

# Consequências de adicionar um atributo a uma tabela que já contém dados

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       |

Suponhamos que em nossa KB da agência de viagens, temos a transação Airport no seguinte estado e com estes atributos. E queremos nesta transação adicionar os atributos CountryId e CountryName da transação Country, para que desta maneira cada Aeroporto tenha um país associado.

Levando em conta que já existem registros na tabela Airport antes da reorganização, com que valor ficará o atributo CountryId?

## Nullable YES

| Name        | Type    | Description  | Formula | Nullable |
|-------------|---------|--------------|---------|----------|
| Airport     | Airport | Airport      |         |          |
| 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 |
|-------------|---------|--------------|---------|----------|
| Airport     | Airport | Airport      |         |          |
| 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         |
| --        | --                    | --        |

Aqui temos duas opções, e dependerá de como configuramos o atributo CountryId.

Se marcarmos que CountryId aceita valores nulos, ao reorganizar e GeneXus adicionar este atributo na tabela, os registros que já estavam, ficarão com este atributo em null.

Agora, se indicamos na transação que esse atributo não aceita nulos, o que acontecerá? Vemos que ao reorganizar, nos lança um aviso, indicando que este atributo não aceita valores nulos e não tem um valor inicial, então será atribuído um valor por padrão ao atributo CountryId.

Será criado um registro na tabela onde este atributo é chave primária e na tabela onde o adicionamos como chave estrangeira, lhe será atribuído um valor 0 se o atributo é do tipo numérico, vazio se for do tipo character e False se é do tipo boolean.

Neste caso, será criado um novo registro na tabela Country com CountryId = 0 e CountryName vazio. E para todos os registros inseridos anteriormente na tabela Airport, será atribuído esse valor ao novo atributo CountryId.

No caso em que existisse registrado um país com ID=0, será referenciado a este.

Sempre teremos a possibilidade de ir registro por registro e modificar este valor para o que quisermos. Ou criar um procedimento que percorra com um for each todos os aeroportos e modifique CountryId de acordo com as

condições que especifiquemos.

Poderíamos também atribuir ao novo atributo CountryId, um valor inicial que sabemos que já existe em nossa base de dados, veremos isto em um 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          |         | No       |
| 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"/> |

Vejamos o seguinte caso.

Queremos, dentro da transação Airport, adicionar um atributo que registre se o aeroporto inserido é internacional ou não.

Para isto, adicionamos o atributo AirportInternational do tipo booleano.

Suponhamos que queremos por padrão que o valor seja verdadeiro. E

para isto criamos uma regra Default.

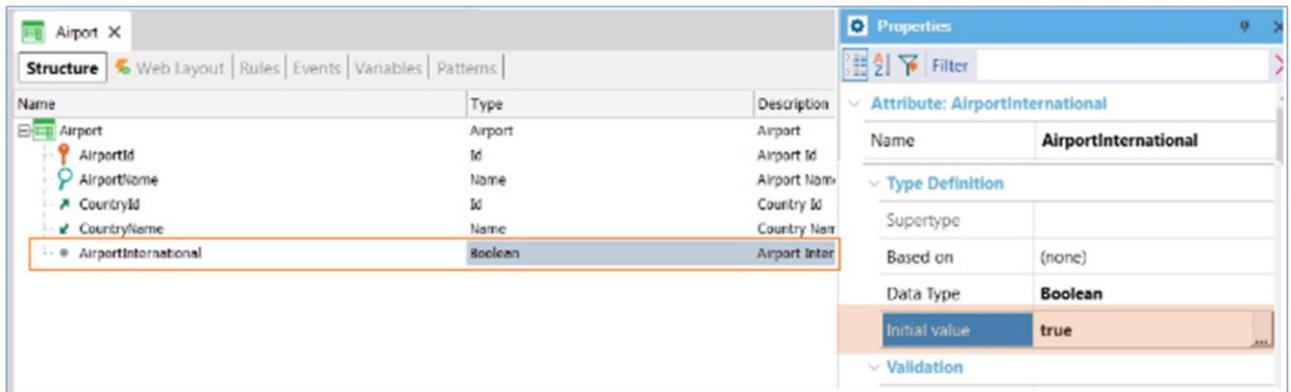
Ao reorganizar, vemos novamente um Warning, que nos diz que o atributo que acabamos de criar não permite valores nulos e não tem valor inicial, portanto, será utilizado como valor predeterminado o vazio.

Se observamos os registros previamente inseridos na tabela Airport, vemos que em todos eles atribuiu valor false a este novo atributo, pois é o valor que é atribuído por padrão a este tipo de dados.

Se queremos inserir um novo registro a partir da transação, vemos que aparece por padrão como true, como o atribuímos na regra, mas não se aplica aos registros que já tínhamos.

