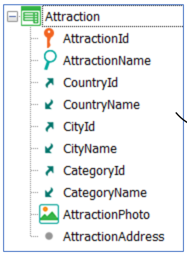
Actualización de la base de datos

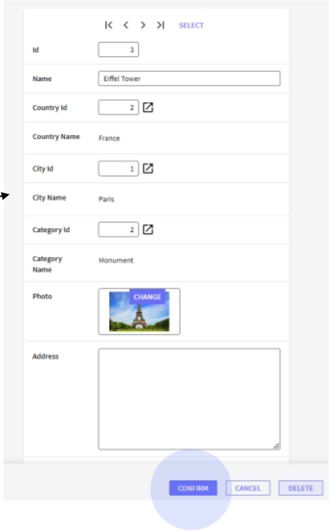
Usando Business Components. Justificación

GeneXus™

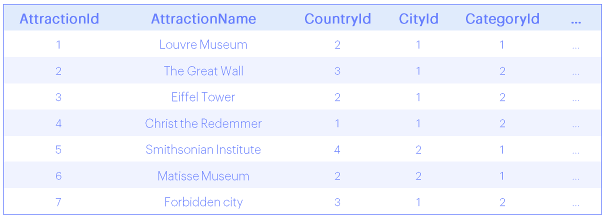
Transaction: Insert, Update, Delete



Attraction
- AttractionId
- AttractionName
- CountryId
- CountryName
- CityId
- CityName
- CategoryId
- CategoryName
- AttractionPhoto
- AttractionAddress



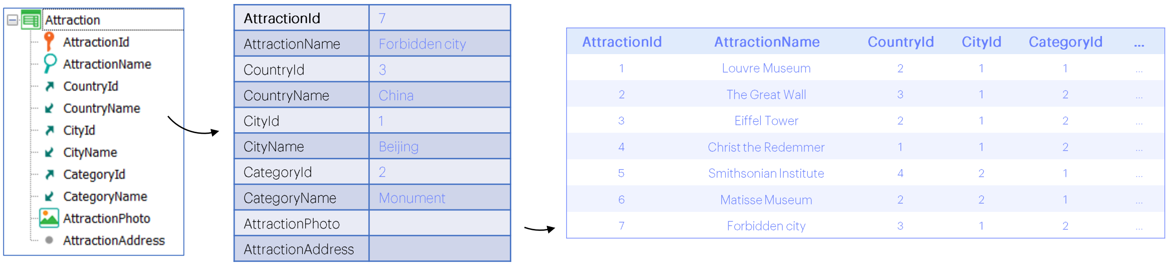
Id	3
Name	Eiffel Tower
Country Id	2
Country Name	France
City Id	1
City Name	Paris
Category Id	2
Category Name	Monument
Photo	
Address	



AttractionId	AttractionName	CountryId	CityId	CategoryId	...
1	Louvre Museum	2	1	1	...
2	The Great Wall	3	1	2	...
3	Eiffel Tower	2	1	2	...
4	Christ the Redemmer	1	1	2	...
5	Smithsonian Institute	4	2	1	...
6	Matisse Museum	2	2	1	...
7	Forbidden city	3	1	2	...

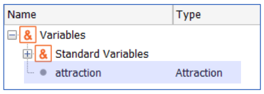
Hasta ahora únicamente actualizamos información de la base de datos a través de las transacciones, esto es, de forma interactiva, a través de una interfaz gráfica.

Business Component: Insert(), Update(), Delete()



AttractionId	7
AttractionName	Forbidden city
CountryId	3
CountryName	China
CityId	1
CityName	Beijing
CategoryId	2
CategoryName	Monument
AttractionPhoto	
AttractionAddress	

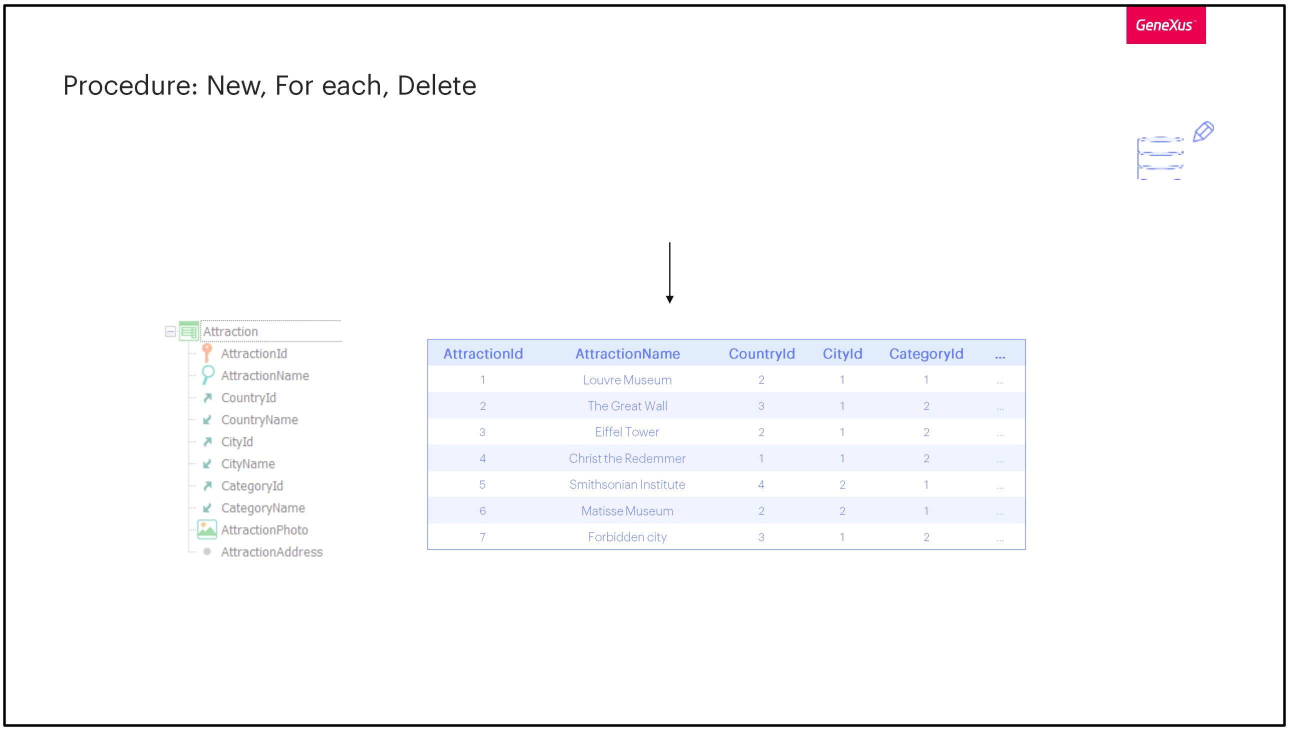
AttractionId	AttractionName	CountryId	CityId	CategoryId	...
1	Louvre Museum	2	1	1	...
2	The Great Wall	3	1	2	...
3	Eiffel Tower	2	1	2	...
4	Christ the Redemmer	1	1	2	...
5	Smithsonian Institute	4	2	1	...
6	Matisse Museum	2	2	1	...
7	Forbidden city	3	1	2	...



