

Universidade Tecnológica Federal do Paraná

Curso: Engenharia de Computação

Disciplina: Banco de dados 2

Professor: Wendel Góes

Anderson Soares da Silva 2150875

Maria Eduarda Pedroso 2150336

Mariana Gonçalves Rodrigues 2151014

Segurança e Autorização em Banco de Dados

TRABALHO 2 – Parte A

1) Considere o esquema de banco de dados relacional da figura a seguir.

EMPREGADO

PNOOME	MINICIAL	UNOME	<u>SSN</u>	DATANASC	ENDereco	SEXO	SALARIO	SUPERSSN	DNO
--------	----------	-------	------------	----------	----------	------	---------	----------	-----

DEPARTAMENTO

DNOOME	<u>DNUMERO</u>	GERSSN	GERDATAINICIO
--------	----------------	--------	---------------

DEPTO_LOCALIZACOES

<u>DNUMERO</u>	<u>DLOCALIZACAO</u>
----------------	---------------------

PROJETO

PJNOME	<u>PNUMERO</u>	PLOCALIZACAO	DNUM
--------	----------------	--------------	------

TRABALHA_EM

<u>ESSN</u>	<u>PNO</u>	HORAS
-------------	------------	-------

DEPENDENTE

<u>ESSN</u>	<u>NOME_DEPENDENTE</u>	SEXO	DATANASC	PARENTESCO
-------------	------------------------	------	----------	------------

Suponha que todas as relações foram criadas pelo usuário X (e, por isso, ele é o proprietário delas), que queira conceder os seguintes privilégios às contas de usuários A,B,C,D e E.

a) A conta A pode recuperar ou modificar quaisquer relações exceto dependentes e pode conceder privilégios para outros usuários.

GRANT SELECT, INSERT, UPDATE, DELETE ON empregado, departamento, depto_localizacoes, projeto, trabalha_em TO userA WITH GRANT OPTION;

b) A conta B pode recuperar todos os atributos de empregado e departamento, exceto salario, o numero de seguro social do gerente e a data de inicio do gerente.

GRANT SELECT (ssn, pnome, minicial, unome, datanasc, endereco, sexo, superssn, dno) ON empregado TO userB;

GRANT SELECT (dnumero, dnome) ON departamento TO userB;

c) A conta C pode recuperar ou modificar TRABALHA_EM, porem, pode apenas recuperar os atributos PNOOME, INICIALM, UNOME e NUMEROD0SEGUROSOCIAL de EMPREGADO e os atributos NOMEP e NUMEROP de PROJETO

GRANT SELECT, INSERT, UPDATE, DELETE ON trabalha_em TO userC;

```
GRANT SELECT (PNOME, MINICIAL, UNOME, SSN) ON empregado TO userC;  
GRANT SELECT (PJNOME, PNUMERO) ON projeto TO userC;
```

```
GRANT GRANT OPTION ON trabalha_em, empregado, projeto TO userC;
```

d) A conta D pode recuperar qualquer atributo de EMPREGADO ou dependente e pode modificar DEPENDENTE.

```
GRANT SELECT ON empregado TO userD;  
GRANT SELECT, INSERT, UPDATE, DELETE ON dependente TO userD;
```

e) A conta E pode recuperar qualquer atributo de EMPREGADO, porém, somente para as tuplas de EMPREGADO que possuam DNO = 3.

