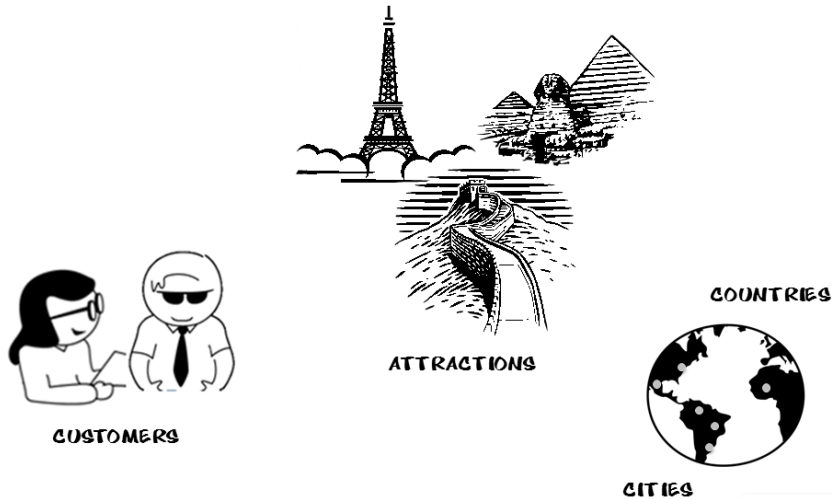


Relações entre os atores da realidade

Relations between actors of reality



GeneXus[®]

Em vários exemplos de nossa agência de viagens, vimos podem ser estabelecidas diferentes relações entre que os atores da realidade. Vimos, por exemplo, que uma atração pode pertencer a uma categoria que, por sua vez, pode ser a categoria de muitas atrações.

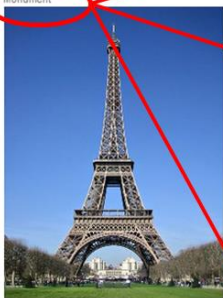
Attraction Information
Name Eiffel Tower

General

Id 3
Name Eiffel Tower
Country Id 2
Country Name [France](#)
Category Id 1
Category Name Monument

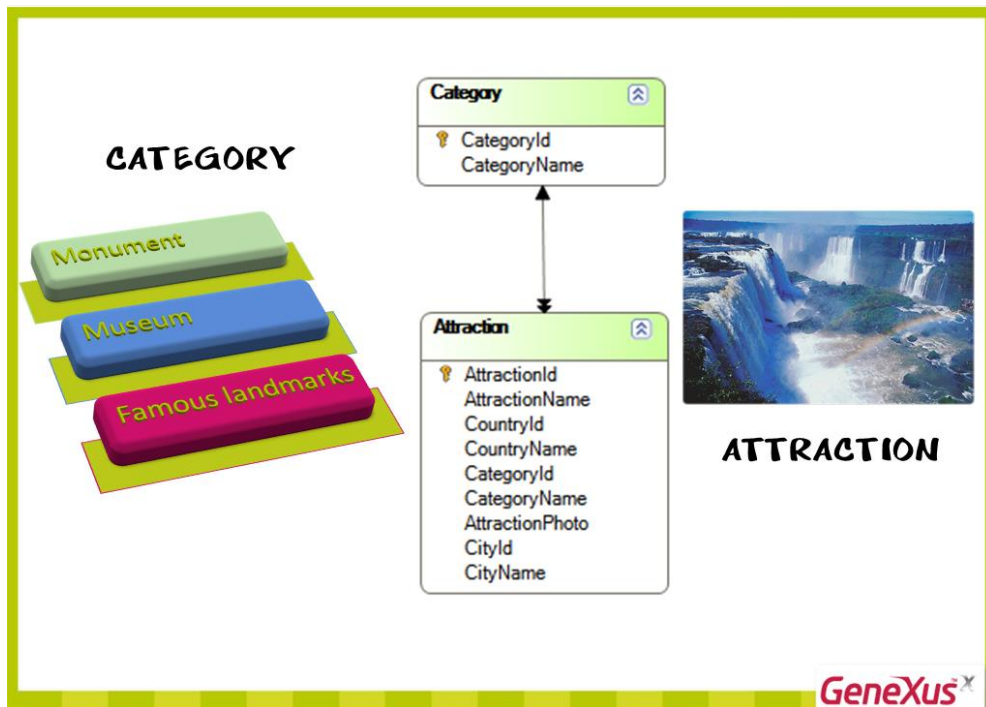
Photo

City Id 1
City Name Paris
[Update](#) [Delete](#)



GeneXus[®]

Vimos que a maneira que temos para representar essas relações é **na hora de criar as transações, incluir atributos de uma transação em outra.**