Name	Type
Variables	
Standard Variables	
attraction	Attraction

En lo que sigue vamos a estudiar cómo actualizar la información de la base de datos por código.

Vamos a privilegiar una de las dos maneras: la actualización a través de business components, ya entenderemos por qué.
Trabajaremos con la estructura de la transacción como si fuera una variable SDT, respetando también las reglas de la transacción. A través de esa variable es que insertaremos, modificaremos o eliminaremos de la base de datos. Será como utilizar la transacción, pero sin su pantalla.



Procedure: New, For each, Delete

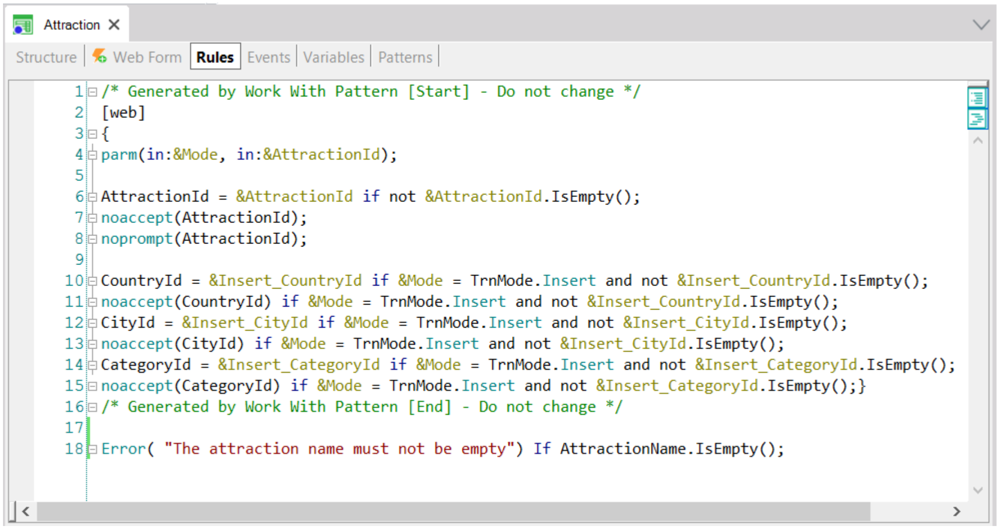
- Attraction
  - AttractionId
  - AttractionName
  - CountryId
  - CountryName
  - CityId
  - CityName
  - CategoryId
  - CategoryName
  - AttractionPhoto
  - AttractionAddress

AttractionId	AttractionName	CountryId	CityId	CategoryId	...
1	Louvre Museum	2	1	1	...
2	The Great Wall	3	1	2	...
3	Eiffel Tower	2	1	2	...
4	Christ the Redemmer	1	1	2	...
5	Smithsonian Institute	4	2	1	...
6	Matisse Museum	2	2	1	...
7	Forbidden city	3	1	2	...

La otra manera es hacerlo a través de comandos especiales que solo pueden utilizarse dentro de objetos de tipo procedimiento. Los veremos más adelante, en otro video. Pero en este caso nos independizamos de la transacción. Trabajamos directamente sobre las tablas, lo que tiene sus desventajas.

Entonces estudiemos esta primera opción, la de más alto nivel y por tanto la que en principio nos resultará más relevante.

Insert through the transaction



```
/* Generated by Work With Pattern [Start] - Do not change */
[web]
{
parm(in:&Mode, in:&AttractionId);

AttractionId = &AttractionId if not &AttractionId.IsEmpty();
noaccept(AttractionId);
noprompt(AttractionId);

CountryId = &Insert_CountryId if &Mode = TrnMode.Insert and not &Insert_CountryId.IsEmpty();
noaccept(CountryId) if &Mode = TrnMode.Insert and not &Insert_CountryId.IsEmpty();
CityId = &Insert_CityId if &Mode = TrnMode.Insert and not &Insert_CityId.IsEmpty();
noaccept(CityId) if &Mode = TrnMode.Insert and not &Insert_CityId.IsEmpty();
CategoryId = &Insert_CategoryId if &Mode = TrnMode.Insert and not &Insert_CategoryId.IsEmpty();
noaccept(CategoryId) if &Mode = TrnMode.Insert and not &Insert_CategoryId.IsEmpty();}
/* Generated by Work With Pattern [End] - Do not change */

Error( "The attraction name must not be empty") If AttractionName.IsEmpty();
```

Observaremos primero con cierta atención lo que sucede cuando queremos insertar una atracción turística nueva utilizando la transacción.

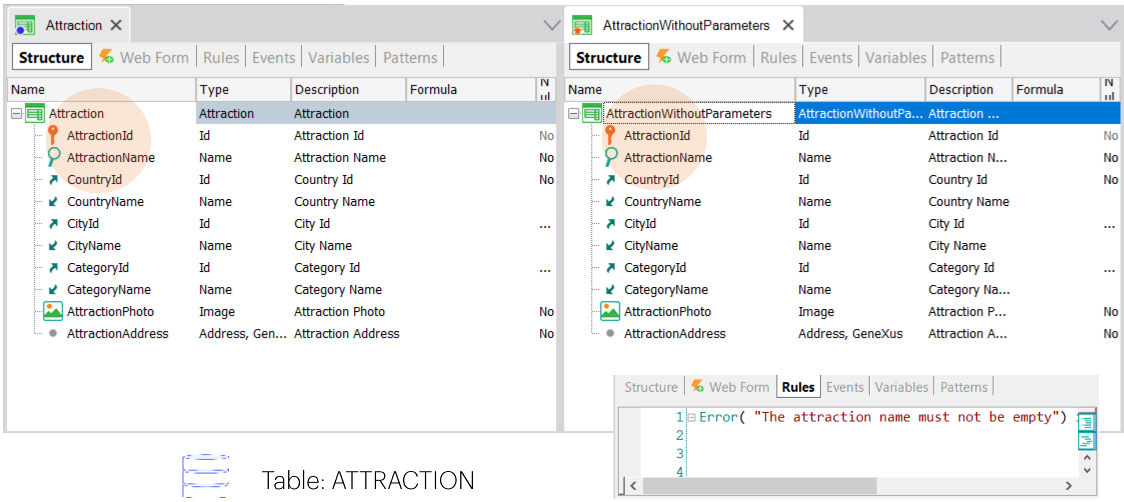
Hemos modificado el orden de los atributos, para que la ciudad nos quede enseguida del país, lo que repercute en el form.

