ESCOLA SUPERIOR DE REDES RNP

Administração de Bancos de Dados

Aluno: **Diego Vitor Soares dos Santos**
Cod. Turma: **DES11**
Data: **13 de dezembro de 2025**

Módulo 4 – Administrando Usuário e Segurança

⚙ Ambiente de Teste

- **Virtualização:** Vagrant + VirtualBox
- **Sistema Operacional:** CentOS Stream 9 (Kernel 5.14.0-642.el9.x86_64)
- **PostgreSQL:** 17.2 (estável)
- **Fonte oficial:** https://ftp.postgresql.org/pub/source/v17.2/postgresql-17.2.tar.gz

---

a) Criação de Roles

Objetivo

1. Criar as seguintes ROLES com permissão de conexão (`LOGIN`) e senha definida:
   - `gerente`
   - `controller`
   - `jsilva`
   - `moliveira`
   - `psouza`
   - `contábil`
2. **Restrição Temporal:** Definir a role **psouza** como válida por apenas **1 mês**.
3. **Permissões Elevadas:** Permitir que a role **gerente** possa:
   - Criar bases de dados (`CREATEDB`).
   - Criar outras roles (`CREATEROLE`).

Comandos Executados

```
-- Criação das roles com LOGIN e senha
CREATE ROLE gerente LOGIN PASSWORD 'atividade4.des11';
CREATE ROLE controller LOGIN PASSWORD 'atividade4.des11';
CREATE ROLE jsilva LOGIN PASSWORD 'atividade4.des11';
CREATE ROLE moliveira LOGIN PASSWORD 'atividade4.des11';
CREATE ROLE psouza LOGIN PASSWORD 'atividade4.des11';
CREATE ROLE "contábil" LOGIN PASSWORD 'atividade4.des11';

-- Definir validade de 1 mês para psouza
\set validade `date -d "+1 month" +"%Y-%m-%d"`
ALTER ROLE psouza VALID UNTIL :'validade';

-- Conceder privilégios administrativos ao gerente
ALTER ROLE gerente CREATEDB CREATEROLE;
```

