

Normalización de Tablas

Un Caso de Estudio

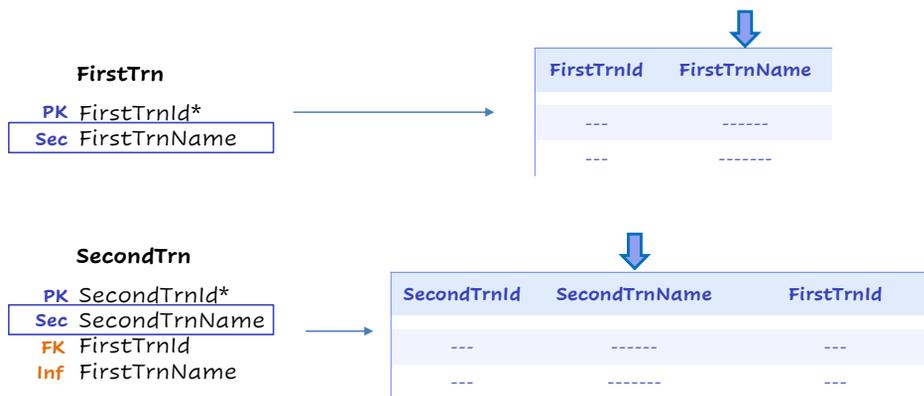
GeneXus™

A lo largo del curso hemos estado viendo cómo GeneXus determina la estructura de la base de datos a partir del diseño de transacciones.

Vimos también que realiza automáticamente los controles de integridad referencial, y que para eso utiliza los índices primarios y foráneos que crea en cada tabla.

Vamos a ver ahora un caso de estudio para analizar cómo GeneXus normaliza la base de datos y determina la estructura de las tablas a partir de un determinado diseño de transacciones.

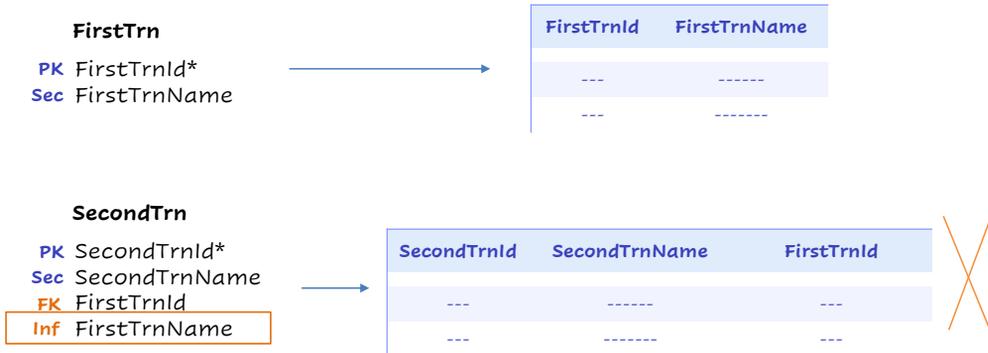
Base de Datos normalizada: Concepto



Comencemos entonces por el concepto de Base de datos normalizada.
¿Qué significa?

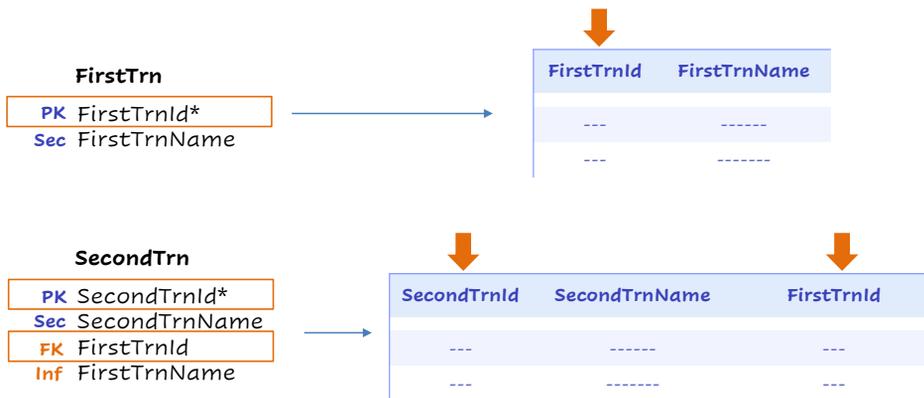
Significa que no hay información duplicada, no hay redundancias. Los atributos secundarios están presentes en una única tabla...

Base de Datos normalizada: Concepto



... los atributos inferidos no se almacenan,

Base de Datos normalizada: Concepto



.. y los únicos atributos que pueden estar presentes en más de una tabla son las claves primarias, por ser también claves foráneas en otras tablas.

Caso de Estudio: Diseño de transacciones

Country

CountryId* (PK)
CountryName (Sec)

TouristGuide

TouristGuideId* (PK)
TouristGuideName (Sec)

Trip

TripId*
TripDate
TripPrice
TouristGuideId
TouristGuideName
CountryId
CountryName

Customer

CustomerId*
CustomerName
CustomerTripsQty

Trip

(
 TripId*
 TripDate
 TripPrice
 CountryId
 CountryName
 CustomerTripMiles
)

Vayamos entonces al caso de estudio. Consideremos el siguiente diseño de transacciones, donde podemos ver que estamos modelando Países, Viajes o excursiones a un determinado país con un Guía turístico a cargo, y los Clientes que pueden realizar varios de estos viajes o excursiones.

