

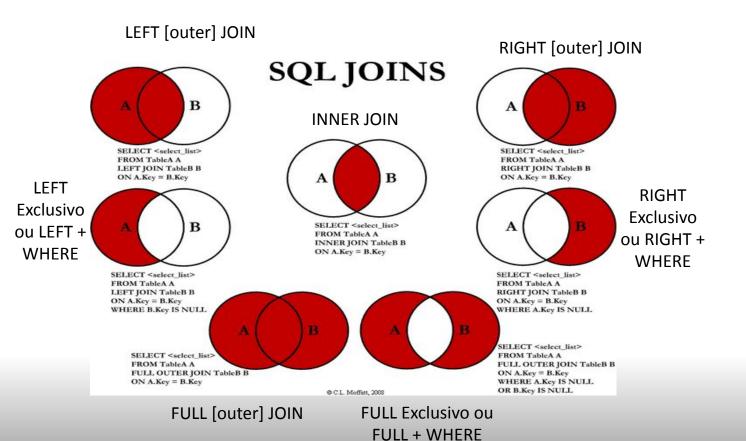
Aula 9

Data Query Language DQL - Continuação









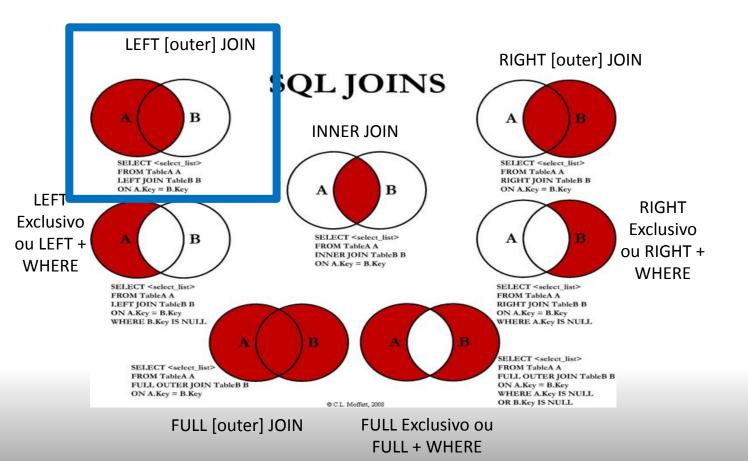
Left Join



Observando a mesma modelagem mostrada anteriormente, para gerar uma lista com TODOS os produtos, mesmo que não haja relação entre as duas tabelas, podemos substituir a cláusula JOIN por LEFT JOIN.

Ou seja, para gerar uma consulta que apresente OBRIGATORIAMENTE todos os produtos, mesmo aqueles em que não encontramos relação na tabela de fornecedores, poderíamos utilizar a cláusula LEFT JOIN, que faz com que todos os registros do objeto (tabela) do lado ESQUERDO da consulta sejam mostrados.





Left Join



Produto

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

Fornecedor

idFornecedor	nome	telefone contat		
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

FROM Produto AS P LEFT JOIN Fornecedor AS F

ON P.idFornecedor = F.idFornecedor

Codigo	Nome do Produto	Estoque	Nome do Fornecedor	Contato	Telefone
XT890A	Asus Zenfone	5	Asus	Felipe	5476-1120
RQ765B	iPhone	0	NULL	NULL	NULL
WD528B	Moto X	3	Motorola	Cristina	7987-9900
TF897A	Xperia	7	Sony	Allan	8498-8732
RF212B	Moto Maxx	2	Motorola	Cristina	7987-9900

