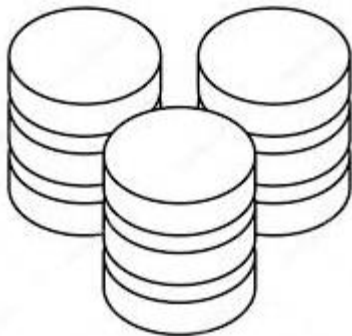


## Banco de Dados II

### O SGBD PostgreSQL



Profa. Damires Souza  
damires@ifpb.edu.br



# PostgreSQL

---

- Implementa o **Modelo Relacional**

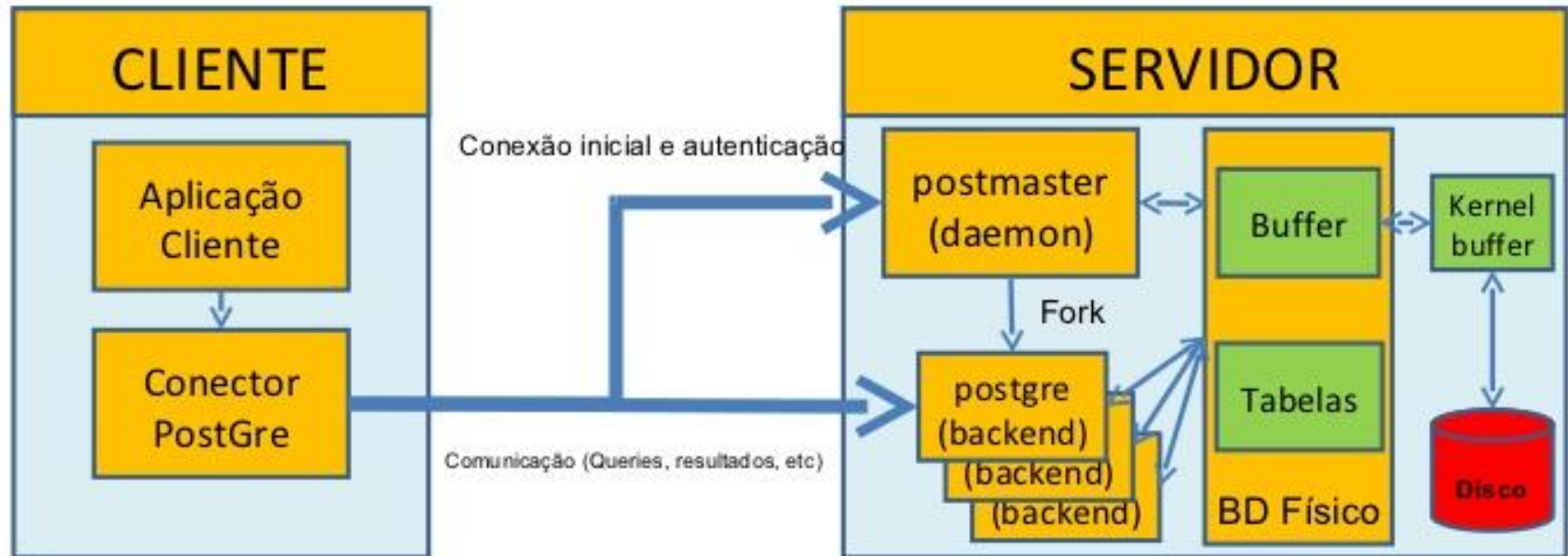
- Mas possui extensões Objeto-relacional, NoSQL...



