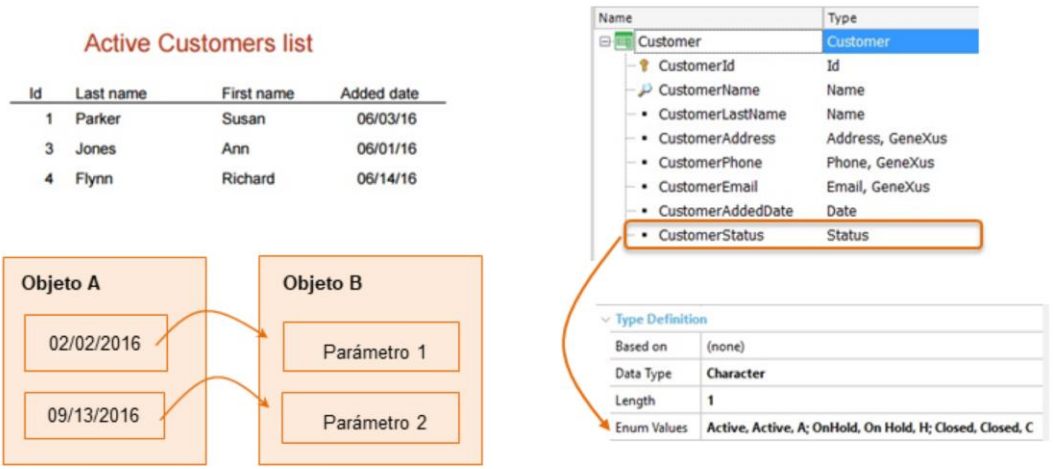
Data Selectors

Reutilizando definiciones

GeneXus 16

Escenario

- Clientes activos ingresados entre dos fechas dadas... en varias consultas

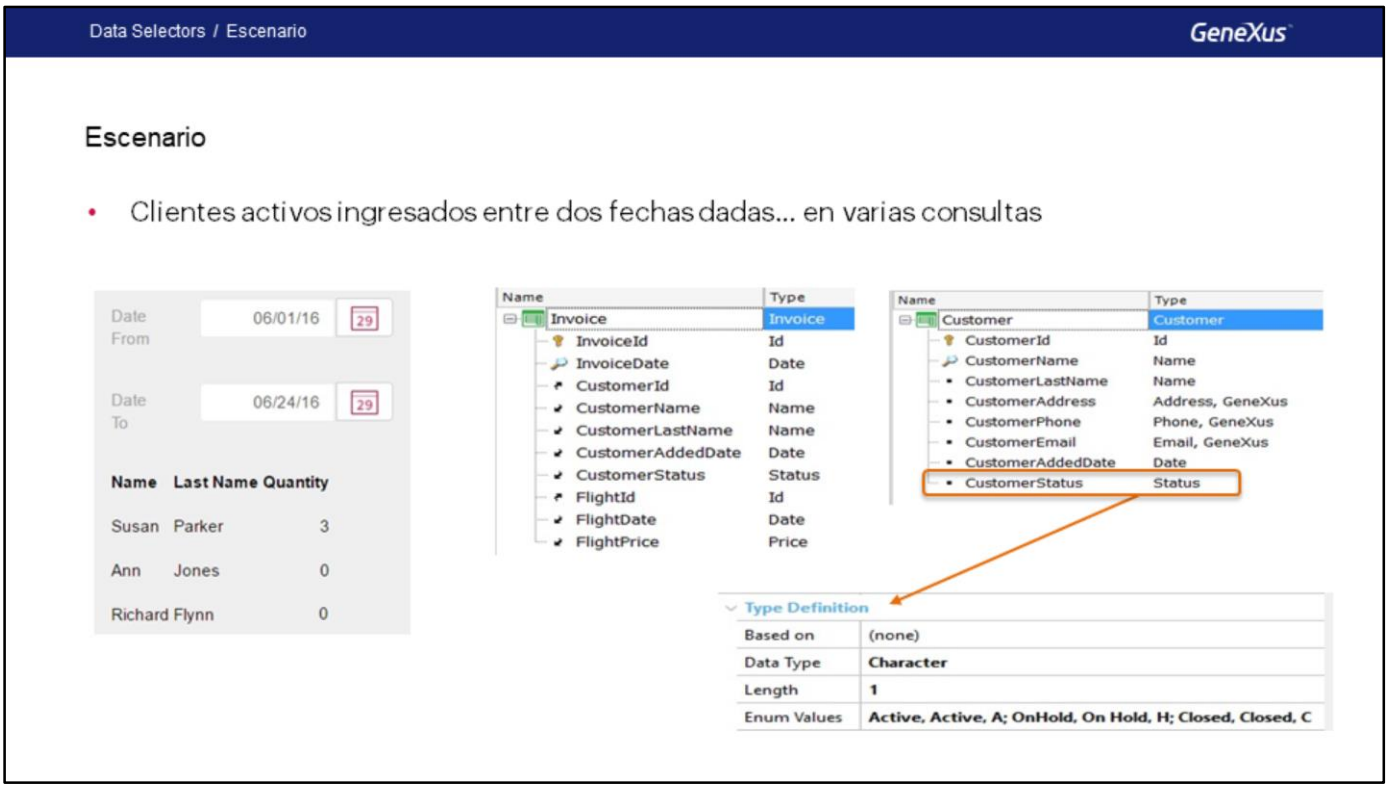


Supongamos que hemos agregado a la transacción Customer el atributo CustomerStatus, para poder representar uno de los tres estados (active, on hold, closed) que puede tener un cliente en el sistema de la agencia de viajes. Para ello se ha definido un tipo de datos Status enumerado, como se ve en la imagen.

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

1. Un listado pdf que reciba un rango de fechas (&start y &end) y muestre los clientes activos que fueron ingresados al sistema entre ese par de fechas dadas.

