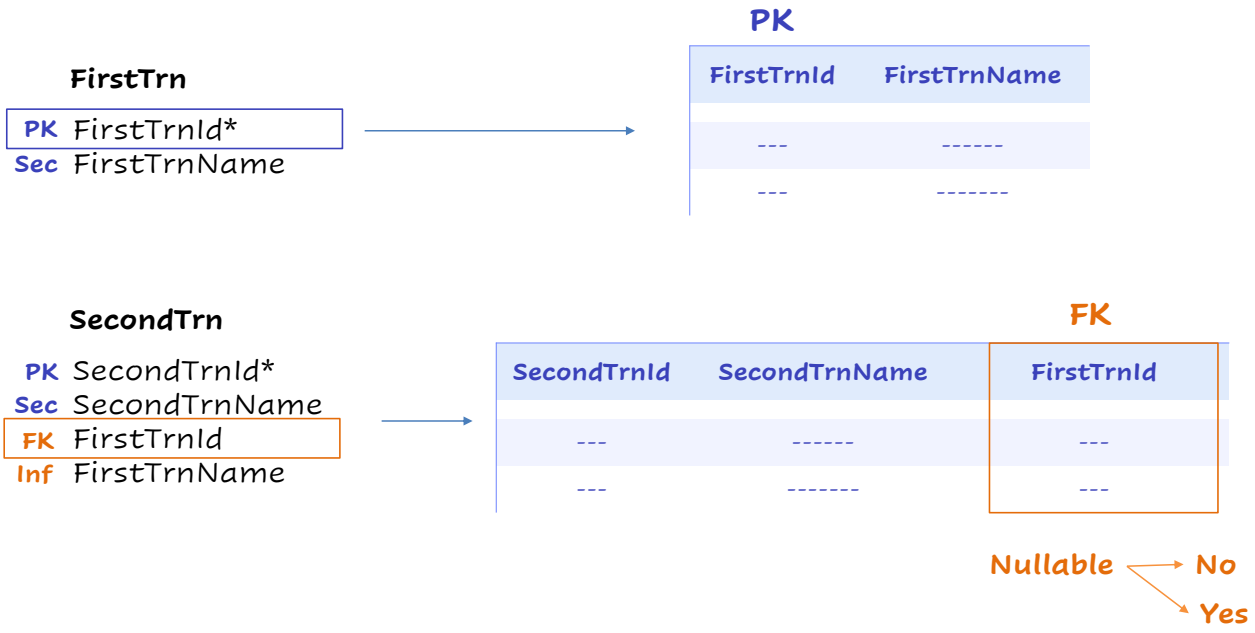
Null values allowed in a compound foreign key

GeneXus™

We have already seen that the Nullable property allows indicating that the value of an attribute that is a foreign key can be left unspecified –unassigned– so that in this case GeneXus does not trigger the corresponding referential integrity checks.

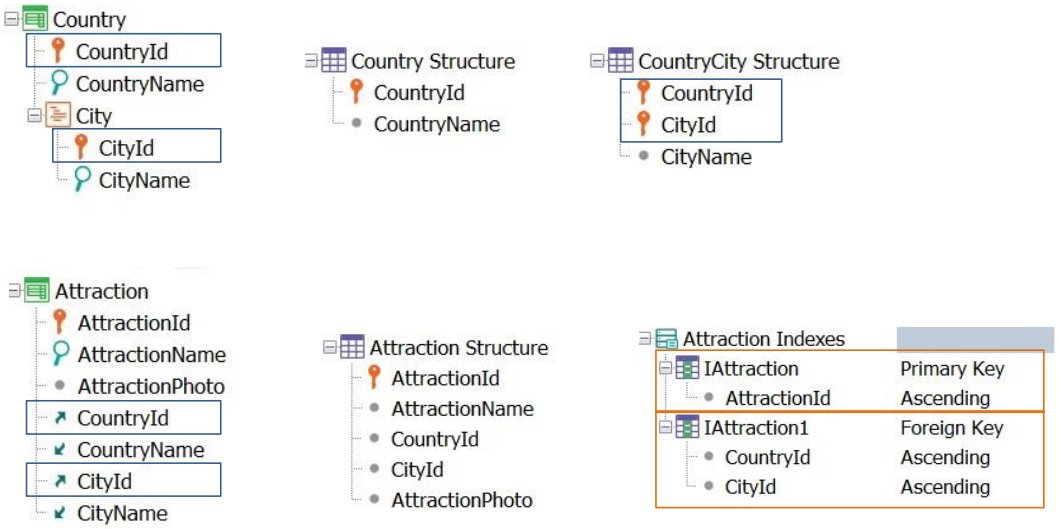
On the other hand, we know that if an attribute does **not** accept nulls, then a valid value must **always** be indicated.

