

## Atividade de BIGDATA

Aluno: Augusto Miguel Faustino Dos Santos

```
-- 1
--A;
INSERT INTO EMPREGADO (ssn, pnome, inicialm, unome, sexo, endereco, datanasc,
superssn, dno, salario)
VALUES ('943775543', 'Roberto', 'F', 'Silva', 'M', 'Rua X, 22 - Araucária -
PR', '1952-06-21', '888665555', '1', 58000.00);
```

```
1 -- 1
2 --A
3 INSERT INTO EMPREGADO (ssn, pnome, inicialm, unome, sexo, endereco, datanasc, superssn, dno, salario)
4 VALUES ('943775543', 'Roberto', 'F', 'Silva', 'M', 'Rua X, 22 - Araucária - PR', '1952-06-21', '888665555', '1', 58000.00);
5
```

Results Chart Export ▾

Success. No rows returned

```
--B;
INSERT INTO PROJETO (pnumero, pjnome, plocal, dnum)
VALUES ('4', 'ProdutoA', 'Araucaria', '2');
```

```
1 --B;
2
3 INSERT INTO PROJETO (pnumero, pjnome, plocal, dnum)
4
5 VALUES ('4', 'ProdutoA', 'Araucaria', '2');
6
```

Results Chart Export ▾

ERROR: 23503: insert or update on table "projeto" violates foreign key constraint "fk\_projeto\_departamento"  
DETAIL: Key (dnum)=(2) is not present in table "departamento".

```
--C;
INSERT INTO DEPARTAMENTO (dnumero, dnome, gerssn, gerdatainicio)
VALUES ('4', 'Produção', '943775543', '1998-10-01');
```

```
4 --C;
5 INSERT INTO DEPARTAMENTO (dnumero, dnome, gerssn, gerdatainicio)
6 VALUES ('4', 'Produção', '943775543', '1998-10-01');
```

Results Chart Export ▾

ERROR: 23505: duplicate key value violates unique constraint "departamento\_pkey"  
DETAIL: Key (dnumero)=(4) already exists.

```
--D
INSERT INTO TRABALHA (essn, pno, horas)
VALUES ('677678989', null, 40.0);
```

```
3 --D
4 INSERT INTO TRABALHA (essn, pno, horas)
5 VALUES ('677678989', null, 40.0);
```

Results Chart Export ▾

ERROR: 23502: null value in column "pno" of relation "trabalha" violates not-null constraint  
DETAIL: Failing row contains (677678989, null, 40).

--E

```
INSERT INTO DEPENDENTE (essn, nomedep, sexodep, datanascdep, parentesco)
VALUES ('453453453', 'Joao', 'M', '1970-12-12', 'CONJUGE');
```

```
6 INSERT INTO DEPENDENTE (essn, nomedep, sexodep, datanascdep, parentesco)
7 VALUES ('453453453', 'Joao', 'M', '1970-12-12', 'CONJUGE');
```

Results Chart Export ▾

Success. No rows returned

--F

```
DELETE FROM TRABALHA
WHERE essn = '333445555';
```

```
5 --F
6 DELETE FROM TRABALHA
7 WHERE essn = '333445555';
```

Results Chart Export ▾

Success. No rows returned

--G

```
DELETE FROM TRABALHA
WHERE essn = '333445555';
```

```
7 --G
8 DELETE FROM TRABALHA
9 WHERE essn = '333445555';
```

Results Chart Export ▾

Success. No rows returned

--H

```
DELETE FROM PROJETO
```

```
WHERE pjnome = 'ProdutoX';
```

```
5 --H
6 DELETE FROM PROJETO
7 WHERE pjnome = 'ProdutoX';
```

Results Chart Export ▾

✓ 📄 ❤️ ≡ Source Primary Database ▾

ERROR: 23503: update or delete on table "projeto" violates foreign key constraint "fk\_trabalha\_projeto" on table "trabalha"  
DETAIL: Key (pnumero)=(1) is still referenced from table "trabalha".

```
--I
```

```
UPDATE DEPARTAMENTO
```

```
SET gerssn = '123456789', gerdatainicio = '1999-01-10'
```

```
WHERE dnumero = '5';
```

```
5 --I
6 UPDATE DEPARTAMENTO
7 SET gerssn = '123456789', gerdatainicio = '1999-01-10'
8 WHERE dnumero = '5';
```

Results Chart Export ▾

Success. No rows returned

```
--J
```

```
UPDATE EMPREGADO
```

```
SET superssn = '943775543'
```

```
WHERE ssn = '999887777';
```

```
5 --J
6 UPDATE EMPREGADO
7 SET superssn = '943775543'
8 WHERE ssn = '999887777';
```

