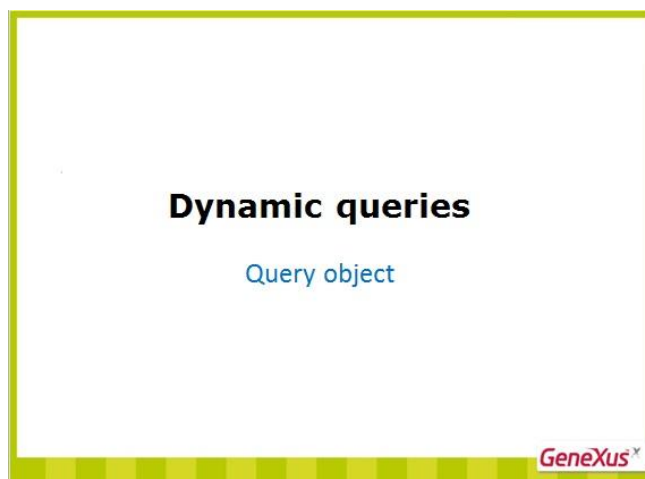
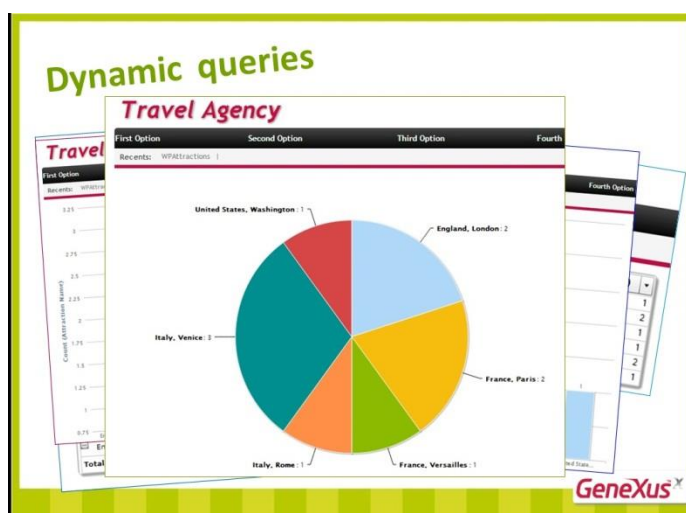


Objeto Query - Más conocimientos

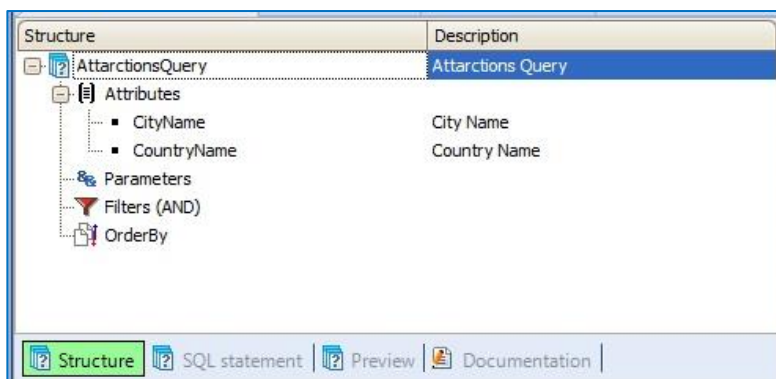


Hemos visto anteriormente las generalidades del objeto Query, los componentes de su estructura y la forma de verlo en ejecución a través del uso del control Query Viewer.



Vamos a concentrarnos ahora en presentar una serie de ejemplos en los cuales iremos aumentando la complejidad en el diseño de las consultas, y de esta forma conocer un poco más el alcance de este objeto GeneXus.

