GeneXus™

Grow thru knowledge

Ejercicios prácticos

GeneXus X Ev.3

Artech™
www.genexus.com

MONTEVIDEO - URUGUAY	Av. Italia 6201- Edif. Los Pinos, P1	(598) 2601 2082
CHICAGO - USA	1143 W Rundell PL, Suite 200	(1 312) 836 9152
SÃO PAULO - BRASIL	Rua Samuel Morse 120 Conj. 141	(5511) 55026722
CIUDAD DE MÉXICO - MÉXICO	Hegel N° 221, Piso 2	(5255) 52554733
TOKYO - JAPAN	2 27 3 Gotanda Front Nishi Gotanda, Shinagawa ku	(813) 6303 9381 (813) 6303 9980

1. El problema

Una agencia de viajes lo contrata para que desarrolle un sistema para almacenar y manipular la información con la que trabaja. Imagine que el sistema se compone de dos módulos:

- **Backend**: parte de la aplicación que deberá correr en un servidor web, de manera tal que los empleados de la agencia puedan manipular la información desde cualquier lugar con conexión a internet.
- **Aplicación sencilla para dispositivos móviles**: parte de la aplicación que será destinada para ser bajada por los clientes de la agencia de viajes, la cual les permitirá consultar las excursiones disponibles así como las principales atracciones turísticas que ofrece cada ciudad.

2. Nuevo proyecto, nueva base de conocimiento

Entrar a GeneXus y crear una base de conocimiento de nombre "TravelAgency" para comenzar el desarrollo de la aplicación.

Sugerimos:

- Elegir como ambiente de desarrollo C#. Asegúrese de tener instalado todo lo necesario (incluyendo SQL Server). Si usa GeneXus Trial, el ambiente de generación ya es predefinido, prototipando en la nube de Amazon.
- No crear la base de conocimiento en la carpeta "Mis Documentos" o cualquier otra carpeta que quede bajo "Documents and Settings", debido a que estas carpetas tienen permisos especiales otorgados por Windows.

Tómese unos minutos para **familiarizarse** con el **IDE (ambiente de desarrollo integrado de GeneXus)**. Pruebe **mover ventanas**, **visualizar ventanas específicas que desee** (View y View/Other Tool Windows) y observe detenidamente el **'Folder View'** dentro de la ventana **'Knowledge Base Navigator'**. Verá que aparecen ya inicializados **dominios**, algunos **objetos**, **imágenes**, etc.

Sugerencia: mantenga la ventana de propiedades abierta (**F4**), pues la utilizará continuamente. Dentro de la ventana 'Knowledge Base Navigator' observe la sección **'Preferences'** donde se configura el **'Environment'**.

3. Primeras transacciones

En las reuniones con la agencia de viajes, le transmiten lo siguiente:

> "Nosotros registramos los datos de nuestros clientes, y a éstos les ofrecemos viajes a distintas ciudades de distintos países, de las que registramos también las atracciones turísticas".

Para empezar a construir la aplicación, debemos empezar por identificar los actores de la realidad, y representarlos mediante **transacciones**. ¿Qué transacciones debemos crear entonces en la base de conocimiento (KB)?

Transacción "Customer"

Le preguntamos a los empleados de la agencia de viajes: ¿qué registran de sus clientes? Sabiendo que la respuesta es:

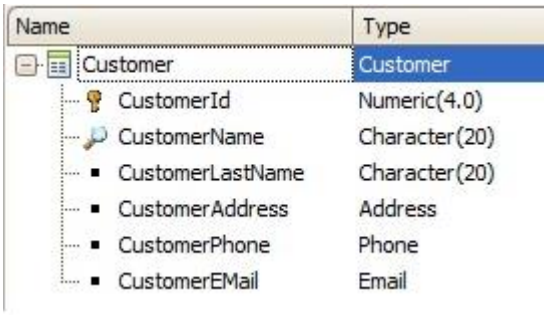
> El **nombre** (que no supera los 20 caracteres), **apellido** (que tampoco los supera), **dirección**, **teléfono** y **e-mail**.

Ya puede crear la transacción "Customer".

Recordar que:

- Para crear objetos existen varias alternativas:
  - Hacerlo desde la "Start Page".
  - Hacerlo desde el menú: File/ New Object
  - Ctrl+N
  - Contamos con un ícono
- Necesitará un atributo que identifique a cada cliente (CustomerId).
- Digitando punto (".") cuando va a ingresar un nuevo atributo, éste se inicializa con el nombre de la transacción.
- **Address, Phone** e **Email** son **dominios semánticos** que se asignan automáticamente a los atributos que se definen conteniendo en su nombre los textos Address, Phone o Email respectivamente.

La estructura de la transacción debería haberle quedado como se muestra:



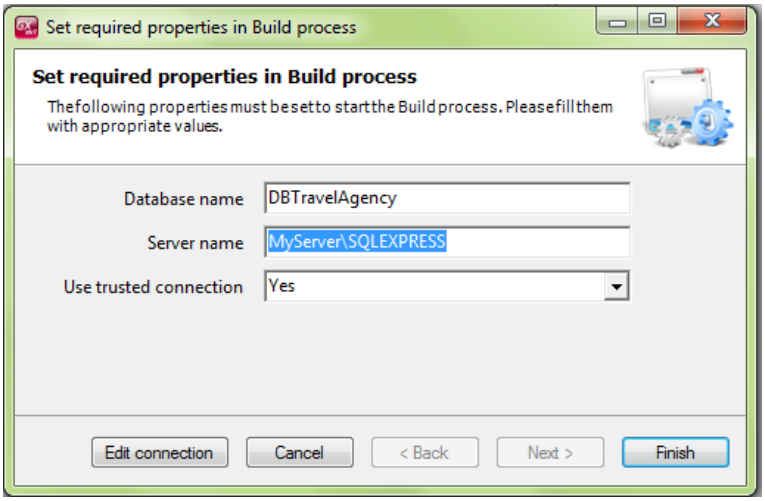
El próximo paso es probar la aplicación en ejecución. Asegúrese de tener la ventana **Output** de GeneXus habilitada y a la vista. **(View/Other Tool Windows /Output)**

Ahora sí **pruebe** la aplicación en ejecución presionando **F5**.

¿Qué sucederá?

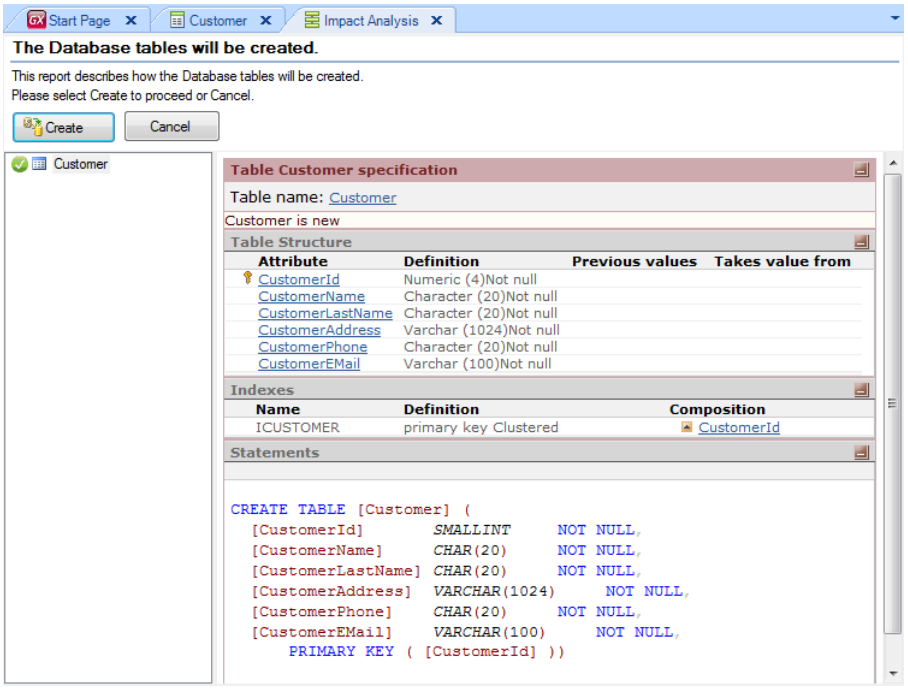
Solución

> Si decide crear la base de datos y programas localmente (esto solamente es posible con GeneXus full), se le abrirá una ventana como la siguiente para que ingrese la información de Base de Datos, Servidor y método de conexión. Recuerde que si no existe una base de datos con el nombre que indicó, en ese servidor, GeneXus la **crea**.



Si en cambio la base de datos y programas se crearán en la nube, el diálogo anterior no aparece puesto que GeneXus conoce los datos del servidor en la nube y configura automáticamente el nombre de la base datos y toda la información de conexión a la misma.

A continuación, se despliega un **Análisis de Impacto** que detalla que se creará la base de datos y la tabla CUSTOMER dentro de la misma:



Si presiona el botón **Create**, GeneXus procederá a ejecutar el programa que llevará a cabo estas creaciones. Al finalizar el proceso, se le abrirá en el navegador que tenga configurado como el predeterminado, el menú con links para ejecutar los objetos definidos. En este caso sólo uno: la transacción Customer.

Ingrese algunos clientes al sistema. **Modifique** algún dato de alguno de los clientes previamente ingresados y **elimine** algún cliente.

También pruebe usar las flechitas ofrecidas para pasar de registro en registro de clientes y entre los íconos la imagen de una lupa, ofrece una "lista de selección" para ver la lista de clientes registrados y seleccionar uno).

Ahora pasemos a identificar y crear la siguiente transacción. Recordemos lo que nos habían enunciado:

"Nosotros registramos los datos de nuestros clientes, y a éstos les ofrecemos viajes a distintas ciudades de distintos países, de las que registramos también las atracciones turísticas".

Transacciones "Attraction" y "Country", relacionadas

Vamos a crear una transacción para registrar las atracciones turísticas y una para registrar los países a los que éstas pertenecen. ¿Qué información maneja de cada atracción la agencia de viajes?

Nombre, **país**, **imagen** de la atracción, **categoría** a la que pertenece.

Ya puede crear las transacciones. Empecemos por "**Country**".

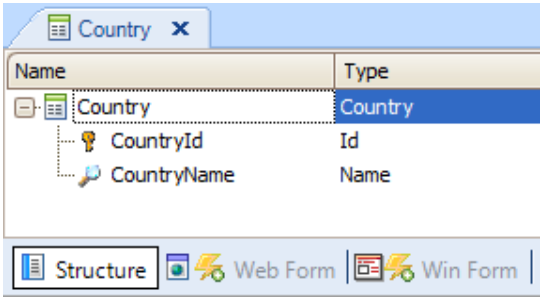
Recuerde que:
- presionando punto (".") cuando está por dar nombre a un atributo en la estructura de la transacción, aparece inicializado con el nombre de la transacción.
- necesitará un atributo identificador, CountryId.

Cuando esté definiendo el tipo de datos del atributo identificador, en vez de utilizar directamente Numeric(4.0), defina el **dominio Id** con ese tipo de datos.

Configure la propiedad **Autonumber** de ese dominio en **True**, para que todos los atributos basados en el mismo se numeren automáticamente, sin que el usuario deba preocuparse.

Observe todos los dominios que ya vienen predefinidos en GeneXus. En particular **Address**, **Email** y **Phone** que habían aparecido automáticamente cuando creó los atributos CustomerAddress, CustomerEMail y CustomerPhone en la transacción "Customer".

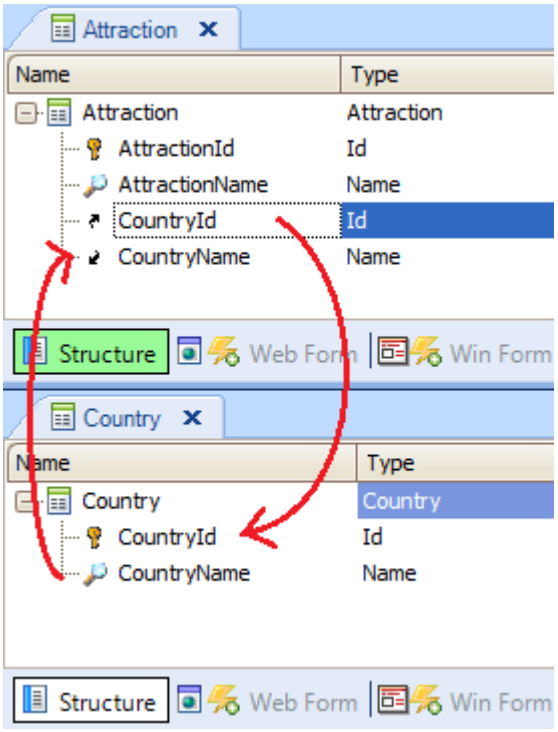
Defina el atributo **CountryName**, con tipo de datos, un nuevo dominio: **Name**=Character(50).



