

Aula 10

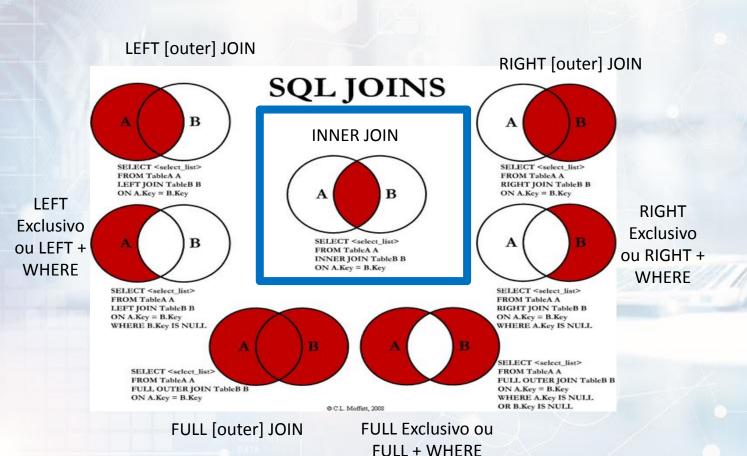
Data Query Language DQL - Continuação

Faculdade TANDA TANDA



# DQL – Tipos de Join







Para ilustrarmos ainda mais o conceito de junção entre mais de duas tabelas, veja o seguinte DER:





#### Gerando a extração com JOIN entre as tabelas:

#### Cor

# idCor descricao 1 Branco 2 Preto 3 Azul 4 Vermelho 5 Amarelo

#### Produto

idProduto	codigo	nome	estoque	descontinuado	idFornecedor	idCor
1	XT890A	Asus Zenfone	5	0	3	4
2	RQ765B	iPhone	0	1	9	4
3	WD528B	Moto X	3	0	2	7
4	TF897A	Xperia	7	0	1	1
5	RF212B	Moto Maxx	2	0	2	1

#### Fornecedor

idFornecedor	nome	telefone	contato
1	Sony	8498-8732	Allan
2	Motorola	7987-9900	Cristina
3	Asus	5476-1120	Felipe
4	Nokia	6755-5656	Fábio

**SELECT** P.Codigo, P.Nome **AS** 'Nome do Produto', P.Estoque

, F.Nome AS 'Nome do Fornecedor', F.Contato, F.Telefone

, C.Descricao AS 'Cor'

FROM Produto AS P JOIN Cor AS C

**ON** C.idCor = P.idCor

**JOIN** Fornecedor **AS** F **ON** F.idFornecedor = P.idFornecedor

Codigo	Nome do Produto	Estoque	Nome do Fornecedor	Contato	Telefone	Cor
XT890A	Asus Zenfone	5	Asus	Felipe	5476-1120	Vermelho
TF897A	Xperia	7	Sony	Allan	8498-8732	Branco
RF212B	Moto Maxx	2	Motorola	Cristina	7987-9900	Branco



#### Quando aplicamos as cláusulas LEFT JOIN ou RIGHT JOIN, percebam os resultados:

_	_	-	
•	•		
_	•		

	1/4					
idCor	descricao					
1	Branco					
2	Preto					
3	Azul					
4	Vermelho					
5	Amarelo					

#### Produto

idProduto	codigo	nome	estoque	descontinuado	idFornecedor	idCor
1	XT890A	Asus Zenfone	5	0	3	4
2	RQ765B	iPhone	0	1	9	4
3	WD528B	Moto X	3	0	2	7
4	TF897A	Xperia	7	0	1	1
5	RF212B	Moto Maxx	2	0	2	1

#### Fornecedor

idFornecedor	nome	telefone	contato	
1	Sony	8498-8732	Allan	
2	Motorola	7987-9900	Cristina	
3	Asus	5476-1120	Felipe	
4	Nokia	6755-5656	Fábio	

#### **SELECT** P.Codigo, P.Nome **AS** 'Nome do Produto', P.Estoque

, F.Nome AS 'Nome do Fornecedor', F.Contato, F.Telefone

, C.Descricao AS 'Cor'

