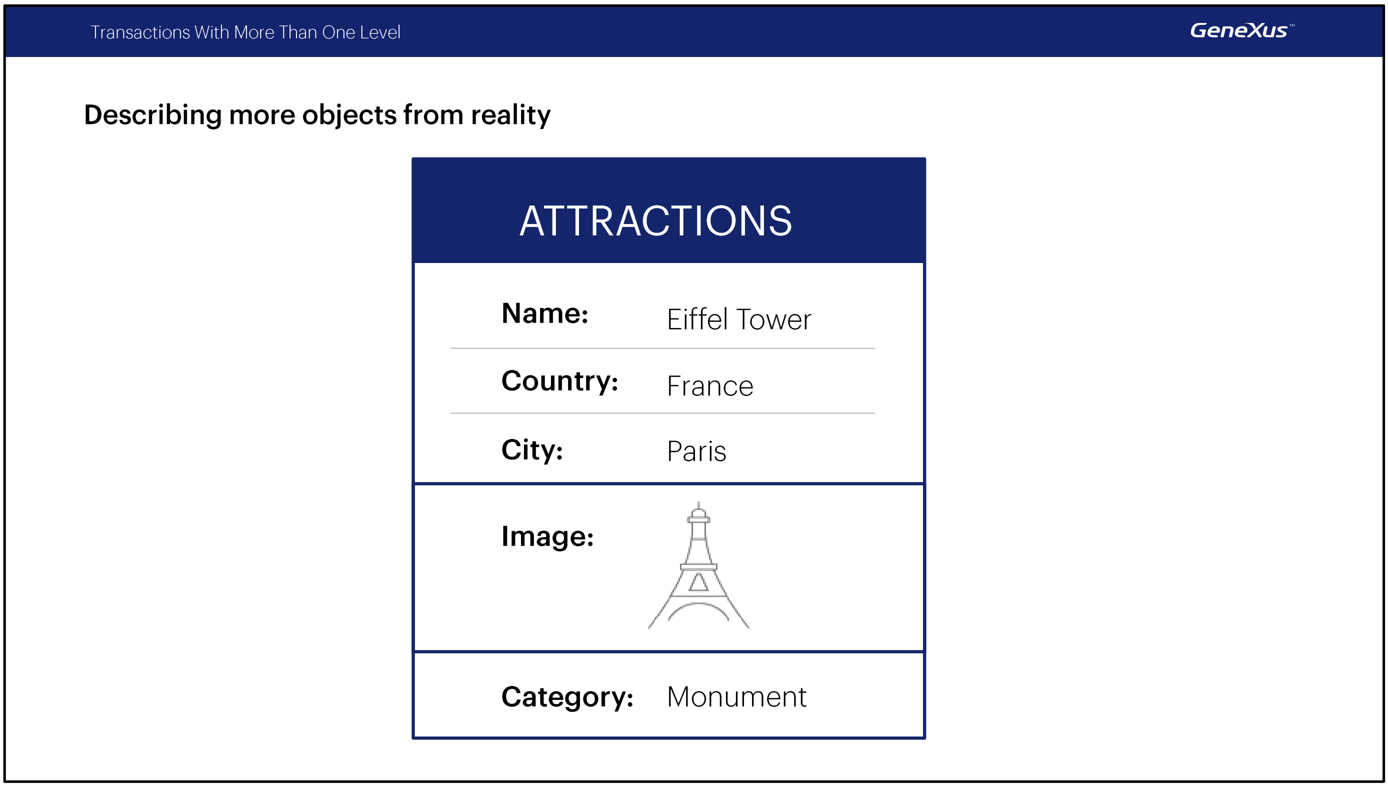
Transactions With More Than One Level

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.

