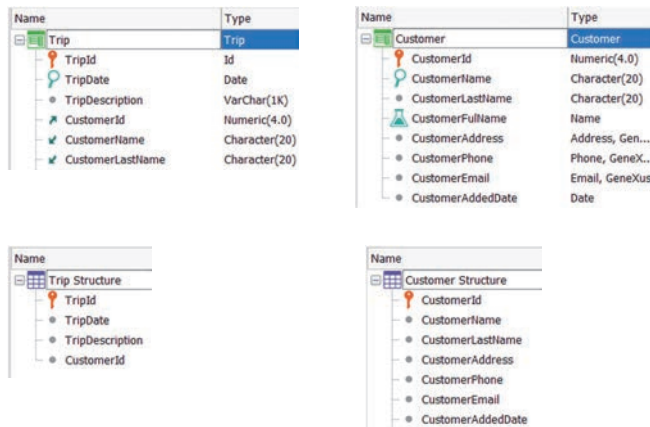


冗長項目属性とその維持

GeneXus[™]

この章では、定義上格納されない推論された項目属性や式を冗長として定義し、データベーステーブルの項目属性となる方法を説明します。



ご存じのように、GeneXus はデータベースを第 3 正規形で自動的に正規化します。つまり、複数のテーブルに存在できる項目属性は、外部キーとして機能する主キーである項目属性のみです。

従属項目属性と呼ばれるその他の項目属性は、1 つのテーブルに格納されます。これらの項目属性を作成元以外のトランザクションに追加すると、GeneXus はこれらを推論します。つまり、外部キーを介して、値が格納されているテーブルから値を取得します。

さらに、項目属性をトランザクションの式として定義すると、項目属性は格納されず、仮想の項目属性になります。

ただし、パフォーマンス上の理由により、推論された項目属性を、項目属性が推論されるトランザクションと関連付けられたテーブルに格納したい場合があります。また、多数の計算を実行する必要があり、値を取得するたびに時間がかかる式の項目属性も、関連付けられたテーブルに格納して迅速に値が取得できるようにしたいことがあります。

GeneXus では、既定でテーブルに格納できない項目属性を冗長と定義することで、格納可能にすることができます。

参照の冗長性

The screenshot displays the GeneXus IDE interface. On the left, the 'Attraction' and 'Country' table structures are shown. The 'Attraction' table includes attributes like AttractionId, AttractionName, CountryId, CountryName, CategoryId, CategoryName, AttractionPhoto, CityId, CityName, and AttractionAddress. The 'Country' table includes CountryId, CountryName, and City. A subroutine named 'Source' is shown with the following code:

```

1 For each Attraction
2   order CountryName
3   where CountryName = "France"
4   print Attractions
5 Endfor

```

The 'Indexes' window is open, showing a list of indexes for the 'Attraction' table. The indexes are as follows:

Attribute	Order	Description
Attraction Indexes		Attraction
IAAttraction	Primary Key	Automatic Index
IAAttraction1	Ascending	Attraction Id
IAAttraction2	Foreign Key	Automatic Index
IAAttraction3	Ascending	Country Id
IAAttraction4	Ascending	City Id
IAAttraction5	Foreign Key	Automatic Index
IAAttraction6	Ascending	Category Id
IAAttraction7	Unique	User Index
IAAttraction8	Ascending	Attraction Name
IAAttraction9	Duplicate	User Index
IAAttraction10	Ascending	CountryName

A red error message is displayed below the 'CountryName' index: 'Attribute CountryName does not exist in table structure'.

