

Transações

Quais nomes de atributos usamos?

Product X

Structure Web Form Rules Events Variables

Name	Type
Product	Product
ProductId	Numeric(4.0)
ProductName	Character(20)
ProductPrice	Numeric(10.2)

Invoice X

Structure Web Form Rules Events Variable

Name	Type
Invoice	Invoice
InvoiceId	Numeric(4.0)
InvoiceDate	Date
Product	Product
ProductId	Numeric(4.0)
ProductName	Character(20)
ProductPrice	Numeric(10.2)
InvoiceProductQuantity	Numeric(4.0)

Nomenclatura GIK

Entidade + Categoria [+ Qualificador]

Product

```
{  
    ProductId* (PK)  
    ProductName (S)  
    ProductPrice (S)  
}
```

Invoice

```
{  
    Invoiceld* (PK)  
    InvoiceDate (S)  
-----
```

Product

```
{  
    ProductId* (PK, FK)  
    ProductName (I)  
    ProductPrice (I)  
    InvoiceProductQuantity (S)  
-----  
}  
}
```

Desenho de Transações

1 – N forte

Cada cliente pertence a um país e um país tem muitos clientes



COUNTRY
CountryId*
CountryName

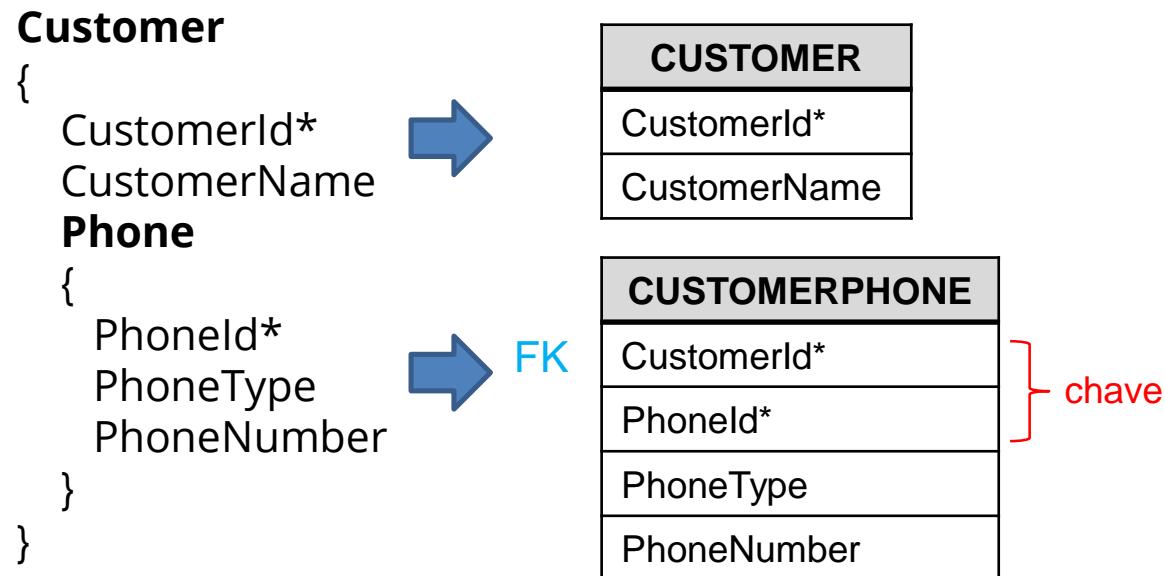
Country
{
 CountryId* (PK)
 CountryName
}

Customer
{
 CustomerId*
 CustomerName
 CountryId (FK)
 CountryName
}

CUSTOMER
CustomerId*
CustomerName
CountryId

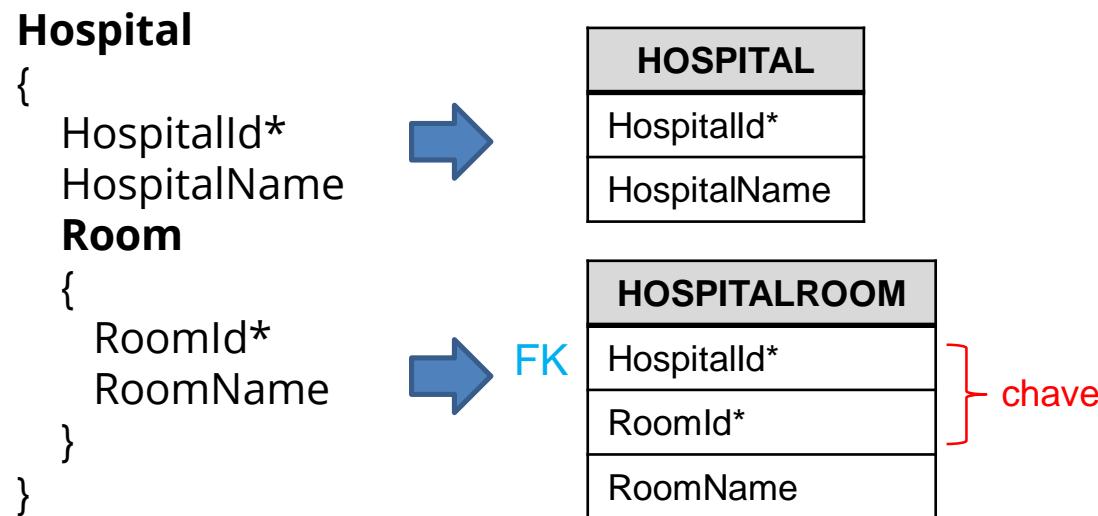
1 – N fraca

Cada cliente tem muitos telefones e cada telefone pertence a um único cliente



1 - N fraca

Cada hospital tem muitas salas e cada sala pertence a um único hospital



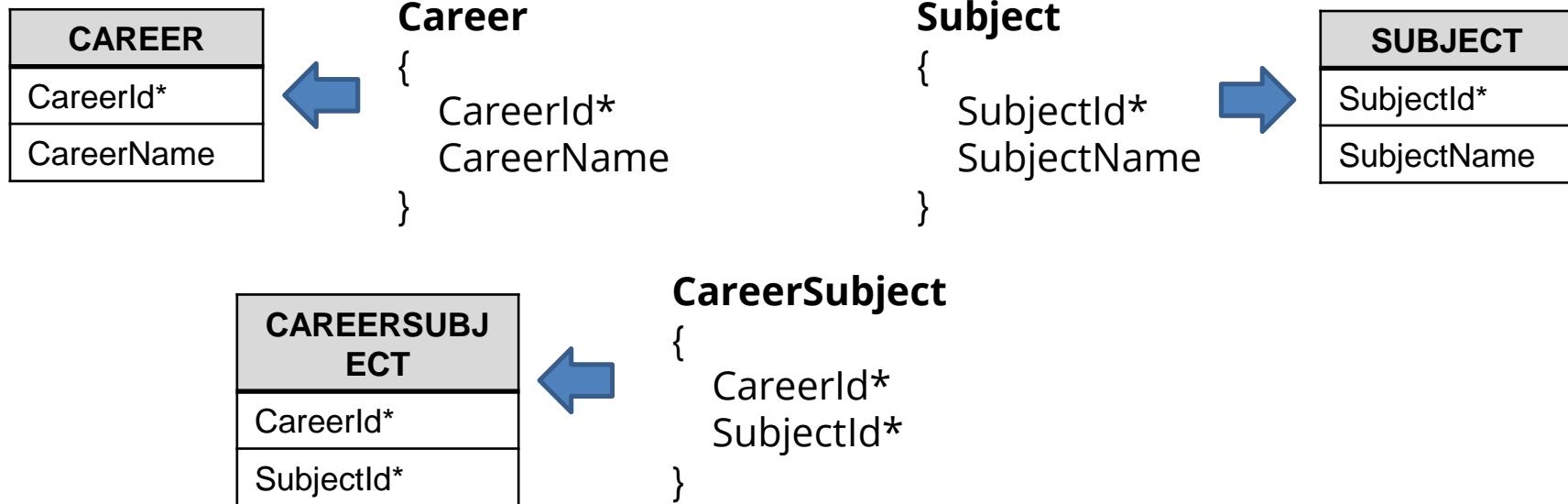
N - N (M)

Cada carreira tem muitas disciplinas e cada disciplina pode estar em muitas carreiras



N - N: Opção 1 de 4

Cada carreira tem muitas disciplinas e cada disciplina pode estar em muitas carreiras



N - N: Opção 1 – Tabelas geradas

CAREER

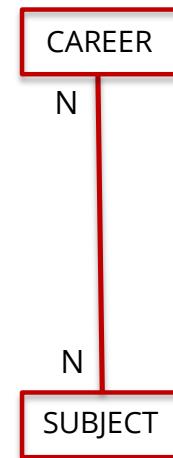
CareerId	CareerName
1	Computer Science
2	Data Science for Health

SUBJECT

SubjectId	SubjectName
1	Computer Logic
2	Programming Fundamentals

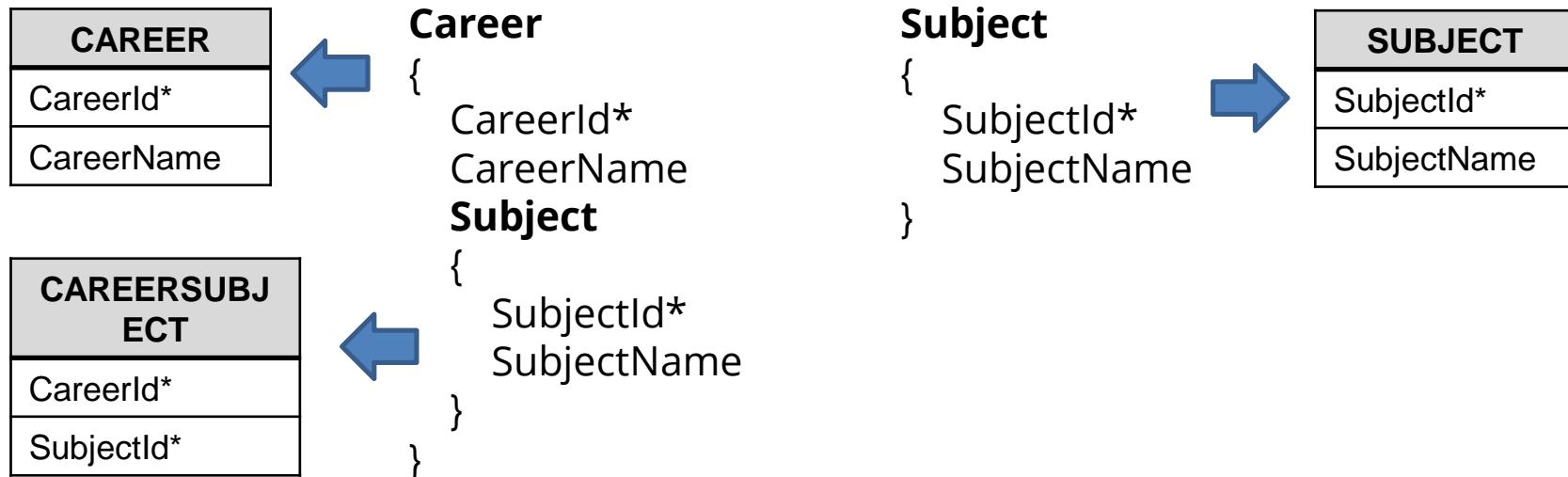
CAREERSUBJECT

CareerId	SubjectId
1	1
1	2
2	2



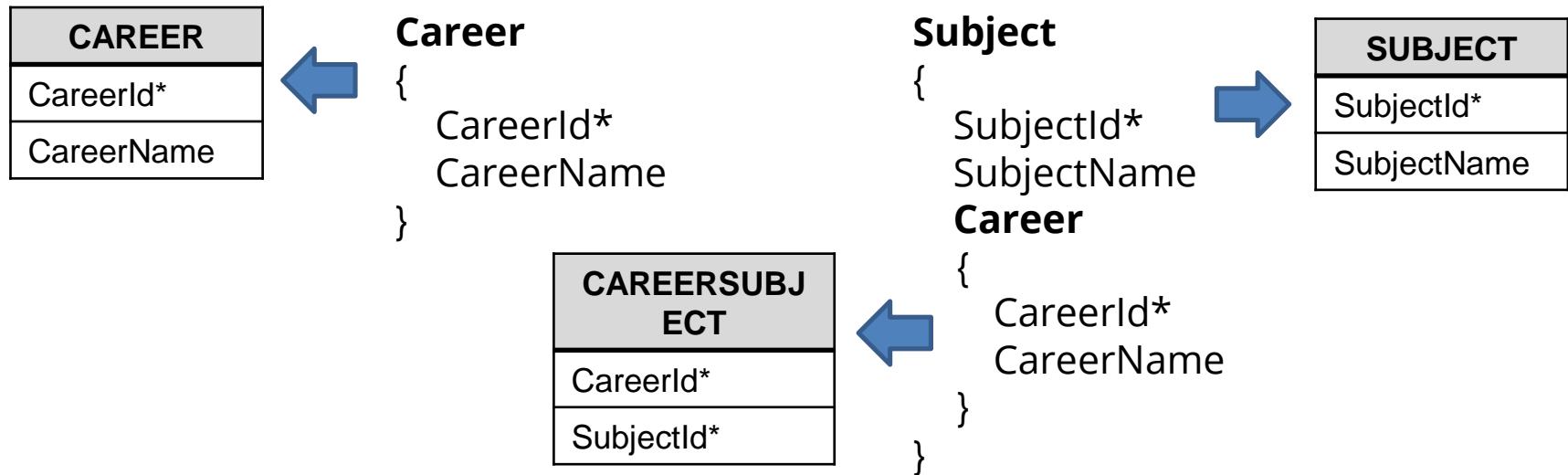
N - N: Opção 2 de 4

Cada carreira tem muitas disciplinas e cada disciplina pode estar em muitas carreiras



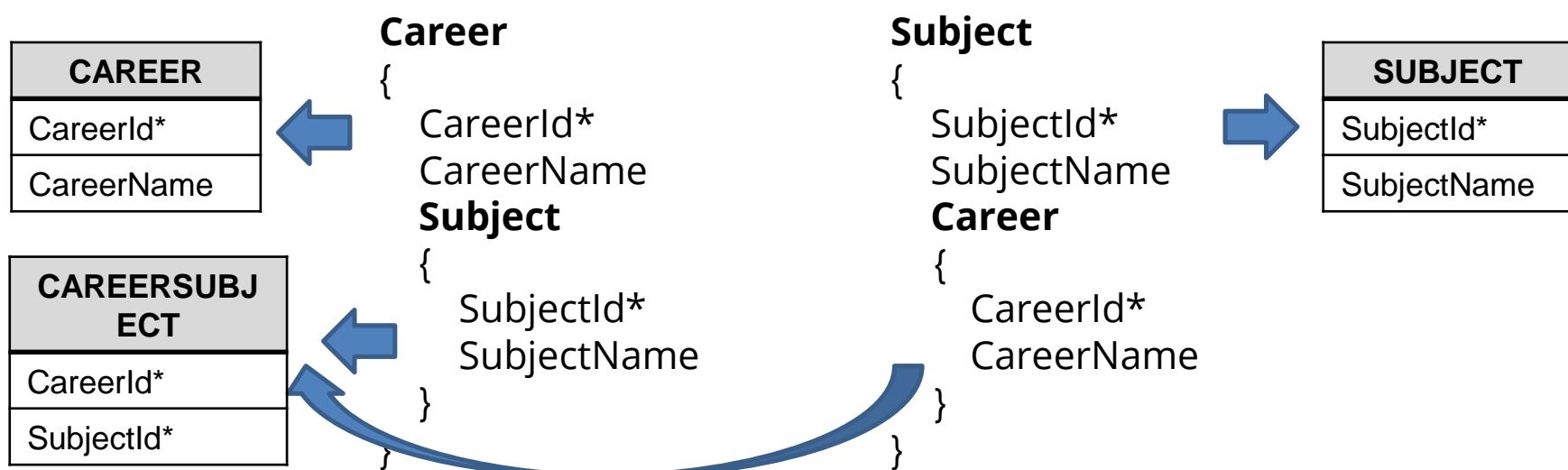
N - N: Opção 3 de 4

Cada carreira tem muitas disciplinas e cada disciplina pode estar em muitas carreiras



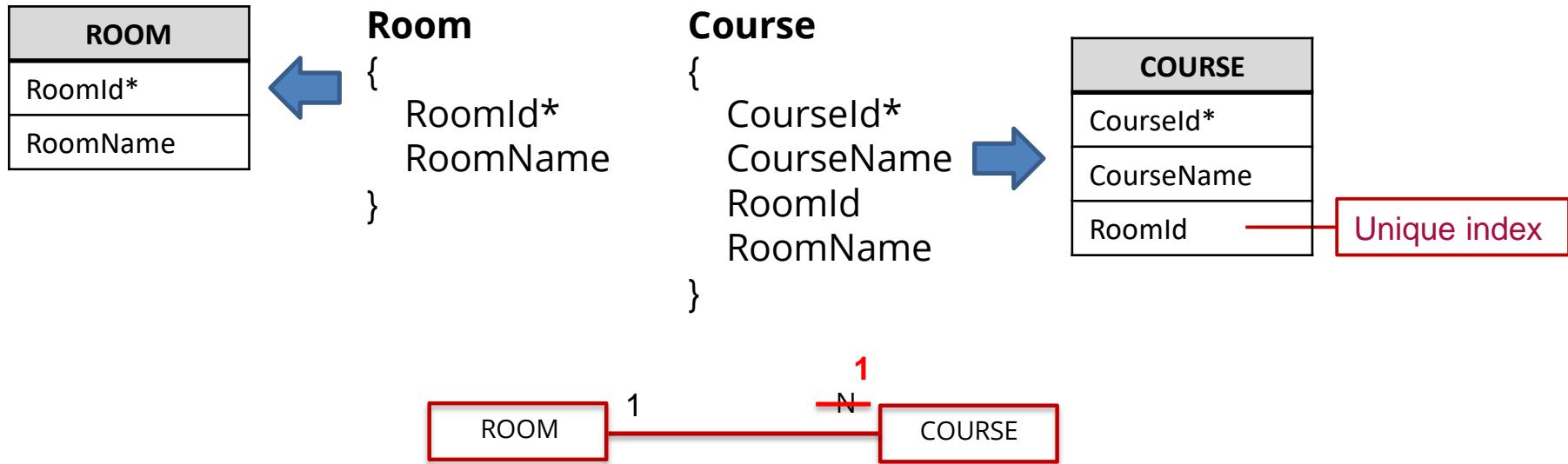
N - N: Opção 4 de 4

Cada carreira tem muitas disciplinas e cada disciplina pode estar em muitas carreiras



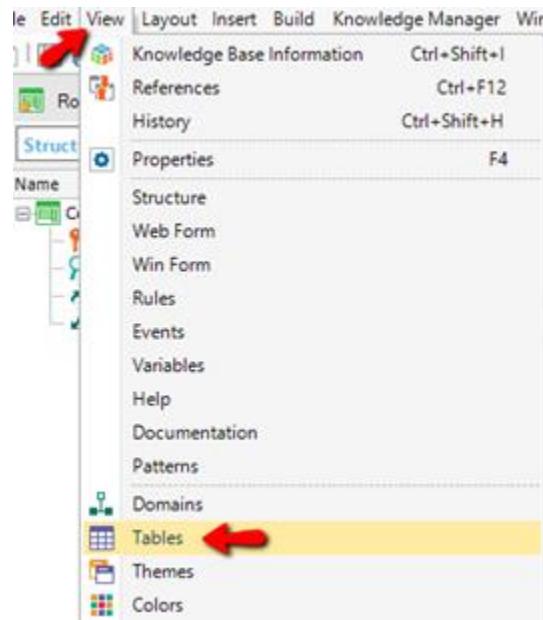
1 - 1

Cada curso é ministrado em uma sala e esta sala só pode ser usada para ministrar esse curso

