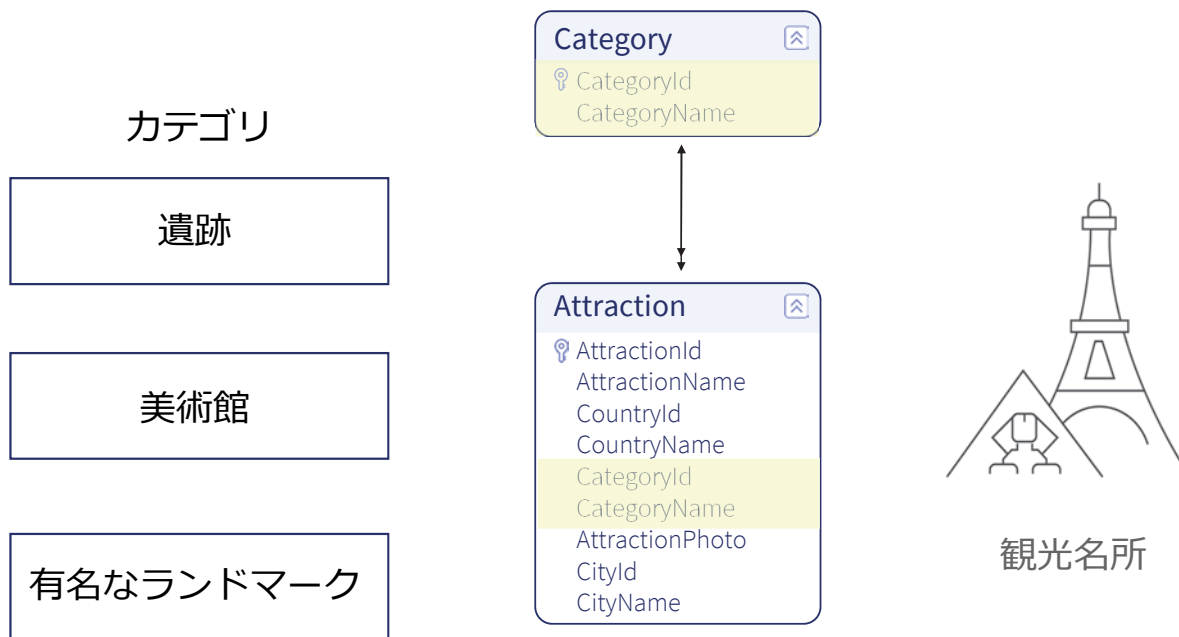


## 現実のエンティティ間の関係

GeneXus™

## 関連付けの実装



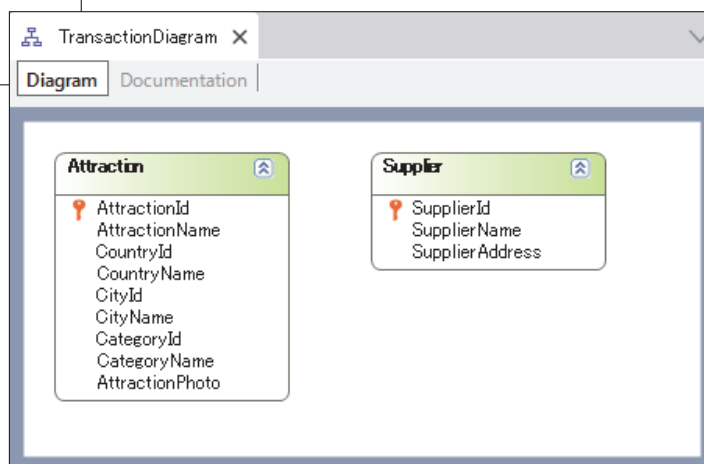
ここまでの章で、現実のエンティティが関連付けられるケースを確認しました。

こうした関連付けの実装は、**トランザクション設計時に、あるトランザクションの項目属性を別のトランザクションに含めることで表せる**、ということを説明しました。

## 新しい Supplier トランザクション

名前	タイプ	DESCRIPTION	式
Supplier	Supplier	サプライヤー	
SupplierId	Id	サプライヤー番号	
SupplierName	Name	サプライヤー名	
SupplierAddress	Address, GeneXus	サプライヤー住所	

