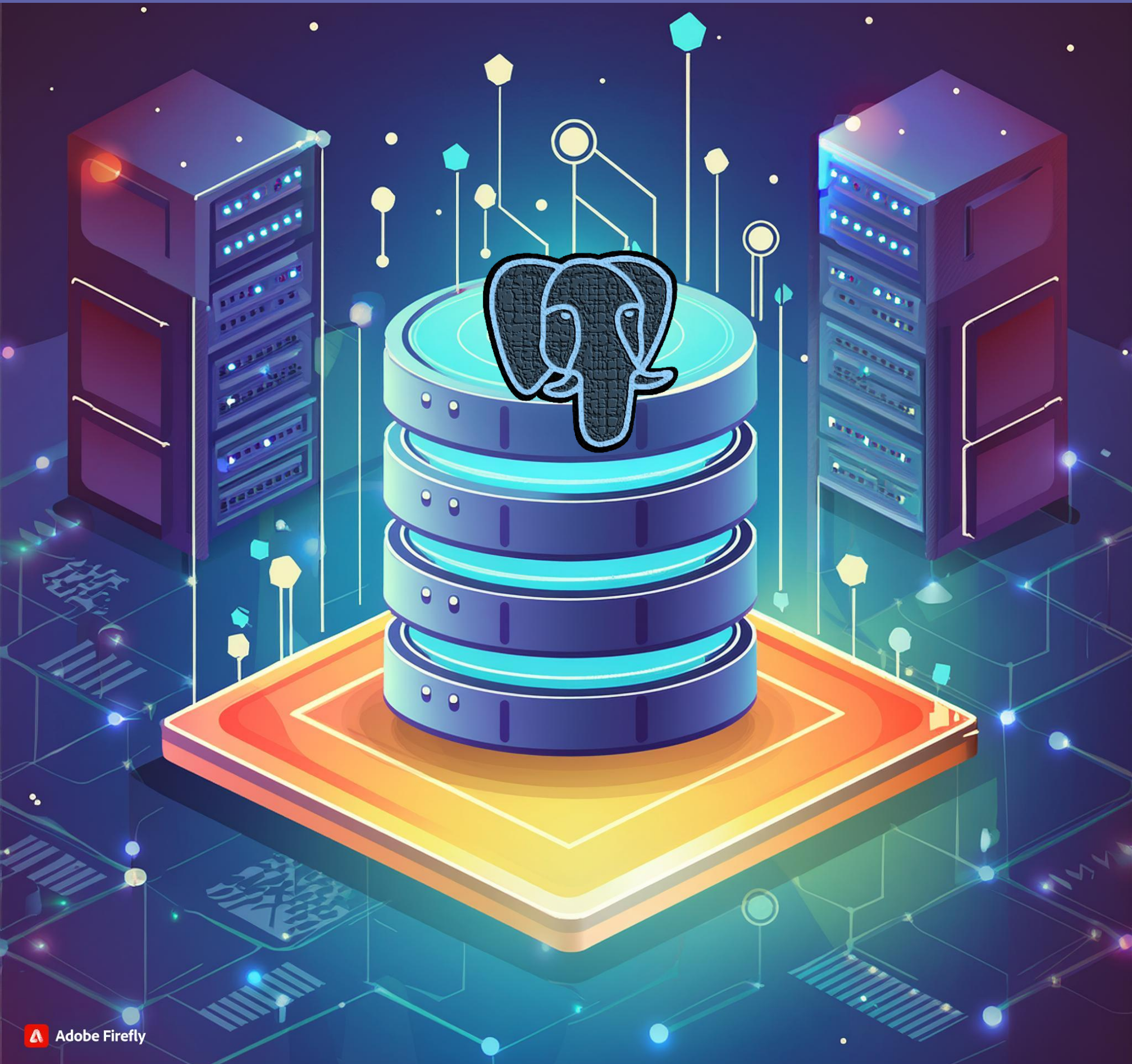
