

# DCL - Linguagem de Controle de Dados

Prof. Igor Avila Pereira  
igor.pereira@riogrande.ifrs.edu.br

Divisão de Computação  
Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS)  
Câmpus Rio Grande

# Agenda

- 1 Introdução
- 2 Motivação
- 3 Controle de Usuários
- 4 Controle de Papéis
- 5 Controle de Privilégios

# Agenda

- 1 Introdução
- 2 Motivação
- 3 Controle de Usuários
- 4 Controle de Papéis
- 5 Controle de Privilégios

## Introdução

A linguagem SQL é uma só, porém ela é dividida em tipos de acordo com a funcionalidade dos comandos:

- **DDL - Data Definition Language - Linguagem de Definição de Dados:** São os comandos que interagem com os objetos do banco.
  - São comandos DDL: CREATE, ALTER e DROP
- **DML - Data Manipulation Language - Linguagem de Manipulação de Dados:** São os comandos que interagem com os dados dentro das tabelas.
  - São comandos DML: INSERT, DELETE e UPDATE

## Introdução

- **DQL - Data Query Language - Linguagem de Consulta de dados:** São os comandos de consulta. São comandos DQL: SELECT (é o comando de consulta)
  - Aqui cabe um parênteses: Em alguns livros o SELECT fica na DML em outros tem esse grupo próprio.
- **DTL - Data Transaction Language - Linguagem de Transação de Dados:** São os comandos para controle de transação.
  - São comandos DTL: BEGIN TRANSACTION, COMMIT E ROLLBACK
- **DCL - Data Control Language - Linguagem de Controle de Dados:** São os comandos para controlar a parte de segurança do banco de dados.
  - São comandos DCL: GRANT, REVOKE E DENY.

# Agenda

- 1 Introdução
- 2 Motivação**
- 3 Controle de Usuários
- 4 Controle de Papéis
- 5 Controle de Privilégios

## Motivação

- Muitas pessoas se conectam a um determinado banco de dados
- Essas pessoas devem ter permissões de acesso e manipulação específicas para:
  - Banco de dados
  - Tabelas
  - Views
  - Stored Procedures
- Em SQL o pacote que manipula as definições de usuário e permissões é a DCL:
  - DCL: Data Control Language

# Agenda

- 1 Introdução
- 2 Motivação
- 3 Controle de Usuários**
- 4 Controle de Papéis
- 5 Controle de Privilégios



## Controle de Usuários

- **Criação de Usuário**
  - CREATE USER

```
CREATE USER <nome_user>  
[WITH <option1> [<option2> ...]]
```

- [] = opcionalidade
- <> = preencher com algum valor
- {} = escolha um dos itens/opções

## Controle de Usuários

### Criação de Usuário:

- **Options:**
  - **CREATEDB** ou **NOCREATEDB**: pode ou não criar BD's
  - **SUPERUSER** ou **NOSUPERUSER**: superusuário ou não
  - **CREATEROLE** ou **NOCREATEROLE**: se pode criar papéis
  - **IN ROLE** <nome papel>: já adiciona o usuário em um ou mais papéis
  - **ENCRYPTED PASSWORD** '<senha>': atribui senha ao novo *user*

**NOCREATEDB, NOSUPERUSER e NOCREATEROLE** são **default**

## Controle de Usuários

### Criação de Usuário:

Ex:

```
-- criando um user <username>
sudo -u postgres createuser <username>
-- criando um database <dbname>
sudo -u postgres createdb <dbname>
-- definindo senha <password> para o user <username>
sudo -u postgres psql
psql=# alter user <username> with encrypted
      password '<password>';
-- 'privileges' de um <dbname> para o user <username>
psql=# grant all privileges on database <dbname>
      to <username>;
```

## Controle de Usuários

### Criação de Usuário:

#### Ex (via PSQL):

```
CREATE DATABASE <yourdbname>;  
CREATE USER <youruser>  
    WITH ENCRYPTED PASSWORD 'yourpass';  
GRANT ALL PRIVILEGES ON DATABASE <yourdbname>  
    TO <youruser>;
```

