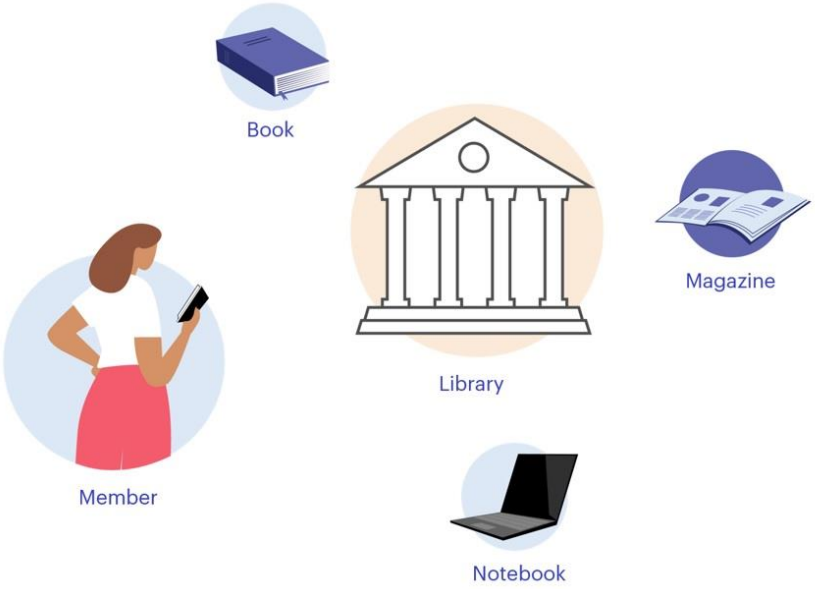
Análisis del modelo de diseño de transacciones

Biblioteca

GeneXus

Biblioteca



A lo largo del curso anterior, hemos estudiado todo lo necesario para poder modelar correctamente una determinada realidad en GeneXus. En este video, a través del estudio de una realidad acotada, analizaremos diferentes opciones para resolver el diseño de transacciones, utilizando un conjunto de recursos esenciales y dando solución a una serie de requerimientos reales.

Supongamos que se necesita diseñar una aplicación GeneXus para la gestión de tareas de una Biblioteca en relación al préstamo de libros, revistas y notebooks.
La misma trabaja con Socios que pueden acceder a diferente material de lectura, ya sea libros o revistas, y a equipos informáticos que la Biblioteca ofrece en calidad de préstamo.

Entidades

Country

- Id
- Name

Member

- Document
- Name
- Address
- Image
- Phone
- Over 20 years old

Author

- Id
- Name
- Image
- Country
- May have several books published

La información que se necesita registrar es la siguiente:

País (Country)
Todo País se registra con un identificador único y su nombre.

Socio (Member)
Todo Socio de la Biblioteca se registra con su Documento de identidad, nombre, dirección, foto y un teléfono de contacto.

Los socios de la Biblioteca deben ser mayores de 20 años.

Autor (Author)
Todo Autor se registra con un identificador único, su nombre, foto y país de origen. Un Autor puede tener varios libros publicados.

Entidades

Country

- Id
- Name

Book

- Id
- Title
- Edition Date
- Copies
- Literary Genre
- Author
- Publishing House in charge of its publication

Member

- Document
- Name
- Address
- Image
- Phone
- Over 20 years old

Magazine

- Id
- Title
- Publication Date
- Image
- Copies

Author

- Id
- Name
- Image
- Country
- May have several books published

Notebook

- Id
- Image
- Description
- Status

Libro (Book)
Todo Libro se registra con un identificador único, título, fecha de edición y cantidad de ejemplares que dispone la Biblioteca. Puede tratarse de una Novela, un Ensayo, un libro de Poesía, etc, por lo tanto, un libro corresponde a un género literario.

Además, un Libro tiene un Autor y una Editorial a cargo de su publicación.

Revista (Magazine)
Toda Revista se registra con un identificador único, título, fecha de publicación, la imagen de la portada y la cantidad de ejemplares disponible.

Notebook (Notebook)
La Biblioteca ofrece notebooks a sus Socios, en calidad de préstamo, ya que muchos son escritores e investigadores. Cada notebook se registra con un identificador único, su imagen y una breve descripción.

Además, se registra su estado (disponible o en préstamo).

Entidades

Country

- Id
- Name

Book

- Id
- Title
- Edition Date
- Copies
- Literary Genre
- Author
- Publishing House in charge of its publication

Loan

- Id
- Date
- Member
- Return Date
- 15-day deadline
- may include a maximum of 3 books, 4 magazines, and may or may not include borrowing a laptop
- Comment

Member

- Document
- Name
- Address
- Image
- Phone
- Over 20 years old

Magazine

- Id
- Title
- Publication Date
- Image
- Copies

Book Request

- Id
- Date
- Publishing House
- Book
- Number of copies

Author

- Id
- Name
- Image
- Country
- May have several books published

Notebook

- Id
- Image
- Description
- Status

Préstamo (Loan)
La Biblioteca ofrece libros, revistas y notebooks en calidad de préstamo a sus Socios.

Todo préstamo se registra con un identificador único, la fecha de retiro, el Socio, y la fecha de devolución que se debe determinar automáticamente.
Todo préstamo se realiza por un plazo de 15 días, puede incluir un máximo de 3 libros, 4 revistas, y puede o no incluir el préstamo de una notebook.

