

1 Introdução

SQL significa **Structured Query Language**. Trata-se de um padrão baseado na álgebra relacional. A Tabela 1.1 mostra os comandos previstos pelo padrão SQL.

Tabela 1.1

Data Definition Language	Data Manipulation Language	Data Control Language	Transaction Control Language	Data Query Language
CREATE (criar tabela)	INSERT (inserir dados em tabelas)	GRANT (atribuir privilégios de acesso)	COMMIT (confirmar operações realizadas em transações)	SELECT (buscar dados em tabelas)
ALTER (alterar estrutura de tabela)	UPDATE (atualizar dados em tabelas)	REVOKE (remover privilégios de acesso)	ROLLBACK (desfazer operações realizadas em transações ainda não concluídas)	
DROP (apagar tabela)	DELETE (remover dados de tabelas)		SAVEPOINT (criar pontos intermediários ao longo de transações, permitindo que apenas parte delas sejam desfeitas, quando necessário)	
TRUNCATE (apagar dados de tabela sem registrar remoções no log)				

Há muitas implementações do padrão SQL. Além das operações padrão previstas na especificação SQL, fabricantes de Sistemas Gerenciadores de Bancos de Dados Relacionais fazem a implementação de extensões, tornando seus produtos mais interessantes para os desenvolvedores. Entre muitos outros recursos, que variam de produto para produto, estas extensões costumam trazer a implementação de **estruturas de seleção e de repetição**, presentes nas linguagens de programação mais comuns. Veja alguns exemplos de extensões ao padrão SQL na Tabela 1.2.

Nome da implementação	Fabricante	SGBD
Transact-SQL (T-SQL)	Microsoft	Microsoft SQL Server
PL/SQL	Oracle	Oracle
PL/pgSQL	PostgreSQL	PostgreSQL

2 Desenvolvimento

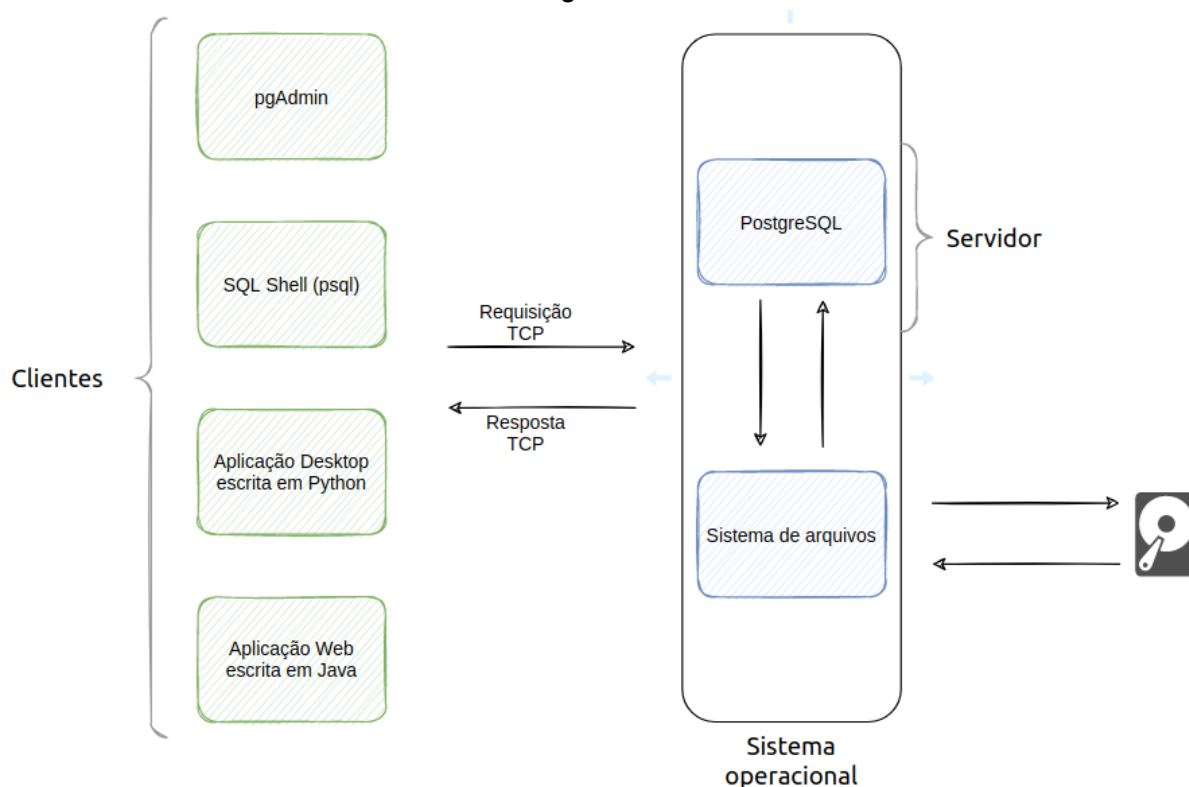
2.1 (Arquitetura cliente/servidor) O uso do PostgreSQL se baseia na arquitetura cliente/servidor. O servidor PostgreSQL é responsável por

