

Ciência da Computação
GBC043 Sistemas de Banco de Dados



Linguagem SQL

Profa. Maria Camila Nardini Barioni

camila.barioni@ufu.br

Bloco B - sala 1B137

1º semestre de 2024

SQL

Structured Query Language

- ◆ Desenvolvida e implementada pelo laboratório de pesquisa da IBM em San Jose – final da década de 60 e início da década de 70
- ◆ Inicialmente chamada de SEQUEL (*Structured English QUery Language*)
- ◆ Criada como interface entre usuários e o primeiro SGBDR – SYSTEM R

SQL

Structured Query Language

- ◆ Uma das mais importantes linguagens relacionais (se não a mais importante)
- ◆ Exemplos de SGBD que utilizam SQL
 - Oracle
 - Informix
 - Ingress
 - SQL Server
 - Interbase
 - SyBase
 - DB2
 - MySQL
 - PostgreSQL

SQL

Structured Query Language

- ◆ Atrativo: pequena quantidade de comandos para realizar todas as operações necessárias para definição e manipulação de relações
 - Simplicidade
 - Grande poder de consulta

- ◆ Padrão facilita migração

Composição do SQL

◆ Linguagem de Definição dos Dados (DDL)

- comandos para a definição, a modificação e a remoção de relações, além da criação e da remoção de índices

◆ Linguagem Interativa de Manipulação dos Dados (DML)

- comandos para a consulta, a inserção, a remoção e a modificação de tuplas no banco de dados

Composição do SQL

- ◆ Linguagem de Manipulação dos Dados Embutida
 - pode ser utilizada a partir de linguagens de programação de propósito geral
- ◆ Definição de visões
 - SQL DDL inclui comandos para a criação e a remoção de visões
- ◆ Restrições de integridade
 - SQL DDL possui comandos para a especificação de restrições de integridade

Composição do SQL

◆ Autorização

- SQL DDL inclui comandos para a especificação de direitos de acesso a relações e visões

◆ Gerenciamento de transações

- introduz comandos para a especificação do início e do fim das transações

◆ Recuperação de falhas

- introduz comandos para utilização do arquivo de *log*



<http://www.postgresql.org/>

Aulas Práticas (SGBD utilizado)

A screenshot of the PostgreSQL website homepage. The browser window shows the URL "postgresql.org" and a tab titled "PostgreSQL: The world's most...". The website header includes a navigation menu with links: Home, About, Download, Documentation, Community, Developers, Support, Donate, and Your account. A search bar and a moon icon for dark mode are also present. A blue banner at the top of the main content area reads "8th February 2024: PostgreSQL 16.2, 15.6, 14.11, 13.14, and 12.18 Released!". Below this, a large image of elephants in a savanna serves as a background for the main heading: "PostgreSQL: The World's Most Advanced Open Source Relational Database". Two buttons are overlaid on the image: "Download →" and "New to PostgreSQL?". At the bottom of the page, there are two columns. The left column features the PostgreSQL logo and the text "New to PostgreSQL?". The right column features a download icon (a blue arrow pointing down into a box) and the text "Latest Releases".

PostgreSQL: The world's most advanced open source relational database

8th February 2024: [PostgreSQL 16.2](#), [15.6](#), [14.11](#), [13.14](#), and [12.18](#) Released!

