

# Curso de Sistemas de Informação

## Disciplina: Administração e Gerenciamento de Banco de Dados

Prof: João Carlos Pinheiro

Data de Entrega: 10-12-2024

Aluno: Rudney Gabriel Gonçalves Fonseca

### Atividade 02 - Segurança em Banco de dados

#### *Cenário 01*

Banco de dados do cenário 01 foi criado com base no script “criação\_cenario\_01” e alguns registros inseridos pelo script “populamento\_basico\_cenario\_01”. Os privilégios exigidos em tal cenário encontram-se no arquivo “privilegios\_cenario\_01” e os testes em testes\_cenario\_01. Abaixo serão mostrados os comandos para a concessão dos privilégios seguidos de seus respectivos testes. Todos os scripts citados encontram-se em anexo.

*Privilégios de A (recuperação de todas as relações menos **DEPENDENTE** e permissão para concessão de privilégios)*

```
-- permissões do usuário A
GRANT SELECT, UPDATE
ON cenario_01.departamento TO 'A' WITH GRANT OPTION;
GRANT SELECT, UPDATE
ON cenario_01.funcionario TO 'A' WITH GRANT OPTION;
GRANT SELECT, UPDATE
ON cenario_01.localizacao_dep TO 'A' WITH GRANT OPTION;
GRANT SELECT, UPDATE
ON cenario_01.trabalha_em TO 'A' WITH GRANT OPTION;
GRANT SELECT, UPDATE
ON cenario_01.projeto TO 'A' WITH GRANT OPTION;
SHOW GRANTS FOR 'A';
```

## Testes A

```
-- teste usuário A
SELECT * FROM dependente;
SELECT * FROM funcionario;
```

✖	1 08:46:40	SELECT * FROM dependente LIMIT 0, 1000	Error Code: 1142. SELECT command denied to user 'A'
✔	2 08:46:42	SELECT * FROM funcionario LIMIT 0, 1000	3 row(s) returned

*Privilégios de B (A conta B pode recuperar todos os atributos de FUNCIONARIO e DEPARTAMENTO, exceto Salario, Cpf\_ger e Data\_inicio\_ger)*

```
-- permissões do usuário B
GRANT SELECT (Dnome,Dnumero) ON cenario_01.departamento to 'B';
GRANT SELECT (Pnome,Minicial,Unome,Cpf,Datanasc,Endereco,Sexo,Cpf_supervisor,Dnr) on cenario_01.funcionario to 'B';
SHOW GRANTS FOR 'B';
```

## Testes B

```
-- teste usuário B
SELECT Cpf from funcionario;
SELECT Salario from funcionario;
```

✔	1 08:51:13	SELECT Cpf from funcionario LIMIT 0, 1000	3 row(s) returned
✖	2 08:51:17	SELECT Salario from funcionario LIMIT 0, 1000	Error Code: 1143. SELECT comm

*Privilégios de C (A conta C pode recuperar ou modificar TRABALHA\_EM, mas só pode recuperar os atributos Pnome, Minicial, Unome e Cpf de FUNCIONARIO e os atributos Projnome e Projnumero de PROJETO.)*

```
-- permissões do usuário C
GRANT SELECT,UPDATE on cenario_01.trabalha_em to 'C';
GRANT SELECT (Pnome, Minicial, Unome,Cpf) ON cenario_01.funcionario to 'C';
GRANT SELECT (Projnome, Projnumero) ON cenario_01.projeto to 'C';
SHOW GRANTS FOR 'C';
```

## Testes C

```
-- testes usuário C
UPDATE trabalha_em set Horas = 30 where Projeto_Projnumero=102 and Fcpf='11122233344';
SELECT Salario FROM funcionario;
SELECT * from projeto;
```

✓	1 08:54:03 UPDATE trabalha_em set Horas = 30 where Projeto_Projnumero=102 and Fcpf='11122233344'	1 row(s) affected Rows matched: 1 Changed: 1
✗	2 08:54:05 SELECT Salario FROM funcionario LIMIT 0, 1000	Error Code: 1143. SELECT command denied to
✗	3 08:54:07 SELECT * from projeto LIMIT 0, 1000	Error Code: 1142. SELECT command denied to

*Privilégios de D (A conta D pode recuperar qualquer atributo de FUNCIONARIO ou DEPENDENTE e pode modificar DEPENDENTE.)*

```
-- permissões do usuário D
GRANT SELECT,UPDATE ON cenario_01.dependente to 'D';
GRANT SELECT ON cenario_01.funcionario to 'D';
SHOW GRANTS FOR 'D';
```

## Testes D

```
-- testes usuário D
UPDATE funcionario SET salario=0 WHERE Cpf='12345678900';
SELECT * FROM dependente;
SELECT * FROM funcionario;
```

✗	2 08:57:21 UPDATE funcionario SET salario=0 WHERE Cpf='12345678900'	Error Code: 1142. UPDATE command denied to
✓	3 08:57:23 SELECT * FROM dependente LIMIT 0, 1000	3 row(s) returned
✓	4 08:57:27 SELECT * FROM funcionario LIMIT 0, 1000	3 row(s) returned