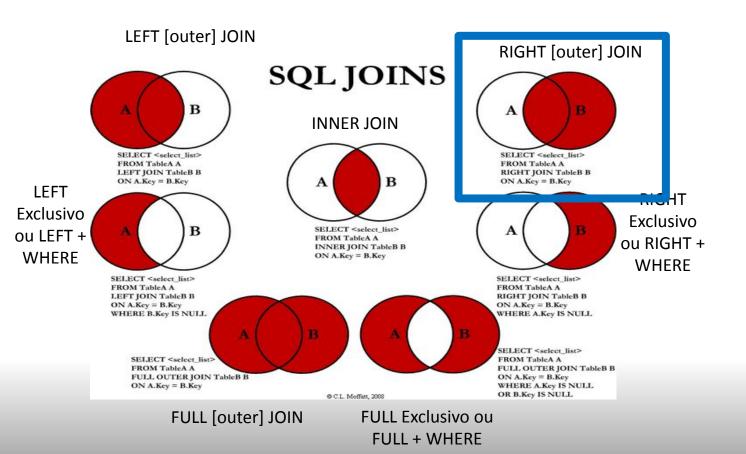
Right Join



Podemos fazer o inverso, ou seja, gerar uma lista com OBRIGATORIAMENTE todos os fornecedores, mesmo aqueles em que não encontramos relação na tabela de produtos, bastando alterar para cláusula RIGHT JOIN, que faz com que todos os registros do lado DIREITO da consulta sejam mostrados.

O controle do modo que queremos o JOIN é realizado apenas pelo posicionamento da tabela ao lado direito ou esquerdo do SELECT ou simplesmente trocando a cláusula de RIGHT JOIN para LEFT JOIN.





Right Join



Por exemplo, as tabelas TB1 e TB2, relacionadas pelas colunas C1 e C2. Se quisermos gerar uma lista com TODOS os registros da tabela TB1, os SELECTs abaixo geram essa lista:

```
SELECT ... FROM TB1 LEFT JOIN TB2 ON TB1.C1 = TB2.C2
OU
```

SELECT ... FROM TB2 RIGHT JOIN TB1 ON TB2.C2 = TB1.C1

Right Join



Produto

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

Fornecedor

idFornecedor	nome	telefone contate 8498-8732 Allan			
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

FROM Produto AS P RIGHT JOIN Fornecedor AS F

ON P.idFornecedor = F.idFornecedor

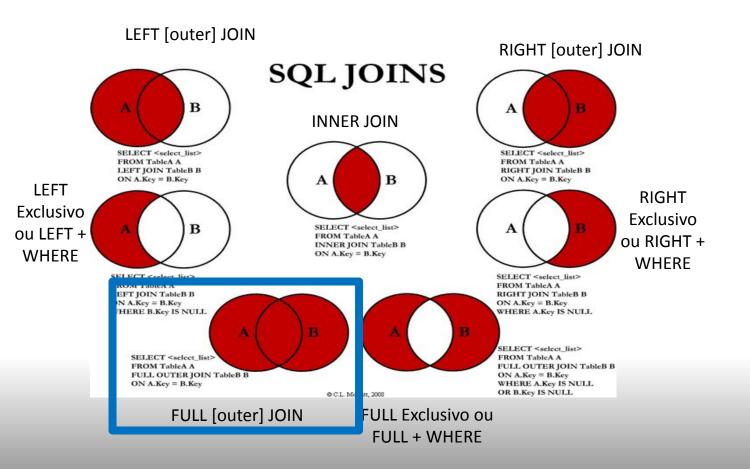
Codigo	Nome do Produto	Estoque	Nome do Fornecedor	Contato	Telefone
XT890A	Asus Zenfone	5	Asus	Felipe	5476-1120
WD528B	Moto X	3	Motorola	Cristina	7987-9900
TF897A	Xperia	7	Sony	Allan	8498-8732
RF212B	Moto Maxx	2	Motorola	Cristina	7987-9900
NULL	NULL	NULL	Nokia	6755-565	Fábio

Full Join



Existe um outro tipo de JOIN que podemos utilizar quando queremos retornar todas as linhas das tabelas envolvidas, retornando todas as relações (como o JOIN) e todos os registros não relacionados (LEFT JOIN e RIGHT JOIN) numa mesma extração. Trata-se da cláusula FULL JOIN que, em geral, não é utilizada sistemicamente, mas sim para encontrarmos possíveis problemas nos relacionamentos.





