GeneXus 17 Junior Analyst Exam

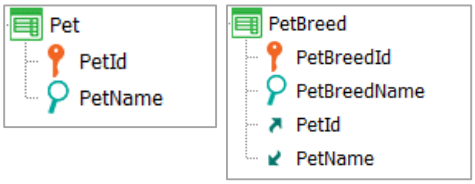
Reality: Pet shop.
About multiple choice questions:

- There is only one correct option.
- In this exam, NO points are deducted for incorrect answers.

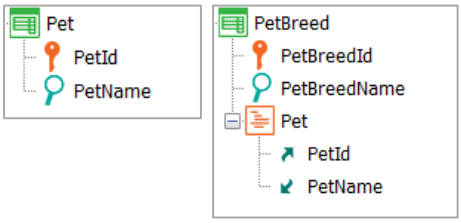
1) There is a GeneXus application for managing a pet shop.

Knowing that a pet (Pet) belongs to a breed (PetBreed), and many pets can belong to the same breed, determine the transaction design you consider correct.

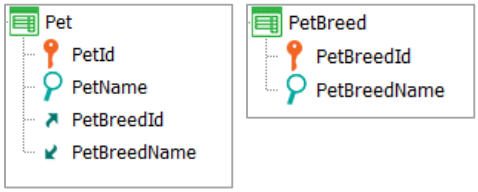
1.1 -



1.2 -



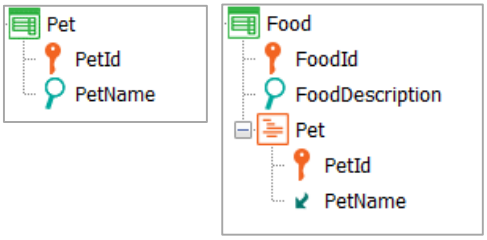
1.3 –



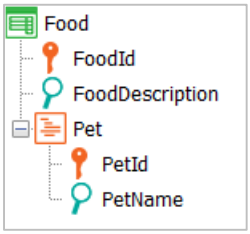
1.4 – None of the above options is correct.

2) Knowing that a pet (Pet) can eat several types of food (Food), and that one type of food can be eaten by several pets, determine the transaction design that you consider correct.

2.1 –



2.2 –



2.3 –

2.4 – None of the above options is correct.

3) Consider the transaction design shown below and determine the option that you consider correct.

3.1 – Every pet (Pet) has a set of specific care guidelines (SpecialCare) that are associated with it and are identified as unique to that pet.

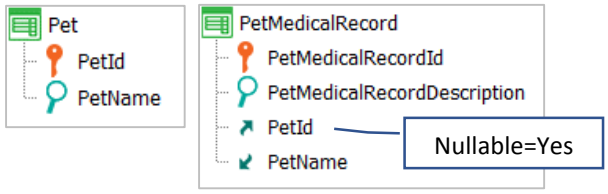
3.2 – Every pet (Pet) has a set of specific care guidelines (SpecialCare) that are associated with it, and the same care is not specific to a single pet, but can be applied to other pets.

3.3 – The design is not valid. It is not possible to create a two-level transaction without defining the second-level entity as a transaction by itself.

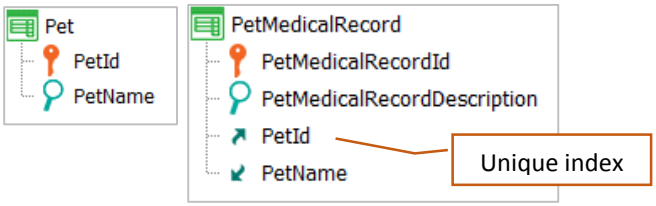
3.4 – None of the above options is correct.

4) Knowing that a pet (Pet) has only one medical record (PetMedicalRecord), and that this medical record is only for that pet, determine the option that you consider correct:

4.1 –



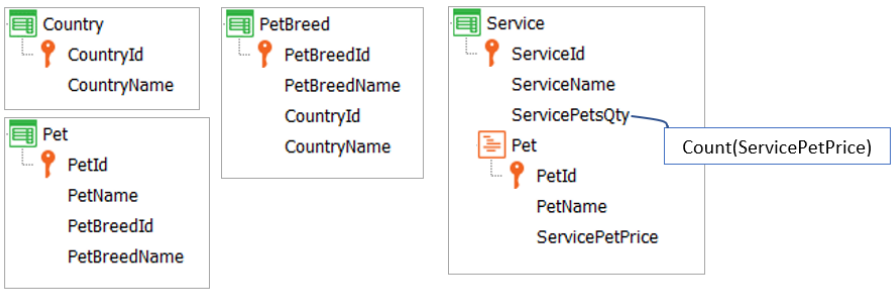
4.2 –



4.3 –

4.4 – None of the above options is correct.

5) Consider the transaction design shown below and determine the physical table structure that will be created by GeneXus.