推論された項目属性を冗長として定義することを参照の冗長性と呼びます。

その目的はパフォーマンスの向上です。たとえば、多くのレコードが関連する場合に、別のテーブルからの推論された値の検索を最小限にしたい場合や、推論された項目属性をユーザーインデックスに含めて、テーブルに項目属性を含むインデックスを定義したい場合などです。

式の冗長性

Name	Type	Description	Formula	Nullable
FlightInstance	FlightInstance	Flight Instance		
FlightInstanceNumber	Id	Flight Instance Number		No
FlightInstanceDate	Date	Flight Instance Date		No
FlightId	Id	Flight Id		No
FlightPrice	Price	Flight Price		
FlightInstanceNumberOfPassengers	Numeric(4,0)	Flight Instance Number Of Pass...		No
FlightInstanceFinalPrice	Price	Flight Instance Final Price	FlightPrice*0.8 IF FlightInstanceNumberOff...	

Formula Editor

```

FlightPrice*0.8 IF FlightInstanceNumberOfPassengers>200;
FlightPrice*0.9 IF FlightInstanceNumberOfPassengers>=100 and FlightInstanceNumberOfPassengers<=200;
FlightPrice OTHERWISE

```

OK Cancel

Name	Type	Description	Formula
Invoice	Invoice	Invoice	
InvoiceId	Id	Invoice Id	
InvoiceDate	Date	Invoice Date	
CustomerId	Numeric(4,0)	Customer Id	
CustomerName	Character(20)	Customer Name	
FlightInstance	FlightInstance	Flight Instance	
FlightInstanceNumber	Id	Flight Instance Number	
FlightInstanceDate	Date	Flight Instance Date	
FlightInstanceFinalPrice	Price	Flight Instance Final Price	FlightPrice*0.8 IF FlightInstanceNumberOfPasse...
InvoiceAmount	Amount	Invoice Amount	sum(FlightInstanceFinalPrice)

式の冗長性も、特に式の値を取得するために多数の計算を実行しなければならない場合などのパフォーマンスの向上が目的です。

これは、特に多くのレコードから値を取得する Aggregate 式の場合に有効です。

そのため、Horizontal 式の場合は冗長を定義してもあまり意味がないと思われるかもしれませんが、場合によって定義する必要があります。

たとえば、式の項目属性 InvoiceAmount を冗長として定義したい場合、まず、Horizontal 式である項目属性 FlightInstanceFinalPrice を冗長として定義する必要があります。

項目属性を冗長として定義する方法

The screenshot illustrates the steps to define a redundant attribute in GeneXus. The main window shows the 'Flight' entity with its attributes. The 'FlightCapacity' attribute is selected. A context menu is open, and the 'Column Chooser' option is selected. The 'Column Chooser' dialog is open, showing a list of attributes. The 'Redundant' option is selected in the dialog.

Name	Type	Description	Formula
Flight	Flight	Flight	
FlightId	Id	Flight Id	
FlightDepartureAirportId	Id	Flight Departure Airport Id	
FlightDepartureAirportName	Name	Flight Departure Airport Name	
FlightDepartureCountryId	Id	Flight Departure Country Id	
FlightDepartureCountryName	Name	Flight Departure Country Name	
FlightDepartureCityId	Id	Flight Departure City Id	
FlightDepartureCityName	Name	Flight Departure City Name	
FlightArrivalAirportId	Id	Flight Arrival Airport Id	
FlightArrivalAirportName	Name	Flight Arrival Airport Name	
FlightArrivalCountryId	Id	Flight Arrival Country Id	
FlightArrivalCountryName	Name	Flight Arrival Country Name	
FlightArrivalCityId	Id	Flight Arrival City Id	
FlightArrivalCityName	Name	Flight Arrival City Name	
FlightPrice	Price	Flight Price	
FlightDiscountPercentage	Percentage	Flight Discount Percentage	
AirlineId	Id	Airline Id	
AirlineName	Name	Airline Name	
AirlineDiscountPercentage	Percentage	Airline Discount Percentage	
FlightFinalPrice	Price	Flight Final Price	FlightPrice * (1-AirlineDiscountPercentage/100) ...
FlightCapacity	Numeric(4,0)	Flight Capacity	count(FlightSeat.location)
Seat	Seat	Seat	
FlightSeatId	Id	Flight Seat Id	
FlightSeatChar	SeatChar	Flight Seat Char	
FlightSeatLocation	Location	Flight Seat Location	