f) Criar tabelas com chave primaria e estrangeira

```
-- CRIAR  
TABELAS COM  
USUARIO  
POSTGREE  
  
CREATE TABLE empregado (  
    SSN int PRIMARY KEY  
    ,pnome varchar(20)  
    ,minicial varchar(50)  
    ,unome int  
    ,datanasc date  
    ,endereco varchar(200)  
    ,sexo varchar(1)  
    ,salario real  
    ,superssn int  
    ,dno int  
);  
  
CREATE TABLE departamento (  
    dnumero int PRIMARY KEY  
    ,dnome varchar(20)  
    ,gerssn int  
    ,gerdatainicio date  
);  
  
CREATE TABLE depto_localizacoes (  
    dlocalizacao int PRIMARY KEY  
    ,dnumero int
```

```
);
```

```
CREATE TABLE projeto (  
    pnumero int PRIMARY KEY  
    ,pjnome varchar(20)  
    ,dnum int  
    ,plocalizacao varchar(200)  
);
```

```
CREATE TABLE trabalha_em (  
    pno int PRIMARY KEY  
    ,essn int  
    ,horas time  
);
```

```
CREATE TABLE dependente (  
    essn int  
    ,nome_dependente varchar(20) PRIMARY KEY  
    ,datanasc date  
    ,parentesco varchar(50)  
    ,sexo varchar(1)  
);
```

```
-----
```

```
--CHAVES ESTRANJEIRAS
```

```
ALTER TABLE depto_localizacoes  
ADD CONSTRAINT chavedlocalizacao FOREIGN KEY (dnumero) REFERENCES  
departamento(dnumero);
```

```
ALTER TABLE trabalha_em  
ADD CONSTRAINT chavessn FOREIGN KEY (essn) REFERENCES empregado(ssn);
```

```
ALTER TABLE trabalha_em  
ADD CONSTRAINT dependente FOREIGN KEY (essn) REFERENCES empregado(ssn);
```

```
-----
```

```
--INSERIR DADOS NAS TABELAS
```

```

INSERT INTO empregado (SSN, pnome, minicial, unome, datanasc, endereco, sexo,
salario, superssn, dno) VALUES (123456789, 'João', 'A', 1, '1990-01-01', 'Rua
A, 123', 'M', 3000, NULL, 1);
INSERT INTO empregado (SSN, pnome, minicial, unome, datanasc, endereco, sexo,
salario, superssn, dno) VALUES (987654321, 'Maria', 'B', 2, '1985-05-10',
'Rua B, 456', 'F', 2500, 123456789, 1);
INSERT INTO empregado (SSN, pnome, minicial, unome, datanasc, endereco, sexo,
salario, superssn, dno) VALUES (345678901, 'Pedro', 'C', 3, '1992-11-20',
'Rua C, 789', 'M', 2800, 123456789, 2);
INSERT INTO empregado (SSN, pnome, minicial, unome, datanasc, endereco, sexo,
salario, superssn, dno) VALUES (111111111, 'Ana', 'D', 4, '1989-03-15', 'Rua
D, 321', 'F', 3500, 123456789, 2);
INSERT INTO empregado (SSN, pnome, minicial, unome, datanasc, endereco, sexo,
salario, superssn, dno) VALUES (222222222, 'Lucas', 'E', 5, '1988-07-22',
'Rua E, 654', 'M', 3200, 234567890, 1);

```

```

INSERT INTO departamento (dnumero, dnome, gerssn, gerdatainicio) VALUES (1,
'Vendas', 123456789, '2019-01-01');
INSERT INTO departamento (dnumero, dnome, gerssn, gerdatainicio) VALUES (2,
'Recursos Humanos', 234567890, '2018-05-01');
INSERT INTO departamento (dnumero, dnome, gerssn, gerdatainicio) VALUES (3,
'Finanças', 345678901, '2017-09-01');
INSERT INTO departamento (dnumero, dnome, gerssn, gerdatainicio) VALUES (4,
'Marketing', 456789012, '2016-01-01');
INSERT INTO departamento (dnumero, dnome, gerssn, gerdatainicio) VALUES (5,
'TI', 567890123, '2015-07-01');

```

```

INSERT INTO depto_localizacoes (dlocalizacao, dnumero) VALUES (1, 1);
INSERT INTO depto_localizacoes (dlocalizacao, dnumero) VALUES (2, 2);
INSERT INTO depto_localizacoes (dlocalizacao, dnumero) VALUES (3, 3);
INSERT INTO depto_localizacoes (dlocalizacao, dnumero) VALUES (4, 4);
INSERT INTO depto_localizacoes (dlocalizacao, dnumero) VALUES (5, 5);

