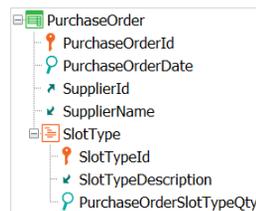
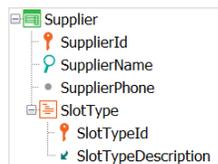
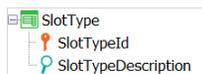
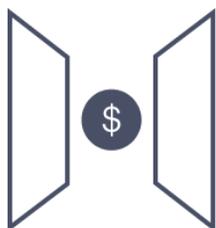


Subordinação lógica

GeneXus™

Neste vídeo, abordaremos o tema chamado subordinação lógica em GeneXus.

Reality: Casino



The type of slot that is being entered in the lines must be provided by the supplier of the order.

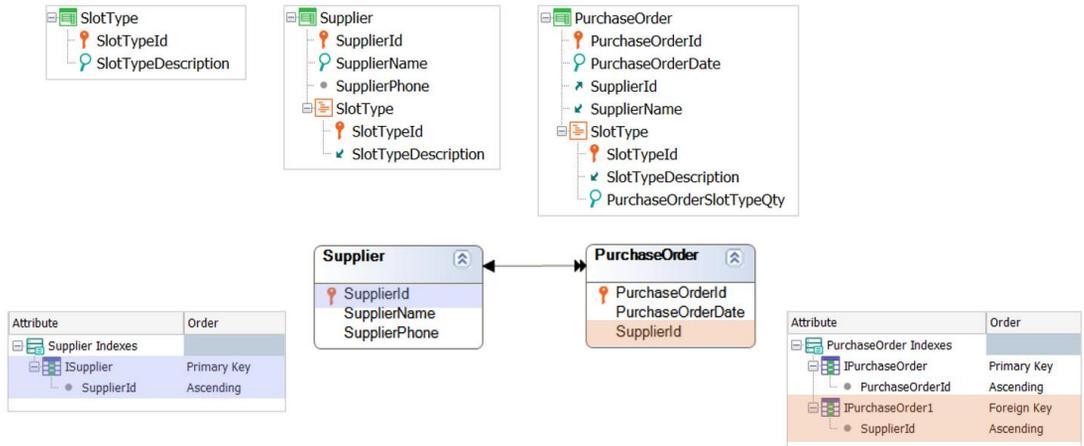
Para isso, vamos nos concentrar em uma KB na qual estamos desenvolvendo uma aplicação para um cassino, que, entre outras coisas, oferece a possibilidade de registrar os pedidos de compra de jogos de caça-níqueis aos seus fornecedores.

Usaremos a transação SlotType para registrar os tipos de jogos de caça-níqueis, a transação Supplier para registrar os fornecedores e os caça-níqueis que eles disponibilizam e, por último, -a transação em que vamos focar para estudar este caso-, PurchaseOrder, para registrar os pedidos de compra de certos tipos de caça-níqueis que o cassino faz para um de seus fornecedores.

Ao inserir um pedido de compra, deve-se controlar na transação PurchaseOrder que o tipo de caça-níqueis que está sendo inserido nas linhas seja fornecido pelo fornecedor do pedido (atributo que se encontra no primeiro nível). Isto está garantido pelo desenho?

A resposta para a pergunta é não. Com este desenho não será realizada essa verificação. Vejamos por quê.

Subordination relationship



Vamos começar com um caso mais simples, para ver bem a diferença. Entre o primeiro nível de PurchaseOrder e o primeiro nível de Supplier existe uma relação de subordinação. Em outras palavras: SupplierId será chave estrangeira na tabela PurchaseOrder, apontando para a tabela Supplier, que tem SupplierId como chave primária. Por este motivo, será criado automaticamente um índice em Supplier por chave primária e um índice em PurchaseOrder por chave estrangeira.

