1) *2 pts.* Se tiene una aplicación GeneXus para una ensambladora de Automóviles.
La misma cuenta con la transacción *Auto* para registrar los automóviles que son ensamblados. De los autos es necesario registrar el país de origen del constructor y el país de origen de los repuestos.
Determine el diseño de transacciones que considere correcto.

a)

Auto
{
AutoId*
AutoDescripcion
ContructorPaisId
ContructorPaisNombre
RepuestoPaisId
RepuestoPaisNombre

Pais
{
PaisId*
PaisNombre
}

Subtype group: ConstructorPaisId
ContructorPaisId subtype or PaisId
Subtype group: ConstructorPaisNombre
ContructorPaisNombre subtype of PaisNombre
Subtype group: RepuestoPaisId
RepuestoPaisId subtype or PaisId
Subtype group: RepuestoPaisNombre
RepuestoPaisNombre subtype of PaisNombre

b)

Auto
{
AutoId*
AutoDescripcion
PaisId
PaisNombre
RepuestoPaisId
RepuestoPaisNombre

Pais
{
PaisId*
PaisNombre
}

c)

Auto
{
AutoId*
AutoDescripcion
ContructorPaisId
ContructorPaisNombre
RepuestoPaisId
RepuestoPaisNombre

Pais
{
PaisId*
PaisNombre
}

Subtype group: ConstructorPais
ContructorPaisId subtype or PaisId
ContructorPaisNombre subtype of PaisNombre

Subtype group: RepuestoPais
RepuestoPaisId subtype or PaisId
RepuestoPaisNombre subtype of PaisNombre

d) Ninguna de las anteriores

2) *4 pts.* Se tiene una aplicación GeneXus para una ensambladora de Automóviles.
Dado el siguiente diseño de transacciones. Suponga que existe un único *País* ingresado en la base con PaisId = 4.
Indique que sucede si se intenta ingresar una nueva Marca con PaisId = 9.

Marca
{
MarcaId *
MarcaDescripcion
PaisId
PaisNombre
}

Pais
{
PaisId *
PaisNombre
}

a) GeneXus verifica que exista el país 9 en la tabla País. Como no existe emite un mensaje de error y no se ingresa la nueva marca.
b) Se ingresa la nueva marca sin un país asociado.
c) Se ingresa la nueva marca y automáticamente crea el país 9 con Nombre de país vacío.
d) Ninguna de las anteriores

3) *5 pts.* Se tiene una aplicación GeneXus para una ensambladora de Automóviles.
La misma cuenta con la transacción *Envío* para registrar los envíos despachados a las automotoras. La hora del envío (EnvíoHora) debe asignarse en el momento exacto en el que se ingresa el envío. Se declara entonces la siguiente regla en la transacción *Envío,* determine la opción correcta*:*

EnvioHora = now() if insert on BeforeComplete

```
Envio
{
    EnvioId*
    EnvioFecha
    EnvioHora
    AutomotoraId
    AutomotoraNom
   Auto
   {
    AutoId *
    AutoDescripcion
    AutoPrecio
   }
```

a) La regla está bien declarada porque al ejecutarse antes del COMMIT y solo en modo INSERT la hora se guarda con el valor deseado.
b) La regla está mal declarada porque se ejecuta una vez para cada registro del segundo nivel.
c) La regla está mal declarada porque si bien se ejecuta antes de COMMIT el registro del cabezal ya fue grabado en la base de datos.
d) Ninguna de las anteriores.

4) *5 pts.* Se tiene una aplicación GeneXus para una ensambladora de Automóviles.
La misma cuenta con la transacción *Envío* para registrar los envíos despachados a las automotoras. Se quiere que al trabajar con cada línea del detalle del envío (EnvioAuto), ya sea alta, baja o modificación, se ejecute el procedimiento logEnvio que recibe como parámetro el identificador del envío.

```
Envio
{
    EnvioId*
    EnvioFecha
    EnvioHora
    AutomotoraId
    AutomotoraNom
   Auto
   {
    AutoId *
    AutoDescripcion
    AutoPrecio
   }
```

a) logEnvio.call(EnvioId) ;

b) logEnvio.call(EnvioId) on AfterValidate;

c) logEnvio.call(EnvioId) on AfterComplete;

d) logEnvio.call(EnvioId) on AfterValidate level AutoId;

5) *7 pts.* Se tiene una aplicación GeneXus para una ensambladora de Automóviles.
La misma cuenta con la transacción *Envío* para registrar los envíos despachados a las automotoras. Se necesita emitir un listado de los envíos por automotora. Se listaran solo aquellas automotoras para lasque hayan envíos.
Determine cual es la implementación correcta.