項目属性を冗長として定義するには、トランザクション構造に移動します。列のバーを右クリックして [表示列の選択] をクリックし、[冗長] を選択して列のバーにドラッグして追加します。

Name	Type	Description	Redundant	Formula
Flight	Flight	Flight		
FlightId	Id	Flight Id		
FlightDepartureAirportId	Id	Flight Departure Airport Id		
FlightDepartureAirportName	Name	Flight Departure Airport Name	<input type="checkbox"/>	
FlightDepartureCountryId	Id	Flight Departure Country Id	<input type="checkbox"/>	
FlightDepartureCountryName	Name	Flight Departure Country Na...	<input type="checkbox"/>	
FlightDepartureCityId	Id	Flight Departure City Id	<input type="checkbox"/>	
FlightDepartureCityName	Name	Flight Departure City Name	<input type="checkbox"/>	
FlightArrivalAirportId	Id	Flight Arrival Airport Id		
FlightArrivalAirportName	Name	Flight Arrival Airport Name	<input type="checkbox"/>	
FlightArrivalCountryId	Id	Flight Arrival Country Id	<input type="checkbox"/>	
FlightArrivalCountryName	Name	Flight Arrival Country Name	<input type="checkbox"/>	
FlightArrivalCityId	Id	Flight Arrival City Id	<input type="checkbox"/>	
FlightArrivalCityName	Name	Flight Arrival City Name	<input type="checkbox"/>	
FlightPrice	Price	Flight Price		
FlightDiscountPercentage	Percentage	Flight Discount Percentage		
AirlineId	Id	Airline Id		
AirlineName	Name	Airline Name	<input type="checkbox"/>	
AirlineDiscountPercentage	Percentage	Airline Discount Percentage	<input type="checkbox"/>	
FlightFinalPrice	Price	Flight Final Price	<input type="checkbox"/>	FlightPrice * (1-AirlineDiscountPercentage/1...
FlightCapacity	Numeric(4,0)	Flight Capacity	<input checked="" type="checkbox"/>	count(FlightSeatLocation)
Seat	Seat	Seat		
FlightSeatId	Id	Flight Seat Id		
FlightSeatChar	SeatChar	Flight Seat Char		
FlightSeatLocation	Location	Flight Seat Location		

[冗長] 列が追加されていることに注目してください。このチェックボックスを選択することで、項目属性を冗長として定義できます。これを行うと、式の記号に冗長であることを示す「+」記号が表示されます。

冗長性の維持



項目属性を冗長として定義すると、GeneXus によりプロシージャが作成され、式が属するレコードを編集すると、格納された値が常に更新されます。

トランザクション画面またはトランザクションのビジネスコンポーネントが実行されることにより、冗長式の項目属性の値が変更されると、式がトリガーされて計算が実行され、その結果がデータベースの物理フィールドに格納されます。

冗長式の項目属性に関連するトランザクションに対応する生成されたプログラムには、再計算されるたびに冗長データを格納するルーチンが組み込まれます。

冗長式の項目属性の値のクエリを実行すると、計算を取得するための式はトリガーされず、代わりにデータベースのフィールドに格納された値が取得されます。

式の計算に属する一部の項目属性の値が変更されると、GeneXus は、冗長項目属性の Update プロシージャもトリガーします。これらのプロシージャは新しい値を計算することを認識しており、結果はデータベースに格納されます。

冗長として定義する際の制限事項

- サブタイプは冗長として定義**できません**。
- 式である項目属性を冗長として定義するには、まず、その式を冗長として定義する必要があります。
- 冗長な式の定義を変更するには、まず、冗長の設定を削除する必要があります。
- 複数レベルの冗長でない式を加算する式は、正しく維持されません。

