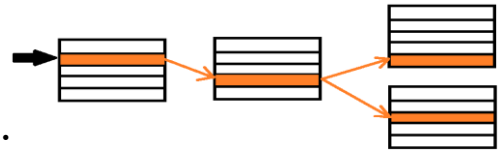
Horizontal Formulas

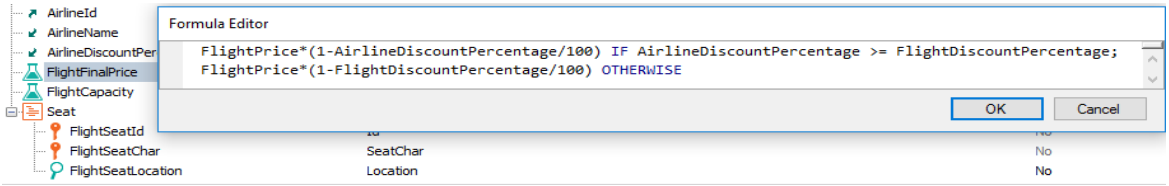
GeneXus™

Review of horizontal formulas

- They allow defining one or more conditional expressions.
- They are called horizontal because they access a single record, involving attributes of the extended table.
- They always have a context.



Attribute = $expression_1$ **if** $condition_1$**;**
$expression_2$ **if** $condition_2$**;**
…
$expression_n$ **if** $condition_n$**;**
$expression_o$ **otherwise;**



Remember what we said about horizontal formulas.

Horizontal formulas allow us to define arithmetic or other expressions, which include conditions that are any valid logical expression.
The attribute that we define as a formula can be assigned several expressions and the first one for which the condition is met will be executed; the rest will not be evaluated. We can also use "otherwise" to assign a default expression, which will be executed if none of the other defined conditions are met.

The attributes involved in the formula definition must belong to the extended table of the table associated with the formula attribute.

These formulas are called horizontal because when they are executed they access a single record of the base table and occasionally those related records belonging to the extended table.

More about horizontal formulas

Attribute =

expression$_1$ **if condition**$_1$**;**
expression$_2$ **if condition**$_2$**;**
…
expression$_n$ **if condition**$_n$**;**
expression$_o$ **otherwise;**

What we can include:

Expression:

- Attributes (from the extended table of the table associated with the formula attribute)
- Constants, Predefined Variables, Functions and Operators (arithmetic, string, and date)
- Invocation to a procedure that returns a value

Condition:

- Attributes (from the extended table of the table associated with the formula attribute)
- Constants, Functions, Logical operators: OR, AND, NOT and relational: (>, >=, <, <=, =, <>, like).

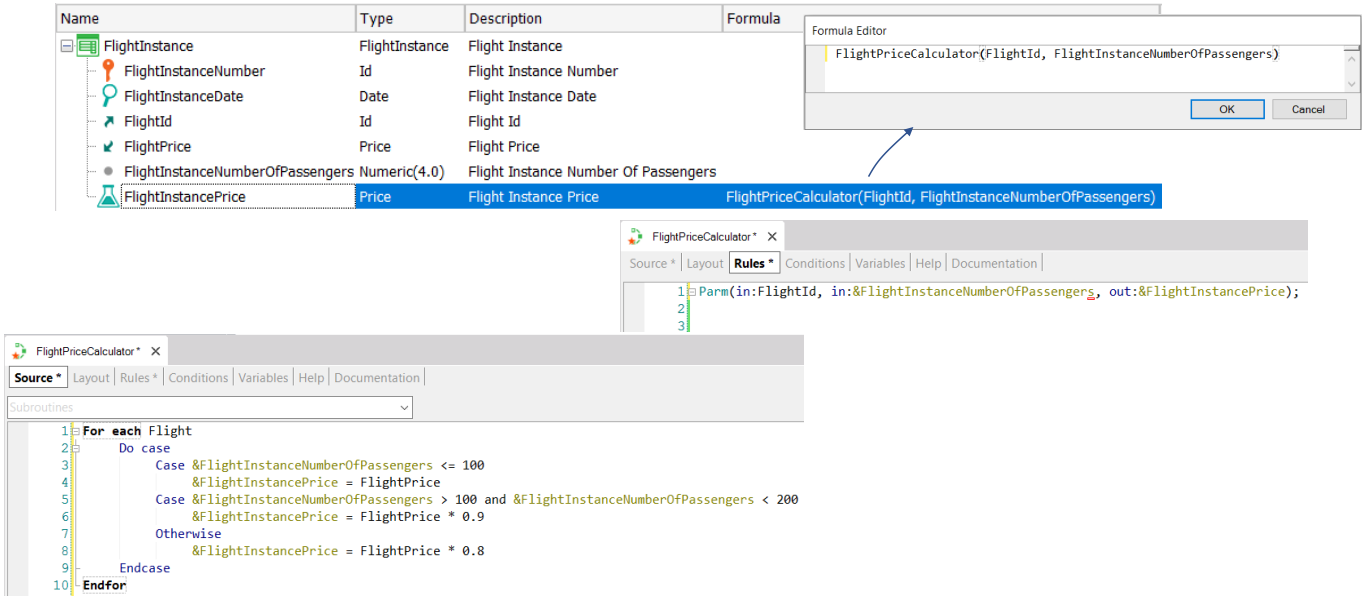
Here we see the general syntax of the formula.