## Controle de Usuários

### Criação de Usuário: +Alguns Exemplos

```
-- um simples user (newuser)
```

```
CREATE USER newuser;
```

```
-- user (newuser) com password 12345
```

```
CREATE USER newuser WITH PASSWORD '12345';
```

```
-- user (newuser) com password que expira (31/12/20)
```

```
CREATE USER newuser WITH PASSWORD  
    VALID UNTIL '2020-12-31';
```

```
-- user (newuser), senha 'pass' e com CREATEDB
```

```
CREATE USER newuser WITH PASSWORD 'pass' CREATEDB;
```

## Controle de Usuários

### Criação de Usuário (SUPERUSER):

```
-- 1) client utility  
sudo -u postgres createuser --superuser <name>  
-- 2) short tag  
sudo -u postgres createuser -s <name>  
-- 3) psql  
CREATE USER <name> SUPERUSER;
```

## Controle de Usuários

### Criação de Usuário (Formas de Atribuir Senha a um Usuário):

```
-- 1) client utility
sudo -u postgres createuser <name> --pwprompt
-- 2) short tag
sudo -u postgres createuser <name> -P
-- 3) psql command
CREATE USER <name> WITH PASSWORD '<password>';
```

## Controle de Usuários

### Criação de Usuário com Adição de Privilégios:

```
-- 1) client utility  
createuser <option> <name>  
-- 2) PSQL command:  
CREATE USER <name> WITH <option>;
```



## Controle de Usuários

Sintaxe <i>Option</i>	PSQL	Descrição
-s -superuser	SUPERUSER	Definindo user como SUPERUSER.
-S -no-superuser	NOSUPERUSER	User não terá o privilégio de SUPERUSER (default).
-d -createdb	CREATEDB	Permite que user crie <i>databases</i> .
-D -no-createdb	NOCREATEDB	Não permite que o user crie <i>databases</i> (default).
-r -createrole	CREATEROLE	Permite que o user crie novos papéis ( <i>roles</i> ).
-R -no-createrole	NOCREATEROLE	Não permite que o user crie novos papéis ( <i>roles</i> ) (default).
-i -inherit	INHERIT	Automaticamente herda privilégios de outros papéis ( <i>roles</i> ) (default).
-I -no-inherit	NOINHERIT	Não herda privilégios de outros papéis.
-l -login	LOGIN	Permite que o usuário logue em uma sessão com seu nome de papel (default).
-L -no-login	NOLOGIN	Não permite que o usuário logue em uma sessão com seu nome de papel.
-P -pwprompt	PASSWORD '<password>'	Inicializa o <i>prompt</i> para a criação da senha ou fornece uma senha para o usuário.
/	PASSWORD NULL	Define a senha para NULL. Toda autenticação falhará para este usuário.
-c <number> -connection-limit=<number>	CONNECTION LIMIT <number>	Define o número máximo de conexões para este usuário. Padrão (default) é sem limite.

## Controle de Usuários

### Criação de Usuário com Adição de Privilégios:

```
-- 1) client utility  
sudo -u postgres createuser -d -r -e <name>  
-- 2) psql equivalent command:  
CREATE USER <name> WITH CREATEROLE CREATEDB;
```

```
~$ sudo -u postgres createuser -d -r -e luke  
SELECT pg_catalog.set_config('search_path', '', false);  
CREATE ROLE luke NOSUPERUSER CREATEDB CREATEROLE INHERIT LOGIN;
```

## Controle de Usuários

### Criação de Usuário com Privilégios (iterativo):

```
sudo -u postgres createuser --interactive
```

## Controle de Usuários

### Criação de Usuário:

- No PostgreSQL:
  - user padrão: **postgres**
  - **postgres** também é o usuário **root** ou **superuser**
  - Para visualizar todos os usuários no PSQL há 3 opções:

```
1) \du
2) SELECT * FROM pg_user;
3) SELECT * FROM pg_roles;
```