項目属性を冗長として定義する際には、いくつか考慮すべき制限事項があります。まず、サブタイプを冗長として定義することはできません。

先ほど見たように、式である項目属性を冗長として定義するには、まず、その式を冗長として定義する必要があります。

冗長式の定義を変更するには、まず冗長の設定を削除してから変更を行い、再度、式を冗長として定義します。

冗長式である項目属性に従って、冗長として定義された Aggregation 式の場合、複数のレベルがネストされていると、冗長性が正しく維持されません。

式の冗長性に関する制限の例

Name	Type	Description	Formula
Customer	Customer	Customer	
CustomerId	Numeric(4,0)	Customer Id	
CustomerName	Character(20)	Customer Name	
CustomerLastName	Character(20)	Customer Last Name	
CustomerFullName	Name	Customer Full Name	CustomerName+' '+CustomerLastName
CustomerAddress	Address, GeneXus	Customer Address	
CustomerPhone	Phone, GeneXus	Customer Phone	
CustomerEmail	Email, GeneXus	Customer Email	
CustomerAddedDate	Date	Customer Added Date	
CustomerTotalPurchases	Amount	Customer Total Purchases	sum(InvoiceAmount)

Name	Type	Description	Formula
Invoice	Invoice	Invoice	
InvoiceId	Id	Invoice Id	
InvoiceDate	Date	Invoice Date	
CustomerId	Numeric(4,0)	Customer Id	
CustomerName	Character(20)	Customer Name	
FlightInstance	FlightInstance	Flight Instance	
FlightInstanceNumber	Id	Flight Instance Number	
FlightInstanceDate	Date	Flight Instance Date	
FlightInstanceFinalPrice	Price	Flight Instance Final Price	FlightPrice*0.8 IF FlightInstanceNumberOfPasse...
InvoiceAmount	Amount	Invoice Amount	sum(FlightInstanceFinalPrice)

Aggregate 式がネストされている場合の制限の例を見てみましょう。

ここで、CustomerTotalPurchases 項目属性は、InvoiceAmount 項目属性の Sum 式の項目属性です。また、InvoiceAmount 項目属性も FlightInstanceFinalPrice 項目属性 (Horizontal 式) の Sum 式の項目属性です。

CustomerTotalPurchases、InvoiceAmount および FlightInstanceFinalPrice 項目属性が冗長式として定義されている場合、InvoiceAmount および FlightInstanceFinalPrice 項目属性のみが初期化されて正しく更新されますが、CustomerTotalPurchases 項目属性は更新されません。

冗長性を維持するために作成されるプロシージャ

GeneXus では次のプロシージャが自動的に作成されます。

- **TableNameUpdateRedundancy**: 推論された項目属性が冗長となるテーブルで、その冗長性が維持されます。
- **TableNameLoadRedundancy**: TableName で、式および推論された項目属性の両方の冗長性が再計算されて更新されます。

冗長性を維持するために GeneXus で使用されるプログラムは、冗長として定義された項目属性を維持する再編成で作成されます。次のような名前になります。

- TableNameUpdateRedundancy
- TableNameLoadRedundancy

TableNameUpdateRedundancy プログラムを使用して、推論された項目属性の冗長性を更新します。TableName は従属項目属性が格納されるテーブルの名前です。

テーブル (このテーブルの従属項目属性が他のテーブルで冗長として定義されている) を定義するトランザクションは、そのテーブルの主キーをパラメーターとして渡し、TableNameUpdateRedundancy プログラムを呼び出します。

TableNameLoadRedundancy プログラムは TableName テーブルの冗長性を再計算します。TableName は冗長項目属性が格納されるテーブルです。