#### **FROM**

Produto AS P JOIN Fornecedor AS F

**ON** P.idFornecedor = F.idFornecedor

**RIGHT JOIN** Cor **AS** C **ON** C.idCor = P.idCor

Codigo	Nome do Produto	Estoque	Nome do Fornecedor	Contato	Telefone	Cor
XT890A	Asus Zenfone	5	Asus	Felipe	5476-1120	Vermelho
TF897A	Xperia	7	Sony	Allan	8498-8732	Branco
RF212B	Moto Maxx	2	Motorola	Cristina	7987-9900	Branco
NULL	NULL	NULL	NULL	NULL	NULL	Preto
NULL	NULL	NULL	NULL	NULL	NULL	Azul
NULL	NULL	NULL	NULL	NULL	NULL	Amarelo



#### Trocando o RIGHT pelo LEFT, o resultado é alterado:

#### Cor

#### Produto

idCor	descricao
1	Branco
2	Preto
3	Azul
4	Vermelho
5	Amarelo

		2 31				
idProduto	codigo	nome	estoque	descontinuado	idFornecedor	idCor
1	XT890A	Asus Zenfone	5	0	3	4
2	RQ765B	iPhone	0	1	9	4
3	WD528B	Moto X	3	0	2	7
4	TF897A	Xperia	7	0	1	1
5	RF212B	Moto Maxx	2	0	2	1

#### Fornecedor

idFornecedor	nome	telefone	contato
1	Sony	8498-8732	Allan
2	Motorola	7987-9900	Cristina
3	Asus	5476-1120	Felipe
4	Nokia	6755-5656	Fábio

**SELECT** P.Codigo, P.Nome **AS** 'Nome do Produto', P.Estoque

, F.Nome **AS** 'Nome do Fornecedor', F.Contato, F.Telefone

, C.Descricao AS 'Cor'

FROM Produto AS P JOIN Fornecedor AS F

**ON** P.idFornecedor = F.idFornecedor

**LEFT JOIN** Cor **AS** C **ON** C.idCor = P.idCor

Codigo	Nome do Produto	Estoque	Nome do Fornecedor	Contato	Telefone	Cor
XT890A	Asus Zenfone	5	Asus	Felipe	5476-1120	Vermelho
WD528B	Moto X	3	Motorola	Cristina	7987-9900	NULL
TF897A	Xperia	7	Sony	Allan	8498-8732	Branco
RF212B	Moto Maxx	2	Motorola	Cristina	7987-9900	Branco



#### Se invertemos a ordem das junções, também teremos diferenças:

#### Cor

idCor	descricao	
1	Branco	
2	Preto	
3	Azul	
4	Vermelho	
5	Amarelo	

#### Produto

idProduto	codigo	nome	estoque	descontinuado	idFornecedor	idCor
1	XT890A	Asus Zenfone	5	0	3	4
2	RQ765B	iPhone	0	1	9	4
3	WD528B	Moto X	3	0	2	7
4	TF897A	Xperia	7	0	1	1
5	RF212B	Moto Maxx	2	0	2	1

#### Fornecedor

idFornecedor	nome	telefone	contato
1	Sony	8498-8732	Allan
2	Motorola	7987-9900	Cristina
3	Asus	5476-1120	Felipe
4	Nokia	6755-5656	Fábio

#### **SELECT** P.Codigo, P.Nome **AS** 'Nome do Produto', P.Estoque

, F.Nome AS 'Nome do Fornecedor', F.Contato, F.Telefone

, C.Descricao AS 'Cor'

#### **FROM** Produto AS P LEFT JOIN Fornecedor AS F ON P.idFornecedor = F.idFornecedor

JOIN Cor AS C ON C.idCor = P.idCor

Codigo	Nome do Produto	Estoque	Nome do Fornecedor	Contato	Telefone	Cor
XT890A	Asus Zenfone	5	Asus	Felipe	5476-1120	Vermelho
RQ765B	iPhone	0	NULL	NULL	NULL	Vermelho
TF897A	Xperia	7	Sony	Allan	8498-8732	Branco
RF212B	Moto Maxx	2	Motorola	Cristina	7987-9900	Branco