PostgreSQL: The World's Most Advanced Open Source Relational Database

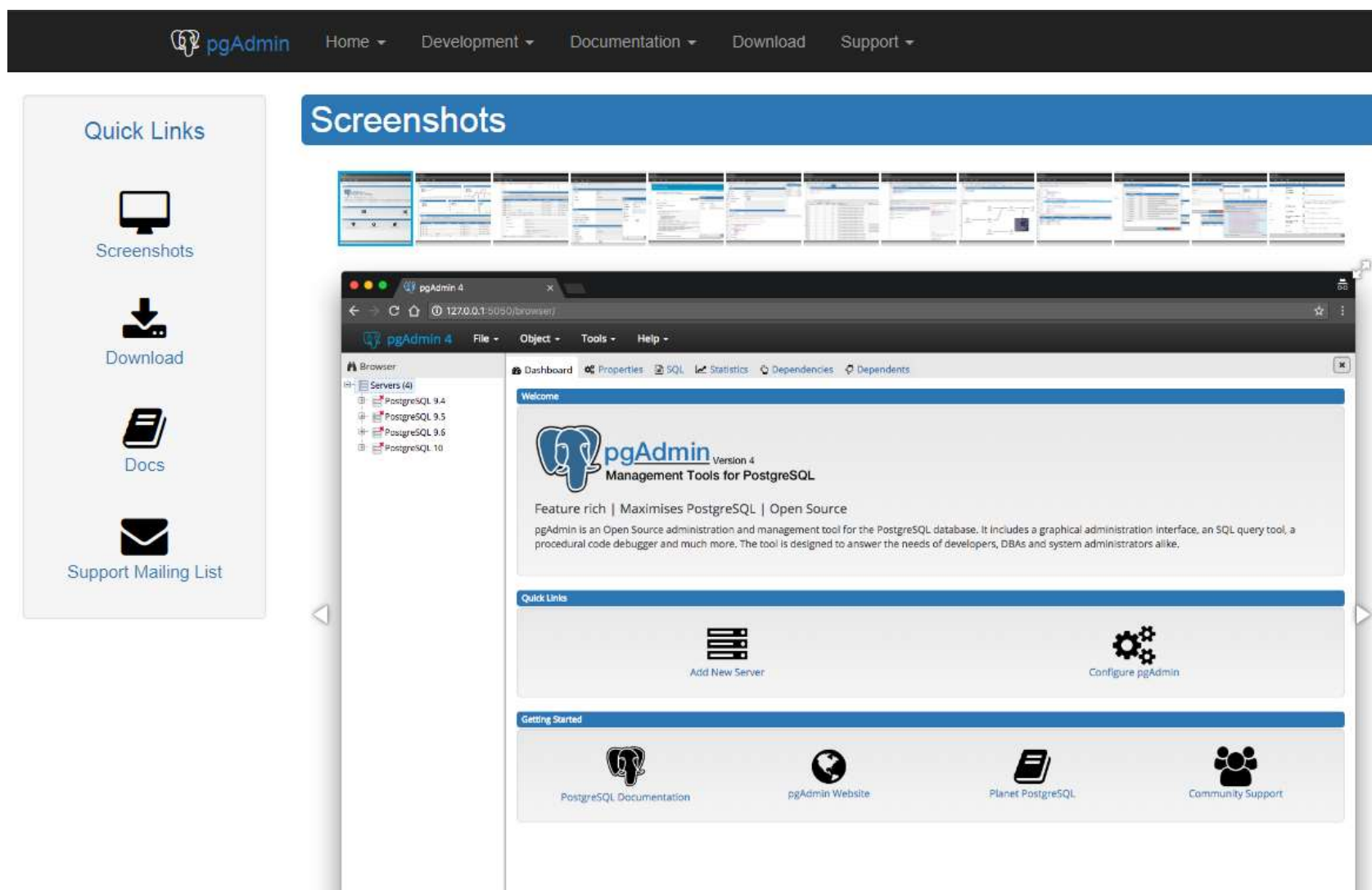
Download → New to PostgreSQL?

New to PostgreSQL?