En este ejemplo nos vamos a adelantar a algo que veremos después: un objeto puede recibir valores de parte del objeto que lo llama, necesarios para realizar su operación (como en nuestro ejemplo, un rango de fechas), pero para ello tiene que ser capaz de recibirlos. Para hacer que un objeto pueda recibir valores (a los que llamamos parámetros) hay que declararlos (después veremos cómo).

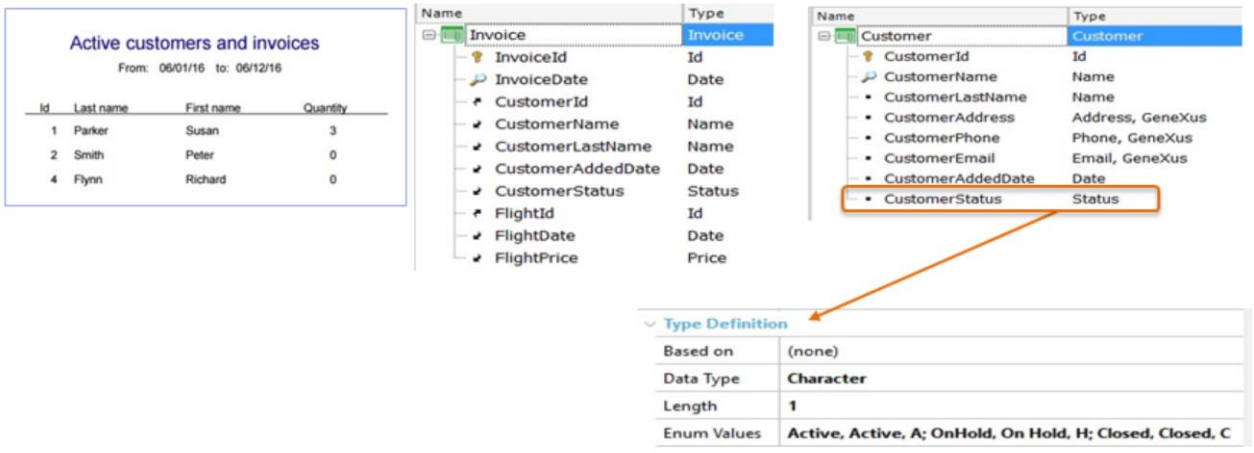


2 - En un web panel que muestra todos los clientes con facturas y su cantidad de facturas dándole al usuario la posibilidad de ingresar un rango de fechas para poder contar únicamente las facturas de los clientes activos ingresados al sistema entre esas fechas. Si el cliente no está activo o fue ingresado fuera de esas fechas, será listado pero su cantidad de facturas será cero.

