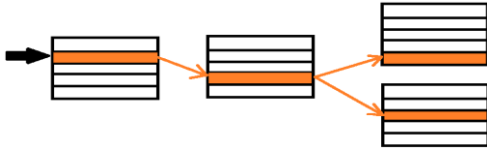


Fórmulas horizontales

GeneXus™

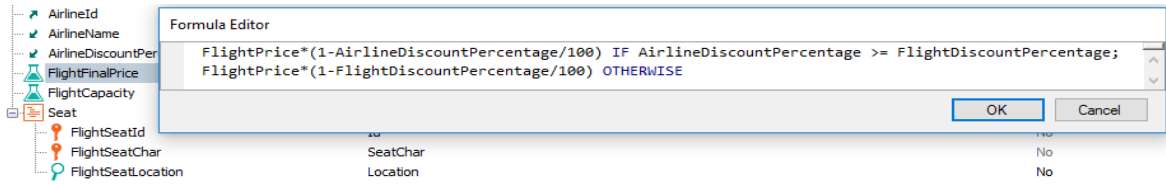
Repaso de fórmulas horizontales

- Permiten definir una o varias expresiones condicionales
- Se llaman horizontales porque acceden a un único registro, involucrando atributos de la tabla extendida.
- Siempre tienen un contexto



Atributo =
$$\begin{aligned} & \text{expresión}_1 \text{ if condición}_1; \\ & \text{expresión}_2 \text{ if condición}_2; \\ & \dots \\ & \text{expresión}_n \text{ if condición}_n; \\ & \text{expresión}_0 \text{ otherwise;} \end{aligned}$$

- Ejemplo:



Recordemos lo que vimos de las fórmulas horizontales.

Las fórmulas horizontales nos permiten definir expresiones aritméticas o de otro tipo, que incluyen condiciones que son cualquier expresión lógica válida.

Al atributo que definimos como fórmula le podemos asignar varias expresiones con condiciones excluyentes y se ejecutará aquella la primera para la que se cumpla la condición, el resto no serán evaluadas. También podemos con "otherwise" asignar una expresión por defecto, que se ejecutará si no se cumple ninguna de las otras condiciones definidas.

Los atributos involucrados en la definición de la fórmula deberán pertenecer a la tabla extendida de la tabla asociada al atributo fórmula.

Estas fórmulas se llaman horizontales porque en su ejecución acceden a un único registro de la tabla base y eventualmente a aquellos relacionados pertenecientes a la tabla extendida.

Más sobre fórmulas horizontales

$$\text{Atributo} = \begin{array}{l} \text{expresión}_1 \text{ if condición}_1; \\ \text{expresión}_2 \text{ if condición}_2; \\ \dots \\ \text{expresión}_n \text{ if condición}_n; \\ \text{expresión}_o \text{ otherwise;} \end{array}$$

Qué podemos incluir:

Expresión:

- Atributos (de la tabla extendida de la tabla asociada al atributo fórmula)
- Constantes, Variables predefinidas, Funciones y Operadores (aritméticos, de strings y de fechas)
- Invocación a un procedimiento que devuelva un valor

Condición:

- Atributos (de la tabla extendida de la tabla asociada al atributo fórmula)
- Constantes, Funciones, Operadores lógicos: OR, AND, NOT y relacionales: (>, >=, <, <=, =, <>, like).

Vemos aquí la sintaxis general de la fórmula.

Las *expresiones* pueden contener **atributos pertenecientes a la tabla extendida de la tabla asociada al atributo que se está definiendo como fórmula**, constantes, variables predefinidas, funciones y operadores aritméticos (como la suma, resta, multiplicación, división, o potencia), operadores de strings (como el +) y operadores de fechas (como el + y el -). También pueden contener una invocación a un objeto procedimiento que retornen un valor. El resultado, ya sea de las expresiones o el retornado por el procedimiento invocado, deberá ser del mismo tipo de datos que el del atributo que se está definiendo como fórmula.

Las *condiciones* son cualquier expresión lógica válida, pudiendo contener **atributos pertenecientes a la tabla extendida de la tabla asociada al atributo que se está definiendo como fórmula**, constantes, funciones y operadores lógicos (and, or, not) y operadores de comparación (como >, >=, <, <=, = y like). La primera condición que al ser evaluada sea True, provocará que el resultado de la fórmula sea el de la expresión de la izquierda de esa condición y las demás no se seguirán evaluando.