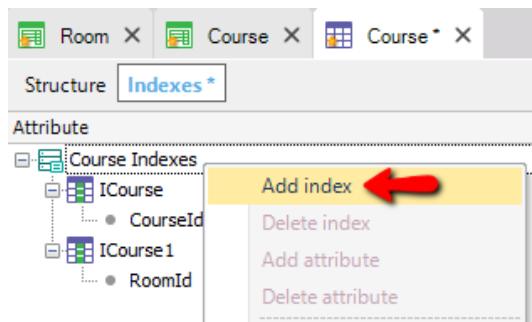


Criação de um índice

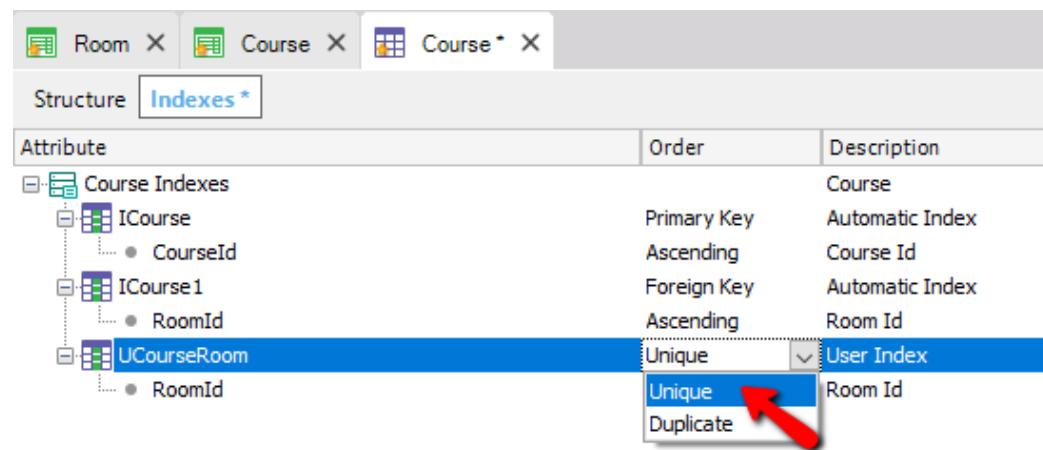
1



2



3



Normalização

GeneXus normaliza as tabelas na Terceira Forma Normal (3NF)

- Atributos inferidos em uma transação não são incluídos na tabela gerada

Continent

```
{  
    ContinentId* (PK)  
    ContinentName  
}
```

Country

```
{  
    CountryId* (PK)  
    CountryName  
    ContinentId (FK)  
    ContinentName (INF)  
}
```

Customer

```
{  
    CustomerId* (PK)  
    CustomerName  
    CountryId (FK)  
    CountryName (INF)  
    ContinentId (INF)  
    ContinentName (INF)  
}
```

CONTINENT

ContinentId*
ContinentName

COUNTRY

CountryId*
CountryName
ContinentId

CUSTOMER

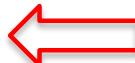
CustomerId*
CustomerName
CountryId

Integridade referencial

Integridade referencial

```
Country
{
    CountryId* (PK)
    CountryName
}
```

CountryId	CountryName
1	URUGUAI
2	ARGENTINA

O registro não é inserido 

```
Customer
{
    CustomerId*
    CustomerName
    CountryId (FK)
    CountryName
}
```

CustomerId	CustomerName	CountryId
1	ANA	1
2	PEDRO	2
3	LUIS	2
4	JOSE	3



Integridade referencial

Country

```
{  
    CountryId* (PK)  
    CountryName  
}
```

CountryId	CountryName
1	URUGUAI 
2	ARGENTINA

O registro não é eliminado

Customer

```
{  
    CustomerId*  
    CustomerName  
    CountryId (FK)  
    CountryName  
}
```

CustomerId	CustomerName	CountryId
1	ANA	1 
2	PEDRO	2
3	LUIS	2
4	JOSE	3

Tabela Base e Estendida

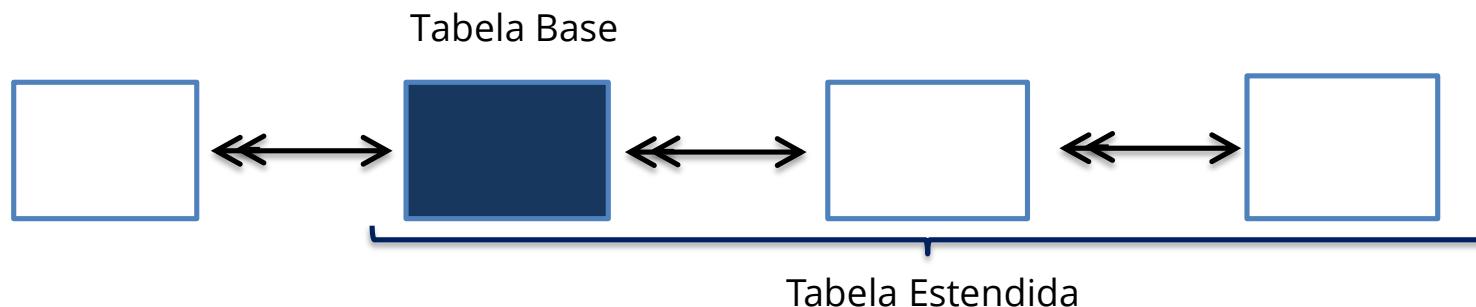
- **Tabela Base**

É qualquer tabela do banco de dados em que estamos posicionados trabalhando em um determinado momento.

- **Tabela Estendida**

Dada uma tabela, sua tabela estendida é um conceito que nos permite pensar todas as informações às quais temos acesso, a partir de suas chaves externas.

É o conjunto de atributos da própria tabela + todos os atributos das tabelas com as quais ela tem um relacionamento N-1 direta ou indiretamente.



Exemplo

```
Customer
{
    CustomerId*
    CustomerName
}

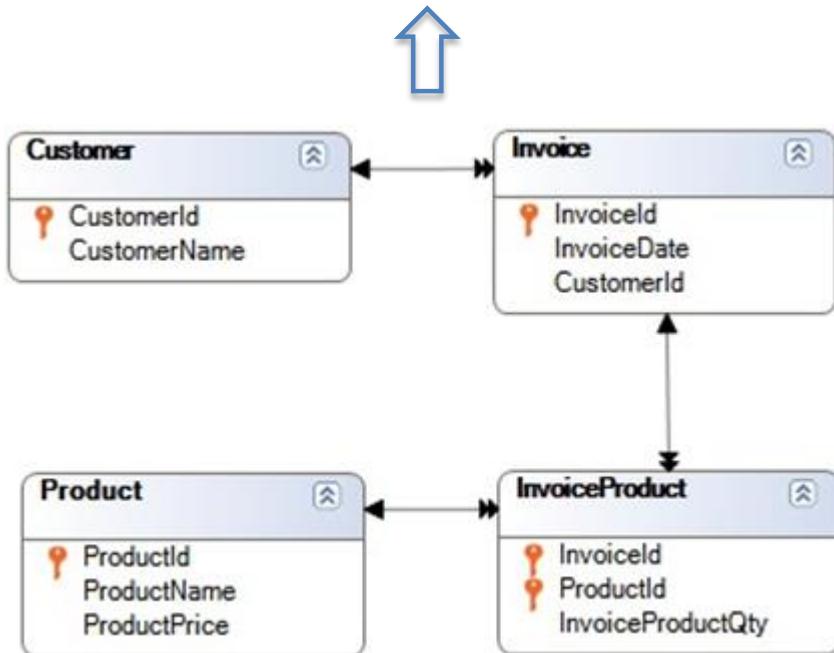
Product
{
    ProductId*
    ProductName
    ProductPrice
}
```

```
Invoice
{
    Invoiceld*
    InvoiceDate
    CustomerId
    CustomerName
}

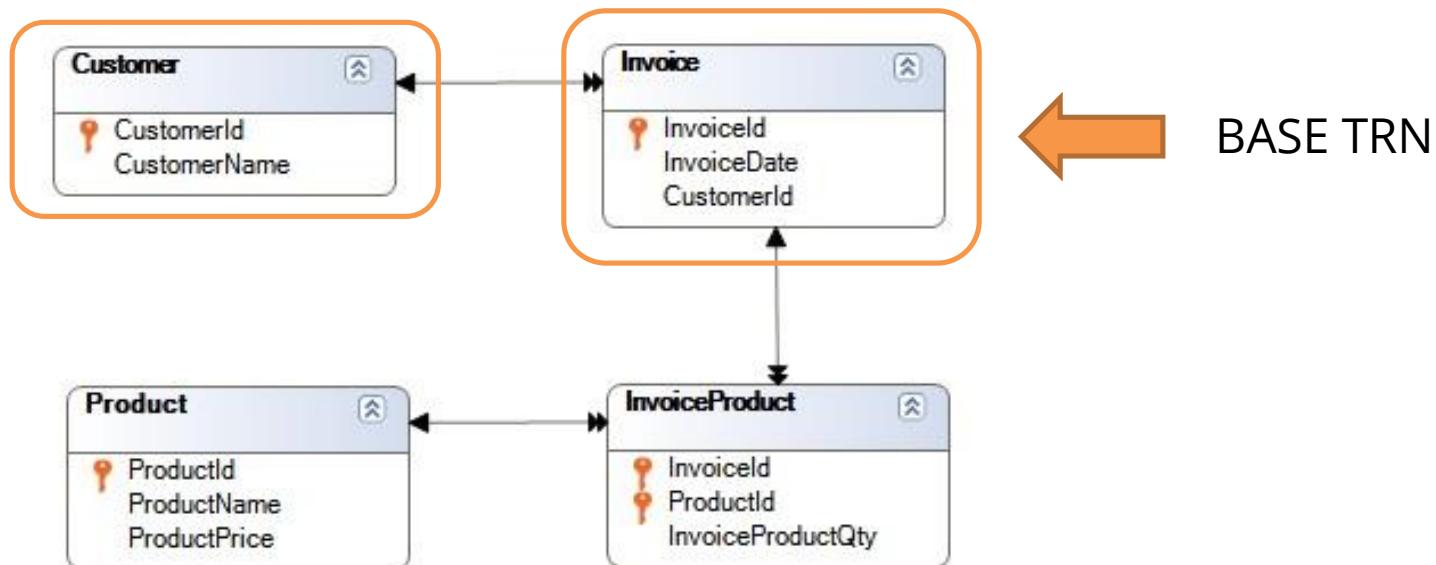
Product
{
    ProductId*
    ProductName
    ProductPrice
    InvoiceProductQuantity
    -----
}

}
```

Diagrama de Tabelas
(diagrama de Bachman)

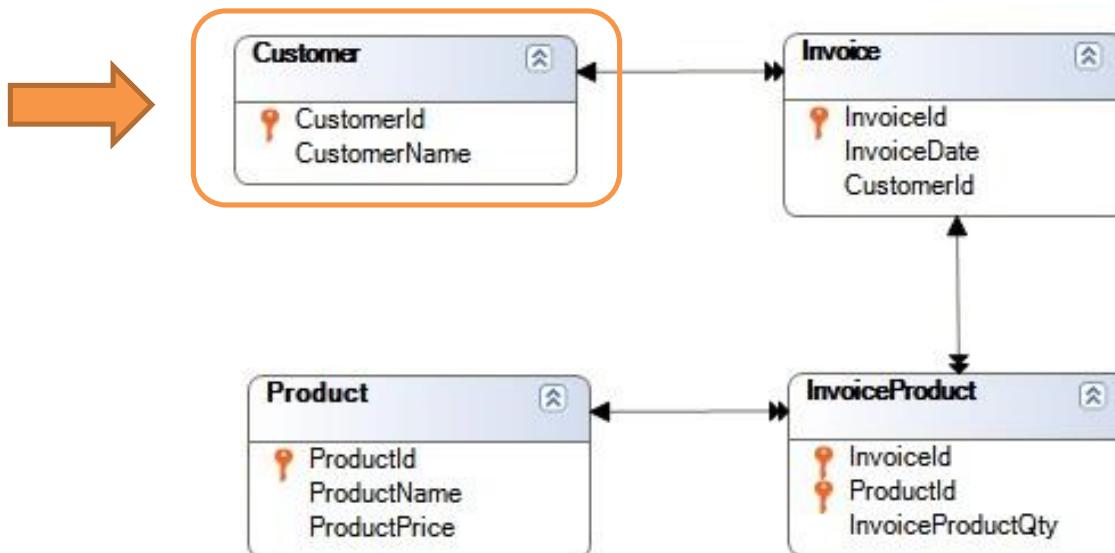


Exemplo: Tabela Estendida de Invoice

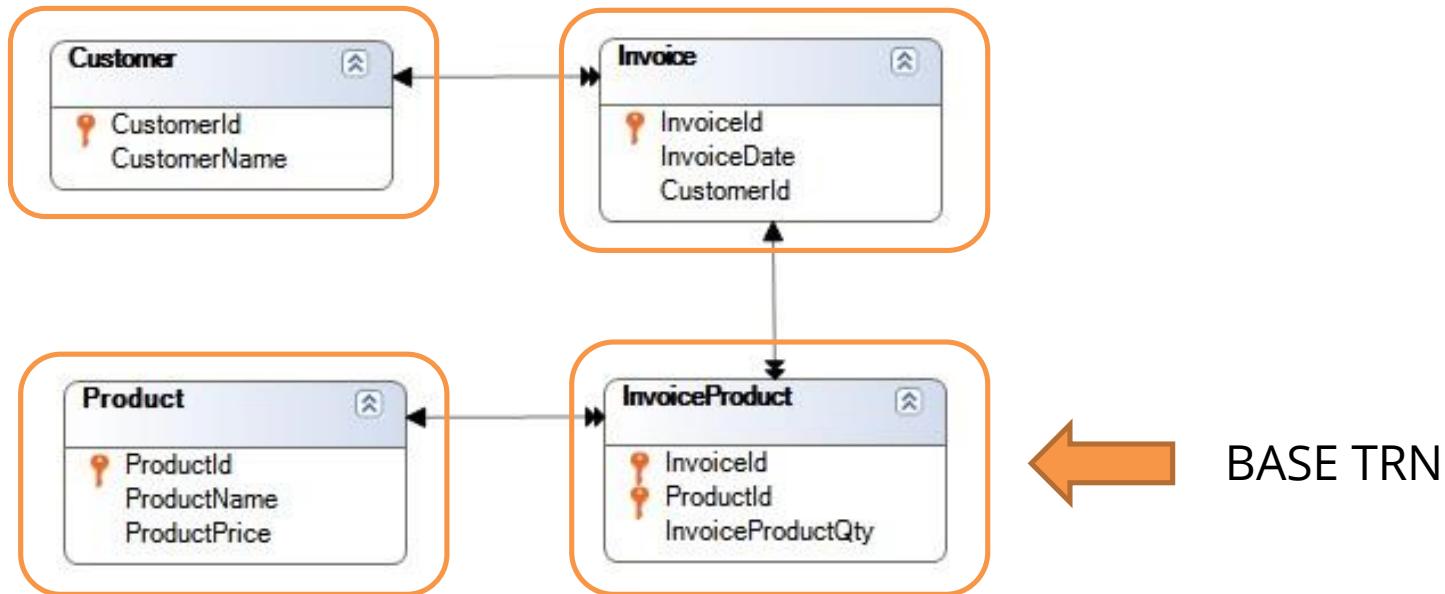


Exemplo: Tabela Estendida de Customer

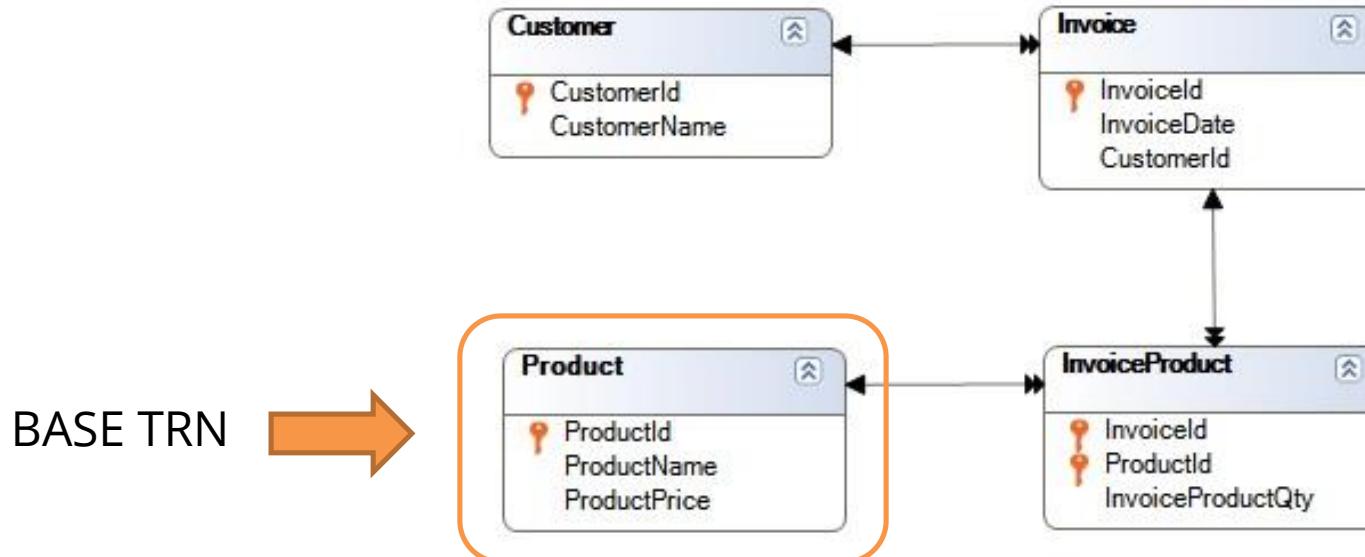
BASE TRN



Exemplo: Tabela Estendida de InvoiceProduct



Exemplo: Tabela Estendida de Product



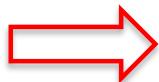
Subtipos

Múltiplas Referências :

É preciso salvar, para cada vôo, o aeroporto de partida e o aeroporto de chegada

```
Airport
{
    AirportId*
    AirportName
}
```

```
Flight
{
    FlightId*
    FlightDate
    AirportId
    AirportName
    AirportId
    AirportName
}
```



Erro por nome de
atributos duplicados

Flight	Flight	Flight
FlightId	Id	Flight Id
FlightDate	Date	Flight Date
AirportId	Id	Airport Id
AirportName	Name	Airport Name
		✖ Duplicate Attribute Name: 'AirportId'

Solução 1 de 3: criar dois grupos de subtipos, um para o aeroporto de partida e outro para o aeroporto de chegada

```
Airport
{
    AirportId*
    AirportName
}
```

```
Flight
{
    FlightId*
    FlightDate
    DepartureAirportId
    DepartureAirportName
    ArrivalAirportId
    ArrivalAirportName
}
```

} Subtype group: DepartureAirport

} Subtype group: ArrivalAirport

Subtype	Description	Supertype
 DepartureAirport		
DepartureAirportId	Departure Airport Id	AirportId

 DepartureAirportName

 Departure Airport Name

 AirportName

Subtype	Description	Supertype
 ArrivalAirport		
ArrivalAirportId	Arrival Airport Id	AirportId

 ArrivalAirportName

 Arrival Airport Name

 AirportName

Solução 2 de 3: criar um grupo de subtipos, somente para o aeroporto de partida

```
Airport
{
    AirportId*
    AirportName
}
```

```
Flight
{
    FlightId*
    FlightDate
    DepartureAirportId
    DepartureAirportName
    AirportId
    AirportName
}
```

} Subtype group: DepartureAirport

Subtype	Description	Supertype
DepartureAirport		
DepartureAirportId	Departure Airport Id	AirportId
DepartureAirportName	Departure Airport Name	AirportName

Solução 3 de 3: criar um grupo de subtipos, somente para o aeroporto de chegada

```
Airport
{
    AirportId*
    AirportName
}
```

```
Flight
{
    FlightId*
    FlightDate
    AirportId
    AirportName
    ArrivalAirportId
    ArrivalAirportName
}
} Subtype group: ArrivalAirport
```

Subtype	Description	Supertype
ArrivalAirport		
ArrivalAirportId	Arrival Airport Id	AirportId
ArrivalAirportName	Arrival Airport Name	AirportName

Múltiplas Referências: É preciso salvar, além do país do cliente, o país onde foi emitida a Fatura

```
Country
{
    CountryId*
    CountryName
}

Customer
{
    CustomerId*
    CustomerName
    CountryId
    CountryName
}
```

```
Invoice
{
    Invoiceld*
    InvoiceDate
    CustomerId
    CustomerName
    CountryId
    CountryName
    InvoiceCountryId
    InvoiceCountryName
}
```

Atributos inferidos

Subtype group: InvoiceCountry

Subtype	Description	Supertype
 InvoiceCountry		
InvoiceCountryId	Invoice Country Id	CountryId
InvoiceCountryName	Invoice Country Name	CountryName

