

E-BOOK

DO ESTUDANTE

»»» PostgreSQL

Indo além do SQL!



Todos os direitos reservados
©2023 Resilia Educação

RESILIA



CONTEXTUALIZANDO	2
FUNDAMENTOS DE POSTGRESQL —.....	3
COMANDOS DE BANCO DE DADOS —.....	6
COMANDOS DE BASE DE DADOS	8
PARA REFLETIR	10
ATIVIDADE	11
PARA IR ALÉM	11
RESUMO DE ESTRUTURAS DE DADOS EM PYTHON	12

PostgreSQL

Indo além do SQL!



O PostgreSQL é um sistema de gerenciamento de banco de dados relacional de código aberto que foi lançado pela primeira vez em 1989. Ele é conhecido por sua capacidade de lidar com grandes quantidades de dados e por sua confiabilidade, segurança e flexibilidade. Como um banco de dados relacional, o PostgreSQL é baseado no modelo de tabela, onde os dados são organizados em tabelas compostas por linhas e colunas, e oferece suporte a uma ampla variedade de recursos avançados, como consultas complexas, transações, integridade referencial e armazenamento de dados em formato **JSON**. Além disso, o PostgreSQL é altamente extensível, permitindo que os usuários personalizem o sistema para atender às suas necessidades específicas. Devido à sua natureza de código aberto e à sua vasta comunidade de desenvolvedores, o PostgreSQL é uma opção popular para empresas e organizações que desejam um banco de dados confiável e escalável.

CONTEXTUALIZANDO

O PostgreSQL é popular no mercado de análise de dados devido à sua capacidade de lidar com grandes volumes de dados e **flexibilidade** para a manipulação de dados. Uma das principais vantagens do PostgreSQL é a possibilidade de ampliar suas funcionalidades por meio de extensões, permitindo que os usuários possam criar funções personalizadas para manipulação de dados e estatísticas. Além disso, o PostgreSQL possui suporte a recursos avançados, como índices textuais, geoespaciais e em JSON, tornando-o adequado para diferentes tipos de análise de dados.

O PostgreSQL também é altamente confiável e seguro, o que é fundamental para o mercado de análise de dados, onde a integridade dos dados é crucial. Ele possui recursos de transação **ACID (Atomicidade, Consistência, Isolamento e Durabilidade)**, que garantem que os dados sejam gravados de forma consistente e segura, mesmo em cenários de falha do sistema.



O PostgreSQL também oferece uma ampla variedade de ferramentas para integração com outras tecnologias de análise de dados, como **R** e **Python**, que são amplamente utilizados no mercado de análise de dados. Ele suporta a interface SQL padrão, permitindo que os usuários realizem consultas complexas e relatórios.

Por essas razões, o PostgreSQL é uma opção muito utilizada no mercado de análise de dados, especialmente para instituições que buscam um banco de dados escalável, confiável e flexível para seus projetos de análise de dados.

FUNDAMENTOS DE POSTGRESQL



O PostgreSQL é um dos **sistemas gerenciadores de banco de dados (SGBD)** mais populares e robustos do mundo, com recursos avançados de segurança e escalabilidade. Ele é amplamente utilizado por empresas e organizações para armazenar e gerenciar grandes volumes de dados. Nesta seção, vamos nos concentrar em como o PostgreSQL pode ser usado para análise de dados.

Mas antes de tudo, vamos aprender a instalar o PostgreSQL na máquina?

Para começar a usar o PostgreSQL, precisamos instalá-lo no nosso sistema operacional. Aqui estão as instruções básicas para baixar e instalar o PostgreSQL no sistema operacional Windows:

- Baixe o instalador do PostgreSQL para Windows no site oficial do PostgreSQL: <https://www.postgresql.org/download/windows/>
- Execute o arquivo de instalação e siga as instruções na tela para instalar o PostgreSQL. Durante a instalação, é possível escolher a pasta de instalação, o idioma e outras configurações.
- Quando a instalação estiver concluída, verifique se o PostgreSQL está funcionando corretamente iniciando o serviço do PostgreSQL. Isso pode ser feito ao clicar no menu Iniciar do Windows, digitar "Serviços" na caixa de pesquisa e abrir o aplicativo "Serviços". Na lista de serviços, procure o serviço "PostgreSQL" e verifique se ele está em execução.

