

Acesso a bases de dados externas

DBRET e DataViews

GeneXus[™]

Muitas vezes precisamos acessar bases de dados externas a partir de nossas aplicações GeneXus.

Acesso a bases de dados externas

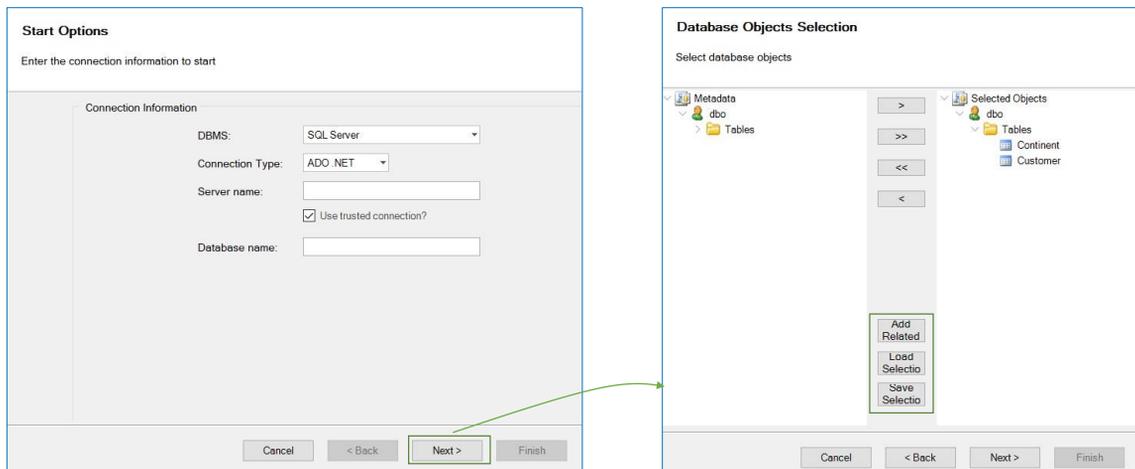


Por exemplo, podemos precisar carregar dados em tabelas de nossa base de dados associada à base de conhecimento com dados que se encontram em determinada base de dados externa, por exemplo, para realizar uma carga inicial... e depois disso, podemos não precisar mais continuar conectados a essa base de dados externa.

Ou podemos também precisar nos conectar e sempre manter conexão com determinada tabela ou certas tabelas de uma ou várias bases de dados externas... e não apenas para lê-las, mas também para inserir e modificar dados nelas.

Vejamos como é possível conseguir tudo isto em GeneXus.

Database Reverse Engineering Tool (DBRET) - Wizard



Em nossa base de conhecimento Travel Agency, observemos no menu: Tools, a opção que diz Database Reverse Engineering.

Selecionando esta opção, acessamos um wizard que nos permite aplicar engenharia reversa em bases de dados externas. Isto significa que para uma base de dados existente que contém tabelas, relações, índices, etc., serão criados no GeneXus todos os objetos e configurações necessárias para poder acessar facilmente.

Nesta primeira página do wizard, o que devemos fazer é indicar as informações de conexão à base de dados que queremos acessar. Dado que esta base de dados que queremos acessar é justamente SQL Server, aqui na opção "DBMS", deixamos "SQL Server". A opção "Connection Type" também permanece como está, e indicamos a seguir o nome do servidor e o nome da base de dados à qual queremos nos conectar. Agora, pressionamos o botão Next... e passamos para a segunda página do wizard.

Aqui, são apresentadas as tabelas que estão dentro da base de dados à qual estamos nos conectando. Devemos ter certeza de deixar na janela da direita apenas aquelas tabelas para as quais desejamos aplicar engenharia reversa, para acessá-las a partir de nossa base de conhecimento.

