

Fórmulas

GeneXus 16

average(atributo₇)

atributo₁ + atributo₂

atributo₃ - atributo₄

fórmulas

count(atributo₆)

sum(atributo₅)

max(...)

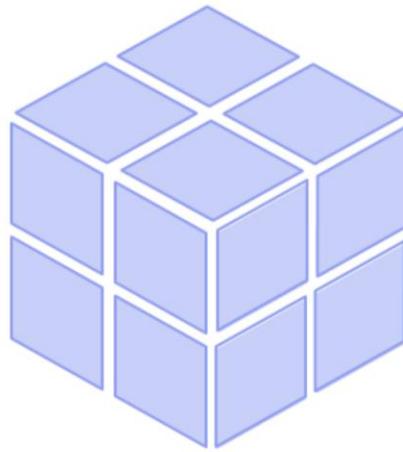
min(...)

Muchas veces necesitamos que la aplicación efectúe un cálculo que involucra valores de determinados atributos, constantes, y/o funciones.

Para estos casos, GeneXus nos provee las **Fórmulas**

Fórmulas Globales

Fórmulas Locales (o inline)



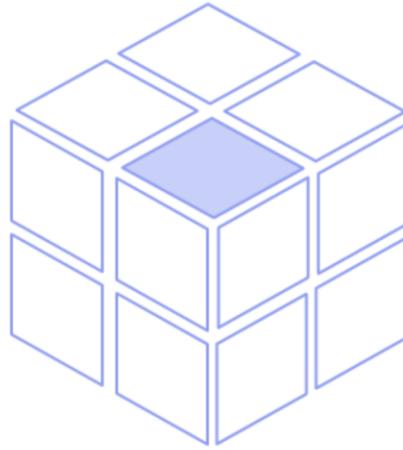
Base de
Conocimiento

Contamos con dos maneras de definir fórmulas:

DE FORMA GLOBAL: y así el cálculo se conocerá en toda la Base de Conocimiento,

Fórmulas **Globales**

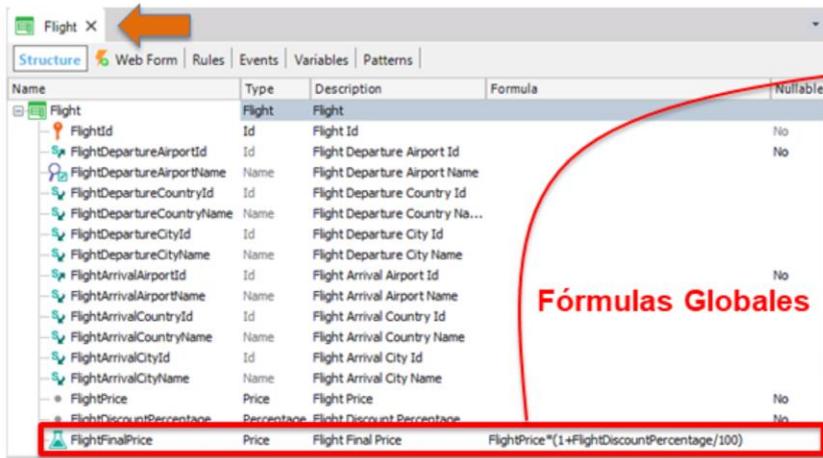
Fórmulas **Locales** (o inline)



Base de
Conocimiento

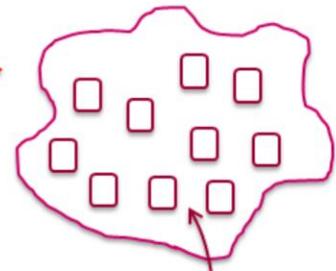
O DE FORMA LOCAL o INLINE: en este caso el cálculo se conocerá solamente en el objeto en el que se ha definido.

Dos tipos de Fórmulas



Name	Type	Description	Formula	Nullable
Flight	Flight	Flight		
FlightId	Id	Flight Id		No
FlightDepartureAirportId	Id	Flight Departure Airport Id		No
FlightDepartureAirportName	Name	Flight Departure Airport Name		
FlightDepartureCountryId	Id	Flight Departure Country Id		
FlightDepartureCountryName	Name	Flight Departure Country Na...		
FlightDepartureCityId	Id	Flight Departure City Id		
FlightDepartureCityName	Name	Flight Departure City Name		
FlightArrivalAirportId	Id	Flight Arrival Airport Id		No
FlightArrivalAirportName	Name	Flight Arrival Airport Name		
FlightArrivalCountryId	Id	Flight Arrival Country Id		
FlightArrivalCountryName	Name	Flight Arrival Country Name		
FlightArrivalCityId	Id	Flight Arrival City Id		
FlightArrivalCityName	Name	Flight Arrival City Name		
FlightPrice	Price	Flight Price		No
FlightDiscountPercentage	Percentage	Flight Discount Percentage		No
FlightFinalPrice	Price	Flight Final Price	FlightPrice*(1+FlightDiscountPercentage/100)	

Base de Conocimiento



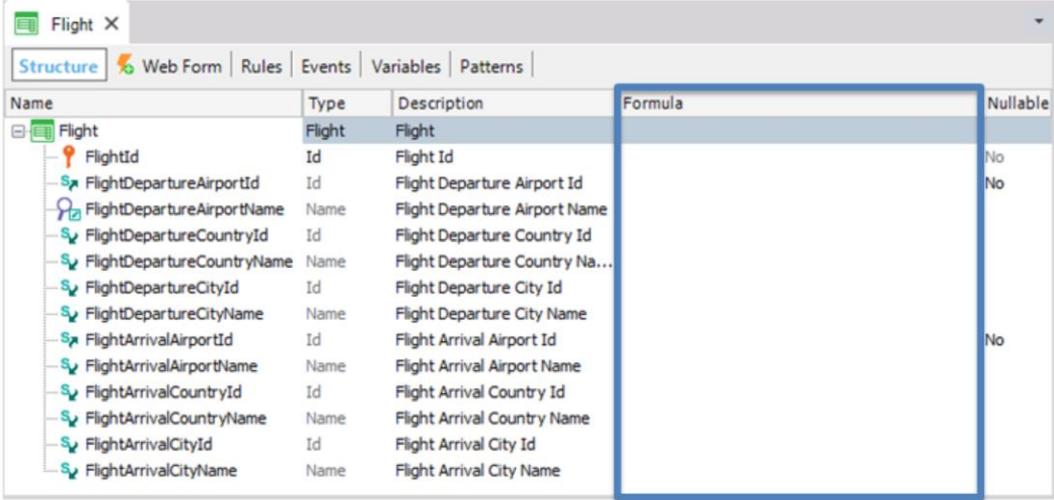
Fórmulas Globales

Fórmulas Inline (o locales)

```
1 Print header
2 For each Country
3   &AttractionQty = Count(AttractionName)
4   print Country
5 -endfor
6
```

Fórmula Global

Es un cálculo que se define asociado a un atributo, de ahí en más será "virtual".



