

# Consecuencias de agregar un atributo a una tabla que ya tiene datos

GeneXus™

## Trn Airport

Name	Type	Description	Formula	Nullable
Airport	Airport	Airport		
AirportId	Id	Airport Id		No
AirportName	Name	Airport Name		No
CountryId	Id	Country Id		No
CountryName	Name	Country Name		No

Supongamos que en nuestra KB de la agencia de viajes, tenemos la transacción Airport en el siguiente estado y con estos atributos. Y queremos en esta transacción agregar los atributos CountryId y CountryName de la transacción Country, para que de esta manera cada Aeropuerto tenga un país asociado.

Teniendo en cuenta que ya existen registros en la tabla Airport antes de la reorganización, ¿con que valor quedara el atributo CountryId?

## Nullable YES

Name	Type	Description	Formula	Nullable
<b>Airport</b>				
AirportId	Id	Airport Id		No
AirportName	Name	Airport Name		No
CountryId	Id	Country Id		Yes
CountryName	Name	Country Name		

AirportId	AirportName	CountryId
1	The Airport of Brazil	Null
2	The Airport of France	Null
3	The Airport of China	Null
--	--	--

## Nullable NO

Name	Type	Description	Formula	Nullable
<b>Airport</b>				
AirportId	Id	Airport Id		No
AirportName	Name	Airport Name		No
CountryId	Id	Country Id		No
CountryName	Name	Country Name		

AirportId	AirportName	CountryId
1	The Airport of Brazil	0
2	The Airport of France	0
3	The Airport of China	0
--	--	--

Aquí tenemos dos opciones, y dependerá de cómo configuremos el atributo CountryId.

Si marcamos que CountryId acepte valores nulos, al reorganizar y GeneXus agregar este atributo en la tabla, los registros que ya estaban quedarán con este atributo en null.

Ahora, si indicamos en la transacción que ese atributo no acepte nulos, ¿qué pasará? Vemos que al reorganizar nos lanza una advertencia, indicando que este atributo no acepta valores nulos y no tiene un valor inicial, por lo que se le asignará un valor por defecto al atributo CountryId.

Se creará un registro en la tabla donde éste atributo es clave primaria y en la tabla donde lo agregamos como clave foránea, se le asignará un valor 0 si el atributo es del tipo numérico, vacío si fuera del tipo character y False si es del tipo boolean.

En este caso se creará un nuevo registro en la tabla Country con CountryId = 0 y CountryName vacío. Y a todos los registros previamente ingresados en la tabla Airport, se asignará ese valor al nuevo atributo CountryId.

En caso de que existiera registrado un país con ID=0, se referenciará a este.

Siempre tendremos la posibilidad de ir registro por registro y modificar este valor por el que queramos. O crear un procedimiento que recorra con un for each todos los aeropuertos, y modifique CountryId según las

condiciones que le especifiquemos.

Podríamos también asignarle al nuevo atributo CountryId, un valor inicial que sepamos existe ya en nuestra base de datos, lo veremos en un momento.

Name	Type	Description	Formula	Nullable
Airport	Airport	Airport		
AirportId	Id	Airport Id		No
AirportName	Name	Airport Name		No
CountryId	Id	Country Id		No
CountryName	Name	Country Name		
AirportInternational	Boolean	Airport International		No

Id	1
Name	The airport of Brazil
Country Id	1
Country Name	Brazil
International	<input type="checkbox"/>

Id	2
Name	The airport of France
Country Id	2
Country Name	France
International	<input type="checkbox"/>

Id	4
Name	
Country Id	0
Country Name	
International	<input checked="" type="checkbox"/>

Veamos el siguiente caso.

Queremos dentro de la transacción Airport, agregar un atributo que registre si el aeropuerto ingresado es internacional o no. Para esto agregamos el atributo AirportInternational del tipo booleano. Supongamos que queremos por defecto el valor sea verdadero. Y para esto creamos una regla Default.

Al reorganizar, vemos nuevamente un Warning, el cual nos dice que el atributo que acabamos de crear no permite valores nulos y no tiene valor inicial, por lo que se utilizará como valor predeterminado el vacío.

Si observamos los registros previamente ingresados en la tabla Airport, vemos que en todos le asignó como valor false a este nuevo atributo, ya que es el valor que se asigna por defecto a este tipo de datos.

Si queremos ingresar un nuevo registro desde la transacción, vemos que aparece por defecto en true, como lo asignamos en la regla, pero no aplica para los registros que ya teníamos.

