Examen Analista GeneXus 18

1. Se está desarrollando una aplicación para una agencia de viajes. Se desean registrar los distintos City tours que se ofrecen para pasear por diferentes atracciones turísticas de la ciudad. Se muestra la estructura de la transacción CityTour únicamente. Se desea lograr el siguiente comportamiento:

   - Que se advierta lo más temprano posible que se ha dejado vacío el campo CityTourName (en cualquiera de los modos).
   - Si se está insertando un city tour, y se selecciona tipo Economy , se quiere prohibir el ingreso de un precio que esté por encima del rango de precios del tipo económico. El chequeo lo hará un proc CheckPrice cuya lógica no importa para este ejercicio. Pero se desea que el control no se realice en el cliente, sino solo en el servidor, inmediatamente antes de que el cabezal se grabe en la tabla CityTour.

   Deberás determinar cuál de las siguientes opciones es la que implementa correctamente las reglas según estos requerimientos.

   a.

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

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

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

b.

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

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

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

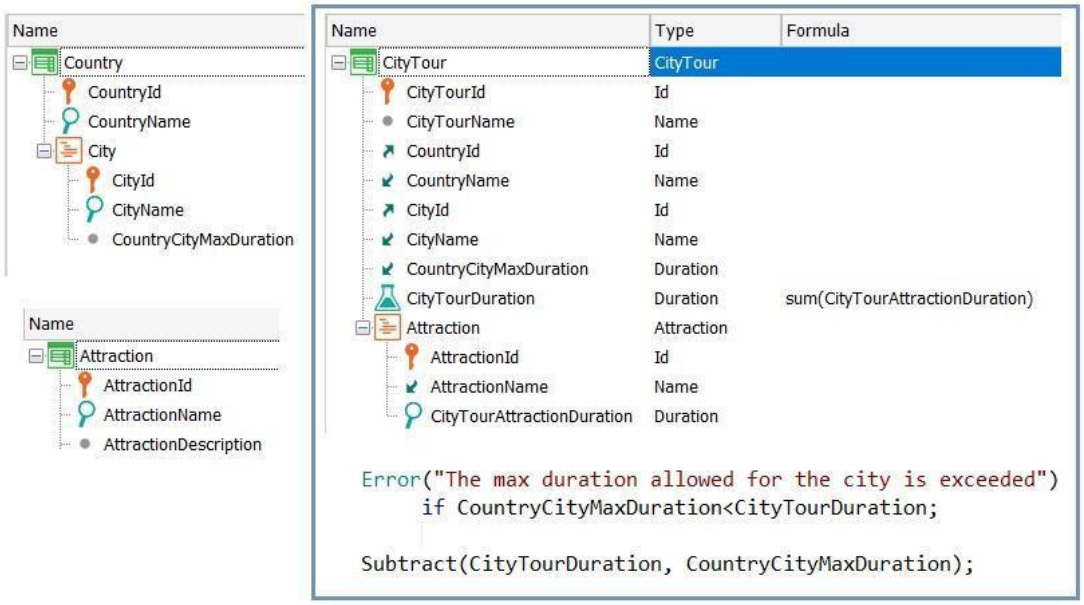
d. Ninguna de las anteriores

2. En el contexto de una aplicación para una agencia de viajes se tiene una transacción para registrar los tours por una ciudad que la agencia ofrece a sus clientes. Se registran las atracciones turísticas que se visitarán, junto con el tiempo (duration) reservado para la visita a cada atracción.

   Suponiendo que cada agencia de viajes tiene una cantidad máxima de tiempo permitido para todos sus city tours por la ciudad, se ha agregado el atributo CountryCityMaxDuration a la transacción Country.

   Cada vez que se ingresa un city tour a la ciudad ese atributo deberá actualizarse de acuerdo a la duración total del city tour.

   Para ello se han especificado en CityTour las dos reglas que mostramos en la imagen. Queremos saber si con ellas cumplimos o no con el requerimiento. De las opciones que damos a continuación, elige la correcta.



   a. No puede utilizarse un atributo fórmula para ser restado, sumado o actualizado de un atributo de la tabla extendida de ese mismo nivel.

   b. La regla de error está mal condicionada, porque dado que debe utilizar el atributo CountryCityMaxDuration, que es actualizado por la regla subtract, siempre se disparará después de ésta. Por tanto, la condición debería ser “If CountryCityMaxDuration < 0”.

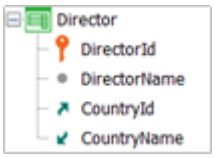
      .

c. Falta condicionar las reglas al evento “On BeforeInsert, BeforeUpdate, BeforeDelete” para que se dispare solo en el server y antes de actualizar el registro. De lo contrario se disparará dos veces y actualizará doble.

d. Ninguna de las anteriores.

3. Una película (Movie) puede ser dirigida por uno o varios directores (Director). Es necesario registrar para las películas el país donde fueron realizadas, e indicar también el país de cada director.

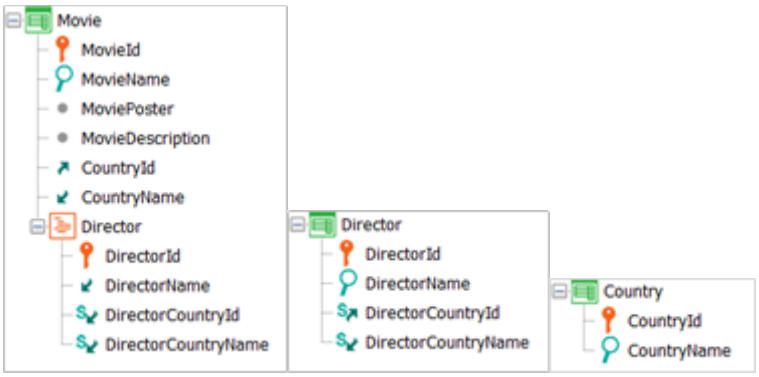
El equipo de desarrollo ha definido la transacción Director de la siguiente manera:



Antes de continuar con las otras transacciones del modelo, el equipo también ha creado todos los programas para resolver los requerimientos solicitados por la empresa que gestionará el Cinema en lo que refiere a los directores (por ejemplo: listados de los directores en PDF, visualización de los directores filtrando por su país desde un Web Panel, procedimientos que realizan diferentes chequeos con sus datos, etc).

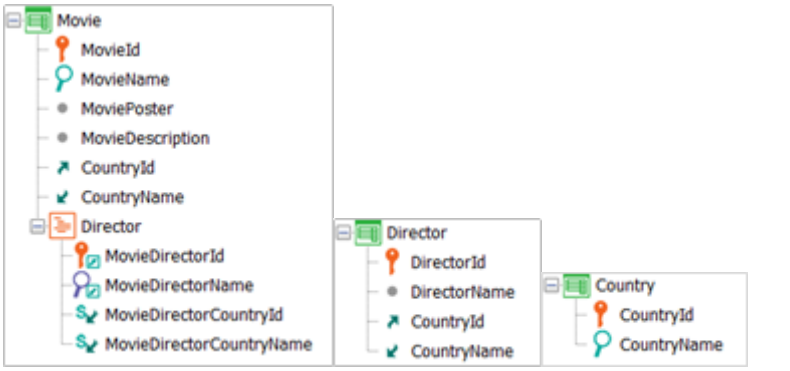
Analiza las opciones 1 y 2 e indica lo que consideres correcto.