Name	Type	Description	Formula	Nullable
Flight	Flight	Flight		
FlightId	Id	Flight Id		No
FlightDepartureAirportId	Id	Flight Departure Airport Id		No
FlightDepartureAirportName	Name	Flight Departure Airport Name		
FlightDepartureCountryId	Id	Flight Departure Country Id		
FlightDepartureCountryName	Name	Flight Departure Country Na...		
FlightDepartureCityId	Id	Flight Departure City Id		
FlightDepartureCityName	Name	Flight Departure City Name		
FlightArrivalAirportId	Id	Flight Arrival Airport Id		No
FlightArrivalAirportName	Name	Flight Arrival Airport Name		
FlightArrivalCountryId	Id	Flight Arrival Country Id		
FlightArrivalCountryName	Name	Flight Arrival Country Name		
FlightArrivalCityId	Id	Flight Arrival City Id		
FlightArrivalCityName	Name	Flight Arrival City Name		

Empecemos por aprender **qué es una fórmula global** y cómo se define.

Una fórmula global es un cálculo que definimos asociado a un atributo.

Observemos que las estructuras de las transacciones contienen una columna de título "Formula".

Si a un atributo, se le define un cálculo en esta columna, **le estaremos diciendo a GeneXus que dicho atributo es virtual**, es decir que no deberá crearse físicamente como campo en la tabla asociada, ya que el valor del atributo se obtendrá efectuando el cálculo que indicamos.

Agregando una fórmula global

Name	Type	Description	Formula	Nullable
Flight	Flight	Flight		
FlightId	Id	Flight Id		No
FlightDepartureAirportId	Id	Flight Departure Airport Id		No
FlightDepartureAirportName	Name	Flight Departure Airport Name		
FlightDepartureCountryId	Id	Flight Departure Country Id		
FlightDepartureCountryName	Name	Flight Departure Country Na...		
FlightDepartureCityId	Id	Flight Departure City Id		
FlightDepartureCityName	Name	Flight Departure City Name		
FlightArrivalAirportId	Id	Flight Arrival Airport Id		No
FlightArrivalAirportName	Name	Flight Arrival Airport Name		
FlightArrivalCountryId	Id	Flight Arrival Country Id		
FlightArrivalCountryName	Name	Flight Arrival Country Name		
FlightArrivalCityId	Id	Flight Arrival City Id		
FlightArrivalCityName	Name	Flight Arrival City Name		
FlightPrice	Price	Flight Price		No
FlightDiscountPercentage	Percentage	Flight Discount Percentage		No

Veamos esto con un ejemplo.

En primer lugar vamos a definir un nuevo atributo en la transacción Flight, con el fin de almacenar el **precio de cada vuelo**

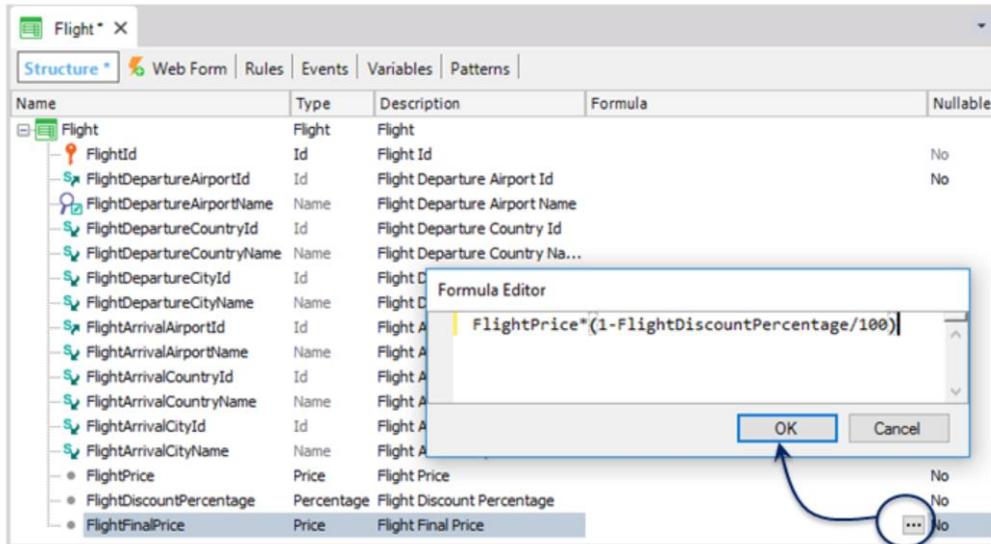
Así que definimos el atributo FlightPrice.

Y creamos el dominio Price.

Grabamos.

Ahora vamos a definir otro atributo nuevo en la misma transacción, **para almacenar el descuento que tiene cada vuelo**. Lo llamamos: FlightDiscountPercentage y su tipo de datos será un dominio también llamado Percentage, numérico de largo 3.

Agregando una fórmula global



Y por último, vamos a definir otro atributo más, de nombre FlightFinalPrice, basado en el dominio Price, **con el objetivo, ahora sí, de definir a este atributo como una fórmula global.**

Para eso, en la columna "Fórmula" de este atributo, vamos a definir el cálculo necesario **para que se ejecute siempre y se disponga en este atributo "del precio actual del vuelo"**, es decir, del precio de haberle descontado a FlightPrice el porcentaje de descuento almacenado en FlightDiscountPercentage.

Así que en esta columna fórmula, vamos a escribir el cálculo correspondiente.

Observar que en esta ventana se indica solamente el cálculo, no hay que escribir la asignación.

Análisis de Impacto

Database needs to be reorganized.

This report describes Database changes and how they will be handled by reorganization programs.
Please select Reorganize to proceed or Cancel.

Reorganize Cancel

Pattern:

Flight

Table Flight specification

Table name: [Flight](#)

Flight needs conversion

Warnings

- ⚠️ rqz0007 Attribute [FlightPrice](#) does not allow nulls and does not have an Initial Value. An empty default value will be used
- ⚠️ rqz0007 Attribute [FlightDiscountPercentage](#) does not allow nulls and does not have an Initial Value. An empty default value will be used

Table Structure

Attribute	Definition	Previous values	Table
FlightId	Numeric (4)Not null		Flight
FlightArrivalAirportId	Numeric (4)Not null		Flight
FlightDepartureAirportId	Numeric (4)Not null		Flight
New FlightPrice	Numeric (9,2)Not null	0	
New FlightDiscountPercentage	Numeric (3)Not null	0	

Indexes

Name	Definition	Composition
IFLIGHT	primary key Clustered	FlightId
IFLIGHT2	duplicate	FlightArrivalAirportId
IFLIGHT1	duplicate	FlightDepartureAirportId

Foreign key constraints

Referenced table	Attributes

0 Errors 1 Warnings 0 Success

Ahora presionemos F5, que automáticamente graba todo lo pendiente... y veamos lo que sucede.