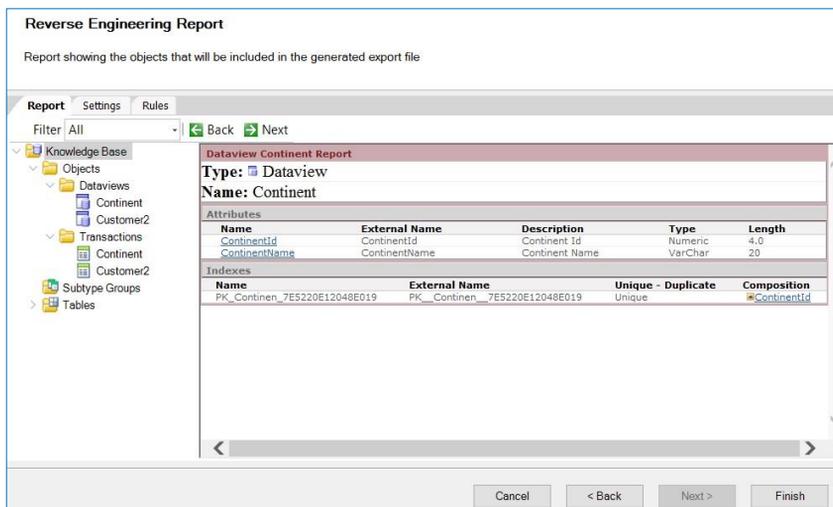
Nossa intenção é acessar as tabelas Customer e Continent, então devem ficar na janela da direita.

Bem, mas antes de clicar no botão Next, observemos estes outros botões:

O botão "Add Related", nos oferece a possibilidade de adicionar automaticamente todas as tabelas que estão em relação N-1 com aquelas tabelas que selecionamos.

Por sua vez, o botão "Save Selection", nos permite salvar a seleção de tabelas que realizamos, para que posteriormente possamos carregar esta mesma seleção rapidamente em outra execução, utilizando o botão "Load Selection".

Database Reverse Engineering Tool (DBRET) - Wizard



Agora sim pressionamos o botão Next

Neste momento começa o processo de engenharia reversa... ou seja, será avaliada a composição e tudo sobre as tabelas que selecionamos e serão definidos os objetos GeneXus, que sejam necessários.

Observemos esta página do Wizard que se abriu. Possui 3 abas:

Report

Settings

Rules

Vamos começar vendo a aba principal: Report. Este relatório que estamos vendo, nos informa sobre os objetos criados durante o processo de engenharia reversa.

Observemos que sob a pasta Objects, temos uma pasta Dataviews e outra pasta Transactions e abaixo de cada uma delas, vemos um objeto chamado: Continent e outro chamado: Customer2.

A primeira questão que nos surge é a seguinte: por que o nome "Continent" foi respeitado exatamente da mesma forma que se chama a tabela externa e em vez disso, no caso de "Customer" que é o nome da tabela externa, foi denominado "Customer2" para objetos da KB? O motivo é que em nossa base de conhecimento já existia uma transação

chamada "Customer" e sua tabela correspondente com o mesmo nome. Assim, outra transação chamada "Customer" não podia ser definida e por este motivo foi denominada "Customer2" como os demais objetos criados para acessar a tabela externa Customer.

Bem, e o que será um objeto Data View? Até agora não o conhecemos. Um objeto Data View permite definir e configurar informações de uma tabela externa. Ou seja, para cada tabela externa que precisamos acessar, deverá ter sido definido um Data View na base de conhecimento.

Objeto DataView

Permite definir e configurar informações de uma tabela externa. Para cada tabela externa à qual se necessita acessar, deverá haver um DataView na base de conhecimento.