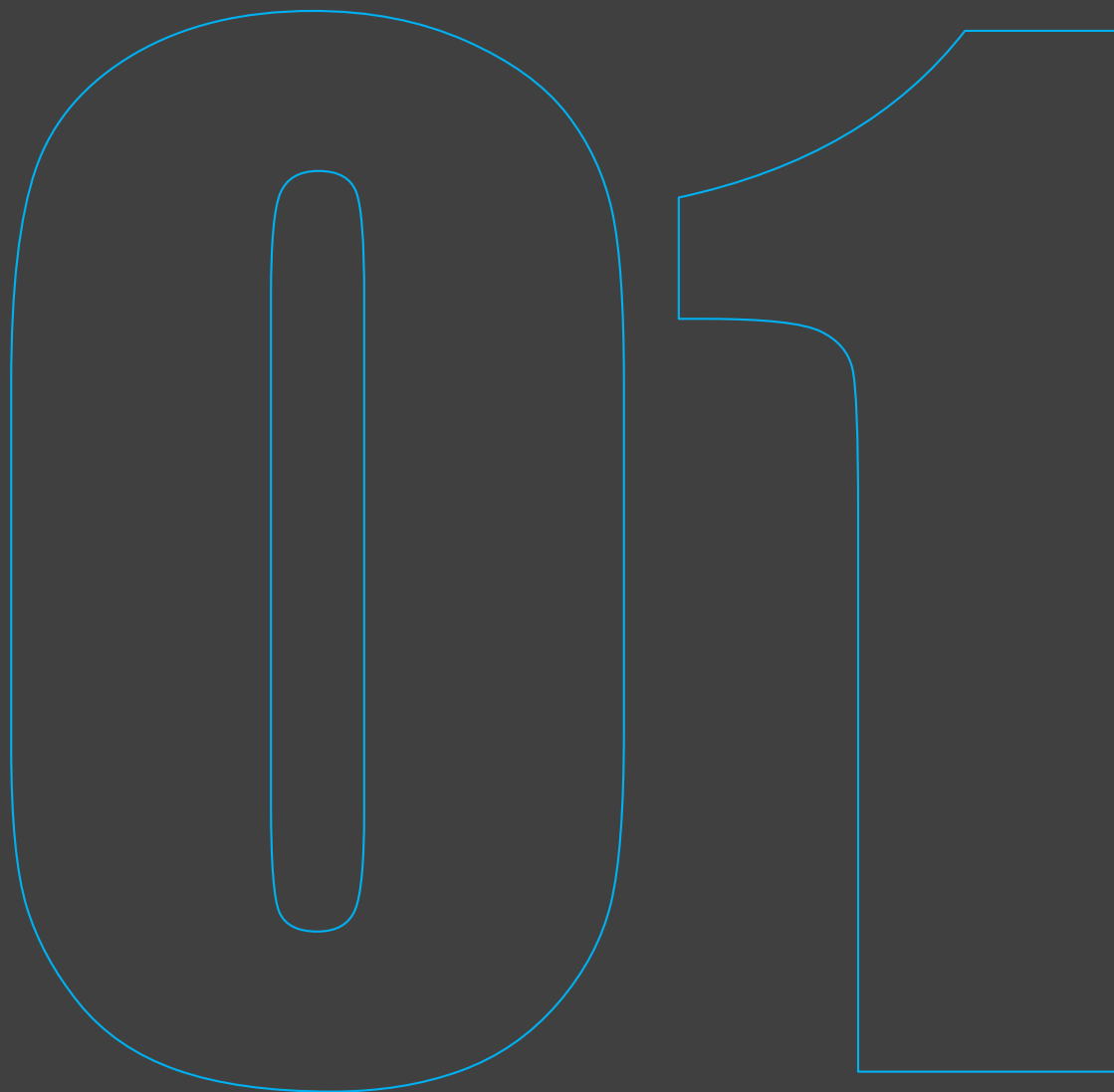


