Instituto Federal

Campus Goiânia

Bacharelado em Sistemas de Informação

Banco de Dados II

Prof. Dory Gonzaga Rodrigues

MySQL®

ORACLE®

PostgreSQL
the world's most advanced open source database

Agenda

- Administração do Banco

  - Controle de Acesso e Permissões

CONTROLE DE ACESSO

Objetivo

- Proteção contra acessos mal intencionados;

- Controlar quais dados um determinado usuário ou grupo de usuários pode ter acesso;

- Controlar quais operações um determinado usuário ou grupo de usuários pode realizar sobre um determinado conjunto de dados;

CONTROLE DE ACESSO

Qual a importância do Controle de Acesso ?

- Uma das maiores preocupações em computação tem sido segurança da informação;

- Nos dias atuais, com o uso da Internet os sistemas estão disponíveis a seus usuários 24horas por dia e 07 dias por semana, vulneráveis a ataques maliciosos;

- Portanto, dentro do nosso escopo de estudo, os SGBDs devem prover uma camada de segurança que visa compor o arsenal de segurança da informação numa corporação;

CONTROLE DE ACESSO

Definindo o que é Segurança da Informação em Banco de Dados

- Segurança em Banco de dados diz respeito à proteção contra acesso e alterações intencionais ou não intencionais utilizando-se ou não de meios computacionais;

- Segurança do banco de dados herda os mesmos princípios da segurança da informação:

  - Impedir o roubo e fraude de dados;
  - Garantir a confidencialidade dos dados;
  - Garantir a integridade dos dados;
  - Garantir a disponibilidade dos dados.

- Uma visão mais detalhada sobre Segurança em Banco de Dados pode ser vista em:
www.perallis.com/news/o-que-a-gartner-pensa-sobre-seguranca-de-banco-de-dados

CONTROLE DE ACESSO

Como é feito o controle de acesso ?

- O subsistema de segurança é responsável por proteger o BD contra o acesso não autorizado.

- Inicialmente, o SGBD possui uma conta denominada DBA (Data Base Administrator), também conhecida como “super usuário”, com plenos poderes no bando de dados;

- O DBA é responsável pela:

  - Criação de contas
  - Concessão/Revogação de privilégios
  - Definição do nível de segurança conforme a política de segurança da empresa

CONTROLE DE ACESSO

Quais são os controles que podem ser implementados ?

- Controles de segurança computacionais:

  - Controle de Autorização e autenticação;
  - Procedimentos de Backup e Recovery;
  - Implementação das Constraints;
  - Stored procedures;
  - Criptografia;
  - Auditoria;
  - Views;
  - Política de Segurança da Informação;
  - Adequado local de trabalho, posicionando equipamentos de forma segura;
  - Controles de acesso físico aos locais de trabalho;
  - Manutenção preventiva dos equipamentos, etc.

CONTROLE DE ACESSO

Comandos de controle de Autorização e Autenticação

Os Comandos SQL são divididos em 3 categorias: DDL, DML e DCL:

- Data Definition Language (DDL)
  são usados para definir a estrutura de banco de dados ou esquema.

- Data Manipulation Language (DML)
  são utilizados na manipulação de dados.

- Data Control Language (DCL)
  são utilizados no controle de acesso ao banco de dados.

CONTROLE DE ACESSO

PostgreSQL – Comandos DCL

- O PostgreSQL utiliza o conceito de **ROLES** (**PAPEIS**) para controlar o acesso à base de dados.
- Uma papel (ROLE) provê todas as funcionalidades necessárias para o desenvolvimento de regras de controle de acesso a objetos e dados contidos em uma base de dados, tais como: tabelas, views, schema, sequences, etc.
- O comando DCL (Data Control Language) utilizado para criar um papel é:

CREATE ROLE <nome_role>

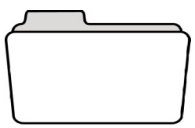
- Já para a exclusão deste papel (função) é:

DROP ROLE <nome_role>

CONTROLE DE ACESSO

Role de Grupo

- Utiliza-se roles de grupo para agregar um conjunto de privilégios.



Role de Grupo
CREATE ROLE nome_do_grupo;

Role do tipo Usuário

- São utilizadas como meio de acesso à base de dados.
- São os usuários do banco de dados, propriamente ditos.
- A estas roles é concedido o direto de iniciar uma nova conexão à base.
- O atributo que faz a distinção entre roles de usuário e roles de grupo: **LOGIN** .

