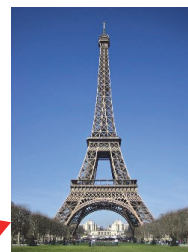


現実のエンティティ間の関連

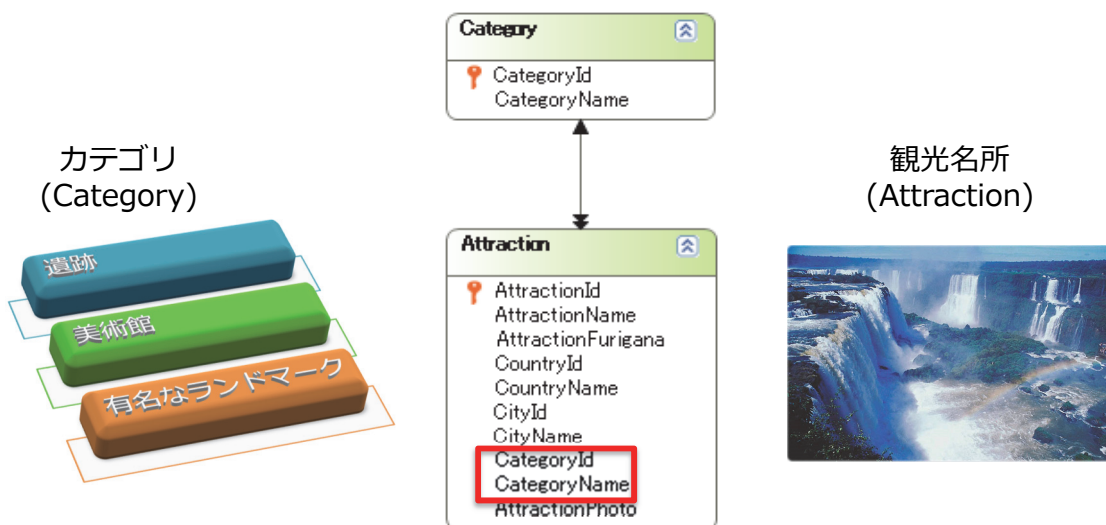
GeneXus™

このコースでは旅行代理店の例を通じて、現実のエンティティ群がいろいろなやり方で、相互に関連付けられていることを学習してきました。

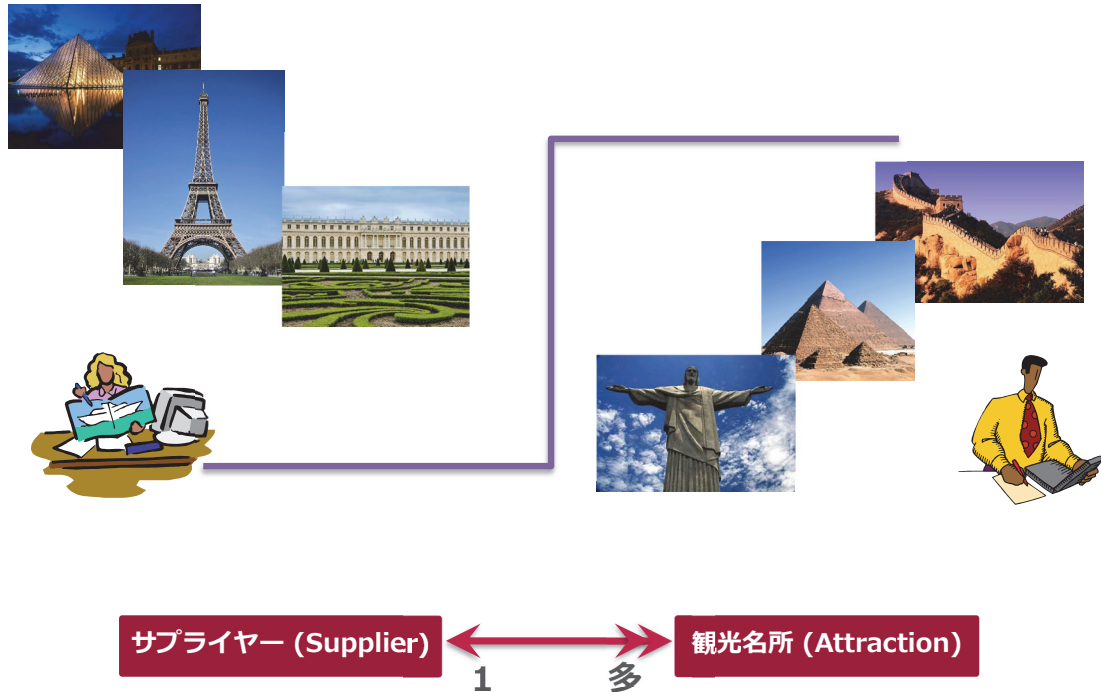
General	サブライヤー
観光名所番号	3
観光名所名	エッフェル塔
観光名所ふりがな	えっふえるとう
国番号	2
国名	フランス
都市番号	0
都市名	
カテゴリ番号	2
カテゴリ名	モニュメント



たとえば、ある観光名所が属しているカテゴリには、多数の観光名所が含まれていることがあります。



こうした関係は、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。



現在、旅行代理店では複数のサプライヤーと連携していて、サプライヤーは世界のさまざまな地域にある観光名所への旅行サービスを定期的に提供しているそうです。

各サプライヤーは多数の観光名所を提供していますが、各観光名所は 1 社のサプライヤーにより管理されます。

新しい Supplier トランザクション

Supplier X

Structure Web Form Win Form Rules Events Variables Help Documentation Patterns

名前	タイプ	DESCRIPTION	式	Null 許容
Supplier	Supplier	サプライヤー		
SupplierId	Id	サプライヤー番号		No
SupplierName	Name	サプライヤー名		No
SupplierAddress	Address, Ge...	サプライヤー住所		No

