

Examen Analista Junior GeneXus 18

Realidad: Tienda de Mascotas.

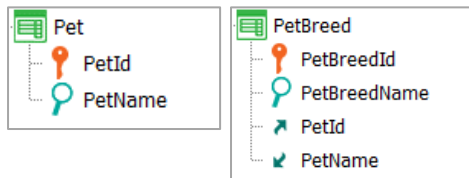
Sobre las preguntas de múltiple opción:

- Hay una sola opción correcta.
- Este examen NO resta puntos por respuestas incorrectas.

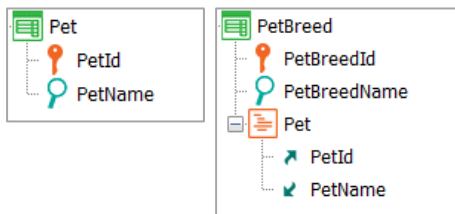
1) Se cuenta con una aplicación GeneXus para la gestión de una tienda de mascotas.

Sabiendo que una mascota (Pet) pertenece a una raza (PetBreed), y que muchas mascotas pueden ser de la misma raza, determina el diseño de transacciones que consideres correcto.

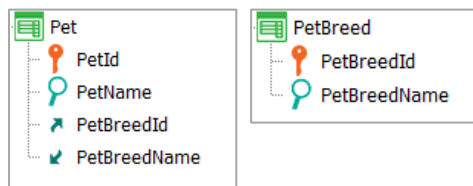
1.1 -



1.2 -



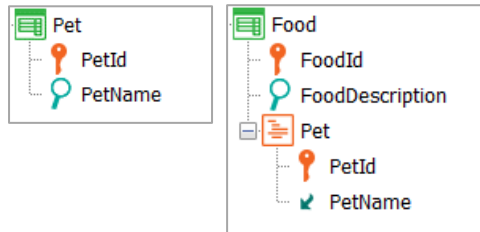
1.3 -



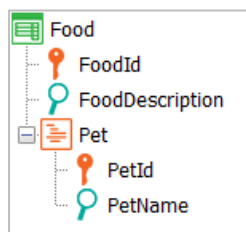
1.4 – Ninguna de las opciones anteriores es correcta.

- 2) Sabiendo que una mascota (Pet) puede consumir varios alimentos (Food), y que un mismo alimento puede ser consumido por varias mascotas, determina el diseño de transacciones que consideres correcto.

2.1 –



2.2 –

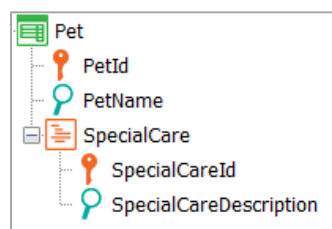


2.3 –



2.4 – Ninguna de las opciones anteriores es correcta.

- 3) Considera el diseño de la transacción que se muestra y determina lo que consideres correcto.



- 3.1 – Toda mascota (Pet) tiene asociado un conjunto de cuidados especiales (SpecialCare) que se identifican como únicos de esa mascota.

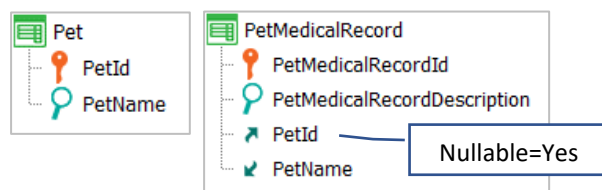
3.2 – Toda mascota (Pet) tiene asociado un conjunto de cuidados especiales (SpecialCare), y esos mismos cuidados no son propios de una única mascota, sino que pueden aplicarse a otras mascotas.

3.3 – El diseño no es válido. No es posible definir una transacción de dos niveles sin que la entidad del segundo nivel deba definirse además como una transacción en sí misma.

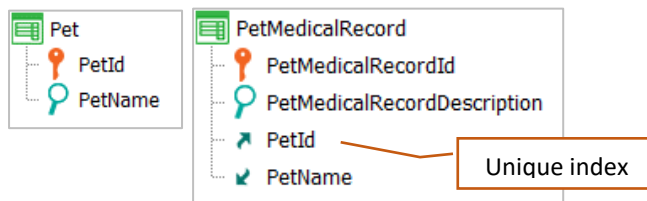
3.4 – Ninguna de las opciones anteriores es correcta.

4) Sabiendo que una mascota (Pet) tiene una única ficha médica (PetMedicalRecord), y que a su vez esa ficha médica es solamente de esa mascota, determina la opción que consideres correcta:

4.1 –



4.2 –

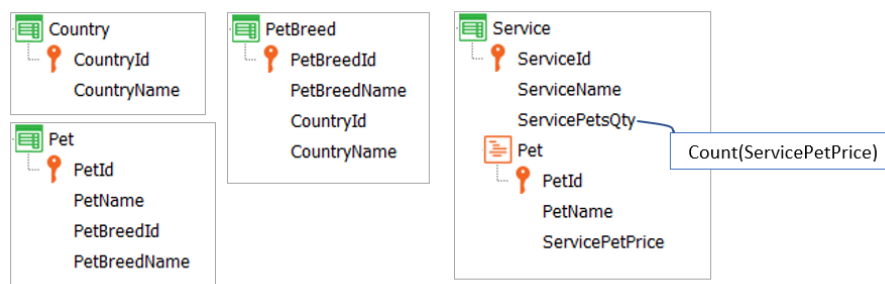


4.3 –

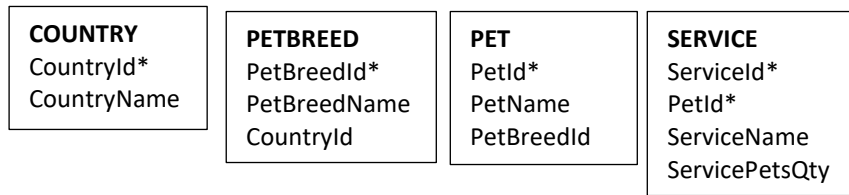


4.4 – Ninguna de las opciones anteriores es correcta.