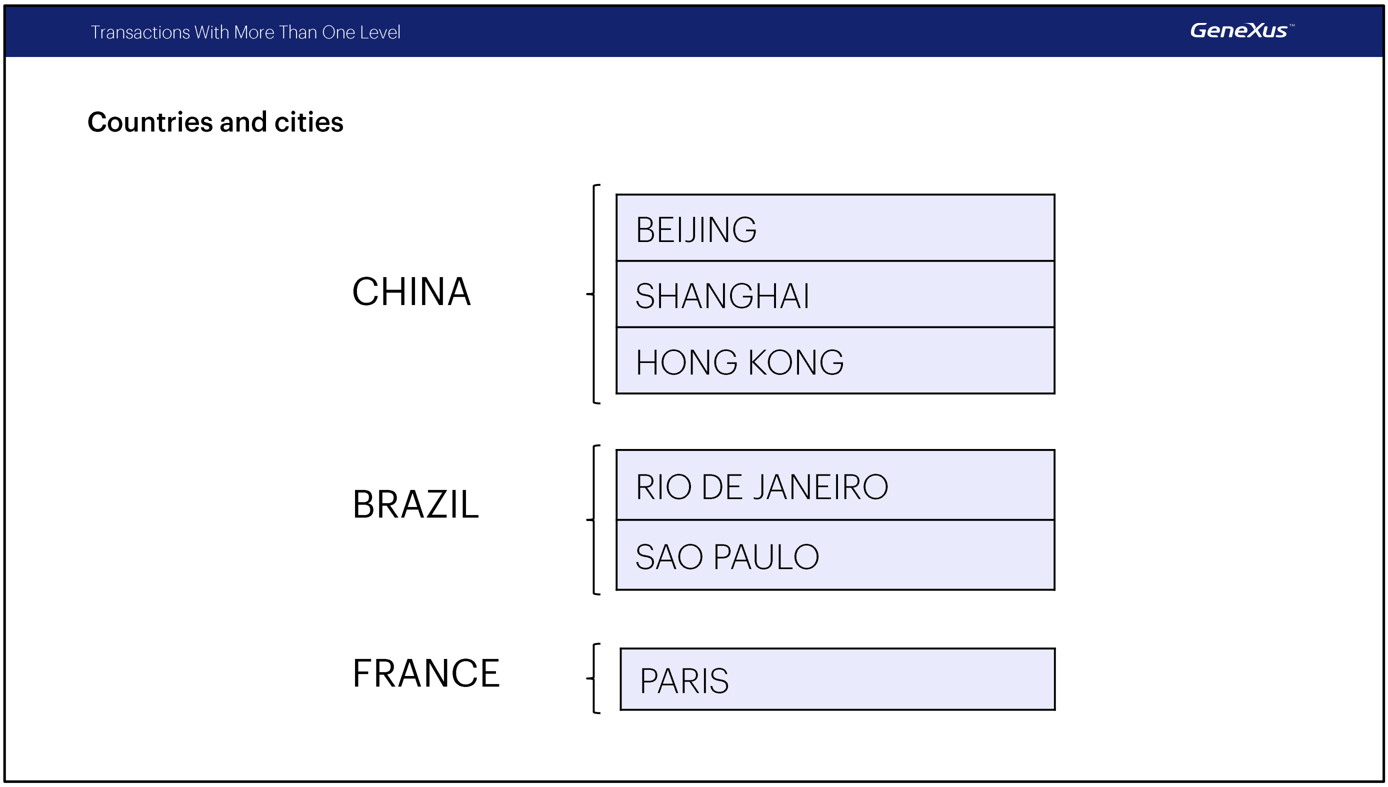


Describing more objects from reality

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.



Transactions With More Than One Level

GeneXus

Countries and cities

CHINA

- BEIJING
- SHANGHAI
- HONG KONG

BRAZIL

- RIO DE JANEIRO
- SAO PAULO

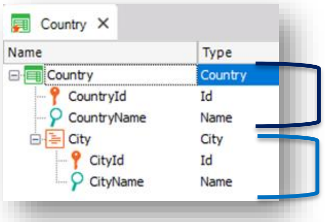
FRANCE

- PARIS

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.

