INSTITUTO FEDERAL
Rio Grande do Norte | Campus Parnamirim

Modelo Físico
Linguagem SQL – Parte 03

Prof. Fábio Procópio

Tecnologia em Sistemas para Internet – Banco de Dados

Relembrando...

- Na aula passada, nós vimos que a **Linguagem de Manipulação de Dados (DML)** permite a atualização de dados com base na álgebra e no cálculo relacional;

- Vimos também os comandos:
  1. INSERT
  2. UPDATE
  3. DELETE

Abrindo parênteses ()...

- Para dar sequência a esta aula, clique aqui para baixar o *script*;
- No phpMyAdmin
  - Acesse a aba **Importar** e clique no botão **Escolher Arquivo**
  - Na opção **Configurar o Mapa de Caracteres do ficheiro**, selecione **iso-8859-1**
  - Pressione o botão **Executar**
- Após a execução do *script*
  - Será criado um banco de dados chamado dbMusical
  - Também serão criadas e "populadas" duas tabelas: TbBanda e TbIntegrante
  - Há um relacionamento entre elas, onde ***uma banda possui N integrantes***

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

SELECT – 1 de 3

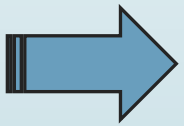
- Usado para consultar registros de uma ou mais tabelas de um banco de dados;
- Operação que não modifica dados. É apenas uma forma de "olhar" os dados armazenados em uma ou mais tabelas;
- Sintaxe básica:

```
SELECT <lista de campos>
 FROM < nome_tabela>
WHERE <critério(s) de filtragem>
ORDER BY <campo1, campo2, ..., campoN> ASC | DESC
```

SELECT – 2 de 3

- O asterisco (*) indica que todos os campos devem ser retornados;
- A ausência da cláusula WHERE permite que todas as linhas da tabela sejam retornadas;
- Na consulta abaixo, todas as linhas e todos os campos da tabela TbBanda serão retornados:

```
SELECT *
 FROM TbBanda
```



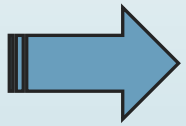
CoBanda	NoBanda	VaCache
1	Jota Quest	130000
2	Skank	100000
3	Paralamas do Sucesso	70000
4	Capital Inicial	150000
5	Roupa Nova	80000
6	Titãs	*NULL*

SELECT – 3 de 3

- Na prática, nem sempre necessita-se obter todos os campos de uma tabela. Para isso, em uma consulta SQL, pode-se especificar qual(is) campo(s) deseja-se retornar;

- A consulta abaixo retorna apenas os campos NoBanda e VaCache da tabela TbBanda:

```
SELECT NoBanda, VaCache
 FROM TbBanda
```



NoBanda	VaCache
Jota Quest	130000
Skank	100000
Paralamas do Sucesso	70000
Capital Inicial	150000
Roupa Nova	80000
Titãs	*NULL*

DISTINCT – 1 de 2

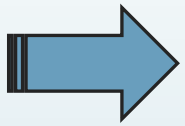
- A instrução é usada para retornar apenas valores distintos;
- Dentro de uma tabela, uma coluna pode conter valores duplicados
  - Caso haja a necessidade de listar valores diferentes, usa-se a instrução SELECT DISTINCT
- Sintaxe básica:

```
SELECT DISTINCT <lista de campos>
 FROM <nome_tabela>
```

DISTINCT – 2 de 2

- Esta consulta mostra os diferentes códigos de bandas existentes na tabela TbIntegrante:

```
SELECT DISTINCT (CoBanda)
 FROM TbIntegrante
```



CoBanda
1
2
3
4
5

WHERE – 1 de 2

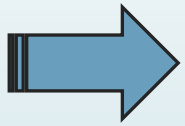
- Esta cláusula é usada para filtrar registros
  - A filtragem extrai apenas os registros que atendem ao(s) critério(s) de filtragem especificado(s)
- É importante destacar que a cláusula WHERE também é usada em outros comandos como o UPDATE e o DELETE;
- No comando SELECT, a sintaxe do uso da cláusula é:

```
SELECT <lista de campos>
 FROM < nome_tabela>
WHERE <critério(s) de filtragem>
```

WHERE – 2 de 2

- Esta consulta lista alfabeticamente todos os integrantes da banda cujo código é 1. A filtragem é feita usando a cláusula WHERE:

```
SELECT *
 FROM TbIntegrante
WHERE CoBanda = 1
```



CoIntegrante	NoIntegrante	CoBanda
1	Rogério Flausino	1
2	Marco Túlio Lara	1
3	PJ	1
4	Paulinho Fonseca	1
5	Márcio Buzelin	1

AND, OR e NOT – 1 de 3

- A cláusula WHERE pode ser combinada com os operadores AND, OR e NOT;
- Operador **AND**
  - Mostra os registros que atendem as condições verdadeiras separadas por AND
- Operador **OR**
  - Mostra os registros que atendem a pelo menos uma das condições verdadeiras separadas por OR
- Operador **NOT**
  - Mostra os registros que atendem a(s) condição (ões) NOT verdadeira

AND, OR e NOT – 2 de 3

- Sintaxe básica do **AND**:

```
SELECT <lista de campos>
 FROM <nome_tabela>
WHERE condição1 AND condição2 AND condiçãoN
```

- Sintaxe básica do **OR**:

```
SELECT <lista de campos>
 FROM <nome_tabela>
WHERE condição1 OR condição2 OR condiçãoN
```

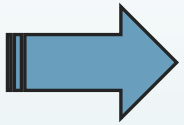
- Sintaxe básica do **NOT**:

```
SELECT <lista de campos>
 FROM <nome_tabela>
WHERE NOT condição
```