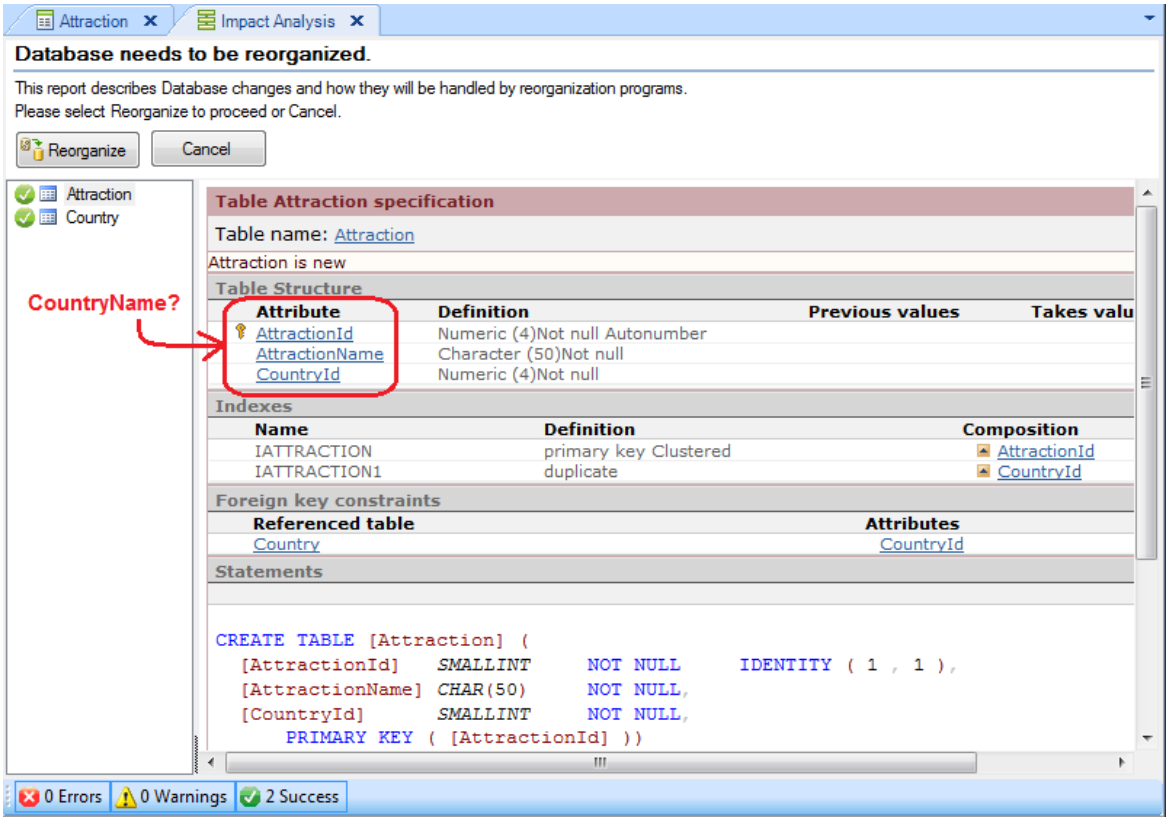
Ahora, creemos la transacción "**Attraction**". Por ahora ingrese solamente identificador, nombre y país.

Observe que al ingresar AttractionId, automáticamente asume el dominio Id. Análogamente, cuando ingrese el atributo AttractionName, asumirá automáticamente el dominio Name.

¿Por qué colocó además de CountryId, el atributo CountryName en "Attraction"?



Ejectue para probar (**F5)** y le aparecerá el siguiente reporte de Análisis de Impacto:



¿Por qué en la tabla Attraction que GeneXus informa que se debe crear en la Base de Datos, no aparece el atributo CountryName? Es decir, ¿por qué la tabla física no lo contendrá, cuando sí está en la estructura de la transacción?

Después de estudiar el reporte, si estamos de acuerdo, presionamos **Reorganize** para que efectivamente se lleve a cabo eso que se informa. Se abrirá el navegador con el menú con links a los 3 programas que corresponden a cada una de las transacciones ("Customer", "Country" y "Attraction").

Ingresar como **países** a: Brasil, Francia y China. Observar que dejando 0 como valor del identificador, al grabar se le asigna automáticamente el número posterior al último asignado (efectivamente, se está autonumerando).

Ingresar como atracción turística: "Louvre Museum", que está en Francia. Si no recuerda el identificador de Francia en el sistema, ¿cómo ingresa el país? Se le ofrece un ícono con una flecha al lado CountryId, para abrir una "Lista de selección" de países, creada automáticamente por GeneXus. Esto es porque CountryId tiene el rol de llave foránea (foreign key) en esta transacción (es decir, está "apuntando" a otra tabla).

Datos relacionados: ¿cómo se mantiene la integridad?

"Attraction" y "Country" están relacionados. Al colocar CountryId en la estructura de Attraction, por tener el mismo nombre exacto que el atributo que es llave primaria en la transacción Country, GeneXus entiende que en "Attraction" CountryId es llave foránea y mantiene automáticamente la integridad de la información. Así, por ejemplo:

- Intente ingresar una atracción con un id de país que no exista. ¿Le permite grabar la atracción turística?
- Elija una atracción previamente ingresada (por ejemplo, Louvre Museum) y cambie el país, por uno que no exista. ¿Pudo grabar la modificación?
- Intente eliminar un país (usando la transacción Country) que tenga alguna atracción asociada (por ejemplo Francia). ¿Se lo permite?

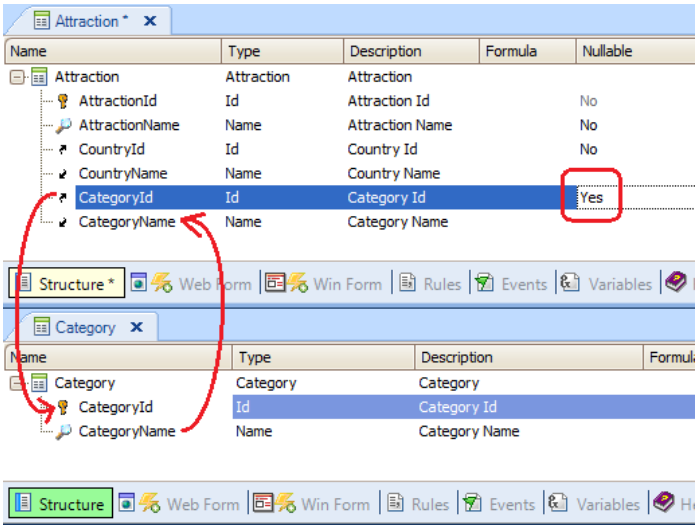
***Conclusión**: los programas correspondientes a las transacciones aseguran la integridad de los datos.*

Transacción 'Category'

Nos falta completar la información de la transacción "Attraction". Los empleados de la agencia de viajes describieron que registran de cada atracción turística, la **categoría** (monumento, museo, parque, etc.) a la que pertenece. Así que necesitaremos crear una transacción para registrar esta información, y agregar la categoría a la transacción "Attraction".

Pero además, han informado que no es obligatorio registrar indefectiblemente la categoría de atracción a la que pertenece una atracción dada que se está manipulando. Se puede dejar vacía. Si sabemos que GeneXus controla automáticamente la integridad, ¿cómo lo conseguimos?

Solución:

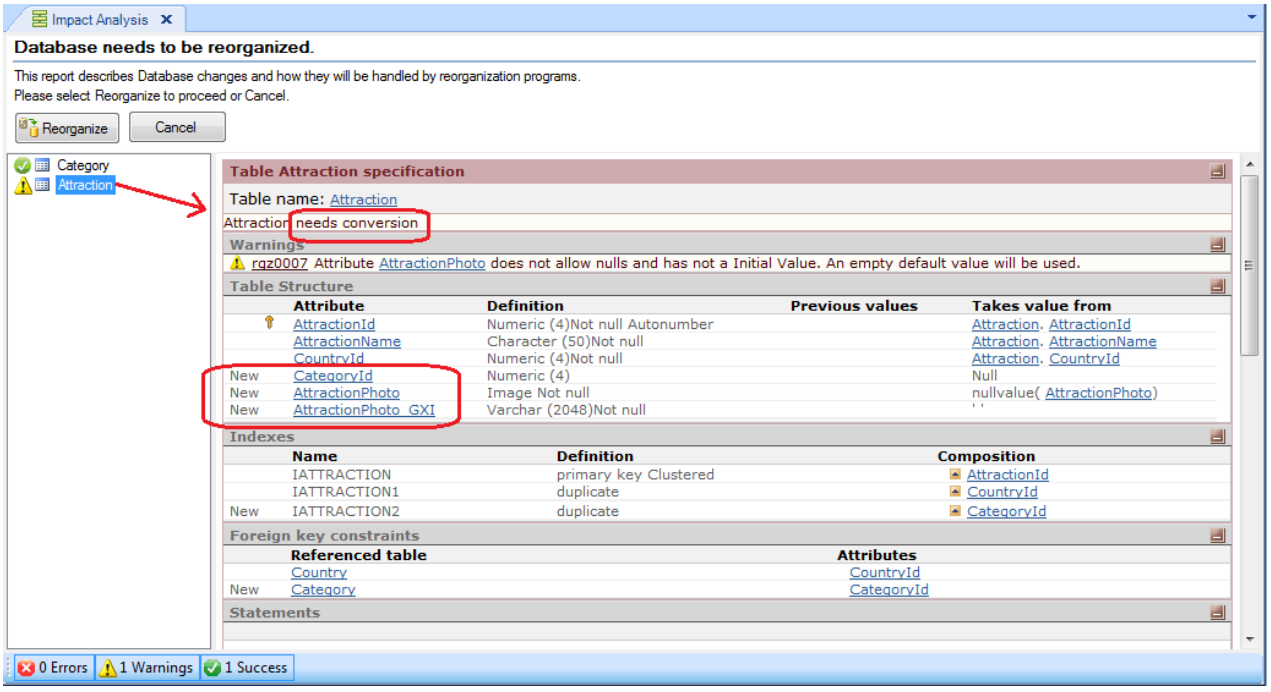


Para terminar la definición de la transacción "Attraction", agreguemos el dato que nos está faltando: la foto.

Para ello, cree el atributo **AttractionPhoto** de tipo de datos **Image**.

Pídale a GeneXus que construya la aplicación, así puede probarla en ejecución. (**F5**)

> **Observe** lo que le informa el reporte de Análisis de Impacto. Deberá crearse la tabla Category, y convertirse la tabla Attraction ya existente, agregando tres elementos de información (uno por el atributo CategoryId y dos por el atributo AttractionPhoto. No se preocupe en entender por qué requiere almacenar dos valores por imagen).

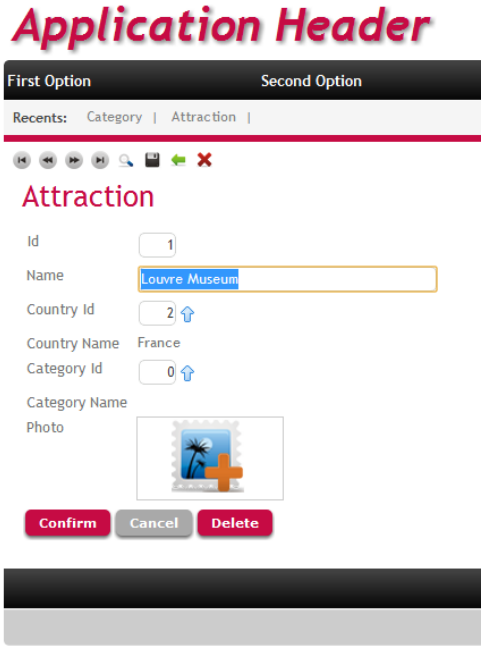


Reorganice y **ejecute**.

Ingrese categorías (como museo y monumento) y acceda a las atracciones turísticas ya ingresadas para completar su información (categoría y foto).

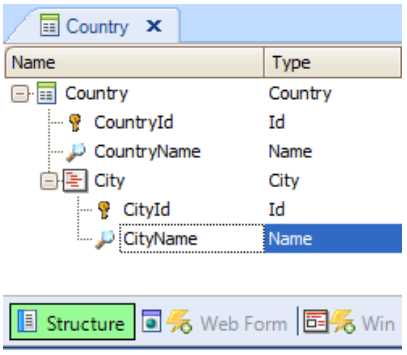
Observe que en este caso puede dejar la categoría vacía (debido a que configuró la propiedad **Nullable** en **Yes** en la estructura de la transacción).

Sin embargo, si intenta poner como valor de CategoryId para la atracción turística un valor inexistente, no le dejará grabar.



Agreguemos las ciudades a la transacción 'Country'

Además de los países con los que trabaja la agencia de viajes, necesitamos registrar la información de sus ciudades. Por tanto, debemos agregar un segundo nivel a la transacción "Country", con el identificador y el nombre de ciudad.

