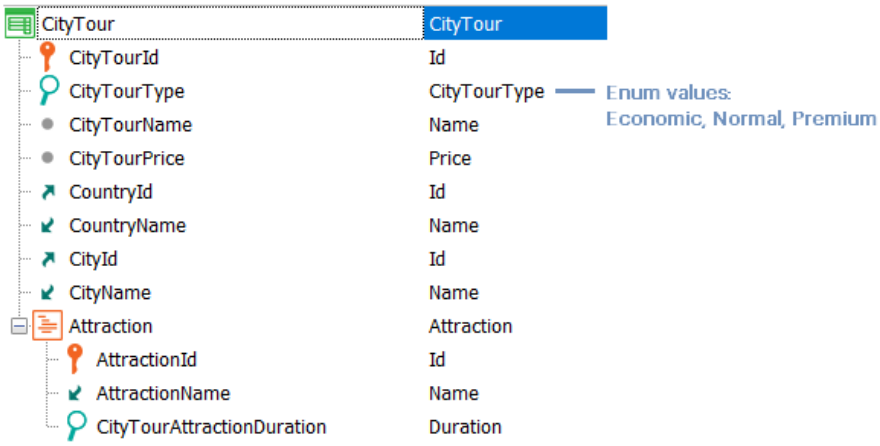
Exame Analista GeneXus 18

1. Está sendo desenvolvida uma aplicação para uma agência de viagens. É desejado registrar os diferentes City tours que são oferecidos para passear por diferentes atrações turísticas da cidade. É apresentada a estrutura da transação CityTour unicamente. É desejado obter o seguinte comportamento:

   - Que seja avisado o quanto antes que foi deixado vazio o campo CityTourName (em qualquer um dos modos).
   - Se estiver sendo inserido um city tour e for selecionado o tipo Economic, é desejado proibir a inserção de um preço que esteja acima da faixa de preços do tipo econômico. A verificação será feita por uma proc CheckPrice cuja lógica não importa para este exercício. Mas é desejado que o controle não seja realizado no cliente, mas apenas no servidor, imediatamente antes que o cabeçalho seja gravado na tabela CityTour.

   Você deverá determinar qual das seguintes opções é a que implementa corretamente as regras de acordo com estes requisitos.



   a.

```
Error("Price more expensive than expected")
     if not &ok and CityTourType=CityTourType.Economic and Insert;

&ok = CheckPrice(CityTourPrice)
     if CityTourType=CityTourType.Economic and Insert;

Msg("City Tour Name is empty")
     if CityTourName.IsEmpty();
```

b.

```
Error("Price more expensive than expected")
      if not &ok and CityTourType= CityTourType.Economic
      on BeforeInsert;
&ok= CheckPrice(CityTourPrice)
      if CityTourType=CityTourType.Economic
      on BeforeInsert;
Msg("City Tour Name is empty")
      if CityTourName.IsEmpty();
```

c.

```
Msg("City Tour Name is empty")
     if CityTourName.IsEmpty();

&ok = CheckPrice(CityTourPrice)
     if CityTourType=CityTourType.Economic and Insert
     on BeforeComplete;

Error("Price more expensive than expected")
     if not &ok and CityTourType=CityTourType.Economic and Insert
     on BeforeComplete;
```

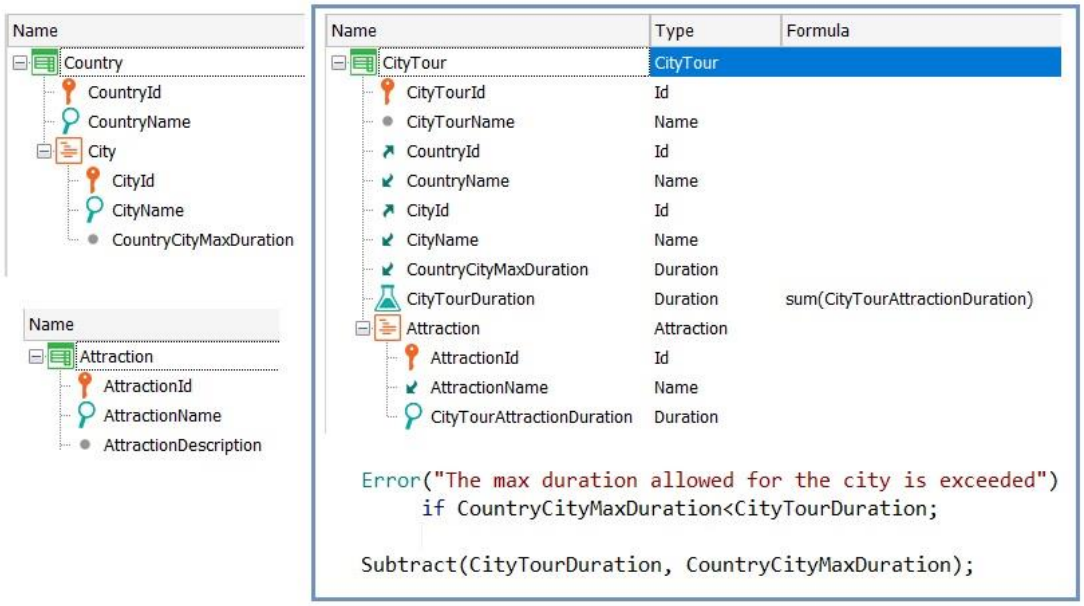
d. Nenhuma das anteriores

2. No contexto de uma aplicação para uma agência de viagens, existe uma transação para registrar os tours por uma cidade que a agência oferece aos seus clientes. São registradas as atrações turísticas que serão visitadas, juntamente com o tempo (duration) reservado para a visita a cada atração.

   Supondo que cada agência de viagens tenha uma quantidade máxima de tempo permitido para todos os seus city tours pela cidade, foi adicionado o atributo CountryCityMaxDuration à transação Country.

   Cada vez que é inserido um city tour para a cidade, esse atributo deverá ser atualizado de acordo com a duração total do city tour.

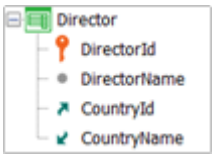
   Para isso, foram especificadas no CityTour as duas regras que mostramos na imagem. Queremos saber se com elas cumprimos ou não o requisito. Das opções que damos a seguir, escolha a correta.



   a. Não pode ser utilizado um atributo fórmula para ser subtraído, somado ou atualizado de um atributo da tabela estendida desse mesmo nível.
   b. A regra error está mal condicionada, pois como deve utilizar o atributo CountryCityMaxDuration, que é atualizado pela regra subtract, sempre será disparada depois desta. Portanto, a condição deveria ser "If CountryCityMaxDuration < 0".
   c. Falta condicionar as regras ao evento "On BeforeInsert, BeforeUpdate,BeforeDelete" para que seja disparada apenas no server e antes de atualizar o registro. Caso contrário, será disparada duas vezes e atualizará duas vezes.
   d. Nenhuma das anteriores.

3. Um filme (Movie) pode ser dirigido por um ou vários diretores (Director). É necessário registrar para os filmes o país onde foram realizados, e indicar também o país de cada diretor.

