



Modelo Físico

Linguagem SQL – Parte 04

Prof. Fábio Procópio



2

Relembrando...

- Na [aula passada](#), começamos a falar da **Linguagem de Consulta a Dados (DQL)** e, como o seu nome sugere, é usada para consultar dados;
- Vimos também a sintaxe básica do comando SQL, além de operadores e cláusulas como:
 1. DISTINCT
 2. WHERE
 3. AND, OR e NOT
 4. ORDER BY
 5. IS NULL
 6. TOP





Comando DQL

- **DDL – Data Definition Language**
 - Comandos: CREATE, ALTER e DROP
- **DML – Data Manipulation Language**
 - Comandos: INSERT, UPDATE e DELETE
- **DQL – Data Query Language**
 - Comando: SELECT
- **DCL – Data Control Language**
 - Comandos: GRANT e REVOKE
- **DTL – Data Transaction Language**
 - Comandos: COMMIT, ROLLBACK e SAVEPOINT



4

COUNT(), MIN() e MAX() – 1 de 2

➤ Função **COUNT()**

- Retorna o número de linhas que atende ao critério especificado

➤ Função **MIN()**

- Retorna o menor valor de uma determinada coluna

➤ Função **MAX()**

- Retorna o maior valor de uma determinada coluna



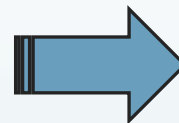
AS define um apelido (alias) para o campo resultante da contagem chamado QtBandas.

5

COUNT(), MIN() e MAX() – 2 de 2

- Esta consulta, usando a função COUNT(), conta a quantidade de bandas:

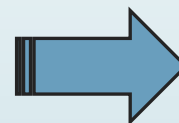
```
SELECT COUNT(*) AS QtBandas  
FROM TbBanda
```



QtBandas
5

- Esta consulta, usando a função MIN(), identifica o curso que apresenta o menor código:

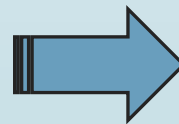
```
SELECT MIN(VaCache) AS 'Menor Cachê'  
FROM TbBanda
```



Menor Cachê
70000,00

- Esta consulta, usando a função MAX(), identifica o curso que apresenta o maior código:

```
SELECT MAX(VaCache) AS 'Maior Cachê'  
FROM TbBanda
```



Maior Cachê
150000,00



SUM() e AVG() – 1 de 2

➤ Função **SUM()**

- Retorna a soma total dos valores de uma coluna do tipo numérico

➤ Função **AVG()**

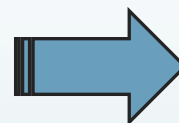
- Retorna o valor médio aritmético dos valores de uma coluna do tipo numérico



SUM() e AVG() – 2 de 2

- Esta consulta, usando a função SUM(), calcula a soma de todos os cachês das bandas:

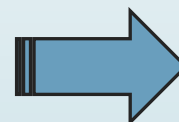
```
SELECT SUM(VaCache) AS TotalCaches  
FROM TbBanda
```



TotalCaches
530000,00

- Esta consulta, usando a função AVG(), calcula a média dos cachês das bandas:

```
SELECT AVG(VaCache) AS MediaCaches  
FROM TbBanda
```



MediaCaches
106000,00



LIKE – 1 de 2

- O operador LIKE é usado em combinação com a cláusula WHERE para procurar um padrão específico em uma determinada coluna;
- Há dois tipos de curingas que podem ser usados com o operador LIKE
 - % - o sinal de percentagem representa nenhum, um ou vários caracteres
 - _ - o *underscore* representa apenas um caractere
 - Os dois curingas podem ser usados em conjunto
- Sintaxe básica:

```
SELECT <lista de campos>  
FROM <nome_tabela>  
WHERE <nome_coluna> LIKE <padrão_procurado>
```




LIKE – 2 de 2

- A seguir, alguns exemplos mostrando diferentes combinações do operador LIKE:

Instrução	Descrição
SELECT * FROM TblIntegrante WHERE NoIntegrante LIKE 'H%'	Retorna integrantes cujos nomes começam com “H”
SELECT * FROM TblIntegrante WHERE NoIntegrante LIKE '%h'	Retorna integrantes cujos nomes finalizam com “h”
SELECT * FROM TblIntegrante WHERE NoIntegrante LIKE '%or%'	Retorna integrantes cujos nomes possuem “or” em qualquer posição
SELECT * FROM TblIntegrante WHERE NoIntegrante LIKE '_a%'	Retorna integrantes cujos nomes possuem “a” na segunda posição
SELECT * FROM TblIntegrante WHERE NoIntegrante LIKE 'F%s'	Retorna integrantes cujos nomes começam com “F” e terminam com “s”



IN – 1 de 2

- O operador IN permite especificar vários valores em uma cláusula WHERE
 - É uma forma abreviada do uso de várias condições OR
- Sintaxe básica:

```
SELECT <lista de campos>  
FROM <nome_tabela>  
WHERE <nome_coluna> IN (valor1, valor2, ...valorN)
```