5) Considera el diseño de transacciones que se muestra y determina la estructura física de las tablas que GeneXus creará.



5.1 -



5.2 -

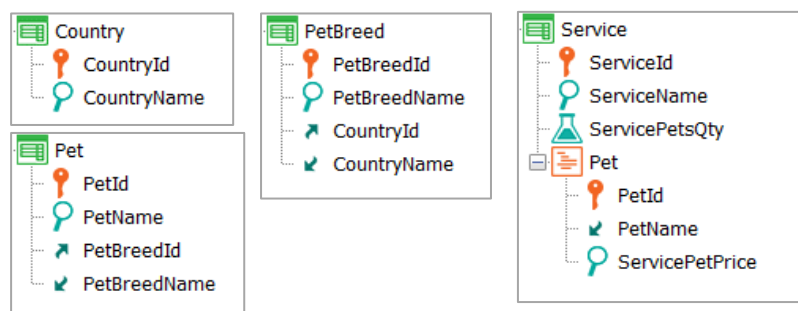


5.3 -



5.4 - Ninguna de las opciones anteriores es correcta.

6) A partir del diseño de transacciones que se muestra, determina la tabla extendida de la tabla SERVICEPET.



6.1 - SERVICEPET, SERVICE, PET

6.2 - SERVICEPET, SERVICE, PET, PETBREED, COUNTRY

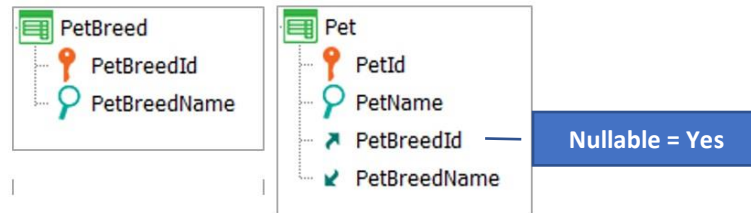
6.3 - SERVICEPET, SERVICE, PET, PETBREED

6.4 - SERVICEPET

- 7) Si bien toda mascota (Pet) tiene una raza (PetBreed), a veces no es posible determinar esa raza, y por lo tanto no se considera un dato obligatorio al registrar una mascota.

En caso de indicar una raza (PetBreedId), dicho valor debe ser válido.

A partir del diseño de transacciones que se muestra, determina lo que consideres correcto:



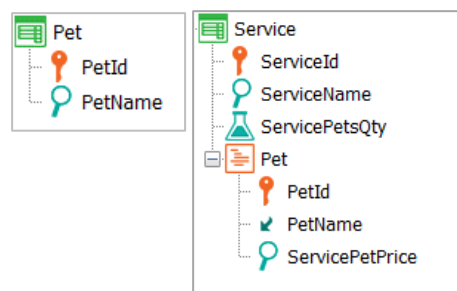
- 7.1) La implementación no resuelve el requerimiento. Al indicar que PetBreedId acepta nulos, GeneXus no realizará los controles de integridad referencial. Se podrá registrar una mascota sin raza, pero si se indica una raza entonces no se controlará si existe como registro en PETBREED.

- 7.2) La implementación no es correcta. Debe definirse un índice unique sobre el atributo PetBreedId en PET.

- 7.3) La implementación es correcta y resuelve el requerimiento.

- 7.4) Ninguna de las opciones anteriores es correcta.

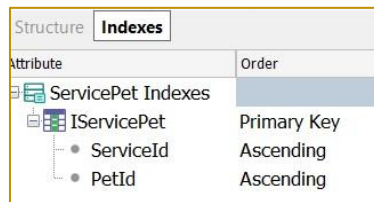
- 8) A partir del diseño de transacciones que se muestra, determina los índices automáticamente creados por GeneXus en la tabla SERVICEPET.



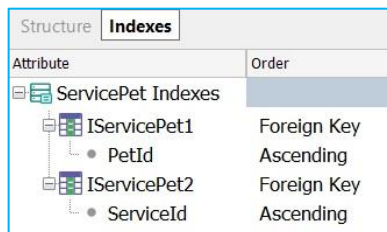
- 8.1)

Structure Indexes	
Attribute	Order
ServicePet Indexes	
IServicePet	Primary Key
• ServiceId	Ascending
• PetId	Ascending
IServicePet1	Foreign Key
• PetId	Ascending
IServicePet2	Foreign Key
• ServiceId	Ascending

8.2)

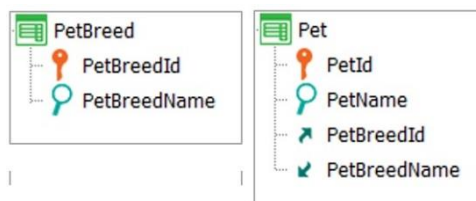


8.3)



8.4) Ninguna de las opciones anteriores es correcta.