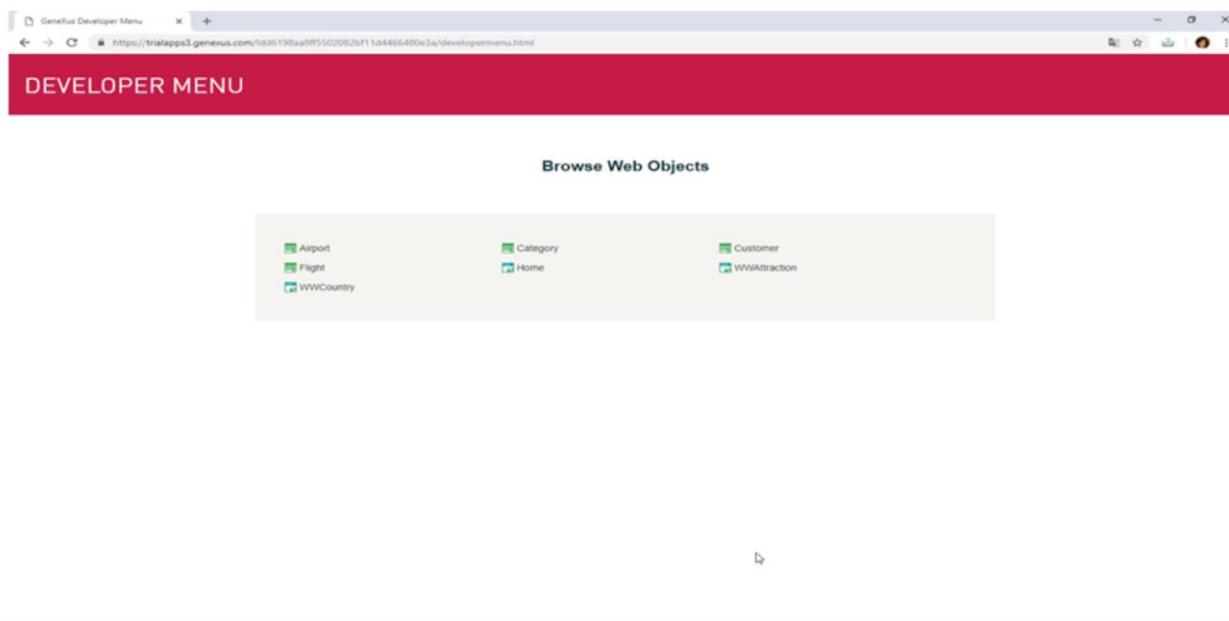
Vemos que en la tabla física Flight se están creando solamente 2 campos, a pesar de que hemos definido **tres atributos nuevos en la estructura de la transacción.**

Por el hecho de tener una definición en la columna fórmula, este atributo no se agrega en la tabla física.

Debido a que el atributo está definido en la base de conocimiento con una fórmula asociada, GeneXus sabe calcular su valor y en todo objeto en el cual ese atributo esté presente, se efectuará el cálculo y se mostrará el resultado.

Reorganizamos... .. y vemos la aplicación en ejecución.

DEMO: ejecutando la aplicación con una fórmula global...



[DEMO: <https://youtu.be/oiOrfa210xw>]

Ejecutamos la transacción Flight, consultamos el vuelo número 1 y vemos en este form, los tres nuevos atributos que definimos:

- el precio del vuelo, habilitado para que lo ingresemos
- el porcentaje de descuento, habilitado también para que lo ingresemos
- y el precio final, deshabilitado porque es el atributo que definimos como fórmula, y su valor no se ingresa, sino que se calculará y se desplegará.

Todo atributo que definamos como fórmula global será un dato de lectura y no será posible digitar valor para él. Esto se debe a que el atributo obtiene su valor del cálculo asociado, el cual se ejecuta cada vez que el atributo se utiliza.

Por esa razón no hay un campo en la tabla física para almacenar el valor de este atributo, así que no tiene sentido que sea editable.

Vamos a ingresar entonces un precio para este vuelo, un porcentaje de descuento: vamos a poner un 10 %.

Y al salir del campo, vemos que inmediatamente se ejecuta la fórmula, mostrándose el precio final del vuelo con el descuento aplicado.

Ejecutando la aplicación con una fórmula global...

FlightId 1

The image shows a web browser window displaying a flight application. The browser address bar shows the URL: `trialapps3.genexus.com/id8562ac14c97c0fe8095a5c284d688e07/flight.aspx`. The main form contains the following data:

Arrival Country Id	2
Arrival Country Name	France
Arrival City Id	1
Arrival City Name	Paris
Price	0.00
Discount Percentage	0
Final Price	0.00

At the bottom of the form are three buttons: CONFIRM, CANCEL, and DELETE.

A modal dialog is overlaid on the form, showing the following data:

Arrival City Name	Paris
Price	1500.00
Discount Percentage	10
Final Price	1350.00

The modal dialog also has three buttons at the bottom: CONFIRM, CANCEL, and DELETE.

Usando atributos de tabla extendida en una fórmula global...

Cree la transacción Airline

Agregue los atributos de Airline a Flight

se determina el precio del vuelo basado en el descuento de la aerolínea:

Name	Type	Description	Formula	Nullable
Airline				
AirlineId	Id			No
AirlineName	Name			No
AirlineDiscountPercentage	Percentage			No
Flight				
FlightId	Id	Flight Id		No
FlightDepartureAirportId		Flight Arrival Country Id		No
FlightDepartureAirportName	Name	Flight Arrival Country Name		No
FlightDepartureCountryId	Id	Flight Arrival City Id		No
FlightDepartureCountryName	Name	Flight Arrival City Name		No
FlightDepartureCityId	Id	Flight Price		No
FlightDepartureCityName	Name	Flight Discount Percentage		No
FlightArrivalAirportId	Percentage	Airline Id		No
FlightArrivalAirportName	Id	Airline Name		No
FlightArrivalCountryId	Name	Airline Discount Percentage		No
FlightArrivalCountryName	Percentage	Flight Final Price	FlightPrice*(1-AirlineDiscountPercentage/100);	Yes
FlightArrivalCityId	Id			
FlightArrivalCityName	Name			
FlightPrice	Price			
FlightDiscountPercentage	Percentage			
AirlineId	Id			
AirlineName	Name			
AirlineDiscountPercentage	Percentage			
FlightFinalPrice	Price	Flight Final Price	FlightPrice*(1-AirlineDiscountPercentage/100);	Yes

Volvamos a GeneXus.

Hemos definido así un atributo **fórmula global**.

Solamente los atributos pueden ser definidos como fórmulas globales de la forma que vimos, haciendo uso de la columna Fórmula en la transacción.

Algo importante a saber, es que si bien en el cálculo del ejemplo, hemos involucrado solamente atributos de la propia tabla asociada a la transacción, es decir, su tabla base, es **posible involucrar también atributos de la tabla extendida**.

Vamos a verlo.

Vamos a crear una nueva transacción de nombre Airline para registrar las aerolíneas.

Definimos

- AirlineId
- AirlineName y...
- AirlineDiscountPercentage, para registrar el descuento que realiza la aerolínea para todos sus vuelos.

Salvamos.

Y ahora vamos a la transacción Flight, para asignarle a cada vuelo la aerolínea que lo opera.