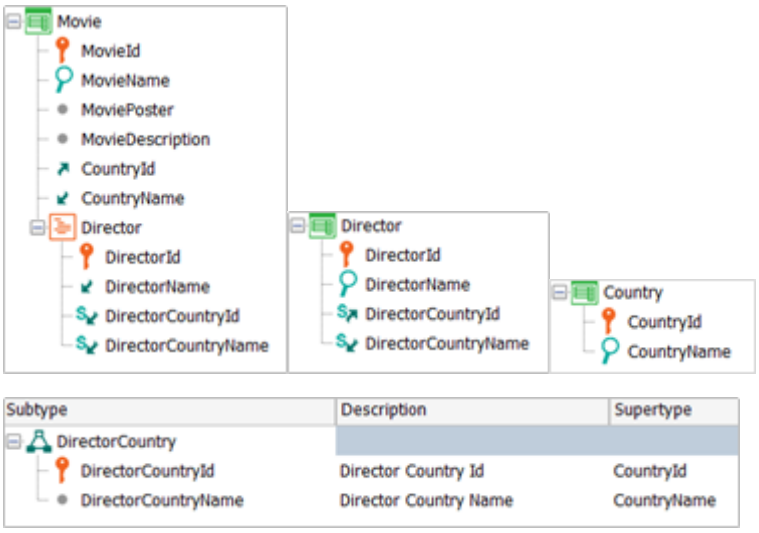
A equipe de desenvolvimento definiu a transação Director da seguinte forma:



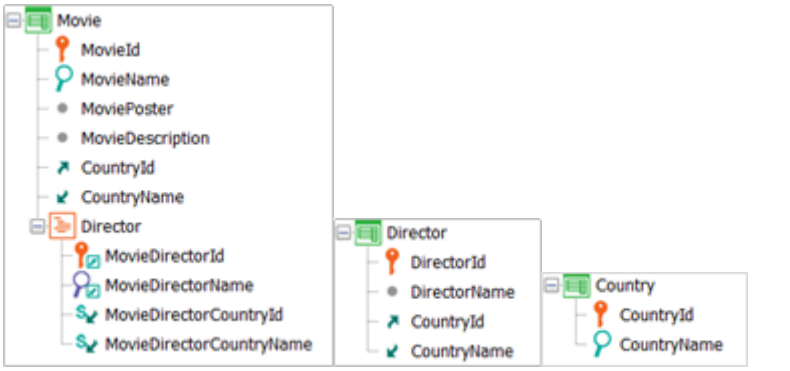
Antes de continuar com as outras transações do modelo, a equipe também criou todos os programas para resolver os requisitos solicitados pela empresa que administrará o Cinema no que se refere aos diretores (por exemplo: listas dos diretores em PDF, visualização dos diretores filtrando por seus país a partir de um Web Panel, procedimentos que realizam diferentes verificações com seus dados, etc).

Analise as opções 1 e 2 e indique o que considere correto.

1)



2)



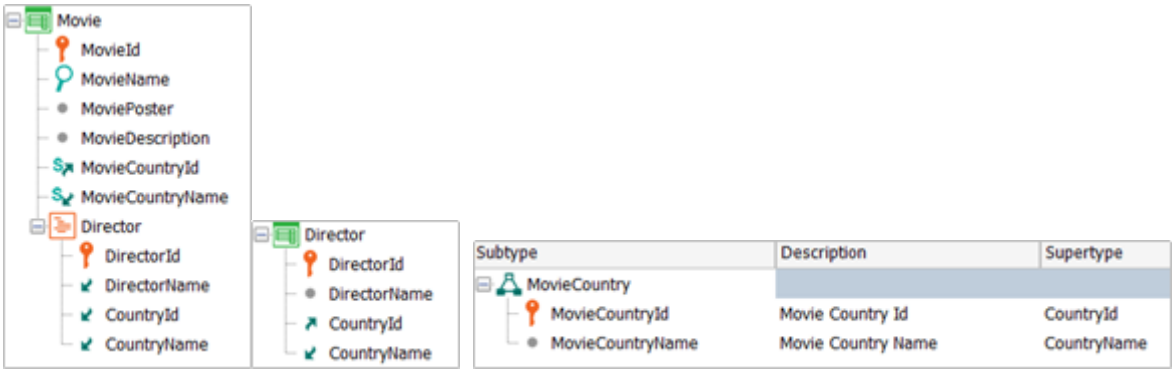
Subtype	Description	Supertype
MovieDirector		
MovieDirectorId	Movie Director Id	DirectorId
MovieDirectorName	Movie Director Name	DirectorName
MovieDirectorCountryId	Movie Director Country Id	CountryId
MovieDirectorCountryName	Movie Director Country Name	CountryName

a) Ambas as opções são equivalentes e resolvem o que é solicitado sem vantagens ou desvantagens uma em relação à outra.

b) Embora ambas as opções resolvam o que é solicitado, a opção 1 é mais adequada neste caso, pois ao ficar o grupo de subtipos com menos elementos, são criadas menos referências, e ao alterar os nomes dos atributos CountryId e CountryName na transação Director, automaticamente são atualizados esses atributos nos objetos onde estão sendo utilizados.

c) Embora ambas as opções resolvam o que foi solicitado, a opção 2 é mais adequada neste caso, pois com ela não há necessidade de ir a cada programa já criado para modificar os atributos CountryId e CountryName, pois eles não mudam.

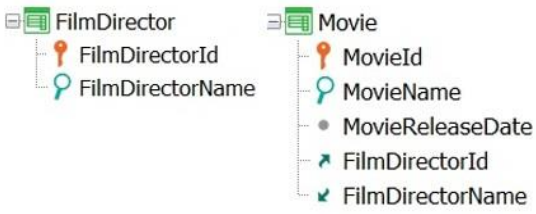
d) Nenhuma das duas opções resolve o que é solicitado. A única solução seria a seguinte:



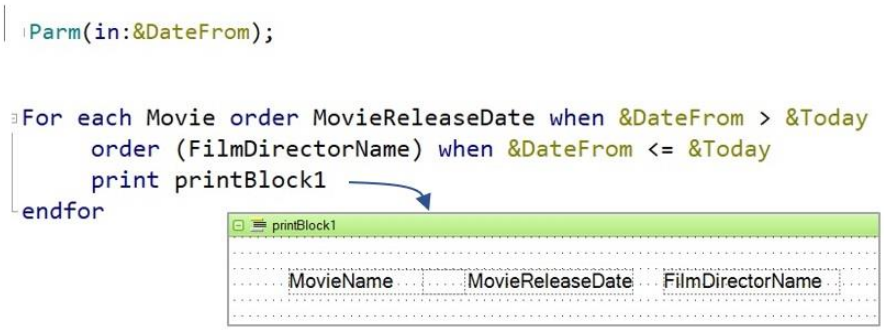
Subtype	Description	Supertype
MovieCountry		
MovieCountryId	Movie Country Id	CountryId
MovieCountryName	Movie Country Name	CountryName

4. Considere o desenho de transações apresentado. É necessária uma lista que receba uma data por parâmetro, e mostre os filmes (Movie) ordenados por data de lançamento (MovieReleaseDate), no caso em que esse parâmetro seja uma data posterior à atual. Caso contrário, a lista será vista ordenada de forma decrescente pelo nome do diretor (FilmDirectorName).

Determine o que considere correto:



a) O requisito é resolvido indicando as ordens desejadas condicionadas às respectivas cláusulas When. Sempre será aplicada a ordem que confirme as condições:



b) Deve ser indicada uma das ordens condicionadas e, no caso de não ser cumprida a condição, é indicada a outra utilizando a cláusula When none do For each:

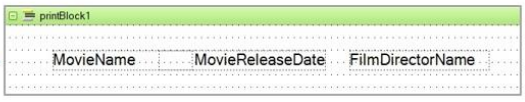
```
Parm(in:&DateFrom);

For each Movie order MovieReleaseDate when &DateFrom > &Today
When none
     For each Movie order (FilmDirectorName)
     print printBlock1
     endfor
endfor
```

c) O requisito é resolvido indicando as ordens desejadas condicionadas com a cláusula If - Else conforme mostrado:

```
Parm(in:&DateFrom);

For each Movie order MovieReleaseDate if &DateFrom > &Today
     order (FilmDirectorName) else

     print printBlock1
endfor
```