Diagram * Documentation

Attraction

- AttractionId
- AttractionName
- AttractionFurigana
- CountryId
- CountryName
- CityId
- CityName
- CategoryId
- CategoryName
- AttractionPhoto

Supplier

- SupplierId
- SupplierName
- SupplierAddress

観光名所と
サプライヤー間の
関連は？

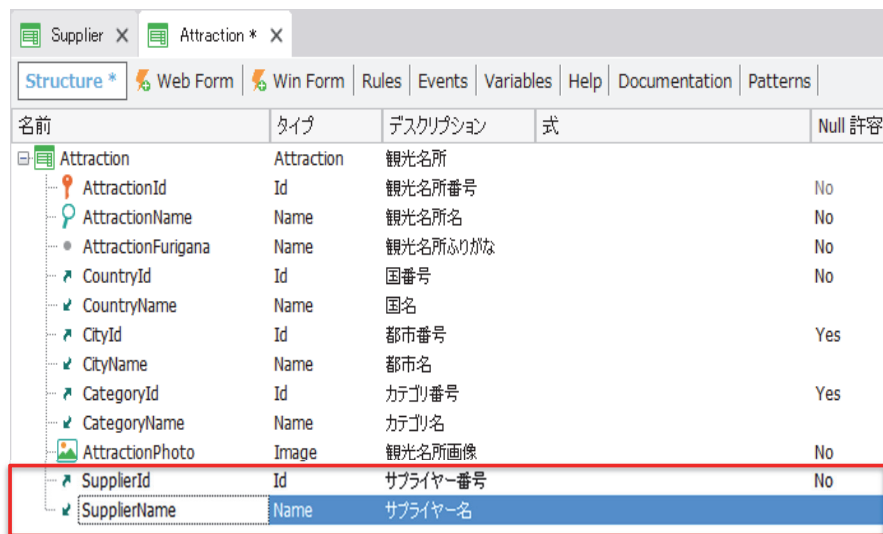
この現実を表すために、将来サプライヤーを登録する Supplier トランザクションを作成します。

トランザクションの Diagram オブジェクトから、サプライヤーと観光名所の関係を分析できます。

[新規] → [オブジェクト] で Diagram のタイプを選択し、KB エクスプローラーから Attraction トランザクションと Supplier トランザクションをドラッグします。

この 2 つのトランザクションの間には、まだ関係が確立されていないことが分かります。

要件: 各観光名所は 1 社のサプライヤーによって提供される

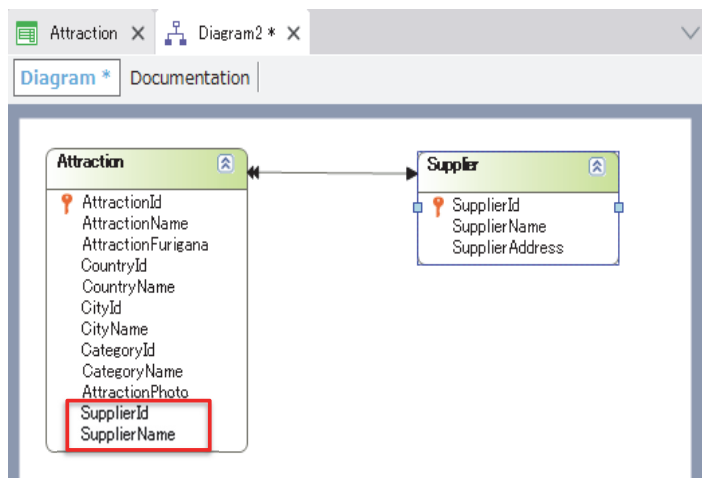


名前	タイプ	デスクリプション	式	Null 許容
Attraction	Attraction	観光名所		
AttractionId	Id	観光名所番号		No
AttractionName	Name	観光名所名		No
AttractionFurigana	Name	観光名所ふりがな		No
CountryId	Id	国番号		No
CountryName	Name	国名		
CityId	Id	都市番号		Yes
CityName	Name	都市名		
CategoryId	Id	カテゴリ番号		Yes
CategoryName	Name	カテゴリ名		
AttractionPhoto	Image	観光名所画像		No
SupplierId	Id	サプライヤー番号		No
SupplierName	Name	サプライヤー名		

1 つの観光名所は 1 社のサプライヤーのみが提供しているので、そのサプライヤーの識別子を Attraction トランザクションの構造に追加します。Attraction トランザクションを開き、SupplierId 項目属性を追加します。

SupplierName 項目属性も追加して、観光名所に対応するサプライヤーの名前を画面で表示できるようにします。

サプライヤーと観光名所の関係



トランザクションの
ダイアグラム
(テーブルのダイア
グラムではない)

トランザクションのダイアグラムを再度開きます。

Supplier 方向を指す一重矢印と、Attraction 方向を指す二重矢印が表示されています。これは、各観光名所のサプライヤーは 1 社のみであり、1 社のサプライヤーが多数の観光名所を提供できることを示しています。

つまり、あるトランザクションの識別子となっている項目属性を別のトランザクション (前述のように、ここでは外部キーとして機能) に追加すると、1 対多 (あるいは「1 対 N」) の関係が確立され、関係の「多」の側に外部キーが格納されます。

設計を実装して作成されたテーブル

名前	タイプ	デスクリプション	式
Supplier 構造		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	

