Acceso a bases de datos externas

DBRET y DataViews

GeneXus™

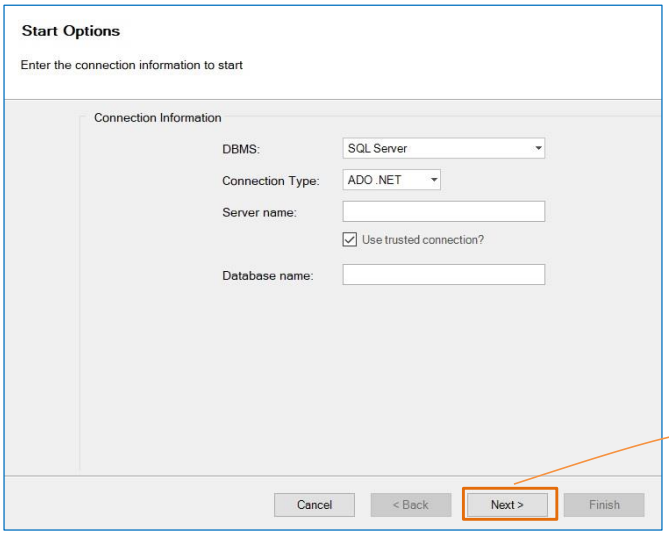
Muchas veces necesitamos acceder **a bases de datos externas** desde nuestras aplicaciones GeneXus.

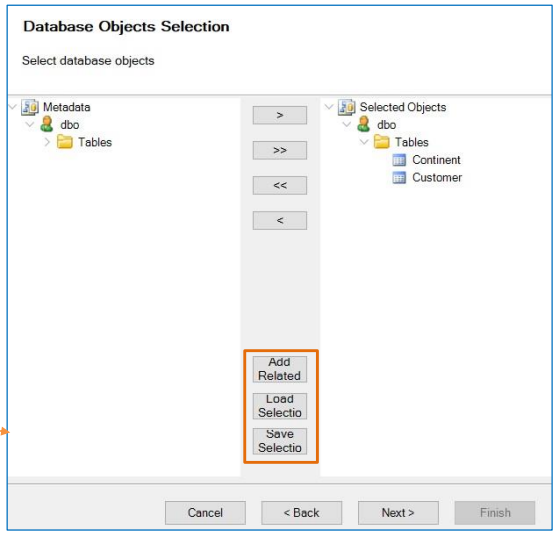
Acceso a bases de datos externas

Por ejemplo, podemos necesitar cargar datos **en tablas de nuestra base de datos asociada a la base de conocimiento** con datos que se encuentran en cierta base de datos externa por ejemplo para realizar una carga inicial... y luego de ello podemos no necesitar más seguir conectados a esa base de datos externa.

O podemos también necesitar conectarnos y mantener conexión **siempre** con determinada tabla o ciertas tablas de una o varias bases de datos externas... y no sólo para leerlas, sino también para ingresar y modificar datos en ellas.
s.
Veamos cómo es posible lograr todo esto en GeneXus.

Database Reverse Engineering Tool (DBRET) - Wizard





En nuestra base de conocimiento Travel Agency observemos en el menú: **Tools, la opción que dice Database Reverse Engineering .**

Seleccionando esta opción, accedemos a un wizard **que nos permite aplicar ingeniería inversa sobre bases de datos externas. Esto significa que para una base de datos existente que contiene tablas, relaciones, índices, etc., se crearán en GeneXus todos los objetos y configuraciones necesarias para poder acceder fácilmente.**