- compilar e executar comandos SQL
- manipular os arquivos utilizados para o armazenamento de dados

entre outras muitas coisas. Ou seja, o servidor PostgreSQL é responsável por prover a abstração por meio da qual enxergamos nossos dados como se fossem armazenados em tabelas.

O acesso a ele requer o uso de um cliente. Podemos usar diferentes clientes, como o **pgAdmin** e **SQL Shell (psql)**. Também podemos desenvolver nossos próprios clientes utilizando linguagens de programação das mais diversas. Veja a Figura 2.1.

Figura 2.1

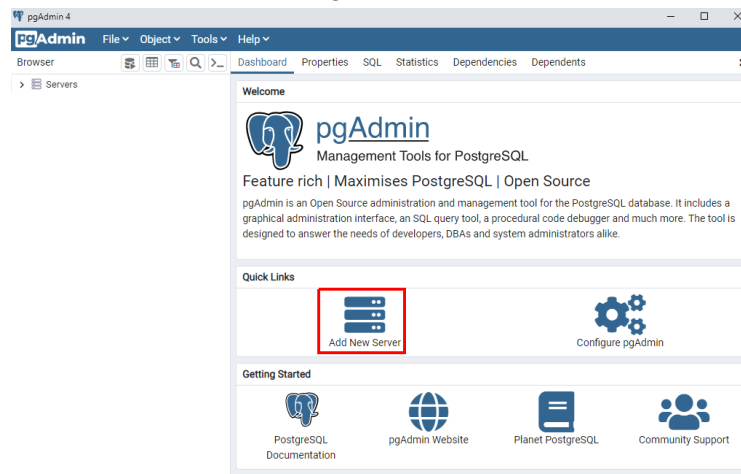


2.2 (Criando um “servidor” no pgAdmin) Como vimos, o pgAdmin é um cliente próprio para o PostgreSQL. Ele pode ser utilizado para acessar múltiplos servidores PostgreSQL. Por isso, precisamos especificar os detalhes que caracterizam o servidor a ser utilizado:

- endereço do host em que ele se encontra executando
- porta em que ele está executando
- usuário e senha a serem utilizados para autenticação/autorização