¿Como podemos hacer para que cuando creamos este atributo, en todos los registros previos queden con valor true? A través de la propiedad Initial Value.

## Propiedad Initial Value

```

1 //Default(AirportInternational, TRUE);

```

The screenshot shows the GeneXus IDE interface. On the left, the 'Structure' view displays a tree of entities for 'Airport'. The 'AirportInternational' attribute is selected, and its properties are shown in the 'Properties' window on the right.

Name	Type	Description
Airport	Airport	Airport
AirportId	Id	Airport Id
AirportName	Name	Airport Name
CountryId	Id	Country Id
CountryName	Name	Country Name
<b>AirportInternational</b>	<b>Boolean</b>	<b>Airport Inter</b>

The 'Properties' window for 'Attribute: AirportInternational' shows the following details:

- Name: **AirportInternational**
- Supertype: (none)
- Based on: (none)
- Data Type: **Boolean**
- Initial value: **true**

Si en lugar de utilizar la regla default, ponemos la propiedad Initial Value del atributo en true, no solo aplicará para los nuevos registros, sino que todos los que ya existen tomarán ese valor. Esa es una de las diferencias entre utilizar la regla default y utilizar la propiedad Initial Value.

## Regla Default vs Initial Value

### **Regla Default:**

Para los registros ya ingresados en la base de datos, el nuevo atributo toma un valor predeterminado.

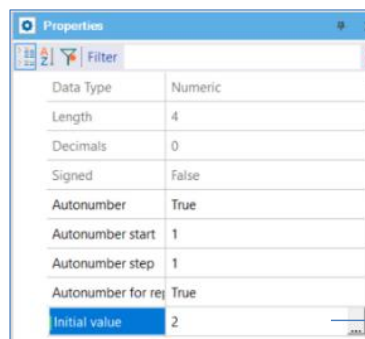
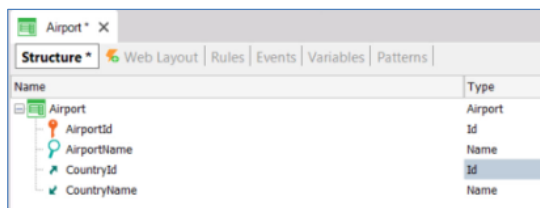
Pertenece a la transacción

### **Initial Value:**

Para los registros ya ingresados en la base de datos, el nuevo atributo toma el valor de la propiedad Initial Value.

Pertenece al atributo.

Otra diferencia, es que la regla default es propia de la transacción, no del atributo. Mientras que la propiedad Initial value es propia del atributo. Y aplicará en cualquier lugar donde se utilice, transacciones, procedimientos, BC, etc.



Airport

« < > » SELECT

Id: 2

Name: Paris Oly Airport

Country Id: 2 [dropdown arrow]

Country Name: France

Volvamos un momento al caso donde agregábamos CountryId a la tabla Airport, veamos qué pasaría si en lugar de haberlo hecho como lo hicimos, le hubiésemos asignado un valor inicial. Por ejemplo, supongamos que sabemos que todos los aeropuertos ya ingresados son del país Francia. Como Francia tiene el id 2, le asignamos este valor a en el Initial Value del atributo Countryid.

Al ejecutar vemos que se reorganiza la tabla Airport agregándose al atributo CountryId, y al ver los datos comprobamos que, en todos los registros previamente ingresados, CountryId tomó el valor 2, correspondiente a Francia.

Incluso cuando queremos agregar un nuevo registro en Airport, Countryid se inicializará por defecto con el valor 2, pudiendo modificarlo si así lo deseamos.

¿Y qué pasaría si hubiésemos puesto en el Initial Value de CountryId un valor que no existe en la tabla country? Por ejemplo 20.



The image displays four components related to the 'Airport' table in GeneXus:

- Structure:** A tree view showing the table structure with fields: AirportId (Airport), AirportName (Name), CountryId (Id), and CountryName (Name).
- Properties:** A window showing the properties of the 'Id' field. Key properties include: Based on: Id, Data Type: Numeric, Length: 4, Decimals: 0, Signed: False, Autonumber: True, Autonumber start: 1, Autonumber step: 1, Autonumber for re: True, and Initial value: 20.
- Data Entry Form:** A form with fields for Id (value: 1), Name (value: Paris Charles de Gaulle Airport), Country Id (value: 20), and Country Name.
- Data Table:** A table with columns 'Id' and 'Name'. It contains records for countries: 1 Brazil, 3 China, 7 England, 2 France, and 6 Italy. The value '20' is shown in the 'Id' column of the first row, corresponding to the 'Initial value' property.