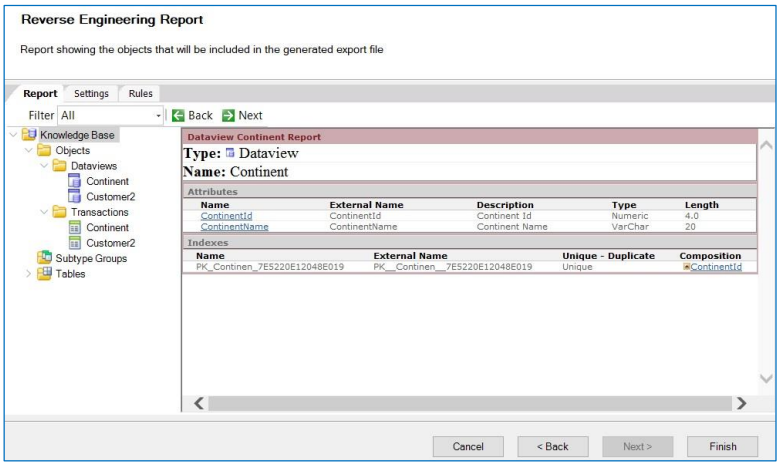
En esta primera página del wizard, lo que debemos hacer es indicar la información de conexión a la base de datos a la cual queremos acceder.
Dado que esta base de datos a la cual queremos acceder es justamente SQL Server, aquí en la opción "DBMS", dejamos "SQL Server".
La opción "Conection Type" también queda como está, e indicamos luego el nombre del servidor y el nombre de la base de datos a la cual nos queremos conectar. Presionamos ahora el botón Next... y pasamos a la segunda página del wizard..

Aquí se nos presentan las tablas que están dentro de la base de datos a la cual nos estamos conectamos. Debemos asegurarnos de dejar en la ventana de la derecha solamente aquellas tablas a las cuales deseamos aplicar ingeniería inversa, para accederlas desde nuestra base de conocimiento.
Nuestra intención es acceder a las tablas Customer y Continent, así que deben quedar en la ventana de la derecha.
Bien, pero antes de presionar el botón Next, observemos estos otros botones:

El botón "Add Related", nos brinda la posibilidad de agregar automáticamente todas las tablas que estén en relación N-1 con aquellas tablas que hayamos seleccionado.

Por su parte el botón "Save Selection", nos permite grabar la selección de tablas que hayamos realizado, para que luego podamos cargar esta misma selección rápidamente en otra ejecución, haciendo uso del botón "Load Selection".

Database Reverse Engineering Tool (DBRET) - Wizard



Ahora sí presionamos el botón Next
En este momento comienza el proceso de ingeniería inversa... es decir, se va a evaluar la composición y todo sobre las tablas que hemos seleccionado y se van a definir los objetos GeneXus, que sean necesarios.

Tiene 3 tabs:

- Report
- Settings
- Rules

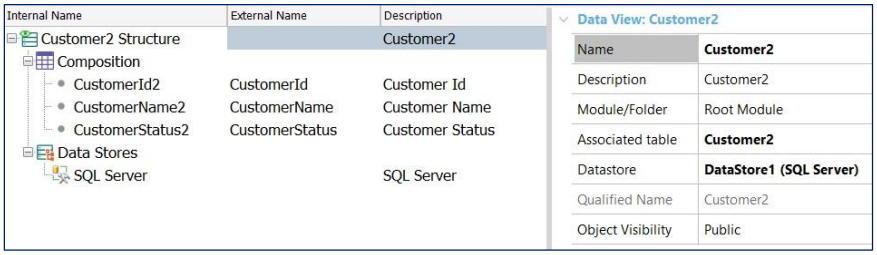
Empecemos por ver el tab principal: Report. Este reporte que estamos viendo, nos informa los objetos creados durante el proceso de ingeniería inversa.
Observemos que bajo el folder **Objects,** tenemos un folder **Dataviews** y otro folder **Transaction** y debajo de cada uno de ellos, se ve un objeto de nombre: Continent y otro de nombre: Customer2.