```

```

INSERT INTO projeto VALUES (1, 'Projeto A', 1, 'São Paulo');
INSERT INTO projeto VALUES (2, 'Projeto B', 1, 'Rio de Janeiro');
INSERT INTO projeto VALUES (3, 'Projeto C', 2, 'Belo Horizonte');
INSERT INTO projeto VALUES (4, 'Projeto D', 2, 'Brasília');
INSERT INTO projeto VALUES (5, 'Projeto E', 3, 'Curitiba');

```

```

INSERT INTO trabalha_em VALUES (1, 123456789, '08:00:00');
INSERT INTO trabalha_em VALUES (2, 987654321, '06:00:00');
INSERT INTO trabalha_em VALUES (3, 123456789, '09:00:00');
INSERT INTO trabalha_em VALUES (4, 111111111, '07:00:00');

```

```

INSERT INTO trabalha_em VALUES (5, 222222222, '08:30:00');

INSERT INTO dependente VALUES (123456789, 'João', '2005-05-20', 'Filho',
'M');
INSERT INTO dependente VALUES (123456789, 'Maria', '2010-09-15', 'Filha',
'F');
INSERT INTO dependente VALUES (987654321, 'Pedro', '2008-12-03', 'Filho',
'M');
INSERT INTO dependente VALUES (111111111, 'Ana', '1998-06-10', 'Filha', 'F');
INSERT INTO dependente VALUES (222222222, 'José', '2001-01-30', 'Filho',
'M');
-----

```

g) Escreva sentenças SQL para conceder esses privilégios, procurando identificar em cada bloco de código, qual usuário está executando. Use visões onde for apropriado.

Todo o código se encontra em:

<https://github.com/anderson-silva21/DB2/blob/main/trabalhodois.sql>

```

-- Database: anotherDB

-- DROP DATABASE IF EXISTS "anotherDB";

CREATE DATABASE "anotherDB"
WITH
  OWNER = postgres
  ENCODING = 'UTF8'
  LC_COLLATE = 'Portuguese_Brazil.1252'
  LC_CTYPE = 'Portuguese_Brazil.1252'
  TABLESPACE = pg_default
  CONNECTION LIMIT = -1
  IS_TEMPLATE = False;

-- CRIAR TABELAS COM USUARIO POSTGREE
CREATE TABLE empregado (
  SSN int PRIMARY KEY
  ,pnome varchar(20)
  ,minicial varchar(50)
  ,unome int
  ,datanasc date
  ,endereco varchar(200)
  ,sexo varchar(1)
  ,salario real
  ,superssn int
  ,dno int
);

CREATE TABLE departamento (
  dnumero int PRIMARY KEY

```

```
,dnome varchar(20)
,gerssn int
,gerdatainicio date
);
```

```
CREATE TABLE depto_localizacoes (
    dlocalizacao int PRIMARY KEY
    ,dnumero int
);
```

```
CREATE TABLE projeto (
    pnumero int PRIMARY KEY
    ,pjnome varchar(20)
    ,dnum int
    ,plocalizacao varchar(200)
);
```

```
CREATE TABLE trabalha_em (
    pno int PRIMARY KEY
    ,essn int
    ,horas time
);
```

```
CREATE TABLE dependente (
    essn int
    ,nome_dependente varchar(20) PRIMARY KEY
    ,datanasc date
    ,parentesco varchar(50)
    ,sexo varchar(1)
);
```

--CHAVES ESTRANJEIRAS

```
ALTER TABLE depto_localizacoes
ADD CONSTRAINT chavedlocalizacao FOREIGN KEY (dnumero) REFERENCES
departamento(dnumero);
```

```
ALTER TABLE trabalha_em
ADD CONSTRAINT chaveessn FOREIGN KEY (essn) REFERENCES empregado(ssn);
```

```
ALTER TABLE trabalha_em
ADD CONSTRAINT dependente FOREIGN KEY (essn) REFERENCES empregado(ssn);
```

