

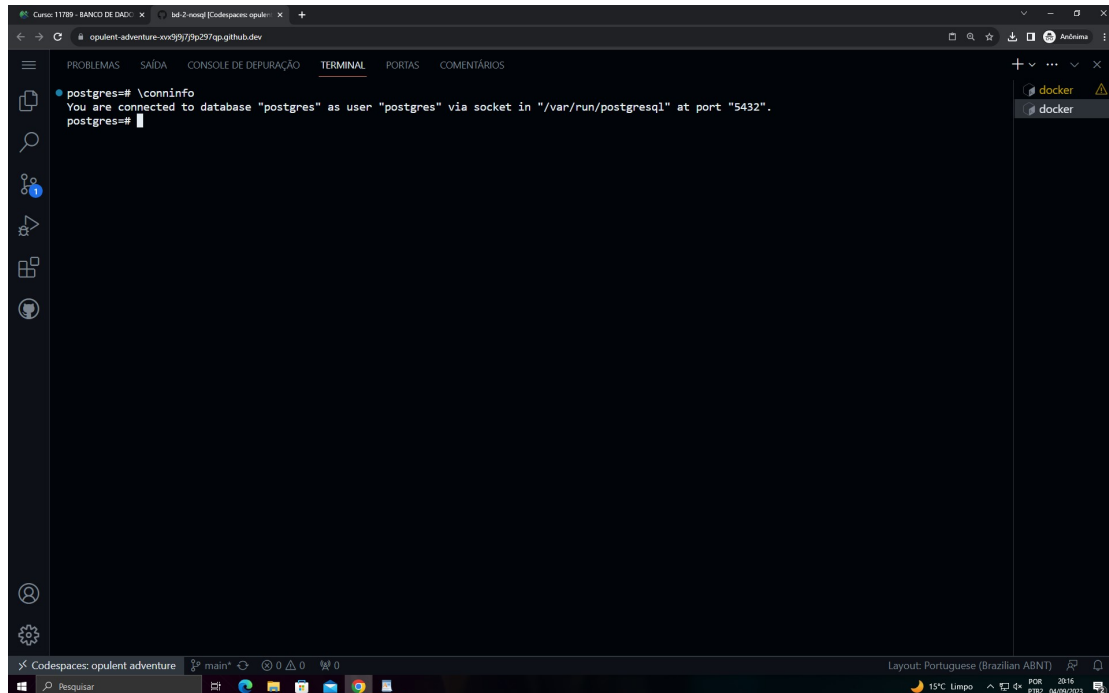
# PostgreSQL

ACADÊMICO: JOÃO MARCELO ZENARO

- **Exercícios com a ferramenta psql**

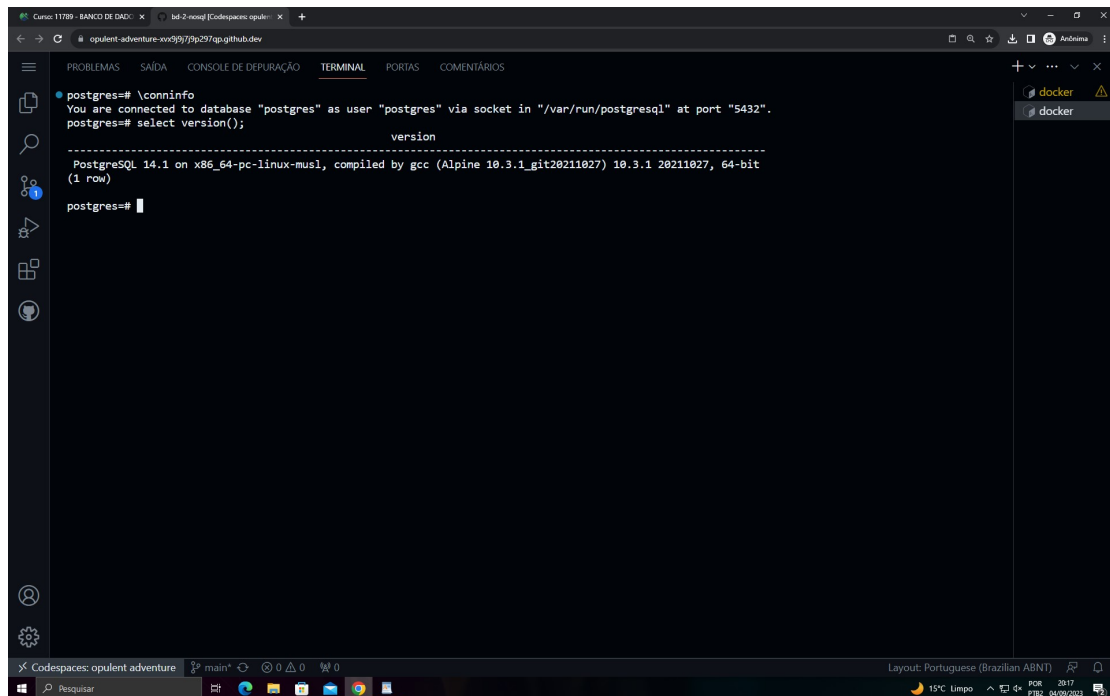
Abra a ferramenta psql e execute os seguintes exercícios:

- Execute o comando `\conninfo`, que mostra informações sobre a conexão atual e cole o resultado abaixo.



```
postgres=# \conninfo
You are connected to database "postgres" as user "postgres" via socket in "/var/run/postgresql" at port "5432".
postgres=#
```

- Execute o comando `SELECT version()` e responda qual a versão do PostgreSQL instalada e qual a arquitetura.



- O comando abaixo mostra o formato atual de data utilizado pelo PostgreSQL:

- `SHOW DATESTYLE;`

Execute o comando e responda abaixo qual é o formato de datas que está sendo utilizado.

```

```
postgres=# show datestyle;
```

```
   DateStyle
```

```
-----
```

```
ISO, MDY
```

```
(1 row)
```

```

"ISO": ISO 8601, formato "AAAA-MM-DD" (Ano-Mês-Dia).

"MDY": Isso indica a ordem de exibição dos campos da data. "MDY" significa Mês-Dia-Ano.