Agregamos entonces el atributo AirlineId, el cual tendrá aquí el rol de llave foránea...y cambiamos el valor de su propiedad Nullable, al valor Yes ... para que de esta forma podamos no especificar en esta primera etapa, la aerolínea del vuelo, ya que aún no tenemos aerolíneas registradas.

Más adelante podemos volver a cambiar el valor de esta propiedad Nullable a No, para que sea obligatorio especificar la aerolínea en el momento de ingresar o modificar un vuelo.

Agregamos también los atributos AirlineName y AirlineDiscountPercentage para poder visualizar también estos datos en el form.

Ahora vamos a modificar la definición de nuestra fórmula, para que calcule el precio final del vuelo,

aplicándole el descuento genérico de la aerolínea, en lugar de aplicarle el descuento propio del vuelo.

Análisis de Impacto

Database needs to be reorganized.

This report describes Database changes and how they will be handled by reorganization programs.
Please select Reorganize to proceed or Cancel.

Pattern:

Airline
 Flight

Table Flight specification

Table name: Flight
Flight needs conversion

Table Structure

Attribute	Definition	Previous values	Takes value from
FlightId	Numeric (4)Not null		Flight. FlightId
FlightArrivalAirportId	Numeric (4)Not null		Flight. FlightArrivalAirportId
FlightDepartureAirportId	Numeric (4)Not null		Flight. FlightDepartureAirportId
FlightPrice	Numeric (9.2)Not null		Flight. FlightPrice
FlightDiscountPercentage	Numeric (3)Not null		Flight. FlightDiscountPercentage
New AirlineId	Numeric (4)		Null

Indexes

Name	Definition	Composition
IFLIGHT	primary key Clustered	FlightId
IFLIGHT2	duplicate	FlightArrivalAirportId
IFLIGHT1	duplicate	FlightDepartureAirportId
New IFLIGHT3	duplicate	AirlineId

Foreign key constraints

Referenced table	Attributes
Airport	FlightArrivalAirportId
Airport	FlightDepartureAirportId
New Airline	AirlineId

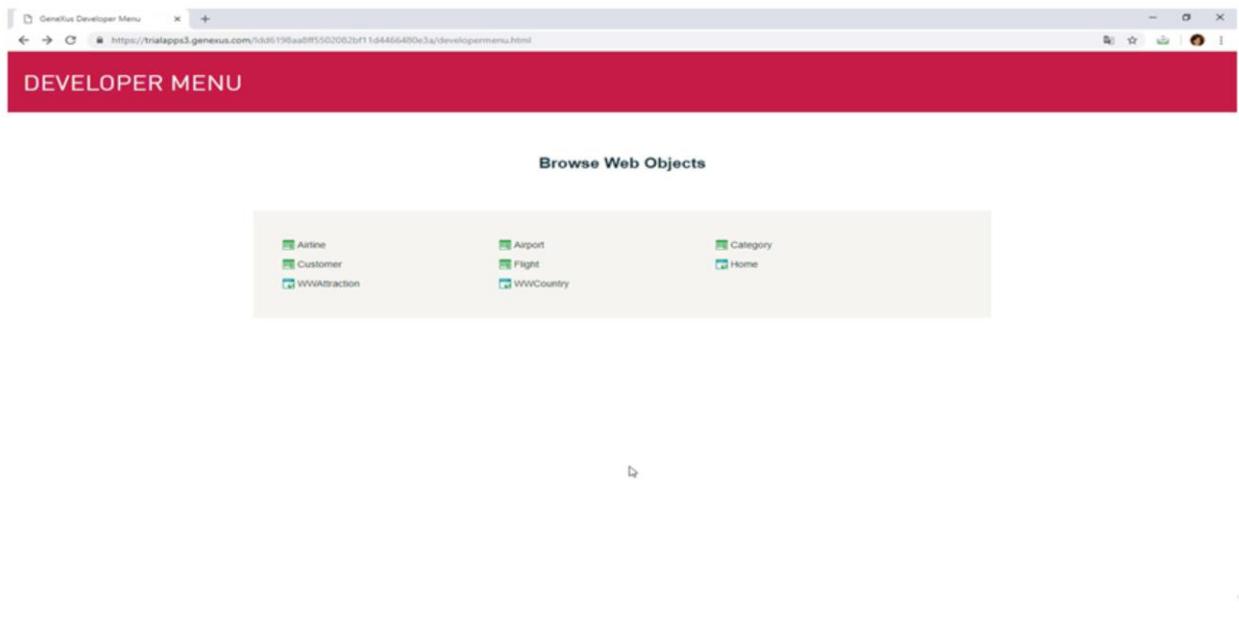
0 Errors 0 Warnings 2 Success

Y presionamos F5...

Vemos la propuesta de creación de la tabla física Airline con los tres atributos que definimos y que en la tabla Flight, se agregará la llave foránea Airlineld.

Procedamos entonces a reorganizar y a ejecutar...

DEMO: testeando la fórmula en tiempo de ejecución



[DEMO: https://youtu.be/VK8BR3faA_g]

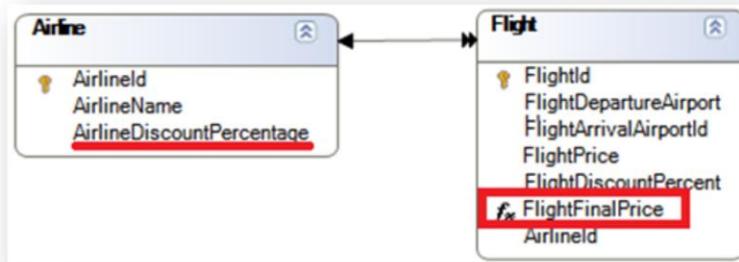
Ejecutamos la transacción Airline y vamos a ingresar una aerolínea, de nombre TAM con un descuento del 30%.

Ahora vamos a asignar esta aerolínea a un vuelo.

Así que abrimos la transacción Flight, nuestro vuelo número 1, y vamos a asociarle la aerolínea número 1...

Vemos que se calculó el nuevo precio final del vuelo, que es un atributo fórmula global.

Testeando la fórmula en tiempo de ejecución



que ahora involucra al porcentaje de descuento de la aerolínea, que es un atributo de la tabla extendida de la tabla base Flight.

Agregando condiciones a la fórmula

$$\text{Atributo} = \begin{matrix} fx \\ \left[\begin{array}{l} \text{expresión}_1 \text{ if } \text{condición}_1; \\ \text{expresión}_2 \text{ if } \text{condición}_2; \\ \dots \\ \text{expresión}_n \text{ if } \text{condición}_n; \end{array} \right. \end{matrix}$$

Bien. Algo que no hemos dicho aún es que las fórmulas pueden evaluar condiciones y que el resultado puede calcularse de diferentes formas dependiendo de si dichas condiciones son verdaderas o falsas.