PostgreSQL

A Saga dos Dados Imortais





CARACTERÍSTICAS DO POSTGRESQL

CARACTERÍSTICAS DO PSQL

O PostgreSQL é uma escolha confiável para uma variedade de aplicativos, desde pequenos projetos até sistemas empresariais robustos, pois ele é:

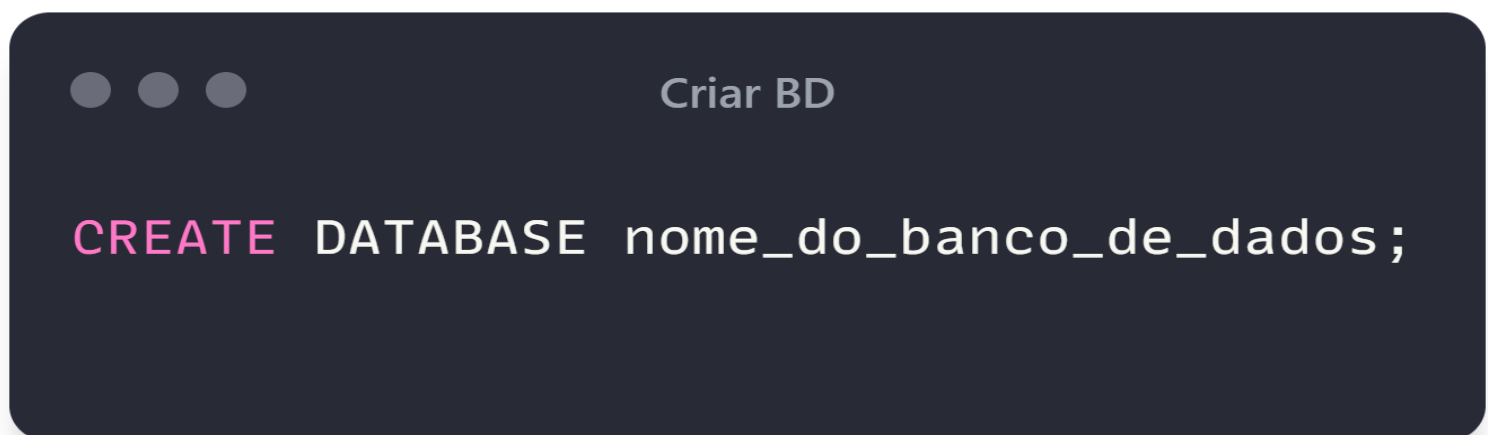
- **Open Source:** O PostgreSQL é um sistema de gerenciamento de banco de dados relacional de código aberto.
- **Conformidade com SQL:** Rigorosamente aderente aos padrões SQL ANSI, facilitando a migração de dados e a escrita de consultas compatíveis.
- **Extensibilidade:** Permite aos usuários estender suas funcionalidades através da criação de novos tipos de dados, funções e operadores.
- **Recursos Avançados:** Suporta transações ACID, replicação síncrona e assíncrona, particionamento de tabelas e manipulação eficiente de dados JSON e XML.
- **Desempenho:** Otimizações avançadas de consulta e índices eficientes garantem um desempenho excepcional mesmo em grandes conjuntos de dados.

02

COMANDOS BÁSICOS

CRIAR BANCO DE DADOS E TABELAS

Para criar um novo banco de dados, utilize o comando `CREATE DATABASE` seguido pelo nome desejado para o banco de dados.



```
CREATE DATABASE nome_do_banco_de_dados;
```

As tabelas são onde os dados são armazenados. Utilize o comando `CREATE TABLE` para criar uma nova tabela, especificando os nomes das colunas e seus tipos de dados.



```
CREATE TABLE usuarios (  
    id SERIAL PRIMARY KEY,  
    nome VARCHAR(100),  
    idade INT  
);
```