Resultado da Execução

```
[postgres@vm02 vagrant]$ psql
psql (17.2)
Type "help" for help.

postgres=# \du
                                   List of roles
 Role name |                         Attributes
-----------+------------------------------------------------------------
```

```
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS

postgres=# CREATE ROLE gerente LOGIN PASSWORD 'atividade4.des11';
CREATE ROLE
postgres=# CREATE ROLE controller LOGIN PASSWORD 'atividade4.des11';
CREATE ROLE
postgres=# CREATE ROLE jsilva LOGIN PASSWORD 'atividade4.des11';
CREATE ROLE
postgres=# CREATE ROLE moliveira LOGIN PASSWORD 'atividade4.des11';
CREATE ROLE
postgres=# CREATE ROLE psouza LOGIN PASSWORD 'atividade4.des11';
CREATE ROLE
postgres=# CREATE ROLE "contábil" LOGIN PASSWORD 'atividade4.des11';
CREATE ROLE
postgres=# \du
                                   List of roles
 Role name  |                         Attributes
------------+------------------------------------------------------------
 controller |
 contábil   |
 gerente    |
 jsilva     |
 moliveira  |
 postgres   | Superuser, Create role, Create DB, Replication, Bypass RLS
 psouza     |

postgres=# exit
[postgres@vm02 vagrant]$ psql -U gerente -d postgres
psql (17.2)
Type "help" for help.

postgres=> SELECT CURRENT_USER;
 current_user
--------------
 gerente
(1 row)

postgres=> SELECT rolname, rolcanlogin
FROM pg_roles
WHERE rolname IN ('gerente','controller','jsilva','moliveira','psouza','contábil');
  rolname   | rolcanlogin
------------+-------------
 gerente    | t
 controller | t
 jsilva     | t
 moliveira  | t
 psouza     | t
 contábil   | t
(6 rows)

postgres=> SELECT NOW();
              now
-------------------------------
 2025-12-13 14:01:21.842454+00
(1 row)

postgres=>
```

```
[postgres@vm02 vagrant]$ psql
psql (17.2)
Type "help" for help.

postgres=# \du
                                   List of roles
 Role name  |                         Attributes
------------+------------------------------------------------------------
 controller |
 contábil   |
 gerente    |
 jsilva     |
 moliveira  |
```

```
 postgres   | Superuser, Create role, Create DB, Replication, Bypass RLS
 psouza     |

postgres=# \set validade `date -d "+1 month" +"%Y-%m-%d"`
postgres=# ALTER ROLE psouza VALID UNTIL :'validade';
ALTER ROLE
postgres=# \du
                                List of roles
 Role name  |                         Attributes
------------+------------------------------------------------------------
 controller |
 contábil   |
 gerente    |
 jsilva     |
 moliveira  |
 postgres   | Superuser, Create role, Create DB, Replication, Bypass RLS
 psouza     | Password valid until 2026-01-13 00:00:00+00
postgres=# SELECT rolname, rolvaliduntil
FROM pg_authid
WHERE rolname = 'psouza';
 rolname |     rolvaliduntil
---------+------------------------
 psouza  | 2026-01-13 00:00:00+00
(1 row)

postgres=#
```

```
postgres=# \du
                                List of roles
 Role name  |                         Attributes
------------+------------------------------------------------------------
 controller |
 contábil   |
 gerente    |
 jsilva     |
 moliveira  |
 postgres   | Superuser, Create role, Create DB, Replication, Bypass RLS
 psouza     | Password valid until 2026-01-13 00:00:00+00

postgres=# ALTER ROLE gerente CREATEDB CREATEROLE;
ALTER ROLE
postgres=# SELECT rolname, rolcreatedb, rolcreaterole
FROM pg_roles
WHERE rolname = 'gerente';
 rolname | rolcreatedb | rolcreaterole
---------+-------------+---------------
 gerente | t           | t
(1 row)

postgres=# \du
                                List of roles
 Role name  |                         Attributes
------------+------------------------------------------------------------
 controller |
 contábil   |
 gerente    | Create role, Create DB
 jsilva     |
 moliveira  |
 postgres   | Superuser, Create role, Create DB, Replication, Bypass RLS
 psouza     | Password valid until 2026-01-13 00:00:00+00
```

---

b) Criação de Grupo e Associação de Membros

Objetivos

1. Conectar com o usuário **gerente**.
2. Criar a **ROLE** para o grupo **contabilidade**.

3. Adicionar as seguintes roles ao grupo **contabilidade**:
    - **jsilva**
    - **moliveira**
    - **psouza**

Comandos Executados

```
-- Criar role de grupo sem permissão de login
CREATE ROLE contabilidade NOLOGIN;

-- Adicionar membros ao grupo
GRANT contabilidade TO jsilva, moliveira, psouza;
```

Resultado da Execução

```
[postgres@vm02 vagrant]$ psql -U gerente -d postgres
psql (17.2)
Type "help" for help.

postgres=> CREATE ROLE contabilidade NOLOGIN;
CREATE ROLE
postgres=> GRANT contabilidade TO jsilva, moliveira, psouza;
GRANT ROLE
postgres=> \du
                                   List of roles
   Role name   |                         Attributes
---------------+------------------------------------------------------------
 contabilidade | Cannot login
 controller    |
 contábil      |
 gerente       | Create role, Create DB
 jsilva        |
 moliveira     |
 postgres      | Superuser, Create role, Create DB, Replication, Bypass RLS
 psouza        | Password valid until 2026-01-13 00:00:00+00

postgres=> SELECT r.rolname AS nome_grupo,
       ARRAY(SELECT u.rolname
             FROM pg_auth_members m
             JOIN pg_roles u ON m.member = u.oid
             WHERE m.roleid = r.oid) as membros
FROM pg_roles r
WHERE r.rolname = 'contabilidade';
  nome_grupo   |              membros
---------------+-----------------------------------
 contabilidade | {gerente,jsilva,moliveira,psouza}
(1 row)

postgres=>
```

---

c) Criação de Base de Dados e Schema

Objetivo

1. Conectar com o usuário **gerente**.
2. Criar a base de dados **sis_contabil**.
3. Dentro da base, criar o schema **controladoria**.
4. Fornecer permissão para a role controller no schema controladoria poder criar objetos e usar o schema.

