

Ejercicios prácticos

Julio 2022

Copyright ©GeneXus S.A. 1988-2019.

All rights reserved. This document may not be reproduced by any means without the express permission of GeneXus S.A. The information contained herein is intended for personal use only.

Registered Trademarks:

GeneXus is trademark or registered trademark of GeneXus S.A. All other trademarks mentioned herein are the property of their respective owners.

CONTENIDO

CONTENIDO	2
EL PROBLEMA	4
NUEVO PROYECTO, NUEVA BASE DE CONOCIMIENTO	4
PRIMERAS TRANSACCIONES	5
Transacción "Employee"	5
Transacciones "AmusementPARK" y "country", relacionadas	8
Datos relacionados: ¿cómo se mantiene la integridad?	10
Transacción 'Show'	11
Transacción 'Game'	11
Transacción 'Category'	11
Transacciones "Employee" y "AmusementPark", relacionadas.....	12
Agreguemos las ciudades a la transacción 'Country'	13
Transacción "AmusementPark": agreguemos la ciudad.	14
AGREGUEMOS COMPORTAMIENTO A LAS TRANSACCIONES (RULES)	15
PATTERNS: MEJORANDO LA INTERFAZ PARA TRABAJAR CON LA INFORMACIÓN.....	16
TRANSACCIONES "REPAIR" Y "TECHNICIAN" Y NECESIDAD DE DEFINIR SUBTIPOS	18
FÓRMULAS	19
CREACIÓN DE SEGUNDO NIVEL	20

EL PROBLEMA

Una multinacional encargada de gestionar parques de diversiones 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:

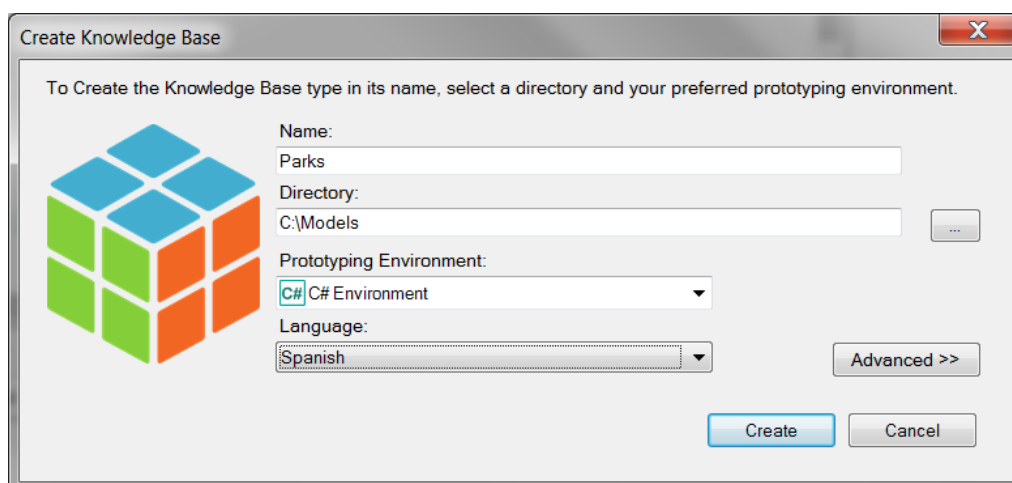
- **Backend:** parte de la aplicación que deberá correr en un servidor web, de manera tal que los empleados de la empresa puedan manipular la información desde cualquier lugar con conexión a internet.
- **Aplicación sencilla para dispositivos móviles:** parte de la aplicación que será destinada para ser descargada por los clientes de la empresa, la cual les permitirá consultar los países disponibles, así como los principales parques de diversiones que ofrece cada ciudad y sus juegos.

NUEVO PROYECTO, NUEVA BASE DE CONOCIMIENTO

Entrar a GeneXus y crear una base de conocimiento de nombre *Parks* 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 con C# y SQL Server 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 contenido de la ventana **KBExplorer** (Knowledge Base Explorer). 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 **'Preferences'** donde se configura el **'Environment'**.

PRIMERAS TRANSACCIONES

En las reuniones con la empresa, le transmiten lo siguiente:

“Nosotros registramos los datos de parques de diversiones, para realizar la gestión tanto de sus empleados como de sus juegos y actividades que ofrece a los visitantes”.

