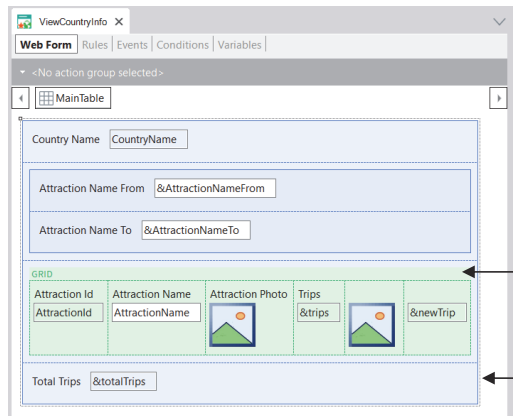


バックオフィスに重点を置いた Web 画面

Web Panel オブジェクト: 複数のグリッド

GeneXus[™]

複数のグリッド



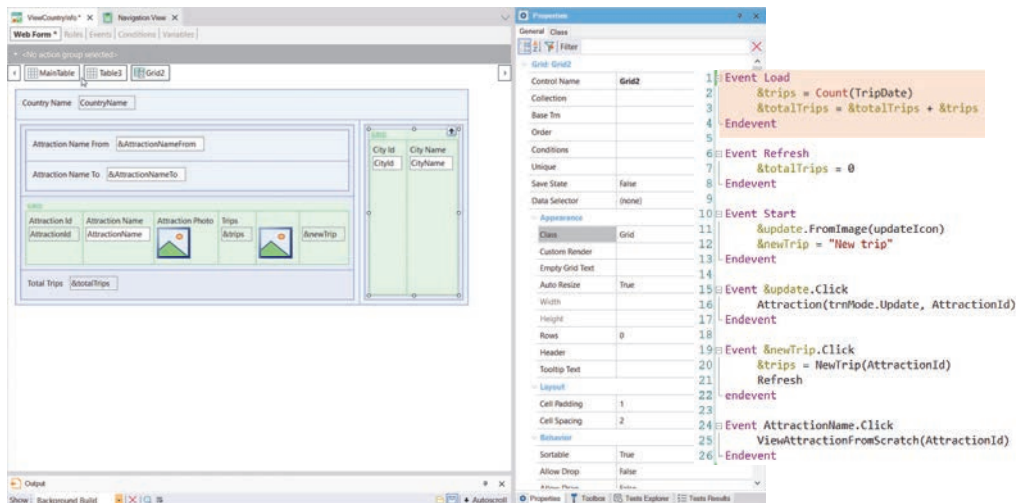
```
Event Grid1.Load
    &trips = Count(TripDate)
    &totalTrips = &totalTrips + &trips
Endevent
```

Grid1

+ Grid2

ここまで何度か言及したように、汎用イベントは、グリッドのない Web パネルや単一グリッドの Web パネルのケースでのみ機能しますが、汎用イベントではなくグリッドの Load イベントを使用する方がよい場合もあります。グリッドの Load イベントを使用すると、将来別のグリッドが必要になったときに対応することができます。

新しいグリッド



国の情報 (国名と観光名所) を表示する Web パネルに、都市のグリッドも追加します。これを行う前にナビゲーション表示を見てみると、現時点では単一グリッドのロードがあることに注目してください。

