

# BANCO DE DADOS

DDL



INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
RIO GRANDE DO NORTE

eliezio.soares@ifrn.edu.br | <https://docente.ifrn.edu.br/elieziosoares>

Msc. Eliezio Soares

# LINGUAGENS DE BANCO DE DADOS

**Um sistema de banco de dados fornece linguagens para definir e manipular dados e também para gerenciar acesso e transações:**

- **Linguagem de Definição de Dados (DDL - Data Definition Language)**
  - **Define o nível lógico de um banco de dados.**
- Linguagem de Manipulação de Dados (DML - Data Manipulation Language)
  - Insere, busca, remove e altera dados.

# LINGUAGENS DE BANCO DE DADOS

**Um sistema de banco de dados fornece linguagens para definir e manipular dados e também para gerenciar acesso e transações:**

- Linguagem de Controle de Dados (DCL – Data Control Language)
  - Linguagem de Controle de Dados, controla quem tem acesso para visualizar ou manipular dados dentro do banco de dados.
- Linguagem de Transação de Dados (DTL - Data Transaction Language)
  - Linguagem de Transação de Dados, usado para o controle de transações no banco de dados. Ex.:
    - ▢ BEGIN WORK (ou START TRANSACTION ou BEGIN TRAN)
    - ▢ COMMIT
    - ▢ ROLLBACK

# SQL

- **SQL - Structured Query Language**
  - Linguagem de Consulta Estruturada
- É uma **linguagem declarativa**, ou seja, não é necessário que você programe qualquer algoritmo para acessar o conteúdo das tabelas.
- SQL usa os termos **tabela, linha e coluna** para os termos do modelo relacional formal *relação, tupla e atributo*.

# CONCEITOS INICIAIS

## 0 Schemas:

- 0 Um schema é uma analogia a um diretório em um S.O, mas schemas não podem ser aninhados.
  - 0 Um schema agrupa tabelas, funções, domínios, permissões etc.
  - 0 Um (grupo de) usuário precisa ter permissão para acessar o conteúdo de um schema.
- 0 Um usuário pode ter acesso a apenas uma parte do banco de dados, caso este esteja disposto em mais de um esquema, impondo restrições de acesso a grupos de usuários distintos.
- 0 Permite muitos usuários usarem um banco de dados sem um interferir no outro.
- 0 Organiza os objetos do banco de dados em agrupamentos lógicos tornando-os mais gerenciáveis.

# CONCEITOS INICIAIS

## o **Catálogos:**

- o Os catálogos são o lugar onde um sistema de gerenciamento de banco de dados relacional armazena os metadados do esquema, tais como informações sobre tabelas e colunas, e informações de registros internos.
  
- o Os catálogos no PostgreSQL são tabelas comuns.

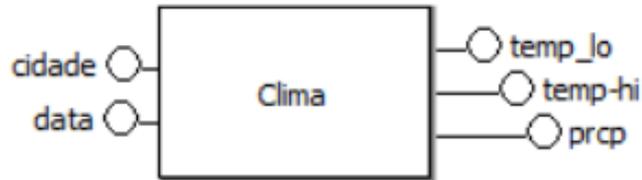
# COMANDO CREATE TABLE

- **CREATE TABLE** é utilizado para especificar uma nova relação, dando-lhe um nome e especificando seus atributos e restrições iniciais.
  - Os atributos recebem **nome**, **tipo (domínio)** e **restrições** (not null, por exemplo).
  - As **restrições de chave, integridade de entidade e integridade referencial** são especificadas em seguida.

## CREATE TABLE FUNCIONARIO...

- As tabelas declaradas nesse comando são criadas e, com suas tuplas, são armazenadas como um arquivo pelo SGBD.
  - São chamadas de relações da base.
- As relações virtuais são chamadas de VIEW (CREATE VIEW) e não necessariamente correspondem a um arquivo físico real.

# COMANDO CREATE TABLE



Clima (cidade, temp\_lo, temp\_hi, prcp, data)

Espaços em branco (espaços, tabs e linhas) são livres em comandos SQL.

```
CREATE TABLE Clima (  
  cidade varchar(80),  
  temp_lo int, -- low temperature  
  temp_hi int, -- high temperature  
  prcp real, -- precipitação  
  data date  
);
```

Dois traços (“--”) introduz comentários ao comando SQL. A partir desse sinal a linha será ignorada.

