

式

GeneXus[™]

$\text{項目属性}_1 + \text{項目属性}_2$ $\text{項目属性}_3 - \text{項目属性}_4$ **式****Average**(項目属性₇)**Sum**(項目属性₅)**Max**(...)**Count**(項目属性₆)**Min**(...)

アプリケーションでは、特定の項目属性、定数、関数を含めた計算が必要になる場合があります。

このような場合のために、GeneXus ではさまざまな**式**を用意しています。

2 タイプの式:

ナレッジベース

名前

名前	タイプ	DESCRIPTION	式
Flight	Flight	フライト	
FlightId	Id	フライト番号	
FlightDepartureAirportId	Id	出発空港番号	
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	No
FlightArrivalCountryId	Id	到着空港国番号	No
FlightArrivalCountryName	Name	到着空港国名	No
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	
FlightDiscountPercentage	Numeric(3,0)	フライト割引率	
FlightFinalPrice	Price	フライト最終価格	$FlightPrice * (1 - FlightDiscountPercentage / 100)$

ローカル (インライン) 式

```

1 Print header
2 For each Country
3   &AttractionQty = Count(AttractionName)
4   print Country
5 -endfor
6

```

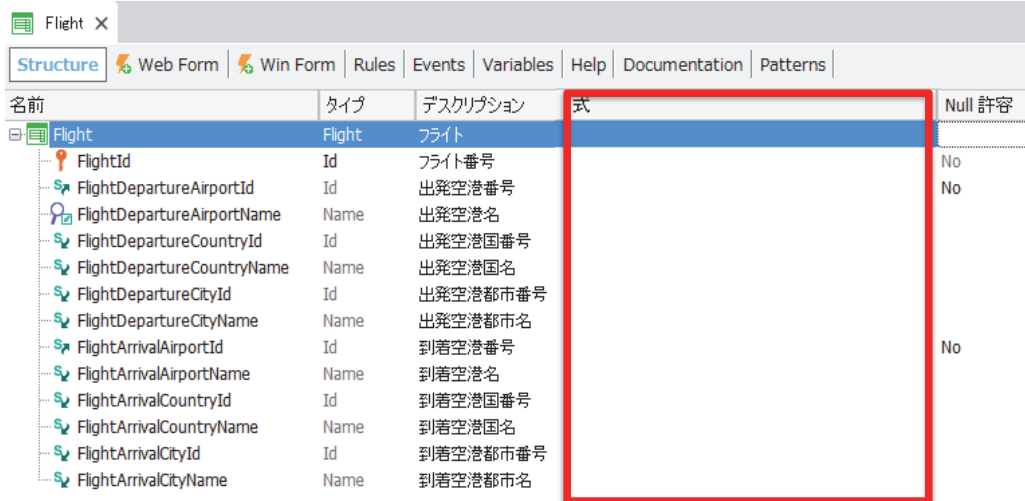
式は、2 つの方法で定義できます。

ナレッジベース全体で認識される**グローバル式**と、定義されたオブジェクト内でのみ認識される**ローカル式**です。

まずは、**グローバル式とはなにか**、どう定義するのかを学習しましょう。

グローバル式

項目属性と関連付けて定義する計算で、定義後は「仮想」項目属性になる



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		

グローバル式とは、項目属性に関連付けて定義する計算です。

トランザクションの [構造] に、[式] という列が含まれています。

この列で項目属性の計算を定義すると、**GeneXus に対してこの項目属性は仮想的なものであると示すことになります。**つまり、項目属性の値は指定された計算を実行することによって取得されるため、フィールドとして物理的に作成する必要はありません。

例を見ていきましょう。

グローバル式の追加

Flight 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
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	Numeric(3.0)	フライト割引率		No
FlightFinalPrice	Price	フライト最終価格	$\text{FlightPrice} * (1 - \text{FlightDiscountPercentage} / 100)$	

FlightPrice 項目属性は Price データタイプで定義されています。このドメインは次のように記述されていました: 「Price = Numeric(9.2)」。

Flight トランザクションに 2 つの新しい項目属性を定義して、フライト料金と各フライトに適用可能な現在の割引率を格納します。

FlightDiscountPercentage という項目属性を入力します。このデータタイプは、3 桁の数値で定義する Percentage ドメインです。

もう 1 つ、FlightFinalPrice という項目属性を定義します。この項目属性は、現在のフライト料金を自動的に計算する式で定義します。

グローバル式の追加

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		
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	Numeric(3.0)	フライト割引率		No
FlightFinalPrice	Price	フライト最終価格	FlightPrice*(1-FlightDiscountPercentage/100)	...