Más sobre fórmulas horizontales

Invocación a un objeto procedimiento:

The image shows a screenshot of the GeneXus IDE. At the top, a table lists attributes for the 'FlightInstance' entity. The 'FlightInstancePrice' attribute is highlighted in blue, and its formula is shown as 'FlightPriceCalculator(FlightId, FlightInstanceNumberOfPassengers)'. To the right, a 'Formula Editor' window shows the same formula. Below, a 'FlightPriceCalculator' procedure window is open, showing its 'Rules' tab with the following code:

```
1 Param(in:FlightId, in:&FlightInstanceNumberOfPassengers, out:&FlightInstancePrice);
2
3
4
```

Below this, the 'Source' tab of the procedure is visible, showing a 'For each Flight' loop with a 'Do case' block:

```
1 For each Flight
2   Do case
3     Case &FlightInstanceNumberOfPassengers <= 100
4       &FlightInstancePrice = FlightPrice
5     Case &FlightInstanceNumberOfPassengers > 100 and &FlightInstanceNumberOfPassengers < 200
6       &FlightInstancePrice = FlightPrice * 0.9
7     Otherwise
8       &FlightInstancePrice = FlightPrice * 0.8
9   Endcase
10 EndFor
```

Como dijimos, las fórmulas horizontales pueden incluir en sus expresiones la invocación a un **objeto procedimiento** que devuelva un valor, del tipo de datos del atributo fórmula.

Por ejemplo, que definimos en nuestra aplicación a una transacción FlightInstance. A diferencia de la transacción Flight donde definimos a un vuelo en forma genérica (con origen, destino, asientos y un precio de base), la transacción FlightInstance nos permite modelar una instancia de vuelo en particular, es decir, un vuelo que sale en una determinada fecha, con un número de vuelo y con un costo que depende por ejemplo, de la cantidad de pasajeros de ese vuelo en particular.

Consideremos al atributo FlightInstancePrice, perteneciente a la transacción FlightInstance, que podría definirse como un atributo fórmula que invoque a un procedimiento que devuelva el valor del precio de la instancia del vuelo, calculando los descuentos correspondientes en función del precio del vuelo y de la cantidad de pasajeros.

En la regla Parm del procedimiento vemos los parámetros recibidos: FlightId y &FlightInstanceNumberOfPassengers y el valor devuelto en la variable &FlightInstancePrice. En el source, podemos ver cómo se realiza el cálculo del precio. Si analizamos el código implementado, vemos que podríamos haber incluido esto mismo en la fórmula global, sin haber utilizado un procedimiento

Vemos que la definición de la fórmula contiene la invocación al

procedimiento `FlightPriceCalculator`, pasándole como parámetros al identificador del vuelo y la cantidad de pasajeros. Notemos que en este caso la expresión de la fórmula horizontal sólo contiene la llamada al procedimiento, pero podría incluir otras expresiones y condiciones.

Las fórmulas horizontales deben tener un contexto

Fórmulas horizontales inline

```

Source | Layout | Rules | Conditions | Variables | Help | Documentation
Subroutines
1 For each FlightInstance
2   Do case
3     Case FlightInstanceNumberOfPassengers <= 100
4       FlightInstancePrice = FlightPrice
5     Case FlightInstanceNumberOfPassengers > 100 and FlightInstanceNumberOfPassengers < 200
6       FlightInstancePrice = FlightPrice * 0.9
7     Otherwise
8       FlightInstancePrice = FlightPrice * 0.8
9   Endcase
10 Endfor

```

Name	Type
FlightInstance	FlightInstance
FlightInstanceNumber	Id
FlightInstanceDate	Date
FlightId	Id
FlightPrice	Price
FlightInstanceNumberOfPassengers	Numeric(4.0)
FlightInstancePrice	Price

Base Table: FLIGHTINSTANCE

Context: FLIGHTINSTANCE extended table

Context: FLIGHT extended table

Formula Editor

```

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

```

OK Cancel

Las fórmulas horizontales pueden ser globales como vimos, pero también inline. Este podría ser el caso si por algún motivo no queremos que el atributo FlightInstancePrice sea fórmula sino un atributo secundario cualquiera, pero luego queremos correr un procedimiento que le asigne valor para todos los registros de FlightInstance.

El atributo que recibe el cálculo, FlightInstancePrice no es un atributo fórmula, sino que se le asigna el cálculo únicamente en el source del procedimiento.

Las fórmulas horizontales, tanto globales como inline, para que tengan un sentido, siempre tienen un **contexto**.

En el caso de las fórmulas horizontales inline, el contexto es dado por el lugar donde fue definida la fórmula, por ejemplo dentro de un comando For Each. En este caso el contexto sería la tabla base del For each (FlightInstance) y su tabla extendida.

En el caso de las fórmulas globales, el contexto está dado por la transacción donde fue definida la fórmula, concretamente la tabla extendida de su tabla base. En el ejemplo que vimos, mostrado abajo, es la tabla extendida de la tabla Flight.

El contexto asegura que puedan instanciarse todos los atributos que forman parte de la fórmula.

En siguientes videos veremos otros tipos de fórmulas.

GeneXus™

training.genexus.com
wiki.genexus.com
training.genexus.com/certifications