

Início

[Acesso A Dados](#)[Arquitetura](#)[Automação](#)[Banco De Dados](#)[Controle De Versão](#)[Desenvolvimento](#)[Infraestrutura](#)[Metodologias](#)[Mobile](#)[Serviços](#)[Testes](#)[Web](#)[Windows](#)Você Está Em : [Home](#) » [Transact-SQL](#) » Transformando Linhas Em Colunas No SQL Server: Utilizando O Operador PIVOT

Transformando linhas em colunas no SQL Server: utilizando o operador PIVOT

👤 Renato Groffe 🕒 25/04/2014 📁 Transact-SQL 💬 2 Comentários



Os bancos de dados relacionais representam, sem sombra de dúvidas, um mecanismo de fundamental importância nas organizações. Projetados para o armazenamento de grandes volumes de informações num formato estruturado, estes repositórios fornecem meios para a condução das operações cotidianas nos mais variados segmentos. A partir de bases relacionais é possível também a geração de diversas análises sob a forma de relatórios, sendo este um importante instrumento no controle e gerenciamento das atividades dentro de uma companhia.

A construção de relatórios implica muitas vezes na necessidade de transformar os dados provenientes de tabelas relacionais, viabilizando com isto a obtenção de informações que atendam a necessidades específicas de um grupo de usuários. Um tipo de modificação comum neste sentido consiste em se inverter a disposição de linhas e colunas obtidas através do retorno de uma query: na prática acontecerá a geração de novas colunas a partir de um campo que corresponda a uma chave, com o agrupamento de valores levando em consideração este fator.

Considerando especificamente o SQL Server, este gerenciador de banco de dados disponibiliza o operador PIVOT para transformações desse gênero. O

Encontre Um Artigo

Últimos Artigos

[C# 7.0: novos Expression-bodied Members](#)[Dicas sobre .NET: Classes Sealed](#)[Validando dados com Fluent Validation](#)[Dicas sobre .NET: convertendo JSON para XML \(e vice-versa\)](#)[Dicas sobre .NET: consultas LINQ em objetos compostos](#)[Acesso a dados com Dapper .NET](#)[Swagger – Documente seu ASP.NET Web API Rest](#)[Dicas sobre .NET: utilizando a classe Stopwatch](#)[Message Broker com RabbitMQ](#)[Cache Distribuído & Session State com Redis NoSQL Server](#)[JS Unit Testing – Utilizando Chutzpah Test Runner com Jasmine JS no Visual Studio](#)[Js Advice – Analisador de código JavaScript](#)[Exploratory Testing \(Preview\) – Extensão Google Chrome](#)[Executando .NET Core Apps no Docker](#)[Criando Web.Config Seguros com ProtectedConfigurationProvider](#)

objetivo deste post é justamente demonstrar como o comando PIVOT pode ser utilizado na conversão de linhas em colunas, auxiliando assim desenvolvedores que façam uso deste produto da Microsoft no dia-a-dia.

Exemplo de utilização do operador PIVOT

O exemplo apresentado a seguir fará uso de uma tabela chamada "CotacoesPorDataMoeda". Essa estrutura armazenará os valores das cotações de moedas estrangeiras (dólar norte-americano, euro e libra esterlina) numa determinada data. Na **Listagem 1** está o script para a criação e inclusão de informações na tabela CotacoesPorDataMoeda.

```
CREATE TABLE [dbo].[CotacoesPorDataMoeda](
    [DataCotacao] [date] NOT NULL,
    [CodMoeda] [varchar](3) NOT NULL,
    [ValorCotacao] [numeric](18, 4) NOT
NULL,
    PRIMARY KEY ([DataCotacao], [CodMoeda])
)
GO

INSERT INTO
dbo.CotacoesPorDataMoeda(DataCotacao, CodMoeda,
ValorCotacao)
VALUES('2014-04-17', 'USD', 2.2357)

INSERT INTO
dbo.CotacoesPorDataMoeda(DataCotacao, CodMoeda,
ValorCotacao)
VALUES('2014-04-16', 'USD', 2.2418)

INSERT INTO
dbo.CotacoesPorDataMoeda(DataCotacao, CodMoeda,
ValorCotacao)
VALUES('2014-04-15', 'USD', 2.2385)

INSERT INTO
dbo.CotacoesPorDataMoeda(DataCotacao, CodMoeda,
ValorCotacao)
VALUES('2014-04-14', 'USD', 2.2147)

INSERT INTO
dbo.CotacoesPorDataMoeda(DataCotacao, CodMoeda,
ValorCotacao)
VALUES('2014-04-17', 'EUR', 3.0927)

INSERT INTO
dbo.CotacoesPorDataMoeda(DataCotacao, CodMoeda,
ValorCotacao)
VALUES('2014-04-16', 'EUR', 3.1012)

INSERT INTO
dbo.CotacoesPorDataMoeda(DataCotacao, CodMoeda,
ValorCotacao)
```