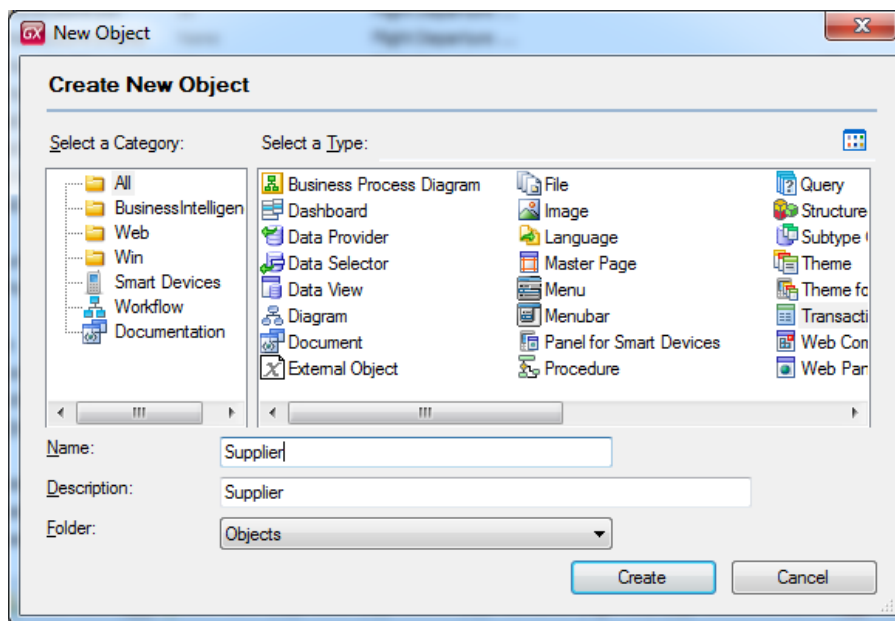


Sabemos que nossa agência trabalha com fornecedores que lhe oferecem periodicamente visitas a atrações turísticas em várias partes do mundo.

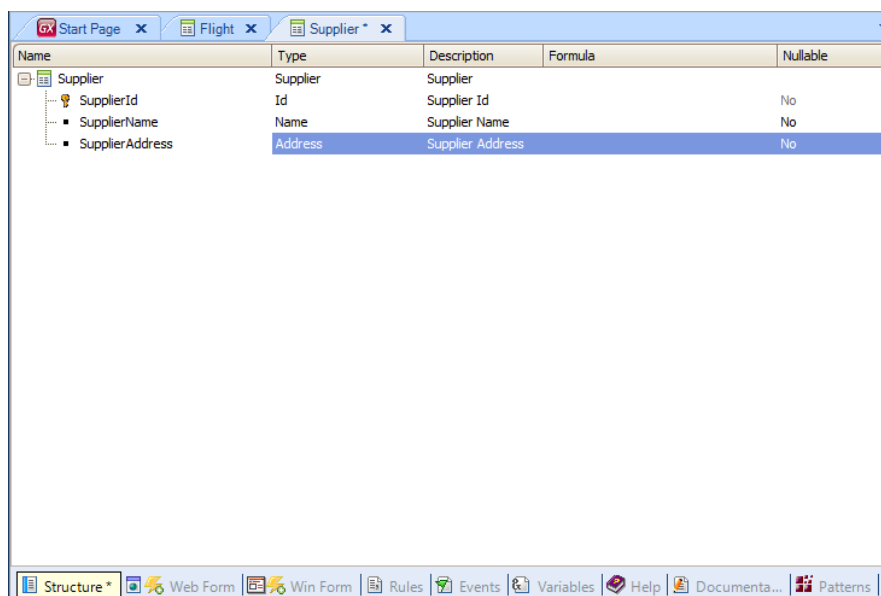


Cada fornecedor oferece muitas atrações turísticas, mas cada atração é administrada por um único fornecedor. Para representar essa realidade, vamos criar a transação Supplier, na qual registraremos os fornecedores...

Clicamos em File... New... Object... chamamos Supplier... E acrescentamos os atributos:



SupplierId como identificador, SupplierName para armazenar o nome do fornecedor e SupplierAddress para o seu endereço.



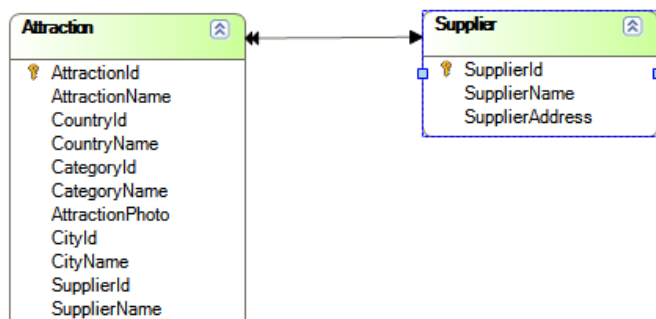
Notemos, que pelo objeto diagrama de transações, a relação entre os fornecedores e as atrações. Clicamos em New... Object do tipo Diagrama e arrastamos as transações Attraction e Supplier do Folder View. Como vemos, ainda não estabelecemos nenhuma relação entre esses dois atores.



Como uma atração turística é oferecida por apenas um único fornecedor, vamos incluir o identificador de fornecedor na estrutura da transação Attraction, abrindo tal transação e acrescentando o atributo SupplierId. Acrescentamos também o atributo Supplier Name para podermos assim mostrar o nome do fornecedor na tela das atrações.

Name	Type	Description	Formula	Nullable
Attraction	Attraction	Attraction		
AttractionId	Id	Attraction Id		No
AttractionName	Name	Attraction Name		No
CountryId	Id	Country Id		No
CountryName	Name	Country Name		
CategoryId	Id	Category Id		Yes
CategoryName	Name	Category Name		
AttractionPhoto	Image	Attraction Photo		No
CityId	Id	City Id		Yes
CityName	Name	City Name		
SupplierId	Id	Supplier Id		No
SupplierName	Name	Supplier Name		

