

# Objeto Data Selector



GeneXus<sup>®</sup>



## Data Selectors - Concepto

- Objeto GeneXus: permite almacenar un conjunto de orders, filtros y un defined by, con el objetivo de **reutilizar navegaciones** (será "referenciado" desde distintas consultas y cálculos).

### Ejemplo:

Customer

```
{  
  CustomerId*  
  CustomerName  
  CustomerAddress  
  CustomerGender  
  CustomerStatus  
}
```

| Domain: Status  |  |
|-----------------|--|
| Name            | Status   |
| Description     | Status   |
| Empty as null   | Yes  |
| Type Definition |  |
| Based on        | (none)   |
| Data Type       | <b>Character</b>   |
| Length          | <b>1</b>   |
| Enum values     | Active, Active, A; OnHold, On Hold, H; Closed, Closed, C ... |

En varios objetos, se repite la necesidad de consultar los clientes activos ( *CustomerStatus= Status.Active* )

### Por ejemplo:

- Procedimiento que imprime clientes activos, agrupados por sexo.
- Procedimiento que aplica un descuento, sólo para clientes activos.
- Web Panel que muestra el total de ventas por cliente, sólo de clientes activos.

CREAMOS DATA SELECTOR CON DICHO FILTRO (+ ORDER + DEFINED BY)...  
Y LO REFERENCIAMOS DESDE LOS LUGARES NECESARIOS

GeneXus<sup>®</sup>

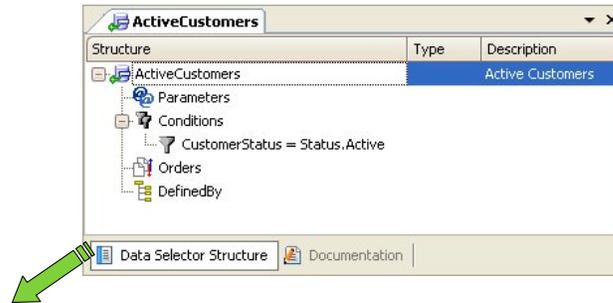
El atributo CustomerStatus está basado en un dominio enumerado, Status, que tiene 3 valores:

- Active
- On Hold
- Closed

Los clientes activos serán aquellos que tengan el valor Status.Active.

## Data Selectors - Definición

- Crear objeto de tipo "Data Selector", darle un nombre y definirlo:



La estructura consta de:

- **Parameters:** Parámetros recibidos desde la "invocación" del Data Selector, para ser usados en conditions.
- **Conditions:** Condiciones para filtrar los datos a ser recuperados.
- **Orders:** Orders condicionales (cada order con sus atributos separados por coma y when opcional).
- **Defined by:** Atributo o lista de atributos que colaboran en la definición de la tabla base final.

**GeneXus**<sup>®</sup>

## Data Selectors - Ejemplo

**Primer ejemplo planteado (Cont.):** En varios objetos de una KB, se repite la necesidad de consultar a los clientes activos

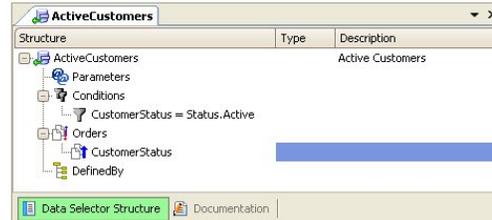
### Ejemplo:

```
Customer
{
  CustomerId*
  CustomerName
  CustomerAddress
  CustomerGender
  CustomerStatus
}
```

*CustomerStatus =  
Status.Active*



### Definimos Data Selector: "ActiveCustomers":



Luego de definido un Data Selector, puede ser referenciado en:

- Comandos For Each
- Grupos de Data Providers
- Fórmulas Aggregate
- Grids en Web Panels

**GeneXus<sup>x</sup>**

La forma de referenciar a un Data Selector dependerá desde cuál de los casos se desee invocar. Veremos la sintaxis para cada caso.



## Data Selectors - Beneficios

- **Ahorro y reutilización de código:** Se realiza cierta definición una vez y se reutiliza en varias consultas y cálculos en la KB.
- **Facilitan el mantenimiento:** Se cambia la definición en un único lugar y el cambio aplica automáticamente a todo lugar de la KB en el que se utilice.
- **Optimizan el conocimiento:** Facilitan el entrenamiento de personal nuevo para un proyecto, gracias al encapsulamiento que proveen.

GeneXus<sup>®</sup>



## Data Selectors – Usos

En For Each...

### Dos maneras de uso:

- Con cláusula **USING**
- Con operador **IN** en cláusula **WHERE**

### Importante:

- Un Data Selector siempre tiene tabla base.
- Según la forma de uso, los atributos presentes en la definición del Data Selector intervienen o no en la determinación de la tabla base del For Each:

- Con cláusula **USING**: interviene en la determinación de la tabla base.
- Con operador **IN**: no interviene en la determinación de la tabla base.

GeneXus<sup>®</sup>



## Data Selectors – Usos

En For Each...

### For Each con **cláusula USING**

Sintaxis: `For Each USING DataSelectorName([[parm1 [,parm2 [, ..] ]])  
.....  
EndFor`

Ejemplo 1: `For Each USING ActiveCustomers()  
.....  
EndFor`

Ejemplo 2: `For Each USING ActiveCustomers()  
Where CountryName = "Uruguay"  
...  
EndFor`

Clientes activos de Uruguay

Esto se expande a:  
`For Each  
Where CustomerStatus = Status.Active  
Where CountryName = "Uruguay"  
...  
EndFor`

ATRIBUTOS DEL DATA SELECTOR PARTICIPAN EN DETERMINACIÓN DE TABLA BASE DEL FOR EACH

GeneXus<sup>x</sup>

Nota:

La performance será la misma si se escriben 2 where en el For Each que si se referencia un Data Selector en el For Each (que tiene definida una condition) + un where en el For Each.