- 9) Considera el diseño de transacciones que se muestra, y determina qué sucederá al momento de intentar eliminar una raza (PetBreed) utilizando el form de la transacción PetBreed.



9.1) GeneXus la eliminará sin realizar ningún tipo de control.

9.2) Automáticamente GeneXus eliminará primero todos los registros en Pet que tengan el PetBreedId como FK, y luego eliminará el correspondiente registro de PetBreed.

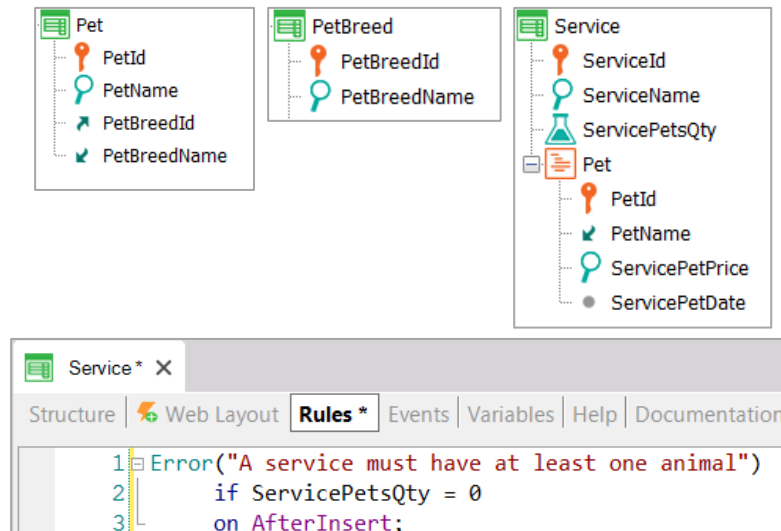
9.3) GeneXus controlará que no existan en Pet registros que tengan el PetBreedId como FK. En caso de existir, emitirá un mensaje indicando que existen registros relacionados y no realizará ninguna acción.

9.4) Ninguna de las opciones anteriores es correcta.

- 10) En el siguiente diseño de transacciones, la transacción Service tiene un atributo fórmula, ServicePetsQty, que cuenta la cantidad de animales registrados para determinado servicio en determinada fecha.

Se necesita controlar que un servicio nunca quede registrado sin ningún animal asociado. Para realizar este control se utiliza la regla Error mostrada a continuación.

Determina lo que consideres correcto:



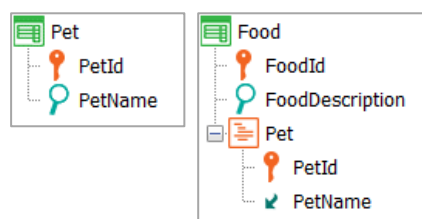
10.1 – La regla no cumple con el requisito solicitado, ya que será disparada en el servidor luego de que se hayan grabado en la base de datos los datos del cabezal (Service) y antes de que comiencen a grabarse los animales (Pet).

10.2 – La regla no cumple el requisito solicitado ya que será disparada en el servidor, luego de haber grabado en la base de datos los datos del cabezal (Service) e inmediatamente después de que se grabe el último animal (Pet).

10.3 – La regla no cumple el requisito solicitado, ya que será disparada en el servidor, exactamente antes de comenzar a grabar los datos del cabezal (Service).

10.4 – La regla cumple con el requisito solicitado, ya que será disparada en el cliente, antes de presionar Confirmar.

11) Considera las transacciones que se muestran y determina el orden en el cual se dispararán las reglas declaradas en la transacción Food.



Rules:

- FoodDetail(FoodId) on AfterComplete;
- Reservation(FoodId) on AfterInsert;
- StockControl(FoodId) on AfterLevel level PetId;

11.1 – b), c), a)

11.2 – c), b), a)

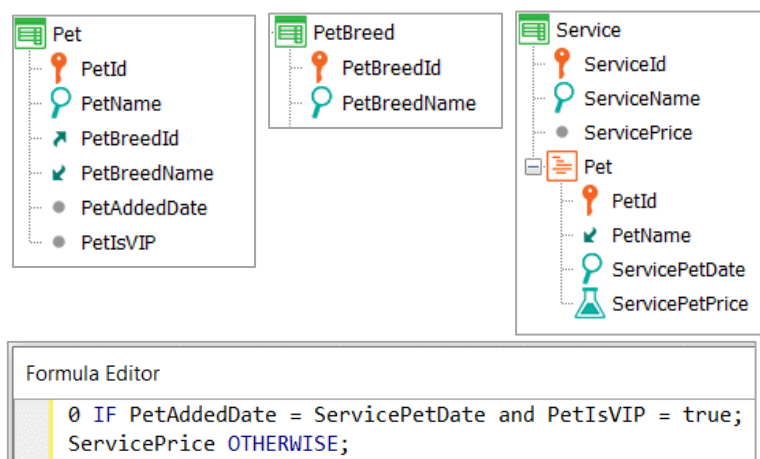
11.3 – c), a), b)

11.4 – Las reglas se disparan en el orden en el que son declaradas.

12) En la tienda de mascotas existen mascotas VIP, es decir, mascotas que cuentan con ciertos beneficios.

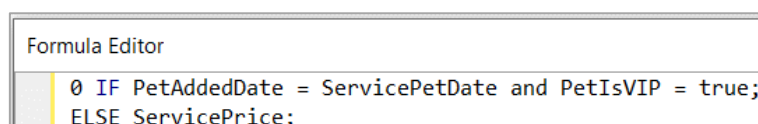
Al asociar una mascota a determinado servicio, si la mascota es VIP y el día para realizar el servicio coincide con el día en el que la mascota fue registrada en la tienda, el mismo no tendrá costo. De lo contrario, asumirá el precio base del servicio.

