

SQL From Hell.com

Um nome bem intuitivo para muita informação!

Transformando linhas em colunas com PIVOT

Publicado em 18/07/2013 por Paulo R. Pereira

Olá pessoas,

A mágica de transformar linhas em colunas no SQL Server com PIVOT não é tão simples, o que requer um pouco de domínio de T-SQL, para saber como utiliza-la, além de não ser pego por alguns erros comuns.

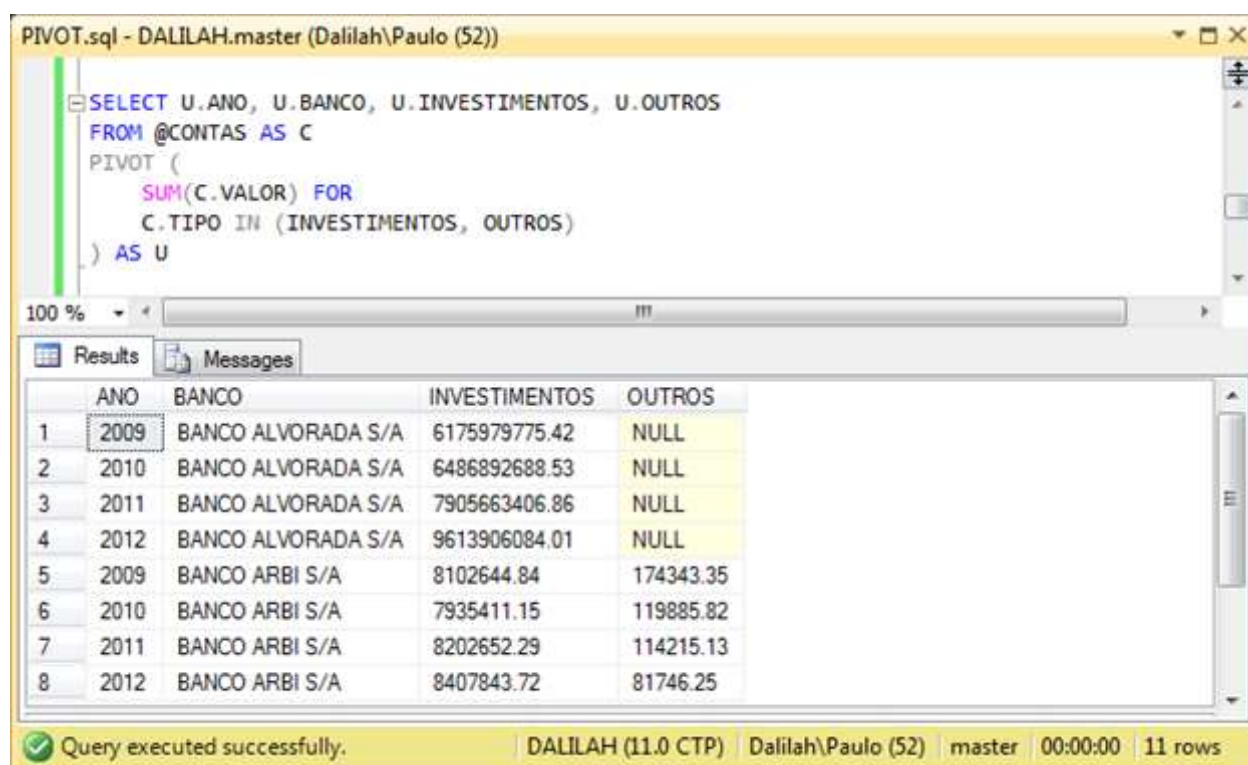
Neste primeiro artigo, vou apresentar alguns exemplos simples, e apresentar um problema comum de se trabalhar com PIVOT.

De início, temos uma massa de dados:

```
DECLARE @CONTAS TABLE (  
    ANO SMALLINT,  
    BANCO VARCHAR(100),  
    TIPO VARCHAR(100),  
    VALOR MONEY  
)  
  
INSERT INTO @CONTAS VALUES  
(2009, 'BANCO ALVORADA S/A', 'INVESTIMENTOS', 6175979775.42),  
(2010, 'BANCO ALVORADA S/A', 'INVESTIMENTOS', 6486892688.53),  
(2011, 'BANCO ALVORADA S/A', 'INVESTIMENTOS', 7905663406.86),  
(2012, 'BANCO ALVORADA S/A', 'INVESTIMENTOS', 9613906084.01),  
(2009, 'BANCO ARBI S/A', 'INVESTIMENTOS', 8102644.84),  
(2009, 'BANCO ARBI S/A', 'OUTROS', 174343.35),  
(2010, 'BANCO ARBI S/A', 'INVESTIMENTOS', 7935411.15),  
(2010, 'BANCO ARBI S/A', 'OUTROS', 119885.82),  
(2011, 'BANCO ARBI S/A', 'INVESTIMENTOS', 8202652.29),  
(2011, 'BANCO ARBI S/A', 'OUTROS', 114215.13),  
(2012, 'BANCO ARBI S/A', 'INVESTIMENTOS', 8407843.72),  
(2012, 'BANCO ARBI S/A', 'OUTROS', 81746.25)  
  
SELECT ANO, BANCO, TIPO, VALOR FROM @CONTAS
```