5.1 -

COUNTRY
CountryId*
CountryName

PETBREED
PetBreedId*
PetBreedName
CountryId

PET
PetId*
PetName
PetBreedId

SERVICE
ServiceId*
PetId*
ServiceName
ServicePetsQty

5.2 –

COUNTRY
CountryId*
CountryName

PETBREED
PetBreedId*
PetBreedName
CountryId

PET
PetId*
PetName
PetBreedId

SERVICE
ServiceId*
ServiceName
ServicePetsQty

SERVICEPET
ServiceId*
PetId*
ServicePetPrice

5.3 –

COUNTRY
CountryId*
CountryName

PETBREED
PetBreedId*
PetBreedName
CountryId

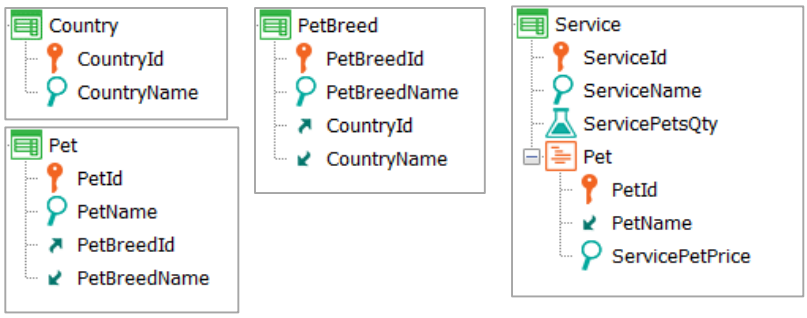
PET
PetId*
PetName
PetBreedId

SERVICE
ServiceId*
ServiceName

SERVICEPET
ServiceId*
PetId*
ServicePetPrice

5.4 – None of the above options is correct.

6) Based on the transaction design displayed, determine the extended table of the SERVICEPET table.



6.1 – SERVICEPET, SERVICE, PET

6.2 – SERVICEPET, SERVICE, PET, PETBREED, COUNTRY

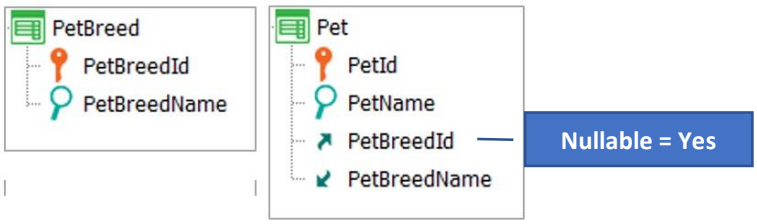
6.3 – SERVICEPET, SERVICE, PET, PETBREED

6.4 – SERVICEPET

7) Although every pet has a breed (PetBreed), sometimes it is not possible to determine that breed, and therefore this information is not considered mandatory when registering a pet.

If a breed (PetBreedId) is specified, this value must be valid.

Based on the transaction design shown below, determine what you consider correct:



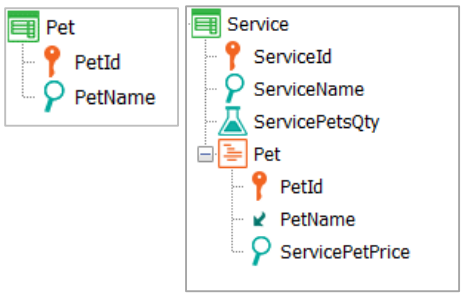
7.1) This implementation doesn't meet the requirement. By indicating that PetBreedId accepts nulls, GeneXus will not perform referential integrity checks. It will be possible to register a pet without a breed, but if a breed is indicated then it will not be checked if it exists as a record in PETBREED.

7.2) The implementation is not correct. A unique index must be defined on the PetBreedId attribute in PET.

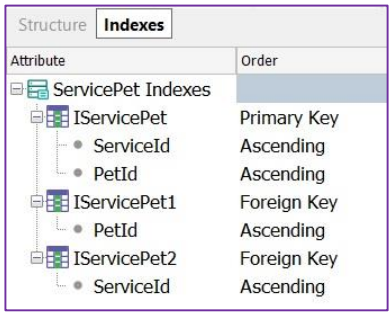
7.3) This implementation is correct and meets the requirement.

7.4) None of the above options is correct.

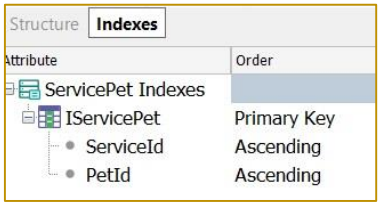
8) Based on the transaction design displayed, determine the indexes automatically created by GeneXus in the SERVICEPET table.