Solamente se puede retirar un ejemplar de cada publicación, y es posible registrar algún comentario que se considere necesario a nivel del préstamo de cada ejemplar (por ejemplo, que la tapa se encuentre dañada, que le falte alguna hoja, etc).

Además, la fecha en que se registra un nuevo préstamo siempre es la actual y no debe ser posible modificarla.

Solicitud de ejemplares (BookRequest)
Muchas veces ocurre que ciertos libros tienen mucha demanda, y la Biblioteca decide solicitar más cantidad de ejemplares.

Para eso, realiza una Solicitud a la Editorial correspondiente. El sistema debe controlar que se soliciten ejemplares de libros a cargo de la Editorial indicada.

Dominios

Country

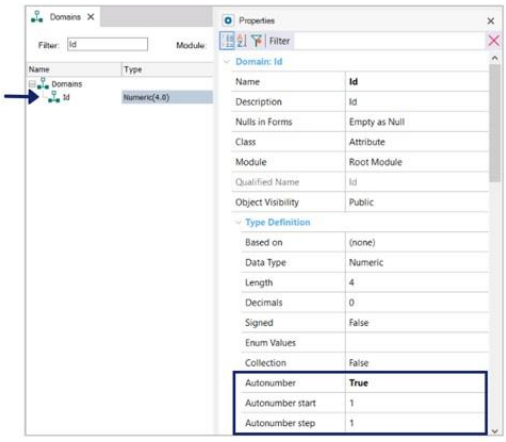
- Id
- Name

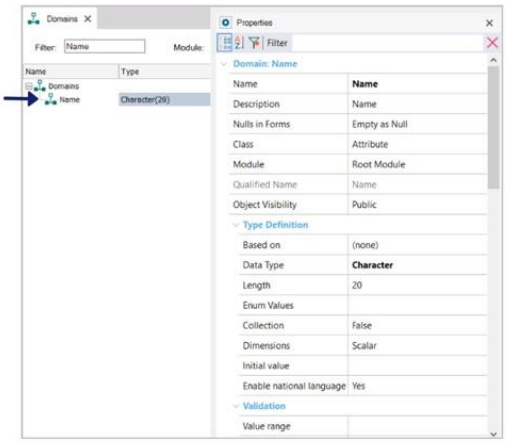
Member

- Document
- Name
- Address
- Image
- Phone
- Over 20 years old

Author

- Id
- Name
- Image
- Country
- May have several books published





Comencemos a analizar esta realidad.

En principio, claramente se distinguen ciertas entidades simples que podemos comenzar a definir, como lo son, por ejemplo, País (Country), Socio (Member) y Autor (Author).

Pero antes, es bueno tomar en cuenta que la mayoría de las entidades se registran con un identificador que perfectamente puede ser autonumerado, a excepción del Nro de Socio, que como mencionamos, se registra con su documento de identidad.
Por lo tanto, definimos el Dominio Id, con la propiedad Autonumber en True.

Definimos también el dominio Name, como carácter de 20.

Definición de transacciones: Country

- Id
- Name

Name	Type	Description
Country	Country	Country
CountryId	Id	Country Id
CountryName	Name	Country Name

Attribute	Order
Country Indexes	
ICountry	Primary Key
CountryId	Ascending
UCountry	Unique
CountryName	Ascending

Vamos a empezar definiendo la transacción Country, con **CountryId** como llave primaria, y **CountryName** como atributo secundario.
CountryId queda basado en el dominio Id, y para controlar que el nombre no se repita, definimos el correspondiente índice unique. Esta misma definición vale en todas las entidades donde es necesario controlar que el nombre no se repita.

Definición de transacciones: Country

- Id
- Name

Weak 1-N

Name	Type	Description
Country	Country	Country
CountryId	Id	Country Id
CountryName	Name	Country Name
City	City	City
CityId	Id	City Id
CityName	Name	City Name

Attribute	Order
Country Indexes	
ICountry	Primary Key
CountryId	Ascending
UCountry	Unique
CountryName	Ascending

Name	Type
CountryCity Structure	
CountryId	Id
CityId	Id
CityName	Name

Si fuera necesario registrar las ciudades de cada país, porque es de interés conocer la ciudad de nacimiento del autor, podríamos modelarla como una entidad débil respecto del país, ya que una ciudad no existe fuera de ese contexto.

Así que se agregaría un segundo nivel a la transacción Country, pero al considerar City como una entidad débil no existirá como una transacción en sí misma.

Esto significa que CityId no existirá como llave primaria en ninguna tabla, y por lo tanto, para poder conocer, por ejemplo, la ciudad de nacimiento de un autor, se necesitará el par compuesto por los atributos CountryId, CityId, que será la llave primaria de la tabla COUNTRYCITY asociada al segundo nivel de la transacción Country.

De todas formas, como la descripción de la realidad que estamos estudiando no incluye el concepto de la ciudad, vamos a modelar el País (Country) como una transacción simple.