Determina lo que consideres correcto a partir del cálculo asociado al atributo ServicePetPrice.



12.1 – La implementación de la fórmula es incorrecta, ya que no es posible utilizar los atributos PetAddedDate y PetIsVIP en la misma, dado que estos no se encuentran en la estructura de la transacción Service (ni en el cabezal, ni en el subnivel Pet).

12.2 – La sintaxis de la fórmula es incorrecta, ya que cuando se usa la estructura IF debe utilizarse ELSE para referirse a los demás. La implementación válida sería así:

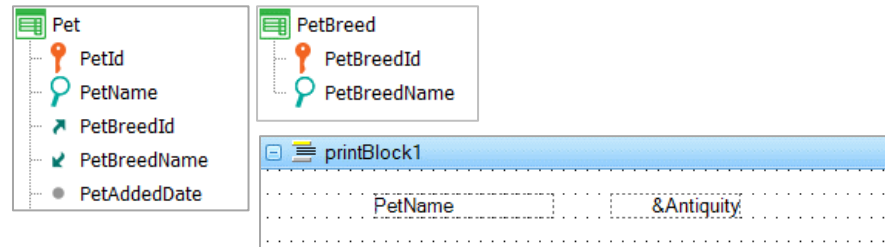


12.3 – La implementación de la fórmula resuelve el requerimiento solicitado.

12.4 – Ninguna de las anteriores es correcta.

- 13) Se necesita mostrar en un listado todos los animales (Pet) de la tienda de mascotas, con su nombre (PetName) y su antigüedad.

Observa la siguiente transacción y el Layout del procedimiento. ¿Cuál debe ser la implementación del source?



13.1 –

```

For each Pet
    &Antiquity = &Today.Year() - PetAddedDate.Year()
Endfor
Print printBlock1

```

13.2 –

```

&Antiquity = &Today.Year() - PetAddedDate.Year()
For each Pet
    Print printBlock1
Endfor

```

13.3 –

```

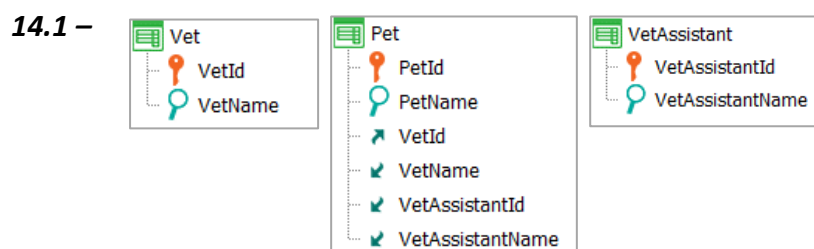
For each Pet
    &Antiquity = &Today.Year() - PetAddedDate.Year()
    Print printBlock1
Endfor

```

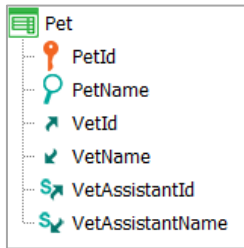
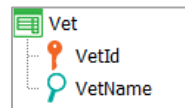
13.4 – Ninguna de las opciones anteriores está correcta.

- 14) Si bien toda mascota (Pet) tiene un veterinario de cabecera, se necesita registrar otro veterinario asistente para los casos en los que el veterinario de cabecera no esté disponible.

Determina cuál de los siguientes diseños de transacciones (y de grupos de subtipos si se incluyen), es el adecuado para modelar la realidad descrita anteriormente.



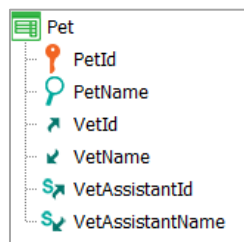
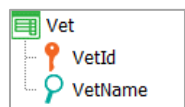
14.2 –



Subtype	Description	Supertype
VetAssistantId	Vet Assistant Id	VetId

Subtype	Description	Supertype
VetAssistantName	Vet Assistant Name	VetName

14.3 –



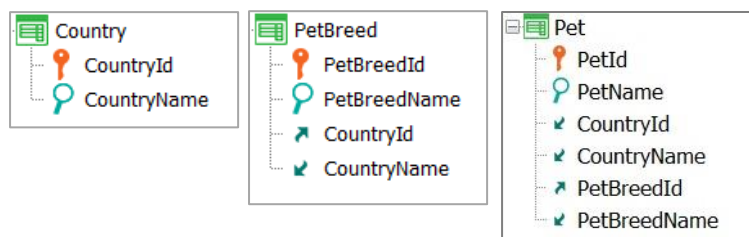
Subtype	Description	Supertype
VetAssistant		
VetAssistantId	Vet Assistant Id	VetId
VetAssistantName	Vet Assistant Name	VetName

14.4 – Ninguna de las opciones anteriores es correcta.