Auto
{
AutoId *
AutoDescripcion
AutoPrecio
}

Automotora
{
AutomotoraId*
AutomotoraNom
}

Envio
{
EnvioId*
EnvioFecha
AutomotoraId
AutomotoraNom
Auto
{
AutoId *
AutoDescripcion
AutoPrecio
}
}

a)
```
For each order AutomotoraId
   Defined by EnvioFecha
   Print --- (AutomotoraId, AutomotoraNom)
   For each
      Print --- (EnvioId, EnvioFecha)
   Endfor
Endfor
```

b)
```
For each
   Print --- (AutomotoraId, AutomotoraNom)
   For each
      Print --- (EnvioId, EnvioFecha)
   Endfor
Endfor
```

c)
```
For each order AutomotoraId
   Print --- (AutomotoraId, AutomotoraNom)
   For each
      Print --- (EnvioId, EnvioFecha)
   Endfor
Endfor
```

d)
```
For each
   Print --- (AutomotoraId, AutomotoraNom)
   For each AutomotoraId
      Print --- (EnvioId, EnvioFecha)
   Endfor
Endfor
```

6) *7 pts.* Se tiene una aplicación GeneXus para una ensambladora de Automóviles.
La misma cuenta con la transacción *Envío* para registrar los envíos despachados a las automotoras. Dado el siguiente diseño de transacciones determine las tablas bases de los for each que aparecen a continuación.

Auto
{
AutoId *
AutoDescripcion
AutoPrecio
}

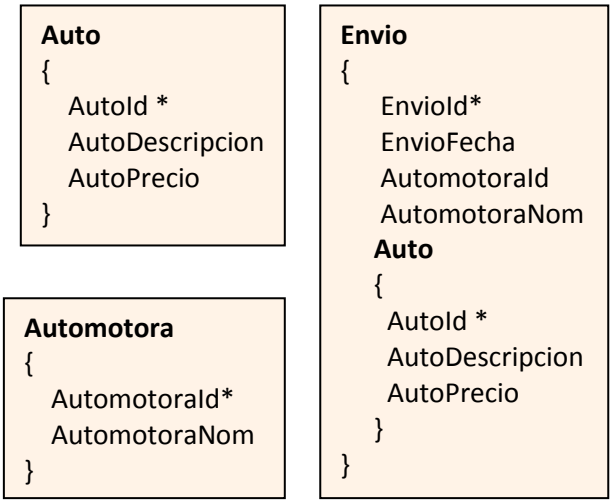
Automotora
{
AutomotoraId*
AutomotoraNom
}

Envio
{
EnvioId*
EnvioFecha
AutomotoraId
AutomotoraNom
Auto
{
AutoId *
AutoDescripcion
AutoPrecio
}
}

```
For each
   Print --- (EnvioId, EnvioFecha)
   For each
      Print --- (AutoId, AutoDescripcion)
   Endfor
Endfor
```

a) For each Externo {ENVIO}, for each interno {AUTO}
b) For each Externo {ENVIO}, for each interno {ENVIOAUTO}
c) For each Externo {ENVIO}, for each interno {ENVIO}
d) For each Externo {ENVIOAUTO }, for each interno {ENVIOAUTO}

7) *5 pts.* Se tiene una aplicación GeneXus para una ensambladora de Automóviles.
La misma cuenta con la transacción *Envío* para registrar los envíos despachados a las automotoras. Dado el siguiente diseño de transacciones determine qué tipo de for each es el siguiente:




```
For each
   Print --- (EnvioId, EnvioFecha)
   For each
      Print --- (AutomotoraId, AutomotoraNom)
   Endfor
Endfor
```

a) Producto cartesiano
b) Corte de control
c) JOIN
d) Ninguna de las anteriores

8) *3 pts.* Determine cuál de las siguientes afirmaciones sobre Data Providers es correcta.

a) El Data Provider es un objeto que me permite procesar datos para obtener información en forma estructurada (SDTs, bussines component, ext.).
b) El Data Provider es un objeto utilizado para actualizar la base de datos.
c) Ninguna de las anteriores es correcta.