Results Chart Export ▾

Success. No rows returned

```
--K
```

```
UPDATE TRABALHA
```

```
SET horas = 5.0
```

```
WHERE essn = '999887777' AND pno = '10';
```

```
5  --K
6  UPDATE TRABALHA
7  SET horas = 5.0
8  WHERE essn = '999887777' AND pno = '10';
```

Results Chart Export ▾

Success. No rows returned

--2 escreva o comando DML adequado que:

-- A ) Mostre o número de segurança social do empregado, o nome do dependente e o

-- parentesco, ordenado por ssn ascendente e parentesco descendente.

```
SELECT essn AS ssn, nomedep, parentesco
```

```
FROM DEPENDENTE
```

```
ORDER BY essn ASC, parentesco DESC;
```

```

4  --2 A
5  SELECT essn AS ssn, nomedep, parentesco
6  FROM DEPENDENTE
7  ORDER BY essn ASC, parentesco DESC;

```

Results Chart Export ▾

ssn	nomedep	parentesco
"123456789"	"Michel"	"FILHO"
"123456789"	"Alice"	"FILHA"
"123456789"	"Elizabete"	"CONJUGE"
"333445555"	"Teodoro"	"FILHO"
"333445555"	"Alice"	"FILHA"
"333445555"	"Joana"	"CONJUGE"
"453453453"	"Joao"	"CONJUGE"
"987654321"	"Abdala"	"CONJUGE"

```

-- B) Mostre o nome e endereço de todos os empregados que trabalham para
-- o departamento Pesquisa.

```

```

SELECT e.pnome, e.unome, e.endereco
FROM EMPREGADO e
JOIN DEPARTAMENTO d ON e.dno = d.dnumero
WHERE d.dnome = 'Pesquisa';

```

```

4  --2 b
5  SELECT e.pnome, e.unome, e.endereco
6  FROM EMPREGADO e
7  JOIN DEPARTAMENTO d ON e.dno = d.dnumero
8  WHERE d.dnome = 'Pesquisa';

```

Results Chart Export ▾

pnome	unome	endereco
"Joao"	"Souza"	"R. 24 de maio, 1500 - Curitiba - PR"
"Fabio"	"Will"	"R. Alagoas, 325 - Curitiba - PR"
"Ricardo"	"Nantes"	"Av. Figueira, 55 - Almirante Tamandare - PR"
"Jussara"	"Pereira"	"Trav. da Lapa, 23 - Curitiba - PR"

-- c) Para todo projeto localizado em Araucaria, liste o nome do projeto, o nome do departamento  
 -- de controle e o último nome, endereço e data de nascimento do gerente do departamento.

```

SELECT
    p.pjnome AS nome_projeto,
    d.dnome AS nome_departamento,
    e.unome AS ultimo_nome_gerente,
    e.endereco AS endereco_gerente,
    e.datanasc AS data_nascimento_gerente
FROM
    PROJETO p
JOIN
    DEPARTAMENTO d ON p.dnum = d.dnumero
JOIN
    EMPREGADO e ON d.gerssn = e.ssn
WHERE
    p.plocal = 'Araucaria';

```

```

4  --2 c
5  SELECT
6      p.pjnome AS nome_projeto,
7      d.dnome AS nome_departamento,
8      e.unome AS ultimo_nome_gerente,
9      e.endereco AS endereco_gerente,
10     e.datanasc AS data_nascimento_gerente
11 FROM
12     PROJETO p
13 JOIN
14     DEPARTAMENTO d ON p.dnum = d.dnumero
15 JOIN
16     EMPREGADO e ON d.gerssn = e.ssn
17 WHERE
18     p.plocal = 'Araucaria';

```

Results Chart Export ▾

nome_projeto	nome_departar	ultimo_nome_g	endereco_gerente	data_nascimer
"ProdutoY"	"Pesquisa"	"Souza"	"R. 24 de maio, 1500 - Curitiba - PR"	"1965-01-09"

-- D) Recupere os nomes de todos os empregados que trabalhem mais de 10 horas por semana

-- no projeto Automizacao.

```

SELECT
    e.pnome, e.unome
FROM
    EMPREGADO e
JOIN
    TRABALHA t ON e.ssn = t.ssn
JOIN
    PROJETO p ON t.pno = p.pnumero
WHERE
    p.pjnome = 'Automizacao' AND t.horas > 10;