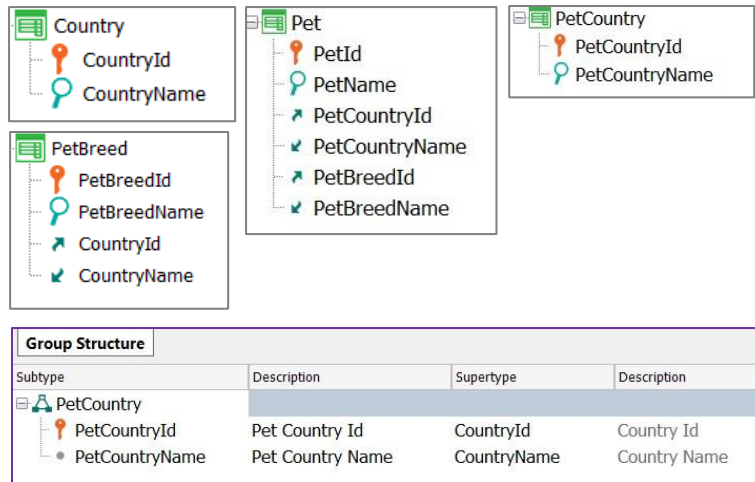
- 15) Si bien toda raza (PetBreed) se asocia con un determinado país de origen (Country), interesa registrar también el país donde la mascota (Pet) fue adquirida (adoptada) por sus dueños.

Determina la opción de implementación que consideres correcta:

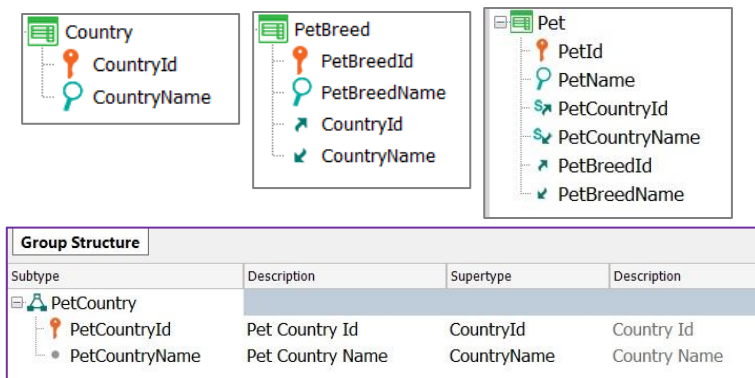
- 15.1) Simplemente por el orden en el cual se sitúan los atributos en la estructura de la transacción Pet, se podrá registrar el país donde fue adoptada la mascota.



15.2) Se debe definir la transacción PetCountry, y declarar el correspondiente grupo de subtipos.



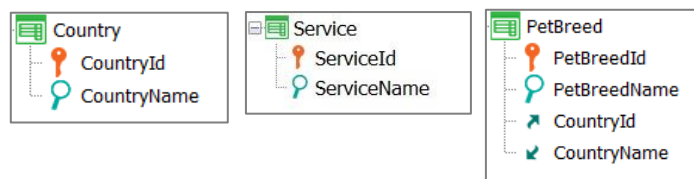
15.3) Se debe definir los atributos y grupo de subtipos como se muestra:



15.4) Ninguna de las opciones anteriores es correcta.

16) Observa el diseño de transacciones y el listado de navegación que se muestra.

¿Qué implementa?



Procedure Procedure2 Navigation Report			
Name:	Procedure2	Environment:	Default (C#)
Description:	Procedure2	Spec. Version:	17_0_3-149782
Output Devices:	File	Form Class:	Graphic
		Program Name:	Procedure2
LEVELS			
For Each Country (Line: 1)			
Order:	CountryId		
	Index: ICOUNTRY		
Navigation filters:	Start from:	FirstRecord	
	Loop while:	NotEndOfTable	
	Country (CountryId)		
For Each Service (Line: 8)			
Order:	ServiceId		
	Index: ISERVICE		
	Service (ServiceId)		

16.1) Producto Cartesiano

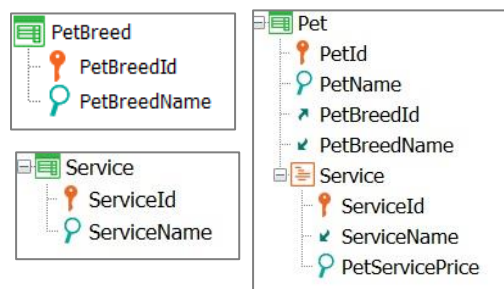
16.2) Corte de Control

16.3) Join

16.4) Ninguna de las opciones anteriores es correcta.

17) Considera el diseño de transacciones que se muestra. Se necesita listar la información completa (cabezal y líneas) de una determinada mascota (Pet) recibida por parámetro.

Determina la implementación que consideres correcta:



17.1) **Parm**(in: PetId);

For each Pet.Service

Print printBlock1

For each Pet.Service

Print printBlock2

Endfor

Endfor

The screenshot shows a report design with two blocks. The first block, 'printBlock1', contains fields for 'PetName' and 'PetBreedName'. The second block, 'printBlock2', contains fields for 'ServiceName' and 'ServicePetPrice'. Arrows from the text 'Print printBlock1' and 'Print printBlock2' point to these respective blocks.