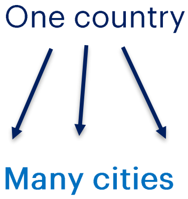


Country ✕

Name	Type
Country	Country
CountryId	Id
CountryName	Name
City	City
CityId	Id
CityName	Name

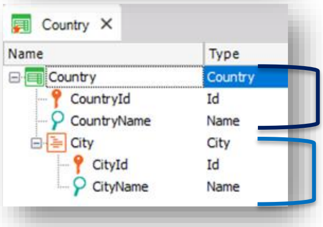
First level

Second level



One country

Many cities

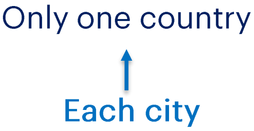


Country ✕

Name	Type
Country	Country
CountryId	Id
CountryName	Name
City	City
CityId	Id
CityName	Name

First level

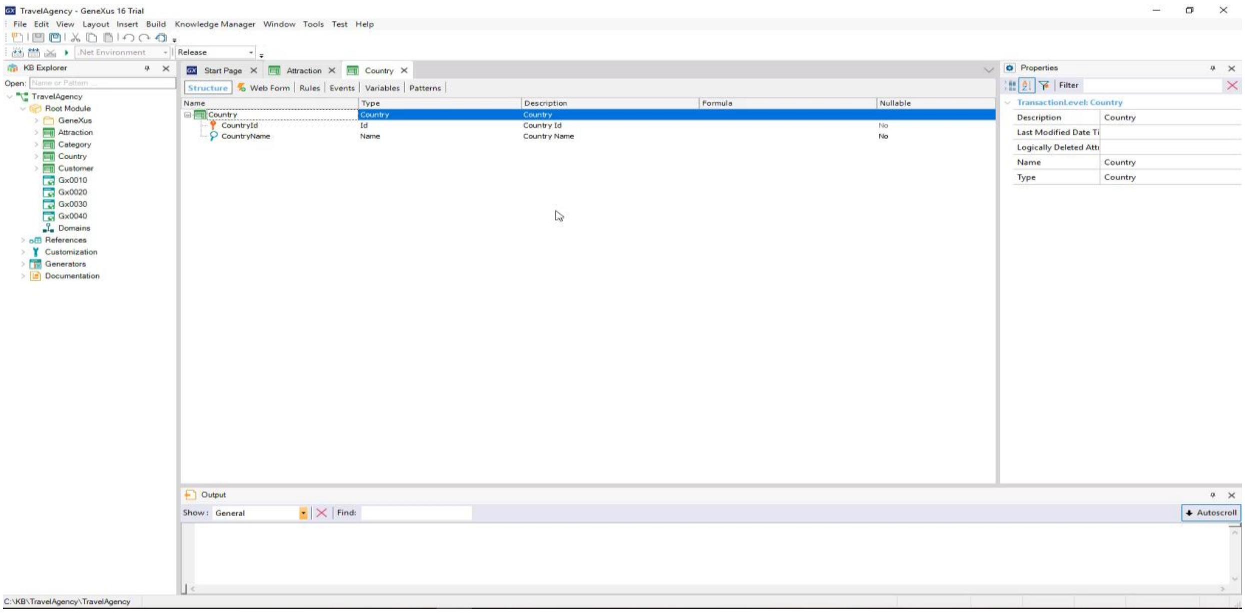
Second level



Only one country

Each city

DEMO



Name	Type	Description	Formula	Nullable
Country	Country	Country		
CountryId	Id	Country Id		No
CountryName	Name	Country Name		No

TransactionLevel: Country

Description	Country
Last Modified Date Ti	
Logically Deleted Att	
Name	Country
Type	Country

[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 2nd level. If we type a period, we see that GeneXus suggests “CountryCity” as prefix, that is to say, the transaction name + the 2nd level name. We would only have to complete the attribute name… in this way.

If, on the other hand, we type inverted comas we see that GeneXus suggests the prefix “City”, the 2nd 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.