Agregando condiciones a la fórmula

Name	Type	Description	Formula	Nullable
Flight	Flight	Flight		
FlightId	Id	Flight Id		No
FlightDepartureAirportId	Id	Flight Departure Airport Id		No
FlightDepartureAirportName	Name	Flight Departure Airport Name		
FlightDepartureCountryId	Id	Flight Departure Country Id		
FlightDepartureCountryName	Name	Flight Departure Country Name		
Formula Editor				
<code>FlightPrice*(1-AirlineDiscountPercentage/100) IF AirlineDiscountPercentage >= FlightDiscountPercentage; FlightPrice*(1-FlightDiscountPercentage/100) OTHERWISE</code>				
FlightArrivalCityName	Name	Flight Arrival City Name		
FlightPrice	Price	Flight Price		No
FlightDiscountPercentage	Percentage	Flight Discount Percentage		No
AirlineId	Id	Airline Id		Yes
AirlineName	Name	Airline Name		
AirlineDiscountPercentage	Percentage	Airline Discount Percentage		
FlightFinalPrice	Price	Flight Final Price	<code>FlightPrice*(1-AirlineDiscountPercentage/100) IF ...</code>	

El resultado será calculado de una forma u otra, dependiendo de qué condición es verdadera.

Veamos esto.

Vamos a hacer clic en este botón para editar la fórmula.

Y vamos a definir que se tenga en cuenta el porcentaje de descuento más alto para calcular el precio final del vuelo, y así poder hacer el mejor descuento posible.

Con esta definición, si la aerolínea tiene un descuento más alto para todos sus vuelos que el porcentaje de descuento que tiene el vuelo en sí, entonces consideramos para el cálculo el descuento de la aerolínea.

También agregamos una condición por defecto

$$\text{Atributo} = f_x \left[\begin{array}{l} \text{expresión}_1 \text{ if } \text{condición}_1; \\ \text{expresión}_2 \text{ if } \text{condición}_2; \\ \dots \\ \text{expresión}_n \text{ if } \text{condición}_n; \\ \text{expresión}_0 \text{ otherwise}; \end{array} \right. \downarrow$$

Y en caso contrario:

empleamos para el cálculo al porcentaje de descuento propio del vuelo.

Testing the new definition of the formula

Country Name	Brazil
City Id	2
City Name	Sao Paulo
Price	3679,00
Discount Percentage	50
Airline Id	1
Airline Name	TAM
Airline Discount Percentage	30
Final Price	1500,00

Observe que las fórmulas se escriben como expresiones, terminando en punto y coma. Para calcular la fórmula GeneXus se queda con la primera expresión para la que se cumpla la condición. Si ninguna se cumple y hay cláusula otherwise, utiliza esa.

Vamos a mover el atributo Final Price para el final, para que nos quede mejor presentada la información.

Presionamos F5.

Ejecutamos la transacción Flight y al primer vuelo le asignamos que su porcentaje de descuento sea mayor al porcentaje de descuento general de la aerolínea, por ejemplo vamos a asignar un 50%.

Salimos del campo y pasamos por el de la aerolínea, de modo que al salir, ya tiene todo para realizar el cálculo de la fórmula.

Vemos que el precio final del vuelo se calculó entonces teniendo en cuenta el mayor descuento.

Fórmulas horizontales

The image shows three instances of the GeneXus Formula Editor. The top editor shows a formula for calculating the final flight price based on discounts. The middle editor shows the same formula with a `Round` function applied to the result. The bottom editor shows a formula for extracting the month from a date. Arrows point from the formula editors to three data tables, illustrating how the formulas are applied to the data.

Formula Editor 1:

```
FlightPrice*(1-AirlineDiscountPercentage/100) IF AirlineDiscountPercentage >= FlightDiscountPercentage;
FlightPrice*(1-FlightDiscountPercentage/100) OTHERWISE
```

Formula Editor 2:

```
FlightPrice*(1-AirlineDiscountPercentage/100) IF AirlineDiscountPercentage >= FlightDiscountPercentage;
Round(
Round(Value: Numeric, Decimals: Numeric): Numeric) OTHERWISE;
```

Formula Editor 3:

```
FlightPrice*(1-AirlineDiscountPercentage/100) IF AirlineDiscountPercentage >= FlightDiscountPercentage;
Month(
Month(Value: Date/DateTime): Numeric)
```

Volvamos a GeneXus.

Como vimos, las fórmulas pueden contener varias líneas seguidas de IF y contener o no una última línea con OTHERWISE

para el caso que no se hayan cumplido ninguna de las condiciones anteriores.

A su vez, si bien en este ejemplo la forma de obtener cada resultado fue mediante un cálculo, también se pueden emplear funciones aplicadas a atributos o expresiones, como `Round`, para obtener un resultado redondeado o `Month`, para extraer el mes de una fecha, etc... incluso se puede llamar a un procedimiento definido por nosotros, que retorne un valor.

Estudiaremos los procedimientos luego.

A este tipo de fórmulas, que realizan un cálculo que se obtiene a partir de datos del registro en el que se está posicionado (uno sólo) y eventualmente de los registros asociados (por tabla extendida), solemos denominarlas fórmulas horizontales.

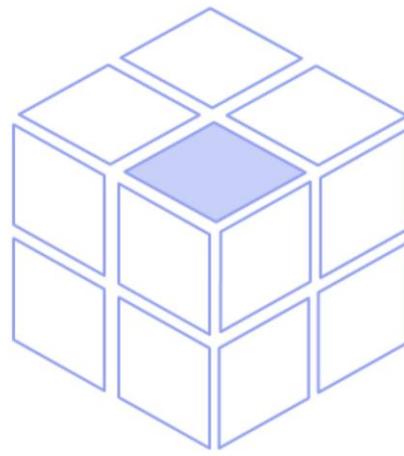
Fórmulas de Agregación

Fórmulas **Globales**

Atributo = fx

Fórmulas **Locales** (o inline)

&Variable = fx



Base de
Conocimiento

Bien. Veremos ahora otro tipo de fórmulas, llamadas: **fórmulas de Agregación**.

Si bien vamos a explicarlas definiendo ejemplos de fórmulas globales, es decir, que los cálculos respectivos los vamos a definir asociados a atributos -y por lo tanto no se van a crear como campos físicos-, lo que expliquemos vale también para ser asignado en otro contexto, a una variable por ejemplo, o localmente en determinada sección de un objeto (esto lo veremos en otro video).

Fórmulas de Agregación: Count, Sum, Average, etc.

Ejemplo: Cree un segundo nivel en la transacción Flight:

Name	Type	Formula	Nullable
Flight	Flight		
FlightId	Id		No
FlightDepartureAirportId	Id		No
FlightDepartureAirportName	Name		
FlightDepartureCountryId	Id		
FlightDepartureCountryName	Name		
FlightDepartureCityId	Id		
FlightDepartureCityName	Name		
FlightArrivalAirportId	Id		No
FlightArrivalAirportName	Name		
FlightArrivalCountryId	Id		
FlightArrivalCountryName	Name		
FlightArrivalCityId	Id		
FlightArrivalCityName	Name		
FlightPrice	Price		No
FlightDiscountPercentage	Percentage		No
AirlineId	Id		Yes
AirlineName	Name		
AirlineDiscountPercentage	Percentage		
FlightFinalPrice	Price	$FlightPrice * (1 - AirlineDiscountPercentage / 100)$ IF Air...	
Seat	Seat		
FlightSeatId	Id		No
FlightSeat.Location	Location		No

Definiremos este dominio con valores enumerados.

