

# INNER, CROSS, LEFT, RIGHT E FULL JOINS SIMPLES ASSIM!

Wellyngton Luiz Cruz Moreira  
wlcmoreira@hotmail.com

**PALAVRAS CHAVES:** inner join, left outer join, cross join, right outer join, full outer join, junções , SQL, junções passo a passo, script sql completo.

## PREPARANDO O AMBIÊNTE DE TRABALHO

As junções entre duas ou mais tabelas podem ser realizadas através de:

- Cross join
- Inner join
- Left outer join
- Right outer join
- Outer Full join

As questões são duas:

- O que são cada uma dessas junções (joins)?
- Como usar?

Para definir e exemplificar as junções acima citadas considere o modelo da figura 1 abaixo:

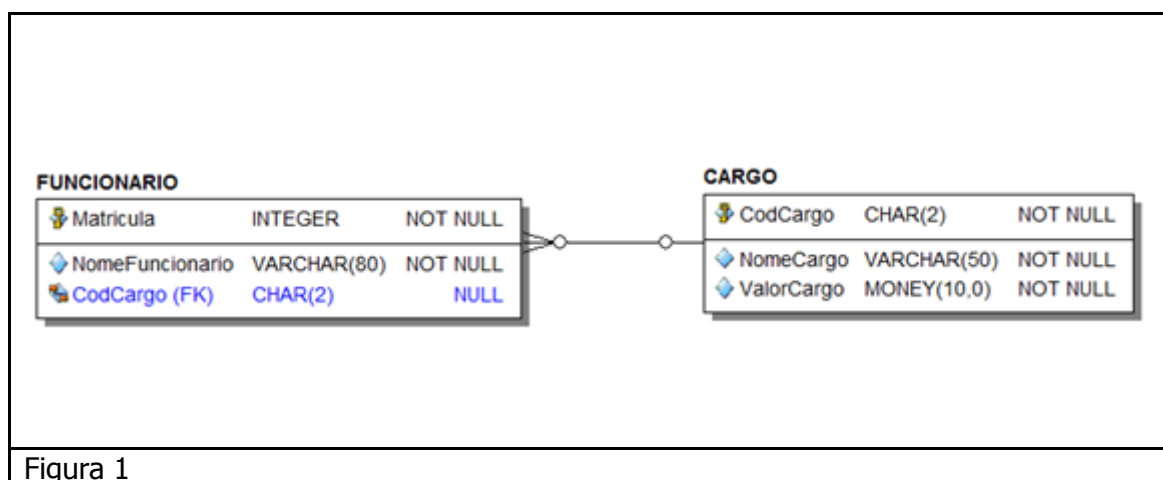


Figura 1

Podemos notar pelo modelo que pode existir no banco de dados funcionários sem cargos e cargos sem funcionários. Para exemplificar melhor, observe o conteúdo das tabelas na figura 2 abaixo:

TODAS AS LINHAS....bTestes (sa (75))

```

SELECT * FROM CARGO AS C --> Apelidamos a tabelas Cargo de C neste artigo
SELECT * FROM FUNCIONARIO AS F --> Apelidamos funcionário de F neste artigo
    
```

---

Results Messages

	CodCargo	NomeCargo	ValorCargo	
1	C1	CAIXA	800,00	<b>CARGO AS C</b>
2	C2	VENDEDOR	1200,00	
3	C3	GERENTE	2400,00	

---

	Matricula	NomeFuncionario	CodCargo	
1	100	JOÃO	C1	<b>FUNCIONARIO AS F</b>
2	110	MARIA	C2	
3	120	CARLOS	C1	
4	130	TADEU	NULL	

Figura 2

Assim, fazemos:

### CROSS JOIN

Quando queremos juntar duas ou mais tabelas por cruzamento. Ou seja, para cada linha da tabela FUNCIONARIO queremos todos os CARGOS ou vice-versa.

CROSS JOIN.sql - ...bTestes (sa (192))

```

SELECT F.NomeFuncionario
      ,C.NomeCargo
FROM      CARGO      AS C
CROSS JOIN FUNCIONARIO AS F
    
```

