

GX

GeneXus by Globant

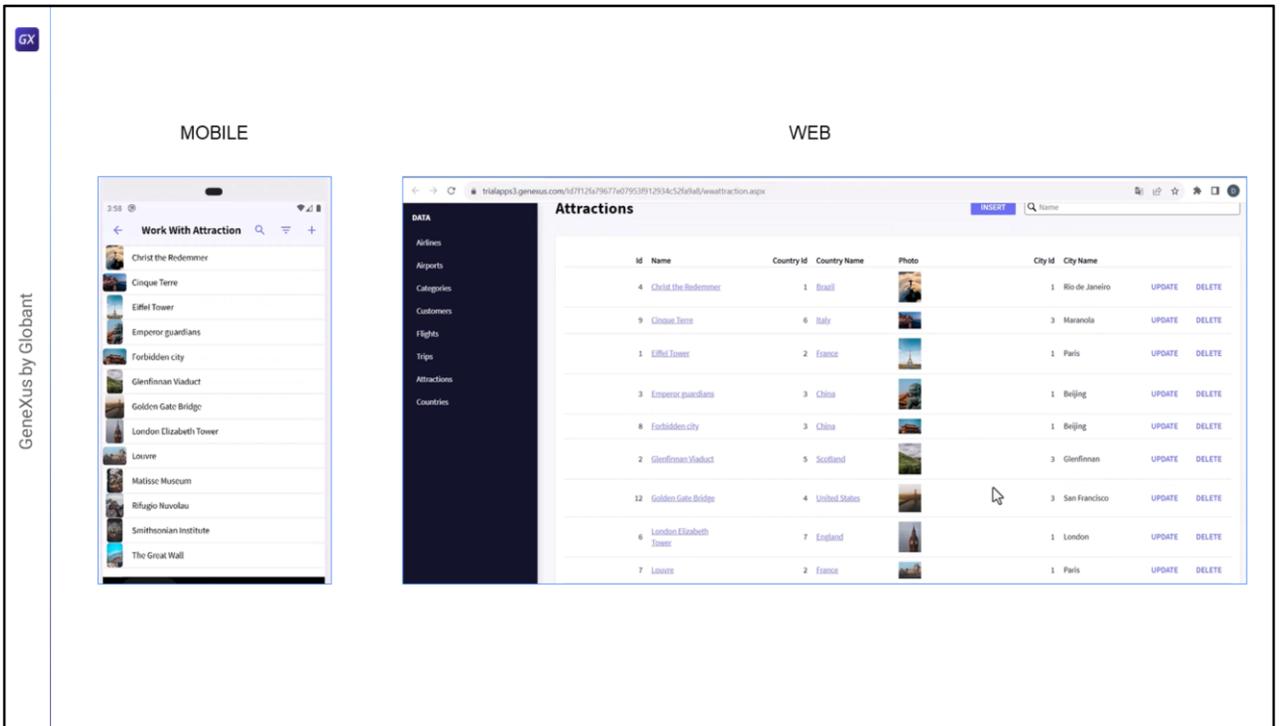
GeneXus[™]
by Globant

training.genexus.com

Offline Database Generation



Diego Marranghello



In this video, we will see how to convert an Online application into an Offline application.

First we are going to see how our Travel Agency application, which is currently an Online application, works.

Let's access the mobile application and the Web application at the same time, with both interfaces open to see the interaction.

To show a clear and simple example, we apply both Work With patterns to the Attraction transaction: for web and for mobile.

First we go to the web, access an attraction, for example the Louvre, edit its name and save.

