



**Digital
College**

ENSINO DE HABILIDADES DIGITAIS

Formação em Data Analytics

Aula 02





Principais Características - PostgreSQL

- SGBD **Objeto-Relacional**;
- Um dos mais avançados **SGBDs open-source e freeware** disponíveis;
- Suporta **muitas plataformas**, entre elas o Windows, Linux, Solaris, AIX, OpenBSD dentre outros;
- **Alto desempenho**: suporta intenso fluxo de dados de maneira segura;
- **Software livre**: permite que os usuários possam ser inovadores e flexíveis aproveitando, também, a ausência de custo de licenças;
- **Escalabilidade**: o software gerencia grandes volumes de dados de maneira fácil;
- **Cloud Computing**: o PostgreSQL é integrado com essa tecnologia, o que permite o acesso à informação quando e onde precisar.

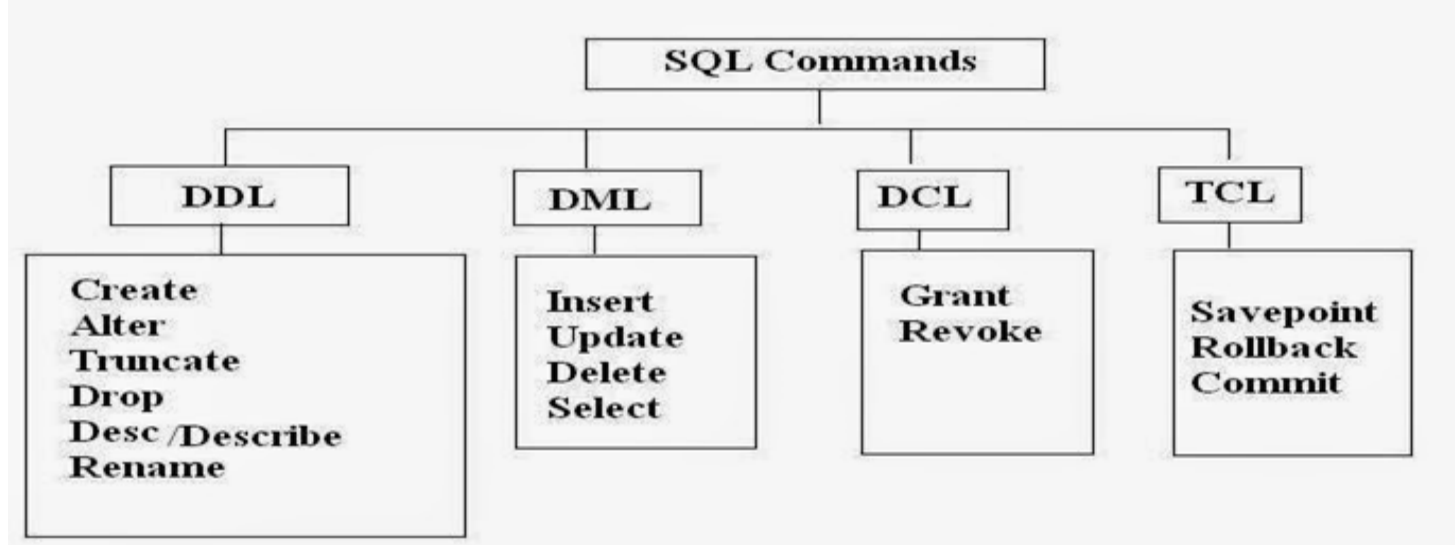
Interfaces de programação suportadas no PostgreSQL

Interfaces de Programação Cliente:

1. **JDBC** - *Java Database Connectivity* ou *JDBC* é um conjunto de classes e interfaces escritas em Java que fazem o envio de instruções SQL para qualquer banco de dados relacional;
1. **ODBC** - *Open Database Connectivity* ou *ODBC* é uma especificação de interface para acesso a dados.
1. **C**
1. **C++**
1. **Python**
1. **Php**



Divisões da Linguagem SQL



- **DDL** – Linguagem de Definição de Dados.
- **DML** – Linguagem de Manipulação de Dados.
- **DCL** – Linguagem de Controle de Dados (autorização de dados e licença de usuários para controlar quem tem acesso aos dados).
- **TCL** – Transação de Controle da Linguagem (são usados para controlar transações em um banco de dados).

DDL - Linguagem de Definição de Dados:

CREATE table

cria uma tabela no banco de dados.

DROP table

Apaga/exclui uma tabela do banco de dados.

ALTER table

Altera a estrutura de uma tabela no banco de dados.

TRUNCATE table

Usada para excluir todas as linhas de uma tabela em um banco de dados.



DML - Linguagem de Manipulação de Dados

SELECT

Permite ao usuário especificar uma consulta com uma descrição do resultado desejado.

INSERT

Incluir um registro em uma tabela.

UPDATE

Alterar os valores dos dados de uma tabela.

DELETE

Remover registros de uma tabela.