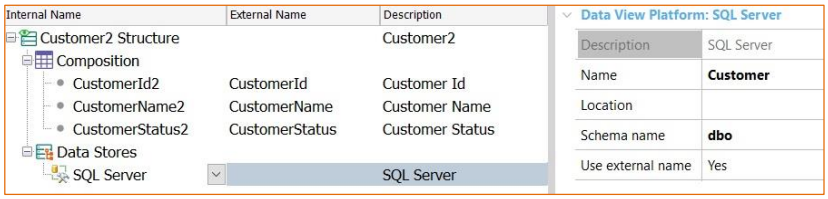
La primera pregunta que nos surge es la siguiente: ¿por qué el nombre "Continent" se respetó exactamente igual a como se llama la tabla externa y en cambio en el caso de "Customer" que es el nombre de la tabla externa, se nombró "Customer2" a los objetos de la KB?
El motivo es que en nuestra base de conocimiento ya había una transacción de nombre "Customer" y su correspondiente tabla de igual nombre. Así que otra transacción de nombre "Customer" no se podía definir y por tal motivo se nombró "Customer2" al igual que a los demás objetos creados para acceder a la tabla externa Customer.

Bien, **¿y qué será un objeto Data View?** Hasta ahora no lo conocemos.
Un objeto Data View permite definir y configurar información de una tabla externa. O dicho de otro modo, **por cada tabla externa a la cual necesitamos acceder, deberá haberse definido un Data View en la base de conocimiento.**

Objeto DataView

Permite definer y configurar información de una table externa. Por cada tabla externa a la cual se necesita acceder, deberá existir un DataView en la base de conocimiento.





Como estamos en la mitad del proceso de ingeniería inversa, para no cancelarlo vamos a observar y comentar en esta imagen cómo quedará definido el Data View "Customer2"

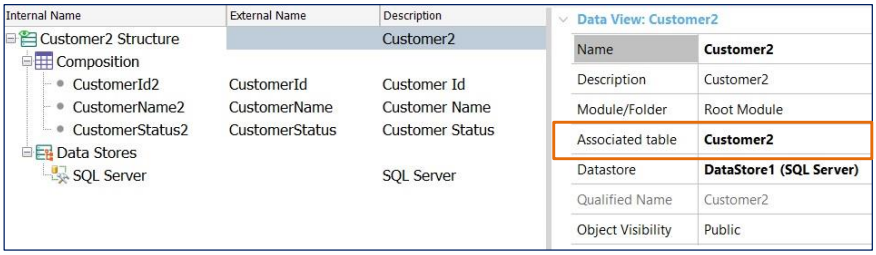
Lo primero que vemos a simple vista, bajo Composition es una lista de nombres de atributos, y observemos que para cada atributo hay un External Name **(que es el nombre que tiene el campo físico en la tabla externa)** y un Internal Name **(que es el nombre dado a nivel de la base de conocimiento)**, es decir, **el Internal Name es el nombre de atributo que está presente en la transacción que se crea automáticamente para acceder de forma interactiva a la tabla externa**, y es el nombre que incluiremos en grids, eventos, etc., en todos los objetos GeneXus.

Así que bajo Composition, está el **mapeo** de nombres de campos externos de la tabla y sus correspondientes nombres de atributos internos con la nomenclatura que usamos en GeneXus.
Nosotros en la base de conocimiento siempre vamos a referirnos a los nombres internos, y GeneXus al momento de generar el código en el lenguaje que corresponda, hará referencia a los nombres de campos físicos, considerando para eso la correspondencia de nombres definida en el Data View.
¿Y dónde está el nombre de la tabla externa para que GeneXus pueda hacer referencia a la tabla cuando genera el código?

Aquí vemos que la plataforma externa es SQL Server y al observar sus propiedades relacionadas, vemos la propiedad **Name** que tiene el nombre de la tabla externa.

Objeto DataView

Associated table:: Tiene por valor el nombre de la transacción asociada al DataView.



¿Y dónde se encuentra la información **de la base de datos externa en la cual se encuentra esta tabla**?

