

GeneXus™

One Day トレーニング

ハンズオンでの GeneXus 開発の概要説明

Copyright © GeneXus S.A. 1988-2023.

All rights reserved. 本書は、GeneXus S.A.の明示的許可なしにはいかなるメディアにも複写することはできません。本書の内容は個人的使用のみを目的として提供するものです。

登録商標:

GeneXus は GeneXus S.A.の商標または登録商標です。本書において取り上げているその他の商標はすべて、それぞれの所有財産です。

目次

序論	3
「GENEXUS」について簡単におさらいしてみましょう	3
GENEXUS の開始	4
新しい WEB アプリケーションの作成	5
TRANSACTION の作成	6
アプリケーションのはじめての生成と実行	12
マスタの切り出し	15
アプリケーションの拡張	25
ビジネスルールの追加	28
計算の定義: 式	250
WEB パネル(一覧表示)	283
PDF 出力	38
バッチ処理(PRC)	45
パターン、WWP の概要(特徴)説明	49
WWP の適用	50
その他製品/特長紹介(GXSERVER、GXTEST、SD 生成、GAM)	61

序論

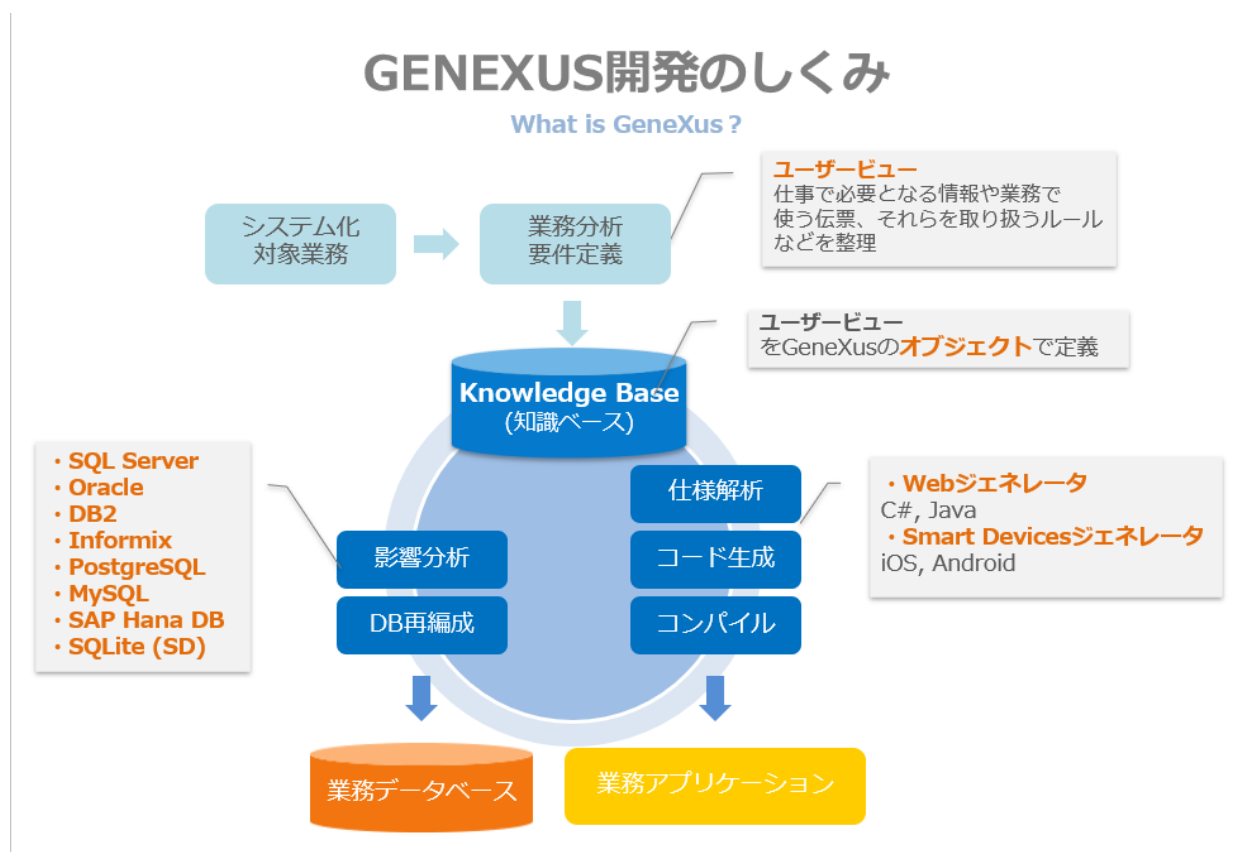
GeneXus は、複数の言語で、異なるプラットフォーム用のソフトウェアアプリケーションを迅速に生成できるローコード開発スイートです。GeneXus には次のような特長があります。習得が簡単で、生産性が高く、クロスプラットフォームで、将来にわたって使用可能であるため、デジタル資産を守ることができ、新しい技術も容易に取り入れることができます。

たとえば、GeneXus では、開発者が選択したターゲットプラットフォーム（特定の言語、データベース、環境、およびプラットフォーム、さらにレスポンシブ Web デザインの有無など）に対応した、Web 用 / モバイルおよびスマートデバイス用のアプリケーションを生成できます。

本書では、GeneXus によるアプリケーション開発がどのようなものか、学習・体験する事ができます。

「GENEXUS」について簡単におさらいしてみましょう

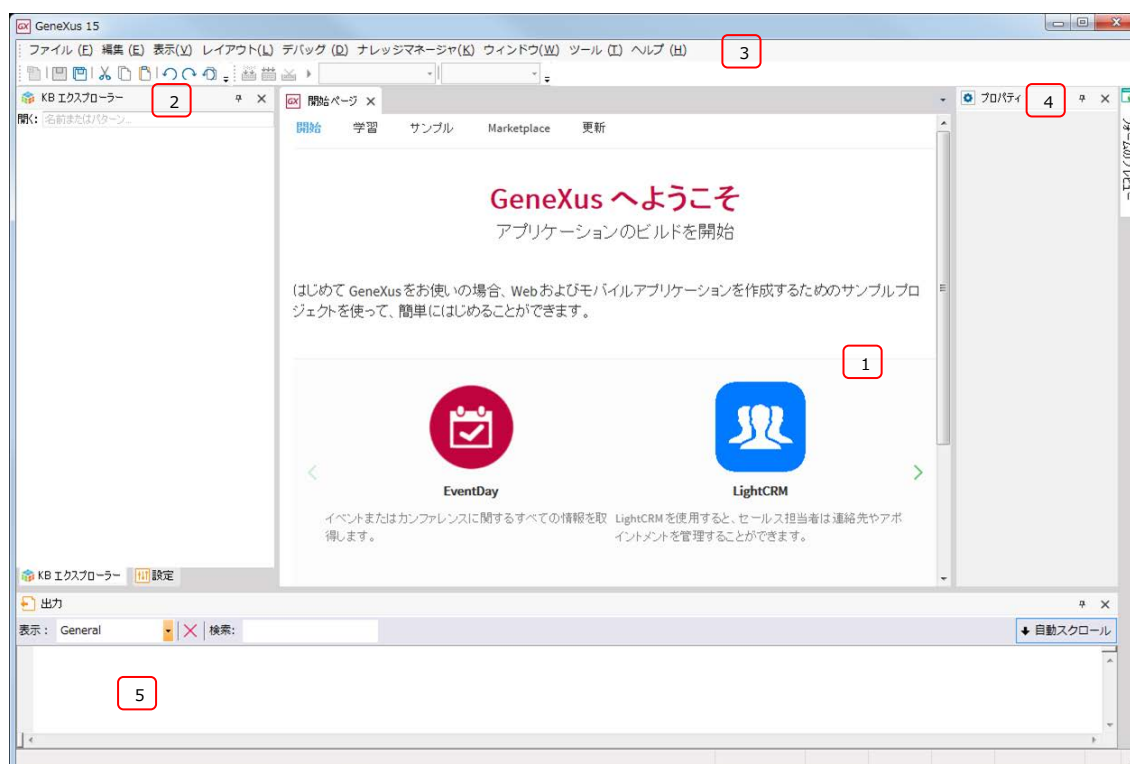
GeneXus の開発のしくみにつきまして、次の画像を参考におさらいしてみましょう。



GENEXUS の開始

GeneXus を起動すると、IDE (統合開発環境) と呼ばれる次のようなインターフェースが表示されます。このインターフェースは使い方が簡単で、各開発者が独自に設定できるようになっています。

このインターフェースは複数のウィンドウで構成されています：



1. **メインウィンドウ (開始ページ):** ツールに関する技術情報や GeneXus コミュニティ (ニュースや、ほかの開発者が投稿したソリューション) が動的に表示されます。また、最近使用されたプロジェクトも表示されるため、ここから開くことができます。新しいプロジェクトを作成することも可能です。
2. **ナレッジ ベース ナビゲーター:** 現在開いているプロジェクトのオブジェクトと設定が表示されます。
3. **ツールバー:** GeneXus でよく使用される機能の使いやすいインターフェースが表示されます。
4. **[プロパティ] ウィンドウ:** 開発者の現在のコンテキスト (選択されているオブジェクト、項目属性、変数、コントロールなど) に関するプロパティが表示されます。
5. **出力:** 実行されたアクションの出力が表示されます。

新しい WEB アプリケーションの作成

GeneXus では、「ナレッジベース」という単位でプロジェクトを作成します。

ここではこの「ナレッジベース」の新規作成を行います。

1. ツールバーより **[ファイル]** → **[新規]** → **[ナレッジベース]** を選択し、下図ダイアログを表示

新規ナレッジベース

名前(N): PharmacySystem

場所(L): C:\Models

基本 詳細

プロトタイプターゲット(I) ローカル ユーザーインターフェースの言語(U) Japanese

バックエンド

プロトタイプ環境(E) C# .NET Framework

データソース(D) SQL Server

フロントエンド

☒ C# Web (.NET Framework)

☐ Android

☐ Apple

☐ Web (Angular)

作成 キャンセル

2. 「名前」に「PharmacySystem」と入力
3. 「プロトタイプ環境」で「.Net Framework」を選択
4. 作成ボタンをクリック

TRANSACTION の作成

新しいナレッジベースを作成したら、次にユーザーの業務内容を記述します。このためには、現実の（実際の業務に関連する）オブジェクトを特定し（商品、請求書、顧客など、ユーザーが説明時に口にした名詞に注目することをお勧めします）、それらを GeneXus **オブジェクト** で定義する必要があります。

GeneXus の開発者は、テーブル定義、正規化、プログラム設計、プログラミングなどの低レベルの作業は行いません。代わりに、ユーザーの現実のオブジェクトを記述するという高レベルの作業を行います。その後、GeneXus が、**完全に自動化された方法**で、定義されたオブジェクトを分析し、選択されたプラットフォームに対応したデータベースとアプリケーションプログラムを設計します。

たとえば、アプリケーションを依頼しているドラッグストアが、販売中の商品を記録できるようにしたいと言っているとします。

特定されたそれぞれの現実のオブジェクトを記述するには、Transaction タイプの GeneXus オブジェクトを作成する必要があります（データベースのトランザクションとは無関係です）。それでは、商品を記述するトランザクションオブジェクトの作成方法について説明します。

1. ツールバーより[ファイル] → [新規] → [オブジェクト] を選択し、下図ダイアログを表示

新規オブジェクト

カテゴリを選択(S):

- データ管理
- ユーザーインターフェイス
- BPM
- Chatbot
- リソース
- ドキュメンテーション
- 拡張性
- デプロイ
- テスト
- レポーティング
- すべて

タイプを選択(T):

- API
- Data Provider
- Data Selector
- Data View
- Domain
- Procedure
- Structured Data Type
- Subtype Group
- Transaction

データベース構造、ビジネスルール、およびデータ操作の UI を定義し、実際のオブジェクトやアクターを記述します。

名前(N):

説明(D):

フォルダ(F):

作成 キャンセル

2. 「データ管理」→「**Transaction**」を選択
3. 「名前」に「**Product**」と入力
4. 「デスクリプション」に「**商品**」と入力
5. 作成ボタンをクリック

これで **Product** トランザクションが作成され、構造を定義できるように Product トランザクションが開きます



6. 初期表示される「**Structure**」タブの「名前」列に「**ProductCode**」と入力

(キーボードのドットキーを押すと、項目属性名の接頭語としてトランザクション名が表示されます)

項目属性を定義するときは、キーボードのドットキーを押します。すると自動的に、接頭語としてトランザクション名が入力されるので、続けて項目属性名の後の部分のみ入力します。

接頭語としてトランザクション名を使うという項目属性の名称設定方法は、項目属性の定義を簡便にするだけではありません。どこにあっても、記述されているオブジェクトが分かるため、項目属性名を見たときに簡単に理解できるようにするための GeneXus コミュニティの規則でもあります。



7. 「タイプ」列に「Numeric(10)」と入力

8. 「デスクリプション」に「商品コード」と入力

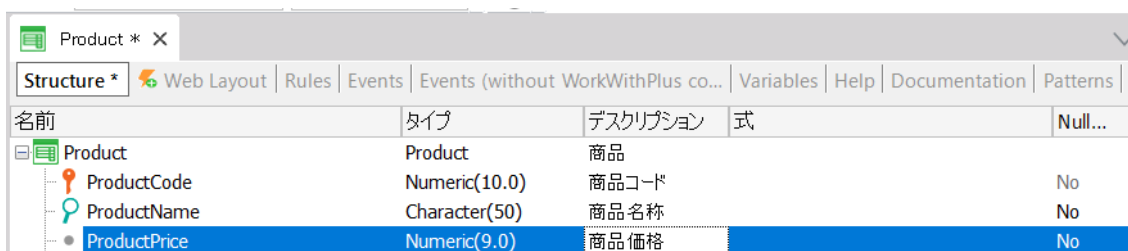


9. 次の行に「**ProductName**」、「Character(50)」、「商品名称」と入力

(Enter キーを押すと、2 つ目の項目属性を定義できる新しい行が開きます)

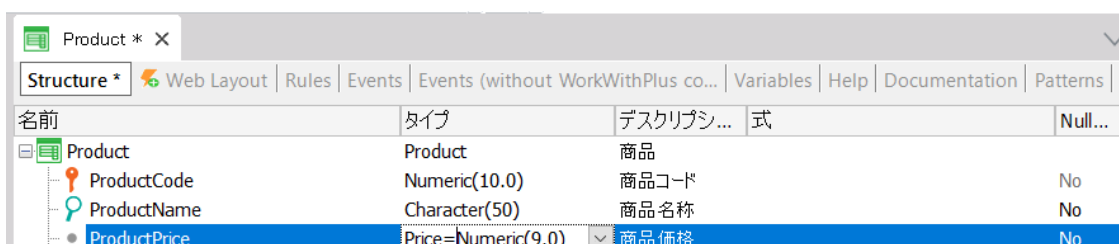


10. 次の行に、「**ProductPrice**」、「Numeric(9)」、「商品価格」と入力

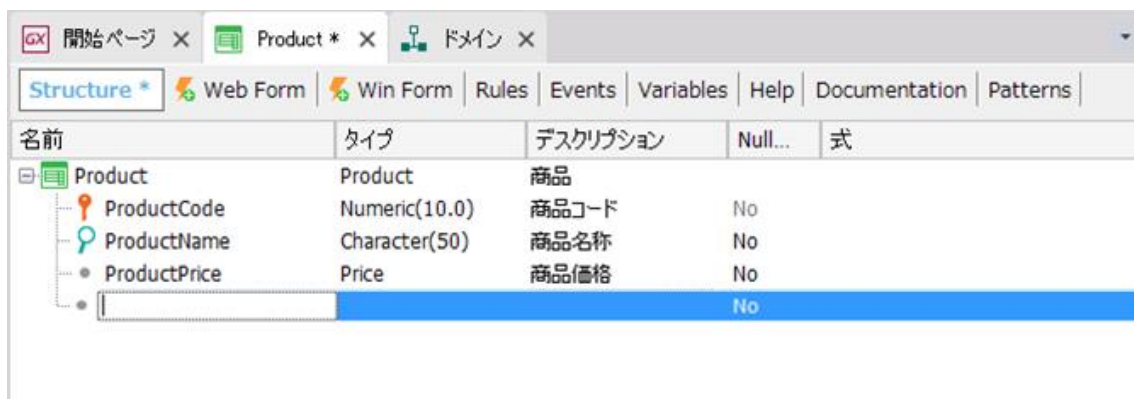


価格や金額を定義する項目属性は、いくつか作成するよう依頼がある可能性があります (ドラッグストアでの商品の購入や販売のため)。したがって、すべての価格で使用できる汎用定義タイプを作成しておくのも一案です。

11. ProductPrice 列のタイプ「Numeric(9.0)」の前に「Price=」と入力



1 2. Enter キーを押す



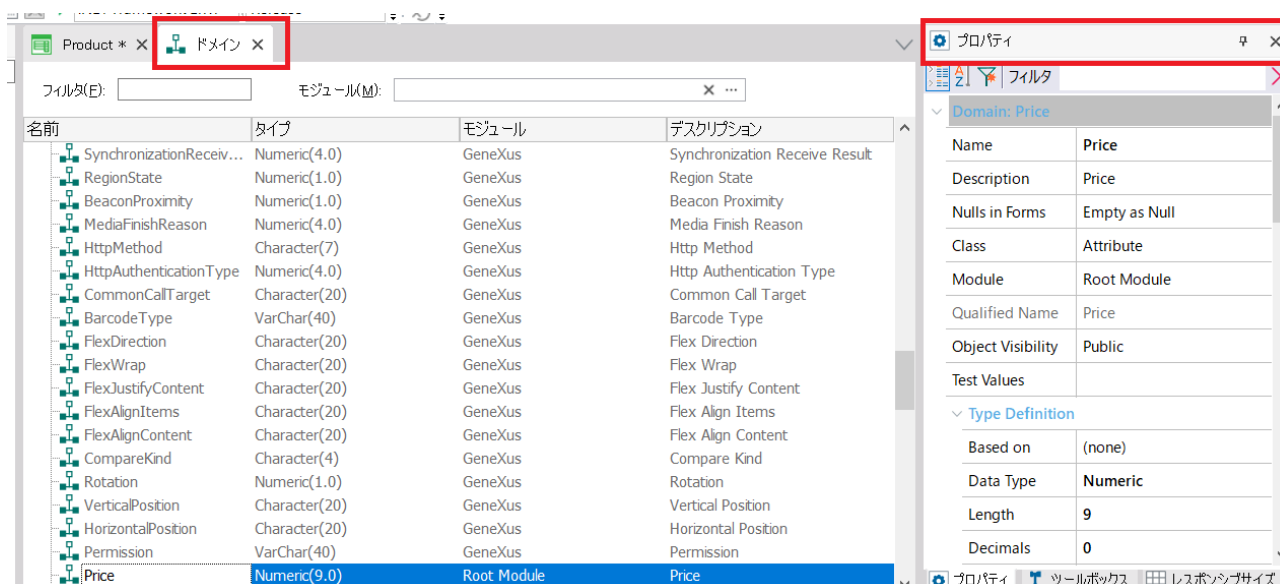
ProductPrice 項目属性が **Price** タイプで設定されたことが確認できます。

