

Aula 8

Data Query Language DQL - Continuação

Faculdade TANDACTA





• **IN** – Fornece uma lista de possibilidades de valores que poderiam atender a consulta.

SELECT SalesOrderID, OrderQty, ProductID, UnitPrice FROM Sales.SalesOrderDetail WHERE ProductID IN (750, 753, 765, 770)

SalesOrderID	OrderQty	ProductID	UnitPrice
43662	5	770	419.4589
43662	3	765	419.4589
43662	1	753	2146.962
43666	1	753	2146.962
43668	2	753	2146.962
43668	6	765	419.4589
43668	2	770	419.4589
43671	1	753	2146.962
43673	2	770	419.4589



 IN – Usa a mesma lógica de múltiplas comparações com o predicado OR entre elas.

SELECT SalesOrderID, OrderQty, ProductID, UnitPrice FROM Sales.SalesOrderDetail WHERE ProductID = 750 OR ProductID = 753 OR ProductID = 765 OR ProductID = 770

SalesOrderID	OrderQty	ProductID	UnitPrice
43662	5	770	419.4589
43662	3	765	419.4589
43662	1	753	2146.962
43666	1	753	2146.962
43668	2	753	2146.962
43668	6	765	419.4589
43668	2	770	419.4589
43671	1	753	2146.962
43673	2	770	419.4589



• **LIKE** – Permite consultas mais refinadas em colunas do tipo string (CHAR, VARCHAR, ...).

WHERE LastName = 'Johnson'

WHERE LastName LIKE 'Johns%n'

FirstName	LastName	MiddleName
Abigail	Johnson	NULL
Alexander	Johnson	М
Alexandra	Johnson	J
Alexis	Johnson	J
Alyssa	Johnson	K
Andrew	Johnson	F
Anna	Johnson	NULL

FirstName	LastName	MiddleName
Meredith	Johnsen	NULL
Rebekah	Johnsen	J
Ross	Johnsen	NULL
Willie	Johnsen	NULL
Abigail	Johnson	NULL
Alexander	Johnson	М
Alexandra	Johnson	J



- **LIKE** Este predicado é usado para verificar padrões dentro de campos strings e utiliza símbolos, chamados de curingas, para permitir a busca desses padrões.
- Símbolos (curingas)
  - % (Percent) representa qualquer string e qualquer quantidade de strings.
  - \_ (Underscore) representa qualquer string, mas apenas uma string.

[<List of characters>] representa possíveis caracteres que atendam a string procurada.

[<Character> - <character>] representa a faixa de caracteres, em ordem alfabética, para a string procurada.

[^<Character list or range>] representa o caractere que não queremos na pesquisa.



• **LIKE** – Este predicado é usado para verificar padrões dentro de campos strings e utiliza símbolos, chamados de curingas, para permitir a busca desses padrões.

SELECT categoryid, categoryname, description FROM Production.Categories WHERE description LIKE 'Sweet%'

# DQL – Utilização do NULL



$$NULL = 0 (zero)$$

# DQL – Utilização do NULL



- NULL É ausência de valor ou valor desconhecido. Nenhuma das sentenças acima é verdadeira, pois o banco de dados não pode comparar um valor desconhecido com outro valor que ele também não conhece.
- Para trabalhar com valores NULL, temos que utilizar os predicados IS NULL e IS NOT NULL.

SELECT custid, city, region, country FROM Sales. Customers WHERE region IS NOT NULL;

## DQL – Cláusula ORDER BY



Conforme mencionado anteriormente, por padrão, não há garantia de ordenação no retorno dos dados de uma consulta.

Para garantir que o retorno da consulta tenha uma ordenação, utilizamos a cláusula ORDER BY.

Elemento	Expressão	Descrição	
SELECT	<li>sta de seleção&gt;</li>	Define quais as colunas que serão retornadas.	
FROM	<tabela de="" origem=""></tabela>	Define a(s) tabela(s) envolvida(s) na consulta.	
WHERE	<condição de="" pesquisa=""></condição>	Filtra as linhas requeridas.	
GROUP BY	<agrupar a="" seleção=""></agrupar>	Agrupa a lista requerida (utiliza colunas).	
HAVING	<condição agrupamento="" de=""></condição>	Filtra as linhas requeridas, pelo agrupamento.	
ORDER	<ordem da="" lista=""></ordem>	Ordena o retorno da lista.	

#### DQL – Cláusula ORDER BY



As cláusulas ASC e DESC podem ser usadas após cada campo do comando ORDER BY. A ordenação ASCendente é a padrão quando não mencionamos explicitamente.