Vamos a agregar una regla de error para evitar que se deje el nombre de la transacción vacío.
Si ahora presionamos la tecla de Control más la barra espaciadora, se nos abre esta ventanita que nos ofrece todas las cosas que podríamos colocar en esta parte del código. Si digitamos la ene, ya encontramos el atributo que buscábamos. Esta es una forma de no cometer errores de tipeo. La otra es con Insert > Attribute, cuyo shortcut es Control+Shift+A, y allí elegimos Attraction Name.

A esta transacción le habíamos aplicado el pattern Work With, y por eso están apareciendo todas estas otras reglas, agregadas automáticamente por el pattern para, por ejemplo, permitir invocar a la transacción desde el Work With, pasándole el modo (insert, update, delete) y el identificador de atracción.

Para entender de forma más simple lo que nos proponemos, preferimos tener la transacción invocable directamente, sin parámetros, tal como estaba antes de aplicarle el pattern. Podemos desaplicárselo, eliminando, recordemos, de aquí la instancia... o, dado que lo necesitaremos para luego, grabar esta transacción con otro nombre; por ejemplo este.

Parallel transactions



Name	Type	Description	Formula	N
Attraction	Attraction	Attraction		
AttractionId	Id	Attraction Id		No
AttractionName	Name	Attraction Name		No
CountryId	Id	Country Id		No
CountryName	Name	Country Name		
CityId	Id	City Id		...
CityName	Name	City Name		
CategoryId	Id	Category Id		...
CategoryName	Name	Category Name		
AttractionPhoto	Image	Attraction Photo		No
AttractionAddress	Address, Gen...	Attraction Address		No

Name	Type	Description	Formula	N
AttractionWithoutParameters	AttractionWithoutPa...	Attraction ...		
AttractionId	Id	Attraction Id		No
AttractionName	Name	Attraction N...		No
CountryId	Id	Country Id		No
CountryName	Name	Country Name		
CityId	Id	City Id		...
CityName	Name	City Name		
CategoryId	Id	Category Id		...
CategoryName	Name	Category Na...		
AttractionPhoto	Image	Attraction P...		No
AttractionAddress	Address, GeneXus	Attraction A...		No

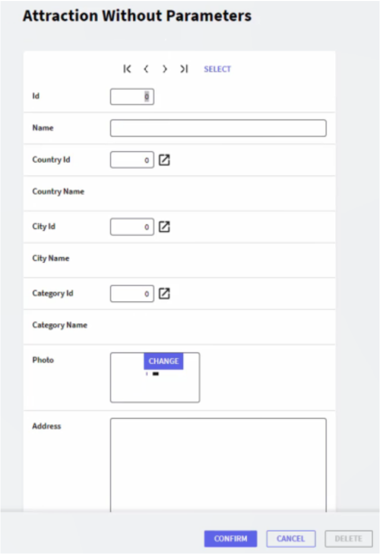
```
Error( "The attraction name must not be empty")
```

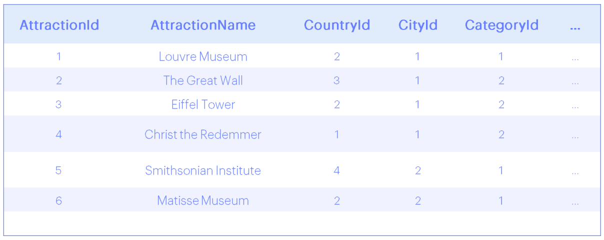
Table: ATTRACTION

Esta transacción será idéntica a la otra, salvo por el nombre y porque no tiene el pattern aplicado, aunque las reglas que venían del pattern y los eventos quedaron, así que simplemente los eliminamos. Dejamos la regla de error y eliminamos todos los eventos que había agregado el patrón. La tabla sobre la que insertará, modificará y eliminará registros esta transacción vemos que es la misma exactamente que la tabla de la transacción Attraction. ¿Por qué? Porque el identificador es el mismo. Estas transacciones reciben el nombre de transacciones paralelas. Comparten la tabla, pero los programas son independientes.
A esta, por ejemplo, podríamos quitarle incluso la regla de error, y por tanto permitiría insertar registros que la otra no.