Definición de transacciones: Member

- Document
- Name
- Address
- Image
- Phone
- Over 20 years old

Name	Type	Description	Formula
Member	Member	Member	
MemberDocument	Numeric(8.0)	Member Document	
MemberName	Name	Member Name	
MemberImage	Image	Member Image	
MemberAddress	Address, GeneXus	Member Address	
MemberPhone	Phone, GeneXus	Member Phone	
MemberBirthDate	Date	Member Birth Date	
MemberAge	Numeric(3.0)	Member Age	age(MemberBirthDate, today())

Member

Structure | Web Layout | Win Form | **Rules** | Events | Variables | Help | Documentation

```
1 Error("The member must be over 20 years old")
2     if MemberAge <= 20;
```

Pasemos ahora al Socio. Para eso definimos la transacción Member, con los siguientes atributos:

- MemberDocument, que corresponde al documento de identidad y por lo tanto no es autonumerado, así que lo definimos como numérico de 8 dígitos
- MemberName, de tipo Name
- MemberImage, de tipo Image
- MemberAddress, basado en el dominio semántico Address
- y MemberPhone, de tipo Phone

Pero, además, la realidad nos dice que los socios deben ser mayores de 20 años, por lo tanto necesitamos su fecha de nacimiento y la edad, que deberá ser calculada en forma automática.
Así que agregamos a la estructura de la transacción los atributos:

- MemberBirthDate, de tipo Date
- y MemberAge, numérico de 3 dígitos

¿Cómo calculamos la edad del socio? Utilizando la función Age, que calcula la edad a partir de la fecha de nacimiento y la fecha actual.
Así que definimos MemberAge como un atributo calculado que obtiene su valor a partir de la siguiente expresión:
Age(MemberBirthDate, Today())

¿Por qué utilizamos la función Today() y no la variable &today?
Porque no es posible utilizar variables en la declaración de fórmulas.

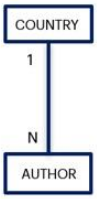
Para controlar que todo socio sea mayor de 20 años alcanza con declarar la siguiente regla Error:
Error(“The member must be over 20 years old”) if MemberAge <= 20;

Es importante destacar que no estamos considerando en este análisis la declaración de reglas para el control básico de ingreso de datos, como por ejemplo, controlar que no se ingrese un socio sin nombre, etc.

Definición de transacciones: Author

- Id
- Name
- Image
- Country
- May have several books published

Name	Type	Description
Author	Author	Author
AuthorId	Id	Author Id
AuthorName	Name	Author Name
AuthorImage	Image	Author Image
CountryId	Id	Country Id
CountryName	Name	Country Name



Consideremos ahora a los Autores. Debemos definir la transacción Author, con los atributos:

- AuthorId
- AuthorName
- AuthorImage

Y sabemos que se debe registrar el país del mismo, ya que todo Autor tiene un país de nacimiento, por lo tanto, agregamos CountryId y CountryName, donde CountryId es una clave foránea y CountryName un atributo que se infiere a partir de dicha clave foránea.
De esta forma representamos que entre Country y Author existe una relación 1-N.

Definición de transacciones: Book

- Id
- Title
- Edition Date
- Copies
- Literary Genre
- Author
- Publishing House in charge of its publication

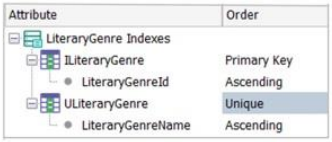
Pensemos ahora en el concepto del Libro. Se trata de un concepto fuerte que se identifica también con un valor autonumerado, tiene un título, una fecha de edición, un autor y también la cantidad de ejemplares adquiridos por la Biblioteca.

Pero, además, la realidad nos dice que un libro corresponde a un género literario, ya que puede tratarse de una Novela, un Ensayo, etc.

Entonces, ¿cómo modelamos el concepto del Género literario? Si pensamos que estos géneros son finitos, una primera opción que podemos considerar es la creación de un dominio enumerado.

Definición de transacciones: Book

- Id
- Title
- Edition Date
- Copies
- Literary Genre
- Author
- Publishing House in charge of its publication



Pero la categorización de los géneros literarios ha ido cambiando a lo largo del tiempo y es posible que siga haciéndolo, entonces no parece que el dominio enumerado sea la mejor opción, ya que los cambios deberían realizarse en forma manual y no sería escalable.

Podemos considerar una transacción GeneroLiterario (LiteraryGenre), con los atributos:
- LiteraryGenreId
- y LiteraryGenreName

Es una buena decisión definir desde ya un índice unique sobre el nombre del género para evitar el registro de géneros literarios con el mismo nombre.

Pero el Libro requiere registrar también el Autor y la Editorial responsable de su publicación. El Autor ya lo tenemos definido como entidad, pero la Editorial no. Y si bien no está declarada explícitamente, en el análisis surge la necesidad de modelar la entidad Editorial.

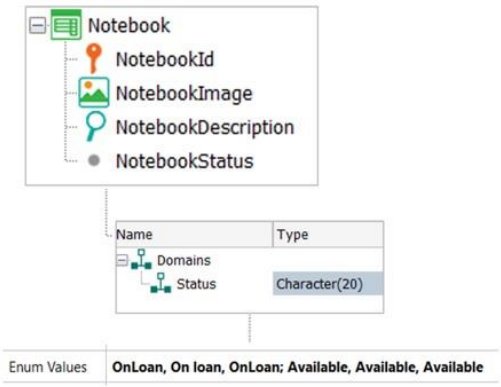
Así que definimos la transacción PublishingHouse con los atributos:
- PublishingHouseId
- y PublishingHouseName

También podemos desde ya crear el correspondiente índice unique sobre el nombre de la editorial.