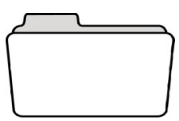
Role do tipo Usuário
CREATE ROLE nome_do_usuario LOGIN;

CONTROLE DE ACESSO

Exemplo do uso de Role com atributo LOGIN

Criação da Role (Usuário)

```
CREATE ROLE gerencia LOGIN;
```



Removendo o direito de LOGIN

```
ALTER  ROLE gerencia NOLOGIN;
```

Adicionando o direito de usuário na Role

```
ALTER ROLE gerencia LOGIN;
```

Excluir uma Role

```
DROP ROLE gerencia;
```

CONTROLE DE ACESSO

Atributos de um Role

SUPERUSER | **NOSUPERUSER**

CREATEDB | **NOCREATEDB**

CREATEROLE | **NOCREATEROLE**

INHERIT | **NOINHERIT**

REPLICATION | **NOREPLICATION**

CONNECTION LIMIT limite_conexão

[ENCRYPTED | UNENCRYPTED] **PASSWORD** 'senha'

VALID UNTIL 'tempo_absoluto'

Exemplo do uso de Role com atributo SUPERUSER

Este atributo sobrepõe qualquer checagem de restrição de acesso, com exceção do direito de replicação da base de dados.

Somente uma role conectada com direto de SUPERUSER pode atribuir o poder de SUPERUSER a outra role.

Criação da Role com atributo SUPERUSER

```
CREATE ROLE financeiro SUPERUSER;
```

Adicionando o atributo

```
ALTER ROLE gerencia SUPERUSER;
```

Removendo o atributo

```
ALTER ROLE gerencia NOSUPERUSER;
```

CONTROLE DE ACESSO

Atributo CREATEDB / NOCREATEDB

Este atributo concede o direito de criar uma nova base de dados.

Criação da Role (valor default é NOCREATEDB**)**
CREATE ROLE financeiro **CREATEDB;**

Alterando o valor do atributo
ALTER ROLE gerencia **CREATEDB;**

Alterando o valor do atributo
ALTER ROLE gerencia **NOCREATEDB;**

CONTROLE DE ACESSO

Atributo CREATEROLE / NOCREATEROLE

Este atributo concede o direito de criar, excluir ou alterar outras roles.

Criação da Role (valor default é NOCREATEROLE**)**
CREATE ROLE financeiro **CREATEROLE;**

Alterando o valor do atributo
ALTER ROLE gerencia **CREATEROLE;**

Alterando o valor do atributo
ALTER ROLE gerencia **NOCREATEROLE;**

CONTROLE DE ACESSO

Atributo INHERIT / NOINHERIT

O atributo INHERIT indica se é possível repassar os privilégios concedidos ao grupo quando uma role for associada direta ou indiretamente.

Já o NOINHERIT requer o uso do comando SET ROLE para alcançar os privilégios.

Criação da Role (valor default é Null**)**

```
CREATE ROLE financeiro INHERIT;
```

Alterando o valor do atributo

```
ALTER ROLE gerencia INHERIT;
```

Alterando o valor do atributo

```
ALTER ROLE financeiro NOINHERIT;
```

CONTROLE DE ACESSO

Atributo REPLICATION / NOREPLICATION

Replicação permite distribuir os dados mantendo uma cópia destes em outro local. Normalmente utilizada para realizar cópia de segurança (backup). Por omissão, o padrão é NOREPLICATION.

Este atributo concede o direito a uma role iniciar um fluxo de replicação.

Também é necessário que a role tenha o direito de LOGIN para que o direito de REPLICATION seja efetivo.

Criação da Role (valor default é NOREPLICATION**)**

```
CREATE ROLE financeiro REPLICATION;
```

Alterando o valor do atributo

```
ALTER ROLE gerencia REPLICATION;
```

Alterando o valor do atributo

```
ALTER ROLE gerencia NOREPLICATION;
```

CONTROLE DE ACESSO

Exemplo do uso de Role com atributo VALID UNTIL

Este atributo estabelece a data de validade para a senha atribuída à role.
Caso não seja especificado um TimeStamp, assume-se que a senha não expira.

Criação da Role (com tempo de validade)

```
CREATE ROLE financeiro VALID UNTIL '2018-01-31 19:00:55';
```

Adicionando tempo de validade

```
ALTER ROLE gerencia VALID UNTIL '2018-05-25 20:30:55';
```

CONTROLE DE ACESSO

Atributo CONNECTION LIMIT

Este atributo é utilizado em uma conexão (role com atributo LOGIN).
O atributo CONNECTION LIMIT especifica quantas conexão simultâneas são permitidas.