--INSERIR DADOS NAS TABELAS

```
INSERT INTO empregado (SSN, pnome, minicial, unome, datanasc, endereco, sexo, salario, superssn, dno)
VALUES (123456789, 'João', 'A', 1, '1990-01-01', 'Rua A, 123', 'M', 3000, NULL, 1);
INSERT INTO empregado (SSN, pnome, minicial, unome, datanasc, endereco, sexo, salario, superssn, dno)
VALUES (987654321, 'Maria', 'B', 2, '1985-05-10', 'Rua B, 456', 'F', 2500, 123456789, 1);
INSERT INTO empregado (SSN, pnome, minicial, unome, datanasc, endereco, sexo, salario, superssn, dno)
VALUES (345678901, 'Pedro', 'C', 3, '1992-11-20', 'Rua C, 789', 'M', 2800, 123456789, 2);
INSERT INTO empregado (SSN, pnome, minicial, unome, datanasc, endereco, sexo, salario, superssn, dno)
VALUES (111111111, 'Ana', 'D', 4, '1989-03-15', 'Rua D, 321', 'F', 3500, 123456789, 2);
INSERT INTO empregado (SSN, pnome, minicial, unome, datanasc, endereco, sexo, salario, superssn, dno)
VALUES (222222222, 'Lucas', 'E', 5, '1988-07-22', 'Rua E, 654', 'M', 3200, 234567890, 1);
```

```

INSERT INTO departamento (dnumero, dnome, gerssn, gerdatainicio) VALUES (1, 'Vendas', 123456789, '2019-01-01');
INSERT INTO departamento (dnumero, dnome, gerssn, gerdatainicio) VALUES (2, 'Recursos Humanos', 234567890, '2018-05-01');
INSERT INTO departamento (dnumero, dnome, gerssn, gerdatainicio) VALUES (3, 'Finanças', 345678901, '2017-09-01');
INSERT INTO departamento (dnumero, dnome, gerssn, gerdatainicio) VALUES (4, 'Marketing', 456789012, '2016-01-01');
INSERT INTO departamento (dnumero, dnome, gerssn, gerdatainicio) VALUES (5, 'TI', 567890123, '2015-07-01');

```

```

INSERT INTO depto_localizacoes (dlocalizacao, dnumero) VALUES (1, 1);
INSERT INTO depto_localizacoes (dlocalizacao, dnumero) VALUES (2, 2);
INSERT INTO depto_localizacoes (dlocalizacao, dnumero) VALUES (3, 3);
INSERT INTO depto_localizacoes (dlocalizacao, dnumero) VALUES (4, 4);
INSERT INTO depto_localizacoes (dlocalizacao, dnumero) VALUES (5, 5);

```

```

INSERT INTO projeto VALUES (1, 'Projeto A', 1, 'São Paulo');
INSERT INTO projeto VALUES (2, 'Projeto B', 1, 'Rio de Janeiro');
INSERT INTO projeto VALUES (3, 'Projeto C', 2, 'Belo Horizonte');
INSERT INTO projeto VALUES (4, 'Projeto D', 2, 'Brasília');
INSERT INTO projeto VALUES (5, 'Projeto E', 3, 'Curitiba');

```

```

INSERT INTO trabalha_em VALUES (1, 123456789, '08:00:00');
INSERT INTO trabalha_em VALUES (2, 987654321, '06:00:00');
INSERT INTO trabalha_em VALUES (3, 123456789, '09:00:00');
INSERT INTO trabalha_em VALUES (4, 111111111, '07:00:00');
INSERT INTO trabalha_em VALUES (5, 222222222, '08:30:00');

```

```

INSERT INTO dependente VALUES (123456789, 'João', '2005-05-20', 'Filho', 'M');
INSERT INTO dependente VALUES (123456789, 'Maria', '2010-09-15', 'Filha', 'F');
INSERT INTO dependente VALUES (987654321, 'Pedro', '2008-12-03', 'Filho', 'M');
INSERT INTO dependente VALUES (111111111, 'Ana', '1998-06-10', 'Filha', 'F');
INSERT INTO dependente VALUES (222222222, 'José', '2001-01-30', 'Filho', 'M');