名前	タイプ	デスクリプション	式	Null 許容
Supplier	Supplier	サプライヤー		
SupplierId	Id	サプライヤー番号		No
SupplierName	Name	サプライヤー名		No
SupplierAddress	Address, Ge...	サプライヤー住所		No

GeneXus でこのようなトランザクション設計を実装すると、Supplier トランザクションからトランザクションと同じ構造の Supplier テーブルが作成されます。

Attraction X				
Structure Indexes				
名前	タイプ	デスクリプション	式	
Attraction 構造		観光名所		
AttractionId	Id	観光名所番号		
AttractionName	Name	観光名所名		
AttractionFurigana	Name	観光名所ふりがな		
CountryId	Id	国番号		
CategoryId	Id	カテゴリ番号		
AttractionPhoto	Image	観光名所画像		
CityId	Id	都市番号		
SupplierId	Id	サプライヤー番号		

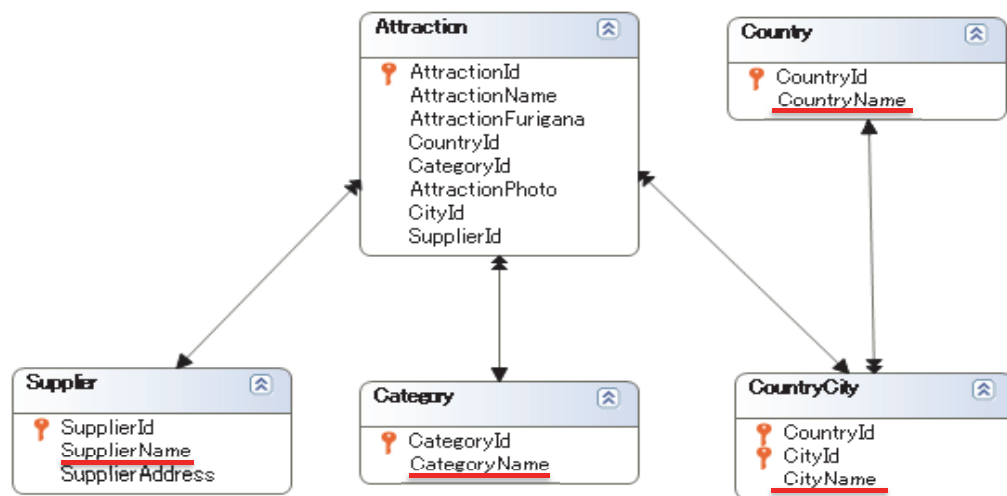
テーブル

Attraction X				
Structure Web Form Win Form Rules Events Variables Help Documentation Patterns				
名前	タイプ	デスクリプション	式	Null 許容
Attraction	Attraction	観光名所		
AttractionId	Id	観光名所番号		No
AttractionName	Name	観光名所名		No
AttractionFurigana	Name	観光名所ふりがな		No
CountryId	Id	国番号		No
CountryName	Name	国名		
CityId	Id	都市番号		
CityName	Name	都市名		
CategoryId	Id	カテゴリ番号		Yes
CategoryName	Name	カテゴリ名		
AttractionPhoto	Image	観光名所画像		No
SupplierId	Id	サプライヤー番号		No
SupplierName	Name	サプライヤー名		

トランザクション

そして、Attraction トランザクションの構造から、ここに示すような Attraction テーブルが作成されます。

Attraction テーブルの構造を Attraction トランザクションの構造と比較すると、CountryName、CategoryName、CityName、SupplierName の各項目属性がテーブルに含まれていないことが分かります。これらの項目属性は推論できるためです。前述のように、これらの項目属性は Attraction テーブルの拡張テーブル内にあるため、物理的に格納されているテーブルから値を取得することができます。





これは、現実の 2 つのエンティティ、つまり、システム内の 2 つのエンティティ間の 1 対多の関係を表す最も一般的な方法です。

ただし、これとは異なる 1 対多の関係もあり、この場合は別の表現を使用します。

フライトの例を思い出してみましょう。1 つのフライトには多数の座席があり、各座席が単一のフライトに割り当てられます。つまり、1 対多の関係です。

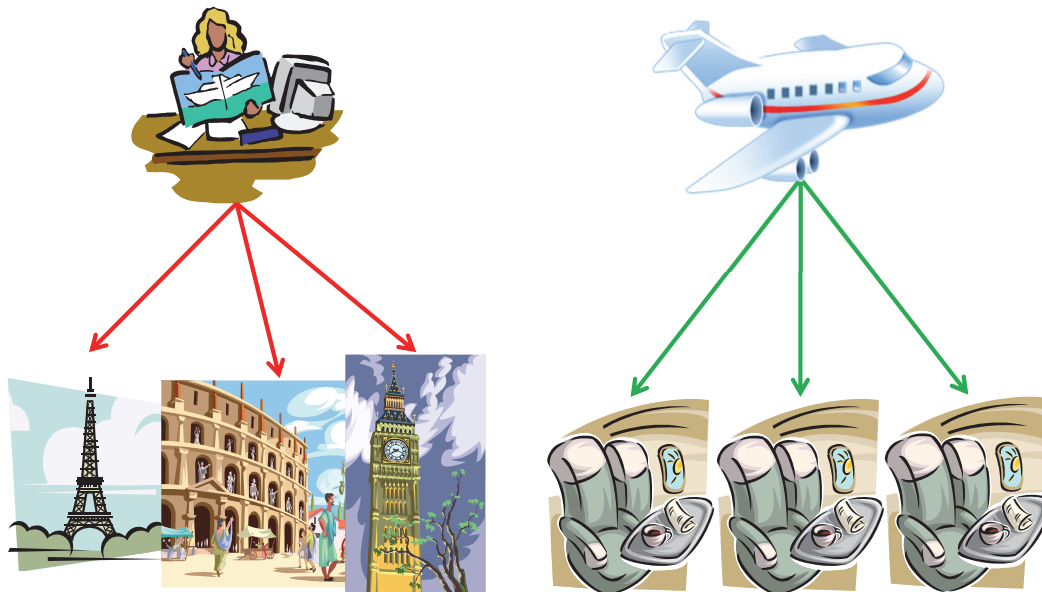
Flight トランザクションの構造を開いて、関係をどのように表現しているかを確認してみましょう。