Como podemos fazer para que, ao criarmos este atributo, todos os registros anteriores fiquem com valor true? Através da propriedade Initial Value.

## Propriedade Initial Value



Se, em vez de utilizar a regra default, colocamos a propriedade Initial Value do atributo como true, não só se aplicará para os novos registros, mas todos os que já existem assumirão esse valor. Essa é uma das diferenças entre utilizar a regra default e utilizar a propriedade Initial Value.

## Regra Default vs Initial Value

### **Regra Default:**

Para os registros já inseridos na base de dados, o novo atributo assume um valor predeterminado.

Pertence à transação

### **Initial Value:**

Para os registros já inseridos na base de dados, o novo atributo assume o valor da propriedade Initial Value. Pertence ao atributo.

Outra diferença é que a regra default pertence à transação, não ao atributo. Enquanto a propriedade Initial value é própria do atributo. E se aplicará em qualquer lugar em que seja utilizado, transações, procedimentos, BC, etc.

The image shows three components of the GeneXus development environment:

- Structure View:** A tree view showing the 'Airport' table with fields: AirportId (Id), AirportName (Name), CountryId (Id), and CountryName (Name).
- Properties View:** A 'Properties' window for the 'CountryId' field. It shows:
 

|                   |         |
|-------------------|---------|
| Data Type         | Numeric |
| Length            | 4       |
| Decimals          | 0       |
| Signed            | False   |
| Autonumber        | True    |
| Autonumber start  | 1       |
| Autonumber step   | 1       |
| Autonumber for re | True    |
| Initial value     | 2       |
- Data Entry Form:** A form titled 'Airport' with fields:
  - Id:** Input field containing '2'.
  - Name:** Input field containing 'Paris Only Airport'.
  - Country Id:** Input field containing '2' with a dropdown arrow.
  - Country Name:** Text field containing 'France'.

A blue arrow points from the 'Initial value' property in the Properties window to the 'Country Id' field in the form.

Voltemos um momento ao caso em que adicionávamos CountryId à tabela Airport, vejamos o que aconteceria se, em vez de ter feito como fizemos, tivéssemos atribuído a ele um valor inicial. Por exemplo, suponhamos que sabemos que todos os aeroportos já inseridos são do país França. Como a França tem o id 2, atribuímos este valor no Initial Value do atributo Countryid.

Ao executar vemos que se reorganiza a tabela Airport adicionando o atributo CountryId, e ao ver os dados verificamos que, em todos os registros previamente inseridos, CountryId assumiu o valor 2, correspondente a França.

Mesmo quando queremos adicionar um novo registro em Airport, Countryid será inicializado por padrão com o valor 2, podendo modificá-lo se desejarmos.

E o que aconteceria se tivéssemos colocado no Initial Value de CountryId um valor que não existe na tabela country? Por exemplo 20.

The image illustrates the configuration of an initial value for a foreign key field in GeneXus. It shows the structure of the 'Airport' table, the properties of the 'CountryId' field, and the resulting data grid.

**Structure of Airport table:**

| Name        | Type    |
|-------------|---------|
| Airport     | Airport |
| AirportId   | Id      |
| AirportName | Name    |
| CountryId   | Id      |
| CountryName | Name    |

**Properties of CountryId field:**

|                   |         |
|-------------------|---------|
| Based on          | Id      |
| Data Type         | Numeric |
| Length            | 4       |
| Decimals          | 0       |
| Signed            | False   |
| Autonumber        | True    |
| Autonumber start  | 1       |
| Autonumber step   | 1       |
| Autonumber for re | True    |
| Initial value     | 20      |

**Data Grid:**

| Id | Name    |
|----|---------|
| 20 |         |
| 1  | Brazil  |
| 3  | China   |
| 7  | England |
| 2  | France  |
| 6  | Italy   |

Ao executar vemos que, nos registros previamente inseridos, ao atributo CountryId é atribuído o valor 20, embora não tivéssemos nenhum país com este id. O que GeneXus fez neste caso para atribuir a CountryId o valor 20 foi criar um país na tabela Country com id 20 e com nome vazio. Já que, caso contrário, seriam apresentados fortes conflitos de integridade referencial.

Se em um momento não quisermos mais que este atributo seja inicializado com um valor, podemos modificar esta propriedade e remover o valor que inserimos. Isto já não impactará sobre os registros que tenhamos inseridos.

É importante lembrar que o valor inicial é atribuído ao novo atributo quando a tabela é reorganizada. Se não tivéssemos atribuído um valor inicial ao atributo quando foi adicionado e é executada a reorganização, depois será necessário percorrer todos os registros para atribuir um valor a ele.

Como também mencionamos, ao criar um novo registro, se houver um valor inicial atribuído, será criado o novo registro com o atributo com aquele 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)     |          |

## Tabela AirportFlight

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

Agora suponhamos que queremos registrar a partir da transação Airport, os voos que são realizados em cada aeroporto, então inserimos um segundo nível com os seguintes atributos da transação Flight.