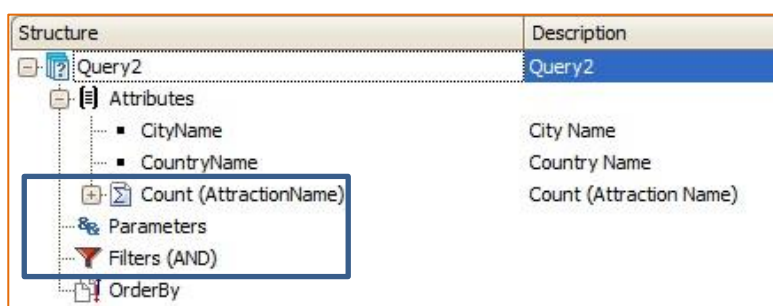
Como ya hemos mencionado, los elementos que definen la consulta aparecen bajo el nodo "Attributes", y deberá existir una tabla extendida que los contenga a todos ellos.



Es así entonces que podemos declarar:

- Atributos
- Funciones de agregación, ya sean simples o condicionadas
- Y Expresiones

Veamos el diseño de esta consulta. Queremos ver la cantidad de atracciones turísticas para cada ciudad, y queremos ver esta información como una tabla dinámica.



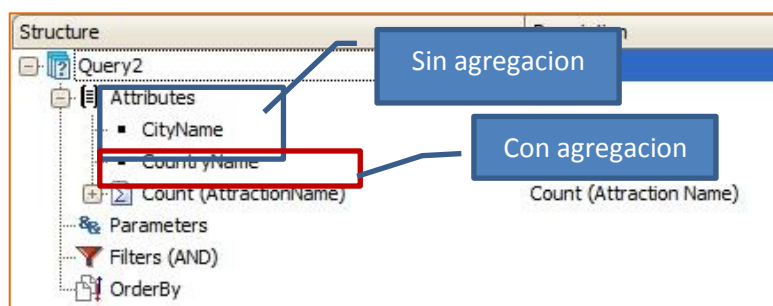
En la definición de una consulta, un atributo puede ser

- Con agregación – Esto corresponde a atributos numéricos con decimales, Sum, Count y Average
- O sin agregación – Que corresponde a descripciones, fechas y números sin decimales

Esto determina entonces su posición en la tabla dinámica, ya sea como Eje o como Dato.

Los Ejes, son los elementos por los cuales se puede pivotear, y son asociados automáticamente a los atributos sin agregación.

Los Datos, en cambio, son los elementos que se sumarizan según los ejes, y están automáticamente asociados a los atributos con alguna agregación.



Veamos la vista previa de nuestro objeto Query.

Count (Attraction Name)		
Country Name	City Name	Count (Attraction Name)
England	London	2
Total of 'England'		2
France	Paris	2
	Versailles	1
Total of 'France'		3
Italy	Rome	1
	Venice	3
Total of 'Italy'		4
United States	Washington	1
Total of 'United States'		1
Grand Total		10

Podemos entonces pivotar por CityName y CountryName que son los ejes, y de esta forma poder reordenar la misma información en diferentes filas y columnas.

Type

PivotTable

Chart type

Column

Count (Attraction Name)

City Name

Country Name

London	Total of 'London'	Madrid	Paris	Rome
England				
Count (Attraction Name)	Count (Attraction Name)	Count (Attraction Name)	Count (Attraction Name)	Count (Attraction Name)
1	1	1	2	

Analicemos

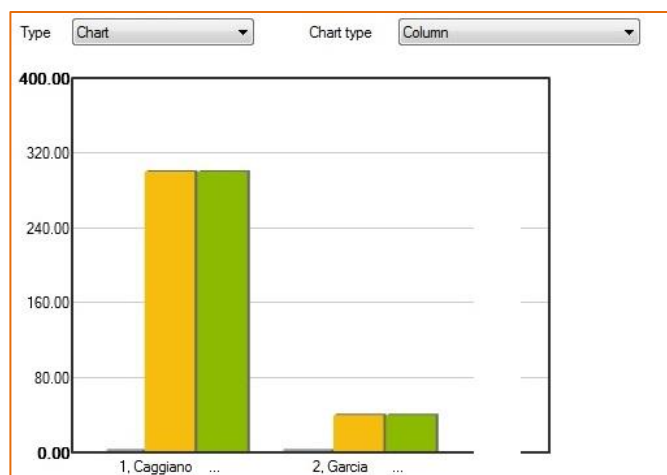
ahora esta nueva consulta que devuelva para cada cliente, el importe total por reservas, la cantidad de reservas realizadas, y el promedio diario de reservas.