Comandos Executados

```
-- Criar base de dados
CREATE DATABASE sis_contabil;

-- Conectar à nova base
```

```
\c sis_contabil

-- Criar schema
CREATE SCHEMA controladoria;

-- Conceder permissões ao controller
GRANT USAGE ON SCHEMA controladoria TO controller;
GRANT CREATE ON SCHEMA controladoria TO controller;
```

Resultado da Execução

```
postgres=> SELECT CURRENT_USER;
 current_user
--------------
 gerente
(1 row)

postgres=> CREATE DATABASE sis_contabil;
CREATE DATABASE
postgres=> SELECT datname
FROM pg_database;
   datname
--------------
 postgres
 template1
 template0
 sis_contabil
(4 rows)

postgres=> \c sis_contabil
You are now connected to database "sis_contabil" as user "gerente".
sis_contabil=> CREATE SCHEMA controladoria;
CREATE SCHEMA
sis_contabil=> \dn
          List of schemas
     Name      |       Owner
---------------+--------------------
 controladoria | gerente
 public        | pg_database_owner
(2 rows)

sis_contabil=> GRANT USAGE ON SCHEMA controladoria TO controller;
GRANT
sis_contabil=> GRANT CREATE ON SCHEMA controladoria TO controller;
GRANT
sis_contabil=> \du+
                                    List of roles
   Role name   |                         Attributes                         | Description
---------------+------------------------------------------------------------+-------------
 contabilidade | Cannot login                                               |
 controller    |                                                            |
 contábil      |                                                            |
 gerente       | Create role, Create DB                                     |
 jsilva        |                                                            |
 moliveira     |                                                            |
 postgres      | Superuser, Create role, Create DB, Replication, Bypass RLS |
 psouza        | Password valid until 2026-01-13 00:00:00+00                |

sis_contabil=>
```

---

d) Configuração de Permissões na Tabela

Objetivo

1. **Conexão com o usuário gerente**
   - Conceder permissão ao grupo **contabilidade** para utilizar o *schema* **controladoria**.
2. **Conexão com o usuário controller**

- Criar a tabela **contas** no *schema* **controladoria** com o seguinte código SQL:

```
    CREATE TABLE controladoria.contas (
        id INT,
        numero INT,
        responsavel VARCHAR(50)
    );
```

3. **Concessão de permissões**
   - Permissão de uso do *schema* **controladoria** para o grupo **contabilidade**.
   - Permissão de **consulta** à tabela **contas** para o grupo **contabilidade**.
   - Permissão de **atualização** da coluna `numero` da tabela **contas** para a *role* **jsilva**.
   - Permissão de **inserção e exclusão** na tabela **contas**, com possibilidade de repassar o privilégio, para a *role* **moliveira**.

Comandos Executados

```
-- Como gerente: conceder USAGE ao grupo contabilidade
GRANT USAGE ON SCHEMA controladoria TO contabilidade;
```

```
-- Como controller: criar tabela
CREATE TABLE controladoria.contas (
    id INT,
    numero INT,
    responsavel VARCHAR(50)
);
```

```
-- Como gerente: conceder permissões específicas
GRANT SELECT ON controladoria.contas TO contabilidade;
GRANT UPDATE (numero) ON controladoria.contas TO jsilva;
GRANT INSERT, DELETE ON controladoria.contas TO moliveira WITH GRANT OPTION;
```