Latest Releases

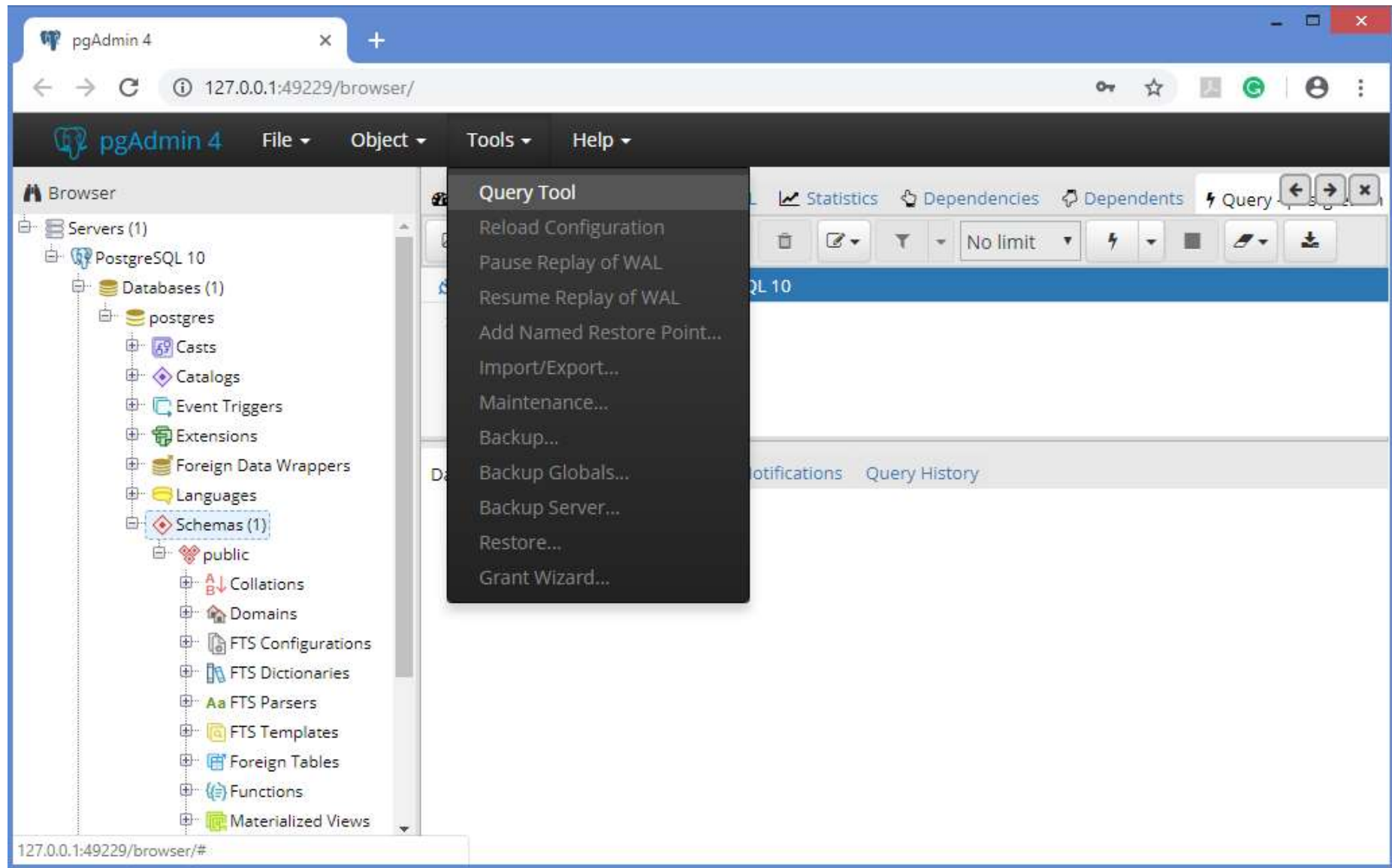
Aulas Práticas

◆ Para rodar os comandos



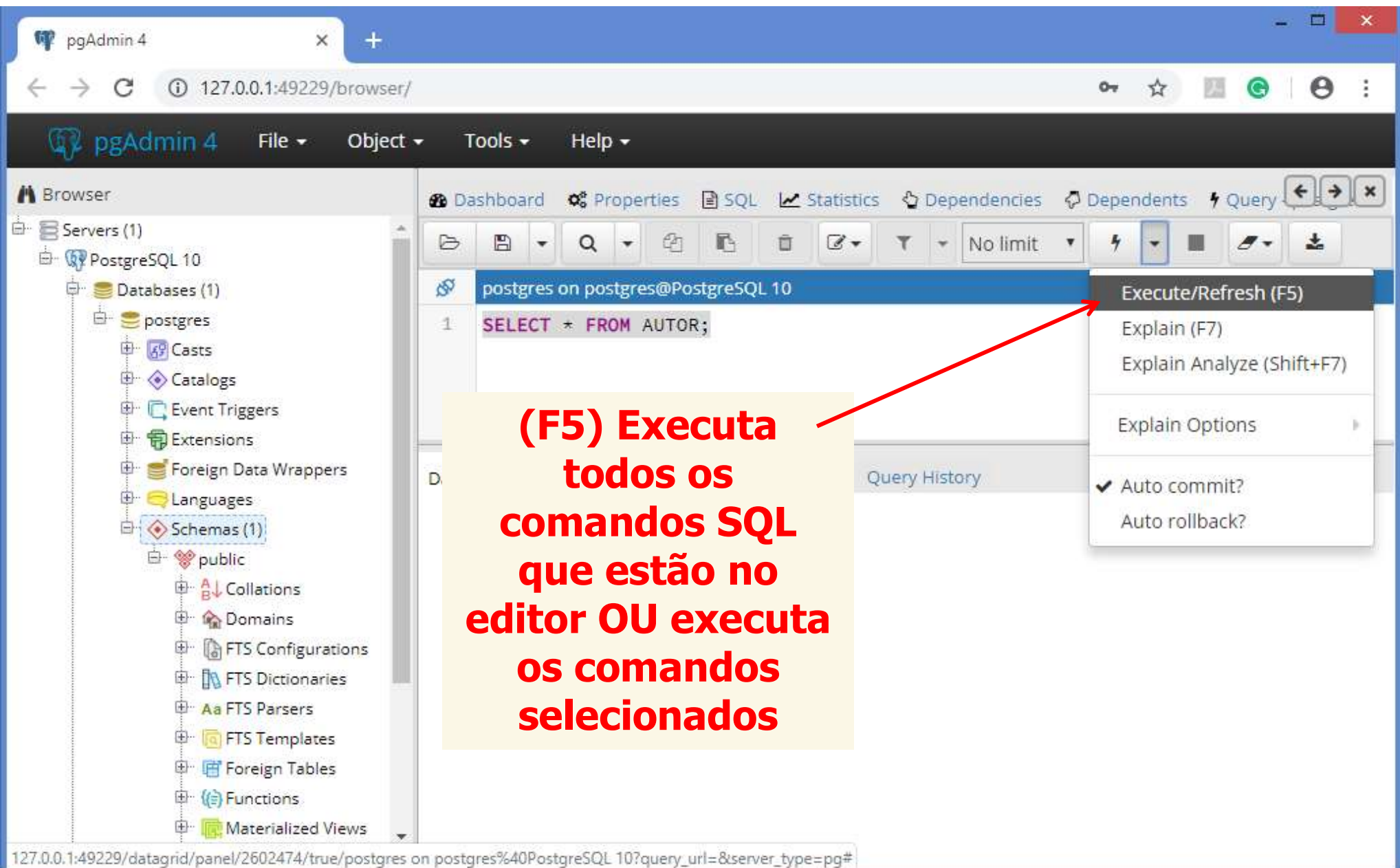
The image shows the pgAdmin 4 website and its user interface. The website header includes the pgAdmin logo and navigation links: Home, Development, Documentation, Download, and Support. A 'Quick Links' sidebar on the left contains icons for Screenshots, Download, Docs, and Support Mailing List. The main content area features a 'Screenshots' section with a row of thumbnail images. Below this is a large screenshot of the pgAdmin 4 web interface. The interface shows a 'Welcome' message, a 'Quick Links' section with 'Add New Server' and 'Configure pgAdmin' buttons, and a 'Getting Started' section with links to PostgreSQL Documentation, pgAdmin Website, Planet PostgreSQL, and Community Support. The browser window title is 'pgAdmin 4' and the address bar shows '127.0.0.1:5050/browser/'.

Aulas Práticas – Ferramenta



Senha: sem senha

Aulas Práticas – Ferramenta



The screenshot displays the pgAdmin 4 web interface in a browser window. The left sidebar shows the database structure: Servers (1) > PostgreSQL 10 > Databases (1) > postgres > Schemas (1) > public. The main panel shows the SQL editor for the 'postgres' database. The query editor contains the text: `SELECT * FROM AUTOR;`. A context menu is open over the query editor, showing options: Execute/Refresh (F5), Explain (F7), Explain Analyze (Shift+F7), Explain Options, Auto commit? (checked), and Auto rollback?. A red arrow points from a text box to the 'Execute/Refresh (F5)' option.

(F5) Executa todos os comandos SQL que estão no editor OU executa os comandos selecionados

Aulas Práticas – Ferramenta

- ◆ A ferramenta está instalada nos laboratórios do Bloco B



SQL – SEGURANÇA POSTGRESQL

Segurança do Banco de Dados

- ◆ Os SGBD incluem recursos de segurança que controlam o modo como um banco de dados é acessado e usado

- ◆ Exemplo:

- No PostgreSQL

- ◆ Papéis
 - ◆ Privilégios
 - ◆ Visões

Mecanismos de Segurança

◆ Papéis (Roles)

- Modo como o PostgreSQL gerencia permissões de acesso
- Podem corresponder a usuários individuais ou a grupos de usuários
- Podem possuir objetos e podem atribuir privilégios nesses objetos para outros papéis

Mecanismos de Segurança

◆ Papéis (Roles)

- Podem permitir que outros papéis se tornem "seus membros" concedendo o direito de uso de seus privilégios para os seus membros
- Existem papéis predefinidos
 - ◆ Exemplo: postgres

O comando CREATE ROLE

◆ Permite criar novos papéis

- NOTA: É necessário ter o privilégio CREATEROLE ou ser um super usuário para usar esse comando.