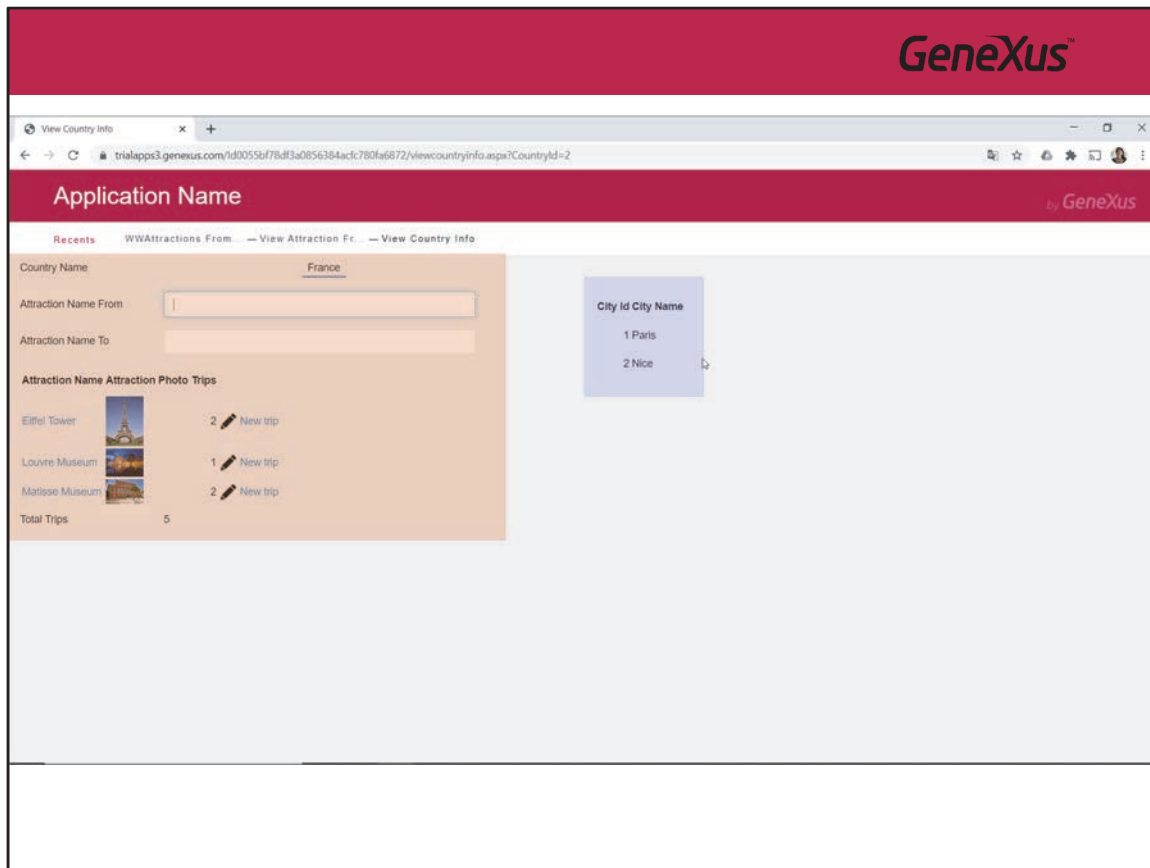
都市のグリッドを追加する前に、画面上にテーブルコントロールを挿入し、そこに国の観光名所に関するグリッドや旅行の数のコントロールを移動させ、一か所にまとめます。

次に、都市の情報のための別のテーブルコントロールを挿入します。そこで、CityId 項目属性と CityName 項目属性で構成される新しいグリッドを挿入します。プロパティを見ると、既定で Grid2 という名前が付けられていることが分かります。

各グリッドには、ベーステーブルが存在する場合と存在しない場合があります。このケースでは、どちらのグリッドにも項目属性が含まれているため、両方ともベーステーブルが存在します。汎用の Load イベントのコードがどちらに適用されるか、どうすれば分かるでしょうか。保存すると、ナビゲーション表示に警告エラーが表示されます。

これは、各グリッドをロードするために必要なナビゲーションを示しています。また、旅行の計算式は Grid1 のロードに属している必要があることも認識されていて、これを指定するよう求められます。それに従います。