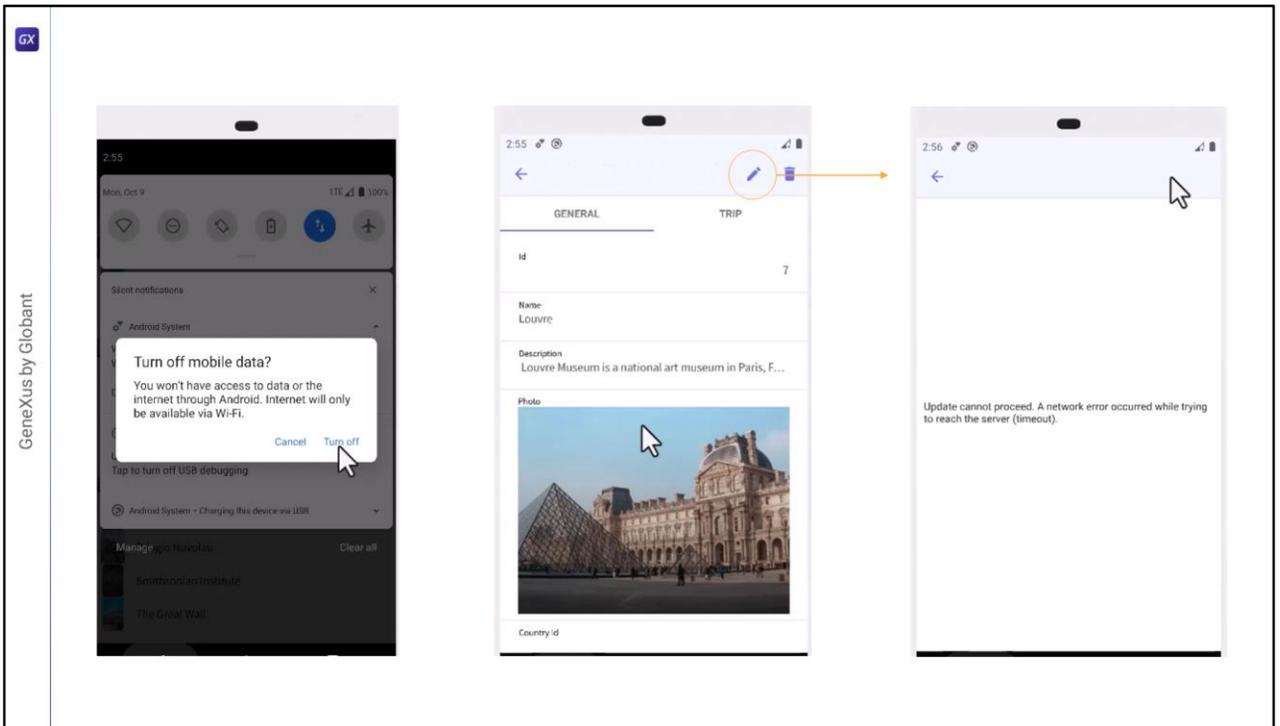
Now we go to the mobile application, refresh, and see that the change has been made automatically.

What happens if we edit it from the device?

We go back to edit the name and leave it as it was in the beginning and save.

We refresh in the web panel and automatically see the change.

This is because the application works in a connected way.



