

## Describing more objects from reality

The application grows up

*GeneXus™ 16*

## Describing more objects from reality



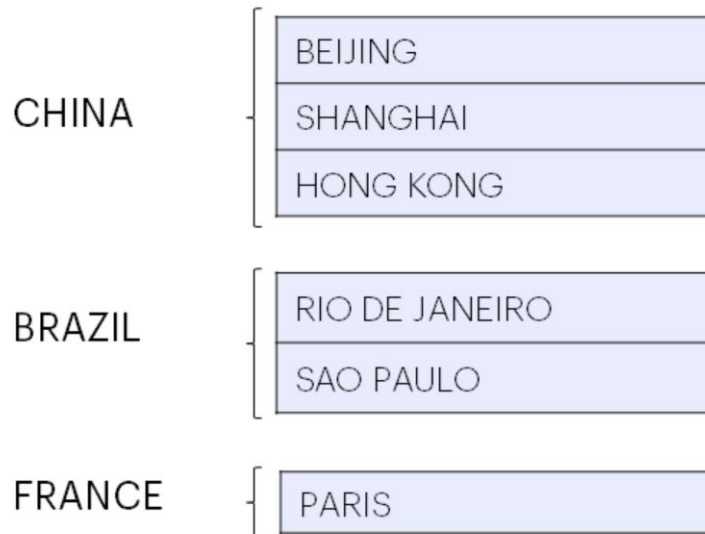
When we show what we've developed so far to our users at the Travel Agency, they approve it and remind us that **for each tourist attraction, they want to record the city in which it is located.**

Therefore, our application must take into account that **countries contain a set of cities**, and also that each tourist attraction belongs to a city.

## Describing more objects from reality

ATTRACTIONS	
Name:	Eiffel Tower
Country:	France
City:	Paris
Image:	
Category:	Monument

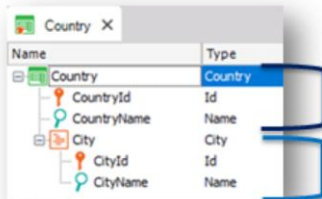
GeneXus provides an easy way to represent a reality such as this of countries and cities.

**Countries and cities**

It is implemented by adding a second level to the Country transaction. Let's see it.

## The idea is to record...

1. Countries contain a set of cities.
2. Each tourist attraction belongs to one city.
3. Define a second level in Country transaction.



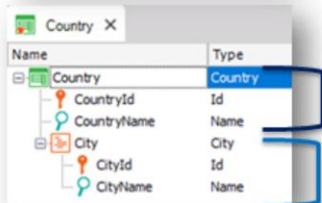
First level

Second level

One country



Many cities



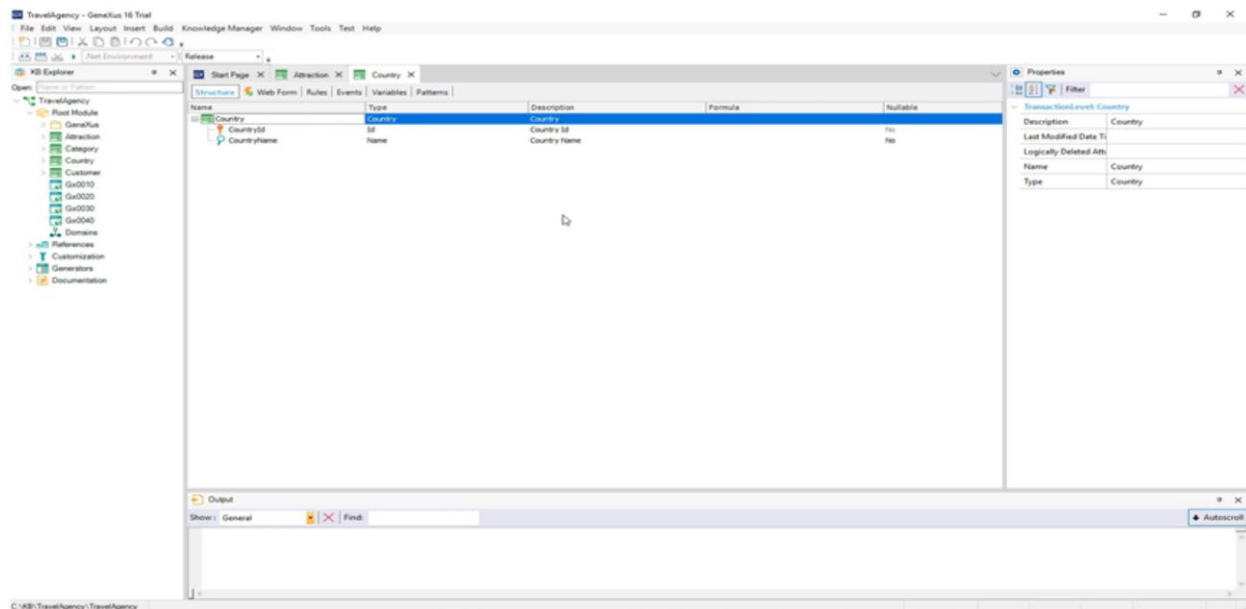
First level

Second level

Only one country



## DEMO



[ DEMO: <https://youtu.be/eLPFwD6-1xU> ]

We open the Country transaction and, while positioned on the last attribute, we right-click and select **Insert Level**.