Para sistemas operacionais baseados em **Unix**, o processo de instalação pode variar dependendo da distribuição e da versão do sistema. Em geral, podemos usar o gerenciador de pacotes do sistema operacional para instalar o PostgreSQL. Por exemplo, no Ubuntu, é possível instalar o PostgreSQL usando o seguinte comando:

```
sudo apt-get install postgresql
```

A instalação do PostgreSQL é um processo simples que pode ser feito seguindo algumas etapas básicas. Após instalar o PostgreSQL, pode-se começar a criar e manipular bancos de dados usando a linguagem SQL. O conhecimento dos fundamentos do PostgreSQL é essencial para utilizar todo o potencial desse sistema de gerenciamento de banco de dados e para lidar com grandes volumes de dados.

Agora, vamos voltar para os fundamentos do PostgreSQL!

Estrutura de Dados do PostgreSQL

Uma das principais razões pelas quais o PostgreSQL é tão popular é sua estrutura de dados altamente flexível. O PostgreSQL suporta vários tipos de dados, incluindo **inteiros, decimais, texto, data/hora, booleanos, JSON** e muitos outros. Ele também permite a criação de tabelas e índices para facilitar a organização e busca dos dados. A estrutura de dados flexível do PostgreSQL é fundamental para a análise de dados, pois permite que os dados sejam armazenados de maneira eficiente e organizada.

Consultas SQL Avançadas

O PostgreSQL suporta a linguagem **SQL (Structured Query Language)**, que é a linguagem padrão para consulta e manipulação de bancos de dados. Com o PostgreSQL é possível escrever consultas SQL avançadas que permitem filtrar e agrupar dados de maneira eficiente. Por exemplo, pode-se usar a cláusula **"WHERE"** para filtrar os dados com base em uma condição específica e a cláusula **"GROUP BY"** para agrupar os dados com base em uma coluna específica. A habilidade de escrever consultas SQL avançadas é fundamental para a análise de dados, pois permite que os dados sejam examinados e resumidos com precisão.

Funções e Procedimentos Armazenados

O PostgreSQL suporta funções e procedimentos armazenados, que são blocos de código que podem ser reutilizados várias vezes. Essas funções e procedimentos podem ser usados para realizar cálculos complexos, manipular dados e executar outras tarefas específicas do usuário. A habilidade de criar funções e procedimentos armazenados é fundamental para a análise de dados, pois permite que tarefas complexas sejam automatizadas e reutilizadas várias vezes. Além disso, o PostgreSQL suporta a análise de dados em tempo real por meio de recursos como o **PostgreSQL Logical Decoding** e o **PostgreSQL Streaming Replication**. Esses recursos permitem que os dados sejam capturados e replicados em tempo real para outros servidores ou aplicativos. A habilidade de realizar análise de dados em tempo real é fundamental para a análise de dados em ambientes em que a velocidade é crucial.

Extensões do PostgreSQL

O PostgreSQL suporta extensões, que são módulos adicionais que fornecem recursos adicionais ao sistema. Existem muitas extensões disponíveis para o PostgreSQL que podem ser usadas para a análise de dados, incluindo extensões para mineração de dados, estatísticas e análise de texto. A habilidade de utilizar extensões é fundamental para a análise de dados em ambientes em que recursos avançados são necessários.

Recursos Adicionais

O PostgreSQL possui recursos para armazenamento e consulta de dados espaciais, o que o torna uma excelente opção para análise de dados geográficos. Ele suporta vários tipos de dados espaciais, incluindo pontos, linhas e polígonos, e fornece funções e operadores para consultar e manipular esses dados. O PostgreSQL também possui recursos para **índices de texto completos**, o que o torna uma excelente opção para análise de dados de texto. Os índices de texto completo permitem que os dados de texto sejam pesquisados de forma eficiente, independentemente do tamanho ou complexidade do texto.

Além disso, o PostgreSQL é conhecido por sua capacidade de gerenciamento de transações. Ele suporta transações **ACID (Atomicidade, Consistência, Isolamento e Durabilidade)**, que garantem que as transações sejam processadas com segurança e confiabilidade. O gerenciamento de transações é essencial para garantir a integridade dos dados e que as transações sejam concluídas corretamente.

O PostgreSQL é compatível com vários sistemas operacionais, incluindo **Windows, Linux e macOS**, o que o torna uma excelente opção para organizações e empresas que usam diferentes sistemas operacionais em seus ambientes de dados. Essa compatibilidade também facilita a integração do PostgreSQL com outras ferramentas de análise de dados.