Vale mencionar que un web panel es un tipo de objeto GeneXus muy flexible que permite diseñar todo tipo de consultas interactivas a la base de datos. Más adelante en el curso veremos este objeto en profundidad.

Escenario

- Clientes activos ingresados entre dos fechas dadas... en varias consultas



3. En un listado pdf necesitamos mostrar lo mismo que en el web panel anterior.

SOLUCIÓN con lo sabido hasta el momento

- Listado de clientes Activos ingresados entre dos fechas.

```
Print Title
For each Customer
    Where CustomerStatus = Status.Active
    Where CustomerAddedDate >= &DateFrom
    Where CustomerAddedDate <= &DateTo
    Print Customer
Endfor
```

```
Parm(in: &DateFrom, in: &DateTo);
```

Valores recibidos de otro objeto (parámetros)

- Listado de clientes Activos ingresados entre dos fechas y su cantidad de facturas.

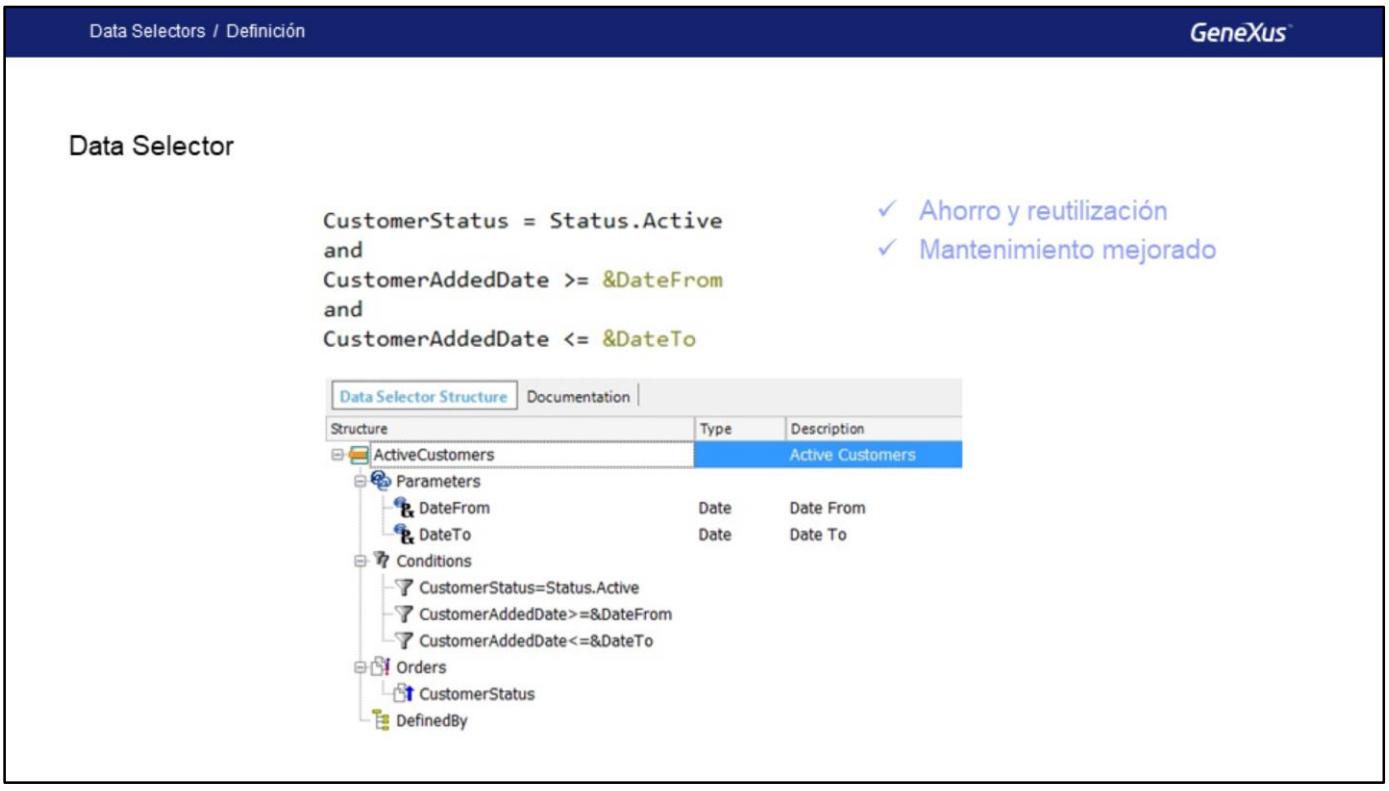
```
Print Title
For each Invoice
    Unique CustomerId
    &Qty = Count(InvoiceDate,
           CustomerStatus=Status.Active and CustomerAddedDate>=&DateFrom and CustomerAddedDate<=&DateTo)
    Print Customer
Endfor
```