◆ Sintaxe simplificada

```
CREATE ROLE role_name
```

O comando CREATE ROLE

◆ Detalhamento...

CREATE ROLE name [[WITH] option [...]]

■ Exemplos de opções:

◆ SUPERUSER | NOSUPERUSER

- Determina se o novo papel é ou não um super usuário

◆ CREATEROLE | NOCREATEROLE

- Determina se o novo papel poderá ou não criar novos papéis

◆ INHERIT | NOINHERIT

- Determina se o novo papel herda ou não os privilégios dos papéis dos quais é membro

O comando CREATE ROLE

◆ Detalhamento...

CREATE ROLE name [[WITH] option [...]]

- Exemplos de opções:

- ◆ LOGIN | NOLOGIN

- Determina se o novo papel tem permissão de login e nesse caso pode ser visto como um usuário
 - Papéis sem esse atributo são úteis para gerenciar os privilégios em um banco de dados

- ◆ [ENCRYPTED | UNENCRYPTED] PASSWORD 'password'

- Maiores detalhes:

<https://www.postgresql.org/docs/current/sql-createrole.html>

Mecanismos de Segurança

◆ Privilégios

- Quando um objeto é criado ele é atribuído a um dono
- Esse dono é normalmente o papel que executou o comando de criação
- Para que outros papéis possam usar o objeto é preciso conceder **privilégios**

Mecanismos de Segurança

◆ Exemplos de tipos de Privilégios

- SELECT (table, view ou sequence)
- INSERT
- UPDATE (pode requerer o privilégio SELECT também)
- DELETE (pode requerer o privilégio SELECT também)
- RULE
- REFERENCES (para a criação de chaves estrangeiras)
- TRIGGER
- CREATE
- EXECUTE
- Sintaxe completa em:
<https://www.postgresql.org/docs/current/sql-grant.html>

Os comandos GRANT e REVOKE

- ◆ GRANT : Permite atribuir privilégios
- ◆ REVOKE : Permite revogar privilégios
- ◆ Sintaxe simplificada para atribuir privilégios

```
GRANT privilege_name  
    ON db_object  
    TO role  
    [WITH GRANT OPTION]
```

- WITH GRANT OPTION: permite que o papel que recebe o privilégio possa atribuí-lo a outros papéis

Os comandos GRANT e REVOKE

- ◆ A palavra reservada `PUBLIC` pode ser usada para conceder um privilégio para todos os papéis do sistema
 - Exemplo: `GRANT UPDATE ON ALUNO TO PUBLIC;`
- ◆ Se substituirmos o nome do privilégio pela palavra reservada `ALL` isso indica que todos os privilégios que se aplicam ao objeto serão concedidos

Mecanismos de Segurança

◆ Associação de Papéis

- Criação de papéis que representam um grupo
- Vantagens:
 - ◆ Concessão reduzida de privilégios
 - Por grupos de usuários
 - ◆ Gerenciamento dinâmico de privilégios
 - Se for necessário alterar privilégios de um grupo de usuários, altera-se apenas no papel
 - ◆ Disponibilidade seletiva de privilégios
 - Permite o controle específico dos privilégios de um usuário em determinada situação

Os comandos GRANT e REVOKE

◆ Associação de Papéis

- A inclusão ou remoção de membros é feita utilizando os comandos GRANT e REVOKE
- Exemplo:
 - ◆ `CREATE ROLE group_role;`
 - ◆ `GRANT group_role TO role1, ... ;`
 - ◆ `REVOKE group_role FROM role1, ... ;`

Os comandos GRANT e REVOKE

◆ Associação de Papéis

- Os membros de um papel podem usar os privilégios do papel de duas maneiras
 - ◆ SET ROLE
 - Para se “tornarem” temporariamente o papel
 - ◆ Os membros do papel que tiverem o atributo INHERIT tem automaticamente o uso dos privilégios do papel

Comandos adicionais

- ◆ Para excluir um papel

```
DROP ROLE role_name
```

- ◆ Para visualizar no dicionário de dados

- Papéis, selecione a tabela

- ◆ pg_roles

Bibliografia

- ◆ Elmasri, Ramez; Navathe, Shamkant B. **Sistemas de banco de dados**. 4 ed. São Paulo: Addison Wesley, 2005, 724 p. Bibliografia: p. [690]-714.
- ◆ Manual do SGBD PostgreSQL