|  |  |
| --- | --- |
| LogoEfeitrans | **Banco de Dados II Prática 1**  Profa. Vanessa Souza |

**Assunto:** Segurança – Aula Prática

Essa prática deve ser executada no SGBD PostgreSQL. REFERÊNCIAS DO POSTGRESQL:

* ROLES : <https://www.postgresql.org/docs/12/user-manag.html>
* GRANT : <https://www.postgresql.org/docs/12/sql-grant.html>
* REVOKE : <https://www.postgresql.org/docs/12/sql-revoke.html>

/\*Seguindo as boas práticas de segurança, você deve criar perfis de usuário

no banco e posteriormente adicionar usuários a esses perfis.

**Questão 1:** Restaure o banco de dados Northwind.

**Questão 2:** Seguindo as boas práticas de segurança, você deve criar perfis de usuário no banco e posteriormente adicionar usuários a esses perfis.

1. Crie uma role chamada ‘programadores’, com o privilégio de CRUD em todas as

tabelas do banco.

CREATE ROLE programadores;

**GRANT** INSERT, SELECT, UPDATE, DELETE **ON ALL TABLES IN SCHEMA** northwind TO programadores;

1. Crie uma role chamada ‘gerente’. Daremos privilégios para essa role nas próximas questões.

CREATE ROLE gerente;

**Questão 3 :** Criação de usuário no banco de dados.

1. Crie um usuário com seu nome no banco de dados Northwind

CREATE USER ivan WITH PASSWORD '12345';

1. Faça *login* com esse novo usuário

Database: Northwind / User e role: ivan

1. Simule falhas de conexão – erro de senha

Com senha erradas ou user e role trocadas nao permite o acesso

1. Verifique os privilégios do usuário

SELECT \* FROM northwind.categories;

/\*ivan não tem acesso ao schema ainda, apenas login\*/

**Questão 4:** Concedendo privilégios ao usuário.

1. Considere que o usuário que você criou na questão 3 é um usuário do tipo

‘programador’. Insira ele na role criada na questão 2.

GRANT programadores TO ivan;

/\*ivan é atrelado a role/grupo programadores, tendo os direitos atribuidos

por meio de programadores: crud sobre os registros das tabelas\*/

1. Verifique o que é possível realizar no banco

AINDA SO PERMITE A CONEXÃO/LOGIN/AUTENTICAÇÃO

**GRANT USAGE ON SCHEMA** northwind **TO** programadores;

/\*ivan terá como usar o schema atraves de programdores \*/

CONSEGUIDO REALIZAR O CRUD NO BANCO AGORA.

SELECT \* FROM northwind.categories; /\* permitido \*/

INSERT INTO northwind.categories(categoryid, categoryname, description) VALUES (200, 'teste', 'testando privilegios');/\* permitido \*/

UPDATE northwind.categories SET categoryname = 'teste 200' WHERE categoryid = 200; /\* permitido \*/

DELETE FROM northwind.categories WHERE categoryid = 200;/\* permitido \*/

CREATE TABLE northwind.teste (a int primary key); /\* ivan nao tem direito de criar tabela \*/

DROP TABLE northwind.teste; /\* ivan nao tem direito de deletar a tabela \*/

1. Simule falhas associadas a falta de privilégios por esse usuário.

AGORA DA PRA ACESSAR COMO USER: IVAN E ROLE PROGRAMDOR TAMBEM

**Questão 5:** Removendo privilégios do usuário.

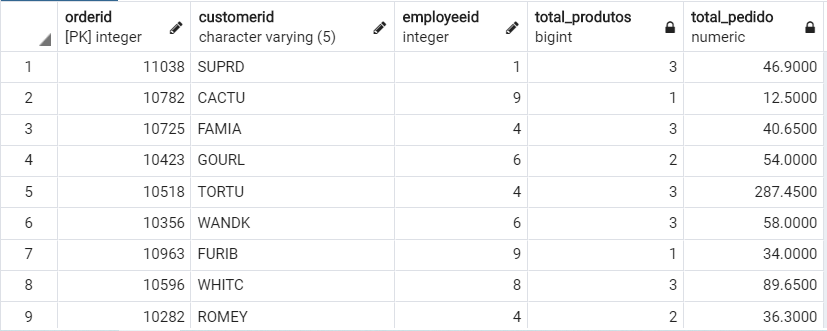
1. Remova da role ‘programadores’ o privilégio de realizar ‘delete’ na tabela ‘Categories’.

**REVOKE DELETE ON TABLE** northwind.categories **FROM** programadores;/\*ivan não terá mais direito de deletar registros da tabela categoria\*/

1. Teste a remoção do privilégio.

DELETE FROM northwind.categories WHERE categoryid = 200;/\* não permitido - sem direito de excluir registros da tabela\*/

**Questão 6 :** É possível conceder privilégios sobre *Views*. A associação dos comandos GRANT e VIEW permite que se limite o acesso de registros a usuários (e não apenas a nível de coluna, como no GRANT). É possível também permitir que usuários tenham acesso apenas a relatórios. Testaremos essa funcionalidade nessa questão.