Uniqueness and referential integrity controls

- The index by primary key in Supplier will be used to control that:

Attribute	Order
Supplier Indexes	
ISupplier	Primary Key
SupplierId	Ascending

- When inserting a supplier, there isn't another one with the same SupplierId value
- When an order for a supplier is inserted, it exists.

Order Id	<input type="text" value="1"/>
Order Date	<input type="text" value="09/15/22"/>
Supplier Id	<input type="text" value="5"/> No matching 'Supplier'.
Supplier Name	<input type="text"/>

Esses índices serão utilizados para realizar com eficiência os controles de unicidade e de integridade referencial. Quer dizer:

- O índice por chave primária em Supplier será utilizado para controlar que:
 - quando for inserido um fornecedor não exista outro com o mesmo valor de SupplierId, e que
 - quando for inserido um pedido para um fornecedor, ele exista

Uniqueness and referential integrity controls

- The index by primary key in Supplier will be used to control that:

Attribute	Order
Supplier Indexes	
ISupplier	Primary Key
SupplierId	Ascending

- When inserting a supplier, there isn't another one with the same SupplierId value
- When an order for a supplier is inserted, it exists.

Order Id	<input type="text" value="1"/>
Order Date	<input type="text" value="09/15/22"/>
Supplier Id	<input type="text" value="5"/> No matching 'Supplier'.
Supplier Name	<input type="text"/>

- The index by foreign key in PurchaseOrder will be used to control that:

Attribute	Order
PurchaseOrder Indexes	
IPurchaseOrder	Primary Key
PurchaseOrderId	Ascending
IPurchaseOrder1	Foreign Key
SupplierId	Ascending

- When deleting a supplier, there is no order referencing it.

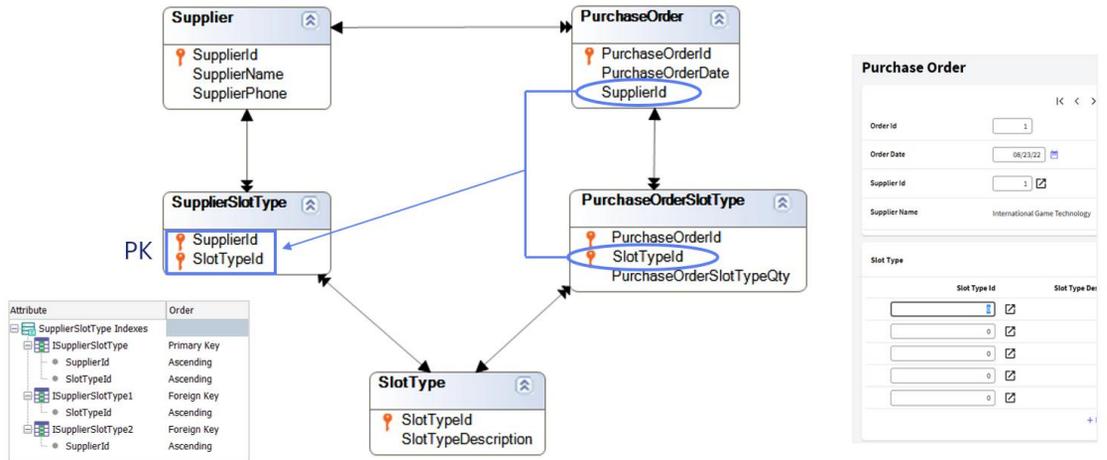
! Invalid delete, related information in Purchase Order

- O índice por chave estrangeira em PurchaseOrder será utilizado para controlar que:
 - quando for excluído um fornecedor, não exista nenhum pedido que o esteja referenciando

Como existe uma subordinação (física) entre Supplier e PurchaseOrder, estas verificações são realizadas.

Agora, o que acontece quando inserimos na transação PurchaseOrder um tipo de caça-níqueis?