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 “EMPLOYEE”

Preguntamos: ¿qué datos registran de los empleados de la empresa? La respuesta es la siguiente:

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

Con estos datos ya puede crear la transacción *Employee*.

Recordar que:

- Para crear objetos existen varias alternativas:
 - Hacerlo desde el menú: File/ New Object
 - Ctrl+N
 - Ícono de la barra de herramientas
- Digitando punto (".") cuando va a ingresar un nuevo atributo, éste se inicializa con el nombre de la transacción.
- Necesitará un atributo que identifique a cada empleado (EmployeeId).

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

Name	Type
Employee	Employee
EmployeeId	Numeric(4,0)
EmployeeName	Character(20)
EmployeeLastName	Character(20)
EmployeeAddress	Address, GeneXus
EmployeePhone	Phone, GeneXus
EmployeeEmail	Email, GeneXus

Recordar que:

- **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.
- Cuando esté definiendo el tipo de datos del atributo identificador, en vez de utilizar directamente Numeric(4.0), defina el **dominio Id** de forma inline: **Id=Numeric(4.0)**. 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.

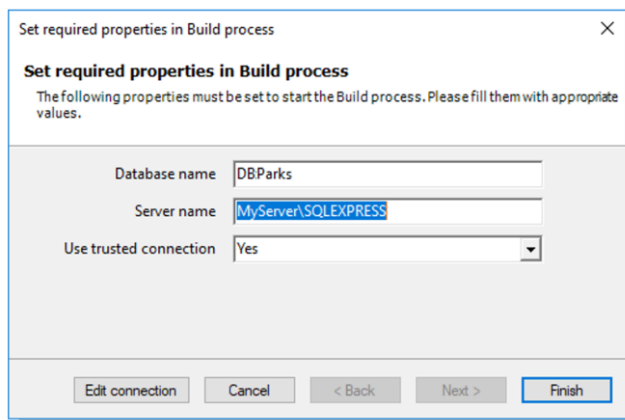
El próximo paso es probar la aplicación en ejecución. Asegúrese de tener la ventana *Output* de GeneXus habilitada y a la vista. (*View/Other Tool Windows/Output*).

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

¿Qué sucederá?

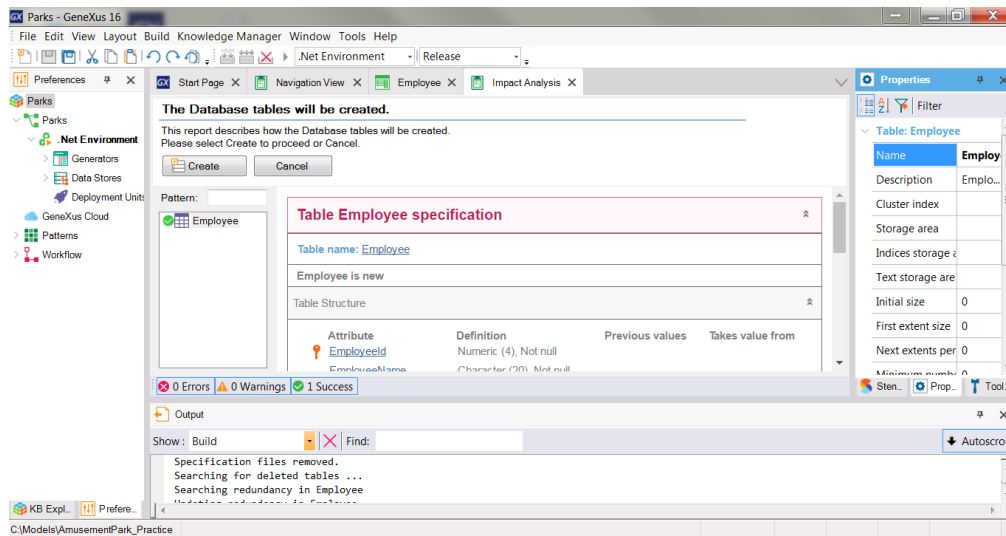
Recuerde:

Si decide crear la base de datos y programas **localmente**, 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 de 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 *EMPLOYEE* dentro de la misma:



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

Ingrese algunos empleados al sistema. Luego, **modifique** algún dato de alguno de los empleados previamente ingresados y **elimine** algún empleado.