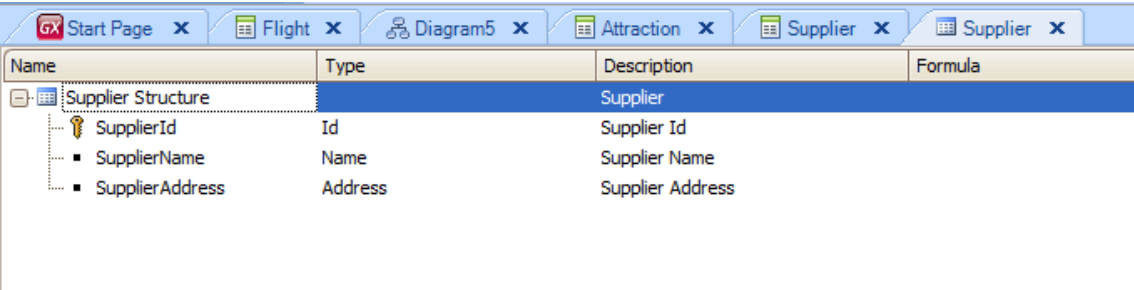
Criamos agora um novo diagrama, arrastamos ambas as transações novamente... e vemos que há uma seta simples apontando em direção a Supplier e uma seta dupla em direção a Attraction, indicando que uma atração tem um único fornecedor e que um fornecedor pode oferecer muitas atrações.



Resumindo: se acrescentarmos o identificador de uma transação a outra transação (que como já vimos, cumprirá o papel de chave estrangeira, fica estabelecida uma relação de 1 para

muitos – também chamada ‘1 para N’), em que o lado “muitos” da relação é onde está a chave estrangeira.

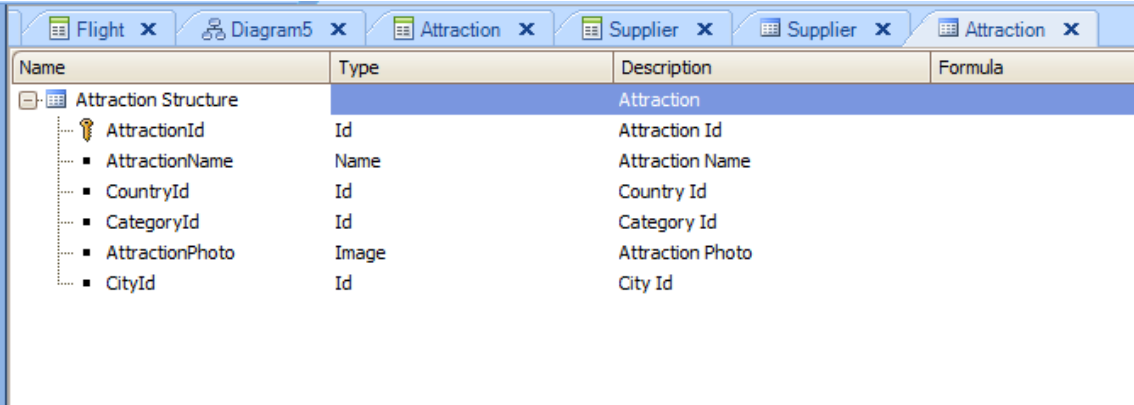
Se analisarmos agora quais seriam as tabelas que GeneXus gera a partir desse desenho de transações, vemos que a partir da transação Supplier, será criada uma tabela SUPPLIER com a mesma estrutura da transação:



The screenshot shows the GeneXus interface with the 'Supplier' transaction selected. The 'Supplier Structure' is displayed in a tree view on the left, and a table on the right shows the mapping of fields to their types and descriptions.

Name	Type	Description	Formula
Supplier Structure		Supplier	
SupplierId	Id	Supplier Id	
SupplierName	Name	Supplier Name	
SupplierAddress	Address	Supplier Address	

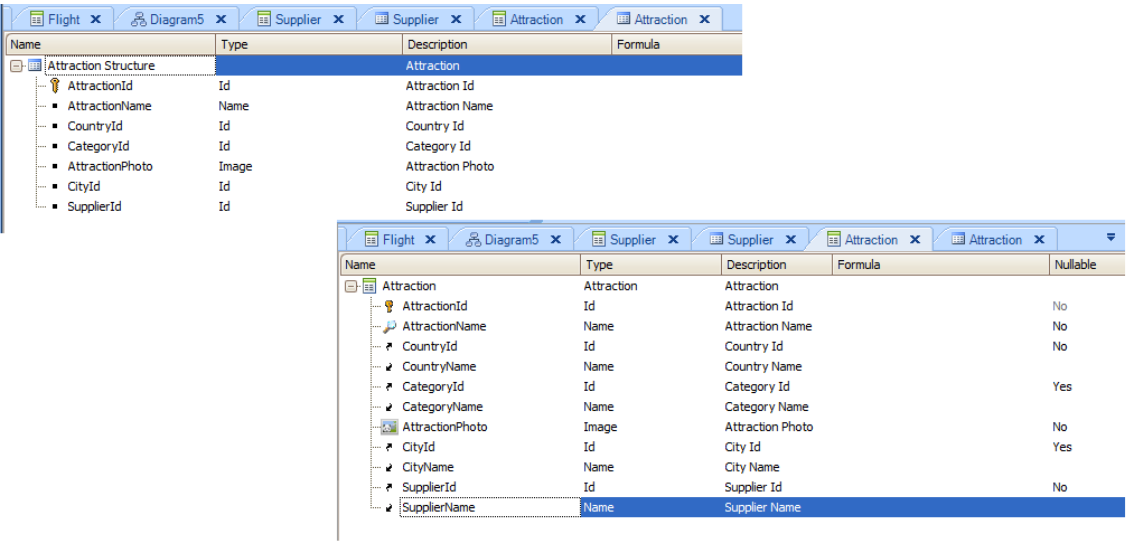
E a partir da estrutura da transação Attraction, GeneXus cria uma tabela ATTRACTION com a seguinte estrutura:



The screenshot shows the GeneXus interface with the 'Attraction' transaction selected. The 'Attraction Structure' is displayed in a tree view on the left, and a table on the right shows the mapping of fields to their types and descriptions.