Bien, ahora vamos a crear un segundo nivel en la transacción Flight... y le ponemos como nombre: Seat

Tal como este nombre del nivel lo describe, lo vamos a utilizar para registrar los asientos que ofrece el vuelo y vamos a registrar para cada asiento, si se encuentra al lado de la ventana, o en el pasillo o está en el medio de estos. Luego vamos a querer conocer siempre la cantidad de asientos que ofrece el vuelo.

Digitamos el punto, y completamos el nombre del atributo: FlightSeatId.

Creamos ahora otro atributo más, que vamos a nombrar FlightSeat Location... que va a ser un caracter de 1.

Fórmulas de Agregación: Count, Sum, Average, etc.

Defina un dominio con valores enumerados:

The screenshot shows the GeneXus IDE interface. On the left, the 'Domains' tree is visible, with 'Location' selected. In the center, the 'Values Editor' dialog is open, showing a table with three rows of values: 'Window' (W), 'Middle' (M), and 'Aisle' (A). On the right, the 'Properties' window for the 'Location' domain is open, showing various attributes. The 'Enum Values' property is highlighted, showing the values 'Window, Window, W; M; A'.

Name	Description	Value
Window	Window	W
Middle	Middle	M
Aisle	Aisle	A

Name	Location
Description	Location
Empty as null	Yes
Class	Attribute
Module	Root Module
Qualified Name	Location
Object Visibility	Public

Property	Value
Based on	(none)
Data Type	Character
Length	1
Enum Values	Window, Window, W; M; A
Collection	False
Initial value	
Enable national l	No

Ahora editamos los dominios, para modificar una propiedad del dominio Location que acabamos de crear.

Ubicamos la propiedad **Enum Values**: y vamos a definir los tres valores que puede tomar este dominio:

- Window... o sea, ventana, y el carácter que se almacenará en este caso será la "W"
- Middle... o sea, al medio, y el valor que se almacenará en este caso será una "M"
- o Aisle... o sea, pasillo, y el valor que se almacenará en este caso será una "A".

Presionamos OK...

Fórmulas de Agregación: Count, Sum, Average, etc.

The screenshot shows the GeneXus web form editor for a transaction named 'Flight X'. The 'Web Form' tab is selected, displaying a form with several input fields. The 'Seat' section is highlighted with a blue border, and the 'Seat Location' dropdown menu is highlighted with a green oval.

.. y observemos el form de la transacción Flight. Vemos que se agregó una grilla para poder ingresar los asientos del vuelo, y que para cada asiento se puede indicar la ubicación del mismo mediante un control combo.

Este combo ofrece los valores "window" "middle" o "aisle", que son los valores posibles que hemos definido para el dominio del atributo FlightSeatLocation.

Fórmulas de Agregación: Count, Sum, Average, etc.

Cambie la clave del nivel Seat para representar mejor el asiento. Queremos identificarlo con un número + una letra de la A a la F.

The screenshot shows the GeneXus interface. On the left, the 'Flight' entity is expanded to show its attributes. The 'Seat' entity is also expanded, showing 'FlightSeatId' (Id), 'FlightSeatChar' (SeatChar), and 'FlightSeatLocation' (Location). The 'FlightSeatChar' attribute is highlighted. A formula 'FlightPrice*(1-Air...' is visible next to it. On the right, the 'Values Editor' dialog is open, showing a table with columns 'Name', 'Description', and 'Value'. The table contains the following data:

Name	Description	Value
A	A	A
B	B	B
C	C	C
D	D	D
E	E	E
F	F	F

Below the dialog, the text 'Dominio Enum' is written, with an arrow pointing to the 'Values Editor' dialog.

Antes de presionar F5, observemos algo en la definición del segundo nivel.

Si la llave se compone de FlightId más FlightSeatId, **para cada vuelo no podremos repetir los números de asiento**. Pero necesitamos que el número de asiento sí se repita, porque lo que lo identifica es ese número más una letra. Así, tendremos el asiento 1A, 1B, 1C, 2A, 2B, etc.

Así que vamos a agregar un nuevo atributo FlightSeatChar. Su tipo lo definimos de un dominio SeatChar, que será un character de 1 y haremos que este atributo sea parte de la llave para que puedan registrarse números iguales de asiento, con distinta letra.

Vamos a restringir las letras posibles a las que van de la A a la F y para eso vamos a editar el dominio SeatChar que acabamos de crear...

Ubicamos su propiedad Enum Values y vamos a definir los valores posibles.

En este caso coinciden los valores de las descripciones con los valores que se almacenan. Presionamos ok.

Fórmulas de Agregación: Count, Sum, Average, etc.

Agregue una nueva fórmula para contar la cantidad de asientos:

Name	Type	Formula	Nullable
Flight	Flight		
FlightId	Id		No
FlightDepartureAirportId	Id		No
FlightDepartureAirportName	Name		
FlightDepartureCountryId	Id		
FlightDepartureCountryName	Name		
FlightDepartureCityId	Id		
FlightDepartureCityName	Name		
FlightArrivalAirportId	Id		No
FlightArrivalAirportName	Name		
FlightArrivalCountryId	Id		
FlightArrivalCountryName	Name		
FlightArrivalCityId	Id		
FlightArrivalCityName	Name		
FlightPrice	Price		No
FlightDiscountPercentage	Percentage		No
AirlineId	Id		Yes
AirlineName	Name		
AirlineDiscountPercentage	Percentage		
FlightFinalPrice	Price	FlightPrice*(1-AirlineDiscountPercentage)	
FlightCapacity	Numeric(4,0)	Count(FlightSeat.Location)	No
Seat	Seat		
FlightSeatId	Id		No
FlightSeatChar	SeatChar		No
FlightSeatLocation	Location		No

Ahora, a efectos de conocer la capacidad de pasajeros que el vuelo permite, en base a la cantidad de asientos, vamos a definir un nuevo atributo en el primer nivel, y en su columna Formula, indicaremos el cálculo que consistirá en contar la cantidad de asientos que ofrece el vuelo...

Creamos entonces el atributo FlightCapacity y su tipo de datos será numérico de 4.

Ahora en su columna Formula vamos a escribir: Count... y entre paréntesis un atributo que elegiremos para que GeneXus interprete que queremos contar asientos. Para eso elegimos al atributo FlightSeatLocation que pertenece a la tabla de los asientos del vuelo, la tabla FlightSeat.

El atributo que se referencia dentro del paréntesis de la fórmula, le da la información a GeneXus de la tabla a navegar para realizar el cálculo.

Fórmulas de Agregación: Count, Sum, Average, etc.

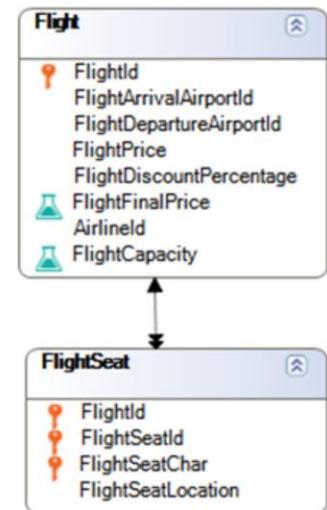
FlightCapacity = Count(FlightSeatLocation)

FLIGHT

FlightId	FlightDepartureAirportId	...	FlightPrice	AirlineId
1	1		1500	1
2	2	...	2500	2
3	1	...	1000	1
...