この Numeric タイプ (整数部 9 桁) の **Price** 定義を**ドメイン**と呼びます。

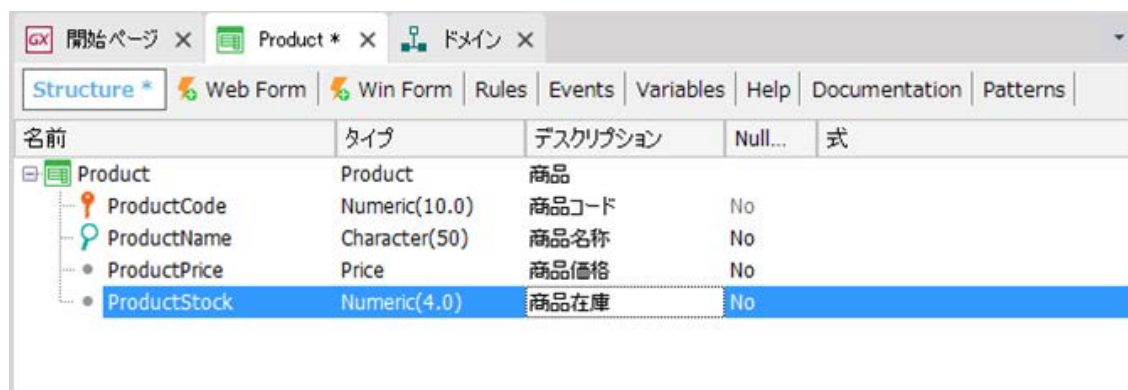
ドメインにより汎用定義が可能になります。ドメインの利点の 1 つは、後になって価格を異なる長さの Numeric にする必要が出てきた場合、ドメイン定義を変更するだけで、そのドメインを使用しているすべての項目属性を更新できるという点です。

1 3. ツールバーで [表示] → [ドメイン] を選択

これでナレッジベースの作成済みドメインを表示できます。:

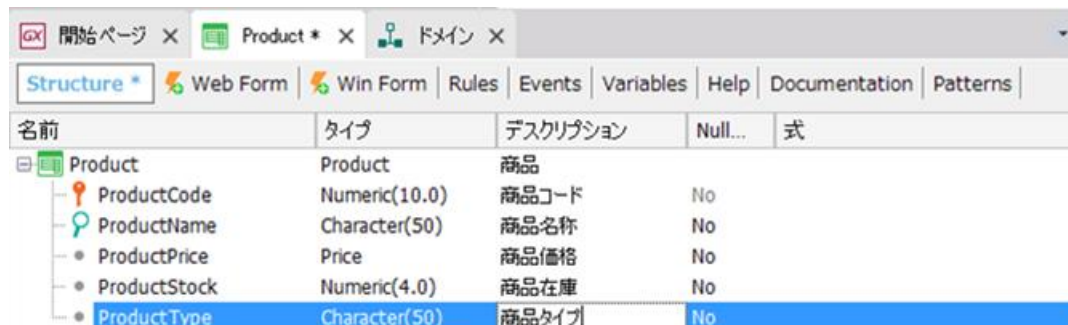


1 4. **Product** トランザクションに戻り、「**ProductStock**」、「Numeric(4)」、「商品在庫」と入力



名前	タイプ	デスクリプション	Null...	式
Product	Product	商品		
ProductCode	Numeric(10.0)	商品コード	No	
ProductName	Character(50)	商品名称	No	
ProductPrice	Price	商品価格	No	
ProductStock	Numeric(4.0)	商品在庫	No	

1 5. 「**ProductType**」、「Character(50)」、「商品タイプ」と入力



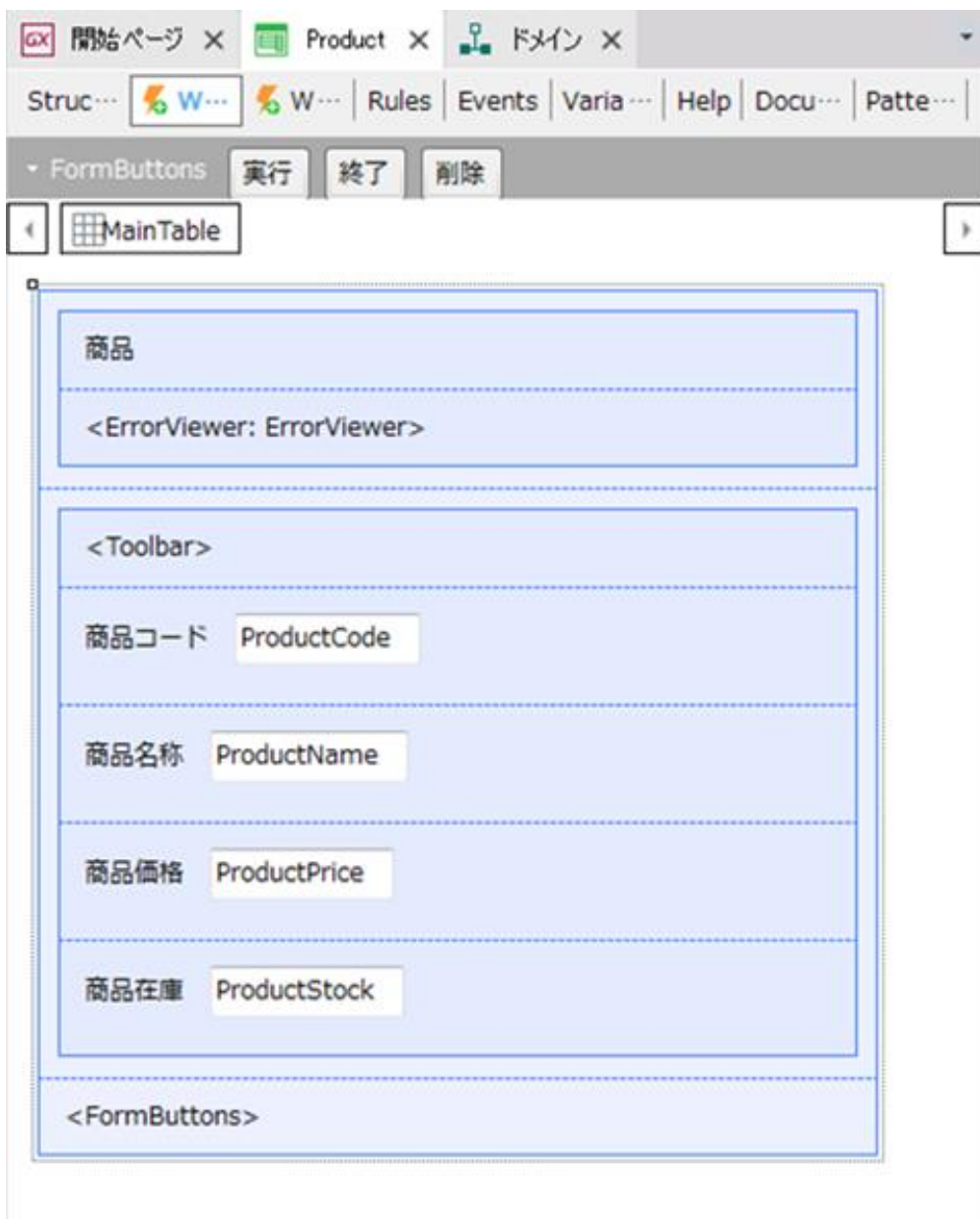
名前	タイプ	デスクリプション	Null...	式
Product	Product	商品		
ProductCode	Numeric(10.0)	商品コード	No	
ProductName	Character(50)	商品名称	No	
ProductPrice	Price	商品価格	No	
ProductStock	Numeric(4.0)	商品在庫	No	
ProductType	Character(50)	商品タイプ	No	

トランザクションの定義はひとまずここまでとして定義された構造を見ていきましょう。

16. トランザクションの「Web Layout」タブをクリック

定義された構造に従って、GeneXus が自動的に Web フォームを設計します。

ユーザーは、実行時にこのフォームで商品の追加、更新、削除を行うことができます：



アプリケーションのはじめでの生成と実行

アプリケーションをはじめて生成し、実行する場合は、次の手順が必要となります。

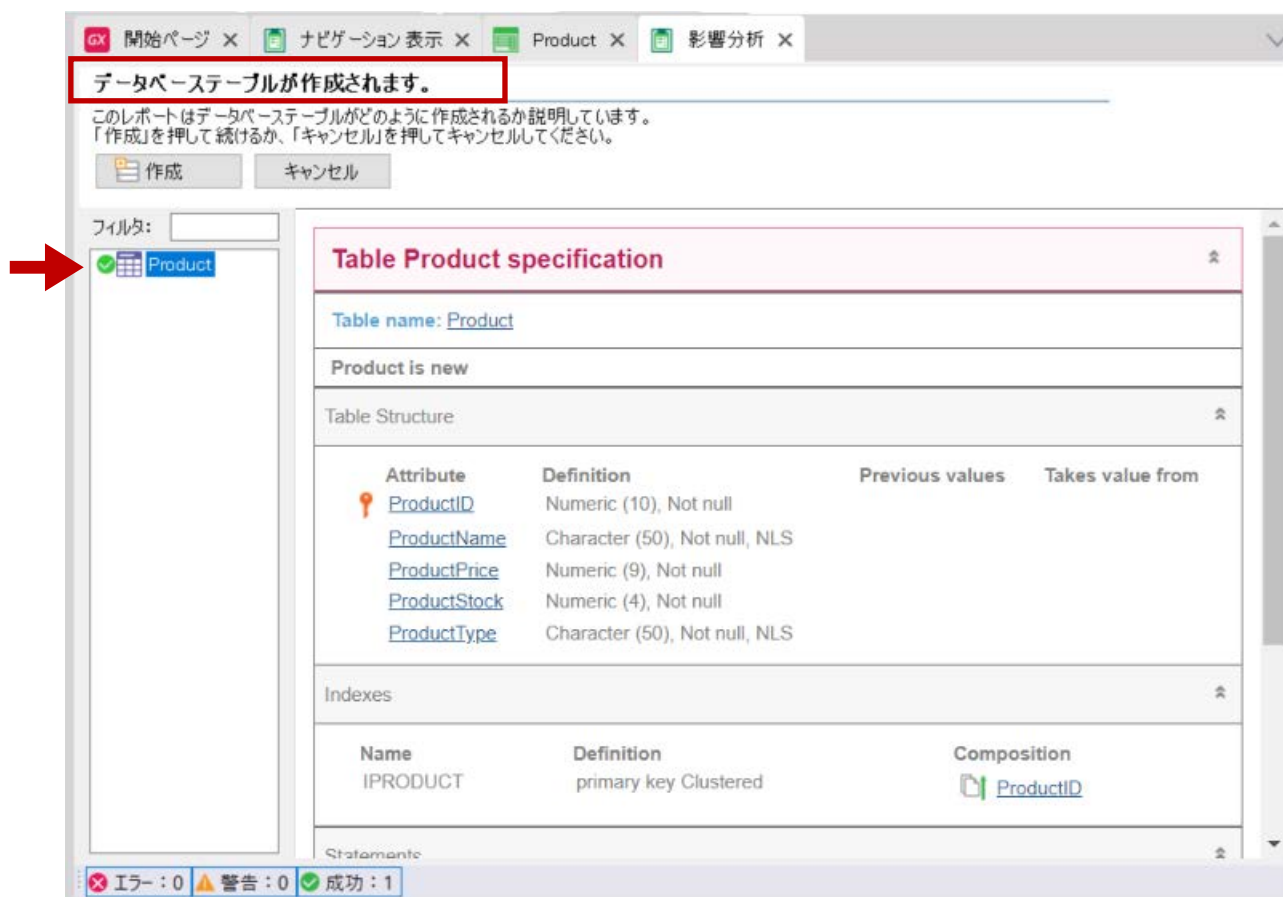
1. F5 キーを押し、下図ダイアログを表示

2. 「Database name」、「Server name」に有効な名前を入力

ここでは「PharmacySystem」、「localhost¥sqlexpress」と入力

3. [完了]ボタンをクリック

ナレッジベース内の新しい定義によって生じる影響が評価され、**[影響分析]** にレポートが表示されます：



このレポートには、データベースで行う必要がある構造変更が表示されます。

このレポートのメインタイトルから、今回は「データベーステーブルが作成される」ことが分かります。

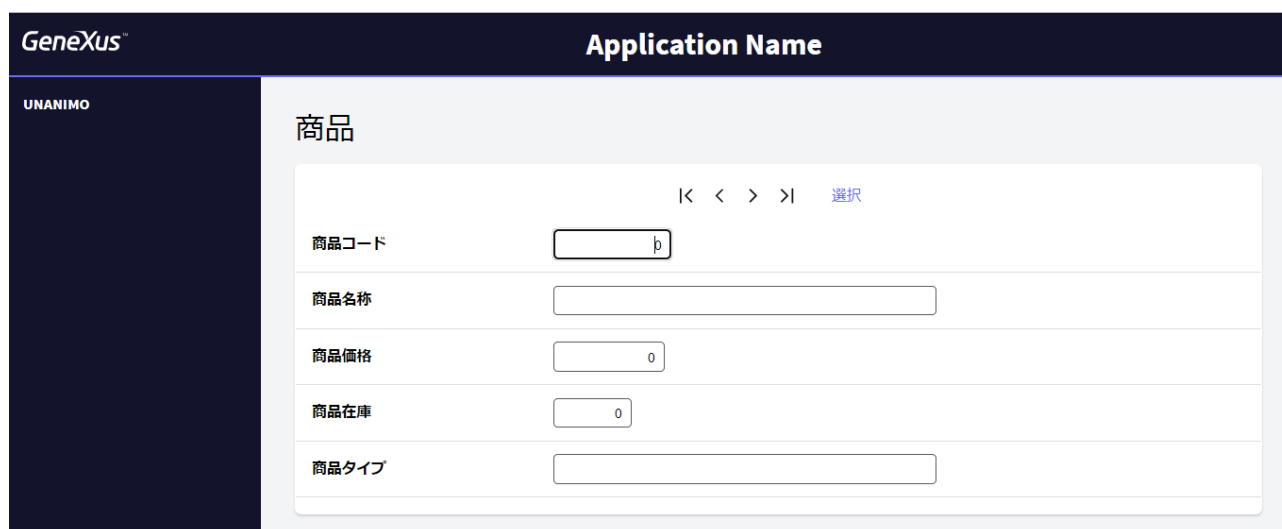
テーブル（左側のウィンドウの **Product**）をクリックすると、右側のウィンドウに、そのテーブルに含まれる項目属性が表示されます。

4. [作成] ボタンをクリック

結果（成功したか、エラーや警告があるか）が表示されます。続けてアプリケーションを実行します：

5. [Launch Pad]内の Product をクリック

Launch Pad とは、定義したオブジェクトを簡単に実行できるシンプルなページです。



こちらの画面にてデータの追加/更新/削除が行えます。

マスタの切り出し

上記までの手順にて簡単な Transaction の作成ができました。

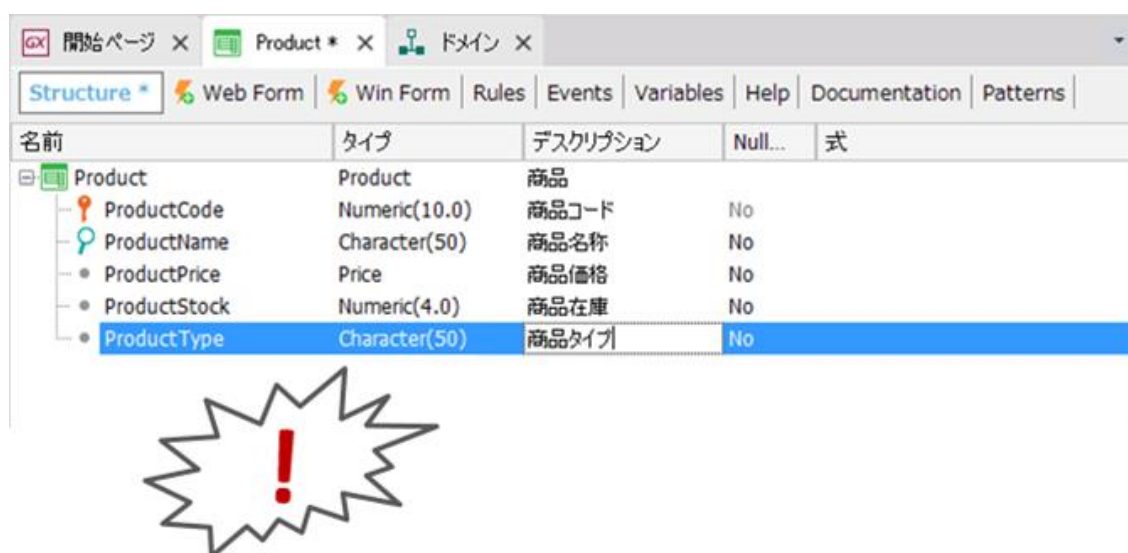
さて、このトランザクションにユーザーが同じタイプの 2 つの商品を入力したい場合はどうなるでしょうか。