Comandos PSQL- Heloísa Silva

MANIPULAÇÃO DE DADOS

Para inserir dados em uma tabela, utilize o comando INSERT INTO, especificando o nome da tabela e os valores a serem inseridos.

```
Insertando dados na tabela

INSERT INTO usuarios (nome, idade)
VALUES ('João', 30);
```

Para recuperar dados de uma tabela, utilize o comando SELECT, especificando as colunas desejadas e, opcionalmente, condições de filtro.

```
Consultar dados na tabela

SELECT * FROM usuarios WHERE idade > 25;
```

MANIPULAÇÃO DE DADOS

Para atualizar registros em uma tabela, utilize o comando UPDATE, especificando a tabela, os novos valores e, opcionalmente, uma cláusula WHERE para especificar quais registros atualizar..



Atualizar dados na tabela

```
UPDATE usuarios SET idade = 35 WHERE  
nome = 'João';
```

Para excluir registros de uma tabela, utilize o comando DELETE FROM, especificando a tabela e, opcionalmente, uma cláusula WHERE para especificar quais registros excluir.



Excluir dados na tabela

```
DELETE FROM usuarios WHERE idade > 40;
```

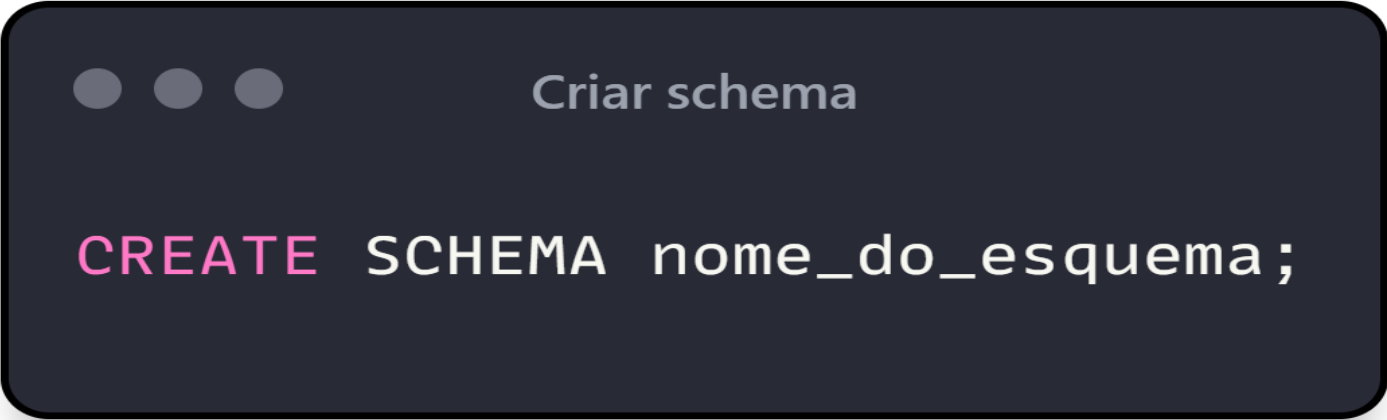
03

COMANDOS AVANÇADOS

Guia de Comandos PostgreSQL: Conceitos avançados

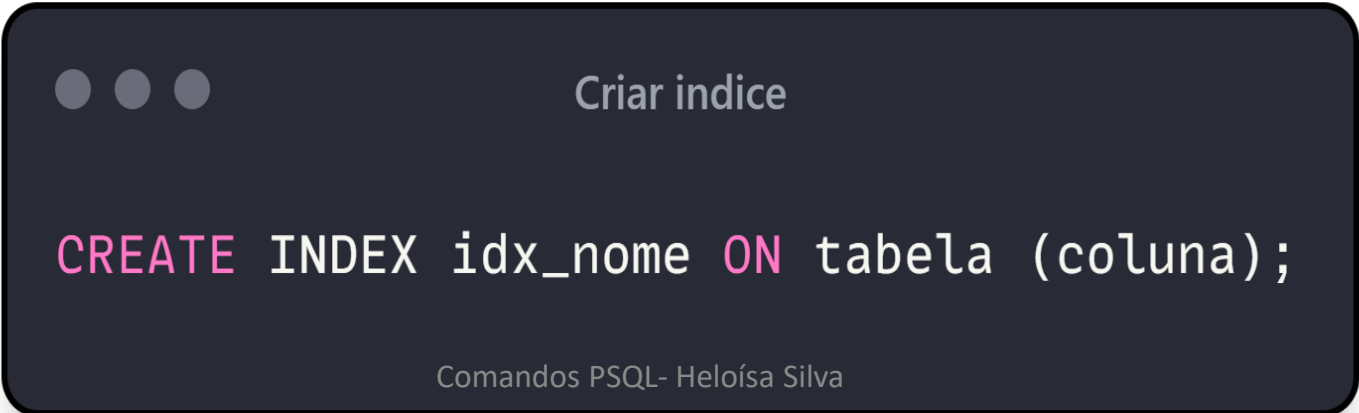
Criando Schemas e Índices

No PostgreSQL, os esquemas são usados para organizar objetos de banco de dados em grupos lógicos. Eles são úteis para dividir e isolar objetos, evitando conflitos de nome. Para criar um esquema, utilize o comando CREATE SCHEMA.



```
● ● ● Criar schema  
  
CREATE SCHEMA nome_do_esquema;
```

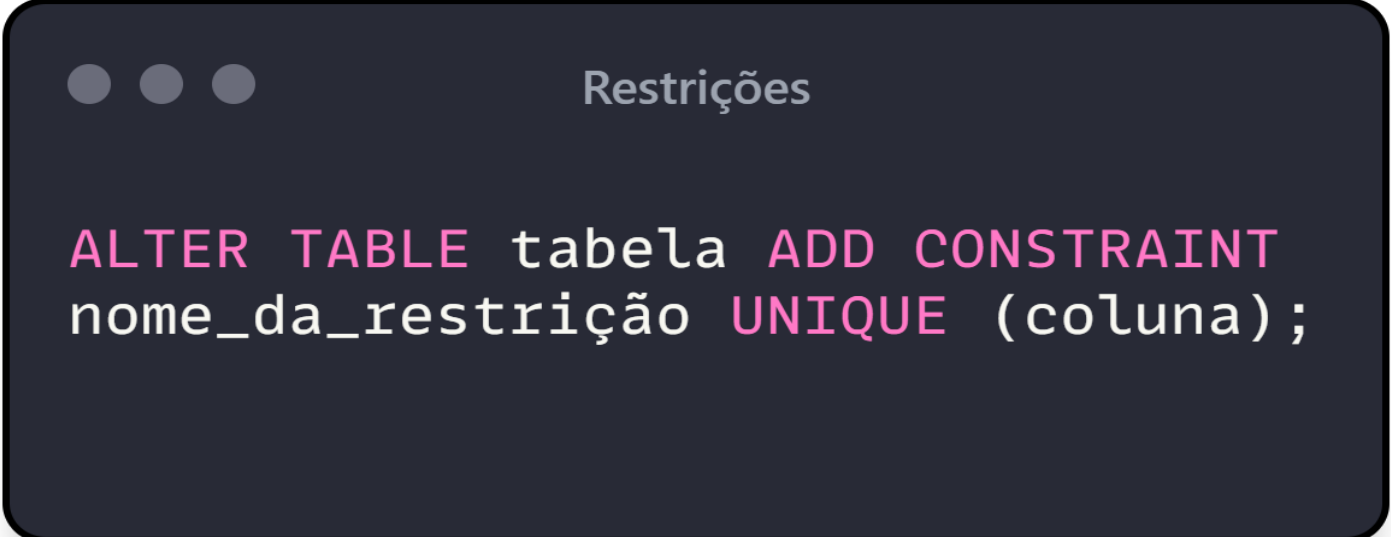
Os índices são estruturas que aceleram a recuperação de dados em tabelas. Eles podem ser criados em colunas específicas para melhorar o desempenho das consultas. Para criar um índice, utilize o comando CREATE INDEX..



```
● ● ● Criar indice  
  
CREATE INDEX idx_nome ON tabela (coluna);
```