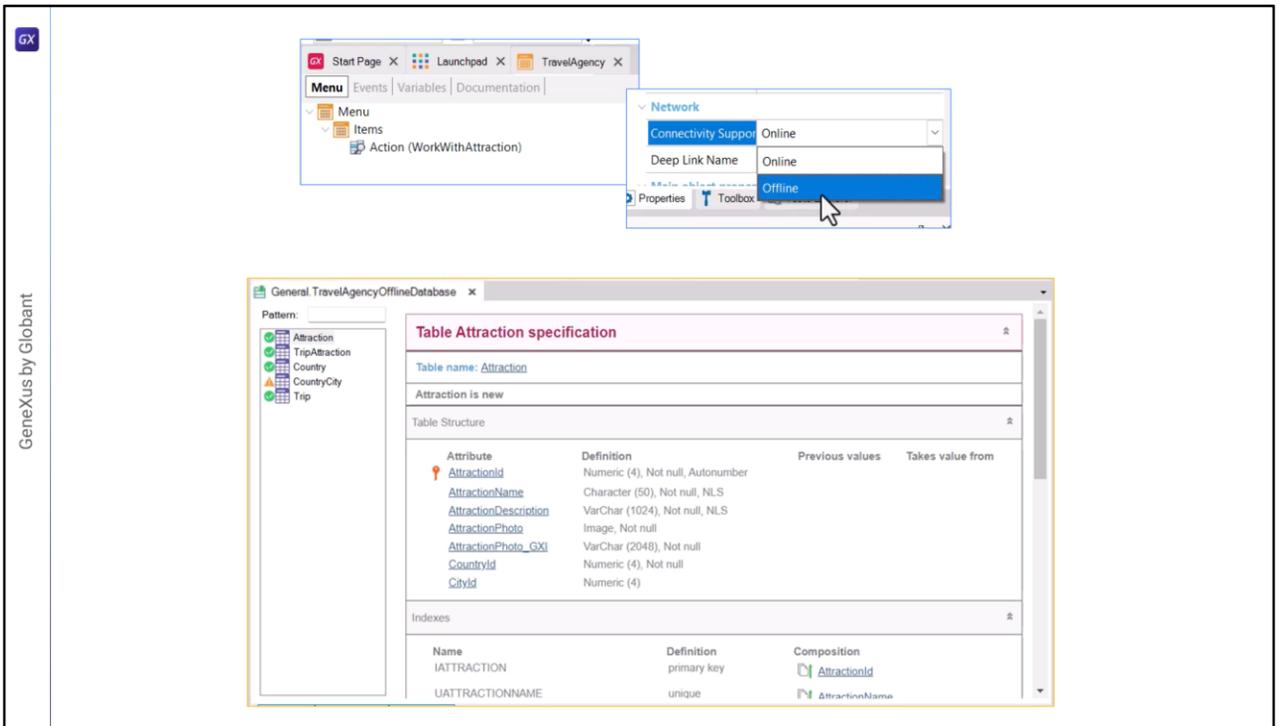
What happens if I now disconnect the application?
Let's simulate this by turning off the Wifi and data connection. Now the emulator has no internet connection, so it cannot access the REST services layer.

If we try to access the attraction, we will be able to do so because this data is cached from the previous access, but if we try to edit it we get an error.

If we try to enter an attraction that we haven't accessed before we can't do it because we have no connection and don't have its data stored.

When we are offline in an Online application, we only have access to data that is cached in the application, but we cannot access new screens.