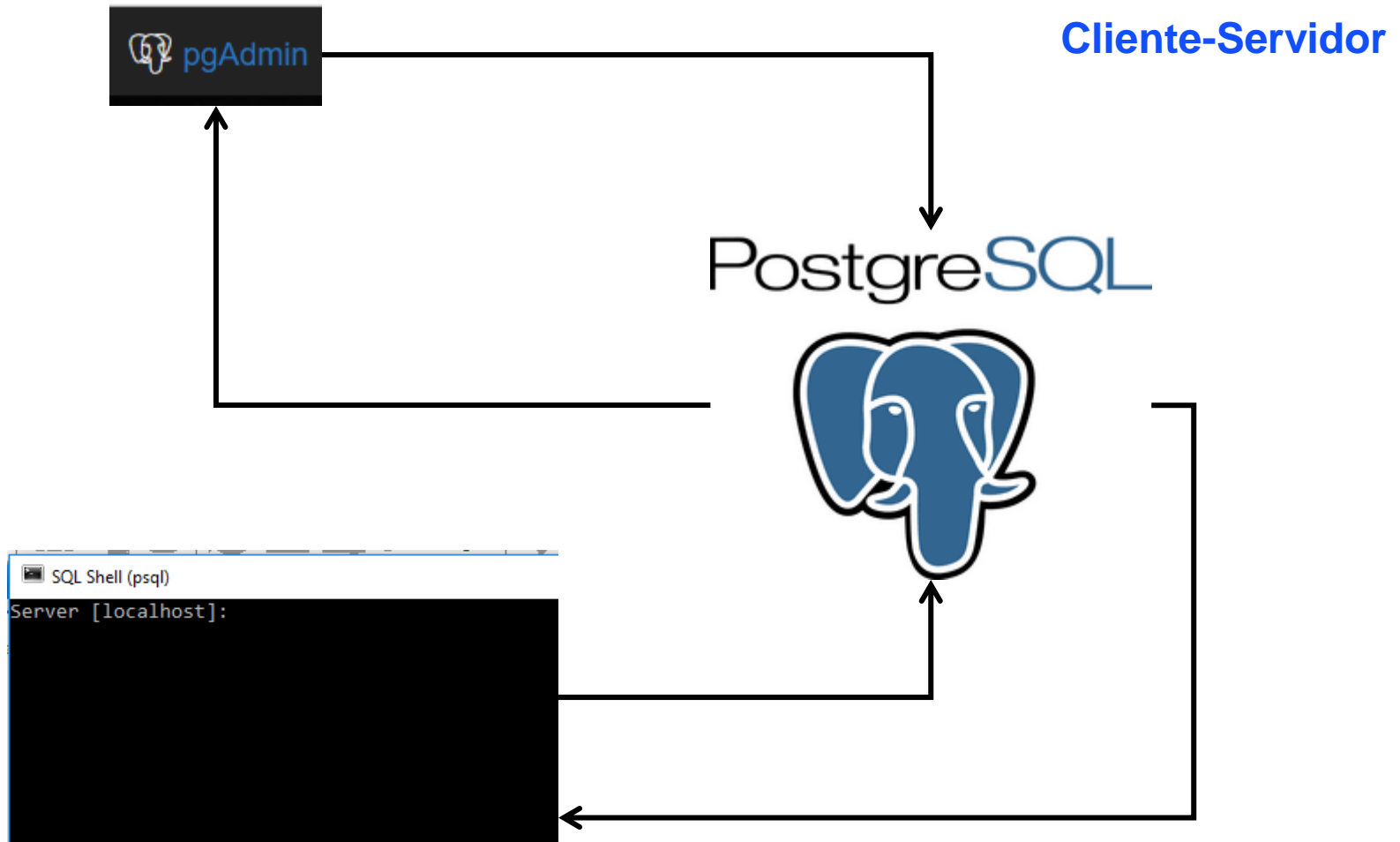
<https://www.postgresql.org/docs/14/>

- Open-source
- Arquitetura cliente-servidor, distribuída
- Iniciado pelo prof. Stonebraker
- SGBD “maduro” e muito usado