Regras

Regras

Error("Enter the student name") if
StudentName.isEmpty();

Msg("The address is empty") if
StudentAddress.isEmpty();

Default(StudentAddedDate, &Today);

Noaccept(StudentAddedDate);

Name	Type
Student	Student
StudentId	Id
StudentName	Name
StudentAddress	Address, GeneXus
StudentBirthDate	Date
StudentAddedDate	Date

& Variables	
& Standard Variables	
GxRemove	Numeric(1.0)
Mode	Character(3)
Pgmdesc	Character(256)
Pgmname	Character(128)
Time	Character(8)
Today	Date

Regras

Name	Type
Country	Country
CountryId	Id
CountryName	Name
CountryLastLine	Numeric(4.0)
City	City
CityId	Id
CityName	Name

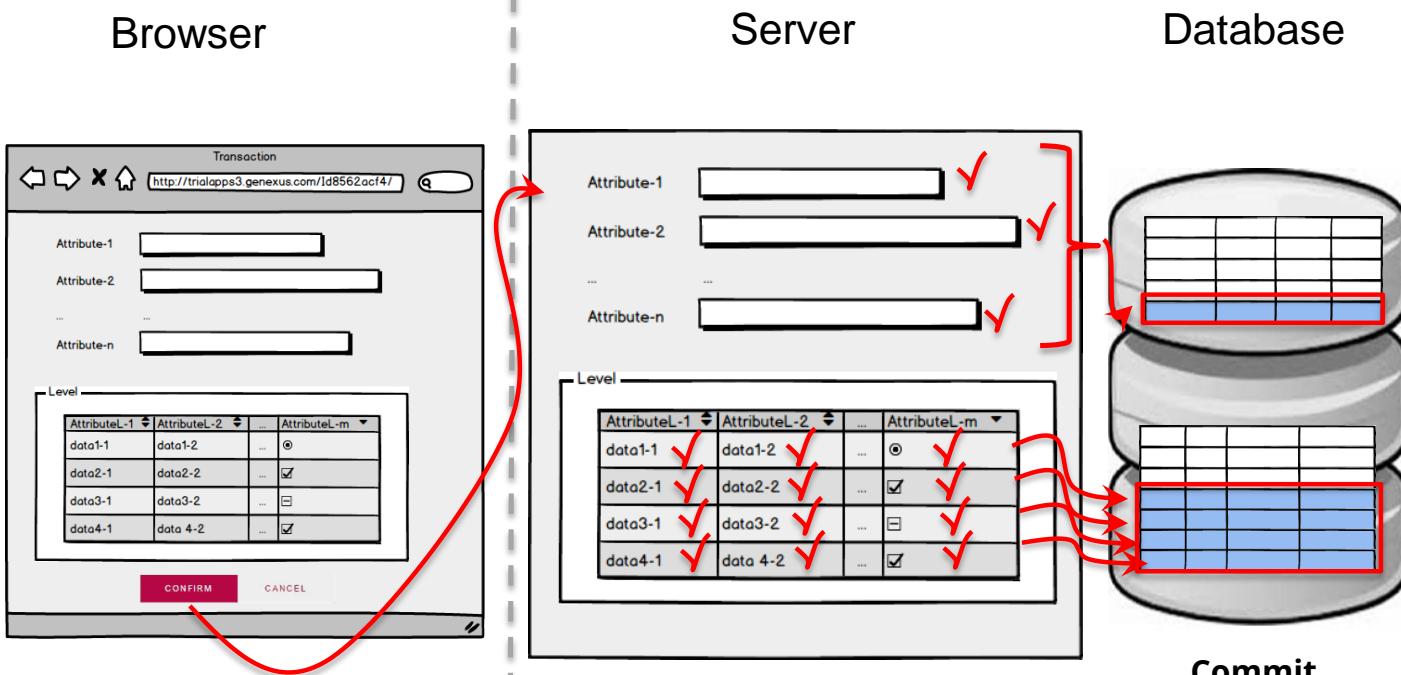
Serial(CityId, CountryLastLine, 1);

Parm(attribute1, &variable1,);



Variável: Espaço em memória que possui um nome e o tipo de dados que pode guardar. São referenciadas através do “&”.

Momentos de disparo



Momentos de disparo de regras

No
servidor,
depois de
pressionar
Confirm

Em transações de um só nível:

On BeforeValidate
VALIDAÇÃO
On AfterValidate
On BeforeInsert/BeforeUpdate/ BeforeDelete
GRAVAÇÃO
On AfterInsert/AfterUpdate/ AfterDelete

On BeforeComplete
COMMIT
On AfterComplete

Em transações de dois níveis:

On BeforeValidate
VALIDAÇÃO do cabeçalho
On AfterValidate
On BeforeInsert/BeforeUpdate/BeforeDelete
GRAVAÇÃO do cabeçalho
On AfterInsert/AfterUpdate/AfterDelete

Para cada linha

On BeforeValidate
VALIDAÇÃO da linha
On AfterValidate
OnBeforeInsert/BeforeUpdate/BeforeDelete
GRAVAÇÃO da linha
On AfterInsert/AfterUpdate/AfterDelete

On AferLevel Level **Atributo das linhas**
On BeforeComplete
COMMIT
On AfterComplete

Momentos de disparo de regras

`PrintCustomer(CustomerId)` on **AfterInsert, AfterUpdate**; ¿Está certo ou não?



Está correto!

Transaction
<http://trialapps3.genexus.com/1d8562acf4/>

Attribute-1
Attribute-2
...
Attribute-n

Level

AttributeL-1	AttributeL-2	...	AttributeL-m
data1-1	data1-2	...	<input checked="" type="radio"/>
data2-1	data2-2	...	<input checked="" type="checkbox"/>
data3-1	data3-2	...	<input type="checkbox"/>
data4-1	data 4-2	...	<input checked="" type="checkbox"/>

CONFIRM CANCEL

VALIDAÇÃO

On Aftervalidate

GRAVAÇÃO

On AfterInsert / On AfterUpdate / On AfterDelete

Momentos de disparo de regras

**PrintCustomer(CustomerId) on
AfterDelete;**

¿Está certo ou não?



Não está correto porque é chamado APÓS a exclusão e o cliente não será encontrado com esse ID na tabela

The screenshot shows a "Transaction" window with the URL <http://trialapps3.genexus.com/Id8562acf4/>. The form contains several input fields labeled "Attribute-1", "Attribute-2", and "Attribute-n". Below the inputs is a table labeled "Level" with columns "AttributeL-1", "AttributeL-2", ..., "AttributeL-m". The table has four rows of data: "data1-1", "data1-2", ..., "data1-m" (with a radio button); "data2-1", "data2-2", ..., "data2-m" (with a checked checkbox); "data3-1", "data3-2", ..., "data3-m" (with an empty checkbox); and "data4-1", "data 4-2", ..., "data4-m" (with a checked checkbox). At the bottom of the form are "CONFIRM" and "CANCEL" buttons.

VALIDAÇÃO
On Aftervalidate

GRAVAÇÃO
On AfterInsert / On AfterUpdate / On AfterDelete

Momentos de disparo de regras

```
Error('The seat quantity should be equal or greater than 8') if FlightCapacity<8
on AfterLevel
Level FlightSeatChar;
```

Transaction
http://trialapps3.genexus.com/ld8562acf4/

Attribute-1	
Attribute-2	
...	
Attribute-n	

Level

AttributeL-1	AttributeL-2	...	AttributeL-m
data1-1	data1-2	...	<input checked="" type="radio"/>
data2-1	data2-2	...	<input checked="" type="checkbox"/>
data3-1	data3-2	...	<input type="checkbox"/>
data4-1	data 4-2	...	<input checked="" type="checkbox"/>

< 8

CONFIRM CANCEL

Em transações de dois níveis:

On BeforeValidate

VALIDAÇÃO do cabeçalho

On AfterValidate

On BeforeInsert/BeforeUpdate/BeforeDelete

GRAVAÇÃO do cabeçalho

On AfterInsert/AfterUpdate/AfterDelete

On BeforeValidate

VALIDAÇÃO da linha

On AfterValidate

OnBeforeInsert/BeforeUpdate/BeforeDelete

GRAVAÇÃO da linha

On AfterInsert/AfterUpdate/AfterDelete

On AfterLevel Level **Atributo das linhas**

On BeforeComplete

COMMIT

On AfterComplete

Para cada linha

Momentos de disparo de regras

```
PrintFlight(FlightId) on AfterComplete;
```

✓ on **AfterComplete**:
Imediatamente depois de
realizar o **Commit** no
banco de dados

Em transações de dois níveis:

On BeforeValidate

VALIDAÇÃO do cabeçalho

On AfterValidate

On BeforeInsert/BeforeUpdate/BeforeDelete

GRAVAÇÃO do cabeçalho

On AfterInsert/AfterUpdate/AfterDelete

On BeforeValidate

VALIDAÇÃO da linha

On AfterValidate

OnBeforeInsert/BeforeUpdate/BeforeDelete

GRAVAÇÃO da linha

On AfterInsert/AfterUpdate/AfterDelete

On AfterLevel Level **Atributo das linhas**

On BeforeComplete

COMMIT

On **AfterComplete**

Para cada linha

Exemplos

Determine se está correto ou não:

Invoice

```
{  
    InvoiceId*  
    InvoiceDate  
    -----
```

Product

```
{  
    ProductId*  
    ProductName  
    ProductPrice  
    InvoiceProductQuantity  
    -----  
}
```

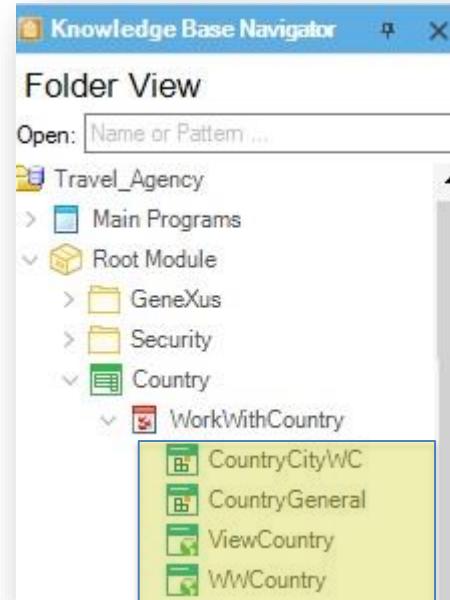
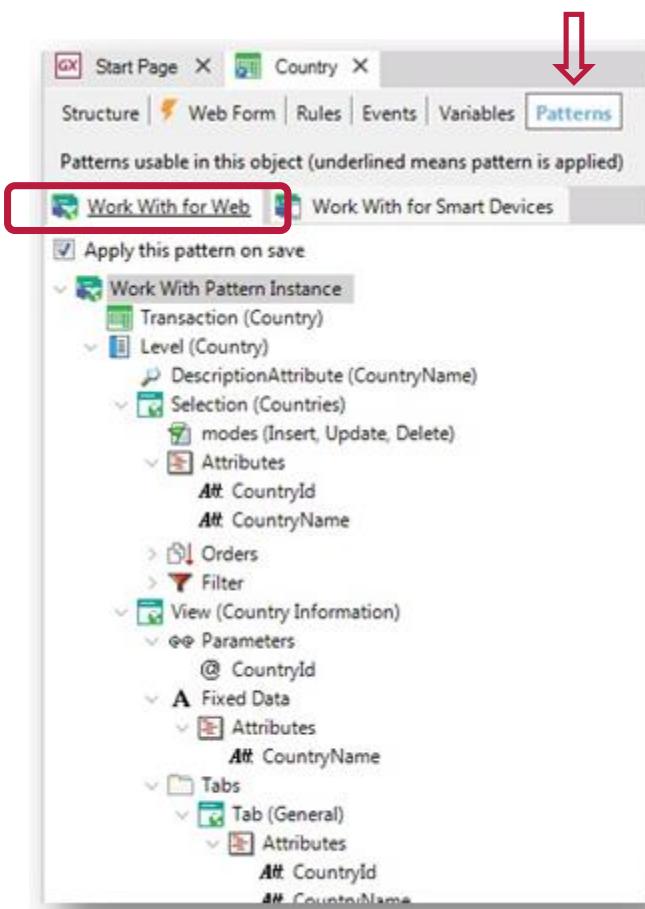
PrintInvoiceDetail(InvoiceId) on **AfterComplete**; 

ProductControl(ProductId) **on BeforeInsert**; 

ProductControl(ProductId) **on AfterComplete**; 

Posso atribuir um valor a um atributo on
AfterInsert?  **NÃO**

Patterns



Gerados automaticamente por GeneXus

Dinamismo



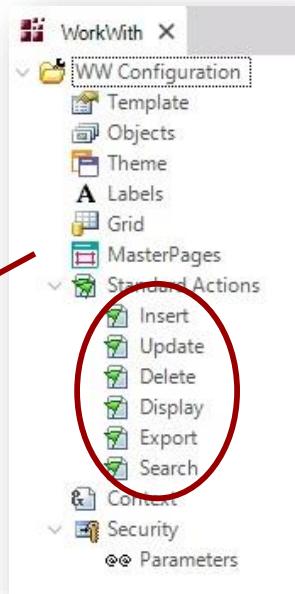
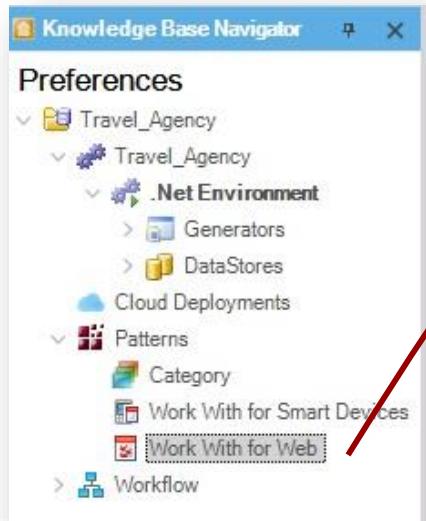
Example: New attribute CountryFlag

The diagram illustrates the dynamic generation of objects from a transaction structure. It shows three main stages:

- Transaction structure:** On the left, a screenshot of a database schema editor shows a **Country** entity with attributes: **CountryId** (Id), **CountryName** (Name), and **CountryFlag** (Image). A red arrow points from this stage to the pattern instance.
- Pattern instance:** In the center, a screenshot of the Genexus pattern editor shows the **Country** entity with the same attributes. A red arrow points from the transaction structure to this stage. The pattern instance includes a **Work With Pattern Instance** configuration panel where the **CountryFlag** attribute is selected under **Attributes**.
- Generated objects:** On the right, a screenshot of a generated web application titled "Travel Agency" shows a list of countries with columns for **ID**, **Name**, and **Flag**. The flag column displays small national flags. Red arrows point from both the transaction structure and the pattern instance to this final stage, indicating the flow from the initial design through the pattern configuration to the final generated user interface.

Configurações Gerais

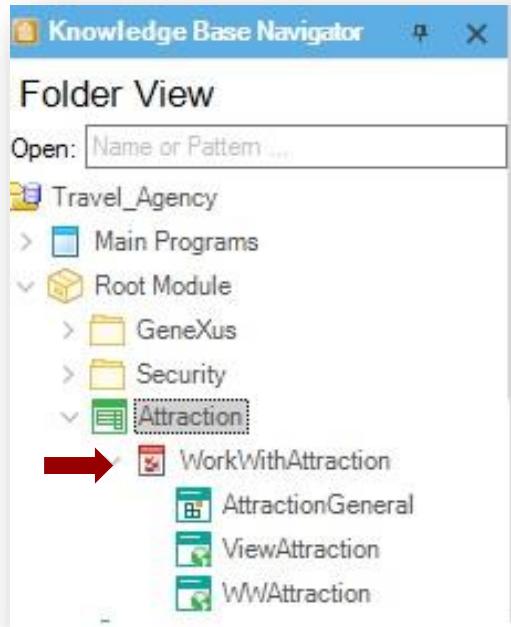
Como são inicializadas todas as instâncias?



Em Country...

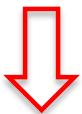
Properties	
modes: Ins: default, Upd: default, ...	
Insert	default
Update	default
Delete	default
Display	default
Export	default

Exclusão da aplicação do pattern

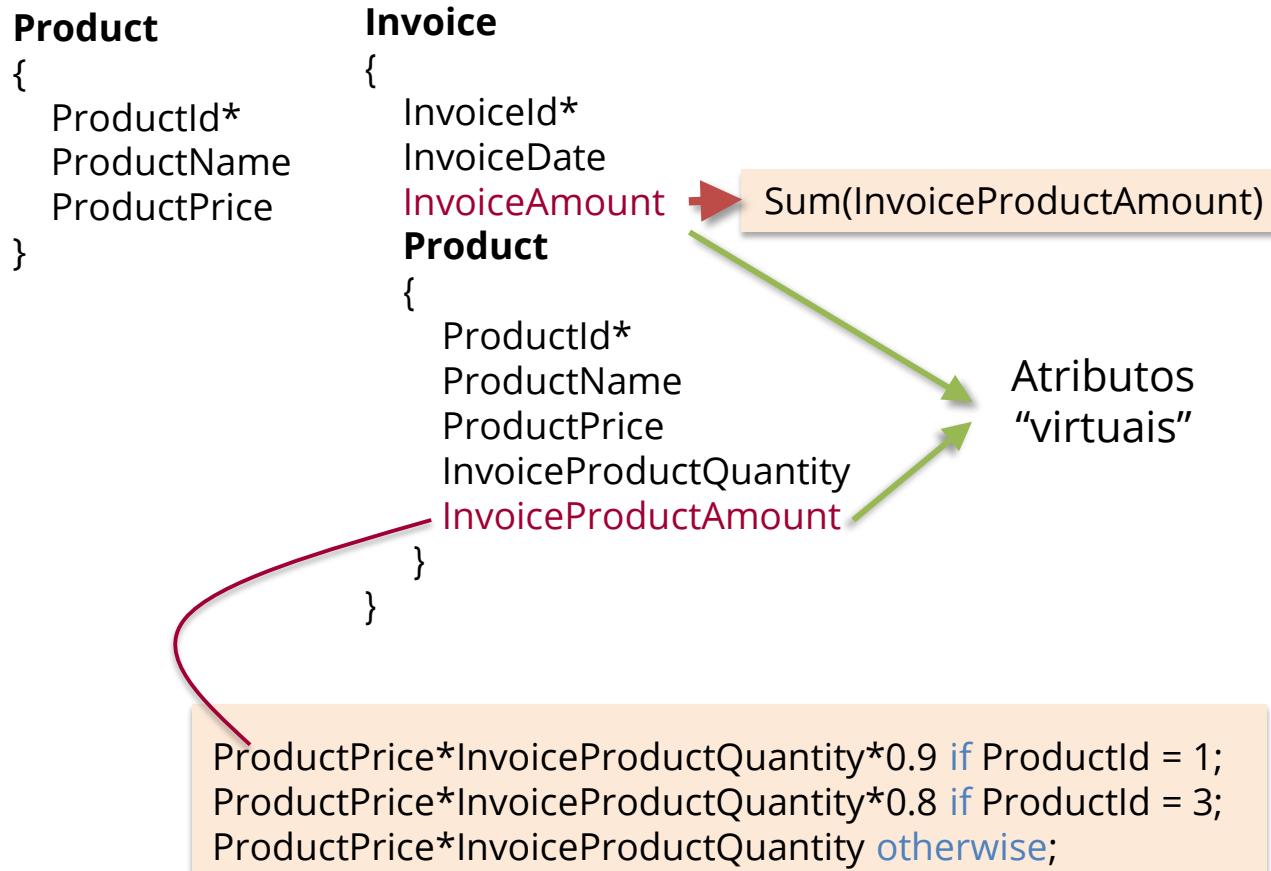


Fórmulas

Fórmulas Globais



- É um cálculo que definimos associado a um atributo
- São acessíveis em toda a KB



```
Country  
{  
    CountryId*  
    CountryName  
    CountryCustomersQuantity  
}
```

```
Customer  
{  
    CustomerId*  
    CustomerName  
    CountryId  
    CountryName  
}
```



Count(CustomerName)

Esta fórmula contará clientes por país ou o número total de clientes?