#### Novamente, trocando LEFT por RIGHT, obtemos resultados diferentes:

Cor					
idCor	descricao				
1	Branco				
2	Preto				
3	Azul				
4	Vermelho				
5	Amarala				

Produto								
idProduto	codigo	nome	estoque	descontinuado	idFornecedor	idCor		
1	XT890A	Asus Zenfone	5	0	3	4		
2	RQ765B	iPhone	0	1	9	4		
3	WD528B	Moto X	3	0	2	7		
4	TF897A	Xperia	7	0	1	1		
5	RF212B	Moto Maxx	2	0	2	1		

Fornecedor						
idFornecedor	nome	telefone	contato			
1	Sony	8498-8732	Allan			
2	Motorola	7987-9900	Cristina			
3	Asus	5476-1120	Felipe			
4	Nokia	6755-5656	Fábio			

**SELECT** P.Codigo, P.Nome **AS** 'Nome do Produto', P.Estoque

, F.Nome AS 'Nome do Fornecedor', F.Contato, F.Telefone

, C.Descricao AS 'Cor'

FROM Produto AS P RIGHT JOIN Fornecedor AS F

**ON** P.idFornecedor = F.idFornecedor

JOIN Cor AS C

**ON** C.idCor = P.idCor

Codigo	Nome do Produto	Estoque	Nome do Fornecedor	Contato	Telefone	Cor
XT890A	Asus Zenfone	5	Asus	Felipe	5476-1120	Vermelho
TF897A	Xperia	7	Sony	Allan	8498-8732	Branco
RF212B	Moto Maxx	2	Motorola	Cristina	7987-9900	Branco



#### Com RIGHT JOIN e LEFT JOIN:

_		
_	$\overline{}$	
٠.		ш
•		

idCor	descricao
1	Branco
2	Preto
3	Azul
4	Vermelho
5	Amarelo

#### Produto

idProduto	codigo	nome	estoque	descontinuado	idFornecedor	idCor
1	XT890A	Asus Zenfone	5	0	3	4
2	RQ765B	iPhone	0	1	9	4
3	WD528B	Moto X	3	0	2	7
4	TF897A	Xperia	7	0	1	1
5	RF212B	Moto Maxx	2	0	2	1

#### Fornecedor

idFornecedor	nome	telefone	contato
1	Sony	8498-8732	Allan
2	Motorola	7987-9900	Cristina
3	Asus	5476-1120	Felipe
4	Nokia	6755-5656	Fábio

#### **SELECT** P.Codigo, P.Nome **AS** 'Nome do Produto', P.Estoque

, F.Nome AS 'Nome do Fornecedor', F.Contato, F.Telefone

, C.Descricao AS 'Cor'

FROM Produto AS P RIGHT JOIN Fornecedor AS F

**ON** P.idFornecedor = F.idFornecedor

**LEFT JOIN** Cor **AS** C **ON** C.idCor = P.idCor

Codigo	Nome do Produto	Estoque	Nome do Fornecedor	Contato	Telefone	Cor
XT890A	Asus Zenfone	5	Asus	Felipe	5476-1120	Vermelho
WD528B	Moto X	3	Motorola	Cristina	7987-9900	NULL
TF897A	Xperia	7	Sony	Allan	8498-8732	Branco
RF212B	Moto Maxx	2	Motorola	Cristina	7987-9900	Branco
NULL	NULL	NULL	Nokia	Fábio	6755-5656	NULL



#### Utilizando as mesmas, mas trocando as posições:

Cor						
idCor	descricao					
1	Branco					
2	Preto					
3	Azul					
4	Vermelho					
5	Amarolo					

Produto									
idProduto	codigo	nome	estoque	descontinuado	idFornecedor	idCor			
1	XT890A	Asus Zenfone	5	0	3	4			
2	RQ765B	iPhone	0	1	9	4			
3	WD528B	Moto X	3	0	2	7			
4	TF897A	Xperia	7	0	1	1			
5	RF212B	Moto Maxx	2	0	2	1			