O PostgreSQL suporta a **escalabilidade horizontal**, o que significa que ele pode ser executado em um cluster de servidores para distribuir a carga de trabalho e aumentar a capacidade de processamento. A escalabilidade horizontal é essencial para organizações que precisam lidar com grandes volumes de dados ou que precisam de maior capacidade de processamento para análises de dados mais complexas.

Esses recursos adicionais ajudam a tornar o PostgreSQL uma opção ainda mais atraente para análise de dados, pois fornecem recursos adicionais para lidar com diferentes tipos de dados e ambientes de dados, além de garantir a

segurança e a confiabilidade do processamento de transações e da escalabilidade. Combinados com os recursos mencionados anteriormente, esses recursos adicionais tornam o PostgreSQL uma ferramenta poderosa para análise de dados.

Até aqui, abordamos como o PostgreSQL pode ser usado para análise de dados. A estrutura de dados flexível do PostgreSQL, a habilidade de escrever consultas SQL avançadas, a criação de funções e procedimentos armazenados, a análise de dados em tempo real e o uso de extensões são recursos fundamentais para a análise de dados no PostgreSQL. Com esses recursos, é possível armazenar, gerenciar, processar e analisar grandes volumes de dados de forma eficiente e organizada.

O PostgreSQL é uma excelente escolha para organizações e empresas que desejam realizar análises de dados sofisticadas e escaláveis. Com o conhecimento e habilidades adequadas, é possível utilizar o PostgreSQL para transformar dados brutos em informações valiosas e insights significativos para tomadas de decisão mais assertivas.

COMANDOS DE BANCO DE DADOS



O PostgreSQL é um sistema de gerenciamento de banco de dados relacional (RDBMS) de código aberto e muito poderoso. É uma excelente opção para aplicativos que exigem alta escalabilidade e capacidade de processamento, com recursos avançados de gerenciamento de dados, como suporte a transações ACID e gerenciamento de esquema robusto. Nesta seção, abordaremos os principais comandos de banco de dados do PostgreSQL, que ajudarão os usuários a gerenciar seus dados de forma eficaz. Vamos conferir quais são eles!

CREATE INDEX: O índice é uma estrutura de dados usada para acelerar a pesquisa de dados em uma tabela. O PostgreSQL oferece suporte a vários tipos de índices, incluindo índices B-tree, índices hash e índices GIN. O comando **CREATE INDEX** é usado para criar um novo índice em uma tabela. Aqui está um exemplo:

```
CREATE INDEX sales_amount_idx ON sales (amount);
```

Esse comando cria um novo índice chamado "sales_amount_idx" na tabela "sales" para a coluna "amount".

CREATE FUNCTION: O PostgreSQL permite que os usuários criem funções e procedimentos armazenados personalizados para executar tarefas específicas. Essas funções podem ser chamadas de outras partes do aplicativo para executar tarefas específicas. O comando **CREATE FUNCTION** é usado para criar uma nova função no PostgreSQL. Aqui está um exemplo:

```
CREATE FUNCTION calculate_total_sales(salesperson
VARCHAR(255), start_date DATE, end_date DATE)
RETURNS DECIMAL AS $$
    SELECT SUM(amount) FROM sales WHERE salesperson = $1
    AND date >= $2 AND date <= $3
$$ LANGUAGE SQL;
```

Esse comando cria uma nova função chamada "calculate_total_sales", que recebe três parâmetros: o nome do vendedor, a data inicial e a data final. A função retorna o total de vendas desse vendedor dentro do intervalo de datas fornecido. A função é escrita em SQL, mas também é possível escrever funções em outras linguagens, como PL/pgSQL ou Python.

CREATE TABLE: O comando **CREATE TABLE** é usado para criar uma nova tabela no banco de dados. É necessário especificar o nome da tabela e as colunas que ela terá. Aqui está um exemplo:

```
CREATE TABLE employees (
    id SERIAL PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    age INTEGER,
    email VARCHAR(255) UNIQUE,
    hire_date DATE
);
```

Este comando cria uma nova tabela chamada "employees" com cinco colunas: id, name, age, email e hire_date. A coluna id é definida como uma chave primária e a coluna email é definida como única.

DROP TABLE: O comando **DROP TABLE** é usado para excluir uma tabela do banco de dados. É necessário especificar o nome da tabela que deseja excluir. Confira o exemplo:

```
DROP TABLE employees;
```

Este comando exclui a tabela "employees" do banco de dados.

DROP DATABASE: O comando **DROP DATABASE** é usado para excluir um banco de dados do PostgreSQL. É necessário especificar o nome do banco de dados que deseja excluir. Exemplo:

```
DROP DATABASE sales;
```

Este comando exclui o banco de dados "sales" do PostgreSQL.

