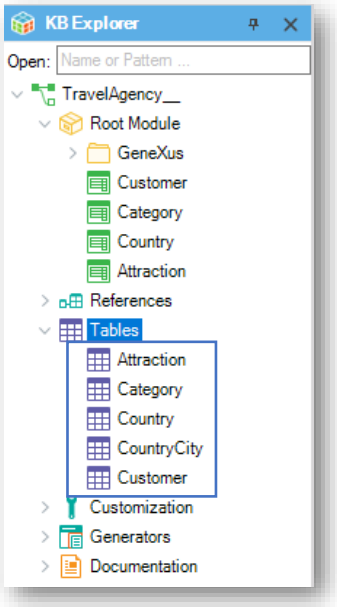
Identifying available attributes

Base table and Extended table

GeneXus™

Tables

View / Tables



We will now explain the definition of **base table** and **extended table**.

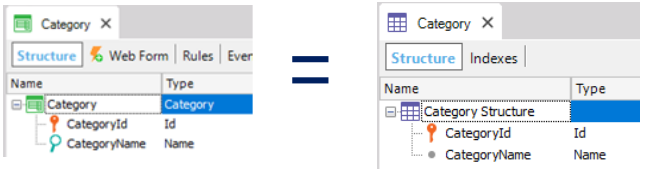
These are two very simple concepts which should be clear because they are regularly applied throughout the overall use of the tool.

Let's take a look at the Tables node.

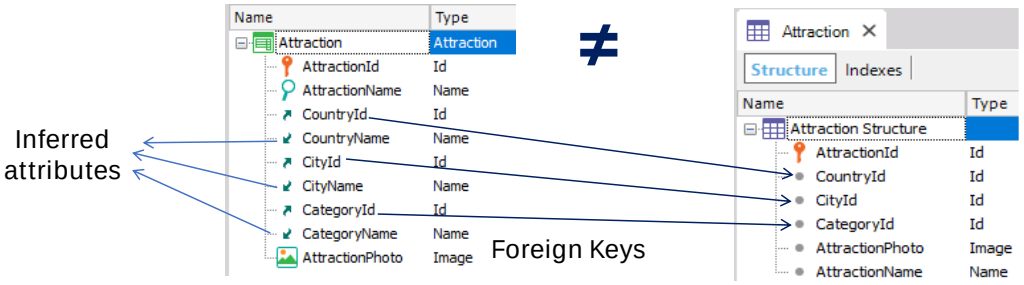
Under this node we can see **which physical tables** GeneXus has determined to be created in the database, **based on the structures of the transactions we defined**.

Transaction - Table

Category: The transaction and the table have the same attributes.



Attraction: The transaction has more attributes than the table.



If, for example, we open the Category table, we will see that the physical table contains exactly the same attributes defined in the structure of the Category transaction.

However, if we open the composition of the Attraction table **we will see that it contains fewer attributes** than those referred in the Attraction transaction.

This is because **there are several foreign keys** in the Attraction transaction and thus, through them there are values of attributes **that we obtain** when the application is executed, from the tables where they are located.

Transaction - Table

Compounded key table

The **table** called **CountryCity** was created by GeneXus in the database, based on what was defined in the second level of the **Country transaction.**

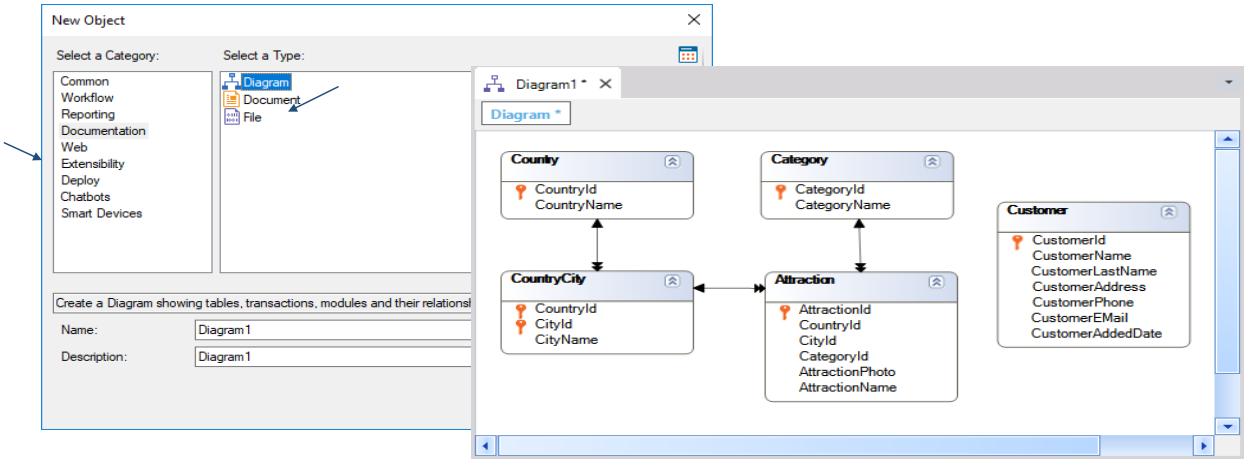
We can also see that under the **Tables** node there is a table called **CountryCity.**

This physical table was created by GeneXus in the database, **upon the definition of the 2nd level we did in the Country transaction.**

As we saw before, when we define a nested level to another in a transaction, a compound key table is created which in this case is the CountryCity table.