Si volvemos a observar las propiedades del nodo principal del Data View, podemos observar la propiedad **Data Store, que apunta a DataStore1, creado automáticamente en el proceso de ingeniería inversa.** Si bien al final de este video veremos dónde están definidos los Data Stores en una base de conocimiento, por ahora los conceptos que debemos aprender son los siguientes:

Cada Data View tiene la información de una tabla externa y cada Data Store tiene la información de acceso a una Base de datos
Luego para cada Data View, hay que referenciar a cuál Data Store pertenece.

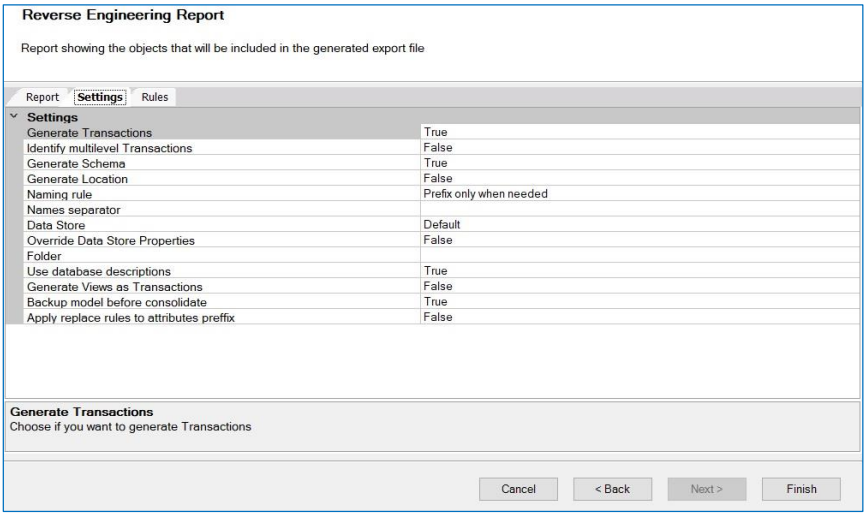
Bien, ahora vamos a observar la propiedad del Data View: "**Associated** table"

Tiene por valor el nombre **de la transacción asociada al Data View**, es decir, la transacción creada también automáticamente con igual nombre que el Data View, para permitir realizar inserciones, modificaciones y eliminaciones en la tabla externa de forma interactiva.

Las transacciones asociadas a Data Views, no provocan la creación de tablas ni reorganizaciones. Si bien son transacciones sin diferencias con las demás, al estar referenciadas en Data Views como tablas asociadas, GeneXus entiende que solamente deberá generar el form y su programación, para permitir interactuar con la tabla externa cuya información se encuentra en el Data View en cuestión.
Vale explicar que las transacciones asociadas a Data Views, pueden tener a lo sumo la misma cantidad de atributos que la tabla externa. Puede tener menos atributos, si algunos campos no se desean acceder, o a lo sumo la misma cantidad que tiene la tabla externa. Pero no más, porque GeneXus no reorganiza tablas externas.

Cabe mencionar también que si en este proceso de ingeniería inversa hubiera ocurrido algún error, se hubiera informado.