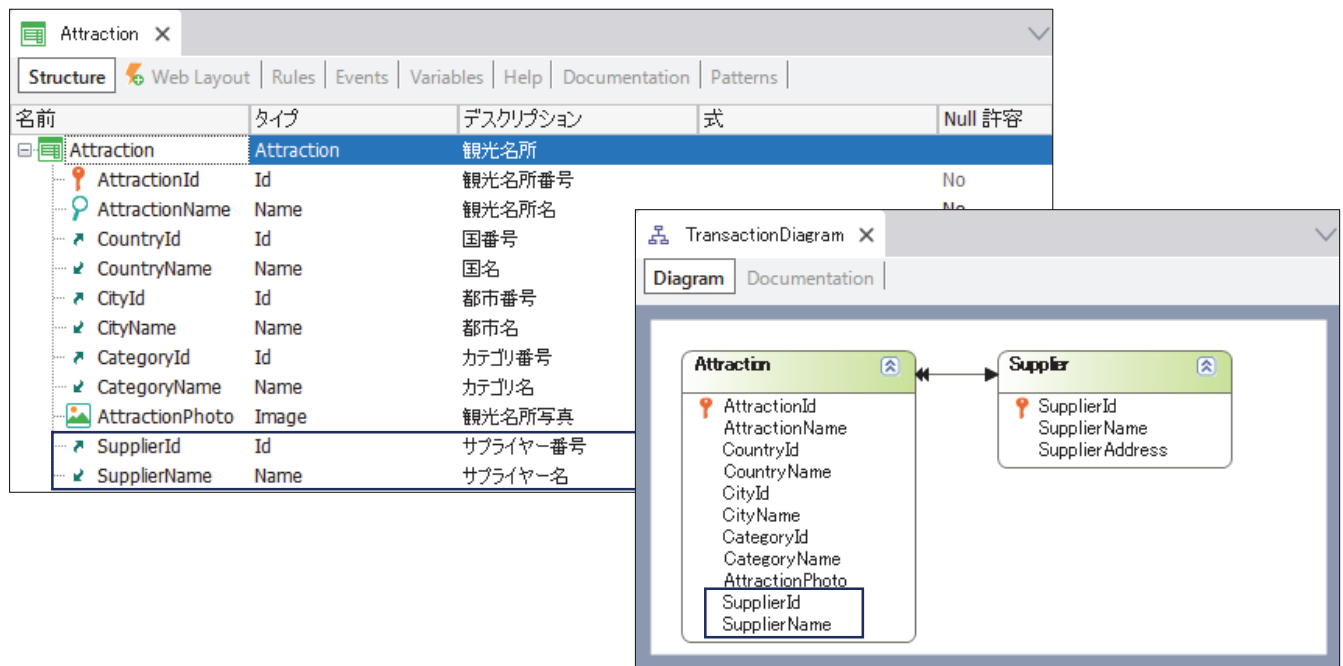
観光名所とサプライヤー間の  
関係は？



例えば、新たなトランザクションが定義された場合、このトランザクションを定義しただけでは、他のトランザクションとの関連性は一切ありません。関連付けを行いたいトランザクションがある場合もちろんこれだけでは何も変わりません。

これは、ここまでに紹介したダイアグラムオブジェクトを利用することでも確認することができます。

## 要件: 各観光名所は 1 つのサプライヤーによって提供される (1 対 N の関係)



もし、新しく定義したトランザクションと既存のトランザクションに関係が必要となる場合、どのような対応が必要でしょうか。

この場合の要望は、新しく定義したトランザクションで登録したデータは、既存のトランザクションで複数回参照される実装です。

つまり、新しいトランザクションを 1 としたとき、既存のトランザクションが N の関係となる必要があります。

GeneXus で、このような実装を行いたい場合、既存のトランザクション側に、新しいトランザクションの定義を参照できるように、主キー項目属性を追加し、外部キーとして利用できるようにする必要があります。

この状態で、ダイアグラムオブジェクトを開きなおすことで、関係を示す矢印が表示されることを確認できます。

## 作成されたテーブル

Supplier	
Structure	Web Layout   Rules   Events   Vari
名前	タイプ
Supplier	Supplier
SupplierId	Id
SupplierName	Name
SupplierAddress	Address, GeneXus

## トランザクション

Attraction	
Structure	Web Layout   Rules   Events   Vari
名前	タイプ
Attraction	Attraction
AttractionId	Id
AttractionName	Name
CountryId	Id
CountryName	Name
CityId	Id
CityName	Name
CategoryId	Id
CategoryName	Name
AttractionPhoto	Image
SupplierId	Id
SupplierName	Name

Supplier	
Structure	Indexes
名前	タイプ
Supplier 構造	
SupplierId	Id
SupplierName	Name
SupplierAddress	Address, GeneXus