ユーザーは同じタイプ名を 2 回入力することになります。

しかも、まったく同じになるように注意しながら入力しなければなりません。

また、後になって、特定のタイプの商品をすべて検索する必要があるかもしれません。

そのときも、まったく同じになるようにタイプ名を入力しなければなりません。

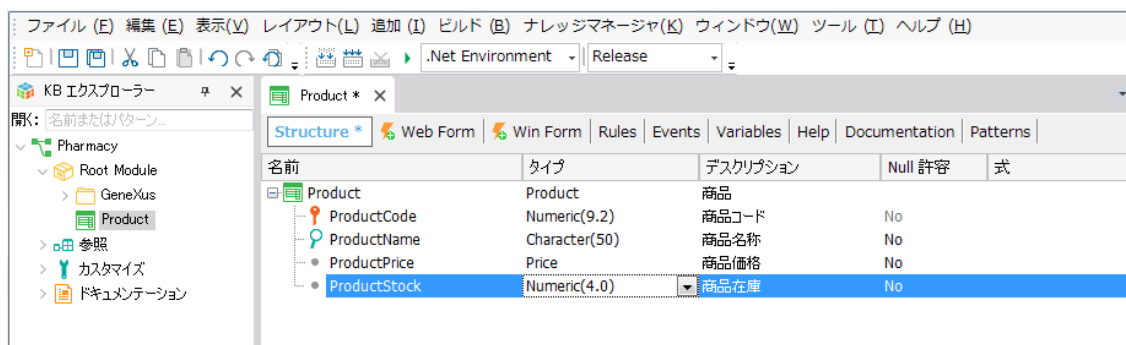


タイプは 1 か所に 1 回だけ入力し、商品ごとに、対応する商品タイプマスタを参照するようにしたいと顧客からリクエストがあったと仮定します。

次の手順にてこうした要望を実装できます。

方法としては、商品タイプを記録する別のマスタ用トランザクションを作成します。それから、各商品に商品タイプを割り当てます：

1. Product トランザクションの Structure から ProductType 項目属性を削除し、保存します

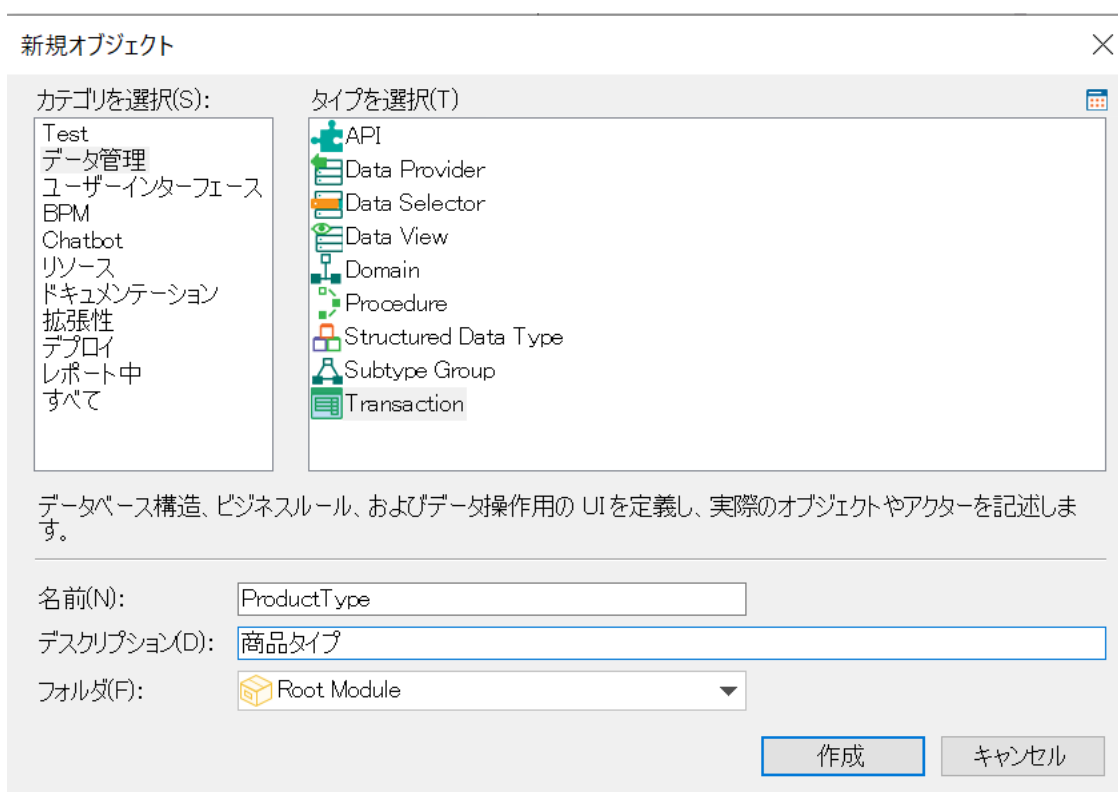


2. ツールバーより[ファイル] → [新規] → [オブジェクト] を選択し、「新規オブジェクト」ダイアログを表示

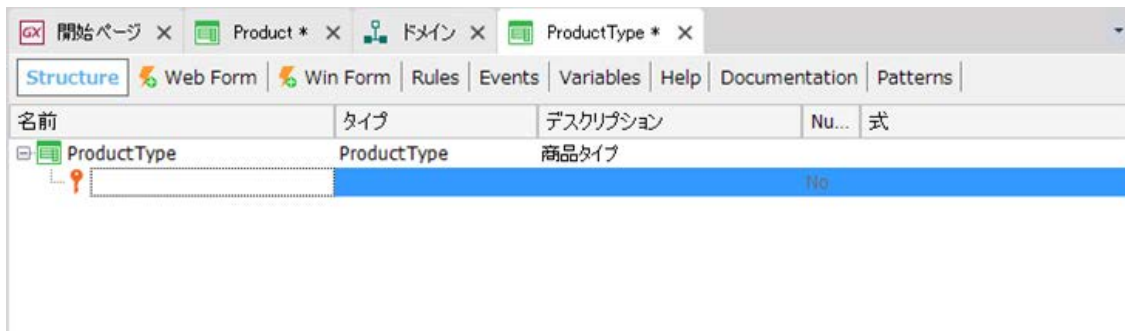
3. 「データ管理」→「Transaction」を選択

4. 「名前」に「ProductType」、「デスクリプション」に「商品タイプ」と入力

5. 作成ボタンをクリック



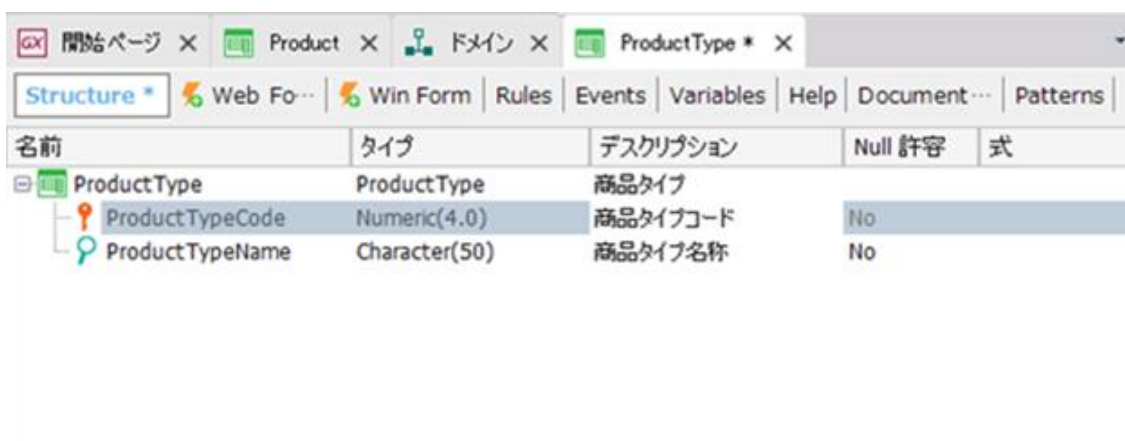
これで「ProductType」トランザクションが作成されます。



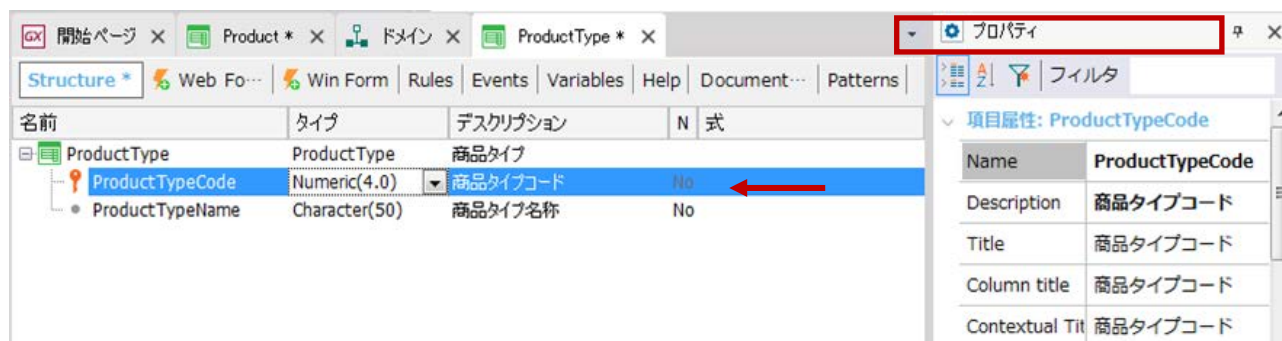
6. Structure に次の項目属性を入力

ProductTypeCode	Numeric(4.0)	商品タイプコード
ProductTypeName	Character(50)	商品タイプ名称

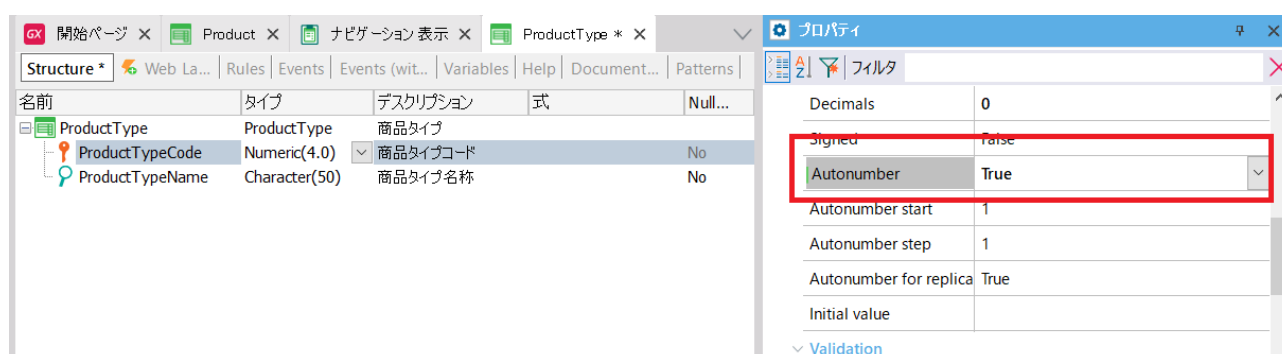
商品タイプごとにコードと名前を格納します:



7. ProductTypeCode 項目属性のプロパティにて「Autonumber」プロパティを「True」に設定



↓



[Autonumber] プロパティについて説明します。既定では [False] に設定されています。これを [True] に変更すると、エンドユーザーが入力した新しい商品タイプはすべて、順番に自動ナンバリングされます。したがって、この識別子項目属性の [Autonumber] プロパティは [True] に設定しておきます。

8. ProductType トランザクションを保存します。

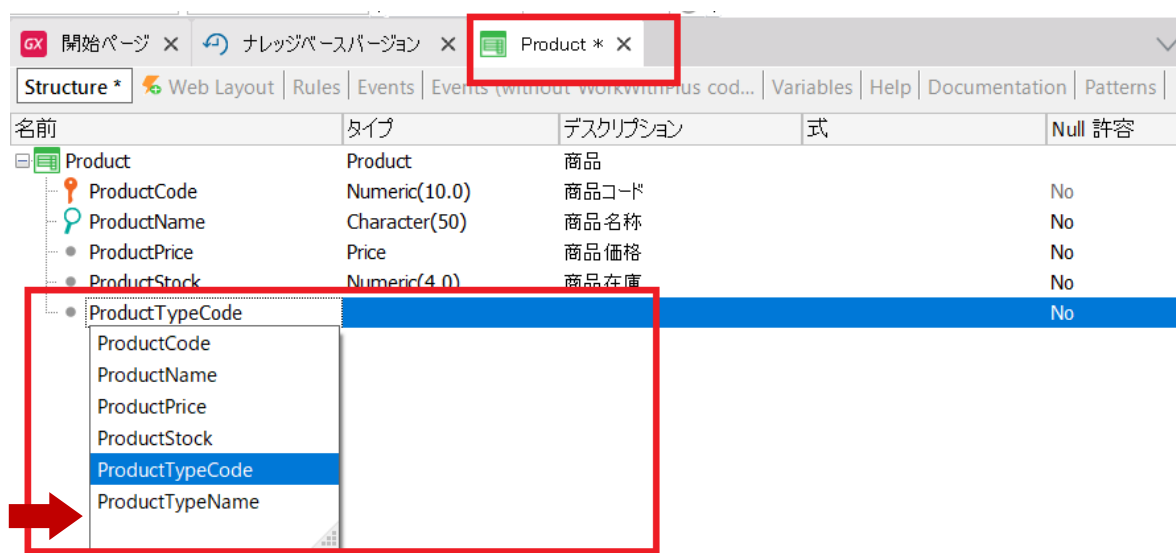
9. Product トランザクションに戻り、Structure に次の項目属性を入力

ProductTypeCode	Numeric(4.0)
ProductTypeName	Character(50)

この際、「P」と入力すると、この文字で始まるナレッジベース内の既存の項目属性のリストが表示されます。

ProductTypeCode、**ProductTypeName** を選択して、その定義をすべて表示します。

※一度 ProductType トランザクションを保存した後、Product トランザクションのウィンドウを「X」で閉じ、開きなおす必要があります。



ProductTypeCode は、**ProductType** トランザクションの識別子項目属性です（すなわち、このトランザクションの主キーです）。したがって、主キーが別のトランザクションに含まれている場合、GeneXus は、その項目属性はそこでは外部キーのロールを果たしていると判断します。

あるトランザクションの主キーである項目属性を別のトランザクションに含めることで、2つのトランザクションを関連付けることができます。

GeneXus は、項目属性名に基づいてリレーションを確立します。したがって、異なるトランザクションで同じ名前の項目属性を見つけると、それらは同じ概念を参照していると見なします。

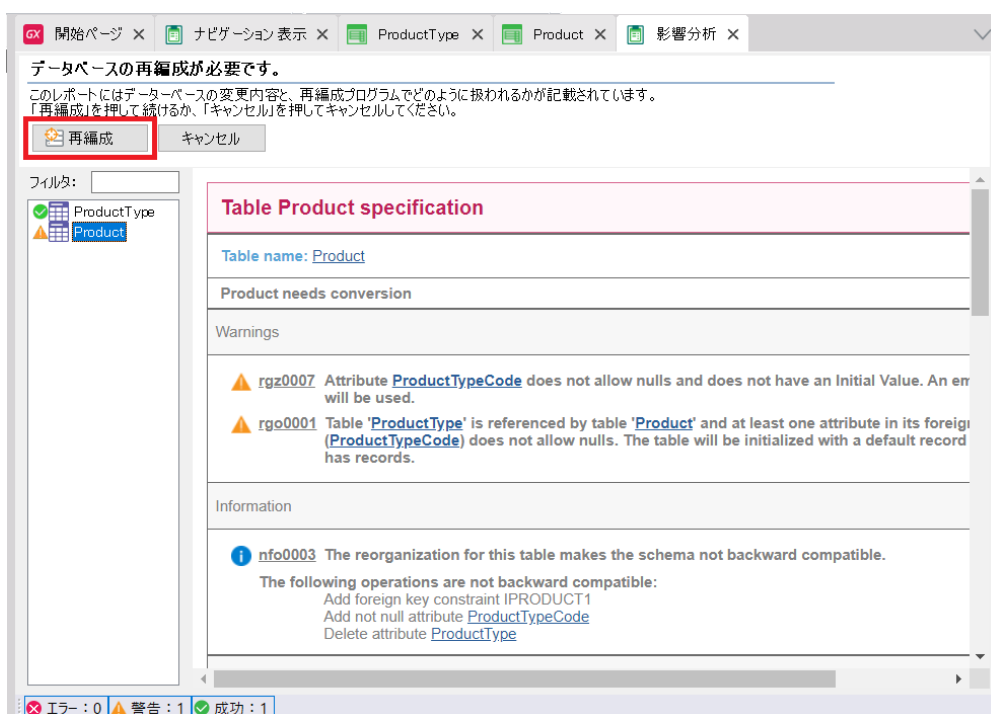
ProductTypeName 項目属性も両方のトランザクションに存在します。しかし、定義済みトランザクションの識別子として設定されていません。したがって、GeneXus は、この項目属性を従属項目属性と見なします。GeneXus は、データベースに作成する **ProductType** 物理テーブルには **ProductTypeName** を含めますが、**Product** 物理テーブルには含めません。

トランザクションの概念と物理テーブルの概念は同じではありません。トランザクションは、現実のオブジェクトを記述するためにナレッジベースに作成する GeneXus オブジェクトです。GeneXus は、当該トランザクション構造と、ナレッジベースに定義されているその他のトランザクション構造を考慮して（さらにいくつかのプロパティも考慮に入れて）、データベースに作成すべき物理テーブルと各テーブルに格納すべき項目属性を決定します。

続いて作成した各トランザクションの実際の動きを見ていきましょう。

10. F5 キーを押下

こちらによりナレッジベース内の新しい定義によって生じる影響が評価され、**[影響分析]** にレポートが表示されます。



注意: 影響分析レポートには、データベースで行う必要がある構造変更が表示されます。

このレポートのメインタイトルから、今回は「データベースの再編成が必要である」ことが分かります。「再編成」という用語は、データベースに変更を加える作業を意味します。このレポートには、**Product** テーブルを更新する必要があると示されています。

1 1. [再編成] ボタンをクリック

こちらにてデータベースを変更するプログラムが作成され、実行されます。そして、アプリケーション自体に対応した必要なプログラムが生成されます。

1 2. [Launch Pad]内の **ProductType** をクリック

The screenshot shows the GeneXus Launchpad interface. At the top, there is a navigation bar with tabs: '開始ページ' (Start Page), 'ナビゲーション表示' (Navigation Display), 'ProductType', 'Product', '影響分析' (Impact Analysis), and 'Launchpad'. Below the navigation bar, there are three main sections: 'WEB アプリ' (Web App), 'ネイティブアプリ' (Native App), and 'API'. The 'WEB アプリ' section is active. Below it, there is a search bar labeled '検索' (Search). The main content area is divided into two sections: 'メイン オブジェクト' (Main Object) and '最近の使用 オブジェクト' (Recently Used Objects). The 'メイン オブジェクト' section is currently empty, displaying the message '表示するメインオブジェクトはありません。' (There are no main objects to display). The '最近の使用 オブジェクト' section contains a table with the following data:

名前	モジュール	タイプ	デスクリプション
Product		Transaction	商品
ProductType		Transaction	商品タイプ

The screenshot shows the '商品タイプ' (Product Type) form in the GeneXus Launchpad. The form has a dark blue sidebar on the left with the text 'UNANIMO'. The main content area is titled '商品タイプ' (Product Type). At the top of the form, there are navigation buttons: '< > >>' and a '選択' (Select) button. Below the navigation buttons, there are two input fields: '商品タイプコード' (Product Type Code) and '商品タイプ名称' (Product Type Name). The '商品タイプコード' field is currently empty, and the '商品タイプ名称' field is also empty.