Volviendo **al Wizard...**



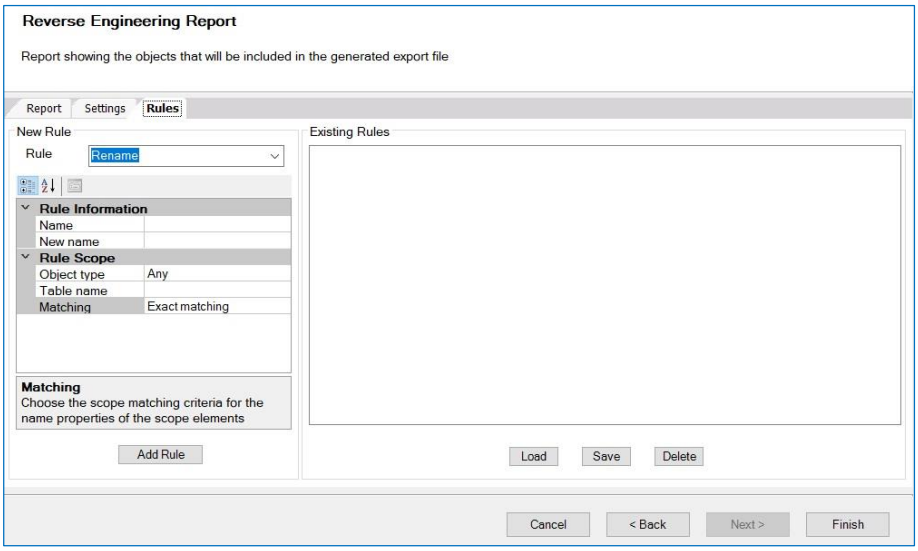
Bien. Ahora sí continuemos viendo los otros tabs que se ofrecen en esta página del wizard.
El tab **Settings**, como su nombre lo indica, permite realizar ciertas configuraciones.

Por ejemplo "**Generate Transactions**" que por defecto tiene el valor True, y tal como aquí se describe, nos permite indicar que se creen transacciones para manipular los datos de las tablas externas de forma interactiva. Si cambiamos el valor a False, entonces no se crean transacciones, sino solamente los Data Views.

La opción "**Identify Multilevel Transactions**" también tal como su nombre lo describe, permite configurar si queremos identificar relaciones de subordinación y que se definan transacciones de más de un nivel siempre que se identifiquen patrones que lo permitan. Su valor por defecto es False, lo que hace que toda tabla a ser importada, defina una sóla transacción de 1 sólo nivel y el Data View relacionado.

La opción "**Generate Schema**" por su parte, tiene el valor True por defecto, y esto significa que la información del schema de la tabla, se almacena en la propiedad Schema del correspondiente Data View.

Volviendo **al Wizard...**



No vamos ahora a continuar viendo cada una de las configuraciones posibles, dado que al posicionarse en ellas, se brinda la ayuda necesaria, así que vamos a continuar con la solapa Rules.

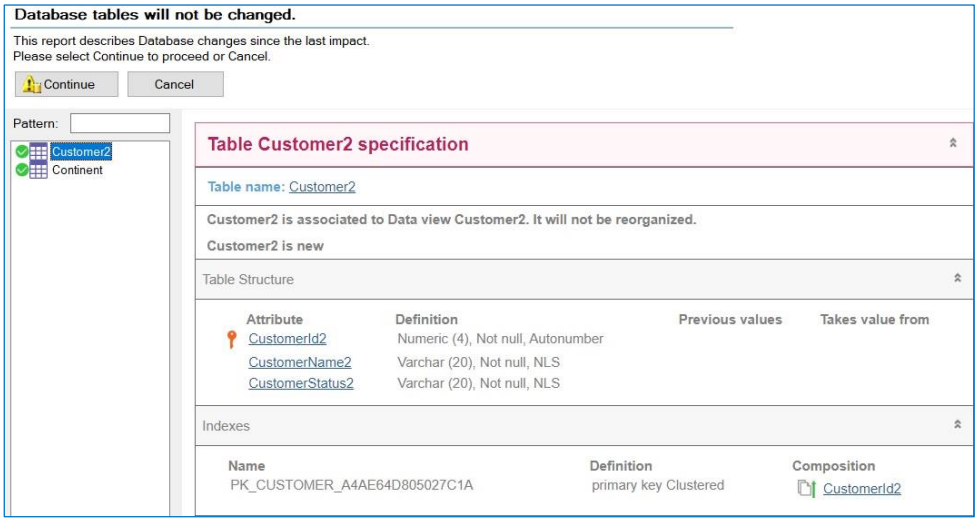
Por el hecho de generar objetos y atributos automáticamente a partir de una base de datos externa, cabe la posibilidad de que se generen algunos conflictos de nombres.

