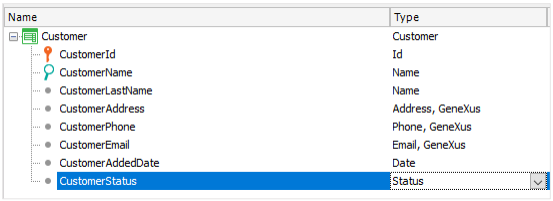
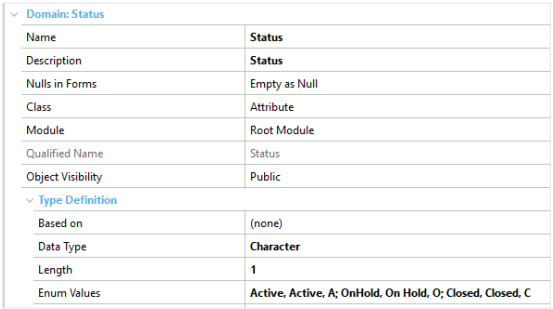
Data Selectors

GeneXus

GeneXus

Data Selectors





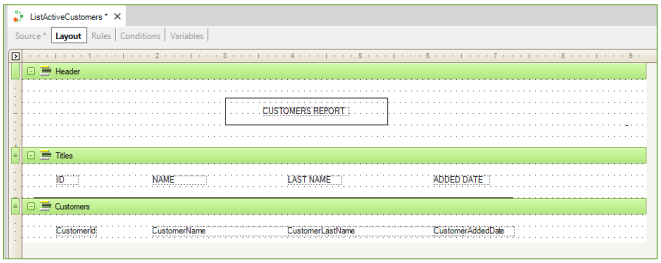
Supongamos que en la transacción Customer de nuestra aplicación, tenemos el atributo CustomerStatus para poder representar uno de los tres estados (active, on hold, closed) que puede tener un cliente en el sistema de la agencia de viajes. Para ello se ha definido un nuevo tipo de datos enumerado.

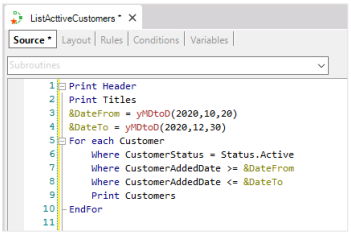
Supongamos que en varios lugares de la aplicación necesitamos trabajar con los clientes activos ingresados entre un par de fechas determinadas. Por ejemplo:

Queremos generar un listado pdf que reciba un rango de fechas y muestre los clientes activos que fueron ingresados al sistema entre esas fechas.

GeneXus

Data Selectors





≈

Para esto creamos un procedimiento de nombre ListActiveCustomers.

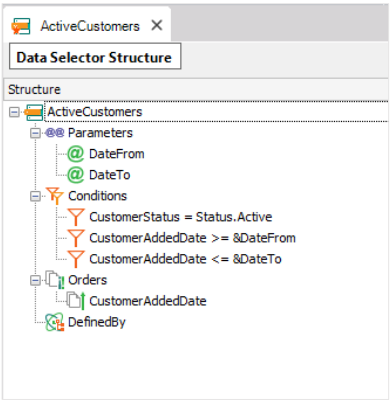
Si implementáramos las consultas con lo que sabemos hasta el momento, sería mediante las siguientes tres condiciones

Ya que éstas especificaciones las vamos a utilizar en varios lugares, para ahorrarnos el trabajo de tener que repetirlas en todos los lugares donde las necesitemos, podemos realizar esas definiciones en un único lugar, dándoles un nombre, y de allí en más utilizar ese nombre como referencia. Ese lugar es el objeto Data Selector.

GeneXus

Data Selectors

Structure:



Creamos un Data Selector al que llamamos “ActiveCustomers”, y allí definimos las condiciones que acabamos de ver.
Observemos la estructura que tiene este objeto.

En parameters, declaramos los parámetros que recibirá el Data Selector, que luego serán utilizados en Conditions. En este ejemplo serán dos variables, &DateFrom y &DateTo.