1)



Subtype	Description	Supertype
DirectorCountry		
DirectorCountryId	Director Country Id	CountryId
DirectorCountryName	Director Country Name	CountryName

2)



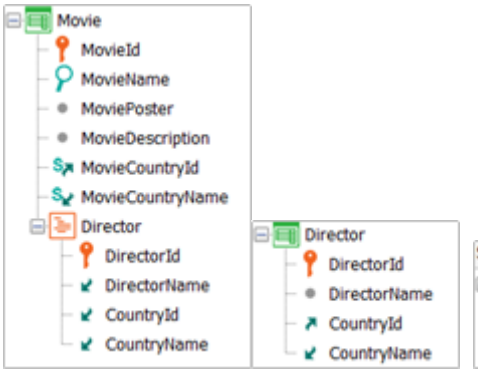
Subtype	Description	Supertype
MovieDirector		
MovieDirectorId	Movie Director Id	DirectorId
MovieDirectorName	Movie Director Name	DirectorName
MovieDirectorCountryId	Movie Director Country Id	CountryId
MovieDirectorCountryName	Movie Director Country Name	CountryName

a) Ambas opciones son equivalentes y resuelven lo solicitado sin ventajas o desventajas una frente a la otra.

b) Si bien ambas opciones resuelven lo solicitado, la opción 1 es más adecuada en este caso, pues al quedar el grupo de subtipos con menos elementos son creadas menos referencias, y al cambiar los nombres de los atributos CountryId y CountryName en la transacción Director, automáticamente son actualizados esos atributos en los objetos donde se estén utilizando.

c) Si bien ambas opciones resuelven lo solicitado, la opción 2 es más adecuada en este caso, pues con ella no existe la necesidad de ir a cada programa ya creado para modificar los atributos CountryId y CountryName, ya que no cambian.

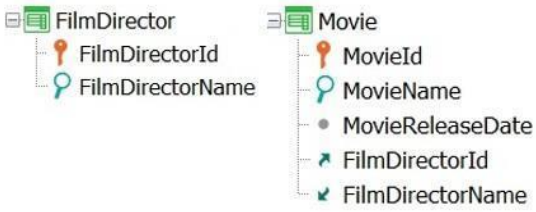
d) Ninguna de las dos opciones resuelve lo solicitado. La única solución sería la siguiente:



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

4. Considera el diseño de transacciones que se muestra. Se necesita un listado que reciba una fecha por parámetro, y muestre las películas (Movie) ordenadas por fecha de estreno (MovieReleaseDate), en caso de que dicho parámetro sea una fecha posterior a la actual. En otro caso, el listado se verá ordenado en forma descendente por el nombre del director (FilmDirectorName).

Determina lo que consideres correcto:



a) El requerimiento se resuelve indicando los órdenes deseados condicionados con las respectivas cláusulas When. Siempre se aplicará el orden que verifique las condiciones:

```
Parm(in:&DateFrom);

For each Movie order MovieReleaseDate when &DateFrom > &Today
     order (FilmDirectorName) when &DateFrom <= &Today
     print printBlock1
endfor
```

b) debe indicar uno de los órdenes condicionados, y en caso de no cumplirse la condición, se indica el otro utilizando la cláusula When none del 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) El requerimiento se resuelve indicando los órdenes deseados condicionados con la cláusula If - Else como se muestra:

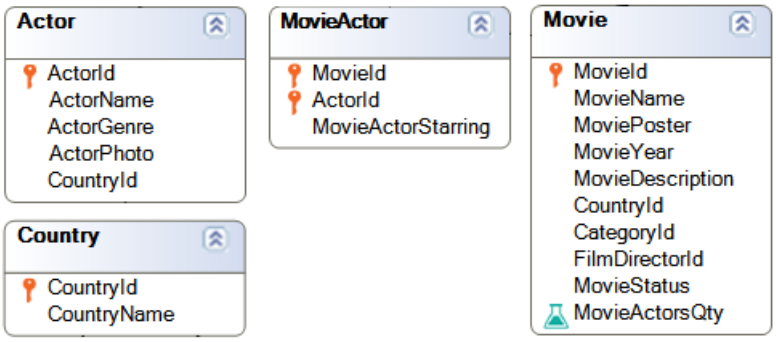
```
Parm(in:&DateFrom);

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

     print printBlock1
endfor
```