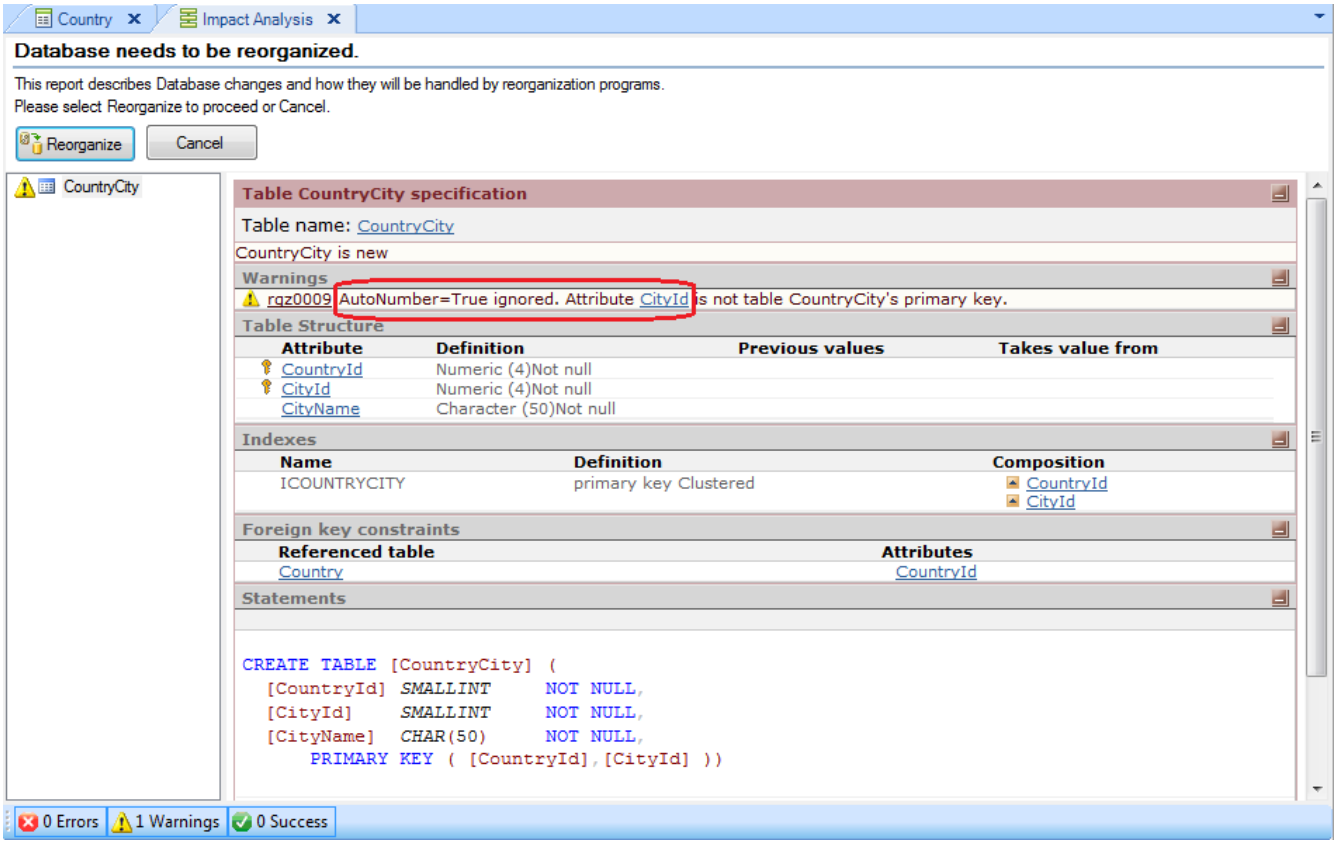


Recuerde que:

- posicionado en el atributo CountryName, con botón derecho / **Insert Level** agrega el subnivel.
- Una vez que le dé un nombre al nuevo nivel, digitando comillas (") en lugar de punto, el atributo que defina se inicializará con el nombre del nivel.
- Las ciudades se identificarán por su propio id en combinación con el del país. Es decir, no podrá identificar a una ciudad sin brindar antes la información del país del que se trata. Así, podría haber una ciudad 1 Rosario tanto para Uruguay como para Argentina:
  País: 1 (Uruguay) – Ciudad: 1 (Rosario)
  País: 2 (Argentina) – Ciudad: 1 (Rosario)

- O incluso podría ser que Rosario para Argentina se identificara con otro número:
  País: 2 (Argentina) – Ciudad: 4 (Rosario)

Reorganice y **ejecute** (**F5**).



Observe que el Listado de Navegación le informará que:

- La propiedad **autonumber** para el caso de **CityId** será ignorada. Esto significa que en ejecución el usuario deberá ingresar manualmente los identificadores de ciudad. La explicación es que la propiedad Autonumber solamente autonumera llaves primarias simples y en este caso CityId es el segundo componente de una llave compuesta.
- Se creará una nueva tabla **CountryCity** para almacenar la información correspondiente a las ciudades.

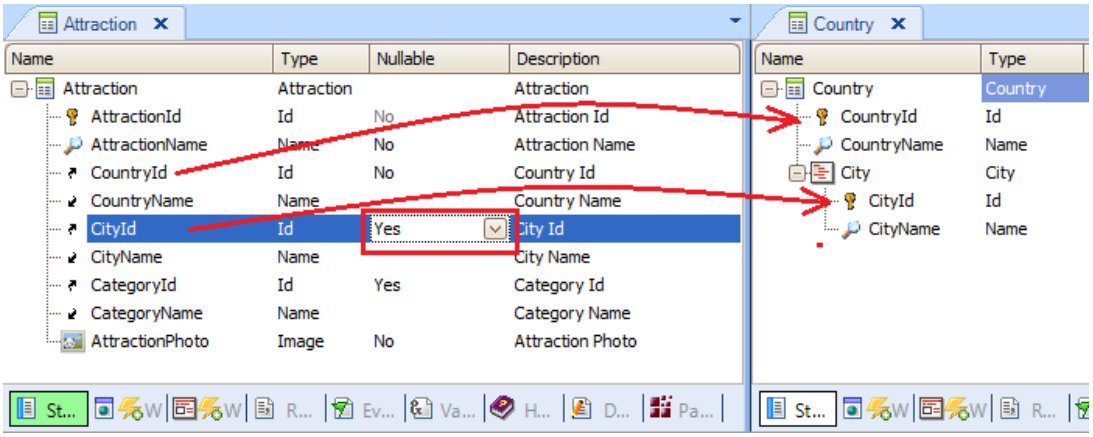
Ingrese ciudades para los países que ya tenía registrados.

Transacción "Attraction": agreguemos la ciudad.

En la transacción "Attraction" agreguemos la ciudad del país a la que la atracción pertenece. ¿Qué debe hacer si la agencia de viajes nos informa que ese valor puede no ser conocido o relevante para una atracción dada en un momento dado?

Construya la aplicación y pruébela (**F5** y **Reorganize**).

Solución



Antes de seguir, abra la transacción "Customer" y modifique el tipo de datos de **CustomerId** para que tenga dominio **Id** (y, así, se autonumere). También modifique los tipos de datos de **CustomerName** y **CustomerLastName** para que pasen a asumir el dominio **Name**. **Reorganice**.

4. Agreguemos comportamiento a las transacciones (rules)

Después de probar con nosotros la aplicación que venimos desarrollando, en la agencia de viajes nos cuentan que para los clientes hay algún comportamiento específico que debemos hacer cumplir a la hora de manipular la información a través del programa (transacción "Customer"). ¿Cuál es este comportamiento?

Nos dicen:

- "El sistema no debe permitir ingresar clientes sin nombre, ni sin apellido".
- "Debe advertirse al usuario si está dejando el teléfono sin asignar, por si fue un descuido".
- Se debe registrar la fecha de ingreso del cliente al sistema (**CustomerAddedDate**) y se debe proponer como valor predeterminado para ese atributo, la fecha de hoy.

Especifique ese comportamiento y pruébelo (**F5** y **Reorganize**).

Recuerde que:
- Las reglas finalizan con punto y coma ";".
- El método **IsEmpty()** aplicado a un atributo devuelve True cuando el atributo está vacío y False en caso contrario.
- La variable **&today** es del sistema y tiene cargado el valor de la fecha del día.

Para escribir una variable dentro de la pantalla **Rules**, cuando digita "&" se le despliegan todas las variables definidas hasta el momento para que seleccione la que necesita. La otra posibilidad es utilizar **Insert** / **Variable.**

Pruebe ingresar un nuevo cliente dejando vacío el nombre. ¿Le permite grabar o pasar al siguiente campo?

Ídem con el apellido. ¿Sucede lo mismo con el teléfono?

Si luego le informan que la fecha de ingreso al sistema no debería ser manipulada por el usuario, sino únicamente visualizada, ¿cómo establece este comportamiento?

Especifíquelo y pruébelo en ejecución.

5. Patterns: mejorando la interfaz para trabajar con la información

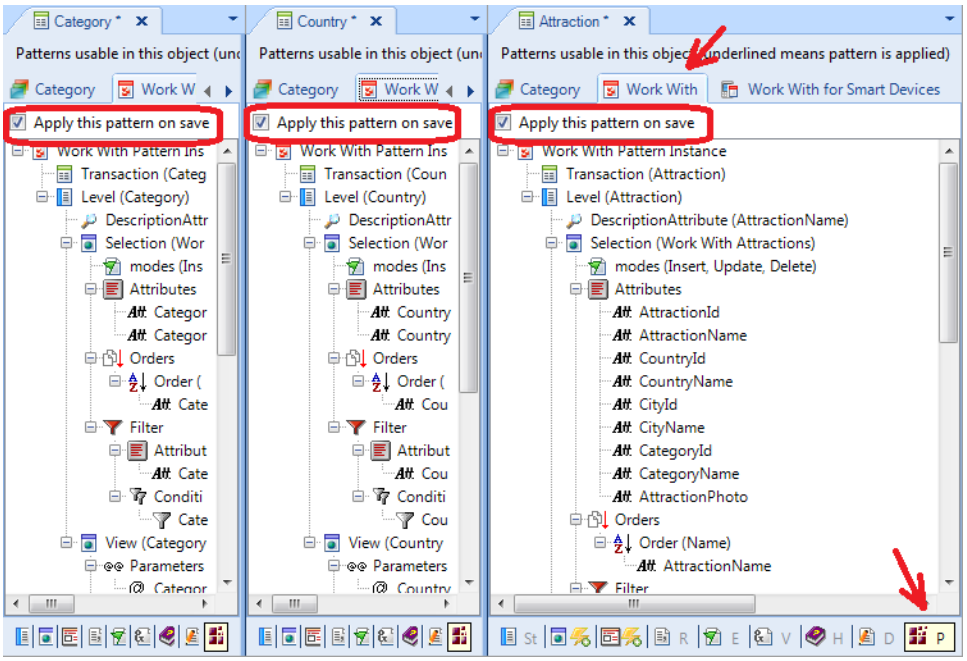
Al mostrarle al cliente lo realizado hasta ahora, nos dice que quisiera poder manipular la información de países, categorías y atracciones turísticas de un modo más potente y vistoso (que ofrezca consulta, posibilidad de filtrar, así como insertar, modificar y eliminar datos, etc.).

Para ello deberá aplicar los patrones Work With a las tres transacciones. Pruébelo y véalo en ejecución.

Observar que:

- existe un Work With para Smart Devices, también. Pero el que usted deberá aplicar es el que corresponde a la aplicación web que está constuyendo.
- GeneXus creará automáticamente varios objetos por transacción, para implementar el "Trabajar con" esa entidad.

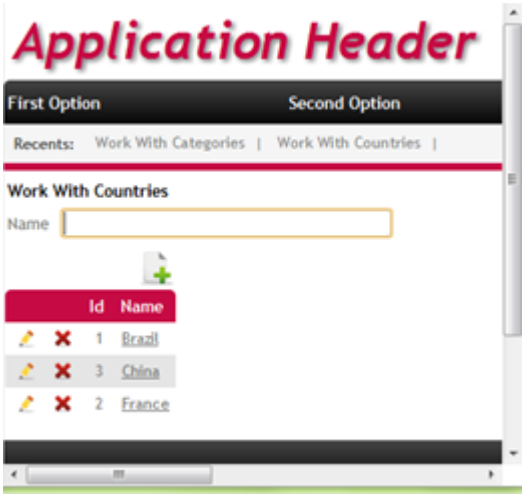
Solución



¿Por qué no aparecen más en el Developer Menu las transacciones Category, Country y Attraction?