[式] 列で、次の計算を定義します:

$\text{FlightPrice} * (1 - \text{FlightDiscountPercentage} / 100)$

F5 キーを押して、どのように処理されるかを確認します。

グローバル式の追加

データベースの再編成が必要です。

このレポートにはデータベースの変更内容と、再編成プログラムでどのように扱われるかが記載されています。
「再編成」を押して続けるか、「キャンセル」を押してキャンセルしてください。

再編成 キャンセル

フィルタ: Flight

Table Flight specification

Table name: [Flight](#)

Flight needs conversion

Warnings

- ▲ [rqz0007Attribute FlightPrice](#) does not allow nulls and does not have an Initial Value. An empty default value will be use
- ▲ [rqz0007Attribute FlightDiscountPercentage](#) does not allow nulls and does not have an Initial Value. An empty default va

Table Structure

Attribute	Definition	Previous values
FlightId	Numeric (4) Not null Autonumber	
FlightArrivalAirportId	Numeric (4) Not null	
FlightDepartureAirportId	Numeric (4) Not null	
FlightPrice	Numeric (9,2) Not null	
New FlightDiscountPercentage	Numeric (3) Not null	

Indexes

Name	Definition	Composition

エラー: 0 警告: 1 成功: 0

Flight トランザクションの構造に新しい項目属性を 2 つ定義しましたが、Flight 物理テーブルにはフィールドが 1 つしか作成されません。

これは、[式] 列を定義したため、**この項目属性は物理テーブルに追加されないからです。**

この項目属性には、関連する式がナレッジベース内で定義されているため、GeneXus で値の計算方法が認識されます。これにより、この項目属性が存在するすべてのオブジェクトで計算が実行され、結果が表示されます。

ここで、再編成します。

グローバル式を含むアプリケーションの実行

FlightId 1

到着空港国番号	2		
到着空港国名	フランス	到着空港都市名	パリ
到着空港都市番号	1		
到着空港都市名	パリ		
フライト価格	0.00	フライト価格	1500.00
フライト割引率	0	フライト割引率	10
フライト最終価格	0.00	フライト最終価格	1350.00

実行
終了
削除



プログラムは生成済みのため、ここでは、実行時のアプリケーションを確認します。

Flight トランザクションを実行し、Flight 1 を参照します。トランザクションのフォームに、定義した 3 つの新しい項目属性が表示されます：

- 入力可能なフライト料金
- 入力可能な割引率
- 最終料金。式を使用して定義した項目属性なので、入力できません。計算の結果が値として表示されます。

グローバル式として定義されたすべての項目属性は、データの読み取りのみとなり、値を入力することはできません。これは、項目属性は関連付けられた計算から値を取得し、計算は使用されている項目属性が変更されるたびに実行されるためです。

そのため、物理テーブルにはこの項目属性の値を格納するフィールドがなく、編集可能とする必要はありません。

このフライトの料金を入力し、フライトの割引率として 10% を入力します。フィールドから移動するとただちに式が実行され、割引率を適用したフライト最終料金が表示されます。

グローバル式での拡張テーブル項目属性の使用

名前

名前	タイプ
Airline	Airline
AirlineId	Id
AirlineName	Name
AirlineDiscountPercentage	Percentage

Airline トランザクションの作成:
Airline の項目属性を Flight に追加

名前

名前	タイプ	デスクリプション	式	Null 許容
FlightDepartureAirportId	Id	出発空港番号		No
FlightDepartureAirportName				
FlightDepartureCountryId				
FlightDepartureCountryName				
FlightDepartureCityId				
FlightDepartureCityName				
FlightArrivalAirportId				
FlightArrivalAirportName				
FlightArrivalCountryId				
FlightArrivalCountryName	Name	到着空港国名		No
FlightArrivalCityId	Id	到着空港都市番号		No
FlightArrivalCityName	Name	到着空港都市名		No
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		No
AirlineName	Name	航空会社名		No
AirlineDiscountPercentage	Percentage	航空会社割引率		Yes
FlightFinalPrice	Price	フライト最終価格	$\text{FlightPrice} * (1 - \text{AirlineDiscountPercentage} / 100)$	

+ 航空会社の割引率に基づいてフライト料金を決定:

ここまでで、**グローバル式**の項目属性を定義しました。

ここで説明した方法で、トランザクションの [式] 列を利用し、**項目属性のみ**が定義されていました。

例に挙げた計算では、トランザクションそのものの項目属性のみを含めましたが、**拡張テーブルの項目属性を含めることもできる**ということを覚えておいてください。

詳しく見ていきましょう。

航空会社を記録するための Airline という新しいトランザクションを作成し、航空会社がそのすべてのフライトに適用する割引率を記録する AirlineDiscountPercentage 項目属性をそれに含めます。
Flight トランザクションで、運航する航空会社を追加します。
そのため、外部キーの役割を果たす AirlineId 項目属性を追加します。
その [Nullable] プロパティの値を [Yes] に変更します。まだ記録されている航空会社がないため、このように設定することで、この段階でのフライト航空会社の指定を回避できます。

後で [Nullable] プロパティの値を [No] に変更して、フライト詳細の入力時または変更時に航空会社の指定を必須にすることができます。
また、AirlineName 項目属性と AirlineDiscountPercentage 項目属性も追加し、フォームにこれらのデータも表示します。
次に、フライト自体の割引率ではなく、航空会社が設定している一般的な割引率を適用してフライトの最終料金を計算するように、式の定義を変更します。

グローバル式での拡張テーブル項目属性の使用

データベースの再編成が必要です。

このレポートにはデータベースの変更内容と、再編成プログラムでどのように扱われるかが記載されています。
「再編成」を押して続けるか、「キャンセル」を押してキャンセルしてください。

フィルタ:

- ☒ Airline
- ☒ Flight

Table Flight specification			
Table name: Flight			
Flight needs conversion			
Table Structure			
Attribute	Definition	Previous values	Takes value from
FlightId	Numeric (4) Not null Autonumber		Flight. FlightId
FlightArrivalAirportId	Numeric (4) Not null		Flight. FlightArrivalAir
FlightDepartureAirportId	Numeric (4) Not null		Flight. FlightDeparture
FlightPrice	Numeric (9.2) Not null		Flight. FlightPrice
FlightDiscountPercentage	Numeric (3) Not null		Flight. FlightDiscountF
New AirlineId	Numeric (4)		Null

Indexes		
Name	Definition	Composition
IFLIGHT	primary key Clustered	FlightId
IFLIGHT1	duplicate	FlightArrivalAirportId
IFLIGHT2	duplicate	FlightDepartureAirportId
New IFLIGHT3	duplicate	AirlineId

Foreign key constraints	
Referenced table	Attributes
Airport	FlightArrivalAirportId
Airport	FlightDepartureAirportId
New Airline	AirlineId

定義した 3 つの項目属性で Airline 物理テーブルを作成し、外部キー AirlineId を Flight テーブルに追加するという影響分析レポートを確認します。

再編成して実行してみます。

式

GeneXus™

実行時の式のテスト

Application Name

Recents 航空会社

航空会社

航空会社番号

1

航空会社名

TAM

航空会社割引率

30

実行

終了

到着空港国名

ブラジル

到着空港都市番号

2

到着空港都市名

サンパウロ

フライト価格

3000.00

フライト割引率

10

航空会社番号

1

航空会社名

TAM

航空会社割引率

30

フライト最終価格

2100.00

実行


終了

削除

Airline トランザクションを開き、TAM という航空会社を 30% の割引率で追加します。

ここで、この航空会社にフライトを割り当てます。フライトナンバー 1 に対して航空会社 1 を割り当てると、**航空会社の割引率**を含むグローバル式の項目属性として、フライトの最終料金が新たに計算されることが分かります。この割引率は、Flight ベーステーブルの拡張テーブルの項目属性です。

式への条件の追加

項目属性 fx =  { 式₁ if 条件₁;
式₂ if 条件₂;
...
式_n if 条件_n;

まだ説明していないことがありました。式で条件を評価し、その条件が True か False かに応じて異なる方法で結果を計算することができます。詳しく見ていきましょう。

式への条件の追加

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		
FlightArrivalCityName	Name	到着空港都市名		No
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/100) ...	