As we can see, a sublevel is opened. We call it: City.

Now there are 2 possible ways to name the attributes on the 2<sup>nd</sup> level. If we type a period, we see that GeneXus suggests "CountryCity" as prefix, that is to say, the transaction name + the 2<sup>nd</sup> level name. We would only have to complete the attribute name... in this way.

If, on the other hand, we type inverted commas we see that GeneXus suggests the prefix "City", the 2<sup>nd</sup> level name... We would only have to complete it by adding Id at the end, and we would have the name CityId.

We will leave it like this.

Note that it takes the Id domain... and in the next line we also type inverted commas and complete the attribute name with Name.

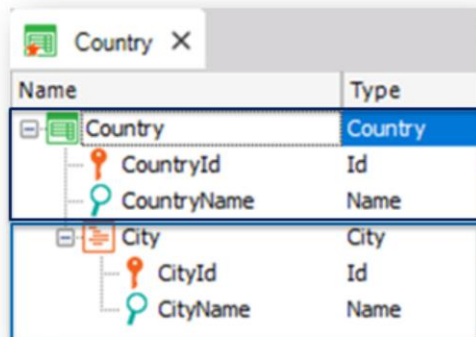
**This two-level transaction indicates that each country has several cities, and that each city belongs to only one country.**

We save the changes and note how GeneXus has changed the Web Form for this transaction after creating a second level with cities.

As we can see, for each country now we can enter **a group of cities**.

Let's go back to the transaction structure.

## Physical tables generated



Country Table

CountryId	CountryName
1	BRAZIL
2	FRANCE
3	CHINA

CountryCity Table

CountryId	CityId	CityName
1	1	RIO DE JANEIRO
1	2	SAO PAULO
2	1	PARIS
3	1	BEIJING
3	2	SHANGHAI
3	3	HONG KONG

The primary key consists of 2 attributes

For every two-level transaction, GeneXus determines that it has to create **two physical tables**:

**One table arises from the first level**, in this case to record the countries, with CountryId primary key.

**And another physical table, associated with the second level**, in this case to record the cities in each country.

Let's focus on the primary key of the second table that will be created. It is composed of 2 attributes: CountryId and CityId.

This means that the **unique identifier** of the cities is composed of both attributes.

## Impact Analysis

The screenshot shows the 'Impact Analysis' window in GeneXus. At the top, a message states: 'Database needs to be reorganized. This report describes Database changes and how they will be handled by reorganization programs. Please select Reorganize to proceed or Cancel.' Below this are 'Reorganize' and 'Cancel' buttons.

On the left, under 'Pattern:', there is a search bar and a list containing 'CountryCity' with a green checkmark icon.

The main area displays the 'Table CountryCity specification'. It includes the following sections:

- Table name:** CountryCity
- CountryCity is new**
- Table Structure**

Attribute	Definition	Previous values	Takes value from
CountryId	Numeric (4), Not null		
CityId	Numeric (4), Not null		
CityName	Character (20), Not null		
- Indexes**

Name	Definition	Composition
ICOUNTRYCITY	primary key Clustered	CountryId CityId
- Foreign key constraints**

Foreign key constraints	Referenced table	Attributes
	Country	

At the bottom left, a status bar shows: 0 Errors, 0 Warnings, 1 Success.

Let's execute the application to see this.

We press F5.

Note that GeneXus is proposing the creation of a table called CountryCity in the database.

This is the table created from the second level that we defined in the Country transaction... and its name is taken from the transaction name plus the name we gave to the second level.

As we explained before, its primary key is composed of the CountryId and CityId attributes.

We execute the creation of this table in the database... The necessary programs are generated... and the application is run.



## DEMO

The screenshot displays a web browser window with the URL <https://trialapps3.genexus.com/1dd619baad8f55020826f11d4466430e3a/country.aspx>. The application has a red header bar with 'Application Name' and the GeneXus logo. Below the header, there are tabs for 'Recents' and 'Country'. The main content area is titled 'Country' and contains a form with the following fields:

- Id**: A text input field with a dropdown arrow.
- Name**: A text input field.
- City**: A section containing a table with columns 'Id' and 'Name'. The table has five rows, each with a dropdown arrow in the 'Id' column and a text input field in the 'Name' column. A '[New row]' link is at the bottom of the table.

At the bottom of the form, there are two buttons: 'CONFIRM' and 'CANCEL'.