```

-- CRIAR USUÁRIOS COM USUARIO POSTGREE

```

create user userA encrypted password '123'; -- pode recuperar ou modificar quaisquer relações exceto dependentes e pode conceder privilégios para outros usuários.
create user userB encrypted password '123'; -- pode recuperar todos os atributos de empregado e departamento, exceto salario, o numero de seguro social do gerente e a data de inicio do gerente.
create user userC encrypted password '123'; -- pode recuperar ou modificar TRABALHA_EM, porem, pode apenas recuperar os atributos PNOOME, INICIALM, UNOME e NUMEROD0SEGUROS0SOCIAL de EMPREGADO e os atributos NOMEPE e NUMEROP de PROJETO
create user userD encrypted password '123'; -- pode recuperar qualquer atributo de EMPREGADO ou dependente e pode modificar DEPENDENTE.
create user userE encrypted password '123'; -- Pode recuperar qualquer atributo de EMPREGADO, porém, somente para as tuplas de EMPREGADO que possuam DNO = 3.

```

```

SELECT * FROM pg_catalog.pg_user;
alter user postgres;
--a
GRANT SELECT, INSERT, UPDATE, DELETE ON empregado, departamento, depto_localizacoes, projeto, trabalha_em TO userA WITH GRANT OPTION;
--teste
set role userA;

```

```
INSERT INTO empregado (  
    SSN  
    ,pnome  
    ,minicial  
    ,unome  
    ,datanasc  
    ,endereco  
    ,sexo  
    ,salario  
    ,superssn  
    ,dno  
)VALUES(  
    1,  
    'mari',  
    'teste',  
    1,  
    '2001-09-28',  
    'apucarana',  
    'f',  
    10000,  
    1,  
    3  
);
```

```
INSERT INTO empregado (  
    SSN  
    ,pnome  
    ,minicial  
    ,unome  
    ,datanasc  
    ,endereco  
    ,sexo  
    ,salario  
    ,superssn  
    ,dno  
)VALUES(  
    1,  
    'caio',  
    'teste',  
    1,  
    '2001-09-29',  
    'apucarana',  
    'm',  
    100,  
    1,  
    1  
);
```

```
--b  
GRANT SELECT (ssn, pnome, minicial, unome, datanasc, endereco, sexo, superssn, dno) ON empregado  
TO userB;  
GRANT SELECT (dnumero, dnome) ON departamento TO userB;  
--teste
```



```
set role userB;
```

```
select ssn from empregado;  
select salario from empregado;
```

```
--c
```

```
GRANT SELECT, INSERT, UPDATE, DELETE ON trabalha_em TO userC WITH GRANT OPTION;  
GRANT SELECT (PNOME, MINICIAL, UNOME, SSN) ON empregado TO userC WITH GRANT  
OPTION;
```

```
GRANT SELECT (PJNOME, PNUMERO) ON projeto TO userC WITH GRANT OPTION;
```

```
--teste
```

```
set role userC;
```

```
select PNOME, MINICIAL, UNOME, SSN from empregado;
```

```
select * from trabalha_em;
```

```
select PJNOME, PNUMERO from projeto;
```

```
--d
```

```
GRANT SELECT ON empregado TO userD;
```

```
GRANT SELECT, INSERT, UPDATE, DELETE ON dependente TO userD;
```

```
--teste
```

```
set role userD;
```

```
select * from empregado;
```

```
select * from dependente;
```

```
insert into dependente (essn, nome_dependente, datanasc, parentesco, sexo)
```

```
values (5, 'jorge', '1998-05-06', 'filho', 'm');
```

```
--e
```

```
CREATE VIEW empregado_dno_3 AS SELECT * FROM empregado WHERE DNO = 3;
```

```
GRANT SELECT ON empregado_dno_3 TO userE;
```

```
--teste
```

```
set role userE;
```

```
select * from empregado_dno_3;
```