# Arquitetura Cliente-Servidor

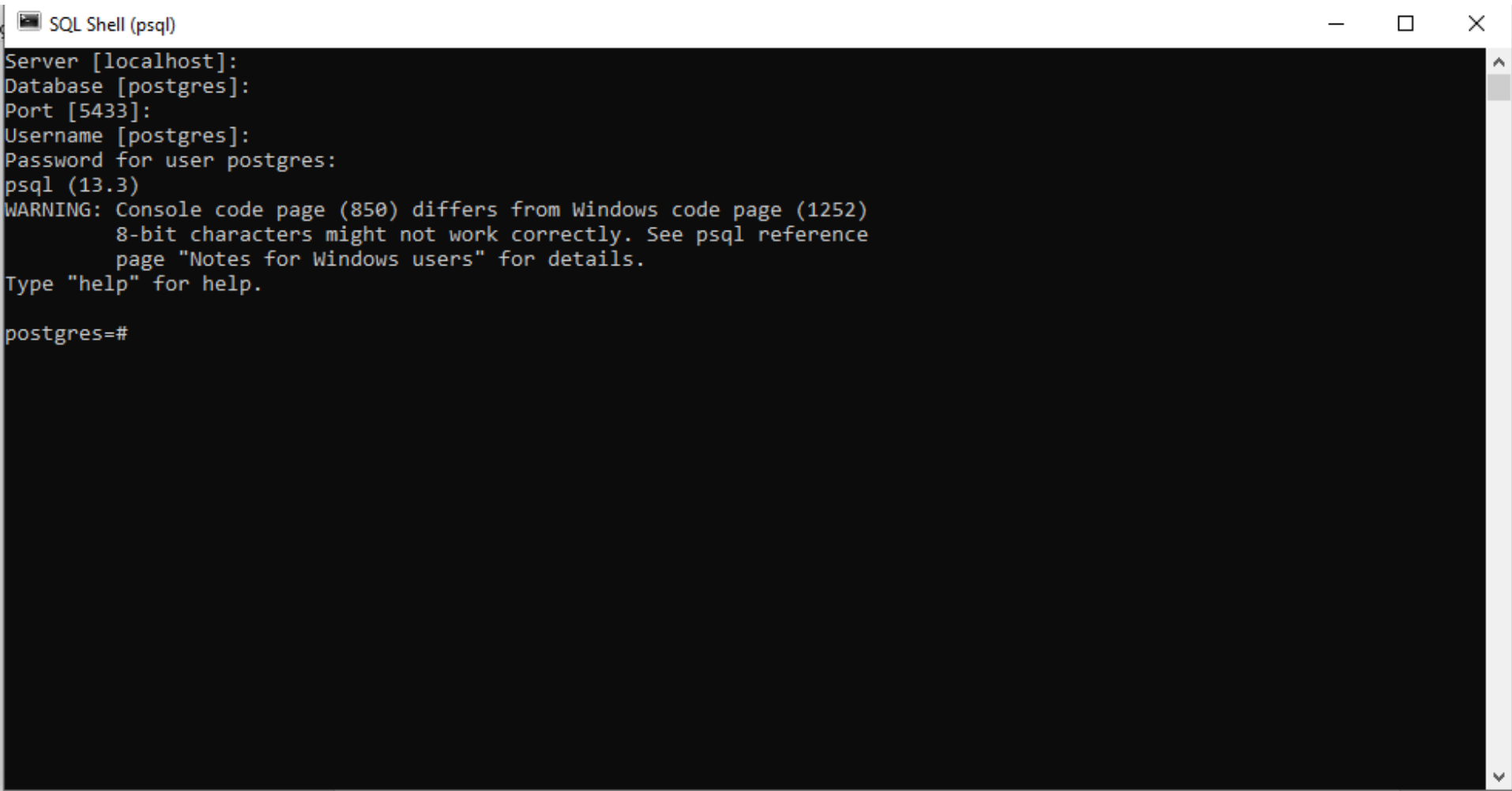


# Interfaces clientes e Postgres



# Conexão ao Servidor Postgres

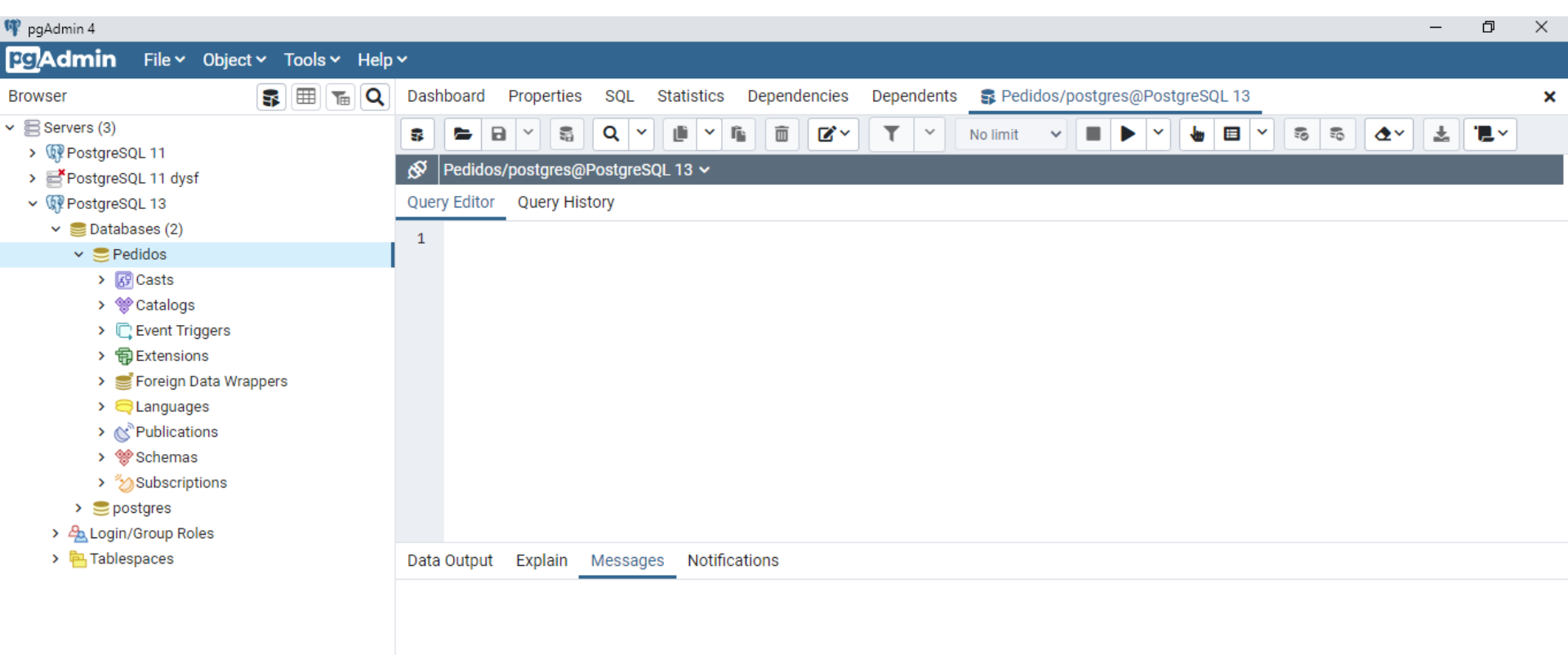
---



```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5433]:
Username [postgres]:
Password for user postgres:
psql (13.3)
WARNING: Console code page (850) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

postgres=#
```

# Conexão ao Servidor Postgres



# PostgreSQL e bancos

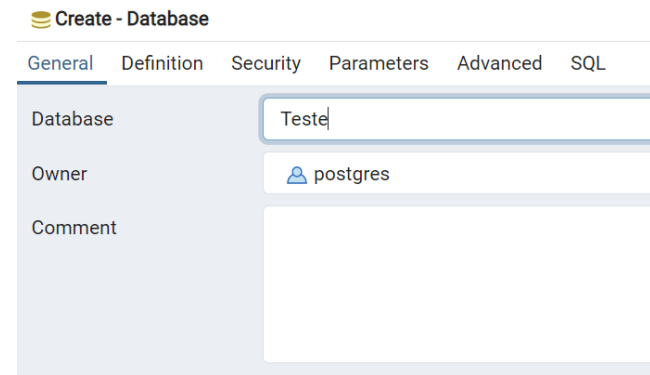
- O SGBD provê meios para o gerenciamento de diferentes **bancos de dados**
- Cada **BD** está associado a um **projeto** (escopo) ou **aplicação**

## ➤ Create database

*(New Database)*

### ➤ **Teste**

- Create database Teste;  
\*\* Super usuário ou usuário  
que tenha privilégio createdb



The screenshot shows the 'Create - Database' dialog box in PostgreSQL. It has tabs for 'General', 'Definition', 'Security', 'Parameters', 'Advanced', and 'SQL'. The 'General' tab is selected. It contains fields for 'Database' (with the value 'Teste'), 'Owner' (with the value 'postgres' and a user icon), and 'Comment' (which is empty).