ALTER TABLE: O comando **ALTER TABLE** é usado para modificar uma tabela existente no banco de dados. É possível adicionar, renomear ou excluir colunas, bem como modificar os tipos de dados ou as restrições de colunas existentes. Confira o exemplo a seguir:

```
ALTER TABLE employees ADD COLUMN department VARCHAR(255);
```

Com esses comandos básicos de criação de tabelas, criação e exclusão de bancos de dados, bem como alteração de tabelas, é possível gerenciar de forma eficaz seus dados no PostgreSQL. É importante conhecer esses comandos para poder utilizar todas as funcionalidades do PostgreSQL e manter seus dados organizados e seguros.

COMANDOS DE BASE DE DADOS



Na seção anterior, abordamos vários conceitos e exemplos sobre comandos de Banco de Dados no PostgreSQL. Agora, vamos incrementar esse conteúdo com os comandos essenciais para Base de Dados no PostgreSQL.

Existem vários comandos de base de dados no PostgreSQL usados para armazenar, recuperar, atualizar e excluir dados. Aqui estão alguns dos principais comandos:

SELECT: O comando **SELECT** é usado para recuperar dados de uma ou mais tabelas. É possível especificar quais colunas você deseja recuperar e aplicar filtros usando a cláusula *WHERE*. Exemplo:

```
SELECT id, name, email FROM employees WHERE age > 30;
```

Esse comando recupera os ids, nomes e endereços de e-mail de todos os funcionários com mais de 30 anos de idade da tabela "employees".

INSERT: O comando **INSERT** é usado para adicionar novas linhas a uma tabela existente. É necessário especificar o nome da tabela e os valores a serem inseridos nas colunas correspondentes. Confira um exemplo:

```
INSERT INTO employees (name, age, email, hire_date) VALUES ('John Smith', 35, 'john@example.com', '2022-01-01');
```

Esse comando adiciona uma nova linha à tabela "employees" com o nome "John Smith", idade de 35 anos, endereço de e-mail "john@example.com" e data de contratação "2022-01-01".

UPDATE: O comando **UPDATE** é usado para atualizar dados existentes em uma tabela. É necessário especificar a tabela que deseja atualizar, as colunas que

deseja modificar e os novos valores correspondentes. Exemplo:

```
DELETE FROM employees WHERE age > 65;
```

Esse comando exclui todas as linhas da tabela "employees" em que a idade dos funcionários é superior a 65 anos.

WHERE: A cláusula **WHERE** é usada para filtrar os resultados de uma consulta **SELECT** ou especificar quais linhas deseja modificar ou excluir em uma consulta **UPDATE** ou **DELETE**. É possível usar operadores como ">", "<", "=", "LIKE" e "IN" para criar filtros mais complexos. Exemplo:

```
SELECT name, email FROM employees WHERE hire_date BETWEEN '2022-01-01'  
AND '2022-06-30';
```

Esse comando calcula a soma dos salários de todos os funcionários em cada departamento da tabela "employees".

IN: O comando "IN" é usado para filtrar resultados baseados em um conjunto de valores. Ele permite que você especifique uma lista de valores possíveis para uma coluna em uma cláusula "WHERE". Por exemplo, se você deseja buscar todos os registros de uma tabela onde o valor da coluna "status" é "ativo" ou "pendente", você pode usar o comando "IN" da seguinte forma:

```
SELECT * FROM minha_tabela WHERE status IN ('ativo', 'pendente');
```

Nesse exemplo, estamos buscando todos os registros da tabela "minha_tabela" onde o valor da coluna "status" é "ativo" ou "pendente". O comando "IN" nos permite especificar uma lista de valores possíveis entre parênteses separados por vírgulas.

AND: O comando "AND" é usado para combinar duas ou mais condições em uma cláusula "WHERE". Ele retorna apenas os registros que atendem a todas as condições especificadas. Por exemplo, se você deseja buscar todos os registros de uma tabela onde o valor da coluna "status" é "ativo" e o valor da coluna "tipo" é "A", você pode usar o comando "AND" da seguinte forma:

```
SELECT * FROM minha_tabela WHERE status = 'ativo' AND tipo = 'A';
```

Nesse exemplo, estamos buscando todos os registros da tabela "minha_tabela" onde o valor da coluna "status" é "ativo" e o valor da coluna "tipo" é "A". O comando "AND" nos permite combinar duas ou mais condições em uma cláusula "WHERE" para refinar nossos resultados.