Contará os clientes por cada país já que GeneXus realiza um filtro automático pelo atributo em comum (*CountryId*).

```
Customer
{
    CustomerId*
    CustomerName
    CustomerTotal
}
```

```
Invoice
{
    InvoiceId*
    InvoiceDate
    InvoiceType
    CustomerId
    CustomerName
    InvoiceAmount
}
```

Dominio InvType
que oferece
dois Enum Values

Credit
Cash



condição de cálculo

Sum(InvoiceAmount, InvoiceType = InvType.Credit)

If CustomerId = 3



condição de disparo

Fórmulas Inline



- Estão escritas no código
- São acessíveis apenas no objeto em que foram definidas

Requisito: Nos pedem um relatório de países com a quantidade de atrações de cada país

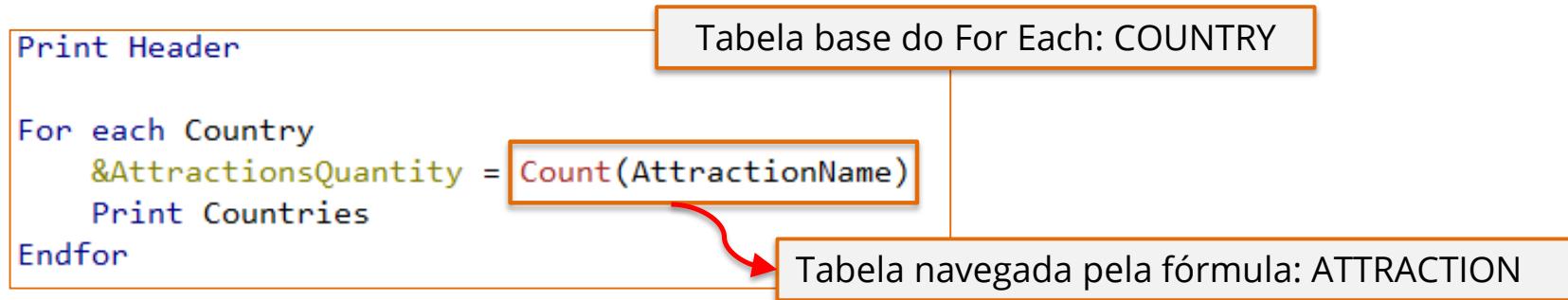
Countries List	
Country	Quantity
Argentina	2
Uruguay	3
Paraguay	1
Estados Unidos	5



```
Country
{
    CountryId*
    CountryName
}
```

```
Attraction
{
    AttractionId*
    AttractionName
    CountryId
    CountryName
}
```

Formula inline no código de um objeto Procedimento



Esta fórmula contará as atrações por país ou o número total de atrações?

Contará as atrações por cada país já que é feito um filtro automático pelo atributo em comum *CountryId* (ambas tabelas estão relacionadas).

For each

Transação Base

```
Flight
{
  FlightId*
  FlightDate
  -----
  Seat
  {
    FlightSeatId*
    FlightSeatChar
  }
}
```

```
For each Flight
-----
Endfor
For each Flight.Seat
-----
Endfor
```

Transação Base

Nome da transação cuja tabela física associada deseja-se percorrer

Ordem

```
Customer
{
    CustomerId*
    CustomerName
    -----
}
```

Requisito: nos pedem um relatório de todos os clientes listados em ordem alfabética.

For each Customer order CustomerName

Endfor

Requisito : nos pedem um relatório de todos os clientes ordenados de forma decrescente pelo nome.

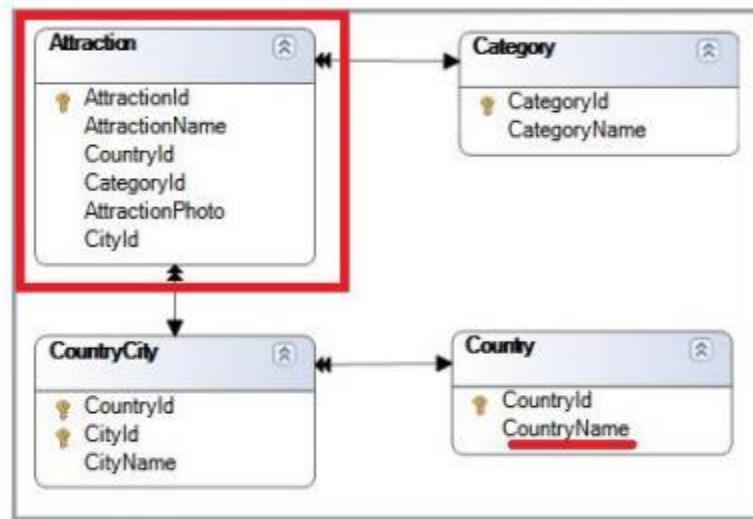
For each Customer order (CustomerName)

Endfor

Ordem

GeneXus permite ordenar pelo valor de um atributo que não está na tabela que estamos percorrendo, mas em sua tabela estendida.

```
Print Header  
For each Attraction order CountryName  
    Print Attractions  
Endfor
```



Filtros com Where

```
Flight
{
    FlightId*
    FlightDate
-----
Seat
{
    FlightSeatId*
    FlightSeatChar
}
}
```

For each Flight

Where FlightDate = Today()

Endfor

For each Flight.Seat

Where FlightId = 1

Endfor

Filtros + Ordem

```
Customer
{
    CustomerId*
    CustomerName
    CustomerAddress
}
```

Index



For each Customer order CustomerName
Where CustomerName >= &NameFrom

Endfor

Warnings

⚠ spc0038 There is no index for order [CustomerName](#); poor performance may be noticed in group starting at line 2.

Customer * X		
Structure	Indexes *	
Attribute	Order	Description
Customer Indexes		Customer
ICustomer	Primary Key	Automatic Index
CustomerId	Ascending	Customer Id
UCustomerName	Duplicate	User Index
CustomerName	Ascending	Customer Name

¡Consulta optimizada!



For each Aninhados + Tabela base diferente + Tabelas NÃO relacionadas = PRODUTO CARTESIANO

```
Country
{
    CountryId*
    CountryName
}
```

```
Room
{
    RoomId*
    RoomName
}
```

Produto Cartesiano

For each **Country**
Print Country

For each **Room**
Print Room
Endfor

Endfor

1 - Brasil	RoomA RoomB RoomC
2 - Uruguay	RoomA RoomB RoomC
3 - Argentina	RoomA RoomB RoomC
4 - Estados Unidos	RoomA RoomB RoomC

For each Aninhados + Tabela base diferente + Tabelas relacionadas = JOIN

Country

```
{  
    CountryId*  
    CountryName  
}
```



Customer

```
{  
    CustomerId*  
    CustomerName  
    CountryId  
    CountryName  
}
```

Join

For each **Country**
Print Country

For each **Customer**
Print Customer
Endfor

Endfor

1 - Brasil	LUIS
2 - Uruguay	JORGE
3 - Argentina	
4 - Estados Unidos	ANA

For each Aninhados + Mesma Tabela base + Tabelas relacionadas = CORTE DE CONTROLE

```
Country
{
    CountryId*
    CountryName
}
```

CORTE DE CONTROLE

For each Customer order CountryId

Print Country

For each Customer

Print Customer

Endfor

Endfor

```
Customer
{
    CustomerId*
    CustomerName
    CountryId
    CountryName
}
```

1 - Brasil	LUIS
	JORGE
4 - Estados Unidos	ANA

Resumo

Produto Cartesiano

For each Country
Print Country

For each Room
Print Room
Endfor

Endfor

Tabelas
diferentes, sem
relacionamento

Join

For each Country
Print Country

For each Customer
Print Customer
Endfor

Endfor

Tabelas
diferentes, com
relacionamento

Corte de Controle

For each Customer order CountryId
Print Country

For each Customer
Print Customer
Endfor

Endfor

Mesma tabela,
Agrupando com
order

Resumo

Produto Cartesiano

1 - Brasil	RoomA RoomB RoomC
2 - Uruguay	RoomA RoomB RoomC
3 - Argentina	RoomA RoomB RoomC

Join

1 - Brasil	LUIS JORGE
2 - Uruguai	
3 - Argentina	
4 - Estados Unidos	ANA

Corte de Controle

1 - Brasil	LUIS JORGE
4 - Estados Unidos	ANA

“Não existe relacionamento entre ambas entidades, mostra todas as possibilidades para cada país”

“Todos os países e seus clientes, independentemente que tenha clientes ou não”

“Somente aqueles países que tenham clientes”

Comunicação entre objetos

Exemplo: passagem de parâmetros

```
Country
{
    CountryId*
    CountryName
}
```

```
Customer
{
    CustomerId*
    CustomerName
    CountryId
    CountryName
}
```

Por exemplo, nas Regras da trn Country:

CustomerList(CountryId) on AfterComplete;



CustomerList

Parm(in: &CountryId);

For each Customer

Where CountryId = &CountryId

Endfor

Variável
Filtro explícito

Exemplo: passagem de parâmetros

```
Country
{
    CountryId*
    CountryName
}
```

```
Customer
{
    CustomerId*
    CustomerName
    CountryId
    CountryName
}
```

Por exemplo, nas Regras da trn Country:

CustomerList(CountryId) on AfterComplete;

CustomerList

Parm(in: CountryId);

For each Customer

~~Where CountryId = &CountryId~~

Endfor



Atributo
Filtro implícito

Exemplo: retorno de um valor

```
Country
{
    CountryId*
    CountryName
}
```

```
Customer
{
    CustomerId*
    CustomerName
    CountryId
    CountryName
}
```

Por exemplo, nas Regras da trn Customer:

&Control = CustomerControl(CustomerId);

CustomerControl

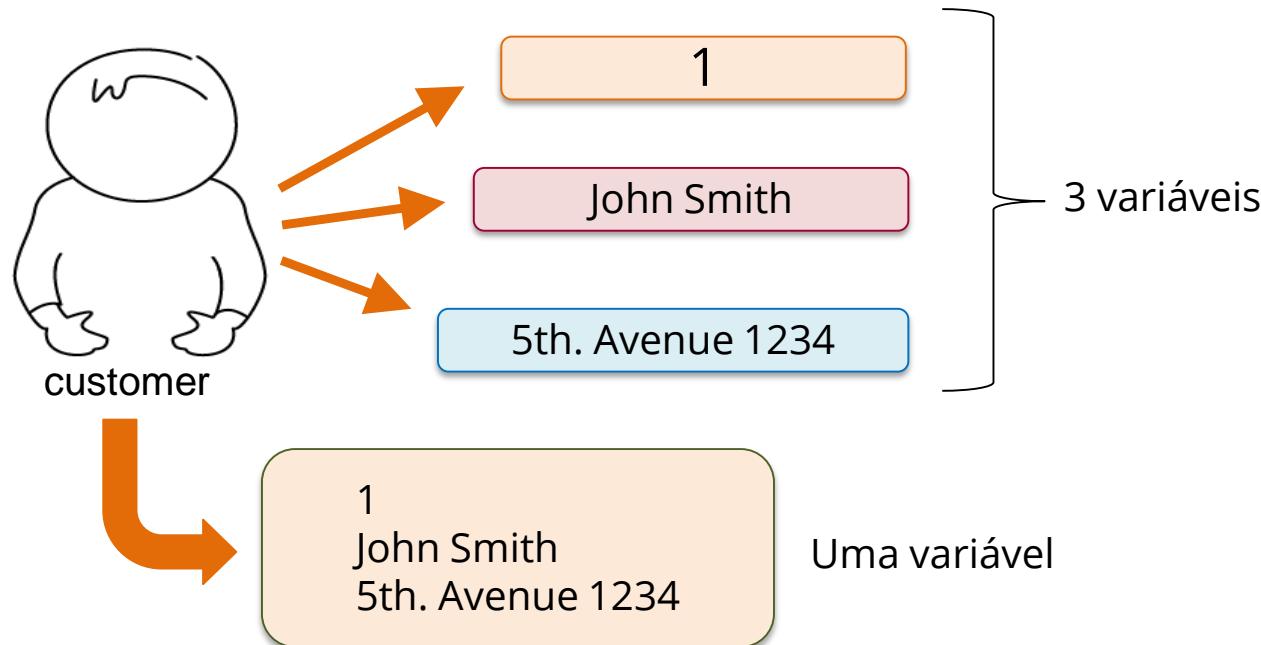
Parm(in: &CustomerId, out: **&Control**);

For each Customer
Where CustomerId = &CustomerId

&Control = True
Endfor

Tipos de Dados Compostos

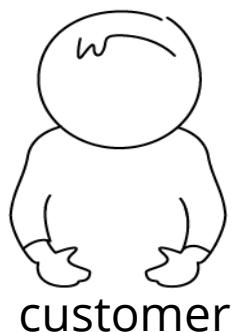
Conceito



Definição

Name	Type	Is Collection
Id	Id	<input type="checkbox"/>
Name	Name	<input type="checkbox"/>
Address	Address, GeneXus	<input type="checkbox"/>

Objeto Structured Data Type



&OneCustomer: SDTCustomer

```
&OneCustomer.Id = 1  
&OneCustomer.Name = 'John Smith'  
&OneCustomer.Address = '5th. Avenue 1234'
```

Data Providers

Id: 1
Name: John Smith
Address: 5th. Avenue 1234

Um cliente

SDTCustomer

Id: 1
Name: John Smith
Address: 5th. Ave.

Id: 2
Name: Susan Brown
Address: 7th.Ave.

Id: 3
Name: Robert Hill
Address: 81th. St..

Id: 4
Name: Peter Jensen
Address: St,Paul Rd.

Coleção de Clientes

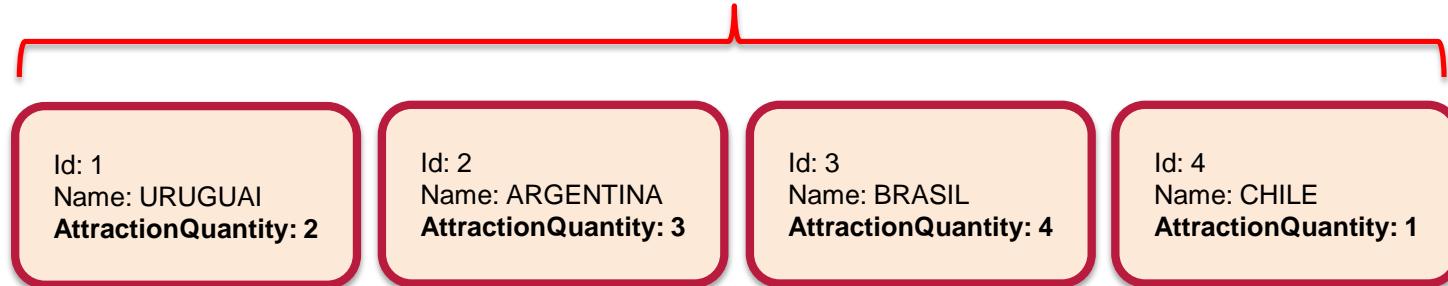
Exemplo: Ranking de quantidade de atrações por país

País	Qtde atrações
BRASIL	4
ARGENTINA	3
URUGUAI	2
CHILE	1
....	
...	

Exemplo: Ranking de quantidade de atrações por país

SDTCountries

Uma coleção de países



Exemplo: Ranking de quantidade de atrações por país

```
Country
{
    CountryId*
    CountryName
}
```

```
Attraction
{
    AttractionId*
    AttractionName
    CountryId
    CountryName
}
```

The screenshot shows the GeneXus interface for defining a structure named "SDTCountries". The "Structure" tab is selected. A table lists the fields: "Name" (SDTCountries), "Type" (SDTCountries), and "Is Collection" (checkbox checked). Under "SDTCountries", there is a child node "SDTCountriesItem" with three fields: "Id" (Type: Id), "Name" (Type: Name), and "CountryAttractionsQuantity" (Type: Numeric(4,0)).

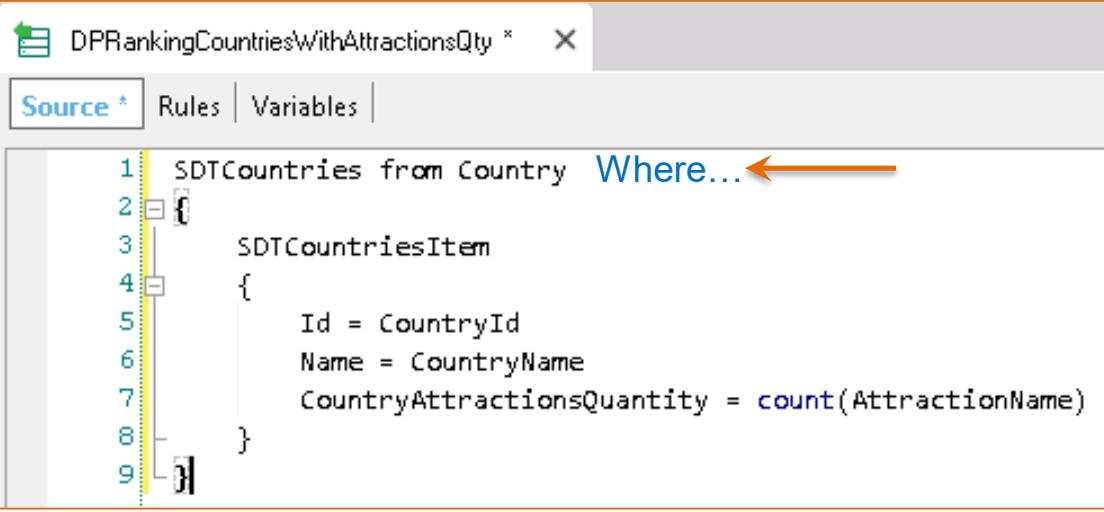
The screenshot shows the GeneXus interface for designing a Data Processing Rule (DPRankingCountriesWithAttractionsQty). The "Source" tab is selected, displaying the following SQL-like query:

```
1 SDTCountries from Country
2 {
3     SDTCountriesItem
4     {
5         Id = CountryId
6         Name = CountryName
7         CountryAttractionsQuantity = count(AttractionName)
8     }
9 }
```

To the right, the "Output" tab is shown with the following settings:

Infer Structure	No
Output	SDTCountries
Collection	False

Exemplo: Ranking de quantidade de atrações por país



The screenshot shows the GeneXus Source editor interface with the title bar "DPRankingCountriesWithAttractionsQty". The tabs "Source", "Rules", and "Variables" are visible, with "Source" being the active tab. The code editor displays the following C#-like pseudocode:

```
1 SDTCountries from Country Where...  
2 {  
3     SDTCountriesItem  
4     {  
5         Id = CountryId  
6         Name = CountryName  
7         CountryAttractionsQuantity = count(AttractionName)  
8     }  
9 }
```

An orange arrow points to the word "Where..." in the first line of the code.

Exemplo: Ranking de quantidade de atrações por país

```
Country
{
    CountryId*
    CountryName
}
```

```
Attraction
{
    AttractionId*
    AttractionName
    CountryId
    CountryName
}
```

PrintCountries * X

Source | Layout | Rules | Conditions | Variables

Subroutines

```

1 &Countries = DPRankingCountriesWithAttractionsQty()
2 &Countries.Sort("[CountryAttractionsQuantity]")
3
4 Print Title
5
6 For &OneCountry in &Countries
7   print Country
8 Endfor
9

```

País	Qtde Atrações
BRASIL	4
ARGENTINA	3
URUGUAI	2
CHILE	1

PrintCountries * X

Source | Layout | Rules | Conditions | Variables *

Name	Type	Is Collection	Description
&Variables			
+& Standard Variables			
● Countries	SDTCountries	<input type="checkbox"/>	Countries
● OneCountry	SDTCountries.SDTCountriesItem	<input type="checkbox"/>	One Country

Variáveis Coleção

&numbers

Name	Type	Is Collection	Description
& Variables			
Standard Variables			
numbers	Numeric(4.0)	<input checked="" type="checkbox"/>	numbers

Name	Type	Description	Is Collection
SDTCountry			
Id	Numeric(4.0)	Id	<input type="checkbox"/>
Name	Character(20)	Name	<input type="checkbox"/>

&Countries

CollectionExample * X			
Web Form Rules Events Conditions Variables *			
Name	Type	Is Collection	Description
& Variables			
Standard Variables			
Countries	SDTCountry	<input checked="" type="checkbox"/>	Countries

Business Components

Conceito: tipo de dados especial baseado em uma transação

The screenshot displays three windows from the Genexus interface:

- Structure Tab:** Shows the schema of the "Category" business component. It contains one category named "Category" with two attributes: "CategoryId" (Id) and "CategoryName" (Name). Both attributes are of type "Category".
- Rules Tab:** Shows a rule definition. Rule 1 is an error message: "Error("Enter de category name, please")". Rule 2 is a condition: "if CategoryName.IsEmpty();".
- Properties Tab:** Shows the properties of the "BusinessComponent: Category" object. A red arrow points to the "Business Component" property, which is set to "True". Other properties include "Name" (Category), "Description" (Category), "Module/Folder" (Root Module), and "Qualified Name" (Category).