# Teste simples

---

Create table **teste1**(cod integer, valor char(2));

Insert into teste1 values(1,'xx');

Insert into teste1 values(1,'xx');

Insert into teste1 values(1,'xx');

Select \* from teste1;

--

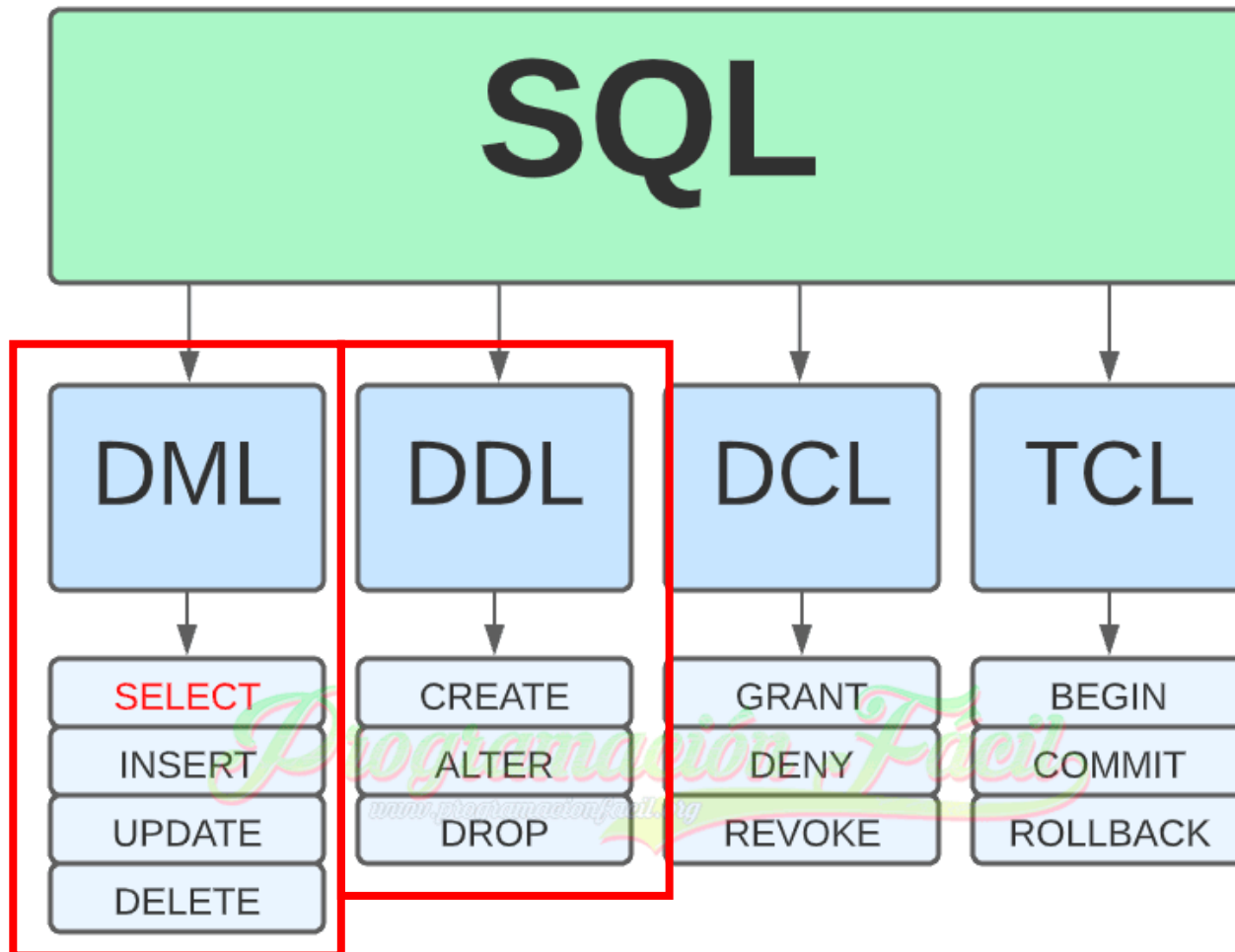
Truncate table teste1;

Select \* from teste1;

Drop table teste1;



# Linguagem SQL



# Linguagem de Manipulação de Dados

---

```
INSERT INTO produto VALUES (1, 'Queijo', 9.99);
```

```
INSERT INTO produto (num, nome, preço)  
VALUES (1, 'Queijo', 9.99);
```

```
INSERT INTO produto (num, nome, preço)  
VALUES (1, 'Queijo', DEFAULT);
```

```
INSERT INTO produto (num, nome, preço) VALUES  
    (1, 'Queijo', 9.99),  
    (2, 'Pão', 1.99),  
    (3, 'Leite', 2.99);
```

# Linguagem de Manipulação de Dados

---

```
UPDATE produto  
SET preço = 10  
WHERE preço = 5;
```

```
UPDATE produto  
SET preço = preço * 1.10;
```

```
UPDATE teste  
SET a = 5, b = 3, c = 1  
WHERE a > 0;
```

# Linguagem de Manipulação de Dados

---

```
DELETE FROM produto  
WHERE preço = 10;
```

```
DELETE FROM  
produto;
```

**Data Manipulation Language (DML)**

# Tipo de dados Serial


---

```
CREATE TABLE tablename (  
    colname SERIAL  
);
```

is equivalent to specifying:

```
CREATE SEQUENCE tablename_colname_seq;  
CREATE TABLE tablename (  
    colname integer NOT NULL DEFAULT nextval('tablename_colname_seq')  
);  
ALTER SEQUENCE tablename_colname_seq OWNED BY tablename.colname;
```