式エディター

```
FlightPrice*(1-AirlineDiscountPercentage/100) IF AirlineDiscountPercentage>=FlightDiscountPercentage;
FlightPrice*(1-FlightDiscountPercentage/100) OTHERWISE;
```

OK(O) キャンセル

結果は、条件が True かどうかに応じて、
どちらかの方法で計算される



[...] ボタン (3 つのドットのボタン) をクリックすると、簡単に式を編集できます。

フライトの最終料金の計算に最も高い割引率を考慮するように定義し、最も安い料金を得ることにします。

この定義では、航空会社の割引率が、フライト個別の割引率よりも大きな割引率を設定している場合に、航空会社の割引率の計算を行います。
それ以外の場合は、フライト個別の割引率を計算で使用します。

次に F5 キーを押します。

式
GeneXus™

到着空港国名	ブラジル
到着空港都市番号	2
到着空港都市名	サンパウロ
フライト価格	3674.00
フライト割引率	50
航空会社番号	1
航空会社名	TAM
航空会社割引率	30
フライト最終価格	1504.00

実行
終了
削除


DEMO

新しい式の定義のテスト

Flight トランザクションを実行し、1 つ目のフライトに対して、航空会社の一般的な割引率よりも大きな割引率を個別に割り当てます。たとえば次の割引率とします：50%。

フライトの最終料金は、大きいほうの割引率を考慮して計算されたことが分かります。

既定の条件も追加

項目属性 fx =  { 式₁ if 条件₁;
式₂ if 条件₂;
...
式_n if 条件_n;
式₀ otherwise;



式には if で続く複数の行を含めることができます。最後の行には、直前のどの条件も適用されない場合のための Otherwise 文を含めることもできます。

Horizontal 式

The image displays three sequential screenshots of the GeneXus formula editor, each showing a different formula structure. Each screenshot has a title bar '式エディター' and buttons 'OK(O)' and 'キャンセル'.

1. The first screenshot shows a formula with an IF statement and a multiplication operation:

```
FlightPrice*(1-AirlineDiscountPercentage/100) IF AirlineDiscountPercentage>=FlightDiscountPercentage;  
FlightPrice*(1-FlightDiscountPercentage/100) OTHERWISE;
```

2. The second screenshot shows the same formula with a `Round` function applied to the result of the multiplication:

```
FlightPrice*(1-AirlineDiscountPercentage/100) IF AirlineDiscountPercentage>=FlightDiscountPercentage;  
Round(  
Round(Value: Numeric, Decimals: Numeric): Numeric/100) OTHERWISE;
```

3. The third screenshot shows the same formula with a `Month` function applied to the result of the multiplication:

```
FlightPrice*(1-AirlineDiscountPercentage/100) IF AirlineDiscountPercentage>=FlightDiscountPercentage;  
FlightPrice*(1-FlightDiscountPercentage/100) OTHERWISE;  
Month(  
Month(Value: Date/DateTime): Numeric
```

この例では、結果を取得する方法として計算を使用していますが、項目属性や計算に適用する関数、たとえば、四捨五入した結果を取得する `Round`、日付から月を取得する `Month` などを使用することもできます。ほかにもさまざまな選択肢があり、値を返すように定義したプロシーチャーを呼び出すこともできます (プロシーチャーについては後で説明します)。

その時点のレコード (1 つのみ) から取得した計算 (または、拡張テーブルを使用して関連するレコードのデータから取得した計算) を実行する式は、通常、Horizontal 式と呼ばれます。

次に、別のタイプのグローバル式について説明します。

Horizontal 式



Aggregate式



次に、別のタイプの **Aggregate 式**について説明します。

ここでは、グローバル式の例を定義して説明します。つまり、対応する計算は項目属性と関連付けて定義し、物理フィールドとしては作成されません。しかし、この説明は、変数や、1つのオブジェクトの特定のセクションでのローカルな定義など、別の文脈でも当てはまります。これについては別の章で説明します。

Aggregate 式: Count、Sum、Average など

例: Flight トランザクションに 2 つ目のレベルを作成する:

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...		
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...		
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	$\text{FlightPrice} * (1 - \text{AirlineDiscountPercentage} / 100)$ IF AirlineDisco...	
Seat	Seat	座席		
FlightSeatId	Id	座席番号		No
FlightSeatLocation	Location	座席位置		No

このドメインを列挙型の
値で定義

Flight トランザクションに 2 つ目のレベルを作成し (Ctrl + L キー)、Seat という名前にします。

レベル名が示すとおり、フライトで利用可能な座席の記録に使用し、各座席について、窓側、通路側、中央のいずれかを記録します。次に、すべての場合に、フライトで利用可能な座席数を取得します。