1 対 N の関係をモデル化する方法

名前	タイプ	デスクリプション	式	Null 許容
Flight	Flight	フライト		
FlightId	Id	フライト番号		No
FlightDepartureAirportId	Id	出発空港番号		No
FlightDepartureAirportName	Name	出発空港名		
FlightDepartureCountryId	Id	出発空港国番号		
FlightDepartureCountryName	Name	出発空港国名		
FlightDepartureCityId	Id	出発空港都市番号		
FlightDepartureCityName	Name	出発空港都市名		
FlightArrivalAirportId	Id	到着空港番号		No
FlightArrivalAirportName	Name	到着空港名		
FlightArrivalCountryId	Id	到着空港国番号		
FlightArrivalCountryName	Name	到着空港国名		
FlightArrivalCityId	Id	到着空港都市番号		
FlightArrivalCityName	Name	到着空港都市名		
FlightPrice	Price	フライト価格		No
FlightDiscountPercentage	Percentage	フライト割引率		No
AirlineId	Id	航空会社番号		Yes
AirlineName	Name	航空会社名		
AirlineDiscountPercentage	Percentage	航空会社割引率		
FlightFinalPrice	Price	フライト最終価格	FlightPrice*(1-AirlineDiscountPercentage... count(FlightSeatLocation)	
FlightCapacity	Numeric(4,0)	フライト座席数		
Seat	Seat	座席		
FlightSeatId	Id	座席番号		No
FlightSeatChar	SeatChar	座席文字		No
FlightSeatLocation	Location	座席位置		No



この例では、Seat は Flight トランザクションの第 2 レベルです。

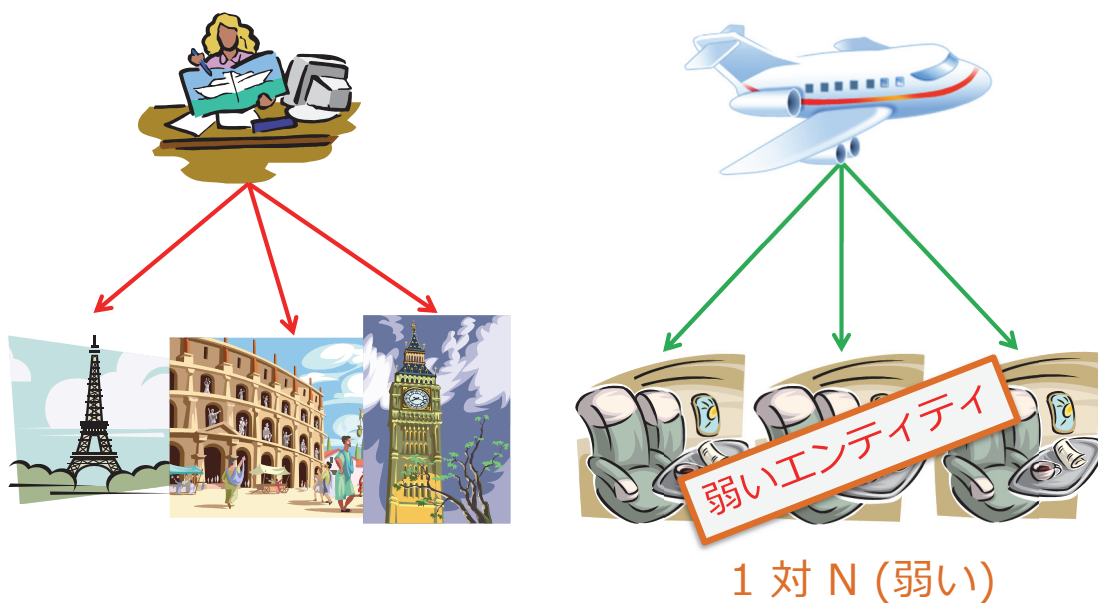


では、この 1 対多の関係と、観光名所とサプライヤーの 1 対多の関係との違いは何でしょうか。

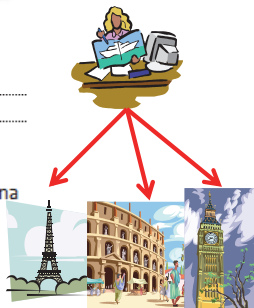
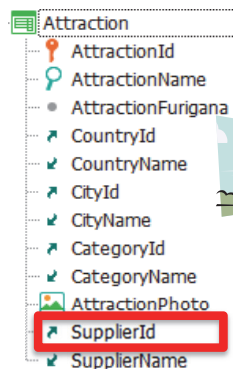
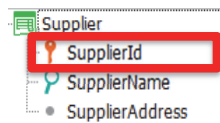
どちらも同じ方法 (同じトランザクション設計) で表さないのはなぜでしょうか。

フライトに関連付けられていない座席は無意味です。座席は、フライトと常に関連付けて考えなければ意味がありません。しかし、観光名所は、その場所への旅行を取り扱うサプライヤーがいなくても、それ自体で存在することができます。