AND, OR e NOT – 3 de 3

- Esta consulta retorna as bandas cujos códigos estão entre 1 e 2, inclusive:

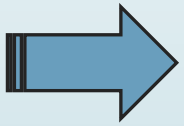
```
SELECT CoBanda, NoBanda
 FROM TbBanda
WHERE CoBanda >= 1 AND CoBanda <= 2
```



CoBanda	NoBanda
1	Jota Quest
2	Skank

- Esta consulta retorna as bandas cujo código é igual a 1 ou maior igual a 5:

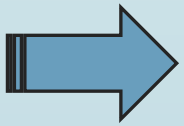
```
SELECT CoBanda, NoBanda
 FROM TbBanda
WHERE CoBanda = 1 OR CoBanda >= 5
```



CoBanda	NoBanda
1	Jota Quest
5	Roupa Nova
6	Titãs

- Esta consulta retorna as bandas cujo código não é maior que 2, isto é, CoBanda <= 2:

```
SELECT CoBanda, NoBanda
 FROM TbBanda
WHERE NOT CoBanda > 2
```



CoBanda	NoBanda
1	Jota Quest
2	Skank

ORDER BY – 1 de 2

- Cláusula usada para classificar o conjunto de resultados em ordem crescente ou decrescente
  - A classificação dos registros é, por padrão, em ordem crescente
  - Para classificar os registros em ordem decrescente, usa-se a palavra-chave **DESC**

- Sintaxe básica:

```
SELECT <lista de campos>
 FROM < nome_tabela>
WHERE <critério(s) de filtragem
ORDER BY <campo1, campo2, ...> ASC | DESC
```

A palavra **ASC** é opcional, quando a classificação é crescente.

ORDER BY – 2 de 2

- Retorna todos os registros de TbBanda ordenados alfabeticamente por NoBanda:

```
SELECT *
 FROM TbBanda
ORDER BY NoBanda
```

CoBanda	NoBanda ▲ 1	VaCache
4	Capital Inicial	150000
1	Jota Quest	130000
3	Paralamas do Sucesso	70000
5	Roupa Nova	80000
2	Skank	100000
6	Titãs	*NULL*

Observe que NoBanda é o 2º campo da listagem. Assim, a ordenação poderia ser informando a sua posição na listagem, isto é: ORDER BY **2**.

- Retorna todos os registros de TbBanda ordenados decrescentemente a partir do campo VaCache:

```
SELECT *
 FROM TbBanda
ORDER BY VaCache DESC
```

CoBanda	NoBanda	VaCache ▼ 1
4	Capital Inicial	150000
1	Jota Quest	130000
2	Skank	100000
5	Roupa Nova	80000
3	Paralamas do Sucesso	70000
6	Titãs	*NULL*

IS NULL e IS NOT NULL – 1 de 2

- Não é possível testar valores NULL com operadores de comparação como =, <, > ou <>
  - Em casos em que é necessário esse tipo de teste, usam-se **IS NULL** ou **IS NOT NULL**

- Sintaxe básica do **IS NULL**:

```
SELECT <lista de campos>
 FROM < nome_tabela>
WHERE <nome_coluna> IS NULL
```

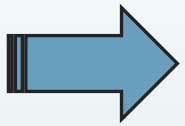
- Sintaxe básica do **IS NOT NULL**:

```
SELECT <lista de campos>
 FROM < nome_tabela>
WHERE <nome_coluna> IS NOT NULL
```

IS NULL e IS NOT NULL – 2 de 2

- Esta consulta retorna as bandas cujo valor do cachê não foi informado:

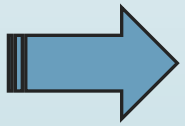
```
SELECT NoBanda, VaCache
 FROM TbBanda
WHERE VaCache IS NULL
```



NoBanda	VaCache
Titãs	*NULL*

- Por outro lado, esta consulta retorna as bandas cujo valor do cachê foi informado:

```
SELECT NoBanda, VaCache
 FROM TbBanda
WHERE VaCache IS NOT NULL
```



NoBanda	VaCache
Jota Quest	130000
Skank	100000
Paralamas do Sucesso	70000
Capital Inicial	150000
Roupa Nova	80000

LIMIT – 1 de 2

- A instrução **LIMIT** é usada para especificar a quantidade de registros a ser retornada pela consulta
  - Útil para tabelas com milhares de registros
  - O retorno de um grande número de registros pode impactar na performance do sistema

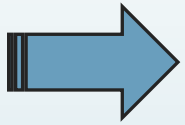
- Sintaxe básica:

```
SELECT <lista de campos>
 FROM <nome_tabela>
 LIMIT <qtde_reg | quantidade OFFSET a_partir_de | a_partir_de, qtde_reg>
```

LIMIT – 2 de 2

- Esta consulta retorna os 2 primeiros registros da tabela TbBanda, ordenados decrescentemente pelo campo VaCache:

```
SELECT *
  FROM TbBanda
ORDER BY VaCache DESC
   LIMIT 2
```



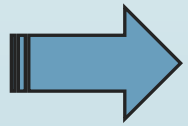
CoBanda	NoBanda	VaCache ▼ 1
4	Capital Inicial	150000
1	Jota Quest	130000

- Esta consulta retorna 3 bandas, começando do registro que ocupa a posição 2 (OFFSET 1):

```
SELECT *
  FROM TbBanda
ORDER BY CoBanda
   LIMIT 3 OFFSET 1
```

OU

```
SELECT *
  FROM TbBanda
ORDER BY CoBanda
   LIMIT 1, 3
```



CoBanda ▲ 1	NoBanda	VaCache
2	Skank	100000
3	Paralamas do Sucesso	70000
4	Capital Inicial	150000

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.