ピリオドを入力し、次の項目属性名となるように書き足します: FlightSeatId。

もう 1 つ FlightSeatLocation という名前の項目属性を作成します。これは 1 桁の Character です。そのため、Location ドメインを定義し、これに特定の値 (列挙値) を割り当てます。

Aggregate 式: Count、Sum、Average など

2) ドメインを列挙型の値で定義する:

The screenshot shows the GeneXus IDE interface. On the left, the 'Domain' editor is open, displaying a list of domains. The 'Location' domain is selected. On the right, the 'Properties' window for the 'Location' domain is open. The 'Enum Values' property is highlighted, showing the values 'Window, 窓, W; Middle, 中間, M; Aisle, 通路側, A'.

名前	タイプ	モジュール	説明
PlaybackState	Numeric(4,0)	GeneXus	Playback State
NetworkAPIConnection...	値エディター		
EventAction			
EventStatus			
EventData			
EventErrors			
ApplicationState			
SynchronizationReceive...			
RegionState			
BeaconProximity			
MediaFinishReason			
HttpMethod			
HttpAuthenticationType			
CommonCallTarget			
BarcodeType			
Id			
Name			
Page	Numeric(4,0)	Root Module	Page
TabCode	Character(50)	Root Module	Tab Code
Price	Numeric(9,2)	Root Module	Price
Percentage	Numeric(3,0)	Root Module	Percentage
Location	Character(1)	Root Module	Location

Domain: Location

Name	Location
Description	Location
Empty as null	Yes
Class	Attribute
Data Dictionary	(None)
Module	Root Module
Qualified Name	Location
Object Visibility	Public

Type Definition

Based on	(none)
Data Type	Character
Length	1
Enum Values	Window, 窓, W; Middle, 中間, M; Aisle, 通路側, A
Collection	False
Initial value	
Enable national language	Yes

Validation

次に、作成したドメインのプロパティである [Enum Values] プロパティを変更するために、ドメインを編集します。

ドメインの取る値を 3 つ定義します:

Window - 窓側が選択されたときに格納される値は文字「W」

Middle - 格納される値は「M」

Aisle - 格納される値は「A」

Aggregate 式: Count, Sum、Average など

The screenshot shows the GeneXus IDE interface for a form titled "Flight". The form is structured as follows:

- FormButtons:** 実行 (Execute), 終了 (End), 削除 (Delete)
- MainTable:**
 - フライト割引率 (FlightDiscountPercentage)
 - 航空会社番号 (AirlineId)
 - 航空会社名 (AirlineName)
 - 航空会社割引率 (AirlineDiscountPercentage)
 - フライト最終価格 (FlightFinalPrice)
- 座席 (Seats):**
 - GRID:**
 - 座席番号 (FlightSeatId)
 - 座席位置 (FlightSeatLocation) - This field is highlighted with a red oval.
- <FormButtons>**

このトランザクションのフォームを見てみましょう。フライトの座席を入力するグリッドが追加され、各座席について、コンボボックスから位置を指定できるようになっています。コンボボックスには [Window]、[Middle]、[Aisle] の値が含まれています。これらの値は、FlightSeatLocation 項目属性のドメインで定義したものです。

Aggregate 式: Count、Sum、Average など

3) 座席をより適切に表せるように Seat レベルのキーを変更する。数値と文字 (A ~ F) で識別する

名前	タイプ	デスクリプション
Flight	Flight	フライト
FlightId	Id	フライト番号
FlightDepartureAirportId	Id	出発空港番号
FlightDepartureAirportName	Name	出発空港名
FlightDepartureCountryId	Id	出発空港
FlightDepartureCountryName	Name	出発空港
FlightDepartureCityId	Id	出発空港
FlightDepartureCityName	Name	出発空港
FlightArrivalAirportId	Id	到着空港
FlightArrivalAirportName	Name	到着空港
FlightArrivalCountryId	Id	到着空港
FlightArrivalCountryName	Name	到着空港
FlightArrivalCityId	Id	到着空港
FlightArrivalCityName	Name	到着空港
FlightPrice	Price	フライト価格
FlightDiscountPercentage	Percentage	フライト割引
AirlineId	Id	航空会社
AirlineName	Name	航空会社
AirlineDiscountPercentage	Percentage	航空会社
FlightFinalPrice	Price	フライト最終価格
Seat	Seat	座席
FlightSeatId	Id	座席番号
FlightSeatChar	SeatChar	座席文字
FlightSeatLocation	Location	座席位置