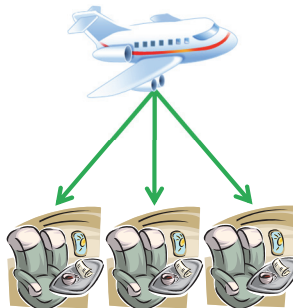
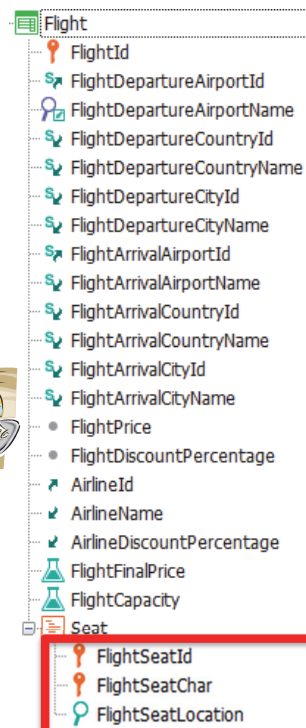
もう 1 つの違いは、フライトのデータを入力する場合は、フライトの座席に関するデータも入力します (請求書とその中の明細行を入力する場合と同様に、すべての情報を同時に入力します)。しかし、サプライヤーと観光名所のデータは、必ずしも同時に入力する必要はありません。



座席のように、別のエンティティ (この例ではフライト) と関連付けることで初めて存在できるエンティティを、**弱いエンティティ**と呼びます。

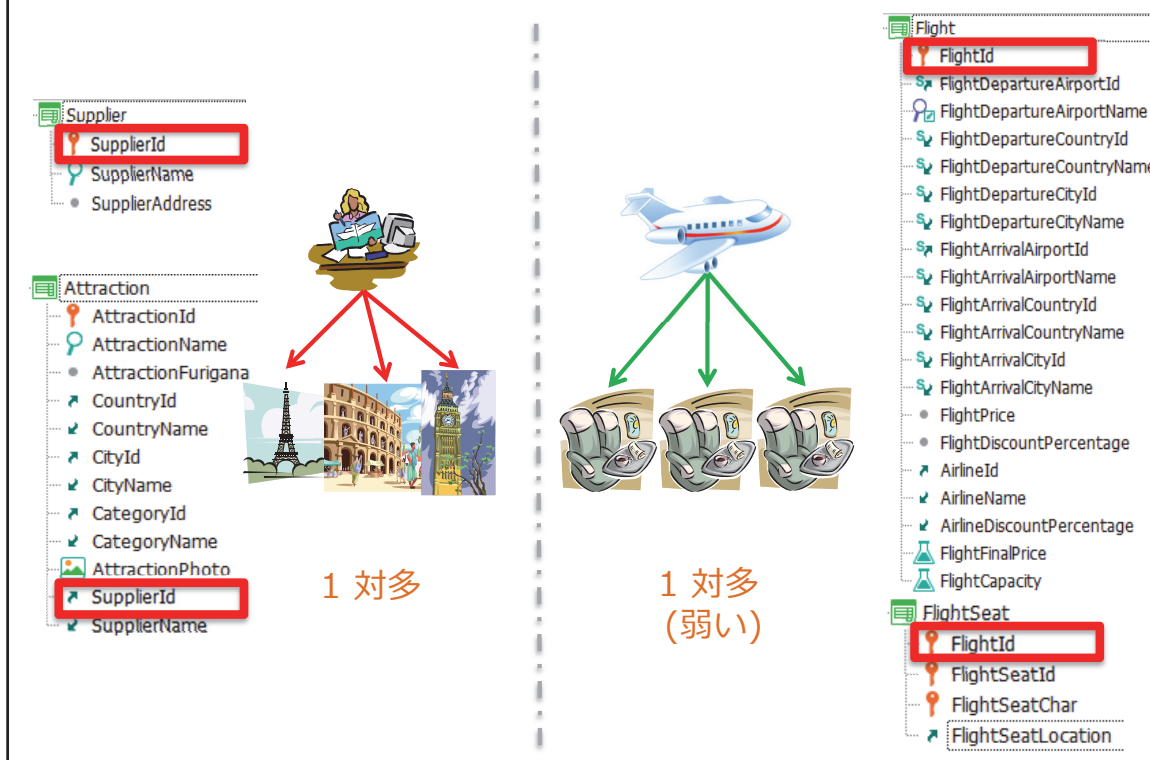


1 対多

1 対多
(弱い)

