

Exame Analista Júnior GeneXus 18

Realidade: Pet Shop.

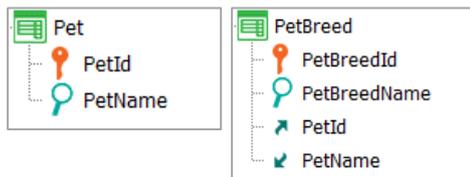
Sobre as questões de múltipla escolha:

- Existe apenas uma opção correta.
- Este exame NÃO subtrai pontos por respostas incorretas.

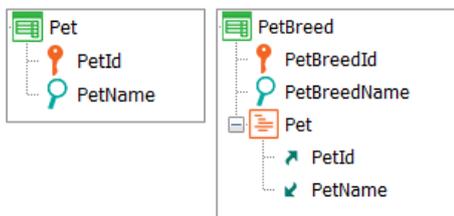
1) Existe uma aplicação GeneXus para a gestão de um pet shop.

Sabendo que um animal de estimação (Pet) pertence a uma raça (PetBreed), e que muitos animais de estimação podem ser da mesma raça, determine o desenho de transações que considere correto.

1.1 -



1.2 -



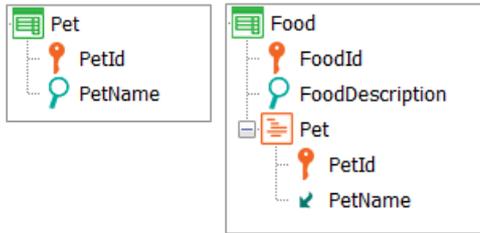
1.3 -



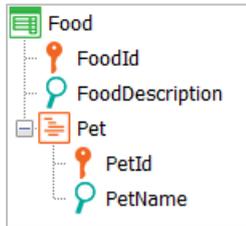
1.4 - Nenhuma das opções anteriores está correta.

2) Sabendo que um animal de estimação (Pet) pode consumir vários alimentos (Food), e que um mesmo alimento pode ser consumido por vários animais de estimação, determine o desenho de transações que considere correto.

2.1 –



2.2 –

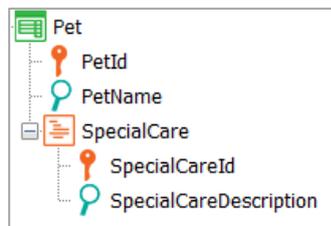


2.3 –



2.4 – Nenhuma das opções anteriores está correta.

3) Considere o desenho da transação apresentado e determine o que considere correto.



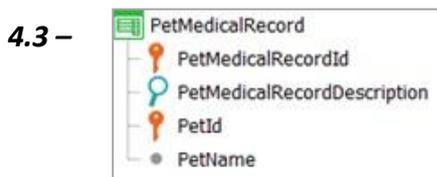
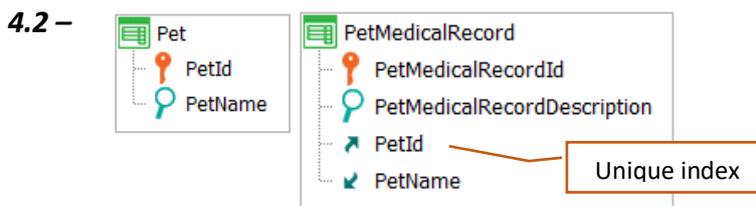
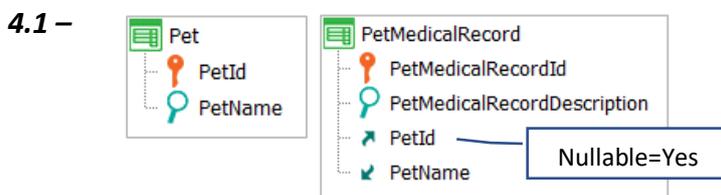
3.1 – Todo animal de estimação (Pet) tem associado um conjunto de cuidados especiais (SpecialCare) que são identificados como exclusivos daquele animal de estimação.

3.2 – Todo animal de estimação (Pet) tem associado um conjunto de cuidados especiais (SpecialCare), e esses mesmos cuidados não são próprios de um único animal de estimação, mas podem ser aplicados a outros animais de estimação.

3.3 – O desenho não é válido. Não é possível definir uma transação de dois níveis sem que a entidade do segundo nível tenha que ser definida também como uma transação em si mesma.

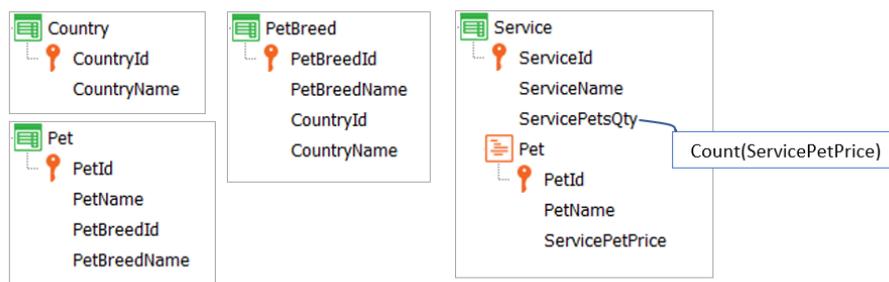
3.4 – Nenhuma das opções anteriores está correta.

4) Sabendo que um animal de estimação (Pet) tem uma única ficha médica (PetMedicalRecord), e que por sua vez essa ficha médica é somente daquele animal de estimação, determine a opção que considere correta:

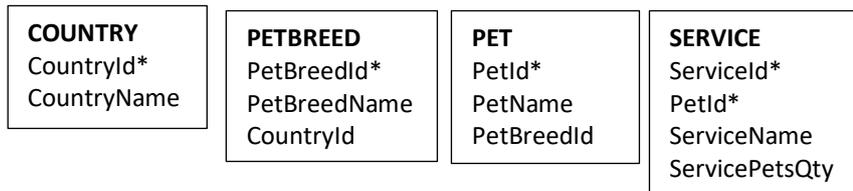


4.4 – Nenhuma das opções anteriores está correta.

5) Considere o desenho de transações apresentado e determine a estrutura física das tabelas que GeneXus criará.



5.1 -



5.2 -

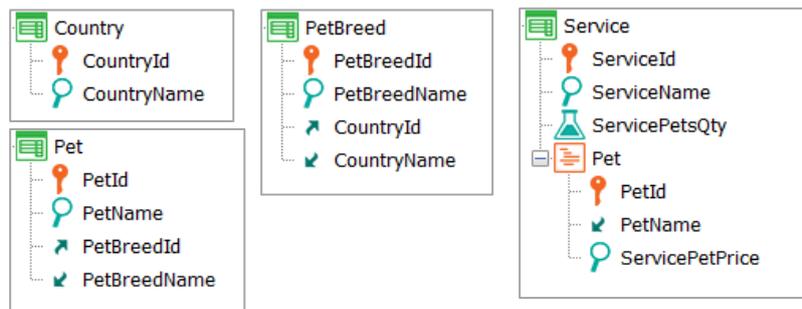


5.3 -



5.4 - Nenhuma das opções anteriores está correta.

6) A partir do desenho de transações apresentado, determine a tabela estendida da tabela SERVICEPET.



6.1 - SERVICEPET, SERVICE, PET

6.2 - SERVICEPET, SERVICE, PET, PETBREED, COUNTRY

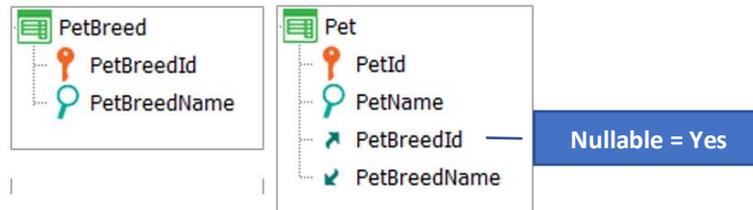
6.3 - SERVICEPET, SERVICE, PET, PETBREED

6.4 - SERVICEPET

- 7) Embora todo animal de estimação (Pet) tenha uma raça (PetBreed), às vezes não é possível determinar essa raça e, portanto, não é considerado um dado obrigatório ao registrar um animal de estimação.

Em caso de indicação de uma raça (PetBreedId), este valor deve ser válido.

A partir do desenho de transações apresentado, determine o que considere correto:



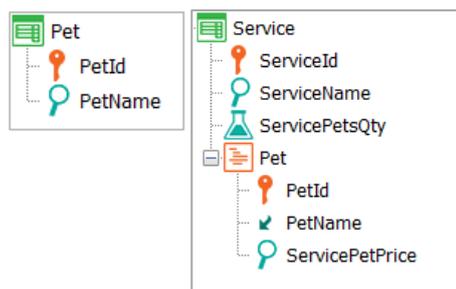
- 7.1) A implementação não resolve o requisito. Ao indicar que PetBreedId aceita nulos, GeneXus não realizará os controles de integridade referencial. Será possível registrar um animal de estimação sem raça, mas se for indicada uma raça, não será controlado se existe como registro em PETBREED.