Conceito: tipo de dados especial baseado em uma transação

Web Form | Rules | Events | Conditions | **Variables**

Name	Type
Variables	Category
Standard Variables	Category
Category	Category

Properties

BusinessComponent: Category	Category
Name	Category
Description	Category
Module/Folder	Root Module
Business Component	True
Qualified Name	Category

Variable: &Category

Name	Category
Description	Category
Column title	Category
Class	Attribute

Type Definition

Based on	(none)
Data Type	Category
Collection	Image
Initial value	LongVarChar
Value range	Numeric
Validation Failed M...	VarChar
Control Info	Video
Control Type	Extended Types
Input Type	Structured Data Types
	Business Components
	Airline
	Attraction
	Category
	Country
	Country.City
	External Objects

Exemplos: inserção e alteração

Category

```
{  
    CategoryId*  
    CategoryName  
}
```

Insert



Source * Layout | Rules | Conditions | Variables *

Subroutines

```
1 &Category.CategoryId = 1  
2 &Category.CategoryName = "Tourist site"  
3 &Category.Save()  
4 commit
```

Update



Source * Layout | Rules | Conditions | Variables *

Subroutines

```
1 &Category.Load(1)  
2 &Category.CategoryName = "New site"  
3 &Category.Save()  
4 commit
```

Exemplo: exclusão

```
Category  
{  
    CategoryId*  
    CategoryName  
}
```

Delete




The screenshot shows the GeneXus Source editor interface. The top navigation bar includes tabs for Source*, Layout, Rules, Conditions, and Variables*. Below the tabs, a dropdown menu is open, showing options like Subroutines, Methods, and Events. The main area displays three lines of code:

```
1 &Category.Load(1)  
2 &Category.Delete()  
3 commit
```

Métodos Insert e Update

```
Category
{
    CategoryId*
    CategoryName
}
```

Insert



Source * Layout | Rules | Conditions | Variables * |

Subroutines

```
1 &Category.CategoryId = 1
2 &Category.CategoryName = "Tourist site"
3 &Category.Insert()
4 commit
```

Update



Source * Layout | Rules | Conditions | Variables * |

Subroutines

```
1 &Category.CategoryId = 1
2 &Category.CategoryName = "Tourist site"
3 &Category.Update()
4 commit
```

Método InsertOrUpdate

Category

{

 CategoryId*
 CategoryName

}

InsertOrUpdate

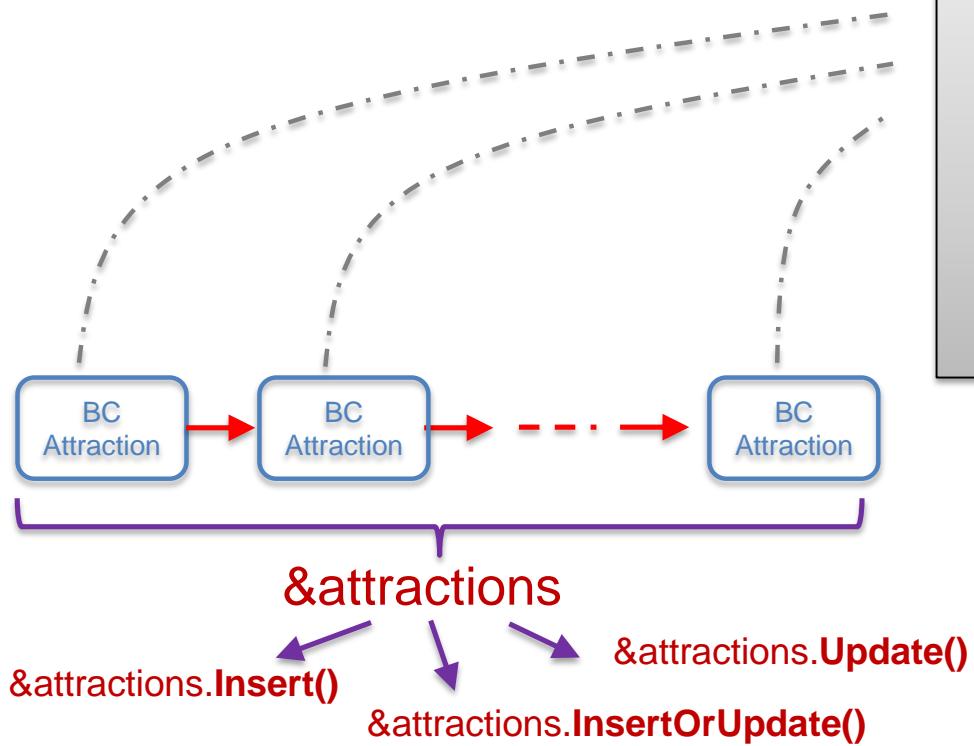


Source * Layout | Rules | Conditions | Variables * |

Subroutines ▾

```
1 &Category.CategoryId = 2
2 &Category.CategoryName = "Tourist site"
3 &Category.InsertOrUpdate()
4 commit
```

Métodos en Colecciones



BC Attraction

AttractionId
AttractionName
CountryId
CityId
CategoryId
AttractionPhoto

Insert / Update / InsertOrUpdate

Recomendamos o uso dos métodos Insert, Update e InsertOrUpdate uma vez que:

- O uso dos métodos Load e Save para modificar o banco de dados é acessado duas vezes - o que diminui o desempenho - enquanto no Update ou InsertOrUpdate é acessado apenas uma vez.
- Os nomes desses novos métodos já são claramente auto-explicativos.

Manuseio de erros trabalhando con BC

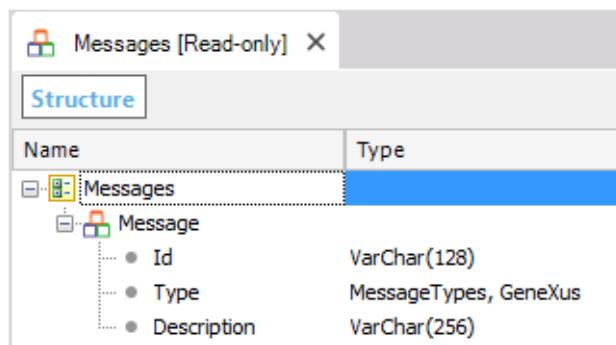
Para cada variável Business Component, uma coleção é carregada na memória com todas as mensagens de aviso ou erro que ocorrem como resultado das operações.



```

Source Layout | Rules | Conditions | Variables | Help
Subroutines
1 &Country.CountryName = "Brasil"
2 &Country.Save()
3
4 &Messages = &Country.GetMessages() ←
5
6 For &oneMessage in &Messages ←
7   msg(&oneMessage.Description)
8 Endfor

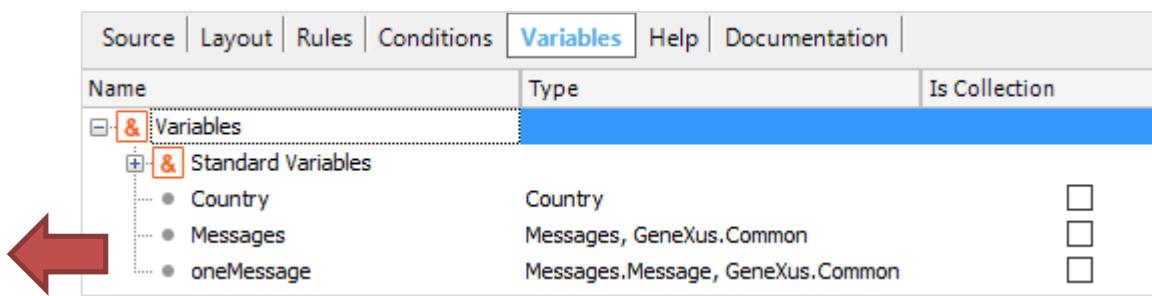
```



Messages [Read-only] X

Structure

Name	Type
Messages	
Message	
Id	VarChar(128)
Type	MessageTypes, GeneXus
Description	VarChar(256)



Source | Layout | Rules | Conditions | **Variables** | Help | Documentation

Name	Type	Is Collection
&Variables		
&Standard Variables		
Country	Country	<input type="checkbox"/>
Messages	Messages, GeneXus.Common	<input type="checkbox"/>
oneMessage	Messages.Message, GeneXus.Common	<input type="checkbox"/>

Carregamento de Dados

Transação

Iniciar dados de forma automática

GeneXus 15 oferece uma alternativa muito fácil para definir os dados com os quais deseja-se popular as tabelas físicas criadas associadas às transações, o que evita ter que criar as cargas de dados de outras formas.

The screenshot shows the GeneXus interface with the 'Category' structure selected. The main window displays the 'Structure' tab, showing a table with columns 'Name', 'Type', and 'Description'. A row for 'Category' is selected, showing 'Category' as the type and 'Category' as the description. Below this, two fields are listed: 'CategoryId' (Id) and 'CategoryName' (Name). A secondary window titled 'Data' is open, showing properties for the selected row. The 'Data Provider' property is set to 'True', and the 'Used to' property is set to 'Populate data'. Both of these properties are highlighted with orange boxes.

Name	Type	Description
Category	Category	Category
CategoryId	Id	Category Id
CategoryName	Name	Category Name

Data	
Data Provider	True
Used to	Populate data
Update Policy	Updatable

Iniciarizar dados

The screenshot shows the GeneXus interface with the following structure:

- Root Module
- GeneXus
- Category
- Category_DataProvider

A red box highlights the "Category" folder, and a red arrow points from it to the "Category_DataProvider" window below.

Category_DataProvider

Source * Rules Variables |

```
1 /*
2 CategoryCollection
3 {
4     Category
5     {
6         CategoryId =
7         CategoryName =
8     }
9 }
10 */
```

Não carregamos o CategoryId
já que foi definido como autonumber

A red arrow points from the text above to this window.

Category_DataProvider

Source Rules Variables |

```
1 CategoryCollection
2 {
3     Category
4     {
5         CategoryName = "Museum"
6     }
7     Category
8     {
9         CategoryName = "Monument"
10    }
11    Category
12    {
13        CategoryName = "Tourist Site"
14    }
15 }
```

Iniciar dados: Somente leitura

Country

```
{  
    CountryId*  
    CountryName  
}
```

Data	
Data Provider	True
Used to	Populate data
Update Policy	Read Only

 United States
 Brazil
 Mexico
 Colombia
 Argentina
 Canada
 Peru
 Venezuela
 Chile
 Ecuador
 Guatemala
 Cuba
 Haiti
 Bolivia
 Dominican Republic
 Honduras
 Paraguay
 Nicaragua
 El Salvador
 Costa Rica
 Panama
 Puerto Rico
 Uruguay
 Jamaica
 Trinidad and Tobago

Carregamento de Dados

Business Components e Data Providers

Exemplo

Country

```
{
    CountryId* ← Autonumber = True
    CountryName
}
```

Web Form Rules | Events | Conditions | Variables |

<No action group selected>

MainTable

Initialize Countries

Events

```
Event 'Initialize Countries'
    &Countries = DPCountries()
    &Countries.Insert()
    commit
Endevent
```

Source Rules | Variables |

```

1 Country
2 {
3     CountryName = "Uruguay"
4 }
5 Country
6 {
7     CountryName = "Argentina"
8 }
9 Country
10 {
11     CountryName = "Brazil"
12 }
```

Output

Infer Structure	No
Output	Country
Collection	True
Collection Name	Countries

Web Form Rules | Events | Conditions | **Variables**

Name	Type	Is Collection	Description
Variables			
Standard Variables			
Countries	Country	<input checked="" type="checkbox"/>	Countries

Carregamento de Dados com Procedimentos Comandos New / For each / Delete

Inserção – Comando NEW

Category

```
{  
    CategoryId*  
    CategoryName  
}
```

```
New  
    CategoryId = 5  
    CategoryName = "Tourist Site"  
Endnew
```

```
New  
    CategoryName = "Tourist Site"  
Endnew
```



Se o atributo é autonumber
não o informamos

Alteração / Comando FOR EACH

```
Category
{
    CategoryId*
    CategoryName
}

For each Attraction
Where CityName = "Beijing" and CategoryName = "Monument"
    CategoryId = find( CategoryId, CategoryName = "Tourist site")
Endfor

Country
{
    CountryId*
    CountryName
}

City
{
    CityId*
    CityName
}

Attraction
{
    AttractionId*
    AttractionName
    CategoryId
    CategoryName
    CountryId
    CountryName
    CityId
    CityName
}
```

Exclusão – Comando DELETE

Attraction

```
{  
    AttractionId*  
    AttractionName  
    CategoryId  
    CategoryName  
    CountryId  
    CountryName  
    CityId  
    CityName  
}
```

```
For each Attraction  
    Delete  
Endfor
```

Considerações

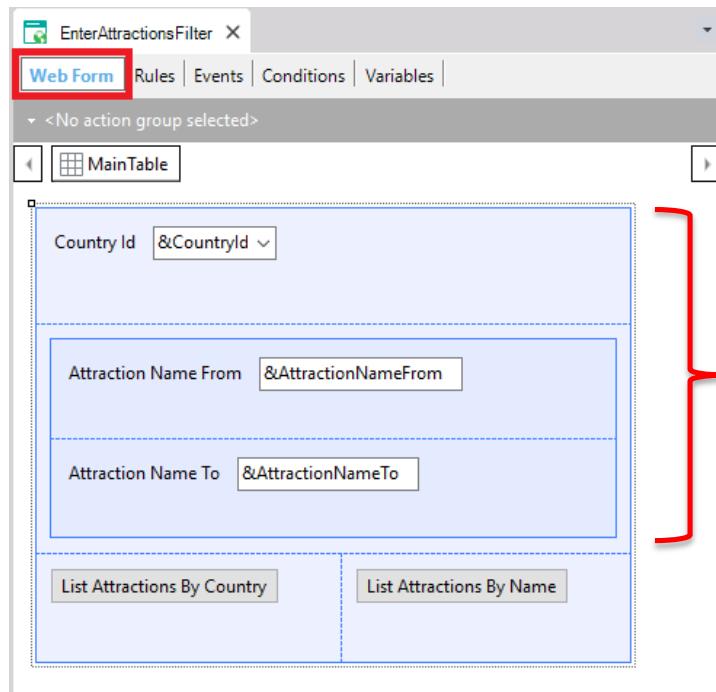
Os comandos New / For Each / Delete são usados para inserir, alterar e excluir informações do banco de dados, mas eles só podem ser usados em Procedimentos.

Embora os comandos a seguir sejam usados para inserir, alterar e excluir informações do banco de dados, recomendamos o uso do Business Component, uma vez que eles:

- Controlam a integridade referencial
- Disparam as regras definidas na transação

Web Panels

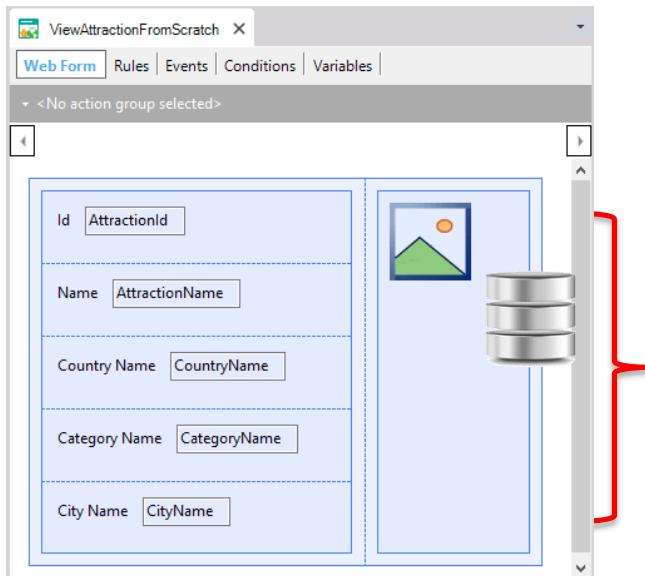
Web Panel sem grid, com variáveis no form



Variáveis: **de entrada**
(não readonly)

Web Panel sem grid mas com atributos no form

Parm(in: AttractionId);



É carregado só **um** registro

Grid: COM TABELA BASE

WwAttractionsFromScratch * X

Web Form * Rules | Events | Conditions | Variables |

<No action group selected>

MainTable Grid1

Country Id &CountryId

Attraction Name From &AttractionNameFrom

Attraction Name To &AttractionNameTo

GRID

Id	Attraction Name	Country	Photo
AttractionId	AttractionName	CountryName	
0			
0			

Properties

General Class

Grid: Grid1

Control Name Grid1

Collection

Base Trn Attraction

Order

Conditions

Data Selector (none)

Appearance

Class Grid

Custom Render

Empty Grid Text

Auto Resize True

Width

Height

Rows 0

Tooltip Text

Layout

Behavior

Transação base

The screenshot illustrates the GeneXus development environment with the following components:

- Web Form (Left):** A form with fields for "Country Id" (dropdown), "Attraction Name From" (dropdown), and "Attraction Name To" (dropdown). Below these is a grid control named "Grid1" with columns: Id, AttractionName, Country, and Photo.
- Subroutine (Source Tab, Middle):** A code editor showing a subroutine named "AttractionsByName". The code prints titles and attraction names from a collection named "Attraction".

```
1 print Title
2 print ColumnTitles
3 For each Attraction <--> CountryName
4   where AttractionName <--> AttrNameFrom
5   where AttractionName <--> AttrNameTo
6   print Attractions
7 endfor
```

A red box highlights the loop condition and body, and a red arrow points from the "Collection" property in the Properties panel to this highlighted code.
- Properties Panel (Right):** Shows properties for "Grid1". The "Collection" property is set to "Base Trn Attraction", which is also highlighted with a red box.

Ordem

The screenshot shows the Genexus IDE interface with the following components:

- Web Form:** A visual representation of the application's user interface. It includes:
 - A dropdown menu for "Country Id" with the value "&CountryId".
 - Text boxes for "Attraction Name From" and "Attraction Name To" with values "&AttractionNameFrom" and "&AttractionNameTo" respectively.
 - A grid control labeled "GRID" with columns: "Id", "Attraction Name", "Country", and "Photo". The "Country" column contains the value "CountryName".
- Source:** A code editor window titled "AttractionsByName" containing the following pseudocode:

```
1 print Title
2 print ColumnTitles
3 For each Attraction order CountryName
4   where AttractionName > &AttractionNameFrom
5   where AttractionName < &AttractionNameTo
6   print Attractions
7 endfor
```
- Properties:** A panel on the right showing properties for "Grid: Grid1".
 - Control Name: Grid1
 - Collection: Attraction
 - Order: CountryName (highlighted in blue)
 - Conditions: (none)
 - Data Selector: (none)
 - Height: 0
 - Rows: 0
 - Tooltip Text: (none)
- Grid1's Order:** A modal dialog box showing the selected order: "CountryName". It has "OK" and "Cancel" buttons.

A red arrow points from the "Order" field in the Properties panel to the "CountryName" entry in the "Grid1's Order" dialog.

Grid: COM TABELA BASE

WwAttractionsFromScratch

Web Form * Rules Events Conditions Variables

<No action group selected>

MainTable Grid1

Country Id &CountryId

Attraction Name From &AttractionNameFrom

Attraction Name To &AttractionNameTo

GRID

Id	Attraction Name	Country	Photo
AttractionId	AttractionName	CountryName	
0			
0			

Properties

General Class

A Z Filter X

Grid: Grid1

Control Name	Grid1
Collection	
Base Trn	Attraction
Order	CountryName
Conditions	CountryId = &CountryId;
Data Selector	(none)

Appearance

X Grid

True

Height

Rows 0

Tooltip Text

Layout

CountryId = &CountryId when not &CountryId.IsEmpty();

Condições de filtro

The screenshot shows the Genexus IDE interface with several windows open:

- Web Form**: Shows three input fields: "Country Id" with value "&CountryId", "Attraction Name From" with value "&AttractionNameFrom", and "Attraction Name To" with value "&AttractionNameTo".
- Source**: Displays a Subroutine with the following code:

```
1 print Title
2 print ColumnTitles
3 for each Attraction order CountryName
4   where CountryId = &CountryId
5   print Attractions
6 endfor
```

A red arrow points from the condition in line 4 to the "Conditions" section in the Properties panel.
- Properties**: The "Grid: Grid1" properties are shown, specifically the "Conditions" section which contains the condition `CountryId = &CountryId;`. A red box highlights this section.
- Grid1**: A grid control with columns: Id (AttractionId), Attraction Name (AttractionName), Country (CountryName), and Photo. The "Photo" column contains a small thumbnail image.