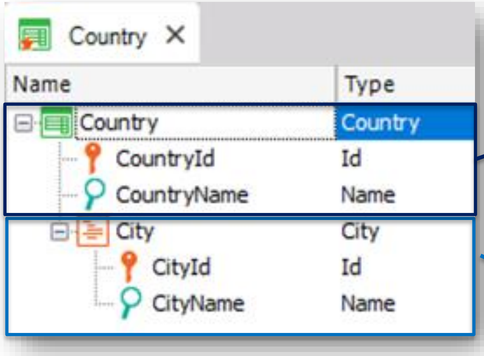
As we can see, for each country now we can enter a group of cities.

Note that the second level of the transaction, named City, is displayed in the form as a grid control. And inside this one, the attributes defined there.
In this way, for each country it is possible to enter a group of cities.

Let's go back to the transaction structure.

This 2-level transaction represents that each country contains several cities and that each city belongs to a single country.

Physical tables generated



Name	Type
Country	Country
CountryId	Id
CountryName	Name
City	City
CityId	Id
CityName	Name

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:

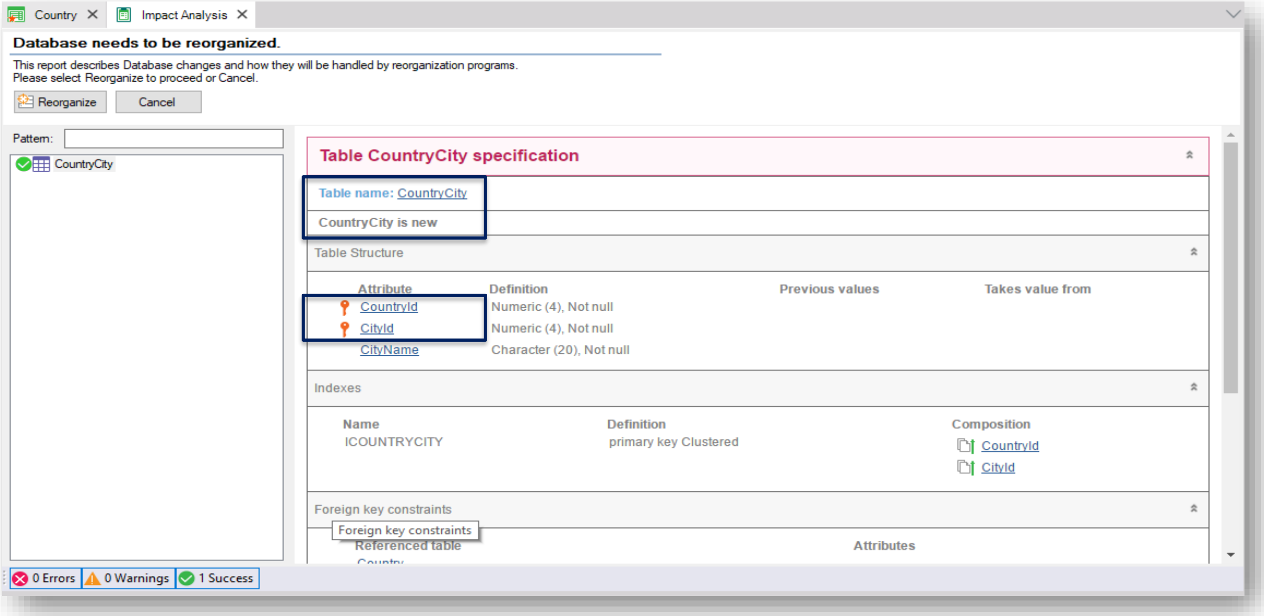
One table arises from the first level, in this case to record the countries, with CountryId primary key.

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



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.

Table CountryCity specification

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	

Let's execute the application to see this.

We press F5.