Si implementáramos las consultas antes mencionadas con lo que sabemos hasta el momento, podemos observar que se repiten en ambos casos las tres condiciones marcadas arriba.

Recordar que escribir tres cláusulas where equivale a escribir una sola, donde sus condiciones se unen con AND.

Observación:
En general un objeto declara los parámetros mediante los que intercambia información con quien lo llama a través de una regla: la regla parm. En nuestro caso en ambos listados crearemos dos variables: &DateFrom y &DateTo, para recibir (por eso se coloca el “in”) el rango de fechas desde el llamador.

Definición



Para ahorrarnos el trabajo de tener que repetir esas mismas especificaciones en todos los lugares donde las necesitemos (el web panel y los procedimientos anteriores, así como en objetos de otro tipo que veremos más adelante), 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.

Para optimizar la consulta en un for each ordenaríamos por CustomerStatus, dado que filtramos por igualdad por ese atributo.

Como vemos, en el ejemplo creamos un data selector al que llamamos “ActiveCustomers”, y allí definimos las condiciones, el order, y declaramos los parámetros &DateFrom y &DateTo, variables que se utilizan en dos de las condiciones.

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).

Veamos cómo, una vez definido el data selector, lo utilizaríamos en los ejemplos que mencionamos.

Uso

En cláusula Using de For each

```
Print Title
For each Customer
    Using ActiveCustomers(&DateFrom, &DateTo)
    Print Customer
Endfor
```

```
Parm(in: &DateFrom, in: &DateTo);
```

Active Customers list

Id	Last name	First name	Added date
1	Parker	Susan	06/03/16
3	Jones	Ann	06/01/16
4	Flynn	Richard	06/14/16

Lo utilizamos a través de la cláusula **using.** El comportamiento es en todo análogo a la especificación anterior. Aquí vemos el caso del primer listado.

Para el caso del comando for each, podría hacerse otro uso del Data Selector, que es ejecutándolo como si fuera una consulta independiente a la base de datos. No lo veremos en este curso, pero es el caso de usar el operador **in**. Por ejemplo, si a los customers les agregáramos el país de procedencia, y quisiéramos listar los países que tengan clientes activos e ingresados al sistema entre un par de fechas dadas, haríamos:

For each Country
Where CountryId **in** ActiveCustomers(&DateFrom, &DateTo)
...
endfor

Teniendo aquí dos consultas a la base de datos: una, la del Data Selector, que devolverá el conjunto de clientes activos e ingresados entre el par de fechas indicados y sus respectivos países, y otra, la del for each, que filtrará los países que estén dentro de ese conjunto.

Puede encontrar más documentación sobre Data selectors en For eachs en nuestro wiki
http://wiki.genexus.com/commwiki/servlet/wiki?5312,Data+Selectors+in+For+Each+command

En Fórmula

```
Print Title
For each Invoice
    Unique CustomerId
    &Qty = Count(InvoiceDate, using ActiveCustomers(&DateFrom, &DateTo))
    Print Customer
Endfor
```