Resultado da Execução

```
[postgres@vm02 vagrant]$ psql -U gerente -d sis_contabil
psql (17.2)
Type "help" for help.

sis_contabil=> \dn+ controladoria
                        List of schemas
     Name      |  Owner  |   Access privileges   | Description
---------------+---------+-----------------------+-------------
 controladoria | gerente | gerente=UC/gerente   +|
               |         | controller=UC/gerente |
(1 row)

sis_contabil=> GRANT USAGE ON SCHEMA controladoria TO contabilidade;
GRANT
sis_contabil=> \dn+ controladoria
                           List of schemas
     Name      |  Owner  |    Access privileges    | Description
---------------+---------+-------------------------+-------------
 controladoria | gerente | gerente=UC/gerente     +|
               |         | controller=UC/gerente  +|
               |         | contabilidade=U/gerente |
(1 row)
```

```
[postgres@vm02 vagrant]$ psql -U controller -d sis_contabil
psql (17.2)
Type "help" for help.

sis_contabil=> CREATE TABLE controladoria.contas (
    id INT,
    numero INT,
    responsavel VARCHAR(50)
```

```
);
CREATE TABLE
sis_contabil=> \dt controladoria.*
              List of relations
    Schema     |  Name  | Type  |   Owner
---------------+--------+-------+------------
 controladoria | contas | table | controller
(1 row)

sis_contabil=>
```

```
sis_contabil=> GRANT USAGE ON SCHEMA controladoria TO GROUP contabilidade;
WARNING:  no privileges were granted for "controladoria"
GRANT
sis_contabil=> \dn+ controladoria
                          List of schemas
     Name      |  Owner  |    Access privileges    | Description
---------------+---------+-------------------------+-------------
 controladoria | gerente | gerente=UC/gerente     +|
               |         | controller=UC/gerente  +|
               |         | contabilidade=U/gerente |
(1 row)

sis_contabil=> GRANT SELECT ON controladoria.contas TO GROUP contabilidade;
GRANT
sis_contabil=> GRANT UPDATE (numero) ON controladoria.contas TO jsilva;
GRANT
sis_contabil=> GRANT INSERT, DELETE ON controladoria.contas TO moliveira WITH GRANT OPTION;
GRANT
sis_contabil=> \dn+ controladoria
                          List of schemas
     Name      |  Owner  |    Access privileges    | Description
---------------+---------+-------------------------+-------------
 controladoria | gerente | gerente=UC/gerente     +|
               |         | controller=UC/gerente  +|
               |         | contabilidade=U/gerente |
(1 row)
sis_contabil=> SELECT nspname AS schema,
       nspowner::regrole AS owner,
       nspacl
FROM pg_namespace
WHERE nspname = 'controladoria';
    schema     |  owner  |                                nspacl
---------------+---------+-----------------------------------------------------------------------
 controladoria | gerente | {gerente=UC/gerente,controller=UC/gerente,contabilidade=U/gerente}
(1 row)
sis_contabil=> SELECT grantee, privilege_type
FROM information_schema.role_table_grants
WHERE table_schema = 'controladoria'
  AND table_name = 'contas';
   grantee    | privilege_type
--------------+----------------
 controller   | INSERT
 controller   | SELECT
 controller   | UPDATE
 controller   | DELETE
 controller   | TRUNCATE
 controller   | REFERENCES
 controller   | TRIGGER
 moliveira    | INSERT
 moliveira    | DELETE
 contabilidade | SELECT
(10 rows)
```

e) Remoção de Usuário do Grupo

Objetivo

Remover o usuário `psouza` do grupo `contabilidade`.

Comando Executado

```
REVOKE contabilidade FROM psouza;
```

Resultado da Execução

```
[postgres@vm02 vagrant]$ psql -U gerente -d sis_contabil
psql (17.2)
Type "help" for help.

sis_contabil=> SELECT u.rolname AS usuario, r.rolname AS role_granted
FROM pg_roles r
JOIN pg_auth_members m ON r.oid = m.roleid
JOIN pg_roles u ON u.oid = m.member
WHERE r.rolname = 'contabilidade';
  usuario  | role_granted
-----------+--------------
 gerente   | contabilidade
 jsilva    | contabilidade
 moliveira | contabilidade
 psouza    | contabilidade
(4 rows)

sis_contabil=> REVOKE contabilidade FROM psouza;
REVOKE ROLE
sis_contabil=> SELECT u.rolname AS usuario, r.rolname AS role_granted
FROM pg_roles r
JOIN pg_auth_members m ON r.oid = m.roleid
JOIN pg_roles u ON u.oid = m.member
WHERE r.rolname = 'contabilidade';
  usuario  | role_granted
-----------+--------------
 gerente   | contabilidade
 jsilva    | contabilidade
 moliveira | contabilidade
(3 rows)
```