Nullable property



Within a knowledge base, the relationship between entities is defined by primary keys, which can be foreign keys when they are included in other entities. Whether a foreign key accepts nulls or not lets GeneXus know how strong the references are between those entities.

Example



For example, consider the Country and Attraction transactions as shown below:

Based on this transaction design, GeneXus will create the COUNTRY and COUNTRYCITY tables we are looking at:
We know that in the case of the Attraction transaction, the attribute pair CountryId, CityId make up a compound foreign key.
As for the structure of the ATTRACTION table, associated with the Attraction transaction, it is the one we are looking at. In addition, GeneXus will also automatically create the following indexes:

The primary index by AttractionId and the foreign index by the attribute pair CountryId, CityId.

Example

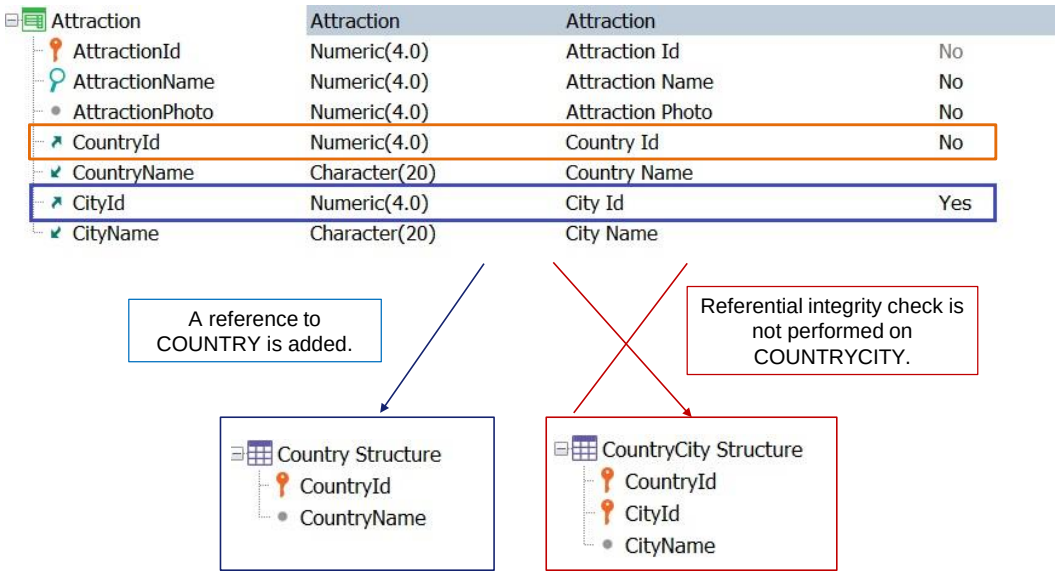
Attraction	Attraction	Attraction	
AttractionId	Numeric(4.0)	Attraction Id	No
AttractionName	Numeric(4.0)	Attraction Name	No
AttractionPhoto	Numeric(4.0)	Attraction Photo	No
CountryId	Numeric(4.0)	Country Id	No
CountryName	Character(20)	Country Name	
CityId	Numeric(4.0)	City Id	No
CityName	Character(20)	City Name	

In the structure of the Attraction transaction, we can see that by default both attributes, CountryId and CityId, have the Nullable property set to No.

This means that the values indicated in them must **always** be valid; i.e. the value pair must exist in the COUNTRYCITY table as a primary key.

Note that the impact analysis report shows that the corresponding reference to the COUNTRYCITY table will be created.

Example



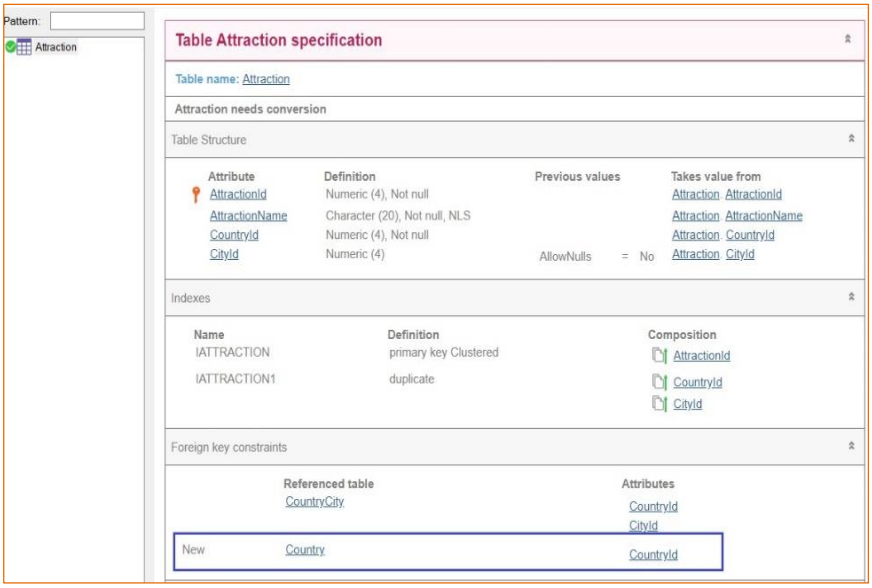
What will happen if, for example, in the structure of the Attraction transaction we indicate that the CityId attribute accepts nulls?