_										
_		201	n	0	_	9	d	$\overline{}$	r	
	u			C	v	u	ч	v		

idFornecedor	nome	telefone	contato
1	Sony	8498-8732	Allan
2	Motorola	7987-9900	Cristina
3	Asus	5476-1120	Felipe
4	Nokia	6755-5656	Fábio

**SELECT** P.Codigo, P.Nome **AS** 'Nome do Produto', P.Estoque

, F.Nome AS 'Nome do Fornecedor', F.Contato, F.Telefone

, C.Descricao AS 'Cor'

FROM Produto AS P LEFT JOIN Fornecedor AS F

**ON** P.idFornecedor = F.idFornecedor

**RIGHT JOIN** Cor **AS** C **ON** C.idCor = P.idCor

Codigo	Nome do Produto	Estoque	Nome do Fornecedor	Contato	Telefone	Cor
XT890A	Asus Zenfone	5	Asus	Felipe	5476-1120	Vermelho
RQ765B	iPhone	0	NULL	NULL	NULL	Vermelho
TF897A	Xperia	7	Sony	Allan	8498-8732	Branco
RF212B	Moto Maxx	2	Motorola	Cristina	7987-9900	Branco
NULL	NULL	NULL	NULL	NULL	NULL	Preto
NULL	NULL	NULL	NULL	NULL	NULL	Azul
NULL	NULL	NULL	NULL	NULL	NULL	Amarelo



#### Por último, com FULL JOIN:

	-		
•		$\overline{}$	-
•	_	u	

idCor	descricao					
1	Branco					
2	Preto					
3	Azul					
4	Vermelho					
5	Amarelo					

#### Produto

idProduto	codigo	nome	estoque	descontinuado	idFornecedor	idCor
1	XT890A	Asus Zenfone	5	0	3	4
2	RQ765B	iPhone	0	1	9	4
3	WD528B	Moto X	3	0	2	7
4	TF897A	Xperia	7	0	1	1
5	RF212B	Moto Maxx	2	0	2	1

#### Fornecedor

idFornecedor	nome	telefone	contato
1	Sony	8498-8732	Allan
2	Motorola	7987-9900	Cristina
3	Asus	5476-1120	Felipe
4	Nokia	6755-5656	Fábio

#### **SELECT** P.Codigo, P.Nome **AS** 'Nome do Produto', P.Estoque

, F.Nome AS 'Nome do Fornecedor', F.Contato, F.Telefone

, C.Descricao AS 'Cor'

FROM Produto AS P FULL JOIN Fornecedor AS F

**ON** P.idFornecedor = F.idFornecedor

**FULL JOIN** Cor **AS** C **ON** C.idCor = P.idCor

Codigo	Nome do Produto	Estoque	Nome do Fornecedor	Contato	Telefone	Cor
XT890A	Asus Zenfone	5	Asus	Felipe	5476-1120	Vermelho
RQ765B	iPhone	0	NULL	NULL	NULL	Vermelho
TF897A	Xperia	7	Sony	Allan	8498-8732	Branco
RF212B	Moto Maxx	2	Motorola	Cristina	7987-9900	Branco
NULL	NULL	NULL	NULL	NULL	NULL	Preto
NULL	NULL	NULL	NULL	NULL	NULL	Azul
NULL	NULL	NULL	NULL	NULL	NULL	Amarelo
WD528B	Moto X	3	Motorola	Cristina	7987-9900	NULL
NULL	NULL	NULL	Nokia	Fábio	6755-5656	NULL



Funções de agregação são funções que estão embutidas (built-in) no banco de dados e utilizamos quando precisamos calcular valores, contabilizar número de registros ou retornar os maiores e menores valores dentro de uma coluna.

Possuem as seguintes características:

- Retornam valores escalares
- ✓ Retornam a coluna sem nome
- ✓ Ignoram colunas NULL (exceção a COUNT(\*))
- ✓ Podem ser usadas nas cláusulas SELECT, HAVING e ORDER BY



São frequentemente utilizadas com a cláusula GROUP BY, mas não são restritas ao uso sem este comando. O exemplo abaixo mostra funções de agregação sem o uso do GROUP BY:



Iremos focar nas funções de agregação de uso comum, mas é importante saber que existem outras categorias de funções agregadas, como estatísticas e outras.