## テーブル

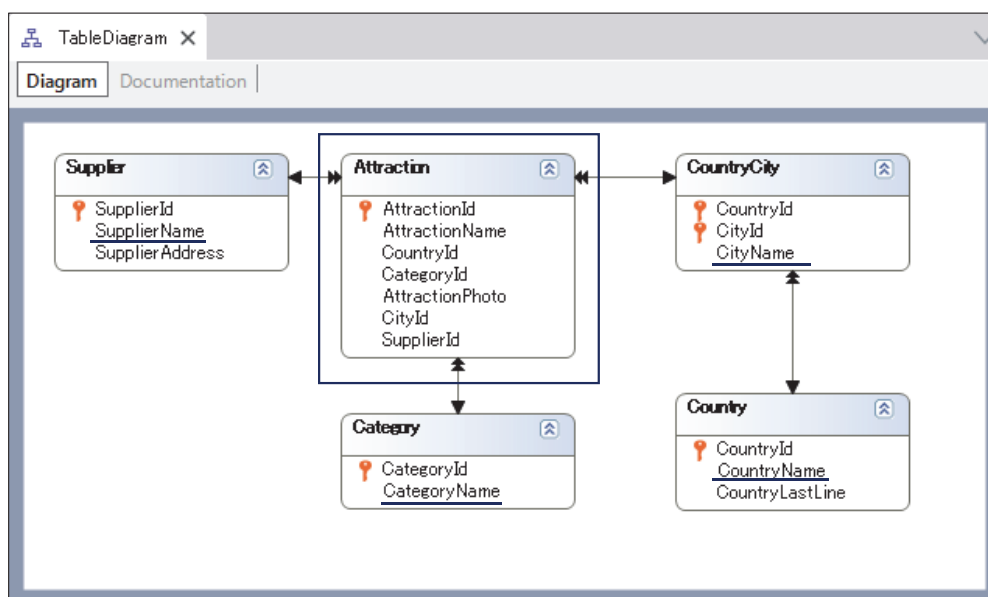
Attraction	
Structure	Indexes
名前	タイプ
Attraction 構造	
AttractionId	Id
AttractionName	Name
CountryId	Id
CategoryId	Id
AttractionPhoto	Image
CityId	Id
SupplierId	Id

このトランザクション設計に基づき、GeneXus によって生成されたテーブルを確認します。

新しく作成したトランザクションは、外部キーを含むものではないため、トランザクションの定義と同じ内容です。

既存のトランザクションには、今回のトランザクションを外部キーとして追加しました。この場合、外部キーの項目属性は、テーブル構造に作成されますが、推論できる項目属性は、含まれないことが確認できます。