Muitas condições

The screenshot shows the GeneXus IDE interface. On the left, the **Properties** window is open for a control named **Grid1**. The **Conditions** section contains the following code:

```
CountryId = &CountryId  
when not &CountryId.IsEmpty();  
  
AttractionName >= &AttractionNameFrom  
when not &AttractionNameFrom.IsEmpty();  
  
AttractionName <= &AttractionNameTo  
when not &AttractionNameTo.IsEmpty();
```

A red curved arrow points from the word "Conditions" in the Properties window to the first condition line in the dialog.

The **Grid1's Conditions** dialog box is displayed, containing the same three conditions. It has **OK** and **Cancel** buttons at the bottom.

At the bottom of the screen, there are tabs for **Properties**, **Toolbox**, and **Responsive Sizes**.

Eventos

Country Id	&CountryId					
Attraction Name From &AttractionNameFrom						
Attraction Name To &AttractionNameTo						
GRID						
Id AttractionId	Attraction Name AttractionName	Country CountryName	Photo 	Trips &trips		&newTrip
Total Trips &totalTrips						

First time

Start

Refresh

Load

User / Control Event

Evento Load em Web Panel COM tabela base

The screenshot shows the Genexus Web Form designer interface. At the top, there's a toolbar with icons for file operations and a title bar 'WwAttractionsFromScratch'. Below the toolbar, a navigation bar includes 'Web Form *' (which is selected), 'Rules', 'Events', 'Conditions', and 'Variables'. A message 'No action group selected' is displayed. In the main area, there are three input fields: 'Country Id' with a dropdown menu '&CountryId', 'Attraction Name From' with a dropdown menu '&AttractionNameFrom', and 'Attraction Name To' with a dropdown menu '&AttractionNameTo'. Below these fields is a 'GRID' section containing four columns: 'Id' (AttractionId), 'Attraction Name' (AttractionName), 'Country' (CountryName), and 'Photo' (an image placeholder). The 'Photo' column contains a small thumbnail of a landscape scene with a sun and mountains.



Evento LOAD

"N vezes,
tantas
quantos
registros
existirem na
tabela
percorrida"

Outro exemplo

```
Country
{
    CountryId*
    CountryName
}
```

```
Attraction
{
    AttractionId*
    AttractionName
    CountryId
    CountryName
}
```

The screenshot shows the GeneXus Web Form designer interface. At the top, there are tabs: 'Web Form *' (which is selected), 'Rules', 'Events', 'Conditions', and 'Variables *'. Below the tabs, a message says '<No action group selected>'. A 'MainTable' component is selected, indicated by a blue border. Inside the MainTable, there is a 'GRID' component. The GRID has three columns: 'Country Id' (containing 'CountryId'), 'Country Name' (containing 'CountryName'), and 'Quantity' (containing '&Quantity'). The 'Quantity' column is highlighted with a red border.

Event **Load**
&Quantity = Count(AttractionName)
endevent

Evento Refresh

Travel Agency

CATEGORIES COUNTRIES ATTRACTIONS ▾

Country Id France

Attraction Name From

Attraction Name To

Attraction Name Country Attraction Photo Trips

Attraction Name	Country	Attraction	Photo	Trips
Christ the Redemer	Brazil			1
Eiffel Tower	France			2
Forbidden city	China			0
Matisse Museum	France			1
Meet the Emperor	China			0
Total Trips				7

2
1

Refresh (once)
Load (2 times)

Event Load
 $\&trips = \text{count(TripDate)}$
 $\&totalTrips = \&totalTrips + \&trips$
Endevent

Event Refresh
 $\&totalTrips = 0$
Endevent

Travel Agency

CATEGORIES COUNTRIES ATTRACTIONS ▾

Country Id France

Attraction Name From

Attraction Name To

Attraction Name Country Attraction Photo Trips

Attraction Name	Country	Attraction	Photo	Trips
Eiffel Tower	France			2
Matisse Museum	France			1
Total Trips				3

Atributos na Grid

Screenshot of the GeneXus IDE showing the 'Attraction' screen in the 'Navigation View' tab.

The interface includes:

- Top navigation bar: Start Page, WWAttractionsFromScratch*, Navigation View, Attraction.
- Toolbars: Web Form*, Rules, Events*, Conditions, Variables.
- Left sidebar: <No action group selected>, MainTable, Grid1, AttractionId.
- Form area:
 - Country Id: &CountryId
 - Attraction Name From: &AttractionNameFrom
 - Attraction Name To: &AttractionNameTo
 - GRID section:
 - Columns: Id, AttractionId, Attraction Name, AttractionName, Country, CountryName, Photo, Trips, &trips.
 - Image: A red arrow points from the 'AttractionId' column header to the 'Visible' property in the Properties panel.
 - Total Trips: &totalTrips
- Properties panel (open for AttractionId):
 - General tab:
 - Attribute: AttractionId
 - Title: Id
 - Class: Attribute
 - Column Class:
 - Return On Click: False
 - On Click Event:
 - Control Info tab:
 - Control Type: Edit
 - Input Type: Values
 - Notify Context Char: False
 - Behavior tab:
 - Input History: True
 - Is Password: False
 - Read Only: True
 - Empty as null: Yes
 - Appearance tab:
 - Auto Resize: True
 - Format: Text
 - Visible: **False** (highlighted)
 - Tooltip Text:
 - Invite Message:
- Output panel: Show: Build, Find: Autoscroll.
- Bottom tabs: Properties, Toolbox.

Web panels
sem tabela base

Web Panels SEM TABELA BASE

Country Id &CountryId

Attraction Name From &AttractionNameFrom

Attraction Name To &AttractionNameTo

GRID

Attraction Id	Attraction Name	Country	Photo	Trips	
&AttractionId	&AttractionName	&CountryName		&trips	 &newTrip

Total Trips &totalTrips

Evento LOAD
“1 vez”



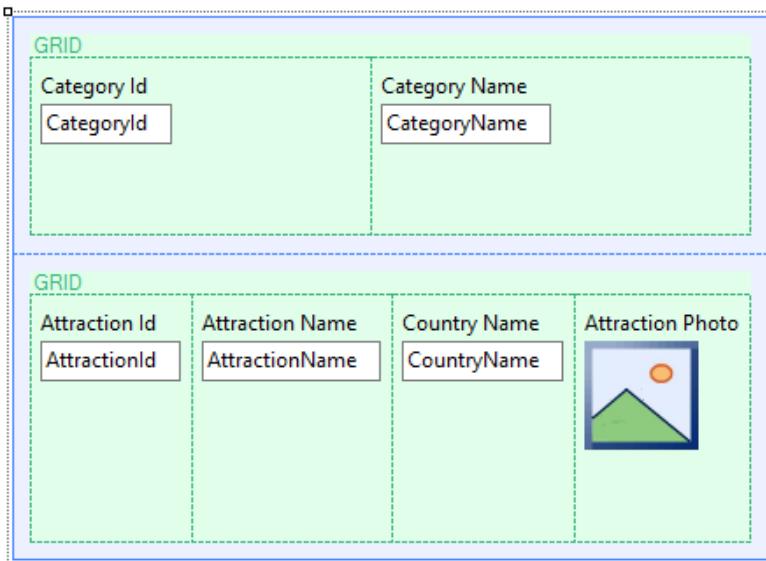
```

For each Attraction
    order CountryId, AttractionName when not &CountryId.IsEmpty()
    order AttractionName
    where CountryId = &CountryId when not &CountryId.IsEmpty()
    where AttractionName >= &AttractionNameFrom when not &AttractionNameFrom.IsEmpty()
    where AttractionName <= &AttractionNameTo when not &AttractionNameTo.IsEmpty()
    &AttractionId = AttractionId
    &AttractionName = AttractionName
    &CountryName = CountryName
    &AttractionPhoto = AttractionPhoto
    &trips = count( TripDate )
    Load
    &totalTrips = &totalTrips + &trips
endfor
Endevent
  
```



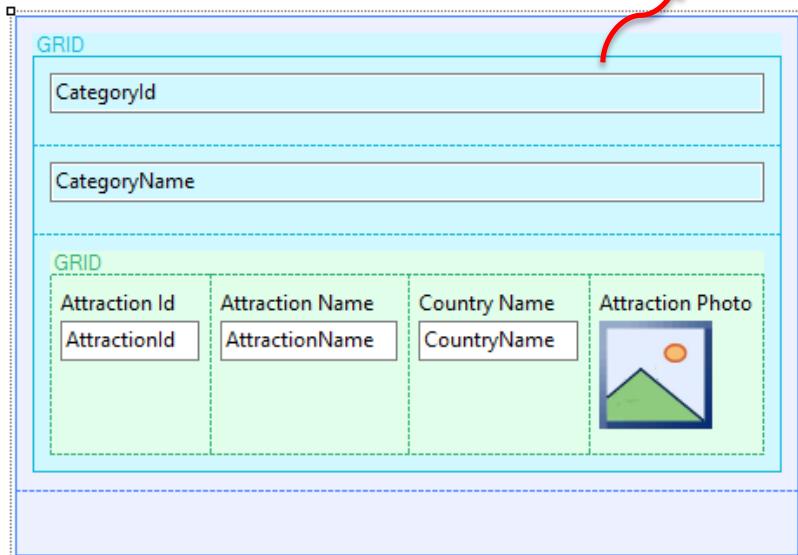
Web panels Múltiplas grids

Múltiplas grids



PARALELO

Navegação independente



ANINHADO

Navegação com tabelas relacionadas

Tipos de Web Panels

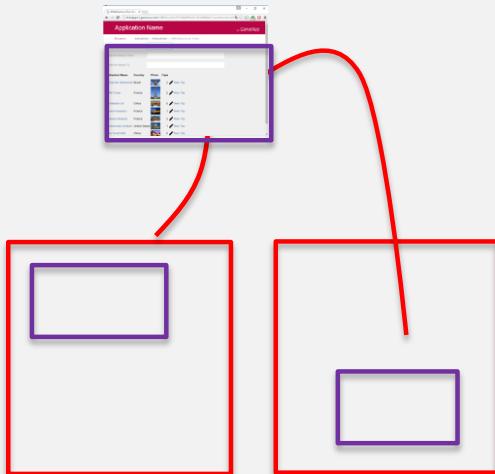
Tipos de Web Panels

Web page (default)

A screenshot of a GeneXus application window titled "Application Name". The interface includes a navigation bar with links like "Attraction — Attractions — WWWAttractions From...". Below the navigation, there are dropdown menus for "Country Id" and "Attraction Name From" and "Attraction Name To". A table lists attractions with columns for "Attraction Name", "Country", "Photo", and "Trips". The table contains the following data:

Attraction Name	Country	Photo	Trips
Christ the Redeemer	Brazil		2 ↗ New Trip
Eiffel Tower	France		2 ↗ New Trip
Forbidden city	China		0 ↗ New Trip
Louvre Museum	France		1 ↗ New Trip
Matisse Museum	France		2 ↗ New Trip
Smithsonian Institute	United States		1 ↗ New Trip
The Great Wall	China		0 ↗ New Trip

Component



Master page

A screenshot of a GeneXus application window titled "Attraction". The interface includes a navigation bar with links like "Attraction — Attractions — Attractions". Below the navigation, there are several input fields and a photo viewer. The fields include:

- Id: 24
- Name: Eiffel Tower
- Country Id: 2
- Country Name: France
- Category Id: 11
- Category Name: Monument
- Photo:

Design Systems

Design Systems

Elementos que dão uniformidade e coerência ao UX

Master Page

e.g. Header / footer

Theme

Classes

Responsive sizes

Responsive tables

RWD

Controls

User control objects

Base styles

CSS libraries

Stencils

Design components

Patterns

Design / behaviour

Stencils

Select a Category: Select a Type:

Common	Color Palette
Workflow	Data Provider
Reporting	Data Selector
Documentation	Data View
Web	Domain
Extensibility	External Object
Deploy	Image
Chatbots	Language
Smart Devices	Procedure
Stencil	
Structured Data Type	

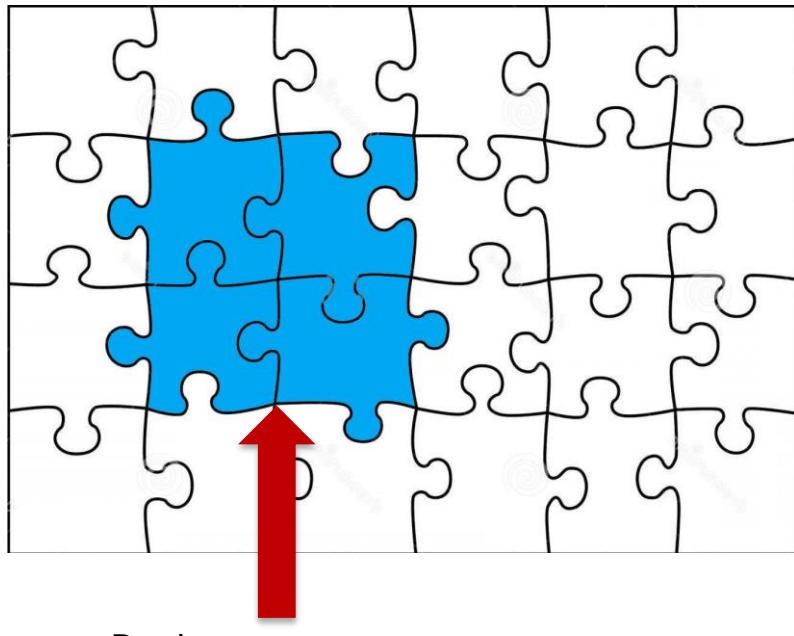
Create a new Stencil

Name: Stencil1

Description: Stencil1

Module/Folder: Root Module

Create **Cancel**



Objeto que permite repetir o desenho da mesma parte da tela (um conjunto de controles), em várias telas.

Responsive Web Design

Responsive Web Design

Attractions

trialapps3.genexus.com/ld8562acf4c97c0fe8095a5c284d688e07/wwattraction.aspx

Application Name by Genexus

Recents Attraction — Attractions

X HIDE FILTERS Attractions Name + INSERT

Ordered By: Name

COUNTRY NAME

	ID	Name	Country Name	Category Name	Photo	City Name	Actions
25	Christ the Redemmer	Brazil	Monument		Rio de Janeiro	UPDATE DELETE	
24	Eiffel Tower	France	Monument			UPDATE DELETE	
28	Forbidden city	China	Tourist Site			UPDATE DELETE	
22	Louvre Museum	France	Museum			UPDATE DELETE	
27	Matisse Museum	France	Museum			UPDATE DELETE	
26	Smithsonian Institute	United States	Museum			UPDATE DELETE	
23	The Great Wall	China	Tourist Site			UPDATE DELETE	

Attractions

trialapps3.genexus.com/ld8562acf4c97c0fe8095a5c284d688e07/wwattraction.aspx

Application Name by Genexus

Recents Attraction — Attractions

X HIDE FILTERS Attractions Name + INSERT

Ordered By: Name

COUNTRY NAME

Name	Actions
Christ the Redemmer	UPDATE DELETE
Eiffel Tower	UPDATE DELETE
Forbidden city	UPDATE DELETE
Louvre Museum	UPDATE DELETE
Matisse Museum	UPDATE DELETE
Smithsonian Institute	UPDATE DELETE

Attractions

trialapps3.genexus.com/ld8562acf4c97c0fe8095a5c284d688e07/wwattraction.aspx

Application Name by Genexus

Recents

Attractions + INSERT

Name

Name	Actions
Christ the Redemmer	UPDATE DELETE
Eiffel Tower	UPDATE DELETE
Forbidden city	UPDATE DELETE
Louvre Museum	UPDATE DELETE
Matisse Museum	UPDATE DELETE
Smithsonian Institute	UPDATE DELETE

Responsive Web Design

WwAttractionsFromScratch X

Web Form | Rules | Events | Conditions | Variables |

< No action group selected >

MainTable CountryId

Country Id &CountryId

Attraction Name From &AttractionNameFrom

Attraction Name To &AttractionNameTo

GRID

Id	Attraction Name	Country	Photo	Trips	&trips	&newtrip
AttractionId	AttractionName	CountryName		&trips		&newtrip

Total Trips &totalTrips

Responsive Sizes

Selection Table: MainTable

Size: Medium (Desktop >= 992 px) inherits from Small

Table Default

Label Width: 25%

1,1 &CountryId
2,1 Table1
3,1 Grid1
4,1 &totalTrips

Values

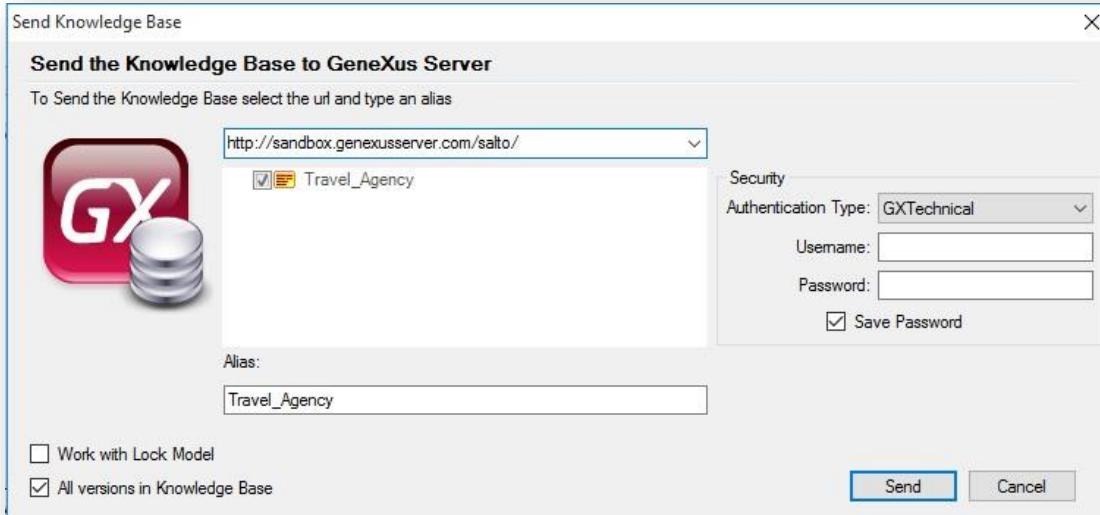
Width: 100% Label Width: 25% Offset: 0% Visible: True Move: < > Default

Use default values for all screen sizes

The screenshot shows the GeneXus interface for a 'Web Form' titled 'WwAttractionsFromScratch'. On the left, there's a visual editor with several controls: a dropdown for 'Country Id' (labeled '&CountryId'), two text inputs for attraction names ('Attraction Name From' and 'Attraction Name To'), and a grid control labeled 'GRID'. The grid has columns for 'Id', 'Attraction Name', 'Country', 'Photo' (with thumbnail images), 'Trips', and '&trips'. At the bottom, there's a text input for 'Total Trips' with the value '&totalTrips'. On the right, the 'Responsive Sizes' panel is open, showing a table structure for different screen sizes. Red arrows point from the 'GRID' label and the '&totalTrips' input field to their corresponding entries in the responsive table. The 'Size' dropdown in the top right of the panel is set to 'Medium (Desktop >= 992 px) inherits from Small'. The 'Values' section on the right side of the panel shows width, label width, offset, visibility, and move options.