Las agregaciones que se pueden obtener como datos de una consulta son tres: Sum (para sumar), Count (para contar) y Average (para promediar), y pueden anidarse.

Structure	Description
Query1	Query1
Attributes	
CustomerId	Customer Id
CustomerFullName	Customer Full Name
Count (ReservationDate)	Count (Reservation Date)
Sum (ReservationPrice)	Sum (Reservation Price)
Average (ReservationPrice) By (ReservationDate)	Average (Reservation Price) By (ReservationD..
Parameters	
Filters (AND)	
OrderBy	

Observemos especialmente que la función Average admite la cláusula **By**, para poder establecer por qué criterio se desea promediar.

Veamos la vista previa de la consulta como un grafico:



Pasemos a un nuevo ejemplo y comparemos ahora el diseño de estas dos consultas.

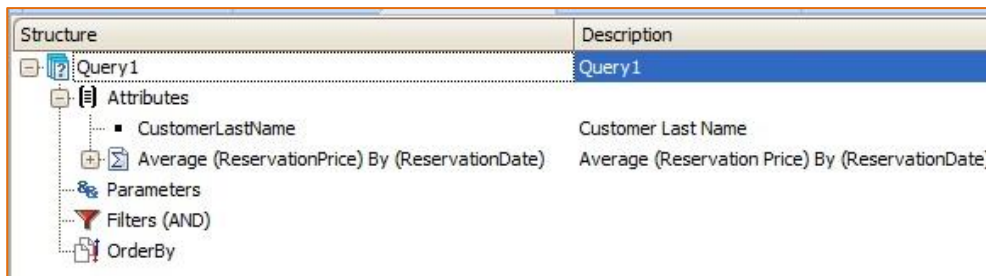
En el primer caso, se suman los ReservationPrice de todas las reservas del cliente, y se dividen por la cantidad de reservas.

En el segundo, en cambio, se agrupan las reservas por fecha, y la suma del ReservationPrice se divide por la cantidad de grupos de fechas. Es decir, se promediará de acuerdo a la cantidad de fechas diferentes.

En este caso necesitamos la cláusula By para lograr ese agrupamiento.

Veamos también la vista previa de cada una de estas consultas y comparemos su resultado.

Structure	Description
Query1	Query1
Attributes	
CustomerLastName	Customer Last Name
Average (ReservationPrice)	Average (Reservation Price)
Parameters	
Filters (AND)	
OrderBy	

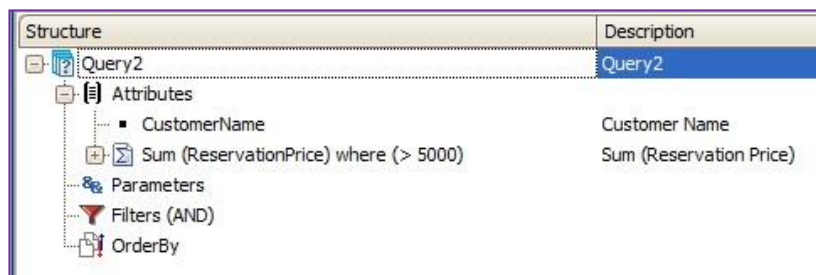
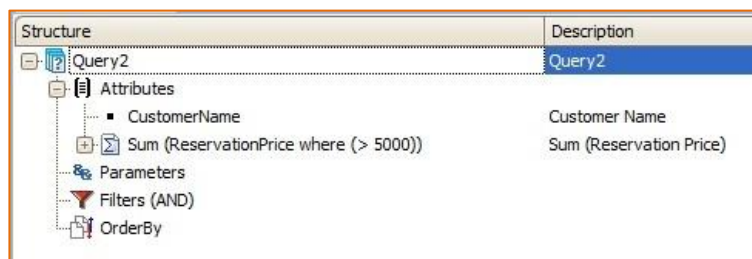


Analicemos ahora un nuevo ejemplo:

En el primer caso, de todas las reservas de un cliente, se van a sumar solamente aquellas que superan los 200 dólares.