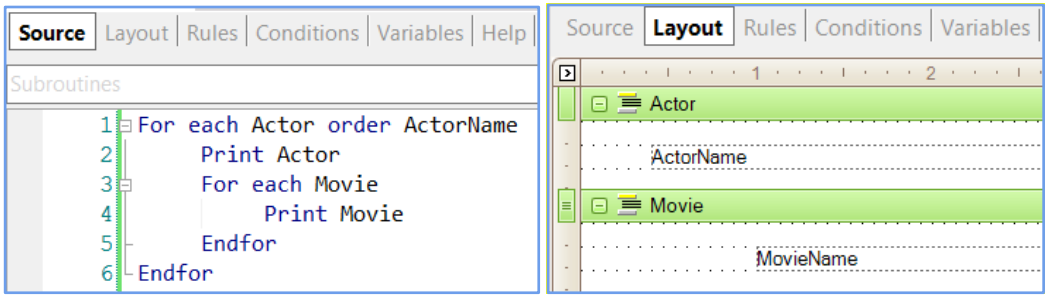
d) No es posible resolver el requerimiento solicitado declarando un solo For each e indicando varios órdenes condicionados. Para resolverlo se debe implementar cada For each según el valor del parámetro recibido:

```
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. Observa las tablas generadas y el procedimiento que se muestran a continuación.

A partir de esta implementación, de las siguientes afirmaciones selecciona la correcta.





a. Se imprimirán todos los actores. Y para cada actor que esté registrado en al menos una película, todas las películas que tengan ingresado a ese actor.

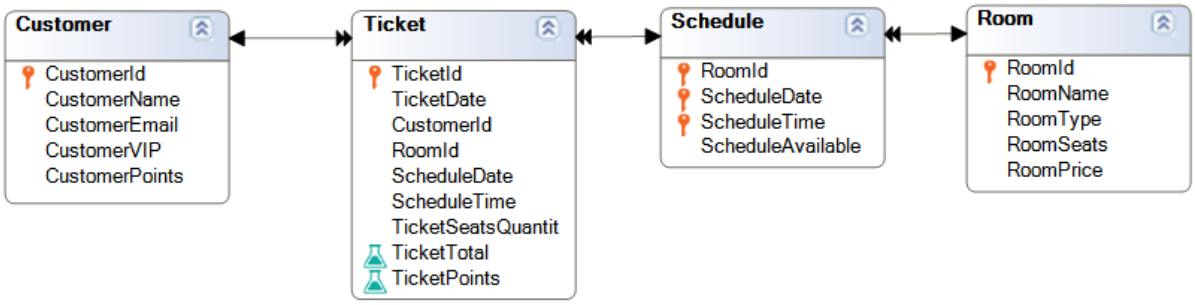
b. Se imprimirán todos los actores. Y para cada actor todas las películas que haya registradas.

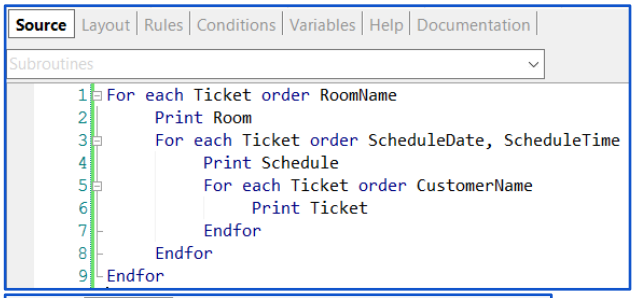
c. Se imprimirán todos los actores. Y para cada actor que esté registrado en al menos una película, todas las películas que haya registradas.

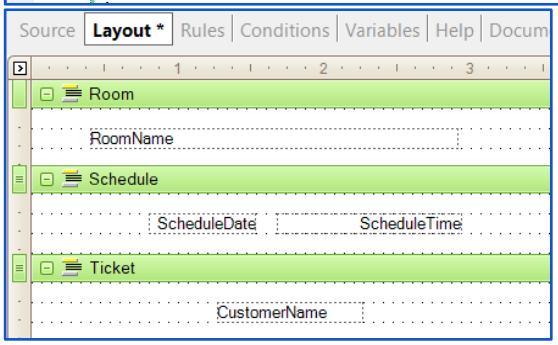
d. Se imprimirán todos los actores. Y para cada actor que esté registrado en al menos una película, se imprimirán las películas cuyo país sea igual al del actor.

6. Observa el diagrama de tablas y el procedimiento que se muestra a continuación.

A partir de esta implementación, selecciona la opción correcta.







a. Se mostrarán en pantalla todas las salas que estén presentes en al menos un ticket.
Para cada una de ellas, todas las funciones ingresadas para esa sala, que sean parte de algún ticket, agrupadas por fecha (ScheduleDate) y hora (ScheduleTime) de la función.

   Y para cada una de esas funciones, todos los clientes que hayan comprado algún ticket de esa función, agrupados por nombre

b. Se mostrarán en pantalla todas las salas que estén presentes en al menos un ticket.
Para cada una de ellas, todas las funciones ingresadas para esa sala, que sean parte de algún ticket, agrupadas por fecha (ScheduleDate) y hora (ScheduleTime) de la función.

   Y para cada una de esas funciones, todos los clientes que hayan comprado algún ticket de esa función, ordenados por nombre

c. Se mostrarán en pantalla todas las salas que estén presentes en al menos un ticket. Para cada sala se mostrarán todas las funciones que haya ingresadas al sistema, ordenadas por fecha (ScheduleDate) y hora (ScheduleTime) de la función (se repetirán si hay funciones con misma fecha y hora). Y para cada una de esas funciones, todos los clientes, ordenados por nombre.

   No se repetirán las funciones, sino que se agruparán.

d. Ninguna de las anteriores.

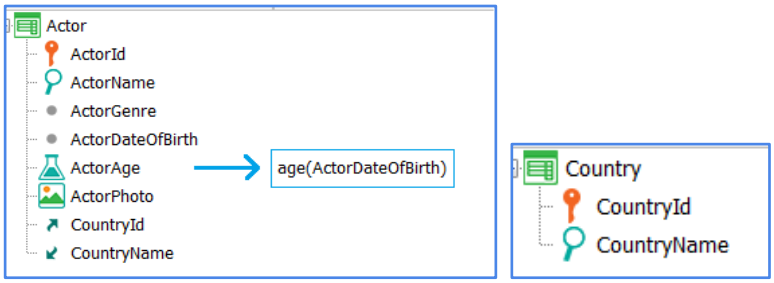
7. En la aplicación para un complejo de cines, tenemos la siguiente implementación.

Observa las transacciones y el procedimiento que se muestran.

Interesa listar todos los países, **sin repetir, que tengan actores registrados**, cada uno con el promedio de edad de sus actores.

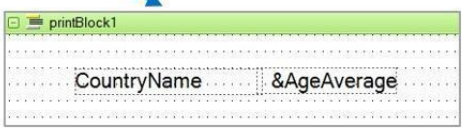
Determina la opción que consideres correcta.

Nota: Una de las funcionalidades de la función Age() es calcular la edad a partir de una fecha de nacimiento.



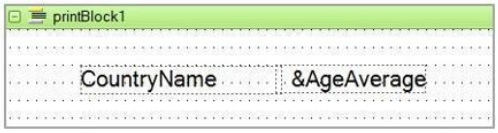
a)

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



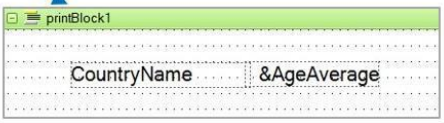
b)

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



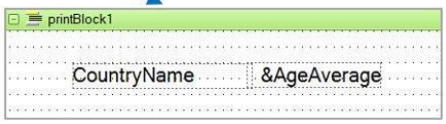
c)

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

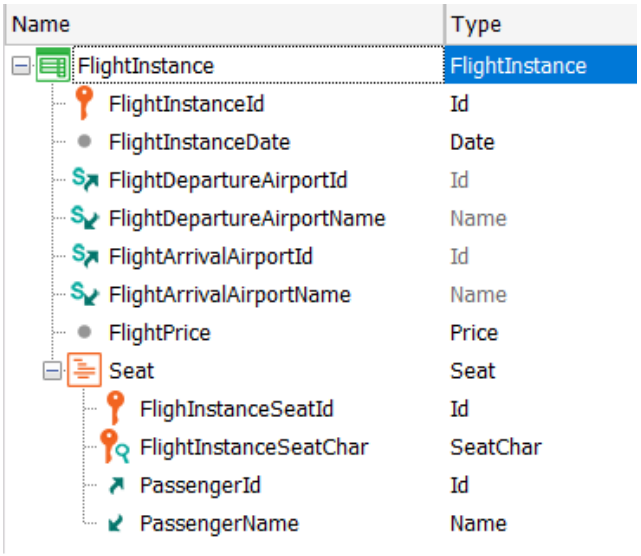


d)

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



8. Se está desarrollando una aplicación simplificada para una agencia de viajes. Se tiene la transacción FlightInstance que registra los vuelos. Para cada uno se registra, además de la fecha del vuelo, su aeropuerto de partida y aeropuerto de llegada (grupos de subtipos), su precio y los pasajeros registrados, con sus asientos.



Queremos cambiar a un pasajero por otro en todos los vuelos posteriores a la fecha de hoy. Para ello, se programará un procedimiento que recibirá por parámetros el pasajero a ser modificado, &OldPassengerId, y el pasajero que lo reemplazará, &NewPassengerId.

En el procedimiento tenemos dos variables de tipo business component:

Las opciones siguientes son posibles implementaciones del requerimiento. Indica la correcta.

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
```