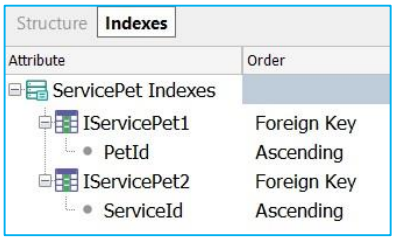
8.1)



8.2)

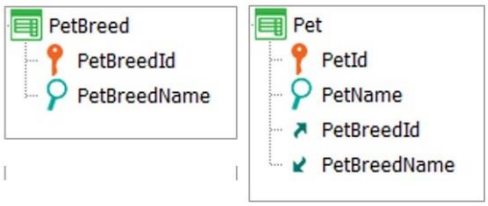


8.3)



8.4) None of the above options is correct.

9) Consider the transaction design shown below, and determine what will happen when trying to delete a breed (PetBreed) using the PetBreed transaction form.



9.1) GeneXus will delete it without making any controls.

9.2) GeneXus will automatically delete all the records in Pet that have PetBreedId as FK first, and then delete the corresponding PetBreed record.

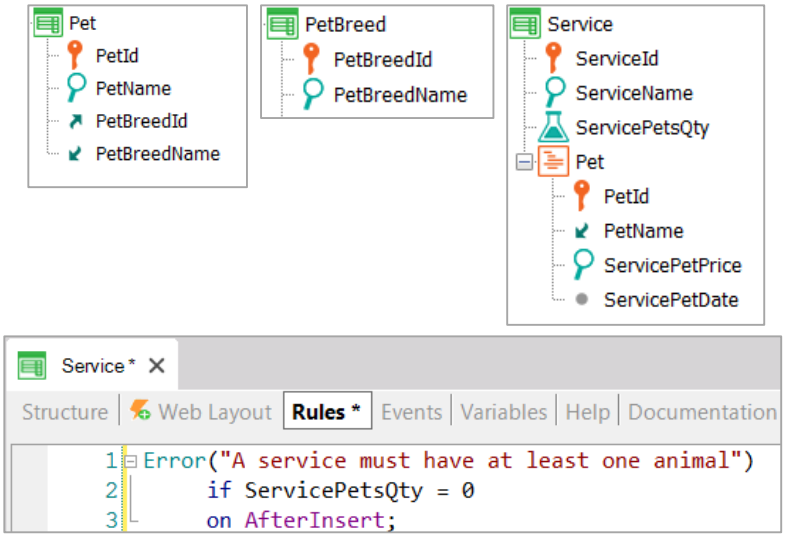
9.3) GeneXus will check that there are no records in Pet that have PetBreedId as FK. If they exist, it will issue a message indicating that related records exist and will not take any action.

9.4) None of the above options is correct.

10) In the following transaction design, the Service transaction has a formula attribute, ServicePetsQty, which counts the number of animals registered for a given service on a given date.

It is necessary to control that a service is never registered without an associated animal. The Error rule shown below is used to perform this control.

Determine what you consider correct:



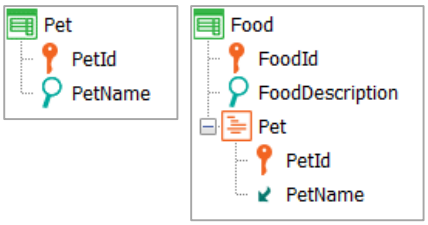
10.1 – The rule doesn't meet the requirement because it will be triggered on the server after the header data (Service) has been saved in the database and before the animals (Pet) start to be saved.

10.2 – The rule doesn't meet the requirement because it will be triggered on the server after the header data (Service) has been saved in the database and immediately after the last animal (Pet) has been saved.

10.3 – The rule doesn't meet the requirement because it will be triggered on the server right before the header data (Service) starts to be saved.

10.4 – The rule meets the requirement because it will be triggered on the client before pressing Confirm.

11) Consider the transactions displayed and determine the order in which the rules declared in the Food transaction will be triggered.



Rules:

a) FoodDetail(FoodId) on AfterComplete;
b) Reservation(FoodId) on AfterInsert;
c) StockControl(FoodId) on AfterLevel level PetId;

11.1 – b), c), a)

11.2 – c), b), a)

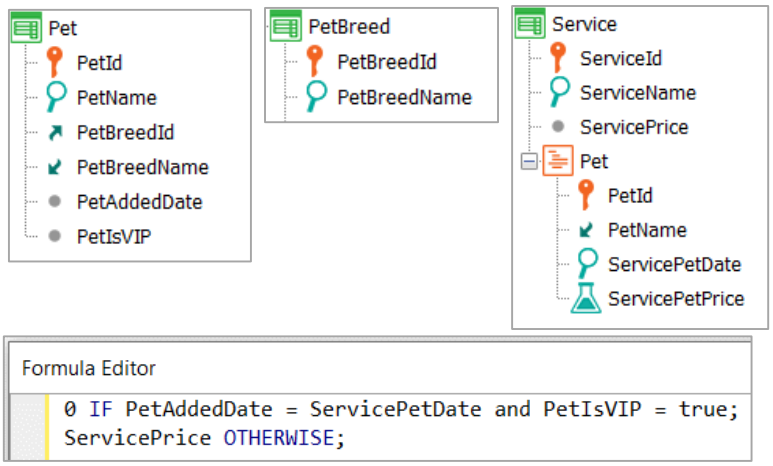
11.3 – c), a), b)

11.4 – The rules are triggered in the order in which they are declared.

