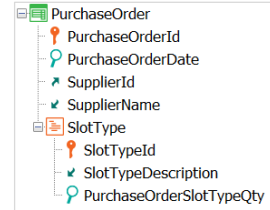
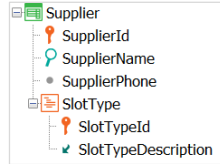
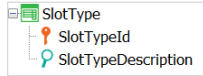
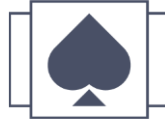


Subordinación lógica

GeneXus™

En este video, abordaremos el tema denominado subordinación lógica en GeneXus.

Reality: Casino



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

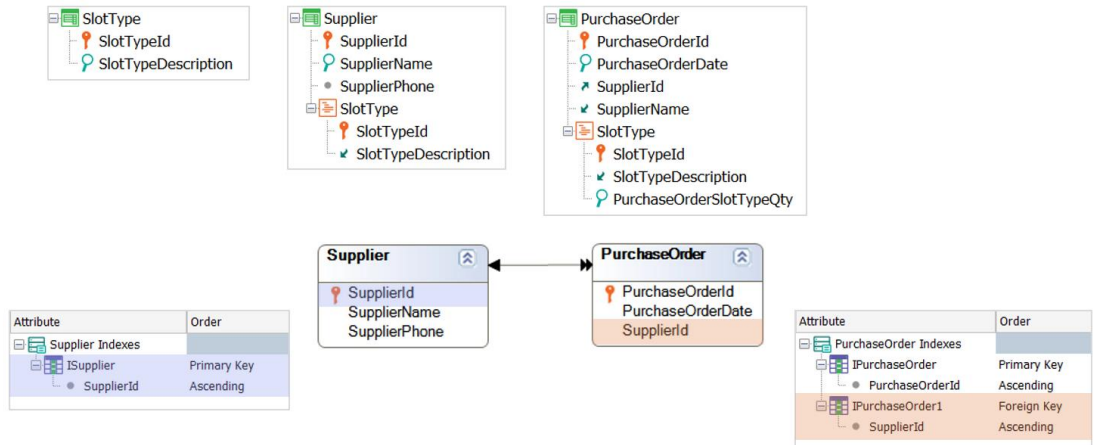
Para ello, nos centraremos en una KB en la que estamos desarrollando una aplicación para un casino, que, entre otras cosas, ofrece la posibilidad de registrar los pedidos de compra de slots de juegos a sus proveedores.

Usaremos la transacción SlotType para registrar los tipos de slots de juegos, la transacción Supplier para registrar los proveedores y los slots que estos disponibilizan, y, por último, -la transacción en la que vamos a enfocarnos para estudiar este caso-, PurchaseOrder, para registrar los pedidos de compra de ciertos tipos de slots que el casino realiza a alguno de sus proveedores.

A la hora de ingresar una orden de compra, debe controlarse en la transacción PurchaseOrder que el tipo de slot que se esté ingresando en las líneas sea provisto por el proveedor de la orden (atributo que se encuentra en el primer nivel). ¿Esto está asegurado por el diseño?

La respuesta a la pregunta es no. Con este diseño no se realizará ese chequeo. Veamos por qué.

Subordination relationship



Empecemos por un caso más simple, para ver bien la diferencia. Entre el primer nivel de PurchaseOrder y el primer nivel de Supplier existe una relación de subordinación. Dicho de otro modo: SupplierId será clave foránea en la tabla PurchaseOrder, apuntando a la tabla Supplier, que tiene a SupplierId como clave primaria. Por este motivo se creará automáticamente un índice en Supplier por clave primaria y un índice en PurchaseOrder por clave foránea.