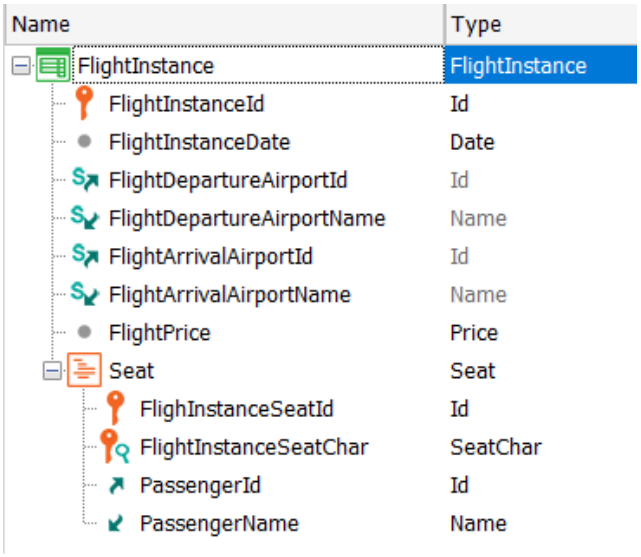
c.

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

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

      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. Se tiene la transacción FlightInstance que registra los vuelos. Para cada uno se registra, además de la fecha del vuelo, su aeropuerto de partida y aeropuerto de llegada (grupos de subtipos), su precio y los pasajeros registrados, con sus asientos.



Se quieren eliminar de un vuelo en particular (el 3546) los asientos 1A y 1F, a través de una variable &flight Business Component de FlightInstance.

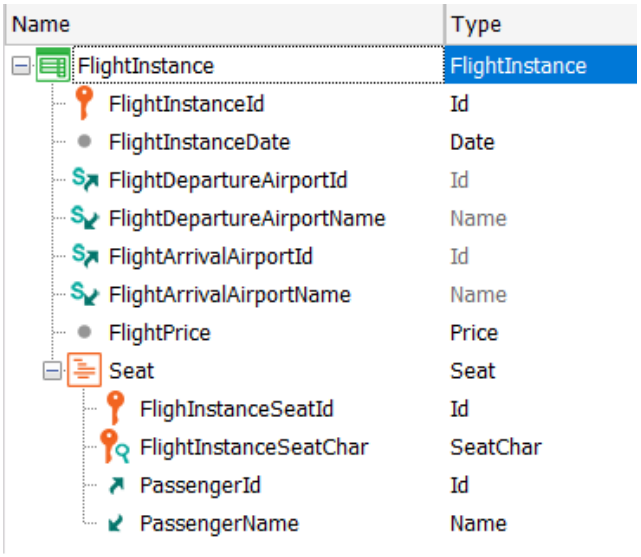
Para ello, se ha escrito el siguiente código:

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

Indica la opción correcta entre las siguientes:

a. El código anterior es correcto y hace lo que queremos
b. El código anterior es incorrecto, porque si bien se eliminarán las dos líneas de la colección Seat en el BC, como el método que se ejecuta es el Delete, se eliminará todo, cabezal y líneas. Tendría que haberse escrito en la 4ta línea &flight.Update() para que fuera correcto.
c. El código anterior es incorrecto. Sería correcto si eliminàramos la 4ta. línea , es decir, &fligh.Delete(), dejando el resto como está.
d. El código anterior es incorrecto por otras razones.

10. Se tiene la transacción FlightInstance que registra los vuelos. Para cada uno se registra, además de la fecha del vuelo, su aeropuerto de partida y aeropuerto de llegada (grupos de subtipos), su precio y los pasajeros registrados, con sus asientos.



Queremos cambiar de asiento a un pasajero dado, el 10, en un vuelo dado, el 5, y modificar el precio de ese vuelo, aumentándole un 10% su valor.

Imaginemos que {FlightInstanceId, PassengerId} es clave candidata de FlightInstanceSeat, para que un mismo pasajero solo pueda estar en un asiento de un mismo vuelo. Para cambiar al pasajero de asiento eliminaremos el **registro** de asiento del pasajero para ese vuelo y lo insertaremos nuevamente, con todo igual y el asiento cambiado.

Supongamos que contamos con el procedimiento GetEmptySeat, al que pasándole el id del vuelo devuelve un SeatId y un SeatChar no ocupados en el vuelo, para poder reasignárselos al pasajero que se quiere mover de lugar.

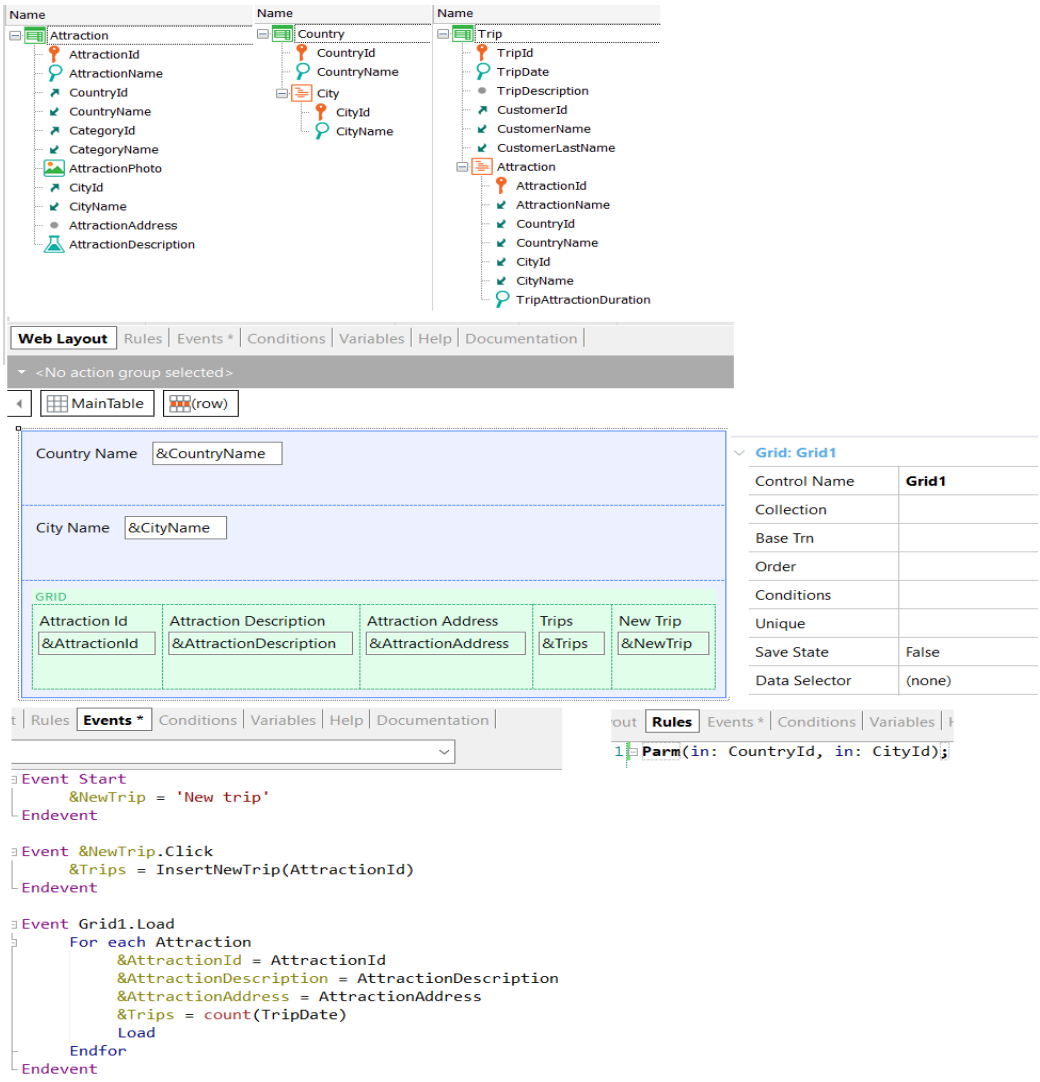
Para implementar el requerimiento, se ha programado el siguiente código en el evento de un Web Panel, donde &flight es de tipo de datos el BC FlightInstance y &flightSeat del tipo de datos el BC FlightInstance.Seat.

```
&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
```