# TIPOS DE DADOS EM SQL

- **Numéricos**
- **Cadeia de Caracteres**
- **Cadeia de bits**
- **Booleano**
- **Data**
- **Hora**

# TIPOS DE DADOS EM SQL

## NUMÉRICOS

Nome	Tamanho Armazenado	Descrição	Range
smallint	2 bytes	Inteiro de baixo range	-32768 a +32767
integer	4 bytes	Inteiro típico	-2147483648 a +2147483647
bigint	8 bytes	Inteiro de alto range	-9223372036854775808 a 9223372036854775807
Decimal	Variável	Precisão especificada pelo usuário NUMERIC( <i>precision</i> )	-
Numeric	Variável	Precisão especificada pelo usuário	-
Real	4 bytes	Precisão varável (1 a 24)	
Double	8 bytes	Precisão varável (25 a 53)	
Serial	4 bytes	<b>Inteiro auto-incremento</b>	1 a 2147483647
bigserial	8 bytes	<b>Inteiro auto-incremento grande</b>	1 a 9223372036854775807

# TIPOS DE DADOS EM SQL

## o **Real e Double**

- o Armazena valores flutuantes e possibilita eficientes operações aritméticas.
- o Se o objetivo é precisão nos cálculos, usar Numeric ou Decimal.

## o **Numeric e Decimal**

- o Pode armazenar números até 1000 dígitos de precisão e realizar cálculos exatos.
- o É recomendado para armazenar valores monetários.
- o Aritmética com valores *numeric* é lenta, se comparada a valores inteiros ou float.

## o **Serial e bigserial**

- o Não são propriamente tipos de dados, mas uma notação.
- o Declara-se identificadores únicos e auto incrementáveis.
- o O tipo será inteiro;

# TIPOS DE DADOS EM SQL

## CARACTERES

Nome	Descrição
Character varying(n), <b>varchar(n)</b>	Tamanho variável com limite.
Character(n), <b>char(n)</b>	Tamanho fixo, espaços em branco preenchem.
<b>text</b>	Tamanho ilimitado.

# TIPOS DE DADOS EM SQL

## CADEIA DE BITS

Nome	Descrição
bit varying(n), <b>bit(n)</b>	Tamanho variável com limite.
bit(n), <b>bit(n)</b>	Tamanho fixo, espaços em branco preenchem.

# TIPOS DE DADOS EM SQL

## DATA / HORA

Nome	Tamanho Armazenado	Descrição
timestamp [ (p) ] [ without time zone ]	8 bytes	Data e Hora
timestamp [ (p) ] with time zone	8 bytes	Data e Hora com fuso horário
date	4 bytes	Somente data
time [ (p) ] [ without time zone ]	8 bytes	Somente hora
time [ (p) ] with time zone	12 bytes	Somente hora com fuso horário
interval [ fields ] [ (p) ]	16 bytes	Intervalo de tempo

# TIPOS DE DADOS EM SQL

## DATA - ENTRADAS

Example	Description
1999-01-08	ISO 8601; January 8 in any mode (recommended format)
January 8, 1999	unambiguous in any <code>datestyle</code> input mode
1/8/1999	January 8 in MDY mode; August 1 in DMY mode
1/18/1999	January 18 in MDY mode; rejected in other modes
01/02/03	January 2, 2003 in MDY mode; February 1, 2003 in DMY mode; February 3, 2001 in YMD mode
1999-Jan-08	January 8 in any mode
Jan-08-1999	January 8 in any mode
08-Jan-1999	January 8 in any mode
99-Jan-08	January 8 in YMD mode, else error
08-Jan-99	January 8, except error in YMD mode
Jan-08-99	January 8, except error in YMD mode
19990108	ISO 8601; January 8, 1999 in any mode
990108	ISO 8601; January 8, 1999 in any mode
1999.008	year and day of year
J2451187	Julian day
January 8, 99 BC	year 99 BC

# TIPOS DE DADOS EM SQL

## DATA - ENTRADAS

Example	Description
04:05:06.789	ISO 8601
04:05:06	ISO 8601
04:05	ISO 8601
040506	ISO 8601
04:05 AM	same as 04:05; AM does not affect value
04:05 PM	same as 16:05; input hour must be <= 12
04:05:06.789-8	ISO 8601
04:05:06-08:00	ISO 8601
04:05-08:00	ISO 8601
040506-08	ISO 8601
04:05:06 PST	time zone specified by abbreviation
2003-04-12 04:05:06 America/New_York	time zone specified by full name