GeneXus

Transaction: Insert, Update, Delete



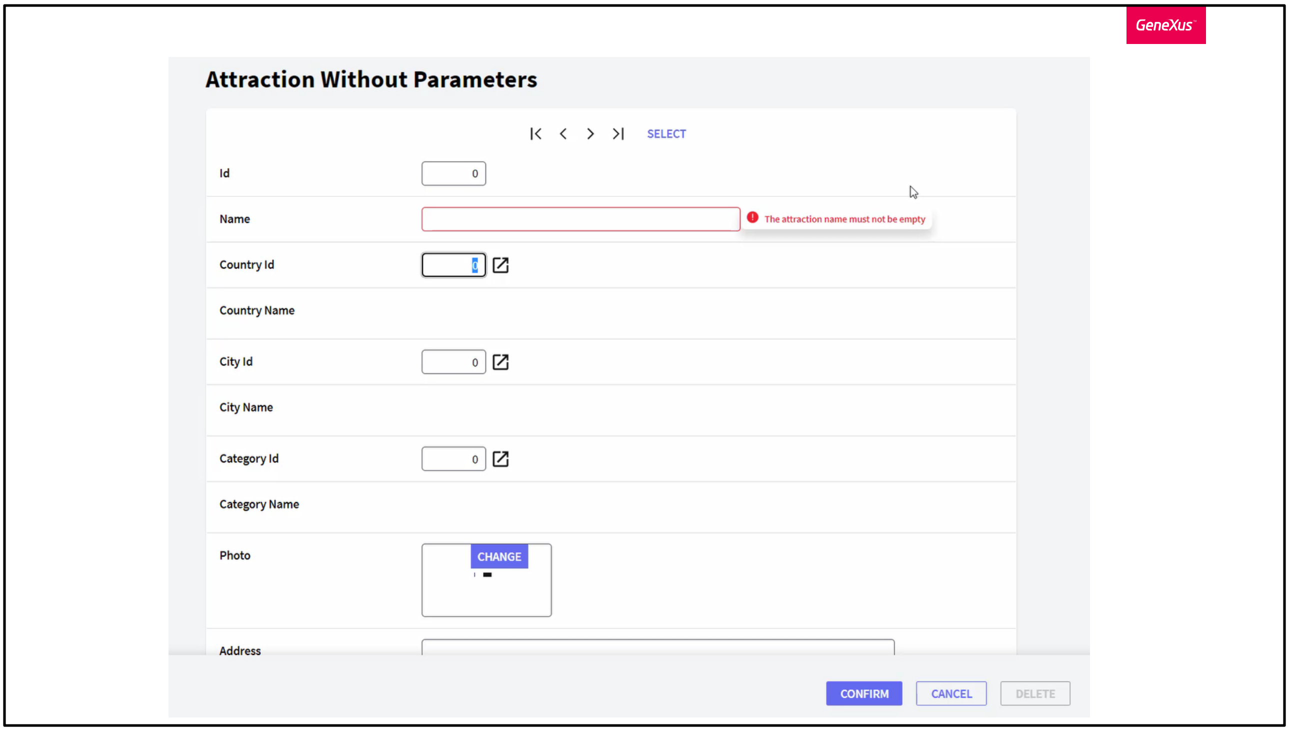


AttractionId	AttractionName	CountryId	CityId	CategoryId	...
1	Louvre Museum	2	1	1	...
2	The Great Wall	3	1	2	...
3	Eiffel Tower	2	1	2	...
4	Christ the Redemmer	1	1	2	...
5	Smithsonian Institute	4	2	1	...
6	Matisse Museum	2	2	1	...

Hagamos F5 para ejecutarla.

Vemos que se abre con los campos vacíos, y activo el del identificador, a la espera de que el usuario ingrese allí un valor, para poder inferir luego de qué operación se tratará. Si de una inserción o, por el contrario una actualización o eliminación.

Así, al salir del campo GeneXus buscará en la tabla si existe un registro con el valor indicado. En caso de existir, la transacción quedará en modo Update, y con los campos con los valores correspondientes. En cambio, en caso de no existir, la transacción quedará en modo Insert y todos los campos estarán vacíos (a menos que hubiera alguna regla Default que dependa solo del modo, o alguna asignación que solo esté condicionada con If Insert. Por ejemplo, AttractionName igual algo if Insert. En este caso no hay ninguna).

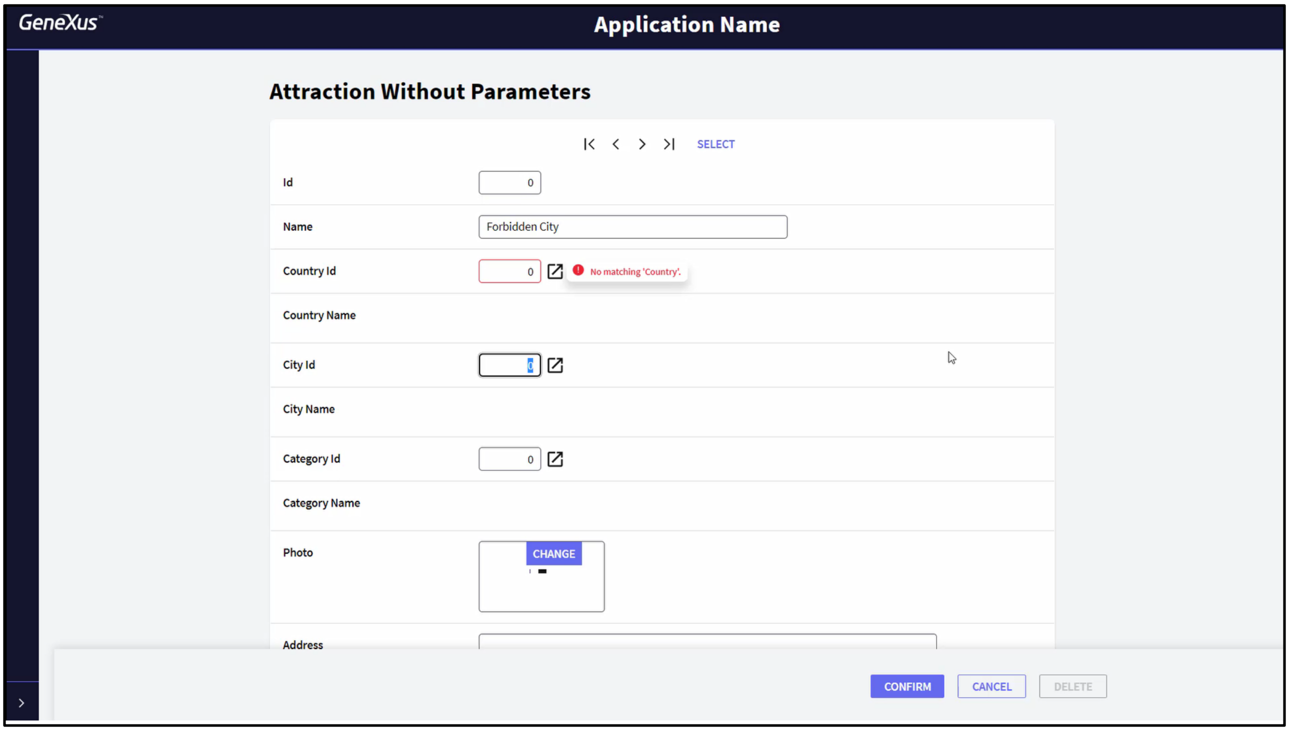


Attraction Without Parameters

Id	0
Name	
Country Id	0
Country Name	
City Id	0
City Name	
Category Id	0
Category Name	
Photo	
Address	