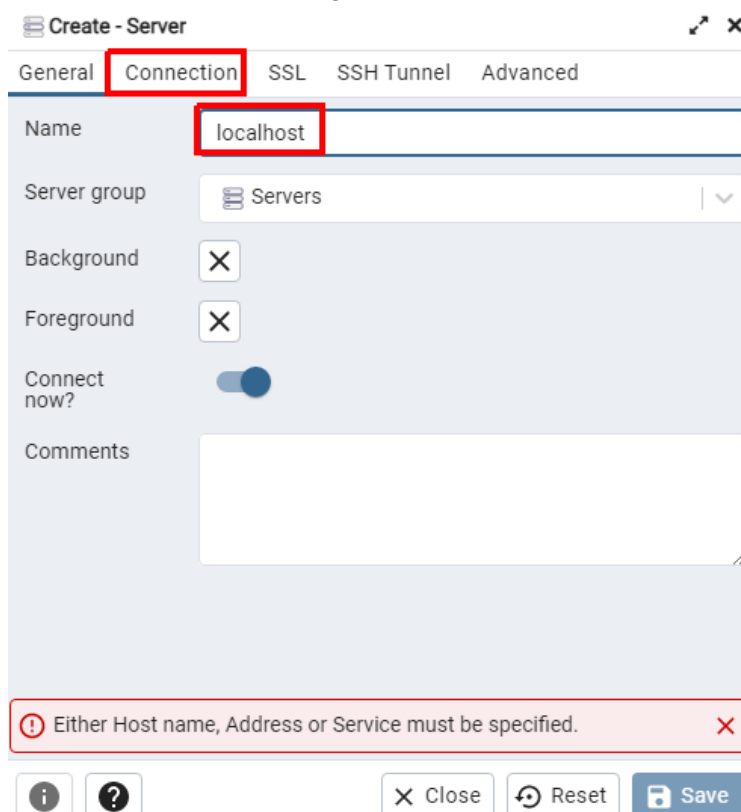
A tela inicial do pgAdmin é exibida pela Figura 2.2.1. Comece clicando em **Add New Server**.

Figura 2.2.1



A seguir, dê um nome para o servidor cujos dados serão armazenados pelo pgAdmin. Esse pode ser um nome qualquer, que você possa utilizar para facilmente para se lembrar a que servidor se refere (localhost, dev, producao são exemplos). Como o servidor está em execução no mesmo computador em que o cliente está, podemos escolher um nome intuitivo como **localhost**. A seguir, clique na aba **Connection**. Veja a Figura 2.2.2.

Figura 2.2.2



Já na aba Connection, como mostra a Figura 2.2.3, preencha os campos destacados e clique em **Save**.

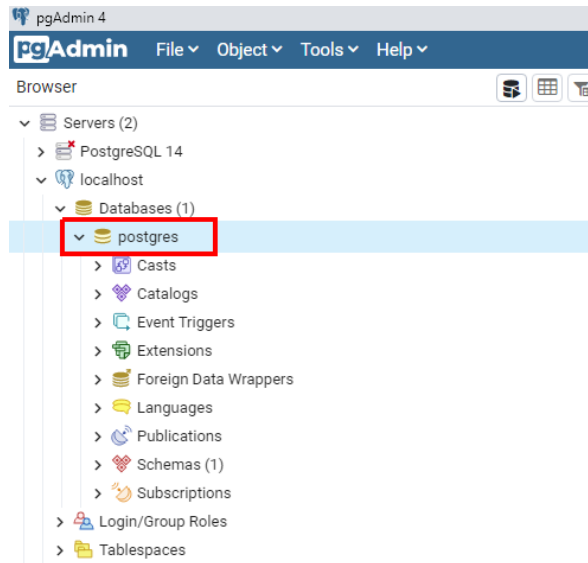
Figura 2.2.3

The image shows the 'Create - Server' dialog box in pgAdmin, specifically the 'Connection' tab. The dialog has several tabs: 'General', 'Connection', 'SSL', 'SSH Tunnel', and 'Advanced'. The 'Connection' tab is active. The fields and their values are: 'Host name/address' (127.0.0.1), 'Port' (5432), 'Maintenance database' (postgres), 'Username' (postgres), 'Kerberos authentication?' (toggle off), 'Password' (masked with dots), 'Save password?' (toggle on), 'Role' (empty), and 'Service' (empty). The 'Save' button at the bottom right is highlighted with a red box. There are also 'Close' and 'Reset' buttons.

Nota. No Windows, a senha é normalmente especificada no momento em que o PostgreSQL é instalado.