Definición de transacciones: Magazine and Notebook

- Id
- Title
- Publication Date
- Image
- Copies

- Id
- Image
- Description
- Status

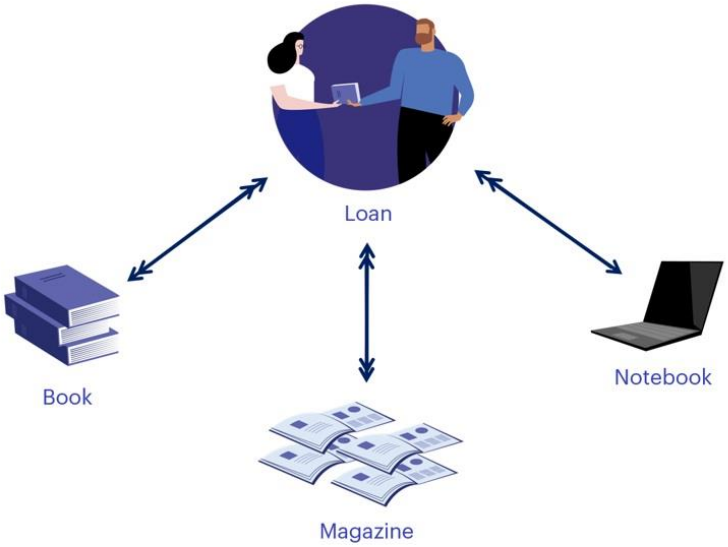


La Biblioteca también ofrece Revistas y Notebooks a sus socios, así que definimos la transacción Magazine, con los atributos:
- MagazineId
- MagazineTitle
- MagazineImage
- MagazinePublicationDate
- y MagazineCopies para registrar la cantidad de ejemplares adquiridos.

Y también la transacción Notebook, con los atributos:
- NotebookId
- NotebookImage
- NotebookDescription de tipo LongVarChar
- y NotebookStatus

Una notebook está “Disponible” o “En préstamo”, así que definimos el dominio enumerado Status con esos valores.

Definición de transacciones: Loan



Analicemos ahora el concepto del Préstamo, que es una entidad fundamental en esta realidad.

La Biblioteca permite que un socio retire, en calidad de préstamo y por 15 días, una notebook, 3 libros y 4 revistas.

Tenemos entonces una relación 1-N entre las entidades Préstamo y Notebook, y una relación N-N entre Préstamo y Libro, y entre Préstamo y Revista.

¿Cómo podemos modelarla?

Definición de transacciones: Loan

Name	Type	Formula	Nullable
Loan	Loan		
LoanId	Id		No
LoanDate	Date		No
MemberDocument	Numeric(8.0)		No
MemberName	Name		
MemberAddress	Address, GeneXus		
LoanReturnDate	Date	LoanDate.adddays(15)	
NotebookId	Id		Yes
NotebookDescription	Character(40)		

Analicemos una primera opción:

Creamos la transacción Loan, y vamos definiendo los siguientes atributos:
- LoanId
- LoanDate
- MemberDocument
- MemberName
- MemberAddress
- LoanReturnDate

Se solicita que la fecha de devolución se calcule automáticamente. Un préstamo se realiza por 15 días, así que declaramos el siguiente cálculo asociado al atributo LoanReturnDate:
LoanDate.AddDays(15)

Además, un préstamo puede incluir o no una notebook, así que agregamos:
- NotebookId
- y NotebookDescription

Pero en esta transacción Loan, NotebookId es una clave foránea no obligatoria, por lo tanto, indicamos su propiedad Nullable con el valor Yes.

Definición de transacciones: Loan

Name	Type	Formula	Nullable
Loan	Loan		
LoanId	Id		No
LoanDate	Date		No
MemberDocument	Numeric(8.0)		No
MemberName	Name		
MemberAddress	Address, GeneXus		
LoanReturnDate	Date	LoanDate.adddays(15)	
NotebookId	Id		Yes
NotebookDescription	Character(40)		
LoanBooksQty	Numeric(4.0)	count(LoanBookComment)	
Book	Book		
BookId	Id		
BookTitle	Character(20)		
LoanBookComment	Character(40)		

```
Error("Only 3 books can be checked out")
     if LoanBooksQty > 3;
```

Un préstamo incluye varios libros, con un máximo de 3. Así que definimos un segundo nivel para registrarlos, incluyendo un atributo LoanBookComment, para registrar algún comentario que se necesite al momento de realizar el préstamo.

¿Cómo controlamos que no se ingresen más de 3 libros?

Podemos definir un nuevo atributo LoanBooksQty, que cuente dicha cantidad y condicionar luego ese valor en una regla Error:
Error(“Only 3 books can be checked out”) if LoanBooksQty > 3;

Definición de transacciones: Loan

Name	Type	Formula	Nullable
Loan	Loan		
LoanId	Id		No
LoanDate	Date		No
MemberDocument	Numeric(8.0)		No
MemberName	Name		
MemberAddress	Address, GeneXus		
LoanReturnDate	Date	LoanDate.adddays(15)	
NotebookId	Id		Yes
NotebookDescription	Character(40)		
LoanBooksQty	Numeric(4.0)	count(LoanBookComment)	
LoanMagazinesQty	Numeric(4.0)	count(LoanMagazineComment)	
Book	Book		
BookId	Id		
BookTitle	Character(20)		
LoanBookComment	Character(40)		
Magazine	Magazine		
MagazineId	Id		
MagazineTitle	Character(20)		
LoanMagazineComment	Character(40)		

```
Error("Only 3 books can be checked out")
     if LoanBooksQty > 3;

Error("Only 4 magazines can be checked out")
     if LoanMagazinesQty > 4;
```

Pero el mismo préstamo puede incluir hasta 4 revistas, así que podemos definir otro segundo nivel, paralelo al libro, para registrar las revistas. De igual manera se puede controlar que no se retiren más de 4.