También pruebe usar las flechitas ofrecidas para pasar de registro en registro de empleados y la opción **SELECT**, que ofrece una “lista de selección” para ver la lista de empleados registrados y seleccionar uno.

Ahora pasemos a identificar y crear la siguiente transacción. Recordemos lo que nos habían enunciado, a lo cual se le hicieron algunos agregados:

“Nosotros registramos los datos de **parques de diversiones** de diferentes **ciudades** en diferentes **países**, para realizar la gestión tanto de sus **empleados** como de sus **juegos** y **actividades** que ofrece a los visitantes”.

Antes de continuar con el desarrollo, publiquemos la KB en GeneXus Server (<http://sandbox.genexusserver.com/v16>)

Recuerde:

- Debe autenticarse con su cuenta GeneXus.
- El nombre con el cual se publique la KB debe ser único. Esto significa que no pueden existir en el Server dos KBs con el mismo nombre.

TRANSACCIONES “AMUSEMENTPARK” Y “COUNTRY”, RELACIONADAS

Le preguntamos a los empleados de la empresa: ¿qué datos registran de los parques de diversiones con los que trabajan? La respuesta es la siguiente:

El **nombre** (que no supera los 40 caracteres), **sitio web** (que no supera los 60 caracteres), **dirección** y **foto representativa**.

Con estos datos ya puede crear la transacción *AmusementPark*.

Vamos a crear una transacción para registrar los países a los que los parques de diversiones pertenecen.

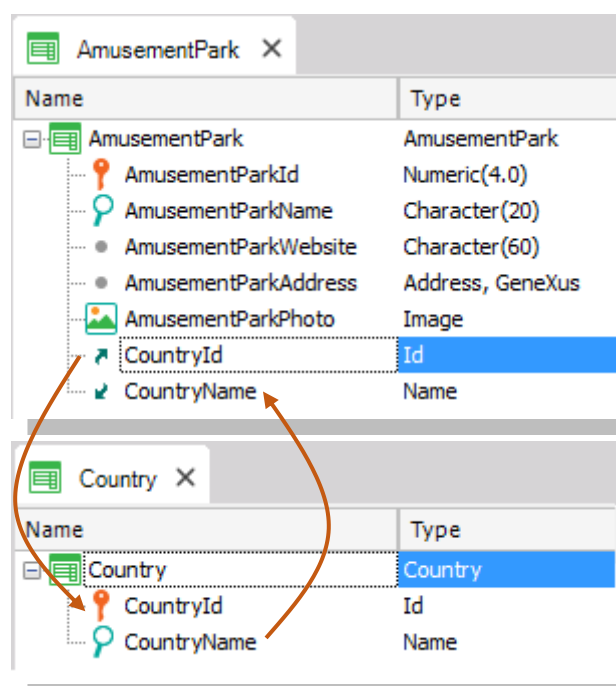
Recuerde que:

- 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.

Defina el atributo *CountryName*, creando y utilizando un nuevo dominio: *Name=Character(50)*.

Ahora, volveremos a la transacción *AmusementPark*, para agregarle los atributos *CountryId* y *CountryName*.

