

Compound 式

GeneXus[™]

Compound 式について見てみましょう。

Compound 式

項目属性 = <エクスプレッション₁> if <条件₁>;
 <エクスプレッション₂> if <条件₂>;
 ...
 <エクスプレッション_n> if <条件_n>;
 <エクスプレッション_o> otherwise;

```
Occupancy.Low IF count(FlightSeatLocation) < 5;  
Occupancy.Medium IF count(FlightSeatLocation) >5 and count(FlightSeatLocation) < 8;  
Occupancy.High OTHERWISE
```

条件：有効な論理エクスプレッション
 - 拡張テーブルに属する項目属性
 - 定数、関数、論理演算子 (and、or、not) および
 関係演算子 (>、>=、<、<=、=、<>、like)

項目属性 = **Count**(<項目属性>, <条件>, <既定値>) if <条件>;

Sum(<エクスプレッション>, <条件>, <既定値>) if <条件>;

Find(<エクスプレッション>, <条件>, <既定値>) if <条件>;

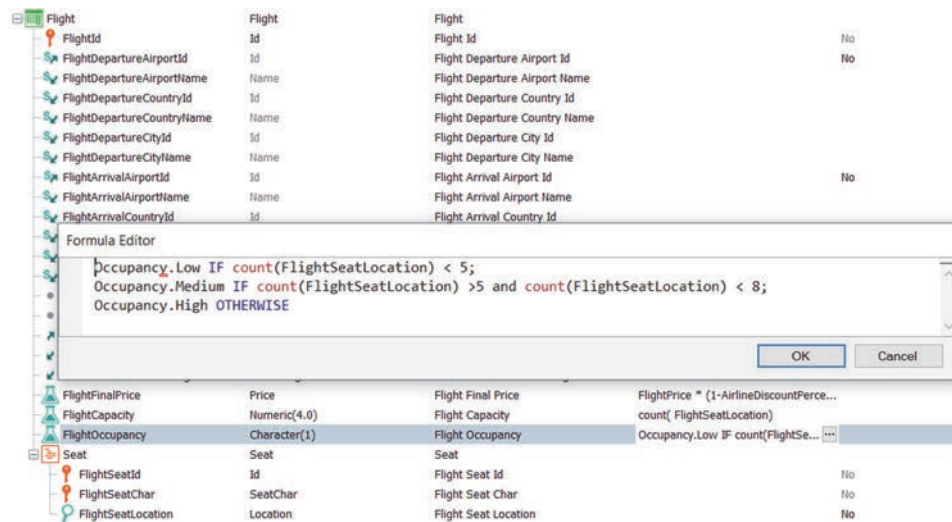
Compound 式にはいくつかの条件付きの Aggregate 式が含まれます。Horizontal エクスプレッションを含めることもできます。

この場合、各エクスプレッションを Aggregate 式または Horizontal 式にすることができます。含まれるすべてのエクスプレッションが Horizontal 式の場合、定義された式は Compound 式ではなく Horizontal 式になります。

条件とは任意の有効な論理エクスプレッションです。式として定義されている項目属性に関連付けられたテーブルの拡張テーブルに属する項目属性、定数、関数、論理演算子 (and、or、not) および関係演算子 (>、>=、<、<=、=、<>、like) を含めることができます。最初に True と評価された条件が、その条件の左側にある式の結果となります (他の条件は評価されません)。

評価された条件のいずれも True でないときに、otherwise 節のある**エクスプレッション**がある場合、この節の前にあるエクスプレッションの結果が式の結果になります。