名前	説明	値
A	A	A
B	B	B
C	C	C
D	D	D
E	E	E
F	F	F

Enum ドメイン

F5 キーを押す前に、2 つ目のレベルの定義内容を見てみましょう。

キーが FlightId と FlightSeatId で構成されていると、**フライトごとに座席番号を繰り返し使用できません**。しかし、座席はこの番号と文字の組み合わせで識別するため、繰り返し使用可能でなければいけません。この場合、座席は 1A、1B、1C、2A、2B のように表されます。

そこで、FlightSeatChar 項目属性を追加します。タイプは SeatChar ドメインとして定義した Character(1) とし、キーの項目属性部分は、異なる文字で同じ座席番号を記録できるようにします。

使用可能な文字を A から F に制限するようにドメインを編集します。

[Enum Values] プロパティを見つけて、使用できる値を定義します。この場合、デスクリプションの値は格納される値と同じです。

Aggregate 式: Count、Sum、Average など

4) 座席数をカウントする新しい式を追加する:

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	$\text{FlightPrice} * (1 - \text{AirlineDiscountPercentage})$	
FlightCapacity	Numeric(4.0)	Flight Capacity	$\text{count}(\text{FlightSeatLocation})$	
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

ここで、フライトに搭乗可能な乗客数を知るために、そのフライトで提供される座席数を保持する新しい項目属性を、1 つ目のレベルに式として定義します。

FlightCapacity 項目属性を Numeric(4.0) とし、[式] 列に次を記述します: Count。次に括弧の中で 2 つ目のレベルの項目属性を参照します。この項目属性を選択することで、GeneXus は、座席数をカウントすることを理解します。そのために、フライトの座席が格納された FlightSeat テーブルに属す、FlightSeatLocation 項目属性を選択します。

あるフライトが与えられた場合、通常は、そのフライトの座席数をカウントする必要があります。つまり、その Flight と関連付けられた FlightSeat テーブルのレコードをカウントする必要があります。

これはグローバル式のため、項目属性 FlightCapacity は Flight テーブルの中に物理的に作成せずに、常にフライトの座席数をカウントします。

Aggregate 式: Count、Sum、Average など

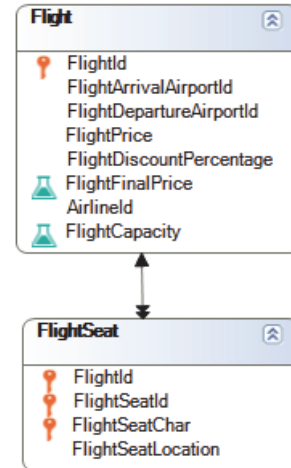
FlightCapacity = Count(FlightSeatLocation)

フライト
(Flight)

FlightId	FlightDepartureAirportId	...	FlightPrice	AirlineId
1	1	...	1500	1
2	3	...	2500	2
3	1	...	1000	1
...

座席
(FlightSeat)

FlightId	FlightSeatId	FlightSeatChar	FlightSeatLocation
1	1	A	窓側
1	1	B	通路側
1	2	A	窓側
1	2	B	通路側
1	3	C	中央
2	1	A	窓側
2	1	B	中央
3
...



この式を計算するために (全フライトの座席レコードを持ち、FlightSeat テーブルと 1 対多の関連がある) Flight テーブルが、1 レコードごとに処理されます。

Count (FlightSeatLocation) と記述すると、記録された座席が集計され、窓側や通路側といった観点では集計されません。特定の 1 レコードのフライトごとに処理されるので、GeneXus はただ、**そのフライト**に対応する座席を集計します。

グローバル式の関連付られたテーブルと、式が参照するテーブルとの関連を GeneXus が検出できた場合、GeneXus は**関連レコードのみ**を利用して計算を実行します。一方、関連を検出できない場合、GeneXus は**ナビゲートしたテーブルのすべてのレコード**を利用して計算を実行します。

これを実行時にテストするには、F5 を押します。

Aggregate 式: Count、Sum、Average など

データベースの再編成が必要です。

このレポートにはデータベースの変更内容と、再編成プログラムでどのように扱われるかが記載されています。
「再編成」を押して続けるか、「キャンセル」を押してキャンセルしてください。

再編成
キャンセル

フィルタ:

FlightSeat

Table FlightSeat specification

Table name: FlightSeat

FlightSeat is new

Warnings

▲ rgz0009AutoNumber=True ignored. Attribute FlightSeatId is not table FlightSeat's primary key