---

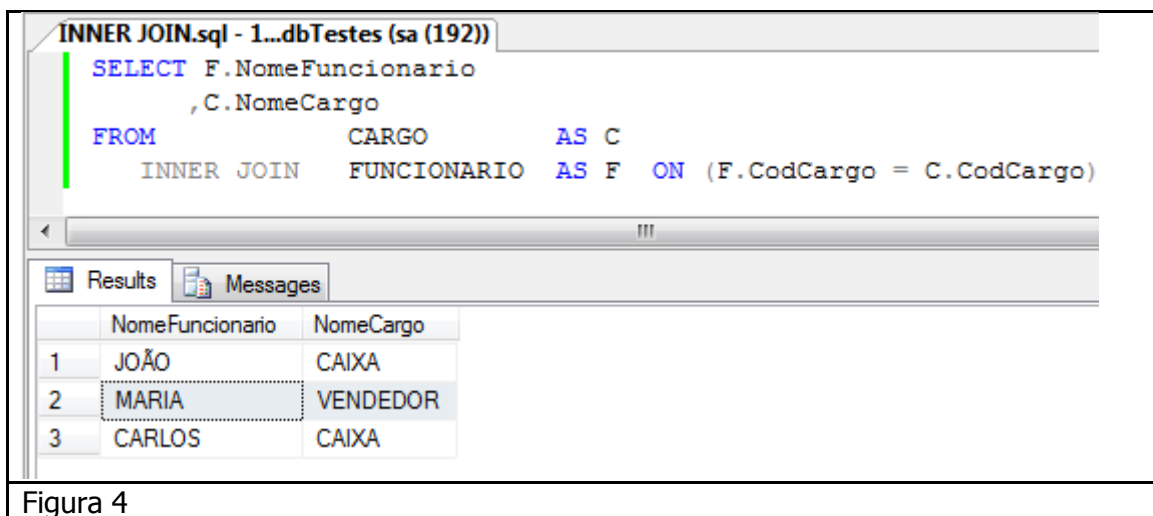
Results Messages

	NomeFuncionario	NomeCargo
1	JOÃO	CAIXA
2	JOÃO	VENDEDOR
3	JOÃO	GERENTE
4	MARIA	CAIXA
5	MARIA	VENDEDOR
6	MARIA	GERENTE
7	CARLOS	CAIXA
8	CARLOS	VENDEDOR
9	CARLOS	GERENTE

Figura 3

## INNER JOIN

Quando queremos juntar duas ou mais tabelas por coincidência. Para cada linha da tabela FUNCINARIO queremos o CARGO correspondente que internamente (INNER), em seus valores de atributos, coincidam. No caso de FUNIONÁRIO e CARGO os atributos internos coincidentes são codigoCargo na tabela **CARGO** e codigoCargo na tabela **FUNCIONARIO**. Veja também a Figura 1 e a Figura 2, lá você notará que codigoCargo é chave primária da tabela **CARGO** e chave estrangeira na tabela **FUNCIONARIO**. Para efetivarmos a junção das duas tabelas se fará necessário ligar (ON) as duas tabelas por seus atributos internos (INNER) coincidentes.



```
INNER JOIN.sql - 1...dbTestes (sa (192))
SELECT F.NomeFuncionario
      ,C.NomeCargo
FROM      CARGO          AS C
INNER JOIN FUNCIONARIO AS F ON (F.CodCargo = C.CodCargo)
```

	NomeFuncionario	NomeCargo
1	JOÃO	CAIXA
2	MARIA	VENDEDOR
3	CARLOS	CAIXA

Figura 4

## LEFT OUTER JOIN

Observando a tabela FUNCIONARIO notamos que o funcionário *Tadeu* não possui cargo associado a ele. Se desejarmos listar **todos os funcionários com seus respectivos cargos, incluindo os funcionários sem cargos**, a exemplo de Tadeu, poderíamos usar todo o poder da junção INNER JOIN adicionando ainda OUTER(EXTERNOS/OUTROS) Funcionários que não fazem parte do INNER JOIN, justamente àqueles sem cargos, como Tadeu. Podemos conseguir esse feito com a junção FUNCIONARIO/CARGO através da declaração **FUNCIONARIO OUTER LEFT JOIN CARGO**, que promove a junção interna (INNER) de todos os funcionários a cargos **e lista ainda** outros (EXTERNOS/OUTER) não associados.

OUTER LEFT JOIN...Testes (sa (232))\*

```

SELECT F.nomeFuncionario
      ,C.nomeCargo
FROM      FUNCIONARIO AS F
LEFT OUTER JOIN CARGO AS C ON ( F.codCargo = C.codCargo )

```

Results Messages

	nomeFuncionario	nomeCargo
1	JOÃO	CAIXA
2	MARIA	VENDEDOR
3	CARLOS	CAIXA
4	TADEU	NULL

Figura 5

Uma observação importante é que a ordem da ligação (ON) não faz diferença, ou seja: "ON (F.codCargo = C.codCargo)" é exatamente igual a "ON (C.codCargo = F.codCargo)"

## RIGHT OUTER JOIN

Observando a tabela CARGO notamos que o cargo GERENTE, com código C3, não é referenciado/associado por/a nenhum funcionário na tabela FUNCIONARIO. Se desejarmos listar todos os CARGOS e seus respectivos FUNCIONARIOS, incluindo os CARGOS sem FUNCIONÁRIOS, poderíamos usar a junção RIGTH OUTER JOIN. Veja o exemplo da figura 6.