例



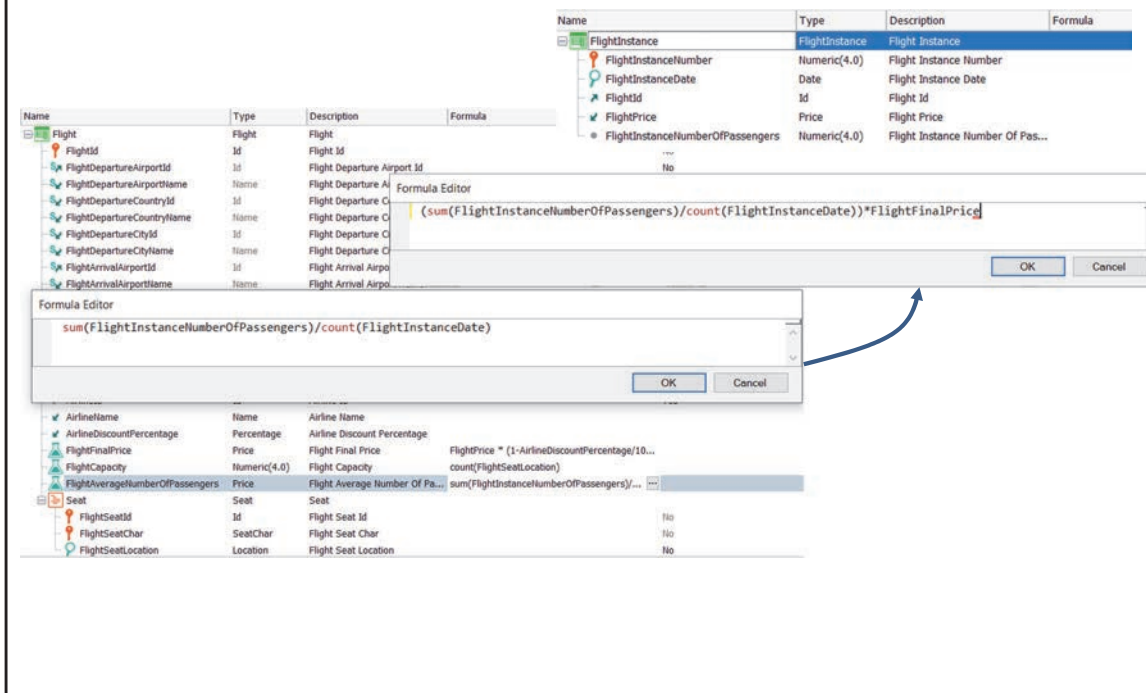
旅行代理店を例に、このようなタイプの Compound 式について見てみましょう。

Horizontal エクスプレッションに基づいて FlightOccupancy 項目属性が定義されています。これにより、フライトの座席数 (Aggregate 式の Count で計算) に応じて、Occupancy ドメインの対応する値 (Low、Medium または High) が割り当てられます。

特に、このケースの場合、Aggregate 式を FlightCapacity 項目属性で置き換えることもできますが、この定義のままでも有効です。

実装は Horizontal 式の構造になっており、Aggregate 式はトリガー条件に含まれていました。

Compound 式のもう 1 つの例



この例では、特定のフライトで旅行する乗客の平均数を計算します。

Flight トランザクションでは一般的な方法でフライトを定義しましたが、FlightInstance トランザクションでは、特定のフライトの実際のインスタンスを、日付、フライト番号、乗客数などによりモデル化します。

各フライトを利用した乗客の平均数を計算するには、そのフライトのすべてのインスタンスの乗客の合計数を加算し、これをフライトインスタンスの数で割る必要があります。

Flight トランザクションの FlightAverageNumberOfPassengers 項目属性を、Aggregate 式の Sum の商として計算するグローバル式として定義します。そして FlightInstanceNumberofPassengers 項目属性を加算し、結果をフライトのインスタンス数で割ります。インスタンス数は、FlightInstanceDate 項目属性を使用してインスタンスをカウントする Aggregate 式の Count として計算します。

Flight トランザクションで式がグローバルとして定義されているため、コンテキストは式の項目属性に関連付けられたテーブル(つまり FLIGHT テーブル)になります。したがって、結果は、対象の特定のフライトの複数のインスタンスの乗客の平均数になります。

また、Aggregate 式の Average があるため、これでこの計算を実行することもできます。しかし、式を構成して Compound 式を作成できることを示すためにこの方法を用いました。

計算の構成を続けることもできます。たとえば、フライトあたりの平均収益を知りたい場合は、乗客の平均数にフライトの最終料金 (FlightFinalPrice 項目属性) を掛けます。

この項目属性は Horizontal 式であるため、既に説明したように、GeneXus で簡単に複雑な計算を実行できます。

その他の Compound 式の例

```
項目属性 = Max(...) if <条件1>;
           (2 * attrx) + 100 if <条件2>;
           Sum(attry) otherwise
```

```
項目属性 = Find(...) if <条件1>;
           1 otherwise
```

```
項目属性 = procedure(...) if <条件1>;
           Min(...) if <条件2>;
           10 if <条件3>
```

```
項目属性 = 2 + Count(<項目属性>, <条件>, <既定値>) *
           Sum(<エクスプレッション>, <条件>, <既定値>) if <条件>;
           Attr1 + Attr2 * Attr3 otherwise;
```

Compound 式を使用すると、柔軟に計算を定義し、さまざまな状況をモデル化することができます。

この章では、式を使用してコードを保存することや、宣言型プログラミングのシンプルさを活用することがいかに便利かを説明しました。