12) In the pet store there are VIP pets, i.e. pets that have certain benefits.

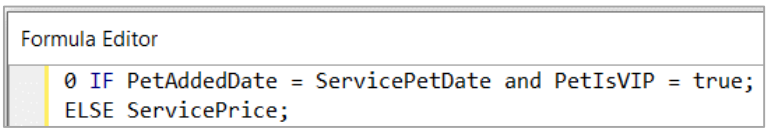
When associating a pet with a certain service, if the pet is a VIP and the day to perform the service matches the day on which the pet was registered in the store, the service will be free of charge. Otherwise, the base price of the service will be applied.

Determine what you consider correct from the calculation associated with the ServicePetPrice attribute.



12.1 – The implementation of the formula is incorrect because it is not possible to use the attributes PetAddedDate and PetIsVIP in it, since they are not found in the structure of the Service transaction (neither in the header nor in the Pet sublevel).

12.2 – The syntax of the formula is incorrect because when the IF structure is used, ELSE should be used to refer to the others. The valid implementation would be as follows:

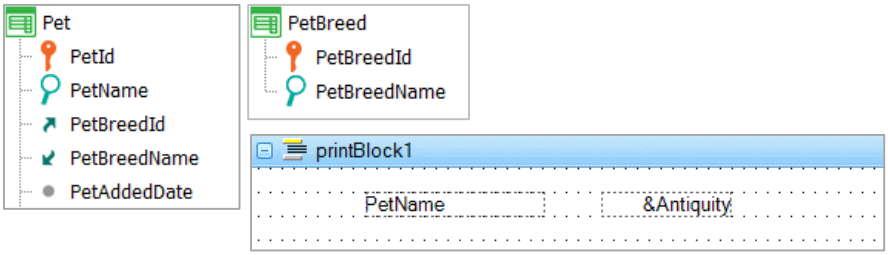


12.3 – The implementation of the formula meets the requirement.

12.4 – None of the above options is correct.

13) A list of all the animals (Pet) in the pet store needs to be displayed, with their name (PetName) and the date on which they were registered.

Look at the following transaction and the procedure Layout. What should be the implementation of the 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 –

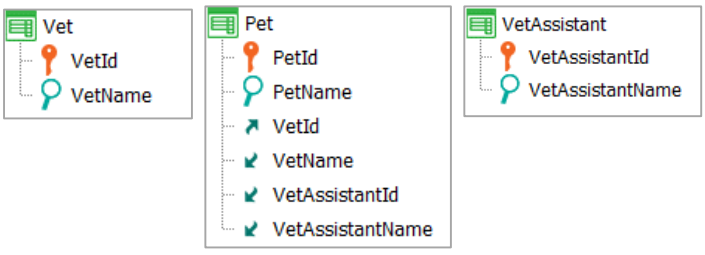
```
For each Pet
      &Antiquity = &Today.Year() - PetAddedDate.Year()
      Print printBlock1
Endfor
```

13.4 – None of the above options is correct.

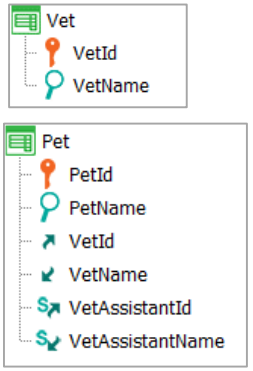
14) Even though every pet (Pet) has a primary veterinarian, a substitute veterinarian needs to be recorded for cases when the primary veterinarian is not available.

Indicate which of the following transaction designs (and of subtype groups if they are included) is adequate to model the above reality.

14.1 –



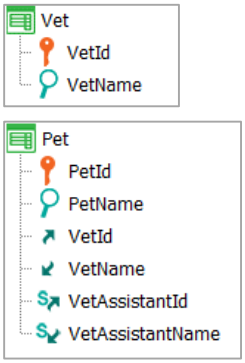
14.2 –



Subtype	Description	Supertype
VetAssistantId		
VetAssistantId	Vet Assistant Id	VetId

Subtype	Description	Supertype
VetAssistantName		
VetAssistantName	Vet Assistant Name	VetName

14.3 –



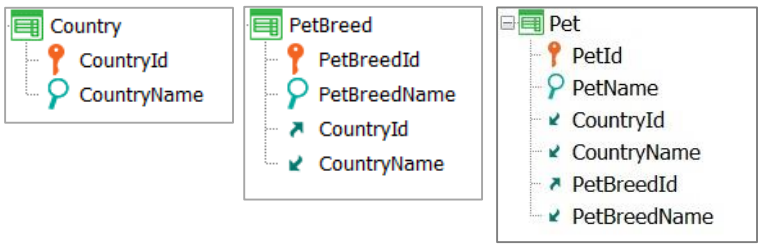
Subtype	Description	Supertype
VetAssistant		
VetAssistantId	Vet Assistant Id	VetId
VetAssistantName	Vet Assistant Name	VetName

14.4 – None of the above options is correct.