Al ejecutar vemos que, en los registros previamente ingresados, al atributo CountryId se le asigna el valor 20, a pesar de que no teníamos ningún país con este id. Lo que hizo GeneXus en este caso para poder asignar a CountryId el valor 20, fue crear un país en la tabla Country con id 20 y con nombre vacío. Ya que de lo contrario se presentarían fuertes conflictos de integridad referencial.

Si en un momento no queremos más que este atributo se inicialice con un valor, podemos modificar esta propiedad y quitarle el valor que ingresamos. Esto ya no impactará sobre los registros que tengamos ingresados.

Es importante recordar, que el valor inicial se asigna al nuevo atributo al momento en que se reorganiza la tabla. Si no hubiéramos asignado un valor inicial al atributo cuando fue agregado y se ejecuta la reorganización, después será necesario recorrer todos los registros para asignarle un valor.

Como mencionamos también, al crear un nuevo registro, si hay un valor inicial asignado, se creará el nuevo registro con el atributo en ese valor.

## Trn Airport

Name	Type	Description	Formula	Nullable
<b>Airport</b>	<b>Airport</b>	<b>Airport</b>		
AirportId	Id	Airport Id		No
AirportName	Name	Airport Name		No
CountryId	Id	Country Id		No
CountryName	Name	Country Name		
AirportInternational	Boolean	Airport International		No
<b>Flight</b>	<b>Flight</b>	<b>Flight</b>		
FlightId	Id	Flight Id		No
AirlineId	Id	Airline Id		
AirlineName	Name	Airline Name		
FlightFinalPrice	Price	Flight Final Price	FlightPrice * (1-AirlineDi...	
FlightCapacity	Numeric(4,0)	Flight Capacity	count(FlightSeatLocation)	

## Tabla AirportFlight

Name	Type	Description	Formula
<b>AirportFlight Structure</b>		<b>Flight</b>	
AirportId	Id	Airport Id	
FlightId	Id	Flight Id	

Ahora supongamos que queremos registrar desde la transacción Airport los vuelos que se realizan en cada aeropuerto, por lo que ingresamos un segundo nivel con los siguientes atributos de la transacción Flight.

Como a nivel de base de datos esta estructura no genera cambios dentro de la tabla Airport, no impacta sobre los registros previamente ingresados. Se genera una nueva tabla AirportFlight la cual quedará vacía.

## Trn Airport

Name	Type	Description	Formula	Nullable
<ul style="list-style-type: none"> <li>Airport               <ul style="list-style-type: none"> <li>AirportId</li> <li>AirportName</li> <li>CountryId</li> <li>CountryName</li> <li>AirportInternational</li> <li>AirportFlightCapacity</li> <li>Flight                   <ul style="list-style-type: none"> <li>FlightId</li> <li>AirlineId</li> <li>AirlineName</li> <li>FlightFinalPrice</li> <li>FlightCapacity</li> </ul> </li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Airport</li> <li>Id</li> <li>Name</li> <li>Id</li> <li>Name</li> <li>Boolean</li> <li>Numeric(4.0)</li> <li>Flight</li> <li>Id</li> <li>Id</li> <li>Name</li> <li>Price</li> <li>Numeric(4.0)</li> </ul>	<ul style="list-style-type: none"> <li>Airport</li> <li>Airport Id</li> <li>Airport Name</li> <li>Country Id</li> <li>Country Name</li> <li>Airport International</li> <li>Airport Flight Capacity</li> <li>Flight</li> <li>Flight Id</li> <li>Airline Id</li> <li>Airline Name</li> <li>Flight Final Price</li> <li>Flight Capacity</li> </ul>	<ul style="list-style-type: none"> <li></li> <li></li> <li></li> <li></li> <li></li> <li>sum(FlightCapacity)</li> <li></li> <li></li> <li></li> <li></li> <li></li> <li>FlightPrice * (1-AirlineDis...</li> <li>count(FlightSeatLocation);</li> </ul>	<ul style="list-style-type: none"> <li>No</li> <li>No</li> <li>No</li> <li>No</li> <li>No</li> <li></li> <li>No</li> <li></li> <li></li> <li></li> <li></li> <li></li> <li></li> </ul>

Supongamos ahora, que en la transacción Flight tenemos un atributo que contabiliza la cantidad de asientos ingresados para ese vuelo. Y queremos desde Airport saber la cantidad total de asientos, sumando todos los vuelos que hay ingresados en ese Aeropuerto, para de esta manera tener un total de capacidad.