Name	Type	Description	Formula
Attraction Structure		Attraction	
AttractionId	Id	Attraction Id	
AttractionName	Name	Attraction Name	
CountryId	Id	Country Id	
CategoryId	Id	Category Id	
AttractionPhoto	Image	Attraction Photo	
CityId	Id	City Id	

Se compararmos a estrutura da tabela ATTRACTION com a da transação Attraction,



The image shows two side-by-side screenshots. The left screenshot shows the 'Attraction Structure' in the GeneXus interface. The right screenshot shows the 'Attraction' table structure in the GeneXus interface, including a 'Nullable' column.

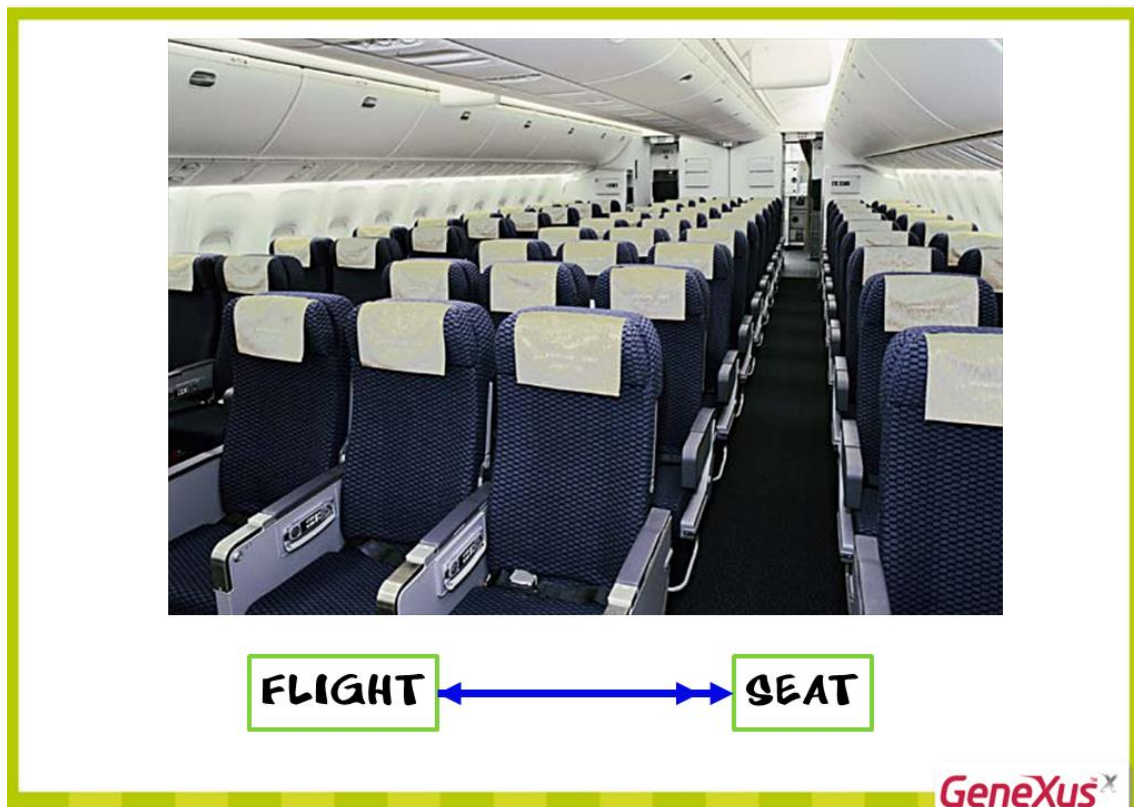
Name	Type	Description	Formula	Nullable
Attraction	Attraction	Attraction		
AttractionId	Id	Attraction Id		No
AttractionName	Name	Attraction Name		No
CountryId	Id	Country Id		No
CountryName	Name	Country Name		
CategoryId	Id	Category Id		Yes
CategoryName	Name	Category Name		
AttractionPhoto	Image	Attraction Photo		No
CityId	Id	City Id		Yes
CityName	Name	City Name		
SupplierId	Id	Supplier Id		No
SupplierName	Name	Supplier Name		

vemos que os atributos CountryName, CategoryName, CityName e SupplierName não estão incluídos na tabela, já que são atributos inferidos e, como vimos anteriormente, como estão na

tabela estendida da tabela ATTRACTION, seu valor pode ser recuperado da tabela onde estão fisicamente armazenados.

Esta é a forma mais comum de representar a relação 1 para muitos entre dois atores da realidade, ou seja, entre 2 entidades do nosso sistema.

No entanto, há outros casos de relações 1 para muitos mas nessas usaremos outro tipo de representação.