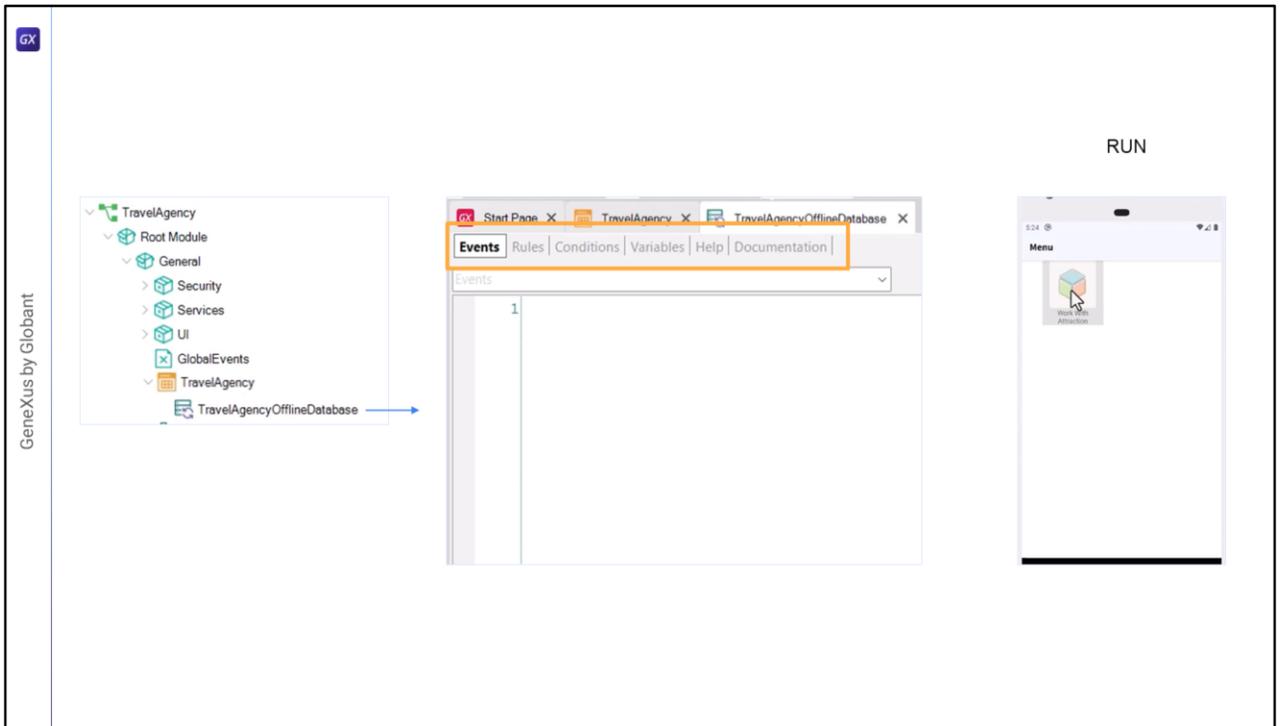
OK, now we go to GeneXus to convert the app into an Offline app.



In our application we have a Menu object named TravelAgency, which is the main object of the application. And inside we only have, at the moment, the attraction work with.

When going to the properties of the menu object, we see one named Connectivity Support which by default has the value Online, and