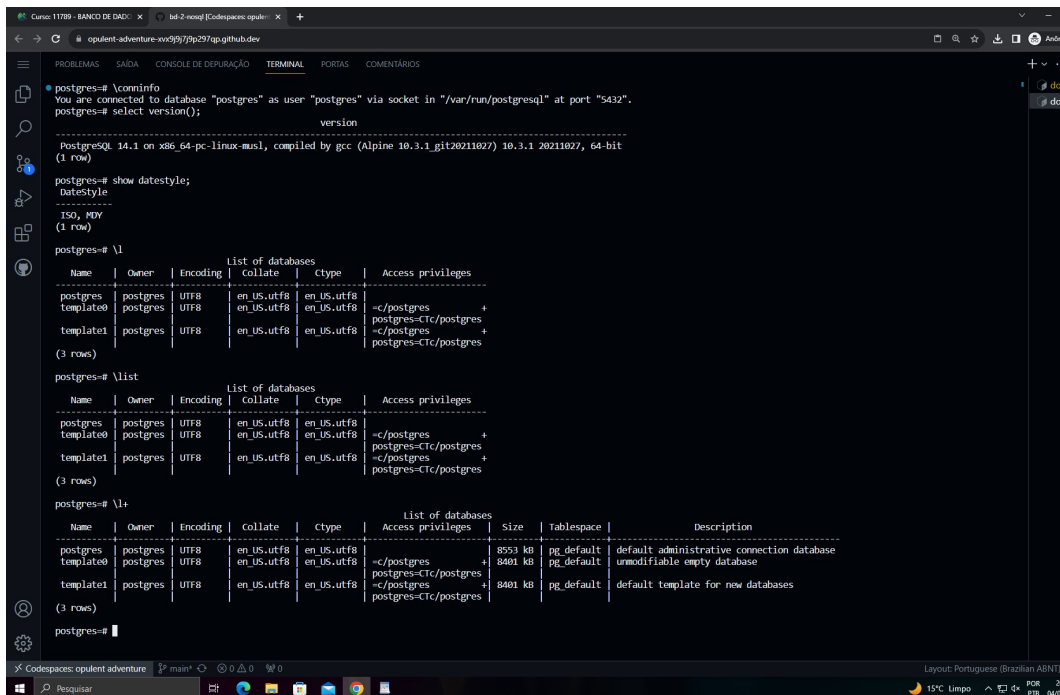
- Para listar os bancos de dados existentes em um determinado servidor, podem ser usados os comandos:

- `\l`
- `\list`
- `\l+`

Execute e responda qual é a diferença da saída dos 2 primeiros comandos em relação ao último?

O ultimo comando mostra informações adicionais. O tamanho que o banco ocupa,

## Tablespace e descrição.



```
postgres=# \conninfo
You are connected to database "postgres" as user "postgres" via socket in "/var/run/postgresql/" at port "5432".
postgres=# select version();
               version
-----
PostgreSQL 14.1 on x86_64-pc-linux-musl, compiled by gcc (Alpine 10.3.1_git20211027) 10.3.1 20211027, 64-bit
(1 row)

postgres=# show datestyle;
datestyle
-----
ISO, MDY
(1 row)

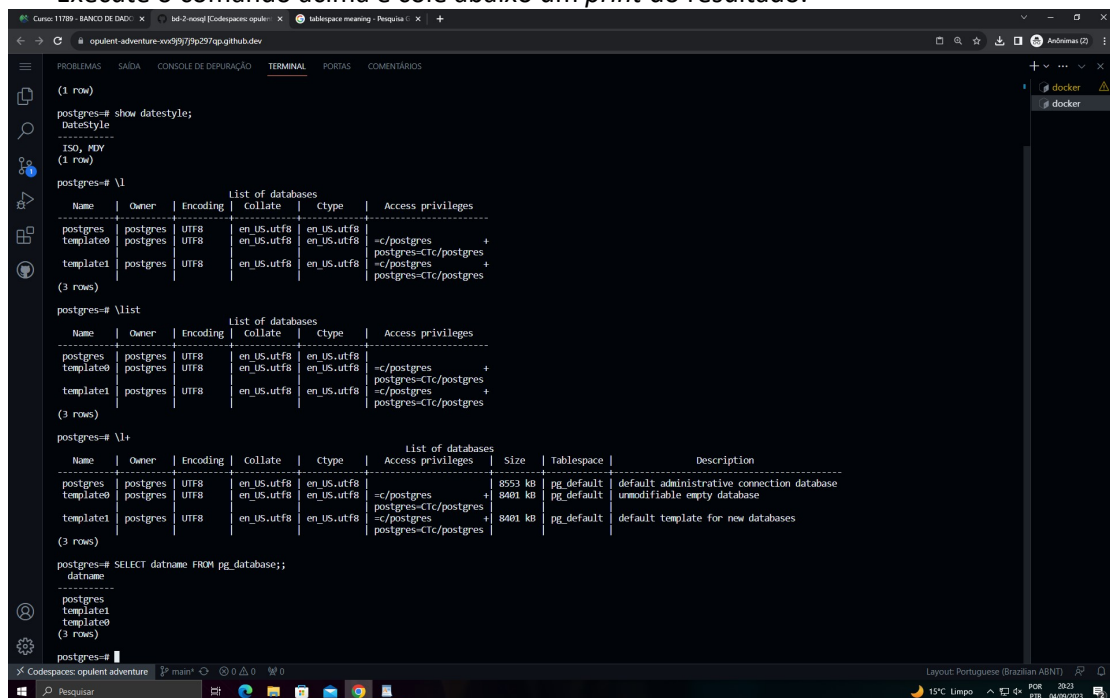
postgres=# \l
               List of databases
  Name | Owner | Encoding | Collate | Ctype | Access privileges
-----+-----+-----+-----+-----+-----
postgres | postgres | UTF8 | en_US.utf8 | en_US.utf8 | =c/postgres
template0 | postgres | UTF8 | en_US.utf8 | en_US.utf8 | postgres=CTc/postgres
template1 | postgres | UTF8 | en_US.utf8 | en_US.utf8 | postgres=CTc/postgres
(3 rows)

postgres=# \list
               List of databases
  Name | Owner | Encoding | Collate | Ctype | Access privileges
-----+-----+-----+-----+-----+-----
postgres | postgres | UTF8 | en_US.utf8 | en_US.utf8 | =c/postgres
template0 | postgres | UTF8 | en_US.utf8 | en_US.utf8 | postgres=CTc/postgres
template1 | postgres | UTF8 | en_US.utf8 | en_US.utf8 | postgres=CTc/postgres
(3 rows)

postgres=# \l+
               List of databases
  Name | Owner | Encoding | Collate | Ctype | Access privileges | Size | Tablespace | Description
-----+-----+-----+-----+-----+-----+-----+-----+-----
postgres | postgres | UTF8 | en_US.utf8 | en_US.utf8 | =c/postgres | 8553 kB | pg_default | default administrative connection database
template0 | postgres | UTF8 | en_US.utf8 | en_US.utf8 | postgres=CTc/postgres | 8401 kB | pg_default | unmodifiable empty database
template1 | postgres | UTF8 | en_US.utf8 | en_US.utf8 | postgres=CTc/postgres | 8401 kB | pg_default | default template for new databases
(3 rows)
```