Para criar transformar as linhas de contas do tipo 'INVESTIMENTOS' e 'OUTROS' em linhas com uma coluna com os valores respectivos a 'INVESTIMENTOS' e outra a 'OUTROS', utilizaremos a expressão PIVOT sobre a tabela @CONTAS, somando os valores da coluna 'VALOR' para onde a coluna 'TIPO' apresenta os valores 'INVESTIMENTOS' e 'OUTROS', assim temos a primeira consulta e resultado:

```
SELECT U.ANO, U.BANCO, U.INVESTIMENTOS, U.OUTROS
FROM @CONTAS AS C
PIVOT (
    SUM(VALOR) FOR
    TIPO IN (INVESTIMENTOS, OUTROS)
) AS U
```



The screenshot shows a SQL query window titled 'PIVOT.sql - DALILAH.master (Dalilah\Paulo (52))'. The query is a PIVOT statement that transforms rows from the @CONTAS table into columns based on the TIPO (INVESTIMENTOS, OUTROS). The results are displayed in a table with 5 columns: ANO, BANCO, INVESTIMENTOS, and OUTROS. The data is grouped by ANO and BANCO. The status bar at the bottom indicates 'Query executed successfully.' and 'DALILAH (11.0 CTP) Dalilah\Paulo (52) master 00:00:00 11 rows'.

	ANO	BANCO	INVESTIMENTOS	OUTROS
1	2009	BANCO ALVORADA S/A	6175979775.42	NULL
2	2010	BANCO ALVORADA S/A	6486892688.53	NULL
3	2011	BANCO ALVORADA S/A	7905663406.86	NULL
4	2012	BANCO ALVORADA S/A	9613906084.01	NULL
5	2009	BANCO ARBI S/A	8102644.84	174343.35
6	2010	BANCO ARBI S/A	7935411.15	119885.82
7	2011	BANCO ARBI S/A	8202652.29	114215.13
8	2012	BANCO ARBI S/A	8407843.72	81746.25

Desta forma, os registros da tabela @CONTAS se transformam na PIVOT com o alias/apelido U (pode ser outra alias, sem problemas), assim as colunas TIPO e VALOR deixaram de existir, e são agrupadas por ANO e BANCO nas colunas INVESTIMENTOS e OUTROS.

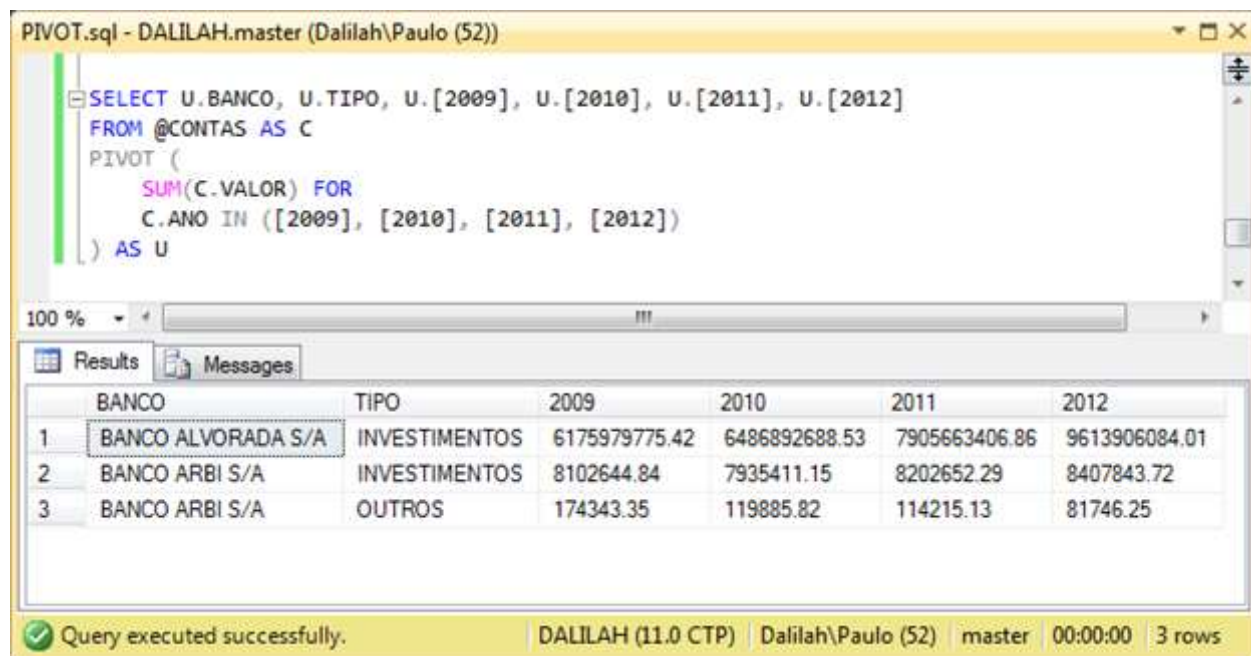
Obs.: Não é obrigatório especificar as colunas no SELECT, pois ele só reconhecerá as colunas do alias U, mas especifiquem para ficar organizado.