この**弱い 1 対 N の関係**は、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。

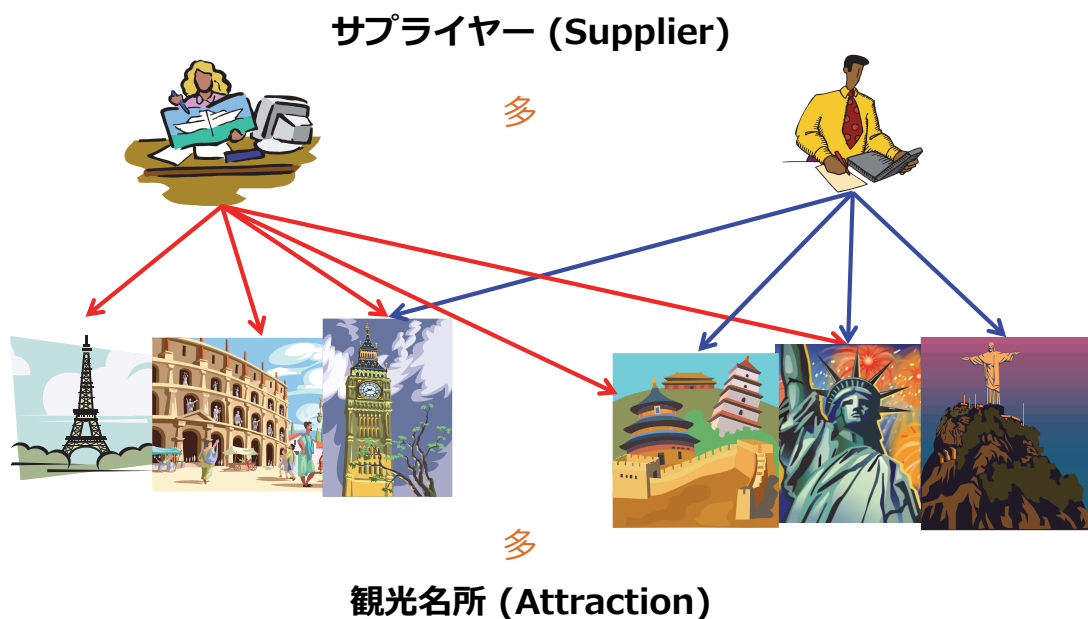
これは、サプライヤーと観光名所の 1 対 N の関係とは異なります。サプライヤーと観光名所の場合は、**2 つ**のトランザクションを作成し、一方に他方の主キーを外部キーとして設定しました。



(前ページとまったく同じデータモデルを目指すために) 弱い 1 対多の関係を 2 つのトランザクションで表すこともできます。FlightSeat トランザクションはその主キーの一部に FlightId 項目属性を含んでいますが、これは Flight テーブルの外部キーになります。

弱いエンティティと強いエンティティには相違点があります。FlightId は主キーの一部であるため、FlightId の値を指定せずにフライトの座席 (2A 窓側など) を定義することはできません。これに対して、SupplierId 項目属性の [Nullable] プロパティの値が「Yes」の場合、サプライヤーを指定せずに観光名所を入力できます。

「多」対「多」の関係 (M 対 N)



これまで 1 対多の関係について説明しましたが、現実に基づき表さなければならぬケースは、これだけではありません。

次のように、**旅行代理店の現実が変わった場合**を考えてみましょう。

各サプライヤーは、(これまで説明したように) 複数の観光名所への旅行を取り扱います。しかし、**各観光名所は、(今までの 1 社のみではなく) 複数のサプライヤーで管理される場合があります。**

つまり、サプライヤーと観光名所の関連は、「1 対多」ではなく「**多対多**」になります。

M 対 N の関連を表す方法



まずデータの入力方法を考えます。各サプライヤーにすべての観光名所を入力する方法と、各観光名所にすべてのサプライヤーを入力する方法があります。

この例の旅行代理店では、ユーザーがサプライヤーを入力するときに、各サプライヤーが取り扱う観光名所を入力できるようにすることを求めています。

その方法をお見せしましょう。

GeneXus における M 対 N の関連のモデリング

The screenshot displays the GeneXus IDE interface with two entity structure windows. The top window shows the 'Attraction' entity, and the bottom window shows the 'Supplier' entity. Both windows have tabs for Structure, Web Form, Win Form, Rules, Events, Variables, Help, Documentation, and Patterns. The 'Structure' tab is active in both.

名前	タイプ	デスクリプション	式	Null 許容
Attraction	Attraction	観光名所		
AttractionId	Id	観光名所番号		No
AttractionName	Name	観光名所名		No
AttractionFurigana	Name	観光名所ふりがな		No
CountryId	Id	国番号		No
CountryName	Name	国名		
CityId	Id	都市番号		Yes
CityName	Name	都市名		
CategoryId	Id	カテゴリ番号		Yes
CategoryName	Name	カテゴリ名		
AttractionPhoto	Image	観光名所画像		No

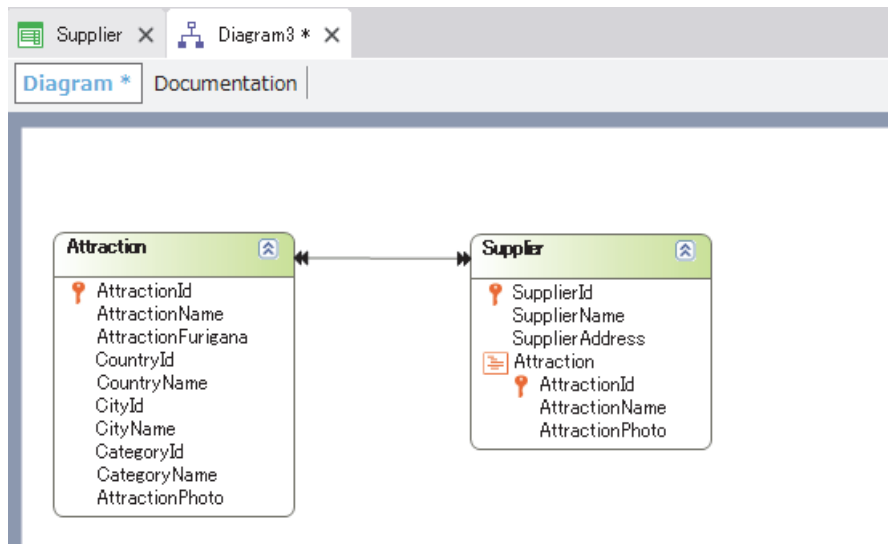
名前	タイプ	デスクリプション	式	Null 許容
Supplier	Supplier	サプライヤー		
SupplierId	Id	サプライヤー番号		No
SupplierName	Name	サプライヤー名		No
SupplierAddress	Address, GeneXus	サプライヤー住所		No
Attraction	Attraction	観光名所		
AttractionId	Id	観光名所番号		No
AttractionName	Name	観光名所名		
AttractionPhoto	Image	観光名所画像		