Criação da Role (valor default é -1 e não há limite de conexões**)**
CREATE ROLE financeiro **CONNECTION LIMIT -1;**

Alterando o valor do atributo
ALTER ROLE gerencia **CONNECTION LIMIT 5;**

CONTROLE DE ACESSO

Atributo PASSWORD / NOPASSWORD

Este atributo define uma senha a ser utilizada para autenticação.
O uso de PASSWORD só faz sentido em roles com o atributo LOGIN.

Criação da Role (valor default é Null**)**
CREATE ROLE financeiro **LOGIN PASSWORD** ‘minha_senha’**;**

Alterando o valor do atributo
ALTER ROLE financeiro **PASSWORD** ‘nova_senha’**;**

Alterando o valor do atributo
ALTER ROLE gerencia **NOPASSWORD;**

CONTROLE DE ACESSO

ROLES de GRUPO

- Roles de grupo são aquelas que não possuem o atributo LOGIN ativo, ou seja, o valor é NOLOGIN. Portanto, não são utilizadas para autenticar clientes na base de dados.

- O objetivo de uma role de grupo é agregar um conjunto de privilégios dentro de uma visão lógica, de negócio.

- O PostgreSQL possui um grupo padrão, chamado PUBLIC, ao qual toda a role automaticamente é membro.

- A associação é realizada no momento da criação da role.

- Assim, a soma dos privilégios de uma role é obtido somando-se os privilégios herdados da associação a grupos, dos privilégios diretamente concedidos mais àqueles oriundos do grupo PUBLIC.

CONTROLE DE ACESSO

O comando GRANT

- O comando GRANT concede privilégios a uma role.

- Através deste comando que uma role de usuário torna-se membro de uma role de grupo.

GRANT <privilégios> ON <objetos> TO <roles> ;

- Pode-se atribuir a uma role os seguintes privilégios:

SELECT	UPDATE	INSERT
DELETE	TRUNCATE	REFERENCES
TRIGGER	CREATE	CONNECT
TEMPORARY/TEMP	EXECUTE	USAGE
ALL PRIVILEGES/ALL		
WITH GRANT OPTION		
WITH ADMIN OPTION		

CONTROLE DE ACESSO

O comando GRANT e seus PRIVILÉGIOS

SELECT

Concede direito de executar comandos de seleção sobre tabelas, views e sequences.

- É possível especificar uma ou mais tabelas ou views.
- É possível especificar uma ou mais colunas de uma tabela ou view.
- Para “sequence”, somente é concedida a execução da função currval.

Ex:

```
GRANT SELECT(nome, sobrenome) ON TABLE cliente TO gerencia;

GRANT SELECT ON TABLE cliente TO gerencia;

GRANT SELECT ON SEQUENCE cliente_id_seq TO gerencia;
```

CONTROLE DE ACESSO

O comando GRANT e seus PRIVILÉGIOS

UPDATE

Concede direito de executar comandos de atualização sobre tabelas e sequences.

- É possível especificar uma ou mais colunas de uma tabela.
- Para “sequence” é concedida a execução das funções:
nextval (retorna o próximo valor) e setval (atribui um valor).

Ex:

```
GRANT UPDATE(nome, sobrenome) ON cliente TO gerencia;

GRANT UPDATE ON cliente TO gerencia;

GRANT UPDATE ON SEQUENCE cliente_id_seq TO gerencia;
```

CONTROLE DE ACESSO

O comando GRANT e seus PRIVILÉGIOS

INSERT
Concede direito de executar comandos de inclusão de novos registros em uma tabela.

- Pode-se especificar a listagem de colunas para as quais será permitido especificar valores.

Ex: GRANT INSERT ON estoque TO gerencia;

DELETE
Concede direito de executar comandos de exclusão de registros de uma tabela.

Ex: GRANT DELETE ON estoque TO gerencia;

CONTROLE DE ACESSO

O comando GRANT e seus PRIVILÉGIOS

TRUNCATE

Concede direito de executar o comando que APAGUE TODOS OS DADOS nas tabelas definidas

```
Ex:  GRANT TRUNCATE ON estoque TO gerencia;
```

REFERENCES

Concede direito do usuário criar chaves estrangeiras(referências). É necessária a concessão tanto para a coluna da chave primária quanto para a coluna da chave estrangeira.

```
Ex:
   GRANT REFERENCES(id_cliente) ON nota_fiscal TO gerencia;
      GRANT REFERENCES(id) ON cliente TO gerencia;
```