- Outra forma de listar os nomes dos bancos de dados é utilizando o dicionário (ou catálogo) do sistema:
- **SELECT** datname **FROM** pg\_database;

Execute o comando acima e cole abaixo um *print* do resultado.



```
postgres=# show datestyle;
datestyle
-----
ISO, MDY
(1 row)

postgres=# \l
               List of databases
  Name | Owner | Encoding | Collate | Ctype | Access privileges
-----+-----+-----+-----+-----+-----
postgres | postgres | UTF8 | en_US.utf8 | en_US.utf8 | =c/postgres
template0 | postgres | UTF8 | en_US.utf8 | en_US.utf8 | postgres=CTc/postgres
template1 | postgres | UTF8 | en_US.utf8 | en_US.utf8 | postgres=CTc/postgres
(3 rows)

postgres=# \list
               List of databases
  Name | Owner | Encoding | Collate | Ctype | Access privileges
-----+-----+-----+-----+-----+-----
postgres | postgres | UTF8 | en_US.utf8 | en_US.utf8 | =c/postgres
template0 | postgres | UTF8 | en_US.utf8 | en_US.utf8 | postgres=CTc/postgres
template1 | postgres | UTF8 | en_US.utf8 | en_US.utf8 | postgres=CTc/postgres
(3 rows)

postgres=# \l+
               List of databases
  Name | Owner | Encoding | Collate | Ctype | Access privileges | Size | Tablespace | Description
-----+-----+-----+-----+-----+-----+-----+-----+-----
postgres | postgres | UTF8 | en_US.utf8 | en_US.utf8 | =c/postgres | 8553 kB | pg_default | default administrative connection database
template0 | postgres | UTF8 | en_US.utf8 | en_US.utf8 | postgres=CTc/postgres | 8401 kB | pg_default | unmodifiable empty database
template1 | postgres | UTF8 | en_US.utf8 | en_US.utf8 | postgres=CTc/postgres | 8401 kB | pg_default | default template for new databases
(3 rows)

postgres=# SELECT datname FROM pg_database;
 datname
-----
postgres
template1
template0
(3 rows)
```

- No MySQL um esquema (*schema*) é um sinônimo de banco de dados. No PostgreSQL, no entanto, um esquema é um espaço de nome (*namespace*) criado dentro de um banco de dados com a finalidade de organizar logicamente as tabelas. Isso permite isolar e segmentar o acesso a tabelas a usuários ou grupos particulares dando a eles permissões específicas.

Para listar os esquemas de um determinado banco de dados, 3 comandos podem ser utilizados:

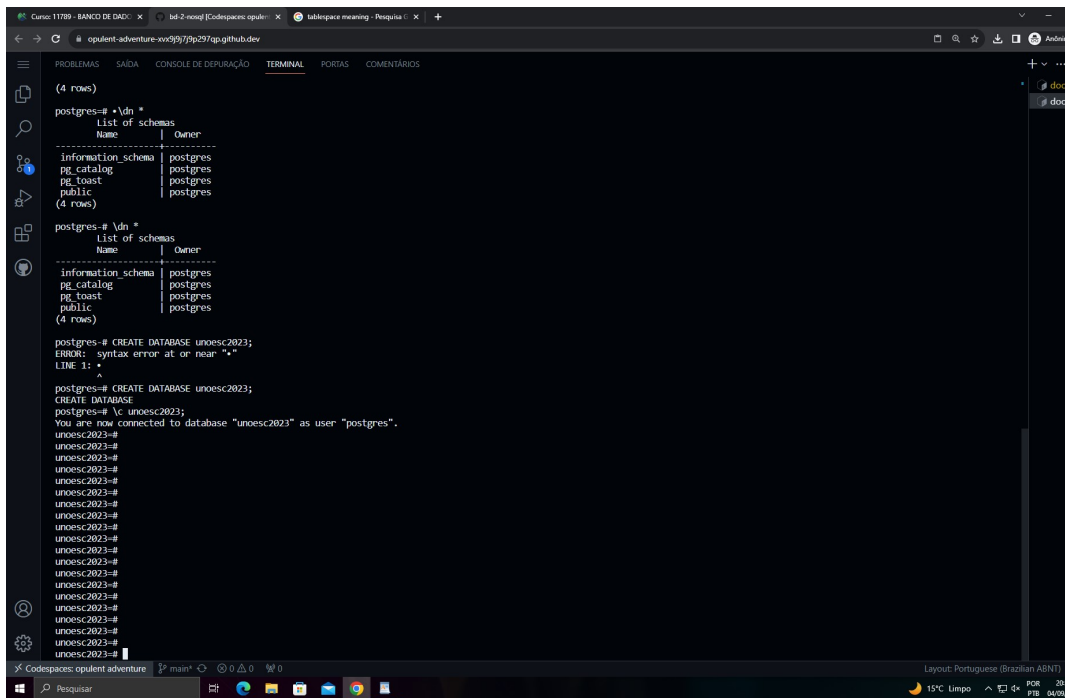
- `SELECT schema_name FROM information_schema.schemata;`
- `SELECT nspname FROM pg_catalog.pg_namespace;`
- `\dn *`

Execute e responda qual é a diferença da saída dos 2 primeiros comandos em relação ao último?