Indica cuál de las siguientes opciones es correcta.

a. La implementación es correcta.
b. Está mal implementado debido a que está mal la inserción de la línea.
c. Está mal implementado debido a que está mal la eliminación de la línea.
d. Ninguna de las anteriores.

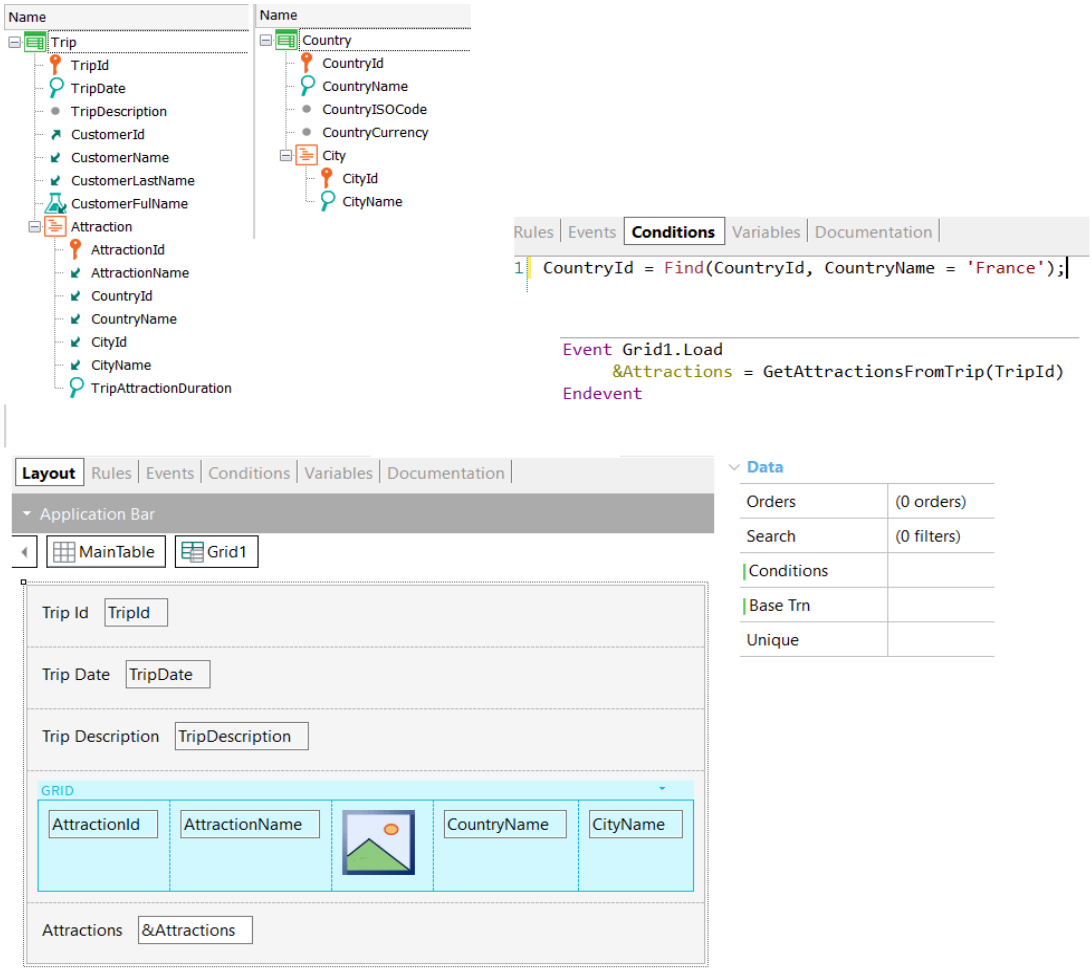
11. En el contexto de una aplicación para una agencia de viajes, con las transacciones que mostramos en la imagen, se desea ver las atracciones de una ciudad de un determinado país, con los viajes que se han realizado a cada atracción. Además, se debe dar la posibilidad de crear un viaje nuevo para visitar una atracción en particular.
    Se ha implementado el siguiente Web Panel. Determina cuál es su tabla base.



a) ATTRACTION
b) COUNTRYCITY
c) COUNTRY
d) TRIPATTRACTION
e) El web panel no tiene tabla base

12. En el marco de una aplicación para una agencia de viaje se implementa el objeto panel que se muestra. Se agregan también las transacciones involucradas. Lo que no se menciona es porque no se modifica respecto al valor default.

Selecciona la opción que consideres correcta.



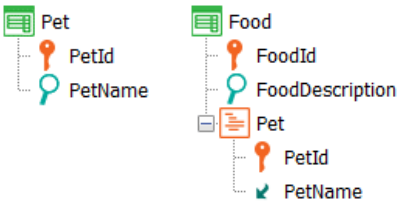
a) La tabla base del objeto panel es TRIPATTRACTION

b) La tabla base de la parte fija es TRIP y la del grid es ATTRACTION

c) La tabla base de la parte fija es TRIP y la del grid es TRIPATTRACTION

d) La tabla base de la parte fija es TRIPATTRACTION y la del grid es TRIPATTRACTION

13. Considera las transacciones que se muestran y determina el orden en el cual se dispararán las reglas declaradas en la transacción 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) Las reglas se disparan en el orden en el que son declaradas.