Esses são apenas alguns dos principais comandos de base de dados no PostgreSQL. Existem muitos outros comandos e funções disponíveis para ajudar a armazenar, recuperar e atualizar dados em suas tabelas do PostgreSQL. É

importante ter um bom entendimento desses comandos e como usá-los efetivamente para garantir que os dados sejam armazenados com precisão e eficiência.

Ao criar um banco de dados no PostgreSQL, é importante planejar sua estrutura cuidadosamente e garantir que suas tabelas sejam normalizadas adequadamente para evitar a duplicação de dados e preservar a integridade dos dados.

Neste e-book, exploramos os fundamentos do PostgreSQL, um dos sistemas gerenciadores de banco de dados mais populares e poderosos disponíveis atualmente. Aprendemos sobre a estrutura de dados flexível do PostgreSQL, que permite armazenar e manipular diversos tipos de dados, e como utilizar o sistema para análise de dados.

Consideramos como criar e gerenciar tabelas e bancos de dados no PostgreSQL, usando comandos como **"CREATE TABLE"**, **"CREATE DATABASE"** e **"DROP TABLE"**. A capacidade de criar e gerenciar esses objetos é fundamental para trabalhar com bancos de dados, e o PostgreSQL oferece uma ampla gama de recursos para simplificar esses processos. Além disso, exploramos os principais comandos de banco de dados e de base de dados do PostgreSQL, incluindo a criação e gerenciamento de tabelas, funções e procedimentos armazenados, bem como comandos como **"SELECT"**, **"INSERT"**, **"UPDATE"**, **"DELETE"**, **"WHERE"**, **"IN"** e **"AND"**.

Agora, você tem uma compreensão básica do PostgreSQL e está pronto para começar a trabalhar com bancos de dados. Lembre-se de que, assim como em qualquer outra área, a prática é fundamental para aprimorar suas habilidades. Continue explorando e aprofundando seus conhecimentos sobre o PostgreSQL e seus comandos para se tornar especialista em banco de dados.



Para refletir

- Como o PostgreSQL se compara a outros sistemas de gerenciamento de banco de dados relacionais em termos de recursos e funcionalidades?
- Como você pode usar o PostgreSQL para análise de dados? Quais são as funcionalidades e recursos que tornam o PostgreSQL uma boa escolha para análise de dados?
- Quais são os principais comandos de base de dados que você precisa conhecer para usar o PostgreSQL efetivamente?



Atividade: Consultando dados em uma tabela no PostgreSQL

Você foi contratado para trabalhar em um projeto que utiliza o PostgreSQL como banco de dados. Para começar, é preciso que você saiba como consultar dados em uma tabela. Sua tarefa é escrever uma consulta SQL que retorne todos os dados da tabela "clientes".

Passo a passo:

1. Abra o terminal do PostgreSQL ou a ferramenta de gerenciamento que preferir;
2. Conecte-se ao banco de dados que contém a tabela "clientes";
3. Digite o comando SQL para selecionar todos os dados da tabela "clientes";
4. Execute a consulta SQL;
5. Verifique se a consulta retorna todos os dados da tabela.

[Clique aqui para acessar o qabarito.](#)

Para ir além



- Leia a **documentação oficial**: o site oficial do PostgreSQL fornece documentação completa e detalhada, que vai ajudar você a entender melhor como o banco de dados funciona. Você pode acessá-la em:
<https://www.postgresql.org/docs/current/index.html>
- Junte-se à **comunidade**: existem muitas comunidades de usuários de PostgreSQL online, onde você pode encontrar ajuda e recursos adicionais. Alguns dos mais populares incluem o fórum oficial do PostgreSQL [PostgreSQL: Community](#) e o subreddit do PostgreSQL [PostgreSQL \(reddit.com\)](#).

RESUMO DE ESTRUTURAS DE POSTGRESQL

PostgreSQL: Sistema de gerenciamento de banco de dados relacional de código aberto.

Introdução

Ele é conhecido por sua capacidade de lidar com grandes quantidades de dados e por sua confiabilidade, segurança e flexibilidade. Como um banco de dados relacional, o PostgreSQL é baseado no modelo de tabela, onde os dados são organizados em tabelas compostas por linhas e colunas.

Contextualização

É popular no mercado de análise de dados devido à sua capacidade de lidar com grandes volumes de dados e flexibilidade para manipulação de dados. Também possui a capacidade de ampliar suas funcionalidades por meio de extensões, permitindo que os usuários possam criar funções personalizadas para manipulação de dados e estatísticas.