上のページでは、ユーザーが**商品タイプ**を追加 / 更新 / 削除できます。

13. 1つ目の商品タイプを入力するため、商品タイプ名称に「化粧品」と入力し「実行」ボタンをクリック

ProductTypeCode 項目属性の [Autonumber] プロパティは [True] に設定されており、自動ナンバリングされるため、ユーザーがこの識別子の値を入力する必要はありません。

商品タイプ

◀ ◁ ▷ ▶ 選択

商品タイプコード	<input type="text" value="0"/>
商品タイプ名称	<input type="text" value="化粧品"/>

データが正常に追加されたというメッセージが表示されます。

同時に、フォームがクリアされ、ほかの商品種類を入力できるようになります：

商品タイプ

▲ データが追加されました。

◀ ◁ ▷ ▶ 選択

商品タイプコード	<input type="text" value="0"/>
商品タイプ名称	<input type="text"/>

1 4. 2 つ目の商品種類を入力するため、商品タイプ名称に「医薬品」と入力し「実行」ボタンをクリック

商品タイプ

商品タイプコード

商品タイプ名称

1 5. 画面上部の矢印をクリック

登録されたデータに移動し、ナンバリングされていることが確認できます。

商品タイプ

商品タイプコード

商品タイプ名称

1 6. [Launch Pad]内の Product をクリック し **Product** トランザクションを実行

開始ページ × ナビゲーション表示 × ProductType × Product × 影響分析 × Launchpad ×

WEB アプリ ネイティブアプリ API

検索

メイン オブジェクト

表示するメインオブジェクトはありません。

最近の使用 オブジェクト

名前	モジュール	タイプ	デスク립ション
Product		Transaction	商品
ProductType		Transaction	商品タイプ

17. 次のデータを入力し「実行」ボタンをクリックし1つ目の商品を追加

商品

商品コード

101010

商品名称

STAR筋肉鎮痛剤

商品価格

2000

商品在庫

120

商品タイプコード

2

🔍

商品タイプ名称

医薬品

商品タイプを指定する必要がありますが、商品タイプコードを覚えている場合は、それを入力します。
別の方法として、商品タイプコードの右側にある矢印をクリックしてリストから選択するやり方もあります。

18. 商品タイプトランザクションを開き、医薬品のデータを表示した状態で「削除」ボタンを押下

商品タイプ

商品タイプコード

2

商品タイプ名称

医薬品

削除

終了

実行

関連データが Product に存在するため削除は実行できないというメッセージが表示されます（この商品タイプには「STAR 筋肉鎮痛剤」という商品が属しています）。

商品タイプ

❗ 無効な削除。詳しい情報は **Product** を参照

アプリケーションの拡張

これまで見てきたように、定義した 2 つのオブジェクトを基にして、GeneXus によってさまざまなものが自動生成されました。

ここで、ドラッグストアから、商品ごとに画像を記録したいという要望が出されたとします。

これを実現するには、GeneXus に戻り次の手順を実行する必要があります。

1. **Product** トランザクションに次の項目属性を入力

ProductPhoto	Image	商品イメージ
--------------	-------	--------

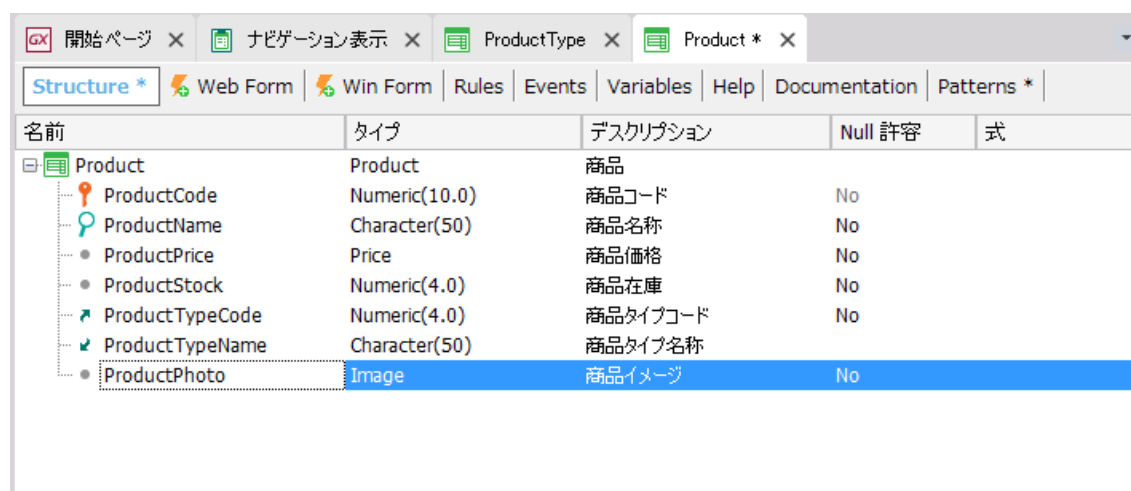


Image タイプでは画像を格納することができます。

Web フォームが自動更新され、**ProductPhoto** 項目属性が組み込まれます。

2. F5 を押下し、再編成をクリック

前回と同様にナレッジベース内の新しい定義によって生じる影響が評価され、**【影響分析】** にレポートが表示されます。

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:

Product

Table Product specification

Table name: [Product](#)

Product needs conversion

Warnings

rgz0007 Attribute [ProductPhoto](#) does not allow nulls and does not have an Initial Value. An empty default value will be used.

Information

nfo0003 The reorganization for this table makes the schema not backward compatible.

The following operations are not backward compatible:

- Add not null attribute [ProductPhoto](#)
- Add not null attribute [ProductPhoto.Uri](#)

Table Structure

Attribute	Definition	Previous values	Takes value from
ProductCode	Numeric (10), Not null		Product ProductCode

エラー: 0 警告: 1 成功: 0

3. ビルドが完了したら[Launch Pad]内の Product をクリックし、Product の画面を表示

Diagram1 x ナビゲーション表示 x ProductType x Product x 影響分析 x Launchpad x

WEB アプリ ネイティブアプリ API

検索

メイン オブジェクト

表示するメインオブジェクトはありません。

最近の使用 オブジェクト

名前	モジュール	タイプ	デスクリプション
Product		Transaction	商品
ProductType		Transaction	商品タイプ

4. 商品コードを覚えている場合は、それを入力、もしくは別の方法として、[選択] ボタンをクリックしてリストから選択し、「STAR 筋肉鎮痛剤」のデータを取得

5. 「商品イメージ」の鉛筆アイコンをクリックし、OneDay トレーニング素材にて同封した「STAR 筋肉鎮痛剤」画像を選択し「実行」ボタンを押下

商品

|< < > >| 選択

商品コード	<input type="text" value="101010"/>
商品名称	<input type="text" value="STAR筋肉鎮痛剤"/>
商品画像	<div style="position: relative; height: 100px;"> × <div style="position: absolute; top: 0; left: 0; bottom: 0; right: 0; background-color: white; padding: 10px; border: 1px solid #ccc;"> <p><input checked="" type="radio"/> ファイルのアップロード</p> <p><input type="radio"/> Web アドレス (URL)</p> <div style="margin-top: 5px;"> <input type="text" value="ファイルの選択"/> <input type="text" value="STAR筋肉鎮痛剤.png"/> </div> <div style="display: flex; justify-content: space-between; margin-top: 10px;"> キャンセル 実行 </div> </div> </div>
商品イメージ	<div style="border: 1px solid #ccc; width: 100px; height: 100px; position: relative; margin: auto;"> </div>

削除
終了
実行

6. 画像が選択できたらトランザクション画面内の「実行」ボタンを押下

これで画像を追加した状態で元のデータを更新できました。

ビジネスルールの追加

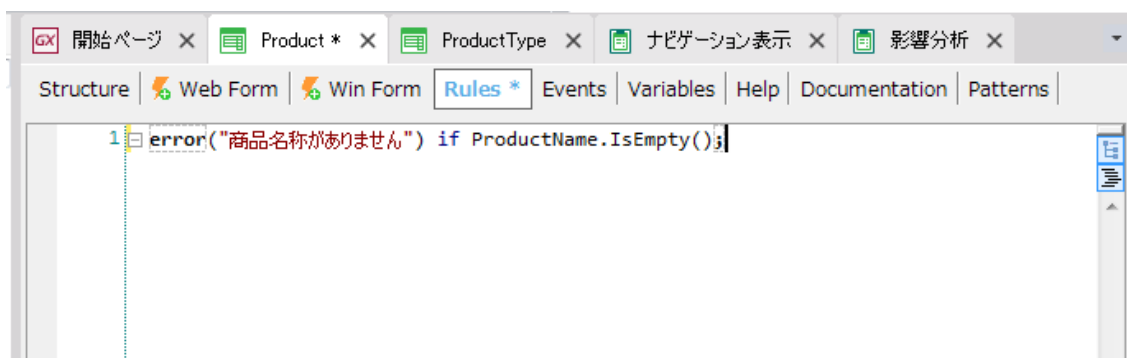
生成されるアプリケーションに GeneXus によって組み込まれる自動コントロールに加えて、ユーザーが特別なコントロールを要求することがあります。トランザクションでは、順守しなければならないルールや検証が必要なコントロールは **【ルール】** セクションで定義します。

たとえば、名前を入力せずに商品を格納できないようにするという要件がある場合は、未入力を回避できる **Error** というルールを使用します。

1. Product トランザクションに移動し、「Rules」を開く

2. 次のような Error ルールを記述

・ `error("商品名称がありません") if ProductName.IsEmpty();`



3. F5 キーを押下し、Product トランザクションを実行

4. 商品名称が未入力のままの状態ですべてのボタンを押下

定義したルールが実行されます。

商品

|< < > >| 選択

商品コード

商品名称 ❗ 商品名称がありません

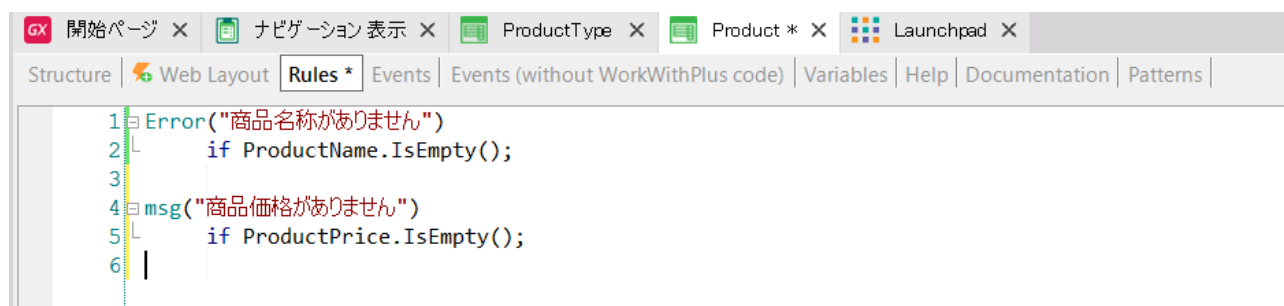
Error ルールとよく似た構文の別のルールもあります。それは **Msg** というルールです。

唯一の違いは、条件を満たした場合、メッセージは通知または警告として表示され、ユーザーは作業を続行できるという点です。

たとえば、商品価格が空白のままになっていることをユーザーに通知し、入力は強制しないようにしたいという要望がありました。その場合 Msg ルールを使用します。

5. Product トランザクションに次のような Msg ルールを記述

- `msg("商品価格がありません") if ProductPrice.IsEmpty();`



6. F5 キーを押下し、Product トランザクションを実行

7. 商品名称が未入力のままの状態です「実行」ボタンを押下

定義したルールが実行されます。

上記までのルールセットは任意の順序で記述でき、実行時の結果はまったく同じになります。

これは、定義された各ルールをトリガーするタイミングは GeneXus が判断するためです（たとえば、関係のあるフィールドからフォーカスが移動したときなど）。

言うまでもなく、GeneXus には、さまざまな検証やアクションを定義するための有用なルールが用意されています。

トランザクションごとに独自の動作ルールを定義する必要がある場合もあります。

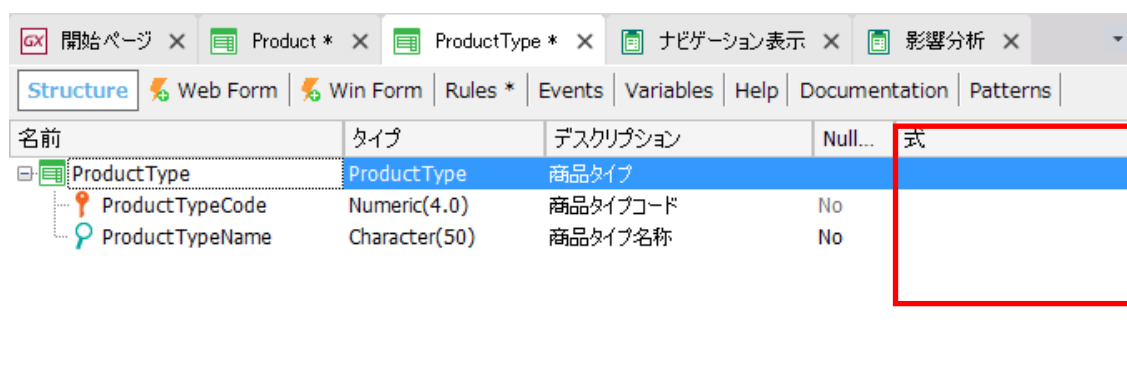
計算の定義：式

アプリケーションでは、特定の項目属性、定数、あるいは関数の値が関与する計算が必要になることが多々あります。これに対応するために、GeneXus には**式**が用意されています。

式は、さまざまな方法で定義できます。

まず、**グローバル式**について説明します。グローバル式とは、開発者が項目属性に関連付けて定義する計算式です。

トランザクション構造には **【式】** というラベルの付いた列があります：



名前	タイプ	デスクリプション	Null...	式
ProductType	ProductType	商品タイプ		
ProductTypeCode	Numeric(4.0)	商品タイプコード	No	
ProductTypeName	Character(50)	商品タイプ名称	No	

項目属性のこの列に計算式が定義されている場合、その項目属性は仮想項目属性になります。つまり、テーブルのフィールドとして物理的に作成されないということです。これは、この項目属性の値が、計算で必要になるたびに取得されるためです。

例を挙げて説明します。ドラッグストアでは、商品に対して割引率を設定し、もともとの商品価格に対してその割引率を計上した商品最終価格を設定する必要があるとします。

そこで、**Product** トランザクションに新しい項目属性を定義し、グローバル式を定義します：

1. 新たに Product トランザクションに次の項目属性を追加

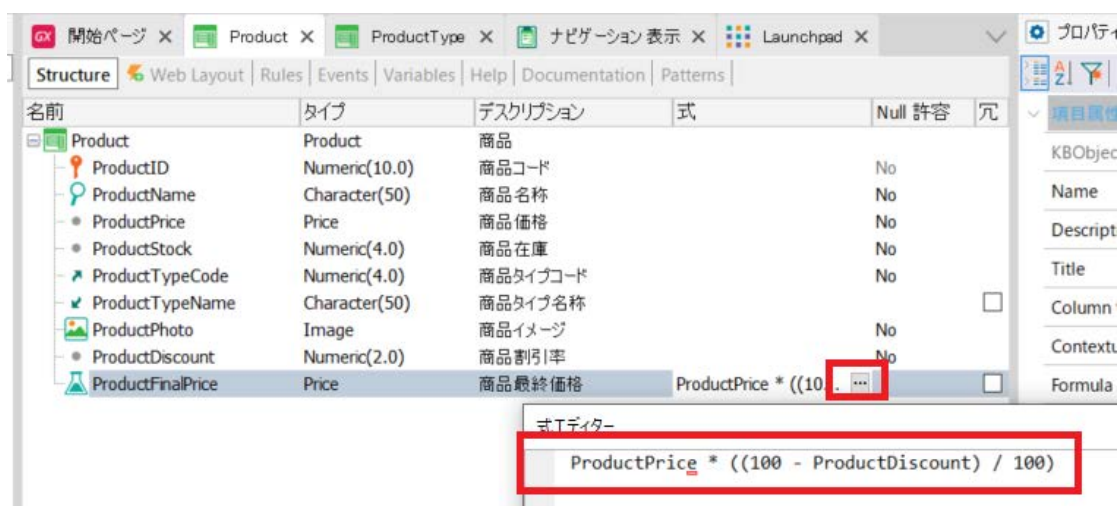
ProductDiscount	Numeric(2.0)	商品割引率
ProductFinalPrice	Price	商品最終価格

※ここでは商品割引率は「%」の表記とします。

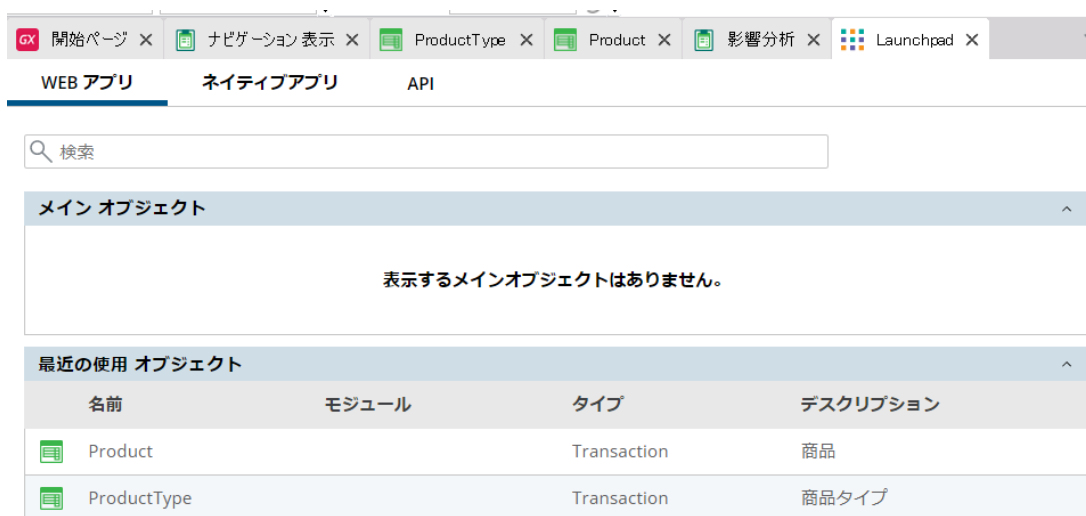
2. ProductFinalPrice 項目属性の「式」に次の記述を追加

・ $\text{ProductPrice} * ((100 - \text{ProductDiscount}) / 100)$

式列の「…」ボタンをクリックすることで式エディタを開くことができます。



3. F5 キーを押下し、再編成ボタンを押下、ビルド完了後に Product トランザクションを実行



4.STAR 筋肉鎮痛剤のレコードを開き、商品割引率に値を入力することで、商品最終価格がどのように計算されるか確認できます。

商品コード 101010

商品名称 STAR筋肉鎮痛剤

商品価格 2000

商品在庫 120

商品タイプコード 2 

商品タイプ名称 医薬品

商品イメージ



商品割引率 10

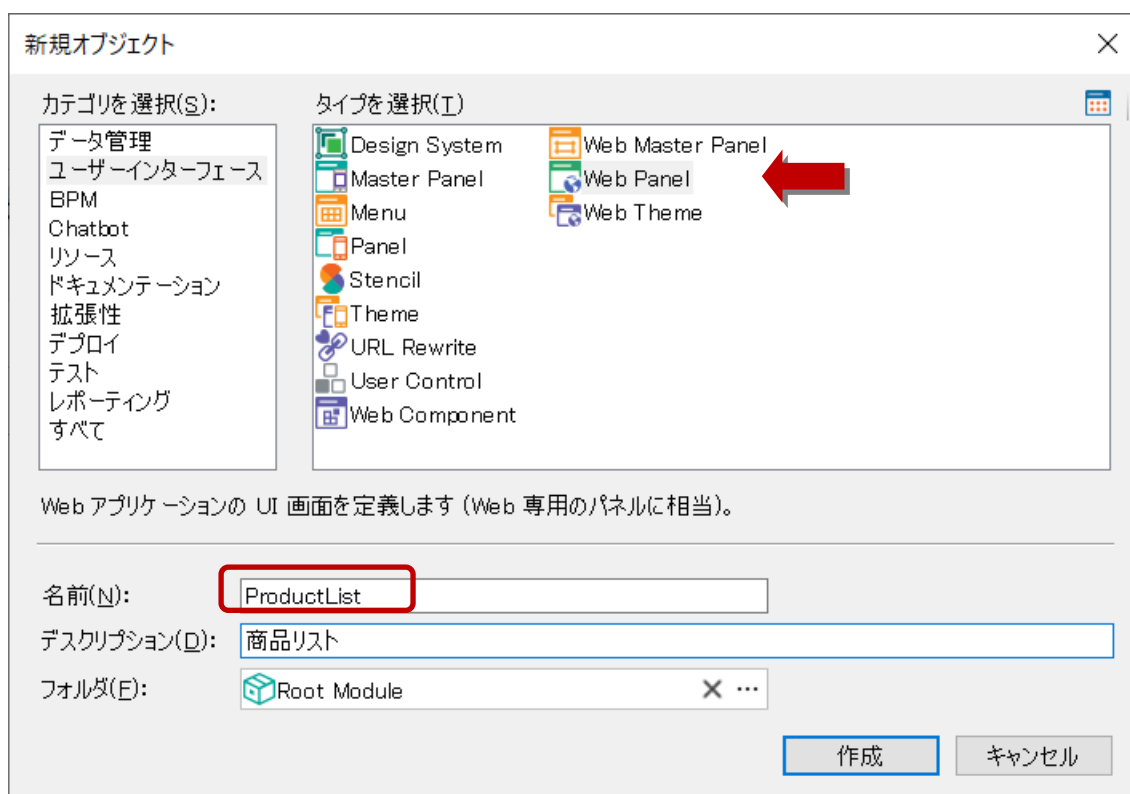
商品最終価格 1800

WEB PANEL(一覧表示)