- 7.2) A implementação não está correta. Deve ser definido um índice unique sobre o atributo PetBreedId em PET.

- 7.3) A implementação está correta e resolve o requisito.

- 7.4) Nenhuma das opções anteriores está correta.

- 8) A partir do desenho de transações apresentado, determine os índices criados automaticamente por GeneXus na tabela SERVICEPET.



- 8.1)

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

8.2)

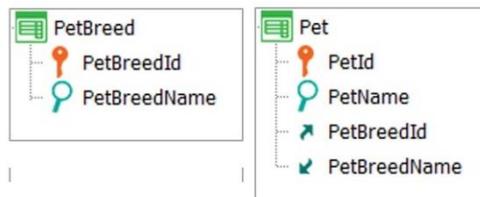
Attribute	Order
ServicePet Indexes	
IServicePet	Primary Key
ServiceId	Ascending
PetId	Ascending

8.3)

Attribute	Order
ServicePet Indexes	
IServicePet1	Foreign Key
PetId	Ascending
IServicePet2	Foreign Key
ServiceId	Ascending

8.4) Nenhuma das opções anteriores está correta.

- 9) Considere o desenho de transações apresentado e determine o que acontecerá no momento de tentar remover uma raça (PetBreed) utilizando o form da transação PetBreed.



9.1) GeneXus a removerá sem realizar nenhum tipo de controle.

9.2) Automaticamente GeneXus removerá primeiro todos os registros em Pet que tenham o PetBreedId como FK, e depois removerá o correspondente registro de PetBreed.

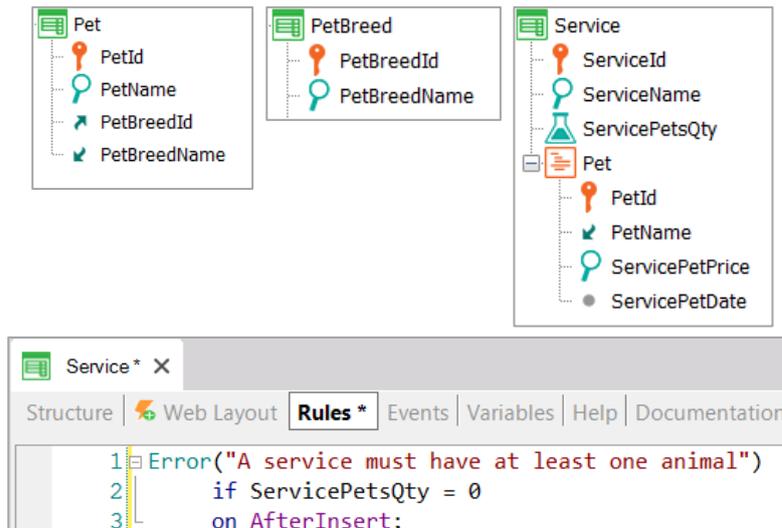
9.3) GeneXus controlará que não existam em Pet registros que tenham o PetBreedId como FK. No caso de existir, emitirá uma mensagem indicando que existem registros relacionados e não executará nenhuma ação.

9.4) Nenhuma das opções anteriores está correta.

- 10) No desenho de transações a seguir, a transação Service tem um atributo fórmula, ServicePetsQty, que conta a quantidade de animais registrados para determinado serviço em determinada data.

É necessário controlar que um serviço nunca seja registrado sem nenhum animal associado. Para realizar este controle, é utilizada a regra Error mostrada a seguir.

Determine o que considere correto:



10.1 – A regra não atende ao requisito solicitado, pois será disparada no servidor após terem sido gravados na base de dados os dados do cabeçalho (Service) e antes que comecem a ser gravados os animais (Pet).

10.2 – A regra não atende ao requisito solicitado, pois será disparada no servidor, após ter gravado na base de dados os dados do cabeçalho (Service) e imediatamente após gravar o último animal (Pet).

10.3 – A regra não atende ao requisito solicitado, pois será disparada no servidor, exatamente antes de começar a gravar os dados do cabeçalho (Service).

10.4 – A regra atende ao requisito solicitado, pois será disparada no cliente, antes de pressionar Confirmar.

11) Considere as transações apresentadas e determine a ordem em que serão disparadas as regras declaradas na transação 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 – As regras são disparadas na ordem em que são declaradas.

12) No pet shop existem animais de estimação VIP, ou seja, animais de estimação que possuem certos benefícios.

Ao associar um animal de estimação a determinado serviço, se o animal de estimação for VIP e o dia para realizar o serviço coincide com o dia em que o animal de estimação foi registrado na loja, o mesmo não terá custo. Caso contrário, assumirá o preço base do serviço.

Determine o que considere correto a partir do cálculo associado ao atributo ServicePetPrice.

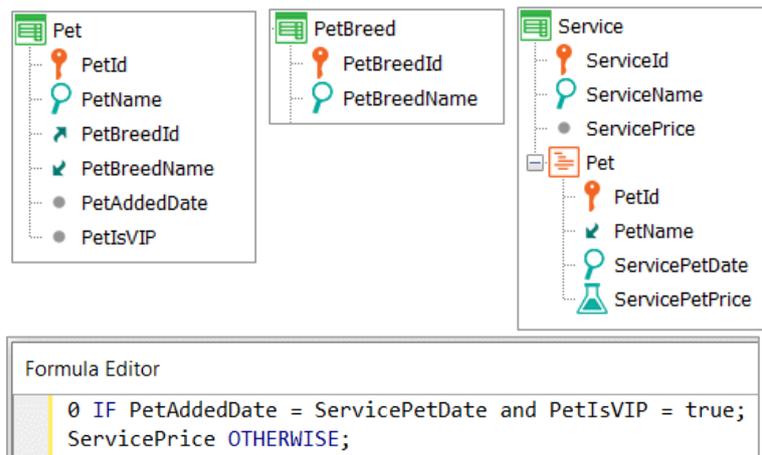


Diagram illustrating the data structure for the pet shop system. It shows three tables: Pet, PetBreed, and Service. The Pet table has attributes: PetId (Primary Key), PetName, PetBreedId (Foreign Key to PetBreed), PetBreedName, PetAddedDate, and PetIsVIP. The PetBreed table has attributes: PetBreedId (Primary Key) and PetBreedName. The Service table has attributes: ServiceId (Primary Key), ServiceName, ServicePrice, and a sub-table Pet. The sub-table Pet has attributes: PetId (Primary Key), PetName, ServicePetDate, and ServicePetPrice. Below the tables is a 'Formula Editor' window containing the formula: IF PetAddedDate = ServicePetDate and PetIsVIP = true; ServicePrice OTHERWISE;

12.1 – A implementação da fórmula está incorreta, pois não é possível utilizar os atributos PetAddedDate e PetIsVIP nela, uma vez que estes não se encontram na estrutura da transação Service (nem no cabeçalho, nem no subnível Pet).

12.2 – A sintaxe da fórmula está incorreta, pois ao usar a estrutura IF deve-se utilizar ELSE para se referir aos demais. A implementação válida seria assim:



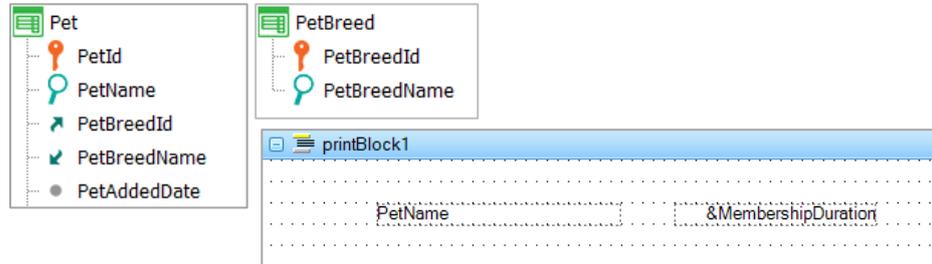
Diagram illustrating the corrected formula implementation. The 'Formula Editor' window contains the formula: IF PetAddedDate = ServicePetDate and PetIsVIP = true; ELSE ServicePrice;

12.3 – A implementação da fórmula resolve o requisito solicitado.

12.4 – Nenhuma das anteriores está correta.

- 13) É necessário apresentar em uma lista todos os animais (Pet) do pet shop, com seu nome (PetName) e seu tempo de cadastro (&MembershipDuration).