The attraction name must not be empty

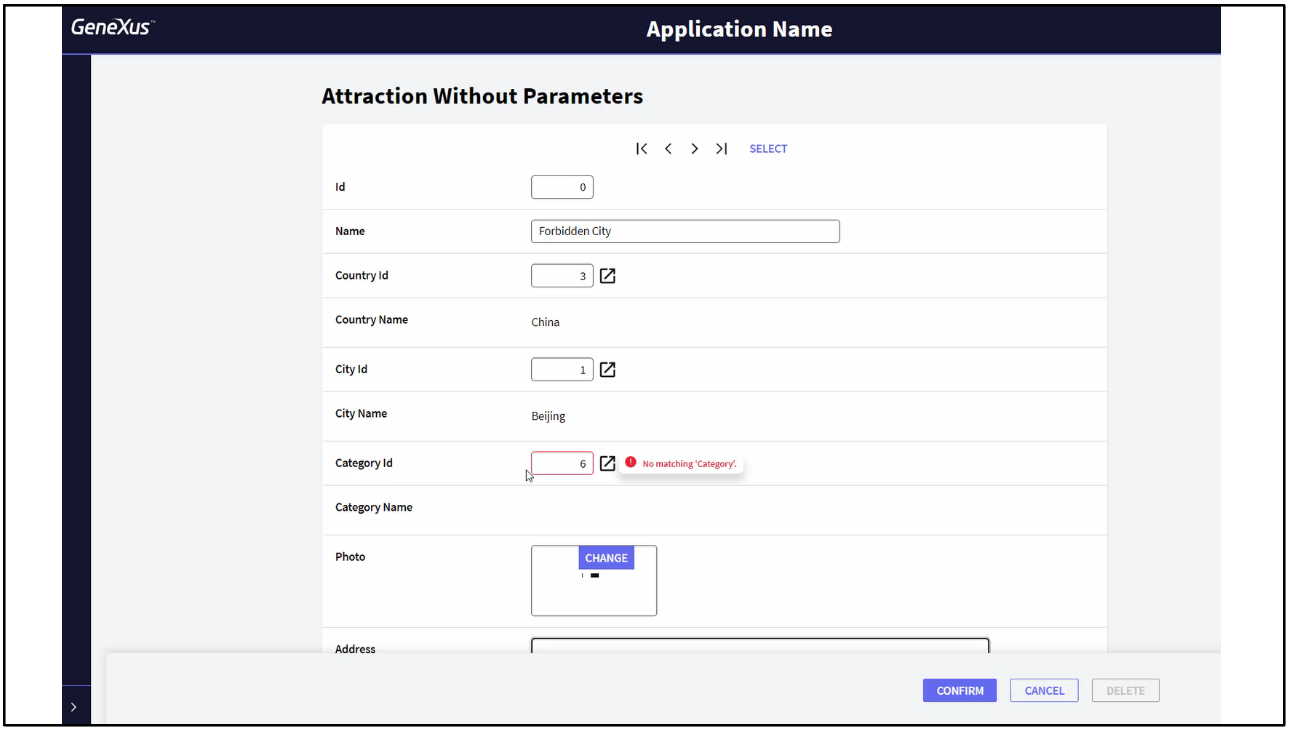
Como AttractionId es autonumerado, el usuario probablemente deje el valor 0 cuando quiera insertar una nueva atracción. La transacción, entonces, quedará en modo Insert. Si ahora sale del campo siguiente, AttractionName, sin ingresarle un valor, se le desplegará un mensaje de error, por la regla que teníamos programada. Pero incluso así, nos permite continuar ingresando los demás porque de todos modos al Confirmar no se nos permitirá grabar, seguirá mostrándonos el mensaje. Ingresemos entonces la ciudad prohibida.



Attraction Without Parameters

Id	0
Name	Forbidden City
Country Id	0 No matching 'Country'.
Country Name	
City Id	0
City Name	
Category Id	0
Category Name	
Photo	
Address	

Cuando se trata de claves foráneas, al salir del campo se chequea que un registro con ese valor exista en la tabla asociada. De lo contrario, arroja el error “No matching” de la integridad referencial. También podemos continuar, pero, al igual que con el otro error, tampoco nos dejará grabar. La ciudad prohibida está en China, y vemos que al elegir su clave ya nos infiere su nombre. Lo mismo ocurre para la ciudad.



Attraction Without Parameters

Id	0
Name	Forbidden City
Country Id	3
Country Name	China
City Id	1
City Name	Beijing
Category Id	6
Category Name	
Photo	
Address	

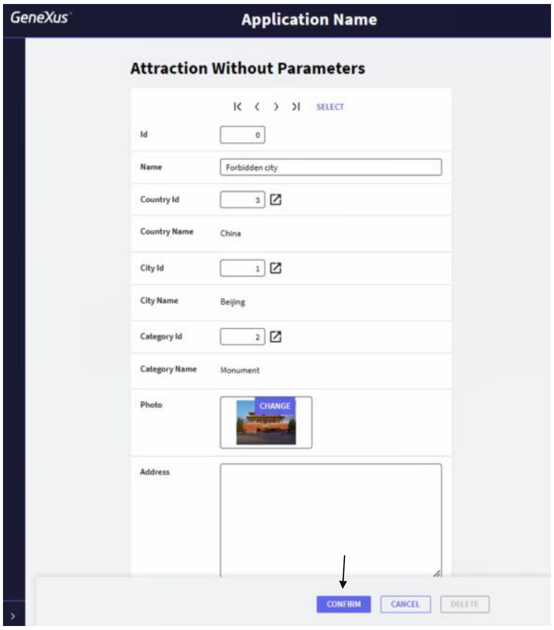
No matching 'Category'.

Aquí tenemos otra clave foránea, la categoría, que no nos está permitido dejar con un valor inexistente, pero sí vacía. Recordemos que indicamos que el atributo aceptaría nulos. Pero en realidad sabemos su categoría, será Monument.

Observemos que para realizar estos controles de integridad referencial y traer los nombres inferidos no tuvo más remedio que ir al servidor, que es el que verdaderamente controla la base de datos. Todas las reglas y controles que se van disparando campo a campo en el navegador del usuario están allí para dar agilidad a la experiencia de usuario, haciéndole sentir que no hay tiempos muertos. A esto le llamamos Client Side Validation, es decir, validación del lado del cliente. Pero todo esto tendrá que repetirse en el servidor cuando el usuario confirme.

Por supuesto, podemos dejar vacíos los campos que no sean claves foráneas default, es decir que no acepten nulos, y los que no tengan reglas explícitas que lo impidan. Coloquemos foto, pero no dirección.

Y ahora sí, confirmemos. Nos está informando que la inserción se realizó exitosamente. Pero ¿qué es lo que pasó detrás?



Attraction Without Parameters

Id	0
Name	Forbidden city
Country Id	3
Country Name	China
City Id	1
City Name	Beijing
Category Id	2
Category Name	Monument
Photo	
Address	

AttractionId	AttractionName	CountryId	CityId	CategoryId	...
1	Louvre Museum	2	1	1	...
2	The Great Wall	3	1	2	...
3	Eiffel Tower	2	1	2	...
4	Christ the Redemmer	1	1	2	...
5	Smithsonian Institute	4	2	1	...
6	Matisse Museum	2	2	1	...
7	Forbidden city	3	1	2	

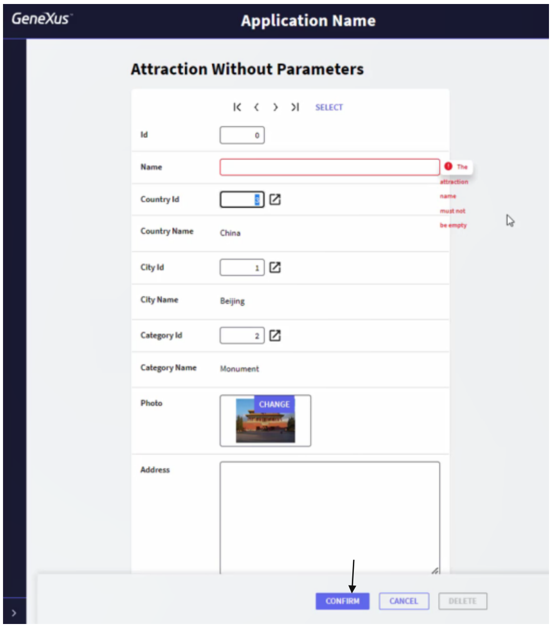
AttractionId	0
AttractionName	Forbidden city
CountryId	3
CountryName	
CityId	1
CityName	
CategoryId	2
CategoryName	
AttractionPhoto	
AttractionAddress	

Al presionar Confirm toda la información ingresada por el usuario en los campos deberá viajar al servidor, que volverá a empezar de cero, porque es él el que asegura que no haya violaciones de seguridad. El browser siempre es un medio hostil. Es el servidor el encargado de que el programa haga lo que tiene codificado. Y en tiempo imperceptible para el usuario. Y es él el único que tiene permitido operar sobre la base de datos.

