Projeto de Banco de Dados

Segurança e Direitos de Acesso no PostgreSQL

PROF. DR. THIAGO ELIAS

Usuários de BD

- Um agrupamento de BD possui um conjunto de usuários.
- Eles possuem objetos de banco de dados e podem conceder privilégios nestes objetos para outros usuários.
- DCL: Linguagem de controle de dados. Controla aspectos de autorização de dados e licenças de usuários.

Criando um Usuário e/ou Regra/Papel

- Usuários X Grupos X *Roles* (papéis)
- No PostgreSQL é possível criar usuários, e também criar papéis (ROLEs) de acessos que podem ser atribuídos a um ou mais usuários, definindo regras e até grupos de acesso.
- O que irá diferenciar um usuário de um papel (ROLE) é o fato de que um usuário necessita de uma senha para autenticar-se no sistema, enquanto um papel, não.
- A este último, apenas define-se acessos a um grupo de usuários que já tenha suas respectivas senhas.

Criando um Usuário e/ou Regra/Papel

- Sintaxe:
  - CREATE USER nome [[WITH] opções […]]
- Obs: O comando CREATE USER é um *alias* para o comando CREATE ROLE.
- As *opções* podem ser:
  - SUPERUSER / NOSUPERUSER
  - CREATEDB / NOCREATEDB
  - CREATEROLE / NOCREATEROLE
  - LOGIN
  - PASSWORD
  - VALID UNTIL ‘tempo’ (o formato é timestamp: ‘yyyy-mm-dd hh:mm:ss’)
  - IN ROLE
  - IN GROUP

Removendo um Usuário do BD

- Uma vez criado no BD, o comando DROP USER remove o usuário especificado.
- O comando não remove os objetos pertencentes ao usuário.
- Se o usuário possuir algum BD, uma mensagem de erro será gerada.
- Sintaxe:
  - DROP USER nome_usuário

Alterando um Usuário do BD

- Em alguns casos, é interessante modificar um usuário, para alterar a sua senha ou incluir ou excluir, por exemplo, uma permissão de criação de usuários.
- Sintaxe:
  - ALTER USER nome [[WITH] opções [...]]
- É possível alterar o nome de um usuário. Apenas um superusuário pode alterar o nome de um outro usuário. A sintaxe é:
  - ALTER USER nome RENAME TO novo_nome

Grupos de Usuários no PostgreSQL

- O conceito de grupos não existe no SQL padrão.
- Os grupos são uma forma lógica de juntar usuários para facilitar o gerenciamento de privilégios.
- Tais privilégios podem ser concedidos ou revogados para o grupo como um todo.
- Sintaxe:
  - CREATE GROUP nome_grupo

Grupos de Usuários no PostgreSQL

- Podemos adicionar ou remover usuários em um grupo existente utilizando o comando ALTER GROUP.
- Sintaxe:
  - ALTER GROUP nome_grupo ADD USER nome_usuário
  - ALTER GROUP nome_grupo DROP USER nome_usuário
- Também podemos renomear um grupo.
  - ALTER GROUP nome_grupo RENAME TO novo_nome

Grupos de Usuários no PostgreSQL

- Uma vez criado o grupo, podemos removê-lo.
- Sintaxe:
  - DROP GROUP nome_grupo
- O comando DROP GROUP não exclui os usuários membros do grupo.
- Obs: O conceito de grupos é o mesmo de *papéis*, que abordaremos em seguida.

Criando Papéis (ROLES) no PostgreSQL

- O comando CREATE ROLE adiciona um novo papel ao BD.
- O papel é uma entidade que pode possuir objetos do BD e possuir privilégios do BD.
- Ele pode ser considerado um “usuário”, um “grupo” ou ambos, dependendo de como é utilizado.
- O comando CREATE ROLE substitui os comandos CREATE USER e CREATE GROUP por possuir mais recursos que os mesmos.
- Sintaxe:
  - CREATE ROLE nome [[WITH] opções [...]]

Removendo um Role no PostgreSQL

- O comando DROP ROLE remove os papeis especificados.
- Sintaxe:
  - DROP ROLE [IF EXISTS] nome
- O papel não poderá ser removido se ainda estiver sendo referenciado em qualquer BD.
- Antes de remover o papel, deve-se remover todos os objetos pertencentes ao mesmo (ou mudar de dono) e revogar todos os privilégios concedidos pelo papel.

Alterando um Role no PostgreSQL

- O comando ALTER ROLE altera os atributos de um papel do PostgreSQL.
- Sintaxe:
  - ALTER ROLE nome [[WITH] opção]

Concedendo e Revogando Privilégios

- Quando um objeto do BD é criado, é atribuído um dono ao mesmo.
- O dono é o usuário que o criou.
- Para mudar o dono, por exemplo, de uma tabela deve ser utilizado o comando ALTER TABLE. Sintaxe:
  - ALTER TABLE nome_tabela OWNER TO novo-dono
- Por padrão, somente o dono (ou superusuário) pode fazer qualquer coisa com o objeto.
- Para permitir o uso por outros usuários, devem ser concedidos privilégios aos mesmos.

Concedendo e Revogando Privilégios

- O comando GRANT é responsável por conceder privilégios aos objetos do BD.
- Ele também concede privilégio de ser membro de um papel.
- Sintaxe:
  - GRANT privilégio ON objeto TO papel [WITH GRANT OPTION]
- Exemplos de privilégios:
  - Select, Insert, Update, Delete, Rule, Trigger, Create, Execute ou ALL PRIVILEGES

Concedendo e Revogando Privilégios

- Em alguns casos também se torna necessário revogar alguns privilégios.
- Para isso, utiliza-se o comando REVOKE.
- Exemplo:
  - REVOKE ALL ON FUNCTION teste(int, int) FROM PUBLIC

EXERCÍCIO

- Crie o BD de um hotel com as tabelas:
  - Apartamento(num_apt, status)
  - Hospede(cod_hosp, nome, idade)
  - Hospedagem(cod_hospedagem, cod_hosp, num_apt, data_ent, data_sai)
  - Obs: o código da hospedagem deve ser *serial*.
- Crie também uma visão que mostra apenas o nome e a idade dos hóspedes.

EXERCÍCIO

- Crie uma função que realiza hospedagem, verificando se o hóspede existe e se o apartamento está desocupado. A função deve receber o código do hóspede e o número do apartamento. Deve ser atribuída a data do sistema para a data de entrada e a data de saída deve ficar em branco. Não esqueça de alterar o status do apartamento.
- Crie também uma função que finaliza uma hospedagem, preenchendo a data de saída e alterando o status do apartamento para desocupado. Ela deverá receber apenas o código da hospedagem.

EXERCÍCIO

- Crie 3 papeis: Gerente, Atendente e Estagiário.
  - O gerente poderá visualizar e inserir todas as tabelas, visualizar a visão, executar as funções e conceder permissões a outros usuários.
  - O atendente poderá apenas executar as funções.
  - O estagiário poderá apenas visualizar o nome e a idade dos hóspedes.
- Crie um usuário com a respectiva senha para cada um dos papéis.
- Realize testes.