En el segundo caso, en cambio, solamente se va a mostrar un cliente, si la suma de todas sus reservas supera los 200 dólares.

Comparemos también los distintos resultados.



Concentrémonos ahora en los filtros.

Como ya se ha mencionado, se tienen disponibles varios tipos: pueden ser rangos, listas, el uso del operador like, el uso de subconsultas, y es posible definir cualquier nivel de anidación de estos filtros y parametrizarlos utilizando los parámetros definidos.

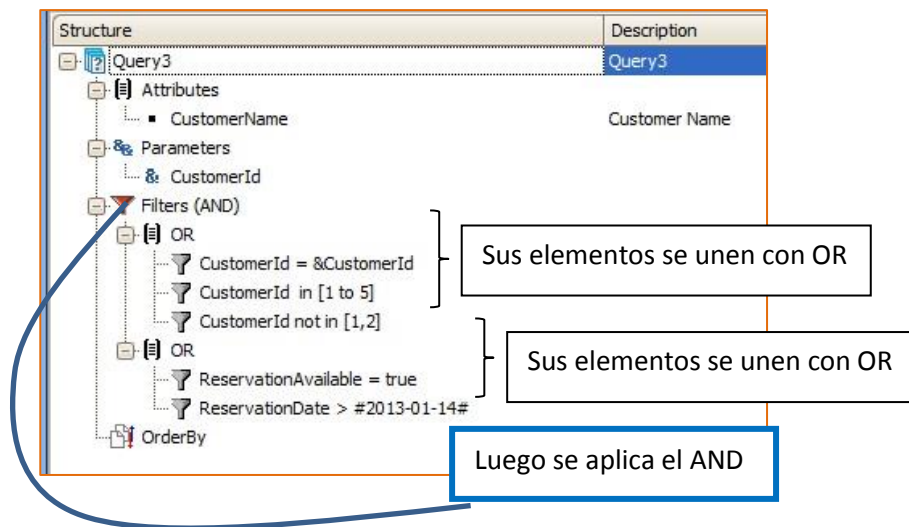
Se ofrecen dos tipos de “filter groups”: AND y OR.

En el ejemplo que estamos viendo ya hemos definido un grupo de filtros cuyos elementos estan unidos por el operador OR.

Vamos a definir un nuevo grupo. Hacemos click con el botón derecho, elegimos “Insert filter group”, y declaramos que estos elementos también van a estar unidos por el operador OR.

Vamos a declarar que el cliente tenga reservas disponibles, y que la fecha de la reserva sea posterior a la del día de hoy.

El resultado final entonces que surja de esta consulta será el de aplicar el operador OR declarado en cada grupo, para finalmente aplicar el operador principal AND.



Hasta acá entonces hemos presentado distintos ejemplos de diseño de consultas dinámicas y hemos visto su salida a través de la vista previa.

Recordemos que para poder ver en ejecución la salida de un objeto Query necesitamos crear un web panel e insertar el user control Query Viewer



Al insertar un control Query Viewer en el web form de un objeto, como sucede con casi todo user control, se consolidan algunos SDT en la KB algunos dominios enumerados, y se crean variables basadas en esos tipos en el objeto, así como también se agregan líneas de código de ejemplo para ayudar a configurar estas propiedades.

De esta forma el desarrollador solamente tendrá que utilizarlos.

Volvamos a GeneXus y veamos algunos ejemplos.

Queremos ver en ejecución la consulta que muestra la cantidad de atracciones turísticas por ciudad.