[ DEMO: <https://youtu.be/0z43AvZDC8E> ]

We will execute the Country transaction to add some cities for the countries that we have entered.

We query the first country that we have entered... it is Brazil, and we add a city to it.

We identify it as city 1 of this country, and it is called: Rio de Janeiro.

Now we will enter another city for Brazil and code it as city 2 of Brazil. It is Sao Paulo.

We confirm.

Now we go to the second country that we have entered, which is France. We add only one city: Paris.

We confirm.

Lastly, we will add cities for China.

We find China... And we add as city no. 1: Beijing; as city no. 2: Shanghai and as city no. 3: Hong Kong.

We confirm.

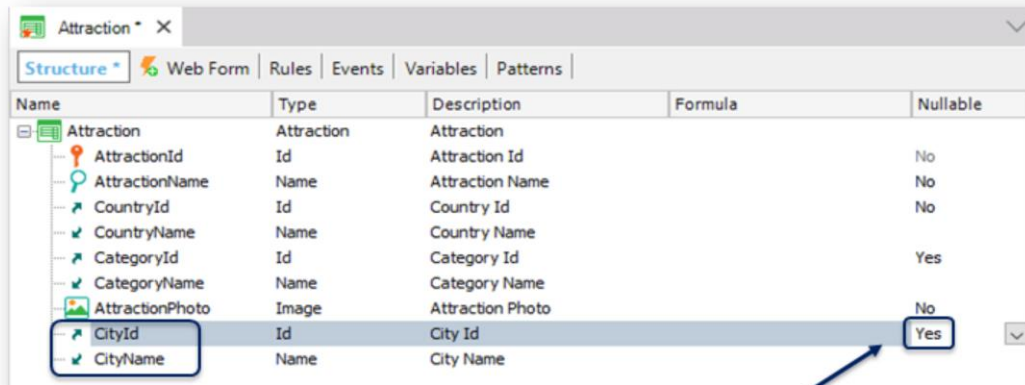
As we can see, different countries can have their cities identified as 1, 2, 3 respectively... because cities are not only identified by their city number, but they are also identified according to the country to which they belong.

Note that for the same country, the city identifier value cannot be repeated.

The application informs us that this country already has a city with this identifier value.

## Each tourist attraction belongs to one city

We add to the Attraction transaction:



To allow the entry of attractions without indicating a value in city (client requirement).

Now let's go back to GeneXus and assign a city to each attraction.

We open the Attraction transaction and add a new attribute. We type "c".

And choose CityId.

We press Enter to add another attribute, and choose CityName:

At the Travel Agency we were told that the attraction city must give the option to not set it. To meet this request, in CityId we will set the Nullable property to Yes.

As we've said before, this property is only set to Yes for foreign keys, because in this case the referenced value is checked for existence. This is the way to also enable the value to not be set.

We save the changes... and upload the modified objects to GXserver.

We then select Knowledge Management/Team development, add a comment and press Commit.

We press F5...

## Impact Analysis

**Database needs to be reorganized.**  
This report describes Database changes and how they will be handled by reorganization programs.  
Please select Reorganize to proceed or Cancel.

Reorganize Cancel

Pattern:

Attraction

**Table Attraction specification**

Table name: [Attraction](#)

Attraction needs conversion

Table Structure

Attribute	Definition	Previous values	Takes value from
<a href="#">AttractionId</a>	Numeric (4), Not null		<a href="#">Attraction</a> <a href="#">AttractionId</a>
<a href="#">AttractionName</a>	Character (20), Not null		<a href="#">Attraction</a> <a href="#">AttractionName</a>
<a href="#">CountryId</a>	Numeric (4), Not null		<a href="#">Attraction</a> <a href="#">CountryId</a>
<a href="#">CategoryId</a>	Numeric (4)		<a href="#">Attraction</a> <a href="#">CategoryId</a>
<a href="#">AttractionPhoto</a>	Image, Not null		<a href="#">Attraction</a> <a href="#">AttractionPhoto</a>
<a href="#">AttractionPhoto_Gxi</a>	Varchar (2048), Not null		<a href="#">Attraction</a> <a href="#">AttractionPhoto.Uri</a>
New <a href="#">CityId</a>	Numeric (4)		Null

Indexes

0 Errors 0 Warnings 1 Success

We are informed that the Attraction table needs to be converted by adding the CityId attribute, which can be left unset.

We proceed to reorganize it...

**So far, we have seen that:**

- A transaction can have more than one level
- Each level will correspond to a physical table
- Subordinate levels will determine that tables have foreign keys
- Cities are weak entities compared to countries.



Videos

[training.genexus.com](http://training.genexus.com)

Documentation

[wiki.genexus.com](http://wiki.genexus.com)

Certifications

[training.genexus.com/certifications](http://training.genexus.com/certifications)