Tipos de Dados - PostgreSQL

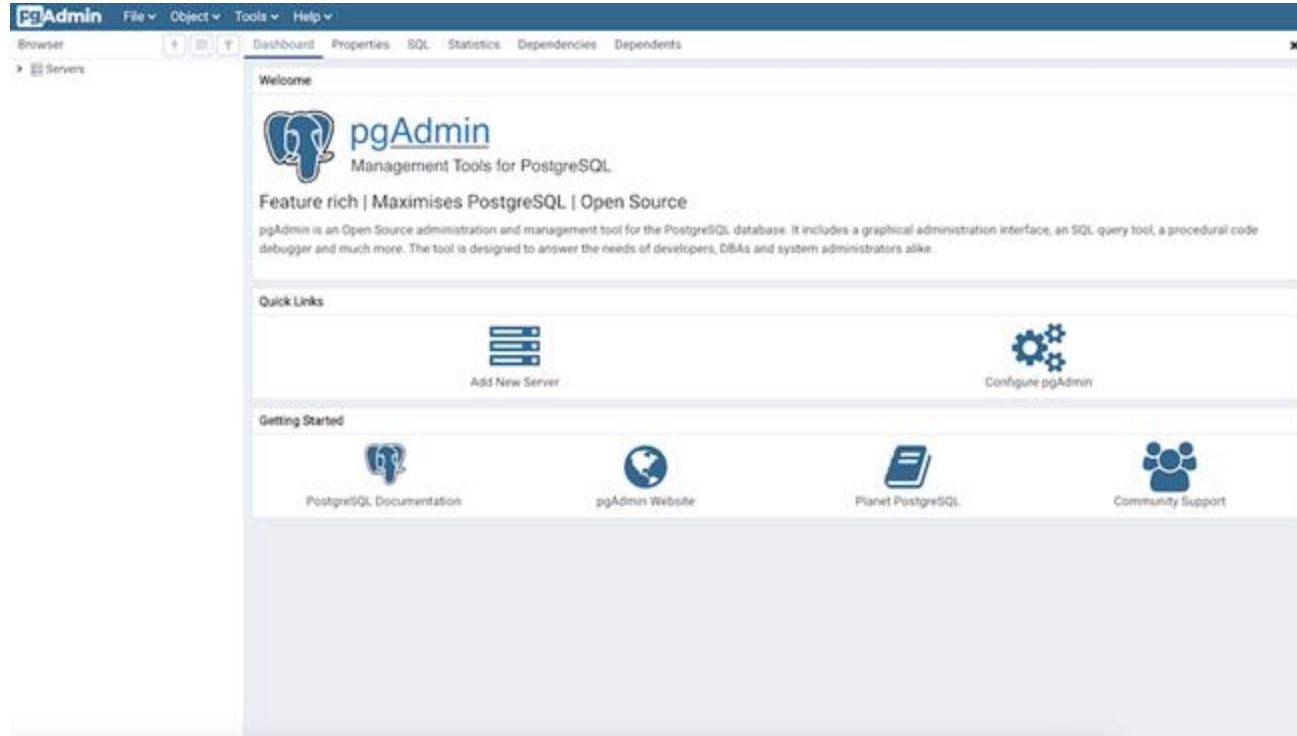
- Além de um nome as colunas devem ser de um tipo específico.
- Os tipos de dados disponíveis no PostgreSQL são:

Tipo	Representação
Integer	4 Bytes
Bigint	8 Bytes
Numeric	Numeric(p, d)
Float	8 Bytes
Char(n)	String de tamanho fixo
Character Varying	String de tamanho variável
Date	Data, 4 Bytes



PgAdmin

- Ferramenta de acesso ao banco de dados



SQL – Structured Query Language

- Definição do Esquema = Criação de Schema e das tabelas.
- Esquema: formado pela base de dados, esquema e tabela
- Create SCHEMA < ... >



SQL – Structured Query Language

- Definição do Esquema = Criação da Tabela:

- Comando: **CREATE TABLE**

```
CREATE TABLE nome-tabela  
(  COLUNA_01 TIPO,  
  COLUNA_02 TIPO,  
  COLUNA_03 TIPO,  
  COLUNA_04 TIPO...);
```



Vamos Praticar?

1. Criar o seguinte banco de dados: digital

```
CREATE DATABASE digital;
```

2. Criar esquema: rh

```
CREATE SCHEMA IF NOT EXISTS rh;
```

3. No esquema rh, criar a tabela: pessoa

```
CREATE TABLE IF NOT EXISTS rh.pessoa  
(id serial,  
nome character varying,  
cpf character varying,  
email character varying,  
nascimento date,  
renda numeric(18,2),  
casa_propria boolean,  
CONSTRAINT pessoa_pkey PRIMARY KEY (id));
```



Definindo e executando comandos SQL

Clique em “Databases” e depois em “postgres”

Você verá que o ícone “SQL” (dentro da lupa) na barra de ferramentas está ativado.

Clique neste ícone “SQL”





Servers

PostgreSQL 14

Databases (29)

- > biblioteca
- > corporativo
- > corporativo4
- > data_warehouse
- > data_warehouse_trab

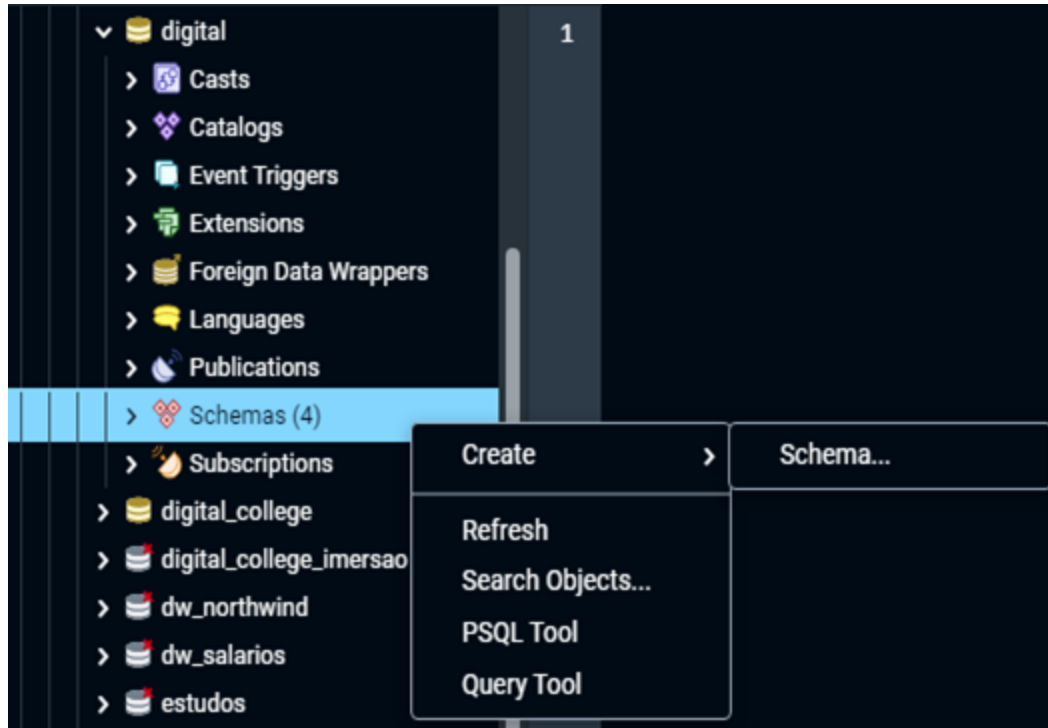
digital

- > Casts
- > Catalogs
- > Event Triggers
- > Extensions
- > Foreign Data Wrappers
- > Languages
- > Publications
- > Schemas (4)