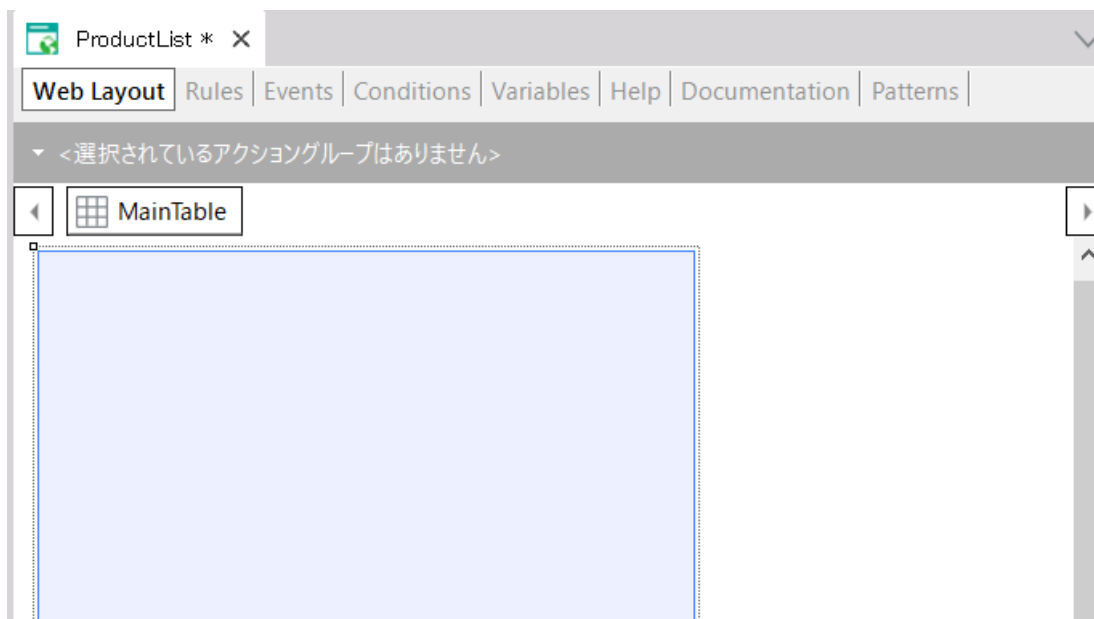
Web Panel はよりカスタマイズ可能なインターフェイスを作成することや詳細な内部処理を行うことが可能です。

今回は商品のリストを表示する画面を作成します。

1. ツールバーより[ファイル] → [新規] → [オブジェクト] を選択し、下図ダイアログを表示
2. 「ユーザーインターフェース」 → 「Web Panel」 を選択
3. 「名前」に「**ProductList**」、「デスクリプション」に「商品リスト」と入力
4. 作成ボタンをクリック



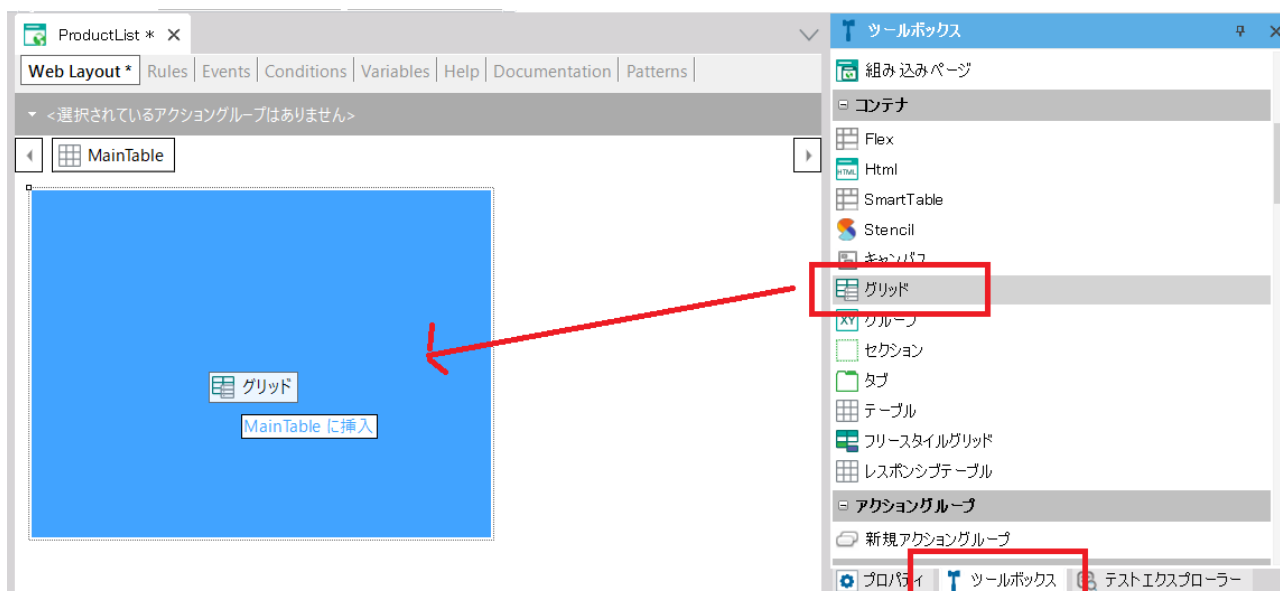
[作成] ボタンをクリックすると、**ProductList** Web Panel 作成され、Web Layout を定義できるように ProductList Web Panel が開きます。



各 Web Panel にはいくつかのセクションがあります。まず、Web Layout では、ドラッグアンドドロップでツールボックス内のコントロールを直感的に配置することが可能です。

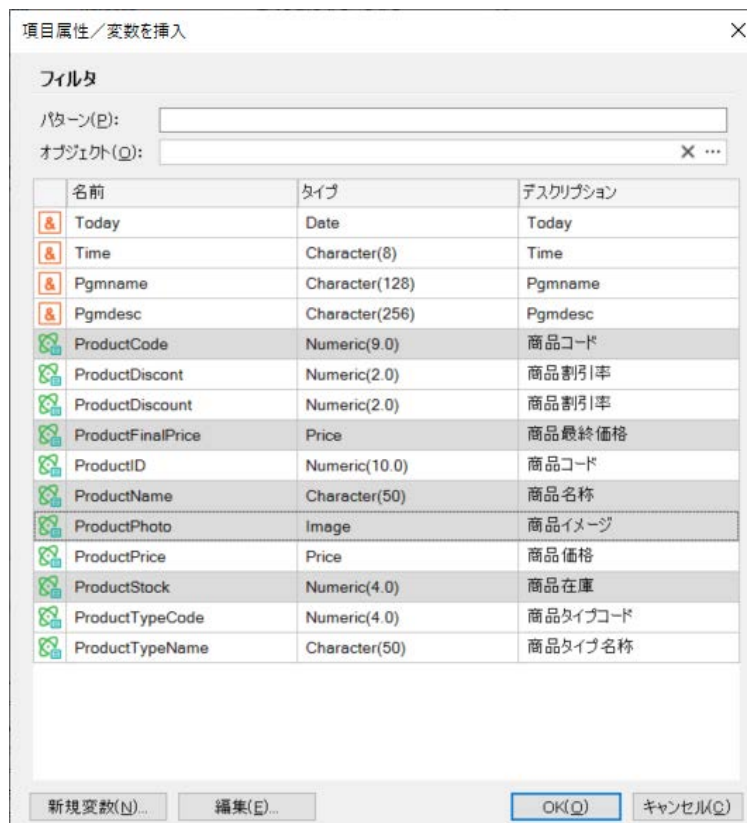
5. ツールボックスからグリッドをドラッグアンドドロップで Main Table に配置

※ツールボックスとは GeneXus が用意した Web Layout を作成するために有効な部品（コントロール）を取りまとめたものとなります。既定ではプロパティウィンドウと同じウィンドウ内のひとつ右側にあります。



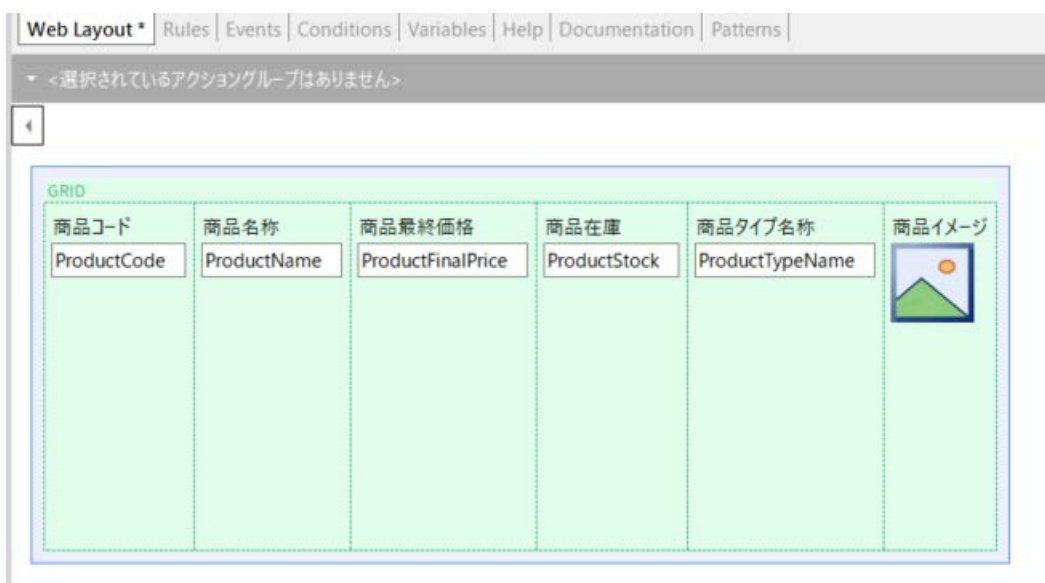
6. ダイアログにて次の項目属性を選択（Ctrl を押しながらで複数選択が可能です。）

- ProductCode
- ProductName
- ProductFinalPrice
- ProductStock
- ProductTypeName
- ProductPhoto



7. 「OK」を押下

これで選択した項目属性がグリッド上に配置されます。



グリッド上の順序が期待した順と違う場合には、ドラッグアンドドロップで順序を変更可能です。

8. F5 を押下してビルドし、Launchpad から作成した ProductList Web Panel オブジェクトを実行

次のように一覧画面が作成されています。

商品コード	商品名称	商品最終価格	商品在庫	商品タイプ名称	商品イメージ
101010	STAR筋肉鎮痛剤	1800	120	医薬品	

PDF 出力 (LAYOUT)

続いて、**Procedure** オブジェクトを使用して、製品(Product)一覧の帳票を作成します。

Procedure オブジェクトは内部処理や帳票出力を行うことが可能です。

今回は商品のリストを表示する帳票を作成します。

1. ツールバーより[ファイル] → [新規] → [オブジェクト] を選択し、下図ダイアログを表示
2. 「データ管理」→「Procedure」を選択
3. 「名前」に「**ProductListOutput**」、「デスクリプション」に「商品帳票出力」と入力
4. 作成ボタンをクリック

新規オブジェクト

カテゴリを選択(S):

- データ管理
- ユーザーインターフェース
- BPM
- Chatbot
- リソース
- ドキュメンテーション
- 拡張性
- デプロイ
- テスト
- レポーティング
- すべて

タイプを選択(T):

- API
- Data Provider
- Data Selector
- Data View
- Domain
- Procedure
- Structured Data Type
- Subtype Group
- Transaction

データベースアクセス、データ更新、および印刷を含むアルゴリズムを実装するプログラムまたはルーチンを定義します。

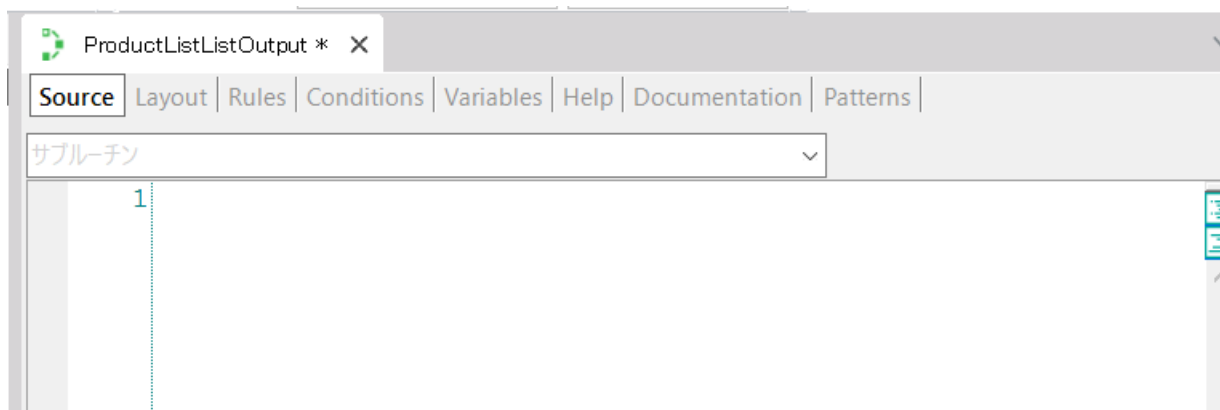
名前(N): ProductListOutput

デスクリプション(D): 商品リスト 帳票出力

フォルダ(F): Root Module

作成 キャンセル

ProductListOutput Procedure が作成され、Source を定義できるように **ProductListOutput** Procedure が開きます:

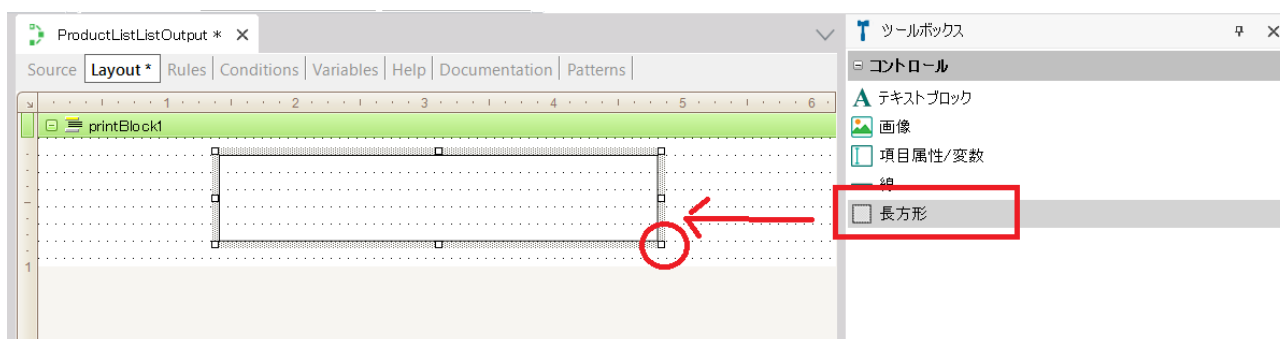


各 Procedure にはいくつかのセクションがあります。まず、Source では、内部処理を記述できます。Layout ではドラッグアンドドロップでツールボックス内のコントロールを直感的に配置することで帳票が作成可能です。

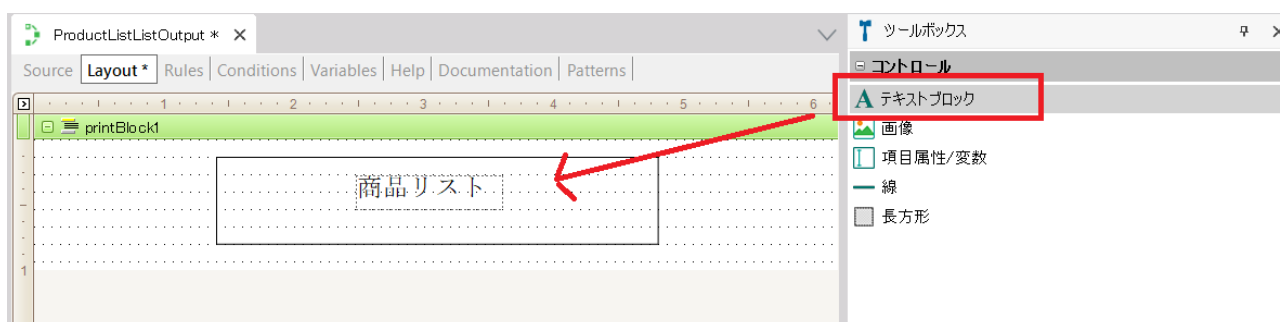
5. Layout のタブをクリックし、移動

まずは帳票のタイトルを作成します。

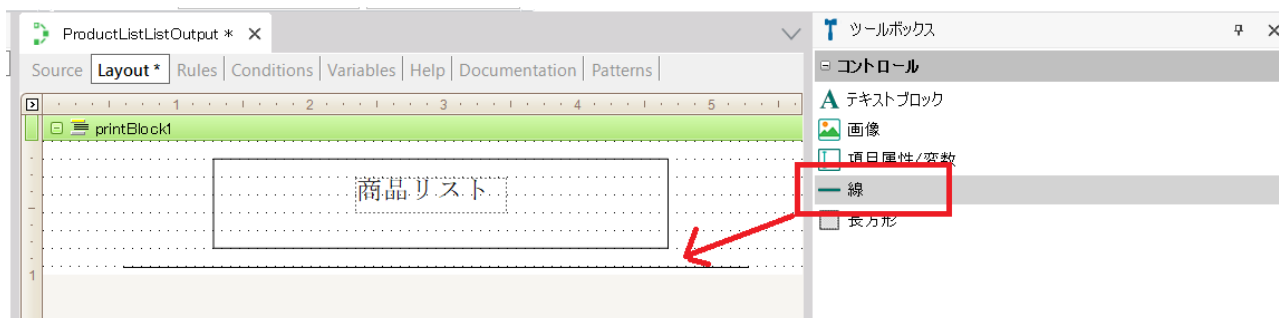
6. 既定として最初から配置されている printBlock1 にツールボックスから**長方形**をドラッグアンドドロップで配置し、右下の四角をドラッグして任意のサイズに変更