Internal Name	External Name	Description
Customer2 Structure		Customer2
Composition		
CustomerId2	CustomerId	Customer Id
CustomerName2	CustomerName	Customer Name
CustomerStatus2	CustomerStatus	Customer Status
Data Stores		
SQL Server		SQL Server

Data View: Customer2	
Name	Customer2
Description	Customer2
Module/Folder	Root Module
Associated table	Customer2
Datastore	DataStore1 (SQL Server)
Qualified Name	Customer2
Object Visibility	Public

Internal Name	External Name	Description
Customer2 Structure		Customer2
Composition		
CustomerId2	CustomerId	Customer Id
CustomerName2	CustomerName	Customer Name
CustomerStatus2	CustomerStatus	Customer Status
Data Stores		
SQL Server		SQL Server

Data View Platform: SQL Server	
Description	SQL Server
Name	Customer
Location	
Schema name	dbo
Use external name	Yes

Como estamos na metade do processo de engenharia reversa, para não o cancelar, vamos observar e comentar nesta imagem como ficará definido o Data View "Customer2"

A primeira coisa que percebemos à primeira vista, sob Composition está uma lista de nomes de atributos, e observemos que para cada atributo existe um External Name (que é o nome que tem o campo físico na tabela externa) e um Internal Name (que é o nome dado no nível da base de conhecimento), ou seja, o Internal Name é o nome do atributo que está presente na transação que é criada automaticamente para acessar de forma interativa a tabela externa, e é o nome que incluiremos em grids, eventos, etc., em todos os objetos GeneXus.

Portanto, sob Composition, está o mapeamento de nomes de campos externos da tabela e seus correspondentes nomes de atributos internos com a nomenclatura que usamos em GeneXus.

Nós, na base de conhecimento sempre vamos nos referir aos nomes internos, e GeneXus ao gerar o código na linguagem correspondente, fará referência aos nomes dos campos físicos, considerando para isso a correspondência de nomes definida no Data View.

E onde está o nome da tabela externa para que GeneXus possa fazer referência à tabela ao gerar o código?

Aqui vemos que a plataforma externa é SQL Server e ao observar suas propriedades relacionadas, vemos a propriedade Name que contém o nome da tabela externa.

Objeto DataView

Associated table:: Tem por valor o nome da transação associada ao DataView.

Internal Name	External Name	Description
Customer2 Structure		Customer2
Composition		
CustomerId2	CustomerId	Customer Id
CustomerName2	CustomerName	Customer Name
CustomerStatus2	CustomerStatus	Customer Status
Data Stores		
SQL Server		SQL Server

Data View: Customer2	
Name	Customer2
Description	Customer2
Module/Folder	Root Module
Associated table	Customer2
Datastore	DataStore1 (SQL Server)
Qualified Name	Customer2
Object Visibility	Public

E onde se encontra a informação da base de dados externa na qual se encontra esta tabela?

Se observamos novamente as propriedades do nó principal do Data View, podemos observar a propriedade Data Store, que aponta para DataStore1, criado automaticamente no processo de engenharia reversa. Embora no final deste vídeo veremos onde estão definidos os Data Stores em uma base de conhecimento, por enquanto os conceitos que devemos aprender são os seguintes:

Cada Data View possui as informações de uma tabela externa e cada Data Store possui as informações de acesso a uma Base de dados. Então, para cada Data View, deve referenciar a qual Data Store pertence.

Bem, agora vamos observar a propriedade do Data View: “Associated table”

Tem por valor o nome da transação associada ao Data View, ou seja, a transação criada também automaticamente com o mesmo nome que o Data View, para permitir realizar inserções, modificações e exclusões na tabela externa de forma interativa.

As transações associadas a Data Views não causam a criação de tabelas ou reorganizações. Embora sejam transações sem diferenças com as demais, ao estarem referenciadas em Data Views como tabelas associadas, GeneXus entende que somente deverá gerar o form e sua programação, para permitir a interação com a tabela externa cuja informação se encontra no Data View em questão.

Vale explicar que as transações associadas a Data Views podem ter no máximo a mesma quantidade de atributos que a tabela externa. Pode ter menos atributos, caso alguns campos não sejam desejados acessar, ou no máximo a mesma quantidade que tem a tabela externa. Mas não mais, porque GeneXus não reorganiza tabelas externas.