Definición de transacciones: Loan

Name	Type	Formula	Nullable
Loan	Loan		
LoanId	Id		No
LoanDate	Date		No
MemberDocument	Numeric(8.0)		No
MemberName	Name		
MemberAddress	Address, GeneXus		
LoanReturnDate	Date	LoanDate.adddays(15)	
NotebookId	Id		Yes
NotebookDescription	Character(40)		
LoanBooksQty	Numeric(4.0)	count(LoanBookComment)	
LoanMagazinesQty	Numeric(4.0)	count(LoanMagazineComment)	
Book	Book		
BookId	Id		
BookTitle	Character(20)		
LoanBookComment	Character(40)		
Magazine	Magazine		
MagazineId	Id		
MagazineTitle	Character(20)		
LoanMagazineComment	Character(40)		

```
Error("Only 3 books can be checked out")
     if LoanBooksQty > 3;

Error("Only 4 magazines can be checked out")
     if LoanMagazinesQty > 4;

Default(LoanDate, today());

noaccept(LoanDate);
```

La realidad describe claramente que la fecha de registro de un préstamo siempre debe ser la actual, sin posibilidad de ser modificada.

Para eso declaramos las siguientes reglas:
Default(LoanDate, today());
noaccept(LoanDate);

Definición de transacciones: Loan

Name	Type	Formula	Nullable
Loan	Loan		
LoanId	Id		No
LoanDate	Date		No
MemberDocument	Numeric(8.0)		No
MemberName	Name		
MemberAddress	Address, GeneXus		
LoanReturnDate	Date	LoanDate.adddays(15)	
NotebookId	Id		Yes
LoanBooksQty	Numeric(4.0)	count(LoanBookComment)	
LoanMagazinesQty	Numeric(4.0)	count(LoanMagazineComment)	
Book	Book		
BookId	Id		
BookTitle	Character(20)		
LoanBookComment	Character(40)		
Magazine	Magazine		
MagazineId	Id		
MagazineTitle	Character(20)		
LoanMagazineComment	Character(40)		

Control Info	
Control Type	**Dynamic Combo Box**
Data Source From	Attributes
Item Values	NotebookId
Item Descriptions	**NotebookDescription**
Sort Descriptions	True
Conditions	**NotebookStatus = Status.Available;**
Instantiated Attributes	
Empty Item	**True**

Si bien el objetivo de este video es concentrarnos en analizar el diseño de transacciones, vamos a sugerir una opción de implementación para que, al momento de registrar un préstamo, se ofrezcan solamente las notebooks que se encuentren en estado disponible (Available). Esta implementación podemos realizarla desde la definición de la propia transacción y por eso la incluimos.

Vamos a eliminar el atributo NotebookDescription, porque "disfrazaremos" el identificador de una notebook con su descripción. Seleccionamos NotebookId y lo definimos como un **Dynamic Combo Box**, con NotebookId en la propiedad Item Values, y NotebookDescription en la propiedad Item Descriptions.

Esto cargará todas las notebooks registradas. Para que solamente se ofrezcan las que tienen estado disponible, declaramos la siguiente condición:
NotebookStatus = Status.Available;

También debemos configurar la propiedad Empty Item con el valor True, ya que NotebookId puede no elegirse, pues puede ser nulo.

Si bien esta no es la única manera de controlar que una notebook se encuentre disponible al momento de registrar un préstamo, es una implementación que podemos resolver desde la definición de la propia transacción.

Cantidades disponibles de Libros y Revistas (Books and Magazines)

Name	Type	Formula
Book	Book	
BookId	Id	
BookTitle	Character(20)	
BookEditionDate	Date	
AuthorId	Id	
AuthorName	Name	
BookCopies	Numeric(4.0)	
LiteraryGenreId	Id	
LiteraryGenreName	Name	
PublishingHouseId	Id	
PublishingHouseName	Name	
BookOnLoanQty	Numeric(4.0)	count(LoanBookComment)

Name	Type	Formula
Magazine	Magazine	
MagazineId	Id	
MagazineTitle	Character(20)	
MagazineImage	Image	
MagazinePubicationDate	Date	
MagazineCopies	Numeric(4.0)	

Otro requerimiento necesario, es conocer la disponibilidad de los libros o revistas de interés.

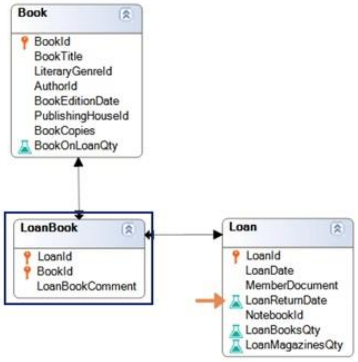
Sabemos la cantidad de ejemplares adquiridos por la Biblioteca para cada libro y revista, pero no sabemos la cantidad que va quedando disponible.