En Conditions se ingresan las condiciones que queremos se cumplan para filtrar los datos a ser recuperados. En este caso indicamos que el status del cliente deberá ser activo. Y que la fecha de ingreso del cliente deberá ser mayor o igual a la variable DateFrom, y menor o igual a DateTo.

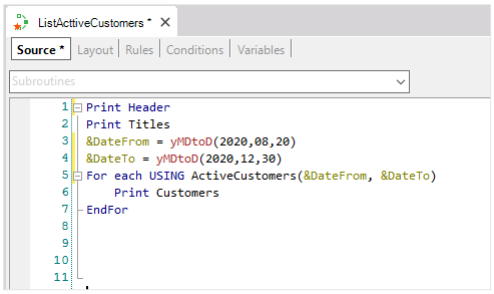
Luego en Orders, podremos indicar en que orden queremos recibir los datos que serán recuperados. Para este caso ingresamos CustomerAddedDate, para ordenar los datos por fecha.

Y por último tenemos Defined By, donde podremos ingresar un atributo o lista de atributos, los cuales colaboran para definir la tabla base final. Esto puede servir para resolver algún problema de ambigüedad en la determinación de la tabla base. Para este ejemplo lo dejamos vacío.

GeneXus

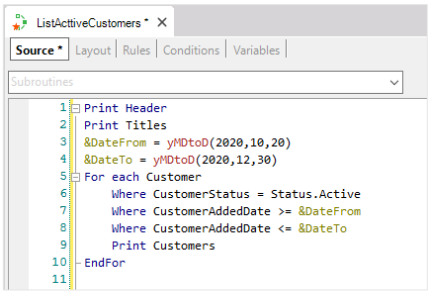
“Using” clause

With Data Selectors

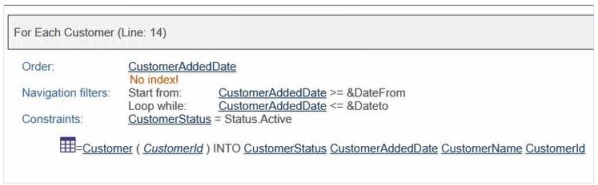


≈

Without Data Selector



Navigation View



Como dijimos, Esta definición centralizada nos permitirá reutilizarla en todos los lugares donde se necesita esa consulta, facilitando el mantenimiento (si se necesita cambiar algo de la definición, se realiza en un único lugar y se aplica automáticamente a todos los lugares de la KB donde se utilice).

Un Data Selector especifica, en base a los parámetros recibidos, un conjunto de condiciones y ordenamientos para la información de manera centralizada, de modo de no tener que repetir las cláusulas order, where y defined by en cada lugar en que se necesiten.

Veamos cómo, una vez definido el data selector, lo utilizaríamos desde nuestro procedimiento que lista los clientes.

Lo utilizamos a través de la cláusula using. Aquí vemos como debería utilizarse para el ejemplo planteado.
Dentro de este for each hacemos un Using al Data Selector ActiveCustomers, pasándole por parámetros dos variables, que corresponderán a las fechas que queremos filtrar los datos.
El comportamiento es en todo análogo a la especificación anterior.

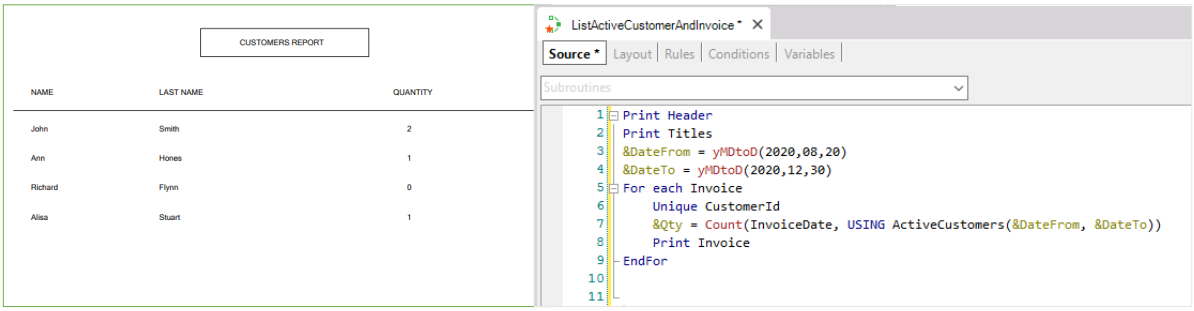
incluso a la hora de determinar la tabla base, o de resolver su navegación,