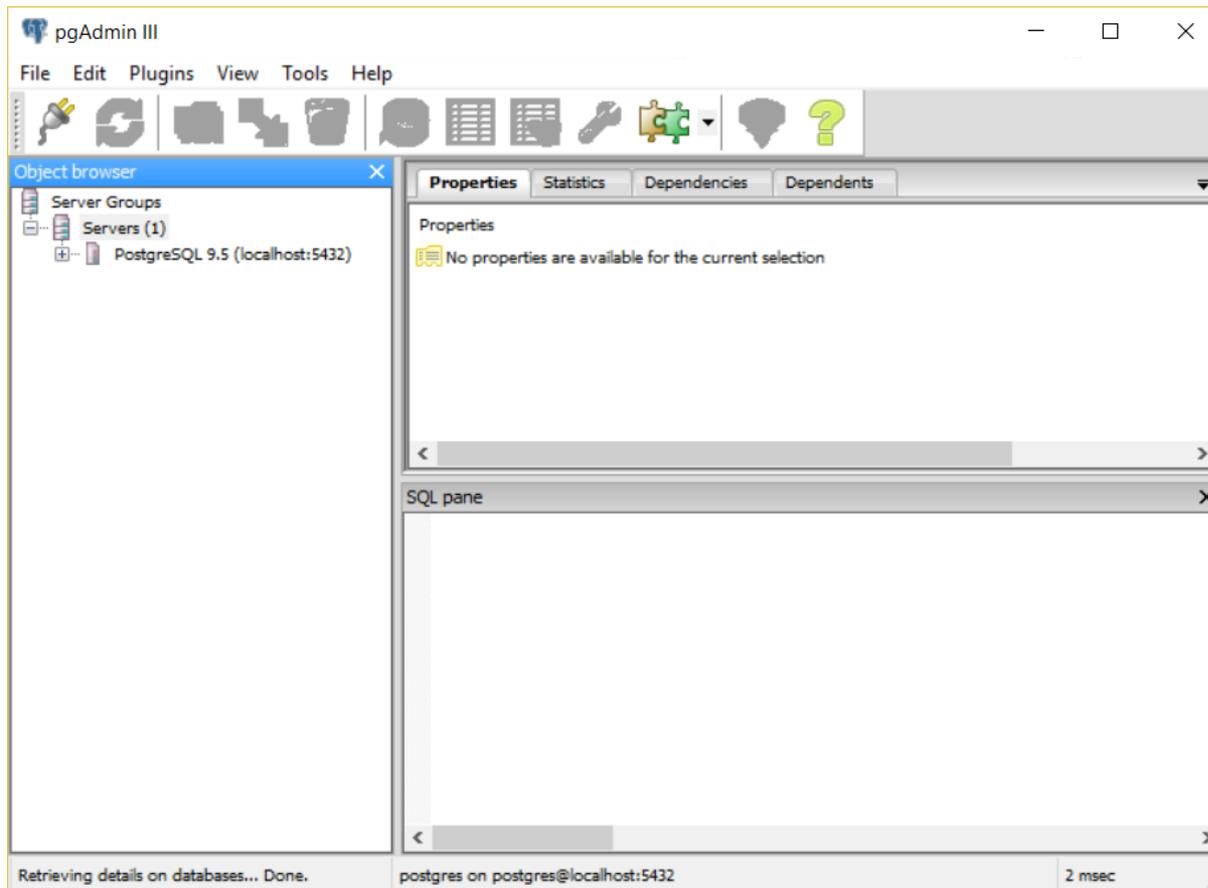
# TIPOS DE DADOS EM SQL

## BOOLEAN

Nome	Tamanho Armazenado	Descrição
boolean	1 byte	Estado de verdadeiro ou falso.

