

# Normalização de tabelas

Um estudo de caso

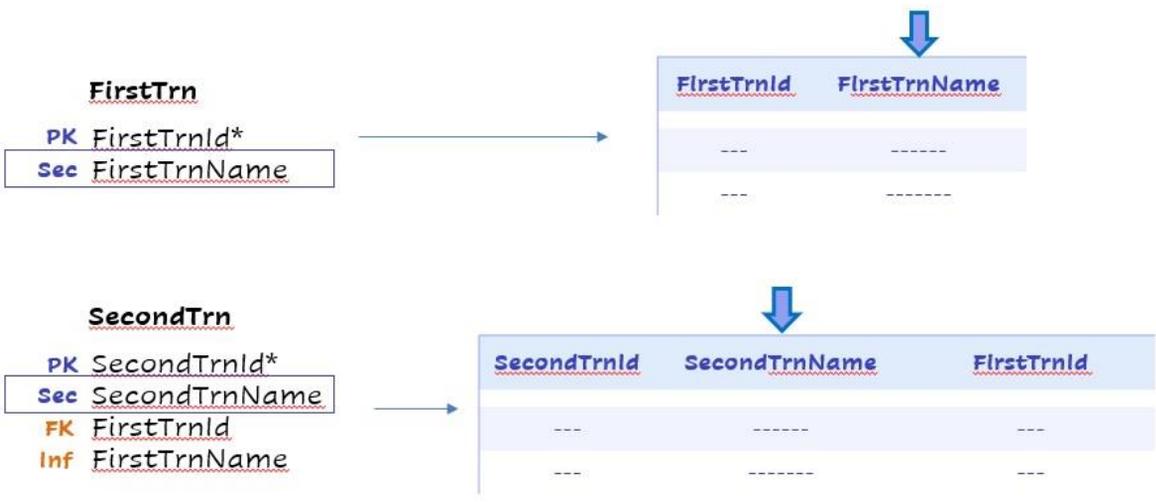
**GeneXus**

Ao longo do curso, temos visto como GeneXus determina a estrutura da base de dados a partir do desenho de transações.

Vimos também que realiza automaticamente os controles de integridade referencial, e que para isso utiliza os índices primários e estrangeiros que cria em cada tabela.

Vamos ver agora um estudo de caso para analisar como GeneXus normaliza a base de dados e determina a estrutura das tabelas a partir de um determinado desenho de transações.

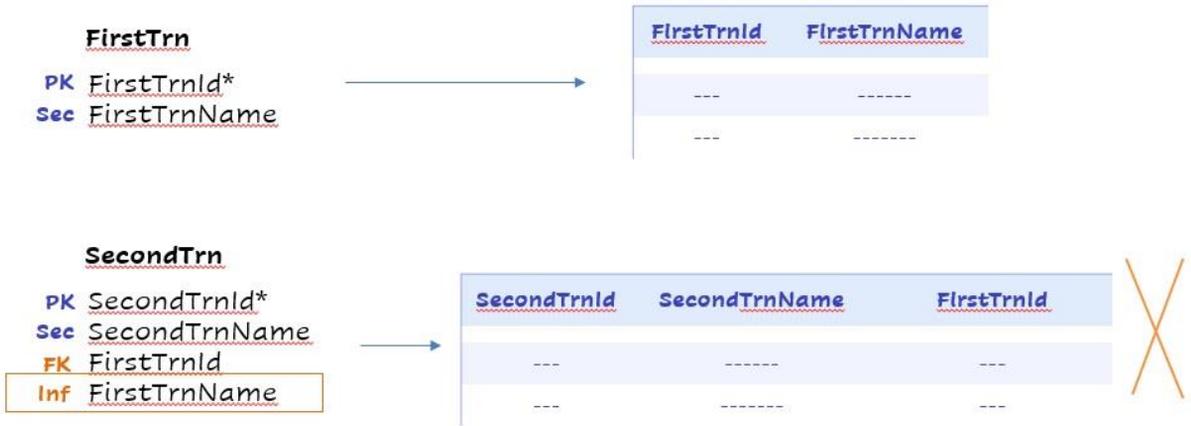
Base de dados normalizada: Conceito



Comecemos então pelo conceito de Base de dados normalizada. O que significa?