Podemos imaginar, solo a efectos prácticos, que es como si se tomaran todos los atributos de la estructura de la transacción y se armara con ellos un SDT que se carga con los valores que interactivamente les dio el usuario en el form (los que importan son los no virtuales, es decir, los que estarán físicamente en la tabla; aquí son estos).

Con esta estructura cargada en el server se ejecuta todo de cero: las validaciones, reglas y fórmulas, pasando por cada elemento al igual que lo hizo el usuario interactivamente.

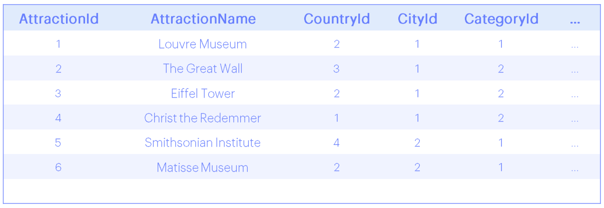
Si logra finalizar sin errores, entonces inserta el registro en la base de datos.



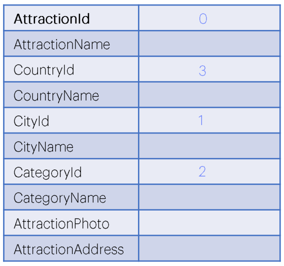
Attraction Without Parameters

Id	0
Name	
Country Id	3
Country Name	China
City Id	1
City Name	Beijing
Category Id	2
Category Name	Monument
Photo	
Address	

The attraction name must not be empty

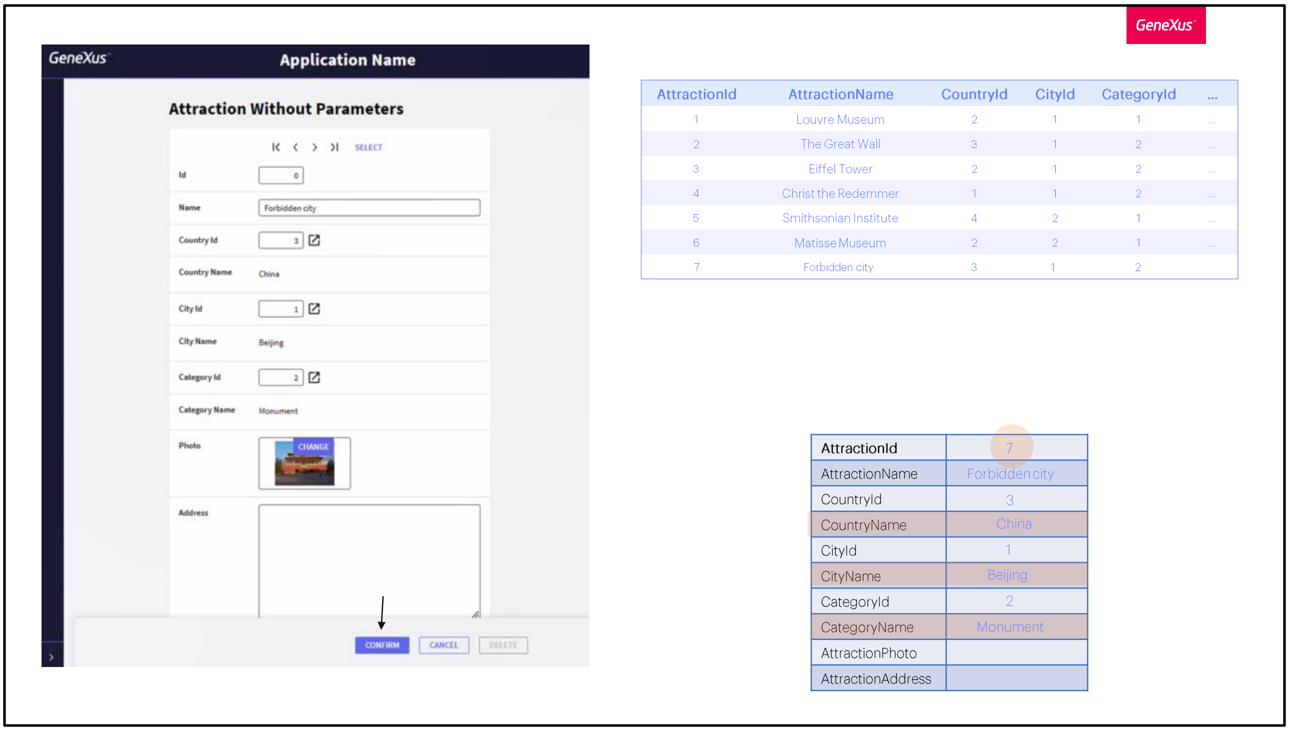


AttractionId	AttractionName	CountryId	CityId	CategoryId	...
1	Louvre Museum	2	1	1	...
2	The Great Wall	3	1	2	...
3	Eiffel Tower	2	1	2	...
4	Christ the Redemmer	1	1	2	...
5	Smithsonian Institute	4	2	1	...
6	Matisse Museum	2	2	1	...



AttractionId	0
AttractionName	
CountryId	3
CountryName	
CityId	1
CityName	
CategoryId	2
CategoryName	
AttractionPhoto	
AttractionAddress	

Si al presionar Confirm hubiésemos dejado vacío el AttractionName, saltará la regla de error y no se permitirá la inserción en la base de datos, mostrándosele el error al usuario en el browser. Lo mismo sucederá si dejamos un valor de clave foránea inexistente.