Table relationships



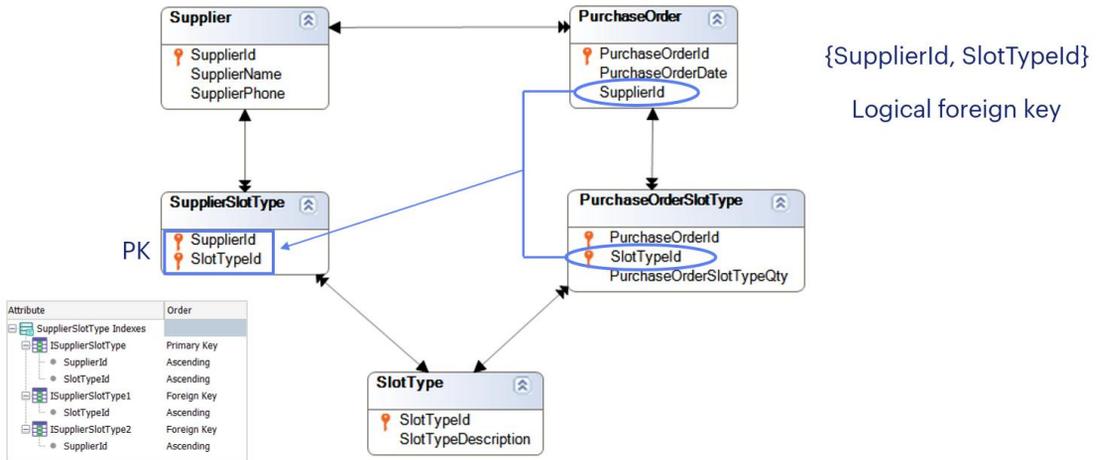
Observemos a relação entre as tabelas:

Nós queremos que quando for inserido um registro na tabela PurchaseOrderSlotType, seja controlado que exista um registro na tabela SupplierSlotType com o valor do SupplierId correspondente ao PurchaseOrderId, e com o valor de SlotTypeId correspondente ao SlotTypeId do registro que está sendo inserido, já que essa informação se tem em memória.

Ou seja, quando deseja-se inserir um novo SlotTypeId no grid da transação PurchaseOrder, tem-se o valor de SlotTypeId que o usuário acabou de digitar ou escolher (usando a lista de seleção) e também se tem o valor de SupplierId (especificado pelo usuário no primeiro nível da transação). Como estes dois atributos juntos constituem a chave primária da tabela SupplierSlotType, então existe um índice nessa tabela por chave primária que permitiria fazer a verificação de integridade de forma eficiente. Mas, a fará?

A resposta é não.

Table relationships



Observemos que a diferença com o caso simples que vimos no início é que na realidade {SupplierId, SlotTypeId} não é uma chave estrangeira em PurchaseOrder, pois os dois atributos não estão na mesma tabela. No entanto, poderíamos dizer que formam uma chave estrangeira lógica, embora tal coisa não exista para as bases de dados relacionais.

O problema é que GeneXus realiza os controles de integridade referencial se entende que existe uma relação de subordinação física entre as tabelas.

Portanto, não os realizará no nosso caso, permitindo inserir um tipo de caça-níqueis que não é fornecido pelo fornecedor do pedido.

Running example

SUPPLIER TABLE

SupplierId	SupplierName	SupplierPhone
1	International Game Technology	11222233
2	Pragmatic Play	55537777

SLOTTYPE TABLE

SlotTypeId	SlotTypeDescription
1	Classic slot
2	Five-reel slot
3	Virtual reality slot
4	3D slot

SUPPLIERSLOTTYPE TABLE

SupplierId	SlotTypeId
1	3
1	4
2	1

Purchase Order

|< < > >| SELECT

Order Id:

Order Date: 📅

Supplier Id: ↩️

Supplier Name: International Game Technology ↩️

Slot Type

	Slot Type Id	Slot Type Description	Type Qty
-	<input type="text" value="1"/> ↩️	Classic slot	<input type="text" value="5"/>
-	<input type="text" value="2"/> ↩️	Five-reel slot	<input type="text" value="3"/>
	<input type="text" value="0"/> ↩️		<input type="text" value="0"/>
	<input type="text" value="0"/> ↩️		<input type="text" value="0"/>
	<input type="text" value="0"/> ↩️		<input type="text" value="0"/>