9) *3 pts.* Determine cuál de las siguientes afirmaciones sobre Business component (BC) es correcta

a) Un objeto BC permite invocar una transacción desde código GeneXus como si esta estuviese siendo ejecutada desde su Web Form.
b) A través de un BC se puede actualizar la base de datos desde un Web Panel
c) Al llamar un BC, este no hace COMMIT sobre la base de datos y queda a cargo del programador la confirmación de los datos en la base de datos.
d) Todas las anteriores son correctas.

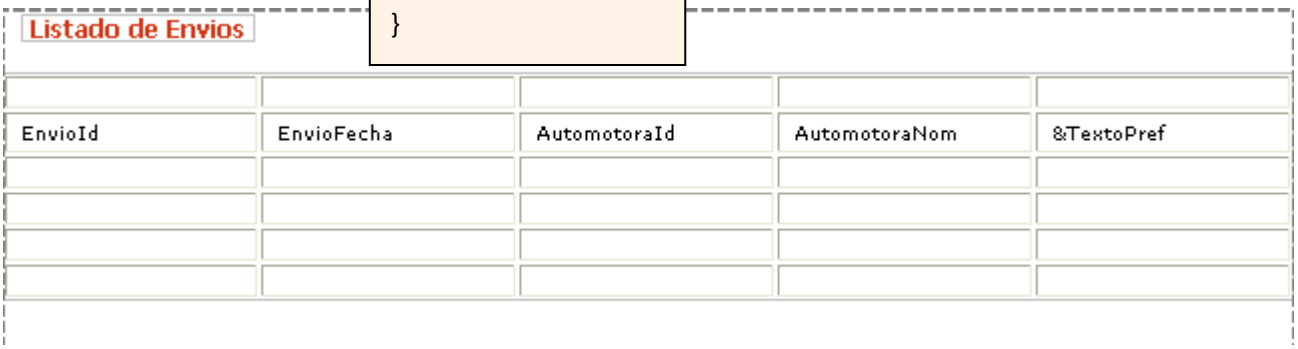
10) *6 pts.* Se tiene un Web Panel SIN TABLA BASE con una grilla para listar todos los Autos que existen en el sistema. Si se sabe que en la tabla *autos* existen 50 autos ingresados. ¿Cuántas veces se ejecuta el evento load de la grilla?

a) 1 vez
b) 10 Veces
c) 50 veces
d) Ninguna vez

11) *8 pts.* Se tiene una aplicación GeneXus para una ensambladora de Automóviles. La misma cuenta con la transacción *Envío* para registrar los envíos despachados a las automotoras. La transacción *Automotora*, que se utiliza para mantener las automotoras a las cuales la ensambladora realiza los envíos, tiene el atributo *AutomotoraEsPref* para identificar aquellas que son clientes preferenciales de la ensambladora. Se quiere construir un Web Panel que liste todos los envíos del sistema mostrando en una columna el texto 'PREFERENCIAL' para aquellos envíos a automotoras que son clientes preferenciales y el texto 'CLIENTE COMUN' para aquellas que no. Determine la opción de implementación que considere correcta.