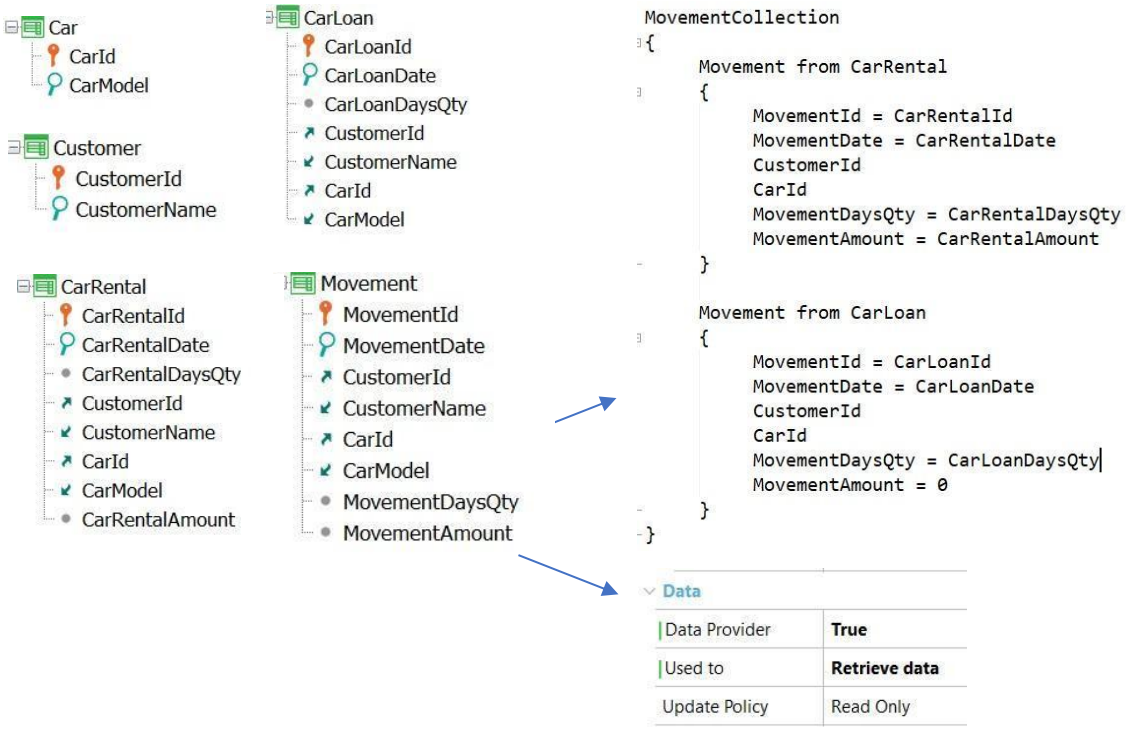
14. Se está diseñando una aplicación para una empresa de Alquiler de autos.

La misma realiza alquileres y préstamos de autos a sus clientes. Se necesita unificar la información de alquileres y préstamos de autos, para finalmente poder listar dicha información ordenada por fecha.

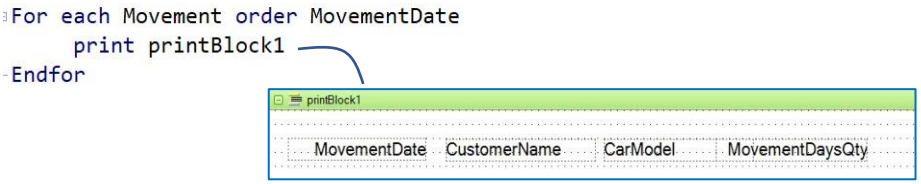
La empresa necesita registrar también promociones (Promotion) de dichos alquileres y préstamos de autos

Nota: Considerar que todos los Id son autonumerados.

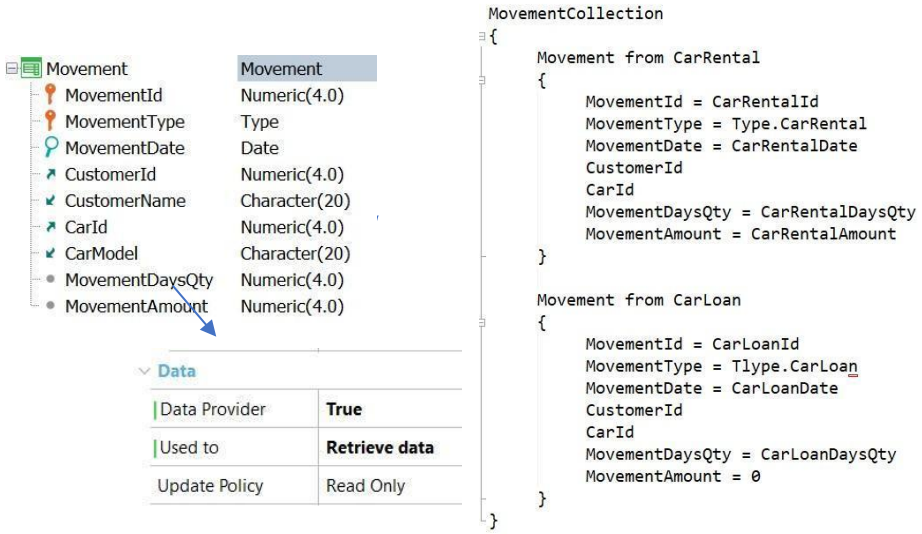
Analiza la realidad, y a partir de la implementación que se muestra, determina lo que consideres correcto:



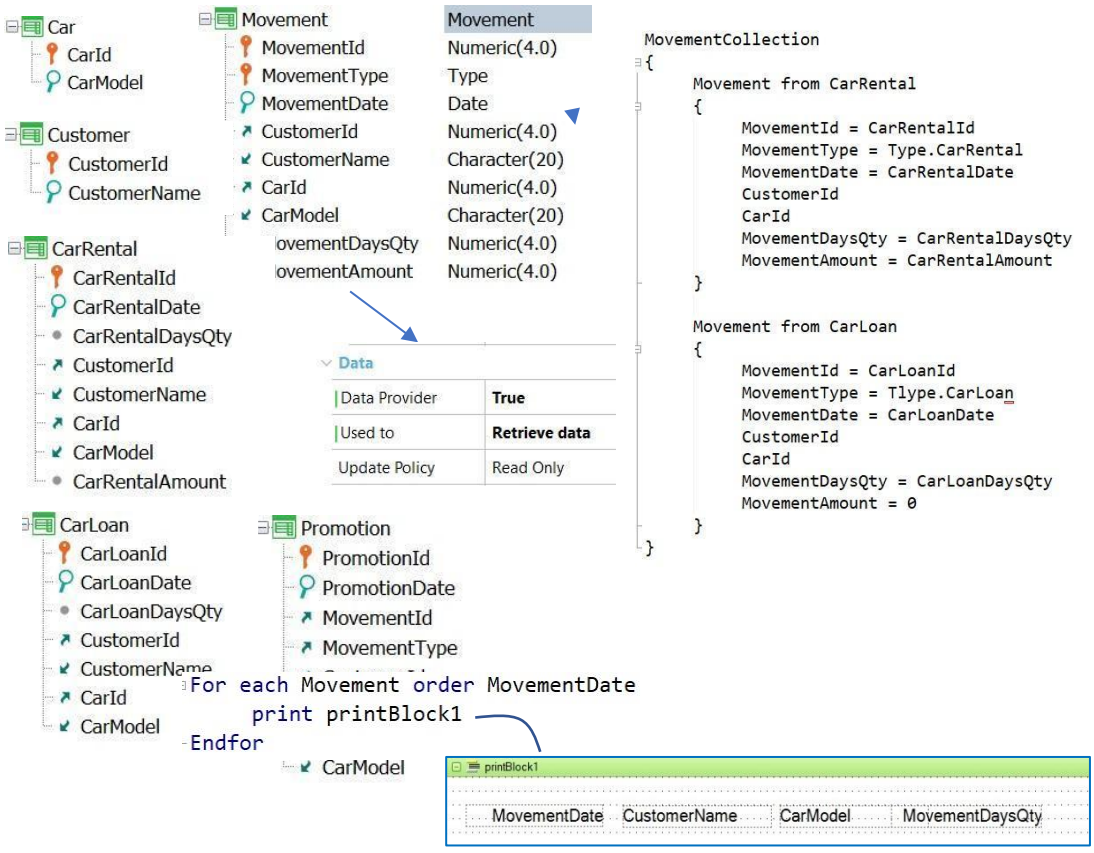
a) La implementación que se muestra es correcta pero incompleta. Para resolver el requerimiento en su totalidad solamente resta el listado que muestre toda la información completa.