Observe a seguinte transação e o Layout do procedimento. Qual deve ser a implementação do source?



13.1 –

```
For each Pet
    &MembershipDuration = &Today.Year() - PetAddedDate.Year()
endfor
Print printBlock1
```

13.2 –

```
&MembershipDuration = &Today.Year() - PetAddedDate.Year()
For each Pet
    Print printBlock1
endfor
```

13.3 –

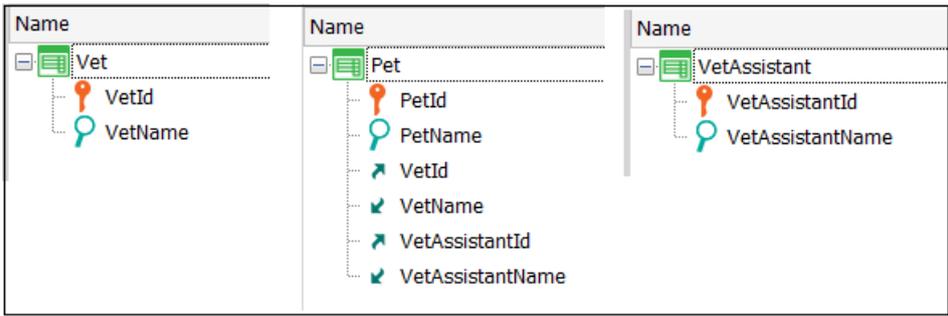
```
For each Pet
    &MembershipDuration = &Today.Year() - PetAddedDate.Year()
    Print printBlock1
endfor
```

13.4 – Nenhuma das opções anteriores está correta.

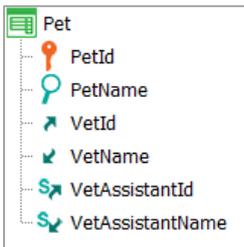
- 14) Embora todo animal de estimação (Pet) tenha um veterinário principal, é necessário registrar outro veterinário assistente para os casos em que o veterinário principal não esteja disponível.

Determine qual dos seguintes desenhos de transações (e de grupos de subtipos, se incluídos) é o adequado para modelar a realidade descrita anteriormente.

14.1 –



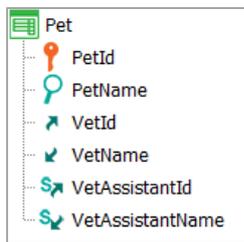
14.2 –



Subtype	Description	Supertype
VetAssistantId		
VetAssistantId	Vet Assistant Id	VetId

Subtype	Description	Supertype
VetAssistantName		
VetAssistantName	Vet Assistant Name	VetName

14.3 –



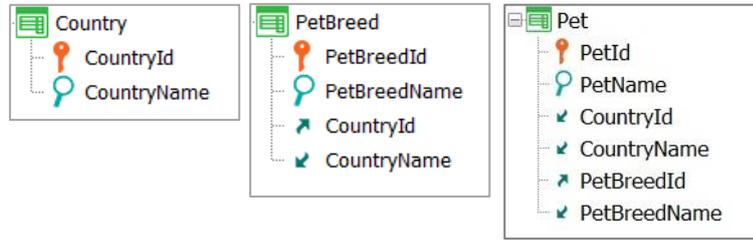
Subtype	Description	Supertype
VetAssistant		
VetAssistantId	Vet Assistant Id	VetId
VetAssistantName	Vet Assistant Name	VetName

14.4 – Nenhuma das opções anteriores está correta.

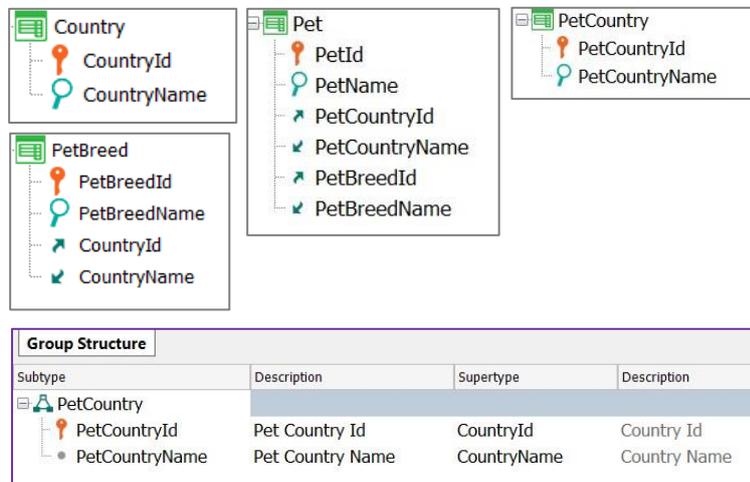
15) Embora toda raça (PetBreed) se associa com um determinado país de origem (Country), interessa registrar também o país onde o animal de estimação (Pet) foi adquirido (adotado) por seus donos.

Determine a opção de implementação que considere correta:

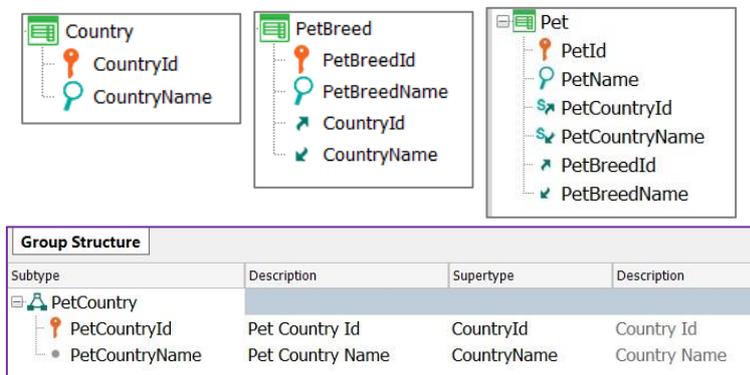
15.1) Simplesmente pela ordem em que são colocados os atributos na estrutura da transação Pet, será possível registrar o país onde o animal de estimação foi adotado.



15.2) Deve ser definida a transação PetCountry, e declarar o correspondente grupo de subtipos.



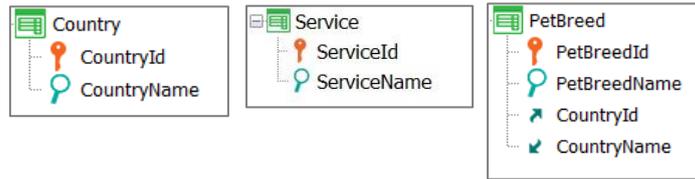
15.3) Devem ser definidos os atributos e grupo de subtipos conforme apresentado:



15.4) Nenhuma das opções anteriores está correta.

16) Observe o desenho de transações e a lista de navegação apresentados.

O que implementa?



Procedure Procedure2 Navigation Report

Name: Procedure2
Description: Procedure2
Output Devices: File

Environment: C# Default (C#)
Spec. Version: 17_0_3-149782
Form Class: Graphic
Program Name: Procedure2

LEVELS

For Each Country (Line: 1)

Order: CountryId
 Index: ICOUNTRY
 Navigation filters: Start from: FirstRecord
 Loop while: NoEndOfTable

Country (CountryId)

For Each Service (Line: 8)

Order: ServiceId
 Index: ISERVICE

Service (ServiceId)

16.1) Produto Cartesiano

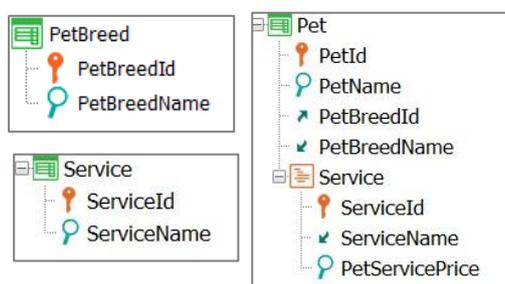
16.2) Corte de Controle

16.3) Join

16.4) Nenhuma das opções anteriores está correta.

17) Considere o desenho de transações apresentado. É necessário listar as informações completas (cabeçalho e linhas) de um determinado animal de estimação (Pet) recebido por parâmetro.