GeneXus siempre tratará de optimizar el código generado, sin embargo no más por utilizar el concepto de Data Selectors , ya que su objetivo como venimos explicando, es: centralizar el conocimiento reusable en una sola definición y luego referenciar esa definición en todas las consultas y cálculos de la KB que aplique.

Dado que en estos ejemplos presentados, se está referenciando en For Each al Data Selector con cláusula USING, los atributos presentes en la definición del Data Selector intervienen en la determinación de la tabla base del For Each. Por esto, en ambos ejemplos, la tabla base del For each es: CUSTOMER.



## Data Selectors – Usos

En For Each...

For Each con **operador IN** (en cláusulas WHERE)

El atributo que antecede al operador **IN** debe pertenecer a la tabla extendida de la tabla base del Data Selector

Sintaxis:

```
For Each
Where <attribute> IN DataSelectorName([[parm1 [,parm2 [, .. ] ]])
...
```

Ejemplo:

```
For Each
Where CustomerId IN InvoicesByDate(&FromDate, &ToDate)
...
```

CONSISTE EN UNA CONSULTA APARTE / INDEPENDIENTE DEL FOR EACH

Definición del Data Selector "InvoicesByDate":

| Structure                                     | Type | Description      |
|---|------|------------------|
| InvoicesByDate                                |      | Invoices By Date |
| Parameters                                    |      |                  |
| FromDate                                      | Date | From Date        |
| ToDate  | Date | To Date          |
| Conditions                                    |      |                  |
| InvoiceDates=FromDate and InvoiceDate<=ToDate |      |                  |
| Orders  |      |                  |
| InvoiceDate                                   |      |                  |
| Definibility                                  |      |                  |

TABLA BASE: INVOICE

• SE GENERA UN SELECT PARA EL DATA SELECTOR, INDEPENDIENTE DEL SELECT QUE SE GENERA PARA EL FOR EACH

• ATRIBUTOS DEL DATA SELECTOR **NO PARTICIPAN** EN DETERMINACIÓN DE TABLA BASE DEL FOR EACH

GeneXus<sup>®</sup>

En este ejemplo:

- La tabla base del Data Selector es: INVOICE
- El atributo que antecede al operador IN pertenece a la tabla extendida de INVOICE (CustomerId)... y la consulta asociada al Data Selector devolverá una lista de valores correspondientes a dicho atributo (devuelve "una lista de clientes" que tienen facturas en el rango de fechas dado).
- El For Each tiene su tabla base determinada por los atributos incluidos en el For Each y sin tener en cuenta a la información almacenada en el Data Selector.
- El For Each navegará su tabla base y extendida, filtrando los registros que contengan a alguno de los clientes de la lista devuelta por el Data Selector.



## Data Selectors – Usos En Fórmulas Aggregate...

- Es posible utilizar Data Selectors en fórmulas Aggregate **tanto globales como locales**.

• Sintaxis: **NombreFormulaAggregate** '( <Expresión> ',' [<CondiciónAgg>] ',' [<ValorDef>] )'

Find | Max | Min | Sum | Count | Average

Expresión a ser buscada, maximizada, minimizada, sumada o promediada

[<CondiciónBúsqueda>] ['USING' <DataSelector> '( <Parámetro1> ',' <Parámetro2> ',' <Parámetro3> )']

Valor a devolver si no hay registros que cumplan con <CondiciónAgg>

- Ejemplo:

atributo fórmula →  
cant. de clientes  
activos del país

| Name                   | Type          | Description              | Formula                           | Nullable |
|------------------------|---------------|--------------------------|-----------------------------------|----------|
| Country                | Country       | Country                  |                                   |          |
| CountryId              | Id            | Country Id               |                                   | No       |
| CountryName            | Name          | Country Name             |                                   | No       |
| CountryActiveCustomers | Numerico(4,0) | Country Active Customers | count(CustomerName,USING Activ... |          |

Formula Editor

count(CustomerName, USING ActiveCustomers ( ))

GeneXus<sup>®</sup>

### Consideraciones acerca de la sintaxis de las fórmulas Aggregate:

Como se detalló en la diapositiva, en la sintaxis de las fórmulas Aggregate la expresión corresponde a la expresión a ser buscada, maximizada, minimizada, sumada o promediada. Puede ser un atributo (almacenado o fórmula) o bien una expresión que involucre atributos, constantes, variables.

Sin embargo hay que tener en cuenta que:

- Variables solo pueden involucrarse en fórmulas locales.
- Para Sum y Average, el resultado de Expresión debe ser un valor numérico.
- Únicamente para el caso de Count, no vale una Expresión sino un atributo.

## Data Selectors – Usos

### Consideraciones

- En los parámetros del Data Selector pueden incluirse tanto variables como atributos.
  - Incluir atributos implica que se filtre por igualdad por ellos.
- Un mismo Data Selector definido, tendrá cierto comportamiento u otro y participará o no en la determinación de la tabla base de la consulta dependiendo de cómo se haya “invocado” en el contexto que sea usado.

GeneXus<sup>®</sup>