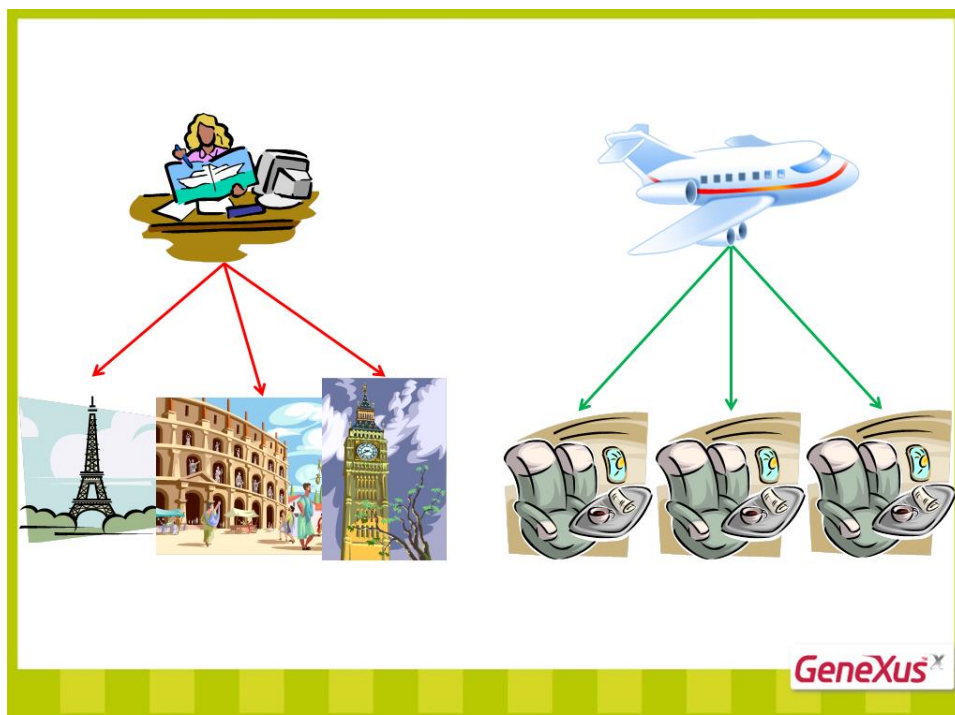


Vamos lembrar o caso dos voos, em que um voo tem muitos assentos e cada assento pertence a um voo, ou seja, uma relação de 1 para muitos. Vamos abrir a estrutura da transação Flight para ver como representamos essa relação...

Name	Type	Description	Formula	Nullable
Flight	Flight	Flight		
FlightId	Id	Flight Id		No
FlightDepartureCountryId	Id	Flight Departure ...		No
FlightDepartureCountryName	Name	Flight Departure ...		
FlightDepartureCityId	Id	Flight Departure ...		No
FlightDepartureCityName	Name	Flight Departure ...		
FlightArrivalCountryId	Id	Flight Arrival Cou...		No
FlightArrivalCountryName	Name	Flight Arrival Cou...		
FlightArrivalCityId	Id	Flight Arrival City...		No
FlightArrivalCityName	Name	Flight Arrival City...		
FlightPrice	Price	Flight Price		No
FlightDiscountPercentage	Percentage	Flight Discount P...		No
FlightFinalPrice	Price	Flight Final Price	$\text{FlightPrice} * (1 - \text{AirlineDiscountPercent} \dots)$	
AirlineId	Id	Airline Id		Yes
AirlineName	Name	Airline Name		
AirlineDiscountPercentage	Percentage	Airline Discount P...		
FlightCapacity	Numeric(4.0)	Flight Capacity	$\text{count}(\text{FlightSeatLocation}, \text{FlightSeat} \dots)$	
FlightSeatQty	Numeric(4.0)	Flight Seat Qty	$\text{count}(\text{FlightSeatChar})$	
Seat	Seat	Seat		
FlightSeatId	Id	Flight Seat Id		No
FlightSeatLocation	Location	Flight Seat Location		No
FlightSeatChar	SeatChar	Flight Seat Char		No

Vemos que, neste caso, o assento (Seat) está como um segundo nível na transação Flight.

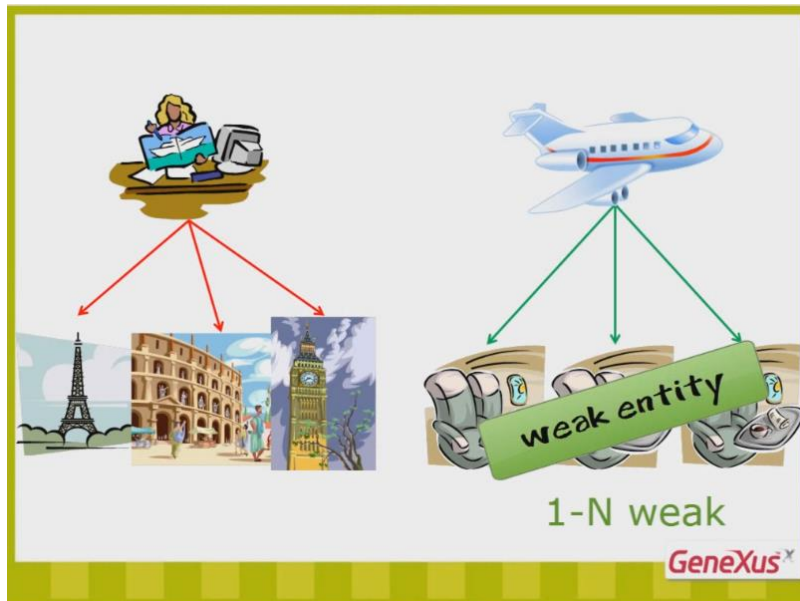
Dessa forma, qual a diferença desta relação de 1 para muitos da relação de 1 para muitos que vimos entre as Atrações e seus Fornecedores?



Por que não representamos da mesma maneira (com o mesmo desenho de transações) ambos os casos?