GeneXus Server

Send Knowledge Base to GeneXus Server



Team Development

Team Development X

Commit to: http://sandbox.genexusserver.com/salto/home.aspx?Travel_Agency.0

Pattern: Recent Comments

Category: *ALL Folder: *ALL

Pending Commits (8/8) Ignored Objects Refresh

Drag a column here to group by that column

<input checked="" type="checkbox"/>	Name	Type	Description	Modified On	Module	Action	Last Synchroniz	User
<input checked="" type="checkbox"/>	Airline	Transaction	Airline	20/6/2016 9:33	Root Module	Modified	20/6/2016 9:30	ARTECH\acaggia...
<input checked="" type="checkbox"/>	AirlineFlight1WC	Web Component	Airline Flight1 WC	20/6/2016 9:33	Root Module	Inserted	20/6/2016 9:30	ARTECH\acaggia...
<input checked="" type="checkbox"/>	AirlineFlightWC	Web Component	Airline Flight WC	20/6/2016 9:33	Root Module	Inserted	20/6/2016 9:30	ARTECH\acaggia...
<input checked="" type="checkbox"/>	AirlineGeneral	Web Component	Airline General	20/6/2016 9:33	Root Module	Inserted	20/6/2016 9:30	ARTECH\acaggia...
<input checked="" type="checkbox"/>	AttractionsAndCategor...	Procedure	Attractions And Ca...	20/6/2016 9:33	Root Module	Modified	20/6/2016 9:30	ARTECH\acaggia...
<input checked="" type="checkbox"/>	ViewAirline	Web Panel	View Airline	20/6/2016 9:33	Root Module	Inserted	20/6/2016 9:30	ARTECH\acaggia...
<input checked="" type="checkbox"/>	WorkWithAirline	Work With for W...	Work With Airline	20/6/2016 9:33	Root Module	Inserted	20/6/2016 9:30	ARTECH\acaggia...
<input checked="" type="checkbox"/>	WWAirline	Web Panel	Work With Airline	20/6/2016 9:33	Root Module	Inserted	20/6/2016 9:30	ARTECH\acaggia...

Add Knowledge Base properties to list Remind me to move changes to... Cancel Commit

Commit Update History Versions

Commit

Knowledge Manager / Team Development

Team Development X

Commit to: http://sandbox.genexusserver.com/salto/home.aspx?Travel_Agency_0

Pattern:

Category: *ALL Folder: *ALL

Pending Commits (8/8) Ignored Objects

Drag a column here to group by that column

<input checked="" type="checkbox"/>	Name	Type	Description	Modified On
<input checked="" type="checkbox"/>	Airline	Transaction	Airline	20/6/2016 9:33
<input checked="" type="checkbox"/>	AirlineFlight1WC	Web Component	Airline Flight1 WC	20/6/2016 9:33
<input checked="" type="checkbox"/>	AirlineFlightWC	Web Component	Airline Flight WC	20/6/2016 9:33
<input checked="" type="checkbox"/>	AirlineGeneral	Web Component	Airline General	20/6/2016 9:33
<input checked="" type="checkbox"/>	AttractionsAndCategor...	Procedure	Attractions And Ca...	20/6/2016 9:33
<input checked="" type="checkbox"/>	ViewAirline	Web Panel	View Airline	20/6/2016 9:33
<input checked="" type="checkbox"/>	WorkWithAirline	Work With for W...	Work With Airline	20/6/2016 9:33
<input checked="" type="checkbox"/>	WWAirline	Web Panel	Work With Airline	20/6/2016 9:33

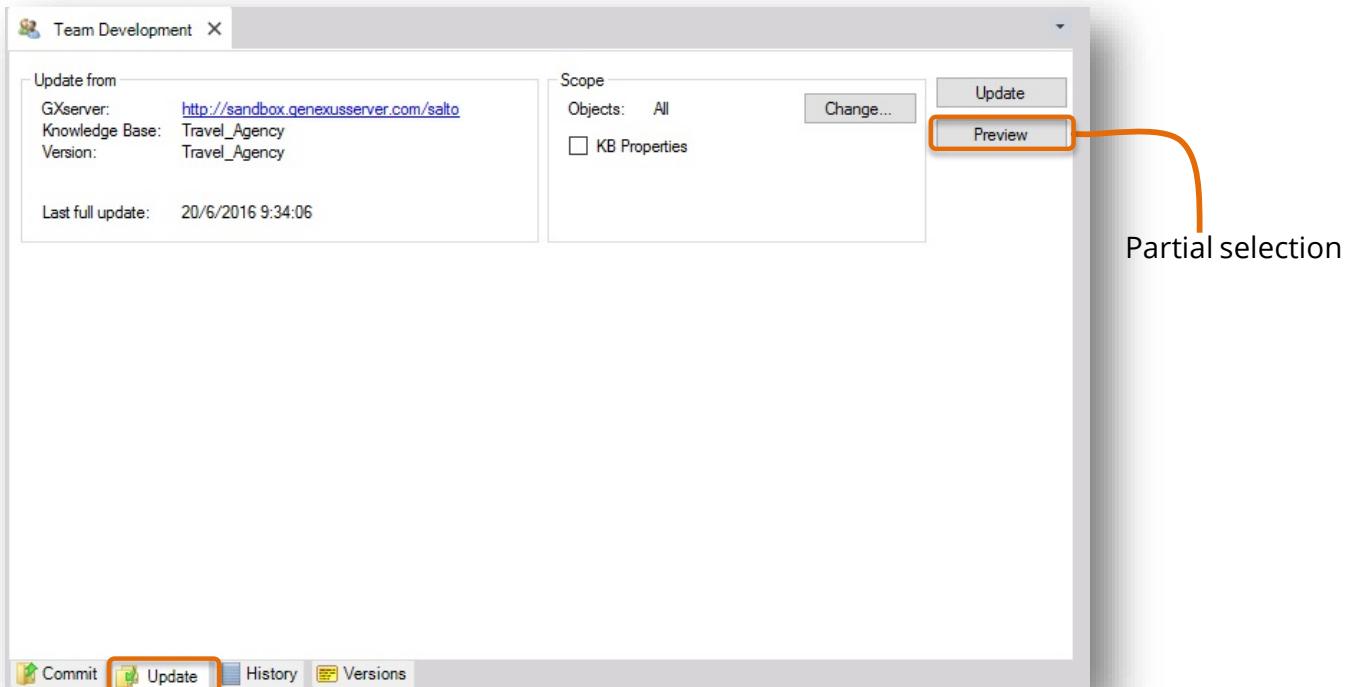
Add Knowledge Base properties to list Remind me to

 Commit  Update  History  Versions

Partial commit



Update



Partial selection

History

Team Development X

Version: Travel_Agency Refresh

From: To: Search: Apply

Drag a column here to group by that column

#	Comment	User	Commit Date
3	WorkWithAirline and AttractionsAndCategories listing	GXTechnical\acaggiano	20/06/2016 10:10 a. m.
1	Travel_Agency is now hosted by GeneXus Server	GXTechnical\acaggiano	20/06/2016 09:33 a. m.

#3 - 20/6/2016 10:10:22 - GXTechnical\acaggiano

WorkWithAirline and AttractionsAndCategories listing

	Name	Type	Description	Action
	Airline	Transaction	Airline	Modified
	AirlineFlight1WC	Web Panel	Airline Flight1 WC	Inserted
	AirlineFlightWC	Web Panel	Airline Flight WC	Inserted
	AirlineGeneral	Web Panel	Airline General	Inserted
	AttractionsAndCategories	Procedure	Attractions And Categories	Modified
	ViewAirline	Web Panel	View Airline	Inserted
	WorkWithAirline	Work With for Web	Work With Airline	Inserted

Commit Update History Versions

Create KB from GeneXus Server

File / New / Knowledge Base from Server



Create Knowledge Base from GeneXus Server

Server Knowledge base

Server KB URL: Select Server KB...

Trunk version
 All versions
 Selected versions:

Local Knowledge base

Name: Knowledge Base
Path: C:\KBTravelAgency

Select Knowledge Base from GeneXus Server

Servers: genexusversalto Open Sandbox Add New Server...

Knowledge Bases: MGC Sto Travel_Agency

Search by pattern here...

Knowledge Base Info:

MGC
Created by GXTechnicalibenedetto
marzo 02, 2016
No description available

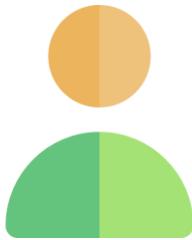
Tags: Untagged

Name: MGC

Select Cancel

Select KB from server

Segurança com GAM

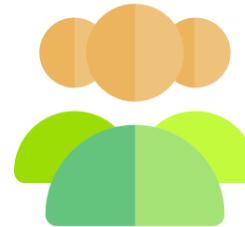
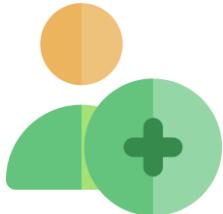


AUTHENTICATION

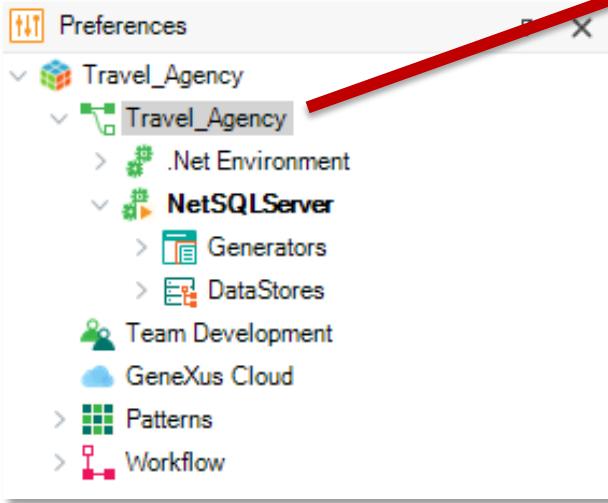


AUTHORIZATION

GeneXus™
ACCESS MANAGER



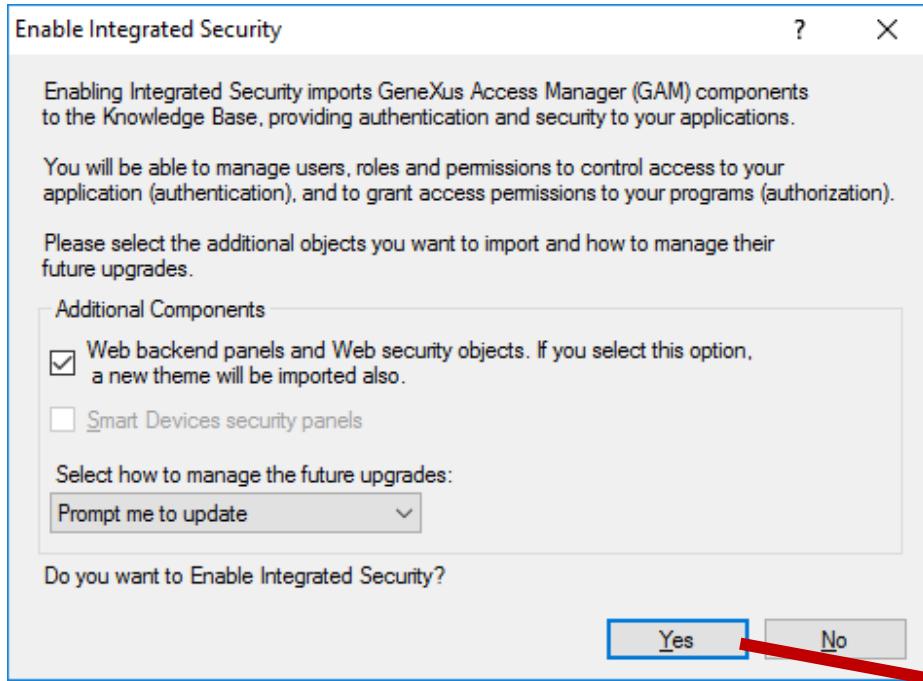
Habilitando o GeneXus Access Manager



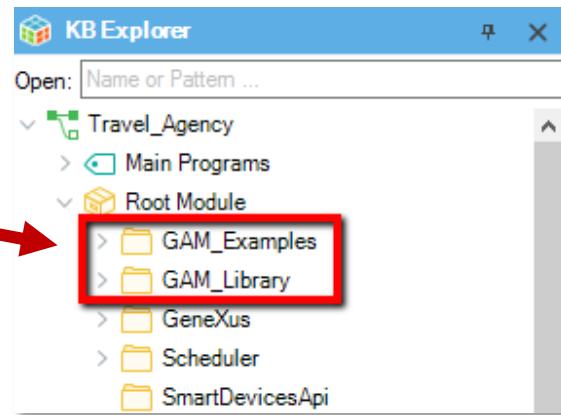
The screenshot shows the 'Properties' dialog box for the 'Travel_Agency' version. The 'Enable Integrated Security' setting is highlighted with a red box. The table below lists various configuration settings:

Setting	Value
Significant attribute name length	30
Significant table name length	30
Significant object name length	128
Preserve Table Casing	True
Generate prompt programs	Yes
LIKE escape character	None
Enable Integrated Security	False

The right pane of the dialog box lists other sections such as Compatibility, User interface, Defaults, Images, Team Development, Workflow, and Patterns.



Importando os objetos do GAM



Properties

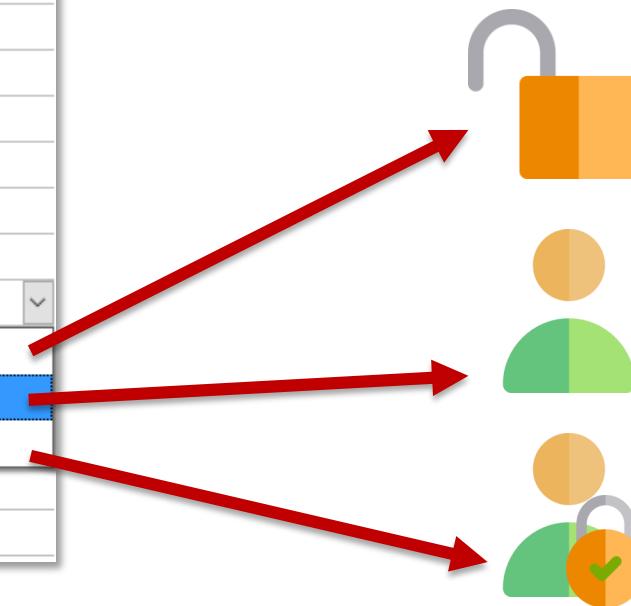
Version: Travel_Agency

Significant attribute name length	30
Significant table name length	30
Significant object name length	128
Preserve Table Casing	True
Generate prompt programs	Yes
LIKE escape character	None
Enable Integrated Security	True

Integrated Security

Integrated Security Level	Authentication
Application ID	None
> Web specific	Authentication
> SmartDevices specific	Authorization
> External Usage	
> Compatibility	

Selecionando o nível de segurança



Acesso ao backend do GAM (objeto GAM Home)

The screenshot shows the Genexus Access Manager application interface. At the top, there is a navigation bar with the following elements:

- GENEXUS ACCESS MANAGER (left side)
- USER (highlighted in red)
- ROLES
- SETTINGS (with a dropdown arrow)
- A circular profile icon with the letter 'A' and the text 'Administrator' (right side)

Below the navigation bar, there is a search bar with the placeholder "Try a Search". To the left of the search bar is a "SHOW FILTERS" button with a checkmark icon. To the right is a "+ ADD" button.

The main content area displays a table with the following columns:

User Name	First Name	Last Name	Authentication	Action
admin	Administrator	User	local	EDIT

At the bottom of the table, there are navigation links: FIRST / PREV / NEXT.

Testing

Objetos para gerar testes unitários e testes de interface

New Object

Select a Category:

- Common
- Workflow
- Reporting
- Documentation
- Web
- Extensibility
- Deploy
- Chatbots
- Smart Devices

Select a Type:

- Color Palette
- Data Provider
- Data Selector
- Data View
- Domain
- External Object
- Image
- Language
- Procedure
- Stencil
- Structured Data Type
- Subtype Group
- Transaction
- UI Test
- Unit Test

- Permite simular
ações do usuário no
navegador

Prova isolada de:
- Procedures
- Data Providers
- Business Components

Teste Unitário

São criados os objetos:

- <ObjectName>UnitTest
- <ObjectName>UnitTestSDT
- <ObjectName>UnitTestDataTable

The screenshot shows the Genexus IDE interface with four tabs at the top: Start Page, AttractionsByName, AttractionList, and AttractionsPerCategory. The AttractionsByName tab is active and displays a script in Source mode. The script content is as follows:

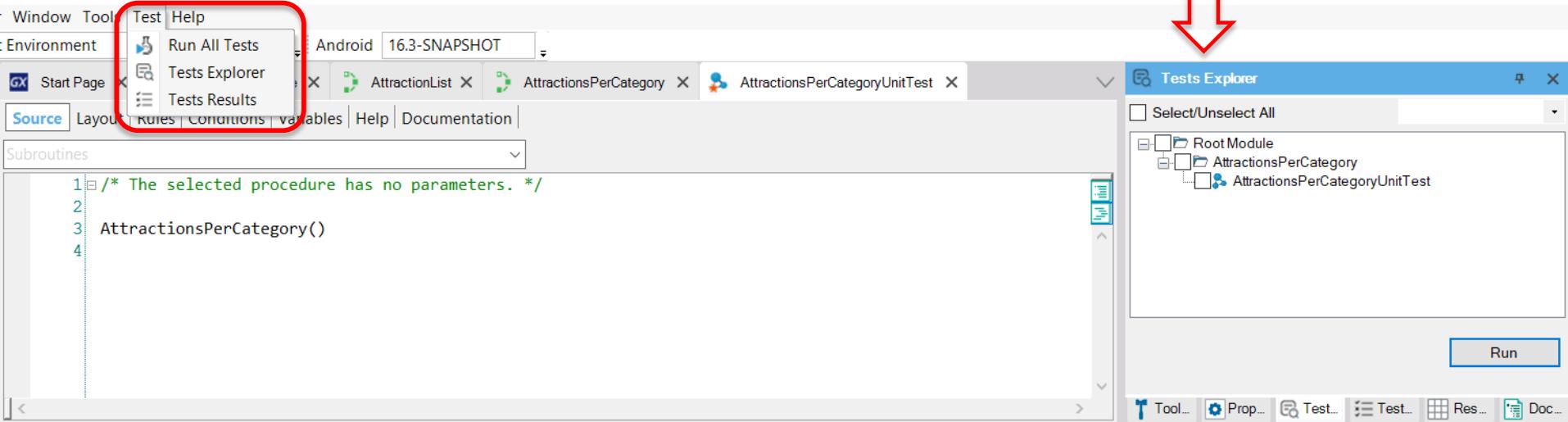
```
1 Print Title
2
3 For each Attraction order CategoryId
4   Print Categories
5   Print ColumnTitles
6   For each Attraction
7     Print Attractions
8   Endfor
9 Endfor
```

The code editor has a vertical scroll bar on the left. To the right of the editor is a context menu with the following options:

- Close
- New Horizontal Tab Group
- New Vertical Tab Group
- Close All But This
- Full Screen
- Open
- Open Part
- Delete
- Save As...
- References
- History
- Team Development
- Properties
- Locate in KB Explorer
- Build
- Rebuild
- Run
- Run Without Building
- Run With This Only
- Build With This Only
- Set As Startup Object
- View Last Navigation
- View Navigation
- Export
- Select Left Side To Compare
- Create Unit Test

The "Create Unit Test" option is highlighted with a yellow background.