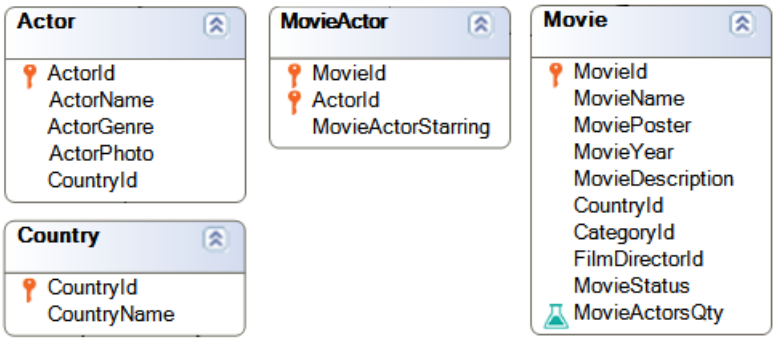
d) Não é possível resolver o requisito solicitado declarando um único For each e indicando várias ordens condicionais. Para resolvê-lo, deve ser implementado cada For each de acordo com o valor do parâmetro recebido:

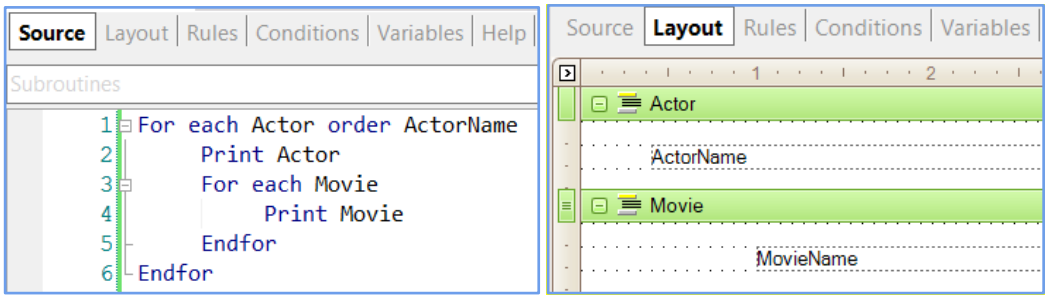
```
Parm(in:&DateFrom);

If &DateFrom > &Today
     For each Movie order MovieReleaseDate
          print printBlock1
     endfor
Else
     For each Movie order (FilmDirectorName)
          print printBlock1
     Endfor
Endif
```



5. Observe as tabelas geradas e o procedimento apresentados a seguir.

A partir desta implementação, das seguintes afirmações selecione a correta.





a. Serão impressos todos os atores. E para cada ator que está registrado em pelo menos um filme, todos os filmes que tenham inserido para esse ator.

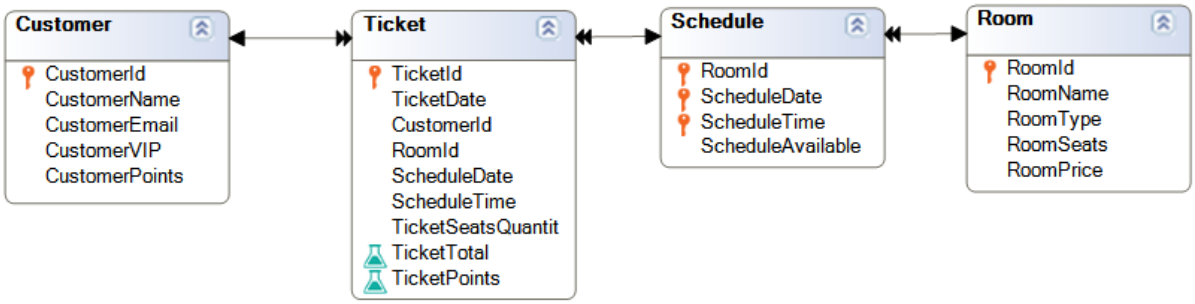
b. Serão impressos todos os atores. E para cada ator todos os filmes que foram registrados.

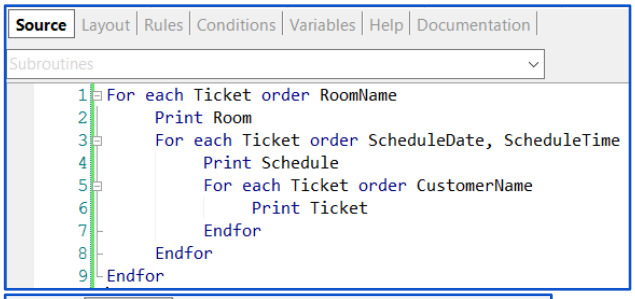
c. Serão impressos todos os atores. E para cada ator que está registrado em pelo menos um filme, todos os filmes que estão registrados.

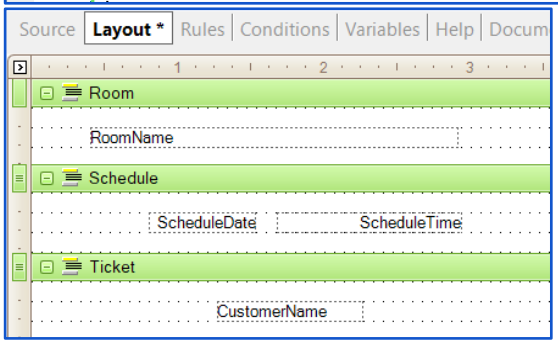
d. Serão impressos todos os atores. E para cada ator que está registrado em pelo menos um filme, serão impressos os filmes cujo país seja igual ao do ator.

6. Observe o diagrama de tabelas e o procedimento apresentados a seguir.

A partir desta implementação, selecione a opção correta.







a. Serão mostradas em tela todas as salas que estiverem presentes em pelo menos um ticket. Para cada uma delas, todas as apresentações inseridas para essa sala, que fazem parte de algum ticket, agrupadas por data (ScheduleDate) e hora (ScheduleTime) da apresentação.

   E para cada uma dessas apresentações, todos os clientes que compraram algum ticket dessa apresentação, agrupados por nome

b. Serão mostradas em tela todas as salas que estiverem presentes em pelo menos um ticket. Para cada uma delas, todas as apresentações inseridas para essa sala, que fazem parte de algum ticket, agrupadas por data (ScheduleDate) e hora (ScheduleTime) da apresentação.

   E para cada uma dessas apresentações, todos os clientes que compraram algum ticket dessa apresentação, ordenados por nome

c. Serão mostradas em tela todas as salas que estiverem presentes em pelo menos um ticket.
Para cada sala, serão mostradas todas as apresentações que foram inseridas no sistema, ordenadas por data (ScheduleDate) e hora (ScheduleTime) da apresentação (serão repetidas se houver apresentações com mesma data e hora). E para cada uma dessas apresentações, todos os clientes, ordenados por nome.

d. Nenhuma das anteriores.

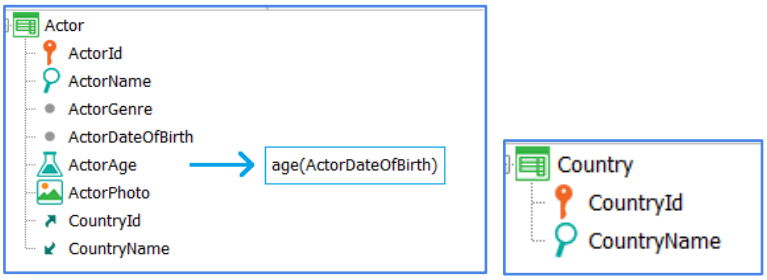
7. Na aplicação para um complexo de cinemas, temos a seguinte implementação.

Observe as transações e o procedimento apresentados.

Interessa listar todos os países, **sem repetir, que tenham atores registrados,** cada um com a média de idade de seus atores.

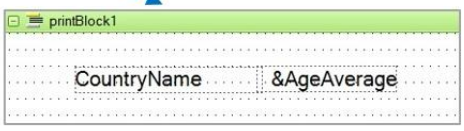
Determine a opção que considere correta.

Nota: Uma das funcionalidades da função Age() é calcular a idade a partir de uma data de nascimento.