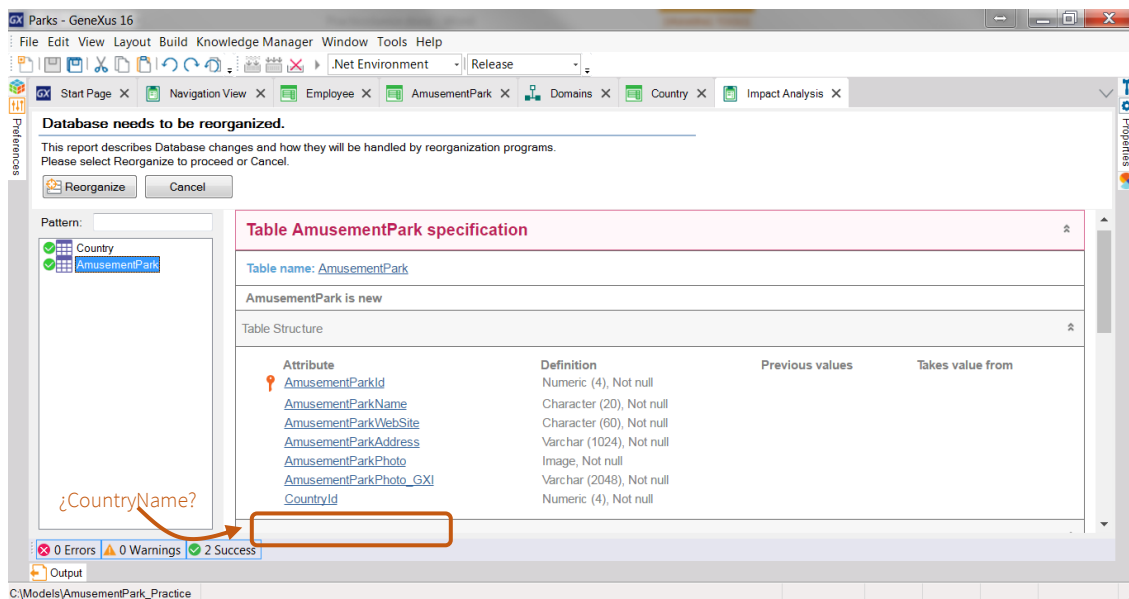
Aprovechemos para cambiar el tipo de datos de *AmusementId*, asignándole el dominio Id previamente creado si es que aún no lo ha configurado.



¿Por qué colocó además de *CountryId*, el atributo *CountryName* en *AmusementPark*?

Respuesta: El atributo *CountryName* es importante que aparezca en la pantalla de la transacción, para mostrarnos el nombre del país, que es el dato que recordamos mejor del país, en lugar de ver solamente su identificador. También es necesario agregar al atributo en la estructura de la transacción si luego lo queremos utilizar, por ejemplo, dentro de una regla.

Ejecute para probar (F5) y le aparecerá el siguiente reporte:



¿Por qué en la tabla *AmusementPark*, que GeneXus informa que se debe alterar 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 (*AmusementPark*, *Country* y *Employee*).

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

Country

Selection List Country

COUNTRY ID	Id	Name
COUNTRY NAME	✓ 1	Brazil
	✓ 2	France
	✓ 3	China

CANCEL

Ingresa como parque de diversiones: “Beto Carrero World”, que está en Brasil. Si no recuerda el identificador de Brasil en el sistema, ¿cómo ingresa el país? Se le ofrece un ícono con una flecha al lado de *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?

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

- Intente ingresar un parque de diversiones con un Id de país que no exista. ¿Le permite grabar ese parque?
- Elija un parque previamente ingresado (por ejemplo, ‘Beto Carrero World’) y cambie el país, por uno que no exista. ¿Pudo grabar la modificación?
- Intente eliminar un país (usando la transacción *Country*) que tenga algún parque asociado (por ejemplo, Brasil). ¿Se lo permite?

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

TRANSACCIÓN 'SHOW'

Nos informan que los parques de diversiones pueden brindar diferentes shows (musicales, de magia, didácticos, etc) a sus asistentes, en determinada fecha y horario. De los shows interesa conocer su nombre y una imagen representativa. Un mismo show puede ser contratado por varios parques de diversiones.

TRANSACCIÓN 'GAME'

Como nos fue solicitado inicialmente, el sistema debe brindar la posibilidad de ingresar los juegos disponibles en cada parque, por lo que debemos crear una transacción que contenga su nombre y el parque de diversiones al que pertenece.

TRANSACCIÓN 'CATEGORY'

Nos falta completar la información de la transacción *Game*. Los empleados describieron que también registran de cada juego la **categoría** (infantil, radical, recreativo, etc.) a la que pertenece. Así que necesitaremos crear una transacción para registrar esta información, y agregar la categoría a la transacción *Game*.

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

Sugerencia: Observe la estructura de la transacción *Game*. La propiedad Nullable permite indicar si el valor de un atributo puede quedar sin especificar.

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

Para ello, cree el atributo *GamePhoto* 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. Deberán crearse las tablas *Show*, *Category* y *Game* (no se preocupe en entender por qué requiere almacenar dos valores por imagen).