Abaixo, um exemplo utilizando a coluna VALOR agrupada por BANCO e TIPO para cada um dos anos da coluna ANO:

```

SELECT U.BANCO, U.TIPO, U.[2009], U.[2010], U.[2011], U.[2012]
FROM @CONTAS AS C
PIVOT (
    SUM(C.VALOR) FOR
    C.ANO IN ([2009], [2010], [2011], [2012])
) AS U

```



	BANCO	TIPO	2009	2010	2011	2012
1	BANCO ALVORADA S/A	INVESTIMENTOS	6175979775.42	6486892688.53	7905663406.86	9613906084.01
2	BANCO ARBI S/A	INVESTIMENTOS	8102644.84	7935411.15	8202652.29	8407843.72
3	BANCO ARBI S/A	OUTROS	174343.35	119885.82	114215.13	81746.25

Query executed successfully. DALILAH (11.0 CTP) Dalilah\Paulo (52) master 00:00:00 3 rows

Uma preocupação que se deve ter ao utilizar PIVOT é especificar claramente as colunas que serão agrupadas e quais as colunas que não serão, pois o SQL Server ainda não lê pensamentos, o que gera um erro bem comum, como apresentado abaixo.

“Então, estou com aquele script de PIVOT (A), mas quando removi a coluna ANO, para agrupar pelas colunas BANCO, INVESTIMENTOS e OUTROS, mas não deu certo, começou a aparecer duplicado a coluna BANCO (B).”

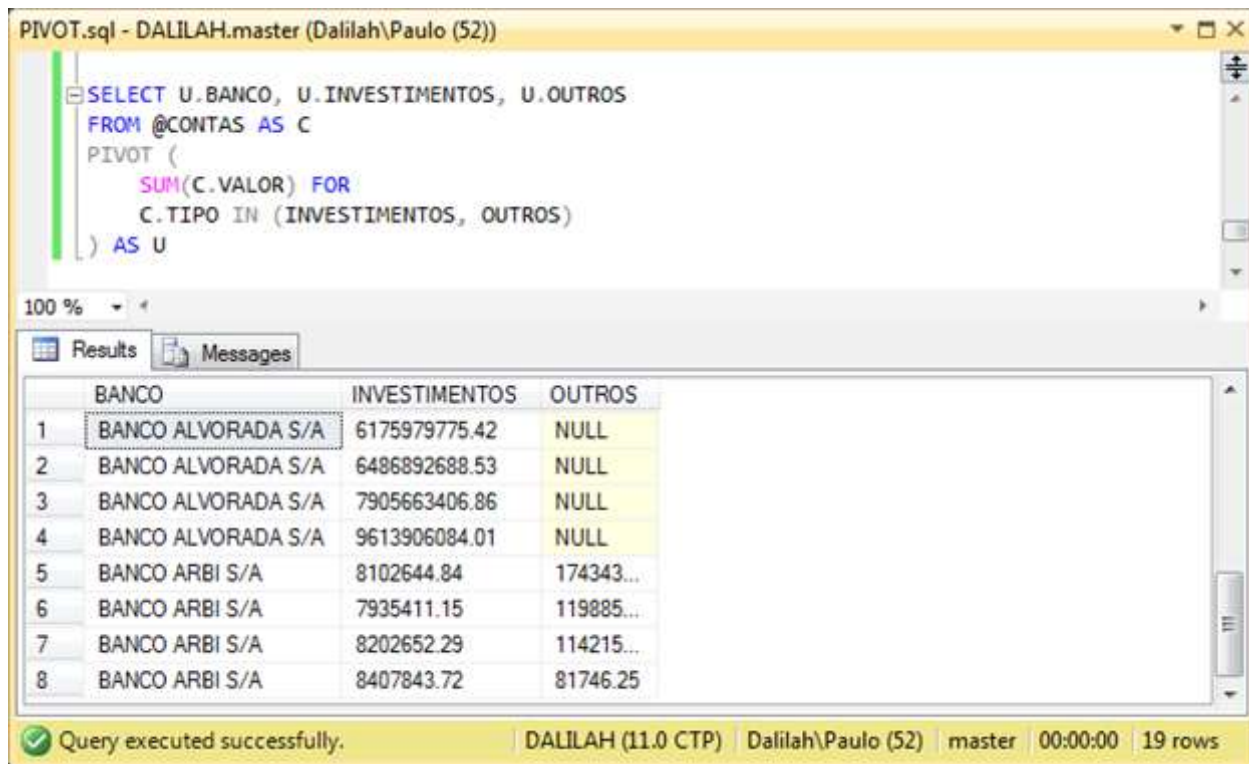
(A):

```

SELECT U.BANCO, U.INVESTIMENTOS, U.OUTROS
FROM @CONTAS AS C
PIVOT (
    SUM(VALOR) FOR
    TIPO IN (INVESTIMENTOS, OUTROS)
) AS U