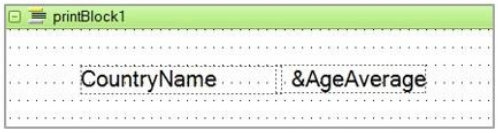
a)

```
For each Country
     &AgeAverage = Average(ActorAge)
     print printBlock1
endfor
```



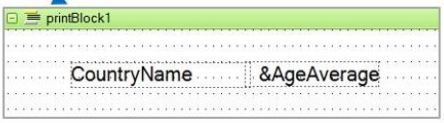
b)

```
For each Actor
     Unique CountryId
     For each Country
          &AgeAverage = Average(ActorAge)
          print printBlock1
     EndFor
endfor
```



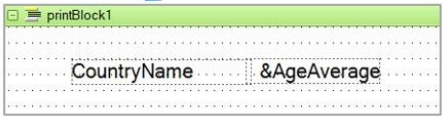
c)

```
For each Actor
     Unique CountryId
     &AgeAverage = Average(ActorAge)
     print printBlock1
endfor
```

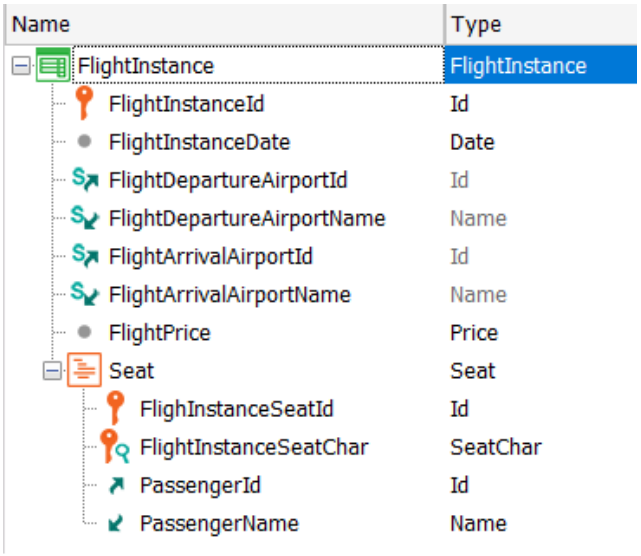


d)

```
For each Actor order CountryId
     &AgeAverage = Average(ActorAge)
     For each Actor
          print printBlock1
     Endfor
endfor
```



8. Está sendo desenvolvida uma aplicação simplificada para uma agência de viagens. Existe a transação FlightInstance que registra os voos. Para cada um são registrados, além da data do voo, seu aeroporto de partida e aeroporto de chegada (grupos de subtipos), seu preço e os passageiros registrados, com seus assentos.



Queremos trocar um passageiro por outro em todos os voos posteriores à data de hoje. Para isso, será programado um procedimento que receberá por parâmetros o passageiro a ser modificado, &OldPassengerId, e o passageiro que o substituirá, &NewPassengerId.

No procedimento temos duas variáveis de tipo business component:

As opções a seguir são possíveis implementações do requisito. Indique a correta.

a.

```
For each FlightInstance.Seat
      where FlightInstanceDate > &Today
      where PassengerId = &oldPassengerId

      &flightInstance.Load(FlightInstanceId)
      &flightSeat = &flightInstance.Seat.GetByKey(PassengerId)
      &flightSeat.PassengerId = &newPassengerId

      if &flightInstance.Update()
            Commit
      endif
endfor
```

b.

```
for each FlightInstance.Seat
      where FlightInstanceDate > &Today
      where PassengerId= &oldPassengerId

      &FlightInstance.Load(FlightInstanceId)
      &FlightSeat= &FlightInstance.Seat.GetByKey(FlightInstanceSeatId, FlightInstanceSeatChar)
      &FlightSeat.PassengerId= &newPassengerId

      if &FlightSeat.Update()
            commit
      endif
endfor
```

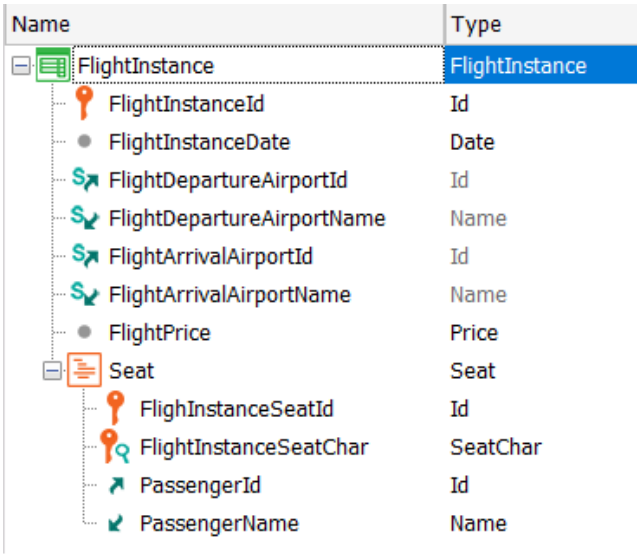
c.

```
for each FlightInstance.Seat
      where FlightInstanceDate > &Today
      where PassengerId= &oldPassengerId

      &FlightInstance.Load(FlightInstanceId)
      &FlightSeat= &FlightInstance.Seat.GetByKey(FlightInstanceSeatId, FlightInstanceSeatChar)
      &FlightSeat.PassengerId= &newPassengerId
      &FlightInstance.Seat.Replace(&FlightSeat)

      if &FlightInstance.Update()
            commit
      endif
endfor
```

d.

```
for each FlightInstance.Seat
      where FlightInstanceDate > &Today
      where PassengerId= &oldPassengerId

      &FlightInstance.Load(FlightInstanceId)
      &FlightSeat= &FlightInstance.Seat.GetByKey(FlightInstanceSeatId, FlightInstanceSeatChar)
      &FlightSeat.PassengerId= &newPassengerId
      &FlightInstance.Seat.Replace(&FlightSeat)

      if &FlightInstance.Update()
            commit
      endif
endfor
```

9. Existe a transação FlightInstance que registra os voos. Para cada um são registrados, além da data do voo, seu aeroporto de partida e aeroporto de chegada (grupos de subtipos), seu preço e os passageiros registrados, com seus assentos.



É desejado remover de um voo específico (o 3546) os assentos 1A e 1F, por meio de uma variável &flight Business Component de FlightInstance.

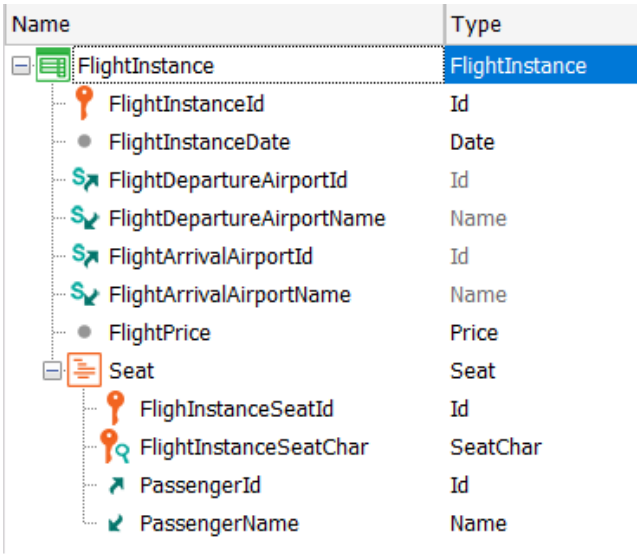
Para isso, foi escrito o seguinte código:

```
&flight.Load(3546)
&flight.Seat.RemoveByKey(1,A)
&flight.Seat.RemoveByKey(1,F)
&flight.Delete()
Commit
```

Indique a opção correta entre as seguintes:

a. O código anterior está correto e faz o que queremos
b. O código anterior está incorreto, pois embora serão eliminadas as duas linhas da coleção Seat no BC, como o método que é executado é o Delete, será eliminado tudo, cabeçalho e linhas. Teria que ter sido escrito na 4ª linha &flight.Update() para estar correto.
c. O código anterior está incorreto. Seria correto se eliminássemos a 4ª linha, ou seja, &flight.Delete(), deixando o resto como está.
d. O código anterior está incorreto por outras razões.

10. Existe a transação FlightInstance que registra os voos. Para cada um são registrados, além da data do voo, seu aeroporto de partida e aeroporto de chegada (grupos de subtipos), seu preço e os passageiros registrados, com seus assentos.