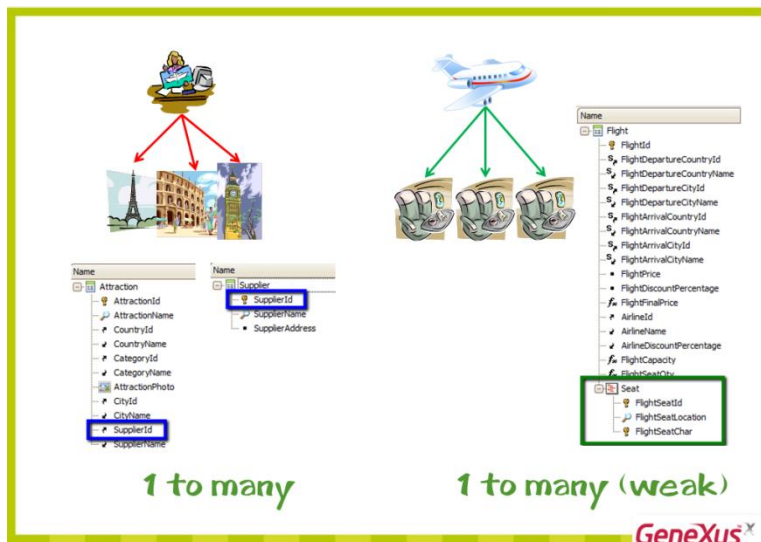
Não faz sentido existir assentos se não estiverem num voo, isto é, não faz sentido considerar um assento sem que esteja relacionado **sempre** ao voo que pertence... Entretanto, uma atração poderia não ter um fornecedor que a oferecesse, existindo por si mesma...

A outra diferença é que quando inserimos os dados de um voo, já inserimos também os dados de seus assentos (da mesma forma, quando inserimos uma fatura com suas linhas, inserimos de uma única vez toda a informação). Por outro lado, não existe razões para inserir os dados dos Fornecedores e das Atrações no mesmo momento.



Dizemos que uma entidade como os assentos, só faz sentido existir quando representada em função de outra entidade (neste caso, os voos), é uma **entidade fraca**.

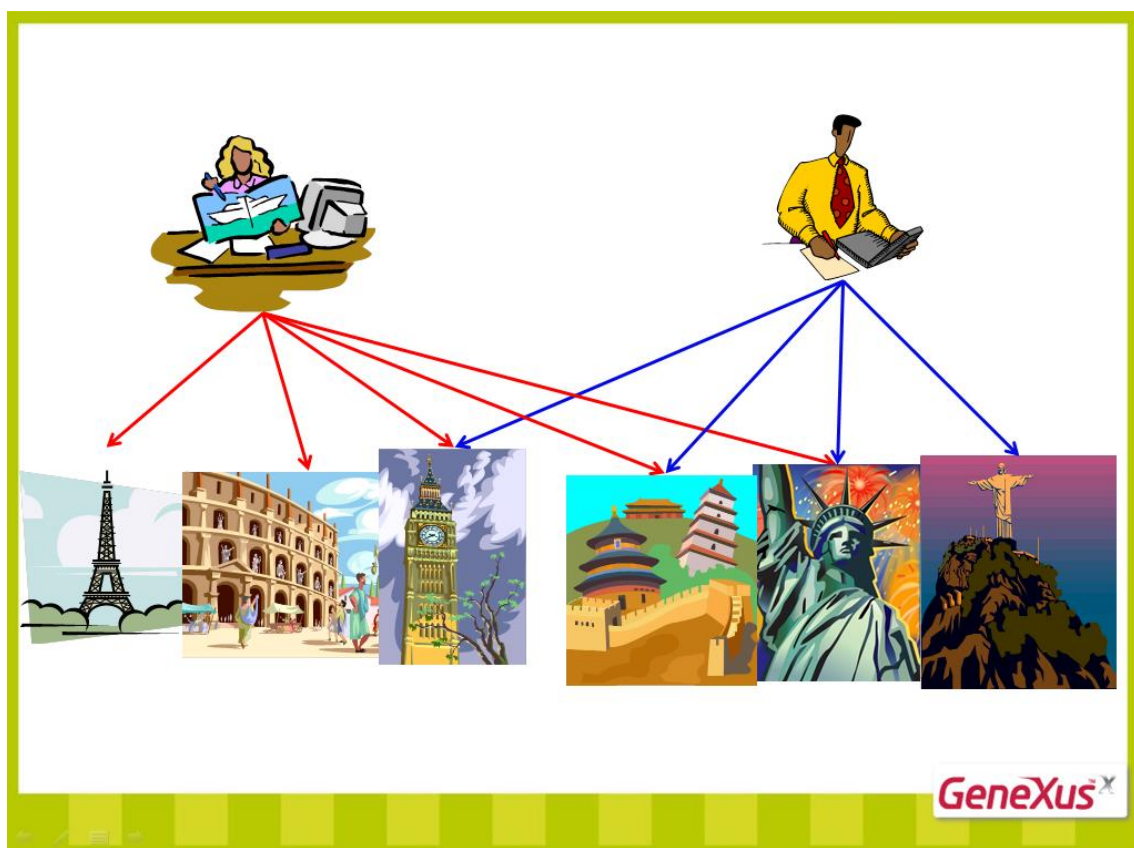
Representamos este tipo de relação **1 para N fraca** com uma única transação de 2 níveis, onde a entidade fraca está no segundo nível; diferente da relação de 1 para N de Fornecedores e Atrações, na qual criamos **2** transações e colocamos uma como chave estrangeira da chave primária de outra.