## Controle de Usuários

### Excluir um Usuário:

```
DROP USER <nome_user>
```

### Alterar um Usuário:

```
ALTER USER <nome_user> WITH <option1> [<option2> ...]
```

## Agenda

- 1 Introdução
- 2 Motivação
- 3 Controle de Usuários
- 4 Controle de Papéis**
- 5 Controle de Privilégios

## Controle de Papéis

- Papéis são uma forma de agrupar usuários
- **Ex:** ao invés de se criar um usuário **aluno** e todos alunos se conectarem por esse **usuário**,
  - cria-se um usuário para cada **aluno**;
  - e um **papel aluno**;
  - vincula-se o **papel** a cada **usuário** criado.
- Os papéis podem estar vinculados a outros papéis
  - Criando uma hierarquia de papéis
  - Desta forma, um usuário terá os privilégios (permissões) atribuídas:
    - Diretamente ao seu usuário
    - Aos seus papéis
    - Aos papéis atribuídos aos seus papéis e, assim sucessivamente.

## Controle de Papéis

### **Criar um Papel:**

```
CREATE ROLE <nome_user> [WITH <option1> [<option2> ...]]
```

### **Excluir um Papel:**

```
DROP ROLE <nome_user> WITH <option1> [<option2> ...]
```

### **Alterar um Papel:**

```
ALTER ROLE <nome_user> WITH <option1> [<option2> ...]
```

### **Atribuindo Papéis:**

```
GRANT <nome_role> TO <user/role>, <user2/role> ...
```

### **Retirando um Papel:**

```
REVOKE <nome_role> FROM <user/role>, <user2/role> ...
```



# Agenda

- 1 Introdução
- 2 Motivação
- 3 Controle de Usuários
- 4 Controle de Papéis
- 5 Controle de Privilégios**

## Controle de Privilégios

### **Alguns tipos de privilégio:**

- CONNECT (em BD)
- SELECT (em tabela)
- UPDATE (em tabela)
- INSERT (em tabela)
- DELETE (em tabela)
- EXECUTE (em stored procedure ou function)

## Controle de Privilégios

### Sintaxe para CONCEDER privilégios:

```
GRANT <nome_privilégio> ON  
    {<nome_tabela>|<nome_bd>|FUNCTION <nome_function>}  
TO <nome_role/nome_user>, ...
```

### Sintaxe para RETIRAR privilégios:

```
REVOKE <nome_privilégio> ON  
    {<nome_tabela>|<nome_bd>|FUNCTION <nome_function>}  
FROM <nome_role/nome_user>, ...
```

## Controle de Privilégios

### Exemplos:

```
REVOKE update (nome) ON departamento  
FROM estagiario, analista
```

```
GRANT ALL PRIVILEGES ON funcionario, projeto  
TO estagiario
```

```
GRANT connect ON DATABASE base1 TO estagiario
```

```
REVOKE execute ON FUNCTION func_1() FROM estagiario
```

## Controle de Privilégios

### No PostgreSQL:

- `\dp` ou `\z`: visualiza privilégios de tabelas, sequências e visões
  - Mostra quando já foi definido algum privilégio diferente dos *defaults* na tabela

**Legenda:** `< user >` = permissões/dono

- a = inserir
- w = update
- r = select
- d = deletar
- D = truncate
- x = references
- t = trigger

## Controle de Privilégios

### No PostgreSQL:

- \l: visualiza privilégios de banco de dados
  - Mostra quando já foi definido algum privilégio diferente dos *defaults* no B.D

**Legenda:** < user > = permissões/dono

- X = execute
- U = usage
- c = connect
- C = create
- T = temporary

## DCL - Linguagem de Controle de Dados

Prof. Igor Avila Pereira  
igor.pereira@riogrande.ifrs.edu.br

Divisão de Computação  
Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS)  
Câmpus Rio Grande