Determine a implementação que considere correta:



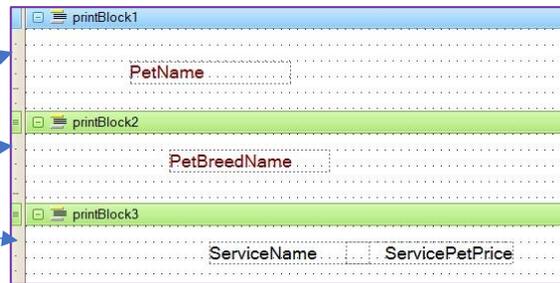
17.1) **Parm**(in: PetId);

For each Pet.Service
Print printBlock1
For each Pet.Service
Print printBlock2
Endfor
Endfor



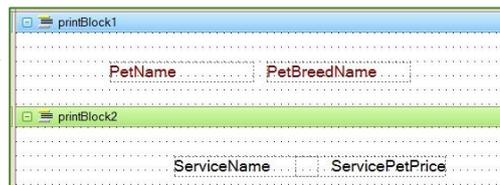
17.2) **Parm**(in: &PetId);

For each Pet
Where PetId = &PetId
Print printBlock1
For each PetBreed
Print printBlock2
For each Pet.Service
Print printBlock3
Endfor
Endfor
Endfor



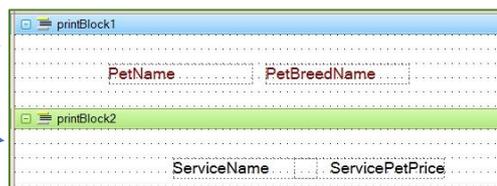
17.3) **Parm**(in: &PetId);

For each Pet
Where PetId = &PetId
Print printBlock1
Print printBlock2
Endfor
Endfor



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

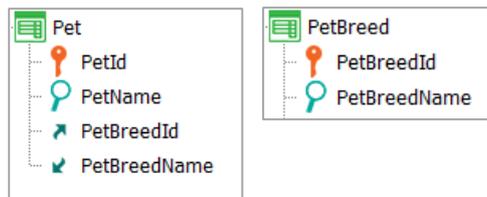
For each Pet
Print printBlock1
For each Pet.Service
Print printBlock2
Endfor
Endfor



18) Considere o desenho de transações apresentado. É necessário definir uma lista que mostre todas as raças (PetBreed) e para cada uma, a lista de animais de estimação (Pet) pertencentes à mesma.

Interessa ver na lista todas as raças, independentemente de terem animais de estimação registrados dessa raça ou não.

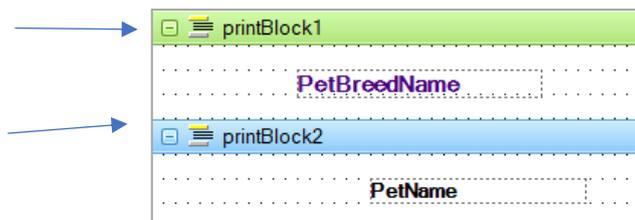
Determine a opção de implementação que considere correta para resolver adequadamente o requisito descrito.



18.1 –

```

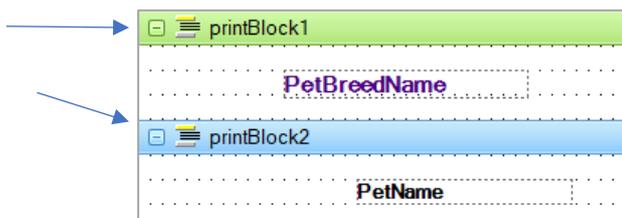
For each PetBreed
  Print printblock1
Endfor
For each Pet
  Print printblock2
Endfor
    
```



18.2 –

```

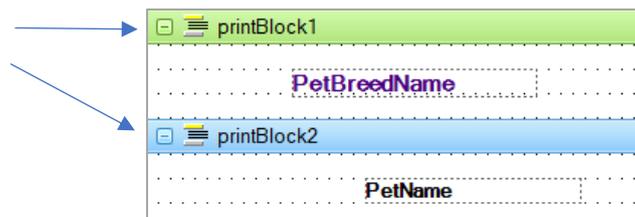
For each Pet
  Print printblock1
  For each Pet
    Print printblock2
  Endfor
Endfor
    
```



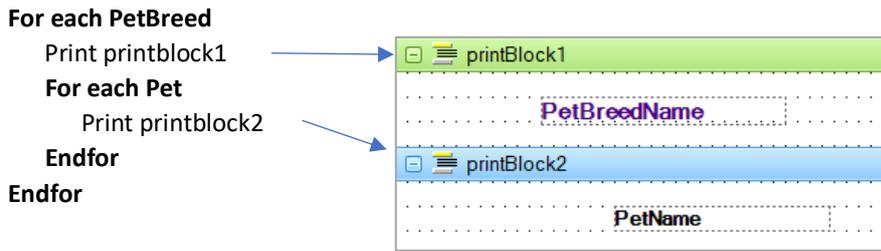
18.3 –

```

For each Pet
  Print printblock1
  Print printblock2
Endfor
    
```

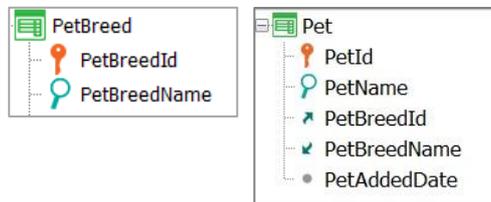


18.4 –



19) Considere o desenho de transações apresentado. É necessário listar os animais de estimação (Pet) da raça “Beagle” (PetBreedId = 4) e “Cocker” (PetBreedId = 7) que foram registrados (PetAddedDate) no ano de 2020.

Determine se a implementação proposta está correta ou não.



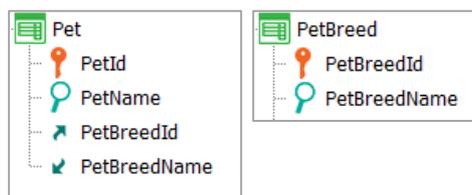
For each Pet
Where PetBreedId = 4 **or** PetBreedId = 7
Where PetAddedDate.Year() = 2020
 Print printBlock1
Endfor



Verdadeiro Falso

20) Considere o desenho de transações apresentado. É necessário definir uma lista que mostre todos os animais de estimação (Pet) agrupados por raça (PetBreed).

Interessa que apareçam na lista apenas aquelas raças que tenham animais de estimação registrados.



For each PetBreed order PetBreadId

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 PetBreadId

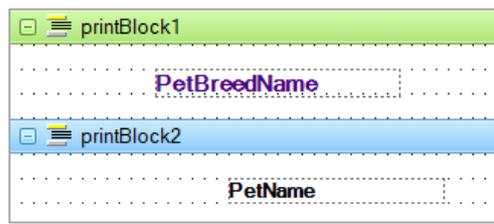
Print printblock1

For each Pet

Print printblock2

Endfor

Endfor



20.4 –

For each Pet

Print printblock1

For each Pet

Print printblock2

Endfor

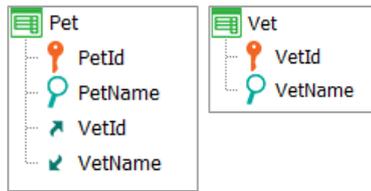
Endfor



21) Considere o desenho de transações apresentado.

É necessário definir uma lista que mostre todos os veterinários (Vet) registrados que tenham pelo menos um animal de estimação sob sua responsabilidade. Se não houver nenhum veterinário que tenha pelo menos um animal de estimação, um texto deve informar isso.

Determine a opção de implementação que considere correta para resolver o 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 –

```

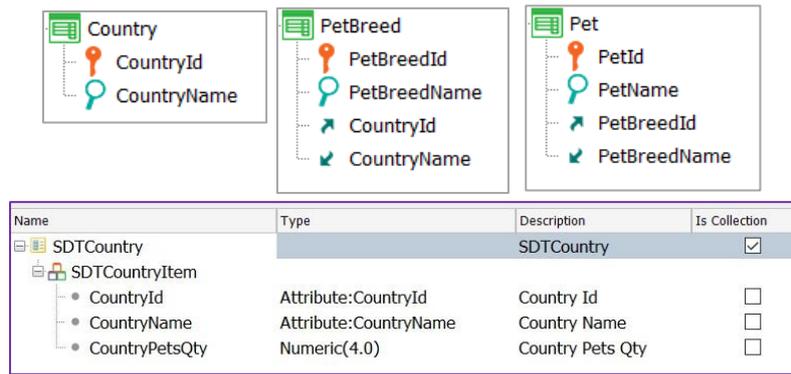
For each Vet
  where Count(PetName) >= 1
  Print printBlock1
when none
  Print printBlock2
Endfor
  
```



21.4 – Nenhuma das opções anteriores está correta.

22) Considere o desenho de transações e a definição do tipo de dados estruturado SDTCountry, apresentados. É necessário desenhar um Data Provider que carregue uma coleção de países (Country), cada um com a quantidade de animais de estimação (Pet) de raças (PetBreed) daquele país.

Determine a opção de implementação que considere correta.