Até agora, vimos relações 1 para muitos, mas este não é o único caso da realidade que devemos representar.

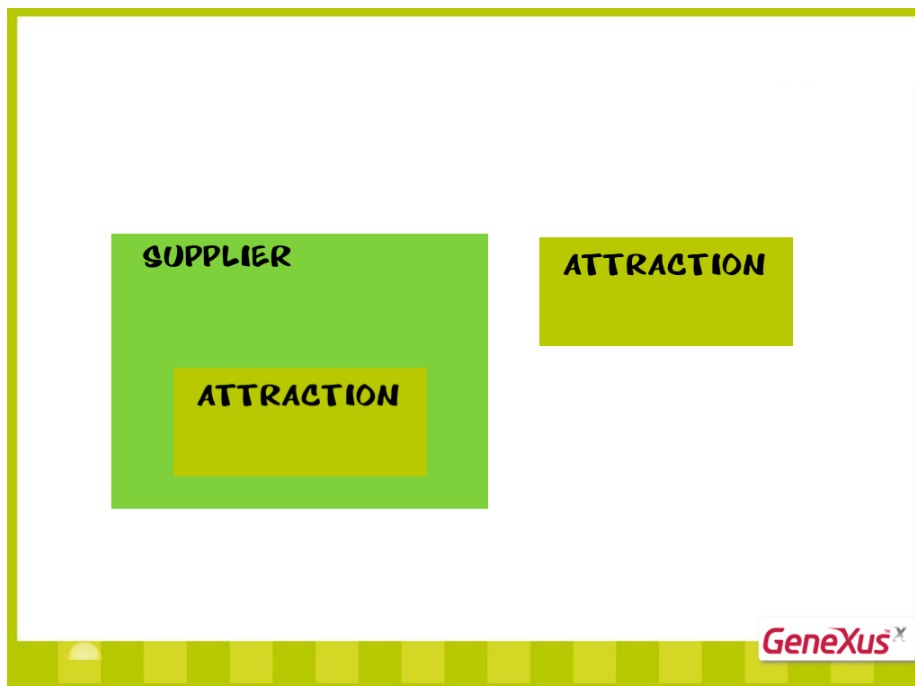
Imaginemos, por exemplo, que a agência de viagens nos informa **uma mudança em sua realidade**:

Cada fornecedor oferece muitas atrações turísticas (como vimos até agora), mas *cada atração pode ser administrada por VÁRIOS fornecedores (e não por 1, como até o momento)*.



Ou seja, a relação entre Fornecedores e Atrações já não seria “1 para muitos”, mas “**muitos para muitos**”.

Como representamos isso em GeneXus?



A solução é utilizar 2 transações, uma para cada entidade. Além disso, acrescentamos em uma delas como segundo nível da outra. Fazemos isso levando em consideração como os dados serão inseridos, se a partir do fornecedor e todas suas atrações turísticas ou a partir da atração e todos os seus fornecedores. Neste caso, o mais lógico é que quando formos inserir os dados de um fornecedor, já inserimos todas as transações que este fornece.

Façamos isso em GeneXus....

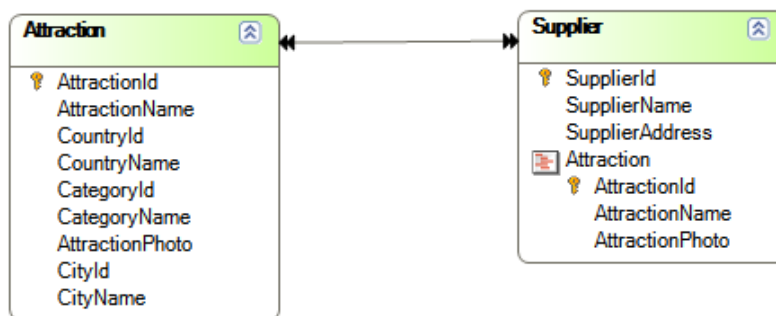
Abrimos a transação Attraction, tiramos os atributos SupplierId e SupplierName e salvamos.

Flight x Supplier x Diagram5 x Diagram6* x Attraction x			
Name	Type	Description	Formula
Attraction	Attraction	Attraction	
AttractionId	Id	Attraction Id	
AttractionName	Name	Attraction Name	
CountryId	Id	Country Id	
CountryName	Name	Country Name	
CategoryId	Id	Category Id	
CategoryName	Name	Category Name	
AttractionPhoto	Image	Attraction Photo	
CityId	Id	City Id	
CityName	Name	City Name	

Agora abrimos a transação Supplier, onde acrescentamos um segundo nível e acrescentamos os atributos: AttractionId, Attraction Name e AttractionPhoto. Como vemos o segundo nível ficou com o nome Attraction.

Name	Type	Description	Formula
Supplier	Supplier	Supplier	
SupplierId	Id	Supplier Id	
SupplierName	Name	Supplier Name	
SupplierAddress	Address	Supplier Address	
Attraction	Attraction	Attraction	
AttractionId	Id	Attraction Id	
AttractionName	Name	Attraction Name	
AttractionPhoto	Image	Attraction Photo	

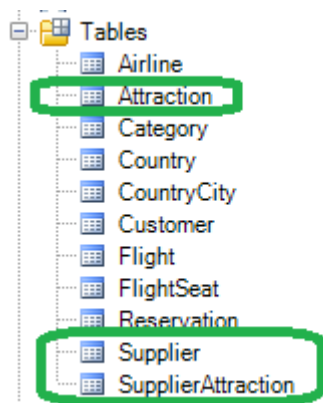
Veremos como ficou esta relação criando um diagrama novo... File... New... Diagram... e arrastamos as transações Attraction e Supplier ao digrama.