+ NEW ROW

CONFIRM CANCEL DELETE

Vejamos isto no seguinte exemplo: o fornecedor com identificador 1, chamado *International Game Technology*, fornece apenas os caça-níqueis com identificadores 3 e 4 (*Virtual reality slot* e *3D slot* respectivamente). No entanto, no pedido de compra indicamos que compraremos desse fornecedor os caça-níqueis de tipo 1 e 2 (justamente aqueles que não fornece).

Como podemos fazer então para resolver este problema?

Possible solution

In the rules, invoke a procedure that performs the control:

A screenshot of a code editor window titled 'PurchaseOrder* X'. The 'Rules' tab is active, showing two lines of code:

```
1 &ok = SupplierProvidesSlot(SupplierId, SlotTypeId);  
2 Error('The slot type is not provided by the supplier') if not &ok;
```

A primeira coisa que pode nos ocorrer é invocar um procedimento nas regras da transação PurchaseOrder que realize a verificação:

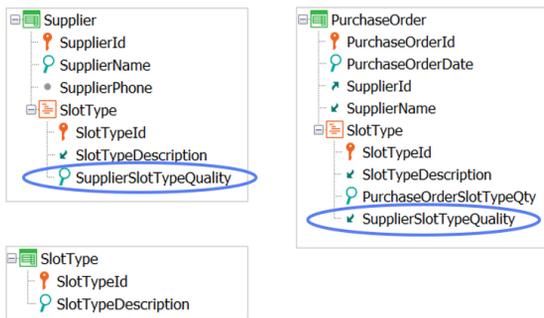
```
&ok = SupplierProvidesSlot(SupplierId, SlotTypeId);
```

E no caso de retornar falso mostrar o seguinte erro:

```
Error('The slot type is not provided by the supplier') if not &ok;
```

Another possible solution

Infer in PurchaseOrder a secondary attribute of Supplier's second level:



Purchase Order

|< < > >| SELECT

Order Id

Order Date

Supplier Id

Supplier Name International Game Technology

Slot Type

Slot Type Id	Slot Type Description	Type Qty
<input type="text" value="1"/>	No matching 'Slot Type'.	<input type="text" value="0"/>
<input type="text" value="0"/>	<input type="checkbox"/>	<input type="text" value="0"/>
<input type="text" value="0"/>	<input type="checkbox"/>	<input type="text" value="0"/>
<input type="text" value="0"/>	<input type="checkbox"/>	<input type="text" value="0"/>
<input type="text" value="0"/>	<input type="checkbox"/>	<input type="text" value="0"/>

Outra maneira pela qual teríamos obtido o mesmo efeito, sem necessidade de escrever nada, teria sido se na transação Supplier houvesse um atributo secundário no segundo nível, de maneira a inferi-lo em PurchaseOrder, e assim forçar GeneXus a perceber a relação.

Nos dará um erro de “No matching” se o tipo de caça-níqueis que está sendo inserido nas linhas não for fornecido pelo fornecedor do pedido.

Another possible solution

Infer in PurchaseOrder a secondary attribute of Supplier's second level:

The diagram illustrates the relationship between the **Supplier** and **PurchaseOrder** entities. The **Supplier** entity has attributes: **SupplierId**, **SupplierName**, **SupplierPhone**, **SlotType**, **SlotTypeId**, **SlotTypeDescription**, and **SupplierSlotTypeQuality**. The **PurchaseOrder** entity has attributes: **PurchaseOrderId**, **PurchaseOrderDate**, **SupplierId**, **SupplierName**, **SlotType**, **SlotTypeId**, **SlotTypeDescription**, **PurchaseOrderSlotTypeQty**, and **SupplierSlotTypeQuality**. A **SlotType** entity is also shown with attributes **SlotTypeId** and **SlotTypeDescription**. A dashed line indicates a relationship between the **SupplierSlotTypeQuality** attribute in the **Supplier** entity and the **SupplierSlotTypeQuality** attribute in the **PurchaseOrder** entity.