22.1)

```
SDTCountry from Pet
{
  SDTCountryItem
  {
    CountryId
    CountryName
    CountryPetQty = 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)
  }
}
```

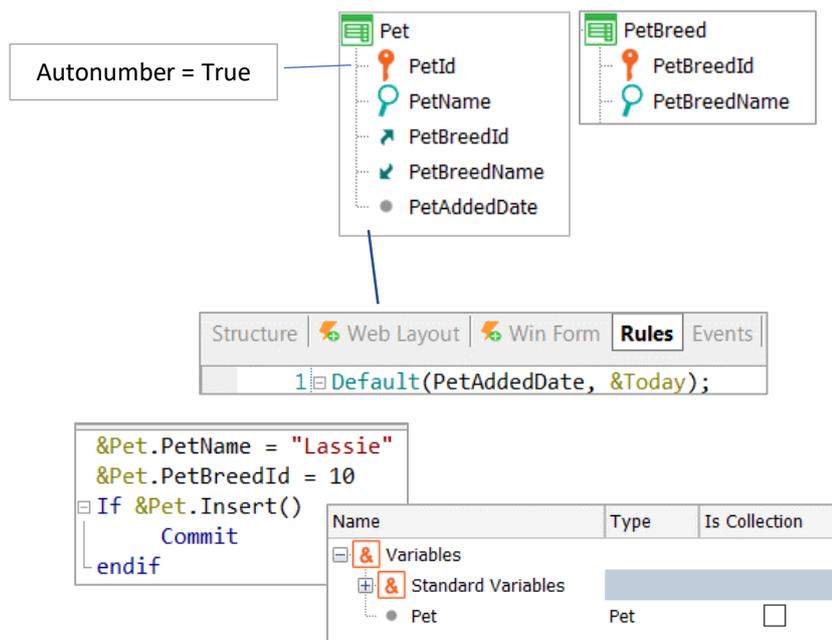
Output	
Infer Structure	No
Output	SDTCountry
Collection	True
Collection Name	Countries

22.4) Nenhuma das opções anteriores está correta.

23) Considere o desenho da transação apresentado. A transação Pet foi configurada como Business Component e o atributo PetId é autonumerado.

Deseja-se inserir um novo animal de estimação (Pet) chamado "Lassie", utilizando um Business Component de Pet.

Foi programado um procedimento com o seguinte código. Indique a opção que considera correta.



23.1 – O animal de estimação só será inserido se existe a raça 10 na tabela PetBreed. Caso contrário, falhará a integridade referencial e o mesmo não será inserido. Se for inserido, ficará com data de inserção vazia, pois nenhuma foi especificada no código.

23.2 - O animal de estimação só será inserido se existe a raça 10 na tabela PetBreed. Caso contrário, falhará a integridade referencial e o mesmo não será inserido. Se for inserido, ficará com data de inserção o dia de hoje.

23.3 – O animal de estimação será inserido sempre, mesmo que não exista uma raça com identificador 10 na tabela PetBreed, pois os Business Components não controlam a integridade referencial. Ficarà com data de inserção vazia, pois nenhuma foi especificada no código.

23.4 – O animal de estimação será inserido sempre, mesmo que não exista uma raça com identificador 10 na tabela PetBreed, pois os Business Components não controlam a integridade referencial. Ficarà com data de inserção do dia de hoje.

24) Considere o desenho de transações e o layout do Web Panel apresentado. Deseja-se desenhar um Web Panel que mostre os identificadores e os nomes dos animais de

estimação (PetName) de uma determinada raça, selecionada pelo usuário, bem como o nome da raça selecionada.

Determine a opção que considere correta.

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

24.1 – Deve codificar o evento Load, conforme apresentado:

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

24.2 – Deve codificar o evento Load, conforme apresentado:

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

24.3 – Deve modificar o evento Start, conforme apresentado:

```
Event Start
  PetBreedId = &PetBreedId
Endevent
```

24.4 – Deve definir a seguinte condição no Grid:

Grid1's Conditions
PetBreedId = &PetBreedId;

25) Considere o desenho de transações e o layout do Web Panel apresentado. Deseja-se desenhar um Web Panel que mostre todas as raças (PetBreed), cada uma com sua respectiva quantidade de animais de estimação registrados. No caso de registrar mais de 10 animais de estimação, será visualizado o comentário "Many pets". Caso contrário, será visualizado "Few pets".

Determine a opção de implementação que considere correta.