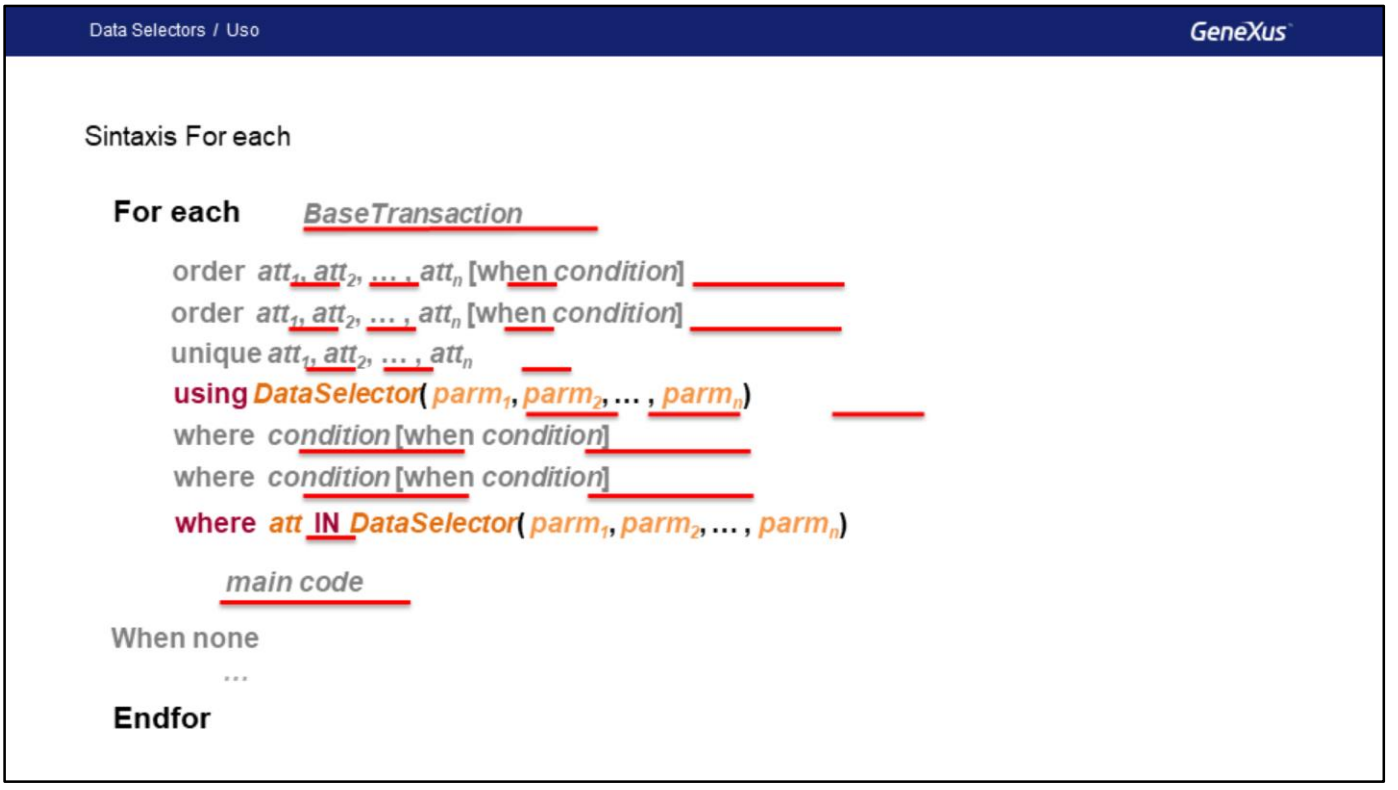
Active customers and invoices

From: 06/01/16 to: 06/12/16

Id	Last name	First name	Quantity
1	Parker	Susan	3
2	Smith	Peter	0
4	Flynn	Richard	0

Aquí vemos el caso del segundo listado, en el que estamos utilizando el data selector 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".

No mostramos aquí el ejemplo del web panel puesto que aún no estudiamos este objeto. Cuando lo hagamos, mostraremos dónde utilizar el Data Selector para filtrar la información que se mostrará en un grid.

