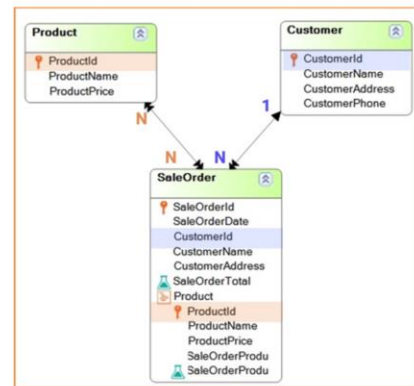
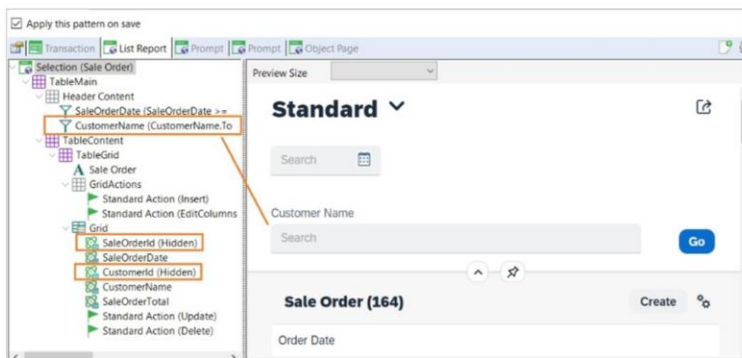


Usando Padrões: Fiori para Web

Parte 3

Avançando em nosso desenvolvimento, vamos aplicar o pattern Fiori for Web à transação que falta em nosso modelo, que é SaleOrder.

Fiori para Web: Ordem de venda



E vamos escolher novamente o floorplan ListReport. Já conhecemos as funcionalidades que serão geradas, então podemos personalizá-lo diretamente antes de sua execução.

A lista de ordens de venda mostra um filtro por data, por ser o atributo descritor, mas também gostaríamos de ter um filtro para poder pesquisar as ordens por nome de cliente. Lembremos que, de acordo com nosso desenho de transações, um cliente pode ter muitas ordens associadas.

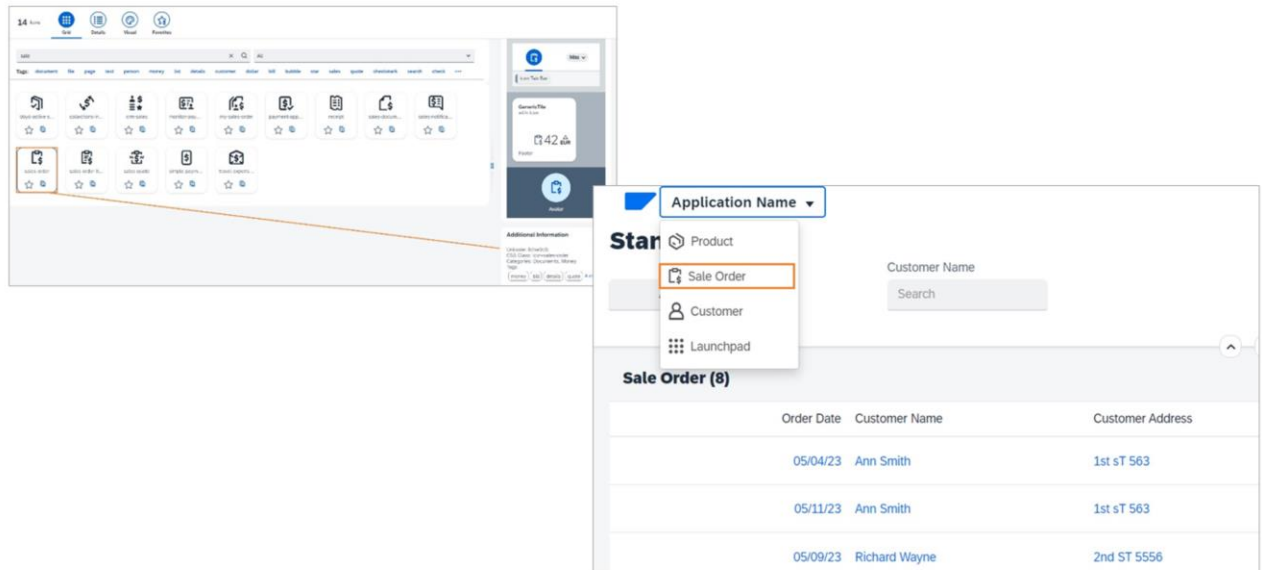
Portanto, no nó Header content da aba ListReport, definimos o novo filtro

No que se refere ao grid que mostra a lista de ordens, não queremos que sejam vistos os identificadores da ordem nem do cliente..