CONTROLE DE ACESSO

O comando GRANT e seus PRIVILÉGIOS

TRIGGER

Permite que sejam criadas triggers nas tabelas sobre as quais o privilégio foi concedido.

```
Ex:  GRANT TRIGGERS ON cliente TO financeiro;
```

CREATE

Concede direito de criação de objetos em uma base de dados, esquema ou tablespace.

```
Ex:  GRANT CREATE ON DATABASE empresaX TO supervisor;

     GRANT CREATE ON SCHEMA address TO supervisor;

     GRANT CREATE ON TABLESPACE ts_empresaX TO supervisor;
```

CONTROLE DE ACESSO

O comando GRANT e seus PRIVILÉGIOS

CONNECT

Este privilégio concede direito a um usuário para conectar-se em determinada base de dados.

```
Ex:  GRANT  CONNECT  ON  DATABASE empresaX  TO  supervisor;
```

TEMPORARY / TEMP

Este privilégio permite a criação de tabelas temporárias na base especificada.

```
Ex:  GRANT  TEMPORARY  ON  DATABASE empresaX  TO  supervisor;

   ou

     GRANT  TEMP  ON  DATABASE empresaX  TO  supervisor;
```

CONTROLE DE ACESSO

O comando GRANT e seus PRIVILÉGIOS

EXECUTE

Este privilégio permite que uma função seja executada.

```
Ex:   GRANT  EXECUTE  ON  FUNCTION f_testeX()  TO  supervisor;
```

USAGE

Este privilégio define comportamentos diferentes para cada objeto sobre o qual ele é definido. Por exemplo:

- Para uma linguagem procedural, permite que seja usada pela role.
- Para esquemas, permite que sejam listados os objetos contidos.
- Para sequences, permite a execução das funções currval e nextval.

```
Ex:   GRANT  USAGE  ON  SCHEMA address  TO  h_nivel_3;
      GRANT  USAGE  ON  LANGUAGE plpgsql TO  h_nivel_3;
```

CONTROLE DE ACESSO

O comando GRANT e seus PRIVILÉGIOS

ALL PRIVILEGES / ALL
Este privilégio concede todos os direitos possíveis de uma só vez.

Ex:

```
GRANT ALL ON TABLE cliente TO supervisor;
```

WITH GRANT OPTION
Esta diretiva permite que a role que está recebendo o privilégio possa concedê-lo a outras.

Ex:

```
GRANT ALL ON TABLE cliente TO supervisor WITH GRANT OPTION;
```

CONTROLE DE ACESSO

O comando GRANT e seus PRIVILÉGIOS

WITH ADMIN OPTION

Concede também o direito ao usuário receptor de realizar associação de outros usuários a este mesmo grupo.

Ex:

```
GRANT supervisor TO dory WITH ADMIN OPTION;
```

CONTROLE DE ACESSO

Concedendo direitos de grupo a um usuário

Roles de grupo são usadas para gerir um conjunto de privilégios compartilhados entre vários usuários (roles com atributo de LOGIN).

Para realizar a associação de uma role de usuário em um determinado grupo, utiliza-se o comando GRANT, de forma semelhante às concessões de privilégios.

O comando abaixo demonstra o ingresso do usuário nos grupos financeiro e venda. A partir desta associação, os privilégios concedidos aos dois grupos ficam disponíveis ao usuário.

```
GRANT financeiro, venda TO dory;
```

Comando REVOKE

Todo os privilégios concedidos através do comando GRANT podem ser revogados pelo comando REVOKE.

```
REVOKE SELECT ON  estoque  FROM  h_nivel_1;
```

Revogando-se determinado privilégio de um grupo, automaticamente revoga-se de todas as roles associadas a este grupo.

Contudo, no caso do mesmo privilégio revogado estar presente em outro grupo, ao qual a role está associada, então o privilégio oriundo do outro grupo permanece.

Comando REVOKE

Quando o privilégio do concessor for revogado, é necessário adicionar a diretiva CASCADE, a qual indica que a revogação deve ser realizada em cascata para todos os dependentes. O uso do CASCADE se faz necessário porque, por padrão, o comando REVOKE considera a cláusula RESTRICT, que limita a ação a role especificada no comando.

```
REVOKE  SELECT ON produtos_estoque FROM h_nivel_2 CASCADE;
```

Revoga o direito de conceder grant (diretiva WITH GRANT OPTION)

```
REVOKE GRANT OPTION FOR select ON produtos_estoque FROM h_nivel_2;
```

Revoga o direito de associar usuários a um grupo (diretiva ADMIN OPTION FOR)

```
REVOKE  ADMIN OPTION FOR h_nivel_1 FROM h_nivel_2;
```