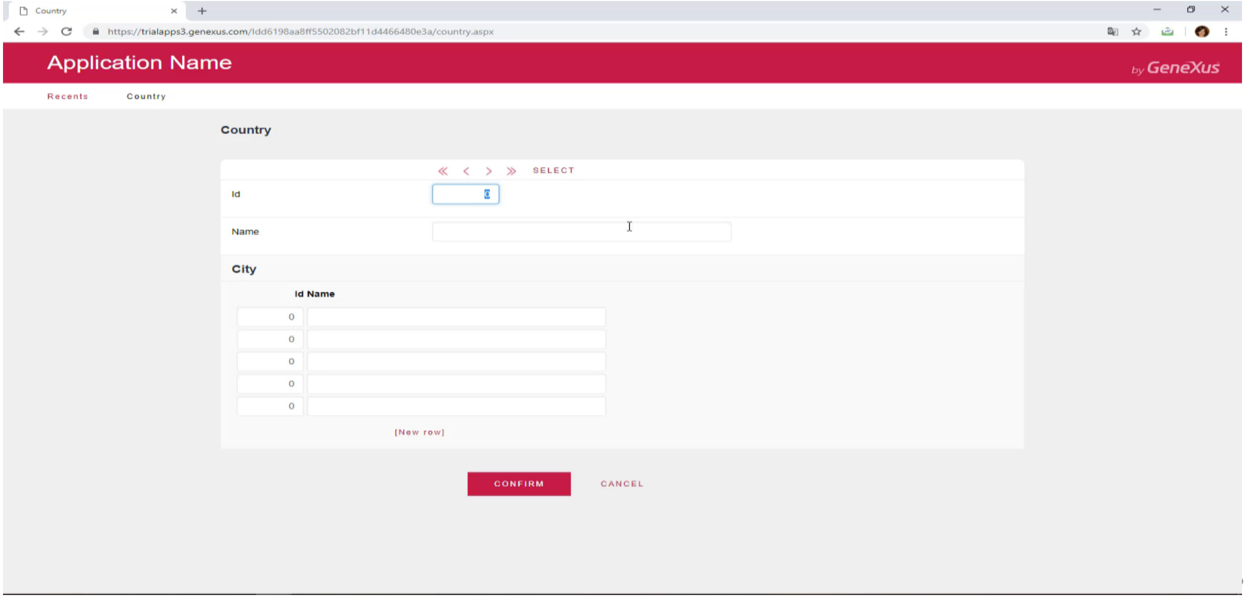
Note that GeneXus is proposing the creation of a table called CountryCity in the database.

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



[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.

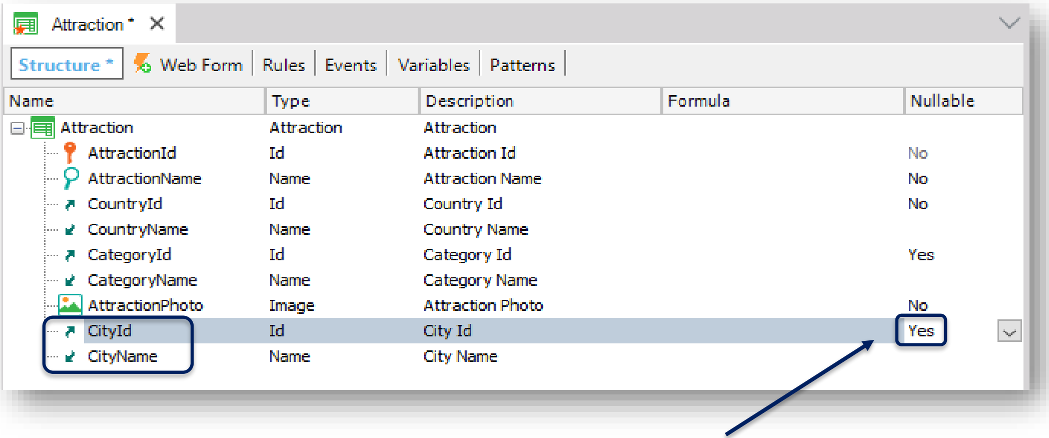
As we can see, different countries can have their cities identified as 1, 2, 3 respectively... because cities are not only identified by their city number, but they are also identified according to the country to which they belong.