Cabe mencionar também que, se neste processo de engenharia reversa tivesse ocorrido algum erro, teria sido informado.

Retornando ao Wizard...

Reverse Engineering Report

Report showing the objects that will be included in the generated export file

Report Settings Rules

Settings

Generate Transactions	True
Identify multilevel Transactions	False
Generate Schema	True
Generate Location	False
Naming rule	Prefix only when needed
Names separator	
Data Store	Default
Override Data Store Properties	False
Folder	
Use database descriptions	True
Generate Views as Transactions	False
Backup model before consolidate	True
Apply replace rules to attributes prefix	False

Generate Transactions
Choose if you want to generate Transactions

Cancel < Back Next > Finish

Bem. Agora sim continuemos vendo as outras abas oferecidas nesta página do wizard.

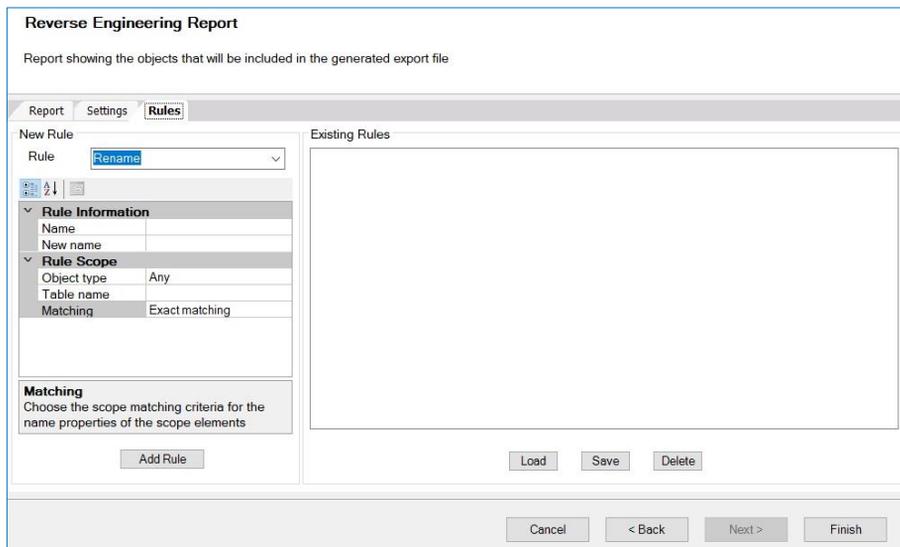
A aba Settings, como seu nome indica, permite realizar certas configurações.

Por exemplo, “Generate Transactions” que por padrão tem o valor True e, conforme aqui se descreve, nos permite indicar que sejam criadas transações para manipular os dados das tabelas externas de forma interativa. Se alterarmos o valor para False, então não são criadas transações, apenas os Data Views.

A opção “Identify Multilevel Transactions”, também como seu nome a descreve, permite configurar se queremos identificar relações de subordinação e que sejam definidas transações de mais de um nível, sempre que se identifiquem padrões que o permitam. Seu valor padrão é False, o que faz com que toda tabela a ser importada, defina uma única transação de apenas 1 nível e o Data View relacionado.

A opção “Generate Schema”, por sua vez, tem por padrão o valor True, e isto significa que as informações do schema da tabela são armazenadas na propriedade Schema do correspondente Data View.

Retornando ao Wizard...



Agora, não vamos continuar vendo cada uma das configurações possíveis, pois ao posicionar-se nelas, é fornecida a ajuda necessária, por isso vamos continuar com a aba Rules.

Devido ao fato de gerar objetos e atributos automaticamente a partir de uma base de dados externa, é possível que sejam gerados alguns conflitos de nomes.

Por este motivo, a aba Rules oferece a possibilidade de definir regras de mapeamento de nomes para “resolver” estas ambiguidades. De modo que esta aba nos permite definir regras para poder renomear atributos, objetos, etc., alterar tipos de dados e mais ajustes que queremos que sejam aplicados antes de gerar os objetos.