Table Structure

Attribute	Definition	Previous values
<u>FlightId</u>	Numeric (4) Not null	
<u>FlightSeatId</u>	Numeric (4) Not null	
<u>FlightSeatChar</u>	Character (1) Not null NLS	
<u>FlightSeatLocation</u>	Character (1) Not null NLS	

Indexes

Name	Definition	Compos
IFLIGHTSEAT	primary key Clustered	FlightId FlightSeatId FlightSeatChar

Foreign key constraints

Referenced table	Attributes
<u>Flight</u>	<u>FlightId</u>

Statements

```
CREATE TABLE [FlightSeat] (
```

エラー : 0
 警告 : 1
 成功 : 0

Flight トランザクションの第 2 レベルに関連した項目属性とキーを含む FlightSeat という物理テーブルが作成されます。項目属性 FlightCapacity は、想定どおり物理的には作成されないため、Flight テーブルの構造は変更されないことが分かります。

変更に同意する場合は、続行するために [再編成] をクリックします。Flight トランザクションを実行します。

Aggregate 式: Count、Sum、Average など

フライト最終価格	2100.00
Capacity	6

座席

	座席番号	座席文字	座席位置
×	1	A ▼	窓 ▼
×	1	B ▼	中間 ▼
×	1	C ▼	通路側 ▼
×	1	D ▼	窓 ▼
×	1	E ▼	中間 ▼
×	1	F ▼	通路側 ▼

[行追加]

実行

終了



フライトに座席を追加するたびに座席数が更新される点に注目してください。

Aggregate 式: Count、Sum、Average など

名前	タイプ	DESCRIPTION	式	Null 許容
Flight	Flight	フライト		
FlightId	Id	フライト番号		
FlightDepartureAirportId	Id	出発空港番号		
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		
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		
FlightDiscountPercentage	Percentage	フライト割引率		
AirlineId	Id	航空会社番号		
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		
FlightCapacity	Numeric(4,0)	フライト座席数	count(FlightSeatLocation)	
Seat	Seat	座席		
FlightSeatId	Id	座席番号		
FlightSeatChar	SeatChar	座席文字		
FlightSeatLocation	Location	座席位置		

Sum(att)

Average(att)

Max(...)

Min(...)

Find(...)

そのほかにも、**複数のレコードを考慮して演算を実行する** Aggregate 式があります。

たとえば、**Sum** は、指定された項目属性の値を合計します。**Average** は、指定された項目属性の値の平均値を算出します。このほかにも、Max、Min、Find もあります。

Aggregate 式: Count、Sum、Average など

条件を追加すると「特定の」行をカウントできる:

名前	タイプ	DESCRIPTION	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		

式エディター

```
count(FlightSeatLocation, FlightSeatLocation = Location.Window)
```

OK(0) キャンセル

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-AirlineDiscountPercent...	
FlightCapacity	Numeric(4,0)	フライト座席数	count(FlightSeatLocation)	
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

また、指定した明示的な条件を満たす特定のレコードのカウント、集計、または平均値算出を行うことができます。

たとえば、フライトの窓際の座席数をカウントする場合は、式の括弧内にカンマと対応する条件を追加します。

FlightSeatLocation 項目属性は Location ドメインに属し、3 つの列挙値が定義されているため、項目属性が使用する値を求める構文は次のようになります: **ドメイン名、ピリオド、フィルタ基準となる値に関連付けられた名前** (この場合は Window)。

Aggregate 式: Count、Sum、Average など

フライト最終価格

2100.00

フライト座席数

3 count(FlightSeatLocation, FlightSeatLocation = Location.Window)

フィルタ条件

座席

座席番号 座席文字 座席位置

×	1	A	窓
×	1	B	中間
×	1	C	通路側
×	1	D	窓
×	1	E	中間
×	1	F	通路側
×	2	A	窓
	0	A	窓

【行追加】

「トリガー条件」も指定可能



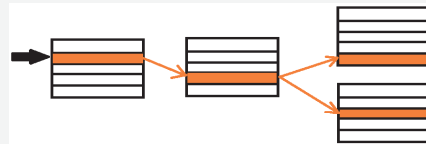
これまで説明したように、暗示的条件（関連レコードがある場合）に加え、**ユーザーが指定した明示的条件を満たすレコード**のカウント、集計、または平均値算出も行うことができます。この条件は通常、「フィルタ条件」と呼ばれます。これは、ユーザーが必要とするレコードのみを扱うことができるためです。