#### DQL – Cláusula ORDER BY



ORDER BY com nome de colunas:

SELECT orderid, custid, orderdate FROM Sales.Orders ORDER BY orderdate;

ORDER BY com apelido:

SELECT orderid, custid, YEAR(orderdate)
AS orderyear
FROM Sales.Orders
ORDER BY orderyear;

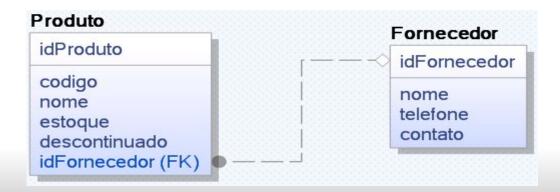
ORDER BY com ordem descendente:

SELECT orderid, custid, orderdate FROM Sales.Orders ORDER BY orderdate DESC;

#### DQL - JOIN



As junções entre as entidades são feitas através da relação de um ou mais atributos entre elas. Por exemplo, no DER abaixo, a tabela **Produto** se relaciona com a tabela **Fornecedor** através das colunas **idFornecedor**.



#### DQL – JOIN



No relacionamento mostrado, dado um determinado produto, através do relacionamento entre a coluna **idFornecedor**, conseguimos mapear as informações do fornecedor daquele produto. Para chegarmos a estas informações, escrevemos o SELECT da seguinte forma:

A cláusula JOIN faz com que o banco de dados retorne informações das tabelas envolvidas, onde a expressão na cláusula ON for atendida; no caso acima, onde a coluna chave da tabela 1 for igual à coluna chave da tabela 2.

#### DQL - JOIN



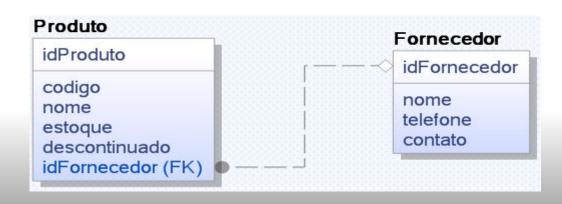
No exemplo Produto x Fornecedor, para trazer informações das duas tabelas, podemos escrever a seguinte query:

**SELECT** Produto.Codigo, Produto.Nome, Produto.Estoque

, Fornecedor.Nome, Fornecedor.Contato, Fornecedor.Telefone

FROM Produto JOIN Fornecedor

**ON** Produto.idFornecedor = Fornecedor.idFornecedor



#### DQL – JOIN



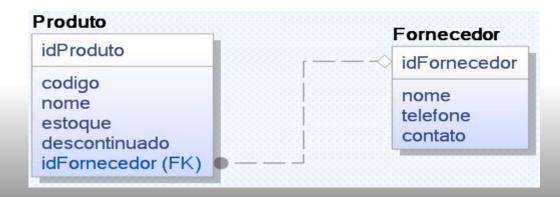
A mesma consulta poderia ser simplificada e/ou melhorada no seu entendimento adotando a utilização de apelidos para tabelas e colunas. O SELECT abaixo é exatamente o mesmo do slide anterior, utilizando Aliases para tabelas e colunas.

**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 JOIN Fornecedor AS F

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



### DQL – JOIN



#### Para o exemplo de dados abaixo:

#### 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	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

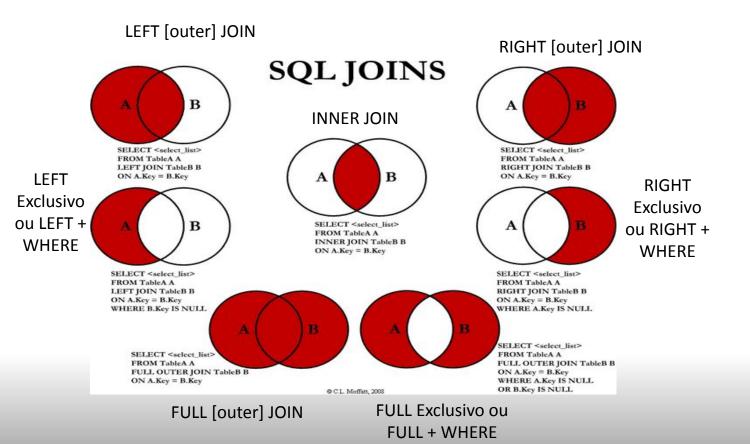
FROM Produto AS P JOIN Fornecedor AS F

**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

# Tipos de Join





# Atividades extraclasse



 Leitura do arquivo PDF disponibilizado na plataforma