

# Aula 6

## Data Query Language DQL



Categoria de subcomando da linguagem SQL que envolve a declaração de recuperação de dados (**SELECT**).

**SELECT** é uma declaração SQL que retorna um conjunto de resultados de registros de uma ou mais tabelas. Ela recupera zero ou mais linhas de uma ou mais tabelas-base, tabelas temporárias, funções ou visões em um banco de dados.

# Data Query Language (DQL)

Existem vários elementos na declaração **SELECT**, mas os principais são:

Elemento	Expressão	Descrição
SELECT	<lista de seleção>	Define quais as colunas que serão retornadas.
FROM	<tabela de origem>	Define a(s) tabela(s) envolvidas na consulta.
WHERE	<condição de pesquisa>	Filtra as linhas requeridas.
GROUP BY	<agrupar a seleção>	Agrupar a lista requerida (utiliza colunas).
HAVING	<condição de agrupamento>	Filtra as linhas requeridas pelo agrupamento.
ORDER BY	<ordem da lista>	Ordena o retorno da lista.

A ordem em que a consulta (**query**) é escrita não significa que será a mesma ordem que o banco de dados utilizará para executar o processamento:

5: SELECT      <select list>

1: FROM        <table source>

2: WHERE       <search condition>

3: GROUP BY    <group by list>

4: HAVING       <search condition>

6: ORDER BY    <order by list>

# Data Query Language (DQL)

A forma mais simples da declaração **SELECT** é a utilização junto ao elemento **FROM**, conforme mostrado abaixo.

Note que no **<select list>** há uma filtragem vertical, ou seja, retorna uma ou mais colunas de tabelas, mencionadas pela cláusula **FROM**.

Elemento	Expressão
<b>SELECT</b>	<b>&lt;select list&gt;</b>
<b>FROM</b>	<b>&lt;table source&gt;</b>

```
SELECT Nome, Sobrenome  
FROM Cliente;
```

Outros exemplos para **SELECT** simples:

- ( \* ) - Retorna todas as colunas da tabela **exemploSQL**:

```
SELECT * FROM exemploSQL
```

- ( coluna ) - Retorna a coluna **texto\_curto\_naonulo** da tabela **exemploSQL**:

```
SELECT texto_curto_naonulo FROM exemploSQL
```

Podemos utilizar diversos operadores matemáticos para cálculo de valores.  
Abaixo, mostramos os principais operadores:

Operador	Descrição
+	Add or concatenate
-	Subtract
*	Multiply
/	Divide
%	Modulo

```
SELECT preco, qtd, (preco * qtd)  
FROM DetalhesDoPedido;
```

Obs.: Operadores possuem precedência entre si.

Exemplos para SELECT simples e operadores:

- Retorna o resultado das operações abaixo:

```
SELECT 20 + 20 / 5 FROM exemploSQL
```

```
SELECT (20 + 20) / 5 FROM exemploSQL
```

```
SELECT 20 + (20 / 5) FROM exemploSQL
```

```
SELECT ( (10+2) / 2 ) * 0.3 ) % 2
```

```
SELECT Nome, Salario * 1.07 FROM Funcionario
```

**Nota:** O operador + se transforma em concatenador quando lidamos com string:

```
SELECT 'Hoje' + ' ' + 'é' + ' terça-feira ' + 'ou' + ' quinta-feira '
```



Pode ser necessário darmos apelidos (Aliases) a colunas para facilitar o entendimento no retorno dos dados:

- Apelidos na coluna utilizando a cláusula AS:

```
SELECT idPedido, preco, qtd AS Quantidade  
FROM DetalhesDoPedido;
```

- Também é possível realizar a mesma operação com =

```
SELECT idPedido, preco, Quantidade = qtd  
FROM DetalhesDoPedido;
```

- Ou mesmo, sem a necessidade do AS:

```
SELECT idPedido, preco ValorProduto  
FROM DetalhesDoPedido;
```

Também pode ser necessário darmos apelidos em tabelas, principalmente quando formos realizar joins:

- Apelidos em tabelas com a cláusula AS:

```
SELECT idPedido, dataPedido  
FROM Pedido AS SO;
```

- Apelidos em tabelas sem AS:

```
SELECT idPedido, dataPedido  
FROM Pedido SO;
```

- Usando os apelidos no SELECT:

```
SELECT SO.idPedido, SO.dataPedido  
FROM Pedido AS SO;
```

- Leitura do arquivo PDF disponibilizado na plataforma