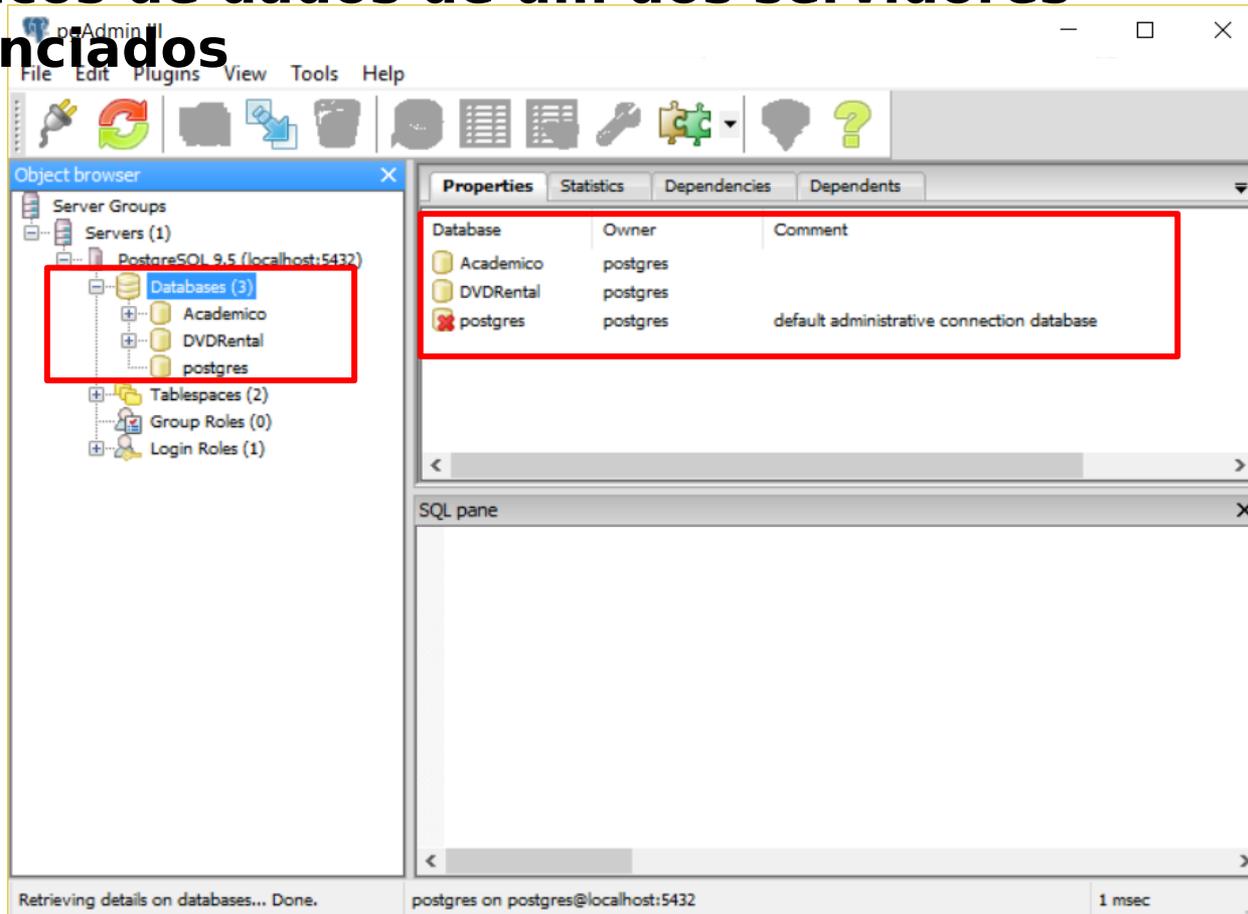
# CRIANDO O PRIMEIRO BANCO DE DADOS

## 0 Tela inicial



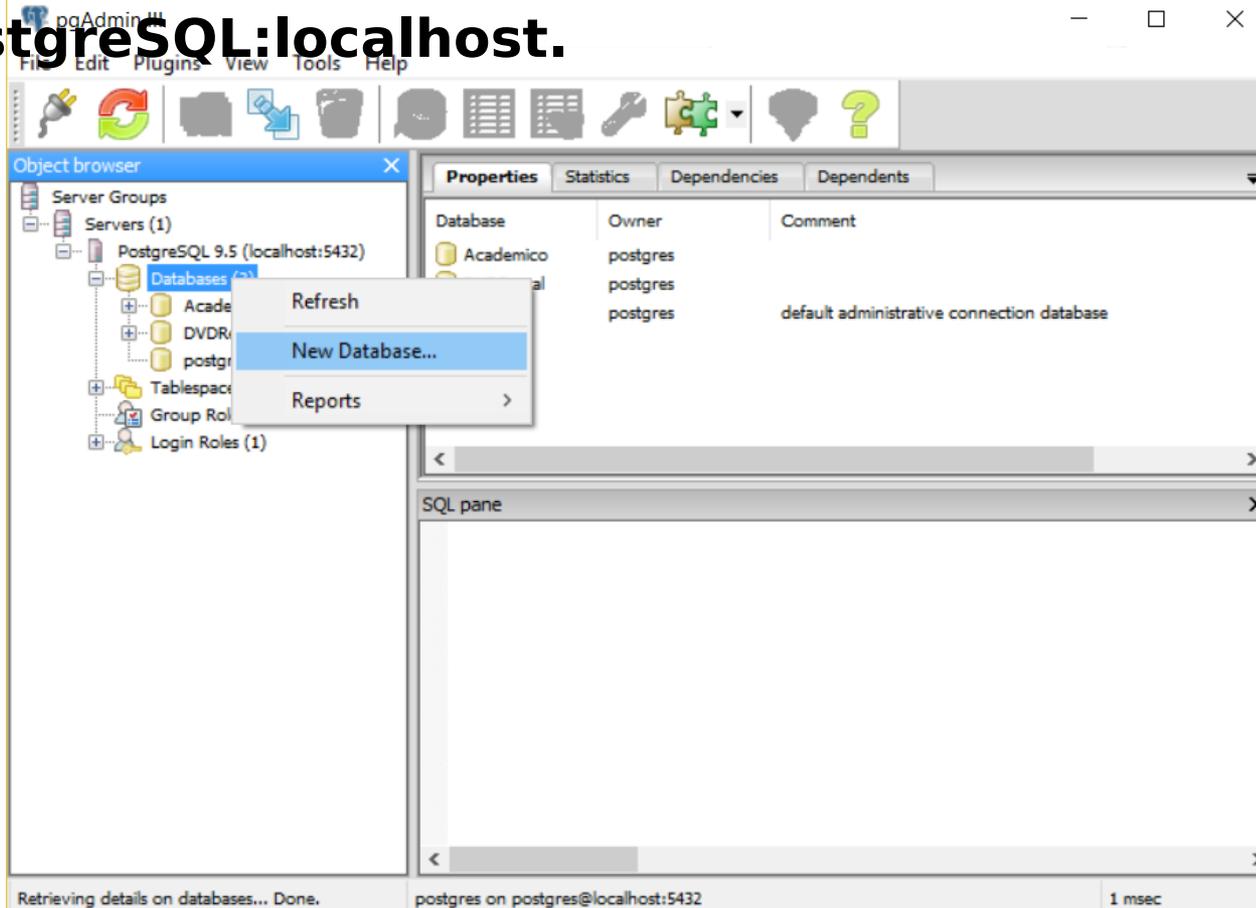
# CRIANDO O PRIMEIRO BANCO DE DADOS

0 Bancos