これで、オブジェクトを保存するときに、ナビゲーション表示にエラーが表示されなくなります。



実行してみます。

これらは 2 つの独立したナビゲーションですが、両方ともパラメーターで受け取った国に関連しているため、どちらも国でフィルタリングされています。

新しいグリッド

```

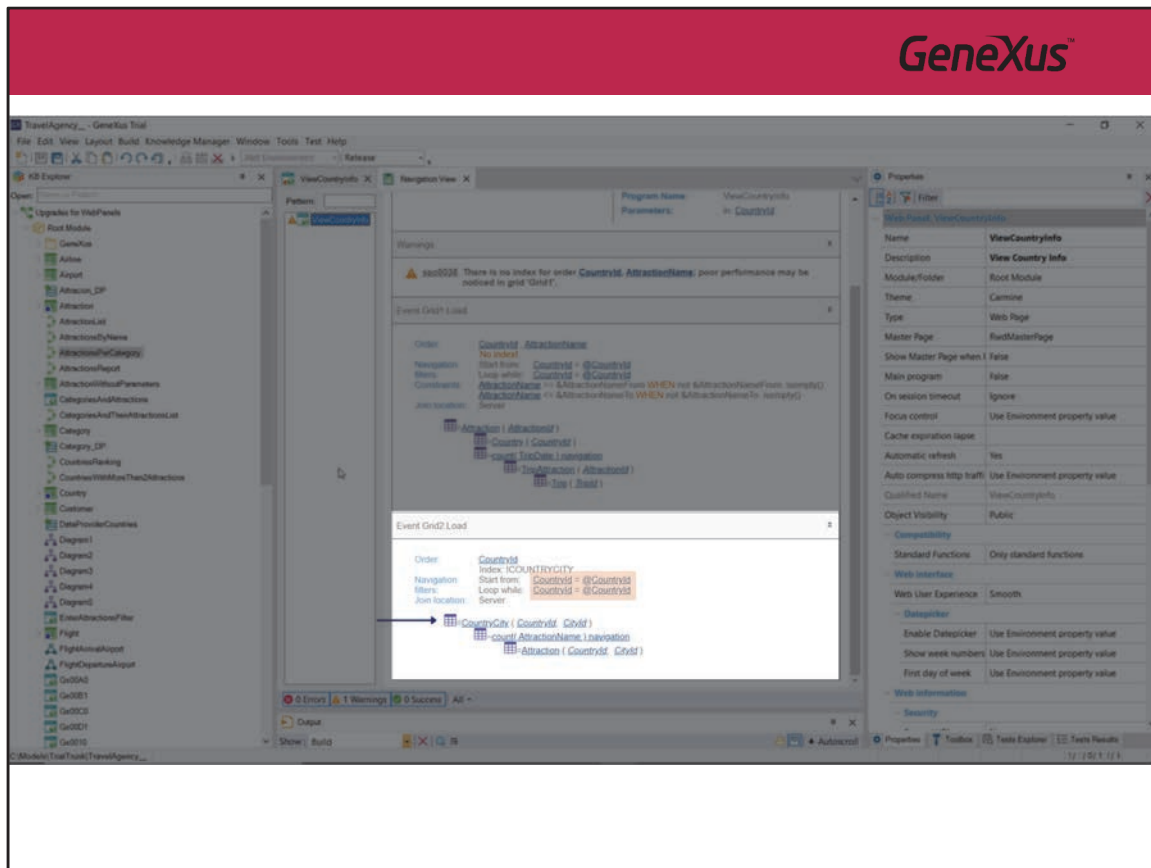
1: Event Grid1.Load
2:   &trips = Count(TripDate)
3:   &totalTrips = &totalTrips + &trips
4: Endevent
5:
6: Event Grid2.Load
7:   &attractions = Count(AttractionName)
8: Endevent
9:
10:
11: Event Refresh
12:   &totalTrips = 0
13: Endevent