Pruebe:

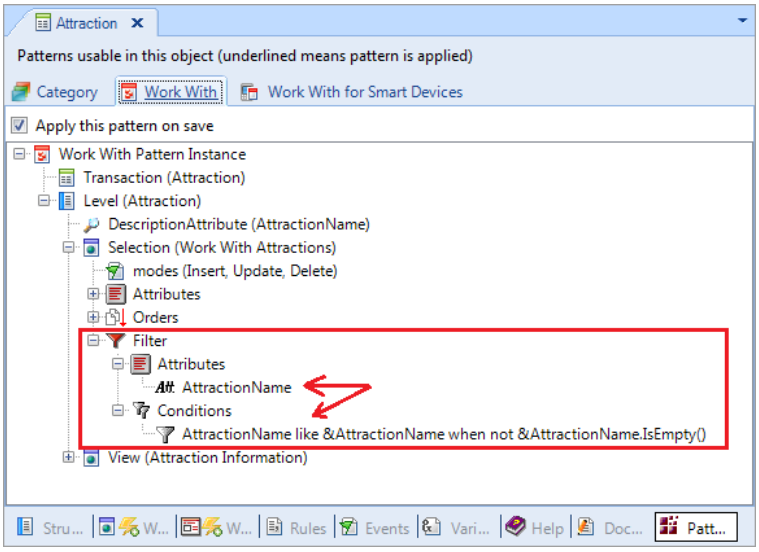
1. Ingresar un nuevo país.
2. Modificar un país existente (por ejemplo, agregándole una ciudad).
3. Eliminar un país existente.
4. Visualizar la información de un país.
5. Realizar una búsqueda por nombre de país.



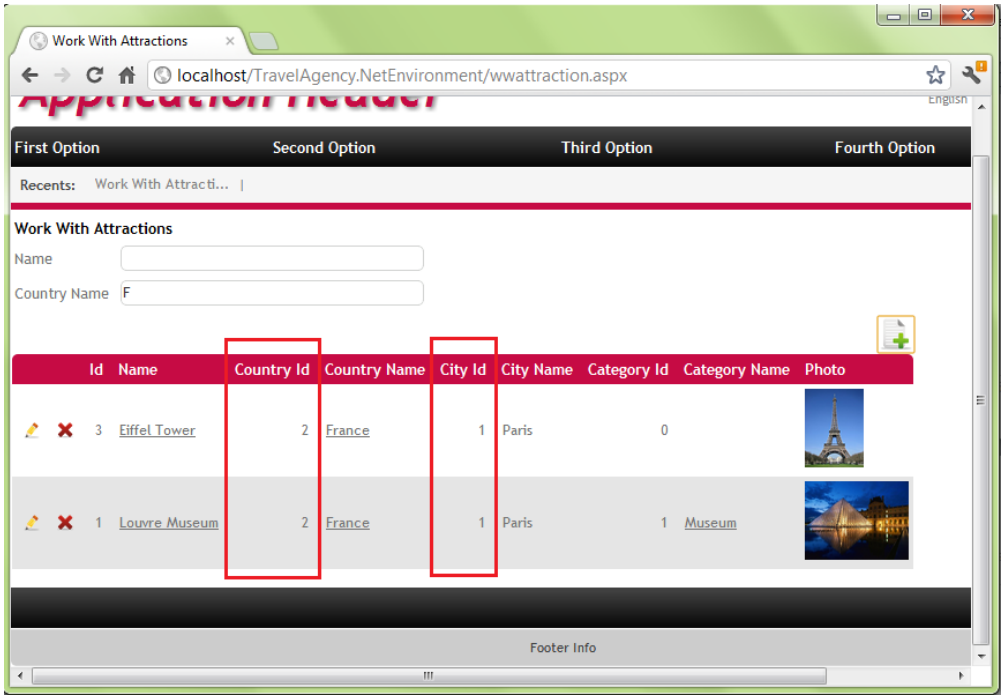
6. Ingrese un par de atracciones turísticas (Ej: La muralla China, de China/Beijing, Eiffel Tower de Francia/París)
7. Filtre las atracciones turísticas cuyo nombre empiece con F. ¿Y si ahora quiere poder visualizar todas las atracciones turísticas de Francia? No está incluida esta posibilidad, por lo que deberemos personalizar el pattern work with de esta transacción, para agregársela. Hágalo **en GeneXus** y **pruebe** en **ejecución**.

Solución

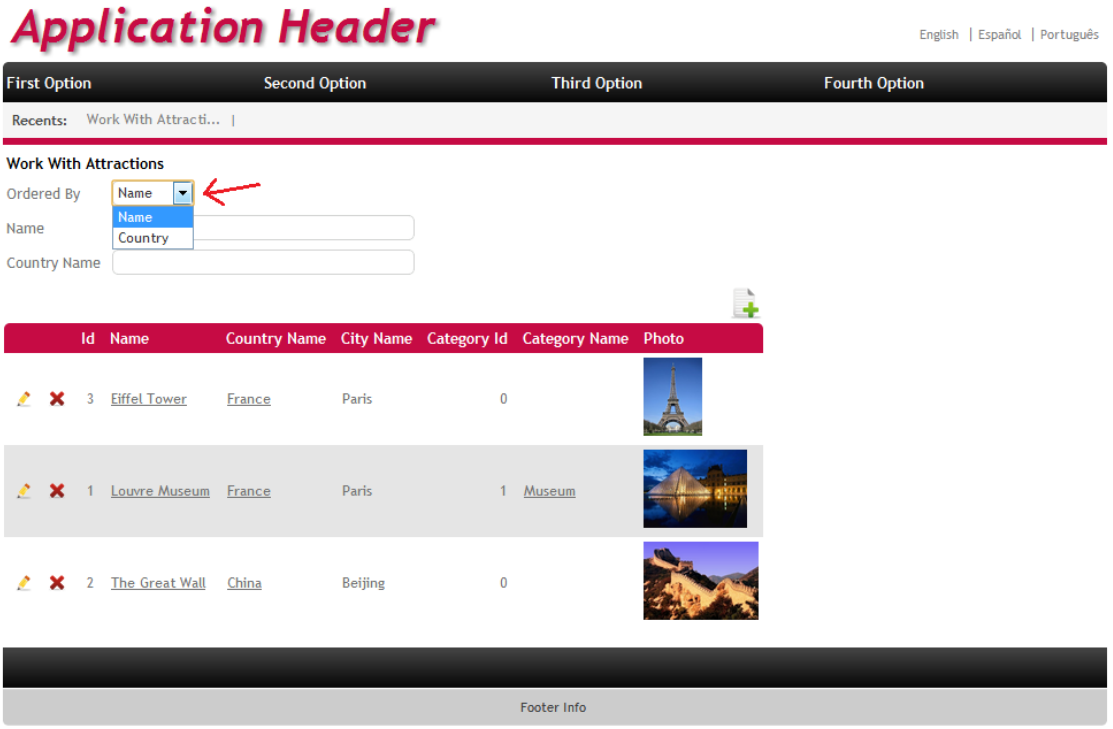
Para ello primero observe cómo está especificado el filtro que sí existe, por nombre de la atracción turística:



8. Ahora quite los identificadores de país y ciudad de la pantalla del Work With y pruébelo en ejecución.



9. Si ahora quiere brindar la posibilidad de que el usuario elija si quiere ver las atracciones ordenadas por nombre de la atracción o por nombre de país, **impleméntelo** y **pruebe**.



6. Transacciones "Flight" y "Airport" y necesidad de definir subtipos

Se necesita ahora registrar los vuelos que la agencia de viajes ofrece. Cada vuelo tiene un identificador y va de un aeropuerto a otro. También cada vuelo tiene un precio. Para el Precio cree un dominio Numeric(10.0).

Cree una transacción para registrar aeropuertos. Cada aeropuerto tiene un identificador, un nombre, un país y una ciudad en la cual se encuentra.

¿Cómo se define que cada vuelo tiene un aeropuerto de origen y otro aeropuerto de destino?

Recuerde

1) Que en la estructura de la transacción:
- un ícono representando una flecha hacia arriba ↗ informa que el atributo es clave foránea (Foreign Key), es decir que apunta a otra tabla.
- Un ícono representado una flecha hacia abajo ↙ informa que el atributo es inferido de otra tabla.
- Un ícono representando una S indica que el atributo es un subitpo

2) Sobre los grupos de subtipos:
- Se definen de la misma manera que cualquier tipo de objeto.
- Cada grupo de subtipos debe contener obligatoriamente un subtipo de un atributo primario (que es llave primaria de una tabla) o conjunto de atributos que forman una llave primaria.
- En cada grupo de subtipos, hay que incluir todos los atributos subtipos que se necesiten conocer, pertenecientes a la tabla base y/o extendida de la clave primaria del grupo.

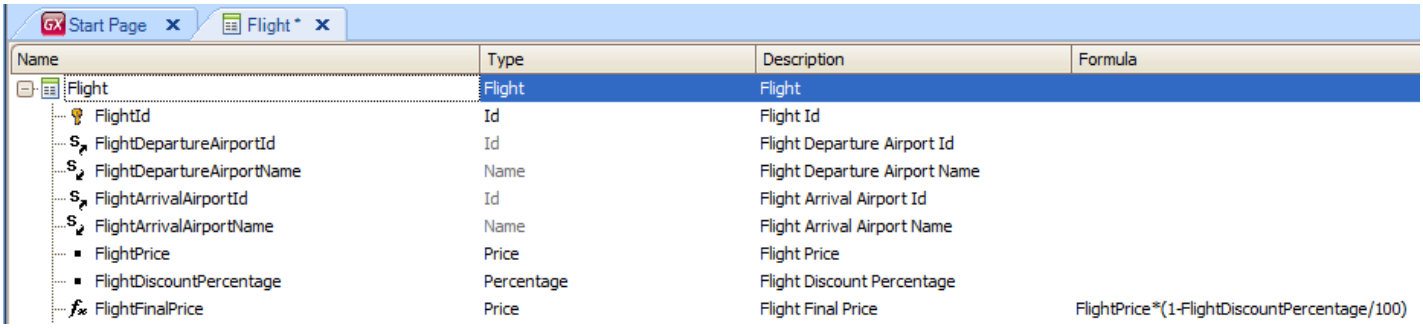
Ejecute y **verifique** que al intentar ingresar un vuelo, se dispare un error si el aeropuerto de partida que está queriendo asignarle al vuelo no existe. Ídem para el aeropuerto de arribo.

No debe permitirse ingresar un vuelo cuyo aeropuerto de partida coincida con el aeropuerto de arribo. Implemente ese comportamiento y pruébelo en ejecución.

7. Fórmulas

Se necesita poder registrar el **el descuento actual que tiene cada vuelo.** Defina un nuevo atributo en la transacción Flight **para almacenar este dato.** Darle al nuevo atributo el nombre: *FlightDiscountPercentage* y que su tipo de datos sea un dominio "Percentage", numérico de largo 3.

Se desea visualizar el precio final del vuelo con el descuento aplicado. Para resolver esto, defina otro atributo más, de nombre *FlightFinalPrice*, que sea **fórmula global** que calcule automáticamente el precio final del vuelo:



Presione F5, **observe en el Análisis de Impacto cuál atributo se creará físicamente y cuál no,** reorganice y pruebe la aplicación en funcionamiento.

Cree un segundo nivel en la transacción "Flight" de nombre: Seat, para registrar sus asientos.

Dado que los asientos se suelen identificar por un número y una letra, la llave primaria (o identificador único) del segundo nivel, deberá componerse de 2 atributos:

- *FlightSeatId:* para registrar el nro de asiento
- *FlightSeatChar:* para ingresar la letra

Defina el dominio SeatChar, de tipo: character(1) con el objetivo que el atributo *FlightSeatChar* pertenezca al mismo. Restrinja las letras posibles para el dominio: que sean válidas las que están entre la A a la F (editando la propiedad **Enum Values** del mismo).

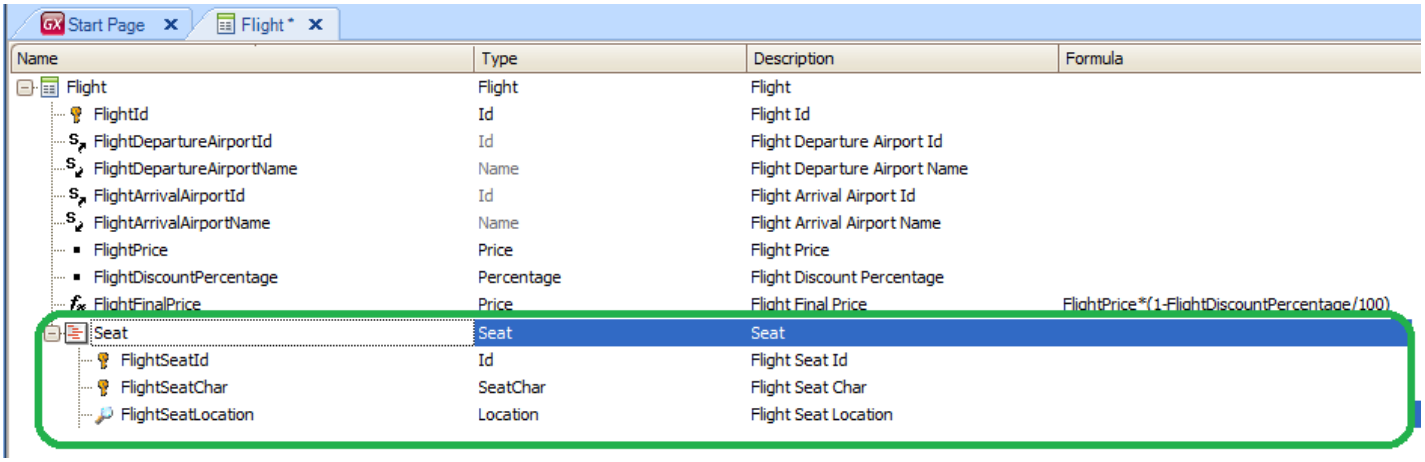
Crear otro atributo en el 2do nivel de la transacción Flight de nombre: FlightSeatLocation. Definir un dominio "Location" asociado al mismo, de tipo carácter(1). En el nodo Domains, editar la propiedad Enum Values del dominio "Location" y definirle los siguiente 3 valores:

Window (valor que se almacenará: "W")

Middle (valor que se almacenará: "M")

Aisle (valor que se almacenará: "A")

Si observa el form de la transacción Flight, verá que para cada asiento podrá indicar la ubicación del mismo mediante un control combo que ofrecerá los valores "Window", "Middle" o "Aisle".