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"/> <input checked="" type="checkbox"/> No matching 'Supplier'.
Supplier Name	<input type="text"/>

Esos índices serán utilizados para hacer eficientemente los controles de unicidad y de integridad referencial. Es decir:

- El índice por clave primaria en Supplier se utilizará para controlar que:
 - cuando se inserte un proveedor no exista otro con el mismo valor de SupplierId, y que
 - cuando se inserte una orden para un proveedor, este 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.

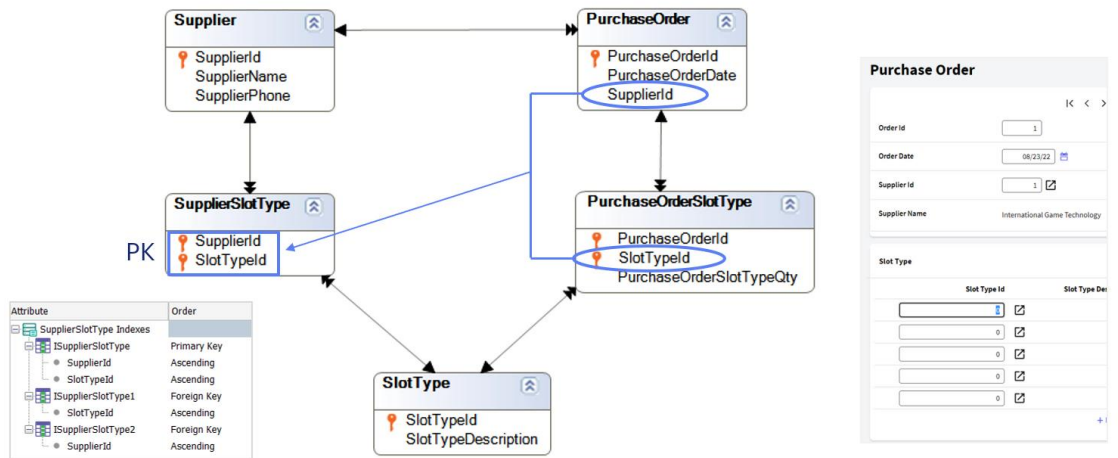
❗ Invalid delete, related information in Purchase Order

- El índice por clave foránea en PurchaseOrder se utilizará para controlar que:
 - cuando se elimine un supplier no exista ninguna orden que lo esté referenciando

Como hay una subordinación (física) entre Supplier y PurchaseOrder, estos chequeos se realizan.

Ahora, ¿qué pasa cuando ingresamos en la transacción PurchaseOrder un tipo de slot?

Table relationships



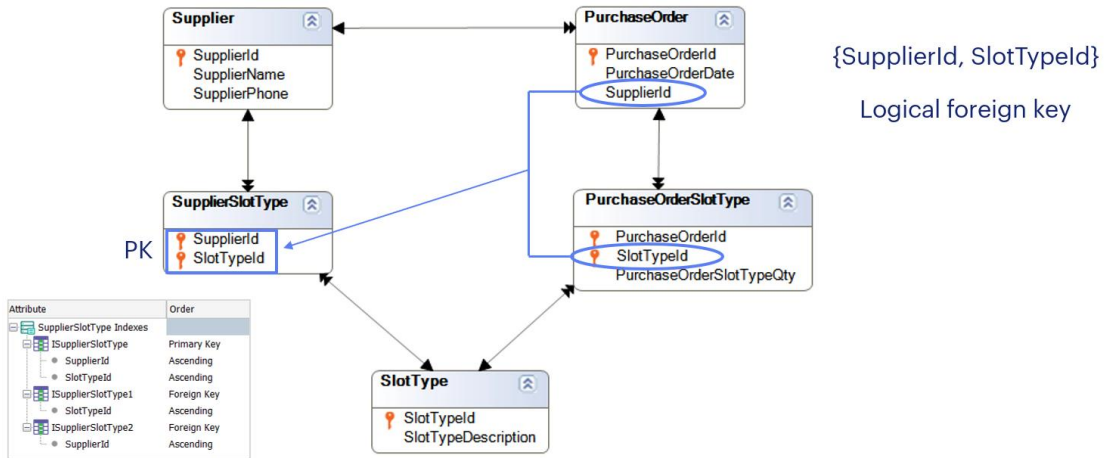
Observemos la relación entre las tablas:

Nosotros queremos que cuando se vaya a ingresar un registro en la tabla PurchaseOrderSlotType se controle que exista un registro en la tabla SupplierSlotType con el valor del SupplierId correspondiente al PurchaseOrderId, y con el valor de SlotTypeId correspondiente al SlotTypeId del registro que se está insertando, puesto que esa información se tiene en memoria.

Es decir, cuando se está queriendo insertar un nuevo SlotTypeId en el grid de la transacción PurchaseOrder, se tiene el valor de SlotTypeId que el usuario acaba de digitar o elegir (usando la lista de selección) y se tiene también el valor de SupplierId (especificado por el usuario en el primer nivel de la transacción). Como estos dos atributos juntos constituyen la clave primaria de la tabla SupplierSlotType, entonces existe un índice en esa tabla por clave primaria que permitiría hacer el chequeo de integridad eficientemente. Pero, ¿lo hará?

La respuesta es no.

Table relationships



Observemos que la diferencia con el caso simple que vimos al principio es que en realidad {SupplierId, SlotTypeId} no es una clave foránea en PurchaseOrder, puesto que los dos atributos no se encuentran en la misma tabla. Sin embargo, podríamos decir que conforman una clave foránea lógica, aunque tal cosa no exista para las bases de datos relacionales.

El problema es que GeneXus realiza los controles de integridad referencial si entiende que hay una relación de subordinación física entre las tablas.

Por lo tanto, no los realizará en nuestro caso, permitiéndonos ingresar un tipo de slot que no es provisto por el proveedor de la orden.

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

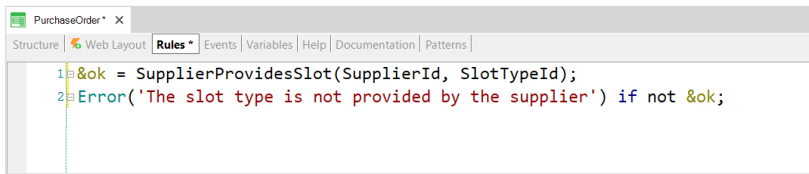
CONFIRM CANCEL DELETE

Veamos esto en el siguiente ejemplo: el proveedor con identificador 1, de nombre *International Game Technology*, provee únicamente los slots con identificadores 3 y 4 (*Virtual reality slot* y *3D slot* respectivamente). Sin embargo, en la orden de compra indicamos que compraremos a ese proveedor los slots de tipo 1 y 2 (justamente los que no provee).

¿Cómo podemos hacer entonces para solucionar este problema?

Possible solution

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



```
PurchaseOrder* X
Structure | Web Layout | Rules | Events | Variables | Help | Documentation | Patterns
1: &ok = SupplierProvidesSlot(SupplierId, SlotTypeId);
2: Error('The slot type is not provided by the supplier') if not &ok;
```

Lo primero que se nos puede ocurrir es invocar un procedimiento en las reglas de la transacción PurchaseOrder que realice el chequeo:

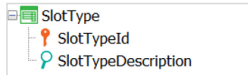
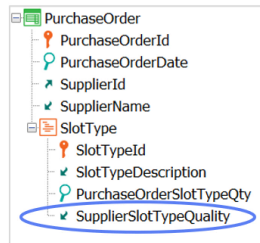
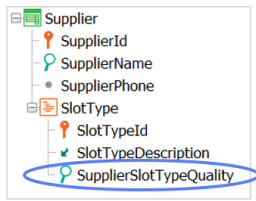
```
&ok = SupplierProvidesSlot(SupplierId, SlotTypeId);
```