# Exemplo

>  teste

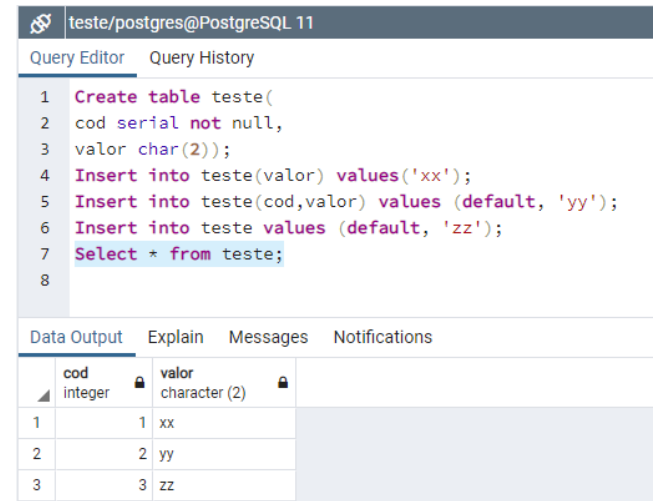
Create table **teste2**(  
    cod **serial** not null,  
    valor char(2));

Insert into teste2(valor) values('xx');

Insert into teste2(cod,valor) values (default, 'yy');

Insert into teste2 values (default, 'zz');

Select \* from teste2;



The screenshot shows a PostgreSQL query editor interface. The top bar indicates the user is 'teste/postgres@PostgreSQL 11'. The 'Query Editor' tab is active, displaying a SQL script with line numbers 1 through 8. The script creates a table 'teste2' with columns 'cod' (serial, not null) and 'valor' (char(2)). It then inserts three rows: one with 'xx', one with 'yy' (using default for cod), and one with 'zz' (using default for cod). The final line is 'Select \* from teste2;'. Below the editor, the 'Data Output' tab is active, showing a table with 3 rows and 2 columns: 'cod' (integer) and 'valor' (character (2)). The rows contain (1, 'xx'), (2, 'yy'), and (3, 'zz').

```
1 Create table teste2(  
2   cod serial not null,  
3   valor char(2));  
4 Insert into teste2(valor) values('xx');  
5 Insert into teste2(cod,valor) values (default, 'yy');  
6 Insert into teste2 values (default, 'zz');  
7 Select * from teste2;  
8
```

