



favorito (7) imprimir anotar marcar como lido dúvidas?

PIVOT no SQL Server: invertendo linhas e colunas em um exemplo real

 (17)  (1)

Veja neste artigo como utilizar o operador PIVOT no SQL Server para inverter as linhas com as colunas de uma consulta.

Imagine que em um dia você se depara com a seguinte situação, precisa transformar as linhas do meu SELECT em COLUNAS, você logo pode pensar “vou fazer pelo meu software”, usando o Delphi, .NET Java, etc. Mas fazer um processamento deste tamanho gera consumo de recurso, o que torna o aplicativo mais lento. O ideal neste tipo de processamento é deixar o nosso servidor/Banco de Dados executar. Para essa situação temos um operador muito interessante e até um pouco desconhecido por muitos, o PIVOT.

Muito se fala sobre PIVOT, mas o pessoal sempre se pergunta: O que é? Qual a sintaxe? Como aplicar em um SELECT com colunas fixas? Como aplicar em um SELECT com colunas dinâmicas? Vamos esclarecer e explicar cada uma destas perguntas.

O que é o PIVOT?

O operador PIVOT no SQL Server, gera uma expressão com seus valores em tabela, ou seja, ele pega os dados que ficam em formato horizontal (linhas) e os coloca em formato vertical (Colunas).

Veja um exemplo bem claro do formato horizontal (linhas):

Receba notificações :)

AA3705322	BLUSA BASICA	M	33
AA3705322	BLUSA BASICA	G	33
AA3705322	BLUSA BASICA	GG	27
AA3705322	BLUSA BASICA	UN	23

Veja um exemplo de como seria usando o PIVOT (transformando as linhas em colunas)

CodProduto	Descricao	P	M	G	GG	UN
AA3705322	BLUSA BASICA	28	33	33	27	23

Qual a sintaxe?

A sintaxe usada no PIVOT parece ser um pouco complexa, mas não é, ela é simples, não iremos detalhar essa sintaxe, pois o objetivo principal é mostrar na prática.

Listagem 1: Sintaxe do PIVOT

```
SELECT <non-pivoted column>,
    [first pivoted column] AS <column name>,
    [second pivoted column] AS <column name>,
    ...
    [last pivoted column] AS <column name>
FROM
    (<SELECT query that produces the data>)
    AS <alias for the source query>
PIVOT
(
    <aggregation function>(<column being aggregated>)
FOR
    [<column that contains the values that will become column headers>]
    IN ( [first pivoted column], [second pivoted column],
        ... [last pivoted column])
) AS <alias for the pivot table>
<optional ORDER BY clause>;
```

Receba notificações :)

Essa é a forma mais comum de aplicar o PIVOT em uma consulta. Você conhece a quantidade exata de colunas desejadas, como por exemplo, um relatório ANUAL das suas vendas.

A forma tradicional seria o seguinte relatório:

ANO	MÊS	VALOR
2013	1	R\$ 785.694,00
2013	2	R\$ 46.582,00
2013	3	R\$ 216.594,00
2013	4	R\$ 56.223,00
2013	5	R\$ 112.546,00
2013	6	R\$ 1.202.355,00
2013	7	R\$ 120.036,00
2013	8	R\$ 804.665,00
2013	9	R\$ 601.128,00
2013	10	R\$ 9.564.122,00
2013	11	R\$ 900.064,00
2013	12	R\$ 487.562,00

Receba notificações :)

Agora, usando o PIVOT teríamos o seguinte resultado:

Ano	Janeiro	Fevereiro	Março	Abril	Etc....
2013	R\$ 785.694,00	R\$ 46.582,00	R\$ 216.594,00	R\$ 56.223,00

Como sabemos que o relatório é anual, temos como prever o seu tamanho. Ele terá que preencher todos os meses do ano, no caso teremos 12 colunas (as colunas foram suprimidas para facilitar a leitura deste artigo), fora a coluna do ano.

```
-----  
    , [1] AS JANEIRO  
        , [2] AS FEVEREIRO  
        , [3] AS MARÇO  
    , [4] AS ABRIL  
        , [5] AS MAIO  
        , [6] AS JUNHO  
        , [7] AS JULHO  
        , [8] AS AGOSTO  
        , [9] AS SETEMBRO  
        , [10] AS OUTUBRO  
    , [11] AS NOVEMBRO  
        , [12] AS DEZEMBRO  
FROM VENDAANUAIS PIVOT (SUM(V valor)  
FOR MES IN ([1],[2],[3],[4],[5],[6],[7],[8],[9],[10],[11],[12]))P  
ORDER BY 1;
```

Na parte de inicial fazemos um SELECT normal, onde colocamos um nome para as colunas, apenas um apelido, usando o 'AS', subentendendo que 1 = JAN, 2 = FEV, 3 = MAR, e assim por diante.