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
```

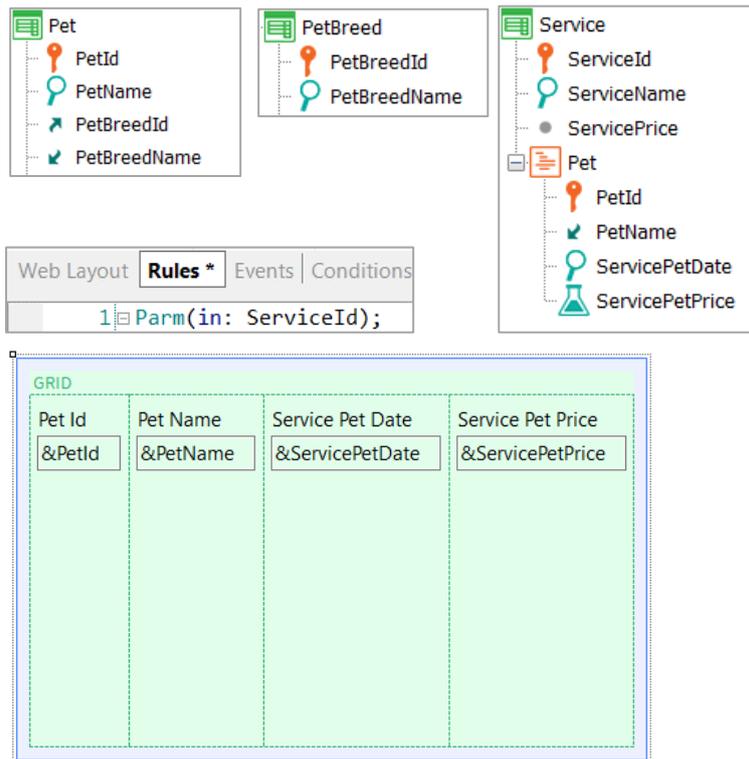
25.3 –

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

25.4 – Nenhuma das opções anteriores está correta.

26) Em um Web Panel, deseja-se mostrar os animais de estimação associados a um determinado serviço recebido por parâmetro.

A partir do desenho de transações e do Web Panel que são detalhados (o que não é mostrado, como as propriedades, é porque não foi modificado, exceto aquelas das variáveis que são todas ReadOnly), determine a opção que considere correta.



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 – Nenhuma das opções anteriores está correta.

RESPOSTAS

1) 3

Justificativa – Quando adicionamos um atributo que é chave primária de uma transação a outra transação, ele se torna uma chave estrangeira e é estabelecida uma relação 1-N forte, na qual a entidade onde está a chave estrangeira é o lado N da relação. Como o enunciado diz que 1 raça tem N animais de estimação (já que muitos animais de estimação podem ser da mesma raça), a resposta correta é a opção 3, pois a transação Pet tem PetBreedId como chave estrangeira. A opção 1 não pode ser porque a relação fica para o outro lado (um animal de estimação tem muitas raças) e a opção 2 representa o modelo de uma relação N para N.

2) 1

Justificativa – O enunciado estabelece uma relação de muitos-para-muitos entre alimentos e animais de estimação. Para modelar uma relação N-N em GeneXus, utilizamos uma transação de dois níveis, onde a entidade do segundo nível é adicionada também como uma transação separada. O modelo que representa o que é solicitado no enunciado é o da opção 1. A opção 2 não está correta porque representa uma relação 1 para N fraca (onde Pet é a entidade fraca, observemos que não é incluída a entidade Pet como transação independente) e a opção 3 representa uma relação 1-N forte, onde 1 alimento é consumido por N animais de estimação, mas não é cumprida a outra parte da relação.

3) 1

Justificativa – Quando usamos uma transação de dois níveis, estamos modelando uma relação 1 para N onde a entidade subordinada (aquela que está no segundo nível) é uma entidade fraca. Isso significa que para identificar corretamente esta entidade precisamos da entidade forte (aquela que está do lado 1 da relação), pois sem essa associação a entidade fraca perde o sentido. No exemplo, os cuidados especiais fazem sentido para um determinado animal de estimação, pois esses cuidados são únicos para ele e outro animal de estimação pode necessitar de outros cuidados. Notemos que com o modelo de transações proposto, GeneXus cria 2 tabelas, uma com os atributos do primeiro nível (Pet) e outra PetSpecialCare cuja chave primária é composta por PetId e SpecialCareId, demonstrando que efetivamente para identificar um cuidado especial é necessário o identificador do animal de estimação.

Portanto, a resposta correta é a opção 1, pois estabelece justamente isso e a prova é a chave primária composta que identificará o cuidado especial. A opção 2 estabelece que os cuidados não são próprios de um único animal de estimação e isso aconteceria se tivéssemos modelado uma relação 1-N forte entre Pet e SpecialCare utilizando duas transações independentes e com PetId como chave estrangeira. A opção 3 está incorreta, pois vimos exemplos de 1-N fraco modelados desta forma, como o caso do voo e seus assentos (entidade fraca).

4) 2

Justificativa – Para modelar uma relação 1 para 1 conforme a descrita no enunciado, em GeneXus utilizamos 2 transações modelando uma relação 1-N forte (usando uma chave estrangeira) e depois criando um índice único pelo atributo que é chave estrangeira. O índice único torna esse atributo uma chave candidata e, portanto, não pode ser repetido. A opção que representa isso corretamente é a 2, onde PetId é chave estrangeira em PetMedicalRecord e também é criado um índice único por esse atributo na tabela PetMedicalRecord. Isso faz com que não possa haver duas fichas médicas para o mesmo animal de estimação. A opção 1 não está correta porque modela a relação 1-N forte, mas em vez do índice único por PetId, é feito que este atributo seja nullable, o que não impede que ele possa ser repetido no caso de ser inserido. A opção 3 tem uma chave composta, mas isso não impede que possa ser utilizado o mesmo PetId para diferentes valores de PetMedicalRecordId.

5) 3

Justificativa – Este é um exercício de normalização e sabemos que GeneXus normaliza a tabelas na Terceira Forma Normal. A opção 1 tem corretas as tabelas COUNTRY, PETBREED e PET, mas a tabela SERVICE está errada pois inclui o atributo ServicePetsQty, que é fórmula e não é criado na tabela, e o atributo PetId, chave do segundo nível. Além disso, falta a tabela SERVICEPET. Na opção 2 as tabelas COUNTRY, PETBREED, PET e SERVICEPET estão corretas, mas a tabela SERVICE continua incluindo o atributo fórmula ServicePetsQty. A opção 3 é a única correta.

6) 2

Justificativa – Para responder corretamente esta questão é útil construir o diagrama de Bachmann das tabelas. A tabela estendida de SERVICEPET é o conjunto dos atributos da tabela base e todos aqueles que podem ser alcançados através de relações N para 1 de forma direta ou indireta. Portanto, a tabela estendida de SERVICEPET é: SERVICEPET (table base), SERVICE, PET, PETBREED e COUNTRY, o que corresponde à opção 2.

7) 3

Justificativa – A propriedade Nullable se aplica fundamentalmente a chaves estrangeiras e se o valor for Yes, permite que seja inserido o registro sem dar valor para essa chave estrangeira e não são realizados os controles de integridade referencial. No caso de ser inserido um valor, então será controlado se o valor existe como chave primária na tabela correspondente. Portanto, a opção correta é a 3, pois ao definir o atributo PetBreedId com Nullable como Yes, não é obrigatório inserir seu valor, mas se for inserido, é controlado que seja válido, como estabelece o enunciado da pergunta. A opção 1 não está correta porque se for inserido um valor, é controlado que exista como registro em PETBREED. A opção 2 não está correta porque se for definido um índice único sobre o atributo PetBreedId em PET, evita que o valor dele seja repetido, mas sempre deverá ser inserido.

8) 1

Justificativa – Quando GeneXus cria uma tabela na base de dados, automaticamente cria índices por chave primária e chaves estrangeiras. Isso é feito para que sejam

eficientes os controles de integridade referencial que são realizados de forma automática. A tabela SERVICEPET tem uma chave primária composta pelos atributos Serviceld e PetId, e também contém o atributo ServicePetPrice. Além disso, o atributo Serviceld é chave estrangeira para a tabela SERVICE e o atributo PetId é chave estrangeira para a tabela PET.

Portanto, GeneXus criará um índice por chave primária pelos atributos Serviceld e PetId, um índice por chave estrangeira por Serviceld e outro índice por chave estrangeira por PetId. A opção correta é a 1, pois é a única que contém os índices anteriormente mencionados. A opção 2 está incorreta porque faltam os índices por chave estrangeira e a opção 3 está incorreta porque falta o índice por chave primária.

9) 3

Justificativa – *Quando é utilizado o form da transação ou a transação como business component, GeneXus realiza os controles de integridade referencial de forma automática. Isso implica que se você deseja eliminar um registro de PetBreed, primeiro verificará se não há registros em Pet que contenham esse valor de PetBreedId como chave estrangeira. Caso seja encontrado, será dada uma mensagem e não será permitida a exclusão. Este comportamento é o explicado na opção 3, que é a opção correta. A opção 1 não está correta porque são realizados controles de integridade referencial quando está sendo utilizado o form da transação conforme estabelecido no enunciado. A opção 2 não está correta porque este comportamento corresponde à eliminação em cascata, que não é implementada em GeneXus.*

10) 1

Justificativa – *Para estabelecer a ordem de disparo das regras e fórmulas, GeneXus monta uma árvore de avaliação. Porém, como às vezes a ordem escolhida por GeneXus não se adapta a determinados requisitos, temos disponíveis determinados momentos de disparo que são realizados sempre no servidor, após ser pressionado Confirm no form da transação. Esses momentos de disparo começam com a cláusula “on” e em uma transação de dois níveis temos momentos antes e depois das instâncias de validação e gravação do cabeçalho, antes e depois da validação e gravação de cada linha, e antes e depois do Commit. A regra especificada no enunciado tem o momento de disparo on AfterInsert e inclui apenas atributos do cabeçalho (neste caso ServicePetsQty), portanto a regra será disparada unicamente quando estiver sendo inserido um registro do cabeçalho e imediatamente depois que o registro seja gravado, mas antes de começar a processar as linhas, ou seja, os registros do segundo nível.*

Como o requisito inicial do enunciado é que a regra deve evitar que seja gravado um serviço se não tiver animais registrados, a opção correta é a 1, pois com esta implementação será gravado primeiro o registro de serviço (cabeçalho) e só depois será disparada a regra, então a regra não realiza o controle evitando que seja gravado o registro. A opção 2 não está correta porque estabelece que a regra será disparada após validar e gravar todas as linhas (animais), que como vimos antes não é assim. A opção 3 não está correta porque indica que a regra será disparada antes de inserir o cabeçalho (quando o momento usado é on AfterInsert). E a opção 4 não

está correta, pois ao condicionar a regra com um momento de disparo, sempre será disparada no servidor após pressionar Confirm, e nunca no cliente.

11) 1