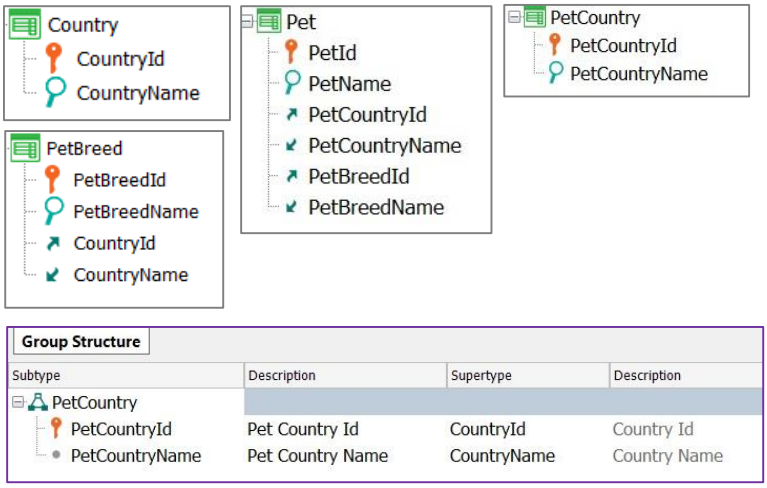
15) Although every breed (PetBreed) is associated with a certain country of origin (Country), the country where the pet (Pet) was obtained (adopted) by its owners should also be recorded.

Determine the implementation option you consider correct:

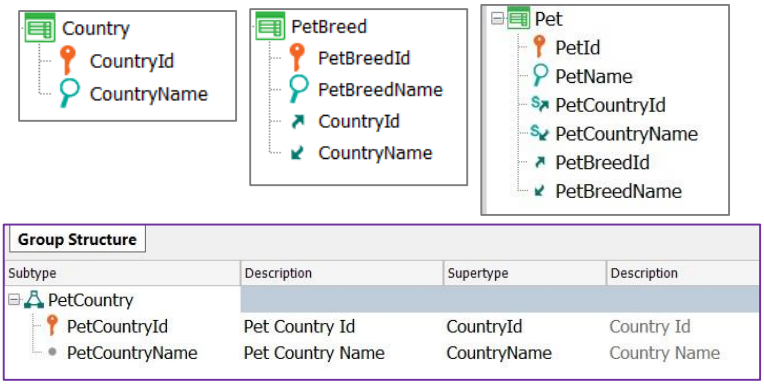
15.1) Simply due to the order in which the attributes are placed in the Pet transaction structure, it will be possible to record the country where the pet was adopted.



15.2) The PetCountry transaction must be defined, and the corresponding group of subtypes must be declared.



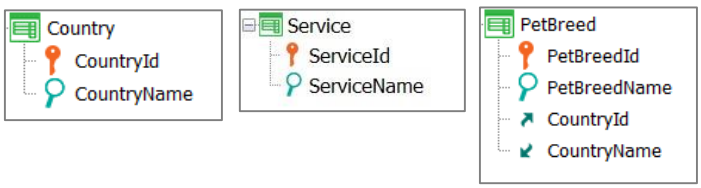
15.3) The attributes and subtype group must be defined as follows:



15.4) None of the above options is correct.

16) Consider the transaction design and navigation list shown below.

What does it implement?



16.1) Cartesian product

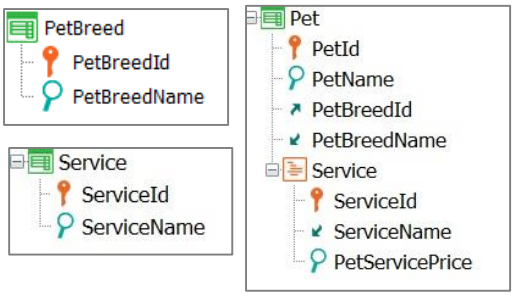
16.2) Control break

16.3) Join

16.4) None of the above options is correct.

17) Consider the transaction design shown below. Complete information (header and lines) about a certain Pet received in a parameter needs to be listed.