FLIGHTSEAT

FlightId	FlightSeatId	FlightSeatChar	FlightSeatLocation
1	1	A	Window
1	1	B	Aisle
1	2	A	Window
1	2	B	Aisle
1	3	C	Middle
2	1	A	Window
2	1	B	Middle
3
...



Dado un vuelo, queremos contar sus asientos (muchos), es decir los registros de la tabla FlightSeat relacionados al vuelo.

Al momento de calcularse la fórmula estaremos posicionados en un registro de la tabla FLIGHT que tiene una relación de 1 a Muchos con la tabla de los asientos, FLIGHTSEAT, donde están los asientos de TODOS los vuelos. Esa relación viene determinada por el atributo FlightId.

Observemos que al escribir Count(FlightSeatLocation), no estamos contando las ventanas o pasillos, sino **los registros** donde están estos asientos. Como estamos posicionados en un vuelo determinado, GeneXus va a contar solamente aquellos asientos correspondientes a **ese** vuelo.

Es decir, si GeneXus detecta relación entre la tabla asociada al atributo fórmula y la tabla que navegará la fórmula para realizar su cálculo **solamente tendrá en cuenta para realizar ese cálculo los registros relacionados...**

En caso de no encontrar relación, GeneXus realizará el cálculo sobre **todos los registros de la tabla navegada.**

Fórmulas de Agregación: Count, Sum, Average, etc.

Name	Type	Formula	Nullable
Flight	Flight		
FlightId	Id		No
FlightDepartureAirportId	Id		No
FlightDepartureAirportName	Name		
FlightDepartureCountryId	Id		
FlightDepartureCountryName	Name		
FlightDepartureCityId	Id		
FlightDepartureCityName	Name		
FlightArrivalAirportId	Id		No
FlightArrivalAirportName	Name		
FlightArrivalCountryId	Id		
FlightArrivalCountryName	Name		
FlightArrivalCityId	Id		
FlightArrivalCityName	Name		
FlightPrice	Price		No
FlightDiscountPercentage	Percentage		No
AirlineId	Id		Yes
AirlineName	Name		
AirlineDiscountPercentage	Percentage		
FlightFinalPrice	Price	FlightPrice*(1-AirlineDiscountPercentage/100) IF Airli...	
Seat	Seat		
FlightSeatId	Id		No
FlightSeatLocation	Location		No

Definiremos este dominio
con valores enumerados

Probemos esto en ejecución... Presionamos F5...

Vemos que se va a crear la tabla física FLIGHTSEAT asociada al 2do nivel de la transacción Flight, con los atributos y la llave que hemos definido... **y no se va a modificar la estructura de la tabla FLIGHT** dado que el atributo FlightCapacity no se creará físicamente, tal como esperábamos.

Estamos de acuerdo así, que procedemos a reorganizar...

DEMO



Browse Web Objects



[DEMO: <https://youtu.be/lv8t1CPDmkA>]

Ejecutamos la transacción Flight... consultamos nuestro vuelo número1 y vamos a registrarle algunos asientos:

- 1A - ventana
- 1B - en el medio
- 1C - pasillo
- 1D - ventana
- 1E - al medio
- 1F - pasillo

Mientras agregamos los asientos, observemos que la cuenta de los asientos se fue actualizando cada vez que vamos agregando un asiento al vuelo.

Vamos a agregar por último

- 2A - ventana ... y dejamos por acá...

Volvamos a GeneXus.

Fórmulas de Agregación: Count, Sum, Average, etc.

The screenshot displays the GeneXus IDE interface for the 'Flight' entity. The 'Structure' tab is active, showing a tree view of attributes and their types. The 'FlightCapacity' attribute is selected, and its formula is shown as 'count(FlightSeat.location)'. Other aggregation formulas are highlighted in blue boxes: 'Sum(Atr)', 'Average(Atr)', 'Max(...)', 'Min(...)', and 'Find(...)'. A large orange grid is overlaid on the right side of the interface.

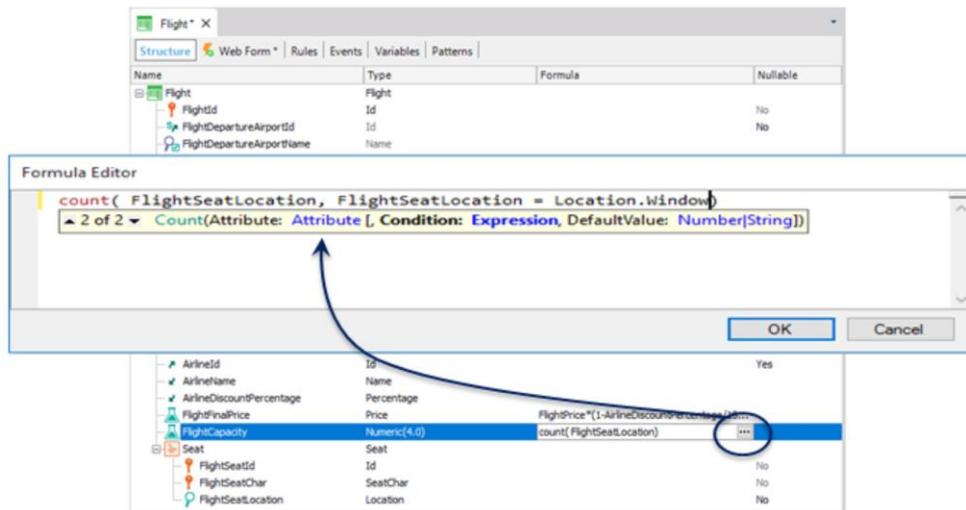
Name	Type	Formula
Flight	Flight	
FlightId	Id	
FlightDepartureAirportId	Id	
FlightDepartureAirportName	Name	
FlightDepartureCountryId	Id	
FlightDepartureCountryName	Name	
FlightDepartureCityId	Id	
FlightDepartureCityName	Name	
FlightArrivalAirportId	Id	
FlightArrivalAirportName	Name	
FlightArrivalCountryId	Id	
FlightArrivalCountryName	Name	
FlightArrivalCityId	Id	
FlightArrivalCityName	Name	
FlightPrice	Price	
FlightDiscountPercentage	Percentage	
AirlineId	Id	
AirlineName	Name	
AirlineDiscountPercentage	Percentage	
FlightFinalPrice	Price	FlightPrice*(1-AirlineDiscountPercentage/10...
FlightCapacity	Numeric(4,0)	count(FlightSeat.location)
Seat	Seat	
FlightSeatId	Id	
FlightSeatChar	SeatChar	
FlightSeatLocation	Location	No

Contamos también con otras fórmulas Aggregate, que realizan operaciones teniendo en cuenta varios registros.