Justificativa – *A regra a) tem unicamente um atributo do cabeçalho (FoodId) e o momento de disparo AfterComplete é o último disponível, após ser efetuado o Commit. A regra b) tem unicamente um atributo do cabeçalho (FoodId) e o momento de disparo AfterInsert faz com que seja disparado imediatamente depois que o registro do cabeçalho é inserido. A regra c) tem um atributo do cabeçalho (FoodId) e usa o momento de disparo on AfterLevel Level PetId, sendo PetId um atributo das linhas (nível subordinado). Portanto, como o servidor primeiro realiza a validação e gravação do cabeçalho, depois a validação e gravação de cada linha, depois termina de processar as linhas e por último realiza o Commit, a ordem das regras será b), c), a), que corresponde à opção 1, que é a opção correta. As opções restantes estão incorretas e em particular a opção 4, devido à árvore de avaliação que GeneXus monta para o disparo (na qual leva em conta dependência entre regras e fórmulas), não podemos assumir que as regras são disparadas na ordem em que são declaradas.*

12) 3

Justificativa – *A fórmula apresentada no enunciado é uma fórmula horizontal, que se expressa de forma declarativa, pode conter uma ou várias atribuições condicionadas e executará unicamente a atribuição que tiver a condição atendida. Este tipo de fórmula navega um único registro e a tabela estendida, portanto os atributos mencionados devem pertencer todos à tabela estendida da tabela associada ao atributo fórmula.*

A fórmula está atribuindo ao atributo ServicePetPrice o valor 0 se o dia para realizar o serviço coincidir com o dia em que o animal de estimação foi registrado na loja e o animal de estimação também for VIP, e atribuirá o preço ServicePrice caso não sejam atendidas as condições anteriores (já que isso está condicionado com OTHERWISE). Portanto, a fórmula está correta e resolve o requisito solicitado, então a resposta correta é a 3. A opção 1 não está correta porque não leva em consideração que os atributos PetAddedDate e PetIsVIP pertencem à tabela estendida de SERVICEPET, que é o necessário. A opção 2 também não está correta porque estabelece que o IF deve ser utilizado com ELSE, sendo que este tipo de cláusula corresponde a uma programação procedural e não à programação declarativa que é a usada na definição das fórmulas.

13) 3

Justificativa – *A duração da adesão do animal de estimação em anos é calculada como a diferença entre o ano atual e o ano da data de adesão na loja de animais de estimação. A data de adesão é dada pelo atributo PetAddedDate e o nome do animal de estimação pelo atributo PetName. Para listar todos os animais da loja de animais de estimação com seu nome e duração da adesão, foi desenhado um layout onde no PrintBlock1 são incluídos o atributo PetName e a variável &MembershipDuration e para percorrer a tabela de animais de estimação é utilizado um comando For each Pet. A opção 1 não está correta pois a instrução para imprimir o PrintBlock1 está fora do For Each e embora a variável &MembershipDuration vá ficar com o valor do último registro de PET, o printBlock não poderá ser instanciado pois está fora do For Each, portanto não pode ser impresso e a lista*

nem sequer mostra a página. A opção 2 também não está correta porque agora é o atributo PetAddedDate que está fora do For each e não pode ser instanciado, portanto a variável &MembershipDuration ficará com o valor do ano atual (ex: 2024). O For Each imprimirá todos os nomes dos animais de estimação, com o valor do ano atual na duração da adesão. A opção correta é a 3, em que está incluído dentro do For each o cálculo da variável &MembershipDuration e a impressão do PrintBlock1 com os dados requeridos na lista. Desta maneira serão listados todos os animais de estimação da loja com seu nome e duração da adesão.

14) 3

Justificativa – *Tanto o veterinário titular quanto o substituto devem poder ser escolhidos entre aqueles armazenados na tabela VET, e caso seja inserido o identificador, ser controlado se este valor existe como identificador de um registro na tabela. A opção 1 não está correta porque embora o valor de VetId e VetAssistantId possam ser escolhidos de forma independente, o lógico é que a tabela com os veterinários seja única e não ter duas tabelas separadas com os nomes dos veterinários, como é o caso da tabela VETASSITANT onde pode inclusive acontecer que existam veterinários em VET que não estejam em VETASSITANT e vice-versa. A opção 2 também não está correta, porque o atributo VetAssistantName está em um grupo de subtipos diferente do VetAssistantId, portanto não será inferido o nome do veterinário quando for escolhido o identificador. A opção 3 é a correta, pois VetAssistantId é subtipo de VetId, portanto se comportará como tal, será chave estrangeira em Pet e serão feitos os controles de integridade referencial sobre Vet. Além disso, como o grupo de subtipos inclui VetAssistantName como subtipo de VetName, ao escolher o valor de VetAssistantId será inferido o valor de VetAssistantName, como pode ser visto na imagem da transação Pet onde é mostrado o atributo VetAssistantName como inferido.*

15) 3

Justificativa – *O que exige o enunciado é que possamos inserir o país do animal de estimação, independentemente do país da raça. Se observarmos a transação PetBreed, estão incluídos os atributos CountryId e CountryName. Ao incluir PetBreedId em Pet, se adicionamos o atributo CountryId, ele não será uma chave estrangeira para Country, mas será um atributo inferido de PetBreedId. Em outras palavras, tomará o valor do país da raça, será read only e não poderemos escolher um país para o animal de estimação. A opção 1 está incorreta porque mesmo que apareçam primeiro os atributos do país antes dos da raça, o país ainda será inferido da raça como pode ser visto na imagem, onde CountryId tem uma seta para baixo indicando que é um atributo inferido. A opção 2 adiciona mais uma transação PetCountry, com os atributos PetCountryId e PetCountryName, que então adiciona à transação Pet. Também adiciona um grupo de subtipos onde PetCountryId é subtipo de CountryId e PetCountryName é subtipo de CountryName, com o que é estabelecida uma relação de especialização relativa a CountryId e CountryName respectivamente. Esta solução, embora funcione, não é adequada porque adiciona objetos desnecessários à solução, como a transação PetCountry ou a relação de especialização. A solução correta é a 3, que inclui na transação Pet os atributos PetCountryId definido como subtipo de CountryId e PetCountryName definido como subtipo de CountryName, no respectivo grupo de subtipos. Esta solução permite inserir um país para o animal de estimação independente do país da raça, utilizando PetCountryId como chave estrangeira e*

PetCountryName como atributo inferido, sem a necessidade de adicionar objetos desnecessários.

16) 1

Justificativa – A lista de navegação do procedimento nos mostra dois *For Eachs* aninhados, o externo com tabela base *COUNTRY* (o *For Each* mais acima) e o aninhado (mostrado recuado) com tabela base *SERVICE*. Se analisarmos as transações, vemos que não existe relação alguma entre as tabelas *COUNTRY* e *SERVICE*, de forma que sabemos que se temos dois *For Eachs* aninhados, com tabela base diferente (neste caso *COUNTRY* e *SERVICE*), a navegação resultante será um produto cartesiano, ou seja, para cada registro da tabela *COUNTRY* serão listados todos os registros existentes na tabela *SERVICE*. Podemos ver isso na lista de navegação porque no nível aninhado que navega a tabela de *SERVICE* não há nenhuma indicação de filtro por contexto. Se *GeneXus* tivesse encontrado alguma relação entre *COUNTRY* e *SERVICE*, apareceria à direita da tabela um texto indicando um filtro implícito na forma: *CountryId = @CountryId*. Como não é o caso, podemos assegurar que *GeneXus* não encontrou nenhuma relação entre as tabelas bases dos *For Eachs* e, portanto, a navegação resultante será um produto cartesiano. A resposta correta é a 1.

17) 4

Justificativa – O enunciado pede para percorrer a tabela *PET*, filtrada pelo animal de estimação recebido por parâmetro e depois percorrer a tabela *PETSERVICE* para trazer os serviços associados a esse animal de estimação. Se percorrermos a tabela *PET*, os atributos de *PETSERVICE* não pertencem à tabela estendida de *PET*, portanto, precisamos de dois *For Eachs* aninhados, um que percorra *PET* e o aninhado que percorra *PETSERVICE*. Na opção 1, é recebido por parâmetro o atributo *PetId*, assim, automaticamente serão filtradas as navegações e apenas serão exibidos registros cujo *PetId* seja igual ao valor recebido por parâmetro. Depois temos 2 *For Eachs* aninhados, ambos com tabela base *PETSERVICE* (indicado pelo uso da transação base *Pet.Service* em ambos), o que causará um corte de controle, mas como falta a *order* por *PetId* no *for each* externo que provoque o agrupamento, serão listados os nomes de animais de estimação repetidos para cada serviço, portanto a opção 1 não está correta. Na opção 2 são utilizados três *For Eachs* aninhados, o que é totalmente desnecessário já que só precisamos de dois e também não é necessário para nada um *For Each* por *PETBREED*, já que esse atributo pertence à estendida de *PET*, então a opção 2 também não está correta. A opção 3 não está correta, pois implementa um único *For Each* com tabela base *PET* e como explicamos anteriormente, não poderão ser alcançados os atributos *ServiceName* e *SevicePetPrice* porque não pertencem à estendida de *PET*. A opção 4 possui os dois *For Eachs* aninhados necessários, o externo por *PET* e o aninhado por *PETSERVICE*. No *For Each* externo, não é incluída a cláusula *Where* para filtrar pelo valor de *PetId* porque o parâmetro é recebido no atributo *PetId*, que como já vimos funciona como filtro por igualdade pelo valor recebido, portanto em *PET* apenas será percorrido o registro do animal de estimação correspondente e por estarem os *For Eachs* aninhados e as tabelas *PET* e *PETSERVICE* estarem relacionadas, serão listados unicamente os serviços que correspondem ao animal de estimação recebido por parâmetro, portanto a opção 4 está correta.

18) 4