Reorganice y ejecute.

Ingrese categorías (como infantil y radical) y juegos (como montañas rusas y carruseles).

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 el juego un valor inexistente, no le dejará grabar.

Recuerde actualizar los cambios en GeneXus Server.

Application Name

by GeneXus

Recents Category — Game

Game	
	« < > » SELECT
Id	<input type="text" value="0"/>
Name	<input type="text" value="Big Tower"/>
Amusement Park Id	<input type="text" value="1"/> 
Amusement Park Name	Beto Carrero World
Category Id	<input type="text" value="2"/> 
Category Name	Radical
Photo	

TRANSACCIONES “EMPLOYEE” Y “AMUSEMENTPARK”, RELACIONADAS

Como nos dijeron al comienzo del desarrollo de la aplicación, el sistema gestionará los parques de diversiones y sus empleados. Por lo tanto, necesitamos vincular a los empleados que son registrados con el parque en el cual trabajan.

Para eso, vayamos a la transacción *Employee* y agreguemos los atributos *AmusementParkId* y *AmusementParkName*.

Nos han informado que no es obligatorio registrar en el momento el parque en donde trabaja el empleado, es decir, se puede dejar vacío. ¿Qué debemos hacer?

Ejecute para probar (F5) y le aparecerá el siguiente reporte indicándonos que el atributo AmusementParkId ahora permite dejar un valor no especificado:

Database needs to be reorganized.

This report describes Database changes and how they will be handled by reorganization programs.
Please select Reorganize to proceed or Cancel.

Reorganize Cancel

Pattern: Employee

Table Employee specification

Table name: Employee

Employee needs conversion

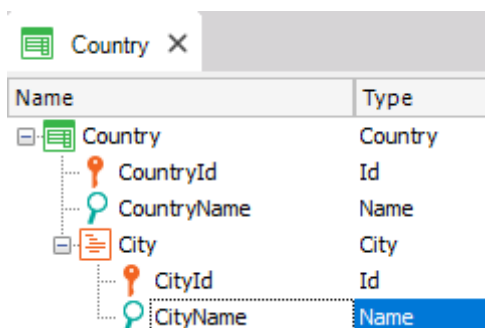
Table Structure

Attribute	Definition	Previous values	Takes value from
EmployeeId	Numeric (4), Not null, Autonumber		Employee EmployeeId
EmployeeName	Character (20), Not null		Employee EmployeeName
EmployeeLastName	Character (20), Not null		Employee EmployeeLastName
EmployeeAddress	Varchar (1024), Not null		Employee EmployeeAddress
EmployeePhone	Character (20), Not null		Employee EmployeePhone
EmployeeEmail	Varchar (100), Not null		Employee EmployeeEmail
New AmusementParkId	Numeric (4)		Null

0 Errors 0 Warnings 1 Success

AGREGUEMOS LAS CIUDADES A LA TRANSACCIÓN 'COUNTRY'

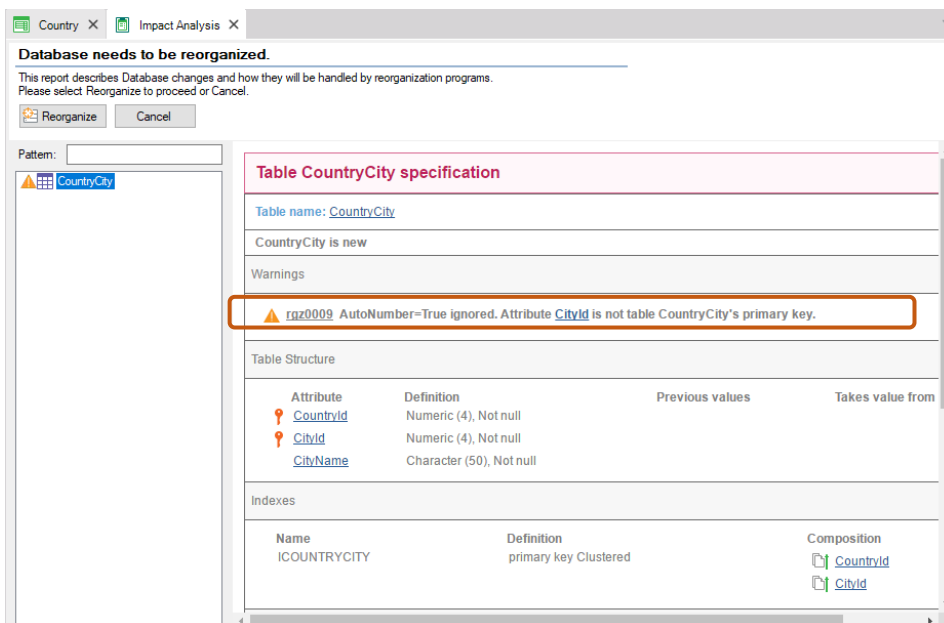
Además de los países, necesitamos registrar la información de sus ciudades. Por lo tanto, debemos agregar un segundo nivel a la transacción *Country*, con el identificador y el nombre de ciudad.