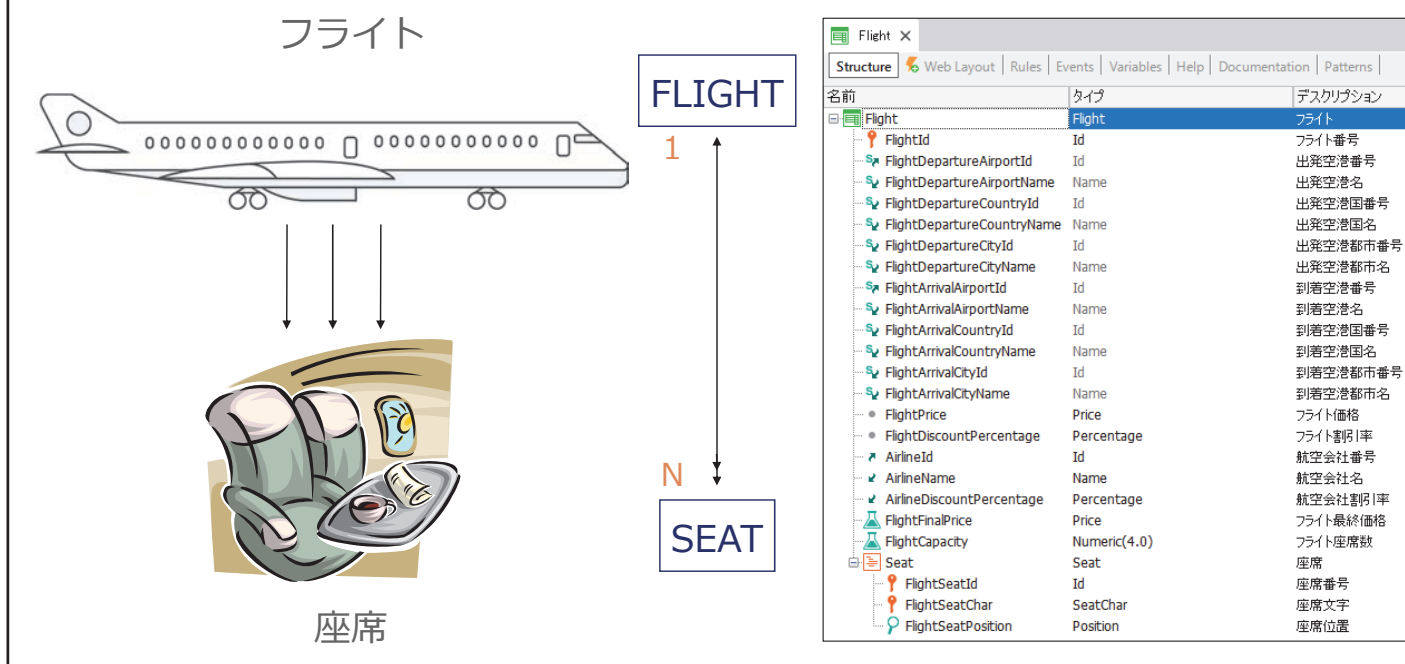
## エンティティ間の関係を確認



このように、1 対 N の関係を必要とする場合、1 とするトランザクションの主キー項目属性を N としたいトランザクションの定義に追加することで実装出来ます。また、N となる側からは、1 側のテーブルの値を取得することができます。

この結果、現実の 2 つのエンティティ、その内容に基づき、定義されたシステム内の 2 つのエンティティ間に 1 対 N の関係を表すことができます。これは、一般的な方法です。

## 1 対 N の関係をモデル化する別の方法

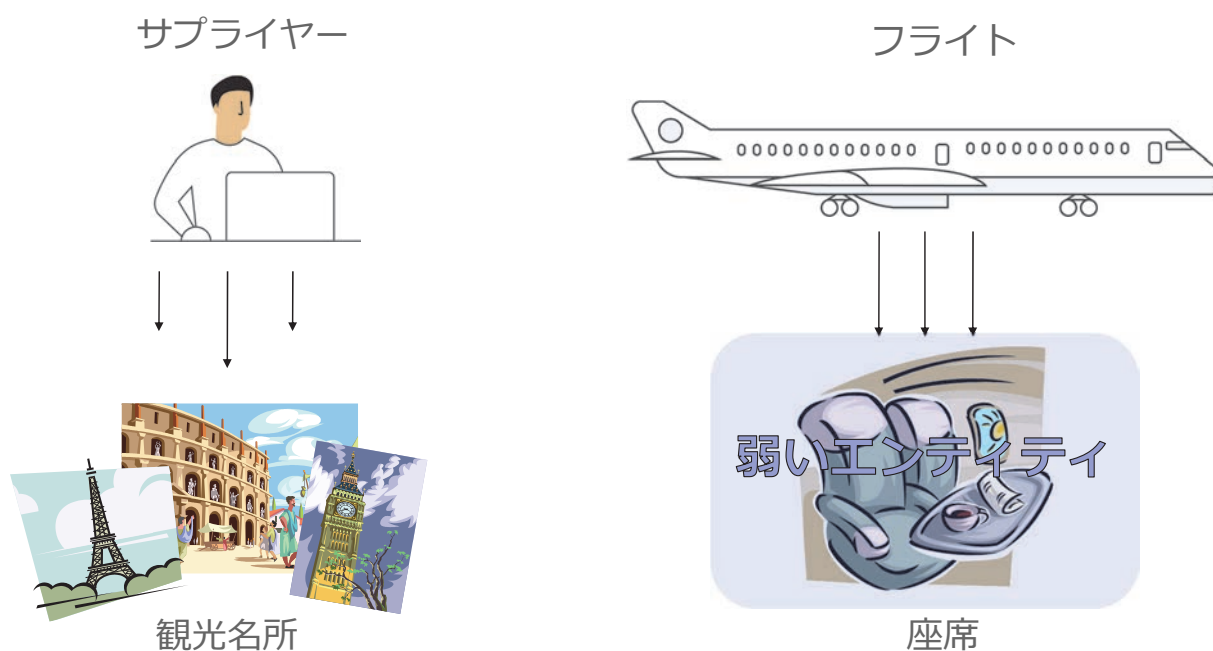


GeneXus で定義できる 1 対 N の関係は、他にもあります。

この方法は、1 つのトランザクション内で、レベルを定義する方法です。  
この方法により、第 1 レベルのレコード 1 件に対し、N 件の第 2 レベルのレコード  
を関連付けることが可能です。

では、前述の 1 対 N の関係と、このスライドの 1 対 N の関係に違いは  
あるのでしょうか。

## 1 対 N の実装比較



大きな違いとして、これらの実装は、同じトランザクション設計で表すことができませんでした。

この理由として、初めに扱った 2 つのトランザクションを利用するケースの場合、1 対 N の関係において、N となるデータは、1 のデータが無くてもデータとして存在することができます。