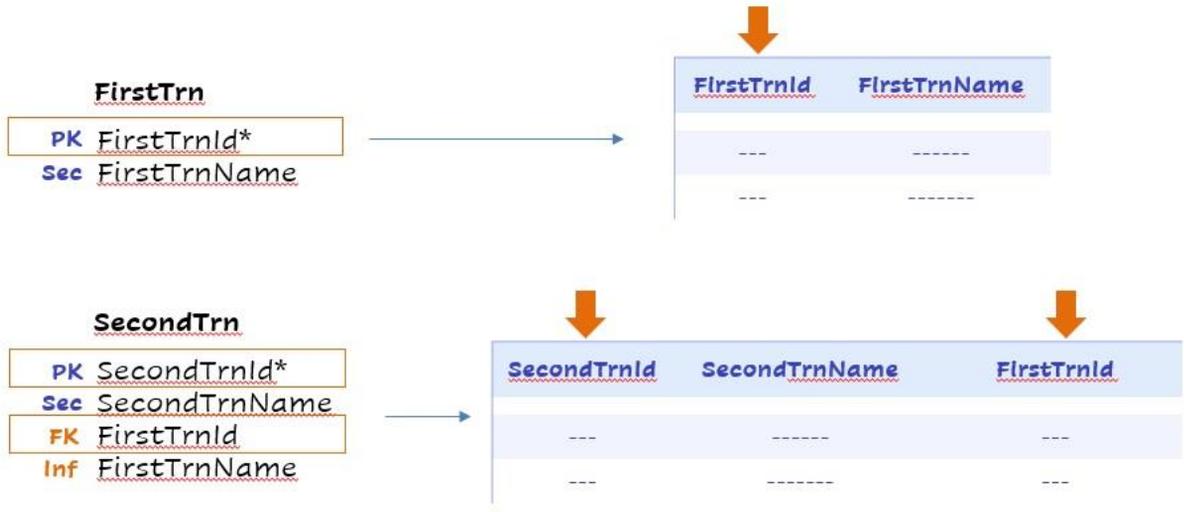
Significa que não há informação duplicada, não há redundâncias. Os atributos secundários estão presentes em uma única tabela...

Base de dados normalizada: Conceito



...os atributos inferidos não se armazenam,

Base de dados normalizada: Conceito



..e os únicos atributos que podem estar presentes em mais de uma tabela, são as chaves primárias, por serem também chaves estrangeiras em outras tabelas.

Estudo de caso: [Desenho de transações](#)**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  
)

Vamos então ao estudo de caso. Consideremos o seguinte desenho de transações, onde podemos ver que estamos modelando Países, Viagens ou excursões a um determinado país com um Guia turístico responsável, e os Clientes que podem realizar várias destas viagens ou excursões.

A primeira coisa que devemos lembrar e ter presente, é que não podemos ver cada transação de forma independente, mas devemos analisar o modelo de dados completo. Cada transação gera um impacto neste modelo de dados.

servemos então, que as transações Country e TouristGuide, são transações simples, de um único nível, cada uma com seu identificador e nome, e sem a presença de chaves estrangeiras. Cada uma tem, então, sua chave primária e um atributo secundário.

## Estudo de caso : Tabelas normalizadas

**COUNTRY**  
CountryId\*  
CountryName

Attribute	Order
Country Indexes	
ICountry	Primary Key
CountryId	Ascending

**TOURISTGUIDE**  
TouristGuideId\*  
TouristGuideName

Attribute	Order
TouristGuide Indexes	
ITouristGuide	Primary Key
TouristGuideId	Ascending