```

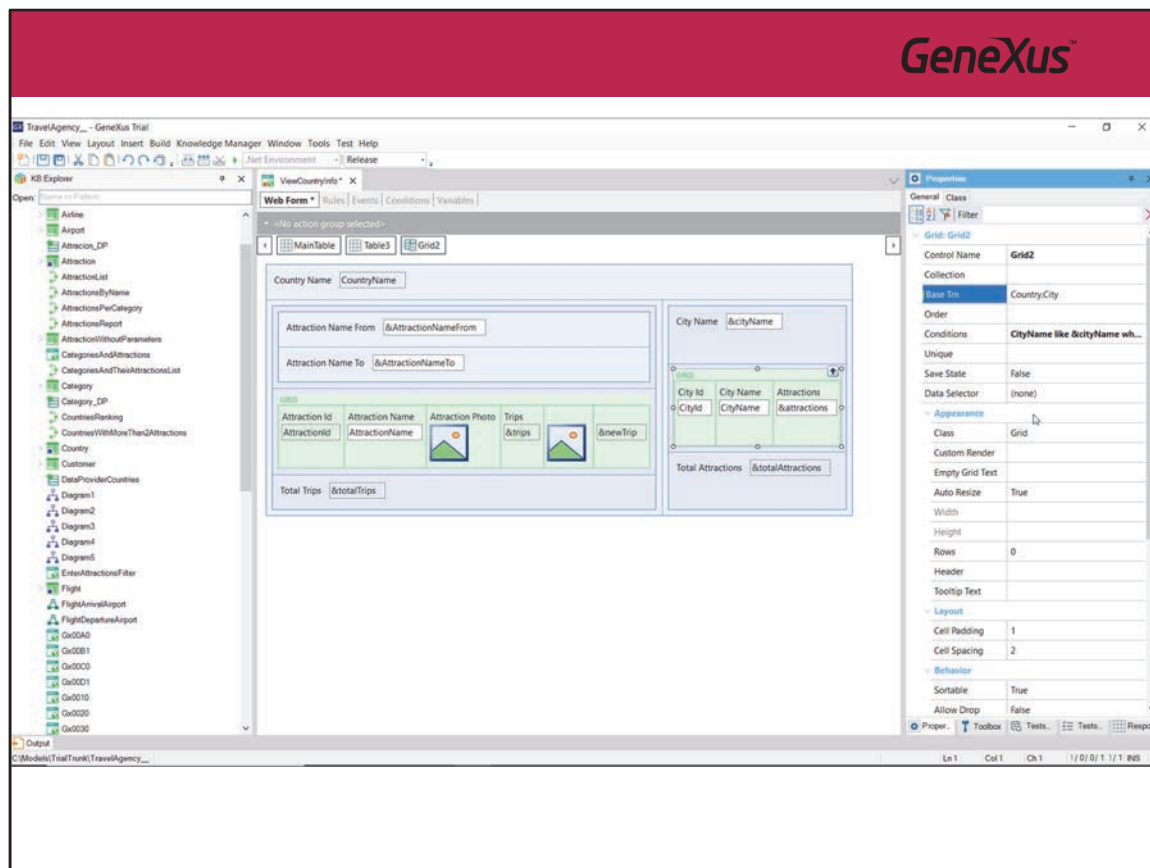
観光名所を含む旅行の数を計算する場合と同様に、各都市にある観光名所の数を計算する場合は、変数 &attractions をグリッドに追加し、明細行がロードされるたびに、つまり Grid2 というグリッドの Load イベントで計算されるようにします。

国と都市の観光名所のみを計算するように、この式を条件付けする必要がないのはなぜでしょうか。

実行してみます。



ビルドしている間に、ナビゲーション表示を見てみましょう。パラメーターで受け取った国に対応する都市のテーブルでレコードが見つかるたびに実行される Load イベント内で、Attraction に対して Count 式の計算がトリガーされ、国と都市でフィルタリングされます。



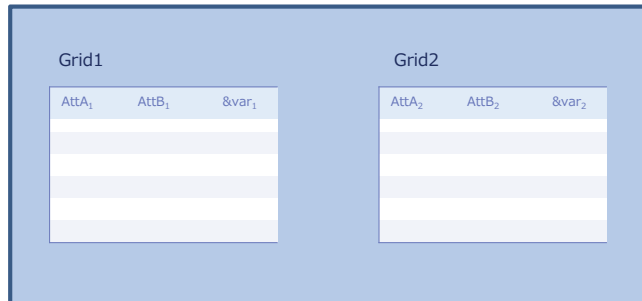
機能的統一を図るために、グリッドに表示される都市をフィルタリングするための変数を追加し、すべての都市の観光名所の合計を表示するように別の変数を追加します。ここでその作業を行いました。

Like 演算子を使用して、フィルタをグリッド条件に配置していることに注目してください。ベーストランザクションは指定しませんでした、GeneXus で自動的に検出されました（これは好都合です）。

汎用の Refresh イベントで &totalTrips 変数を初期化しており、&totalAttractions 変数も初期化する必要があります。

ただし、実際には 3 つの Refresh イベントがあります。ここまでにプログラムした汎用 Refresh イベントと、各グリッドの Refresh イベントです。