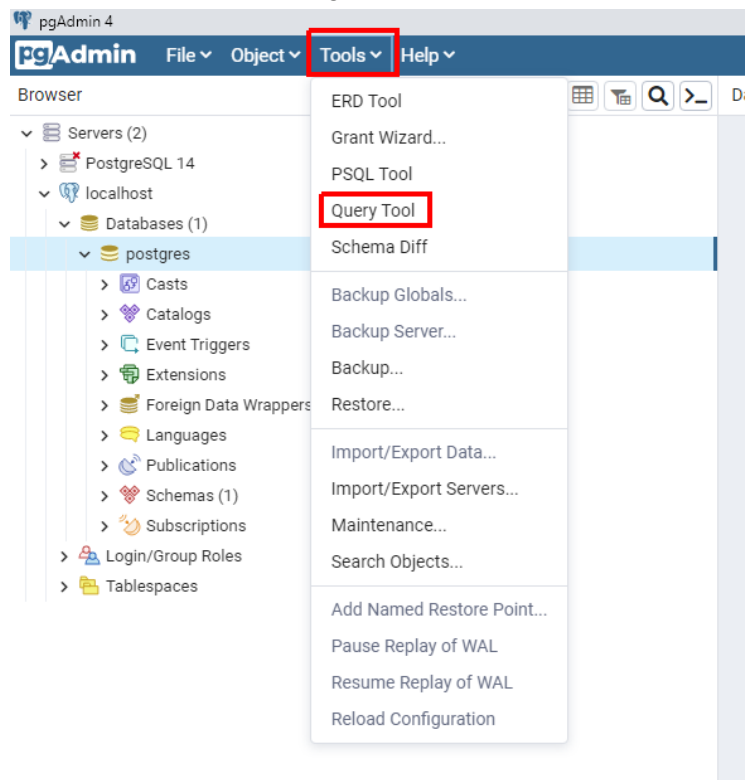
2.3 (Databases) No PostgreSQL, cada usuário tem, por padrão, uma base de dados com o seu nome. Ela pode ser vista no canto esquerdo do pgAdmin, como destaca a Figura 2.3.1. Neste momento, **clique sobre ela**.

Figura 2.3.1



2.4 (Editor SQL) Clicando **Tools >> Query Tool**, como na Figura 2.4.1, podemos abrir um editor de texto em que comandos SQL podem ser digitados.

Figura 2.4.1



2.5 (Teste: criando uma tabela, fazendo uma inserção e uma recuperação de dados)

Use os comandos do Bloco de Código 2.5.1 para

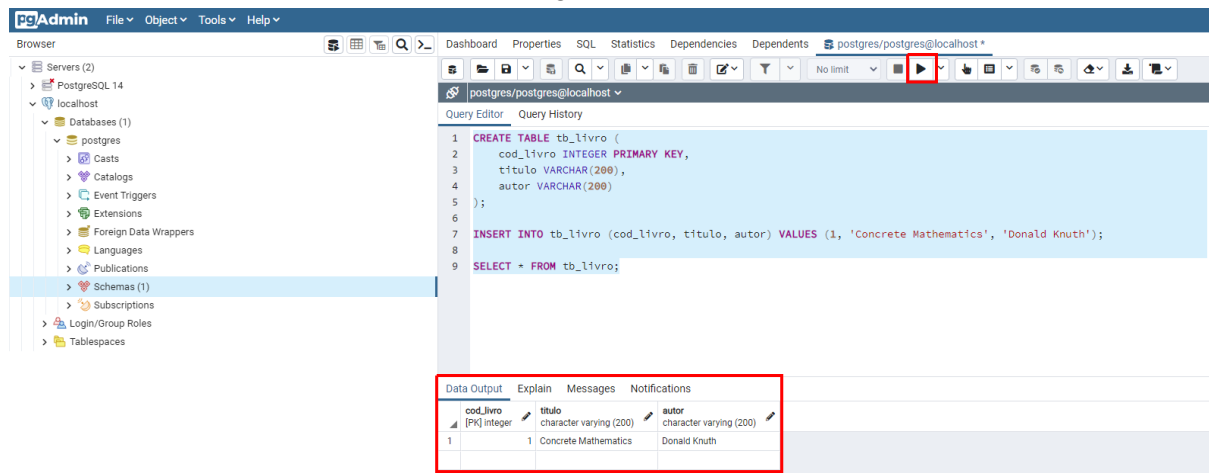
- criar uma tabela
- inserir uma linha
- buscar os dados tabela