Para esto, tenemos ya creado un atributo fórmula en el primer nivel de la transacción Airport, con la fórmula SUM, que sumará el valor del atributo FlightCapacity de los vuelos ingresados para cada aeropuerto.

Como sabemos, los atributos definidos como fórmula global no se guardan en la base de datos, se calcula su valor cada vez que se necesita, pero esos datos no son persistentes.

Supongamos que con frecuencia se necesita acceder y trabajar con este dato. Al ser calculado en tiempo de ejecución, si son muchos los registros que debe recorrer (por ejemplo, en el caso de tener una cantidad muy grande de vuelos ingresados), puede tener un fuerte impacto en el rendimiento.

Para solucionar esta situación, GeneXus nos permite definir un atributo fórmula global como redundante. Esto significa que dejará de ser un atributo virtual, este quedará almacenado en su tabla asociada y GeneXus mantendrá el conocimiento de que el atributo es una fórmula y cómo calcular su valor.

De esta manera, al ejecutar la transacción que contiene un atributo con fórmula redundante, la fórmula será evaluada, calculada y el resultado será

almacenado en la base de datos.

Luego, cuando necesitemos volver a consultar o trabajar con ese atributo, GeneXus obtendrá el valor almacenado de la base de datos en lugar de dedicarle tiempo y trabajo a realizar el cálculo.

## Atributo Redundante

Name	Redundant	Type	Description	Formula	Nullable
<ul style="list-style-type: none"> <li>Airport           <ul style="list-style-type: none"> <li>AirportId</li> <li>AirportName</li> <li>CountryId</li> <li>CountryName</li> <li>AirportInternational</li> <li><b>AirportFlightCapacity</b></li> </ul> </li> <li>Flight           <ul style="list-style-type: none"> <li>FlightId</li> <li>AirlineId</li> <li>AirlineName</li> <li>FlightFinalPrice</li> <li>FlightCapacity</li> </ul> </li> </ul>	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/>	Airport Id Name Id Name Boolean <b>Numeric(4,0)</b> Flight Id Id Name Price Numeric(4,0)	Airport Airport Id Airport Name Country Id Country Name Airport International <b>Airport Flight Capacity</b> Flight Flight Id Airline Id Airline Name Flight Final Price Flight Capacity	sum(FlightCapacity) FlightPrice * (1-Airlin... count(FlightSeatLoca...	No No No No No No No No No No

Para definir este atributo como redundante, hacemos clic derecho en la fila de encabezados de columna, elegimos columna chooser y ahí vemos todas las columnas que podemos incluir en el editor de transacciones. Elegimos redundante y lo arrastramos hasta el editor. Vemos que, mediante un check nos da la opción de marcar el atributo como redundante. Lo seleccionamos y volvemos a ejecutar.

Nos pedirá reorganizar la base de datos, ya que como vimos, se agregará en ella el atributo que acabamos de marcar como redundante.

Ante un registro nuevo de Airport, se dispara la fórmula como siempre en la transacción, sumando el total de vuelos ingresados. Y al confirmar ahora ya almacena también este valor en la base de datos.

Ahora, ¿qué pasará con los registros de Airport que ya existían en la base de datos? En esos registros ¿con qué valor habrá quedado este atributo en la tabla?

Al hacer la reorganización y agregar este atributo, GeneXus recorre uno por uno los registros previamente ingresados, disparando la fórmula y guardando su valor en la tabla. Al ir a los registros anteriores, ya el nuevo atributo nos aparece con su valor correspondiente, el cual como dijimos, ya está guardado en la base de datos.

Hasta aquí vimos brevemente como impacta en una transacción con registros previos, el agregar atributos nuevos, ya sean atributos inferidos, atributos con fórmula redundante o con valor en la propiedad Initial Value.

Para más información los invitamos a indagar en nuestra Wiki.

# GeneXus™

[training.genexus.com](http://training.genexus.com)  
[wiki.genexus.com](http://wiki.genexus.com)  
[training.genexus.com/certifications](http://training.genexus.com/certifications)