Queremos alterar o assento de um passageiro determinado, o 10, em um voo determinado, o 5, e modificar o preço desse voo, aumentando em 10% seu valor.

Vamos imaginar que {FlightInstanceId, PassengerId} é chave candidata de FlightInstanceSeat, para que um mesmo passageiro só possa estar em um assento de um mesmo voo. Para alterar o assento do passageiro, excluiremos o **registro** de assento do passageiro para esse voo e o inseriremos novamente, com tudo igual e o assento alterado.

Suponhamos que temos o procedimento GetEmptySeat, ao qual, passando o id do voo, retorna um SeatId e um SeatChar não ocupados no voo, para ser possível reatribuí-los ao passageiro que se deseja mover de lugar.

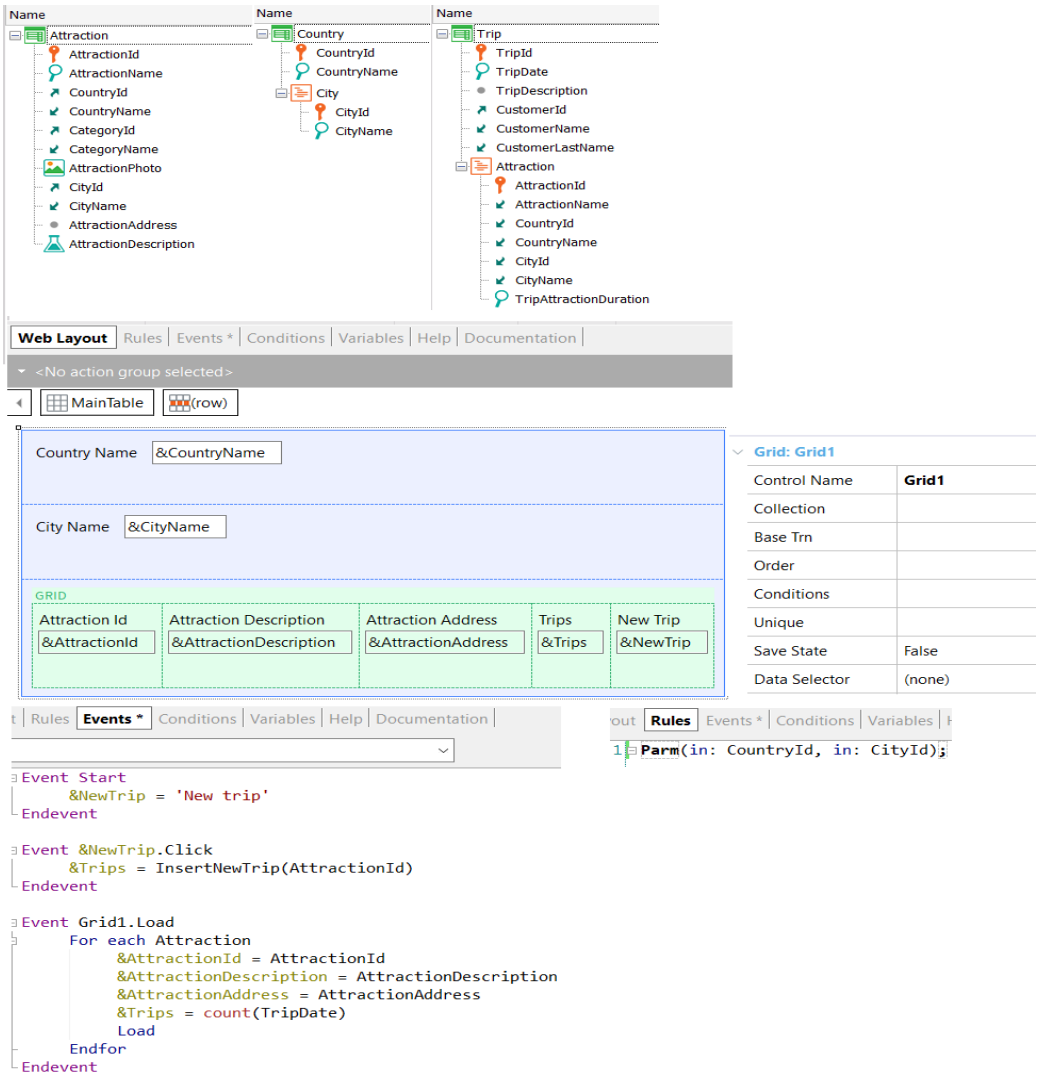
Para implementar o requisito, foi programado o seguinte código no evento de um Web Panel, onde &flight é do tipo de dados BC FlightInstance e &flightSeat do tipo de dados BC FlightInstance.Seat.

```
&flight.Load(5)
&flight.FlightPrice = &flight.FlightPrice * 1.1

GetEmptySeat(5,&seatId, &seatChar)
&flightSeat = new()
&flightSeat.FlightInstanceSeatId = &seatId
&flightSeat.FlightInstanceSeatChar = &seatChar
&flightSeat.PassengerId = 10

For each FlightInstance.Seat
Where FlightIntanceId = 5 and PassengerId = 10
     &flight.Seat.RemoveByKey(FlightInstanceSeatId, FlightInstanceSeatChar)
Endfor
&flight.Seat.Add(&flightSeat)

If &flight.Update()
     Commit
Endif
```

Indique qual das seguintes opções está correta.

a. A implementação está correta.
b. Está implementado errado porque está errada a inserção da linha.
c. Está implementado errado porque está errada a remoção da linha.
d. Nenhuma das anteriores.

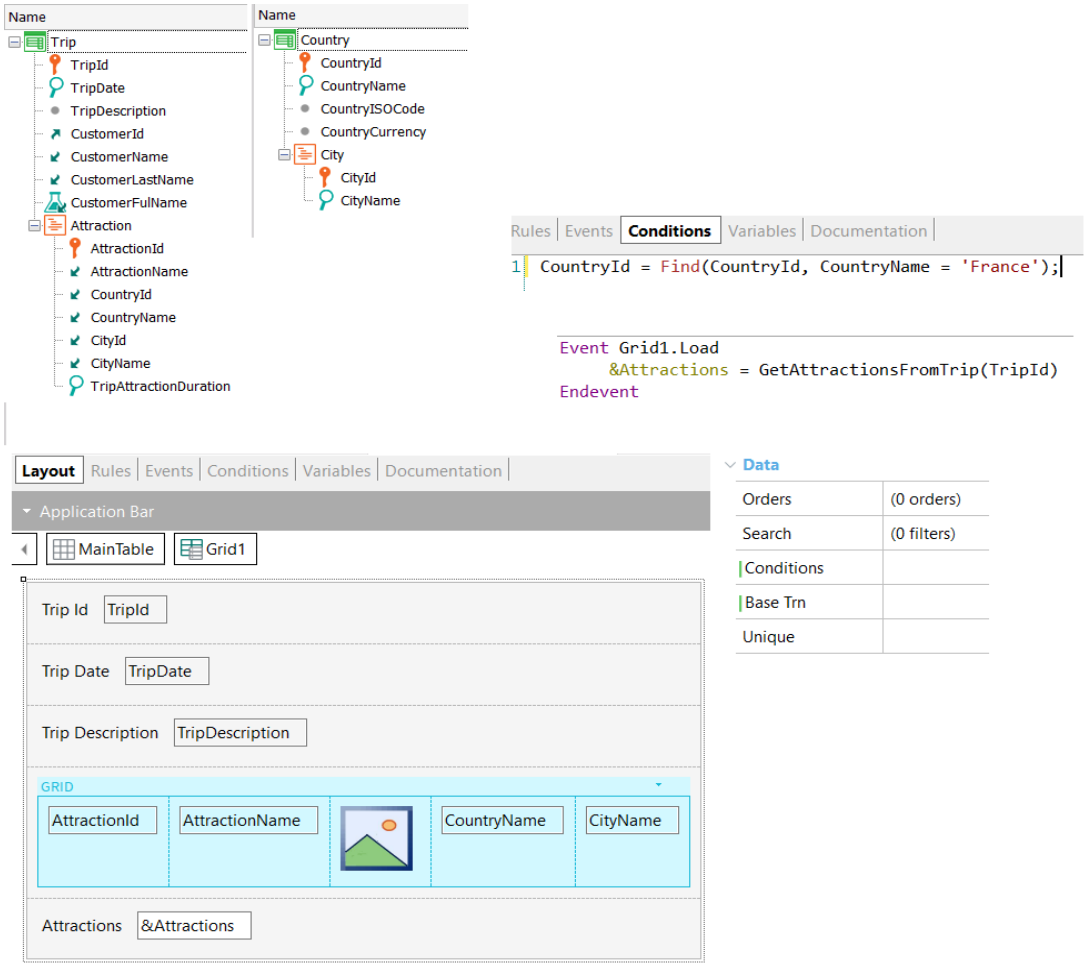
11. No contexto de uma aplicação para uma agência de viagens, com as transações que mostramos na imagem, é desejado ver as atrações de uma cidade de um determinado país, com as viagens que foram feitas para cada atração. Além disso, deve ser dada a possibilidade de criar uma viagem nova para visitar uma atração em particular.
Foi implementado o seguinte Web Panel. Determine qual é sua tabela base.