Expressions can contain attributes belonging to the extended table of the table associated with the attribute being defined as a formula, constants, predefined variables, functions and arithmetic operators (such as addition, subtraction, multiplication, division, or power), string operators (such as +) and date operators (such as + and -). They can also contain an invocation to a procedure object that returns a value. The result, either of the expressions or the one returned by the invoked procedure, must be of the same data type as that of the attribute being defined as a formula.

Conditions are any valid logical expression, and may contain attributes belonging to the extended table of the table associated with the attribute being defined as a formula, constants, functions and logical operators (and, or, not) and comparison operators (such as >, >=, <, <=, = and like). The first condition that evaluates to True will cause the result of the formula to be that of the expression to the left of that condition and the others will not continue to be evaluated.

More about horizontal formulas

Invoking a procedure object:



As we said, horizontal formulas can include in their expressions the invocation to a procedure object that returns a value, of the data type of the formula attribute.

For example, we define in our application a FlightInstance transaction. Unlike the Flight transaction where we define a flight in a generic way (with origin, destination, seats and a base price), the FlightInstance transaction allows us to model a particular flight instance, i.e. a flight departing on a certain date, with a flight number and with a cost that depends, for example, on the number of passengers on that particular flight.

Consider the FlightInstancePrice attribute, belonging to the FlightInstance transaction, which could be defined as a formula attribute that invokes a procedure that returns the price value of the flight instance, calculating the corresponding discounts based on the price of the flight and the number of passengers.

In the Parm rule of the procedure we see the parameters received: FlightId and &FlightInstanceNumberOfPassengers and the value returned in the variable &FlightInstancePrice. In the source, we can see how the price calculation is performed. If we analyze the implemented code, we see that we could have included this same thing in the global formula, without having used a procedure.

We can see that the formula definition contains the invocation to the FlightPriceCalculator procedure, sending it as parameters the flight identifier and the number of passengers. Note that in this case the horizontal formula expression only contains the procedure call, but it could include other expressions and conditions.

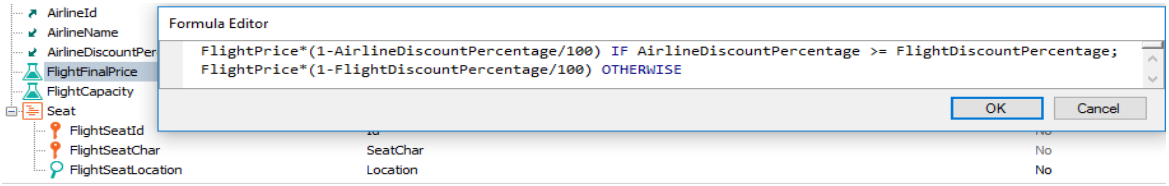
Horizontal formulas must have a context

Inline horizontal formulas

Base table: FLIGHTINSTANCE

Context: FLIGHTINSTANCE extended table

Context: FLIGHT extended table



Horizontal formulas can be global as we have seen, but also inline. This could be the case if for some reason we don't want the FlightInstancePrice attribute to be a formula but an ordinary secondary attribute, but then we want to run a procedure that assigns a value to it for all FlightInstance records.

The attribute that receives the calculation, FlightInstancePrice, is not a formula attribute, but the calculation is assigned only in the source of the procedure.

Horizontal formulas, both global and inline, in order to make sense, always have a context.

In the case of horizontal inline formulas, the context is given by the place where the formula was defined; for example, within a For Each command. In this case, the context would be the base table of the For each (FlightInstance) and its extended table.

In the case of global formulas, the context is given by the transaction where the formula was defined, namely the extended table of its base table. In the example we saw, shown below, it is the extended table of the Flight table.

The context ensures that all attributes that are part of the formula can be instantiated.

In the following videos we will see other types of formulas.

GeneXus™

training.genexus.com
wiki.genexus.com