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 "AMUSEMENTPARK": AGREGUEMOS LA CIUDAD.

En la transacción *AmusementPark* agreguemos la ciudad del país a la que el parque pertenece. ¿Qué debe hacer si la empresa nos informa que ese valor puede no ser conocido o relevante para un parque dado en un momento dado?

Construya la aplicación y pruébela (F5 y Reorganize).

Actualice los cambios en GeneXus Server.

AGREGUEMOS COMPORTAMIENTO A LAS TRANSACCIONES (RULES)

Después de probar con nosotros la aplicación que venimos desarrollando, en la empresa nos cuentan que para los empleados hay algún comportamiento específico que debemos hacer cumplir a la hora de manipular la información a través del programa (transacción *Employee*).

¿Cuál es este comportamiento?

Nos dicen:

- “El sistema no debe permitir ingresar empleados 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 empleado al sistema (**EmployeeAddedDate**) 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 empleado dejando vacío el nombre. ¿Le permite grabar o pasar al siguiente campo?

Ídem con el apellido. ¿Sucedre 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?

Por último, nos solicitan que cuando se registre un nuevo país (transacción Country) al ingresar sus ciudades, se autonumeren y el atributo CityId quede inhabilitado.

Especifíquelo y pruébelo en ejecución.

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, parques de diversiones, empleados, shows, categorías y juegos 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 el patrón *Work With for Web* a las 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 construyendo.
- GeneXus creará automáticamente varios objetos por transacción, para implementar el "Trabajar con" esa entidad.

¿Por qué no aparecen más en el *Developer Menu* las transacciones *Country*, *AmusementPark*, *Employee*, *Show*, *Category* y *Game*?

Pruebe hacer lo siguiente:

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.

Application Name
by GeneXus

Recents
Countries

Countries
 Name
+ INSERT

Name		
Brazil	UPDATE	DELETE
China	UPDATE	DELETE
France	UPDATE	DELETE

- Ingrese un par de parques de diversiones (Ej: Shangai Disney Resort, de China/Shangai, Parc Du Bocasse de Francia/Normandía, Happy Valley, de China/Beijing).
- Filtre los parques de diversiones cuyo nombre empiece con P. ¿Y si ahora quiere poder visualizar todos los parques de China? No está incluida esta posibilidad, por lo que deberemos personalizar el pattern *Work With* de esta transacción, para agregárselo. Hágalo en GeneXus y pruebe en ejecución.





Sugerencia: Observe cómo está especificado el filtro que sí existe, por nombre del parque de diversiones. Intente definir el filtro solicitado de igual forma.

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

Application Name
by GeneXus

Recents
Amusement Parks

Amusement Parks
 Park Name
+ INSERT

Park Id	Park Name	Park Website	Park Address	Park Photo	Country Id	Country Name	City Id	City Name		
1	Beto Carrero World	www.betocarrero.com.br	Rua Inácio Francisco de Souza, 1597		1	Brazil	2	São Paulo	UPDATE	DELETE
4	Happy Valley	bj.happyvalley.cn	Beijing Olympic Tower		3	China	1	Beijing	UPDATE	DELETE
3	Parc Du Bocasse	www.parcduboasse.fr	226 Route de Clères, 76690		2	France	2	Normandia	UPDATE	DELETE
2	Shangai Disney Resor	www.shangaidisneyresort.com	Shanghai Disney Resort, Pudong, Xangai		3	China	2	Shangai	UPDATE	DELETE

9. Si ahora quiere brindar la posibilidad de que el usuario elija si quiere ver los parques de diversiones ordenado por nombre del parque o por nombre de país, **implementelo y pruebe**.