また、ほかのすべてのグローバル式と同様に、式に「トリガー条件」を指定することもできます。つまり、条件が満たされた場合にのみ、その式が計算されます。

まとめ

- Horizontal:

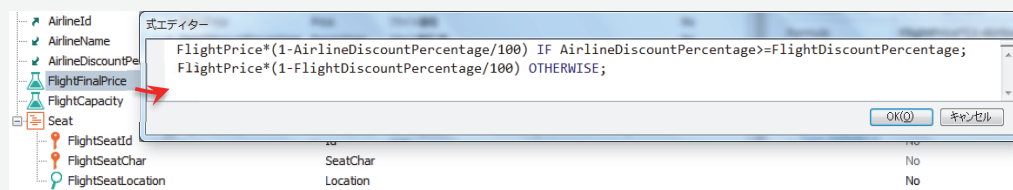
- 計算を行うためにレコードにアクセスし、場合によっては拡張テーブルを経由する



項目属性 =

式₁ if 条件₁;
 式₂ if 条件₂;
 ...
 式_n if 条件_n;
 式₀ otherwise;

- 例: FlightFinalPrice



FlightFinalPrice の計算では、Flight テーブルの項目属性と Airline テーブルの項目属性を使用し、対応するトリガー条件をそれぞれの計算に追加します。

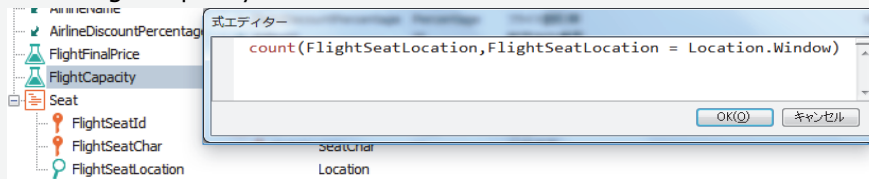
まとめ

- **Aggregate:**

- 計算を行うために、同じテーブル内の多数のレコードをナビゲートする必要がある

FlightId	FlightDepartureAirportId	...	FlightId	FlightSeatId	FlightSeatLocation
1	1	...	1	1	A 窓側
2	3	...	1	1	B 通路側
3	1	...	1	2	A 窓側
...	1	2	B 通路側
			1	3	C 中央
			2	1	A 窓側
			2	1	B 中央
			3
		

- 例: FlightCapacity



FlightCapacity 項目属性の計算では、式の項目属性と関連付けられた Flight テーブルから、count の引数で FlightSeatLocation 項目属性が格納されている FlightSeat テーブルに対して計算が実行されます。

この場合、1 対多の関係を持つため、関連レコードのみがカウントされます。この関連が存在しない場合は、すべてのレコードがカウントされます。また、カウントするレコードのフィルタ条件を指定しているため、関連レコードは、その条件を満たす場合にのみカウントされます。

まとめ

- **Aggregate:**
 - Sum
 - Average
 - Max
 - Min
 - Find
- 式の括弧内で参照される項目属性は、計算を実行するときに検索するテーブルに関する情報を GeneXus に提供する
- GeneXus では、検索対象テーブルと式の項目属性が記述されているテーブルの間に関連があると、関連レコードのみを考慮して計算を実行する
- 関連がない場合は、GeneXus は**検索対象テーブルのすべてのレコード**を考慮して操作を実行する

まとめ

- **Aggregate:**

$$fx$$

項目属性 = Count(項目属性, 条件, 既定値)
 Sum(式, 条件, 既定値)
 Find(式, 条件, 既定値)
 ...

- Horizontal 式を Aggregate 式と組み合わせることも可能 (このコースでは説明しません):

$$fx$$

項目属性 = 2 + Count(項目属性, 条件, 既定値) *
 Sum(式, 条件, 既定値) if 条件;
 Att₁ + Att₂ * Att₃ otherwise;

式の 2 つ目のパラメーターでフィルタ条件を指定し、3 つ目のパラメーターで既定値を指定できます。既定値は、カウントや集計などの対象となるレコードが見つからなかった場合にその式が返す値です。

Horizontal 式と同様に、Aggregate 式でもトリガー条件を指定できます。

このコースでは説明しませんが、Horizontal 式を Aggregate 式と組み合わせることもできます。組み合わせることで、高度な表現が可能になります。