es igual a si en vez del DataSelector se hubieran escrito directamente sus cláusulas.
¿Se pueden agregar cláusulas order y cláusulas where al for each, además de lo que ya viene del Data Selector? La respuesta es sí. Las cláusulas se suman.

En el listado de navegación, vemos que se recorrerá la tabla Customer, se ordenará por Customer AddedDate, se recorrerán los registros según los filtros que aplicamos dentro de Conditions, y como restricción filtrara sólo los clientes con status activo.

GeneXus

Using Data Selectors in formula



Presentamos ahora un tercer ejemplo donde podemos utilizar nuestro Data Selector.
Para este caso, supongamos que necesitamos mostrar en un listado PDF, todos los clientes con facturas y su cantidad, ingresando un rango de fechas para poder contar únicamente las facturas de los clientes activos ingresados al sistema entre esas fechas. Si el cliente no está activo o fue ingresado fuera de esas fechas, será listado pero su cantidad de facturas mostrará cero.

Aquí utilizamos el data selector dentro de una fórmula, específicamente dentro de la fórmula Count.
Recordemos que el segundo parámetro de una fórmula aggregate, es para escribir las condiciones que deberán cumplir los registros para ser "agregados".

Si observamos con detenimiento este caso, estamos recorriendo en el for each la tabla Invoice. Como cada cliente puede tener N facturas, colocamos la cláusula Unique para quedarnos con una de esas facturas para el cliente, así con la fórmula Count sobre la misma tabla, contamos las facturas de ese cliente, que además cumplan con las cláusulas del DataSelector: es decir, si el cliente no está activo, el Count dará 0, porque para todas sus facturas el cliente no será uno activo. Si en cambio sí está

activo, solamente se contaran las facturas cuyo rango esté entre el considerado.

GeneXus

Ways to use Data Selector in For each

- **USING** clause ≈ Add where clauses in the For each

The attributes intervene in the determination of the base table of the For Each

- **IN** operator

Different queries to the database.

The attributes do **NOT** intervene in the determination of the base table of the For Each

Conceptualizando lo visto hasta el momento, tenemos dos usos muy distintos del Data Selector, dependiendo de cómo se lo utilice. Si es mediante la cláusula USING, no habrá ninguna diferencia con tener directamente dentro del for each, las cláusulas order o where del Data Selector. O cómo vimos recién, si lo usamos dentro de una fórmula agreggate, es análogo a ingresar allí las condiciones que pusimos dentro de "Conditions" en el Data Selector.

Como mencionamos, al declarar la cláusula Using, le estaremos diciendo que agregue los órdenes y filtros del Data Selector a los del for each. Por eso los atributos que integren el Data Selector deberán pertenecer a la tabla extendida de la tabla base del For each.

En cambio, si utilizamos el operador IN en una cláusula where, por ejemplo, estamos, sí, estableciendo una consulta independiente, por lo que ese Data Selector, para resolverse, tendrá que navegar su tabla base. Como si fuera un for each independiente.
El IN se utiliza cuando se desea realizar una subconsulta, cuyo resultado será utilizado luego como filtro en el For each, como vimos recién.