we will change it to Offline. We save the changes and build the app.

One of the things we will see next is an impact analysis of the Offline database; these will be the tables that will be taken and created in the device.



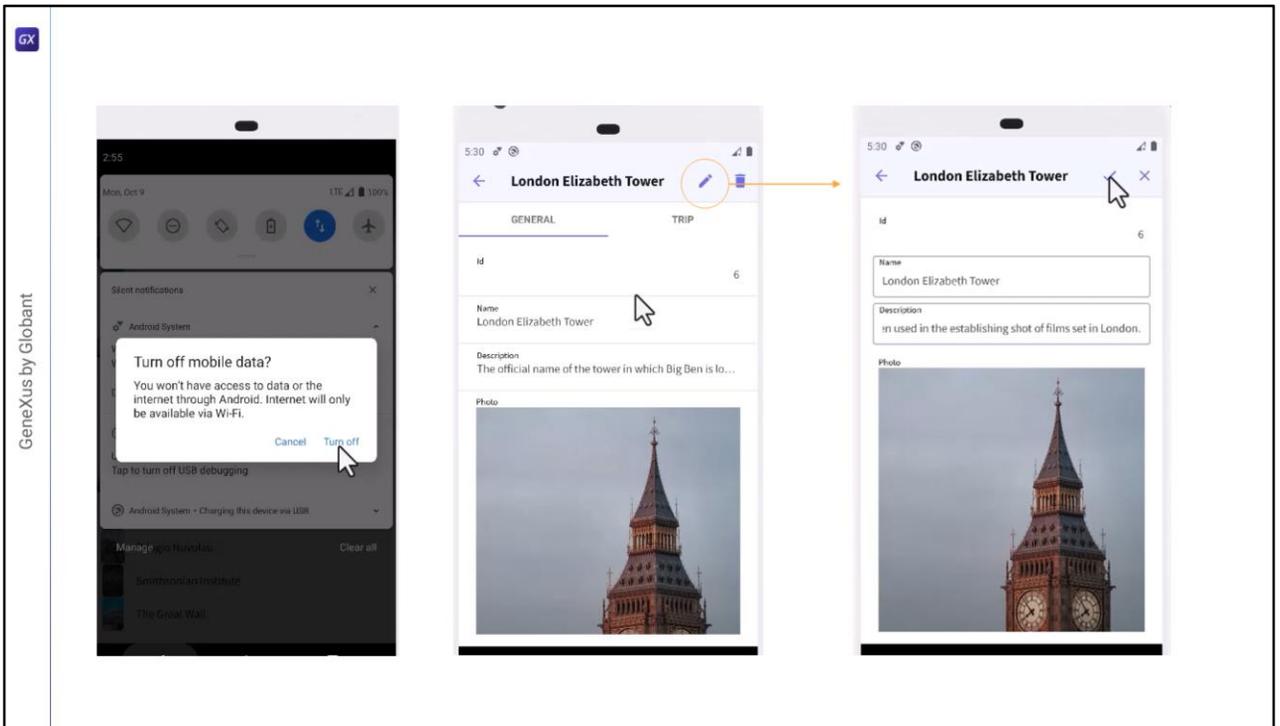
Let's see that now we will have a new object, `TravelAgencyOfflineDatabase` that will be associated with the `TravelAgency` menu object.