Justificativa – Como o enunciado solicita listar todas as raças registradas independentemente de haver animais de estimação dessa raça e para cada raça que tenha animais de estimação, os animais de estimação associados, estão solicitando um JOIN. Isto implica dois For Eachs aninhados, o externo com tabela base PETBREED (para que percorra todas as raças registradas) e o aninhado com tabela base PET. Para que haja um JOIN também é necessário que as tabelas bases estejam relacionadas, que é o caso entre PETBREED e PET. Se olharmos as opções, a opção 1 está incorreta porque apresenta dois For Each paralelos em vez de estarem aninhados. A opção 2 mostra dois For Eachs aninhados, mas sobre a mesma tabela base PET, que não é o que precisamos, portanto também não está correta. A opção 3 tem um único For Each com tabela base PET, com o qual não serão formados grupos por raça, mas será repetido o nome da raça para cada registro de PET, que não é o que se pretende, de que para cada raça sejam mostrados os animais de estimação dessa raça, portanto a opção 3 também não está correta. Se analisarmos a opção 4, vemos que agora temos dois For Eachs aninhados, o externo com tabela base PETBREED e o aninhado por PET, ou seja, diferentes tabelas base, mas relacionadas. Serão listadas todas as raças registradas em PETBREED e para cada uma apenas os animais de estimação dessa raça, pois devido a esta relação será efetuado um JOIN. Esta opção 4 é a correta.

19) True

Justificativa – A implementação mostra um For Each com tabela base PET, que percorrerá esta tabela unicamente para os animais de estimação com raça com PetBreedId = 4 ou PetBreedId = 7 e que também o ano da data de registro seja 2020. Isso atende aos requisitos estabelecidos no enunciado da pergunta. A dúvida que podemos ter é se todos os atributos incluídos nas cláusulas Where e no printblock pertencem ou não à tabela estendida de PET, que é a tabela base do For Each. Verificamos isso e vemos que é cumprido, então a implementação está correta e a resposta é VERDADEIRO.

20) 3

Justificativa – Como o enunciado solicita listar somente as raças que tenham animais de estimação registrados, cada uma com seus correspondentes animais de estimação, estão solicitando um CORTE DE CONTROLE. Não poderemos listar as raças a partir da tabela PETBREED, pois ali estão todas as raças, tenham ou não tenham animais de estimação registrados, portanto devemos recuperar as raças da tabela PET. Como neste caso recuperamos a raça de cada animal de estimação, será uma raça que tem animais de estimação registrados. Para implementar um corte de controle entre raças e animais de estimação, precisamos de dois For Eachs aninhados, ambos com a mesma tabela base e com a cláusula ORDER pelo atributo que queremos fazer o corte (agrupando por PetBreedId) no For Each externo. A opção 1 não está correta porque existe apenas um For Each e os animais de estimação não saem agrupados por raça. A opção 2 tem o For Each externo com tabela base PETBREED, que já vimos que não está correta. A opção 3 atende a tudo o que precisamos para implementar corretamente o corte de controle, portanto a opção 3 é a correta. Para a opção 4 falta a order que permite os agrupamentos, por isso está incorreta.

21) 3

Justificativa – O enunciado pede para listar todos os veterinários que tenham pelo menos um animal de estimação sob sua responsabilidade e caso não haja nenhum veterinário que tenha animais de estimação atribuídos a ele, deve ser exibida uma mensagem. Para isso

devemos percorrer a tabela VET filtrando aqueles registros que a contagem de animais de estimação é maior ou igual a 1. Todas as opções fazem isso corretamente, o problema é como implementamos a segunda parte, que é mostrar uma mensagem se não há nenhum veterinário com animais de estimação. A opção 1 não está correta porque usa um ELSE que não faz parte da sintaxe do For Each. A opção 2 após imprimir o printBlock1 usa uma cláusula UNIQUE, mas esta cláusula é usada para evitar repetidos e não se aplica a este caso, portanto também não está correta. A opção 3 está correta porque utiliza WHEN NONE, que é justamente a cláusula adequada na sintaxe do For Each para usar nesses casos, ou seja, quando não há nenhum registro que atenda às condições dos filtros.

22) 2

Justificativa – O tipo de dados estruturado SDTCountry é coleção e para carregar essa coleção, o Data Provider deverá percorrer a tabela COUNTRY para obter o identificador do país, o nome e a quantidade de animais de estimação. Para isso, o Data Provider deve incluir a cláusula from Country para indicar a transação base, cuja tabela base será a tabela percorrida pelo Data Provider. Além disso, o Data Provider deve ter a propriedade Collection definida como False, pois o output será a estrutura SDTCountry que já é coleção. As três opções têm a mesma implementação para o item SDTCountryItem e esta implementação está correta. A opção 1 não está correta porque a transação base é Pet e não Country. A opção 2 é a correta, pois cumpre que seja percorrida a tabela base COUNTRY e que a propriedade Collection está definida como False. A opção 3 não está correta, pois tenta percorrer a tabela PETBREED.

23) 2

Justificativa – Para inserir um registro na tabela PET usando a variável &Pet definida como business component de Pet, o procedimento deveria atribuir valores ao PetName, PetBreedId e PetAddedDate, pois ao PetId não é necessário por ser autonumerado. Porém, como a transação Pet tem definida a regra Default que atribui a data do dia para PetAddedDate, não é necessário atribuir este valor na implementação do procedimento. E como sabemos que GeneXus controla a integridade referencial quando são usados business component, o valor de PetBreedId deve ser válido, ou seja, deve existir como chave primária na tabela PETBREED. Após atribuir esses valores, o source do procedimento invoca o método Insert() e se o resultado for verdadeiro, é realizado o Commit, o que está corretamente implementado. A opção 1 tem a primeira parte correta quando fala do valor de PetBreedId, mas não está certo que, se for inserido o registro fica vazia a data PetAddedDate, pois a regra Default é disparada e é atribuído esse valor. A opção 2 é a correta porque explica exatamente o que acontecerá ao ser executado o código implementado. A opção 3 não está correta porque não está certo que, se usarmos BC não são implementados os controles de integridade referencial e também devido à regra a data não ficará vazia. A opção 4 também não está correta, porque embora seja correto que é atribuída a data de hoje, não está certa a afirmação de que ao usar BC não serão realizados os controles de integridade referencial.

24) 4

Justificativa – O Grid tem tabela base pois tem os atributos PetBreedName, PetId e PetName. Para filtrar o Grid com tabela base, utilizamos sua propriedade Conditions com a condição de que PetBreedId seja igual ao valor selecionado na variável &PetBreedId. A opção 1 está

incorreta, pois como o Grid tem tabela base, GeneXus cria um For Each implícito sobre a tabela PET que permite obter os valores de PetId e PetName, bem como PetBreedName pela tabela estendida e, portanto, não é necessário implementar um For each no evento Load para recuperar esses dados. A opção 2 também não está correta porque para filtrar pelo valor de PetBreedId selecionado na variável, devem ser utilizadas as Conditions do Grid e não um For Each no evento Load, pois este For Each ficaria aninhado ao For Each implícito e este where não filtrará o referido For Each implícito. A opção 3 também não está correta porque o evento Start é disparado apenas uma vez e com essa atribuição não ficará filtrado o For Each implícito pelo valor da raça selecionada. A opção 4 é a opção correta, pois com essa condição adicionada às propriedades do Grid GeneXus conseguirá filtrar os registros do grid pelo valor escolhido de &PetBreedId.

25) 3

Justificativa – *Como o Grid tem tabela base PETBREED, devido ao valor da propriedade Base Trn, sabemos que o evento Load será disparado N vezes, uma vez para cada registro da tabela PETBREED. É no evento Load que devemos obter a quantidade de animais de estimação -utilizando a fórmula count(PetName)- e também dependendo desta quantidade atribuir o valor da variável &Comment. A opção 1 não está correta porque, por se tratar de um Grid com tabela base, será criado um For Each implícito e não é necessário adicionar um For Each no evento Load. A opção 2, embora tente uma implementação diferente da 1, também não está correta porque também adiciona um For Each desnecessário, que também ficará aninhado ao implícito. A opção 3 é a opção correta, pois para cada registro de PETBREED será disparado o evento Load e será calculada a quantidade de animais de estimação para a raça que está sendo percorrida e então se a quantidade for maior que 10 será atribuída a mensagem de “muitos animais de estimação” e para o caso contrário “poucos animais de estimação”. A opção 4, portanto, também não está correta.*

26) 2

Justificativa – *Como no Grid não existem atributos e fica esclarecido que o restante das propriedades do Grid e seções do Web Panel possuem valores padrão, podemos afirmar que o Web Panel não tem tabela base. Neste caso sabemos que o evento Load será disparado apenas uma vez e, portanto, neste evento devemos implementar um For Each que percorra a tabela SERVICEPET para que através de sua estendida possamos recuperar os valores dos atributos PetId, PetName, ServicePetDate e ServicePetPrice e atribuir os valores das variáveis do Grid, utilizando o comando Load para copiar os sucessivos valores para cada linha do Grid. A opção 1 não está correta porque a transação base do For Each é Service, e a partir da tabela base SERVICE não é possível alcançar os atributos que necessitamos. A opção 2 é a correta, pois atribuindo a transação base Service.Pet, a tabela base do For each será SERVICEPET e nos permitirá obter os valores necessários para carregar o Grid. A opção 3 não está correta, porque não é necessário o For Each Service, pois com um único For Each com tabela base SERVICEPET já podemos recuperar todos os atributos através de sua tabela estendida. A opção 4 também não está correta.*