# Unidade 6: SEGURANÇA E AUTORIZAÇÃO EM BANCO DE DADOS

Prof. Rodrigo Baroni

PUC Minas Virtual

# Segurança em BD

- DCL (*Data Control Language)*
- O banco de dados  envolve armazenamento de informações estratégicas, e às vezes sigilosas, da organização.
- Cabe ao DBA em conjunto com o analista de sistema e o usuário gestor do sistema estabelecer a política de privilégios.
- DCL / DML: O aspecto segurança de dados relaciona qual usuário ou grupo de usuários tem privilégio de INSERT, DELETE, SELECT ou UPDATE em uma tabela ou visão.

# Tipos de Privilégios

**Dois tipos de privilégios podem ser dados aos usuários:**

**Privilégio de Sistema**

- Habilita usuários realizarem ações particulares no banco de dados

**Privilégio de Objeto**

- Habilita usuários acessarem e manipularem um objeto específico

Fonte: CASTRO, Profa. Laura (2018)

# Privilégios de Sistema: Exemplos

- CREATE SESSION
- UNLIMITED TABLESPACE
- CREATE TABLE
- CREATE TABLESPACE
- CREATE TRIGGER
- CREATE USER
- CREATE VIEW
- DELETE ANY TABLE
- DROP ANY TABLE
- DROP USER

Fonte: CASTRO, Profa. Laura (2018)

# Concessão de Privilégios de Objeto

Usuários que podem conceder privilégio em um determinado objeto:

- dono do objeto
- DBA
- qualquer usuário que tenha sido dado explicitamente permissão de conceder privilégio no objeto

Fonte: CASTRO, Profa. Laura (2018)

# Privilégios de Objeto: Exemplos

**Tables**

- select, insert, update, delete, alter

**Views**

- select, insert, update, delete

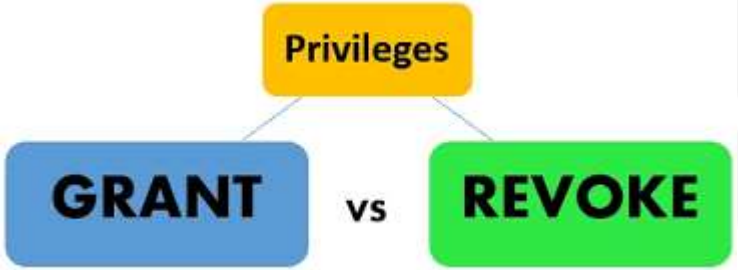
**Packages, Procedures, Functions**

- execute, debug

Fonte: CASTRO, Profa. Laura (2018)

# Segurança em BD

- O privilégio de SELECT é o mais usual, sendo os outros mais restritos.
- Quando se deseja conceder todos os privilégios, a palavra chave ALL é utilizada.
- Quando se deseja conceder privilégios para todos os usuários, a palavra chave PUBLIC é utilizada.



# Comando GRANT

- A sintaxe do comando GRANT é a seguinte:

**GRANT <privilégios> ON <objeto> TO <usuário, grupo de usuários ou PUBLIC>**

- Exemplos:

GRANT SELECT, INSERT ON FUNCIONARIOS TO RAQUEL

GRANT ALL ON FUNCIONARIOS TO CARLOS, ANTONIO

GRANT SELECT ON ORGANIZACAO TO PUBLIC

GRANT ALL ON FUNCIONARIOS TO PUBLIC

(Obs: Concede privilégios para todos na tabela)

# GRANT with GRANT OPTION

- Concede o direito ao usuário de repassar a permissão para frente. Deve ser usada com muito critério.
- Não é a opção default

Exemplos:

GRANT SELECT, INSERT ON FUNCIONARIOS TO RAQUEL with GRANT OPTION

Após esse comando, a usuária Raquel poderá repassar para quem quiser os privilégios de Select e Insert na tabela Funcionarios

# Comando REVOKE

- O comando REVOKE tem o efeito revogar uma permissão (GRANT) pois revoga privilégios proibições já concedidos.
- A sintaxe do comando REVOKE é a seguinte:

REVOKE <privilégios> ON <objeto> FROM <usuário, grupos de usuários ou PUBLIC>

# Exemplos do Comando REVOKE

1. REVOKE SELECT, INSERT ON FUNCIONARIOS FROM MARIA, CARLOS
2. REVOKE ALL ON FUNCIONARIOS FROM PUBLIC(Obs: Retira privilégios de todos na tabela)
3. REVOKE GRANT OPTION FOR SELECT, INSERT ON FUNCIONARIOS FROM RAQUEL

Esse comando de Revoke mantém os direitos da usuária Raquel, mas cancela a possibilidade de passá-los adiante.

# Role (papel)

É um grupo nomeado de privilégios relacionados que são concedidos para usuários ou outras roles.