1. Crie uma *view* chamada ‘relatorio’, sobre as tabelas *Orders e OrderDetails*, cujo resultado seja idêntico à figura 1. Onde:
   * Total\_produtos é o total de produtos comprados no pedido
   * Total\_pedido é a soma dos valores de todos os produtos comprados

/\*É possível conceder privilégios sobre Views. A associação dos comandos GRANT e VIEW permite que se limite o acesso de registros a usuários (e não apenas a nível de coluna, como no GRANT). É possível também permitir que usuários tenham acesso apenas a relatórios.

Crie uma view chamada ‘relatorio’, sobre as tabelas Orders e OrderDetails onde: Total\_produtos é o total de produtos comprados no pedido e Total\_pedido é a soma dos valores de todos os produtos comprados.\*/

SELECT \* FROM northwind.orders

SELECT \* FROM northwind.order\_details where orderid = 11038

CREATE VIEW relatorio AS (SELECT o.orderid, o.customerid, o.employeeid, count(od.orderid) as total\_produtos, sum(od.unitprice) as total\_pedido

FROM northwind.orders o, northwind.order\_details od

WHERE o.orderid = od.orderid

GROUP BY (o.orderid));

SELECT \* FROM relatorio;

1. Conceda o privilégio de leitura sobre a View para a role ‘gerente’.

**GRANT SELECT ON** relatorio **TO** gerente;

1. Crie o usuário ‘gestor’ e adicione ele na role

/\*Crie o usuário ‘gestor’ e adicione ele na role\*/

CREATE USER gestor WITH PASSWORD '123'; /\*permite o login pelo gestor\*/

GRANT gerente TO gestor; /\*coloca o gestor no grupo gerente\*/

GRANT USAGE ON SCHEMA northwind TO gerente;

/\* Permiteao grupo gerente o uso do banco\*/

1. Teste os privilégios do usuário gestor no banco de dados.

SELECT \* FROM northwind.orders;

/\* sem direito de leitura dos registros da tabela\*/

SELECT \* FROM northwind.order\_details;

/\* sem direito de leitura dos registros da tabela\*/

SELECT \* FROM relatorio; /\*leitura realizadacom sucesso\*/

**Questão 7:** O banco de dados PostgreSQL é do tipo ‘objeto-relacional’. Isso significa que ele implementa conceitos da orientação objeto em seu modelo. Dessa forma, tabelas, roles, usuários, são objetos no banco e possuem ‘um dono’. Ou seja, quem criou aquele ‘objeto’ tem direitos sobre ele. No caso dos comandos SQL GRANT e REVOKE, isso tem diversas implicações. Vamos testá-las nessa atividade.

\* Comandos para verificar as roles e seus privilégios no banco\*/

SELECT \* FROM pg\_roles where rolname = 'gerente';

SELECT \* FROM information\_schema.role\_table\_grants where grantee='gerente';

1. Crie uma role, cujo perfil é de ‘DBA de Banco de Dados’. Ou seja, ela terá todos

os privilégios sobre o banco Northwind.

CREATE ROLE dba\_northwind;

GRANT ALL PRIVILEGES ON DATABASE northwind TO dba\_northwind;

/\* concede todos os privilegios sobre o banco\*/

GRANT ALL ON ALL TABLES IN SCHEMA northwind TO dba\_northwind;

/\* concede todos os privilegios sobre as tabelas\*/

GRANT USAGE ON SCHEMA northwind TO dba\_northwind; /\*"superusuário"\*/

1. Crie um usuário e associe ele a esse perfil.

CREATE USER dba WITH PASSWORD '123';

GRANT dba\_northwind TO dba;

1. Faça login no banco com o perfil criado e execute a seguinte operação no banco: inserir uma coluna na tabela ‘categories’.

select \* from northwind.categories;

ALTER TABLE northwind.categories ADD COLUMN teste int default 0;/\*erro\*/

1. Qual o resultado da operação executada na letra c?

erro, deve ser o proprietário das categorias da tabela\*/

**ALTER TABLE** northwind.categories **OWNER TO** dba; /\*dba passa a ter direito de proprietário da tabela\*/

ALTER TABLE northwind.categories ADD COLUMN teste2 int default 0; /\*adiciona tabela\*/

SELECT \* FROM northwind.categories;

1. Ainda logado como o usuário criado na letra b (que é um DBA), remova da role

‘programadores’ o privilégio de realizar ‘delete’ na tabela ‘Orders.