また、それぞれのトランザクションで必要なデータの入力を完結するため、すべてのデータを一度に入力する必要がありません。

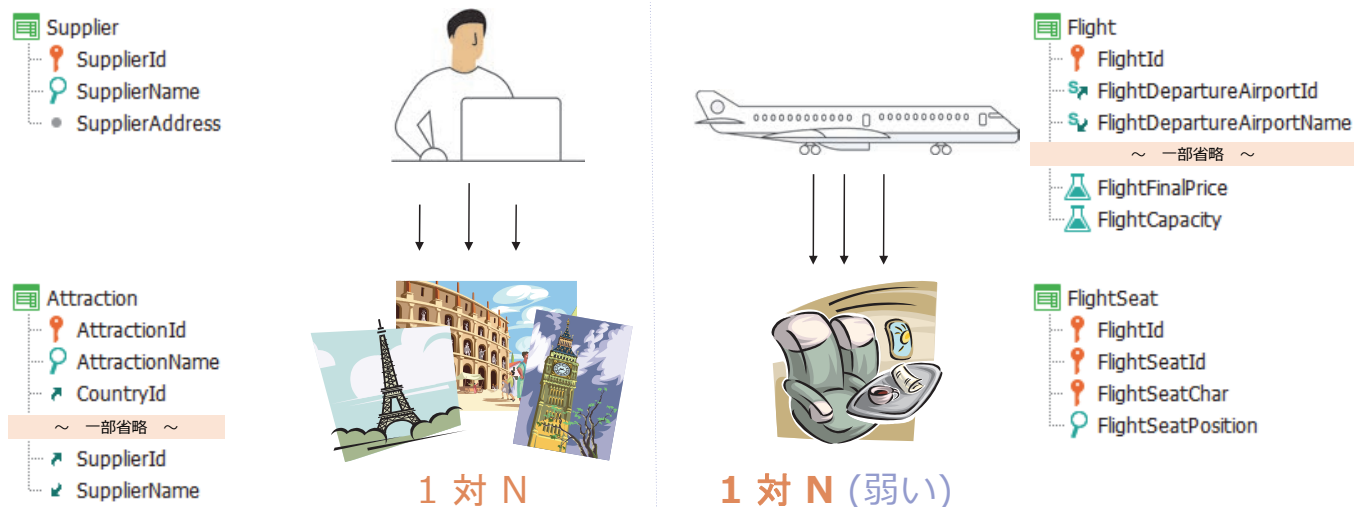
一方、1 つのトランザクションに複数のレベルがある場合、1 対 N の関係において、N となるデータは、1 のデータがなければ意味のないデータです。

そして、1 となるデータとともに N となるデータもすべて一度に入力する必要があります。

GeneXus では、後者の 1 つのトランザクションを利用し、1 対 N の関係を実装する場合、N 側となるエンティティを**弱いエンティティ**と呼びます。



## 別の弱いエンティティ定義方法



この**弱い 1 対 N の関係**は、前述の通り、通常、単一のトランザクション内に 2 つのレベルを使用して表し、弱いエンティティが 2 つ目のレベルになります。これは、**2 つ**のトランザクションを作成し、外部キーとして他方の主キーを追加して実現する 1 対 N の関係とは異なります。

弱い 1 対 N の関係を 2 つのトランザクションで表すこともできます (データモデル化の目的でもまったく同じです)。  
第 2 レベルの定義をトランザクションとして作成します。この時、第 1 レベルの主キーを複合キーの一部に含めます。  
つまり、主キーの一部に外部キーが含まれています。

通常のエンティティと弱いエンティティには相違点があります。  
通常のエンティティの場合、外部キー項目属性の [Null 許容] 列を Yes に設定していれば、外部キーを指定せずに登録できます。  
しかし、弱いエンティティの場合、外部キーは、主キーの一部となるため、Null を設定することができず、必ず外部キーの値が必要となります。

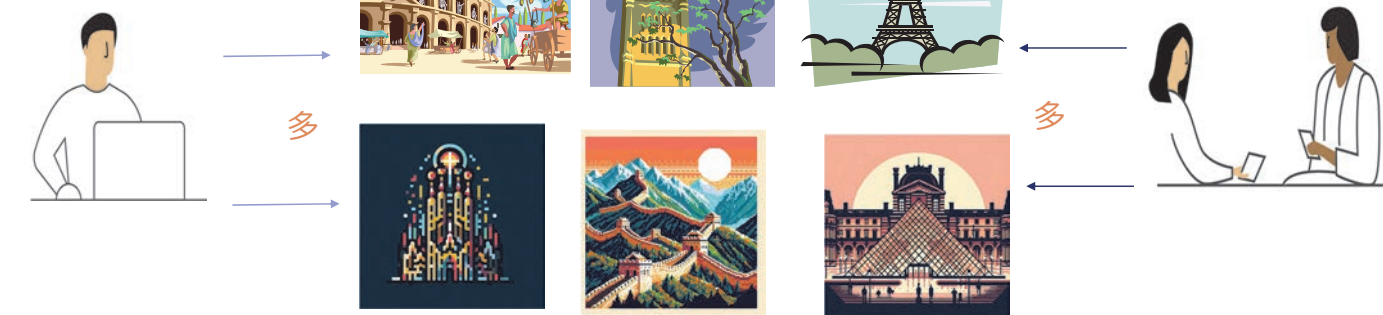
これまで 1 対 N の関係について説明しましたが、表さなければならない現実は、それ以外にもあります。