Automotora
{
AutomotoraId *
AutomotoraNom
AutomotoraEsPref
}

Envio
{
EnvioId*
EnvioFecha
AutomotoraId
AutomotoraNom
AutomotoraEsPref
}



a)
```
Event grid1.load
    for each
        if AutomotoraEsPref = true
            &TextoPref = 'PREFERENCIAL'
        else
            &TextoPref = 'CLIENTE COMUN'
        endif
    endfor
EndEvent
```

b)
```
Event grid1.Load
    if AutomotoraEsPref = true
        &TextoPref = 'PREFERENCIAL'
    else
        &TextoPref = 'CLIENTE COMUN'
    endif
EndEvent
```

c)
```
Event Start
    if AutomotoraEsPref = true
        &TextoPref = 'PREFERENCIAL'
    else
        &TextoPref = 'CLIENTE COMUN'
    endif
EndEvent
```

2 d) Ninguna de las anteriores

12) *10 pts.* Se tiene una aplicación GeneXus para una ensambladora de Automóviles

Se ensamblan distintas marcas de autos (marca). Un *Auto* tiene una M*arca*. A su vez pueden haber muchos autos de una *marca*

Determine el diseño de transacciones que considere correcto para representar dicha realidad.

a)

Marca
{
MarcaId*
MarcaDescripcion
Auto
{
AutoId*
AutoDescripcion
}
}

b)

Auto
{
AutoId *
AutoDescripcion
}

Marca
{
MarcaId *
MarcaDescripcion
AutoId
AutoDescripcion
}

c)

Auto
{
AutoId *
AutoDescripcion
MarcaId
MarcaDescripcion
}

Marca
{
MarcaId *
MarcaDescripcion
}

d) Ninguna de las anteriores

13) *10 pts.* Se tiene una aplicación GeneXus para una ensambladora de Automóviles

Un *Auto* está compuesto por varios tipos de Piezas (PiezaTipo). Un tipo de pieza puede ser parte de muchos autos.

Determine el diseño de transacciones que considere correcto para representar dicha realidad.

a)

Auto
{
AutoId *
AutoDescripcion
}

PiezaTipo
{
PiezaTipoId *
AutoId*
PiezaTipoDesc
}

b)

Auto
{
AutoId*
AutoDescripcion
PiezaTipo
{
PiezaTipoId*
PiezaTipoDesc
}
}

PiezaTipo
{
PiezaTipoId *
PiezaTipoDesc
}

c)

Auto
{
AutoId *
AutoDescripcion
PiezaTipoId
PiezaTipoDesc
}

PiezaTipo
{
PiezaTipoId *
PiezaTipoDesc
}

d) Ninguna de las anteriores

14) *10 pts.* Se tiene una aplicación GeneXus para una ensambladora de Automóviles

Un *Auto* está compuesto por varias Piezas (Pieza). Estás piezas son utilizadas únicamente para esos A*utos* por lo que no puede haber una *Pieza* que no esté asociada a un *Auto*.

Determine el diseño de transacciones que considere correcto para representar dicha realidad.

a)

Auto
{
AutoId *
AutoDescripcion
PiezaId
PiezaNombre
}

Pieza
{
PiezaId *
PiezaNombre
}

b)

Pieza
{
PiezaId *
PiezaNombre
Auto
{
AutoId *
AutoDescripcion
}
}

Auto
{
AutoId *
AutoDescripcion
}

c)

Auto
{
AutoId*
AutoDescripcion
Pieza
{
PiezaId*
PiezaNombre
}
}

Pieza
{
PiezaId *
PiezaNombre
}

d)

Auto
{
AutoId*
AutoDescripcion
Pieza
{
PiezaId*
PiezaNombre
}
}

15) *5 pts.* Se tiene una aplicación GeneXus para una ensambladora de Automóviles.

A partir del siguiente diseño de transacciones, determine la estructura física de las tablas que GeneXus creará.