```

(B):



The screenshot shows a SQL Server window titled "PIVOT.sql - DALILAH.master (Dalilah\Paulo (52))". The query editor contains the following SQL code:

```
SELECT U.BANCO, U.INVESTIMENTOS, U.OUTROS
FROM @CONTAS AS C
PIVOT (
    SUM(C.VALOR) FOR
    C.TIPO IN (INVESTIMENTOS, OUTROS)
) AS U
```

The "Results" tab is active, displaying the following data:

	BANCO	INVESTIMENTOS	OUTROS
1	BANCO ALVORADA S/A	6175979775.42	NULL
2	BANCO ALVORADA S/A	6486892688.53	NULL
3	BANCO ALVORADA S/A	7905663406.86	NULL
4	BANCO ALVORADA S/A	9613906084.01	NULL
5	BANCO ARBI S/A	8102644.84	174343...
6	BANCO ARBI S/A	7935411.15	119885...
7	BANCO ARBI S/A	8202652.29	114215...
8	BANCO ARBI S/A	8407843.72	81746.25

The status bar at the bottom indicates: "Query executed successfully. DALILAH (11.0 CTP) Dalilah\Paulo (52) master 00:00:00 19 rows".

Este problema se dá ao fato, que independente de você tirar as colunas especificadas no SELECT, a PIVOT continuará sendo feita sobre todas as colunas da tabela, ou seja, tirou a coluna 'ANO' do SELECT, não quer dizer que a PIVOT vai esquecer que a coluna 'ANO' existe.

Para evitar este problema, ao invés de utilizar diretamente a tabela para a PIVOT, especifique em uma subconsulta, quais colunas a PIVOT utilizará, exemplo:

```
SELECT U.BANCO, U.INVESTIMENTOS, U.OUTROS
FROM (
    SELECT BANCO, TIPO, VALOR
    FROM @CONTAS
) AS C
PIVOT (
    SUM(C.VALOR) FOR
    C.TIPO IN (INVESTIMENTOS, OUTROS)
) AS U
```



The screenshot shows a SQL Server query window titled "PIVOT.sql - DALILAH.master (Dalilah\Paulo (52))". The query is as follows:

```
SELECT U.BANCO, U.INVESTIMENTOS, U.OUTROS
FROM (SELECT BANCO, TIPO, VALOR FROM @CONTAS) AS C
PIVOT (
    SUM(C.VALOR) FOR
    C.TIPO IN (INVESTIMENTOS, OUTROS)
) AS U
```

The query results are displayed in a table with the following data:

	BANCO	INVESTIMENTOS	OUTROS
1	BANCO ALVORADA S/A	30182441954.82	NULL
2	BANCO ARBI S/A	32648552.00	490190.55

The status bar at the bottom indicates: "Query executed successfully. DALILAH (11.0 CTP) Dalilah\Paulo (52) master 00:00:00 2 rows".

Próximas semanas, eu vou apresentar o UNPIVOT e também ensinar as limitações destas cláusulas e mágicas, assim como explicar como fazer as mesmas coisas sem utilizar PIVOT e UNPIVOT.

Anúncios

Esse post foi publicado em [Introdução](#), [Virtual PASS BR](#) e marcado [PIVOT](#), [Pivot Table](#), [SQL Server 2005](#). Guardar [link permanente](#).

Um pensamento sobre “Transformando linhas em colunas com PIVOT”

Pingback: [Transformando colunas em linhas sem UNPIVOT | SQL From Hell.com](https://sqlfromhell.wordpress.com/2013/07/18/transformando-colunas-em-linhas-sem-UNPIVOT/)