RESTRIÇÕES E TRANSAÇÕES

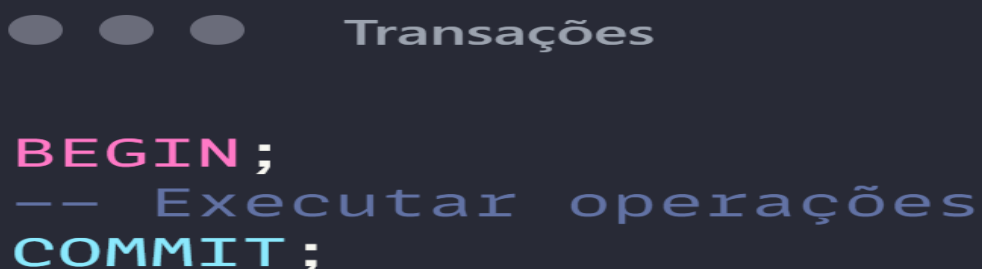
As restrições são regras aplicadas aos dados em uma tabela para garantir sua integridade. Algumas das restrições mais comuns incluem PRIMARY KEY, FOREIGN KEY, UNIQUE e CHECK. Vejamos um exemplo de criação de uma restrição UNIQUE.



```
Restrições

ALTER TABLE tabela ADD CONSTRAINT
nome_da_restrição UNIQUE (coluna);
```

As transações no PostgreSQL garantem a atomicidade, consistência, isolamento e durabilidade (ACID) das operações no banco de dados. O comando BEGIN inicia uma nova transação, COMMIT confirma a transação e ROLLBACK a reverte.



```
Transações

BEGIN;
-- Executar operações
COMMIT;
```