Table diagram

File / New Object



So... we have seen the Tables node under which we can see the physical tables created in the database and their composition.

We will now see that we can create a **table diagram**, to view them on a scheme showing, **in addition to their composition, how they relate with one another**.

For this we select **File / New / Object**

We choose to create an object of the **Diagram** type and leave the name proposed by default: **Diagram1**

We select Create, and drag all the tables from the Tables node to the diagram...

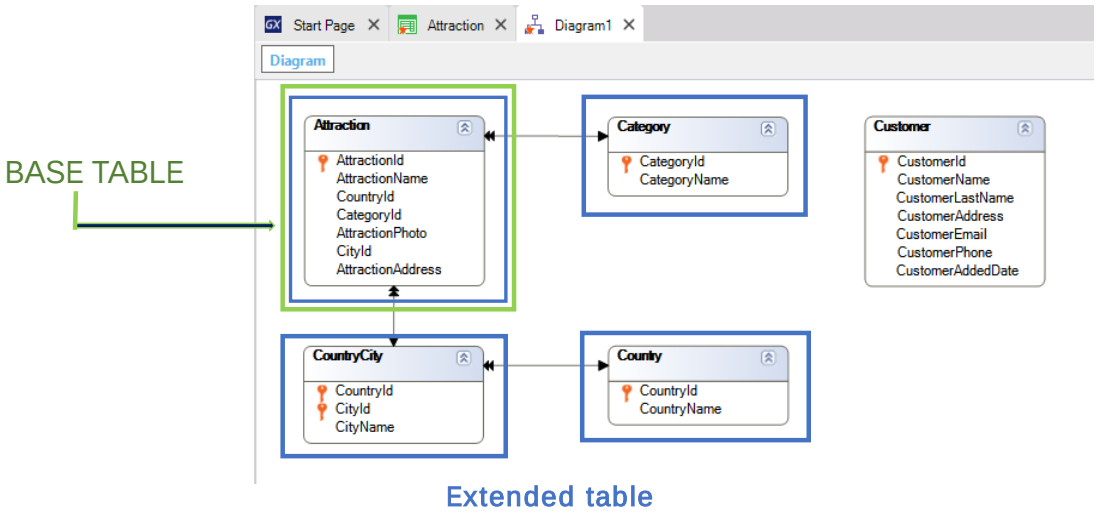
Now we click the right button on the mouse and select "Arrange nodes" for distributing the tables on the diagram and for making the arrows representing their inter-relations clearly visible.

Considering this table diagram we will explain what we call **base table** and what we call **extended table** in GeneXus.

We call "base table" any table from the database where we are positioned at a certain time, for example, for deploying or modifying its data.

Example

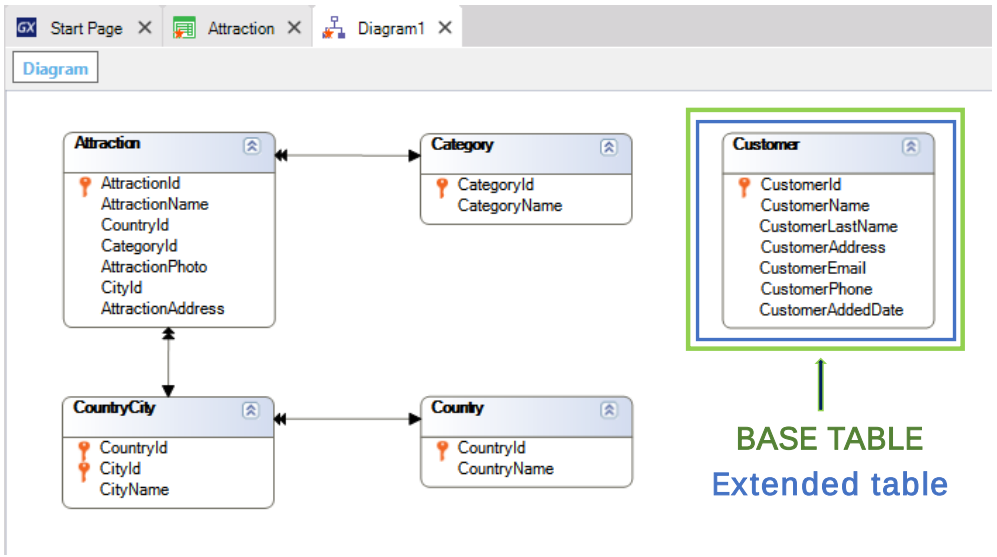
- Extended table of the “Attraction base table”:



It could be this one...