OUTER RIGHT JOIN...Testes (sa (232))

```

SELECT F.nomeFuncionario
      ,C.nomeCargo
FROM      FUNCIONARIO AS F
RIGHT OUTER JOIN CARGO AS C ON ( F.codCargo = C.codCargo )

```

Results Messages

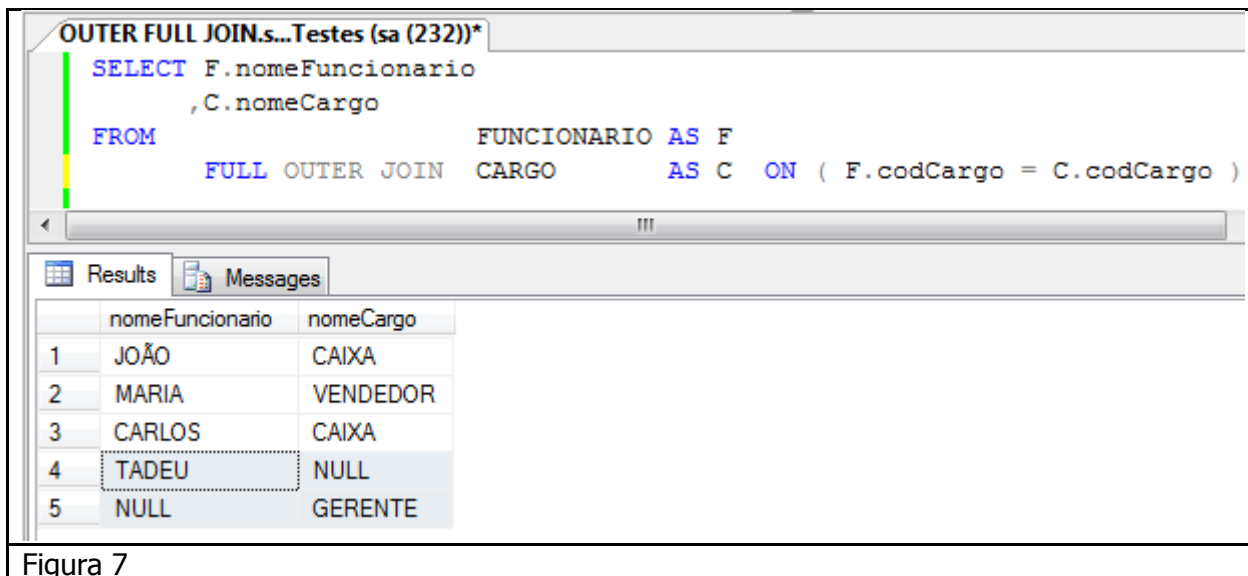
	nomeFuncionario	nomeCargo
1	JOÃO	CAIXA
2	CARLOS	CAIXA
3	MARIA	VENDEDOR
4	NULL	GERENTE

Figura 6

Uma observação importante é que a ordem da ligação (ON) não faz diferença, ou seja: "ON (F.codCargo = C.codCargo)" é exatamente igual a "ON (C.codCargo = F.codCargo)"

## OUTER FULL JOIN

Aqui juntamos o poder das junções (JOIN) internas (INNER), a listagem de todas as outras linhas não associadas, tanto do lado direito (RIGHT) da junção como do lado ESQUEDO (LEFT). Veja o exemplo abaixo:



The screenshot shows a SQL query window titled "OUTER FULL JOIN.s...Testes (sa (232))\*". The query is as follows:

```
SELECT F.nomeFuncionario
      ,C.nomeCargo
FROM      FUNCIONARIO AS F
FULL OUTER JOIN CARGO      AS C ON ( F.codCargo = C.codCargo )
```

Below the query, the "Results" tab is active, displaying a table with the following data:

	nomeFuncionario	nomeCargo
1	JOÃO	CAIXA
2	MARIA	VENDEDOR
3	CARLOS	CAIXA
4	TADEU	NULL
5	NULL	GERENTE

Figura 7

Uma observação importante é que a ordem da ligação (ON) não faz diferença, ou seja: "ON (F.codCargo = C.codCargo)" é exatamente igual a "ON (C.codCargo = F.codCargo)".