Por este motivo, la solapa Rules ofrece la posibilidad de definir reglas de mapeo de nombres para "resolver" estas ambigüedades.
De modo que esta solapa nos permite definir reglas para poder renombrar atributos, objetos, etc., cambiar tipos de datos, y más ajustes que deseemos que se apliquen antes de generar los objetos.

En nuestro ejemplo no necesitamos definir reglas de este tipo, pero es muy intuitiva la forma de uso, como en general la ofrecen los editores.

Ahora sí vamos a culminar la ejecución del wizard. Presionamos el botón Finish.

Reporte de análisis de impacto



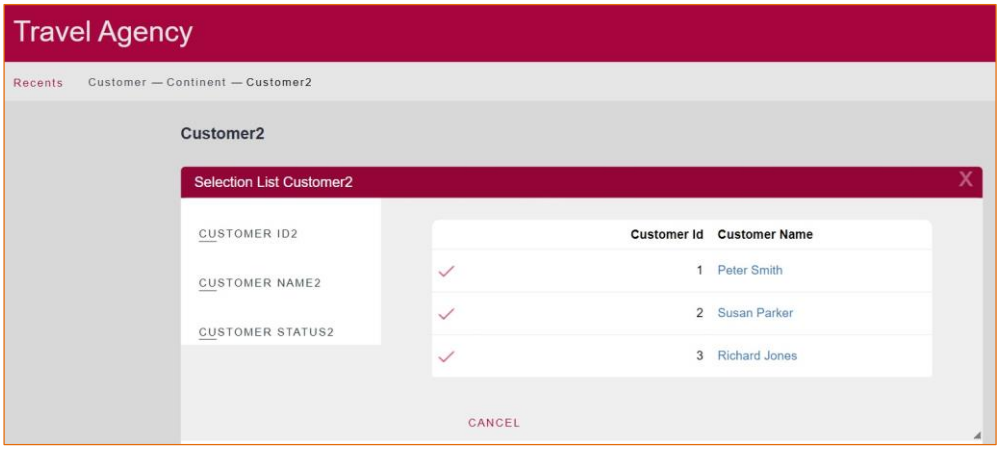
Veamos lo que se generó
El Data View Continent y el Data View Customer2 , además de las correspondientes transacciones Continent y Customer2.

Bien. Presionemos F5

En el reporte de análisis de impacto, podemos verificar que las tablas asociadas a los Data Views no se reorganizarán.

Continuemos...

En ejecución...



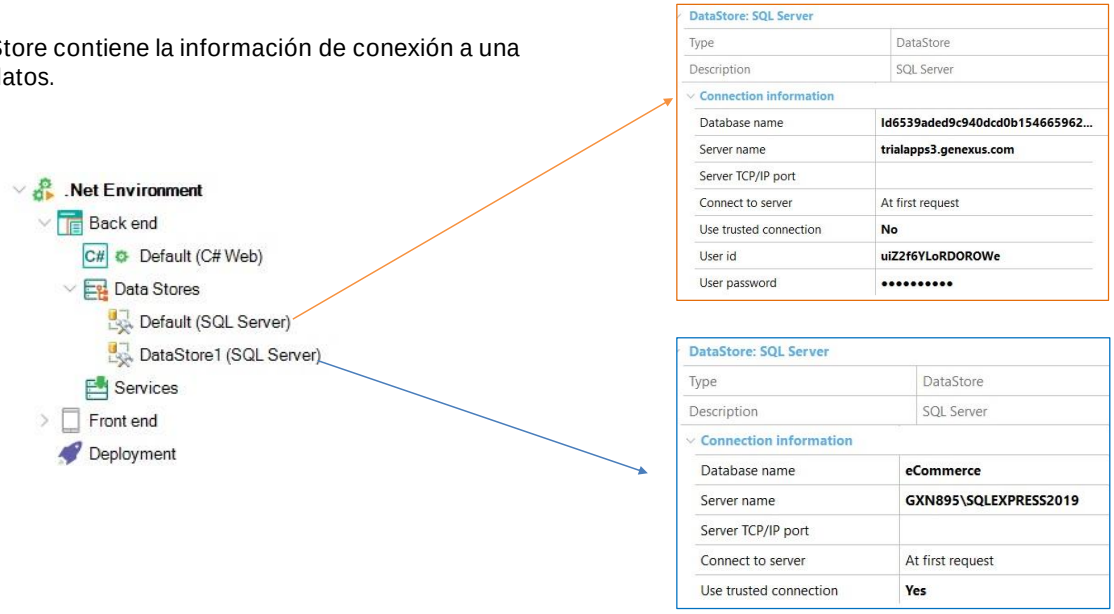
Y vamos a ingresar a la transacción Customer2. Recordemos que a través de la transacción Customer2 estamos accediendo directamente a la tabla CUSTOMER de la base de datos externa.