FUNÇÕES

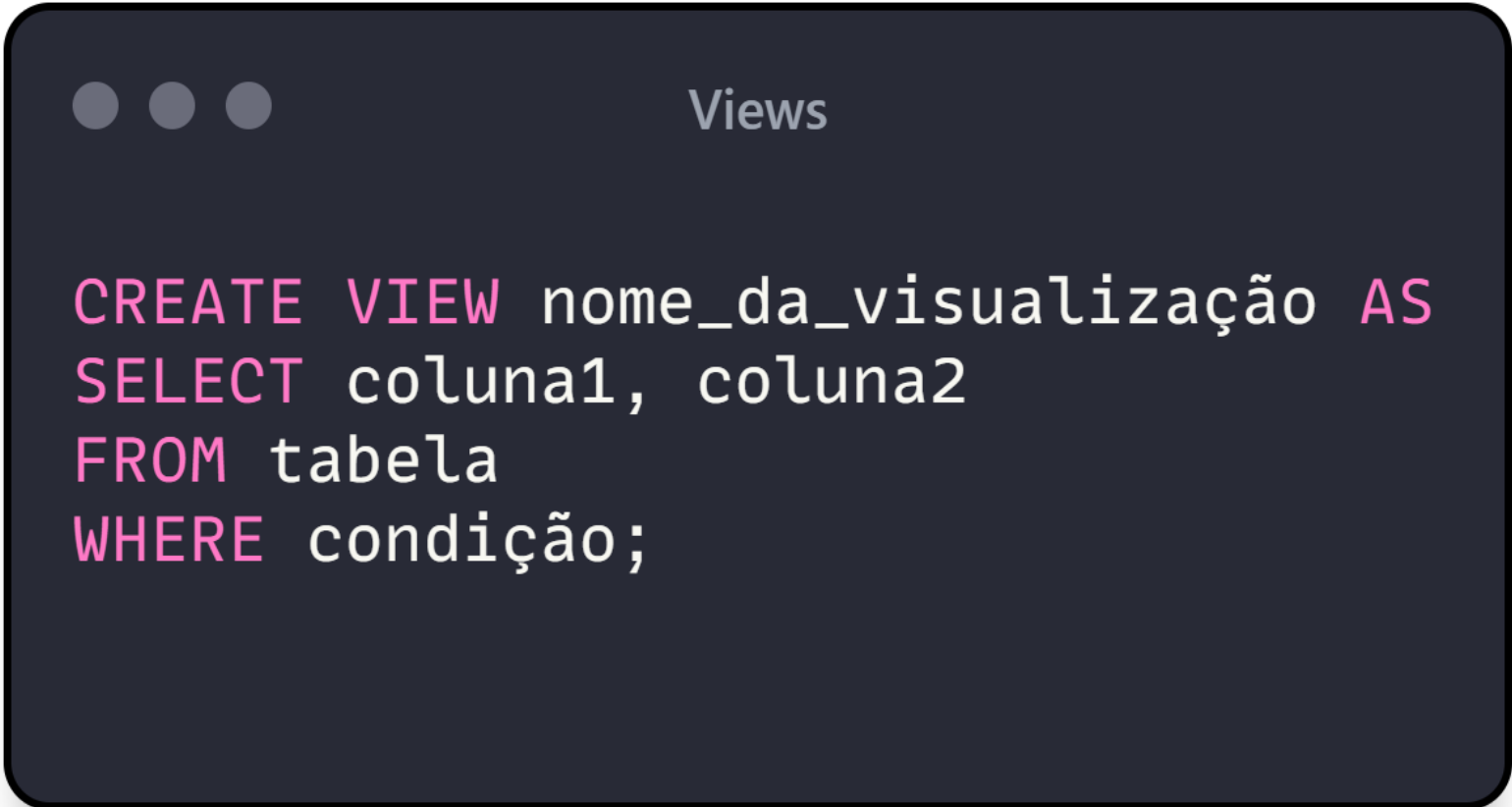
As funções armazenadas são blocos de código SQL que podem ser reutilizados em várias consultas. Elas aumentam a modularidade e a reutilização do código. Vejamos um exemplo de criação de uma função.

```
Funções

CREATE FUNCTION nome_da_função(parametro tipo)
RETURNS tipo_retorno AS $$
DECLARE
    variavel tipo;
BEGIN
    -- Lógica da função
END;
$$ LANGUAGE plpgsql;
```

VIEWS

As visualizações são consultas armazenadas que aparecem como tabelas virtuais. Elas simplificam consultas complexas e ocultam a complexidade dos dados subjacentes. Para criar uma visualização, utilize o comando CREATE VIEW.