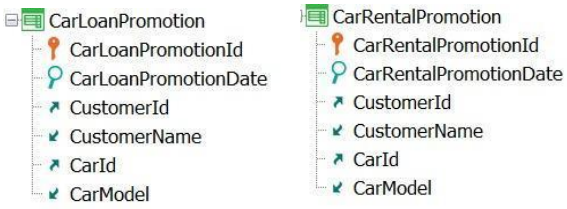
b) La definición de la transacción dinámica (Movement) no es correcta, ya que no es posible identificar si se trata de un alquiler o préstamo. La definición correcta es la que se muestra, y el requerimiento se resuelve con esta implementación junto con la opción a).



c) Lo implementado en b) es correcto. Para registrar además las promociones de alquileres y préstamos (Movement) se debe definir la transacción que se muestra. La solución total al requerimiento es:



d) La solución que propone la opción c) es correcta excepto la definición de la transacción Promotion. El par compuesto por MovementId, MovementType no conforma una clave foránea por no existir como atributos almacenados en una tabla. Se deben registrar por separado las promociones de alquileres y las promociones de préstamos, como se muestra:



e) Ninguna de las opciones anteriores es correcta

Respuestas

1. b

Justificación por incisos:

a. *Incorrecta. Las primeras 2 reglas no están cumpliendo con lo solicitado de que se evalúen en el servidor y no del lado del cliente. Lo que hace que no cumplan con el requerimiento solicitado.*

b. *Correcta. La regla Msg cumple con lo solicitado en el primer punto ya que mostrará el mensaje de que el nombre de la ciudad está vacío al salir del campo. Para cumplir con el segundo requerimiento se debe asignar a ambas reglas un evento de disparo para ser ejecutado después de confirmar pero antes de insertar los datos del cabezal y esto se cumple con el evento BeforeInsert.*

c. *Incorrecta. La regla Msg está bien definida, pero las reglas del segundo requerimiento no cumplen con el requerimiento, si bien están asociadas a un evento de disparo, este se ejecutará después de la inserción de los datos del cabezal y del detalle. Un momento antes de hacer el commit y no antes de insertar el primer nivel como se solicita.*

d. *Incorrecta. La opción b) resuelve el requerimiento.*

2. b

Justificación por incisos:

a. *Incorrecta. Es posible utilizar un atributo fórmula para realizar las operaciones que se mencionan en el enunciado.*

b. *Correcta. Cuando hay 2 reglas que utilizan un mismo atributo, primero se ejecutará la regla que modifica la Base de datos y después la que consulta, por tanto, se disparará primero la regla Subtract y en segundo lugar la regla Error, razón por lo cual debe evaluar a ContryCityMaxDuration < 0 y no contra CityTourDuration.*

c. *Incorrecta. Lo que dice el enunciado es incorrecto. La regla Subtract tiene la inteligencia para determinar el cálculo a realizar dependiendo del modo en que se esté ejecutando, por ello no requiere de condicionar a un evento de disparo que además no permitirá que la validación sea del lado del cliente.*

d. *Incorrecta. La opción b) resuelve el requerimiento.*

3. c

Justificación por incisos:

a. *Incorrecta. Si bien ambas opciones cumplen con lo solicitado, no son equivalentes, en el caso de la opción 1 al editar el diseño de la transacción Director va afectar el resto de los objetos ya definidos que referencian al director lo que va implicar realizar modificaciones a los mismos.*

b. *Incorrecta, ya que no se actualizan automáticamente los objetos que referencian a una transacción que haya tenido atributos modificados, esos cambios deben realizarse manualmente.*

c. *Correcta. En la opción 2 se identificó que no se debía modificar la estructura de Director para no afectar los objetos previamente definidos, por lo cual la mejor alternativa era hacer un grupo de subtipos de toda la estructura de Director para poder invocarla en Movie y cumplir con el requerimiento.*

d. *Incorrecta. La solución propuesta también es viable y resuelve el requerimiento, lo que la hace incorrecta es el enunciado indicando que las opciones anteriores no resuelven lo solicitado.*

4. a

Justificación por incisos:

a. *Correcta. La cláusula when evalúa que se cumpla la condición que le precede y en caso afirmativo ejecuta el orden especificado.*

b. *Incorrecta. El primer order está bien definido pero no hace nada porque al no tener ningún Where no hay registros que no cumplan el filtro y no entra nunca al When none.*

c. *Incorrecta. El condicionamiento de los órdenes se realiza con la cláusula When aplicada a cada uno. La sintaxis que se muestra es incorrecta.*

d. *Incorrecta. El requerimiento se resuelve con un solo For each, declarando los posibles órdenes condicionados con la cláusula When.*

5. d

Justificación por incisos:

a. *Incorrecta. No va imprimir todas las películas que tenga esa actor, solo aquellas cuyo país sea igual al del actor ya que se está haciendo un Join indirecto por CountryId que es el atributo en común entre ambas tablas.*

b. *Incorrecta. Se imprimirán las películas cuyo país sea igual al del actor y no todas las que haya registradas. Se debe identificar que se está haciendo un Joín indirecto por CountryId que es el atributo en común entre ambas tablas.*

c. *Incorrecta. No se imprimirán todas las películas registradas por Actor, ya que se está haciendo un Joín indirecto por CountryId que es el atributo en común entre ambas tablas, por lo cual imprimirá las películas cuyo país sea igual al del actor.*

d. *Correcta. La relación entre la tabla Actor y Movie es 1-N indirecta, a través de la tabla Country. El atributo en común de ambas tablas es CountryId, y por él realizará un Join dando como resultado lo que indica la opción.*