Ao criar uma nova ordem, também não queremos ver seu identificador, pois, como vimos anteriormente, os atributos identificadores estão baseados em um domínio autonumerado.

Assim, na tela da aba Transaction, ocultamos o Id.

Fiori para Web: Ordem de venda



Contudo. Também queremos incluir o acesso a Sale Order no menu da master page. Voltamos então à raiz da aba List Report e habilitamos a propriedade Include in Master page menu.

Não esqueçamos de procurar o código do ícone que representa esta entidade...

Vejamos agora todas estas mudanças juntas em execução. Pressionamos F5..

Na lista de ordens, vemos o filtro por nome de cliente e não vemos os identificadores da ordem e do cliente.

Se criarmos uma nova ordem... também não vemos o Id...

E vemos o acesso no menu da master page. Perfeito! Já vemos as telas como queríamos.

Voltamos para o launchpad... Vamos personalizar o cartão

Queremos que neste cartão seja visto um gráfico com a quantidade de ordens de vendas por data.

Como podemos conseguir isso?

Objeto Query

Query object

Structure	Description
SaleOrdersByDate	Sale Orders By Date
Attributes	
SaleOrderDate	Sale Order Date
Count(SaleOrderId)	Quantity of Sale Order Id
Parameters	
Filters	
Order	

Web component object

Web Layout * Roles | Events | Conditions | Variables | Help | Documentation | Patterns

< No action group selected >

MainTable QueryViewer1

Properties

General

Filter

Query Viewer: UCWeb

Control Name: QueryViewer1

Data bindings

Object	SaleOrdersByDate
Elements	&Elements
Allow Elements On	False
Parameters	&Parameters

Appearance

Fiori launchpad tile

GeneXus conta com o objeto Query que permite definir consultas dinâmicas de uma forma intuitiva e declarativa, e podemos ver sua saída de diferentes formas, por exemplo, como um gráfico.

Para ver o objeto Query em execução devemos definir um Web panel com um controle do tipo QueryViewer que será o encarregado de nos mostrar o resultado da consulta dinâmica.

Por sua vez, para poder ver este web panel no cartão do Fiori launchpad, devemos defini-lo como um Web Component. E o que é um Web component? É um objeto web que pode ser executado de forma independente, como qualquer objeto web, ou pode fazer parte de outro objeto web, como é o caso que precisamos.

Mais sobre objeto Query:

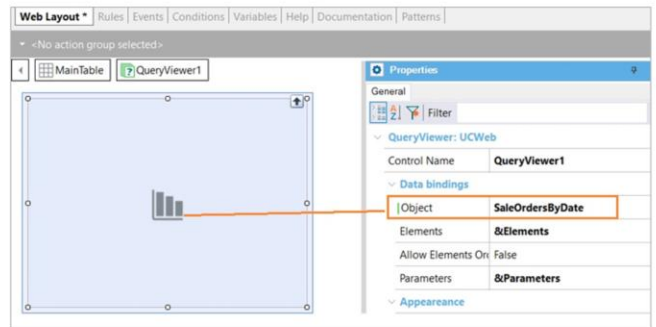
<tps://wiki.genexus.com/commwiki/servlet/wiki?9026,Category%3AQuery+object>

Mais sobre objeto Web component:

<https://wiki.genexus.com/commwiki/servlet/wiki?1864,Category%3AWeb+Component+object>

Fiori para Web: Personalização de cartão (Web component)

Structure	Description
SaleOrdersByDate	Sale Orders By Date
Attributes	
• SaleOrderDate	Sale Order Date
• Count(SaleOrderId)	Quantity of Sale Order Id
Parameters	
Filters	
OrderBy	



Vamos então ao GeneXus para definir a consulta dinâmica e o web component.

Criamos um objeto Query chamado SaleOrdersByDate e definimos sua estrutura,

Queremos ver cada data com sua quantidade de ordens registrada s

Salvamos, e definimos agora um objeto do tipo Web component...