---

f) Configuração de Autenticação (pg_hba.conf)

Objetivo

1. **Configuração geral**
   - Definir autenticação por senha utilizando o método **MD5**.
2. **Regras específicas de acesso**
   - **Grupo contabilidade**
     - Base de dados: **sis_contabil**
     - Rede: **172.15.10.0/24**
     - Método: **md5**
   - **Usuário contabil**
     - Base de dados: **sis_contabil**
     - Servidor: **172.2.18.25/32**
     - Método: **md5**
   - **Usuário gerente**
     - Bases: **todas**
     - Rede: **2001:db8:3003::/48** (IPv6)
     - Método: **md5**

Regras Configuradas

```
# Grupo contabilidade - rede 172.15.10.0/24
host    sis_contabil    +contabilidade    172.15.10.0/24    md5
```

```
# Usuário contábil - servidor específico
host    sis_contabil    contábil          172.2.18.25/32    md5

# Usuário gerente - rede IPv6
host    all             gerente           2001:db8:3003::/48    md5
```

Resultado da Execução

Localização do arquivo:

```
[postgres@vm02 vagrant]$ psql -c "SHOW hba_file;"
       hba_file
----------------------
 /db/data/pg_hba.conf
(1 row)
```

```
[postgres@vm02 vagrant]$ cat >> /db/data/pg_hba.conf << 'EOF'

# Grupo contabilidade - rede 172.15.10.0/24
host    sis_contabil    +contabilidade    172.15.10.0/24    md5

# Usuário contábil - servidor específico
host    sis_contabil    contábil          172.2.18.25/32    md5

# Usuário gerente - rede IPv6
host    all             gerente           2001:db8:3003::/48    md5
EOF
[postgres@vm02 vagrant]$ tail -10 /db/data/pg_hba.conf
host all all 0.0.0.0/0 trust

# Grupo contabilidade - rede 172.15.10.0/24
host    sis_contabil    +contabilidade    172.15.10.0/24    md5

# Usuário contábil - servidor específico
host    sis_contabil    contábil          172.2.18.25/32    md5

# Usuário gerente - rede IPv6
host    all             gerente           2001:db8:3003::/48    md5
[postgres@vm02 vagrant]$ psql -c "SELECT pg_reload_conf();"
 pg_reload_conf
----------------
 t
(1 row)
```

---

g) Inserção de Dados e Row-Level Security (RLS)

Objetivo

1. **Conexão**
   - Conectar com um usuário que possua permissão de **inserir dados** na tabela `controladoria.contas` (conforme definido no item d).
2. **Inserção de registros**

```
INSERT INTO controladoria.contas(id, numero, responsavel) VALUES (1, 1000, 'jsilva');
INSERT INTO controladoria.contas(id, numero, responsavel) VALUES (2, 2000, 'psouza');
INSERT INTO controladoria.contas(id, numero, responsavel) VALUES (3, 2000, 'moliveira');
```

3. **Habilitar segurança por registro (Row-Level Security)**

```
ALTER TABLE controladoria.contas ENABLE ROW LEVEL SECURITY;
```

4. **Criar política de acesso por responsável**
   - Cada usuário só pode acessar os registros cujo campo `responsavel` corresponda ao seu nome de usuário:

```
CREATE POLICY contas_responsavel_policy
ON controladoria.contas
FOR ALL
USING (responsavel = current_user);
```

> Essa política garante que cada usuário veja apenas os registros onde ele é o responsável.

5. **Consulta com usuários específicos**
   - Conectar como `jsilva`, `psouza` ou `moliveira` e executar:

```
SELECT * FROM controladoria.contas;
```

   - O resultado mostrará apenas os registros cujo `responsavel` seja o usuário conectado.

6. **Consulta com superusuário**
   - Conectar como superusuário (ex.: `controller`) e executar:

```
SELECT * FROM controladoria.contas;
```

   - O superusuário visualizará **todos os registros**, independentemente da política de segurança.