Lo primero que debemos recordar y tener presente es que no podemos ver cada transacción en forma independiente sino que debemos analizar el modelo de datos completo. Cada transacción genera un impacto en dicho modelo de datos.

Observemos entonces que las transacciones Country y TouristGuide son transacciones simples, de un solo nivel, cada una con su identificador y nombre, y sin la presencia de claves foráneas. Cada una tiene entonces su clave primaria y un atributo secundario.

Caso de Estudio: Tablas normalizadas

COUNTRY

CountryId*
CountryName



Country X	
Structure Indexes	
Attribute	Order
Country Indexes	
ICountry	Primary Key
CountryId	Ascending

TOURISTGUIDE

TouristGuideId*
TouristGuideName



TouristGuide X	
Structure Indexes	
Attribute	Order
TouristGuide Indexes	
ITouristGuide	Primary Key
TouristGuideId	Ascending

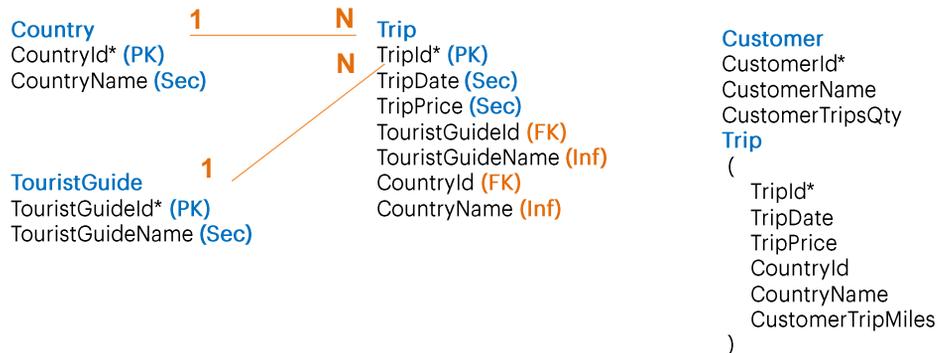
Por lo tanto, ya sabemos que a partir de ellas GeneXus creará las tablas COUNTRY y TOURISTGUIDE con la estructura que estamos viendo:

La tabla COUNTRY tiene CountryId como clave primaria y CountryName como atributo secundario. Sobre esta tabla entonces, GeneXus creará el índice primario por CountryId.

Recordemos que GeneXus crea automáticamente los índices primarios para controlar la unicidad de la clave primaria y para realizar en forma eficiente los controles de integridad referencial.

Por su parte, la tabla TOURISTGUIDE tiene TouristGuideId como clave primaria y TouristGuideName como atributo secundario. En esta tabla GeneXus creará el correspondiente índice primario por el atributo TouristGuideId

Caso de Estudio: Diseño de transacciones



Observemos ahora la transacción Trip. Tiene TripId como atributo primario, y TripDate y TripPrice como atributos secundarios. Luego TouristGuideId es clave foránea en esta transacción. Y TouristGuideName se infiere a partir de dicho valor. Tenemos una relación 1-N entre las entidades TouristGuide y Trip.

Algo similar ocurre con el atributo CountryId que aquí es clave foránea, y CountryName que se infiere a partir de dicho valor. Entre Country y Trip existe también una relación 1-N.

Caso de Estudio: Tablas normalizadas

COUNTRY

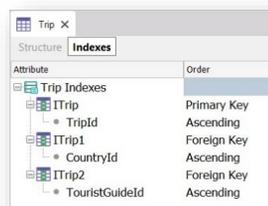
CountryId*
CountryName

TOURISTGUIDE

TouristGuideId*
TouristGuideName

TRIP

TripId*
TripDate
TripPrice
TouristGuideId
CountryId



Ya sabemos que los atributos inferidos no se almacenan porque GeneXus obtiene su valor a partir de las correspondientes claves foráneas, entonces ¿qué estructura va a tener la tabla asociada a la transacción Trip?

Su clave primaria TripId, y los atributos TripDate, TripPrice, TouristGuideId y CountryId.

Si pensamos ahora en los índices, ¿qué índices va a crear GeneXus sobre la tabla TRIP? El índice primario por TripId, y los índices foráneos por CountryId y TouristGuideId.

Esto significa que, por ejemplo, al insertar un viaje, el índice primario por TripId controlará la unicidad de su valor, o sea, que ya no exista otro viaje con el mismo valor de clave primaria.

El índice primario por TouristGuideId en la tabla TOURISTGUIDE, permitirá realizar los controles de integridad referencial verificando que el valor de la clave foránea TouristGuideId que se indique en Trip exista previamente como clave primaria en la tabla TOURISTGUIDE.

El mismo control realizará el índice primario por CountryId definido en la tabla COUNTRY. Verificará que el valor aquí indicado exista previamente como clave primaria en COUNTRY.