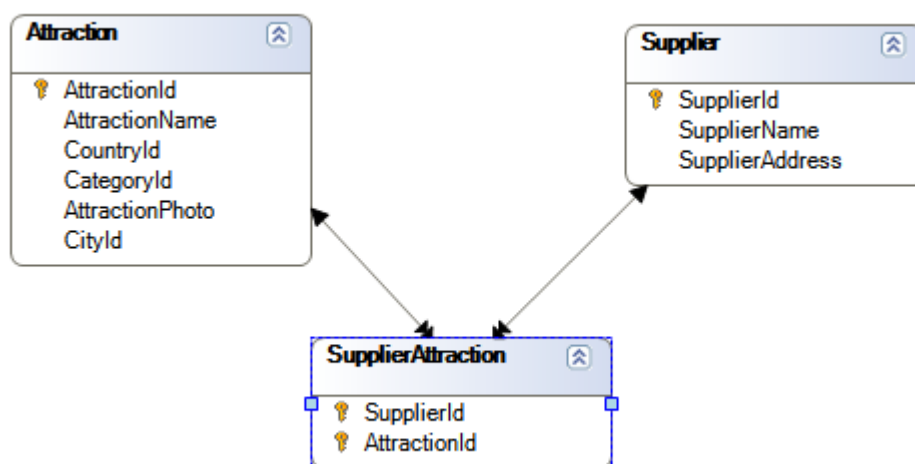
Agora já existe uma seta dupla em cada extremo da relação, o que indica que a relação é de “muitos” para “muitos”, isto é, uma atração é oferecida por muitos fornecedores e um fornecedor fornece muitas atrações.

Vejamos agora as tabelas que o GeneXus criará a partir do desenho anterior...

Vemos que encontramos uma tabela ATTRACTION, uma tabela SUPPLIER e uma tabela chamada SUPPLIERATTRACTION.



Devemos criar um objeto diagrama novo e arrastar as 3 tabelas ao digrama...

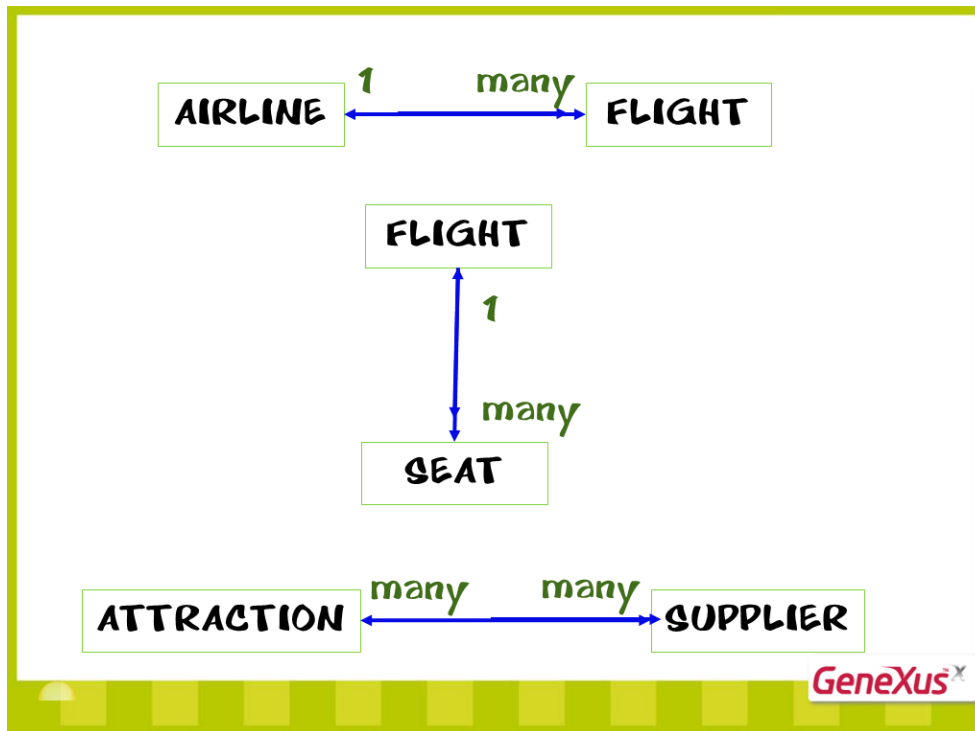


Devemos observar que, neste caso, GeneXus cria uma tabela para cada transação que intervém na relação muitos para muitos (ATTRACTION e SUPPLIER), mas cria também uma terceira tabela chamada SUPPLIERATTRACTION para estabelecer a relação.

Se observarmos a estrutura dessa terceira tabela, vamos perceber que são incluídos somente os atributos identificadores das outras duas tabelas.

Portanto, cada vez que GeneXus estabelecer uma relação de muitos para muitos, tal relação será representada no banco de dados por 3 tabelas, uma para cada entidade que intervém e uma terceira com os dois identificadores de ambas.

Vemos que a relação de muitos para muitos entre Attraction e Supplier se desmembra em 2 relações um para muitos, utilizando a tabela SUPPLIERATTRACTION para estabelecer a relação entre as anteriores.



Vimos assim que mediante as transações e seus atributos, podemos representar as diferentes relações entre os atores da nossa realidade.