a) ATTRACTION
b) COUNTRYCITY
c) COUNTRY
d) TRIPATTRACTION
e) O web panel não possui tabela base

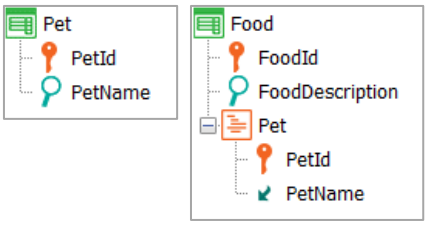
12. No contexto de uma aplicação para uma agência de viagens, é implementado o objeto panel apresentado. São adicionadas também as transações envolvidas. O que não é mencionado é porque não é modificado em relação ao valor default.

Selecione a opção que considere correta.



a) A tabela base do objeto panel é TRIPATTRACTION
b) A tabela base da parte fixa é TRIP e a do grid é ATTRACTION
c) A tabela base da parte fixa é TRIP e a do grid é TRIPATTRACTION
d) A tabela base da parte fixa é TRIPATTRACTION e a do grid é TRIPATTRACTION

13. Considere as transações apresentadas e determine a ordem em que serão disparadas as regras declaradas na transação Food.



Rules:

1. FoodDetail(FoodId) on AfterComplete;
2. Reservation(FoodId) on AfterInsert;
3. StockControl(FoodId) on AfterLevel level PetId;

a) 2), 3), 1)

b) 3), 2), 1)

c) 3), 1), 2)

d) As regras são disparadas na ordem em que são declaradas.

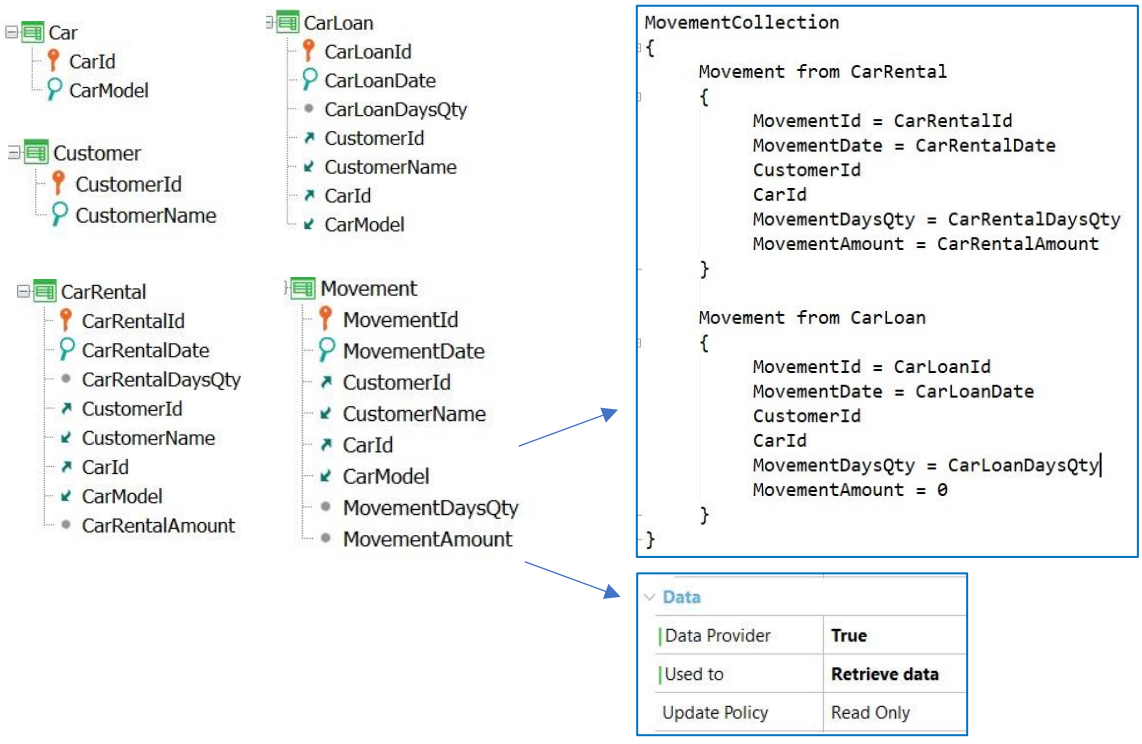
14. Está sendo desenhada uma aplicação para uma empresa de Aluguel de veículos.

Ela realiza aluguéis e empréstimos de veículos aos seus clientes. É necessário unificar a informação sobre aluguéis e empréstimos de veículos, para finalmente poder listar essa informação ordenada por data.

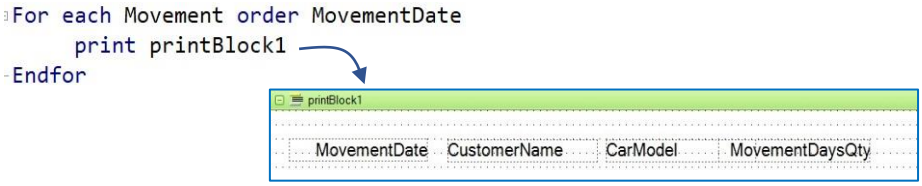
A empresa precisa registrar também promoções (Promotion) dos referidos aluguéis e empréstimos de veículos

Nota: Considerar que todos os Id são autonumerados.

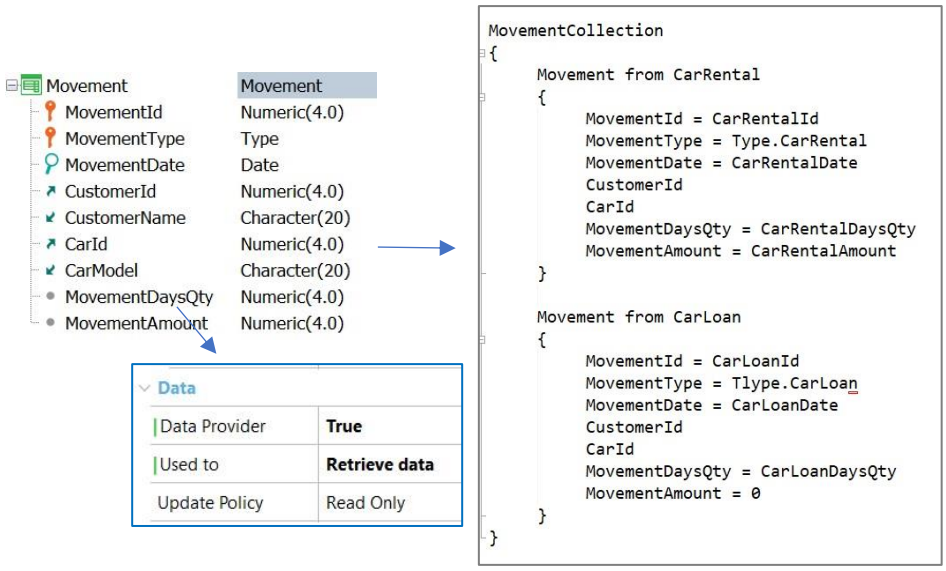
Analise a realidade e, a partir da implementação apresentada, determine o que considere correto:



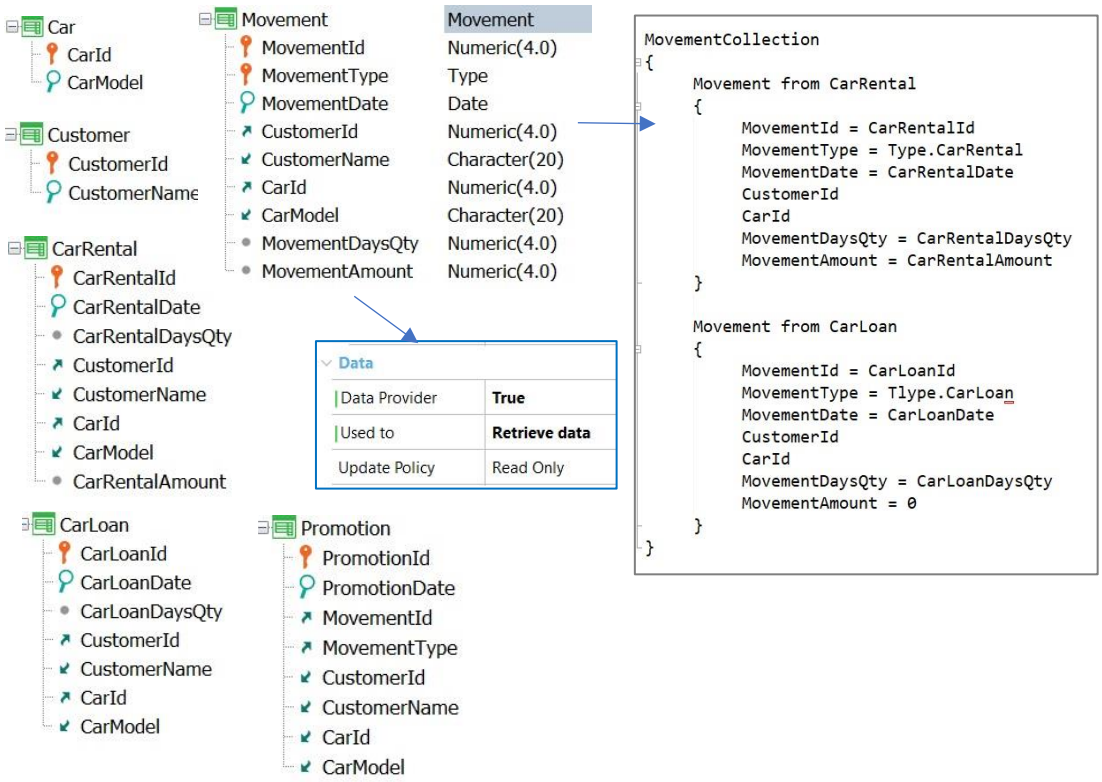
a) A implementação mostrada está correta, mas incompleta. Para resolver o requisito na sua totalidade, falta apenas a lista que mostra toda a informação completa.



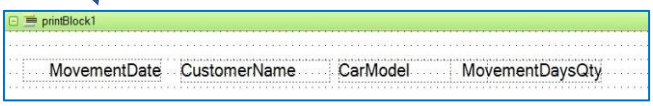
b) A definição da transação dinâmica (Movement) não está correta, pois não é possível identificar se trata-se de um aluguel ou empréstimo. A definição correta é a mostrada, e o requisito é resolvido com esta implementação juntamente com a opção a).



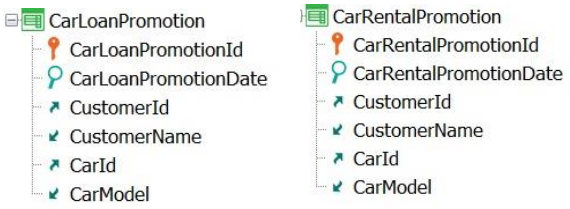
c) O que foi implementado em b) está correto. Para registrar também as promoções de aluguéis e empréstimos (Movement), deve-se definir a transação apresentada. A solução total para o requisito é:



```
For each Movement order MovementDate
      print printBlock1
Endfor
```



d) A solução proposta pela opção c) está correta exceto pela definição da transação Promotion. O par composto por MovementId, MovementType não compõe uma chave estrangeira por não existir como atributos armazenados em uma tabela. Devem ser registradas separadamente as promoções de aluguéis e as promoções de empréstimos, conforme mostrado:



e) Nenhuma das opções anteriores está correta

Respostas

1. b

Justificativa por opções:

a. *Incorreta. As 2 primeiras regras não estão atendendo à solicitação para que sejam avaliadas no servidor e não no lado do cliente. O que significa que não atendem ao requisito solicitado.*

b. *Correta. A regra Msg atende ao que foi solicitado no primeiro ponto, pois exibirá a mensagem de que o nome da cidade está vazio ao sair do campo. Para atender ao segundo requisito, deve ser atribuído a ambas as regras um evento de disparo para ser executado depois de confirmar, mas antes de inserir os dados do cabeçalho, e isso é cumprido com o evento BeforeInsert.*

c. *Incorreta. A regra Msg está bem definida, mas as regras do segundo requisito não atendem ao requisito, embora estejam associadas a um evento de disparo, este será executado depois da inserção dos dados do cabeçalho e do detalhe. Um momento antes de fazer o commit e não antes de inserir o primeiro nível conforme solicitado.*

d. *Incorreta. A opção b) resolve o requisito.*

2. b

Justificativa por opções:

a. *Incorreta. É possível utilizar um atributo fórmula para realizar as operações mencionadas no enunciado.*

b. *Correta. Quando houver 2 regras que utilizem o mesmo atributo, primeiro será executada a regra que modifica a Base de dados e depois a que consulta, portanto, será disparada primeiro a regra Subtract e em segundo lugar a regra Error, razão pela qual deve avaliar para ContryCityMaxDuration < 0 e não contra CityTourDuration.*

c. *Incorreta. O que diz o enunciado está incorreto. A regra Subtract tem a inteligência para determinar o cálculo a ser realizado dependendo do modo em que se esteja executando, portanto, não necessita condicionar a um evento de disparo que também não permitirá que a validação seja do lado do cliente.*

d. *Incorreta. A opção b) resolve o requisito.*

3. c

Justificativa por opções:

a. *Incorreta. Embora ambas as opções cumpram o que foi solicitado, não são equivalentes, no caso da opção 1, ao editar o desenho da transação Director, afetará o restante dos objetos já definidos que referenciam ao diretor, o que implicará realizar modificações para eles.*

b. *Incorreta, pois não são atualizados automaticamente os objetos que referenciam a uma transação que teve atributos modificados, essas alterações devem ser realizadas manualmente.*

c. *Correta. Na opção 2 foi identificado que não deveria ser modificada a estrutura de Director para não afetar os objetos previamente definidos, então a melhor alternativa foi fazer um grupo de subtipos de toda a estrutura de Director para poder invocá-la em Movie e cumprir o requisito.*

d. *Incorreta. A solução proposta também é viável e resolve o requisito, o que a torna incorreta é o enunciado indicando que as opções anteriores não resolvem o que foi solicitado.*

4. a

Justificativa por opções:

a. *Correta. A cláusula when avalia se é cumprida a condição que precede e, em caso afirmativo, executa a ordem especificada.*

b. *Incorreta. A primeira order está bem definida mas não faz nada pois como não possui nenhum Where, não há registros que não atendam ao filtro e nunca entra no When none.*

c. *Incorreta. O condicionamento das ordens é realizado com a cláusula When aplicada a cada uma. A sintaxe mostrada está incorreta.*

d. *Incorreta. O requisito é resolvido com um único For each, declarando as possíveis ordens condicionadas com a cláusula When.*