Auto
{
AutoId *
AutoDescripcion
AutoPrecio
}

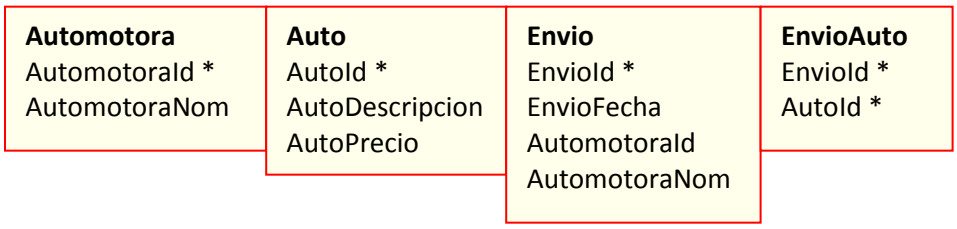
Automotora
{
AutomotoraId*
AutomotoraNom
}

Envio
{
EnvioId*
EnvioFecha
AutomotoraId
AutomotoraNom
Fx EnvioTotal → Sum(AutoPrecio)
Auto
{
AutoId *
AutoDescripcion
AutoPrecio
}
}

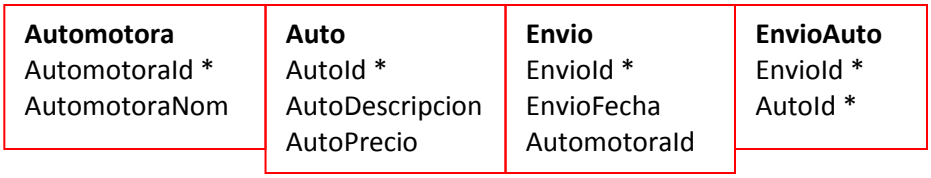
a)

Automotora	**Auto**	**Envio**	**EnvioAuto**
AutomotoraId *	AutoId *	EnvioId *	EnvioId *
AutomotoraNom	AutoDescripcion	EnvioFecha	AutoId *
	AutoPrecio	EnvioTotal	AutoDescripcion
			AutoPrecio

b)

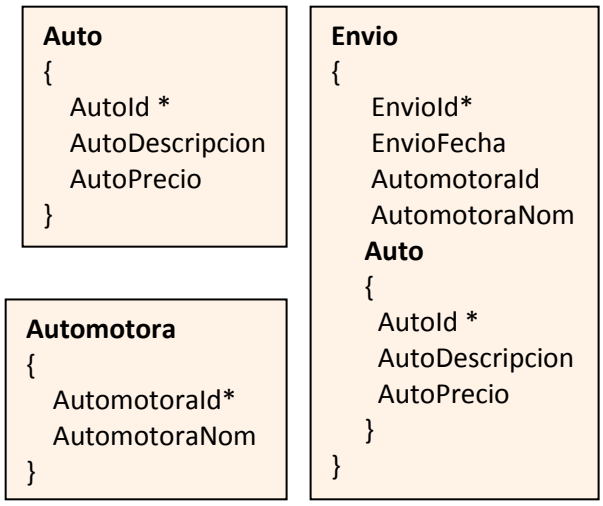


c)



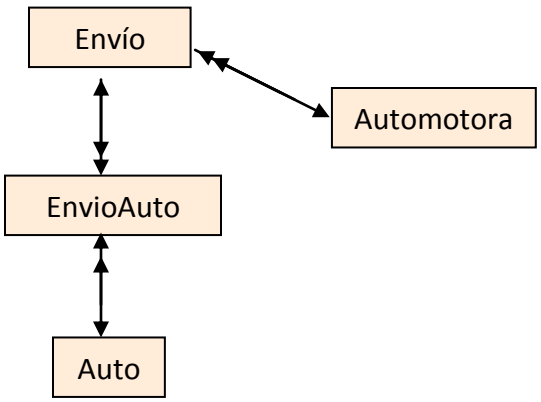
d) Ninguna de las anteriores

16) *5 pts.* Se tiene una aplicación GeneXus para una ensambladora de Automóviles.
Dado el siguiente diseño de transacciones, determine la tabla extendida de la tabla *Envío*.



a) No posee tabla extendida
b) {Envío, EnvioAuto}
c) { Envío, Automotora, EnvioAuto,Auto }
d) { Envío, Automotora}

17) *5 pts.* Se tiene una aplicación GeneXus para una ensambladora de Automóviles.
Dado el siguiente diagrama de Bachman, determine la tabla extendida de *EnvioAuto.*



a) No posee tabla extendida
b) { Envío, Automotora, EnvioAuto,Auto }
c) {Envío, Automotora}
d) { Envío, EnvioAuto}