Comandos Executados

```
-- Como moliveira: inserir dados
INSERT INTO controladoria.contas(id, numero, responsavel) VALUES (1, 1000, 'jsilva');
INSERT INTO controladoria.contas(id, numero, responsavel) VALUES (2, 2000, 'psouza');
INSERT INTO controladoria.contas(id, numero, responsavel) VALUES (3, 2000, 'moliveira');
```

```
-- Como controller: habilitar RLS e criar política
ALTER TABLE controladoria.contas ENABLE ROW LEVEL SECURITY;

CREATE POLICY contas_responsavel_policy
ON controladoria.contas
FOR ALL
USING (responsavel = current_user);
```

Resultado da Execução

```
[postgres@vm02 vagrant]$ psql -U gerente -d sis_contabil
psql (17.2)
Type "help" for help.

sis_contabil=> SELECT relname AS table_name,
       relacl
FROM pg_class
WHERE relname = 'contas';
 table_name |                                        relacl
------------+-----------------------------------------------------------------------------------------
---
 contas     |
{controller=arwdDxtm/controller,contabilidade=r/controller,moliveira=a*d*/controller}
(1 row)

sis_contabil=> exit
[postgres@vm02 vagrant]$ psql -U moliveira -d sis_contabil
psql (17.2)
Type "help" for help.

sis_contabil=> INSERT INTO controladoria.contas(id, numero, responsavel) VALUES (1, 1000,
'jsilva');
INSERT 0 1
sis_contabil=> INSERT INTO controladoria.contas(id, numero, responsavel) VALUES (2, 2000,
'psouza');
INSERT 0 1
```

```
        sis_contabil=> INSERT INTO controladoria.contas(id, numero, responsavel) VALUES (3, 2000,
'moliveira');
        INSERT 0 1
        sis_contabil=> ALTER TABLE controladoria.contas ENABLE ROW LEVEL SECURITY;
        ERROR:  must be owner of table contas
        sis_contabil=> exit
        [postgres@vm02 vagrant]$ psql -U controller -d sis_contabil
        psql (17.2)
        Type "help" for help.

        sis_contabil=> ALTER TABLE controladoria.contas ENABLE ROW LEVEL SECURITY;
        ALTER TABLE
        sis_contabil=>    CREATE POLICY contas_responsavel_policy
           ON controladoria.contas
           FOR ALL
           USING (responsavel = current_user);
        CREATE POLICY
        sis_contabil=> exit
        [postgres@vm02 vagrant]$ psql -U jsilva -d sis_contabil
        psql (17.2)
        Type "help" for help.

        sis_contabil=> SELECT * FROM controladoria.contas;
         id | numero | responsavel
        ----+--------+-------------
          1 |   1000 | jsilva
        (1 row)

        sis_contabil=> exit
        [postgres@vm02 vagrant]$ psql -U moliveira -d sis_contabil
        psql (17.2)
        Type "help" for help.

        sis_contabil=> SELECT * FROM controladoria.contas;
         id | numero | responsavel
        ----+--------+-------------
          3 |   2000 | moliveira
        (1 row)
        sis_contabil=> exit
        [postgres@vm02 vagrant]$ psql -U controller -d sis_contabil
        psql (17.2)
        Type "help" for help.

        sis_contabil=> SELECT * FROM controladoria.contas;
         id | numero | responsavel
        ----+--------+-------------
          1 |   1000 | jsilva
          2 |   2000 | psouza
          3 |   2000 | moliveira
        (3 rows)
```

Projetos

- Repositório Github Admin Banco de Dados DES11: Repositório contendo todos os scripts SQL, configurações, exercícios práticos e atividades desenvolvidas durante o curso de Administração de Banco de Dados (DES11), abordando tópicos como gerenciamento de usuários, roles, permissões, segurança e otimização de banco de dados PostgreSQL.

Referências (Material do Curso)

- ESCOLA SUPERIOR DE REDES (RNP). **Administração de Banco de Dados DES11: Capítulo 4 - Administrando Usuário e Segurança**. Material do curso DES11. (Arquivo: `DES6-Mod04-Apresentacao.pdf`).
- Hans-Jürgen Schönig (Packt). **Mastering PostgreSQL 17**. Elevate your database skills with advanced deployment, optimization, and security strategies (6th Edition).