17.2) **Parm**(in: &PetId);

For each Pet

Where PetId = &PetId

Print printBlock1

For each PetBreed

Print printBlock2

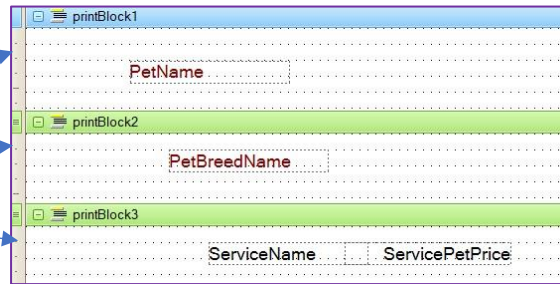
For each Pet.Service

Print printBlock3

Endfor

Endfor

Endfor



17.3) **Parm**(in: &PetId);

For each Pet

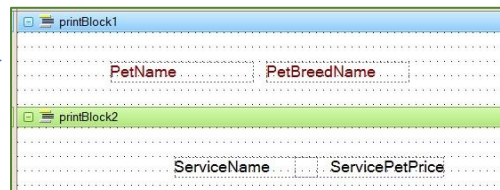
Where PetId = &PetId

Print printBlock1

Print printBlock2

Endfor

Endfor



17.4) **Parm**(in: PetId);

For each Pet

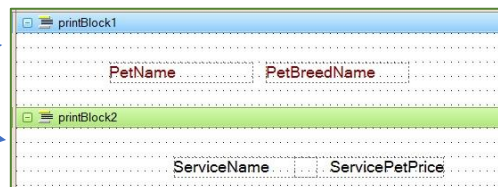
Print printBlock1

For each Pet.Service

Print printBlock2

Endfor

Endfor



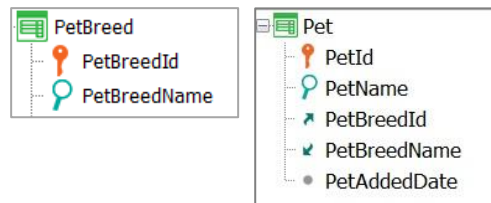
18) Considere el diseño de transacciones que se muestra. Se necesita definir un listado que muestre todas las razas (PetBreed) y para cada una, la lista de mascotas (Pet) pertenecientes a la misma.

Interesa ver en el listado todas las razas, independientemente de que tengan mascotas registradas de esa raza o no.

Determina la opción de implementación que consideres correcta para resolver adecuadamente el requisito descrito.

- 19) Considera el diseño de transacciones que se muestra. Se necesita listar las mascotas (Pet) de raza "Beagle" (PetBreedId = 4) y "Cocker" (PetBreedId = 7) que fueron registradas (PetAddedDate) en el año 2020.

Determina si la implementación propuesta es correcta o no.



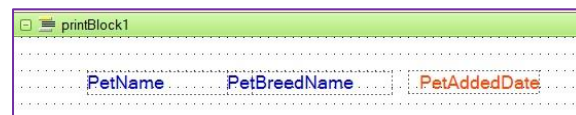
For each Pet

Where PetBreedId = 4 **or** PetBreedId = 7

Where PetAddedDate.Year() = 2020

Print printBlock1

Endfor


☐

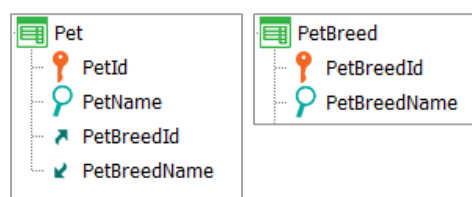
Verdadero

☐

Falso

- 20) Considera el diseño de transacciones que se muestra. Se necesita definir un listado que muestre todas las mascotas (Pet) agrupadas por raza (PetBreed).

Interesa que salgan en el listado solamente aquellas razas que tengan mascotas registradas.



20.1 –

For each PetBreed order PetBreedId

Print printblock1

Print printblock2

Endfor



20.2 –

For each PetBreed order PetBreadId

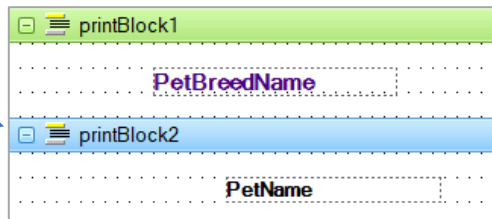
Print printblock1

For each Pet

Print printblock2

Endfor

Endfor



20.3 –

For each Pet order PetBreedId

Print printblock1

For each Pet

Print printblock2

Endfor

Endfor



20.4 –

For each Pet

Print printblock1

For each Pet

Print printblock2

Endfor

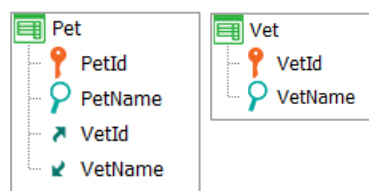
Endfor



21) Considera el diseño de transacciones que se muestra.

Se necesita definir un listado que muestre todos los veterinarios (Vet) registrados que tengan al menos una mascota a su cargo. Si no tiene ninguna, un texto debe informarlo.

Determina la opción de implementación que consideres correcta para resolver el requisito descrito.



21.1 –

```
For each Vet
  where Count(PetName) >= 1
  Print printBlock1
else
  Print printBlock2
Endfor
```



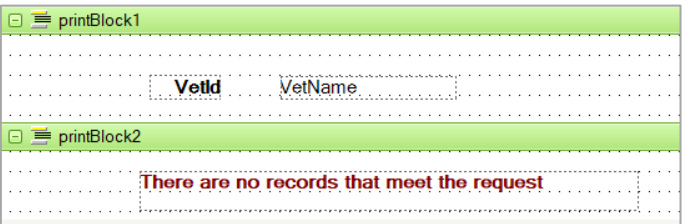
21.2 –

```
For each Vet
  where Count(PetName) >= 1
  Print printBlock1
unique
  Print printBlock2
Endfor
```