Recuerde que para definir que cierto atributo es parte de la llave primaria, debe presionar el botón derecho del mouse sobre el atributo y el menú contextual le ofrecerá la opción **Toggle Key**.

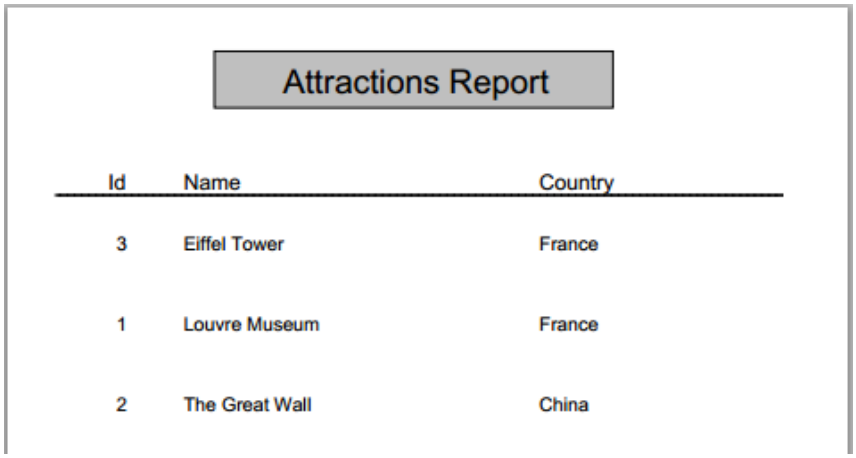
A efectos de conocer la capacidad de pasajeros que el vuelo permite, cree un atributo nuevo en el primer nivel de la transacción Flight, de nombre: *FlightCapacity*, tipo: numéric(4) y defínalo fórmula global que cuente la cantidad de asientos que ofrece el vuelo.

Se desea ver la capacidad del vuelo en el formulario web y controlar que no se registre ningún vuelo con menos de 4 asientos. Este control deberá realizarse cuando terminen de ingresar todos los asientos y luego de haber presionado el botón Confirm.

8. Listados pdf

Ahora supongamos que como parte de la aplicación, deberá implementarse la posibilidad de que a pedido del usuario, se le desplieguen listados pdf con la información requerida. Por ejemplo, suponga que se necesita un listado que muestre en orden alfabético, las atracciones turísticas almacenadas en la base de datos.

Si sabe que debe lucir más o menos como sigue:



Attractions Report

Id	Name	Country
3	Eiffel Tower	France
1	Louvre Museum	France
2	The Great Wall	China

Impleméntelo en GeneXus.

Recordar que para poder visualizar directamente desde el browser un listado, el mismo debe ser generado como PDF. Para esto debe configurar las siguientes propiedades del objeto procedimiento:

- Main program = 'True'
- Call Protocol = http
- Report output = Only to File

Y la siguiente regla:

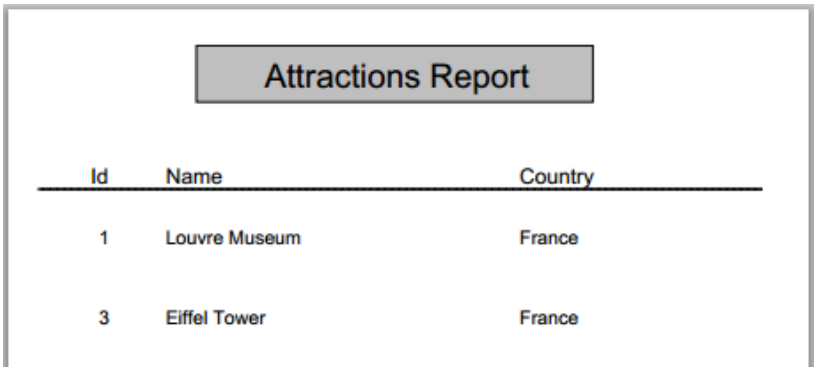
- Output_file('nombre-archivo.pdf' , 'PDF')

Para ejecutar el listado, sobre la pestaña del objeto, botón derecho / **Run with this only**

¿Reparó en lo que le informa el listado de navegación del procedimiento?

¿Y si ahora se necesita que el listado salga ordenado por nombre de país? Impleméntelo, observe lo informado en el listado de navegación y pruébelo.

¿Y si ahora necesita solamente listas las atracciones de Francia? Pruébelo (observando listado de navegación)

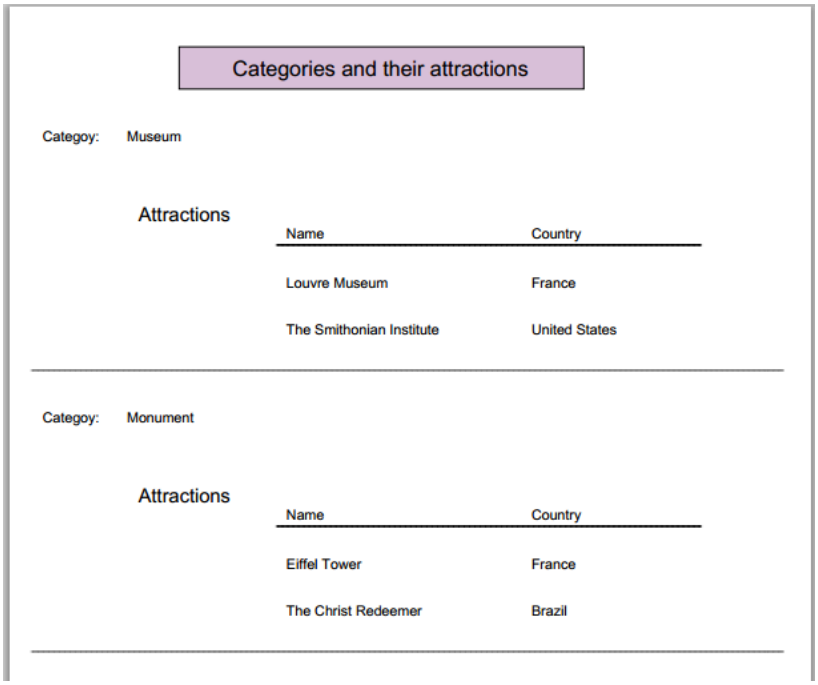


Attractions Report

Id	Name	Country
1	Louvre Museum	France
3	Eiffel Tower	France

En cada caso, deduzca qué tabla de la base de datos se está recorriendo para realizar la consulta del for each.

También se necesita un listado como el que sigue (que muestre cada categoría, y por cada una de ellas, sus atracciones turísticas). Impleméntelo y pruebe lo realizado.



Agregue una nueva categoría al sistema, por ejemplo "Art Gallery" (galería de arte). Vuelva a ejecutar el listado anterior. ¿Salió listada la categoría?

Modifique el listado anterior para que no salgan listadas categorías que no tengan atracciones turísticas relacionadas.

¿Qué modificaciones encontró en el listado de navegación?

Otra solicitud de la agencia de viajes es un listado que muestre todos los nombres de países y para cada país, **la cantidad de atracciones turísticas que ofrece**:

Y nos pide otro listado también que muestre todos los países que tienen **más de 2 atracciones para visitar**:

9. Business Components

Realizaremos algunas operaciones sobre la base de datos, a través de business components.

Aumento de precio de los vuelos

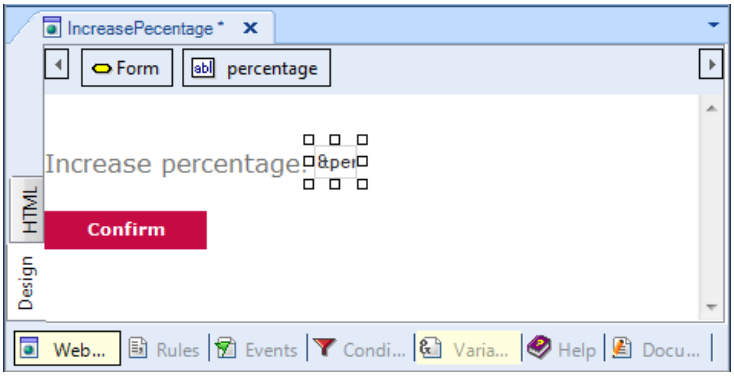
Cada tanto la agencia de viajes necesita incrementar los precios de los vuelos en un porcentaje determinado. Para ello, deberemos implementar una **pantalla** que permita al usuario especificar ese **porcentaje de aumento**, y dar la **orden** de aplicarlo a todos los vuelos de la base de datos. **Impleméntelo** y **pruebe**.

Recuerde que:

- El objeto web panel le permite implementar pantallas flexibles para entrada y salida de información.
- Para entrada de información (que el usuario pueda ingresar valores a la pantalla), se insertan controles variable en el form.
- Si quiere editar barras de menú, posicionándose en GeneXus arriba, sobre la barra, con botón derecho podrá insertar, por ejemplo, la barra Formatting.
- Para que el web panel tome alguna acción determinada, pueden insertarse botones y programar el "evento" asociado.
- Los business components son tipos de datos que se crean al configurar la propiedad de nombre Business Component del objeto transacción con valor Yes. Al hacerlo, para insertar, modificar o eliminar registros de las tablas correspondientes, podrá utilizarse, además de la transacción, una variable de tipo de datos Business Component en cualquier otro objeto (por ejemplo, un web panel) y realizar las mismas operaciones a través de los métodos Load(), Save() y Delete().
- Para que las operaciones realizadas a través del business component queden efectuadas de manera permanente, deberá ejecutar a continuación el comando Commit.
- Si a un valor X se le quiere incrementar un 20%, alcanza con hacer X = X*(1+20/100) = X*1,20

Solución

Una solución posible (intente implementar ud. lo pedido sin mirar lo que sigue): creamos un Web panel, con una variable &percentage, de tipo de datos: Numeric(3.0)



Al hacer doble clic sobre el botón, nos lleva a la sección Events, editando el evento Enter, asociado al mismo.

Allí programaremos la lógica que queremos se ejecute cuando el usuario presione el botón.

Necesitamos recorrer todos los vuelos y sobreescribir el precio que tenían, aumentándolo en ese porcentaje indicado por el usuario en la variable de pantalla.

Para recorrer todos los vuelos de la base de datos, usamos el comando for each.

¿Y para cada vuelo, cómo lo editamos para modificarle el valor de FlightPrice?

Necesitaremos una variable para ello, que tenga toda la estructura de la transacción Flight, y además nos permita grabar en la base de datos. ¿Qué tipo de datos tendrá, entonces, esa variable? El **business component Flight** (observar que el tipo de datos business component tiene el mismo nombre que la transacción). Pero GeneXus no crea ese tipo de datos automáticamente para cada transacción. Se lo tenemos que pedir. ¿Cómo lo hacíamos? Prendiendo la propiedad correspondiente de la transacción.

Una vez hecho esto, entonces:

```
Event Enter
 for each Flight
       &flight.Load(FlightId)
       &flight.FlightPrice = &flight.FlightPrice*(1+&percentage/100)
       &flight.Save()
       ...
 endfor

EndEvent
```

Como las operaciones de grabación o eliminación de la base de datos (a través de los métodos Save() y Delete()) pueden provocar errores, es importante saber qué ocurrió. Para ello tenemos el método Success() que devolverá True si no hubo errores, y False en caso contrario.

