかチェックされます。

## BC: Delete



Insert, Update, Delete

Attraction

- AttractionId
- AttractionName
- CountryId
- CountryName
- CityId
- CityName
- CategoryId
- CategoryName

AttractionId	AttractionName	CountryId	CityId	CategoryId
1	ルーブル美術館	2	1	1
3	エッフェル塔	2	1	2
5	コルコバードのキリスト像	1	2	2
4	紫禁城	3	1	2
5	コルコバードのキリスト像	1	2	2

```

For each Attraction
Where CountryName = "China" and CityName = "Beijing"
Where CategoryName = "Monument"

    &attraction.Load(AttractionId)
    &attraction.Delete()

    If &attraction.Success()
        Commit
    endif
endfor
  
```

それでは、北京の遺跡タイプのすべての観光名所を削除したい場合はどうでしょう。この場合も、For Each コマンドで観光名所を1つずつ参照し、Load アクションでロードして、Delete メソッドを使用して削除します。

これは単純ですが、2 レベルの  
ビジネスコンポーネントの場合はどうでしょうか。

単一レベルのビジネスコンポーネントの場合はとても簡単でしたが、2 レベルの場合はどうでしょうか。行の挿入、更新、削除をどう実行すればいいでしょうか。

これについては次の資料で説明します。