Por otra parte, si intentamos por ejemplo eliminar un guía de turismo través de la transacción ToruistGuide, el índice foráneo por TouristGuideId definido en Trip, controlará que no exista ningún viaje registrado que tenga dicho valor de TouristGuideId. En caso de tenerlo, GeneXus avisará que existen registros en Trip que tienen en TouristGuideId el valor que se intenta eliminar, y no permitirá que se elimine.

Caso de Estudio: Diseño de transacciones

Country CountryId* (PK) CountryName (Sec)	Trip TripId* (PK) TripDate (Sec) TripPrice (Sec) TouristGuideId (FK) TouristGuideName (Inf) CountryId (FK) CountryName (Inf)	Customer CustomerId* (PK) CustomerName (Sec) CustomerTripsQty (GLOBAL FORMULA) Trip (TripId* (PK) - (FK) TripDate (Inf) TripPrice (Inf) CountryId (Inf) CountryName (Inf) CustomerTripMiles (Sec))
TouristGuide TouristGuideId* (PK) TouristGuideName (Sec)		

Observemos ahora la transacción Customer. Es una transacción de dos niveles, con TripId como clave primaria del segundo nivel. Esto nos indica una relación N-N entre Customer y Trip. Y ya sabemos que a partir de este diseño de transacciones GeneXus™ creará dos tablas: CUSTOMER Y CUSTOMERTRIP.

Pero vamos a concentrarnos en el primer nivel de la transacción.

Tenemos CustomerId como clave primaria, CustomerName como atributo secundario y CustomerTripsQty que tiene asociado un cálculo, o sea, es una fórmula global, y ya sabemos que es un atributo calculado no se almacena en la base de datos.

Concentrémonos ahora en el segundo nivel. TripId es su clave primaria, pero también es clave foránea. y vemos a TripDate, TripPrice, CountryId y CountryName.

Si volvemos a mirar la estructura de la transacción Trip, vemos que estos atributos están allí presentes y por lo tanto todos ellos se obtienen, se infieren, a partir del valor de su clave primaria TripId.

Esto significa que ahora en el segundo nivel de Customer, estos atributos serán inferidos por el valor de TripId. Entonces CountryId que era una clave foránea directa, almacenada, en Trip, es ahora una clave foránea inferida en este segundo nivel de Customer. Y ¿qué pasa con el atributo CustomerTripMiles? Es un atributo secundario de este nivel.

Caso de Estudio: Tablas normalizadas

COUNTRY

CountryId*
CountryName

TOURISTGUIDE

TouristGuideId*
TouristGuideName

TRIP

TripId*
TripDate
TripPrice
TouristGuideId
CountryId

CUSTOMER

CustomerId*
CustomerName

CUSTOMERTRIP

CustomerId*
TripId*
CustomerTripMiles

Attribute	Order
Customer Indexes	
ICustomer	Primary Key
CustomerId	Ascending

Attribute	Order
CustomerTrip Indexes	
ICustomerTrip	Primary Key
CustomerId	Ascending
TripId	Ascending
ICustomerTrip1	Foreign Key
TripId	Ascending
ICustomerTrip2	Foreign Key
CustomerId	Ascending

Entonces, ¿cómo queda la estructura de las tablas que surgen a partir de la transacción Customer?

La tabla CUSTOMER asociada al primer nivel, con CustomerId como clave primaria y CustomerName. Recordemos que el atributo CustomerTripsQty, al ser calculado, no se guarda en la tabla asociada. Y luego la tabla CUSTOMERTRIP asociada al segundo nivel de la transacción Customer, con el par CustomerId, TripId como clave primaria y el atributo secundario CustomerTripMiles.

En cuanto a los índices, en CUSTOMER se creará el índice primario por CustomerId, y en CUSTOMERTRIP, se creará su correspondiente índice primario por el par de atributos CustomerId, TripId, y luego los correspondientes índices foráneos por TripId y CustomerId.

Observemos que no se crea sobre esta tabla un índice por CountryId, porque como hemos explicado anteriormente, en esta tabla no es una clave foránea directa sino inferida a través del valor de TripId.

Finalmente...

Diseño de transacciones	Country CountryId* CountryName	Trip TripId* TripDate TripPrice TouristGuideId TouristGuideName CountryId CountryName	Customer CustomerId* CustomerName CustomerTripsQty – Count(TripId) Trip (TripId* TripDate TripPrice CountryId CountryName CustomerTripMiles)
	TouristGuide TouristGuideId* TouristGuideName		

COUNTRY

CountryId*
CountryName

TRIP

TripId*
TripDate
TripPrice
TouristGuideId
CountryId

CUSTOMER

CustomerId*
CustomerName

Tablas normalizadas

TOURISTGUIDE

TouristGuideId*
TouristGuideName

CUSTOMERTRIP

CustomerId*
TripId*
CustomerTripMiles

De esta forma entonces, hemos analizado que a partir de este diseño de transacciones, GeneXus crea esta estructura normalizada en la base de datos.

GeneXus™

training.genexus.com
wiki.genexus.com
training.genexus.com/certifications