Em seguida informamos qual seria a tabela onde os dados estão localizados. Logo após, colocamos o operador PIVOT, este por sua vez será usado com uma função de agregação, neste caso o SUM e agrupando os valores de cada mês.

Finalizando, usamos o 'FOR IN', para preencher os respectivos valores em suas colunas. E finalizando com um ORDER BY comum.

Receba notificações :)

Como aplicar em um SELECT dinâmico?

Neste momento é o ponto mais interessante do nosso artigo, temos a parte dinâmica, onde você não sabe qual é o número de COLUNAS que o seu SELECT irá gerar. O código abaixo mostra uma situação REAL do dia-a-dia de muitos desenvolvedores, temos a seguinte estrutura para o nosso banco de dados de exemplo:

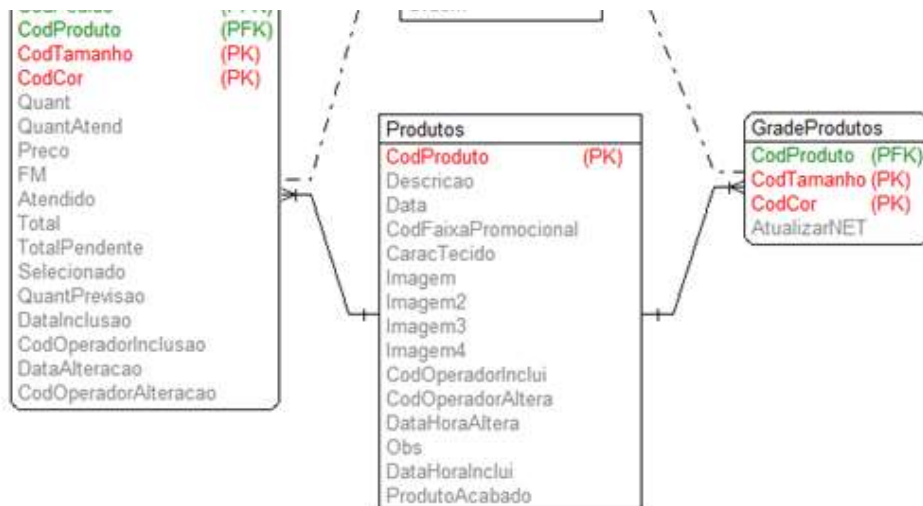


Figura 1: Modelagem das tabelas

Com base no modelo das tabelas acima, iremos destrinchar de forma parcelada o SELECT com o PIVOT, para que fique de fácil entendimento.

O código abaixo mostra o começo da declaração do nosso SCRIPT, e tamanho definimos a quantidade de colunas dinâmicas.

Listagem 3: Preparando o SCRIPT

```
DECLARE@SQLStrVARCHAR(5000)
SET@SQLStr=' '
SELECT@SQLStr=@SQLStr+'['+[a].[Column]+'], '
FROM
(SELECTDISTINCTCONVERT(VARCHAR(2),NomeTamanho)as[Column]
FROMTamanhos
)asa
SET@SQLStr=LEFT(@SQLStr,len(@SQLStr)-1)
```

Como podemos notar, o script inicial tem apenas uma declaração de uma variável do tipo VARCHAR.

Logo após é declarado um apelido para nossa tabela chamado de 'a' e em seguida é definido o nome da coluna, neste caso usamos o 'Column'.

Posteriormente é feito um SELECT na tabela TAMANHOS e assim podemos descobrir todos (nesse momento estamos trabalhando com o SELECT dinâmico, pois podemos ter 10 ou 100 tamanhos, isso não importa para o SCRIPT) os tamanhos que temos e com isso preparar as nossas colunas. Note que está

Foi definido um apelido para a tabela assim como acima o 'a'. Em seguida atribuímos o conteúdo acima para a nossa variável @SQLStr.

Neste momento chegamos ao ponto, onde podemos definir a nossa regra de negocio, ou seja, escolher as colunas que vão ser exibidas, escolher as clausulas, definirmos agrupamentos, etc. Não se prenda nas regras de negocio, elas estão ali apenas pra ilustrar o SCRIPT proposto com base na **Figura 1**.