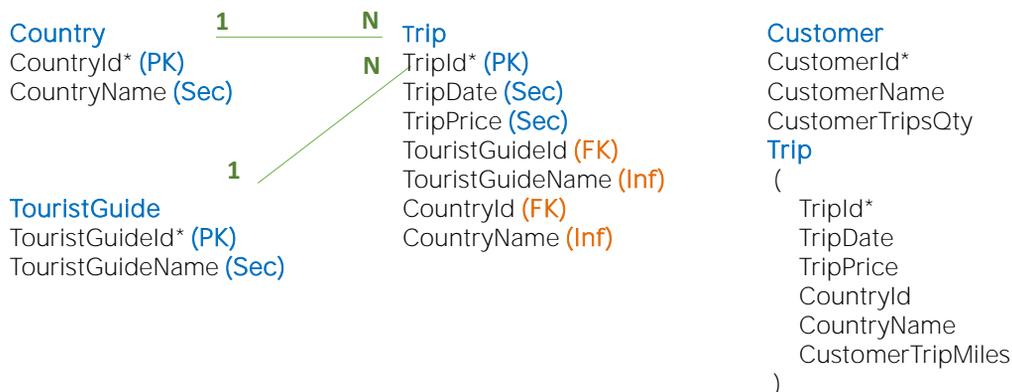
Portanto, já sabemos que a partir delas, GeneXus criará as tabelas COUNTRY e TOURISTGUIDE com a estrutura que estamos vendo.

A tabela COUNTRY tem CountryId como chave primária e CountryName como atributo secundário. Nesta tabela então, GeneXus criará o índice primário por CountryId.

Lembremos que GeneXus cria automaticamente os índices primários para controlar a unicidade da chave primária e para realizar de forma eficiente os controles de integridade referencial.

Por sua vez, a tabela TOURISTGUIDE tem TouristGuideId como chave primária e TouristGuideName como atributo secundário. Nesta tabela, GeneXus criará o correspondente índice primário pelo atributo TouristGuideId.

## Estudo de caso: Desenho de transações



Observemos agora a transação Trip. Tem TripId como atributo primário, e TripDate e TripPrice como atributos secundários. Logo, TouristGuideId é chave estrangeira nesta transação. E TouristGuideName infere-se a partir deste valor. Temos uma relação 1-N entre as entidades TouristGuide e Trip.

Algo similar ocorre com o atributo CountryId que aqui é chave estrangeira, e CountryName que infere-se a partir deste valor. Entre Country e Trip existe também uma relação 1-N.

## Estudo de caso : Tabelas normalizadas

**COUNTRY**

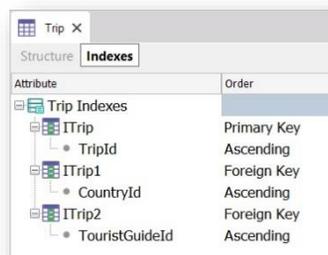
CountryId\*  
CountryName

**TOURISTGUIDE**

TouristGuideId\*  
TouristGuideName

**TRIP**

TripId\*  
TripDate  
TripPrice  
TouristGuideId  
CountryId



Já sabemos que os atributos inferidos não se armazenam porque GeneXus obtém seu valor a partir das correspondentes chaves estrangeiras, então, que estrutura terá a tabela associada à transação Trip?

Sua chave primária TripId, e os atributos TripDate, TripPrice, TouristGuideId e CountryId.

Se pensamos agora nos índices, quais índices GeneXus criará sobre a tabela TRIP? O índice primário por TripId, e os índices estrangeiros por CountryId e TouristGuideId

Isto significa que, por exemplo, ao inserir uma viagem, o índice primário por TripId, controlará a unicidade de seu valor, ou seja, que já não exista outra viagem com o mesmo valor de chave primária.

O índice primário por TouristGuideId na tabela TOURISTGUIDE, permitirá realizar os controles de integridade referencial verificando que o valor da chave estrangeira TouristGuideId que se indique em Trip exista previamente como chave primária na tabela TOURISTGUIDE.

O mesmo controle realizará o índice primário por CountryId definido na tabela COUNTRY. Verificará que o valor aqui indicado, exista previamente como chave primária em COUNTRY.

Por outro lado, se tentamos, por exemplo, excluir um guia turístico através da transação TouristGuide, o índice estrangeiro por TouristGuideId definido em Trip, controlará para que não exista nenhuma viagem registrada, que tenha este valor de TouristGuideId. Caso tenha, GeneXus avisará que existem registros em Trip que têm em TouristGuideId o valor que se tenta excluir, e não permitirá que seja excluído.