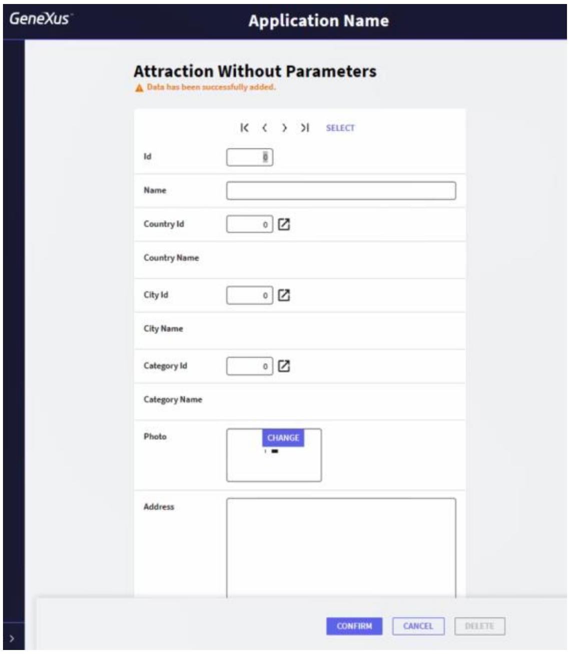
Attraction Without Parameters

Id	0
Name	Forbidden city
Country Id	3
Country Name	China
City Id	1
City Name	Beijing
Category Id	2
Category Name	Monument
Photo	
Address	

AttractionId	AttractionName	CountryId	CityId	CategoryId	...
1	Louvre Museum	2	1	1	...
2	The Great Wall	3	1	2	...
3	Eiffel Tower	2	1	2	...
4	Christ the Redemmer	1	1	2	...
5	Smithsonian Institute	4	2	1	...
6	Matisse Museum	2	2	1	...
7	Forbidden city	3	1	2	

AttractionId	7
AttractionName	Forbidden city
CountryId	3
CountryName	China
CityId	1
CityName	Beijing
CategoryId	2
CategoryName	Monument
AttractionPhoto	
AttractionAddress	

Pero si todo anduvo bien, en este caso como AttractionId es autonumerado, al insertarlo en la tabla se le dará el número que corresponda y el SDT también quedará actualizado, con ese dato y con los datos inferidos y fórmulas que correspondan (aquí no hay fórmulas) por si hubiera que hacer algo más con él (si se tratara de una transacción de dos niveles, todavía faltaría trabajar con las líneas).



Application Name

Attraction Without Parameters

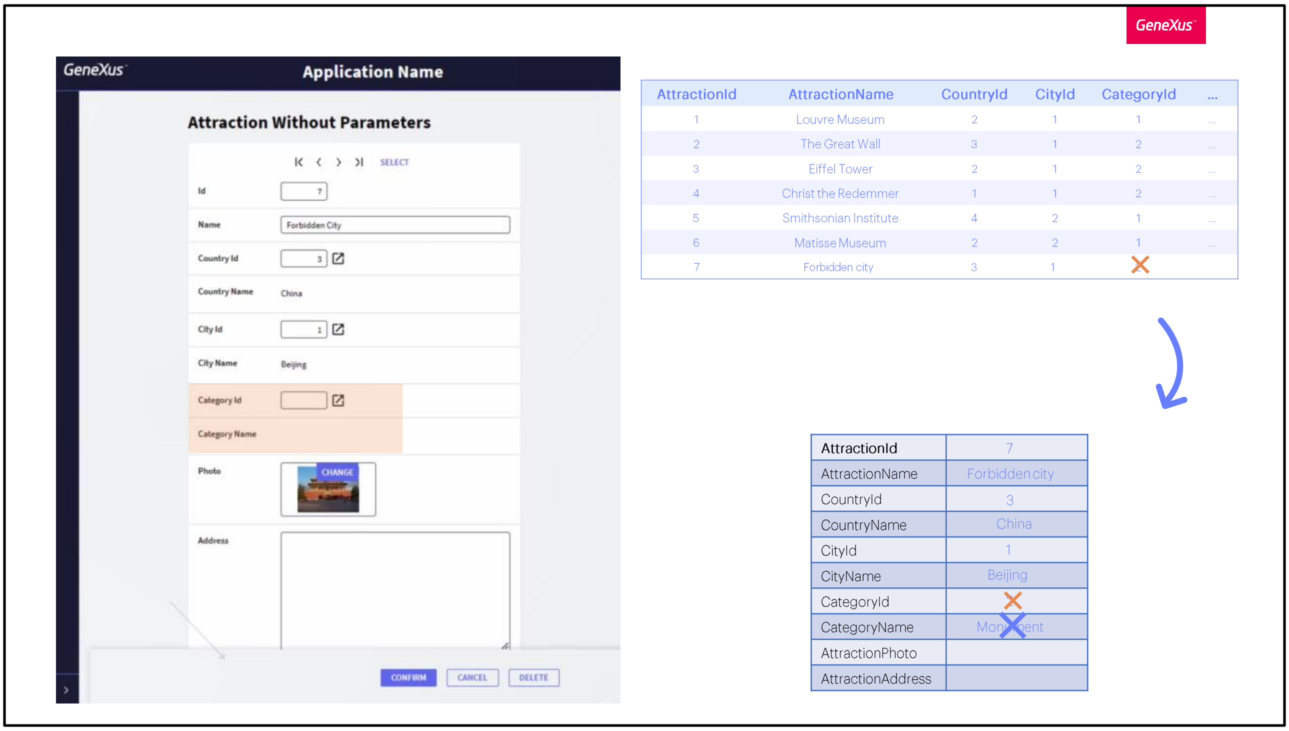
⚠ Data has been successfully added.

Id	0
Name	
Country Id	0
Country Name	
City Id	0
City Name	
Category Id	0
Category Name	
Photo	
Address	

AttractionId	AttractionName	CountryId	CityId	CategoryId	...
1	Louvre Museum	2	1	1	...
2	The Great Wall	3	1	2	...
3	Eiffel Tower	2	1	2	...
4	Christ the Redemmer	1	1	2	...
5	Smithsonian Institute	4	2	1	...
6	Matisse Museum	2	2	1	...
7	Forbidden city	3	1	2	

AttractionId	
AttractionName	
CountryId	
CountryName	
CityId	
CityName	
CategoryId	
CategoryName	
AttractionPhoto	
AttractionAddress	

Lo que vemos en ejecución es que luego de la inserción la pantalla se vacía, con el mensaje de que los datos fueron exitosamente ingresados. La transacción vuelve a quedar en modo Insert, es decir, vuelve a estar lista para que el usuario ingrese una nueva atracción. Podemos pensar también que se elimina esa estructura en el server, para ser nuevamente creada cuando se reinicie el proceso.