推論された冗長項目属性の場合、このプロシージャは従属項目属性が格納されるテーブルにアクセスし、その値で冗長項目属性が推論されるテーブルを更新します。

冗長式の項目属性の場合、プロシージャは式に関連する項目属性を含むテーブルにアクセスし、計算を実行して、冗長式の項目属性が定義されたテーブルのその項目属性を更新します。

例を見てみましょう。

従属項目属性の冗長性

Name	Type	Description	Redundant	Formula
Trip	Trip	Trip		
TripId	Id	Trip Id		
TripDate	Date	Trip Date		
TripDescription	VarChar(1K)	Trip Description		
CustomerId	Numeric(4,0)	Customer Id		
CustomerName	Character(20)	Customer Name	<input checked="" type="checkbox"/>	
CustomerLastName	Character(20)	Customer Last Name	<input checked="" type="checkbox"/>	

Name	Type
Customer	Customer
CustomerId	Numeric(4,0)
CustomerName	Character(20)
CustomerLastName	Character(20)
CustomerFullName	Name
CustomerAddress	Address, Gen...
CustomerPhone	Phone, GeneX...
CustomerEmail	Email, GeneXus
CustomerAddedDate	Date

Trip トランザクションで、推論された項目属性 CustomerName と CustomerLastName を冗長にするとします。

Customer トランザクションで項目属性 CustomerName または CustomerLastName が変更されるたびに、CustomerUpdateRedundancy プロシージャが自動的に呼び出され、これらの値が最新に保たれます。

これを影響分析で見てみましょう。

従属項目属性の冗長性 (続き)

Database needs to be reorganized.

This report describes Database changes and how they will be handled by reorganization programs.
Please select Reorganize to proceed or Cancel.

Pattern:

☒ Trip
☒ Customer
☒ Trip

Table Trip specification

Table name: [Trip](#)

Trip needs conversion

Table Structure

	Attribute	Definition	Previous values	Takes value from
	TripId	Numeric (4), Not null, Autonumber		Trip TripId
	TripDate	Date, Not null		Trip TripDate
	TripDescription	Varchar (1024), Not null, NLS		Trip TripDescription
	CustomerId	Numeric (4), Not null		Trip CustomerId
New	CustomerName	Character (20), Not null, NLS		Customer CustomerName
New	CustomerLastName	Character (20), Not null, NLS		Customer CustomerLastName

まず、Trip テーブルを再編成する必要があります。推論され、格納された項目属性 CustomerName と CustomerLastName が作成されるためです。

さらに、2つのプロシージャがあることが確認できます。1つは Customer、もう1つは Trip という名前です。

従属項目属性の冗長性 (続き)

The screenshot displays two procedure configurations in the GeneXus IDE. On the left, the 'Table Customer update redundancy procedure' is shown. It has a 'Pattern' of 'Trip' and 'Customer'. The 'Redundant attributes to update' are 'CustomerName' and 'CustomerLastName'. The 'From attributes to update' are 'CustomerName' and 'CustomerLastName'. The 'Procedure Name' is 'CustomerUpdateRedundancy'. The procedure is structured as 'For First Customer (Line: 1)' with an 'Order' of 'CustomerId', 'Index: ICUSTOMER', and 'Navigation filters: Start from: CustomerId = @CustomerId, Loop while: CustomerId = @CustomerId'. It then enters a loop 'For Each Trip (Line: 4)' with an 'Order' of 'CustomerId', 'Index: ITRIP', and 'Navigation filters: Start from: CustomerId = @CustomerId, Loop while: CustomerId = @CustomerId'. The loop contains an 'UPDATE Trip (CustomerId, CustomerLastName)' statement. On the right, the 'Table Trip load redundancy procedure' is shown. It has a 'Pattern' of 'Trip' and 'Customer'. The 'Redundant attributes' are 'CustomerName' and 'CustomerLastName'. The 'Procedure Name' is 'TripLoadRedundancy'. The procedure is structured as 'For Each Trip (Line: 2)' with an 'Order' of 'TripId', 'Index: ITRIP', and 'Navigation filters: Start from: FirstRecord, Loop while: NotEndOfTable'. It then enters a loop 'For Each Customer (Line: 4)' with an 'Order' of 'CustomerId', 'Index: ICUSTOMER', and 'Navigation filters: Start from: CustomerId = @CustomerId, Loop while: CustomerId = @CustomerId'. The loop contains an 'UPDATE Trip (CustomerLastName, CustomerName)' statement.