イベントの実行順序



Start
Refresh

Grid1.Refresh
Grid1.Load

Grid2.Refresh
Grid2.Load

Web パネルが初めて実行されるときイベントの実行順序は次のとおりです。

Start イベント

最初に汎用 Refresh イベントが実行されます。

次に 1 つ目のグリッドの Refresh イベントが実行されます。ベーステーブルが存在する場合は、そのテーブル全体を対象として、対応するレコードがフィルタリングされ、各レコードに対してそのグリッドの Load イベントが実行されます。ベーステーブルが存在しない場合は、グリッドの Load イベントは 1 回だけ実行されます。

その後、2 つ目のグリッドの Refresh イベントと Load イベントが同じように実行されます。

イベントの実行順序

The screenshot shows a web application interface with two main sections. The left section contains a search form for attractions with fields for 'Country Name', 'Attraction Name From', and 'Attraction Name To'. Below this is a grid with columns: 'Attraction Id', 'Attraction Name', 'Attraction Photo', 'Trips', and a button '&newTrip'. At the bottom of this section is a 'Total Trips' label and the variable '&totalTrips'. The right section contains a search form for cities with a 'City Name' field. Below this is a grid with columns: 'City Id', 'City Name', and 'Attractions'. At the bottom of this section is a 'Total Attractions' label and the variable '&totalAttractions'.

```

Event Grid1.Refresh
  &totalTrips = 0
Endevent

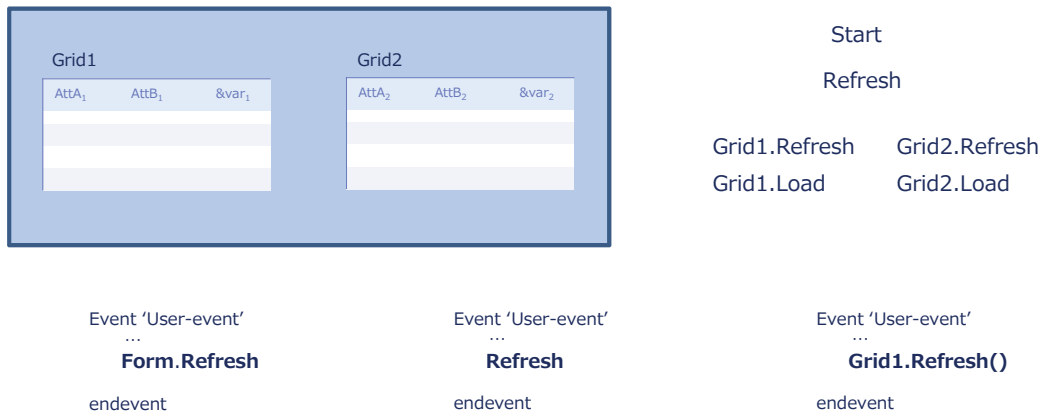
Event Grid2.Refresh
  &totalAttractions = 0
Endevent

```

この例では、汎用の Refresh イベントではなく、各グリッドの Refresh イベントで変数 `&totalTrips` と `&totalAttractions` を初期化する必要があります。その理由は、後でグリッドのフィルタ変数を変更する必要がある場合に、画面全体を再表示しなくても、そのグリッドのみを再表示すればよいからです。

次に、この方法でイベントを変更します。

再表示の対象



もちろん、グリッドをさらに追加するということは、別の章で見たユーザーイベントに記載された Refresh コマンドも、特定のグリッドに適用することができるということです。