Structure	Description
Query2	Query2
Attributes	
CityName	City Name
CountryName	Country Name
Count (AttractionName)	Count (Attraction Name)
Parameters	
Filters (AND)	
OrderBy	

Así que hemos creamos un web panel de nombre WPQuery y arrastramos el user control Query viewer desde la Toolbox.

Si queremos ver la consulta como un gráfico podemos directamente indicar el nombre de nuestro objeto Query en la propiedad Object del control, y declarar el Output de tipo Chart.

QueryViewer: QueryViewer1	
AutoRefreshGroup	
ControlName	QueryViewer1
+ EventParameters	
- Data Bindings	
Object	Query6
Axes	&Axes
Parameters	&Parameters
UseCache	False
+ Appearance	
+ Export	
- Output	
Type	Chart
ChartType	Pie 3D

Pero si el usuario final desea poder elegir en ejecución la forma de visualizar la consulta y no declararla previamente en forma estática, vamos a tener que configurar algunas propiedades en ejecución.

Como primer paso hemos definido en el form del web panel un combo box para que el usuario pueda elegir la forma de la salida, e insertamos un botón.

Analicemos el evento Enter asociado.

Estamos asignando a la propiedad Type del Query viewer el valor seleccionado en el combo, e indicamos también que si se elije la opción Chart entonces se visualice como un gráfico Column 3D.

```

Event Enter
  Queryviewer1.Type = &OutputType
  if &OutputType = "Chart"
    Queryviewer1.ChartType = QueryviewerCharttype.Column3d
  endif
Endevent

```

Veamos ahora su comportamiento en ejecución. Presionemos F5.

Seleccionamos el web panel WPQuery.

Podemos pivotear para ver los datos ordenados de distinta forma, seleccionar la salida como una tabla estática, o como un gráfico que se verá en columnas 3D.

Pivot Table View

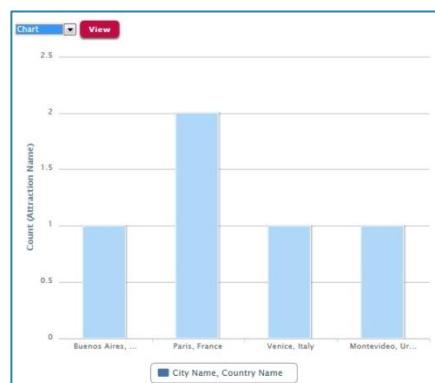
Pivot Table
Table
Chart

(Attraction Name)

City Name	Country Name	Count (Attraction Name)
Buenos Aires	Argentina	1
Total of 'Buenos Aires'		1
Montevideo	Uruguay	1
Total of 'Montevideo'		1
Paris	France	2
Total of 'Paris'		2
Venice	Italy	1
Total of 'Venice'		1
Grand Total		5

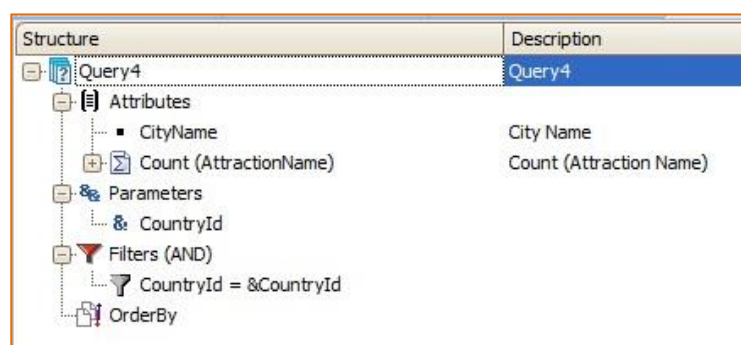
Table View

City Name	Country Name	Count (Attraction Name)
Buenos Aires	Argentina	1
Paris	France	2
Venice	Italy	1
Montevideo	Uruguay	1