Attraction トランザクションを開き、SupplierId および SupplierName 項目属性を削除して保存します。

Supplier トランザクションを開き、2 つ目のレベルを追加して AttractionId 項目属性を追加します (Attraction で始まるキー項目属性を記述する際に、レベル名が自動的に Attraction に変わります)。さらに、AttractionName と AttractionPhoto も追加します。

サプライヤーと観光名所間の関連



どのような関連かを確認するために、Attraction トランザクションと Supplier トランザクションの間のトランザクションダイアグラムを開きます。

今度は、関連の両端に二重矢印があります。これは、関連が「多」対「多」であり、1 つの観光名所への旅行を複数のサプライヤーが取り扱っていて、1 社のサプライヤーが複数の観光名所への旅行を取り扱っていることを意味します。

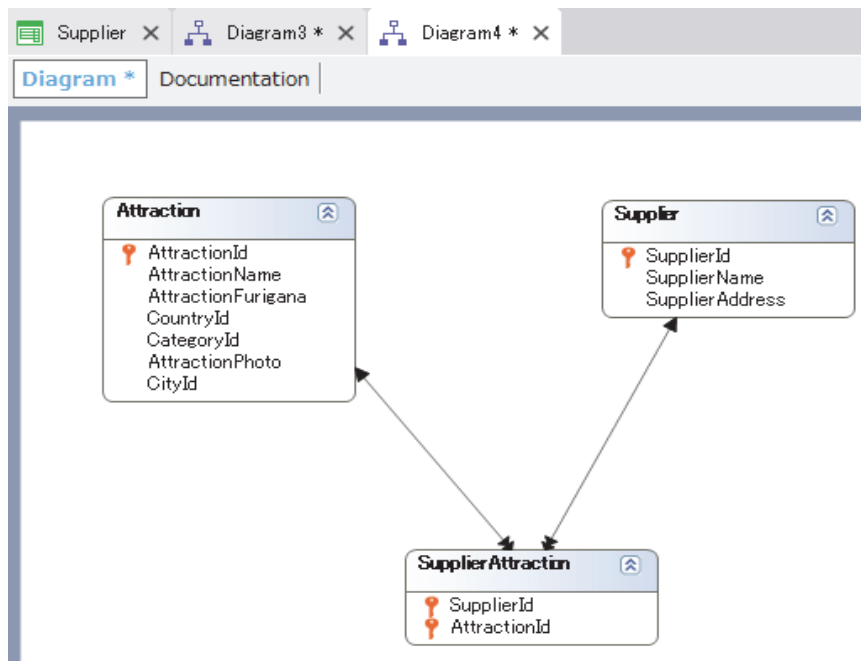
設計を実装して作成されたテーブル



前述の設計に基づいて作成されるテーブルを見てみましょう ([表示] → [テーブル])。

Attraction テーブル、Supplier テーブル、および SupplierAttraction というテーブルがあることが分かります。

テーブル間の関連: M 対 N



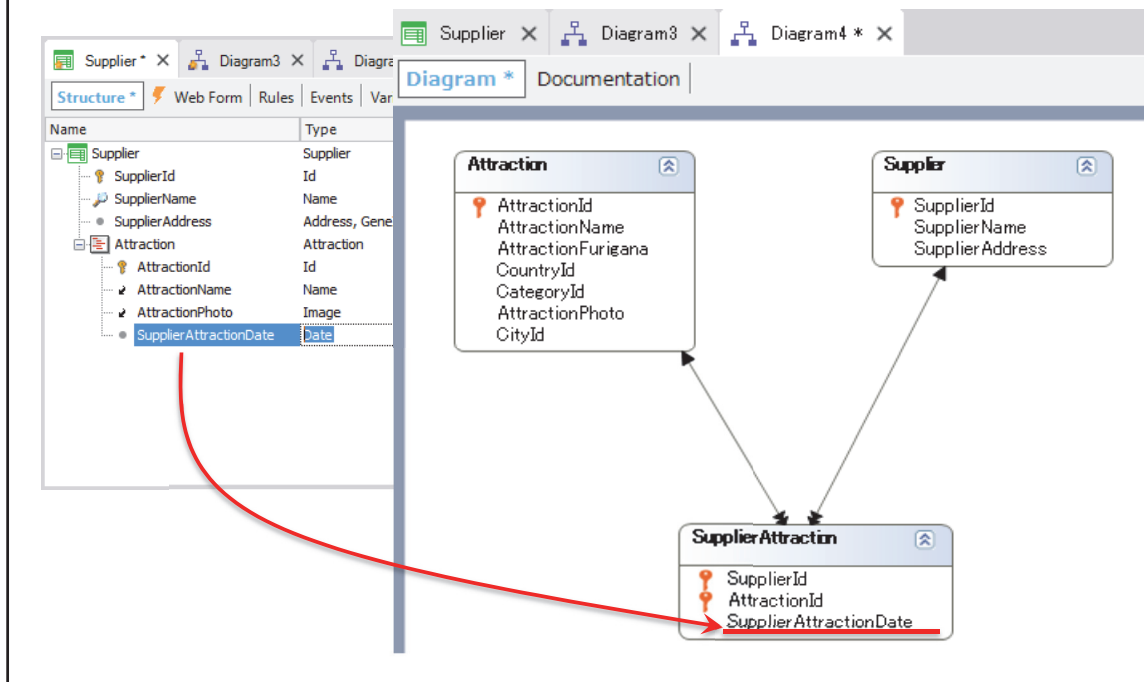
次は、新しい Diagram オブジェクトを作成し、3 つのテーブルをそのダイアグラムにドラッグします。