7. **テキストブロック**をドラッグアンドドロップで長方形内に配置し、その後テキストブロックをダブルクリックし、**商品リスト**と書き換え



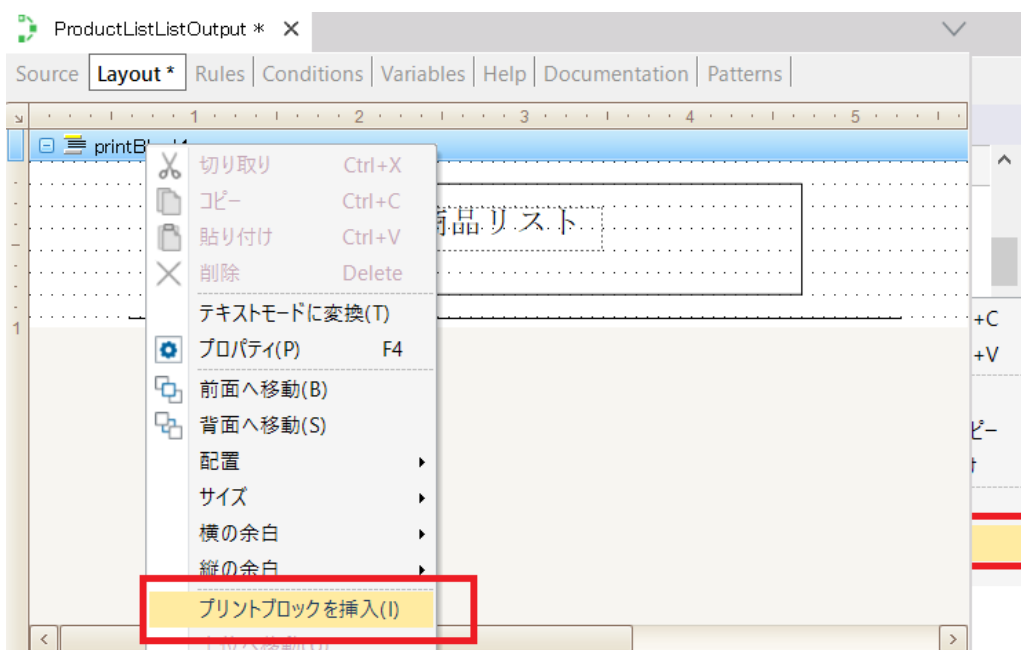
8. タイトルと帳票データの境界線として線を配置し、ドラッグで任意のサイズに変更



帳票のタイトルにつきましては完了しましたので、次は帳票のデータについて作成します。

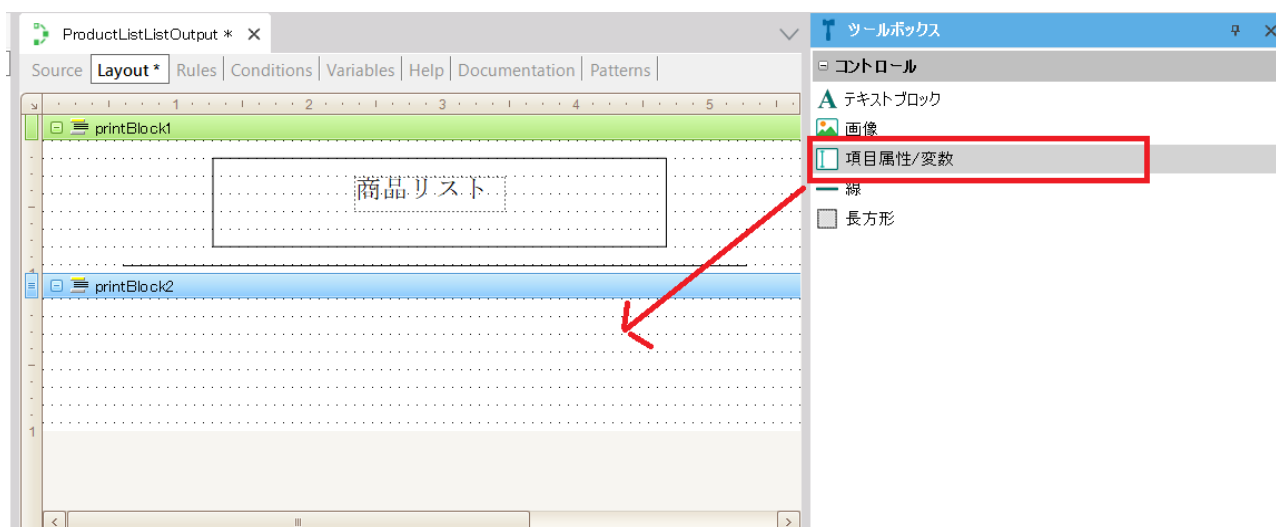
9. Layout 内の任意の場所を右クリックし、「プリントブロックを挿入」を選択

こちらの操作により、新規 printBlock が作成されます。



新規作成した printBlock2 がタイトルの printBlock1 よりも上に配置されてしまった場合には、printBlock2 を右クリックして、**下位へ移動**を選択することによって配置を変更できます。

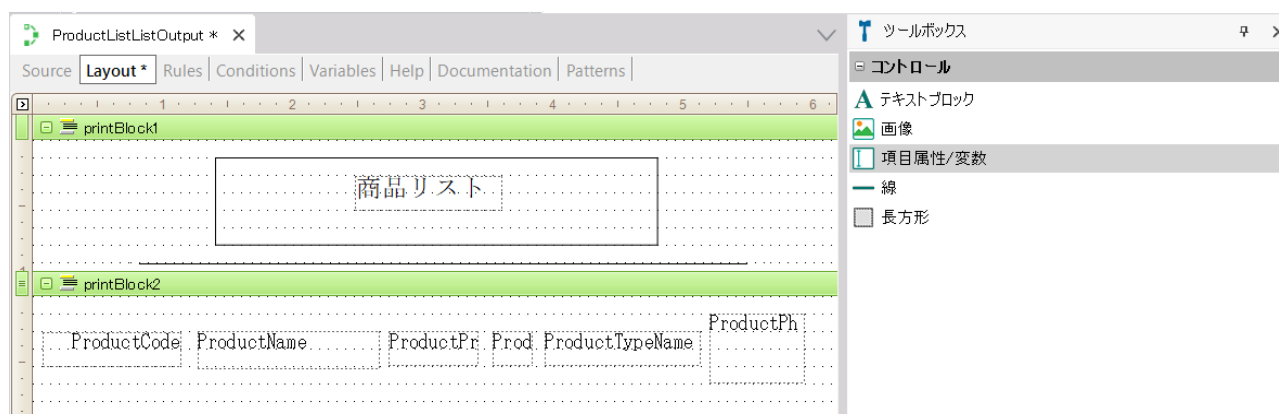
1 0. 作成した printBlock2 にツールボックスから**項目属性/変数**をドラッグアンドドロップで配置



1 1. 項目属性選択のダイアログにて、次の項目属性を選択して配置

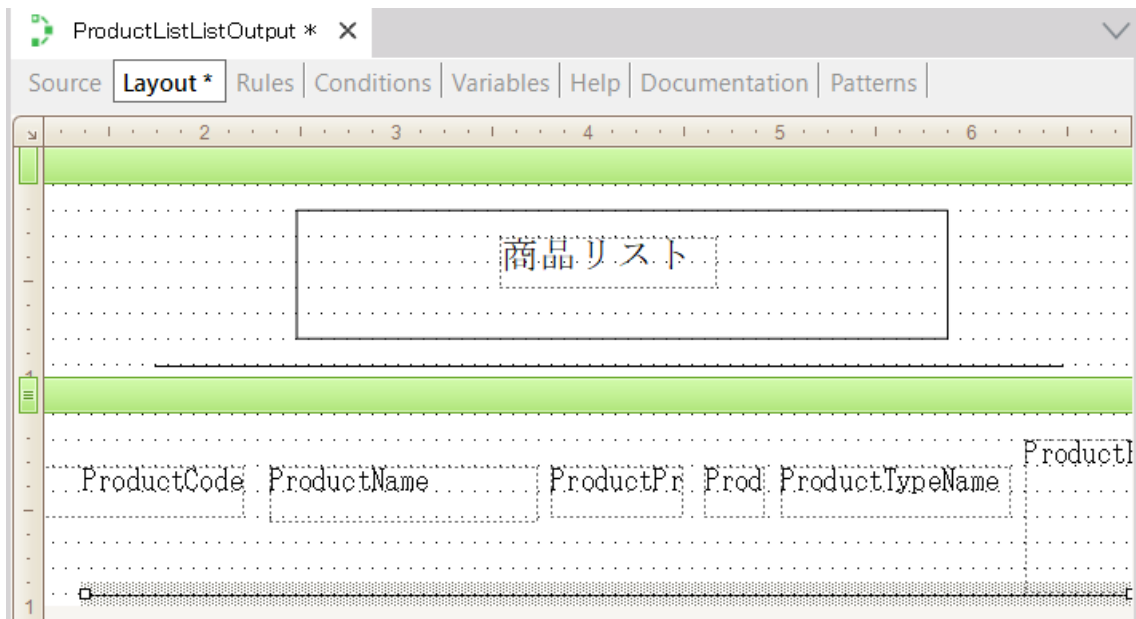
※Web Panel の時と違い、Procedure の Layout では項目属性は一つずつしか配置できません。

- ProductCode
- ProductName
- ProductPrice
- ProductStock
- ProductTypeName
- ProductPhoto



長方形やテキストブロック、線と同様に項目属性/変数も詳細な設定が可能ですので、任意で追加の設定を行ってください。

1 2. 下部に各データの区切りとして線を配置



ここまで Layout の設定は終了し、帳票出力の内部処理の設定を行います。

作成した Layout を帳票として出力するには、次の手順が必要となります。

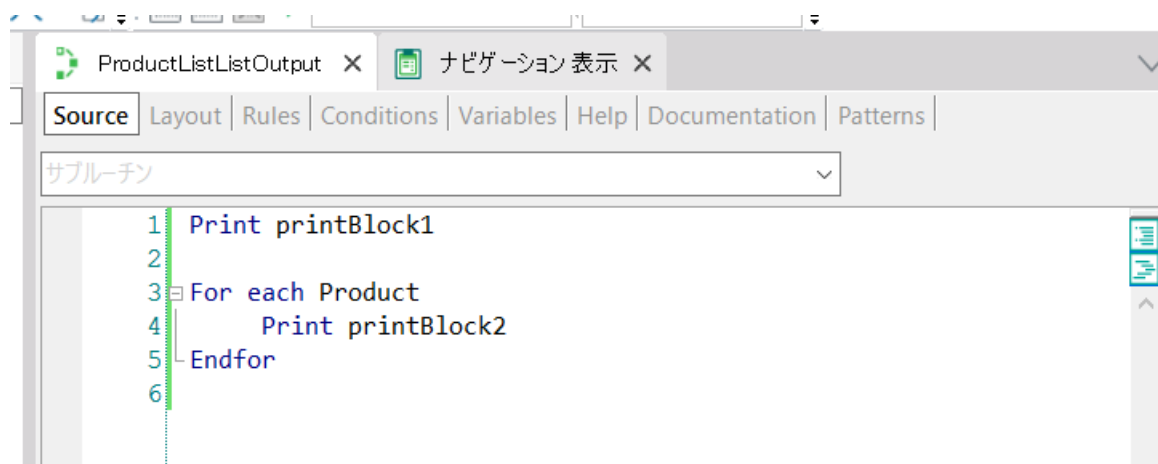
1. **Source** のタブを開き、次のように作成した printBlock を出力する処理を記述

```
Print printBlock1
```

```
For Each Product
```

```
    Print printBlock2
```

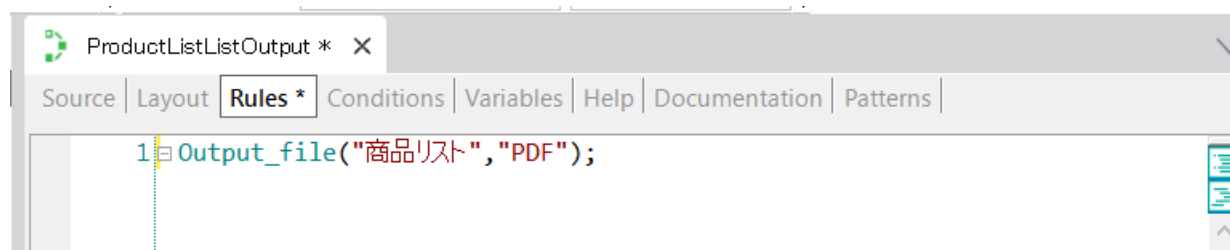
```
Endfor
```



2. **Rules** のタブを開き、次のように記述

- OutPut_file("商品リスト","PDF");

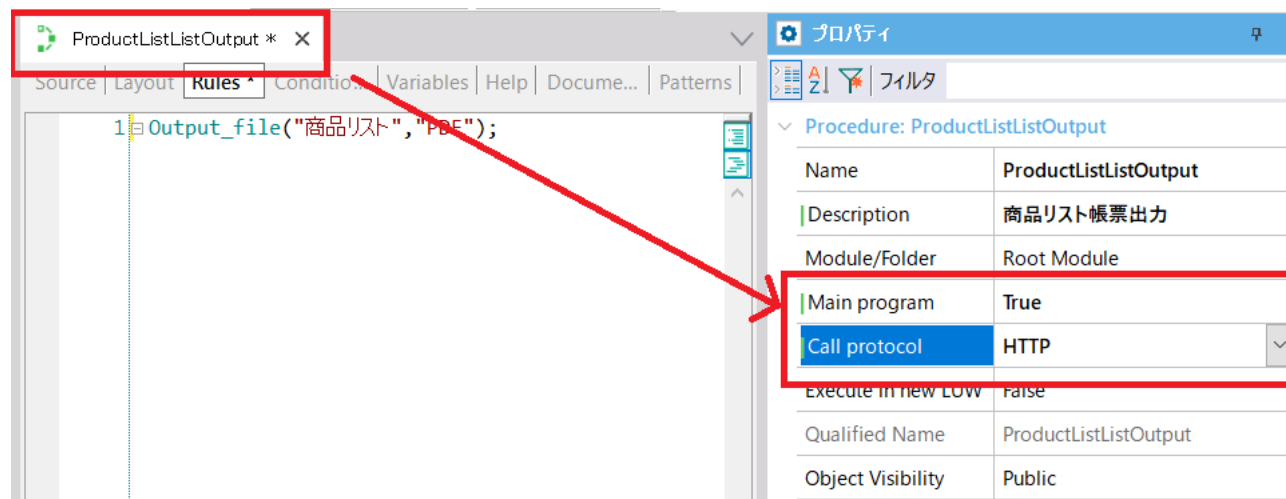
Output_File ルールを使用して PDF として出力することを宣言できます。



4. ProductListOutput プロシーダをクリックして表示されるプロパティにて、次のようにプロパティを設定

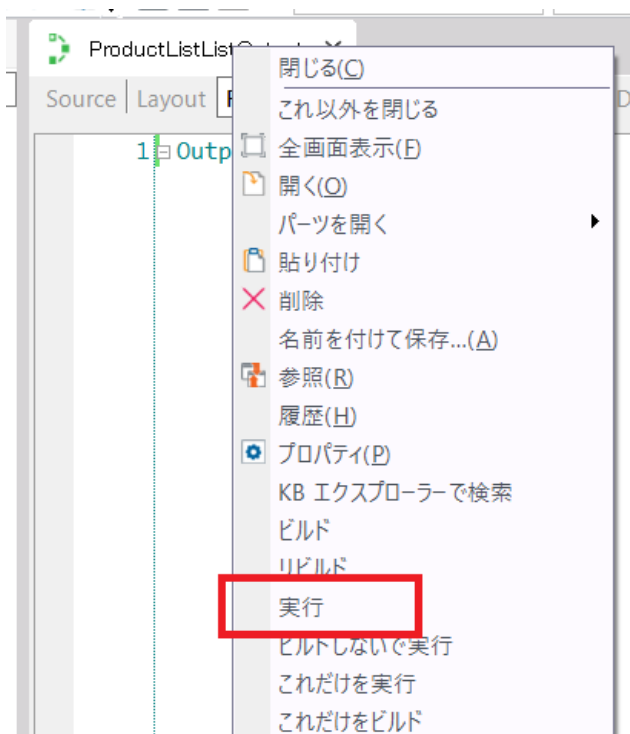
- Main program = True
- Call protocol = HTTP

※ツールボックスウィンドウが開かれている場合にはプロパティウィンドウを開くようにタブ移動してください。



これで Procedure を帳票として出力する設定は完了です。

5. ProductListOutput を右クリックして**実行**を選択



こちらで実際にどのように出力されるか確認できます。

商品リスト

1	STAR筋肉鎮痛剤	2000	120	医薬品
2	傷薬	300	200	医薬品



バッチ処理 (PROCEDURE)

上記で **Procedure** オブジェクトを使用しての PDF 出力の方法を学びましたが、**Procedure** オブジェクトはそのほかにもバッチ処理などの処理を行うことも可能です。

例えば会社の方針で商品の価格改定が必要となり、すべての医薬品の商品価格を 1.2 倍にしたいという要望がありました。

1. ツールバーより[ファイル] → [新規] → [オブジェクト] を選択し、下図ダイアログを表示
2. 「データ管理」 → 「Procedure」を選択
3. 「名前」に「**ProductPriceUp**」、「デスクリプション」に「商品価格改定」と入力
4. 作成ボタンをクリック

新規オブジェクト

カテゴリを選択(S):

- データ管理
- ユーザーインターフェイス
- BPM
- Chatbot
- リソース
- ドキュメンテーション
- 拡張性
- デプロイ
- テスト
- レポーティング
- すべて

タイプを選択(T):

- API
- Data Provider
- Data Selector
- Data View
- Domain
- Procedure
- Structured Data Type
- Subtype Group
- Transaction