Fundamentos de PostgreSQL: O PostgreSQL é um dos sistemas gerenciadores de banco de dados (SGBD) mais populares e robustos do mundo.

Instalando PostgreSQL

- Baixe o instalador do PostgreSQL para Windows no site oficial do PostgreSQL: <https://www.postgresql.org/download/windows/>
- Execute o arquivo de instalação e siga as instruções na tela para instalar o PostgreSQL.
- Quando a instalação estiver concluída, você pode verificar se o PostgreSQL está funcionando corretamente iniciando o serviço do PostgreSQL.

No Ubuntu, você pode instalar o PostgreSQL usando o seguinte comando:

```
sudo apt-get install postgresql
```

Estrutura de Dados do PostgreSQL

Uma das principais razões pelas quais o PostgreSQL é tão popular é sua estrutura de dados altamente flexível. Ele suporta vários tipos de dados, incluindo inteiros, decimais, texto, data/hora, booleanos, JSON e muitos outros.

Consultas SQL Avançadas

O PostgreSQL suporta a linguagem SQL (Structured Query Language), que é a linguagem padrão para consulta e manipulação de bancos de dados.

Extensões do PostgreSQL

O PostgreSQL suporta extensões, que são módulos adicionais que fornecem recursos adicionais ao sistema.

Recursos Adicionais

Índices de texto completos

ACID (Atomicidade, Consistência, Isolamento e Durabilidade)

Compatível com vários sistemas operacionais, incluindo Windows, Linux e macOS

Escalabilidade horizontal

Comandos de Base de Dados: Principais comandos de base de dados do PostgreSQL; essenciais.

SELECT

É usado para recuperar dados de uma ou mais tabelas. Exemplo:

```
SELECT id, name, email FROM employees WHERE age > 30;
```

INSERT

É usado para adicionar novas linhas a uma tabela existente. Exemplo:

```
INSERT INTO employees (name, age, email, hire_date) VALUES ('John Smith', 35, 'john@example.com', '2022-01-01');
```

UPDATE

É usado para atualizar dados existentes em uma tabela. Exemplo:

```
DELETE FROM employees WHERE age > 65;
```

WHERE

É usada para filtrar os resultados de uma consulta SELECT ou especificar quais linhas deseja modificar ou excluir em uma consulta UPDATE ou DELETE. Exemplo:

```
SELECT name, email FROM employees WHERE hire_date BETWEEN '2022-01-01' AND '2022-06-30';
```

IN

É usado para filtrar resultados baseados em um conjunto de valores. Exemplo:

```
SELECT * FROM minha_tabela WHERE status IN ('ativo', 'pendente');
```

AND

É usado para combinar duas ou mais condições em uma cláusula "WHERE". Exemplo:

```
SELECT * FROM minha_tabela WHERE status = 'ativo' AND tipo = 'A';
```

RESUMO DE ESTRUTURAS DE POSTGRESQL

Comandos de Banco de Dados: Principais comandos de banco de dados do PostgreSQL que ajudarão os usuários a gerenciar seus dados de forma eficaz.

CREATE INDEX

É usado para criar um novo índice em uma tabela. Exemplo:

```
CREATE INDEX sales_amount_idx ON sales (amount);
```

CREATE FUNCTION

É usado para criar uma nova função no PostgreSQL. Exemplo:

```
CREATE FUNCTION calculate_total_sales (salesperson VARCHAR(255), start_date DATE, end_date DATE) RETURNS DECIMAL AS $$ SELECT SUM(amount) FROM sales WHERE salesperson = $1 AND date >= $2 AND date <= $3 $$ LANGUAGE SQL;
```

CREATE TABLE

É usado para criar uma nova tabela no banco de dados. Exemplo:

```
CREATE TABLE employees ( id SERIAL PRIMARY KEY, name VARCHAR(255) NOT NULL, age INTEGER, email VARCHAR(255) UNIQUE, hire_date DATE );
```

DROP TABLE

É usado para excluir uma tabela do banco de dados. Exemplo:

```
DROP TABLE employees;
```

DROP DATABASE

É usado para excluir um banco de dados do PostgreSQL. Exemplo:

```
DROP DATABASE sales;
```

ALTER TABLE

É usado para modificar uma tabela existente no banco de dados. Exemplo:

```
ALTER TABLE employees ADD COLUMN department VARCHAR(255);
```



**Até a próxima e
#confianoprocesso**