## 「多」対「多」の関係 (N 対 M)

### 観光名所

サプライヤー

サプライヤー

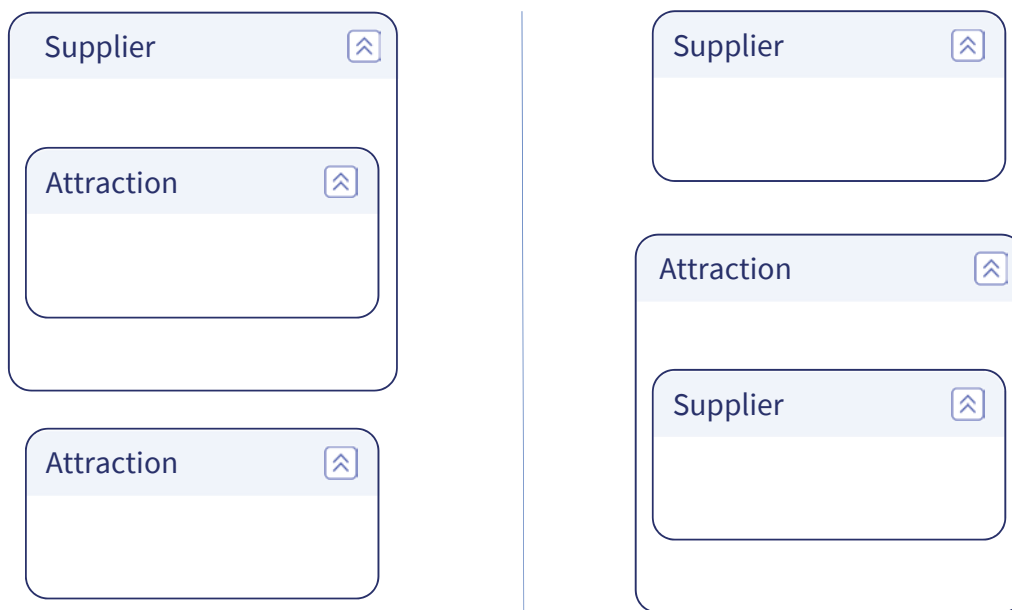


想定される現実の一つは、「**多対多**」 (N 対 M) のケースです。

つまり、トランザクション A と、トランザクション B があった場合、A に多数の B を紐づけ、逆に B に対しても、多数の A を紐づけるケースです。

これまでの 1 対 N の関係をお互いに持っているような状況です。  
このような実装は、GeneXus でどのように表せるでしょうか。

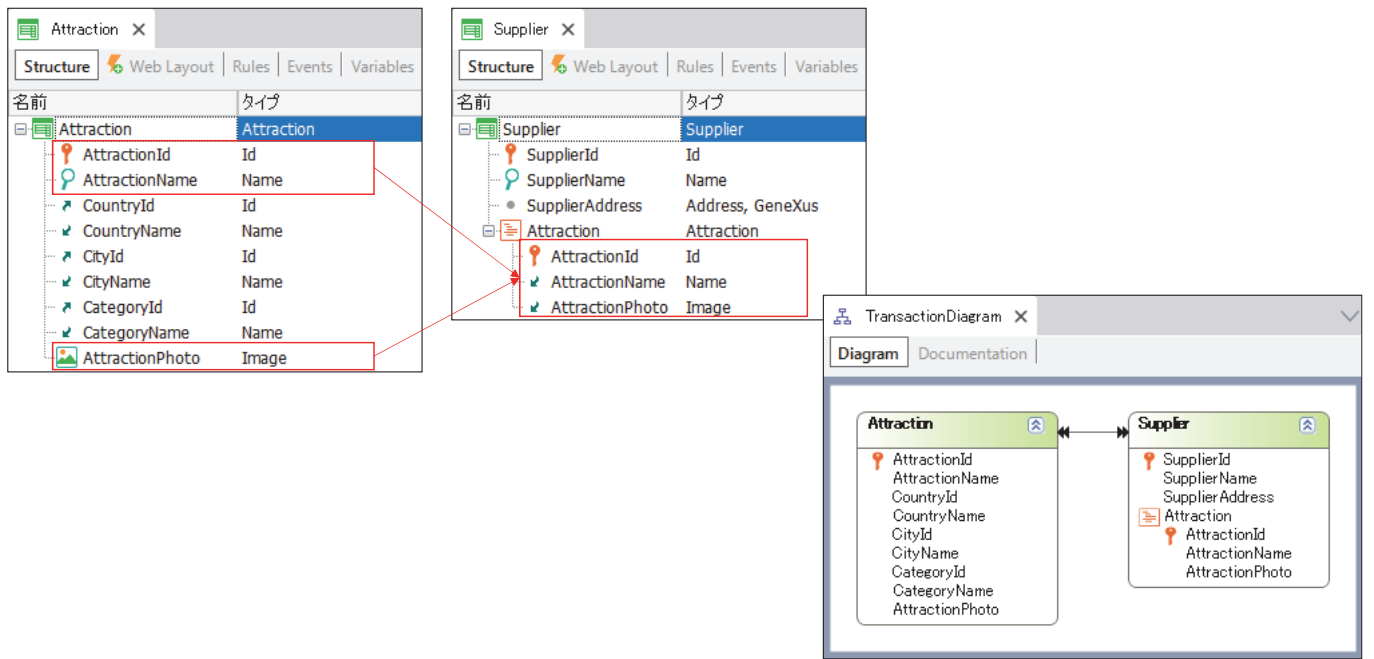
## N 対 M の関連を表す方法



N 対 M の関係を表す方法について、まずは概念を確認します。  
エンティティごとに 1 つずつ、2 つのトランザクションを定義します。  
さらに、一方を他方の第 2 レベルとして追加します。  