Customer というプロシージャーをクリックすると、実際の名前は CustomerUpdateRedundancy であることが分かります。

これは CustomerId に基づいてフィルタされる Customer テーブルにアクセスし (For First Customer)、次に CustomerId 外部キーの値でフィルタされる Trip テーブルにアクセスして、それに対して UPDATE を実行します。

このプロシージャーは、Customer テーブルの CustomerName および CustomerLastName 項目属性が変更されたときにトリガーされ、Trip で冗長になっている CustomerName と CustomerLastName の値を更新します。

Trip というプロシージャーを選択すると、名前が TripLoadRedundancy であることが分かります。

このプロシージャーは Trip テーブルの CustomerName および CustomerLastName 項目属性の更新をトリガーします。

式項目属性の冗長性

Name	Type	Description	Redundant	Formula
FlightInstance	FlightInstance	Flight Instance		
FlightInstanceNumber	Id	Flight Instance Number		
FlightInstanceDate	Date	Flight Instance Date		
FlightId	Id	Flight Id		
FlightPrice	Price	Flight Price		
FlightInstanceNumberOfPassengers	Numeric(4,0)	Flight Instance Number Of P...		
FlightInstanceFinalPrice	Price	Flight Instance Final Price	<input checked="" type="checkbox"/>	FlightPrice*6.8 IF FlightInstanceNumberOf...

Pattern:

FlightInstance

Flight

FlightInstance

Table Flight update redundancy procedure

Redundant attributes to update: FlightInstanceFinalPrice

From attributes to update: FlightPrice

Procedure Name: FlightUpdateRedundancy

For First Flight (Line: 1)

Order: FlightId
Index: IFLIGHT
Navigation filters: Start from: FlightId = @FlightId
Loop while: FlightId = @FlightId

Flight (FlightId)

For Each FlightInstance (Line: 3)

Order: FlightId
Index: IFLIGHTINSTANCE1
Navigation filters: Start from: FlightId = @FlightId
Loop while: FlightId = @FlightId

FlightInstance (FlightInstanceNumber)

UPDATE FlightInstance (FlightInstanceFinalPrice)

Pattern:

FlightInstance

Flight

FlightInstance

Table FlightInstance load redundancy procedure

Redundant attributes: FlightInstanceFinalPrice

Procedure Name: FlightInstanceLoadRedundancy

For Each FlightInstance (Line: 2)

Order: FlightInstanceNumber
Index: IFLIGHTINSTANCE
Navigation filters: Start from: FirstRecord
Loop while: NotEndOfTable
Join location: Server

FlightInstance (FlightInstanceNumber)

Flight (FlightId)

UPDATE FlightInstance (FlightInstanceFinalPrice)

この例では、式項目属性 FlightInstanceFinalPrice を冗長として定義します。ご覧のように、FlightInstanceFinalPrice 項目属性が格納された項目属性として作成される FlightInstance テーブルの再編成に加えて、Flight および FlightInstance というプロシージャが表示されています。

Flight プロシージャの影響分析を見ると、名前が FlightUpdateRedundancy であることが分かります。これにより、式を統合する FlightPrice 項目属性の値から、冗長項目属性 FlightInstanceFinalPrice を更新します。

そのために、主キー FlightId によりフィルタされる Flight テーブルにアクセスして、外部キー FlightId によりフィルタされる FlightInstance テーブルに対して For each を実行します。そして UPDATE を実行して FlightInstanceFinalPrice 項目属性を更新します。