**O ultimo comando também mostra o dono (Owner) da base de dados.**

- Execute os comandos abaixo para remover o banco de dados unoesc2023 (caso já existe) e criá-lo:
  - `DROP DATABASE IF EXISTS unoesc2023;`
  - `CREATE DATABASE unoesc2023;`
- Para conectar com um banco específico usa-se o comando `\c`. Conecte ao banco recém criado.
  - `\c unoesc2023;`

Cole abaixo o resultado, se assegurando que você está agora conectado ao banco unoesc2023.



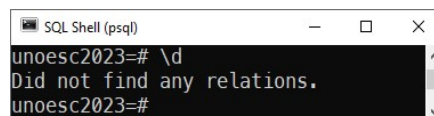
```
postgres=# \d
List of schemas
Name | Owner
-----|-----
information_schema | postgres
pg_catalog | postgres
pg_toast | postgres
public | postgres
(4 rows)

postgres=# \dn
List of schemas
Name | Owner
-----|-----
information_schema | postgres
pg_catalog | postgres
pg_toast | postgres
public | postgres
(4 rows)

postgres=# CREATE DATABASE unesc2023;
ERROR: syntax error at or near "\"
LINE 1: "
        ^

postgres=# CREATE DATABASE unesc2023;
CREATE DATABASE
postgres=# \c unesc2023;
You are now connected to database "unesc2023" as user "postgres".
unesc2023=#
unesc2023=#
unesc2023=#
unesc2023=#
unesc2023=#
unesc2023=#
unesc2023=#
unesc2023=#
unesc2023=#
unesc2023=#
unesc2023=#
unesc2023=#
unesc2023=#
unesc2023=#
unesc2023=#
```

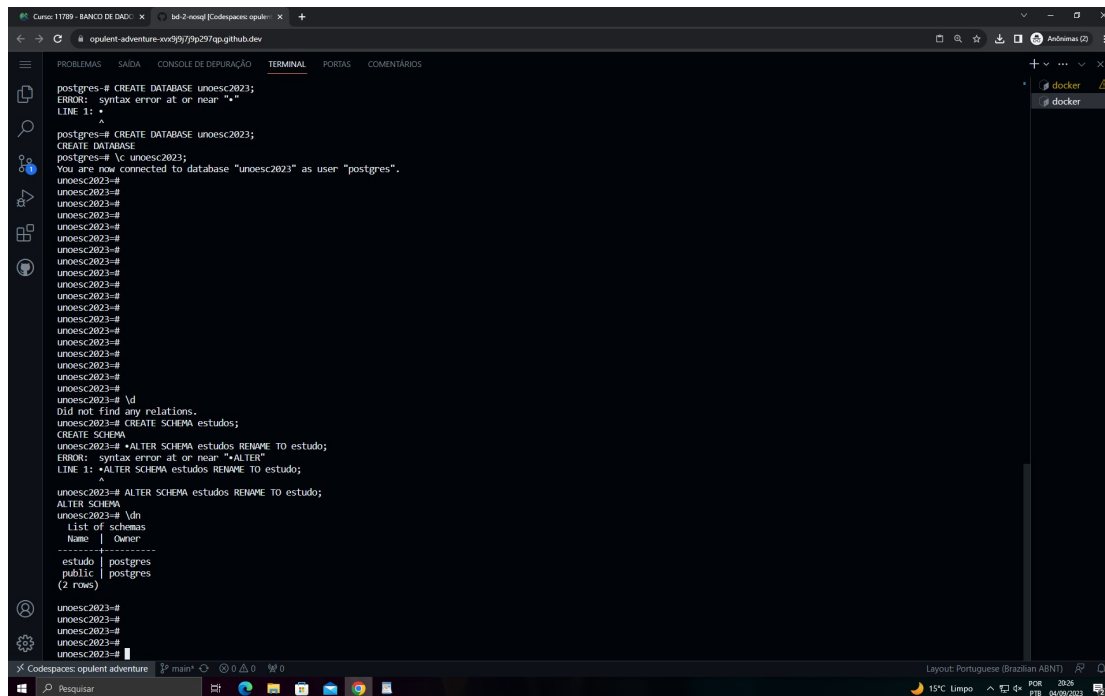
- O comando `\d` é utilizado para listar as relações de um banco de dados. No PostgreSQL as relações podem ser tabelas, *views* (visões) e sequências. Como ainda nenhuma relação foi criada, ao executar este comando, a saída abaixo irá aparecer.



```
SQL Shell (psql)
unesc2023=# \d
Did not find any relations.
unesc2023=#
```

- Execute os comandos abaixo para criar um *schema* e alterar seu nome:
  - **CREATE SCHEMA** estudos;
  - **ALTER SCHEMA** estudos **RENAME TO** estudo;