¿Podemos obtener esos valores de una forma simple y basándonos en el diseño de transacciones?
Si en la transacción Book, definimos un nuevo atributo, BookOnLoanQty, podemos calcular la cantidad de ejemplares prestados, y por lo tanto saber la cantidad que quedan disponibles.

¿Qué sucede si a ese nuevo atributo BookOnLoanQty le asociamos el cálculo Count(LoanBookComment)? ¿Sumará efectivamente la cantidad de ejemplares de ese libro que se encuentran en préstamo?
La respuesta es SÍ, porque la tabla asociada a la transacción donde definimos el cálculo es BOOK. Y la tabla donde se resuelve el cálculo es LOANBOOK, ya que queda determinada por el atributo LoanBookComment.
Entre ambas tablas existe un atributo en común que es BookId, y por lo tanto es un filtro implícito que GeneXus aplicará, o sea que ese cálculo devolverá el total de ejemplares de dicho libro que ha sido prestado.

Cantidades disponibles de Libros y Revistas (Books and Magazines)

Name	Type	Formula
Book	Book	
BookId	Id	
BookTitle	Character(20)	
BookEditionDate	Date	
AuthorId	Id	
AuthorName	Name	
BookCopies	Numeric(4.0)	
LiteraryGenreId	Id	
LiteraryGenreName	Name	
PublishingHouseId	Id	
PublishingHouseName	Name	
BookOnLoanQty	Numeric(4.0)	count(LoanBookComment,LoanReturnDate >= servernow())



Name	Type	Formula
Magazine	Magazine	
MagazineId	Id	
MagazineTitle	Character(20)	
MagazineImage	Image	
MagazinePubicationDate	Date	
MagazineCopies	Numeric(4.0)	

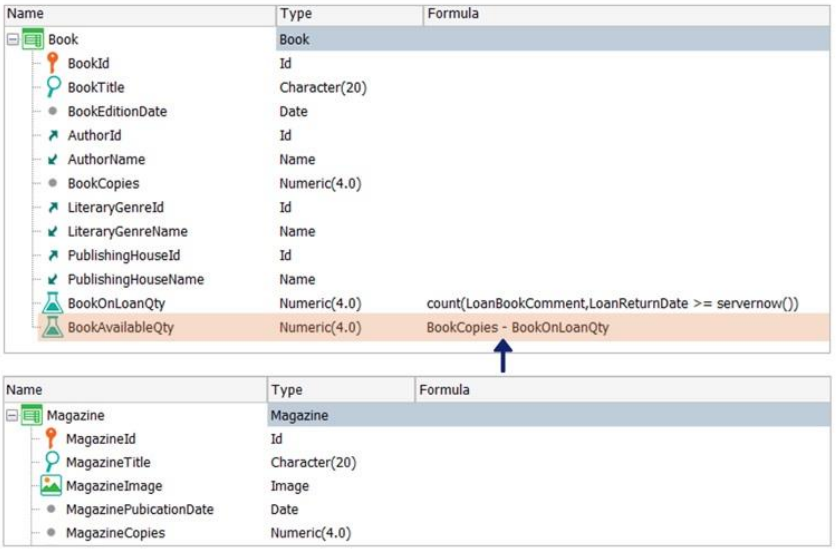
Pero necesitamos saber la cantidad de ejemplares actualmente en préstamo, así que necesitamos agregar una condición al cálculo que permita contar la cantidad de ejemplares en préstamos actualmente vigentes. Para eso consideramos los préstamos cuya fecha de devolución sea mayor o igual a la actual.

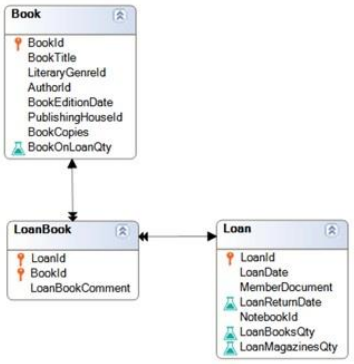
Podemos declarar lo siguiente:
Count(LoanBookComment, LoanReturnDate >= servernow())

La función servernow() permite obtener la fecha y hora actual del server. Se podría utilizar también la función Today().

¿Podemos declarar esa condición de cálculo? Sí, porque el atributo involucrado (LoanReturnDate) pertenece a la tabla extendida de la tabla donde se resuelve el cálculo declarado, que es LOANBOOK.

Cantidades disponibles de Libros y Revistas (Books and Magazines)





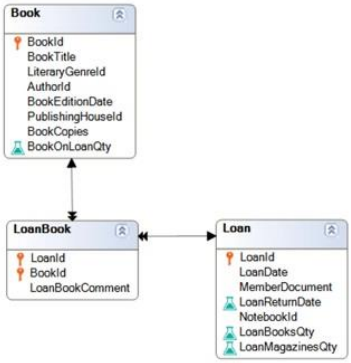
Por lo tanto, si sabemos la cantidad total de ejemplares, y la cantidad de ejemplares en préstamos vigentes, entonces podemos saber la cantidad de ejemplares actualmente disponibles.