5. d

Justificativa por opções:

a. *Incorreta. Não vai imprimir todos os filmes que tenha esse ator, apenas aqueles cujo país seja igual ao do ator, já que está sendo feito um Join indireto por CountryId que é o atributo em comum entre ambas as tabelas.*

b. *Incorreta. Serão impressos os filmes cujo país seja igual ao do ator e não todos aqueles que foram registrados. Deve-se identificar que está sendo feito um Join indireto por CountryId que é o atributo em comum entre ambas as tabelas.*

c. *Incorreta. Não serão impressos todos os filmes registrados por Actor, pois está sendo feito um Join indireto por CountryId que é o atributo em comum entre ambas as tabelas, portanto imprimirá os filmes cujo país seja igual ao do ator.*

d. *Correta. A relação entre a tabela Actor e Movie é 1-N indireta, através da tabela Country. O atributo em comum de ambas as tabelas é CountryId, e através dele realizará um Join, resultando no que indica a opção.*

6. b

Justificativa por opções:

a. *Incorreta. Não será agrupado por nome no último for each, mas será ordenado por ele.*

b. *Correta. O que é mostrado no source é um duplo corte de controle onde os 2 primeiros for each indicam o critério pelo qual será agrupada a informação, e o for each mais interno os dados ordenados dos clientes que compraram tickets para essa sala nessa apresentação.*

c. *Incorreta. Não serão repetidas as apresentações, mas serão agrupadas.*

d. *Incorreta. A opção b) resolve o requisito.*

7. c

Justificativa por opções:

a. *Incorreta. Embora a implementação mostre as médias de idades por país, como a tabela base do For each é COUNTRY, na lista podem ser mostrados países que não tenham atores registrados.*

b. *Incorreta. A cláusula unique não se aplica para for each aninhados.*

c. *Correta. A cláusula unique nos permitirá ter uma lista sem elementos repetidos e também serve como condição na fórmula Average para calcular a média apenas daquelas idades de clientes que pertencem a esse país.*

d. *Incorreta. É feita uma tentativa de resolver através de um corte de controle, mas não é correto.*

8. c

Justificativa por opções:

a. *Incorreta. O erro se encontra quando é desejado posicionar no registro do assento do passageiro, pois o correto é se posicionar na PK do subnível para identificar o passageiro*

que será atualizado ficando da seguinte maneira: &FlightSeat= &FlightInstance.Seat.GetbyKey(FlightInstanceSeatId,FlightInstanceSeatChar)

b. *Incorreta. O erro se encontra no momento em que quer fazer a atualização pois o método Update não se associa à variável &FlightSeat do subnível, mas deve ser feito sobre &FlightInstace.Update()*

c. *Correta. É acessada a tabela FlightInstance.Seat e são feitos os filtros corretamente para identificar os voos posteriores à data atual, está correto o método Load para se posicionar nos voos e da mesma maneira o uso do método GetByKey posicionando-se na PK do subnível onde se encontra o passageiro, é feita a alteração e em seguida o método Update associado à variável &FlightInstance, finalizando com o commit*

d. *Incorreta. Não existe o método Replace para atualizar um atributo.*

9. b

Justificativa por opções:

a. *Incorreta. A resposta correta é a opção 2. O método Delete excluirá o Voo todo e não faria o Update dele, que é o necessário.*

b. *Correta. A solicitação é excluir 2 assentos de um determinado voo, o que implica uma atualização do voo e após indicar os assentos que serão removidos nas linhas 2 e 3, é necessário usar o método Update conforme indicado no enunciado.*

c. *Incorreta. Trabalhar com BC é análogo a trabalhar com a transação. Não basta selecionar as linhas a serem removidas, deve-se finalizar com uma operação para a BD e se obtém usando o método Update.*

d. *Incorreta. A opção b) resolve o requisito.*

10. a

Justificativa geral:

A resposta correta é a opção a). A implementação está correta. O método Load nos posiciona no voo com o qual queremos trabalhar e a linha 2 atualiza o preço do voo. É invocado o Procedimento que retornará os valores do assento livre, onde faremos a inserção conforme mostrado da linha 4 até a 7. Para ser possível fazer a exclusão do assento anterior, é feito um

for each para identificar o registro a ser excluído, uma vez removido, a linha &Flight.Seat.Add(&FlightSeat) adiciona o novo assento definido da linha 4 até a 7 para atribuir o passageiro ao seu novo assento. O método Update faz atualização de todas as alterações.

11. a

Justificativa geral:

Quando se possui um Web Panel com grid, devem ser consideradas as seguintes partes para determinar se tem tabela base: Atributos soltos no form (layout), atributos visíveis e não visíveis de um grid, propriedade Base Transaction, Order, Conditions, Unique e Data Selector do grid, bem como os atributos em eventos sem contexto, ou seja, fora de um for each ou fórmula. Neste caso a tabela base é Attraction e é determinada pelo atributo AttractionId que se encontra no evento &NewTrip.Click. Os atributos da regra Parm não participam da determinação da tabela base.

É importante mencionar que o Web Panel ficou mal implementado, pois o for each do evento Load formará um corte de controle por chave primária com a tabela base do grid.

12. d

Justificativa por opções:

a. *Incorreta. Diferentemente do Web Panel, em um objeto Panel a determinação da tabela base da parte fixa é independente da determinação da tabela base do grid. Por esta razão, não está correta, porque não é possível falar de "tabela base do objeto panel", mesmo quando não existe grid porque nesse caso seria a tabela base da parte fixa do panel.*

b. *Incorreta: Encontramos TripId, TripDate e TripDescription na parte fixa e não há botões ou controles que possam conter atributos, nem atributos na Application Bar. Encontramos CountryId na Tab Conditions do Panel (CountryName não participa porque está em uma fórmula) e não há evento Refresh, portanto a mínima tabela estendida que contém esses atributos é TRIPATTRACTION e não TRIP.*

c. *Incorreta. Como foi dito na opção b), a tabela base da parte fixa é TRIPATTRACTION.*

d. *Correta. Como foi dito na opção b), a tabela base da parte fixa é TRIPATTRACTION. Para determinar a tabela base do grid temos os atributos AttractionId, AttractionName, AttractionPhoto, CountryName e CityName. Nas condições gerais do Panel temos CountryId e no evento Load temos o atributo TripId que está fora de algum contexto (nem em um For Each, nem em uma fórmula) portanto participa da determinação da tabela base do grid. A mínima tabela estendida que contém esses atributos é TRIPATTRACTION, portanto a tabela base do grid é TRIPATTRACTION*

13. a

Justificativa geral:

A opção correta é a a) ficando na ordem: 2,3,1, pois a primeira regra a ser executada é a do ponto 2, que será disparada após a inserção do primeiro nível da transação Food, depois o ponto 3, que será executada quando for deixado o nível Food.Pet, ficando em último lugar o ponto 1 sendo a última a ser executada depois de ter feito commit.

14. c

Justificativa por opções:

a. *Incorreta. A definição de Movement está incompleta, deve ser modificada a chave primária para ser possível identificar se trata-se de um empréstimo ou de um aluguel e também falta uma entidade para inserir as promoções.*

b. *Incorreta. Embora seja corrigida a definição da transação Movement para ser possível identificar um aluguel de um empréstimo, falta uma transação para poder definir as promoções.*

c. *Correta. É mostrada uma correta definição da transação Movement e é adicionada a de Empréstimos, bem como a Lista PDF, por isso cumpre o requisito solicitado.*

d. *Incorreta. A PK de uma transação dinâmica de tipo Retrieve pode ser utilizada como FK em outra transação, GeneXus criará uma pseudo-chave para poder executar os controles de integridade, por isso não está correto o que diz esta opção.*

e. *Incorreta. A opção c) resolve o requisito.*