Utilize o comando `\dn` para listar os esquemas e cole abaixo o resultado.



```
postgres=# CREATE DATABASE unesc2023;
ERROR: syntax error at or near "*"
LINE 1: *
        ^

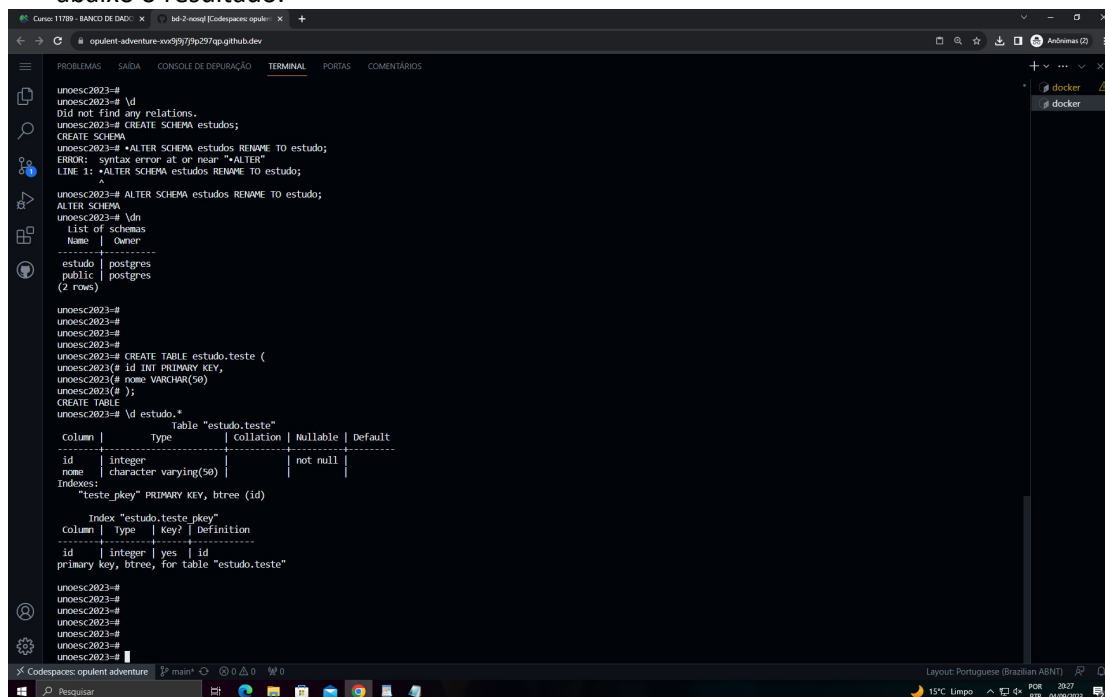
postgres=# CREATE DATABASE unesc2023;
CREATE DATABASE
postgres=# \c unesc2023;
You are now connected to database "unesc2023" as user "postgres".
unesc2023=#
unesc2023=#
unesc2023=#
unesc2023=#
unesc2023=#
unesc2023=#
unesc2023=#
unesc2023=#
unesc2023=#
unesc2023=#
unesc2023=#
unesc2023=#
unesc2023=#
unesc2023=#
unesc2023=# \d
Did not find any relations.
unesc2023=# CREATE SCHEMA estudos;
CREATE SCHEMA
unesc2023=# *ALTER SCHEMA estudos RENAME TO estudo;
ERROR: syntax error at or near "*"
LINE 1: *ALTER SCHEMA estudos RENAME TO estudo;
        ^
unesc2023=# ALTER SCHEMA estudos RENAME TO estudo;
ALTER SCHEMA
unesc2023=# \dn
List of schemas
Name | Owner
-----|-----
estudo | postgres
public | postgres
(2 rows)

unesc2023=#
unesc2023=#
unesc2023=#
```

- Execute o comando abaixo para criar uma tabela no *schema* estudo:

```
CREATE TABLE estudo.teste (
    id INT PRIMARY KEY,
    nome VARCHAR(50)
);
```

Utilize o comando `\d estudo.*` para listar todas as relações do *schema* estudo e cole abaixo o resultado.



```
unesc2023=# \d
Did not find any relations.
unesc2023=# CREATE SCHEMA estudos;
CREATE SCHEMA
unesc2023=# *ALTER SCHEMA estudos RENAME TO estudo;
ERROR: syntax error at or near "*"
LINE 1: *ALTER SCHEMA estudos RENAME TO estudo;
        ^
unesc2023=# ALTER SCHEMA estudos RENAME TO estudo;
ALTER SCHEMA
unesc2023=# \dn
List of schemas
Name | Owner
-----|-----
estudo | postgres
public | postgres
(2 rows)

unesc2023=#
unesc2023=#
unesc2023=#
unesc2023=# CREATE TABLE estudo.teste (
unesc2023=#   id INT PRIMARY KEY,
unesc2023=#   nome VARCHAR(50)
unesc2023=# );
CREATE TABLE
unesc2023=# \d estudo,*
Table "estudo.teste"
Column | Type | Collation | Nullable | Default
-----|-----|-----|-----|-----
id      | integer |          | not null |
nome    | character varying(50) |          |          |
Indexes:
    "teste_pkey" PRIMARY KEY, btree (id)

Index "estudo.teste_pkey"
Column | Type | Key | Definition
-----|-----|-----|-----
id     | integer | yes | id
primary key, btree, for table "estudo.teste"

unesc2023=#
unesc2023=#
unesc2023=#
unesc2023=#
```

- O comando `\dt` lista somente as relações do tipo tabela. O comando `\dt+` mostra informações mais detalhadas sobre uma tabela. O comando `SELECT` abaixo usa o dicionário de dados (catálogo do sistema) para mostrar as mesmas informações. Execute os comandos a seguir e cole o resultado.
- `\dt estudo.*`
- `\dt+ estudo.teste`
- `SELECT * FROM information_schema.tables WHERE table_catalog='unoes2023' AND table_schema='estudo';`

```
unoes2023=# \dt estudo.*
unoes2023=#
unoes2023=#
unoes2023=#
unoes2023=# \dt estudo.*
List of relations
Schema | Name | Type | Owner
-----+-----+-----+-----
estudo | teste | table | postgres
(1 row)

unoes2023=# \dt+ estudo.teste
List of relations
Schema | Name | Type | Owner | Persistence | Access method | Size | Description
-----+-----+-----+-----+-----+-----+-----+-----
estudo | teste | table | postgres | permanent | heap | 0 bytes |
(1 row)

unoes2023=# *SELECT * FROM information_schema.tables WHERE table_catalog='unoes2023' AND table_schema='estudo';
ERROR: syntax error at or near "*"SELECT"
LINE 1: *SELECT * FROM information_schema.tables WHERE table_catalog...
      ^

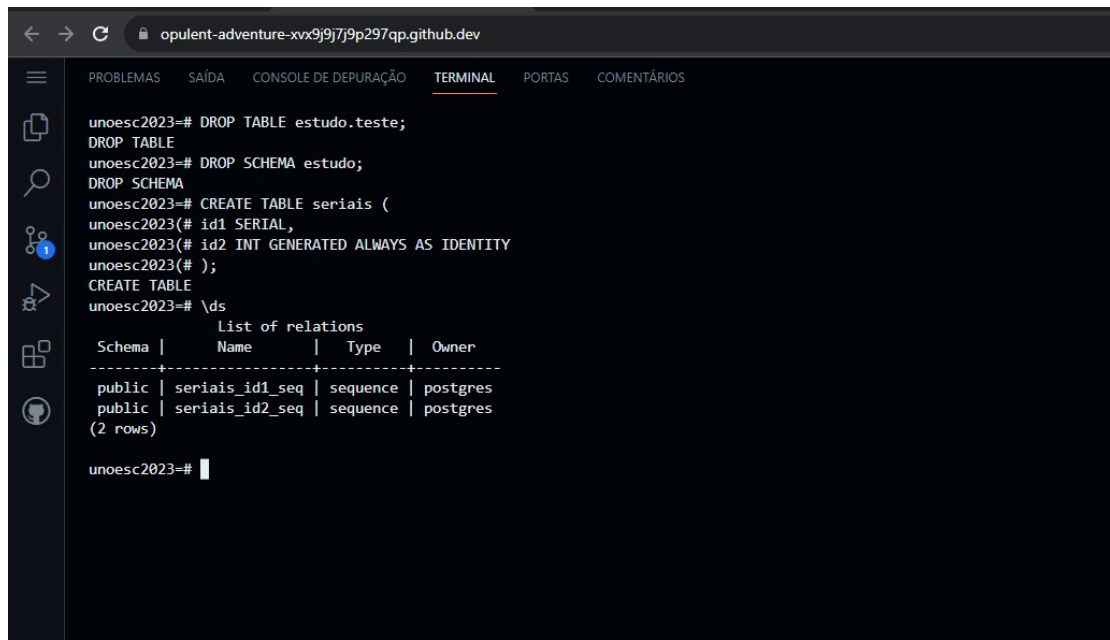
unoes2023=# SELECT * FROM information_schema.tables WHERE table_catalog='unoes2023' AND table_schema='estudo';
 table_catalog | table_schema | table_name | table_type | self_referencing_colan_name | reference_generation | user_defined_type_catalog | user_defined_type_schema | user_defined_type_name | is_insertable_into | is_typed | commit_action
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
unoes2023 | estudo | teste | BASE TABLE | | | | | | | YES | NO |
(1 row)
```