This object will allow us to configure certain aspects related to the Offline database –the one that will be generated on the device– such as granularity. Then we will study these properties in more detail.

In addition, this object has Events, Rules, Conditions and variables. Later on, we'll see how they are used, but for now we leave the default settings.

Note that now when the mobile application is started, we can see in the status bar that the synchronization is running.

Once the process is finished, we will be able to use the application as usual.

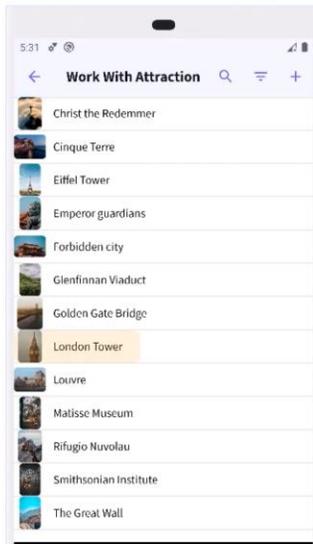


We enter to see the attractions, so we will see the list.
We disconnect the device.

Now we try to access an attraction and don't see the error that we got before when we were connected and lost the connection. This is because now the data is on the device.

I can even edit the records.

For the user there is no difference compared to being connected.



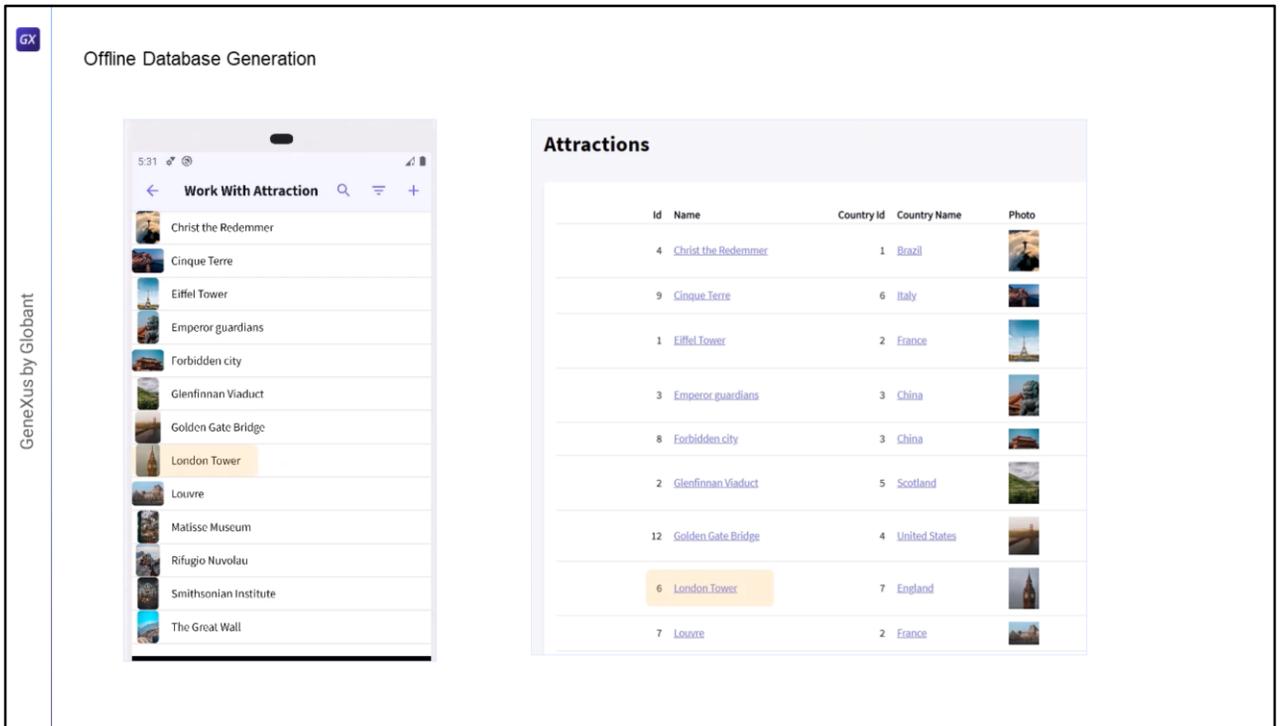
Attractions

Id	Name	Country Id	Country Name	Photo
4	Christ the Redemmer	1	Brazil	
9	Cinque Terre	6	Italy	
1	Eiffel Tower	2	France	
3	Emperor guardians	3	China	
8	Forbidden city	3	China	
2	Glenfinnan Viaduct	5	Scotland	
12	Golden Gate Bridge	4	United States	
6	London Elizabeth Tower	7	England	
7	Louvre	2	France	

Let's go back to editing the name of an attraction and save.

Note that it is not a problem to be offline. The data is always recorded on the device.

And if we go to the web interface and refresh we won't see that change. That's because the device is offline.



Let's reconnect the emulator to the network.

To force the synchronization I will close and open the application again. This happens because we haven't configured anything of the OfflineDatabase object yet.

When opening it again it runs the synchronization.

Now if we go to the web and update we see the change.

This is the end of this basic Demo, where we saw how simple and easy it is to create an Offline application with GeneXus.

In addition, we have seen the process of synchronizing the data between the device working Offline and the web server, and the CRUD operations that will always be performed on the local database.

Offline Database Generation

The screenshot shows the GeneXus interface for Offline Database Generation. On the left, a 'Network' panel is visible with 'Connectivity Support' and 'Offline' options. The main workspace shows two objects: 'TravelAgency' and 'TravelAgencyOfflineDatabase'. An arrow points from 'TravelAgency' to 'TravelAgencyOfflineDatabase'. On the right, a 'Table Attraction specification' window is open, displaying the table structure and indexes for the 'Attraction' table.

Attribute	Definition	Previous values	Takes value from
AttractionId	Numeric (4), Not null, Autonumber		
AttractionName	Character (50), Not null, NLS		
AttractionDescription	VarChar (1024), Not null, NLS		
AttractionPhoto	Image, Not null		
AttractionPhoto_GX	VarChar (2048), Not null		
CountryId	Numeric (4), Not null		
CityId	Numeric (4)		