É útil para facilitar a administração de privilégios no banco de dados e portanto melhorar a segurança.

```
CREATE  ROLE  nome_role;
```

```
DROP  ROLE  nome_role;
```

Fonte: CASTRO, Profa. Laura (2018)

# Role

**1) Concessão de privilégio para role**

```
GRANT <privilégios> TO <nome_role>;
```

**2) Concessão de role a usuário**

```
GRANT <nome_role>
TO <usuário>
[WITH ADMIN OPTION];
```

Fonte: CASTRO, Profa. Laura (2018)

# Profile (Perfil)

É um conjunto de limites de recursos e senhas

O banco de dados Oracle cria automaticamente o perfil DEFAULT que não possui nenhum limite

Fonte: CASTRO, Profa. Laura (2018)

# Motivações para uso de perfil

- Impedir que os usuários executem operações que requeiram uso excessivo de recursos do sistema.
- Garantir que usuários desconectem do banco depois de deixar suas sessões sem uso por algum tempo.
- Possibilitar o agrupamento de recursos e atribuição a um grupo de usuários.
- Controlar o uso de senhas

Fonte: CASTRO, Profa. Laura (2018)

# Criação/Remoção de Perfil

```
CREATE PROFILE <perfil> LIMIT
{ parametro_recurso | parametro_senha }
[parametro_recurso | parametro_senha]...;
```

```
DROP PROFILE <perfil> [CASCADE];
```

**Cascade:**
O perfil é retirado de todos os usuários aos quais o perfil havia sido definido e o profile DEFAULT é definido para tais usuários.

Fonte: CASTRO, Profa. Laura (2018)

# Parâmetro de Recursos

```
{ { SESSIONS_PER_USER   |
    CPU_PER_SESSION   |
    CPU_PER_CALL   |
    CONNECT_TIME   |
    IDLE_TIME   |
    LOGICAL_READS_PER_SESSION   |
    LOGICAL_READS_PER_CALL   }
{ integer | UNLIMITED | DEFAULT }}
```

Fonte: CASTRO, Profa. Laura (2018)

# Parâmetros de Recursos

**CPU_PER_SESSION:**

- total de tempo de CPU medido em centésimos de segundo.

**SESSIONS_PER_USER:**

- número de sessões concorrentes permitidas por cada usuário.

**CONNECT_TIME:**

- tempo de conexão medidos em minutos.

**IDLE_TIME:**

- tempo inatividade de sessão medido em minutos.

Fonte: CASTRO, Profa. Laura (2018)

# Parâmetros de Recursos (Continuação)

**LOGICAL_READS_PER_SESSION:**

- número de blocos de dados lidos (leitura lógica e física, ou seja, memória e disco)

**CPU_PER_CALL:**

- tempo de CPU por chamada (um SQL executado) em centésimos de segundo.

**LOGICAL_READS_PER_CALL:**

- número de blocos de dados lidos.

Fonte: CASTRO, Profa. Laura (2018)

# Parâmetros de Senha

```
{
 { FAILED_LOGIN_ATTEMPTS   |
     PASSWORD_LIFE_TIME    |
   PASSWORD_REUSE_TIME     |
   PASSWORD_REUSE_MAX      |
   PASSWORD_LOCK_TIME      }
 {integer| UNLIMITED | DEFAULT}|
  PASSWORD_VERIFY_FUNCTION { function | NULL | DEFAULT }
}
```

Fonte: CASTRO, Profa. Laura (2018)

# Parâmetros de Senha

**FAILED_LOGIN_ATTEMPTS:**

- Número de tentativas sem sucesso na conexão antes da conta ser bloqueada.

**PASSWORD_LIFE_TIME:**

- Tempo de vida da senha em dias antes da senha expirar.

**PASSWORD_REUSE_TIME:**

- Número de dias antes que a senha possa ser reusada.

**PASSWORD_REUSE_MAX:**

- Número de trocas de senhas requeridas antes da senha corrente poder ser reusada.

Fonte: CASTRO, Profa. Laura (2018)

# Parâmetros de Senha (Continuação)

## PASSWORD_LOCK_TIME:

- Número de dias que a conta ficará bloqueada depois de um determinado número de tentativas consecutivas sem sucesso de conexão ao banco.

## PASSWORD_VERIFY_FUNCTION:

- Possibilita verificação de tamanho, conteúdo e complexidade senhas.
- Função PL/SQL que faz a verificação da complexidade da senha antes da senha ser atribuída.

Fonte: CASTRO, Profa. Laura (2018)

# Função de Verificação

```
CREATE OR REPLACE FUNCTION verifica_senha
(username VARCHAR2,
 password VARCHAR2,
 old_password VARCHAR2)
RETURN BOOLEAN AS
BEGIN
        IF LENGTH(password) < 4 THEN
                RETURN FALSE;
        ELSE
                RETURN TRUE;
        END IF;
END verifica_senha;
```

Fonte: CASTRO, Profa. Laura (2018)

# Passos para uso de perfil

**1. Criar perfil**

Comando : create profile

```
CREATE PROFILE desenvolvedor
LIMIT
SESSIONS_PER_USER 2
IDLE_TIME 5
CONNECT_TIME 10
FAILED_LOGIN_ATTEMPTS 3
PASSWORD_VERIFY_FUNCTION
verifica_senha();
```

**2. Definir perfil para os usuários**

Comandos:

- create user
- alter user

```
create user usu1
identified by xyz
profile desenvolvedor;
```

Fonte: CASTRO, Profa. Laura (2018)

PUC Minas
Virtual