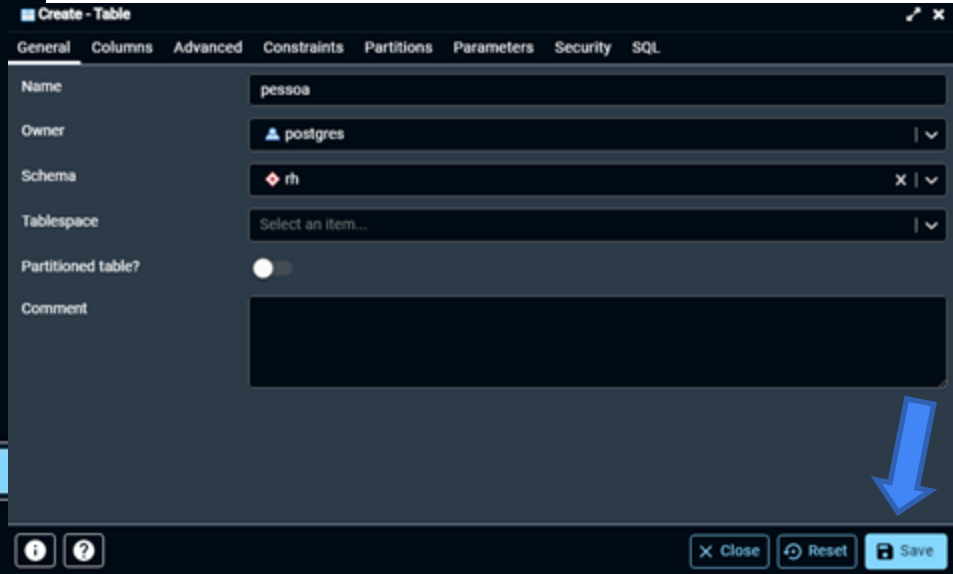
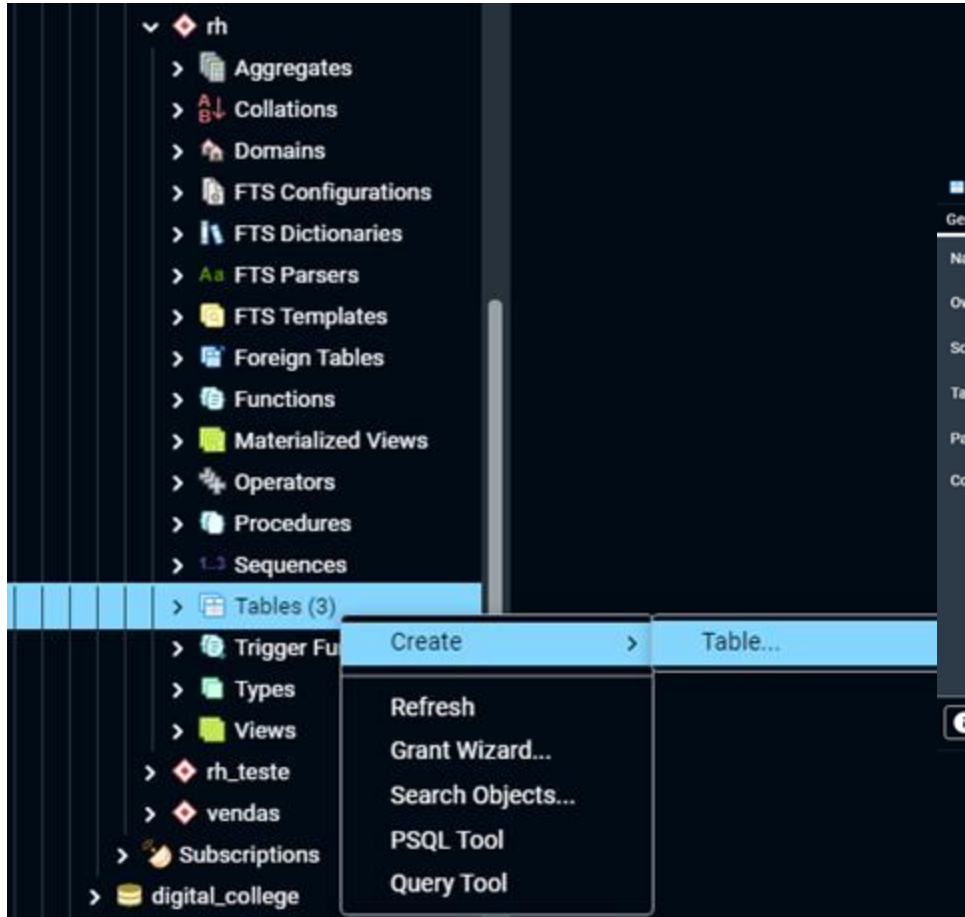
```
1  -- Database: digital
2
3  -- DROP DATABASE IF EXISTS digital;
4
5  CREATE DATABASE digital
6      WITH
7      OWNER = postgres
8      ENCODING = 'UTF8'
9      LC_COLLATE = 'Portuguese_Brazil.1252'
10     LC_CTYPE = 'Portuguese_Brazil.1252'
11     TABLESPACE = pg_default
12     CONNECTION LIMIT = -1;
```



Criando Esquema: rh



Criando no Esquema rh, tabela pessoa



Criando no Esquema rh, tabela pessoa

Create - Table

General Columns Advanced Constraints Partitions Parameters Security SQL

Name:

Owner:

Schema:

Tablespace:

Partitioned table? ☐

Comment:

Create - Table

General Columns Advanced Constraints Partitions Parameters Security SQL

Inherited from table(s):