	cod integer	valor character (2)
1	1	xx
2	2	yy
3	3	zz

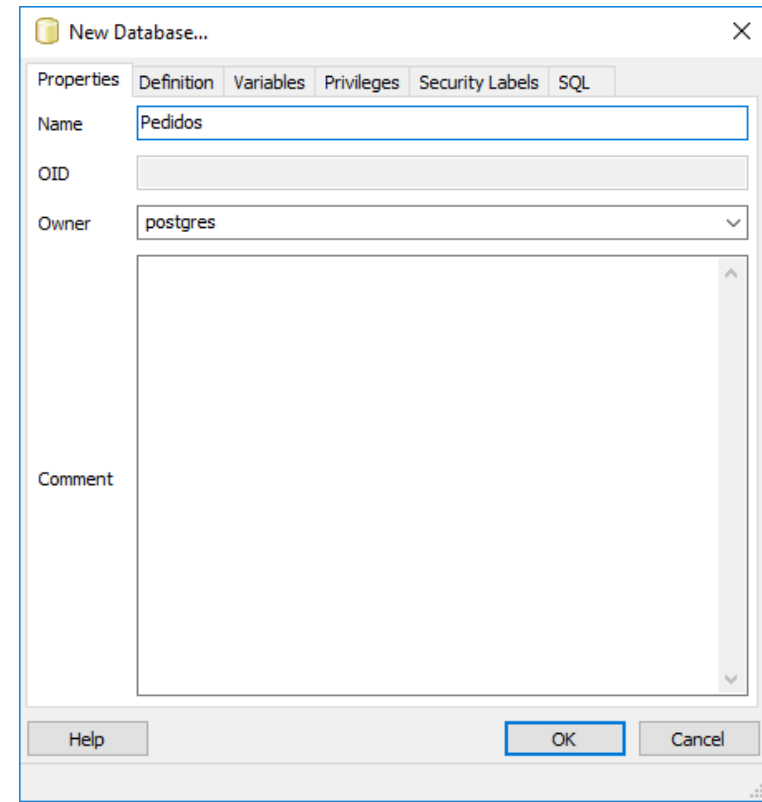
# Outro Banco de Dados

## ➤ Create database

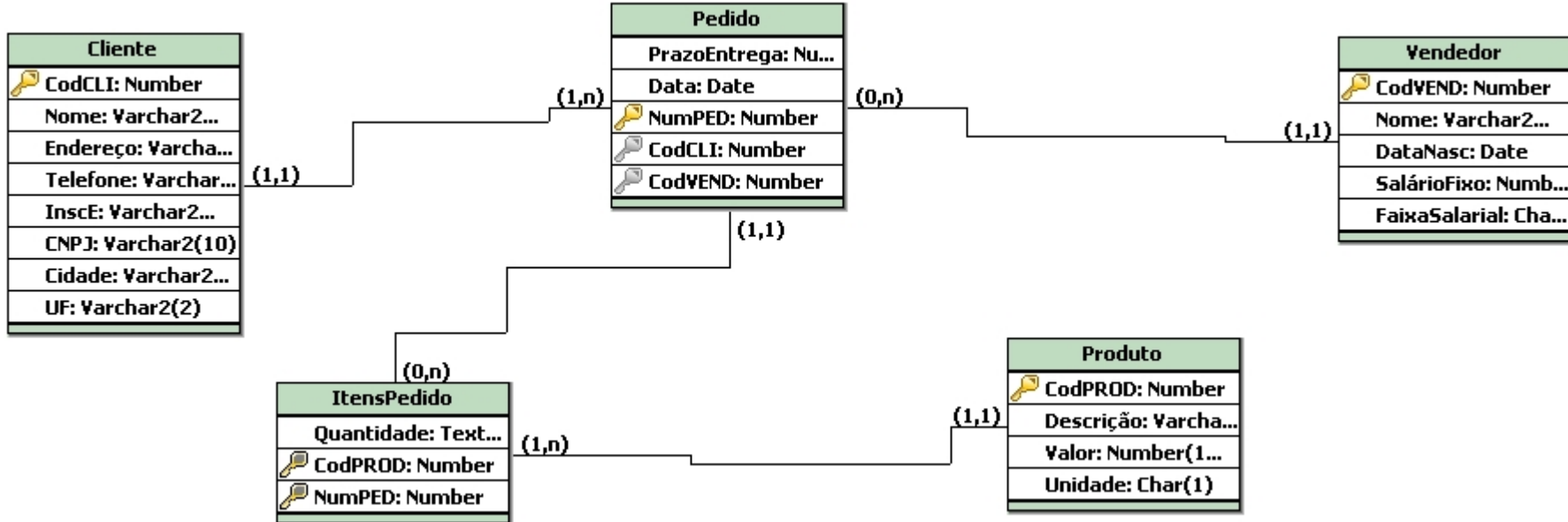
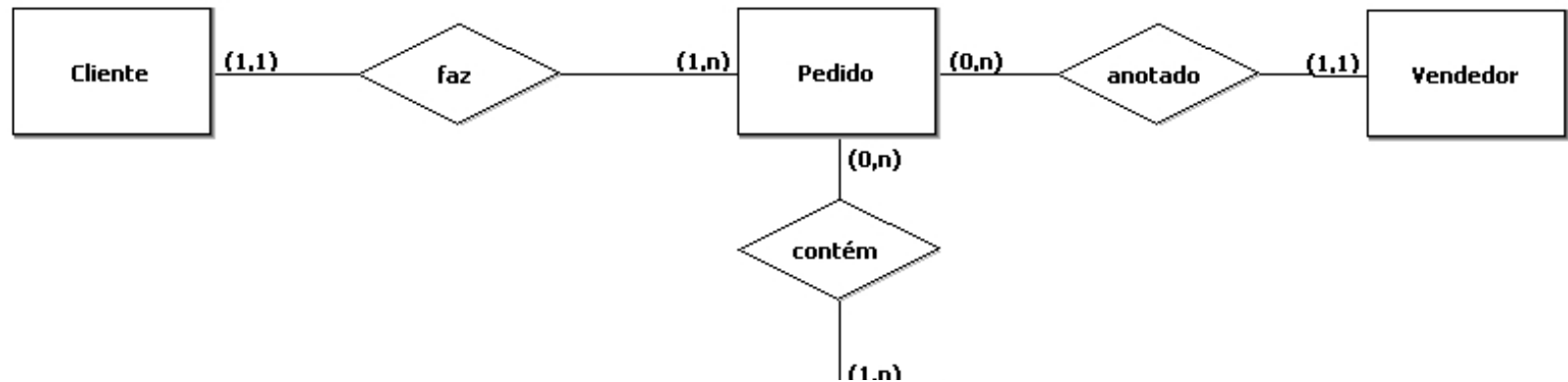
*(New Database)*

## ➤ Pedidos

- Create database Pedidos;  
\*\* Super usuário ou usuário  
que tenha privilégio createdb



# BD Pedidos





# Create database

---

```
CREATE DATABASE "Pedidos"  
WITH OWNER = postgres  
ENCODING = 'UTF8'  
TABLESPACE = pg_default  
LC_COLLATE = 'Portuguese_Brazil.1252'  
LC_CTYPE = 'Portuguese_Brazil.1252'  
CONNECTION LIMIT = -1;
```