REVOKE DELETE ON TABLE northwind.orders FROM programadores;

REVOKE DELETE ON TABLE northwind.categories FROM programadores;/\*realizado\*/

1. Qual o resultado da operação executada na letra e?

Não consegue remover privilegio da tabela orders apenas de categories

**select \* from information\_schema.role\_table\_grants where grantee=**'programadores';

/\* mostra programador com todos direito exceto sem direito de delete sobre a tabela categories\*/

**select tablename, tableowner From pg\_tables**

/\*mostra dba como proprietario decategories\*/

**/\* Exclusão da role\*/**

DROP ROLE programadores;

/\* Como existe privelegios atrelados a ivan, não será possivel excluir.

devera ser retirados todos direitos atribuidos pela role. \*/

REVOKE ALL PRIVILEGES ON schema northwind FROM programadores;

/\* remove direito de uso sobre o schema para programadores\*/

REVOKE ALL PRIVILEGES ON ALL TABLES IN schema northwind FROM programadores;

/\* remove todos os direitos de todas as tabelas \*/

DROP ROLE programadores;

/\* Apos a remoção de todos os direitos é excluida a role\*/

8- Testando o ‘WITH GRANT OPTION’.

1. Crie um novo usuario no banco chamado ‘user1’ e dê a ele privilégios de CRUD nas tabelas categories, customers e products. Utilize o ‘with grant option’.

Lembrem-se que essa não é uma boa prática de segurança!

CREATE USER user1 WITH PASSWORD '12345';

GRANT INSERT, SELECT, UPDATE, DELETE ON TABLE northwind.categories, northwind.customers, northwind.products TO user1 WITH GRANT OPTION;

GRANT USAGE ON SCHEMA northwind TO user1;

1. Crie um novo usuario no banco chamado ‘user2’ e dê a ele privilégios de CRUD nas tabelas orders e orderdetails. Utilize o ‘with grant option’.

CREATE USER user2 WITH PASSWORD '12345';

GRANT INSERT, SELECT, UPDATE, DELETE ON TABLE northwind.categories, northwind.orders, northwind.order\_details TO user2 WITH GRANT OPTION;

GRANT USAGE ON SCHEMA northwind TO user2;

1. Crie um novo usuario no banco chamado ‘user3’ e dê a ele privilégios de

SELECT em todas as tabelas do schema northwind.

CREATE USER user3 WITH PASSWORD '12345';

GRANT SELECT ON ALL TABLES IN SCHEMA northwind TO user3;

GRANT USAGE ON SCHEMA northwind TO user3;

1. Agora, o user1 vai repassar os privilégios dele para o user3.

GRANT INSERT, SELECT, UPDATE, DELETE ON TABLE northwind.categories, northwind.customers, northwind.products TO user3 WITH GRANT OPTION;

1. De forma análoga, o user2 repassará os privilégios dele para o user3.

GRANT INSERT, SELECT, UPDATE, DELETE ON TABLE northwind.categories, northwind.orders, northwind.order\_details TO user3 WITH GRANT OPTION;

1. Qual o efeito das operações executadas nas letras d e e no banco?

user 3 passa a ter mais direitos do que o de leitura concedidos inicialmente.

g) Logado como postgres, remova os privilégios no user3. O que acontece? Como resolver?

REVOKE ALL PRIVILEGES ON ALL TABLES IN SCHEMA northwind FROM user3;

REVOKE ALL PRIVILEGES ON schema northwind FROM user3;

remove todos os privilegios acima concedidos pelo usuario postgres\*/

DROP ROLE user3; /\* não consegue excluir usuario 3, pois o mesmo apresenta

direios concedidos pelos usuarios 1 e 2 \*/

select \* from information\_schema.role\_table\_grants where grantee='user3' ORDER BY grantor;

**A única forma é excluido todos os provilegios concedidos a todos os usuarios que concederam ao user 3 de forma cascata\*/**

REVOKE ALL PRIVILEGES ON ALL TABLES IN SCHEMA northwind FROM user1 CASCADE;

REVOKE ALL PRIVILEGES ON schema northwind FROM user1 CASCADE;

REVOKE ALL PRIVILEGES ON ALL TABLES IN SCHEMA northwind FROM user2 CASCADE;

REVOKE ALL PRIVILEGES ON schema northwind FROM user2 CASCADE;

DROP ROLE user3;/\*user 3 excluido\*/

1. Crie um novo usuario no banco chamado ‘user2’ e dê a ele privilégios de CRUD nas tabelas orders e orderdetails. Utilize o ‘with grant option’.
2. Crie um novo usuario no banco chamado ‘user3’ e dê a ele privilégios de SELECT

em todas as tabelas do schema *northwind*.

1. Agora, o user1 vai repassar os privilégios dele para o user3.
2. De forma análoga, o user2 repassará os privilégios dele para o user3.
3. Qual o efeito das operações executadas nas letras d e e no banco?
4. Logado como postgres, remova os privilégios no user3. O que acontece? Como resolver?