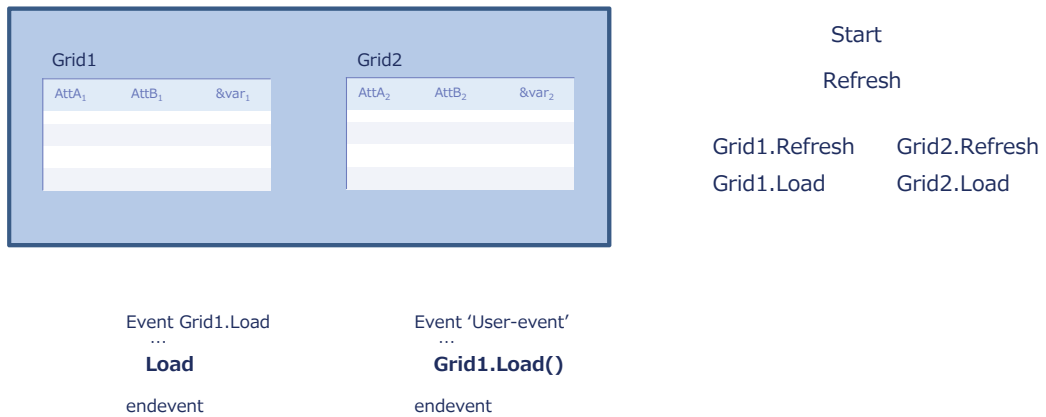
次に例を示します：

Form.Refresh というコマンドがあり、このコマンドを使用すると、Start、汎用の Refresh、および各グリッドの Refresh と Load が実行され、ページ全体が再表示されます。

汎用の **Refresh** コマンド (これまで説明してきたもの) を使用すると、汎用の Refresh、および各グリッドの Refresh と Load (つまり、Start 以外のすべて) が実行されます。

さらに、グリッドの Refresh メソッドがあり、このメソッドではグリッドのみが再表示されます。つまり、グリッドの Refresh と Load が (ベーステーブルの有無に応じて 1 回または n 回) 実行されます。