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:



Name	Type	Description	Formula	Nullable
Attraction	Attraction	Attraction		
AttractionId	Id	Attraction Id		No
AttractionName	Name	Attraction Name		No
CountryId	Id	Country Id		No
CountryName	Name	Country Name		
CategoryId	Id	Category Id		Yes
CategoryName	Name	Category Name		
AttractionPhoto	Image	Attraction Photo		No
CityId	Id	City Id		Yes
CityName	Name	City Name		

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.

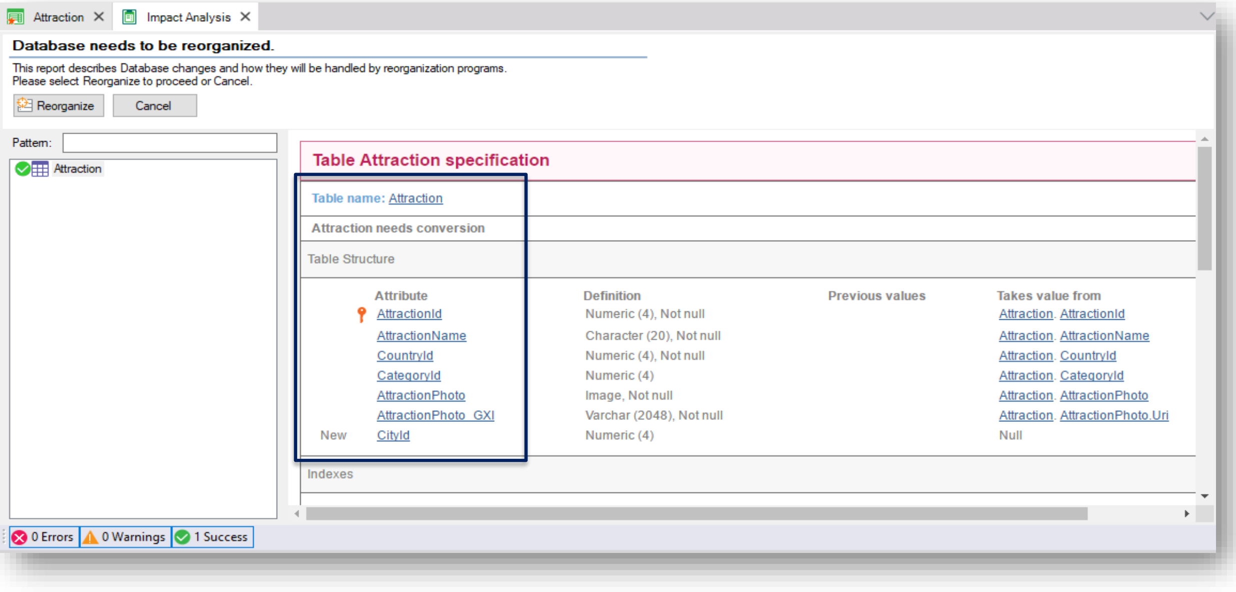
The order in which the attributes are entered in the transaction is the same as the order in which they will be displayed in the web form.
If for example, we move the CityId and CityName attributes, leaving them under the CountryName attribute and save the changes...
You can see that the order is also changed in the web form.

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.