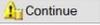
Em nosso exemplo não precisamos definir regras deste tipo, mas é muito intuitiva a forma de uso, como geralmente oferecem os editores.

Agora sim vamos concluir a execução do wizard. Pressionamos o botão Finish.

Relatório de análise de impacto

Database tables will not be changed.

This report describes Database changes since the last impact.
Please select Continue to proceed or Cancel.

Pattern:

Customer2
 Continent

Table Customer2 specification 

Table name: [Customer2](#)

Customer2 is associated to Data view Customer2. It will not be reorganized.
Customer2 is new

Table Structure 

Attribute	Definition	Previous values	Takes value from
 CustomerId2	Numeric (4), Not null, Autonumber		
CustomerName2	Varchar (20), Not null, NLS		
CustomerStatus2	Varchar (20), Not null, NLS		

Indexes 

Name	Definition	Composition
PK_CUSTOMER_A4AE64D805027C1A	primary key Clustered	 CustomerId2

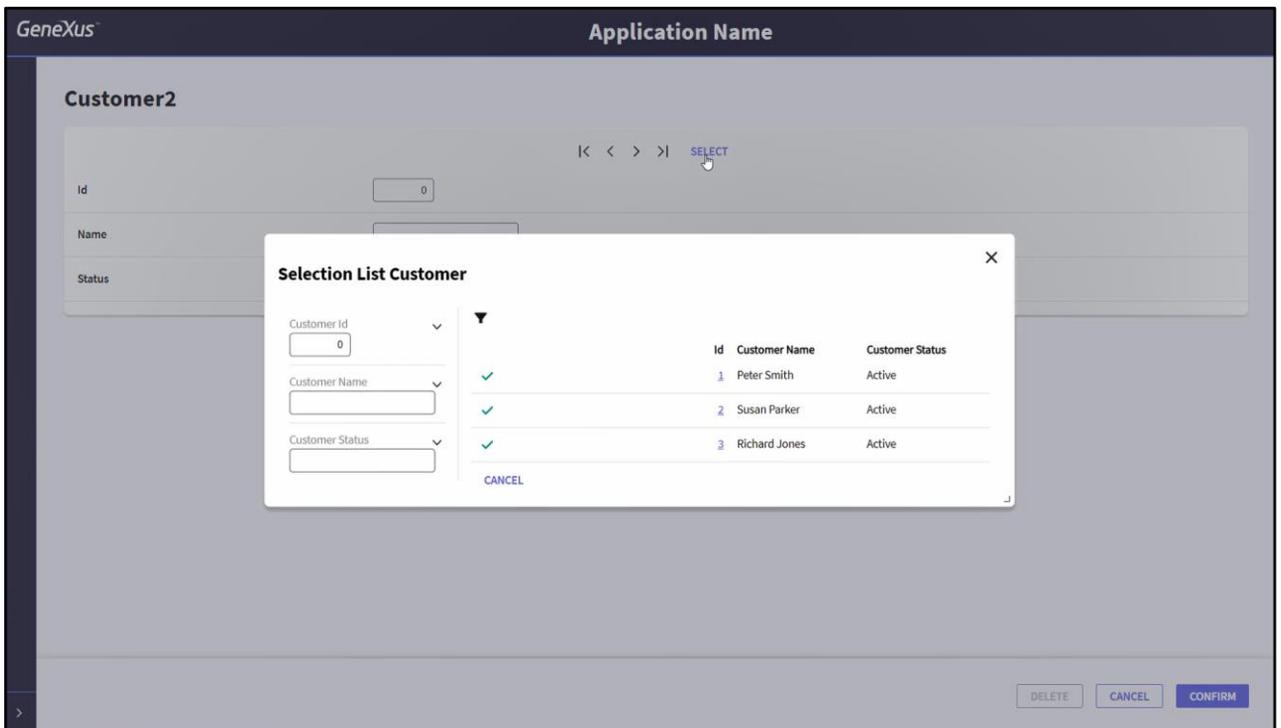
Vejamos o que foi gerado

O Data View Continent e o Data View Customer2, além das transações correspondentes Continent e Customer2.