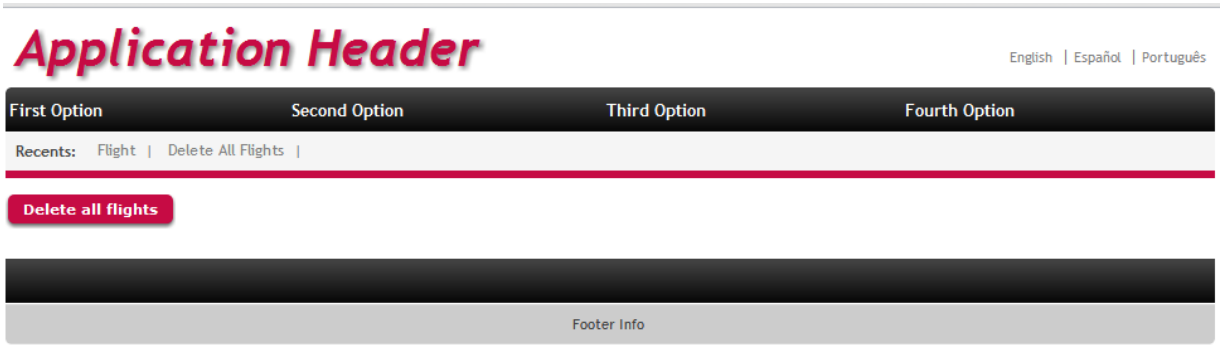
```
Event Enter
  for each Flight
  &flight.Load(FlightId)
  &flight.FlightPrice = &flight.FlightPrice*(1+&percentage/100)
  &flight.Save()
  If &flight.Success()
        Commit
  else
        Rollback
  endif
  endfor

EndEvent
```

Al insertar, modificar o eliminar registros de una tabla de la base de datos, mientras que no se diga: "todo lo hecho que quede permanente", esas modificaciones serán provisorias. ¿Qué quiere decir? Que por ejemplo, si hay un apagón, esas modificaciones se perderán. La forma de decir "lo hecho, que quede permanente" es

ejecutar el comando Commit. Si por el contrario, queremos que lo que hicimos se deshaga, ejecutamos el comando Rollback.

Pantalla para eliminación de todos los vuelos

Grabe el web panel anterior con otro nombre (para ello alcanza con posicionarse sobre la pestaña y con botón derecho, hacer **Save as**) y modifique el Form para que sólo contenga un botón con el texto "Delete all flights", de modo que en ejecución este web panel se vea así:



Cuando el usuario presione el botón, deberán eliminarse todos los vuelos de la base de datos.

¿Qué debe modificar del evento Enter que tenía programado?

Nota: si se posiciona sobre el botón y ve sus propiedades, en la de nombre Caption puede modificar el texto del botón.

Solución

Alcanzará con :

```
Event Enter
	for each Flight
		&flight.Load(FlightId)
		&flight.Delete()
		if &flight.Success()
			Commit
			Msg( 'Successful deletion')
		else
			Msg( 'Deletion could not be done')
			Rollback
		endif
	endfor
EndEvent
```

Con Msg logrará que se despliegue ese mensaje en la pantalla del web panel.

10. Procedimientos para actualizar registros

Aumento de precios de los vuelos

Suponga que los vuelos a los que debe aumentar el precio en un porcentaje dado (vuelva al primer ejercicio del punto 9 del práctico) son miles. Sabiendo que el aumento de precio es un procedimiento sencillo que no hará fallar la integridad de ningún modo, ejercite resolverlo con un procedimiento, sin utilizar business components.

Recuerde que:

- Con el comando for each dentro de un procedimiento, puede actualizar los atributos de su tabla extendida a través de simples asignaciones.
- La actualización "directa" a través de procedimientos no controla la integridad de la información.
- Todo objeto debe declarar los parámetros que recibe y los parámetros que devuelve. Si no los declara, ni recibirá ni devolverá valores.
- Los parámetros se declaran a través de la regla **parm**.
- Las variables son locales al objeto donde se utilizan. Esto significa que si quiero recibir un valor como parámetro en una variable &X, debo declararla en el objeto.

Eliminación de todos los vuelos

¿Y si ahora quisiera eliminar todos los vuelos, tal como lo hizo en el punto 9 del práctico, pero a través de un procedimiento?

Solución

Crear un procedimiento FlightsDeletion que no recibe parámetros, con el siguiente código:

```
for each Flight
   for each Flight.Seat
      delete
   endfor
   delete
endfor
```

Haga un "Save as" del web panel que tenía implementado en el punto 9 (el que hacía la eliminación a través de Business Component), y cambie el evento Enter:

```
Event Enter
  FlightsDeletion()
EndEvent
```

¿Y si ahora quiere borrar toda la información de la base de datos?

Solución

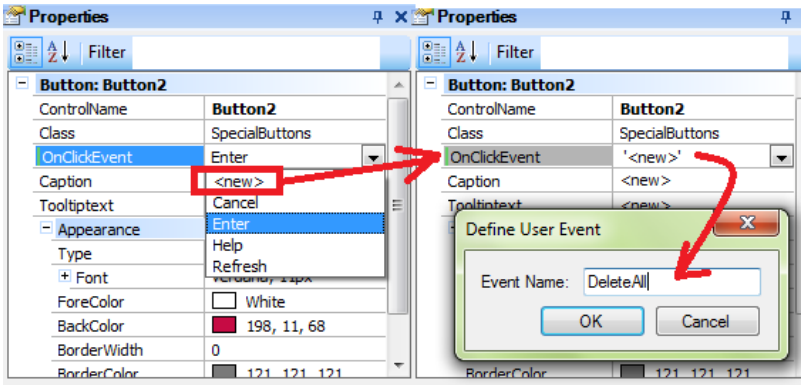
Crear un procedimiento DeleteAll con Source:

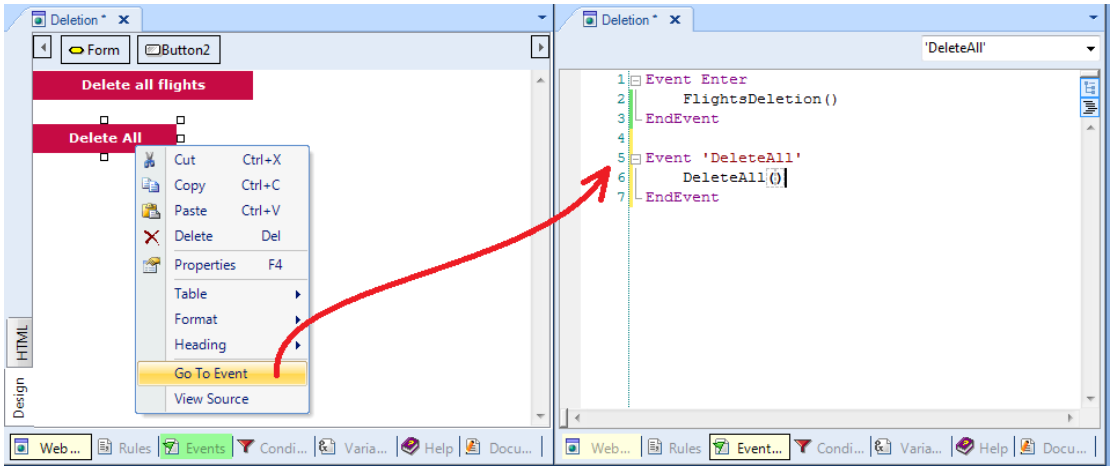
```
for each Flight
   for each Flight.Seat
      delete
   endfor
   delete
endfor

for each Airport
  delete
endfor

for each Attraction
  delete
endfor

for each Category
  delete
endfor

for each Customer
  delete
endfor

for each Country
  for each Country.City
    delete
  endfor
  delete
endfor
```

Recuerde que los procedimientos no validan la consistencia de los datos, pero la base de datos sí lo hace. Es decir, la base de datos valida la consistencia de los datos interrelacionados, por lo tanto el orden en el que ud. intente eliminar los datos, es importante. Por ejemplo, si intenta eliminar los países antes que las atracciones, la base de datos lo impedirá, el programa cancelará y no resultará amigable para el usuario.

Y desde un web panel con un botón, al cual le asociaremos un "**evento de usuario**", haremos la llamada al procedimiento.

Editando las propiedades del botón, en OnClickEvent elegimos <new> y le damos el nombre que queramos





Luego presionando el botón derecho del mouse sobre el botón y seleccionando **Go To Event**, nos posicionamos en el evento asociado al botón y le definimos dentro, la invocación al procedimiento.

Inicialización de la información de la base de datos [opcional]

Cuando la aplicación que usted está desarrollando se ponga en producción (es decir, empiece a ser utilizada por la agencia de viajes) deberán cargarse todos los datos de clientes, países, categorías, atracciones, vuelos. Escriba un procedimiento que inicialice esas tablas con información que nos ha brindado la agencia de viajes y pruébelo.

Tener en cuenta que:

- El comando new ingresa **un** registro en **una** tabla física.
- Si no asigna valor a un atributo de la tabla, entonces:
  - Si el atributo es autonumerado, al grabarse el registro en la base de datos (al alcanzarse el "endnew"), en memoria quedará para ese atributo el valor dado por la base de datos al mismo. Por ejemplo, si se está grabando una categoría:

    New
    CategoryName = 'Monument'
    Endnew

    Sabemos que luego del Endnew si utilizamos el atributo CategoryId éste tendrá el valor que la base de datos le dio al registro al ingresarlo.
- Hay que insertar la imagen de la Torre Eiffel en la KB para poder utilizarla. Una forma de hacerlo es en la ventana Folder View ir al folder Customization y eligiendo Images, donde se listan todas las imágenes actualmente almacenadas en la KB, insertar una nueva (desde un archivo) y darle un nombre.

Solución

Crear un procedimiento DBInitialize con el Source que sigue e invocarlo desde un web panel:

```
new
       CustomerName = 'John'
       CustomerLastName = 'Cage'
       CustomerEMail = 'jcage@gmail.com'
       CustomerAddress = '11235 NE 4 Ave, Miami'
endnew
//Brazil
```

```
new
	CountryName = 'Brazil'
endnew
new
	CountryId = CountryId
	CityId = 1
	CityName = 'Rio de Janeiro'
endnew

//France
new
	CountryName = 'France'
endnew
new
	CountryId = CountryId
	CityId = 1
	CityName = 'Paris'
endnew
new
	CategoryName = 'Monument'
endnew
new
	AttractionName = 'Eiffel Tower'
	CountryId = CountryId
	CityId = CityId
	CategoryId = CategoryId
	AttractionPhoto = EiffelTower.Link()
endnew

new
	CountryId = CountryId
	CityId = 2
	CityName = 'Niza'
endnew

//China
new
	CountryName = 'China'
endnew
new
	CountryId = CountryId
	CityId = 1
	CityName = 'Beijing'
endnew
new
	CountryId = CountryId
	CityId = 2
	CityName = 'Shangai'

endnew
```

11. Pasaje de parámetros

Muchas veces necesitamos que un objeto reciba parámetros para que, en base a los mismos, ejecute su lógica. Por ejemplo, una cosa es realizar un listado pdf de todas las atracciones turísticas de la base de datos, y otra es realizar un listado de aquellas cuyo nombre se encuentra dentro de un rango dado.

Listado de atracciones en un rango determinado

Grabe con otro nombre el procedimiento que había creado en el punto 6 para listar las atracciones turísticas, y modifíquelo para que ahora solamente liste aquellas cuyo nombre se encuentra dentro de un rango recibo por parámetro (el valor inicial y el final del rango serán los parámetros recibidos).

Implemente una pantalla que pida los valores de ese rango al usuario, e invoque a este objeto, pasándole esos valores por parámetro. Pruebe en ejecución.

Recuerde que:

- Si define una variable basada en (**based on**) un atributo, quedará ligada al tipo de datos del atributo, es decir, si éste se modifica, el de la variable también, acorde al cambio.
- Las variables utilizadas para realizar una invocación en el objeto que llama y las utilizadas para declarar los parámetros que se reciben en el objeto llamado, no tienen por qué coincidir en nombre, pero sí deben tener tipos de datos compatibles.

Solución:

Al procedimiento (llamémosle "AttractionsFromTo" le agregamos la regla parm (y definimos ambas variables en el procedimiento):

```
parm( in: &fromName, in: &toName );
```

Y en su Source:

```
print Title
print ColumnTitles
for each Attraction
  where AttractionName>= &fromName
  where AttractionName <= &toName
        print Attractions
endfor
```

Y luego creamos un web panel, en el que definimos dos variables, que pueden llamarse de cualquier manera, por ejemplo A y B, pero que deben tener como tipo de datos uno compatible con el de las variables &fromName y &toName. ¿Por qué? Porque será en estas variables que el usuario ingresará valores que en el evento que programemos serán enviadas por parámetro al procedimiento, así:

AttractionsFromTo(&A, &B)

12. Web panels

La agencia de viajes solicita una página que muestre todos los países y para cada uno, la cantidad de ciudades que ofrece visitar.