Example

- Extended table of the “Customer base table”:

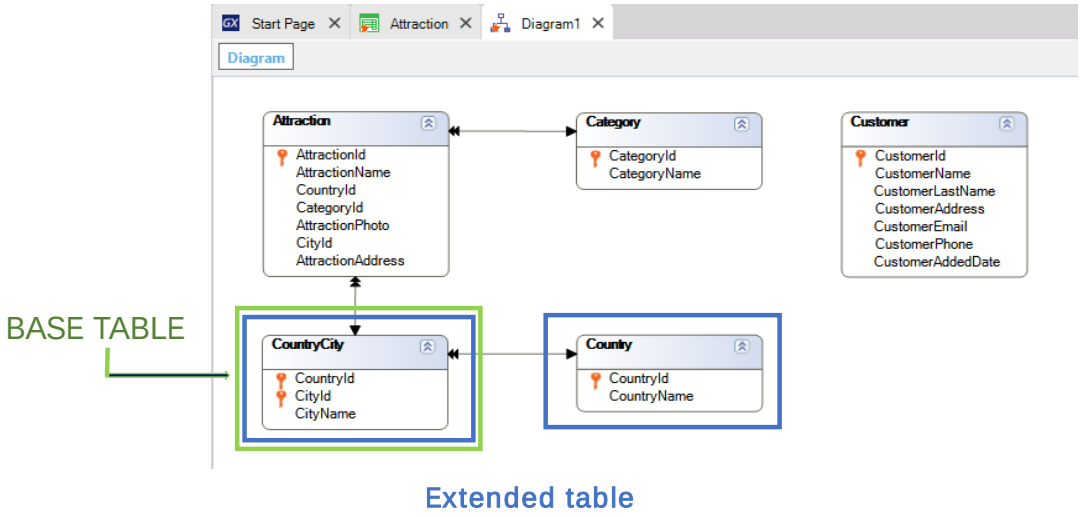


Or this one...

Or any of the tables in the diagram.

Example

- Extended table of the "CountryCity base table":



When we execute a transaction **with a single level**, it will have **one associated base table** that is: a physical table where the insertions, modifications and deletions that we operate – like for example interactively through the transaction screen – take place.

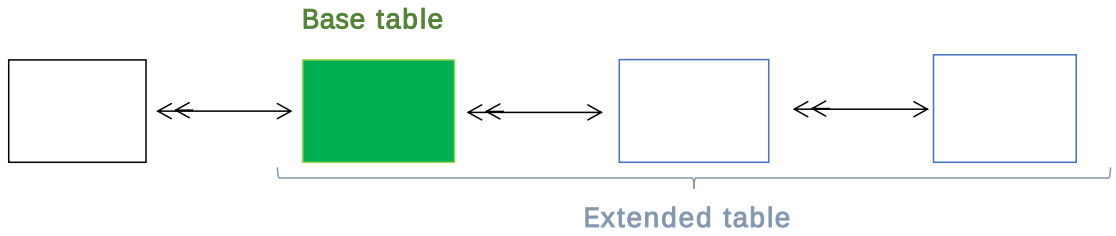
When we work with a transaction with **more than one level**, **each level** will have an associated base table where the insertions, modifications and deletions that we process through that level take place.

Also, when we define queries (a listing, for instance), **we will be navigating a specific base table**.

Fundamental concepts

- **Base table:**
  Any table in the database where we may be working at a given moment.

- **Extended table:**
  For a given table, its extended table is a concept that allows us to consider all the information that we can access from it, using its foreign keys.
  It is the set of attributes of the table itself **+** all the attributes of the tables with which it has an N to 1 relation, either directly or indirectly.



The **base table** is then any **physical table** in the database where we are **positioned** and working at a certain time.

Let's now see the **extended table** concept. This concept is meant to **simplify the way to find out which are the tables we have access to** when we are positioned in a specific **base table.**

Intuitively, we have already accessed the **extended table of a certain base table.**

For example, the "Attraction" transaction's base table, or associated physical table, is the ATTRACTION table.