- O comando `DROP SCHEMA` pode ser usado para excluir um *schema*. Um *schema* não poderá ser excluído se existirem objetos que dependam dele. Execute os comandos abaixo para remover a tabela e o esquema e então cole o resultado.
- `DROP TABLE estudo.teste;`
- `DROP SCHEMA estudo;`
- O PostgreSQL disponibiliza duas formas de se criar um campo do tipo autoincremento:
  - Pseudotipo [SERIAL](#)
  - Restrição [GENERATED \[BY DEFAULT | ALWAYS\] AS IDENTITY](#)

Execute o comando para criar uma tabela contendo dois campos de autoincremento:

```
CREATE TABLE seriais (
    id1 SERIAL,
    id2 INT GENERATED ALWAYS AS IDENTITY
);
```

Execute o comando `\ds` para listar as relações de sequência do esquema atual e cole o resultado abaixo:



The screenshot shows a web browser window with the address bar displaying `opulent-adventure-xvx9j7j9p297qp.github.dev`. The browser interface includes tabs for 'PROBLEMAS', 'SAÍDA', 'CONSOLE DE DEPURACÃO', 'TERMINAL' (which is active), 'PORTAS', and 'COMENTÁRIOS'. On the left side of the terminal, there is a vertical toolbar with icons for file operations, search, and other utilities. The terminal window contains the following text:

```
unoesc2023=# DROP TABLE estudo.teste;
DROP TABLE
unoesc2023=# DROP SCHEMA estudo;
DROP SCHEMA
unoesc2023=# CREATE TABLE serials (
unoesc2023(# id1 SERIAL,
unoesc2023(# id2 INT GENERATED ALWAYS AS IDENTITY
unoesc2023(# );
CREATE TABLE
unoesc2023=# \ds
List of relations
Schema |      Name      | Type   | Owner
-----+-----+-----+-----
public | serials_id1_seq | sequence | postgres
public | serials_id2_seq | sequence | postgres
(2 rows)

unoesc2023=#
```

• Ainda com os de autoincremento, experimente executar os seguintes comandos:

- `INSERT INTO serials VALUES (1, 1);`
- `INSERT INTO serials (id1) VALUES (1);`
- `INSERT INTO serials (id2) VALUES (2);`
- `INSERT INTO serials (id1, id2) VALUES (2, null);`

Somente o segundo deve ter sido executado com sucesso. Os outros não foram executados pois campos de identidade criados com a opção `ALWAYS` não permitem que se tente atribuir um valor a eles.



```

Curso: 11789 - BANCO DE DADOS x docker-compose.yml - bd-2-nos x +
opulent-adventure-vx9j7j9p297qp.github.dev

PROBLEMAS SAÍDA CONSOLE DE DEPURACÃO TERMINAL PORTAS COMENTÁRIOS

List of relations
Schema | Name | Type | Owner
-----+-----+-----+-----
public | serials_id1_seq | sequence | postgres
public | serials_id2_seq | sequence | postgres
(2 rows)

unoesc2023=# INSERT INTO serials VALUES (1, 1);
ERROR: cannot insert a non-DEFAULT value into column "id2"
DETAIL: Column "id2" is an identity column defined as GENERATED ALWAYS.
HINT: Use OVERRIDING SYSTEM VALUE to override.
unoesc2023=# INSERT INTO serials (id1) VALUES (1);
ALUES (2);
INSERT INTO serials (id1, id2) VALUES (2, null);
INSERT 0 1
unoesc2023=# INSERT INTO serials (id2) VALUES (2);
ERROR: cannot insert a non-DEFAULT value into column "id2"
DETAIL: Column "id2" is an identity column defined as GENERATED ALWAYS.
HINT: Use OVERRIDING SYSTEM VALUE to override.
unoesc2023=# INSERT INTO serials (id1, id2) VALUES (2, null);
ERROR: cannot insert a non-DEFAULT value into column "id2"
DETAIL: Column "id2" is an identity column defined as GENERATED ALWAYS.
HINT: Use OVERRIDING SYSTEM VALUE to override.
unoesc2023=#
unoesc2023=#
unoesc2023=#
unoesc2023=#