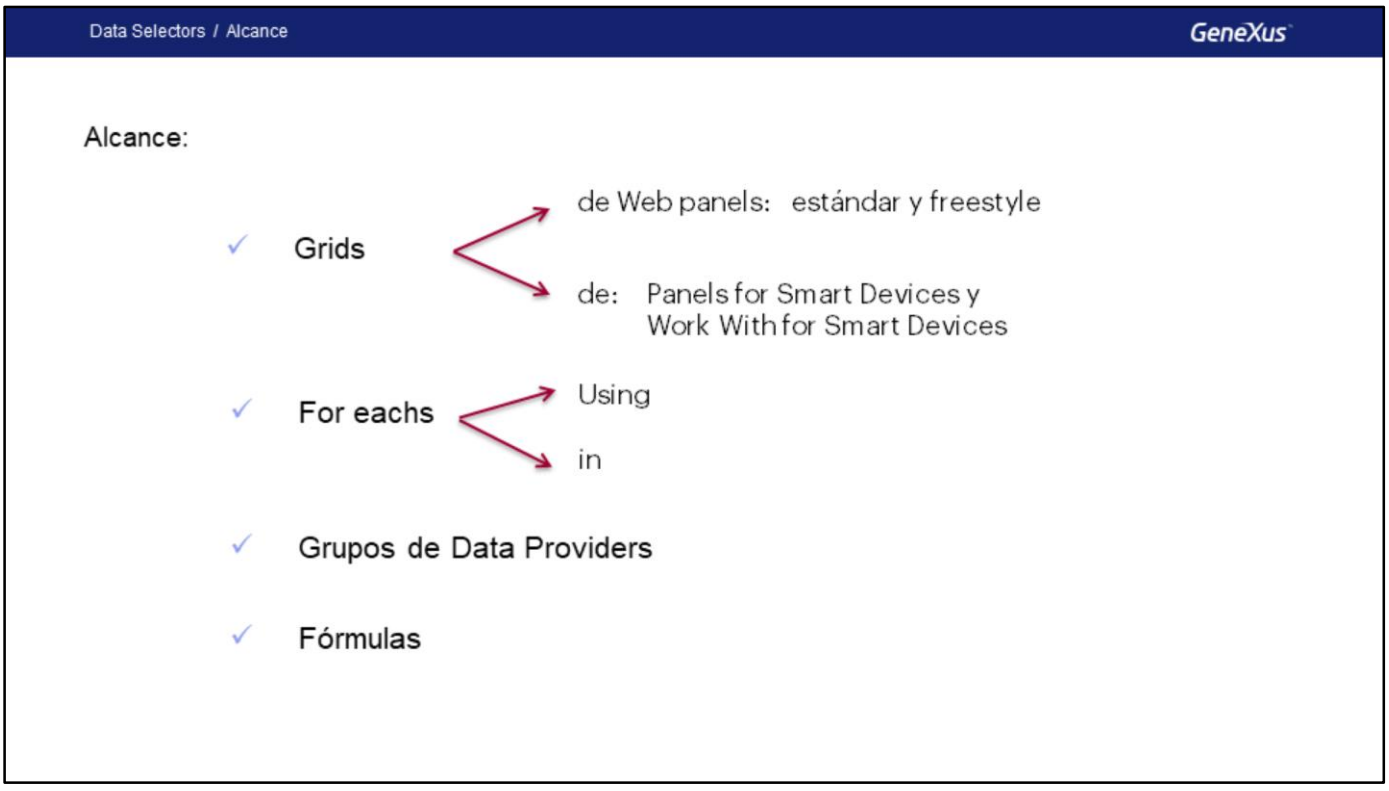


Un Data Selector especifica, en base a los parámetros recibidos, un conjunto de condiciones y ordenamientos para la información de manera centralizada, de modo de no tener que repetir las cláusulas order, where y defined by en cada lugar en que se necesiten.

Al indicarle al for each que use (using) un Data Selector, le estaremos diciendo que agregue sus órdenes y filtros 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.

La otra posibilidad mencionada, de utilizar el operador in para filtrar en un where, se muestra en la sintaxis pero ha sido dejada fuera de esta explicación. Si lo desea, puede aprender sobre esto en el siguiente link: http://wiki.genexus.com/commwiki/servlet/wiki?5312,Data+Selectors+in+For+Each+command,.

Alcance de los Data Selectors



El Data selector es un objeto para almacenar un conjunto de parámetros, conditions, orders y definedby, 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.

Hasta aquí solamente conocemos los for eachs y las fórmulas. Luego estudiaremos los grids en paneles Web y Smart Devices y los Data Providers.

Puede encontrar más documentación sobre Data selectors en nuestro wiki:
http://wiki.genexus.com/commwiki/servlet/wiki?5271,Category%3AData+Selector+object

GeneXus™

The power of doing.

Videos	training.genexus.com
Documentation	wiki.genexus.com
Certifications	training.genexus.com/certifications