The screenshot shows the **Purchase Order** form. The **Order Id** is 2, **Order Date** is 08/30/22, **Supplier Id** is 1, and **Supplier Name** is International Game Technology. The **Slot Type** table shows a validation error for the first row: "The slot type is not provided by the supplier".

Slot Type Id	Slot Type Description	Type Qty
1	The slot type is not provided by the supplier	
0		0
0		0
0		0
0		0

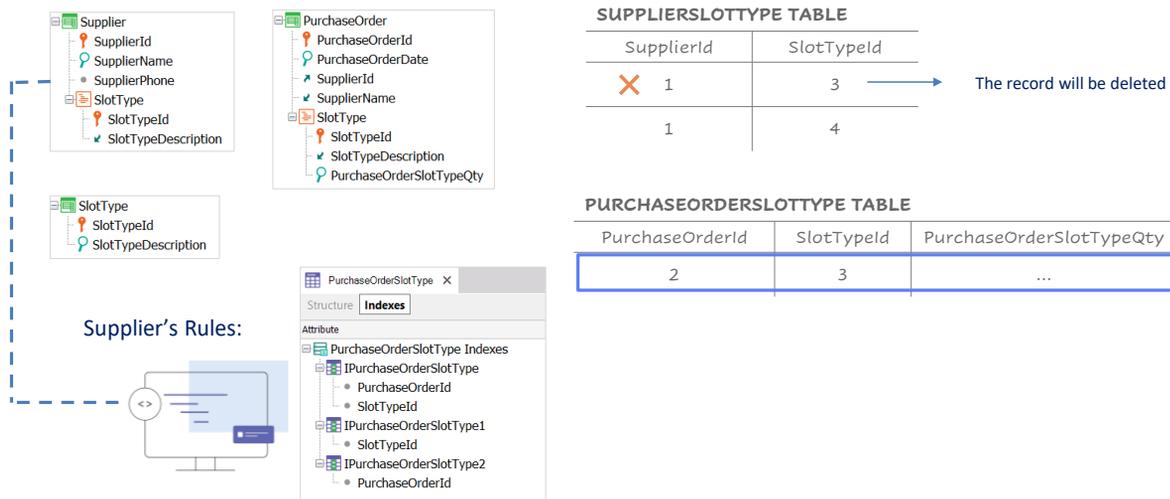
```

1= RefMsg('The slot type is not provided by the supplier',
2= SupplierId, SlotTypeId);
  
```

Poderíamos utilizar a regra `RefMsg`, que serve para personalizar a mensagem do erro quando é realizada uma verificação de integridade referencial em insert ou update a partir de uma chave estrangeira, e ela falha. Esta regra tem como parâmetros - além da mensagem que queremos que seja exibida em caso de falha do controle de IR- ou os atributos que compõem a chave estrangeira.

Something to think about:

What will happen in this solution if you delete a slot type from a supplier for which orders already exist?



Uma pergunta interessante é o que acontecerá com esta solução se for desejado remover um tipo de caça-níqueis de um fornecedor, para o qual já existam pedidos.

Ou seja, se o usuário, através da transação "Supplier", tenta eliminar uma linha para a qual exista algum "PurchaseOrder" com aquele fornecedor e tipo de caça-níqueis.

Neste caso, GeneXus não poderá verificar a integridade, pois não existe um índice por {SupplierId, SlotTypeId} na tabela PurchaseOrderSlotType. Como essa chave estrangeira só existe em um nível lógico (ou seja, na tabela estendida), não existirá índice algum –pois um índice corresponde a um conceito físico. Portanto, esta verificação deveremos programar nas regras da transação "Supplier", invocando um procedimento que a realize e então disparando um erro de acordo com o resultado retornado.

GeneXus™

training.genexus.com

wiki.genexus.com

training.genexus.com/certifications