```

- É possível inserir registros usando a palavra-chave `DEFAULT`. A palavra chave `RETURNING` retorna os valores inseridos. Execute os comandos abaixo e cole o `print` abaixo:
- `INSERT INTO serials VALUES(DEFAULT) RETURNING id1;`
- `INSERT INTO serials VALUES(DEFAULT, DEFAULT) RETURNING *;`

```

unoesc2023=#
unoesc2023=#
unoesc2023=#
unoesc2023=#
unoesc2023=# INSERT INTO serials VALUES(DEFAULT) RETURNING id1;
RETURNING id1;
id1
----
1
(1 row)

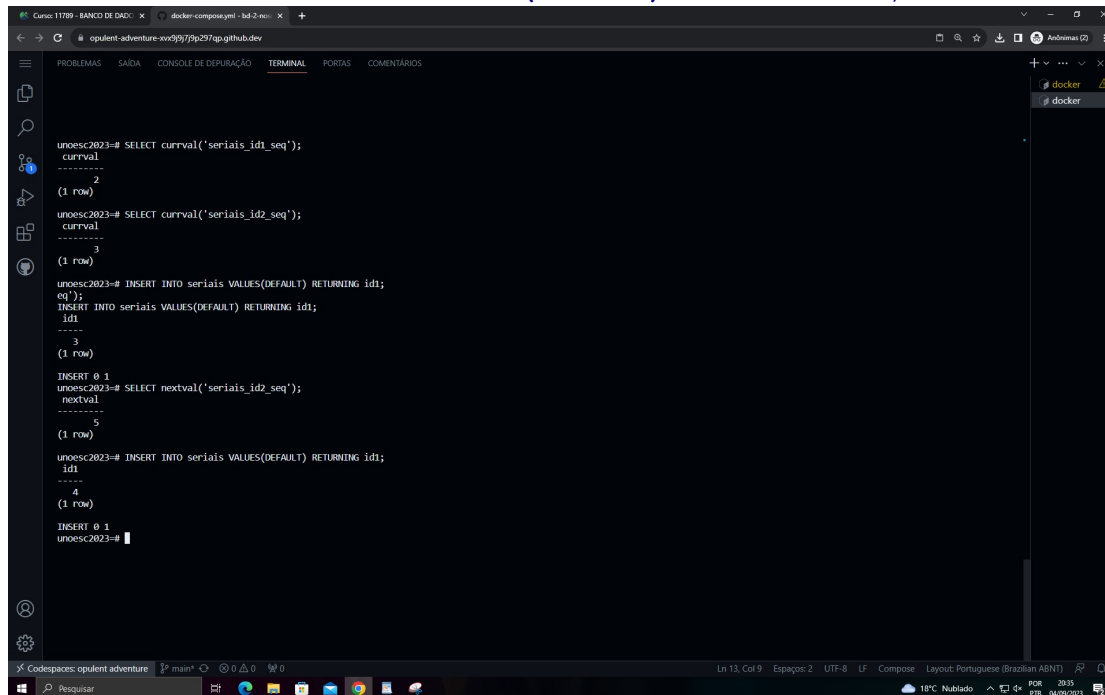
INSERT 0 1
unoesc2023=# INSERT INTO serials VALUES(DEFAULT, DEFAULT) RETURNING *;
id1 | id2
-----+-----
2 | 3
(1 row)

INSERT 0 1
unoesc2023=#
unoesc2023=#
unoesc2023=#
unoesc2023=#

```

- A função `currval()` retorna o valor atual da sequência. A função `nextval()` avança a sequência para o seu próximo valor. Execute os comandos abaixo e cole o *print* abaixo:

- `SELECT currval('seriais_id1_seq');`
- `SELECT currval('seriais_id2_seq');`
- `INSERT INTO seriais VALUES(DEFAULT) RETURNING id1;`
- `SELECT nextval('seriais_id2_seq');`
- `INSERT INTO seriais VALUES(DEFAULT) RETURNING id1;`



```

unoesc2023=# SELECT currval('seriais_id1_seq');
currval
-----
2
(1 row)

unoesc2023=# SELECT currval('seriais_id2_seq');
currval
-----
3
(1 row)

unoesc2023=# INSERT INTO seriais VALUES(DEFAULT) RETURNING id1;
INSERT INTO seriais VALUES(DEFAULT) RETURNING id1;
id1
---
3
(1 row)

INSERT 0 1
unoesc2023=# SELECT nextval('seriais_id2_seq');
nextval
-----
5
(1 row)

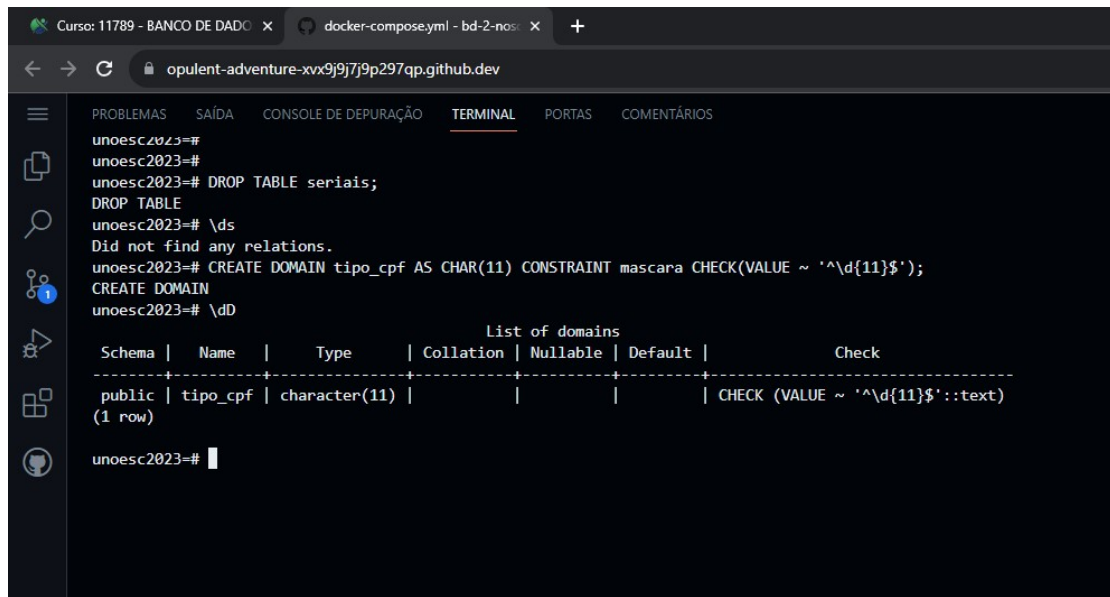
unoesc2023=# INSERT INTO seriais VALUES(DEFAULT) RETURNING id1;
INSERT INTO seriais VALUES(DEFAULT) RETURNING id1;
id1
---
4
(1 row)

INSERT 0 1
unoesc2023=#