#### **Uso Comum**

- SUM
- MIN
- MAX
- AVG
- COUNT
- COUNT\_BIG

#### **Estatísticas**

- STDEV
- STDEVP
- VAR
- VARP



Podemos combinar a cláusula DISTINCT com funções de agregação para sumarizar somente valores ÚNICOS.

A agregação com a cláusula DISTINCT elimina valores duplicados, não linhas (exceto se utilizarmos SELECT DISTINCT).

Compare os resultados parciais do exemplo abaixo, com e sem o uso da cláusula DISTINCT:

SELECT empid, YEAR(orderdate) AS orderyear, COUNT(custid) AS all\_custs, COUNT(DISTINCT custid) AS unique\_custs FROM Sales.Orders GROUP BY empid, YEAR(orderdate);

empid	order	year	all_custs	unique_custs
1	2006	26	22	
1	2007 2008	55 42	40 32	
2	2006	16	15	



A maioria das funções de agregação simplesmente ignoram o NULL e não geram nenhum erro. As funções a seguir ignoram o NULL:

AVG(<coluna>), COUNT (<coluna>), ...

A função COUNT usada com \* é uma EXCEÇÃO à regra acima, contabilizando TODAS AS LINHAS: **COUNT(\*)** 

Esse tipo de comportamento das funções agregadas perante o NULL pode produzir resultados INCORRETOS, como é o caso abaixo, utilizando a função AVG.

SELECT AVG(c2) AS AvgWithNULLs, AVG(ISNULL(c2,0)) AS AvgWithNULLReplace FROM dbo.t2;



GROUP BY cria grupos no retorno das linhas de acordo com a combinação da(s) coluna(s) escrita(s) na cláusula GROUP BY.

```
SELECT <select_list> FROM <table_source> WHERE <search_condition> GROUP BY <group_by_list>;
```

O GROUP BY retira os "detalhes" das linhas, fazendo um cálculo com a função de agregação selecionada para a coluna escolhida.

SELECT empid, SUM(freight) AS fht FROM Sales.Orders GROUP BY empid;

SELECT empid, COUNT(\*) AS cnt FROM Sales.Orders GROUP BY empid



Ordem Lógica	Fase	Comentário
5	SELECT	
1	FROM	
2	WHERE	
3	GROUP BY	Cria Grupos
4	HAVING	Opera na Filtragem dos Grupos

- Se a consulta (query) usa GROUP BY, todas as fases subsequentes irão operar nos grupos.
- HAVING, SELECT e ORDER BY precisam NECESSARIAMENTE retornar apenas um valor por grupo.

• Todas as colunas que aparecerem no SELECT, HAVING e ORDER BY precisam OBRIGATORIAMENTE estar ou no GROUP BY ou numa Função de Agregação.



# SELECT orderid, empid, custid FROM Sales.Orders;

orderid	empid	custid	orderid	empid	custid	empid	COUNT(
10643	6	1	10643	6	1	empia	*)
10692	4	1	10692	4	1	6	1
10926	4	2	10926	4	2	4	2
10625	3	2	10625	3	2	3	1
10365	3	3					

SELECT output

WHERE custid IN(1,2)

**GROUP BY empid** 



Funções de agregação são comumente usadas em cláusula SELECT, sumarizando a(s) coluna(s) colocada(s) no GROUP BY.

SELECT custid, COUNT(\*) AS cnt FROM Sales.Orders GROUP BY custid;

Funções de agregação podem referir qualquer coluna, não apenas as que estiverem escritas no GROUP BY.

SELECT productid, MAX(qty) AS largest\_order FROM Sales.OrderDetails GROUP BY productid;

### Cláusula HAVING



- HAVING filtra os dados obtidos através do GROUP BY.
- HAVING fornece condição de pesquisa que precisa ser satisfeita para cada grupo.
- HAVING é processado após a execução do GROUP BY.

SELECT custid, COUNT(\*) AS count FROM Sales.Orders GROUP BY custid HAVING COUNT(\*) > 10;

# Atividades extraclasse



 Leitura do arquivo PDF disponibilizado na plataforma