Como no nível da base de dados esta estrutura não gera alterações na tabela Airport, não impacta os registros previamente inseridos. É gerada uma nova tabela AirportFlight a qual ficará vazia.

## 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></li> <li>No</li> <li></li> <li></li> <li></li> <li></li> </ul> |

Suponhamos agora que na transação Flight temos um atributo que contabiliza a quantidade de assentos inseridos para esse voo. E queremos, a partir de Airport, saber a quantidade total de assentos, somando todos os voos que existem inseridos nesse Aeroporto para, desta maneira, ter um total de capacidade.

Para isto, já criamos um atributo fórmula no primeiro nível da transação Airport, com a fórmula SUM, que somará o valor do atributo FlightCapacity dos voos inseridos para cada aeroporto.

Como sabemos, os atributos definidos como fórmula global não são salvos na base de dados, é calculado seu valor toda vez que for necessário, mas esses dados não são persistidos.

Suponhamos que com frequência é necessário acessar e trabalhar com este dado. Como é calculado em tempo de execução, se são muitos os registros que deve percorrer (por exemplo, no caso de ter uma quantidade muito grande de voos inseridos), pode ter um forte impacto no desempenho.

Para resolver esta situação, Genexus permite definir um atributo fórmula global como redundante. Isto significa que deixará de ser um atributo virtual, ficará armazenado em sua tabela associada e GeneXus manterá o conhecimento de que o atributo é uma fórmula e como calcular seu valor. Desta maneira, ao executar a transação que contém um atributo com fórmula redundante, a fórmula será avaliada, calculada e o resultado será armazenado na base de dados.

Posteriormente, quando precisarmos novamente consultar ou trabalhar com esse atributo, GeneXus obterá o valor armazenado da base de dados em vez de dedicar tempo e trabalho para realizar o cálculo.

## Atributo Redundante

| Name                  | Redundant                           | Type         | Description             | Formula                    | Nullable |
|-----------------------|-------------------------------------|--------------|-------------------------|----------------------------|----------|
| Airport               |                                     | Airport      | Airport                 |                            |          |
| AirportId             |                                     | Id           | Airport Id              |                            | No       |
| AirportName           |                                     | Name         | Airport Name            |                            | No       |
| CountryId             |                                     | Id           | Country Id              |                            | No       |
| CountryName           | <input type="checkbox"/>            | Name         | Country Name            |                            |          |
| AirportInternational  |                                     | Boolean      | Airport International   |                            | No       |
| AirportFlightCapacity | <input checked="" type="checkbox"/> | Numeric(4,0) | Airport Flight Capacity | sum(FlightCapacity)        |          |
| Flight                |                                     | Flight       | Flight                  |                            |          |
| FlightId              |                                     | Id           | Flight Id               |                            | No       |
| AirlineId             | <input type="checkbox"/>            | Id           | Airline Id              |                            |          |
| AirlineName           | <input type="checkbox"/>            | Name         | Airline Name            |                            |          |
| FlightFinalPrice      | <input type="checkbox"/>            | Price        | Flight Final Price      | FlightPrice * (1-Airlin... |          |
| FlightCapacity        | <input type="checkbox"/>            | Numeric(4,0) | Flight Capacity         | count(FlightSeatLoca...    |          |

Para definir este atributo como redundante, clicamos com o botão direito do mouse na linha dos cabeçalhos de coluna, escolhemos Column Chooser e lá vemos todas as colunas que podemos incluir no editor de transações. Escolhemos redundante e a arrastamos para o editor. Vemos que, por meio de um check, nos dá a opção de marcar o atributo como redundante. O selecionamos e executamos novamente.

Nos pedirá para reorganizar a base de dados, uma vez que, como vimos, será adicionada nela o atributo que acabamos de marcar como redundante.

Diante de um novo registro de Airport, é disparada a fórmula como sempre na transação, somando o total de voos inseridos. E ao confirmar, agora também armazena este valor na base de dados.

Agora, o que acontecerá com os registros de Airport que já existiam na base de dados? Nesses registros, com qual valor terá ficado este atributo na tabela? Ao fazer a reorganização e adicionar este atributo, GeneXus percorre um por um os registros previamente inseridos, disparando a fórmula e salvando seu valor na tabela. Ao ir para os registros anteriores, o novo atributo aparece com seu valor correspondente, que, como dissemos, já está salvo na base de dados.

Até aqui vimos brevemente como impacta em uma transação com registros prévios, inserir novos atributos, sejam eles atributos inferidos, atributos com fórmula redundante ou com valor na propriedade Initial

Value.

Para mais informações, os convidamos a pesquisar em nossa 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)