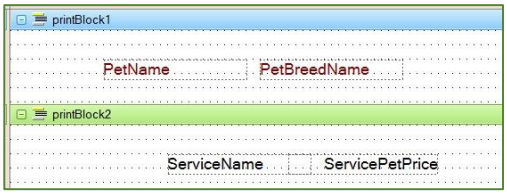
Determine the implementation that you consider correct:



17.1)

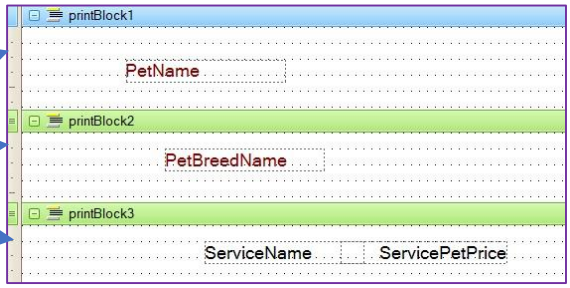
Parm(**in**: PetId);

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



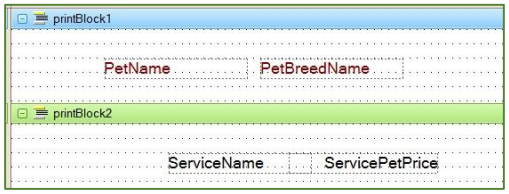
17.2) **Parm**(**in**: &PetId);

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



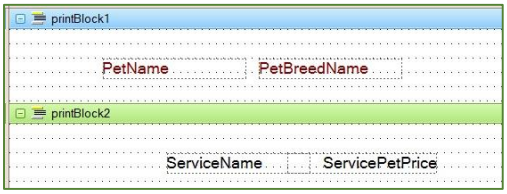
17.3) **Parm**(**in**: &PetId);

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



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

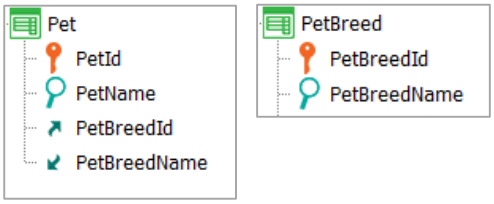
```
For each Pet
   Print printBlock1
   For each Pet.Service
    Print printBlock2
   Endfor
Endfor
```



18) Consider the transaction design shown below. A list has to be defined to show all the breeds (PetBreed) and, for each one of them, the list of pets (Pet) that belong to it.

All the breeds should be listed, regardless of whether they have registered pets of that breed or not.

Select the implementation option you consider correct to meet the request described.

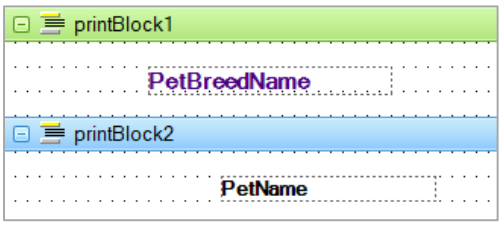


18.1 –

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 –

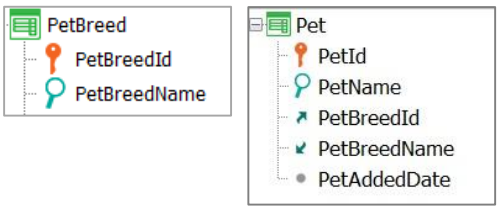
For each Pet
    Print printblock1
    Print printblock2
Endfor

18.4 –

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

19) Consider the transaction design shown below. The pets (Pet) of breed “Beagle” (PetBreedId = 4) and “Cocker” (PetBreedId = 7) that were registered (PetAddedDate) in the year 2020 need to be listed.

Determine if the proposed implementation is correct or not.

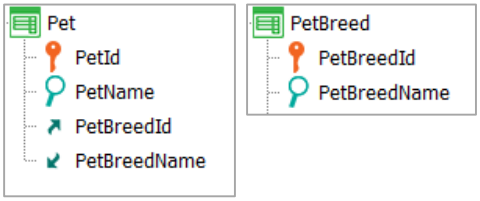


For each Pet
  Where PetBreedId = 4 **or** PetBreedId = 7
  Where PetAddedDate.Year() = 2020
    Print printBlock1
Endfor

☐ True ☐ False

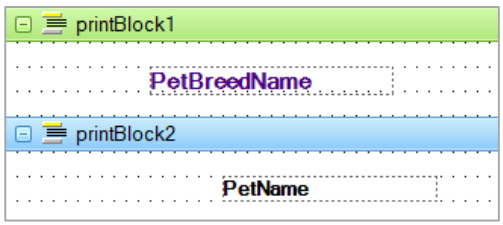
20) Consider the transaction design shown below. A list has to be defined to show all the pets (Pet) grouped by breed (PetBreed).

Only those breeds that have pets registered should be included in the list.



