

INFORMAÇÕES

MÓDULO	AULA	INSTRUTOR
01 Fundamentos de Banco de Dados	01 - Fundamentos do SQL e PostgreSQL	Matheus Laureto

CONTEÚDO

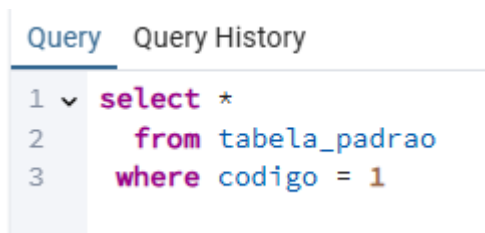
1. Fundamentos do SQL

SQL, sigla para Structured Query Language (Linguagem de Comando Estruturada) é a linguagem padrão utilizada para interagir com bancos de dados relacionais. Dominar esses comandos é essencial para uma boa análise de dados, correção de problemas ou outros desafios que possam aparecer.

Os comandos SQL podem ser divididos em três categorias:

- **DML - Data Manipulation Language**
 - (select, insert, update, delete)
- **DDL - Data Definition Language**
 - (create, alter, drop, truncate)
- **DCL - Data Control Language**
 - (grant, revoke)

2. DML - Data Manipulation Language



```

Query  Query History
1  select *
2    from tabela_padrao
3   where codigo = 1
  
```

SELECT:

- Trata-se do principal comando da linguagem SQL, onde nós selecionamos (SELECT rs) os dados que serão listados na nossa consulta;

FROM:

- É o comando de onde informamos a origem das informações que serão listadas;

WHERE:

- Comando responsável por filtrar os dados que serão listados na consulta. Obs: dos tres comandos listados, o comando where nao é obrigatório, porém é altamente recomendado a sua utilização.

```
insert into tabela_padrao (codigo, descricao)
values (1, 'PRODUTO 1')
```

INSERT: Comando responsável pela inserção dos dados dentro das tabelas previamente criadas no banco de dados.

Obs: O comando **INSERT** pode ter variações na sua execução, que veremos no decorrer do curso.

```
update tabela_padrao
set descricao = 'PRODUTO NOVO'
where codigo = 1
```

UPDATE: Com o comando update, podemos alterar os dados que já estão salvos dentro das tabelas do banco de dados.

Da mesma forma que no comando **SELECT**, a instrução **WHERE** não é obrigatória para a sua execução, porém se for executada dessa maneira, todos os registros da tabela serão alterados pelo comando **UPDATE** (famoso update sem where).

```
delete
from tabela_padrao
where codigo = 1
```

DELETE: O comando **DELETE** é utilizado para excluir registros existentes em uma tabela do banco de dados. Ele atua linha por linha, removendo os dados conforme os critérios definidos no filtro **WHERE**.

OBS: O comando **WHERE** não é obrigatório, mas altamente recomendado.

A execução de um **DELETE** sem **WHERE** poderá apagar **TODOS** os registros da tabela.

3. DDL - Data Definition Language

```
create table tabela_valores (
    codigo int,
    descricao varchar,
    valor decimal(12,2))
```

CREATE: O comando **CREATE** já faz parte da linguagem DDL e é utilizado para criar estruturas dentro do banco de dados.

Ex: (tabelas, outros bancos de dados, views, índices, procedures, etc...)

A sintaxe do comando **CREATE** varia de acordo com o tipo de estrutura a ser criada, mas no exemplo acima temos a criação de uma tabela no banco.

```
alter table tabela_valores  
add column categoria varchar(50)
```

ALTER: O comando **ALTER** é responsável pela alteração de uma estrutura previamente criada. Como o exemplo anterior, criamos uma tabela para armazenar VALORES, e posteriormente a tabela foi alterada para também se armazenar a categoria daquele registro.

Poderia apagar a tabela e criar novamente? Sim, se a tabela estiver vazia. Se ela estiver preenchida com dados, ao apagar essa tabela, os registros automaticamente também serão apagados.

```
drop table tabela_valores
```

DROP: Cuidado com o comando **DROP**! Este comando faz a exclusão do registro físico da tabela, desta forma, os registros que estavam inseridos na tabela também serão excluídos.

```
truncate table tabela_valores
```

TRUNCATE: Este comando também exige atenção. O **TRUNCATE** elimina todos os dados da tabela, não permitindo a seleção dos mesmos como é feito no **DELETE** com **WHERE**.

4. DCL - Data Control Language

GRANT: O comando **GRANT** atribui permissões de um objeto para um usuário.

REVOKE: Da mesma maneira que o **GRANT** atribui permissões, o comando **REVOKE** retira permissões dos usuários.

Exemplo:

```
grant select, insert, update, delete on tabela_valores to postgres
```

Nesse exemplo, é atribuído permissões de **SELECT**, **INSERT**, **UPDATE** e **DELETE** na tabela **TABELA_VALORES**.

Obs: Neste momento começamos a falar sobre hierarquia de acessos no banco de dados, ou seja, só é possível dar **GRANT** para um determinado usuário se o seu próprio usuário tiver essa mesma permissão.

Ex: Usuário "A" quer dar **GRANT** de leitura para o usuário "B".

Para isso, o usuário "A" obrigatoriamente já deve ter permissão de leitura na tabela e permissão para atribuir esse **GRANT** a outro usuário (através do comando **WITH GRANT OPTION**).

```
revoke select, insert, update, delete on tabela_valores to postgres
```

Nesse exemplo, podemos ver o comando **REVOKE**, removendo permissões de **SELECT**, **INSERT**, **DELETE** e **UPDATE** de determinado usuário no banco