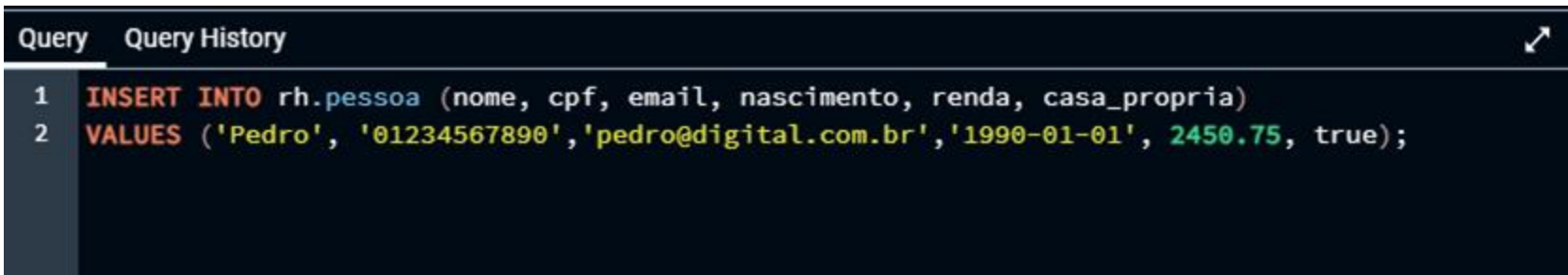
Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
<input type="checkbox"/>	id	serial			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input type="checkbox"/>	nome	character varying			<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	cpf	character varying			<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	email	character varying			<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	nascimento	date			<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	renda	numeric	18	2	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/>	casa_propria	boolean			<input type="checkbox"/>	<input type="checkbox"/>	

Armazenando Valores

- Adicionar dados:
 - Comando: **INSERT INTO**

```
INSERT INTO tabela (A1, A2, A3...)  
VALUES (V_A1, V_A2, V_A3...);
```



The screenshot shows a SQL query editor with a dark background. At the top, there are two tabs: 'Query' and 'Query History'. The 'Query' tab is active, showing a SQL statement with line numbers 1 and 2 on the left. The statement is: `1 INSERT INTO rh.pessoa (nome, cpf, email, nascimento, renda, casa_propria)` and `2 VALUES ('Pedro', '01234567890', 'pedro@digital.com.br', '1990-01-01', 2450.75, true);`. The text is color-coded: 'INSERT INTO' is orange, 'rh.pessoa' is blue, and 'VALUES' is orange. The values are in various colors: 'Pedro' is red, '01234567890' is red, 'pedro@digital.com.br' is green, '1990-01-01' is green, '2450.75' is green, and 'true' is red. A small icon of a cursor is visible in the top right corner of the editor area.

```
Query  Query History  
1  INSERT INTO rh.pessoa (nome, cpf, email, nascimento, renda, casa_propria)  
2  VALUES ('Pedro', '01234567890', 'pedro@digital.com.br', '1990-01-01', 2450.75, true);
```

Consultando Valores

Query

Query History

1

SELECT * FROM rh.pessoa;