データベースアクセス、データ更新、および印刷を含むアルゴリズムを実装するプログラムまたはルーチンを定義します。

名前(N): ProductPriceUp

デスクリプション(D): 商品価格改定

フォルダ(F): Root Module

作成 キャンセル

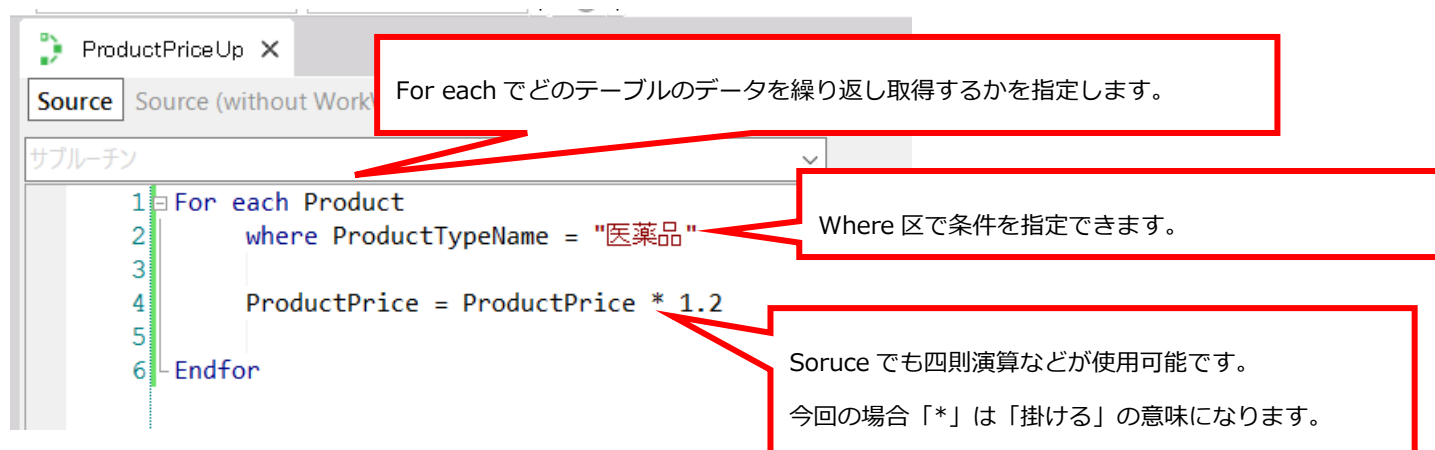
5. Source に移動し、次のように記述

For each Product

where ProductTypeName = "医薬品"

ProductPrice = ProductPrice * 1.2

Endfor



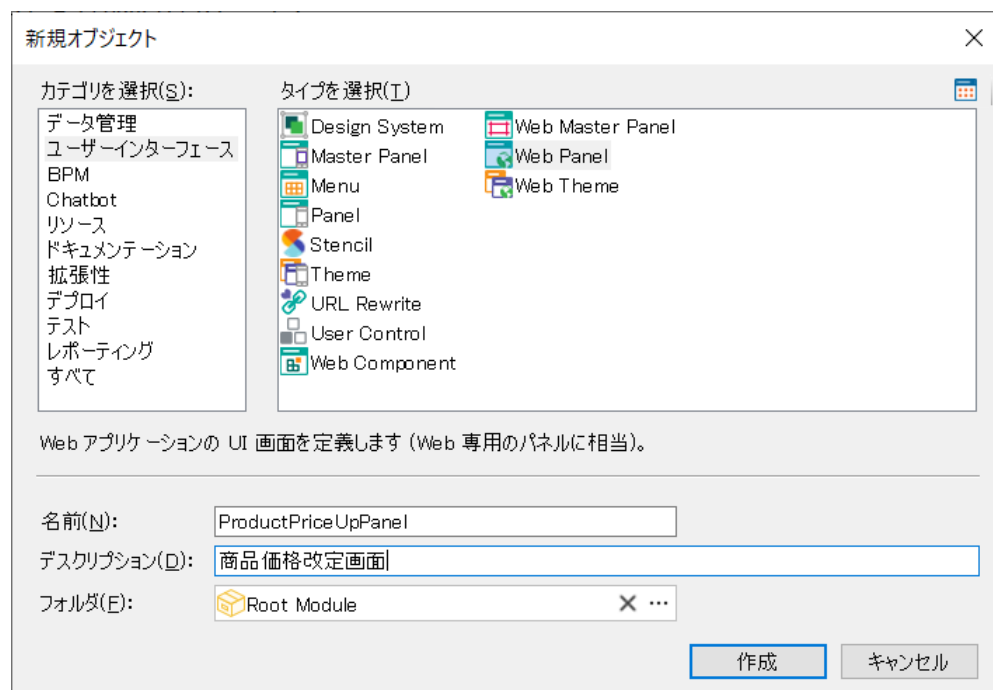
続いていま作成した Procedure を呼び出す Web パネルを作成します。

6. ツールバーより[ファイル] → [新規] → [オブジェクト] を選択し、下図ダイアログを表示

7. 「ユーザーインターフェース」 → 「Web Panel」を選択

8. 「名前」に「**ProductPriceUpPanel**」、「デスクリプション」に「商品価格改定画面」と入力

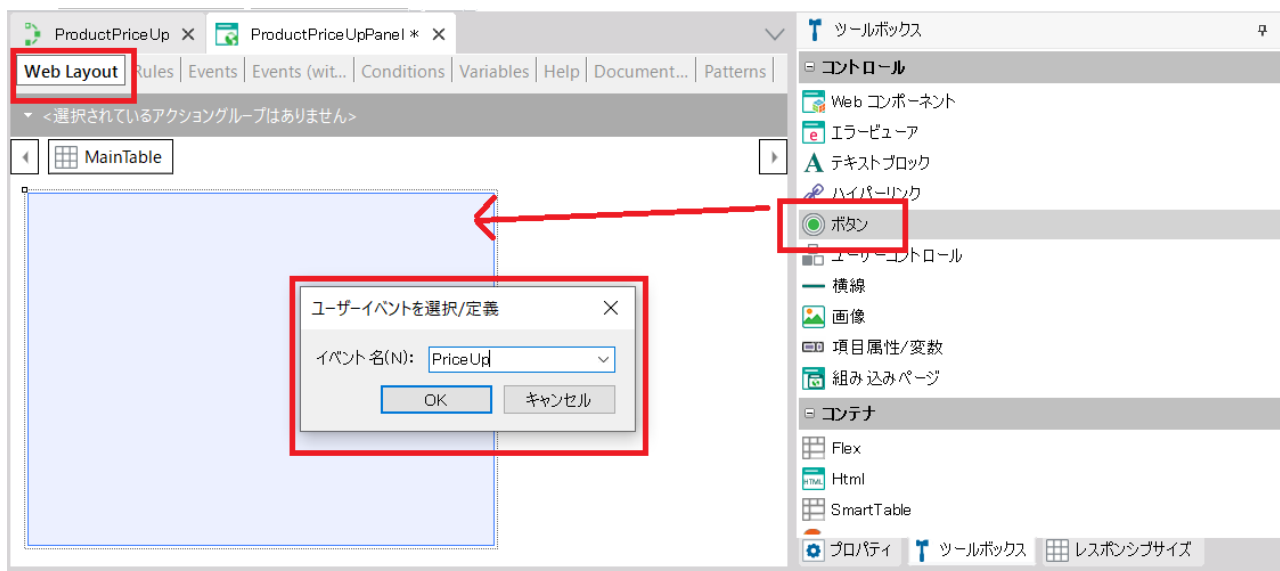
9. 作成ボタンをクリック



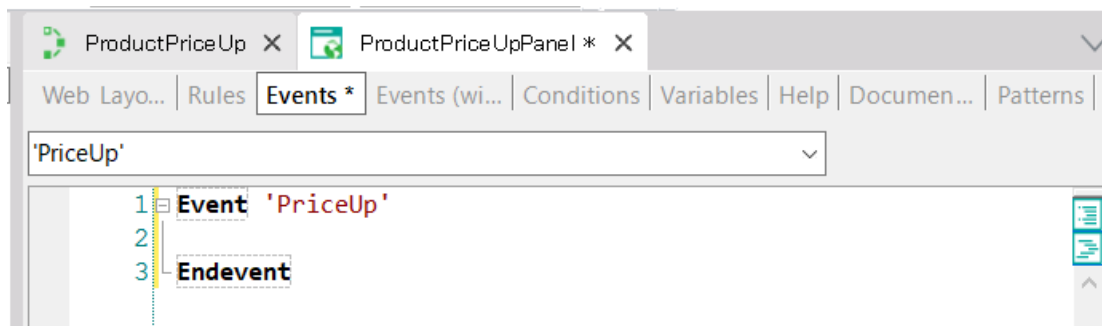
続いて Procedure を呼び出すためのボタンを作成する必要があります。

1 0. **Web Layout** タブにツールボックス内の**ボタン**をドラッグアンドドロップして配置

1 1. ドラッグアンドドロップするとボタンで発生させるイベント名を決めるためのダイアログが表示されるので次のように設定

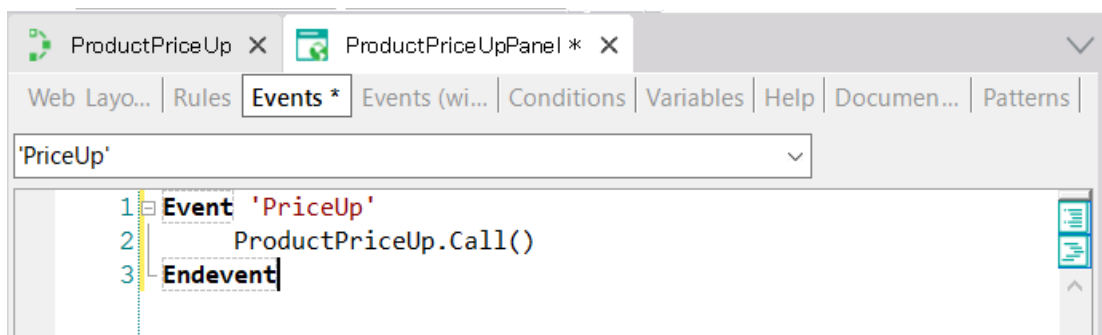


1 2. 配置したボタンをダブルクリックすることで Events タブ内のそのイベントに対応した箇所に移動



1 3. この Event 'PriceUp'内に次のように記載

- ProductPriceUp.Call()



1 4. F5 で実行してボタンを押下

その後商品トランザクションにて医薬品の商品価格が上がっているのを確認してこのパートは終了です。

Application Name

Recents商品価格改定画面

Price Up

クリック

商品

《 < > 》 選択

商品コード

1

商品名称

STAR筋肉鎮痛剤

商品価格

2400

商品在庫

120

パターン (WORK WITH PLUS) の概要 (特徴) 説明

パターンを使用することで、アプリケーションを自動的にさらに強化できます。

パターンの適用はきわめて簡単です。パターンを適用すると、GeneXus によって直ちにオブジェクト、コード、設定が作成され、開発者がプログラミングすることなく便利な動作を実現できます。

パターンには複数の種類が存在しますが、今回はオプション製品である **Work With Plus for Web**(以降 WWP と呼称します)について説明します。

WWP をパターンに適用するだけでユーザーフレンドリーな UI/UX を備えた Web アプリケーションの作成が可能です。

それらを素早く実現するためのアイデアが数多く含まれており、開発チームはより短時間で、より優れたアプリケーションの作成が可能です。

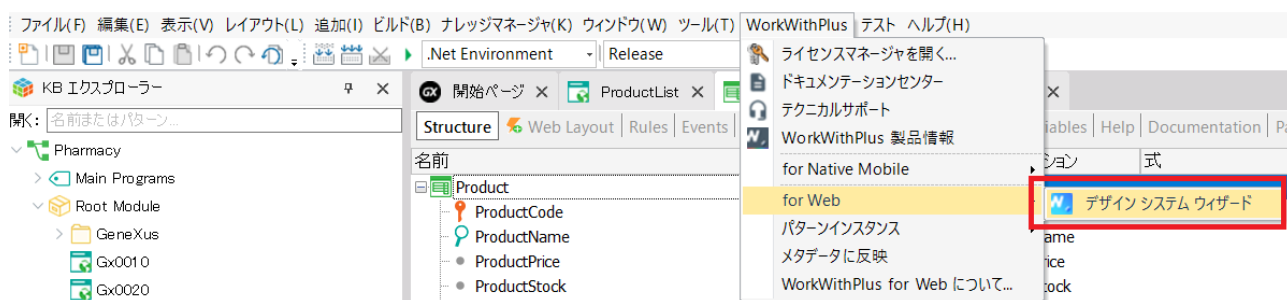
また、高度な機能を簡単に実装できるようにテンプレート化されており、生産性向上にも役立ちます。

WWP の適用

それでは実際にトランザクションに WWP パターンを適用してみましょう。

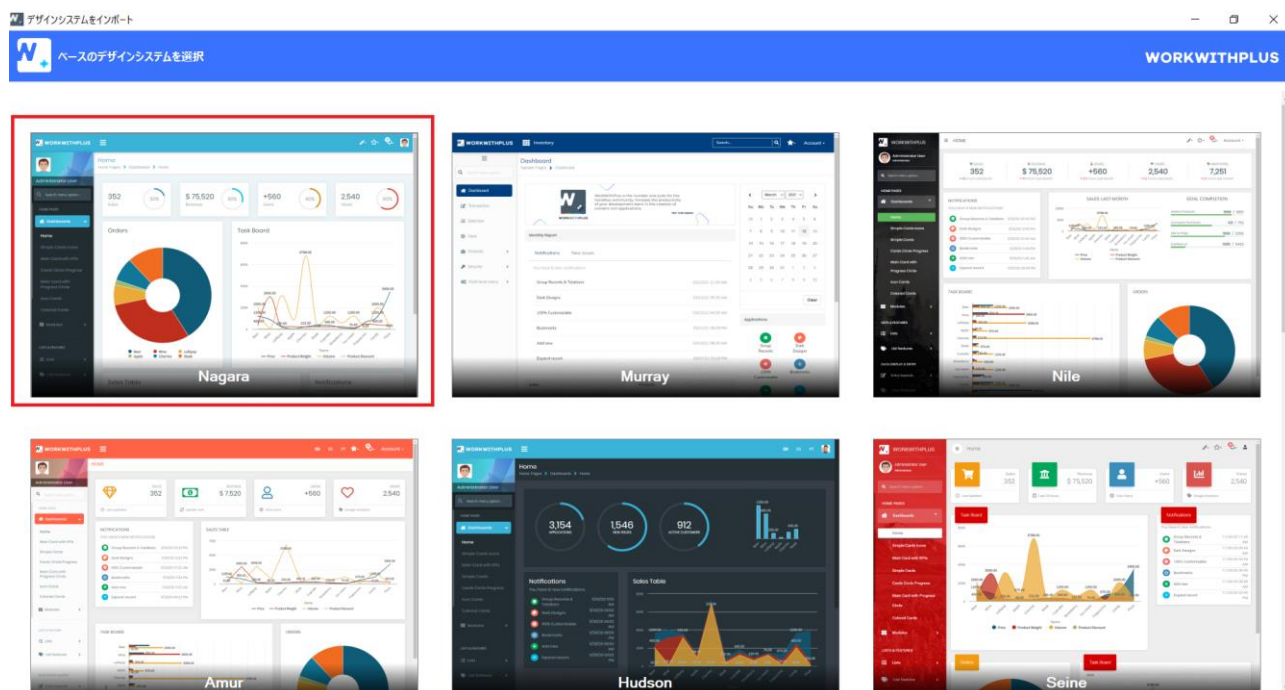
WWP をインストール済みの場合、ツールバーに WorkWithPlus のメニューが表示されます。

1. ツールバーより「WorkWithPlus」→「for Web」→「デザインシステムウィザード」を選択



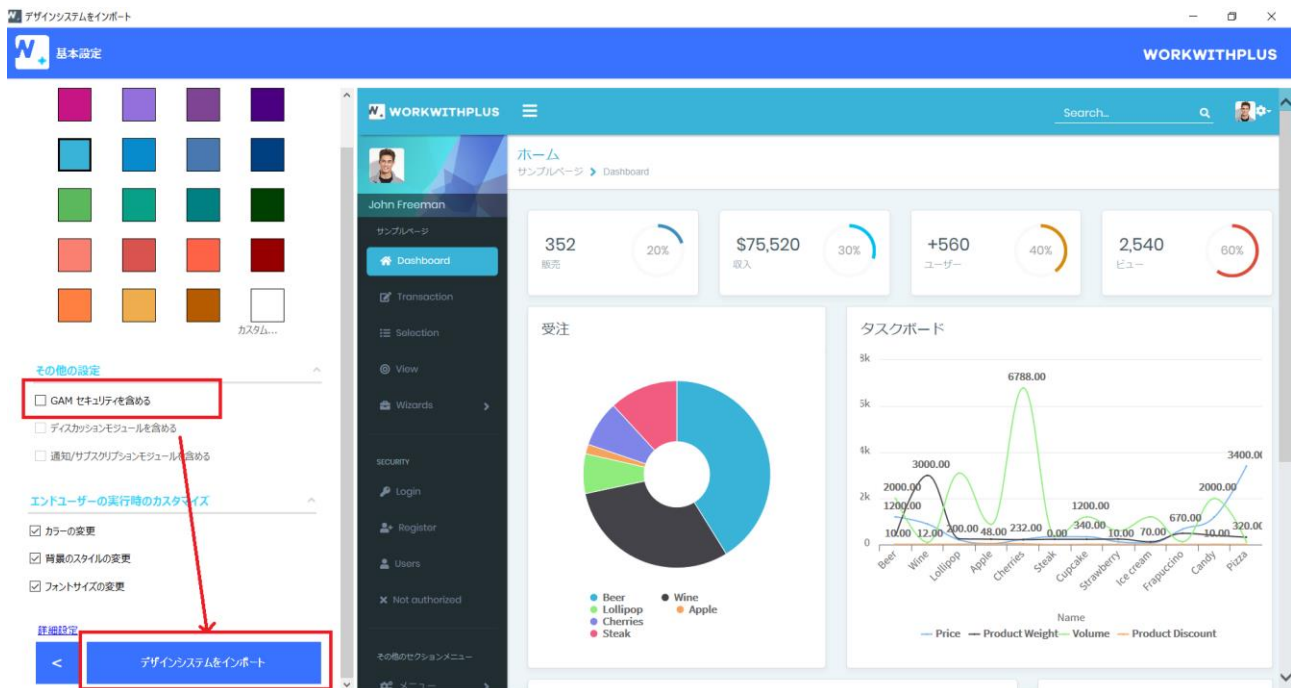
こちらを選択すると WWP のデザインシステムウィザードが表示され、ウィザードに従ってテンプレートを選択できます。