Load コマンド



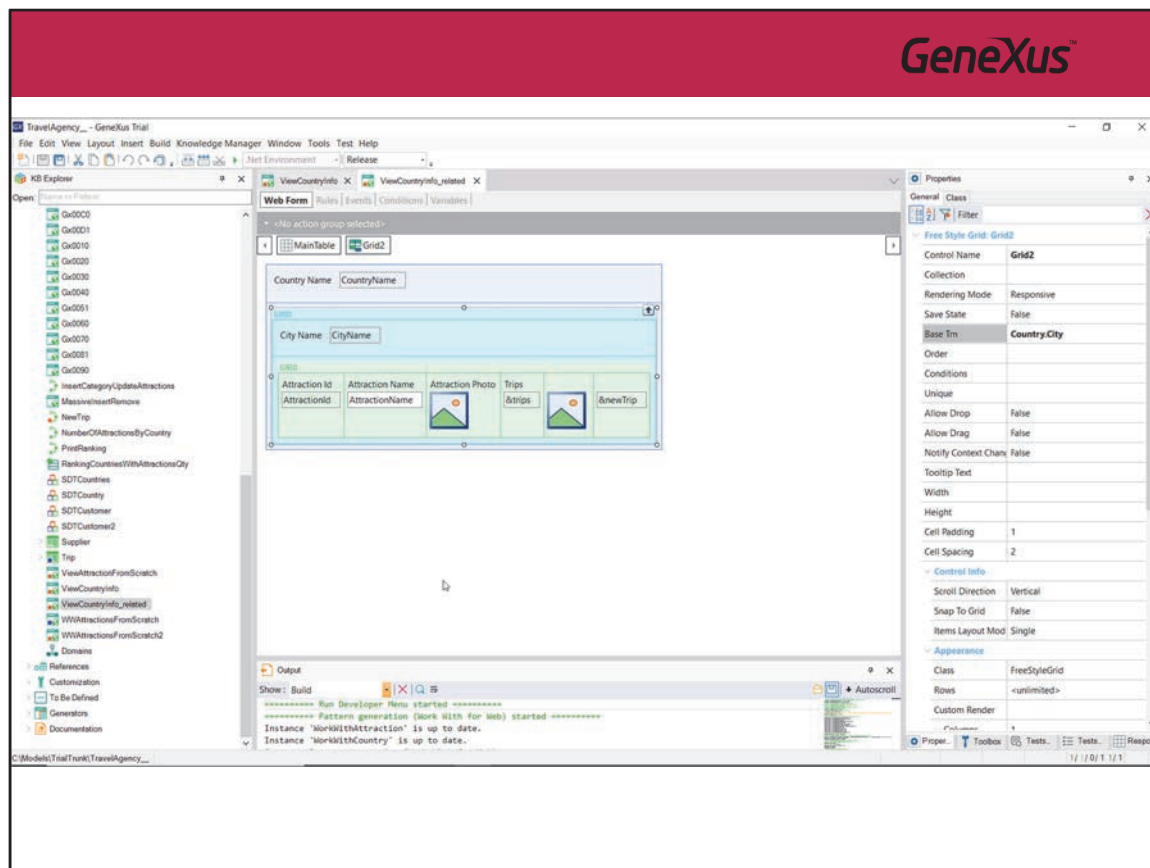
Load コマンドに関しては、状況が少し異なります。

複数のグリッドがある場合は、対象グリッドの **Load イベント**内に **Load コマンド**のみを記述できます。

また、ユーザーイベントからグリッドの 1 つに明細行をロードするには、必然的に、対象グリッドの **Load メソッド**を使用する必要があります。

並列のグリッドとネストされたグリッド

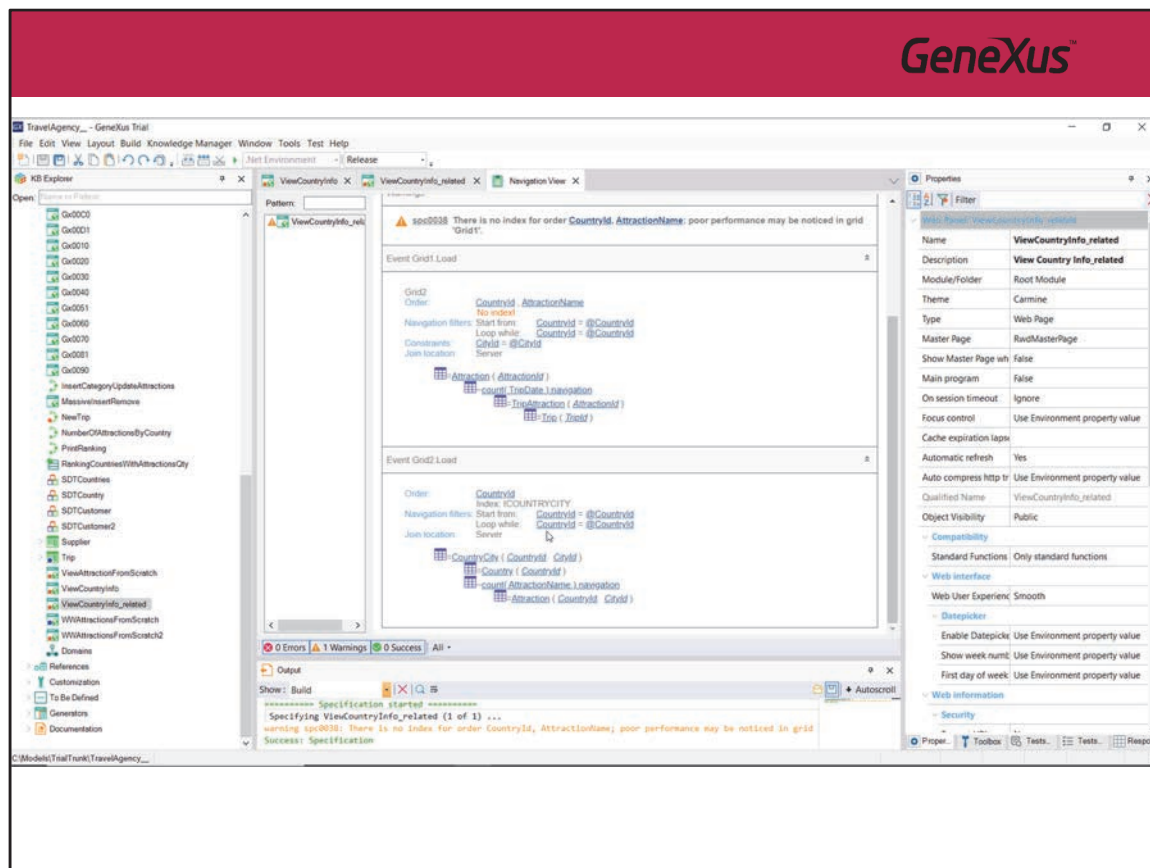
ここでは、並列のグリッドの例だけを見てきましたが、グリッドは、For Each コマンドのようにネストすることもできます。