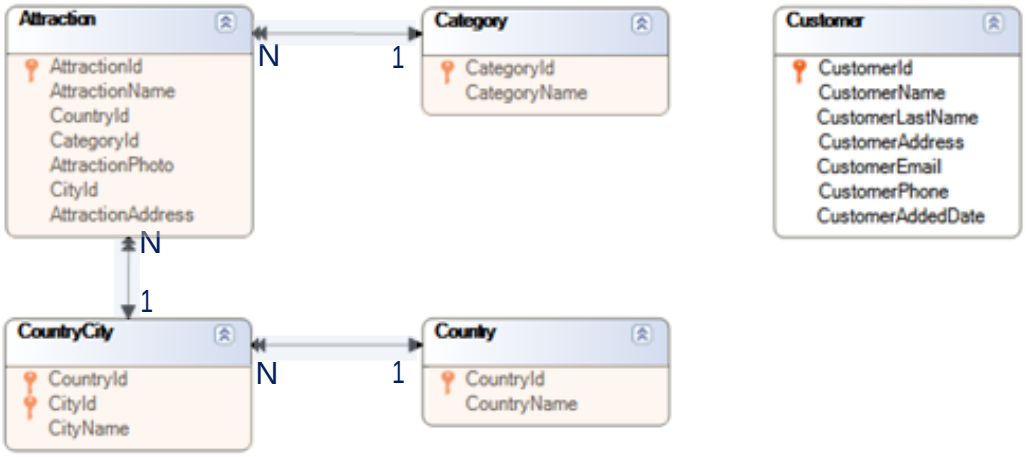
And as we saw before, because CategoryId is a foreign key attribute in the "Attraction" transaction, we can then reference the CategoryName attribute.

CategoryName **does not belong to the base table associated with the "Attraction" transaction, but we can obtain its value, since we have it in the extended table.**

In general, and based on a foreign key – like CategoryId, CountryId and CityId in "Attraction" – we can always obtain the values of its secondary attributes from the tables where they are located.

And if those tables **also include another or other foreign keys**, then the chain goes on and we can obtain their related data as well.

Extended table



Having intuitively comprehended the concept, we can formally state that:

Upon any given table that we consider base table at a certain time, its **extended table** is the total of **all of the attributes of the base table itself, plus all the attributes of the tables directly or indirectly related to it through an N-1 relation**.

If we now take another look at the table diagram we had created, **we will see another way of determining the extended table of a specific base table**.

If we take **ATTRACTION** as base table, its extended table comprises the ATTRACTION table itself and, following the double arrow, we will see that CATEGORY is also included.

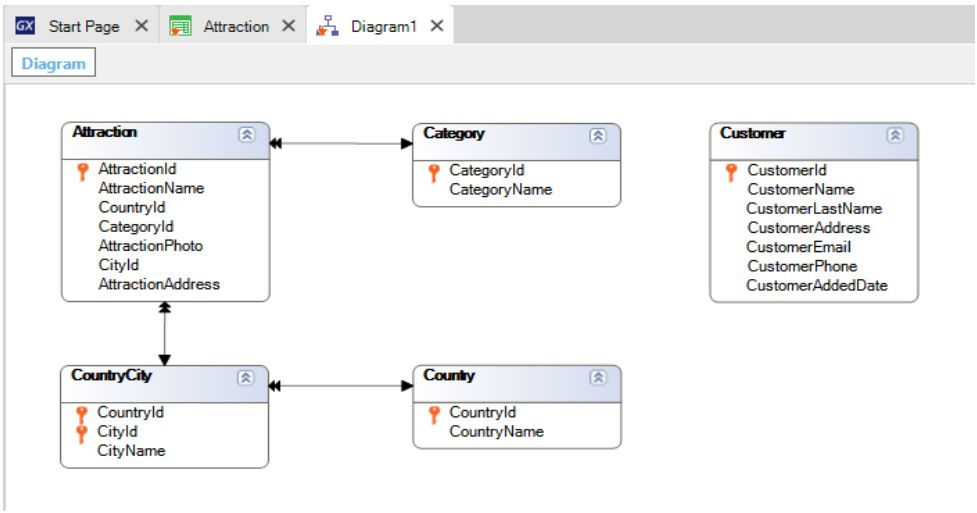
On the other side there is a double arrow indicating that COUNTRYCITY is also included... and from COUNTRYCITY there is a double arrow, meaning that COUNTRY also belongs to the extended table of the ATTRACTION base table.

Thus, upon a table diagram, to obtain the extended table of a specific base table, we can follow the arrows with double pointer coming from it and with single pointer at the other end, and the table we arrive at will be also be part of its extended table... and all the tables to which we may arrive by following double arrows will be part of the extended table.

In sum, we will be navigating in the direction of the N to 1 relations.

Example

- Extended table of the "Country base table":



If we go back to the diagram we can see that if **COUNTRYCITY is the base table on which we are located at a certain time**, its extended table includes itself and the COUNTRY table only.

And given the COUNTRY base table, we can see that its extended table comprises only itself, because it does not have any double arrow we could follow.

GeneXus™

training.genexus.com
wiki.genexus.com