第 2 レベルとして追加する理由は、データの入力方法を考慮した結果です。

この場合、どちらの定義をもう一方の定義へ追加しても登録できるデータに差はありません。

## GeneXus における N 対 M の関連のモデリング



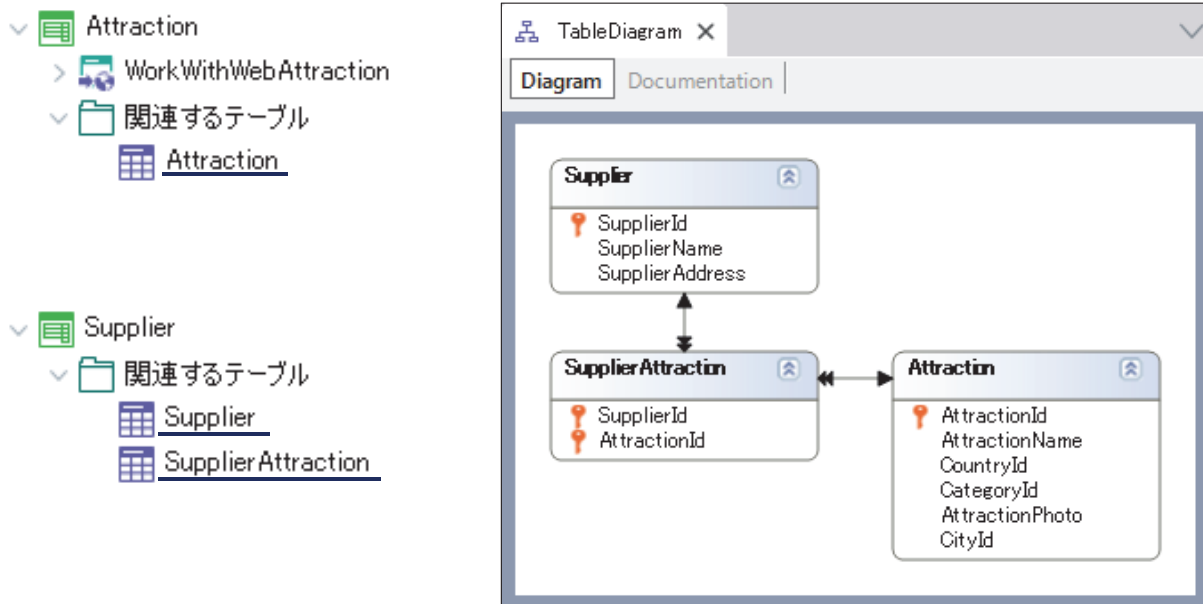
GeneXus における実装を確認します。

もし、この N 対 M の関係としたいトランザクション間に、1 対 N の関係がある場合には、まずはこれを削除します。

あるトランザクションで定義した項目属性を、別のトランザクションに追加した第 2 レベルの項目として追加します。

このトランザクションの実装を行うことで、ダイアグラムオブジェクトを確認すると、2 つのトランザクションが両端に二重の矢印で関係があることを確認できます。これは、N 対 M の関係性となっていることを表し、意図した通りの実装となっていることが確認できます。