### **Simples assim!**

Abaixo segue o SCRIPT SQL de criação das tabelas, seus relacionamentos, seu povoamento e dos exemplos utilizados. Todos os exemplos deste artigo foram testados no SGBDR MS-SQL Server 2008 Express.

#### SCRIPT DE TESTES EM TRANSACT-SQL ( SQL SERVER )

```
-----
-- OS SCRIPTS ABAIXO FORAM DESENVOLVIDOS E TESTADOS
-- NO SGBD SQL SERVER 2008 EXPRESS
-- por Wellyngton Luiz Cruz Moreira
-----

-- Criando um dataBase para testes
CREATE DATABASE dbTESTE_JOINS
GO

-- Usando o DataBase para criar as tabelas e povoá-las
USE dbTESTE_JOINS
GO
```

```

-- Criando a tabela Cago
CREATE TABLE CARGO
(
    CodCargo char(2) NOT NULL,
    NomeCargo varchar(50) NULL,
    ValorCargo money NULL,
    PRIMARY KEY(CodCargo)
)

GO

-- Criando a tabela Funcionario
CREATE TABLE FUNCIONARIO
(
    Matricula int,
    NomeFuncionario varchar(50) NOT NULL,
    CodCargo char(2) NULL,
    PRIMARY KEY(Matricula),
    FOREIGN KEY (CodCargo) REFERENCES CARGO (CodCargo)
)

GO

-- Povoando a tabela CARGO
INSERT CARGO (CodCargo, NomeCargo, ValorCargo) VALUES ('C1', 'CAIXA'
, 800.00)
INSERT CARGO (CodCargo, NomeCargo, ValorCargo) VALUES ('C2',
'VENDEDOR', 1200.00)
INSERT CARGO (CodCargo, NomeCargo, ValorCargo) VALUES ('C3', 'GERENTE'
, 2400.00)

GO

-- Povoando a tabela FUNCIONARIO
INSERT FUNCIONARIO (Matricula, NomeFuncionario, CodCargo) VALUES (100,
'JOÃO' , 'C1')
INSERT FUNCIONARIO (Matricula, NomeFuncionario, CodCargo) VALUES (110,
'MARIA' , 'C2')
INSERT FUNCIONARIO (Matricula, NomeFuncionario, CodCargo) VALUES (120,
'CARLOS' , 'C1')
INSERT FUNCIONARIO (Matricula, NomeFuncionario, CodCargo) VALUES (130,
'TADEU' , NULL)

GO

-----
-- EXEMPLOS DE JOIN ABORDADOS NO ARTIGO
-----

-- Seleção Simples ( Todos os Cargos e Todos os Funcionario ) - Veja
Figura 2
SELECT * FROM CARGO AS C --> Apelidamos a tabelas Cargo de C
neste artigo
SELECT * FROM FUNCIONARIO AS F --> Apelidamos funcionário de F neste
artigo

GO

-- CROSS JOIN ( Junção Cruzada ) - Veja Figura 3
SELECT F.NomeFuncionario
, C.NomeCargo

```

```

FROM          CARGO          AS C
CROSS JOIN    FUNCIONARIO    AS F

-- INNER JOIN ( Junção Interna ) - Veja Figura 4
SELECT F.NomeFuncionario
      ,C.NomeCargo
FROM          CARGO          AS C
INNER JOIN    FUNCIONARIO    AS F ON (F.CodCargo = C.CodCargo)

-- LEFT OUTER JOIN ou simplesmente LEFT JOIN ( Junção Externa Esquerda
) - Veja Figura 5
SELECT F.nomeFuncionario
      ,C.nomeCargo
FROM          FUNCIONARIO AS F
LEFT OUTER JOIN CARGO      AS C ON ( C.codCargo = F.codCargo
)

-- RIGHT OUTER JOIN ou simplesmente RIGHT JOIN ( Junção Externa
Direita) - Veja Figura 6
SELECT F.nomeFuncionario
      ,C.nomeCargo
FROM          FUNCIONARIO AS F
RIGHT OUTER JOIN CARGO      AS C ON ( F.codCargo =
C.codCargo )

-- FULL OUTER JOIN ou simplesmente FULL JOIN (Junção Externa
Completa)
SELECT F.nomeFuncionario
      ,C.nomeCargo
FROM          FUNCIONARIO AS F
FULL OUTER JOIN CARGO      AS C ON ( C.codCargo = F.codCargo
)

-- É só isso!
-- Até a próxima e AVANTE!

```

Avante e até a próxima.

*Wellyngton Luiz Cruz Moreira*

[wlcmoreira@hotmail.com](mailto:wlcmoreira@hotmail.com)

<http://wlcmoreira.wordpress.com>