Table Attraction specification

Table name: Attraction

Attraction needs conversion

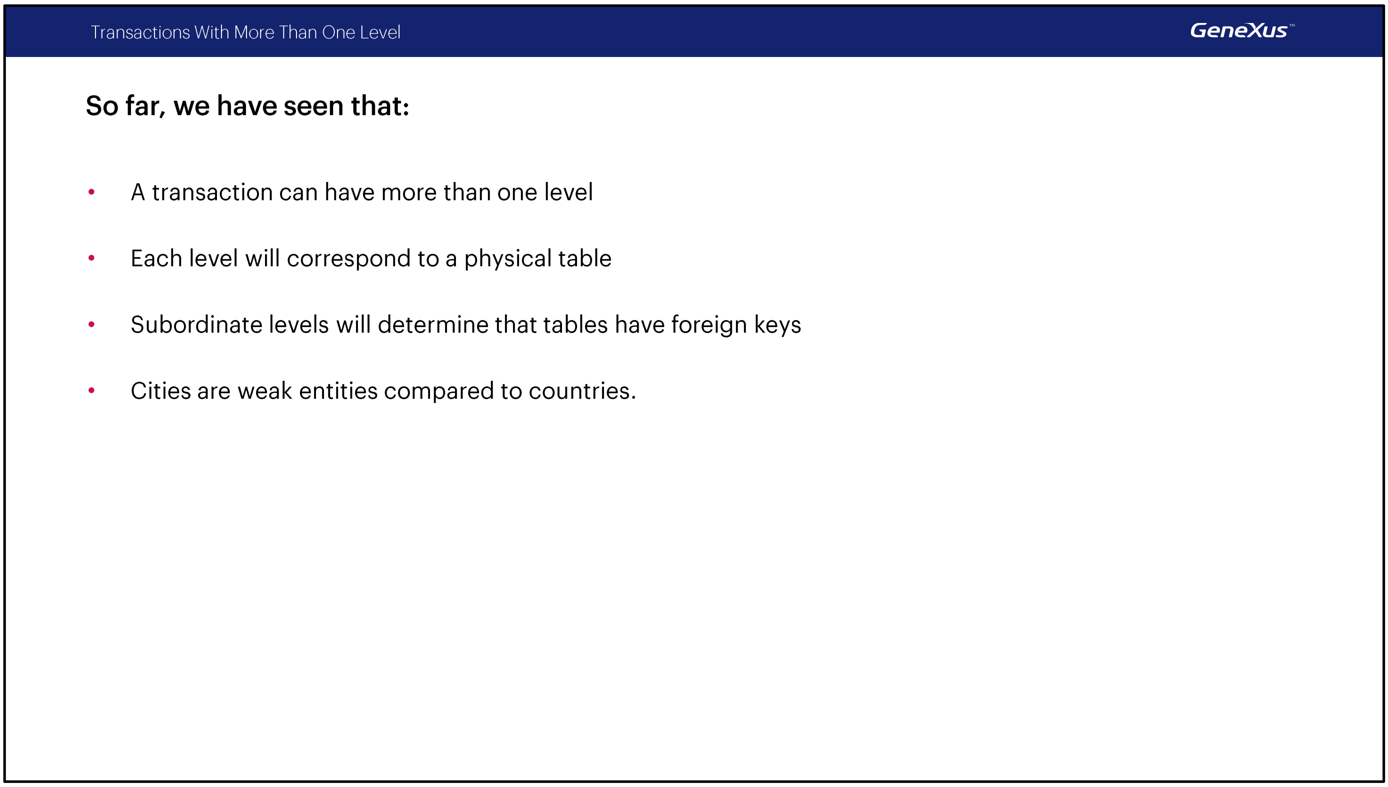
Table Structure

	Attribute	Definition	Previous values	Takes value from
	AttractionId	Numeric (4), Not null		Attraction. AttractionId
	AttractionName	Character (20), Not null		Attraction. AttractionName
	CountryId	Numeric (4), Not null		Attraction. CountryId
	CategoryId	Numeric (4)		Attraction. CategoryId
	AttractionPhoto	Image, Not null		Attraction. AttractionPhoto
	AttractionPhoto_GXI	Varchar (2048), Not null		Attraction. AttractionPhoto.Uri
New	CityId	Numeric (4)		Null

Indexes

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.

To sum up what we have seen:

When a transaction has more than one level, each level will have a table associated with it.

The primary key of the subordinate-level tables ends up being the combination of the identifying attributes of the level itself and those of the parent levels. In this case, CityId plus the City level identifier, which is CountryId.

This CountryId identifier in City silently converts it into a foreign key of the Country table and this is what creates the relationship between the two tables.
However, because it is not only a foreign key but also part of the primary key in the city table, cities are a weak entity in relation to countries, since in order to identify a city we NEED to identify the country to which it belongs.

GeneXus™

The power of doing.

Videos	training.genexus.com
Documentation	wiki.genexus.com
Certifications	training.genexus.com/certifications