Application Name
by GeneXus

Recents Amusement Parks
Amusement Parks
Q Park Name
+ INSERT

Ordered By : Park Name

Park Name

Country Name

Park Id	Park Name	Park Website	Park Address	Park Photo	Country Name	City Name		
1	Beto Carrero World	www.betocarrero.com.br	Rua Inácio Francisco de Souza, 1597		Brazil	São Paulo	UPDATE	DELETE
4	Happy Valley	bj.happyvalley.cn	Beijing Olympic Tower		China	Beijing	UPDATE	DELETE
3	Parc Du Bocasse	www.parcdubocasse.fr	226 Route de Clères, 76690		France	Normandia	UPDATE	DELETE
2	Shanghai Disney Resor	www.shangaidisneyresort.com	Shanghai Disney Resort, Pudong, Xangai		China	Shangai	UPDATE	DELETE



TRANSACCIONES “REPAIR” Y “TECHNICIAN” Y NECESIDAD DE DEFINIR SUBTIPOS

Se necesita ahora registrar los juegos que entran en estado de reparación. Cada reparación tiene un identificador, una fecha desde la cual el juego deja de estar disponible para su uso, la cantidad de días estimada para su reparación, el identificador del juego, su nombre, el técnico titular y el técnico suplente. También cada reparación tiene un costo. Para el costo cree un dominio llamado *Cost*, del tipo Numeric(8.2).

Cree una transacción para registrar a los técnicos que trabajarán en las reparaciones. Cada técnico tiene un identificador, un nombre y apellido, un teléfono, un país y una ciudad en la cual se encuentra.

¿Cómo se define que cada reparación tiene un técnico titular y otro suplente?

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 subtipo.
- 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 una reparación, se dispare un error si el técnico titular que está queriendo asignarle a la reparación no existe. Ídem para el técnico suplente.

No debe permitirse ingresar una reparación cuyo técnico titular coincida con el técnico suplente. Implemente ese comportamiento y pruébelo en ejecución.

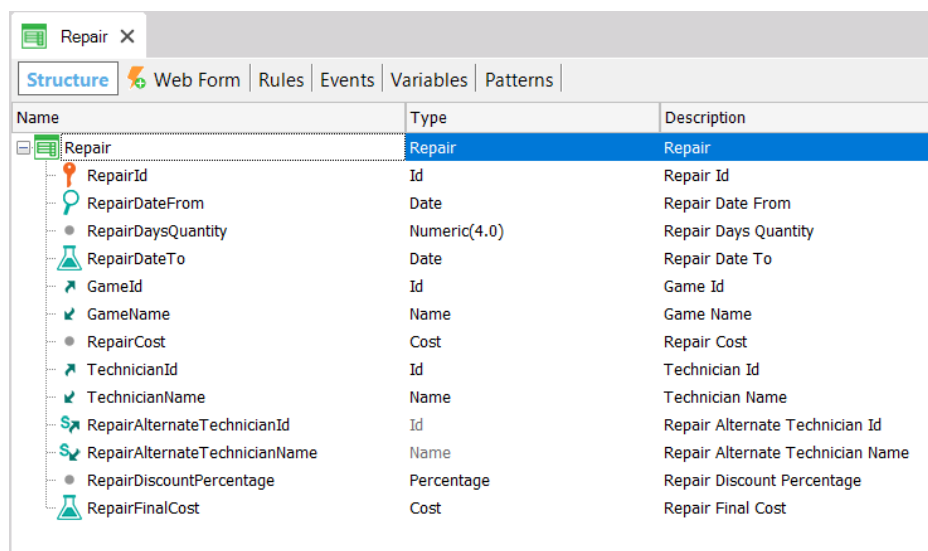
Actualice los cambios en GeneXus Server.

FÓRMULAS

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

Se desea visualizar el precio final de la reparación con el descuento aplicado. Para resolver esto, defina otro atributo más, de nombre *RepairFinalCost*, que sea **fórmula global** y que calcule automáticamente el precio final de la reparación.