Según la forma en que lo invoquemos los atributos definidos dentro del Data Selector intervendrán o no en la determinación de la table base del

for each:

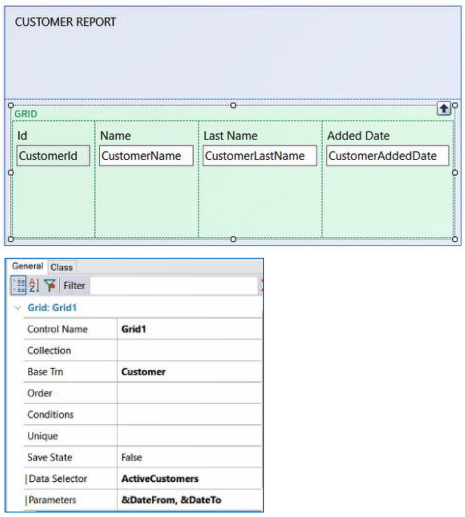
Si invocamos al Data Selector a través de la cláusula USING, los atributos declarados dentro de este objeto, intervienen en la determinación de la tabla base del for each donde lo estamos llamando.

Si lo invocamos a través de la cláusula IN, los atributos del Data selector no intervienen en la determinación de la tabla base del for each.

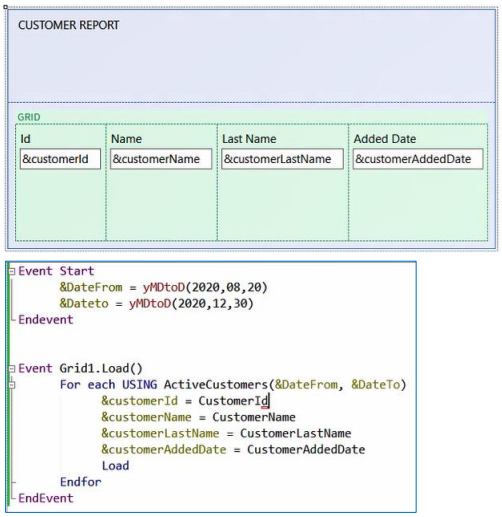
GeneXus

Data Selector In Web Panel

Web Panel with base table



Web Panel without base table



Los tres ejemplos que acabamos de plantear, los hicimos sobre objetos procedimiento, ya que queríamos listar la información en archivos PDF. Pero si simplemente quisiéramos mostrar la información en pantalla mediante un Web Panel, ¿cómo lo haríamos?.

Volvamos un momento sobre el primer ejemplo, en el que nos interesaba listar los clientes activos, en un rango de fechas determinado, utilizando para esto el Data Selector que creamos. Pero en esta oportunidad mostraremos los registros en pantalla con un Web Panel.

En el Web Layaout de nuestro Web Panel, declaramos un grid, y agregamos los atributos que nos interesan listar.
Dentro de las propiedades del Grid, vemos una de nombre Data Selector. Allí deberemos ingresar el nombre de nuestro Data Selector que queremos utilizar. Esto es análogo a cuando en un objeto procedimiento lo invocábamos desde el for each a través de la cláusula Using.

Ejecutemos para probar. Vemos que nos lanza un error, ya que recordemos que el Data Selector ActiveCustomers recibía por parámetro dos variables, las que contendrán las fechas. Por lo que deberemos pasarle esta información.
Para esto creamos las variables DateFrom y DateTo del tipo Date. En el evento Start las declaramos y para esta prueba las inicializamos con estas fechas.