When a foreign key is composed of more than one attribute, and the null value is allowed in any of its parts, then new references must be defined if the rest of the attributes are also a foreign key, as it happens in this case, since CountryId is also a foreign key.

So, if in the Attraction transaction we indicate that the CityId attribute accepts nulls, it will imply that GeneXus will not make the corresponding referential integrity checks on the COUNTRYCITY table when the value of this attribute is not indicated.

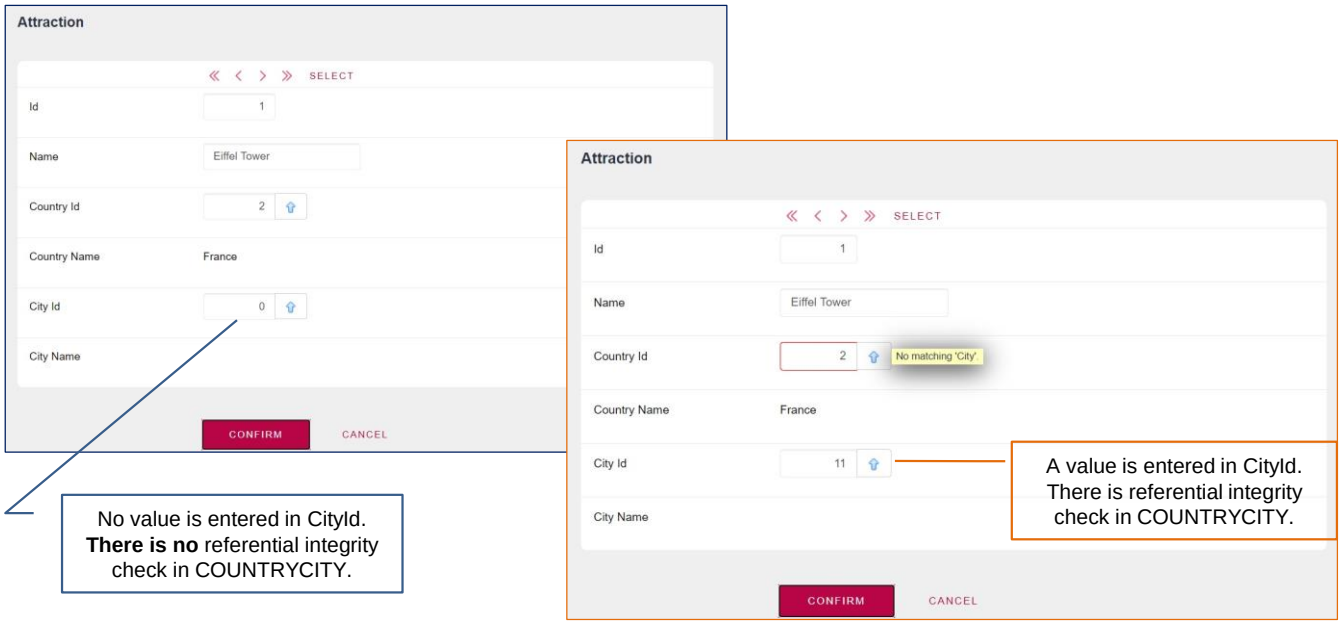
GeneXus will then create a reference to the COUNTRY table to avoid indicating a CountryId value that doesn't exist in that table as a primary key, when the CityId value is not entered.

Example



If we look at the impact analysis report, we see that GeneXus added a new reference over the COUNTRY table:

At runtime:



If we enter the Eiffel Tower, but indicating only a valid value in CountryId and leaving CityId blank, GeneXus only checked that the CountryId value was valid; that is, that it existed as a primary key in the COUNTRY table.

When an invalid value is entered in CityId, GeneXus displays the corresponding error message.
The reason for this is that when a CityId value is entered, GeneXus does make the corresponding controls on the COUNTRYCITY table; therefore, the pair indicated in CountryId and CityId must exist as a primary key in that table.

GeneXus™

training.genexus.com
wiki.genexus.com
training.genexus.com/certifications