## 設計を実装して作成されたテーブル / テーブル間の関連



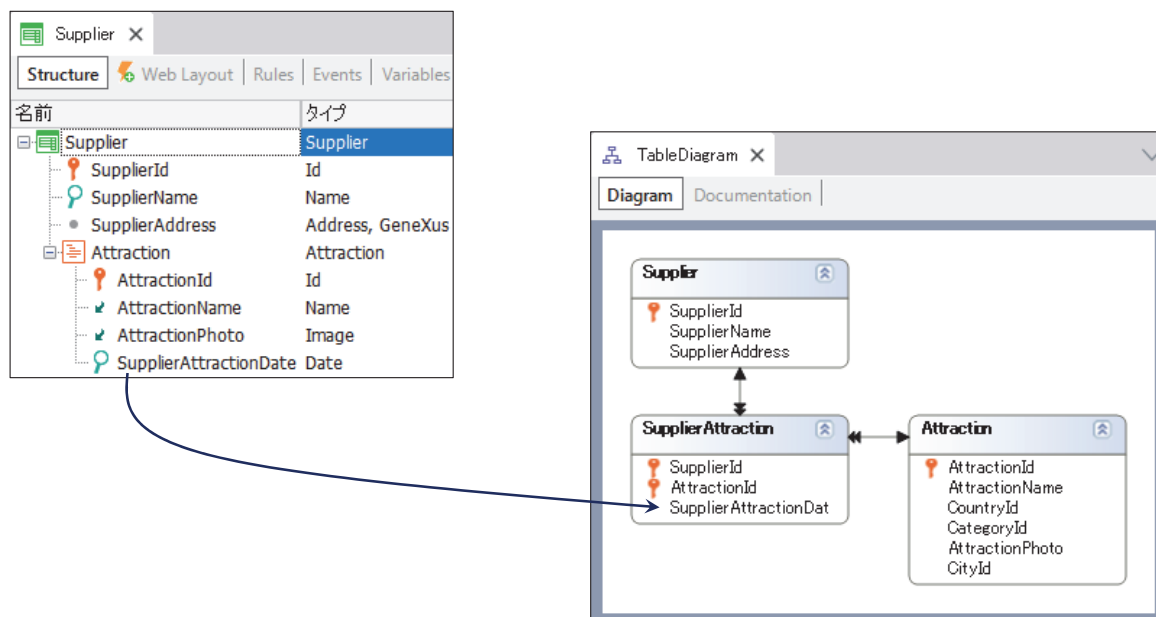
前述の通り、2 つのトランザクション定義があり、一方に第 2 レベルを定義し、N 対 M の関係を構築した場合、テーブルは 3 つ作成されます。このうち、2 つは、各トランザクションの第 1 レベルに対応するものであり、残りの 1 つが、第 2 レベルに対応するものです。

第 2 レベルに対応するテーブルには、各トランザクションの主キー項目が複合キーとしてのみ含まれている定義であることが確認できます。

ダイアグラムオブジェクトに、この 3 つのテーブルオブジェクトを配置すると、テーブルに基づく関係性が確認できます。トランザクションオブジェクトの関係性と異なり、テーブル間の場合、N 対 M の関係を表す矢印はなく、各第 1 レベルに対応するテーブルを 1 とし、第 2 レベルに対応するテーブルが N の関係で表されていることが確認できます。

この第 2 レベルに対応するテーブルによって、N 対 M の関係が実装されている状態となります。

## 外部参照以外の項目属性の追加



補足として、N 対 M の関係を実装するために追加した第 2 レベルには、独自の項目属性を追加することも可能です。  
この場合、追加された項目属性は、第 2 レベルに対応するテーブルに列として追加されます。

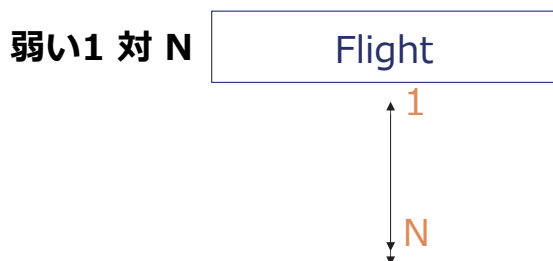
そのため、N 対 M として紐づけた際の関連付けに付加的な項目属性が必要な場合は、追加可能です。

## 実装可能な関係性

## 1 対 N



## 弱い1 対 N



## N 対 M



ここまでにご紹介してきた GeneXus で実装可能な関係性はスライドの通りです。

現実のエンティティ間のさまざまな関係は、トランザクションとその項目属性の定義によってあらわすことができます。

*GeneXus*<sup>TM</sup>

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