*Privilégios de E (A conta E pode recuperar qualquer atributo de FUNCIONARIO, mas somente para tuplas de FUNCIONARIO que têm Dnr = 3.)*

```
-- permissões do usuário E
CREATE VIEW granularidade_usuario_E AS
SELECT * FROM cenario_01.funcionario where Dnr=3;

GRANT SELECT ON granularidade_usuario_E to 'E';
```

## Testes E

```
-- testes usuário E
```

```
SELECT * FROM funcionario where Dnr=1;
```

```
1 08:59:23 SELECT * FROM funcionario where Dnr=1 LIMIT 0, 1000
```

## Cenário 02

O banco de dados do cenário 02 foi criado e populado com os scripts disponibilizados na própria atividade. O script usado concessão de permissão as roles é “criacao\_roles” e para a criação e associação de usuários as roles é “criação\_usuarios”.

*Conteúdo de “criacao\_roles” com os privilégios exigidos a cada uma*

```
CREATE ROLE 'role_admin';
CREATE ROLE 'role_gerente';
CREATE ROLE 'role_atendente';

-- privilégios admin
GRANT ALL PRIVILEGES ON vendasdb.* to 'role_admin';
SHOW GRANTS FOR 'role_admin';

-- privilégios gerente
GRANT SELECT,UPDATE,INSERT ON vendasdb.supplier to 'role_gerente';
GRANT SELECT, UPDATE,INSERT ON vendasdb.product to 'role_gerente';
GRANT SELECT, UPDATE,INSERT ON vendasdb.order to 'role_gerente';
SHOW GRANTS FOR 'role_gerente';

-- privilégios atendente
GRANT SELECT ON vendasdb.customer to 'role_atendente';
GRANT SELECT ON vendasdb.order to 'role_atendente';
SHOW GRANTS FOR 'role_atendente';
```

Conteúdos de “criação\_usuario”

```
CREATE USER 'admin_vendas' IDENTIFIED BY 'admin';
CREATE USER 'gerente_vendas' IDENTIFIED BY 'gerente';
CREATE USER 'atendente_vendas' IDENTIFIED BY 'atendente';
```

```
GRANT 'role_admin' TO 'admin_vendas';
SET DEFAULT ROLE 'role_admin' TO 'admin_vendas';
SHOW GRANTS FOR 'admin_vendas';
```

```
GRANT 'role_gerente' TO 'gerente_vendas';
SET DEFAULT ROLE 'role_gerente' TO 'gerente_vendas';
SHOW GRANTS FOR 'gerente_vendas';
```

```
GRANT 'role_atendente' TO 'atendente_vendas';
SET DEFAULT ROLE 'role_atendente' TO 'atendente_vendas';
SHOW GRANTS FOR 'atendente_vendas';
```

## Teste Gerente

-- Inserção de novos registros na tabela product com o usuário gerente\_vendas.

```
INSERT INTO Product (Id,ProductName,SupplierId,UnitPrice,Package,IsDiscontinued)VALUES(79,'Lasanha',1,30.00,'10 boxes x 20 bags',0)
SELECT * FROM Product where id=79;
```

	id	productname	supplierid	unitprice	package	isdiscontinued
▶	79	Lasanha	1	30.00	10 boxes x 20 bags	0
*	NULL	NULL	NULL	NULL	NULL	NULL

✓ 4 10:08:44 INSERT INTO Product (Id,ProductName,SupplierId,UnitPrice,Package,IsDiscontinued)VALUES(79,'Lasanha',1,30.00,'10 boxes x 20 bags',0)

## Testes Atendente

-- Consulta de dados na tabela customer com o usuário atendente\_vendas.

```
SELECT * FROM vendasdb.customer where firstname like 'R%';
```

	id	firstname	lastname	city	country	phone
▶	20	Roland	Menffvdel	Graz	Austria	7675-3425
	44	Renate	Mcdessner	Frankfurt a.M.	Germany	069-0245984
	55	Rene	Phicddllips	Anchorage	USA	(907) 555-7584
	86	Rita	Müledler	Stuttgart	Germany	0711-020361
*	NULL	NULL	NULL	NULL	NULL	NULL

✓ 2 10:11:47 SELECT \* FROM vendasdb.customer where firstname like 'R%' LIMIT 0, 1000

```
-- Tentativa de exclusão na tabela supplier com o usuário atendente_vendas (espera-se que ocorra erro).  
DELETE FROM vendasdb.supplier where id=1;
```

```
1 12:24:02 DELETE FROM vendasdb.supplier where id=1
```

## Conclusão

É notória a importância das políticas de segurança no gerenciamento de banco de dados uma vez que um banco de dados desprotegido e sem o devido gerenciamento de permissões pode acarretar variados prejuízos à organização e até mesmo a terceiros. Os conhecimentos oferecidos e testados nesta atividade e ao longo de toda a primeira etapa desta cadeira (como a manipulação da integridade do banco de dados, criação de usuários, concessão de permissões, entre outros) são de notável valor a um administrador de banco de dados.