

BANCO DE DADOS RELACIONAL

Nesta aula vamos aprender sobre:

- Recapitulando DDL
- Comandos DQL e DML
- Comando INSERT INTO
- Comando UPDATE
- Comando DELETE
- Atividade

Prof. Júnior Gonçalves



Recapitulando...

A linguagem SQL (***Struct Query Language***) é uma linguagem padrão para a manipulação de dados dentro de um SGBD. Desenvolvida pela IBM, possibilita o armazenamento, organização, atualização e exclusão de informações dentro do banco de dados.

- DDL (*Data Definition Language* – linguagem de definição de dados) permite determinar o esquema do banco de dados, bem como alterá-lo e excluí-lo, e trabalhar com os metadados;
- DQL (*Data Query Language* – linguagem de consulta de dados) proporciona a consulta de dados;
- DML (*Data Manipulation Language* – linguagem de manipulação de dados) permite a manipulação dos dados, ou seja, a inclusão, alteração e exclusão dos dados;

Comandos SQL

```
> CREATE DATABASE biblioteca_aula2;

> USE biblioteca_aula2;

> CREATE TABLE tb_autores (
    idautor INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(60),
    nacionalidade VARCHAR(30) );

> CREATE TABLE tb_editoras(
    ideditora INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(60),
    telefone VARCHAR(15) );
```



Comandos SQL

```
MySQL 8.0 Command Line Cli × + ▾

mysql> DESCRIBE tb_autores;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| idautor        | int           | NO   | PRI | NULL    | auto_increment |
| nome           | varchar(60)   | YES  |     | NULL    |                |
| nacionalidade  | varchar(30)   | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.03 sec)

mysql> DESCRIBE tb_editoras;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| ideditora      | int           | NO   | PRI | NULL    | auto_increment |
| nome           | varchar(60)   | YES  |     | NULL    |                |
| telefone       | varchar(15)   | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> |
```



Comandos SQL

- > **SHOW tables;**
- > **ALTER TABLE tb_autores
MODIFY COLUMN nome VARCHAR(50) NOT NULL;**
- > **ALTER TABLE tb_editoras
ADD COLUMN representante VARCHAR(50),
MODIFY COLUMN nome VARCHAR(50) NOT NULL;**

```
MySQL 8.0 Command Line Cli x + v
mysql> DESCRIBE tb_editoras;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| ideditora      | int           | NO   | PRI | NULL    | auto_increment |
| nome           | varchar(50)   | NO   |     | NULL    |                |
| telefone       | varchar(15)   | YES  |     | NULL    |                |
| representante  | varchar(50)   | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```



Comandos SQL

A DQL (*Data Query Language* – Linguagem de Consulta de Dados) é responsável por permitir a consulta e recuperação de dados em um banco de dados.

Dentro do subgrupo DQL, temos um único comando: o SELECT.

```
> SELECT * FROM tb_autores;
```

O comando SELECT é composto por diversas cláusulas e opções, o que possibilita a elaboração de consultas desde as mais simples até as mais complexas. Por meio dessas cláusulas e opções, os usuários podem definir critérios de filtragem, ordenação, agrupamento e outras operações que direcionam o resultado da consulta de acordo com suas necessidades específicas.



Comandos SQL

Uma das principais partes da linguagem SQL é a DML (*Data Manipulation Language*), que inclui os comandos de manipulação dos dados. Quando dizemos manipulação, estamos falando sobre inserção, atualização e exclusão de dados em uma tabela.

**> INSERT INTO tb_autores (nome, nacionalidade)
VALUES ('Monteiro Lobato', 'Brasil');**

```
MySQL 8.0 Command Line Cli x + v

mysql> INSERT INTO tb_autores (nome, nacionalidade)
-> VALUES ('Monteiro Lobato', 'Brasil');
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM tb_autores;
+-----+-----+-----+
| idautor | nome          | nacionalidade |
+-----+-----+-----+
|      1 | Monteiro Lobato | Brasil        |
+-----+-----+-----+
1 row in set (0.00 sec)
```



Comandos SQL

Alguns campos podem ser deixados em branco, não indicando o nome do campo e, conseqüentemente, não atribuindo valor para ele

```
> INSERT INTO tb_autores (nome)  
VALUES ('Ariano Suassuna');
```

Se o campo “nacionalidade” fosse criado com a opção NOT NULL não seria possível executar este comando, pois teríamos uma restrição que obriga a entrar com um valor assim que o registro é inserido

Se mantivermos a ordem dos campos na mesma sequencia da tabela, podemos omitir os nomes dos campos

```
> INSERT INTO tb_autores  
VALUES (NULL, 'Clarice Lispector', 'Brasil');
```



Comandos SQL

> SELECT * FROM tb_autores;

```
mysql> SELECT * FROM tb_autores;
+-----+-----+-----+
| idautor | nome           | nacionalidade |
+-----+-----+-----+
|      1 | Monteiro Lobato | Brasil        |
|      2 | Ariano Suassuna | NULL          |
|      3 | Clarice Lispector | Brasil        |
+-----+-----+-----+
3 rows in set (0.01 sec)
```

> SELECT nome, nacionalidade FROM tb_autores;

```
mysql> SELECT nome, nacionalidade FROM tb_autores;
+-----+-----+
| nome           | nacionalidade |
+-----+-----+
| Monteiro Lobato | Brasil        |
| Ariano Suassuna | NULL          |
| Clarice Lispector | Brasil        |
+-----+-----+
3 rows in set (0.01 sec)
```



Comandos SQL

Podemos aproveitar e incluir vários registros de uma só vez, basta indicar os campos e os valores de cada registro separados por vírgula, como no exemplo a seguir:

```
> INSERT INTO tb_autores (nome, nacionalidade) VALUES  
    ('J.R.R. Tolkien', 'Inglaterra'),  
    ('Agatha Christie', 'Inglaterra'),  
    ('Isaac Asimov', 'Rússia');
```

```
mysql> INSERT INTO tb_autores (nome, nacionalidade) VALUES  
-> ('J.R.R. Tolkien', 'Inglaterra'),  
-> ('Agatha Christie', 'Inglaterra'),  
-> ('Isaac Asimov', 'Rússia');
```

```
Query OK, 3 rows affected (0.03 sec)
```

```
Records: 3  Duplicates: 0  Warnings: 0
```

```
mysql> SELECT * FROM tb_autores;
```

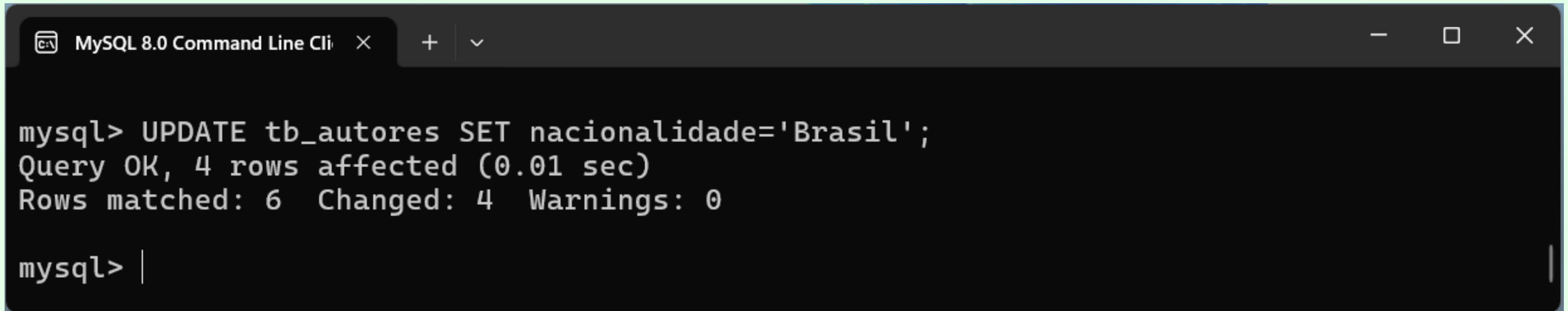
idautor	nome	nacionalidade
1	Monteiro Lobato	Brasil
2	Ariano Suassuna	NULL
3	Clarice Lispector	Brasil
4	J.R.R. Tolkien	Inglaterra
5	Agatha Christie	Inglaterra
6	Isaac Asimov	Rússia



Comandos SQL

É possível alterar os campos de uma tabela utilizando o UPDATE

> UPDATE tb_autores SET nacionalidade='Brasil';

A screenshot of the MySQL 8.0 Command Line Client interface. The window title is "MySQL 8.0 Command Line Client". The terminal shows the command "mysql> UPDATE tb_autores SET nacionalidade='Brasil';" being executed. The output is "Query OK, 4 rows affected (0.01 sec)" followed by "Rows matched: 6 Changed: 4 Warnings: 0". The prompt "mysql> |" is visible at the bottom.

```
mysql> UPDATE tb_autores SET nacionalidade='Brasil';
Query OK, 4 rows affected (0.01 sec)
Rows matched: 6  Changed: 4  Warnings: 0

mysql> |
```



Comandos SQL

> **SELECT * FROM tb_autores;**

```
mysql> SELECT * FROM tb_autores;
```

idautor	nome	nacionalidade
1	Monteiro Lobato	Brasil
2	Ariano Suassuna	Brasil
3	Clarice Lispector	Brasil
4	J.R.R. Tolkien	Brasil
5	Agatha Christie	Brasil
6	Isaac Asimov	Brasil

6 rows in set (0.00 sec)

Isso aconteceu por que não colocamos a cláusula **WHERE** no comando **UPDATE**, que impõe uma condição para que essa alteração afete apenas um conjunto de registros



Comandos SQL

Vamos corrigir nossa tabela autores:

- > **UPDATE tb_autores
SET nacionalidade='Inglaterra'
WHERE idautor=4;**
- > **UPDATE tb_autores
SET nacionalidade='Inglaterra'
WHERE idautor=5;**
- > **UPDATE tb_autores
SET nacionalidade='Rússia'
WHERE nome='Isaac Asimov';**



Comandos SQL

Para concluir os comandos de manipulação de dados, temos o comando DELETE, que é utilizado para a remoção de registros:

- > **DELETE FROM tb_autores WHERE idautor=2;**
- > **DELETE FROM tb_autores
WHERE nacionalidade='Inglaterra';**

```
mysql> SELECT * FROM tb_autores;
```

idautor	nome	nacionalidade
1	Monteiro Lobato	Brasil
3	Clarice Lispector	Brasil
6	Isaac Asimov	Rússia

```
3 rows in set (0.00 sec)
```



Dúvidas, perguntas,
sugestões, reclamações,
opiniões, comentários...

**Não existe pergunta boba,
bobo é quem não pergunta.**

Vamos praticar com um exercício >>



Atividade 03: Comandos DML

Lista de exercícios com o banco biblioteca