Bloco de Código 2.5.1

```
CREATE TABLE tb_livro (  
    cod_livro INTEGER PRIMARY KEY,  
    titulo VARCHAR(200),  
    autor VARCHAR(200)  
);  
  
INSERT INTO tb_livro (cod_livro, titulo, autor) VALUES (1, 'Concrete Mathematics',  
'Donald Knuth');  
  
SELECT * FROM tb_livro;
```

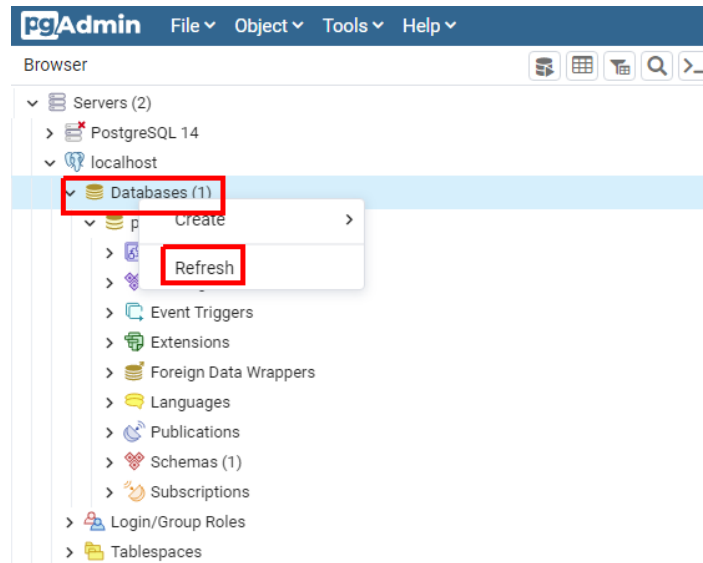
Selecione todo o conteúdo e clique em **Execute**, como destaca a Figura 2.5.1. Ela também destaca o resultado esperado, logo abaixo.

Figura 2.5.1



Clique com o direito em **Databases >> Refresh**, como na Figura 2.5.2.

Figura 2.5.2



A seguir, faça as expansões destacadas pela Figura 2.5.2 e veja como é possível interagir com a tabela criada graficamente.

Figura 2.5.2



Clique duas vezes na “linha em branco” logo abaixo da primeira linha preenchida, como destaca a Figura 2.5.3. Repare como é possível inserir dados na tabela graficamente. Clique duas vezes em cada campo da linha em branco para registrar um dado de interesse. Ao final, clique em **Save Data Changes**, opção cujo botão também é destacado.

Figura 2.5.3

The screenshot shows the PostgreSQL GUI interface. The top menu bar includes Dashboard, Properties, SQL, Statistics, Dependencies, and Dependents. The current tab is 'postgres/postgres@localhost *'. The toolbar contains various icons, with the 'Save Data Changes' icon (a disk with a checkmark) highlighted by a red box. Below the toolbar, the 'Query Editor' tab is active, displaying the following SQL code:

```
1 CREATE TABLE tb_livro (  
2     cod_livro INTEGER PRIMARY KEY,  
3     titulo VARCHAR(200),  
4     autor VARCHAR(200)  
5 );  
6  
7 INSERT INTO tb_livro (cod_livro, titulo, autor) VALUES (1, 'Concrete Mathematics', 'Donald Knuth');  
8  
9 SELECT * FROM tb_livro;
```

Below the query editor, the 'Data Output' tab is active, showing a table with the following data:

	cod_livro [PK] integer	titulo character varying (200)	autor character varying (200)
1	1	Concrete Mathematics	Donald Knuth
2	2	How to prove it	Daniel Velleman

The second row of the table is highlighted with a red box, indicating the new data entry.

Bibliografia

HEUSER, Carlos Alberto. Projeto de Banco de Dados. 6a Ed., Bookman, 2008.