de dados de um dos servidores gerenciados



# CRIANDO O PRIMEIRO BANCO DE DADOS

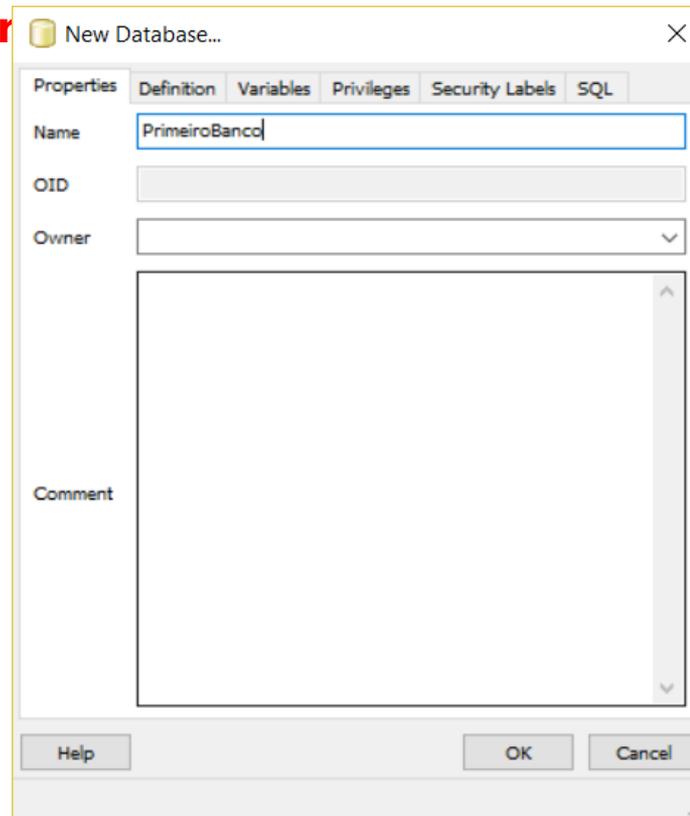
0 Criando um novo banco de dados em PostgreSQL:localhost.