20.1 –

For each PetBreed order PetBreedId
    Print printblock1
    Print printblock2
Endfor



20.2 –

For each PetBreed order PetBreedId

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

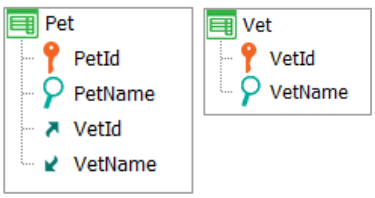
20.4 –

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

21) Consider the transaction design shown below.

A list needs to be defined showing all registered veterinarians (Vet) who have at least one pet in their care. If they don't have any, a text must inform it.

Determine the implementation option you consider correct to meet the requirement described.



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
```

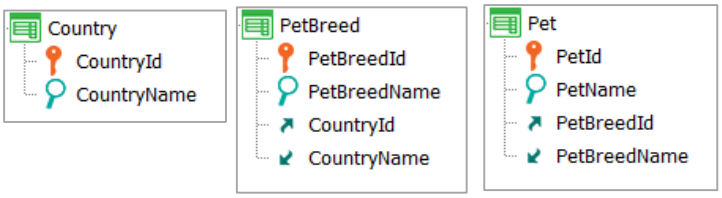
21.3 –

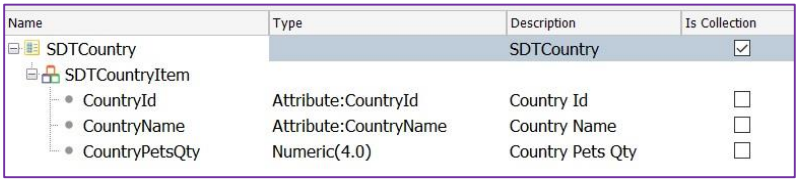
```
For each Vet
     where Count(PetName) >= 1
     Print printBlock1
when none
     Print printBlock2
Endfor
```

21.4 – None of the above options is correct.

22) Consider the transaction design and the definition of the SDTCountry structured data type that are shown below. A Data Provider needs to be designed to load a collection of countries (Country), each with the number of pets (Pet) of breeds (PetBreed) of that country.

Select the implementation option you consider correct.





Name	Type	Description	Is Collection
SDTCountry		SDTCountry	☑
SDTCountryItem			
CountryId	Attribute:CountryId	Country Id	☐
CountryName	Attribute:CountryName	Country Name	☐
CountryPetsQty	Numeric(4.0)	Country Pets Qty	☐

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)
     }
}
```

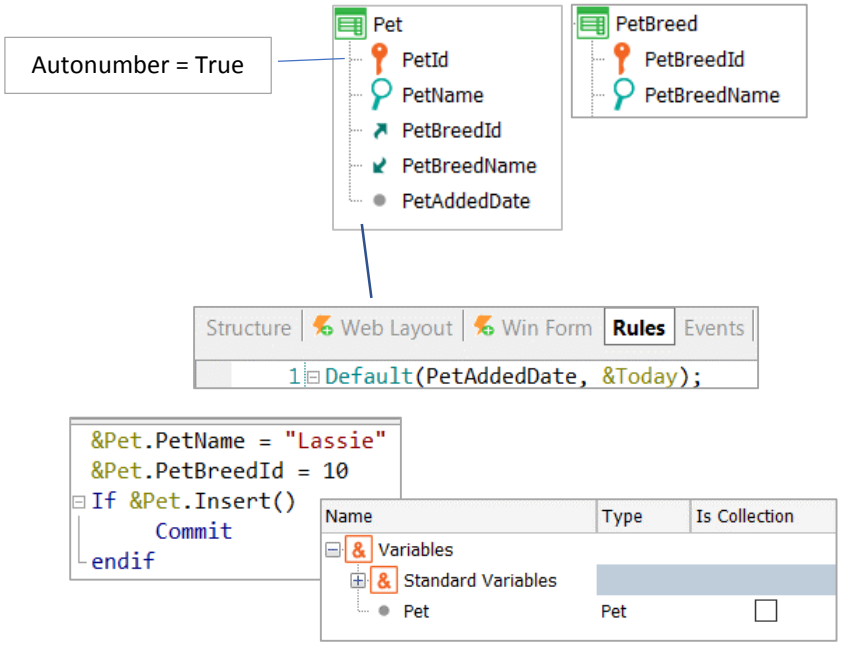
Output	
Infer Structure	No
Output	**SDTCountry**
Collection	**True**
Collection Name	**Countries**

22.4) None of the above options is correct.

23) Consider the transaction design shown below. The Pet transaction has been configured as a Business Component and the PetId attribute is autonumbered.

A new pet (Pet) called “Lassie” has to be added using a Business Component of Pet.

A procedure has been programmed with the following code. Select the correct option.



23.1 – The pet will only be inserted if breed 10 exists in the PetBreed table. Otherwise, the referential integrity will fail and it will not be inserted. If it is inserted, it will have an empty entry date, since no entry date was specified in the code.