A partir da Toolbox arrastamos um controle do tipo Query viewer... e em sua propriedade Object declaramos o objeto Query que acabamos de criar.

No conjunto de propriedades sob Output, vamos indicar de que forma queremos ver a saída da consulta. Escolhemos Chart.

Colocamos um título para o gráfico

E escolhemos o tipo de gráfico. Temos muitas opções para escolher... e desta vez escolhemos Doughnut.

Podemos deixar que mostre os valores, mas a princípio optamos por não mostrá-los. Salvamos

Fiori para Web: Personalização de cartão (Web component)

The screenshot displays the SAP Fiori development environment. On the left, a tree view shows the project structure for a 'Sale Order' transaction, including components like 'TableMain', 'Header Content', 'TableContent', and 'Grid'. The central preview window shows a 'Standard' view with a search bar for 'Customer Name' and a 'Go' button. Below this is a 'My Home' section with a 'Customer' card and a 'Sale orders by date' chart. The right-hand pane shows the configuration for the 'Launchpad' section, with the following properties:

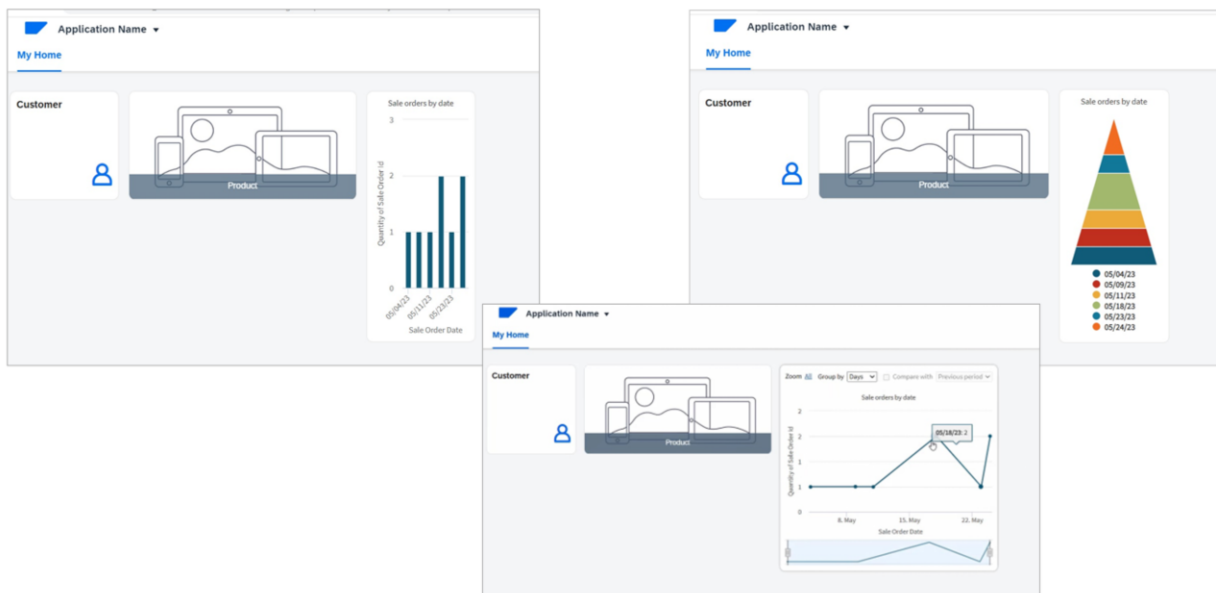
Is Main	False
Master Page	<default>
Form Theme Class	
Is Split Screen Template	False
Generation options	
Update Object	Create default
Web User Experience	Use model property
Base Web Form	Use model property
Launchpad	
Include In Launchpad	True
Tile Category	
Tile Type	Web Component
Tile Web Component	WCSaleOrdersByDate
Tile Size	Automatic
Tile Order Index	0

Bom, e o que devemos fazer agora? Integrar este web component ao cartão de Sale Order no Fiori launchpad.

Vamos para a transação, e na raiz da aba ListReport .. vamos para a propriedade Tile Type e escolhemos Web component.

Aqui indicamos o objeto a mostrar... e indicamos que o tamanho do cartão é automático.

Fiori para Web: Personalização de cartão (mais opções)



Agora sim, estamos em condições de ver tudo em execução. Pressionamos F5
E mais opções que convidamos você a descobrir...

GeneXus[™]
by **Globant**

training.genexus.com