Así que la estructura de la transacción Book nos queda así:

Cantidades disponibles de Libros y Revistas (Books and Magazines)

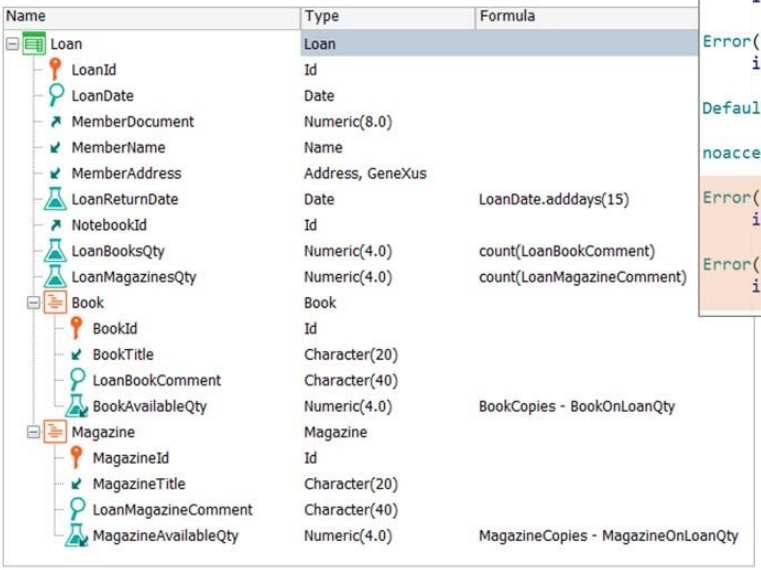
Name	Type	Formula
Book	Book	
BookId	Id	
BookTitle	Character(20)	
BookEditionDate	Date	
AuthorId	Id	
AuthorName	Name	
BookCopies	Numeric(4.0)	
LiteraryGenreId	Id	
LiteraryGenreName	Name	
PublishingHouseId	Id	
PublishingHouseName	Name	
BookOnLoanQty	Numeric(4.0)	count(LoanBookComment,LoanReturnDate >= servernow())
BookAvailableQty	Numeric(4.0)	BookCopies - BookOnLoanQty

Name	Type	Formula
Magazine	Magazine	
MagazineId	Id	
MagazineTitle	Character(20)	
MagazineImage	Image	
MagazinePubicationDate	Date	
MagazineCopies	Numeric(4.0)	
MagazineOnLoanQty	Numeric(4.0)	count(LoanMagazineComment, LoanReturnDate >= servernow())
MagazineAvailableQty	Numeric(4.0)	MagazineCopies - MagazineOnLoanQty



De igual forma podemos saber la cantidad de ejemplares disponibles de una determinada revista:

Definición de transacciones: Loan



Name	Type	Formula
Loan	Loan	
LoanId	Id	
LoanDate	Date	
MemberDocument	Numeric(8.0)	
MemberName	Name	
MemberAddress	Address, GeneXus	
LoanReturnDate	Date	LoanDate.adddays(15)
NotebookId	Id	
LoanBooksQty	Numeric(4.0)	count(LoanBookComment)
LoanMagazinesQty	Numeric(4.0)	count(LoanMagazineComment)
Book	Book	
BookId	Id	
BookTitle	Character(20)	
LoanBookComment	Character(40)	
BookAvailableQty	Numeric(4.0)	BookCopies - BookOnLoanQty
Magazine	Magazine	
MagazineId	Id	
MagazineTitle	Character(20)	
LoanMagazineComment	Character(40)	
MagazineAvailableQty	Numeric(4.0)	MagazineCopies - MagazineOnLoanQty

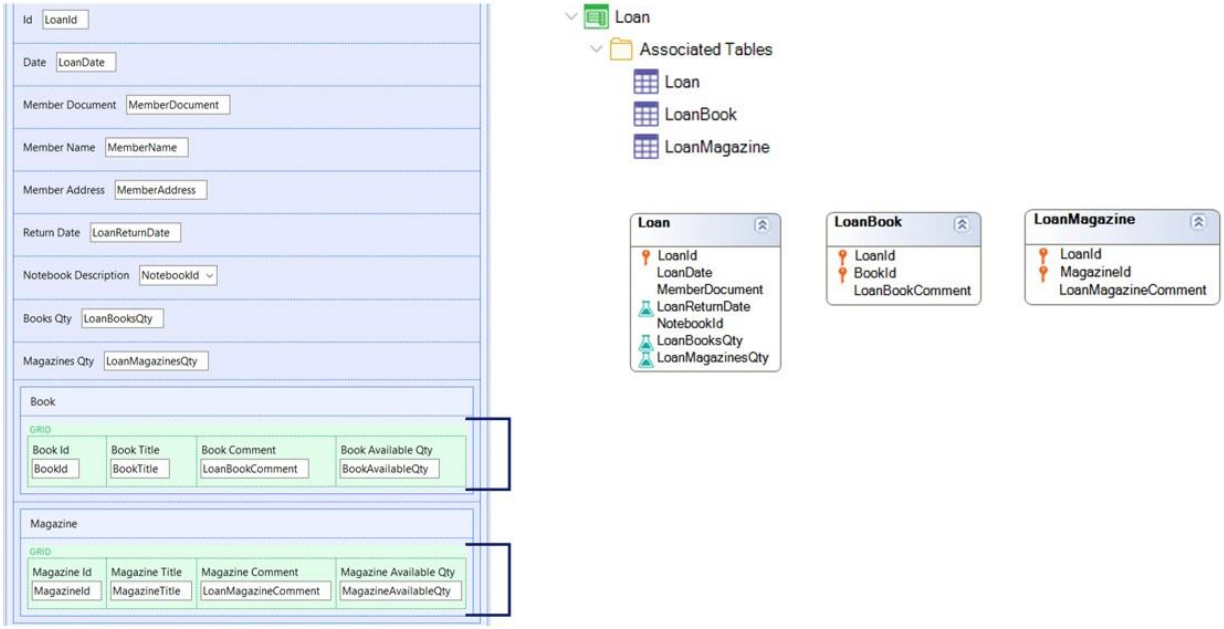
```
Error("Only 3 books can be checked out")
     if LoanBooksQty > 3;

Error("Only 4 magazines can be checked out")
     if LoanMagazinesQty > 4;

Default(LoanDate, today());

noaccept(LoanDate);

Error("There are no available copies of this book")
     if BookAvailableQty < 0;

Error("There are no available copies of this magazine")
     if MagazineAvailableQty < 0;
```