Test Explorer



Teste de Interface: GXTest

chrome web store 

Extensions 1 of 1 extensions

« Home

Extensions

Themes

Features

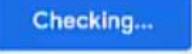
Runs Offline

By Google

Free

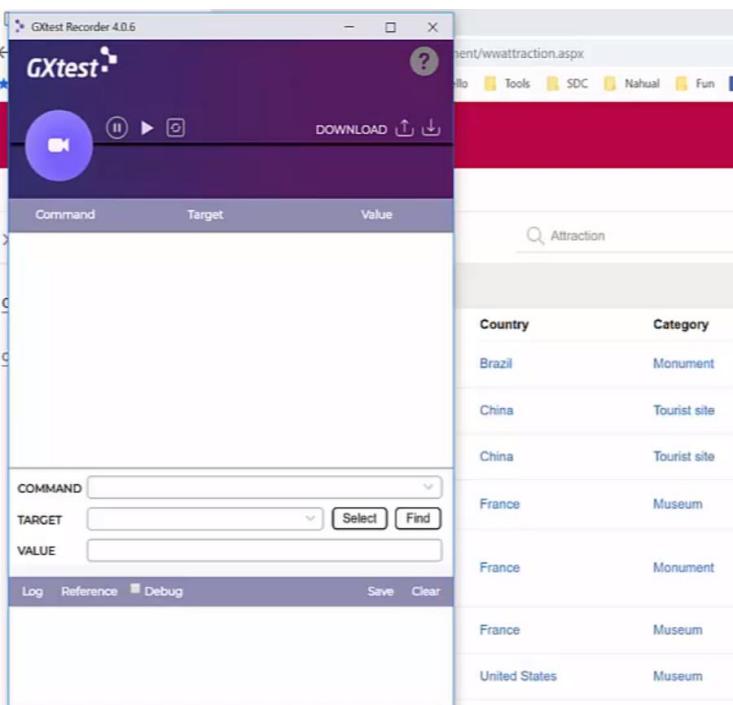
Gxtest recorder

Offered by: www.abstracta.us 

Allows developers and testers to record test cases actions over GeneXus a

 1 Developer Tools

Teste de Interface



Command	Target	Value
ClickBy	id=BTNINSERT	Estadio Centenario
ClickBy	id=COUNTRYID	
SelectBy	id=COUNTRYID	label=Uruguay
ClickBy	id=COUNTRYID	
ClickBy	id=ATTRACTIONNAME_Balloon	
AssertTextBy	id=ATTRACTIONNAME_Balloon	Attraction,Country already exists
ClickBy	id=CITYID	
ClickBy	id=ATTRACTIONNAME	
TypeBy	id=ATTRACTIONNAME	Estadio Centenario 2

COMMAND: ClickBy
TARGET: id=COUNTRYID
VALUE: label=Uruguay

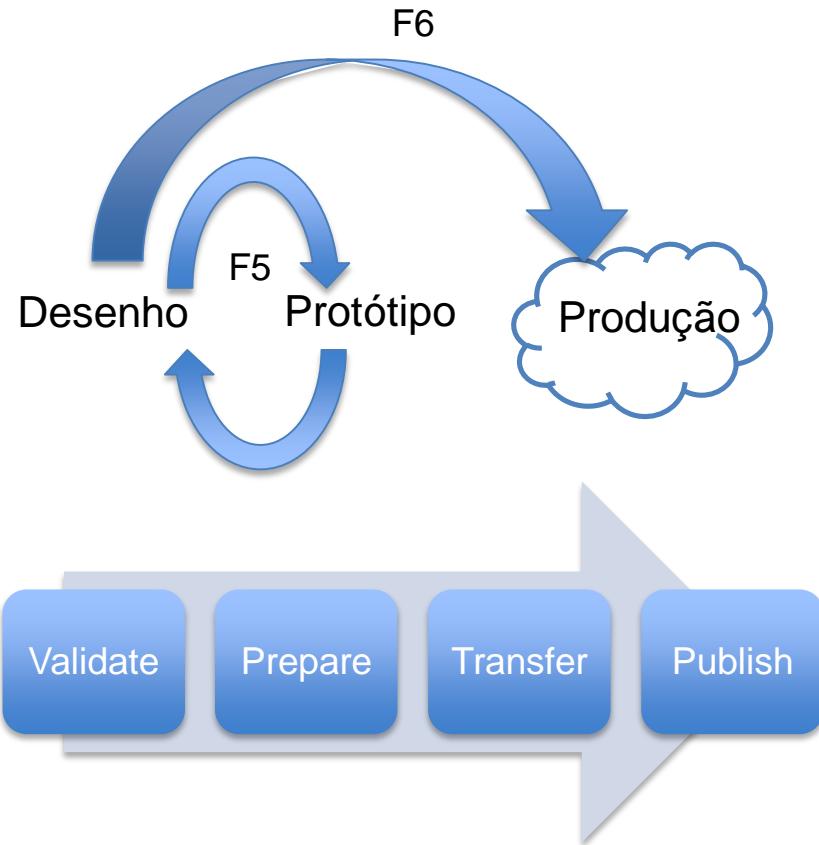
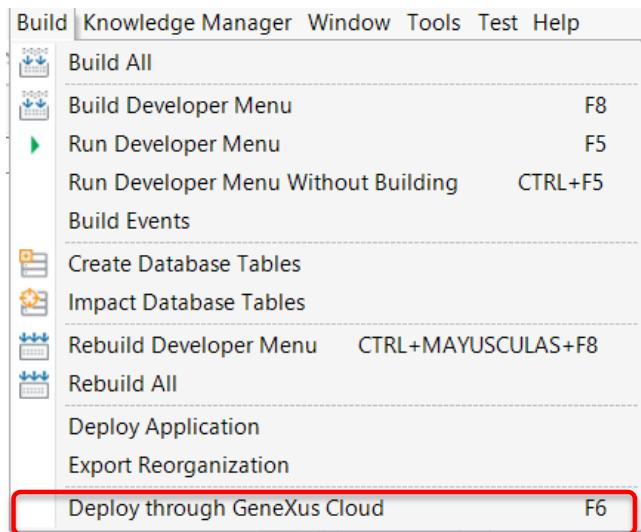
```
[info] Executing: [ClickBy | id=BTNINSERT | ]
[info] Executing: [ClickBy | id=COUNTRYID | ]
[info] Executing: [SelectBy | id=COUNTRYID | label=Uruguay]
[info] Executing: [ClickBy | id=COUNTRYID | ]
[info] Executing: [ClickBy | id=ATTRACTIONNAME_Balloon | ]
[info] Executing: [AssertTextBy | id=ATTRACTIONNAME_Balloon | Attraction,Country already exists]
[info] Executing: [ClickBy | id=CITYID | ]
[info] Executing: [ClickBy | id=ATTRACTIONNAME | ]
[info] Executing: [TypeBy | id=ATTRACTIONNAME | Estadio Centenario 2]
```



(Requer licença GXTest)

Deployment

Deploy automático com F6



Build | Knowledge Manager Window Tools Test Help

Build All
Build MenuTravelAgency F8
Run MenuTravelAgency F5
Run MenuTravelAgency Without Building CTRL+F5
Build Events
Create Database Tables
Impact Database Tables
Rebuild MenuTravelAgency CTRL+MAYUSCULAS+F8
Rebuild All
Deploy Application

Export Reorganization
Deploy through GeneXus Cloud

Deploy Application

Select objects to deploy

New Deployment Unit ... Add...

Web Panel
 Home

Objetos main

Target: Local Options

Application Server: Microsoft IIS 7

Deploy

Integração

Módulo de Inteligência Artificial

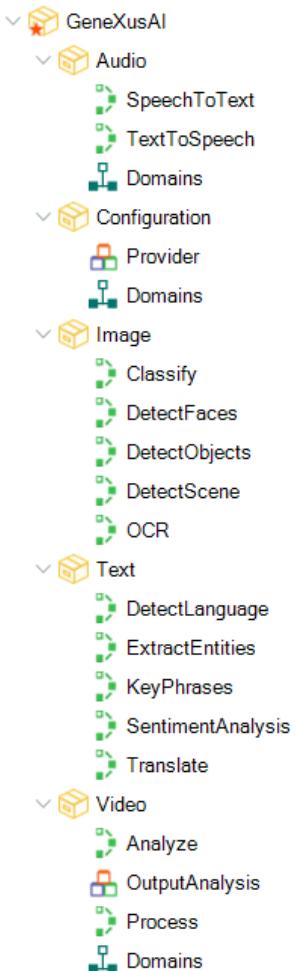
Manage Module References X

Modules: Search by pattern here...

Chatbot (2.1.10.129299)	GeneXus Chatbot module is a basic set of interfaces and implementations of data structures and algorithms needed to implement a Chatbot solution.
GeneXusAI (1.1.21.129329)	GeneXusAI contains a common set of Artificial Intelligence tasks, including audio, text and image processing, all of them provided by several Cloud Platforms (e.g. IBM Watson, Microsoft Azure Cognitive Services, SAP Leonardo)
GeneXus (2.1.7.129290)	GeneXus Core Module is a basic set of interfaces and implementations of data structures and algorithms needed to implement a Chatbot solution.
GXtest (0.4.2)	GXtest Module provides core functionality for creating, running and reporting tests on genexus and over ci/cd pipelines. https://wiki.genexus.com/commwiki/servlet/wiki?name=GXtest

MyKB

- > Main Programs
- > Root Module
- > References
 - > GeneXus
 - > **GeneXusAI**
- > Customization
- > Documentation



IBM Watson

Microsoft Cognitive Services

SAP Leonardo

Google Cloud Services

Amazon Web Services

Chatbots: objeto Conversational Flow

New Object

Select a Category: Select a Type:

- Common
- Workflow
- Reporting
- Documentation
- Web
- Deploy
- Chatbots**
- Smart Devices

Conversational Flows

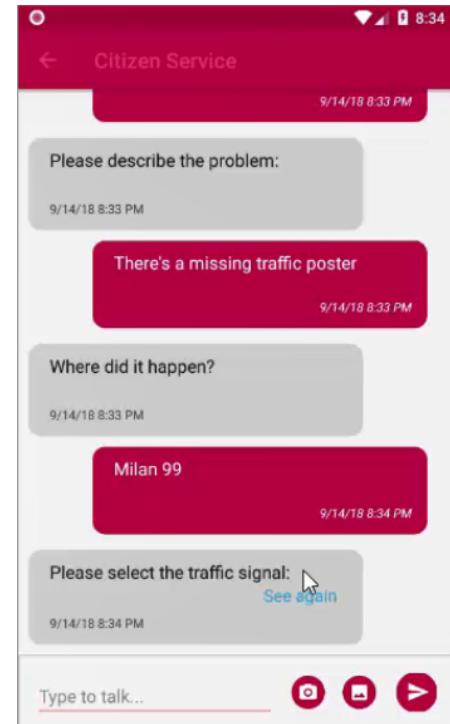
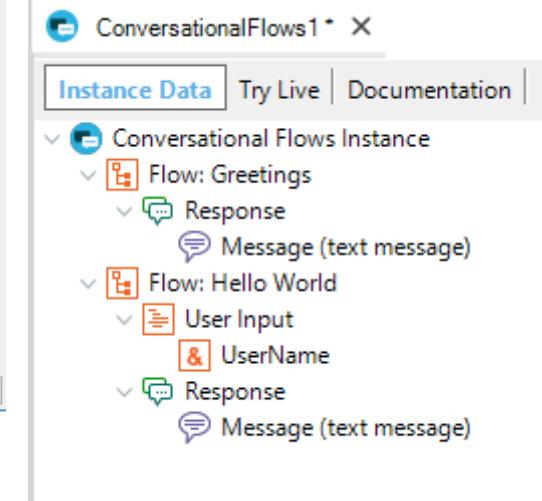
Create a pattern instance of Conversational

Name: ConversationalFlows1

Description: Conversational Flows1

Module/Folder: Root Module

Create **Cancel**



Smart Devices

Aplicar Pattern em Transação

Country X

Structure | Web Form | Win Form | Rules | Events | Variables | Help | Documentation | Patterns

Patterns usable in this object (underlined means pattern is applied)

Category | Work With for Web | Work With for Smart Devices

Apply this pattern on save

Level (Country)
List
Detail
Section (General)
Section (Place)
Section (Sale)

Select List, Detail or Section on the tree to edit it

Aplicar Pattern em Transação

The screenshot shows the Genexus IDE interface with the following components:

- KB Explorer:** On the left, it shows the project structure with nodes like "Main Programs", "Root Module", "Gx0021", "Country", "Country_DataProvider", and "WorkWithDevicesCountry". A red arrow points to the "WorkWithDevicesCountry" node.
- Country Object View:** The main workspace shows the "Country" object with tabs for "Structure", "Web Form", "Win Form", "Rules", "Events", "Variables", "Help", "Documentation", and "Patterns". The "Patterns" tab is selected. It displays patterns applicable to the object, such as "Category", "Work With for Web", and "Work With for Smart Devices".
- Properties View:** On the right, the "Properties" view is open, showing various configuration options for the "WorkWithDevicesCountry" object. One of the properties, "Main program", is highlighted with a red border.
- Contextual Menu:** A context menu is open over the "WorkWithDevicesCountry" node in the KB Explorer. The menu includes options like "New", "Open", "Copy", "Delete", "Save As...", "References", "History", "Properties" (which is highlighted in yellow), "Build", "Rebuild", and "Run".

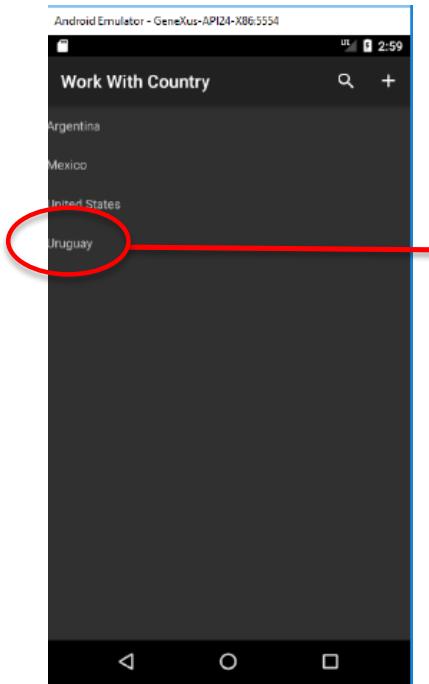
A tooltip message "Select List, Detail or Section on the tree to edit it" is visible near the bottom center of the screen.

Text Description:

Tem que configurar sua propriedade **Main program = True** e, sobre o objeto, clicar com o **botão direito e Run**

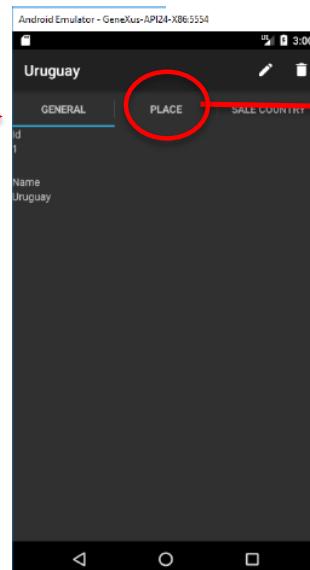
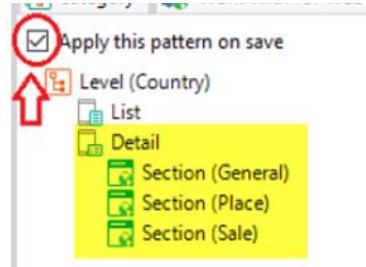
Aplicar Pattern em Transação

botão direito e Run:

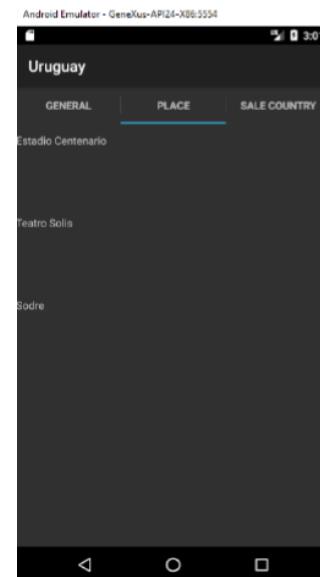


GENERAL

Escolho 1 country

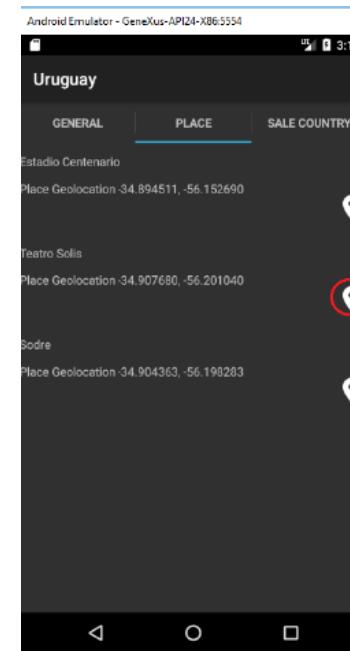
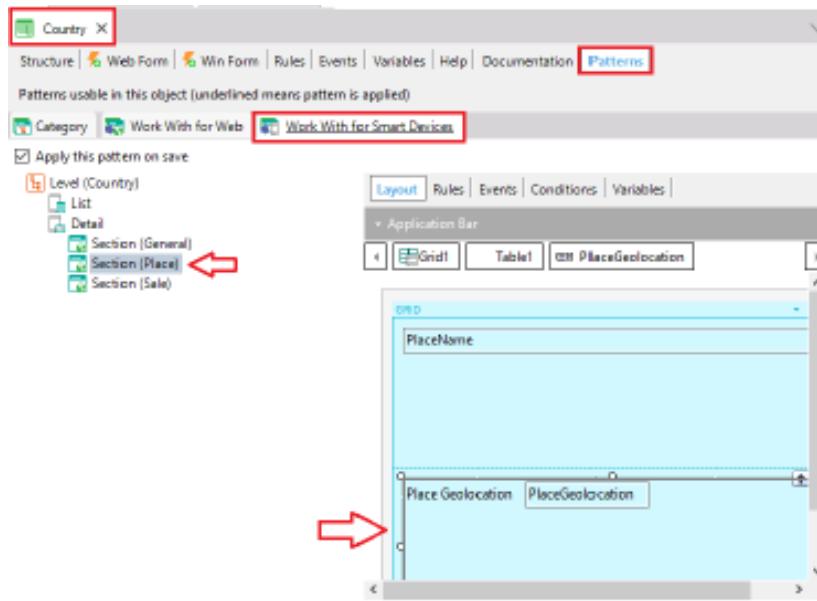


PLACE



Aplicar Pattern em Transação

Queremos, para cada lugar, além de ver seu nome, ver sua geolocalização.



Menu de acesso: Menu for Smart Devices

Criação:

New Object

Select a Category: Select a Type:

- Common
- Workflow
- Reporting
- Documentation
- Web
- Smart Devices**

Menu for Smart Devices

Panel for Smart Devices

Theme for Smart Devices

Work With for Smart Devices

Show application options using different layouts.

Name: MenuSmartDevices1

Description: Menu Smart Devices1

Properties

Menu for Smart Devices: MenuSmartDevices1

Name	MenuSmartDevices1
Description	Menu Smart Devices1
Module/Folder	Root Module
Qualified Name	MenuSmartDevices1
Object Visibility	Public
Auto Update	False
Main program	True

Add action...

MenuSmartDevices1*

Menu Events Variables

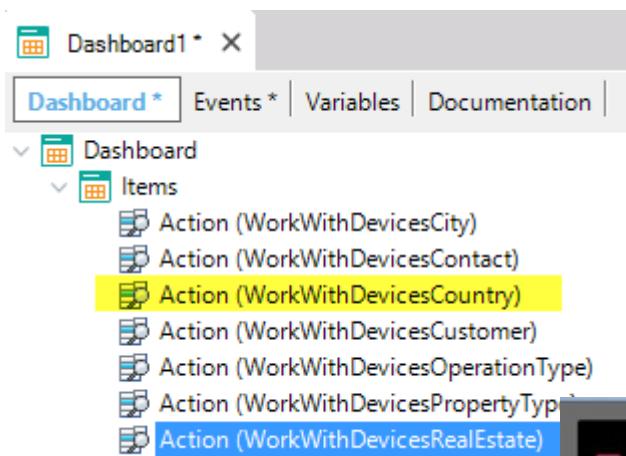
Menu Items

Add

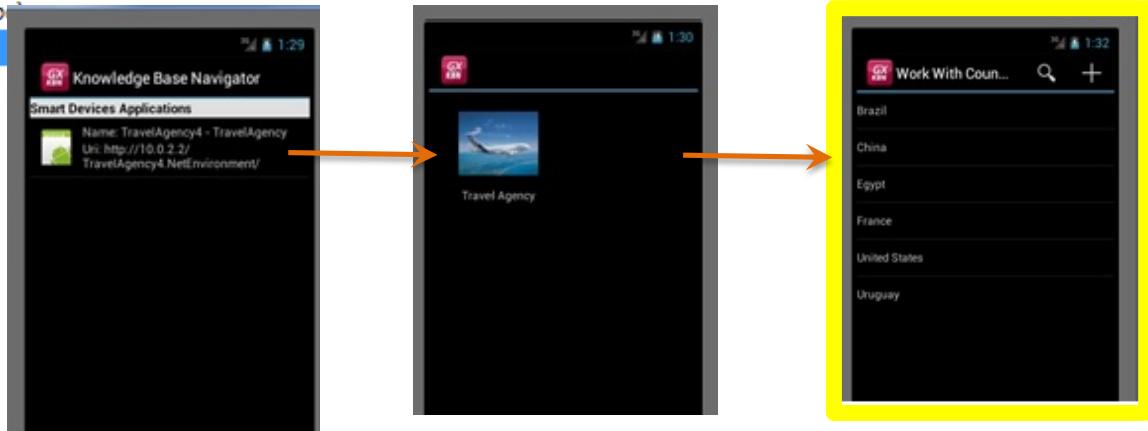
- Delete SUPR
- Cut CTRL+X
- Copy CTRL+C
- Paste CTRL+V
- Expand All
- Collapse All
- Properties F4

Action

Link



Ao pressionar F5...



GeneXus™

Videos

training.genexus.com

Documentation

wiki.genexus.com

Certifications

training.genexus.com/certifications