2. 「Nagara」を選択



詳細設定を選択することでより詳細に各種設定が行えますが、今回はデフォルトでインポートします。

3. その他の設定での「GAM セキュリティを含める」のチェックを外す



4. 「デザインシステムをインポート」ボタンを押下

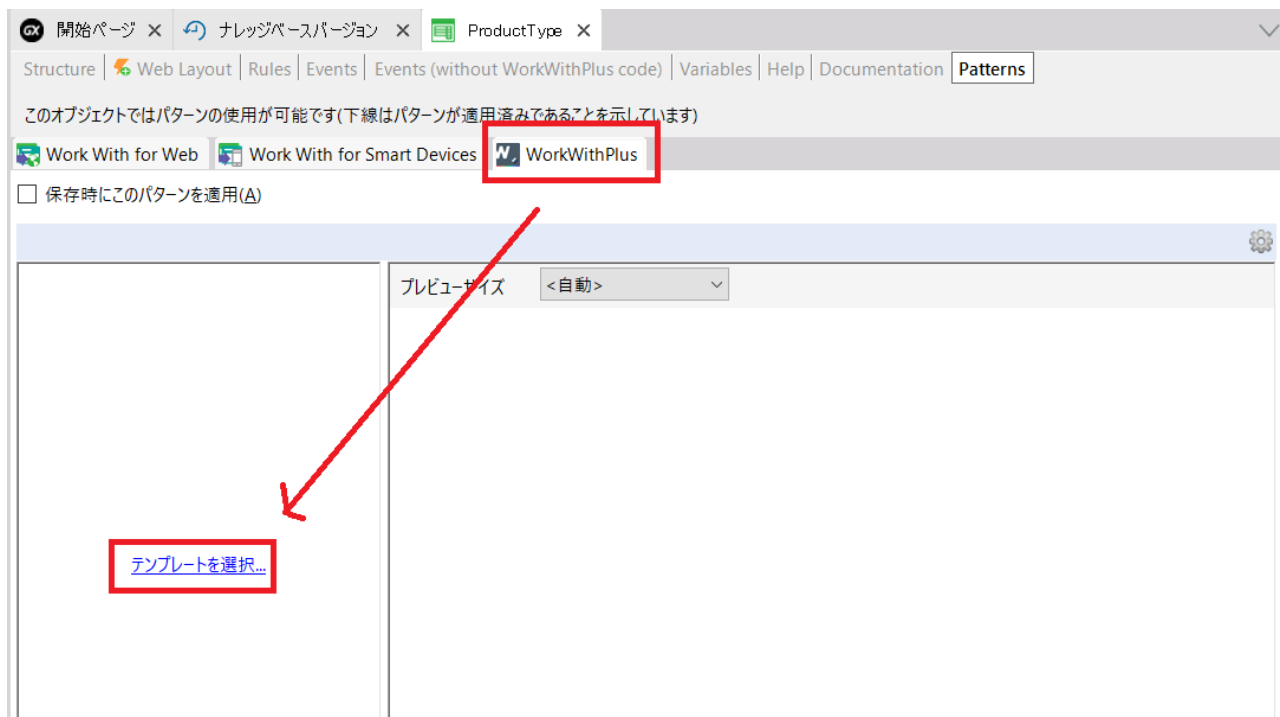
こちらの手順で WWP の選択したデザインシステムに対応する各種オブジェクトがインポートされ、一部プロパティも WWP を適用する設定に変更されます。

インポートが完了しましたら、例として、**ProductType** トランザクションで WWP を試してみます。

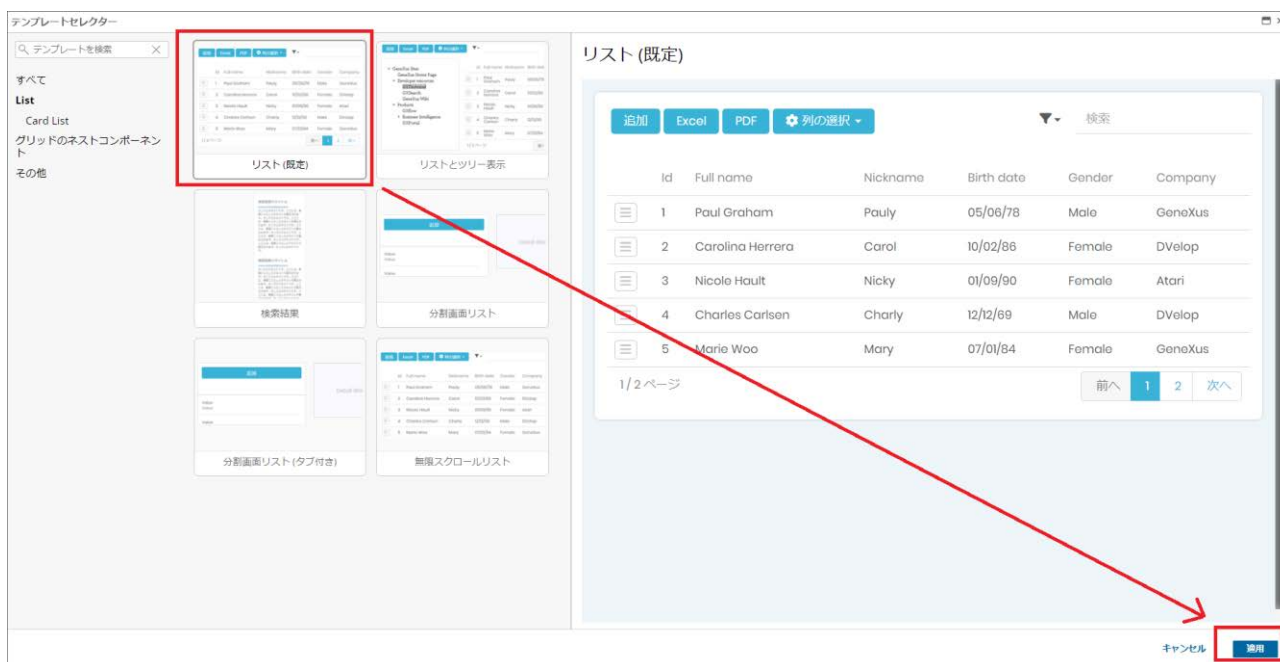
5. ProductType トランザクションの [パターン] を選択



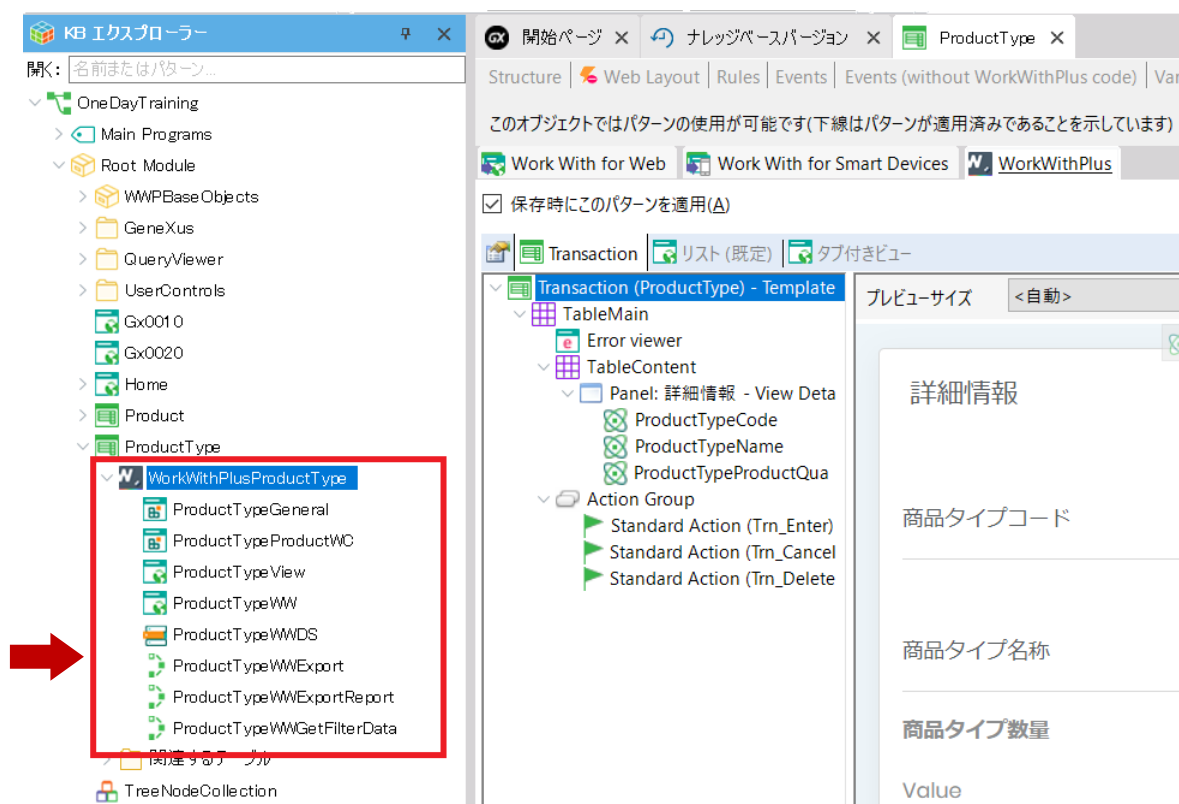
パターンセレクションに移動した時点で自動的にどの WWP テンプレートを適用するかウィンドウが表示されます。自動的に表示されていない場合、**WorkWithPlus** のタブへ移動し、**テンプレートの選択**を押下することでテンプレートの適用画面が表示されます。



6. リスト（既定）を選択し、適用ボタンを押下



[KB エクスプローラー] で **ProductType** トランザクションを確認すると、このトランザクションの下にいくつかのオブジェクトがあることが分かります。



これらのオブジェクトは、WWP パターンが適用されたときに自動的に作成されたものです。

WWP を適用することで、WWP 既定の設定により F5 キー押下時の挙動が変更され、以前までの開発者メニューとは別の画面が表示されるようになっております。



9. 左のサイドバーから開発者メニュー→商品タイプを選ぶことで格納されている商品タイプがすべて表示されます。

ユーザーは、このページで、さまざまな機能を使用して商品タイプの処理を行うことができます。

10. 最初の行の左ハンバーガーメニュー（≡となっているアイコン）内の [更新] をクリック




ProductType トランザクションが開き、この行の商品タイプの詳細を編集できるようになります。

1 1. タイプ名称を編集して実行ボタンを押下

詳細情報

商品タイプコード

1

商品タイプ名称

10代の化粧品

商品タイプ数量

0

実行

終了

実行後、アプリケーションは **ProductTypeWW** ページに戻ります:

追加

Excel

PDF

列の選択

検索

	商品タイプコード ↑	商品タイプ名称 ▼	商品タイプ数量 ▼
	1	10代の化粧品	0
<div><div> 更新</div><div> 削除</div></div>	2	医薬品	1

1/1ページ

前へ

1

次へ

1 2. [削除] ボタンを押下することで、ユーザーがその行の商品タイプを削除できます。

1 3. 追加ボタンをクリックして次のように新しい商品タイプを入力し、実行ボタンをクリック

(前述のとおり、キー項目属性の [Autonumber] プロパティを [True] に設定してあるため、入力するのは商品タイプ名称だけです。)

商品タイプ

その他のオプション > 開発者メニュー > 商品タイプ

詳細情報

商品タイプコード

0

商品タイプ名称

ベビーケア

商品タイプ数量

0

実行

終了

追加後もアプリケーションは **ProductTypeWW** ページに戻ります:

追加 Excel PDF 列の選択

検索

	商品タイプコード ↑	商品タイプ名称 ▼	商品タイプ数量 ▼
	1	10代の化粧品	0
	2	医薬品	1
	3	ベビーケア	0

1/1ページ

前へ 1 次へ

1 4. 次の商品タイプのリンクをクリック: 医薬品

追加
Excel
PDF
列の選択

▼
検索

商品タイプコード ↑
商品タイプ名称 ▼
商品タイプ数量 ▼

☰	1	10代の化粧品	0
☰	2	医薬品	1
☰	3	ベビーケア	0

1/1ページ
前へ
1
次へ

次のように、最初のタブには、選択した商品タイプのすべての詳細が表示され、次のタブには、その商品タイプに属している商品のリストが表示されます。

General
商品
商品タイプへ戻る

商品タイプコード
2

商品タイプ名称
医薬品

商品タイプ数量
1

更新
削除

General
商品
商品タイプへ戻る

追加
▼
検索

商品コード ▼	商品名称 ↑	商品価格 ▼	商品在庫 ▼	Photo	
☰	1	STAR筋肉鎮痛剤	2400	120	
☰	2	傷薬	360	200	

1/1ページ
前へ
1
次へ

各商品タイプには関連する商品がいくつか存在するため、[商品] タブが自動的に生成されました。各商品タイプにほかの種類の関連データも存在する場合は、商品タイプの各関連データのリストが表示されるように、さらにタブが生成されます。

15. ページの左メニューにある**商品タイプ**のリンクをクリックして **ProductTypeWW** ページに戻る

WORKWITHPLUS

医薬品
その他のオプション > 開発者メニュー > 商品タイプ

General 商品 商品タイプへ戻る

追加 検索

商品コード	商品名称	商品価格	商品在庫	商品イメージ
1	STAR筋肉鎮痛剤	2000	120	
2	傷薬	300	200	

1/1ページ 前へ 1 次へ

16. その他の機能として、名前で検索することもできます。たとえば、「医」と入力すると、この文字で始まる商品タイプのみ表示されます。

追加 Excel PDF 列の選択

医

商品タイプコード	商品タイプ名称	商品タイプ数量
2	医薬品	1

1/1ページ メインのフィルタでフィルタリング 前へ 1 次へ

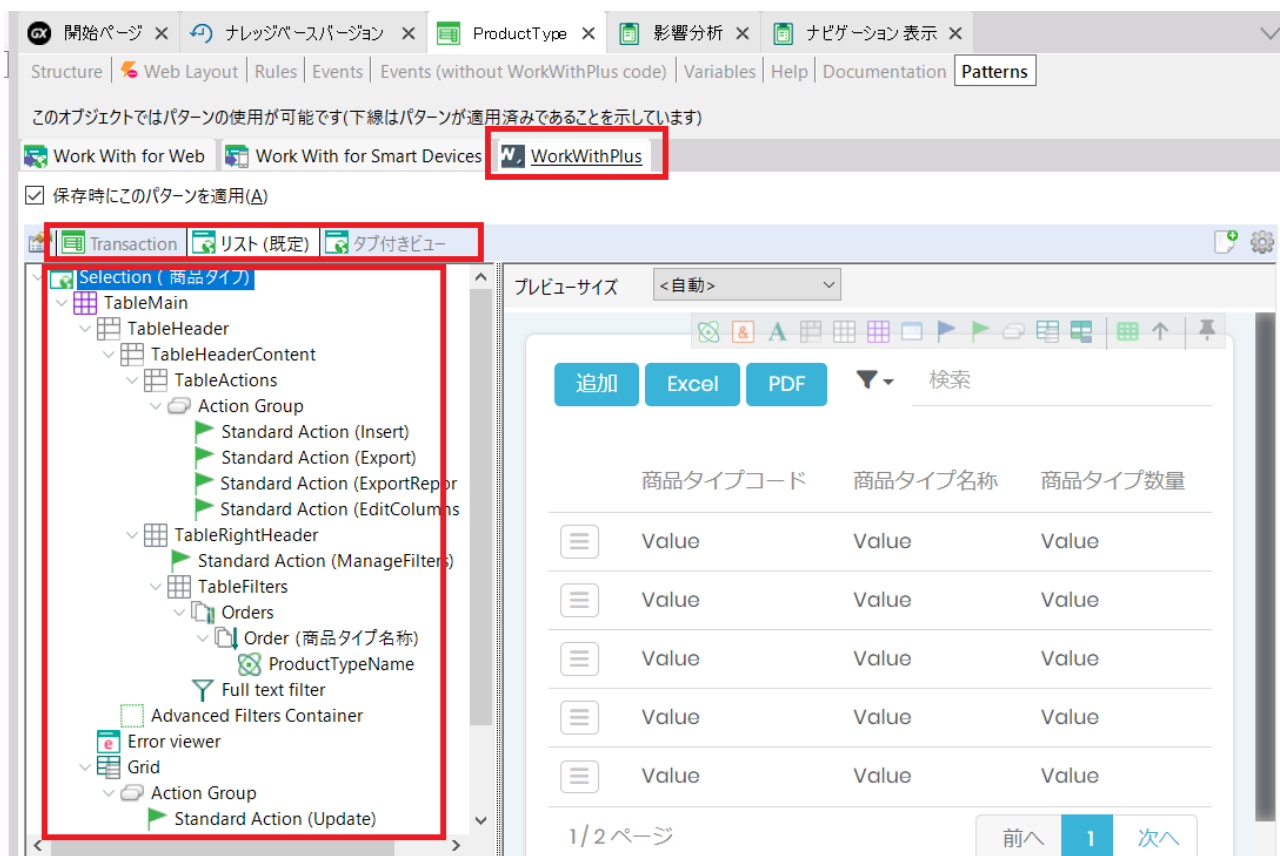
その他本画面からの機能として Excel や PDF の出力機能、表示する列の取捨選択、ソート順の変更などの機能もあります。

GeneXusに戻ります。

これまでの手順では、**ProductType** トランザクションの **[WorkWithPlus]** タブにてテンプレートを選択して **【保存時にこのパターンを適用】** だけを選択しました。

保存後は、自動的に生成された機能がすべて表示されました。

このツリーで設定ができることに気付かなかったかもしれません：



今回では触れませんが、ここでは各種表示画面で設定可能なノード、サブノード、エレメントが含まれており、生成する動作をカスタマイズできるようになっています（たとえば検索条件の変更など）。

その他製品/特長紹介(GXSERVER、GXTEST、SD 生成、GAM)

- チーム開発をサポート
 - o **GeneXus Server** では、ナレッジベースをサーバーにアップロードすることもできます。その後、新しい開発者は、どのような場所からでも必要に応じて、サーバーのナレッジベースからローカルコピーを作成できます。

開発者は（常にローカルで）作業を行い、変更したものをサーバーにアップロードできます。言うまでもなく、競合を解決するためのメカニズムも備えています。このソリューションには、ナレッジベースのモデルのバージョン管理など数多くのメリットがあります。
- GXtest でのアプリケーションのテスト
 - o 新しい機能やバリエーションを実装したら、（変更前の）既存の機能が引き続き適切に動作するか確認する必要があります。アプリケーションが大きく成長すると、テストすべきものが増えるため、このような確認タスクは非常に長い時間がかかる場合があります。GeneXus では、**GXtest** ソフトウェアによってこの種のテストを自動化できるため、開発者の一連のテスト作業が不要になります。テストは自動的に再生され、システムが適切に動作しているか検証されます。
- パターン適用によるスマート デバイス (ネイティブ モバイル)生成
 - o パターン利用では Web 用の画面だけでなく、Android、iOS、(Angular)といったいわゆるスマートフォンアプリ用の画面を簡単に生成可能です。

オプション製品である Work With Plus for Native Mobile を使用することでよりリッチな画面テンプレートをインポート可能です。
- セキュリティの管理
 - o GeneXus には、GeneXus Access Manager (GAM) という（GeneXus に完全に統合された）セキュリティモジュールがあります。これを有効にすることによって、Web アプリケーションとモバイルおよびスマート デバイス アプリケーション両方の認証機能とオーソライズ機能を解決することができます。

次のステップ

- GeneXus はきわめて広範な開発プラットフォームであり、学ぶべきことが山ほどあります。

次の場所で深く掘り下げて学ぶことができます：

- o GeneXus トレーニングコース：

<https://www.genexus.jp/training>

- o GeneXus Wiki：

<http://wiki.genexus.jp/hwikibypageid.aspx?46071>

開発に役立つサイト

<https://www.genexus.jp/community-and-support-jp/support-overview#link05>