Listagem 4: Definindo a regra de negocio do seu PIVOT e preenchendo as colunas

```
SET@SQLStr='SELECT Descricao, CodProduto, '
+@SQLStr
+' FROM (SELECT P.Descricao, P.CodProduto, T.NomeTamanho, SUM(IP.Quant) QTDETOTAL '+
' FROM Produtos P, GradeProdutos GP, Tamanhos T, ItensPedidos IP '+
' WHERE P.CodProduto = GP.CodProduto '+
' AND P.CodProduto = IP.CodProduto '+
' AND GP.CodTamanho = T.CodTamanho '+
' AND IP.CodTamanho = T.CodTamanho '+
' AND SUBSTRING(P.CodProduto, 1,2 ) = ''CL'' '+
' AND SUBSTRING(P.CodProduto, 5,3 ) = ''053'' '+
' AND YEAR(P.DATA) >= 2012 '+
' GROUP BY P.CodProduto, P.Descricao, T.NomeTamanho '+
' ) sq PIVOT (SUM(QTDETOTAL) FOR NomeTamanho IN( '
+@SQLStr+')) AS pt'
PRINT@SQLStr
EXEC(@SQLStr)
```

Receba notificações :)

Podemos notar que temos um SCRIPT simples, onde usamos os campos, Descricao, CodProduto, NomeTamanho e QTDETOTAL. As tabelas envolvidas, conforme a **Figura 1**, e posteriormente a instrução WHERE, com as condições de relacionamento, nele você pode fazer vários e vários tipos de filtro.

Assim como o PIVOT fixo, temos a mesma forma de implementar no nosso PIVOT dinâmico, a diferença é que não precisamos colocar as colunas fixas [1], [2], [3], [4], [5], [etc...], pois este trabalho fica a cargo da primeira parte do SCRIPT.

Quase no final, usamos o comando PRINT, apenas pra imprimir no box mensagem, o comando que foi executado, veja como ficou quando executamos em um banco de dados:

Listagem 5: Exemplo do comando Print

```

WHERE P.CodProduto = GP.CodProduto
AND P.CodProduto = IP.CodProduto
AND GP.CodTamanho = T.CodTamanho
AND IP.CodTamanho = T.CodTamanho
AND SUBSTRING(P.CodProduto, 1,2 ) = 'CL'
AND SUBSTRING(P.CodProduto, 5,3 ) = '053'
AND YEAR(P.DATA) >= 2012
GROUP BY P.CodProduto, P.Descricao, T.NomeTamanho ) sq
PIVOT (SUM(QTDETOTAL) FOR NomeTamanho IN ([G ], [GG], [M ], [P ], [PP], [RN], [UN], [XG])) AS pt

```

Agora finalizando o nosso SCRIPT, temos o comando que executa a nossa consulta, o EXEC, e com isso fechamos o SCRIPT do PIVOT, veja o resultado da consulta:

Descricao	CodProduto	G	GG	M	P	PP	RN	UN	XG
BATA	NN1405302	27	NULL	25	NULL	25	NULL	NULL	NULL
CALCA LINHO	NN2105304	30	NULL	28	NULL	NULL	NULL	NULL	NULL
CASACO MOLETON	NN3205313	24	33	24	NULL	NULL	NULL	NULL	33
BLUSA BASICA	NN3705309	29	NULL	26	25	NULL	NULL	NULL	NUL
BLUSA BASICA MG	NN3705314	29	33	25	25	NULL	NULL	NULL	NUL
BLUSA LACO	NN3805304	32	NULL	31	NULL	NULL	NULL	33	NUL
BLUSA LISTRADA	NN3805306	31	NULL	30	NULL	54	NULL	NULL	NUL
BLUSA MG	NN3805308	31	NULL	28	NULL	NULL	NULL	NULL	55
VESTIDO VERMELHO	NN4205302	34	NULL	35	NULL	NULL	NULL	NULL	NULL
BLUSA BOTAO	NN5105314	35	11	33	NULL	NULL	11	22	NULL
SAIA ESTAMPADAS	NN5205309	32	NULL	31	NULL	NULL	NULL	NULL	NULL
BLUSA XADREZ	NN5705321	30	NULL	27	25	NULL	NULL	22	NULL

Receba notificações :)

Usar o recurso de operador PIVOT em um primeiro momento pode assustar, porém, como podemos notar ele é um poderoso recurso e não é tão complexo quanto parece, agora basta adaptar o exemplo para a sua realidade. Use a sua imaginação e as suas consultas ficarão perfeitas.

Receba nossas novidades

Informe o seu e-mail...

Receber Newsletter!

por **Wesley Yamazack**



(17)



(1)

Ficou com alguma dúvida?

Receba notificações :)

ASSINE

FALE CONOSCO

APP

Hospedagem web por Porta 80 Web Hosting