21.3 –

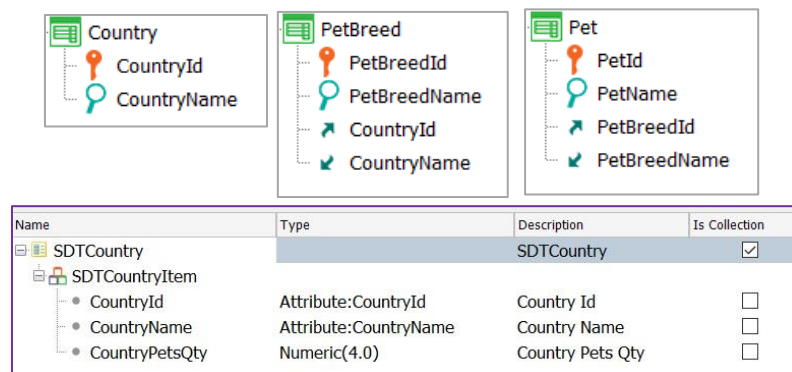
```
For each Vet
  where Count(PetName) >= 1
  Print printBlock1
when none
  Print printBlock2
Endfor
```



21.4 – Ninguna de las opciones anteriores es correcta.

- 22) Considera el diseño de transacciones y la definición del tipo de datos estructurado SDTCountry, que se muestran. Se necesita diseñar un Data Provider que cargue una colección de países (Country), cada uno con la cantidad de mascotas (Pet) de razas (PetBreed) de dicho país.

Determina la opción de implementación que consideres correcta.



22.1)

```
SDTCountry
{
    SDTCountryItem from Pet
    {
        CountryId
        CountryName
        CountryPetsQty = count(PetName)
    }
}
```

Output	
Infer Structure	No
Output	SDTCountry
Collection	False

22.2)

```
SDTCountry from Country
{
    SDTCountryItem
    {
        CountryId
        CountryName
        CountryPetsQty = count(PetName)
    }
}
```

Output	
Infer Structure	No
Output	SDTCountry
Collection	False

22.3)

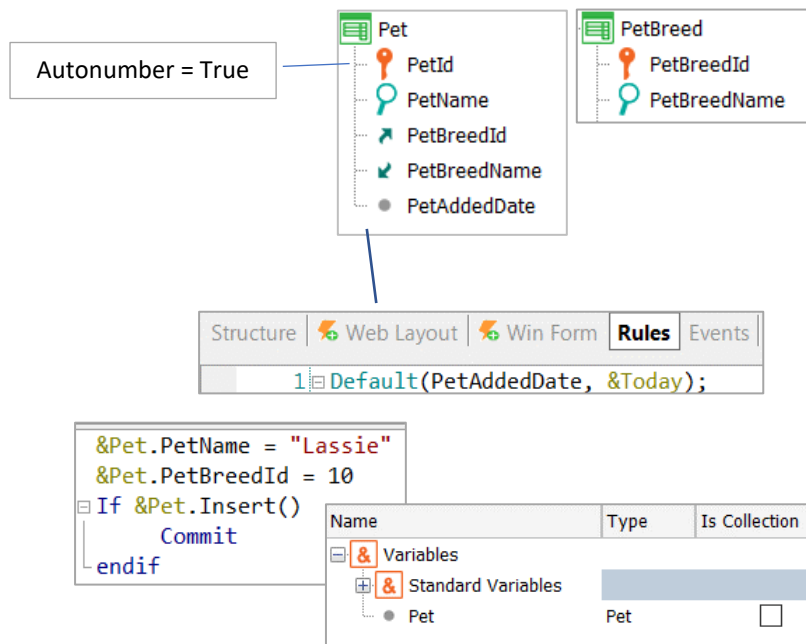
```
SDTCountry from PetBreed
{
    SDTCountryItem
    {
        CountryId
        CountryName
        CountryPetsQty = count(PetName)
    }
}
```

Output	
Infer Structure	No
Output	SDTCountry
Collection	True
Collection Name	Countries

22.4) Ninguna de las opciones anteriores es correcta.

23) Considere el diseño de la transacción que se muestra. La transacción Pet fue configurada como Business Component y el atributo PetId es autonumerado. Se desea insertar una nueva mascota (Pet) de nombre "Lassie", utilizando un Business Component de Pet.

Se ha programado un procedimiento con el siguiente código. Indique la opción que considera correcta.



23.1 – La mascota solo será insertada si existe la raza 10 en la tabla PetBreed. De lo contrario, fallará la integridad referencial y la misma no se insertará. Si es insertada, quedará con fecha de ingreso vacía, pues no se especificó ninguna en el código.

23.2 - La mascota solo será insertada si existe la raza 10 en la tabla PetBreed. De lo contrario, fallará la integridad referencial y la misma no se insertará. Si es insertada, quedará con fecha la fecha de ingreso del día de hoy.

23.3 – La mascota será insertada siempre, aunque no exista una raza con identificador 10 en la tabla PetBreed, pues los Business Components no controlan la integridad referencial. Quedará con fecha de ingreso vacía, pues no se especificó ninguna en el código.

23.4 – La mascota será insertada siempre, aunque no exista una raza con identificador 10 en la tabla PetBreed, pues los Business Components no controlan la integridad referencial. Quedará con fecha de ingreso del día de hoy.

24) Considera el diseño de transacciones y el layout del Web Panel que se muestra. Se desea diseñar un Web Panel que muestre los nombres de las mascotas (PetName) de determinada raza, seleccionada por el usuario.

Determina la opción que consideres correcta.