6. b

Justificación por incisos:

a. *Incorrecta. No se agrupara por nombre en el último for each, sino que se ordenará por él.*

b. *Correcta. Lo que se muestra en el source es un doble corte de control donde los 2 primeros for each indican el criterio por el cual se estará agrupando la información, y el for each más interno los datos ordenados de los clientes que compraron tickets para esa sala en esa función.*

c. *Incorrecta. No se repetirán las funciones, sino que se agruparán.*

d. *Incorrecta. La opción b) resuelve el requerimiento.*

7. c

Justificación por incisos:

a. *Incorrecta. Si bien la implementación muestra los promedios de edades por país, como la tabla base del For each es COUNTRY, en el listado pueden mostrarse países que no tengan actores registrados.*

b. *Incorrecta. La cláusula unique no aplica para for each anidados.*

c. *Correcta. La cláusula unique nos va permitir tener una lista sin elementos repetidos y además sirve como condición en la fórmula Average para promediar solo aquellas edades de clientes que pertenezcan a ese país.*

d. *Incorrecta. Se intenta resolver mediante un corte de control, pero no es correcto.*

8. c

Justificación por incisos:

a. *Incorrecta. El error se encuentra cuando se quiere posicionar en el registro del asiento del pasajero ya que lo correcto es posicionarse en la PK del subnivel para identificar al*

pasajero que se va actualizar quedando de la siguiente manera: &FlightSeat= &FlightInstance.Seat.GetbyKey(FlightInstanceSeatId,FlightInstanceSeatChar)

b. *Incorrecta. El error se encuentra al momento de querer hacer la actualización ya que el método Update no se asocia a la variable &FlightSeat del subnivel, si no que debe hacerse sobre &FlightInstace.Update()*

c. *Correcta. Se accede a la tabla FlightInstance.Seat y se hacen los filtros correctamente para identificar los vuelos posteriores a la fecha actual, es correcto el método Load para posicionarse en los vuelos y de la misma manera el uso del método GetByKey posicionándose en la PK del subnivel donde se encuentra el pasajero, se hace el cambio y luego el método Update asociado a la variable &FlightInstance, finalizando con el commit*

d. *Incorrecta. No existe el método Replace para actualizar un atributo.*

9. b

Justificación por incisos:

a. *Incorrecta. La respuesta correcta es la opción 2. El método Delete eliminará el Vuelo completo y no haría el Update del mismo que es lo que se necesita.*

b. *Correcta. La solicitud es eliminar 2 asientos, de un vuelo en particular, lo que implica una actualización del vuelo y después de indicar los asientos que serán removidos en la línea 2 y 3 se necesita usar el método Update como lo indica el enunciado.*

c. *Incorrecta. El trabajar con BC es análogo a realizarlo con la transacción. No alcanza con seleccionar las líneas a ser removidas, se debe finalizar con una operación a la BD y se obtiene usando el método Update.*

d. *Incorrecta. La opción b) resuelve el requerimiento.*

10. a

Justificación general:

La respuesta correcta es la opción a). La implementación es correcta. El método Load nos posiciona en el vuelo con el que queremos trabajar y la línea 2 actualiza el precio del vuelo. Se invoca al Procedimiento el cual va retornar los valores del asiento libre, donde haremos la inserción como se muestra de la línea 4 a la 7. Para poder hacer la eliminación del asiento

anterior se hace un for each para identificar el registro a dar de baja, una vez eliminado, la línea &Flight.Seat.Add(&FlightSeat) agrega el nuevo asiento definido de la línea 4 a la 7 para asignar al pasajero en su nuevo asiento. El Método Update hace la actualización de todos los cambios.

11. a

Justificación general:

Cuando se tiene un Web Panel con grid se deben considerar las siguientes partes para determinar si tiene tabla base: Atributos sueltos en el form (layout), atributos visibles y no visibles de un grid, propiedad Base Transaction, Order, Conditions, Unique y Data Selector del grid así como los atributos en eventos sin contexto, es decir fuera de un for each o fórmula. En este caso la tabla base es Attraction y se determina por el atributo AttractionId que se encuentra en el evento &NewTrip.Click . Los atributos de la regla Parm no participan en la determinación de la tabla base.

Es importante mencionar que el Web Panel quedó mal implementado, ya que el for each del evento Load conformará un corte de control por clave primaria con la tabla base del grid.

12. d

Justificación por incisos:

a. *Incorrecta. A diferencia del Web Panel, en un objeto Panel la determinación de la tabla base de la parte fija es independiente de la determinación de la tabla base del grid. Por esa razón, no es correcta porque no se puede hablar de "tabla base del objeto panel", aun cuando no haya grid porque en ese caso sería la tabla base de la parte fija del panel.*

b. *Incorrecta: Encontramos TripId, TripDate y TripDescription en la parte fija y no hay botones o controles que puedan contener atributos, ni atributos en la Application Bar. Sí encontramos a CountryId en el Tab Conditions del Panel (CountryName no participa porque está en una fórmula) y no hay evento Refresh, por lo que la mínima tabla extendida que contiene a esos atributos es TRIPATTRACTION, y no TRIP.*

c. *Incorrecta. Como se dijo en la opción b), la tabla base de la parte fija es TRIPATTRACTION.*

d. *Correcta. Como se dijo en la opción b), la tabla base de la parte fija es TRIPATTRACTION. Para determinar la tabla base del grid tenemos a los atributos AttractionId, AttractionName, AttractionPhoto, CountryName y CityName. En las condiciones generales del Panel tenemos a CountryId y en el evento Load tenemos al atributo TripId que está fuera de algún contexto (ni en un For Each, ni en una fórmula) por lo que participa de la determinación de la tabla base del grid. La mínima tabla extendida que contiene a esos atributos es TRIPATTRACTION, por lo que la tabla base del grid es TRIPATTRACTION*

13. a

Justificación general:

La opción correcta es la a) quedando en el orden: 2,3,1, debido a que la primera regla en ejecutarse es la del punto 2, la cual se disparará después de la inserción del primer nivel de la transacción Food, después el punto 3, la cual se ejecutará cuando se abandone el nivel Food.Pet, quedando en último lugar el punto 1 siendo la última en ejecutarse después de haber hecho commit.

14. c

Justificación por incisos:

a. *Incorrecta. La definición de Movement está incompleta, se debe modificar la llave primaria para poder identificar si se trata de un préstamo o de un alquiler y también falta una entidad para ingresar las promociones.*

b. *Incorrecta. Si bien se corrige la definición de la transacción Movement para poder identificar un alquiler de un préstamo, falta una transacción para poder definir las promociones.*

c. *Correcta. Se muestra una correcta definición de la transacción Movement y se agrega la de Prestamos, así como el Listado PDF por lo cual cumple el requerimiento solicitado.*

d. *Incorrecta. La PK de una transacción dinámica de tipo Retrieve puede ser utilizada como FK en otra transacción, GeneXus creará una seudo llave para poder ejecutar los controles de integridad, por lo cual no es correcto lo que dice esta opción.*

e. *Incorrecta. La opción c) resuelve el requerimiento.*