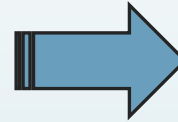
OU

```
SELECT <lista de campos>  
FROM <nome_tabela>  
WHERE <nome_coluna> IN (SELECT <valor> FROM ...)
```

IN – 2 de 2

- Esta consulta retorna os integrantes das bandas cujos códigos são 1 ou 2:

```
SELECT NoIntegrante, CoBanda
FROM TbIntegrante
WHERE CoBanda IN (1, 2)
```



NoIntegrante	CoBanda
Rogério Flausino	1
Marco Túlio Lara	1
PJ	1
Paulinho Fonseca	1
Márcio Buzelin	1
Samuel Rosa	2
Henrique Portugal	2
Lelo Zaneti	2
Haroldo Ferretti	2

- A consulta anterior ainda poderia ser rescrita de formas diferentes e retornaria o mesmo resultado. Por exemplo:

```
SELECT NoIntegrante, CoBanda
FROM TbIntegrante
WHERE CoBanda = 1
    OR CoBanda = 2
```

```
SELECT NoIntegrante, CoBanda
FROM TbIntegrante
WHERE CoBanda >= 1
    AND CoBanda <= 2
```



BETWEEN – 1 de 2

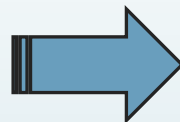
- O operador BETWEEN seleciona valores dentro de um intervalo determinado
 - Os valores podem ser números, textos ou datas
 - No caso de números e datas, o intervalo é fechado
 - O intervalo fechado [a, b] incluem os valores maiores ou iguais a “a” e menores ou iguais a “b”.
- Sintaxe básica:

```
SELECT <lista de campos>  
FROM <nome_tabela>  
WHERE <nome_coluna> BETWEEN <valor1> AND <valor2>
```

BETWEEN – 2 de 2

- Esta consulta retorna os integrantes das bandas cujos códigos estão no intervalo [1, 2]:

```
SELECT NoIntegrante, CoBanda
FROM TbIntegrante
WHERE CoBanda BETWEEN 1 AND 2
```



NoIntegrante	CoBanda
Rogério Flausino	1
Marco Túlio Lara	1
PJ	1
Paulinho Fonseca	1
Márcio Buzelin	1
Samuel Rosa	2
Henrique Portugal	2
Lelo Zaneti	2
Haroldo Ferretti	2

- A consulta anterior ainda poderia ser rescrita de formas diferentes e retornaria o mesmo resultado:

```
SELECT NoIntegrante, CoBanda
FROM TbIntegrante
WHERE CoBanda IN (1, 2)
```

```
SELECT NoIntegrante, CoBanda
FROM TbIntegrante
WHERE CoBanda >= 1
AND CoBanda <= 2
```



Apelidos (Aliases) – 1 de 2

- São usados para apelidar uma coluna ou uma tabela, temporariamente
 - Em geral, são usadas para deixar o nome de uma coluna mais amigável
 - Existem apenas enquanto a consulta está sendo executada

- Sintaxe básica para apelidar colunas:

```
SELECT nome_campo AS campo_apelidado  
FROM <nome_tabela>
```

- Sintaxe básica para apelidar tabelas:

```
SELECT <lista de campos>  
FROM <nome_tabela> AS tabela_apelidada
```

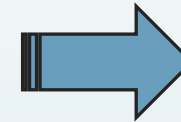


15

Apelidos (Aliases) – 2 de 2

- Esta consulta apelida a coluna **NoIntegrante** de **Nome do Integrante**:

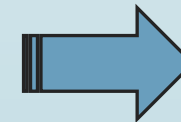
```
SELECT NoIntegrante AS 'Nome do Integrante'  
FROM TbIntegrante  
WHERE CoBanda = 1
```



Nome do Integrante
Rogério Flausino
Marco Túlio Lara
PJ
Paulinho Fonseca
Márcio Buzelin

- A consulta abaixo apelida as colunas NoIntegrante e NoBanda. Além disso, as tabelas TbBanda e TbIntegrante também são apelidadas, respectivamente, de **i** e **b**:

```
SELECT NoIntegrante AS 'Nome do Integrante',  
       NoBanda AS 'Banda'  
FROM TbIntegrante AS i, TbBanda AS b  
WHERE i.CoBanda = 1  
      AND i.CoBanda = b.CoBanda
```



Nome do Integrante	Banda
Rogério Flausino	Jota Quest
Marco Túlio Lara	Jota Quest
PJ	Jota Quest
Paulinho Fonseca	Jota Quest
Márcio Buzelin	Jota Quest



Principais Referências

- 1) w3SCHOOLS.COM. **SQL TUTORIAL.** Disponível em: <https://www.w3schools.com/sql/default.asp>. Acessado em: 22 jan 2019.
- 2) DAVE, Pinal. **SQL SERVER – What is – DML, DDL, DCL and TCL – Introduction and Examples.** Disponível em: <http://blog.sqlauthority.com/2008/01/15/sql-server-what-is-dml-ddl-dcl-and-tcl-introduction-and-examples/>. Acessado em: 22 jan. 2019.