Sem  
limite

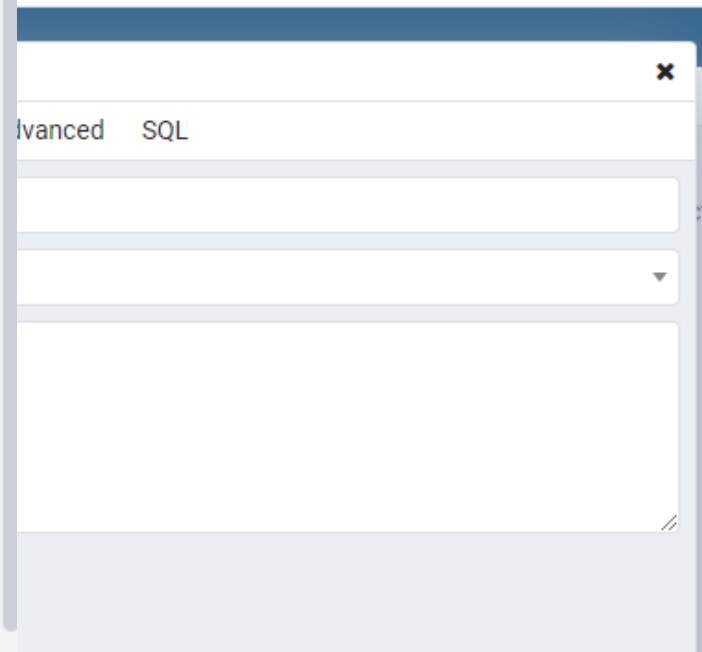
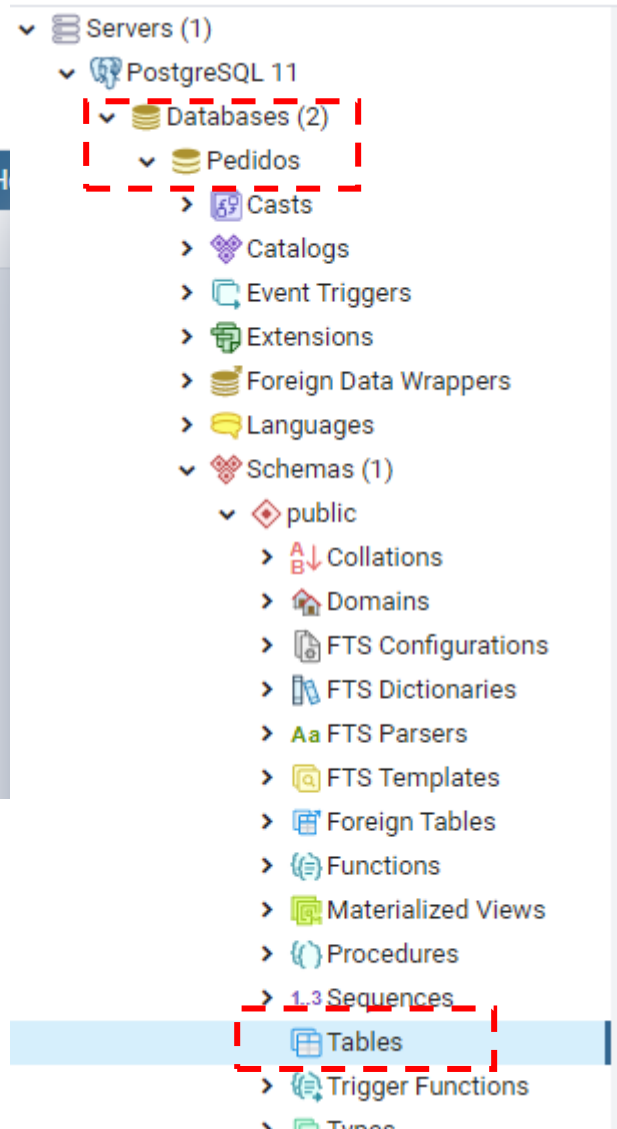
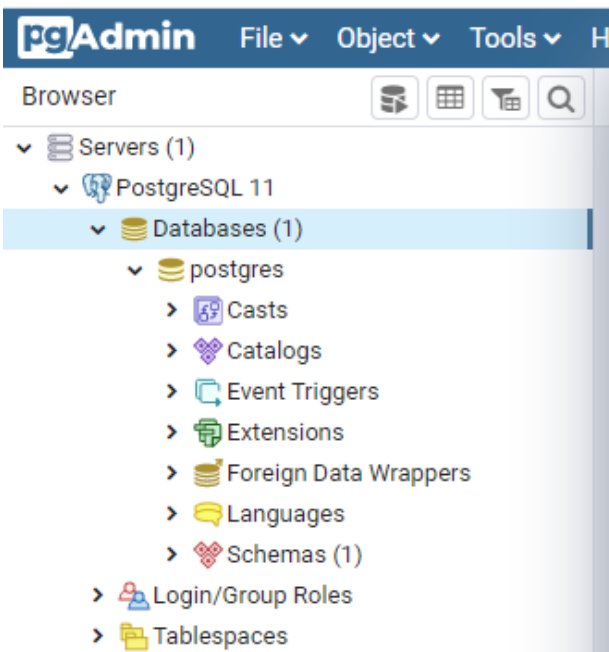
Ou createdb Pedidos;

Para remover: **Drop Database Pedidos;**

Para alterar: Alter Database...

# PostgreSQL e SQL

## ➤ BD Pedidos



# Scripts SQL

---

- Um **script SQL** é um conjunto de comandos SQL salvos como um arquivo SQL
- Pode-se usar scripts SQL para criar, editar, visualizar, executar e excluir registros

**\*\* Vamos criar um banco para uma aplicação**

## **exemplo: Pedidos**

- Requisitos funcionais:
  - Cadastrar Pedido
  - Cadastrar Cliente
  - Cadastrar Vendedor
  - Cadastrar Produto

# PostgreSQL e scripts SQL

➤ BD Pedidos

Script: criarPedidos.sql

pgAdmin 4

File Object Tools Help

Browser

- Servers (3)
  - PostgreSQL 11
  - PostgreSQL 11 dysf
  - PostgreSQL 13
    - Databases (2)
      - Pedidos
        - Casts
        - Catalogs
        - Event Triggers
        - Extensions
        - Foreign Data Wrappers
        - Languages
        - Publications
        - Schemas
        - Subscriptions
      - postgres
      - Login/Group Roles
      - Tablespaces

Dashboard Properties SQL Statistics Dependencies Dependents Pedidos/postgres@PostgreSQL 13 \*

Pedidos/postgres@PostgreSQL 13

Query Editor Query History

```
1 CREATE TABLE Cliente (  
2   CodCLI Serial PRIMARY KEY,  
3   Nome Varchar(30),  
4   Endereco Varchar(30),  
5   Telefone Varchar(12),  
6   InscE Varchar(10),  
7   CNPJ Varchar(10),  
8   Cidade Varchar(15),  
9   UF Varchar(2)  
10 );  
11  
12 CREATE TABLE Produto  
13   CodPROD Serial not null,  
14   Descricao Varchar(30)
```

Select file

C:\Users\Damir\Documents\Damires\IFPB\Sistemas pa...