Tags

.net .NET Core .NET Native
AForge.NET Android SDK Angular AOP
apps híbridos arquitetura
Asp.net **ASP.NET 5**
ASP.NET Core ASP.NET
IDENTITY ASP.NET MVC 5
ASP.NET MVC 6
ASP.NET SIGNALR **ASP.NET**
Web API ASP.NET WEB API
2.2 Automação Backup BACKup
Hyper-V Banco de Dados BDD Bower
BPM BPMN Business
Intelligence **C#** C# 5.0
C# 6.0 C# 7 C# 7.0 Cache
Captcha Client-side **Code**
Analysis Code Cracker Code
First Code Metrics Code Smell Code
Snippets Computação Distribuída
Cordova CQRS CSS Data
Binding DDD design pattern
design patterns DNVM
DNX Engenharia Reversa **Entity**
Framework 6 Ferramentas Fiddler
Foreign Key Free Formatter FxCop
GAC Git **GitHub** Google Charts Google
Play Gulp GZip HtmlAgilityPack
Hyper-V ICommand ILSPY
INotifyPropertyChanged IOC **ionic**
framework iTextSharp
JavaScript JQuery **JSON**
JSON2CSHARP lambda Linux
Log4Net Microsoft **Microsoft Test**
Manager Mobile Mocking Moq
MVVM NDepend npm **Nuget** OCR
OWIN Parse Notification PicResize
PMBOK PostSharp **Primary Key** Push
Notification Refactoring Reporting
Services **REST** **RESTful**
RestSharp Scrum Selenium
Serviços SharpZipLib Smart Unit Test
SOA **solid** SPA SpecFlow SQL
Server **SQL Server**

```
VALUES('2014-04-15', 'EUR', 3.0874)

INSERT INTO
dbo.CotacoesPorDataMoeda(DataCotacao, CodMoeda,
ValorCotacao)
VALUES('2014-04-14', 'EUR', 3.0616)

INSERT INTO
dbo.CotacoesPorDataMoeda(DataCotacao, CodMoeda,
ValorCotacao)
VALUES('2014-04-17', 'LIB', 3.7593)

INSERT INTO
dbo.CotacoesPorDataMoeda(DataCotacao, CodMoeda,
ValorCotacao)
VALUES('2014-04-16', 'LIB', 3.7708)

INSERT INTO
dbo.CotacoesPorDataMoeda(DataCotacao, CodMoeda,
ValorCotacao)
VALUES('2014-04-15', 'LIB', 3.7400)

INSERT INTO
dbo.CotacoesPorDataMoeda(DataCotacao, CodMoeda,
ValorCotacao)
VALUES('2014-04-14', 'LIB', 3.7048)

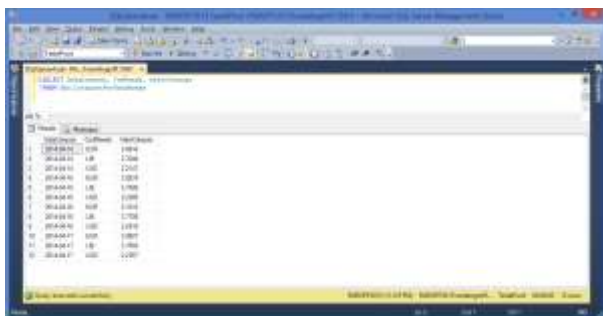
GO
```

Listagem 1: Scripts para criação e preenchimento da tabelaCotacoesPorDataMoeda

Executando uma consulta como a que consta na **Listagem 2** será possível observar as diferentes cotações, com tais dados estando agrupados por data e código da moeda (**Imagem 1**).

```
SELECT DataCotacao, CodMoeda, ValorCotacao
FROM dbo.CotacoesPorDataMoeda
```