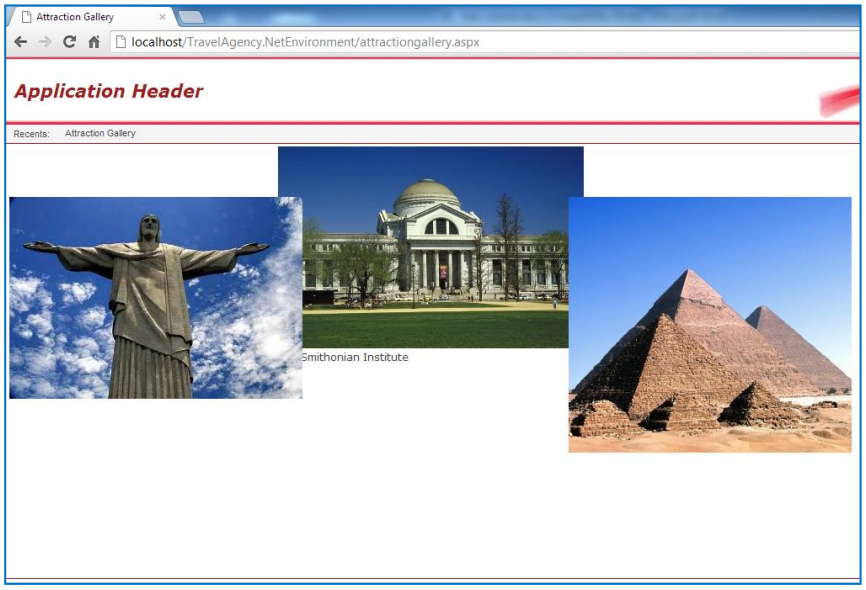
Recuerde que el **evento Load en un web panel con tabla base,** se ejecuta justo antes de cargar cada línea en el grid. Este evento es adecuado para asignar a la variable, el cálculo que devuelve la cuenta de ciudades de cada país que se está navegando y a punto de cargar en una línea del grid.

Agregue ahora al web panel definiDo, dos variables (&CountryNameFrom y &CountryNameTo) y defina las condiciones necesarias para filtrar los países incluidos en dicho rango.

13. Extended controls

Utilizando el extended control ImageGallery, diseñe un web panel que muestre la galería de fotos de todas las atracciones turísticas.

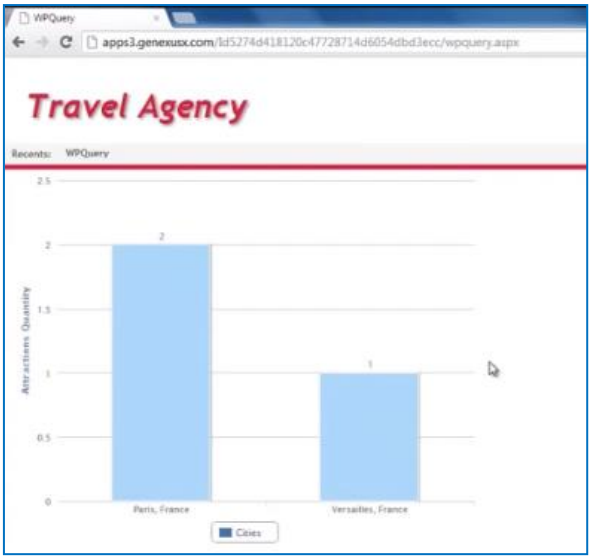
Personalice las propiedades del extended control, estableciendo Width=1000, Height= 500, y Type=Slider:



14. Objeto Query

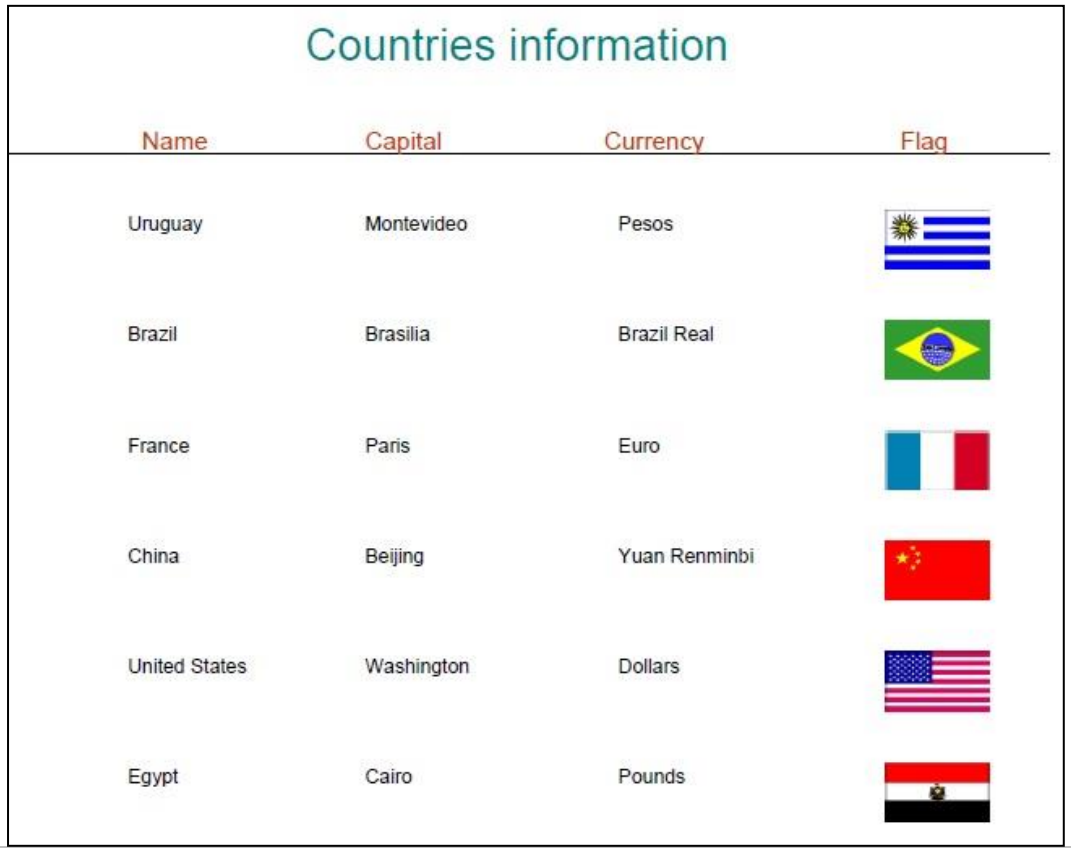
Defina un objeto Query que devuelva solamente las ciudades de Francia, ordenadas por orden alfabético, cada una de ellas con su respectiva cantidad de atracciones turísticas.

Defina un web panel y utilizando el control QueryViewer, visualice la consulta anterior como un gráfico.



15. Web services

La agencia de viajes nos ha solicitado una nueva funcionalidad: desea poder ofrecer a los clientes, un listado de todos los países, mostrando para cada uno su cuidad capital, su moneda de uso y bandera:



Para resolver esto sugerimos importar y hacer uso de un Web Service específico: CountryInfoService, que devuelve dicha información y más.

La ubicación del web service es: http://www.oorsprong.org/websamples.countryinfo/CountryInfoService.wso

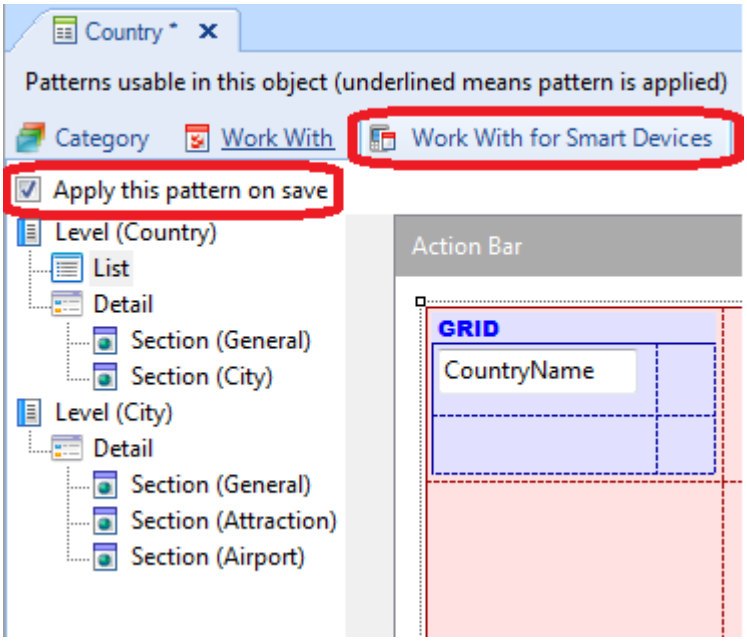
Recuerde que seleccionando en el menú de GeneXus: **Tools / Application Integration / WSDL Import**, se abrirá un wizard, que importará todas las especificaciones del web service, métodos y parámetros en un objeto externo (external object) y se definirán SDTs de ser necesario. Recuerde que definiendo variables del tipo del objeto externo, pordrá ejecutar los métodos provistos por el web service.

16. Se necesita una parte para Smart Devices

La agencia desea ofrecer también una pequeña aplicación para dispositivos inteligentes para ser utilizada por los usuarios finales.

El objetivo es que cualquier persona pueda consultar desde su smart device todos los países que la agencia ofrece visitar, y para cada uno de ellos sus ciudades, y atracciones turísticas.

Para ello, hay que aplicar el patrón "Work With for Smart Devices" a la transacción Country:



Y simplemente grabar, para posteriormente crear un objeto Dashboard y agregarle el objeto de nombre: WorkWithDevicesCountry generado por el patrón.

Recuerde que por el hecho de haber creado dentro de nuestra base de conocimiento, objetos propios para Smart Devices, al presionar F5 automáticamente se ejecutará el emulador para Android, pudiendo también acceder a

nuestra aplicación web desde el Developer menú.

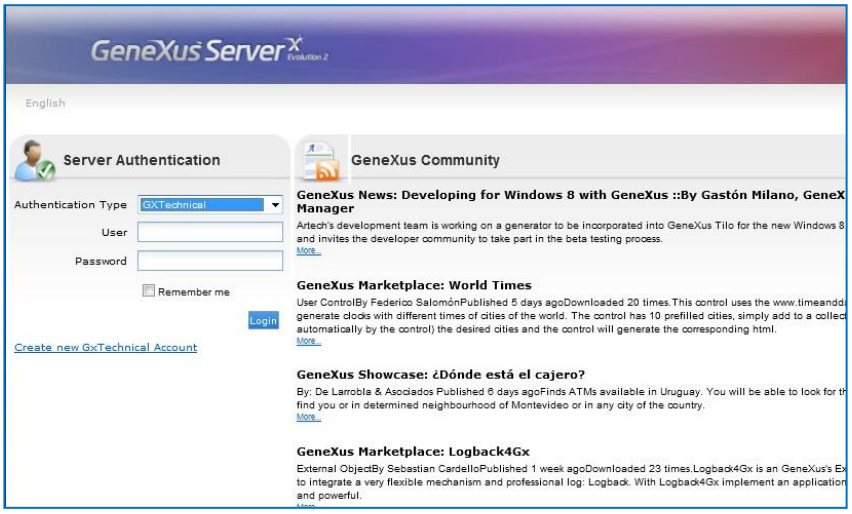
17. GXserver

Publique la base de conocimiento en el servidor http://sandbox.genexusserver.com/xev2/main.aspx

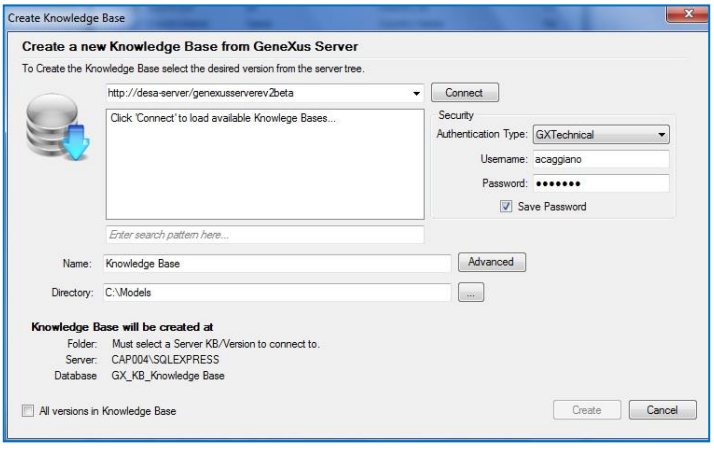
Cree un nuevo web panel (WPCustomers) que muestre la lista de clientes de la Agencia.

Envíe este nuevo objeto al server para que se integre a la base de conocimiento centralizada.

Ingrese a la consola web y verifique el estado final de la KB.



Cierre la base de conocimiento anterior, y cree una nueva sincronizándose con la KB previamente publicada. De esta forma se recibe localmente una copia de la KB administrada por GXserver.



En esta nueva copia local, edite la transacción Country y defina el nuevo atributo *CountryFlagImage*, de tipo Image. Envíe este cambio al servidor.

Cierre esta KB y abra nuevamente la KB inicial. Realice la operación Update para recibir el cambio antes realizado.

---

18. Definición de un modelo de proceso de negocios (BPM) para la reserva de una atracción [opcional]

El proceso de reserva de una atracción turística consiste en una serie de tareas que se puede considerar como un proceso de negocio.