Attraction Without Parameters

Id	7
Name	Forbidden City
Country Id	3
Country Name	China
City Id	1
City Name	Beijing
Category Id	
Category Name	
Photo	
Address	

AttractionId	AttractionName	CountryId	CityId	CategoryId	...
1	Louvre Museum	2	1	1	...
2	The Great Wall	3	1	2	...
3	Eiffel Tower	2	1	2	...
4	Christ the Redemmer	1	1	2	...
5	Smithsonian Institute	4	2	1	...
6	Matisse Museum	2	2	1	...
7	Forbidden city	3	1	~~2~~	

AttractionId	7
AttractionName	Forbidden city
CountryId	3
CountryName	China
CityId	1
CityName	Beijing
CategoryId	~~2~~
CategoryName	~~Monument~~
AttractionPhoto	
AttractionAddress	

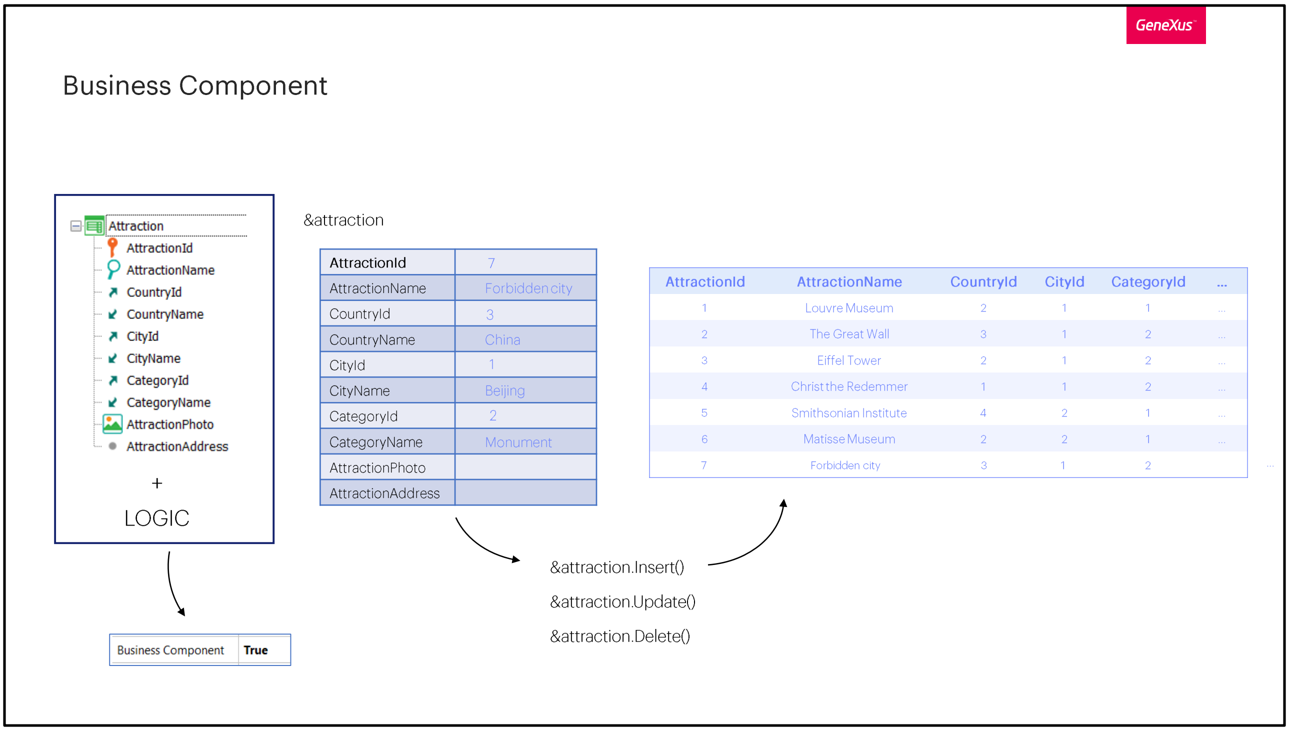
Si ahora ingresamos en la clave este valor, 7, y abandonamos el campo...lo que vemos en el browser será esto. La transacción habrá ido al servidor que irá a la base de datos a consultar por la existencia de un registro con ese valor. Lo encontrará. Podemos, otra vez, pensar que carga todos sus valores (los físicos y los inferidos y fórmulas) en una estructura como la anterior y la envía al cliente, con la información de que, entonces ahora el modo de la transacción será Update.

Otra vez, el usuario interactivamente hará las modificaciones que desee, por ejemplo, elimina la categoría (que como acepta nulos, será permitido). Al confirmar, todo se vuelve a realizar en el server: se carga el registro de la base de datos, se le realizan las modificaciones hechas en el cliente, y se va validando campo a campo, disparando las reglas que correspondan. Si nada falla, se actualizará el registro de la tabla, en este caso quitando la categoría, y actualizará la estructura, quitando también el nombre de categoría, que era inferido.

Si ninguna otra regla de la transacción lo impide, entonces en el browser queda la información actualizada, y un mensaje que lo indica. Observemos que la transacción queda en modo Update. Podríamos volver a modificar este registro, por ejemplo, agregándole otra vez la categoría.

Si ahora quisiéramos eliminarla, bastaría con presionar el botón Delete, y en el server, con la estructura cargada será fácil ir a buscar a la tabla la atracción 7 para eliminarla.

Con esto ya nos queda bastante claro que si lográramos trabajar con una variable estructurada como la que imaginamos utiliza internamente la transacción en el servidor, que encapsule las reglas de la transacción y además nos permita realizar las operaciones sobre la tabla, podríamos insertar, modificar y eliminar de la base de datos por código, respetando la lógica declarada en la transacción.



Business Component

Business Component	**True**

&attraction

AttractionId	7
AttractionName	Forbidden city
CountryId	3
CountryName	China
CityId	1
CityName	Beijing
CategoryId	2
CategoryName	Monument
AttractionPhoto	
AttractionAddress	

&attraction.Insert()

&attraction.Update()

&attraction.Delete()

AttractionId	AttractionName	CountryId	CityId	CategoryId	...
1	Louvre Museum	2	1	1	...
2	The Great Wall	3	1	2	...
3	Eiffel Tower	2	1	2	...
4	Christ the Redemmer	1	1	2	...
5	Smithsonian Institute	4	2	1	...
6	Matisse Museum	2	2	1	...
7	Forbidden city	3	1	2	...

Esto es ni más ni menos que lo que se conoce como Business Component.

De la estructura de la transacción con su lógica (y por lógica entendemos los controles de duplicados –no sólo los de clave primaria, sino también de claves candidatas-, de integridad referencial, sus reglas y algunos de sus eventos), se obtiene una suerte de tipo de datos similar a un SDT, pero mucho más potente.

Luego, alcanzará con definir en casi cualquier programa una variable de ese tipo de datos y manipularla, que es lo que veremos a continuación.

GeneXus™

training.genexus.com
wiki.genexus.com