

Consultando dados além do SELECT

Comandos que todo DBA deveria
ensinar aos desenvolvedores

Nossos Patrocinadores

DELL EMC

TmaxSoft
Brasil



STROHL
Brasil

 **Timbira**
A empresa brasileira de PostgreSQL

GREEN
tecnologia

DBA4All
We care about your data

 **Academy**

 **ORAMASTER**

Apresentação Pessoal

- Ignez Mello
- Dinossaura de CPD (em TI desde 1988)
 - Dbase, Clipper, VO, VB, SQL Server 6.5, Erwin, etc...
- Especialista em SQL Server
 - Curiosa com Oracle, MySQL, Caché, MongoDB, Cassandra...
- Contatos
 - www.facebook.com/ignez.mello.3
 - www.linkedin.com/in/ignezmello
 - ignez.mello@outlook.com

Agenda

- CTE
- MERGE
- PIVOT e UNPIVOT
- Homework:
 - CROSS e OUTER APPLY
 - OUTPUT e INTO
 - OFFSET e FETCH
 - OVER

CTE

```
;WITH expressao_cte ([coluna1, coluna2, coluna4...])
```

```
AS
```

```
(SELECT ...),
```

```
outra_expressao_cte ([colunaA, colunaB, ...])
```

```
AS
```

```
(SELECT ...),
```

```
outra_expressao_cte ([coluna1, colunaC, ...])
```

```
AS
```

```
(SELECT ...)
```

Comando (Select, Insert, Update, Delete, Merge)

MERGE

```
MERGE #Tabela1 AS Destino  
USING #Tabela2 AS Origem  
ON Destino.ColunaChave = Origem.ChaveColuna
```

-- Há registro no destino e na origem

```
WHEN MATCHED THEN  
UPDATE SET Ativo = 0, Alteracao = GETDATE()
```

-- Não há registro no destino e há na origem

```
WHEN NOT MATCHED THEN  
INSERT (Coluna_A, Coluna_B, Coluna_C)  
VALUES ('Valor A', 999, Getdate())
```

-- Há registro no destino mas não há na origem e que sejam diferente de Valor

```
WHEN NOT MATCHED BY origem AND Coluna_A <> 'Valor' THEN  
UPDATE SET Ativo = NULL, Alteracao = GETDATE();
```

PIVOT

```
SELECT <coluna não pivot>,  
    [primeira coluna pivot] AS <coluna>,  
    [segunda coluna pivot] AS <coluna>,  
    ...  
    [última coluna pivot] AS <coluna>  
  
FROM (<SELECT query que produz os dados>) AS <alias da query origem>  
  
PIVOT (<função de agregação>(<coluna sendo agregada>)  
    FOR [<coluna que contem o valor que será o cabeçalho das colunas>]  
    IN ([valor1], [valor2],... [valorN]) ) AS <alias para a tabela pivot>  
  
<ORDER BY>;
```


UNPIVOT

```
SELECT <coluna nao pivot>,  
    [primeira coluna pivot] AS <coluna>, [segunda coluna pivot] AS <coluna>,  
    ... [ultima coluna pivot] AS <coluna>  
  
FROM (<SELECT query que produz os dados>) AS <alias da query origem>  
  
UNPIVOT (<funcao de agregacao>(<coluna sendo agregada>)  
    FOR [<coluna com o valor que será o cabeçalho das colunas>]  
    IN ([valor1], [valor2],... [valorN])) AS <alias para a tabela unpivot>  
  
<ORDER BY>;
```


APPLY – CROSS e OUTER

```
SELECT TA.Coluna_A, TA.Coluna_B, ..., TC.Coluna_1
```

```
FROM Tabela_A AS TA
```

```
CROSS APPLY
```

```
(SELECT Coluna_1, ..., Coluna_N
```

```
FROM Tabela_B AS TB
```

```
WHERE TA.Coluna_A = TB.Coluna_B) AS TC
```

```
SELECT TA.Coluna_A, TA.Coluna_B, ..., TC.Coluna_1
```

```
FROM Tabela_A AS TA
```

```
OUTER APPLY
```

```
(SELECT Coluna_1, ..., Coluna_N
```

```
FROM Tabela_B AS TB
```

```
WHERE TA.Coluna_A = TB.Coluna_B) AS TC
```

OUTPUT e INTO

INSERT (ou UPDATE, DELETE, MERGE...)

OUTPUT inserted.Col_A, Col_B, ...

deleted.Col_A, Col_B, ...

INTO Tabela / #TabTemp

VALUES....

Ou

SELECT...

OFFSET e FETCH

SELECT coluna_A, coluna_B, ..., coluna_X

FROM Tabela

WHERE ...

ORDER BY...

OFFSET N ROWS

FETCH NEXT X ROWS ONLY

OVER

```
SELECT Coluna_A, Coluna_B, ... Coluna_X  
       ROW_NUMBER() OVER(ORDER BY Coluna_A asc) AS Linha_A  
       ROW_NUMBER() OVER(PARTITION BY Coluna_B  
                           ORDER BY Coluna_A asc) AS Linha_B,  
       SUM(Coluna_B) OVER(PARTITION BY Coluna_X) AS Total,  
       SUM(Coluna_C) OVER(PARTITION BY Coluna_B ORDER BY Coluna_A  
                           ROWS BETWEEN 3 PRECEDING AND CURRENT ROW) AS XXX  
       SUM(Coluna_C) OVER(PARTITION BY Coluna_B ORDER BY Coluna_A  
                           ROWS BETWEEN 3 CURRENT ROW AND 3 FOLLOWING)  
       SUM(Coluna_D) OVER(PARTITION BY Coluna_A ORDER BY Coluna_C  
                           ROWS UNBOUNDED PRECEDING)  
FROM Tabela
```



Dúvidas?!

Nossos Patrocinadores

DELL EMC

TmaxSoft
Brasil



STROHL
Brasil

 **Timbira**
A empresa brasileira de PostgreSQL

GREEN
tecnologia

DBA4All
We care about your data

 **Academy**

 **ORAMASTER**