Listagem 2: Efetuando uma consulta à tabela CotacoesPorDataMoeda



DataCotacao	CodMoeda	ValorCotacao
2014-04-14	EUR	3.0616
2014-04-15	EUR	3.0874
2014-04-16	LIB	3.7708
2014-04-17	LIB	3.7593
2014-04-15	LIB	3.7400
2014-04-14	LIB	3.7048

Imagem 1. Resultado de uma consulta à tabela CotacoesPorDataMoeda

2012 SQL Server Profiler

StyleCop Tag Helpers TDD Team
Foundation Server Tesseract

Transact-SQL Try/Catch

Ubuntu Universal Apps UWP

Visual Studio Visual Studio

"15" Visual

Studio 2015 Visual

Studio Code Visual Studio Online WCF

Web Web API web crawler Web
Robot Web Scraping Web Services Web

Spider Windows 10 Windows
Authentication Windows Backup

Windows Phone Windows SDK
Windows Server 2012R2 WinTask wpf

xamarin Xamarin Forms

Xaml XML XUnit

Redes Sociais

História do Grupo
Palestras e Mini Cursos
Linkedin
Facebook

Patrocinadores



Parceiros

O uso do operador PIVOT permitirá reagrupar estas informações, com a geração de colunas para cada tipo de moeda. Esses campos seriam então preenchidos com valores correspondentes à cotação numa determinada data. A **Listagem 3** apresenta o código que possibilitará tal transformação; já na **Imagem 2** está o resultado da execução desta expressão SQL.



Conforme é possível observar, a utilização do operador PIVOT envolve:



- Um Sub-SELECT, com os dados que deverão ser transformados (no caso, um SELECT simples consultando dados da tabela CotacoesPorDataMoeda);
- Uma seção formada pelo comando PIVOT, em que constará primeiramente uma expressão de agrupamento (para este exemplo foi utilizada a função AVG para obtenção da cotação média, mas também seria possível o uso de SUM, MAX, MIN). Além disso, nota-se a presença da cláusula FOR com o nome da coluna a ser transformado (no caso o campo CodMoeda), assim como os valores desta última que servirão de base para a geração de novas colunas (para campos do tipo VARCHAR, não utilizar aspas simples; uma prática comum é representar os valores entre colchetes, de forma a permitir a geração de colunas cujo nome também possa conter espaços);
- Por fim, no SELECT principal desta instrução SQL estão os campos de data de cotação, bem como as novas colunas representando os valores de moedas transformados por meio do operador PIVOT (estas três colunas foram renomeadas através aliases indicados pela cláusula AS).



Vídeo Aulas

[ComunidadeNetCoders](#)

9 vídeos

Inscriver-se

724

[eXcript](#)

581 vídeos

Inscriver-se

999+

```
SELECT DataCotacao,  
       [USD] AS VlDolar,  
       [EUR] AS VlEuro,  
       [LIB] AS VlLibra  
FROM  
(  
    SELECT DataCotacao, CodMoeda, ValorCotacao  
    FROM dbo.CotacoesPorDataMoeda  
) C  
PIVOT  
(  
    AVG(ValorCotacao)  
    FOR CodMoeda IN ([USD], [EUR], [LIB])  
) AS P
```

Listagem 3: Exemplo de utilização do operador PIVOT

0 comentários

Classificar por

Mais antigos



Adicionar um comentário...

[Plugin de comentários do Facebook](#)

No related posts.

SQL Server 2012, Transact-SQL

Related Posts

**Primary e Foreign Keys –
Parte II****Ferramentas Online para
Desenvolvedores****Primary e Foreign Keys –
Parte I****Introdução a Banco de
Dados**

2 Comments Already

[Subscribe to comments feed](#)**samuelcazelli** - abril 28th, 2014 at 09:53

Legal, gostei.



Alan - março 10th, 2016 at 13:38

Obrigado! Foi muito útil

Deixe uma resposta

O seu endereço de e-mail não será publicado. Campos obrigatórios são marcados com *

Comentário

Nome *

E-mail *

Site

Publicar comentário

« SQL Server Reporting Services – Dicas:
quebras de página e geração de
documentos PDF

Transformando colunas em linhas no SQL
Server: utilizando o operador UNPIVOT »

Copyright © 2017. .NET Coders