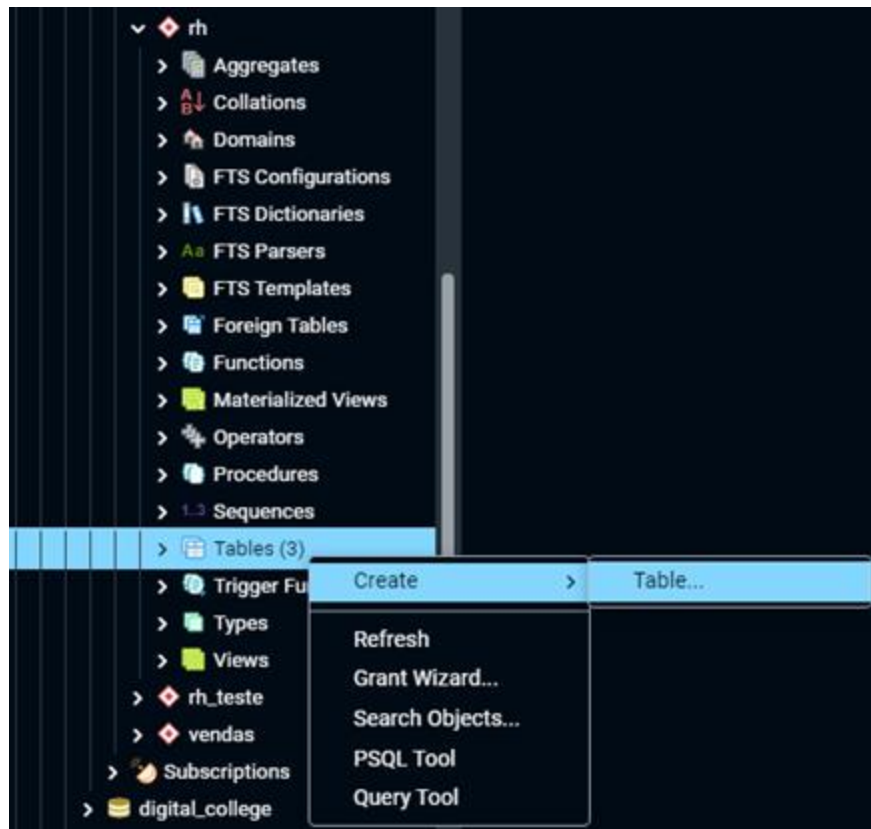
Data output

Messages

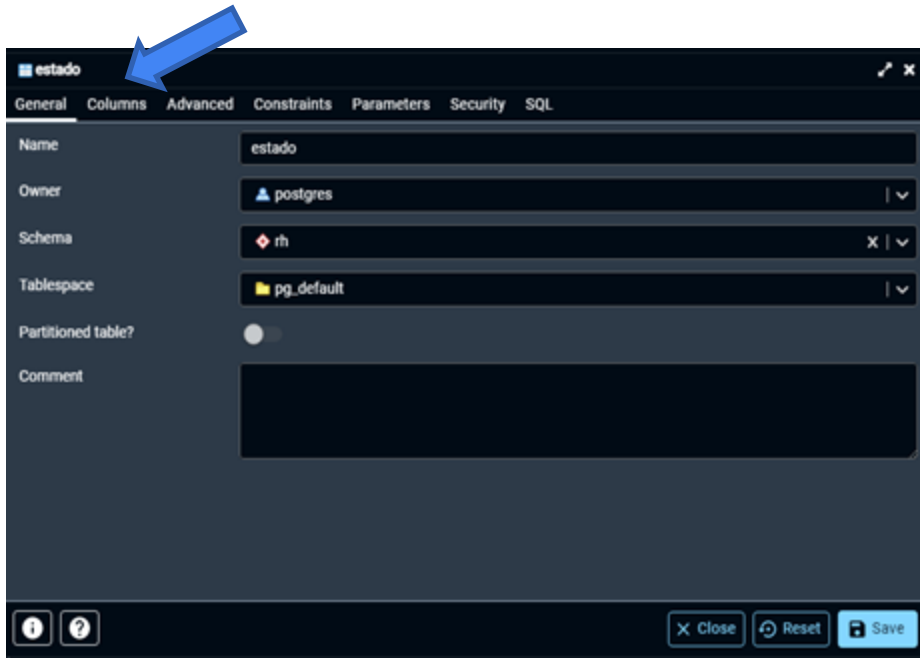
Notifications

	<div><div>id</div><div>[PK] integer</div></div>	<div><div>nome</div><div>character varying</div></div>	<div><div>cpf</div><div>character varying</div></div>	<div><div>email</div><div>character varying</div></div>	<div><div>nascimento</div><div>date</div></div>	<div><div>renda</div><div>numeric (18,2)</div></div>	<div><div>casa_propria</div><div>boolean</div></div>	
1	1	Pedro	01234567890	pedro@digital.co...	1990-01-01	2450.75	true	

Criando uma tabela através de interface gráfica



No esquema rh, criar tabela: estado



estado

General Columns Advanced Constraints Parameters Security SQL

Name: estado

Owner: postgres

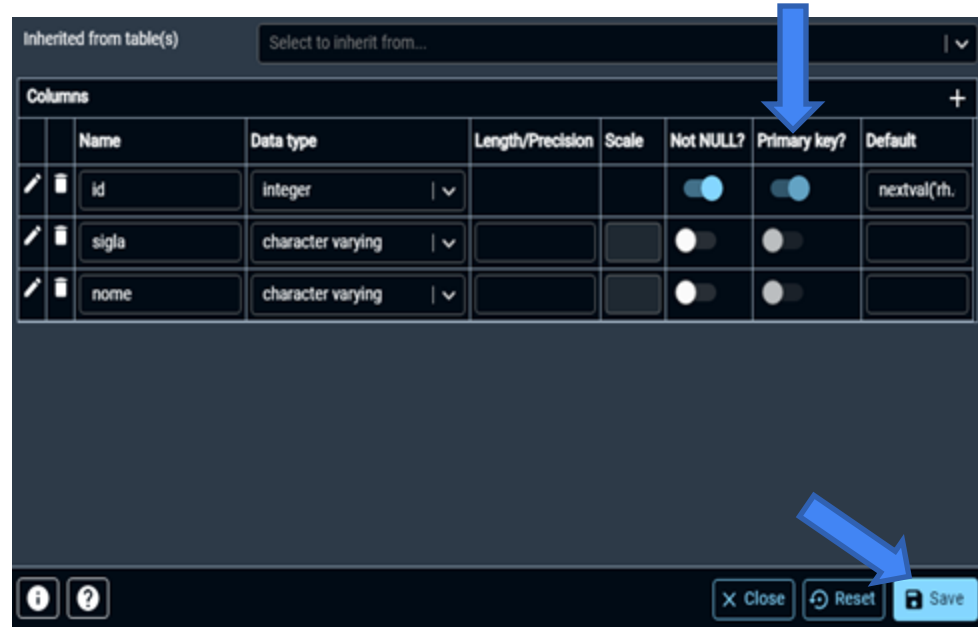
Schema: rh

Tablespace: pg_default

Partitioned table?: ☐

Comment:

Close Reset Save

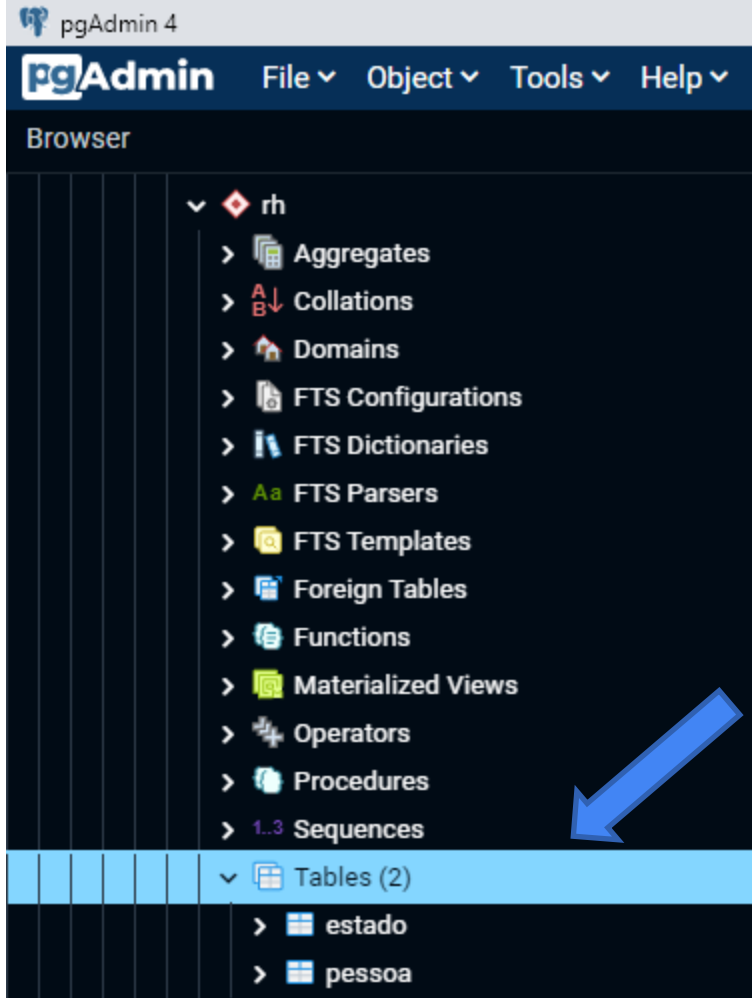


Inherited from table(s): Select to inherit from...

Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
	id	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	nextval('rh.
	sigla	character varying			<input type="checkbox"/>	<input type="checkbox"/>	
	nome	character varying			<input type="checkbox"/>	<input type="checkbox"/>	

Close Reset Save



Exercício – Inserir os 26 Estados e o DF

- Acre (AC)
- Alagoas (AL)
- Amapá (AP)
- Amazonas (AM)
- Bahia (BA)
- Ceará (CE)
- Distrito Federal (DF)
- Espírito Santo (ES)
- Goiás (GO)
- Maranhão (MA)
- Mato Grosso (MT)
- Mato Grosso do Sul (MS)
- Minas Gerais (MG)
- Pará (PA)
- Paraíba (PB)
- Paraná (PR)
- Pernambuco (PE)
- Piauí (PI)
- Rio de Janeiro (RJ)
- Rio Grande do Norte (RN)
- Rio Grande do Sul (RS)
- Rondônia (RO)
- Roraima (RR)
- Santa Catarina (SC)
- São Paulo (SP)
- Sergipe (SE)
- Tocantins (TO)

`INSERT INTO rh.estado (sigla, nome) VALUES ('AC', 'ACRE');`



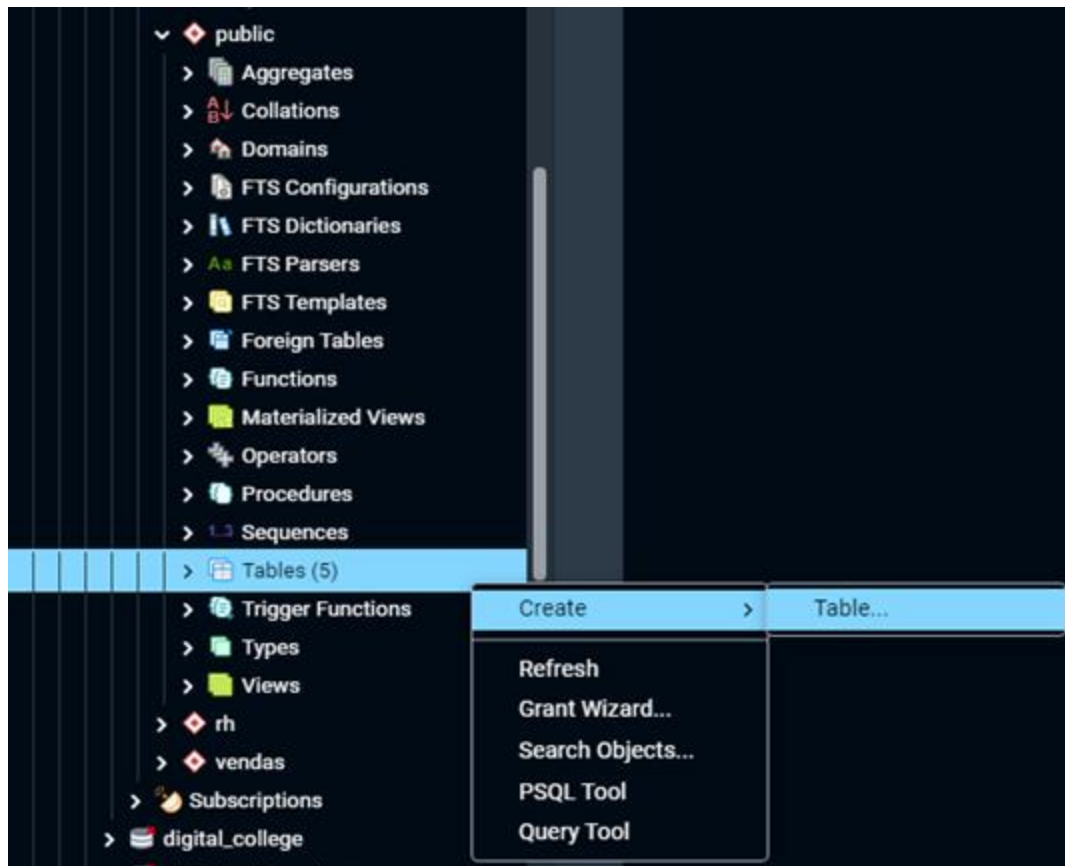
Criando as tabelas através de interface gráfica

TABELA DEPARTAMENTO		
ID	NOME	ID_REGIAO
10	VENDAS	1
20	OPERAÇÕES	1
30	ADMINISTRATIVO	2
40	FINANCEIRO	3

TABELA REGIÃO	
ID	NOME
1	NORTE
2	CENTRO
3	SUL



Criando uma tabela através de interface gráfica



Criando a tabela região

Create - Table

General Columns Advanced Constraints Partitions Parameters Security SQL

Name:

Owner:

Schema:

Tablespace:

Partitioned table? ☐

Comment:

Create - Table

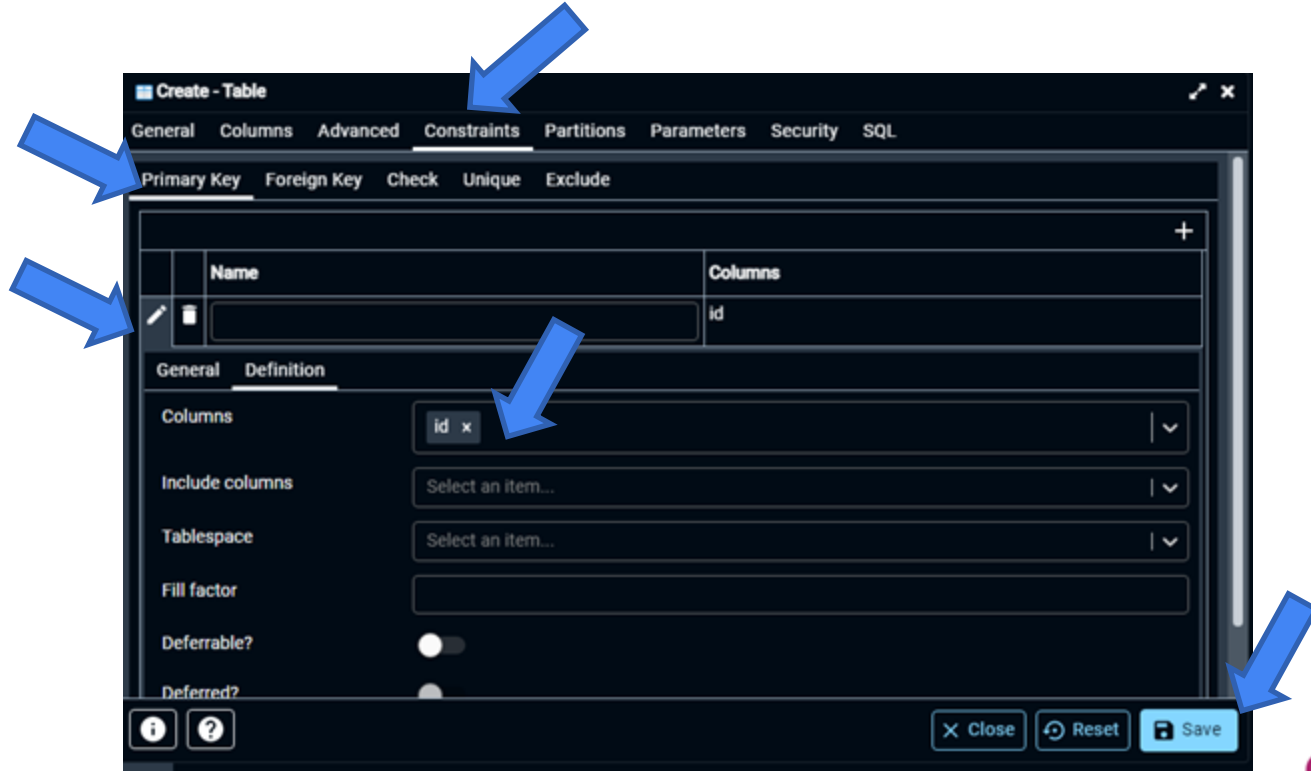
General Columns Advanced Constraints Partitions Parameters Security SQL

Inherited from table(s):

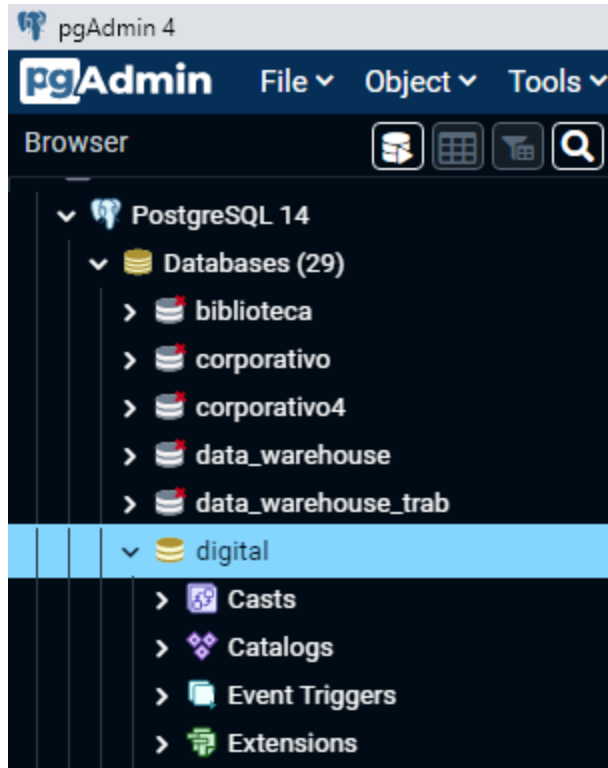
Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
<input type="checkbox"/>	id	serial			<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>
<input type="checkbox"/>	nome	character varying			<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>

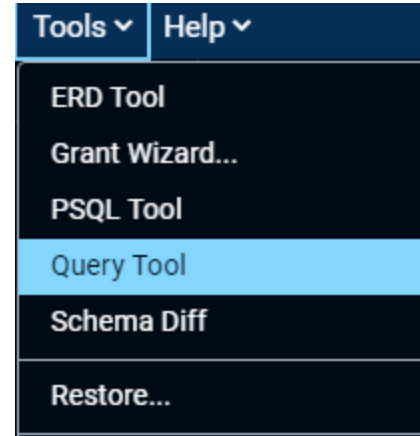
Definindo Primary Key



Criando a tabela região via código



Com o banco de dados selecionado, ir no menu Tools -> Query Tool



Criando a tabela região via código

```
Query  Query History
1
2 CREATE TABLE IF NOT EXISTS public.regiao
3 (
4     id serial NOT NULL,
5     nome character varying,
6     CONSTRAINT regiao_pkey PRIMARY KEY (id)
7 );
```



Criando a tabela departamento

Create - Table

GeneralColumnsAdvancedConstraintsPartitionsParametersSecuritySQL

Name

departamento

Owner

postgres

Schema

public

Tablespace

Select an item...

Partitioned table?

☐

Comment

Close

Reset

Save

Create - Table

GeneralColumnsAdvancedConstraintsPartitionsParametersSecuritySQL

Inherited from table(s)

Select to inherit from...

Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
	id	serial			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	nome	character varying			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	id_regiao	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Close

Reset

Save

Definindo Primary Key e Foreign key

The image displays two side-by-side screenshots of a database management tool's 'Create Table' dialog, illustrating the steps to define a Primary Key and a Foreign Key.

Left Screenshot (Primary Key):

- The **Constraints** tab is selected, with the **Primary Key** sub-tab active.
- A table with one column, **id**, is shown in the main area.
- The **Definition** sub-tab is active, showing the **Columns** list with **id** selected.
- Buttons at the bottom include **Close**, **Reset**, and **Save**.

Right Screenshot (Foreign Key):

- The **Constraints** tab is selected, with the **Foreign Key** sub-tab active.
- The main area shows a table with three columns: **Name**, **Columns**, and **Referenced Table**.
- The **Columns** sub-tab is active, showing the **Columns** list with **id_regiao** selected.
- The **References** list shows **public.regiao** as the referenced table.
- The **Referencing** list shows **id** as the referencing column.
- An **Add** button is visible at the bottom right.