Bem. Pressionemos F5

No relatório de análise de impacto, podemos verificar que as tabelas associadas aos Data Views não serão reorganizadas.

Continuemos...



E vamos entrar na transação Customer2. Lembremos que através da transação Customer2 estamos acessando diretamente a tabela CUSTOMER da base de dados externa.

Podemos ver os registros que já estão inseridos. Vamos, por exemplo, inserir um novo cliente, e vemos que o registro foi inserido corretamente.

Podemos também, por exemplo, modificar o estado de outro cliente, e podemos até mesmo excluir o cliente que acabamos de inserir. Observamos, então, que realmente estamos trabalhando diretamente na tabela CUSTOMER por meio da transação Customer2 criada pelo processo de engenharia reversa.

Data Stores

Um Data Store contém as informações de conexão a uma base de dados.

The screenshot shows the .Net Environment tree on the left with two Data Store nodes highlighted. Two configuration windows are shown on the right, each linked to a node in the tree by an arrow.

DataStore: SQL Server (Top Window)

Type	DataStore
Description	SQL Server
▼ Connection information	
Database name	Id6539aded9c940dcd0b154665962...
Server name	trialapps3.genexus.com
Server TCP/IP port	
Connect to server	At first request
Use trusted connection	No
User id	uiZ2f6VLoRDOROWe
User password	*****

DataStore: SQL Server (Bottom Window)

Type	DataStore
Description	SQL Server
▼ Connection information	
Database name	eCommerce
Server name	GXN895\SQLEXPRESS2019
Server TCP/IP port	
Connect to server	At first request
Use trusted connection	Yes

Por último, vejamos os Data Stores. Na visualização Preferences, sob o nó .Net Environment que é nosso ambiente de execução, podemos ver sob o nó Backend, os DataStores.

Vemos Default Data Store e DataStore1. A que correspondem?

Como já mencionamos, um Data Store contém as informações de conexão com uma base de dados.

Quando pressionamos F5 pela primeira vez em uma base de conhecimento e prototipamos de forma local, nos solicita as informações relacionadas à base de dados, e é criado um Data Store Default com estas informações.

Se observamos suas propriedades, podemos ver que conecta com a base de dados criada naquela oportunidade. Corresponde à base de dados relacionada à nossa base de conhecimento, e dentro dela são criadas as tabelas que GeneXus determina, analisando nossas transações definidas.

Agora, com qual base de dados conectará o DataStore1? Com a base de dados externa eCommerce à qual nos conectamos.

Acessando informação externa

DataView Continent

Internal Name	External Name	Description
Continent Structure		Continent
Composition		
• ContinentId	ContinentId	Continent Id
• ContinentName	ContinentName	Continent Name
Data Stores		
• SQL Server		SQL Server

Data View: Continent	
Name	Continent
Description	Continent
Module/Folder	Root Module
Associated table	Continent
Datastore	DataStore1 (SQL Server)
Qualified Name	Continent

Print Title

```
For each Continent order ContinentName
  Print ContinentDetail
Endfor
```

List of Continents	
	
	Africa
	America
	Antarctica
	Asia
	Europe
	Oceania

Bem. Então, se finalmente observamos as propriedades dos DataViews Continent e Customer2, vemos que apontam para o referido DataStore1 criado automaticamente por GeneXus neste processo de engenharia reversa.

Para finalizar e poder ver em execução mais um exemplo, criamos uma lista muito simples com a lista de continentes. Observemos no source deste procedimento que estamos fazendo referência à transação base Continent, ordenada pelo atributo ContinentName.

Para executá-lo, botão direito, Run.

GeneXus[™]

training.genexus.com
wiki.genexus.com