Control Info	
Control Type	Dynamic Combo Box
Data Source From	Attributes
Item Values	PetBreedId
Item Descriptions	PetBreedName

24.1 – Debe codificar el evento Load, como se muestra:

```

Event Load
  For each Pet
    where PetBreedId = &PetBreedId
      &PetBreedName = PetBreedName
      &PetId = PetID
      &PetName = PetName
      Load
    Endfor
  Endevent

```

24.2 – Debe codificar el evento Load, como se muestra:

```

Event Load
  For each Pet
    where PetBreedId = &PetBreedId
      Load
    Endfor
  Endevent

```

24.3 – Debe modificar el evento Start, como se muestra:

```

Event Start
  PetBreedId = &PetBreedId
Endevent

```

24.4 – Debe establecer la siguiente condición en el Grid:

```

Grid1's Conditions
  PetBreedId = &PetBreedId;

```

- 25) Considera el diseño de transacciones y el layout del Web Panel que se muestra. Se desea diseñar un Web Panel que muestre todas las razas (PetBreed), cada una con su respectiva cantidad de mascotas registradas. En caso de registrar más de 10 mascotas se visualizará el comentario “Many pets”. De lo contrario se visualizará “Few pets”.

Determina la opción de implementación que consideres correcta.



25.1 –

```

Event Load
  For each PetBreed
    &Quantity = Count(PetName)
    If &Quantity > 10
      &Comment = "Many pets"
    Else
      &Comment = "Few pets"
    Endif
    Load
  Endfor
EndEvent
  
```

25.2 –

```

Event Load
  &Quantity = Count(PetName)
  For each PetBreed
    Where &Quantity > 10
      &Comment = "Many pets"
    When none
      &Comment = "Few pets"
    Endfor
EndEvent
  
```

25.3 –

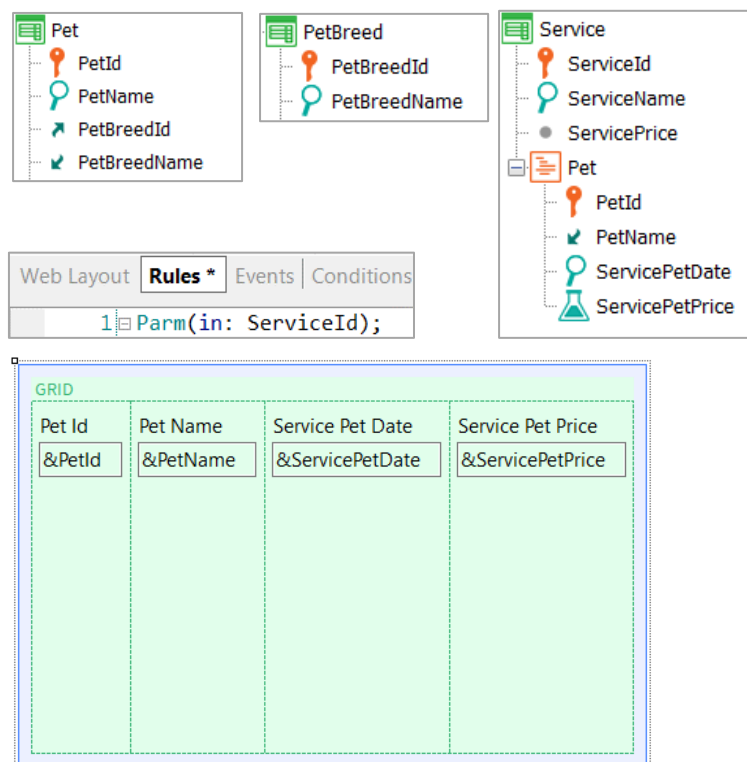
Event Load

```
&Quantity = Count(PetName)
If &Quantity > 10
    &Comment = "Many pets"
Else
    &Comment = "Few pets"
Endif
EndEvent
```

25.4 – Ninguna de las opciones anteriores es correcta.

26) En un Web Panel, se desea mostrar las mascotas asociadas a determinado servicio recibido por parámetro.

A partir del diseño de transacciones y del Web Panel que se detallan (lo que no se muestra, como las propiedades, es porque no se ha modificado salvo las de las variables que son todas ReadOnly), determina la opción que consideres correcta.



26.1 –

```
Event Load
  For each Service
    &PetId = PetId
    &PetName = PetName
    &ServicePetDate = ServicePetDate
    &ServicePetPrice = ServicePetPrice
    Load
  Endfor
Endevent
```

26.2 –

```
Event Load
  For each Service.Pet
    &PetId = PetId
    &PetName = PetName
    &ServicePetDate = ServicePetDate
    &ServicePetPrice = ServicePetPrice
    Load
  Endfor
Endevent
```

26.3 –

```
Event Load
  For each Service
    &PetId = PetId
    &PetName = PetName
    For each Service.Pet
      &ServicePetDate = ServicePetDate
      &ServicePetPrice = ServicePetPrice
      Load
    Endfor
  Endfor
Endevent
```

26.4 – Ninguna de las opciones anteriores es correcta.

RESPUESTAS

- 1) 3
- 2) 1
- 3) 1
- 4) 2
- 5) 3
- 6) 2
- 7) 3
- 8) 1
- 9) 3
- 10) 1
- 11) 1
- 12) 3
- 13) 3
- 14) 3
- 15) 3
- 16) 1
- 17) 4
- 18) 4
- 19) True
- 20) 3
- 21) 3
- 22) 2
- 23) 2
- 24) 4
- 25) 3
- 26) 2