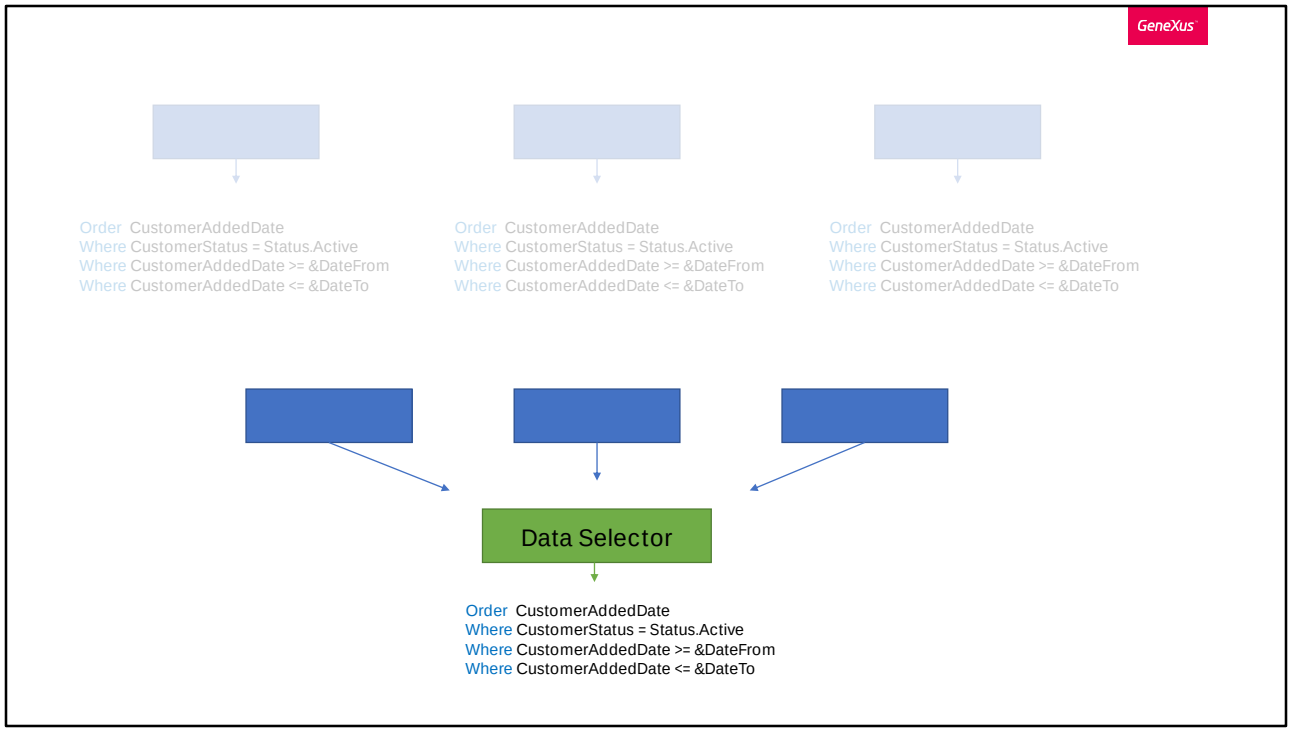
Para pasarle los parámetros al Data Selector lo hacemos desde la propiedad Parameters del grid.

Ahora sí, volvemos a ejecutar.
Vemos que se nos muestran en pantalla los clientes, filtrando por los activos y comprendidos entre las fechas indicadas. Ordenados por fecha de ingresados, tal como definimos en el Data Selector.

Supongamos el mismo ejemplo, pero que el Web Panel en lugar de tener atributos, tenga variables. En ese caso se trata de un Web Panel sin tabla base, a diferencia del anterior. Por este motivo, aquí no nos servirá utilizar la propiedad Data Selector del grid, ya que para recorrer todos los clientes deberemos hacerlo con un for each, y es allí donde debemos llamar a nuestro Data Selector. Por lo que en estos casos, la declaración será similar a la que vimos con el objeto procedimiento.

En la sección eventos, debemos declarar el evento Load del grid. Dentro un for each con la cláusula USING seguido del nombre de nuestro Data Selector, pasándole por parámetro las variables que recibirá este para filtrar las fechas.
Y dentro del for each, a las variables del grid le asignamos el valor de los atributos que nos interesa mostrar.
Por último definimos el comando Load, y cerramos el for each y el evento Load.

Ejecutamos, y vemos los resultados deseados.



Resumiendo, el Data selector es un objeto que nos permite almacenar un conjunto de parámetros, conditions, orders, para utilizarlo/invocarlo desde diferentes consultas y cálculos, y reutilizar la misma navegación varias veces.

Por tanto, ¿en qué lugares podremos utilizar un Data Selector? En todos los que especifiquen consultas a la base de datos. Por ejemplo, en grids de paneles y web panels o en grupos de Data Providers.

Puede encontrar más información sobre Data selectors en nuestro wiki.

GeneXus™

training.genexus.com
wiki.genexus.com