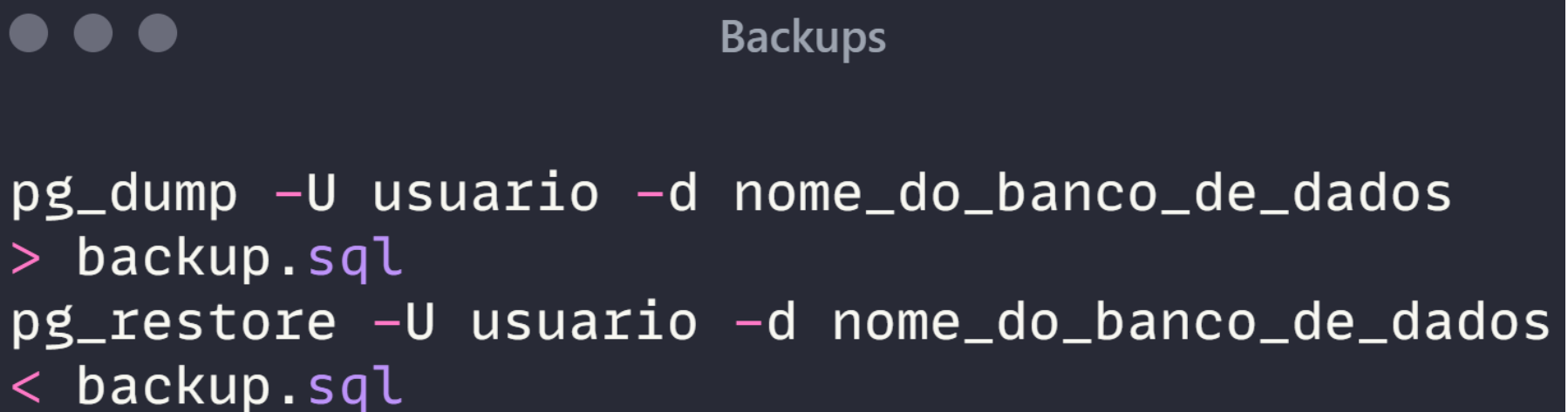
A dark-themed terminal window with a title bar containing three window control buttons and the title "Views". The terminal displays the following SQL code:

```
CREATE VIEW nome_da_visualização AS  
SELECT coluna1, coluna2  
FROM tabela  
WHERE condição;
```

```
CREATE VIEW nome_da_visualização AS  
SELECT coluna1, coluna2  
FROM tabela  
WHERE condição;
```

Backups

Fazer backups regulares do banco de dados é essencial para garantir a segurança dos dados. Utilize os utilitários `pg_dump` e `pg_restore` para fazer e restaurar backups.



```
pg_dump -U usuario -d nome_do_banco_de_dados  
> backup.sql  
pg_restore -U usuario -d nome_do_banco_de_dados  
< backup.sql
```

CONCLUSÃO

CONCLUSÃO

Desbravando as possibilidades com o PSQL

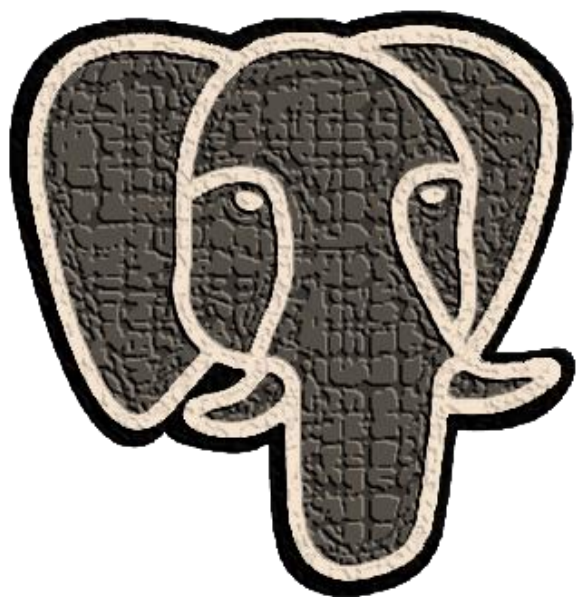
Nesta jornada através do PostgreSQL, é mais do que apenas um banco de dados, é uma solução robusta e confiável para uma variedade de necessidades de gerenciamento de dados. Sua natureza de código aberto, conformidade com padrões, extensibilidade e desempenho excepcional o tornam uma escolha popular para projetos de todos os tamanhos. Com uma comunidade ativa oferecendo suporte contínuo, o PostgreSQL continua a ser uma ferramenta essencial para desenvolvedores e empresas que buscam soluções de banco de dados eficazes.

AGRADECIMENTOS

OBRIGADO POR LER ATÉ AQUI!

Este ebook foi criado para fins didáticos, para o bootcamp do Santander na DIO.

Quero agradecer os ensinamentos dos professores e a oportunidade e experiência que está sendo proporcionada.



PostgreSQL