



INSTITUTO FEDERAL
PIAUÍ
Campus Parnaíba

Revisão - SQL - MySQL

Prof. Msc Denival A. dos Santos

Base de Dados

- Para criarmos uma base de dados no MySQL Server, utilizamos:

```
mysql> CREATE DATABASE livraria;  
Query OK, 1 row affected (0.02 sec)
```

- Para seleccionar uma bases de dados existente, utilizamos USE:

```
mysql> USE livraria;  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A  
  
Database changed
```

Base de Dados

- Para listar as bases de dados existentes, utilizamos:

```
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| livraria |  
| mysql |  
| test |  
+-----+  
4 rows in set (0.03 sec)
```

- Para remover uma base de dados existente, utilizamos:

```
mysql> DROP DATABASE livraria;  
Query OK, 0 rows affected (0.08 sec)
```

Tabelas

- Para criarmos uma tabela no MySQL Server, utilizamos:

```
mysql> CREATE TABLE 'livraria'.'Livro' (  
-> 'titulo' VARCHAR(255),  
-> 'preco' DOUBLE  
-> )  
-> ENGINE=MyISAM;  
Query OK, 0 rows affected (0.14 sec)
```

- Para listar as tabelas existentes, utilizamos:

```
mysql> SHOW TABLES;  
+-----+  
| Tables_in_livraria |  
+-----+  
| Livro               |  
+-----+  
1 row in set (0.00 sec)
```

- Para remover uma tabela existente, utilizamos:

```
mysql> DROP TABLE Livro;  
Query OK, 0 rows affected (0.00 sec)
```

Tabelas

- Para modificar a estrutura de uma tabela no MySQL Server, utilizamos:

```
mysql> ALTER TABLE Livro RENAME livros;  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> ALTER TABLE Livro ADD paginas INTEGER;  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> ALTER TABLE Livro DROP COLUMN paginas;  
Query OK, 0 rows affected (0.00 sec)
```

Tabelas

▪ CRUD

- As operações básicas para manipular os dados persistidos são: Inserção (INSERT), ler (SELECT), alterar (UPDATE) e remover (DELETE).

- INSERT

```
mysql> INSERT INTO Livro (titulo, preco) VALUES ('Java', 98.75);  
Query OK, 1 row affected (0.00 sec)
```

Tabelas

- **CRUD**

- SELECT

```
mysql> SELECT * FROM Livro;  
+-----+-----+  
| titulo | preco |  
+-----+-----+  
| Java   | 98.75 |  
+-----+-----+  
1 row in set (0.00 sec)
```

- **CRUD**

- UPDATE

```
mysql> UPDATE Livro SET preco = 115.9 WHERE titulo = 'Java';  
Query OK, 1 row affected (0.00 sec)  
Rows matched: 1   Changed: 1   Warnings: 0
```


- **CRUD**

- DELETE

```
mysql> DELETE FROM Livro WHERE titulo = 'Java';  
Query OK, 1 row affected (0.00 sec)
```

Tabelas - Seleção

- Para recuperar colunas de uma tabela, utilizamos:

```
SELECT nome, email FROM Aluno;
```

```
SELECT * FROM Aluno;
```

Tabelas - Seleção

- Para filtro de condições, utilizamos:

```
SELECT * FROM Aluno WHERE altura > 1.80;
```

```
SELECT * FROM Aluno WHERE altura BETWEEN 1.5 AND 1.8;
```

```
SELECT * FROM Aluno WHERE nome LIKE 'Rafael%';
```

```
SELECT * FROM Aluno WHERE altura IN (1.5, 1.6, 1.7, 1.8);
```

```
SELECT * FROM Aluno WHERE altura < 1.8 AND nome LIKE 'Rafael%';
```

```
SELECT * FROM Aluno WHERE nome IS NULL;
```

Tabelas - Seleção

- Para o ordenamento dos resultados, utilizamos:

```
SELECT * FROM Aluno ORDER BY altura;
```

```
SELECT * FROM Aluno ORDER BY altura DESC, nome ASC;
```

Tabelas - Seleção

- Funções agregadas:

```
SELECT COUNT(*) FROM Aluno;
```

```
SELECT AVG(altura) FROM Aluno;
```

```
SELECT SUM(altura) FROM Aluno;
```

```
SELECT MAX(altura) FROM Aluno;
```

```
SELECT MIN(altura) FROM Aluno;
```

Tabelas - Seleção

- Para agrupamos valores, utilizamos;

```
SELECT aprovado, COUNT(*) FROM Aluno GROUP BY aprovado;
```

```
SELECT sexo, aprovado, COUNT(*) FROM Aluno GROUP BY sexo, aprovado;
```

Tabelas - Seleção

- Para junções de tabelas, utilizamos;

```
SELECT *  
FROM Usuario AS u  
JOIN Perfil AS p  
ON u.id = p.id;
```