Full Join



Diferentemente do LEFT JOIN ou RIGHT JOIN, em que o posicionamento das tabelas ou escolha da cláusula gera efeito na listagem, na cláusula FULL JOIN, a ordem em que as tabelas estiverem não fará diferença, ou seja, o resultado na geração da lista será o mesmo.

Full Join



Produto

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

Fornecedor

idFornecedor	nome	telefone conta		
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

FROM Produto AS P FULL JOIN Fornecedor AS F

ON P.idFornecedor = F.idFornecedor

Codigo	Nome do Produto	Estoque	Nome do Fornecedor	Contato	Telefone
XT890A	Asus Zenfone	5	Asus	Felipe	5476-1120
RQ765B	iPhone	0	NULL	NULL	NULL
WD528B	Moto X	3	Motorola	Cristina	7987-9900
TF897A	Xperia	7	Sony	Allan	8498-8732
RF212B	Moto Maxx	2	Motorola	Cristina	7987-9900
NULL	NULL	NULL	Nokia	6755-565	Fábio



A ordem em que colocarmos as tabelas ou as cláusulas JOIN, LEFT JOIN ou RIGHT JOIN ocasiona diferenças na geração dos resultados e, conforme já visto, devemos estar atentos a isso.

Mas, se tratarmos a consulta envolvendo três ou mais tabelas como conjunto de resultados, não teremos maiores problemas na construção das queries.



resultados, e faça as junções através deles.

No script abaixo perceba que, independente das junções que tivermos, se tratarmos cada duas tabelas como um conjunto e a próxima junção como um novo conjunto, o entendimento será mais fácil.

Quando aplicamos JOIN entre as tabelas, a ordem em que estão não irá afetar os resultados, assim podemos inverter as ordens das tabelas que o efeito será o mesmo:



Se alternarmos os JOINS entre JOIN, LEFT JOIN ou RIGHT JOIN, devemos ter cuidado, pois o resultado final será afetado.

Veja o exemplo abaixo e perceba como há diferença entre as junções:

<tabela 3>.<coluna chave>

1º. Conjunto de Resultados

```
SELECT <tabela 1>.<coluna 1>, ..., <tabela 1>.<coluna n>
, <tabela 2>.<coluna 1>, ..., <tabela 2>.<coluna n>
, <tabela 2>.<coluna 1>, ..., <tabela 2>.<coluna n>
FROM <tabela 1> LEFT JOIN <tabela 2>

ON <tabela 1>.<coluna chave> = <tabela 2>.<coluna chave>
JOIN <tabela 3>

ON <tabela 1>.<coluna chave> =
```

2º. Conjunto de Resultados



É totalmente diferente se invertermos os JOINS conforme mostrado abaixo:

```
SELECT <tabela 1>.<coluna 1>, ..., <tabela 1>.<coluna n>
, <tabela 2>.<coluna 1>, ..., <tabela 2>.<coluna n>
, <tabela 2>.<coluna 1>, ..., <tabela 2>.<coluna n>
```



```
FROM <tabela 1> JOIN <tabela 2>

ON <tabela 1>.<coluna chave> = <tabela 2>.<coluna chave>

LEFT JOIN <tabela 3>

ON <tabela 1>.<coluna chave> = <tabela 3>.<coluna chave>
```



2º. Conjunto de Resultados



A mesma diferença será estendida se trocarmos pela cláusula RIGHT JOIN, ou seja, quando temos apenas a cláusula JOIN, não precisamos nos preocupar com a ordem em que as tabelas são posicionadas.

Mas, se em nossas junções tivermos qualquer tipo de LEFT JOIN ou RIGHT JOIN, temos que ter atenção no posicionamento e ordem em que as tabelas estarão, assim como cada cláusula de junção.

Atividades extraclasse



 Leitura do arquivo PDF disponibilizado na plataforma