FlightInstance をクリックすると、それが FlightInstance テーブルを参照し、Flight テーブルにアクセスして式を再計算し、FlightInstance テーブルに格納された FlightInstanceFinalPrice 値を更新していることが分かります。

冗長性のリビルド

プログラム: gxlred

Name	Type	Description	Redundant	Formula
Customer	Customer	Customer		
CustomerId	Numeric(4,0)	Customer Id		
CustomerName	Character(20)	Customer Name		
CustomerLastName	Character(20)	Customer Last Name		
CustomerFullName	Name	Customer Full Name	<input checked="" type="checkbox"/>	CustomerName+" "+CustomerLastName
CustomerAddress	Address, Gen...	Customer Address		
CustomerPhone	Phone, Gene...	Customer Phone		
CustomerEmail	Email, GeneX...	Customer Email		
CustomerAddedDate	Date	Customer Added Date		
CustomerTotalPurchases	Amount	Customer Total Purchases		

CustomerFullName 項目属性は自動的に更新されないため、gxlred を呼び出して更新する必要があります。

Source * Layout Rules Conditions Variables Help

Subroutines

```

1 New
2   CustomerName = 'Anna'
3   CustomerLastName = 'Morgan'
4 EndNew
5
6 New
7   CustomerName = 'Peter'
8   CustomerLastName = 'Smith'
9 EndNew
10

```

```

1 Event 'Update redundant attributes'
2   call("gxlred")
3   Commit
4 EndEvent

```

GeneXus では冗長性が自動的に維持されますが、場合によっては明示的に更新する必要があります。

たとえば、New を使用してプロシーチャーから更新したい冗長項目属性がある場合などです。

この例では、式の項目属性 CustomerFullName は、冗長として定義されており、項目属性 CustomerName と CustomerLastName の連結から値を取得します。プロシーチャーのソースでは、並行した New 節により Customer テーブルに 2 つのレコードが作成されています。

この場合、Customer テーブルで冗長として定義された CustomerFullName 項目属性の値は更新されません。

このような場合のために、ナレッジベースで定義された一部の冗長性を更新できる [冗長性をリビルド] ユーティリティが用意されています。

このユーティリティを実行するには、gxlred プログラムを呼び出す必要があります。これにより <テーブル名> loadredundancy という名前のすべての冗長性プログラムが呼び出されます (たとえば、CustomerLoadRedundancy など)。

Call メソッドを使用して、このプログラムをイベントから呼び出してコミットを実行することもできます。

冗長項目属性と null

GeneXus では冗長項目属性が null になるかどうか自動的に管理されます。

- 推論された冗長項目属性:
 - 元の項目属性が null の場合、冗長項目属性も null になります。
 - 元の項目属性が Null 許容 = No の場合、外部キーが null の場合のみ冗長項目属性が null になります。
- 冗長式の項目属性

冗長項目属性の詳細については、<http://wiki.genexus.jp/hwikibypageid.aspx?6661> を参照してください。

考慮すべきもう 1 つの点は、冗長項目属性への Null 許容の割り当てです。

開発者は、冗長項目属性の Null 許容を管理できません。次の基準に従って、GeneXus が自動的に計算します。

推論された項目属性は、冗長性の元の項目属性が Null 許容の場合、冗長項目属性も Null 許容になります。

元の項目属性の Null 許容が No に設定されている場合、冗長項目属性が Null 許容になるかどうかは対応する外部キーの Null 許容の設定によります。対応する外部キーが null の場合は、冗長項目属性も null になります。それ以外の場合は、冗長項目属性は null になりません。

冗長として定義されたグローバル式は null 値を受け入れません。

このトピックの詳細と、推論された項目属性や式の項目属性を冗長として定義するための一般的な情報については、GeneXus の wiki を参照してください。
<http://wiki.genexus.jp/hwikibypageid.aspx?6661>