Ahora tenemos disponible toda la información que necesitamos para validar un préstamo, así que agreguemos los atributos necesarios para validar que exista disponibilidad de los ejemplares que se desean retirar en el préstamo:

- BookAvailableQty, en el nivel correspondiente a los libros, y
- MagazineAvailableQty, en el nivel correspondiente a las revistas

Y controlamos con las siguientes reglas Error:

Transacción Loan: Form y estructura de tablas



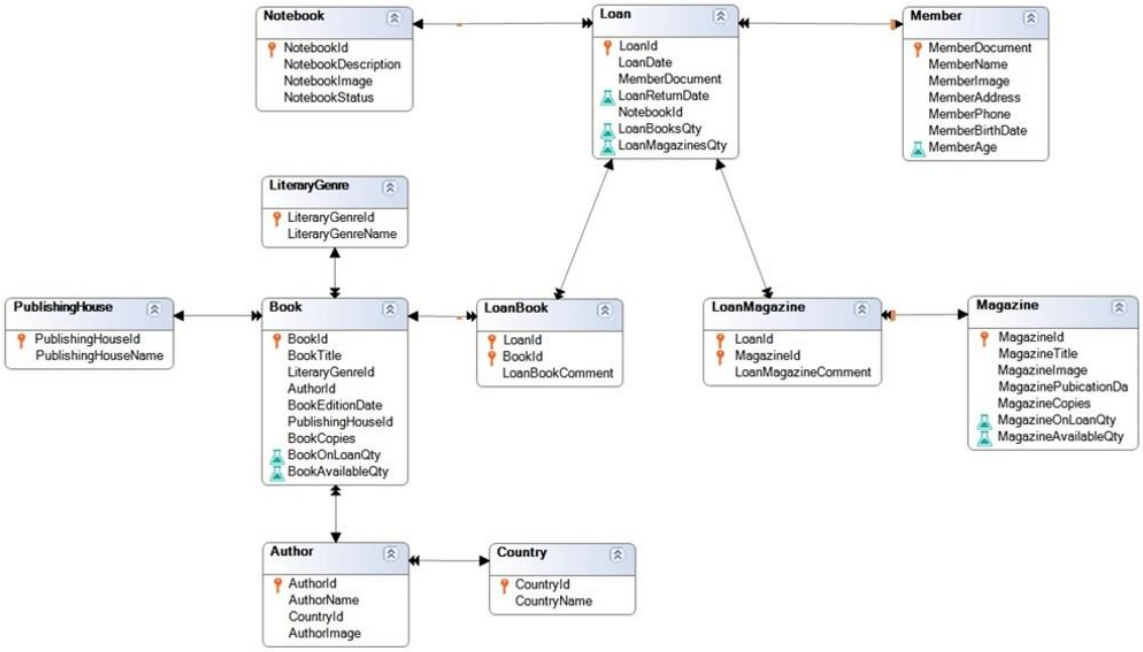
Hemos resuelto hasta aquí los requerimientos para registrar Préstamos. Pero observemos el form generado para el diseño que hemos hecho.

¿Qué tablas asociadas crea GeneXus para esta transacción Loan?

Crea 3 tablas:
- **LOAN**, asociada al primer nivel, con LoanId como PK.
- **LOANBOOK**, asociada al nivel de los libros, con LoanId y BookId como PK compuesta
- **LOANMAGAZINE**, asociada al nivel de las revistas, con LoanId y MagazineId como PK compuesta.

Al definir dos niveles paralelos, vemos en el form grillas paralelas. Esto tal vez puede ser engorroso para el usuario final, que puede solicitar simplificar dicho diseño, y manejar la información en diferentes pantallas más simples.

Diagrama de Tablas



Para el diseño actual, GeneXus ha creado las siguientes tablas.

Resumen

Analizar, evaluar la realidad a modelar e implementar la opción que consideremos correcta trabajando siempre junto al usuario final

Recordemos que siempre debemos analizar, evaluar la realidad a modelar e implementar la opción que consideremos correcta, y para eso es fundamental trabajar siempre junto al usuario final, que será quien nos guíe en la selección del diseño.

No olvidemos que la aplicación debe ser una herramienta de apoyo al usuario final para un mejor manejo y desarrollo de su negocio.

GeneXus™

training.genexus.com
wiki.genexus.com