Como por ejemplo: Sum, para sumar los valores del atributo que se especifique Average, para realizar el promedio de los valores del atributo que se especifique y otras, como Max, Min, Find.

Fórmulas de Agregación: Count, Sum, Average, etc.

Se pueden agregar condiciones para contar "ciertas" líneas:



Si lo que queremos contar no es solamente los asientos del vuelo en el que estamos posicionados, sino aquellos que además cumplan con alguna otra condición –por ejemplo los asientos ubicados junto a la ventana- contamos con la posibilidad de agregar esa condición a la fórmula. Así, dado que el atributo FlightSeatLocation es del dominio Location y el mismo tiene definidos 3 valores enumerados, la sintaxis para preguntar por el valor que tomó el atributo, es la siguiente:

nombre del dominio, punto, y el nombre asociado al valor que nos interesa filtrar en este caso Window.

Presionamos ok.

Fórmulas de Agregación: Count, Sum, Average, etc.

Final Price: 2100.00

Capacity: 3

`count(FlightSeatLocation, FlightSeatLocation = Location.Window)`

condición de filtro

También pueden tener una "condición de disparo"

Seat id	Seat Char	Seat Location
X 1	A	Window
X 1	B	Middle
X 1	C	Aisle
X 1	D	Window
X 1	E	Middle
X 1	F	Aisle
X 2	A	Window
[New row]	A	Window

Presionemos F5

Ejecutamos la transacción Flight, el registro número 1 y vemos que la capacidad ahora muestra la cantidad: 3, que corresponde a la cantidad de asientos ubicados al lado de 1 ventana, lo cual coincide con los ingresos que hicimos en la grilla de asientos.

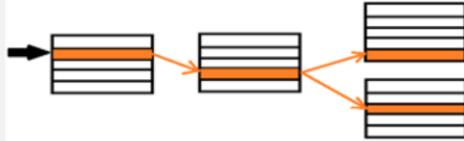
En definitiva, lo que hemos visto es que además de la condición implícita (cuando hay registros relacionados) es posible también contar, sumar, buscar, maximizar, promediar, en una palabra: **agregar, aquellos registros que cumplan con cierta condición explícita indicada por nosotros**. A esta condición usualmente la llamamos "condición de filtro" porque permite quedarnos únicamente con aquellos registros que nos interesen.

Por último debemos saber que como toda fórmula global, las fórmulas Aggregate también pueden tener "condición de disparo", es decir que solamente la fórmula se calcule cuando se cumple esa condición.

Resumiendo

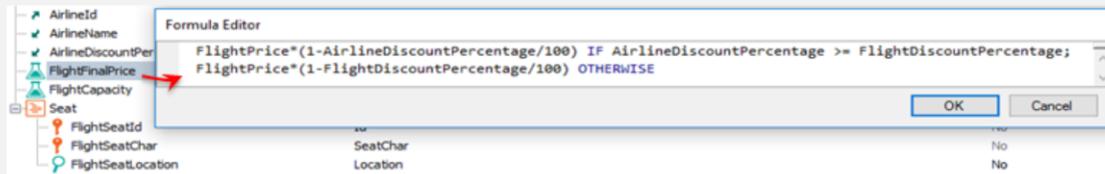
Horizontal:

- Para hacer un cálculo, acceden a un registro y ocasionalmente, a aquellos relacionados a través de una table extendida.



Attributo = $expresión_1$ **if** $condición_1$;
 $expresión_2$ **if** $condición_2$;
 ...
 $expresión_n$ **if** $condición_n$;
 $expresión_o$ **otherwise**;

- Ejemplo: FlightFinalPrice



En resumen: vimos dos tipos de fórmulas:

las **horizontales**, que para realizar un cálculo acceden a un registro y eventualmente a los relacionados por tabla extendida.

Era el caso de FlightFinalPrice:

Donde estos atributos eran de la tabla FLIGHT y estos otros de la tabla Airline.

Como vimos en el ejemplo, era posible definir que un atributo fórmula se calculara de distintas maneras de acuerdo al valor de una condición.

Resumiendo

Aggregate:

- Para hacer un cálculo necesitan navegar varios registros de la misma tabla.

FlightId	FlightDepartureAirportId	...
1	1	...
2	3	...
3	1	...
...



FlightId	FlightSeatId	FlightSeatLocation	
1	1	A Window	←
1	1	B Aisle	
1	2	A Window	←
1	2	B Aisle	
1	3	C Middle	
2	1	A Window	
2	1	B Middle	
3	
...	

- Ejemplo: FlightCapacity

Formula Editor

```
count( FlightSeatLocation, FlightSeatLocation = Location.Window)
```

OK Cancel

y por otro lado teníamos las fórmulas **aggregate**, que para realizar su cálculo deben navegar muchos registros de una misma tabla.

Era el caso de FlightCapacity que desde la tabla FLIGHT asociada al atributo fórmula, lanzaba un cálculo sobre la tabla FLIGHTSEAT, donde se encuentra el atributo FlightSeatLocation.

En este caso, como el atributo fórmula está asociado a una tabla, Flight, que tiene una relación de 1 a muchos con la tabla sobre la que se realizará el Count, se contarán únicamente los registros relacionados. Si no hubiera relación, se contarían todos. Pero además, como indicamos condiciones para que los registros sean contados de los registros relacionados se contarán únicamente los que además cumplan esa condición.

Resumiendo

Atributo = *Count*(Atributo, condición, Valor por defecto) **if** condición;

Sum(Expresión, condición, Valor por defecto) **if** condición;

Find(Expresión, condición, Valor por defecto) **if** condición;

...

La condición de filtro es el segundo parámetro de la fórmula, y como tercer parámetro vemos que podemos indicar un valor por defecto, esto es, el valor que devolverá la fórmula si no se encuentra ningún registro para contar, sumar, etcétera.

Al igual que en el caso de las fórmulas horizontales, las aggregate también pueden definirse condicionalmente.

Resumiendo

Atributo = 2 + **Count**(Atributo, condición, Valor por defecto) *

Sum(Expresión, condición, Valor por defecto) **if** condición;

Atr₁ + **Atr₂** * **Atr₃** **otherwise**;

...

No lo veremos en este curso, pero de hecho pueden combinarse las fórmulas horizontales con las aggregate, logrando un enorme grado de expresividad en los cálculos.

Para finalizar quitemos la condición de filtro al atributo FlightCapacity...

Y ahora enviemos las nuevas definiciones a GeneXus Server.

Resumiendo

- Aggregate:
 - Sum
 - Average
 - Max
 - Min
 - Find
- El atributo al que se hace referencia dentro de los paréntesis de la fórmula proporciona información a GeneXus sobre la tabla a navegar para realizar el cálculo.
- Si GeneXus encuentra una relación entre la tabla navegada y aquella en la que se establece el atributo de fórmula, solo considerará los registros relacionados para realizar el cálculo.
- Si no encuentra una relación, GeneXus realizará la operación considerando todos los registros en la tabla navegada.

GeneXus™

The power of doing.

Videos

training.genexus.com

Documentation

wiki.genexus.com

Certifications

training.genexus.com/certifications