Este proceso comienza cuando una persona va a la agencia de viajes a reservar un paquete turístico para una determinada atracción. Lo primero que debe hacer el empleado de la agencia, es ingresar una reserva para dicho paquete.

Luego de ingresada la reserva, hay que verificar que la persona que solicite el paquete sea un cliente de la agencia y en caso de que no lo sea, se deberá ingresar como cliente y asociar dicho cliente a la reserva.

Una vez completados estos pasos, es necesario verificar que haya paquetes disponibles para la cantidad de personas que deseen realizar el viaje.

Si hay lugares disponibles, la reserva será asignada y finaliza el proceso de negocios. En caso contrario, se le debe ofrecer al pasajero otro paquete.

Para implementar el modelo, deberá crear un objeto Business Process Diagram, pero antes debe crear una transacción Reservation con los siguiente atributos:

- ReservationId - Id
- ReservationDate - Date
- ReservationQty – N(4)
- CustomerId
- CustomerName
- ReservationAvailable – Boolean

El atributo ReservationAvailable se utilizar para marcar si el paquete turístico está disponible o no.

Para el atributo CustomerId, seleccionamos su columna Nullable con el valor Yes, para indicar que al momento de ingresar una reserva, podemos no tener aún al identificador de cliente, de la persona que está adquiriendo la reserva

Defina el modelo de negocios para la reserva de una atracción turística y ejecútelo, probando todos los casos mencionados en la descripción, de forma de recorrer todo el diagrama.

Solución

Cree la transacción Reservation, de acuerdo a lo solicitado.

Cree un objeto Business Process Diagram y nómbrelo AttractionReservation. Desde la toolbar arrastre al diagrama un símbolo de NoneStartEvent.

Ubique a la transacción Reservation en la ventana de Folder View y arrástrela hacia el diagrama. Para conectar el nodo de Start con la transacción, haga clic en la parte inferior del círculo verde y sin soltar, arrastre la conexión hasta que la punta de la flecha toque el borde superior del rectángulo de la transacción.

Para representar la decisión en el diagrama para verificar si la persona es cliente de la empresa, arrastre un nodo Exclusive Gateway sobre el diagrama y únalo desde la transacción Reservation.

Ahora se debe definir la condición del Gateway que hará que el flujo siga el curso normal (hacia abajo) cuando no hay que agregar a la persona como cliente, o el flujo alternativo (hacia la derecha), cuando se debe agregar a la persona como cliente. Primero arrastre la transacción Customer al diagrama y conéctela a la derecha del Gateway.

Luego haga doble clic en la flecha verde que une el Gateway con la transacción Customer y escriba la expresión: Reservation.CustomerId = 0, para indicar que el flujo debe tomar ese camino si cuando se ingresó la reserva, el atributo CustomerId se dejó con valor 0. Observe que debe usar el nombre de la transacción para calificar al atributo CustomerId, para distinguir al atributo que es clave foránea en la transacción Reservation, del atributo CustomerId que es clave primaria de la transacción Customer.

Para asociar el cliente recién creado a la reserva, debe crear un procedimiento llamdo AssignCustomerToReservation y en la sección de reglas escriba una regla parm, con los parámetros &ReservationId y &CustomerId.

En el source del mismo escriba un For each con una cláusula Where filtrando el atributo ReservationId por la variable ReservationId recibida por parámetro. Luego asigne a CustomerId el valor de la variable &CustomerId y cierre el For Each.

De esta forma se le asignara el cliente recién creado, a la reserva ingresada previamente.

Ahora arrastre el procedimiento recién creado hacia el diagrama y conéctelo desde la transacción Customer.

Continuando con el flujo habitual (hacia abajo del Gateway), debe contar con una pantalla donde se pueda confirmar o cancelar la reserva. Para eso, puede utilizar nuevamente la transacción Reservation y se marcará la reserva como disponible o no, mediante el atributo ReservationAvailable.

Arrastre a la transacción Reservation desde la ventana de Folder View hacia el diagrama. Renombre a la tarea como ReservationAvailability y conéctela desde el Gateway.

Para indicar que el flujo hacia abajo es el flujo habitual, seleccione la conexión y en la ventana Propiedades asigne la propiedad ConditionType en el valor Default. Observe en el diagrama que el flujo quedó señalizado con una raya de color verde que cruza al mismo. Aproveche ahora para conectar la tarea AssigntCustomerToReservation, a la tarea ReservationAvailability.

Para evaluar el valor ingresado en el check box de la transacción Reservation, inserte desde la toolbar un Exclusive Gateway y conéctelo desde la tarea ReservationAvailability. Luego conecte el flujo alternativo del mismo (que lo dibujamos hacia la izquierda) a la tarea Reservation, para que se pueda agregar una reserva nueva. Haga doble clic sobre dicho conector y agregue la condición ReservationAvailable=False. Observe que en este caso no es necesario calificar el atributo, porque está únicamente en la transacción Reservation.

Lo que falta hacer ahora es tomar en cuenta el caso de que la reserva se confirme, en cuyo caso ya no habrá más tareas y deberá finalizar el proceso. Para indicar que debe terminar el diagrama, inserte desde la toolbar un símbolo de NoneEndEvent y conéctelo desde el Gateway.

Esta conexión hacia abajo es el flujo normal del Gateway, de modo que cuando la reserva se confirma, finalizará el proceso.

Para indicar esto, seleccione la conexión y en la ventana Propiedades asigne la propiedad ConditionType con el valor Default.

Ahora debe ejecutar el diagrama. Sobre la solapa con el nombre del diagrama, de botón derecho y elija Run.

Se abrirá una pantalla con el Cliente GXflow. Seleccione la tarea Reservation y para ejecutarla presione el botón de Execute, o hagaa doble clic sobre la tarea. Ingrese una reserva, dejando sin ingresar el identificador del cliente. Presione Confirmar y cierre la ventana. Para pasar a la tarea siguiente presione Send.

Como no se ingresó el cliente, la próxima tarea pendiente es la transacción Customer. Ejecute la tarea e ingrese a la persona que solicitó la reserva, como cliente. Cierre la ventana y presione Send.

Observe que la próxima tarea es ReservationAvailability, ya que la tarea AssignCustomerToReservation es no interactiva y por lo tanto se ejecutó en forma batch. Ejecute la tarea, marque el check box y presione Confirmar. Cierre la ventana y presione Send.

Note que ahora la bandeja de entrada se muestra vacía, indicando que no hay más tareas pendientes para ejecutar ya que se ha llegado al fin del proceso.

Ejecute nuevamente el diagrama, pero ahora pruebe los otros caminos del diagrama, por ejemplo ingresando un identificador de cliente en la transacción Reservation, o dejando desmarcado el check box al confirmar la reserva.

Para ver la historia de las tareas ejecutadas, seleccione My Processes en la ventana Navigator y haga doble clic sobre el Proceso AttractionReservation, en la ventana de la derecha. Podrá ver en una ventana, todas las tareas que se fueron ejecutando

Para ver la historia en forma de animación, seleccione More Actions, View Diagram y presionel el botón de Play.

19. Testear la aplicación con GXtest [opcional]

Nota: Descargar aquí GXtest. Y para más información http://abstracta.com.uy/

GXtest nos permite grabar secuencias de operaciones para probar nuestras pantallas y verificar ante nuevos cambios, que el sistema sigue funcionando correctamente.

En este caso, vamos a verificar que el precio de un vuelo se calcule sin errores.

Este precio se representa por el atributo FlightFinalPrice definido con una fórmula global. Esta fórmula calcula el precio final del vuelo, aplicando al precio inicial el descuento del vuelo.

Utilizando GXtest, defina un caso de prueba ingresando un vuelo con los siguientes valores:

Aeropuerto de partida: 2

Aeropuerto de destino: 1

Precio del vuelo: 5000

Descuento del vuelo: 50%

El precio calculado del vuelo será de 2500, ya que como el descuento del vuelo es de 50%, la fórmula aplicará dicho descuento para calcular el valor del atributo FlightFinalPrice.

Continuando con el ejemplo, ingrese los asientos del vuelo y no presione Confirmar.

Defina en GXtest, que el valor correcto para el texto que muestra el valor del precio del vuelo, es de 2500. Presione Confirmar y cierre el navegador.

Verifique en GXtest que el caso de prueba se haya creado correctamente. Verifique que se haya creado el pool de datos con los valores ingresados.

Luego cambie la fórmula del atributo FlightFinalPrice (por ejemplo cambie el 100 del cociente, por el valor 10) y ejecute el caso de prueba ingresado anteriormente.

Verifique que GXtest informe de que encontró un error y vea el valor calculado incorrectamente en la pantalla de la transacción Flight, que se muestra en la ventana de resultados del caso de prueba.

Luego corrija la fórmula (poniendo el valor 100 en lugar del 10) y ejecute nuevamente el caso de prueba. Verifique que en este caso, GXtest no encuentra errores y el resultado de la prueba muestra el valor del precio del vuelo calculado correctamente.

Solución

Abra la aplicación GXtest Designer, desde el acceso directo en el escritorio. Cree un proyecto nuevo y asígnele un nombre.

Ahora presione el signo de más (color verde), elija su KB de la carpeta donde está almacenada, agregue una descripción y presione OK. Cuando finalice la importación de datos de la KB, presione OK.

La URL que nos solicita GXtest es la de la aplicación a probar. Vuelva a GeneXus, ejecute la transacción Flight, copie la URL de la misma y péguela en el campo URL de la ventana de Gxtest. Presione OK.

En la parte derecha de la ventana del Diseñador de GXtest, presione botón derecho sobre Test Cases y elija Record Test Case. Asígnele un nombre al caso de test, una breve descripción y marque la casilla para que se autogenere el conjunto de datos de prueba.

Presione el botón rojo, en la parte superior derecha de la ventana. Notará que se abrirá el navegador y se ejecutará la transacción Flight. Ingrese un vuelo nuevo con los datos especificados en la letra del ejercicio, incluyendo los asientos, pero no presione Confirmar para terminar el ingreso del vuelo.

Antes, seleccione con el mouse el precio del vuelo (que muestra el valor 2500) y presione el símbolo de check, en la barra de herramientas de GXtest del navegador. Elija VerifyControlText y agregue una descripción. Presione Aceptar y nuevamente Aceptar.

GXtest avisará que agregó correctamente la validación solicitada. Presione OK.

Vuelva a la pantalla de la transacción Flight y presione Confirmar. Cierre el navegador y vuelva al diseñador de GXtest.

Vea que se creó un caso de prueba. En el diagrama seleccione el componente llamado Flight y vea que en la ventana de comandos se detallan todos los pasos realizados al ingresar el vuelo en la transacción Flight.

En el panel de Project, a la derecha de la pantalla y bajo DataPools haga clic en el elemento del conjunto de datos. Verifique en dicha ventana, los datos que usó para ingresar el vuelo.

Vuelva a GeneXus, edite la fórmula del atributo FlightFinalPrice y en la división escriba un 10 en lugar del 100. Presione F5 para actualizar la aplicación. Para ejecutar el caso de test definido, vuelva a GXtest, seleccione la solapa del caso de prueba y presione el botón de Play, que se encuentra arriba a la izquierda de la pantalla.

Verá que se abre la ventana del navegador, se muestra la pantalla de la transacción Flight y empieza la ejecución automática del caso de prueba, ingresándose automáticamente los mismos datos que Ud. había ingresado antes, como si fuera una persona que lo va haciendo.

Preste atención a los valores y vea que el precio del vuelo se calculó incorrectamente. Ahora muestra el valor -20000. Cuando finalice el caso de prueba, notará que se activa nuevamente la ventana del Diseñador de GXtest. Abra el diseñador.

Notará que se agregó una solapa con los resultados del test. En la columna Result encontrará el símbolo de un insecto, representando un "bug" encontrado. En el panel de la izquierda, si hace clic en el símbolo de + al lado del insecto, se abrirán varias opciones y a la derecha, los test realizados.

Si hace doble clic en Flight, se abrirá una pantalla donde puede observar la pantalla de la transacción Flight, los pasos que integran el test realizado, el tiempo que insumió cada paso y su resultado

Bajando la barra, verá que aparece el error de validación del precio del vuelo, indicando cuál fue el valor esperado y cuál fue el valor real obtenido. Si hace clic sobre el renglón del error, abajo se desplegará la pantalla del navegador, con el valor calculado incorrectamente.

Vuelva a la transacción Flight, deje la fórmula como estaba antes (vuelva a poner 100 en lugar de 10) y vuelva a GXtest para ejecutar nuevamente el caso de prueba.

Verifique que ahora no aparece ningún símbolo de error y que el resultado del cálculo del vuelo sea el correcto, mostrando nuevamente el valor 2500. Verifique el valor haciendo click sobre el renglón que muestra la validación de dicho precio y verifique en la pantalla del navegador que se mostrará debajo, que el valor mostrado sea también 2500.