Agregue un nuevo campo llamado *RepairDateTo*, el cual será una suma entre la fecha de inicio de reparación y la cantidad de días que la misma llevará.



Name	Type	Description
Repair	Repair	Repair
RepairId	Id	Repair Id
RepairDateFrom	Date	Repair Date From
RepairDaysQuantity	Numeric(4.0)	Repair Days Quantity
RepairDateTo	Date	Repair Date To
GameId	Id	Game Id
GameName	Name	Game Name
RepairCost	Cost	Repair Cost
TechnicianId	Id	Technician Id
TechnicianName	Name	Technician Name
RepairAlternateTechnicianId	Id	Repair Alternate Technician Id
RepairAlternateTechnicianName	Name	Repair Alternate Technician Name
RepairDiscountPercentage	Percentage	Repair Discount Percentage
RepairFinalCost	Cost	Repair Final Cost

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

CREACIÓN DE SEGUNDO NIVEL

Se desea crear un segundo nivel en la transacción *Repair* para allí guardar un detalle del tipo de problema encontrado para reparar.

Para ello, antes que nada se deberá crear un dominio llamado *KindName*, Character(1). Restrinja los valores posibles para el dominio: que sean válidos los valores “E”, “M” y “R” (editando la propiedad **Enum Values** del mismo como se ve a continuación).

Name	Description	Value
Electricity	Electricity	E
Mechanical	Mechanical	M
Replacement	Replacement	R

Cree un segundo nivel en la transacción *Repair* de nombre *Kind* para registrar el tipo de reparación. Este nivel tendrá estos tres atributos:

- *RepairKindId* – Numeric(4) (será clave en este segundo nivel)
- *RepairKindName* – Basado en dominio *KindName* (Genexus lo sugerirá automáticamente).
- *RepairKindRemarks* – Character(120), contendrá observaciones sobre el problema, un pequeño detalle del problema encontrado o la pieza a remplazar por ejemplo.



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**. En este caso, no será necesario ya que sólo el primer atributo forma parte de la clave primaria y éste ya aparece con el ícono de llave.

A efectos de conocer la cantidad de tipos de problema que implica una reparación, cree un nuevo atributo en el primer nivel de la transacción *Repair*, de nombre *RepairProblems*, Numeric(1) y defínalo como fórmula global, deberá contar los tipos de problemas encontrados.

Una reparación podría implicar problemas de electricidad, de mecánica o que sea necesario el remplazo de alguna pieza. Podría incluso tener más de un problema del mismo tipo, dos registros de electricidad, por ejemplo. Si hubiera muchos, se puede detallar algo más con el uso del atributo de observaciones, el cual permite escribir un pequeño texto.

Se desea ver la cantidad de problemas en el formulario web y se debe controlar que se ingresen entre 1 y 3 líneas de tipos de problema.

Este control deberá realizarse cuando se termine de ingresar datos en el segundo nivel y luego de haber presionado el botón *Confirm*.

Name	Type	Description
 Repair	Repair	Repair
RepairId	Id	Repair Id
RepairDateFrom	Date	Repair Date From
RepairDaysQuantity	Numeric(4.0)	Repair Days Quantity
RepairDateTo	Date	Repair Date To
GameId	Id	Game Id
GameName	Name	Game Name
RepairCost	Cost	Repair Cost
TechnicianId	Id	Technician Id
TechnicianName	Name	Technician Name
RepairAlternateTechnicianId	Id	Repair Alternate Technician Id
RepairAlternateTechnicianName	Name	Repair Alternate Technician Name
RepairDiscountPercentage	Percentage	Repair Discount Percentage
RepairFinalCost	Cost	Repair Final Cost
RepairProblems	Numeric(1.0)	Repair Problems
 Kind	Kind	Kind
RepairKindId	Numeric(1.0)	Repair Kind Id
RepairKindName	KindName	Repair Kind Name
RepairKindRemarks	Character(120)	Repair Kind Remarks

El valor de *RepairKindId* se va ingresando manualmente. Lo normal sería ir dándole valores a partir de 1. Recordemos que **no es posible autonumerar** este atributo utilizando la propiedad *Autonumber*.

Actualice los cambios en GeneXus Server.