Name	Size	Modified
criarPedidos.sql	1.0 kB	Tue Mar 13 08:35:33 2018
inscliente.sql	874.0 B	Mon Feb 1 10:51:55 2021
insitenspedido.sql	304.0 B	Mon Feb 1 11:01:59 2021
inspedido.sql	354.0 B	Mon Feb 1 11:00:16 2021
insproduto.sql	363.0 B	Mon Feb 1 10:57:31 2021
insvendedor.sql	407.0 B	Mon Feb 1 10:56:20 2021
QPedidos1.sql	1.0 kB	Thu Feb 11 10:22:03 2021

# Query tool: BD **Pedidos** - Scripts de Criação e População

---

1. CriarPedidos.sql
2. Inscliente.sql
3. Insvendedor.sql
4. Insproduto.sql
5. Inspedido.sql
6. Insitenspedido.sql



**Fique atento(a) às *constraints***



# Inscliente.sql

The image shows a PostgreSQL query editor interface. The top toolbar contains various icons for file operations, search, and execution. The main window is titled "Pedidos/postgres@PostgreSQL 11" and has tabs for "Query Editor" and "Query History".

The "Query Editor" tab displays the following SQL script:

```
1 insert into cliente values(default,'Claudia Dias',null,null, '564325','786534','Recife','PE');
2 insert into cliente values(default,'Joaquim Moraes','Epitacio Pessoa, 123','32425643', '500925','789004','Joao Pessoa',
3 insert into cliente values(default,'Janaina Rodrigues','Rui Carneiro, 342',null, '764325','386534','Joao Pessoa','PB');
4 insert into cliente values(default,'Maria Portela','Boa Viagem, 345','76435678', null,null,'Recife','PE');
5 insert into cliente values(default,'Ana Moura','Nego, 321','32465432', '87325','780978','Joao Pessoa','PB');
6 insert into cliente values(default,'Cassandra Doura',null,null, '786525','79876','Recife','PE');
7 insert into cliente values(default,'Cicero Novaes',null,null, '123525','432534','Natal','RN');
8 insert into cliente values(de
9
```

The "Query History" tab shows the executed query:

```
1 select * from cliente
```

The "Data Output" tab displays the results of the query in a table format:

	codcli [PK] integer	nome character varying (30)	endereco character varying (30)	telefone character varying (12)	insce character varying (10)	cnpj character varying (10)	cidade character
1	1	Claudia Dias	[null]	[null]	564325	786534	Recife
2	2	Joaquim Moraes	Epitacio Pessoa, 123	32425643	500925	789004	Joao I
3	3	Janaina Rodrigues	Rui Carneiro, 342	[null]	764325	386534	Joao I
4	4	Maria Portela	Boa Viagem, 345	76435678	[null]	[null]	Recife
5	5	Ana Moura	Nego, 321	32465432	87325	780978	Joao I
6	6	Cassandra Doura	[null]	[null]	786525	79876	Recife
7	7	Cicero Novaes	[null]	[null]	123525	432534	Natal

# Verifique as tabelas criadas e populadas e faça:

---

1. `select * from cliente;`
2. `select * from vendedor;`
3. `select * from produto;`
4. `select * from pedido;`
5. `select * from itenspedido;`
6. Insira mais três registros em cada uma das cinco tabelas

# Principais Tipos de Dados no PostgreSQL

Tipo	Descrição	Alcance
smallint	[numérico] número inteiro pequeno alcance	-32.768 a 32.767
integer	[numérico] escolha típica para inteiro	-2147483648 a 2147483647
bigint	[numérico] número inteiro grande gama-	-9223372036854775808 para 9223372036854775807
numeric	[numérico] precisão especificada pelo usuário, exato	até 131072 dígitos antes do ponto decimal; até 16383 dígitos depois do ponto decimal
real	[numérico] precisão variável, inexata	precisão 6 dígitos decimais
serial	[numérico] integer autoincrementável	1-2147483647



# Principais Tipos de Dados no PostgreSQL

---

Tipo	Descrição
Varchar(n)	[caracter] comprimento variável com limite
Char(n)	[caracter] comprimento fixo
text	[caracter] comprimento ilimitado variável
date	[data] somente data
timestamp [(p) ] [sem fuso horário]	[data] tanto a data e hora
timestamp [(p) ] com fuso horário	[data] tanto a data e a hora, com fuso horário
booleano	estado de verdadeiro ou falso
point	[geométrico] (x, y)

# Queries: verifique

---

Select numped, codprod, quantidade  
From itenspedido  
Where quantidade > 20;

Select descricao  
From produto  
Where unidade = 'KG' and valor > 4.0

Select codprod, descricao  
From produto  
Where valor between 5.00 and 15.50;

# Verifique

---

Select nome

From vendedor

Where faixacomissao IN ('A','B');

Select nome

From cliente

Where InscE is null;

Select nome, (salariofixo \*1.15) + 120 as "Novo  
Salário"

From vendedor

Where faixacomissao = 'C'

Order by nome;

# Verifique

---

Select \*  
From Cliente  
Where nome like '\_\_\_\_\_A';

X

Select \*  
From Cliente  
Where nome like '\_\_\_\_\_a'