# CRIANDO O PRIMEIRO BANCO DE DADOS

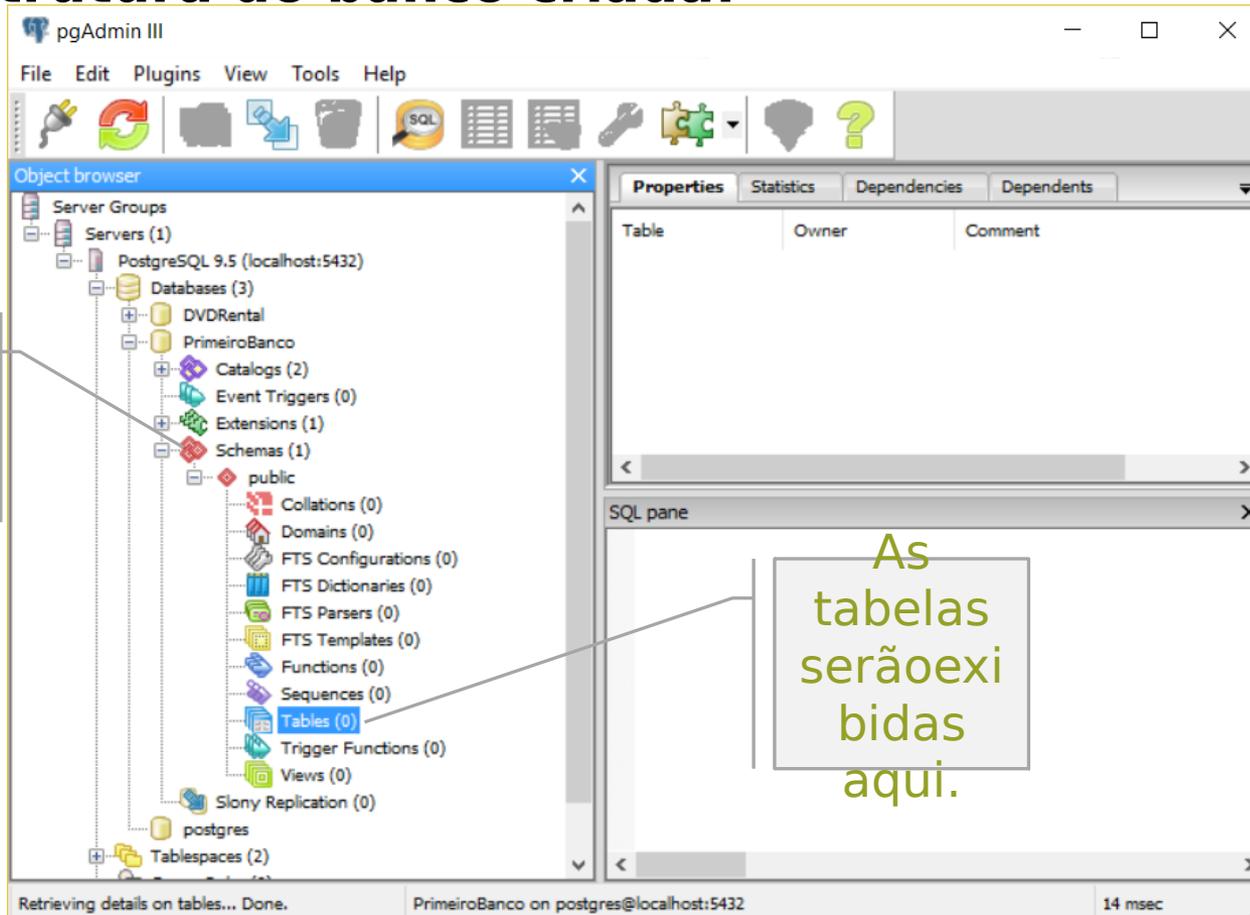
○ Criando um novo banco de dados em PostgreSQL:localhost.

○ Nome: **Primeiro**



# CRIANDO O PRIMEIRO BANCO DE DADOS

o Estrutura do banco criada:

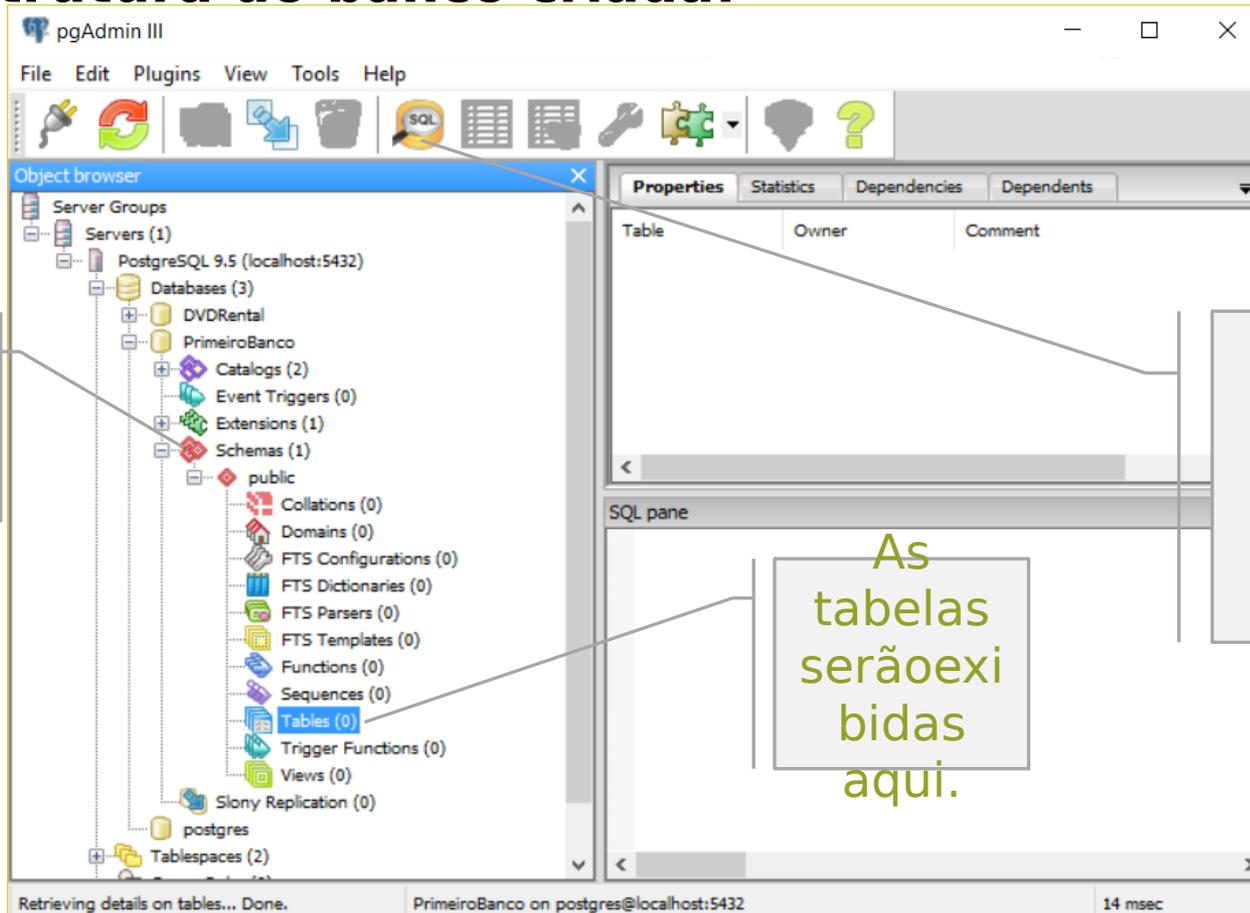


Schem  
a  
padrão  
criado;

As  
tabelas  
serão exi  
bidas  
aqui.

# CRIANDO O PRIMEIRO BANCO DE DADOS

o Estrutura do banco criada:



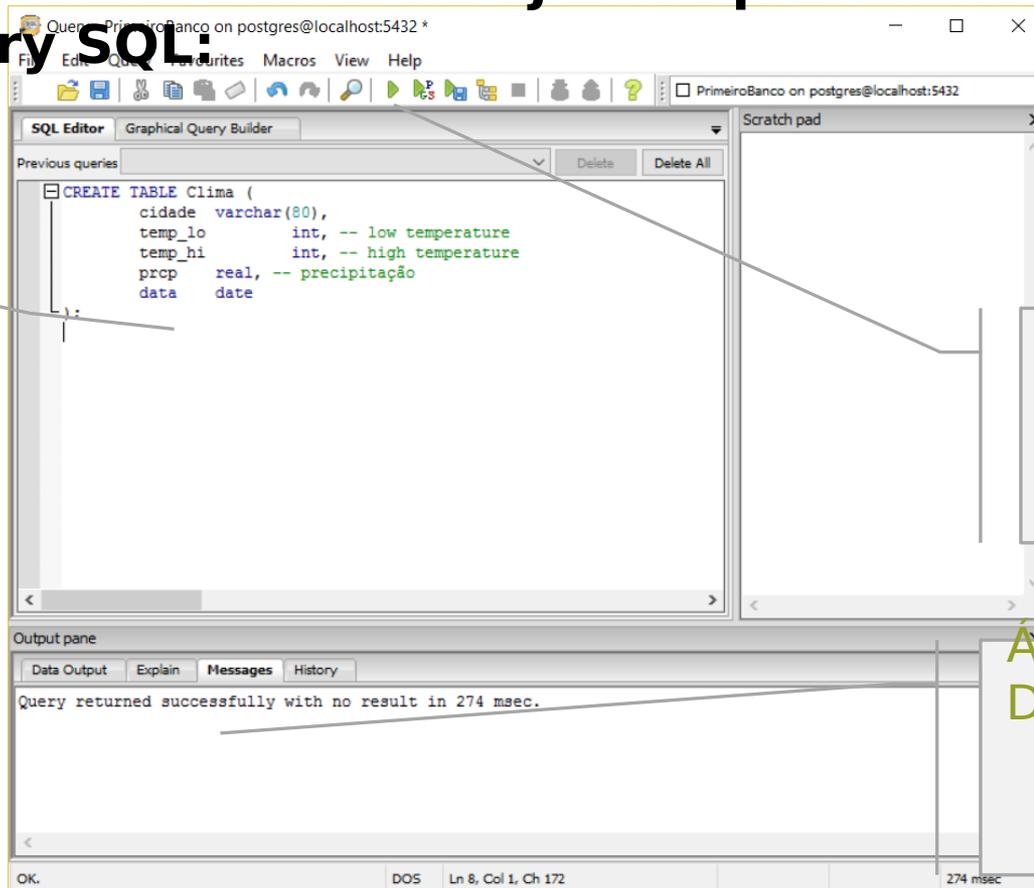
Schem  
a  
padrão  
criado;

Estando no  
contexto do  
banco criado,  
clique no  
botão SQL

As  
tabelas  
serão exi  
bidas  
aqui.

# CRIANDO O PRIMEIRO BANCO DE DADOS

- O PGAdmin abrirá uma janela para a execução de query SQL:



Área de edição de query

Copie o comando de criação da tabela clima e clique em executar.

Área de resultados: Dados, Mensagens, Historico, Detalhamento da query.

# CRIANDO O PRIMEIRO BANCO DE DADOS

O PGAdmin abrirá uma janela para a execução de query SQL:

The screenshot displays the PGAdmin interface. On the left, the 'Object browser' shows a tree view of the database structure. Under the 'public' schema, a table named 'clima' is highlighted. A callout box points to this table with the text 'Tabela criada'. The 'Properties' pane on the right shows details for the 'clima' table, including its name, OID, owner, tablespace, ACL, and primary key. The 'SQL pane' at the bottom shows the SQL command used to create the table:

```
-- Table: public.clima
-- DROP TABLE public.clima;

CREATE TABLE public.clima
(
  cidade character varying(80),
  temp_lo integer,
  temp_hi integer,
  prcp real,
  data date
)
```

# CRIANDO O PRIMEIRO BANCO DE DADOS

○ **Apenas para visualizarmos, execute os comandos abaixo:**

○ `insert into Clima(cidade, temp_lo, temp_hi, prcp, data) values('Mossoró', -10,15,94.5, current_timestamp);`

○ `select * from Clima;`

# VALOR PADRÃO DE UMA COLUNA

- Uma coluna pode ser declarada com um valor padrão.
- Ao se criar uma nova linha, caso não seja especificado um valor, a coluna será inserida com o valor padrão.
- Se não há um valor padrão, a coluna é inserida com valor NULL.

```
CREATE TABLE Produto (
  produto_id integer, name
  text, price numeric DEFAULT 9.99 );
  nome text,
  preco numeric DEFAULT 9.99
);
```

# DÚVIDAS?



# REFERÊNCIAS BIBLIOGRÁFICAS

ELMASRI, Ramez. Sistemas de Banco de Dados. **Projeto de Banco de Dados**. 6ª Edição. São Paulo. Pearson Addison Wesley, 2011.

HEUSER, Carlos Alberto. **Projeto de banco de dados**. 6ª Edição. Porto Alegre. Bookman, 2009.

PostgreSQL 9.0.22 Documentation. Disponível em: <<https://www.postgresql.org/files/documentation/pdf/9.0/postgresql-9.0-US.pdf>>. Acesso em 27 Set. 2016.