Name	Definition	Composition
IATTRACTION	primary key	
UATTRACTIONNAME	unique	

0 Errors 1 Warnings 4 Success

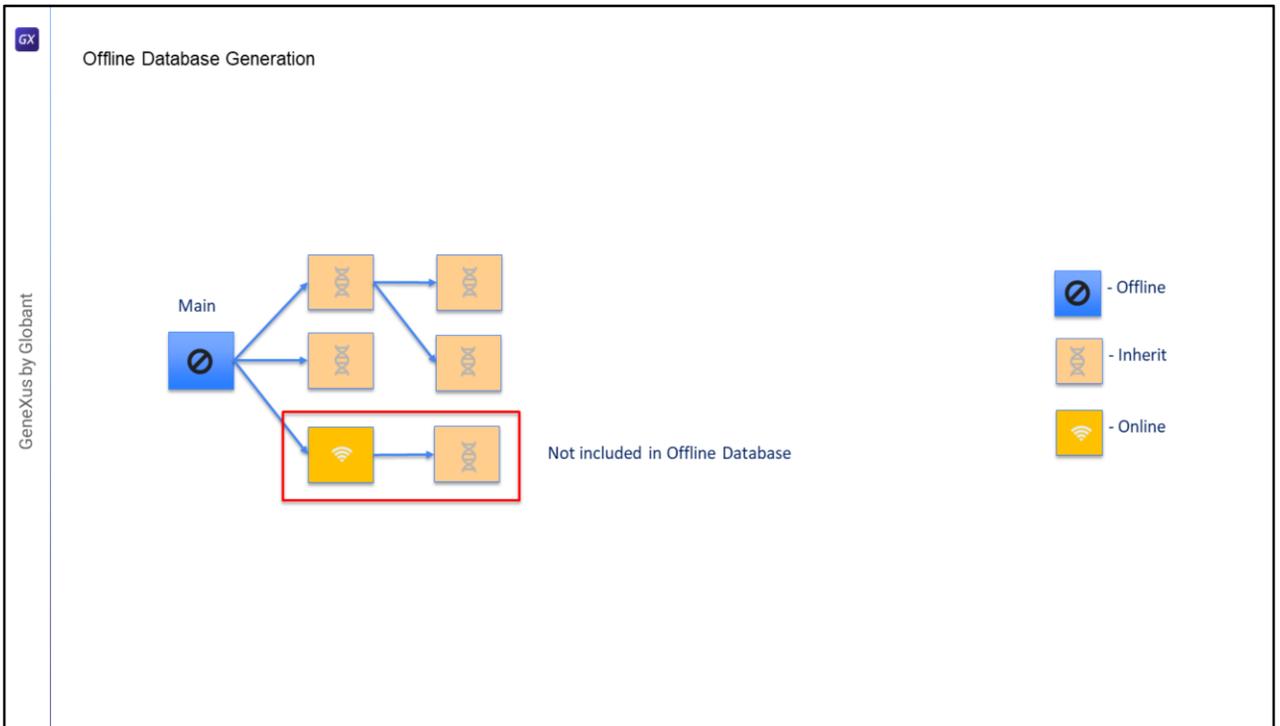
As we have just seen, when we indicated that the main object was Offline and executed the Build operation, an OfflineDatabase object associated with the Main object was automatically generated.

We've also seen an impact analysis, which is similar to when we select to create the database. However, the tables displayed here are the tables that the GeneXus analysis has determined that it will take to the SQLite database on the device.

When the OfflineDatabase object is created, the programs to create this local database are also created in the device's native language.

This object is responsible for determining when the synchronization is run, which tables will be created in the local database, and which data is sent to them when they are synchronized with the server tables.

Also, the OfflineDatabase object is where the properties we've seen for Send and Receive are configured.



What tables are effectively sent when an application is set to offline?

Suppose we have this call tree from the Main object. We can have Objects with the Connectivity Support property set to Offline, such as the Main object; there may be others with the default value, which is Inherit –they inherit the type according to the object calling them– and others may be Online.

If we have this invocation tree, the tables that will be sent to the device are all those of the offline objects, and those that inherit the Offline connectivity of the caller object.

In addition, referential integrity and the attributes mentioned both in panels and in prompts will be taken into account.

This object that is Online and calls this other one that inherits the type of connection will not be included, they will be online.

In the following videos, we'll see more details about the synchronization and properties of the OfflineDatabase object

GX

GeneXus by Globant

GeneXus[™]
by Globant

training.genexus.com