Podemos ver los registros que ya están ingresados. Vamos, por ejemplo, a ingresar un nuevo cliente, y vemos que el registro se ingresó correctamente.

Podemos también, por ejemplo, modificar el estado de otro cliente, e incluso podemos eliminar el cliente que acabamos de ingresar. Hemos observado, entonces, que realmente estamos trabajando directamente sobre la tabla CUSTOMER a través de la transacción Customer2 creada por el proceso de ingeniería inversa.

Data Stores

Un Data Store contiene la información de conexión a una base de datos.



Por último, veamos los Data Stores. En la vista Preferences, bajo el nodo .Net Envirnment que es nuestro ambiente de ejecución, podemos ver bajo el nodo Backend, los DataStores.

Vemos Default Data Store y DataStore1 ¿A qué corresponden?

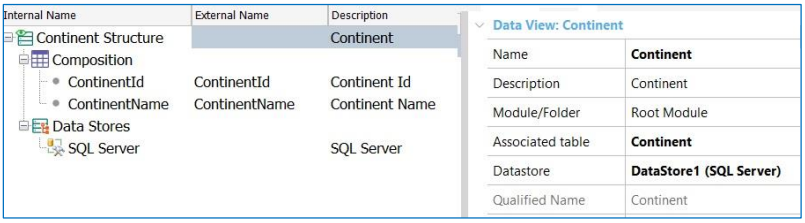
Como ya hemos mencionado, **un Data Store contiene la información de conexión a una base de datos**.
Cuando presionamos F5 por primera vez en una base de conocimiento y prototipamos en forma local, se nos pide la información relativa a la base de datos, y se crea **un Data Store Default**, con dicha información.

Si observamos sus propiedades, podemos observar que conecta con la base de datos creada en su momento. Corresponde a **la base de datos relacionada a nuestra base de conocimiento**. y dentro de la misma se crean las tablas que GeneXus determina, analizando nuestras transacciones definidas.

Ahora bien, ¿con qué base de datos conectará el DataStore1? Con la base de datos externa eCommerce a la cual nos hemos conectado.

Accediendo a información externa

DataView Continent




```
Print Title

For each Continent order ContinentName
     Print ContinentDetail
Endfor
```

Bien. Entonces, si finalmente observamos las propiedades de los DataViews Continent y Customer2 vemos que apuntan a dicho DataStore1 creado automáticamente por GeneXus en este proceso de ingeniería inversa.

Para finalizar y poder ver en ejecución un ejemplo más, hemos creado un listado muy simple con la lista de continentes.
Observemos en el source de este procedimiento que estamos haciendo referencia a la transacción base Continent, ordenando por el atributo ContinentName.

Para ejecutarlo, botón derecho, Run.

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