```

- Exclua a tabela acima com o comando `DROP TABLE seriais`. Verifique com o comando `\ds` que as sequências também foram automaticamente excluídas.
- Um domínio permite a definição de um tipo com restrições. Execute o comando a seguir para criar um domínio que usa uma expressão regular para verificar se o valor possui 11 caracteres numéricos.
  - `CREATE DOMAIN tipo_cpf AS CHAR(11) CONSTRAINT mascara CHECK (VALUE ~ '^\\d{11}$');`

Execute o comando `\\dD` para listar os domínios do esquema atual e cole o resultado abaixo:



```
Curso: 11789 - BANCO DE DADO x docker-compose.yml - bd-2-nos: x +
opulent-adventure-xx9j7j9p297qp.github.dev

PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL PORTAS COMENTÁRIOS

unoesc2023=#
unoesc2023=#
unoesc2023=# DROP TABLE seriais;
DROP TABLE
unoesc2023=# \ds
Did not find any relations.
unoesc2023=# CREATE DOMAIN tipo_cpf AS CHAR(11) CONSTRAINT mascara CHECK(VALUE ~ '^d{11}$');
CREATE DOMAIN
unoesc2023=# \dD

List of domains
Schema | Name | Type | Collation | Nullable | Default | Check
-----|-----|-----|-----|-----|-----|-----
public | tipo_cpf | character(11) | | | | CHECK (VALUE ~ '^d{11}$::text)
(1 row)

unoesc2023=#
```

- Um tipo definido pelo usuário permite definir uma enumeração. Execute o comando abaixo:
  - `CREATE TYPE tipo_pagamento AS ENUM('cartao', 'boleto', 'dinheiro');`

Execute o comando `\dT+ tipo_pagamento` para mostrar detalhes sobre este tipo e cole abaixo o resultado:

```

unoesc2023=#
unoesc2023=#
unoesc2023=# DROP TABLE seriais;
DROP TABLE
unoesc2023=# \ds
Did not find any relations.
unoesc2023=# CREATE DOMAIN tipo_cpf AS CHAR(11) CONSTRAINT mascara CHECK(VALUE ~ '^d{11}$');
CREATE DOMAIN
unoesc2023=# \dD

          List of domains
Schema | Name | Type | Collation | Nullable | Default | Check
-----+-----+-----+-----+-----+-----+-----
public | tipo_cpf | character(11) | | | | CHECK (VALUE ~ '^d{11}$'::text)
(1 row)

unoesc2023=# CREATE TYPE tipo_pagamento AS ENUM('cartao', 'boleto', 'dinheiro');
CREATE TYPE
unoesc2023=# \dT+

          List of data types
Schema | Name | Internal name | Size | Elements | Owner | Access privileges | Description
-----+-----+-----+-----+-----+-----+-----+-----
public | tipo_cpf | tipo_cpf | var | | postgres | | 
public | tipo_pagamento | tipo_pagamento | 4 | cartao + boleto + dinheiro | postgres | | 
(2 rows)

unoesc2023=#

```

- Crie a tabela a seguir:

```

CREATE TABLE testa_restricoes (
    cpf tipo_cpf,
    forma_pgmtto tipo_pagamento NOT NULL
);

```

Tente agora realizar as seguintes inserções e cole ao lado o respectivo erro (se existir):

Comando	Erro
<pre> INSERT INTO testa_restricoes (cpf) VALUES ('123.123.123-12'); INSERT INTO testa_restricoes (cpf) VALUES ('abc12312312'); </pre>	<p>ERROR: value too long for type character(11)</p> <p>ERROR: value for domain tipo_cpf violates check constraint "mascara"</p>
<pre> INSERT INTO testa_restricoes (forma_pgmtto) VALUES ('cheque'); </pre>	<p>ERROR: invalid input value for enum tipo_pagamento: "cheque"</p> <p>LINE 1: ...NSERT INTO testa_restricoes (forma_pgmtto) VALUE ('cheque');</p>
<pre> INSERT INTO testa_restricoes (cpf) VALUES ('12312312312'); </pre>	<p>ERROR: null value in column "forma_pgmtto" of relation "testa_restricoes" violates not-null constraint</p> <p>DETAIL: Failing row contains (12312312312, null).</p>
<pre> INSERT INTO testa_restricoes VALUES ('11111111111', 'boleto'); </pre>	<p>Sem erro</p>

- Apague os tipos e tabela criadas acima:
  - **DROP TABLE** testa\_restricoes;

- `DROP TYPE` tipo\_cpf;
- `DROP TYPE` tipo\_pagamento;
- O PostgreSQL suporta a criação de tipos calculados. O exemplo a seguir cria um campo que automaticamente calcula o valor em polegadas baseado no valor em centímetros fornecido.

```
CREATE TABLE medidas (
    cm NUMERIC,
    polegadas NUMERIC GENERATED ALWAYS AS (cm * 2.54) STORED
);
INSERT INTO medidas VALUES (1), (10);
```

Liste o conteúdo da tabela e cole abaixo o resultado. Repare que o valor em polegadas foi automaticamente calculado. Após isso exclua a tabela criada.

```
unoesc2023=# CREATE TABLE medidas (
unoesc2023=# cm NUMERIC,
unoesc2023=# polegadas NUMERIC GENERATED ALWAYS AS (cm * 2.54) STORED
unoesc2023=# );
unoesc2023=# INSERT INTO medidas VALUES (1), (10);
unoesc2023=# INSERT 0 2
unoesc2023=# select * from medidas;
  cm | polegadas
-----+-----
   1 |       2.54
  10 |      25.40
(2 rows)

unoesc2023=#
```