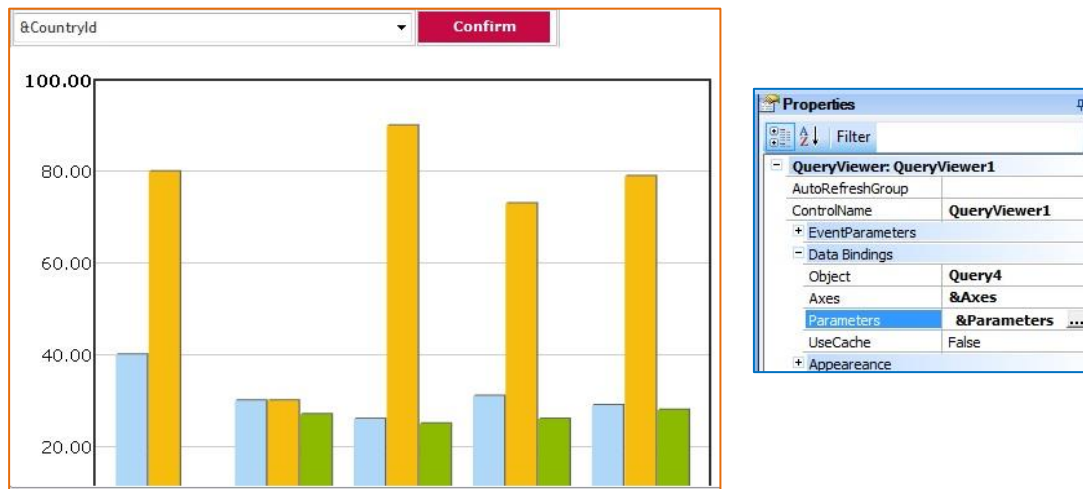


Veamos por último un ejemplo que requiere del pasaje de parámetros.

Queremos poder elegir en ejecución un determinado país, y graficar la cantidad de atracciones turísticas por ciudad.



Hemos creado también el siguiente web panel de nombre AttractionsPerCountry, donde es posible seleccionar un país desde un combo dinámico,



Observemos las propiedades del control Query Viewer, y veamos que en la propiedad Parameters tiene asociada la variable &Parameters.

Esta variable está basada en el sdt QueryViewerParameters que se crea automáticamente al arrastrar el control Query viewer. Y corresponde a una colección de parámetros donde cada uno tiene su nombre y su valor.

Name	Type	Description	Is Collection
QueryViewerParameters		Query Viewer Par...	<input checked="" type="checkbox"/>
Parameter		Parameter	<input type="checkbox"/>
Name	Character(20)	Name	<input type="checkbox"/>
Value	Character(20)	Value	<input type="checkbox"/>

Volvamos al web panel y observemos entonces el evento Enter asociado al botón:

```

Event Enter
    &Parameters = new()
    &Parameter.Name = "CountryId"
    &Parameter.Value = &CountryId.ToString().Trim()
    &Parameters.Add(&Parameter)
    QueryViewer1.Type = QueryViewerOutputType.Chart
    QueryViewer1.ChartType = QueryViewerChartType.Column3d

```

Endevent

Ambas variables, &Parameters y &Parameter, han sido creadas automáticamente en el web panel al arrastrar el Query viewer.

En el caso de la variable &Parameter representa a un elemento de la colección de parámetros, mientras que la variable &Parameters representa a la colección de parámetros.

Así que inicializamos ambas variables y le asociamos a la variable &Parameter el nombre y el valor correspondiente del parámetro que utilizamos, o sea CountryId, teniendo en cuenta que debido a la estructura del SDT los elementos deben ser de tipo Character. Luego agregamos el parámetro a la colección.

Y finalmente asociamos la salida como un gráfico, en particular, como columnas 3d.

Solamente nos está faltando asociar el objeto Query al control Query viewer, y asociándole al control Queryviewer1, a la propiedad ObjectName el nombre de nuestro objeto Query.

Veamos el comportamiento en ejecución. Presionemos F5.

Y seleccionamos AttractionsPerCountry.