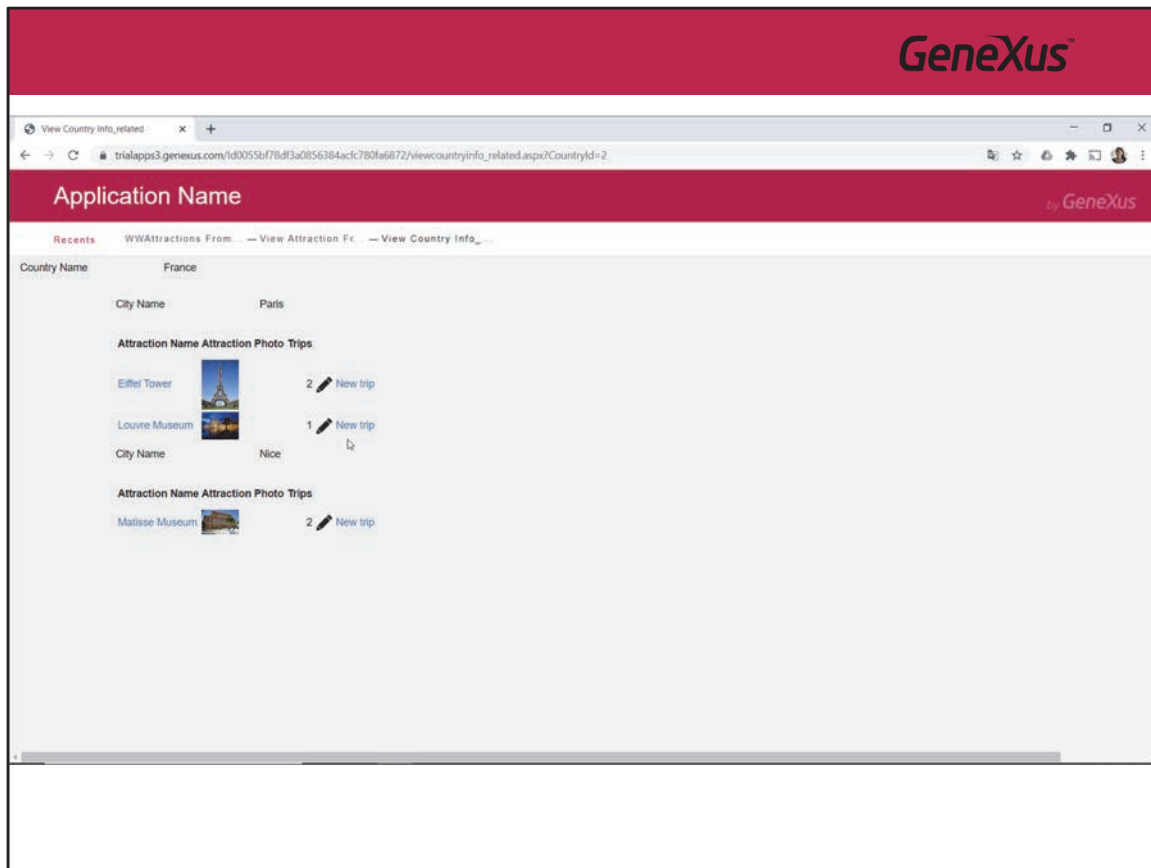
たとえば、選択した国と、前に説明したような情報を関連付けて表示したい場合があります。

ここでそれを実装します。グリッドに別のグリッドを含めるには、特殊なタイプのグリッド (テーブル形式ではなく、フリースタイル) を使用する必要があります。フリースタイルグリッドと呼ばれるグリッドです。

このグリッドは CountryCity テーブル全体を対象とするものであり、見つけた都市ごとに、その国/都市の観光名所を検索する 2 つ目のグリッド (ネストされたグリッド) の Refresh と Load が実行されます。そこで、情報に関連があることを認識できます。



ナビゲーション表示では、CountryCity テーブル全体を対象として、パラメーターで受け取った国で Grid2 (都市のグリッド) がフィルタリングされていることが分かります。次に、観光名所に対応する Grid1 の Load が Attraction テーブル全体を対象として実行され、CountryCity の各レコードに配置されている国と都市でフィルタリングされます。そのため、CityId にアットマーク (@) が表示されています。



実行すると、このようになります。パリには 2 つの観光名所があります。ニースには 1 つの観光名所があります。

並列のグリッドとネストされたグリッド

このトピックに関しては、さらに学習すべきことがありますが、別の章で改めてご紹介します。