Criando a tabela departamento via código

Query Query History

```
1
2 CREATE TABLE IF NOT EXISTS public.departamento
3 (
4     id serial NOT NULL,
5     nome character varying,
6     id_regiao integer,
7     CONSTRAINT departamento_pkey PRIMARY KEY (id),
8     CONSTRAINT departamento_id_regiao_fkey FOREIGN KEY (id_regiao)
9     REFERENCES public.regiao (id)
10 );
```



Inserindo dados na tabela região

Query Query History

```
1 INSERT INTO public.regiao(id, nome)
2 VALUES (1, 'Norte');
3
4 INSERT INTO public.regiao(id, nome)
5 VALUES (2, 'Centro');
6
7 INSERT INTO public.regiao(id, nome)
8 VALUES (3, 'Sul');
```



Visualizando os dados da tabela região

Query

Query History

1

SELECT * FROM regioao

Data output

Messages

Notifications

	<div><div>id</div><div>[PK] integer</div></div>	<div><div>nome</div><div>character varying</div></div>
1	1	Norte
2	2	Centro
3	3	Sul



Inserindo dados na tabela departamento

Query Query History

```
1 INSERT INTO public.departamento (id, nome, id_regiao)
2 VALUES (10, 'Vendas', 1);
3
4 INSERT INTO public.departamento (id, nome, id_regiao)
5 VALUES (20, 'Operações', 1);
6
7 INSERT INTO public.departamento (id, nome, id_regiao)
8 VALUES (30, 'Administrativo', 2);
9
10 INSERT INTO public.departamento (id, nome, id_regiao)
11 VALUES (40, 'Financeiro', 3);
```



Visualizando os dados da tabela departamento

Query

Query History

1

SELECT * FROM departamento;

Data output

Messages

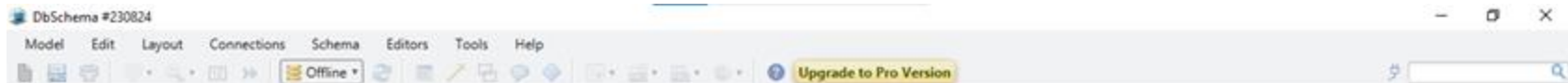
Notifications

	<div><div>Id</div><div>[PK] integer</div></div>	<div><div>nome</div><div>character varying</div></div>	<div><div>Id_regiao</div><div>integer</div></div>
1	10	Vendas	1
2	20	Operações	1
3	30	Administrativo	2
4	40	Financeiro	3





DbSchema



DbSchema

Start New Project

[Connect to Database](#) 1

[Open SQL DDL File](#) 2

By performing a reverse engineering process on the database, you gain the ability to edit the schema directly and execute queries seamlessly.

[Watch Video](#)

Start from Scratch

[Logical Design](#) 1

[Physical Design](#) 2

Create and design the schema, saving the resulting design model to a file (Pro). This schema can be deployed to any database at a later stage.

Reopen Saved Design Model (Pro)

DbSchema Pro features the design model, a copy of the schema, making possible to design the schema in a team, schema versioning and deployment across diverse databases.

[Choose Model File \(Pro\)](#)

[Sample Models](#)

Tutorials

[Video Tutorials](#)

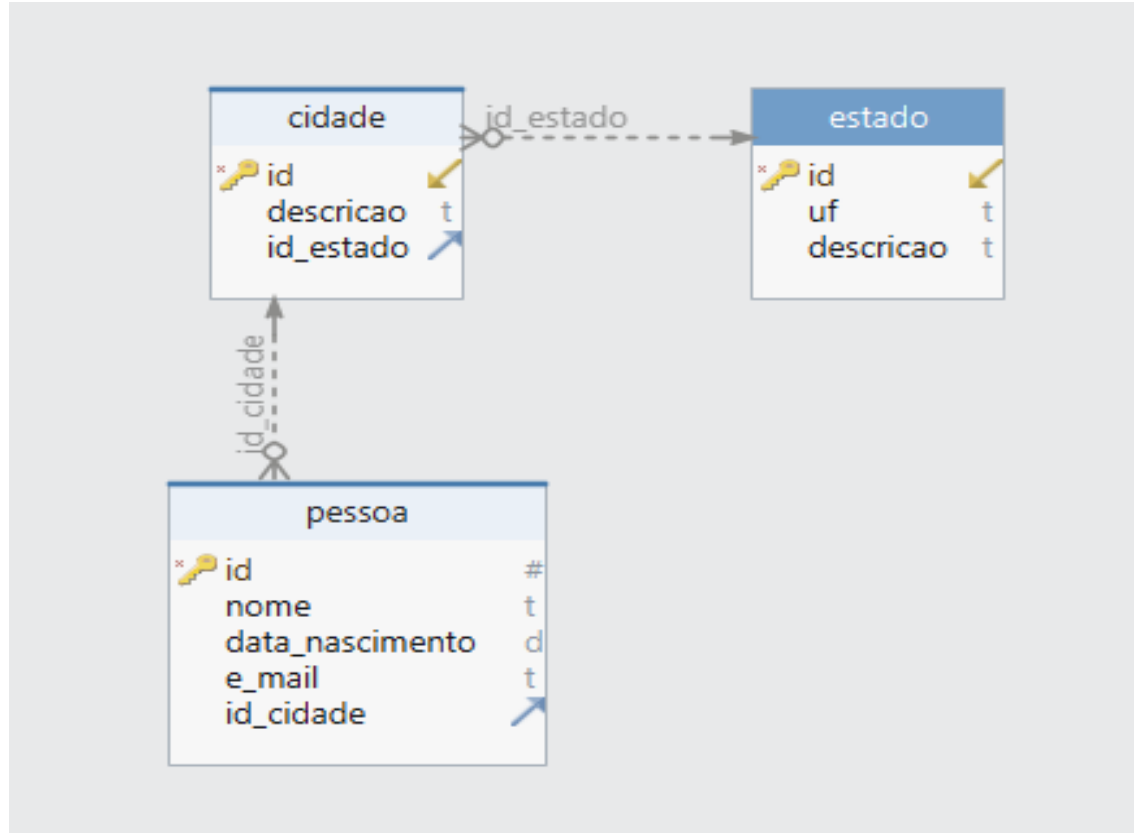
[Quick Tour](#)

[Documentation](#)

[Scaling & Dark Theme](#)

Design - Document - Generate Scripts - Deploy on Multiple Databases - Explore and Query. Details in the Quick Tour

Criar modelo lógico usando DBSchema



FIM