Y en caso de devolver falso mostrar el siguiente error:

```
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="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"/>		<input type="text" value="0"/>

Otra manera en que hubiéramos logrado el mismo efecto, sin necesidad de escribir nada, hubiera sido si en la transacción Supplier hubiera habido un atributo secundario en el segundo nivel, de manera tal de inferirlo en PurchaseOrder, y así forzar a GeneXus a darse cuenta de la relación.

Nos dará un error de “No matching” si el tipo de slot que se esté ingresando en las líneas no es provisto por el proveedor de la orden.

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 has attributes: `SlotTypeId` and `SlotTypeDescription`. A dashed line indicates a relationship between the `SupplierSlotTypeQuality` attribute in both entities.

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

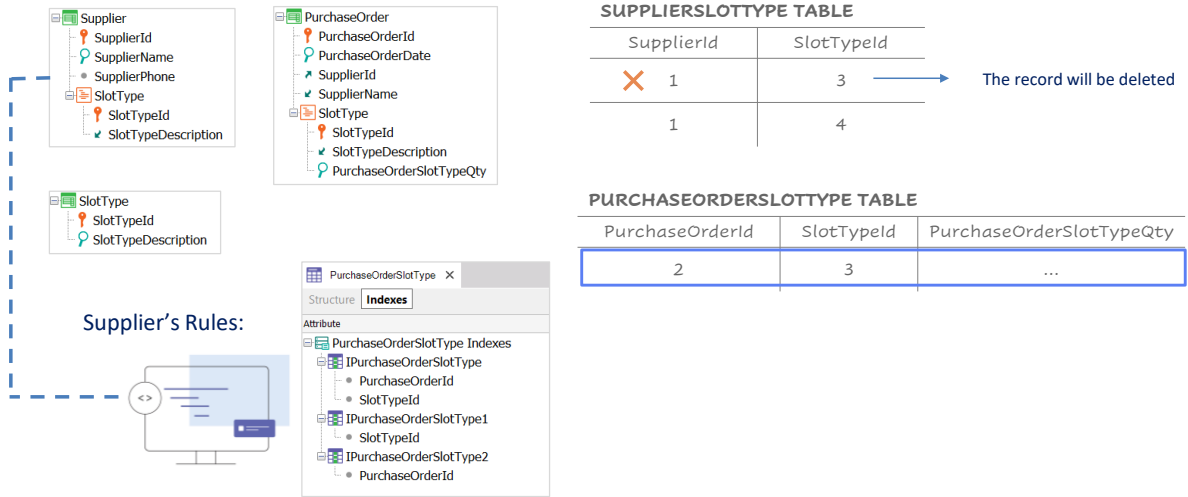
```

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

Podríamos utilizar la regla RefMsg, que sirve para personalizar el mensaje del error cuando se realiza un chequeo de integridad referencial en insert o update desde una clave foránea, y este falla. Esta regla tiene como parámetros - además del mensaje que queremos que se despliegue en caso de fallar el control de IR- el o los atributos que conforman la clave foránea.

Something to think about:

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



Una pregunta interesante es qué sucederá con esta solución si se desea eliminar un tipo de slot de un proveedor, para el que ya existan órdenes.

Es decir, si el usuario, a través de la transacción "Supplier" intenta eliminar una línea para la cual exista alguna "PurchaseOrder" con ese proveedor y tipo de slot.

En este caso GeneXus no podrá chequear la integridad, debido a que no existe un índice por {SupplierId, SlotTypeId} en la tabla PurchaseOrderSlotType. Como esa clave foránea solo existe en un nivel lógico (en la tabla extendida), no existirá índice alguno –dado que un índice corresponde a un concepto físico. Por lo tanto este chequeo deberemos programarlo en las reglas de la transacción "Supplier", invocando a un procedimiento que lo realice y luego disparando un error de acuerdo al resultado devuelto.

GeneXus™

training.genexus.com

wiki.genexus.com

training.genexus.com/certifications