## Estudo de caso: Desenho de transações

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

Observemos agora a transação Customer. É uma transação de dois níveis, com TripId como chave primária do segundo nível. Isso nos indica uma relação N-N entre Customer e Trip. E já sabemos que, a partir deste desenho de transações, GeneXus criará duas tabelas: CUSTOMER e CUSTOMERTRIP.

Temos CustomerId como chave primária, CustomerName como atributo secundário e CustomerTripsQty, que tem associado um cálculo, ou seja, é uma fórmula global, e já sabemos que é um atributo calculado, que não se armazena na base de dados.

Concentremo-nos agora no segundo nível. TripId é a sua chave primária, mas também é chave estrangeira, e vemos TripDate, TripPrice, CountryId e CountryName.

Se voltarmos a olhar a estrutura da transação Trip, vemos que estes atributos estão ali presentes e, portanto todos eles são obtidos, são inferidos, a partir do valor de sua chave primária TripId.

Isto significa que agora no segundo nível de Customer, estes atributos serão inferidos pelo valor de TripId. Então CountryId que era uma chave estrangeira direta, armazenada, em Trip, agora é uma chave estrangeira inferida neste segundo nível de Customer. E o que acontece com o atributo CustomerTripMiles? É um atributo secundário deste nível.

Estudo de caso : Tabelas normalizadas

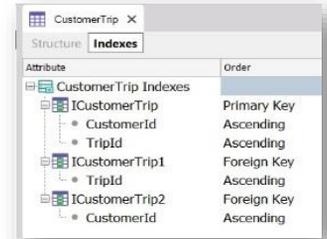
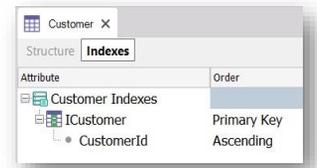
**COUNTRY**  
CountryId\*  
CountryName

**TRIP**  
TripId\*  
TripDate  
TripPrice  
TouristGuideId  
CountryId

**TOURISTGUIDE**  
TouristGuideId\*  
TouristGuideName

**CUSTOMER**  
CustomerId\*  
CustomerName

**CUSTOMERTRIP**  
CustomerId\*  
TripId\*  
CustomerTripMiles



Então, como fica a estrutura das tabelas que surgem a partir da transação Customer?

A tabela CUSTOMER associada ao primeiro nível, com CustomerId como chave primária e CustomerName. Lembremos que o atributo CustomerTripsQty, ao ser calculado, não se guarda na tabela associada. E então a tabela CUSTOMERTRIP associada ao segundo nível da transação Customer, com o par CustomerId, TripId como chave primária e o atributo secundário CustomerTripMiles.

quanto aos índices, em CUSTOMER será criado o índice primário por CustomerId, e em CUSTOMERTRIP, será criado seu correspondente índice primário pelo par de atributos CustomerId, TripId, e logo, os correspondentes índices estrangeiros por TripId e CustomerId.

Observemos que não é criado sobre esta tabela, um índice por CountryId, porque como explicamos anteriormente, nesta tabela não é uma chave estrangeira direta, mas inferida através do valor de TripId.

## Finalmente...

## Desenho de transações

**Country**

CountryId\*  
CountryName

**TouristGuide**

TouristGuideId\*  
TouristGuideName

**Trip**

TripId\*  
TripDate  
TripPrice  
TouristGuideId  
TouristGuideName  
CountryId  
CountryName

**Customer**

CustomerId\*  
CustomerName  
CustomerTripsQty – Count(TripId)

**Trip**

(  
TripId\*  
TripDate  
TripPrice  
CountryId  
CountryName  
CustomerTripMiles  
)

**COUNTRY**

CountryId\*  
CountryName

**TRIP**

TripId\*  
TripDate  
TripPrice  
TouristGuideId  
CountryId

**CUSTOMER**

CustomerId\*  
CustomerName

**TOURISTGUIDE**

TouristGuideId\*  
TouristGuideName

**CUSTOMERTRIP**

CustomerId\*  
TripId\*  
CustomerTripMiles

## Tabelas normalizadas

Desta forma então, analisamos que a partir deste desenho de transações GeneXus cria esta estrutura normalizada na base de dados

*GeneXus*<sup>TM</sup>

[training.genexus.com](http://training.genexus.com)  
[wiki.genexus.com](http://wiki.genexus.com)