この例では、多対多の関連の各トランザクション (Attraction と Supplier) に対してテーブルが 1 つずつ作成されます。また、この関係を確認するために、SupplierAttraction という 3 つ目のテーブルも作成されます。

この 3 つ目のテーブルの構造を分析すると、ほかの 2 つのテーブルを識別する項目属性のみが含まれていることが分かります。

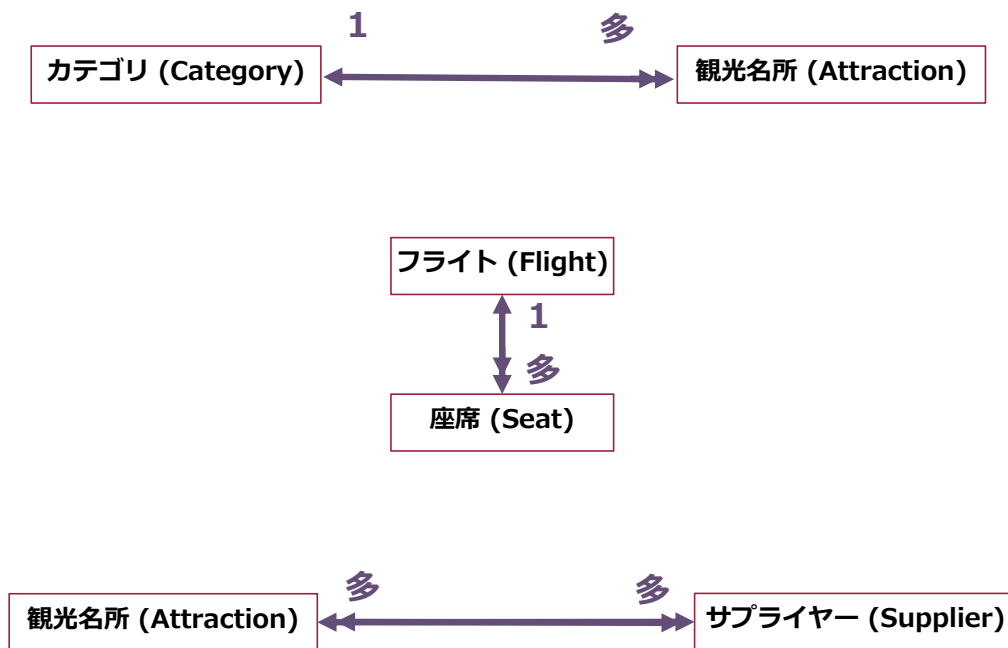
このように、GeneXus で多対多の関連付けが行われるたびに、その関連はデータベース内の 3 つのテーブルで表されます。すなわち、各エンティティが含まれている 2 つのテーブルと、これらの識別子の 3 つ目のテーブルです。

テーブル間の関連: M 対 N

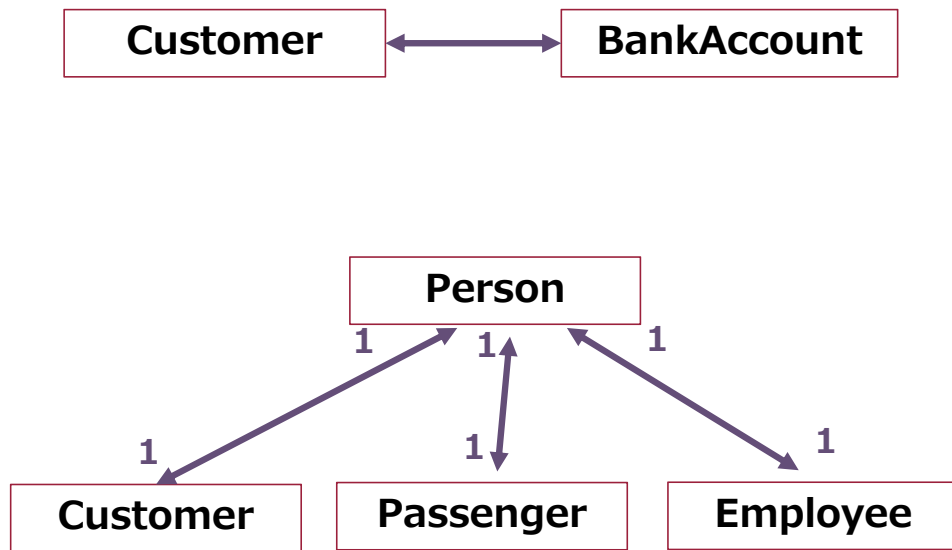


この 3 つ目のテーブルには、サプライヤーが観光名所への旅行サービスを開始した日付など、独自の項目属性を含めることができます。ダイアグラムを再度開くと、関連テーブルに項目属性が表示されます。

Attraction と Supplier の「多対多」の関連は、2 つの「1 対多」の関連に分解され、SupplierAttraction テーブルを使用してこれらの関連付けが行われます。



このように、現実のエンティティ間の関連は、トランザクションとその項目属性を使用して表すことができます。



このコースでは 1 対 1 の関係については学習しません。たとえば、旅行代理店で顧客と旅行代金の支払口座を関連付ける必要がある場合、顧客と口座は 1 対 1 の関係です。

1 対 1 の関係の別のシナリオについては、サブタイプについて学習したときに説明しました。これはあるエンティティが、そのほかのエンティティに特化するスペシャライゼーションのケースでした。

これらのシナリオに興味がある場合は、関連する資料を参照してください。それでは次のトピックに進みましょう。