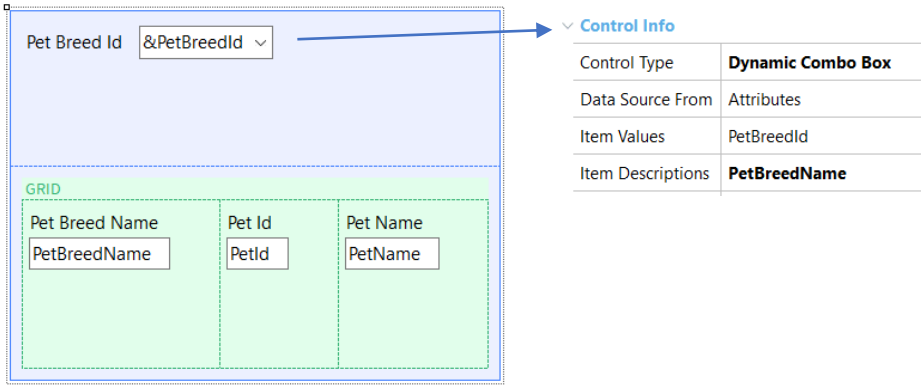
23.2 – The pet will only be inserted if breed 10 exists in the PetBreed table. Otherwise, the referential integrity will fail and it will not be inserted. If it is inserted, its entry date will be today's date.

23.3 – The pet will always be inserted, even if there is no breed with identifier 10 in the PetBreed table, because Business Components do not control the referential integrity. It will have an empty entry date, since no entry date was specified in the code.

23.4 – The pet will always be inserted, even if there is no breed with identifier 10 in the PetBreed table, because Business Components do not control the referential integrity. Its entry date will be today's date.

24) Consider the transaction design and the Web Panel layout shown below. A Web Panel has to be designed to show the names of the pets (PetName) of a given breed selected by the user.

Determine the option that you consider correct.



24.1 – The Load event must be coded as shown:

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

24.2 – The Load event must be coded as shown:

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

24.3 – The Start event must be modified as shown:

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

24.4 – The following condition must be stated in the Grid:

```
Grid1's Conditions
    PetBreedId = &PetBreedId;
```

25) Consider the transaction design and the Web Panel layout shown below. A Web Panel has to be designed to show all the breeds (PetBreed), each one with their corresponding number of pets registered. If more than 10 pets are registered, the comment "Many pets" will be displayed. Otherwise, "Few pets" will be displayed.

Select the implementation option you consider correct.

25.1 –

```
Event Load
    For each PetBreed
        &Quantity = Count(PetName)
        If &Quantilty >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
```

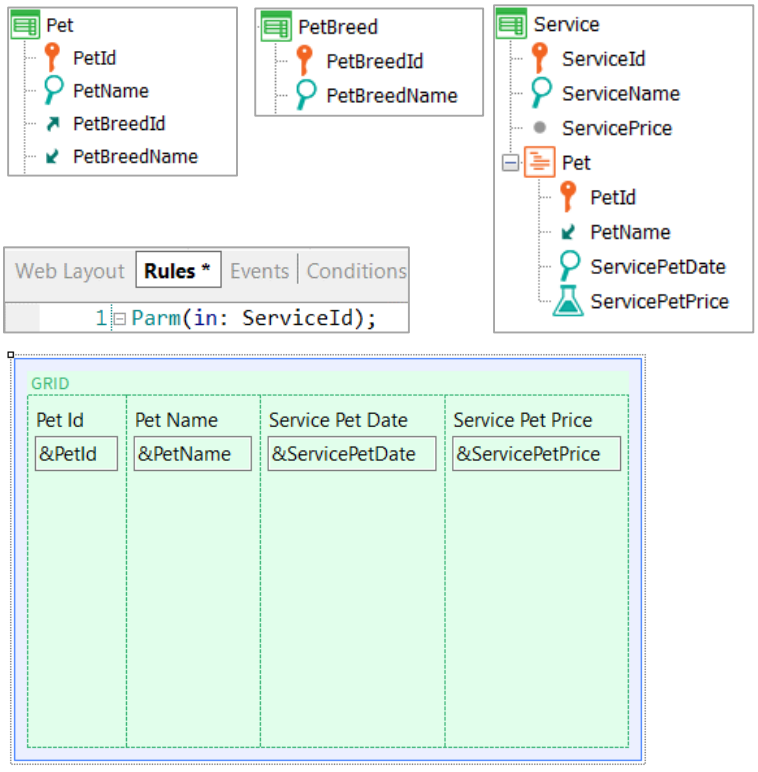
25.3 –

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

25.4 – None of the above options is correct.

26) In a Web Panel, the pets associated with a certain service received in a parameter are to be displayed.

Based on the detailed transaction and Web Panel design (what is not shown, such as properties, has not been modified except for the variables, which are all ReadOnly), determine the option you consider correct.



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 – None of the above options is correct.

ANSWERS

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