```

```

4  --2 d
5  SELECT
6  |  | e.pnome, e.unome
7  FROM
8  |  | EMPREGADO e
9  JOIN
10 |  | TRABALHA t ON e.ssn = t.essn
11 JOIN
12 |  | PROJETO p ON t.pno = p.pnumero
13 WHERE
14 |  | p.pjnome = 'Automatizacao' AND t.horas > 10;

```

Results Chart Export ▾

pnome	unome	
"Alberto"	"Medeiros"	

-- E) Mostre o nome dos empregados que têm os três maiores salários.

```

SELECT pnome, inicialm, unome
FROM EMPREGADO
ORDER BY salario DESC
LIMIT 3;

```

```

4  --2 e
5  SELECT pnome, inicialm, unome
6  FROM EMPREGADO
7  ORDER BY salario DESC
8  LIMIT 3;

```

Results Chart Export ▾

pnome	inicialm	unome	
"Roberto"	"F"	"Silva"	
"Joaquim"	"E"	"Brito"	
"Jennifer"	"S"	"Wallace"	

-- F) Mostre o nome e o salário dos supervisores com aumento de 20%.



```
SELECT e.pnome, e.unome, e.salario * 1.20 AS novo_salario
FROM EMPREGADO e
JOIN DEPARTAMENTO d ON e.ssn = d.gerssn;
```

```
4 --2 F
5 SELECT e.pnome, e.unome, e.salario * 1.20 AS novo_salario
6 FROM EMPREGADO e
7 JOIN DEPARTAMENTO d ON e.ssn = d.gerssn;
```

Results Chart Export ▾

pnome	unome	novo_salario
"Jennifer"	"Wallace"	51600
"Joaquim"	"Brito"	66000
"Joao"	"Souza"	36000

-- G) Mostre os departamentos e suas localidades.

```
SELECT d.dnome AS nome_departamento, l.dlocalizacao AS localizacao
FROM DEPARTAMENTO d
JOIN LOCALIZACAO l ON d.dnumero = l.dnum;
```

```
4 --2 G
5 SELECT d.dnome AS nome_departamento, l.dlocalizacao AS localizacao
6 FROM DEPARTAMENTO d
7 JOIN LOCALIZACAO l ON d.dnumero = l.dnum;
```

Results Chart Export ▾

nome_departamento	localizacao
"Sede Administrativa"	"Curitiba"
"Administracao"	"Colombo"
"Pesquisa"	"Araucaria"
"Pesquisa"	"Curitiba"
"Pesquisa"	"Pinhais"

-- H) Mostre os departamentos e seus projetos.

```
SELECT d.dnome AS nome_departamento, p.pjnome AS nome_projeto
FROM DEPARTAMENTO d
JOIN PROJETO p ON d.dnumero = p.dnum;
```

4 --2 H

```
5 SELECT d.dnome AS nome_departamento, p.pjnome AS nome_projeto
6 FROM DEPARTAMENTO d
7 JOIN PROJETO p ON d.dnumero = p.dnum;
8
```

Results Chart Export ▾

nome_departamento	nome_projeto
"Pesquisa"	"ProdutoX"
"Pesquisa"	"ProdutoY"
"Pesquisa"	"ProdutoZ"
"Administracao"	"Automatizacao"
"Sede Administrativa"	"Reorganizacao"
"Administracao"	"Novos Beneficios"

-- I) Mostre os empregados do sexo feminino e a data de nascimento cujo salário é maior que 30.000.

```
SELECT pnome, inicialm, unome, datanasc
FROM EMPREGADO
WHERE sexo = 'F' AND salario > 30000;
```

4 --2 I

```
5 SELECT pnome, inicialm, unome, datanasc
6 FROM EMPREGADO
7 WHERE sexo = 'F' AND salario > 30000;
```

Results Chart Export ▾

pnome	inicialm	unome	datanasc
"Jennifer"	"S"	"Wallace"	"1941-06-20"

```
-- J) Mostre os projetos em que o empregado 'Fábio' trabalha.  
SELECT p.pjnome AS nome_projeto  
FROM PROJETO p  
JOIN TRABALHA t ON p.pnumero = t.pno  
JOIN EMPREGADO e ON t.essn = e.ssn  
WHERE e.pnome = 'Fábio';
```

```
4  --2 j  
5  SELECT p.pjnome AS nome_projeto  
6  FROM PROJETO p  
7  JOIN TRABALHA t ON p.pnumero = t.pno  
8  JOIN EMPREGADO e ON t.essn = e.ssn  
9  WHERE e.pnome = 'Fábio';
```

Results Chart Export ▾

Success. No rows returned

```
--3) Joins: Dadas as tabelas A e B e seus registros (abaixo):  
-- 3 registros para a empresa 1 (que existe na tabela A);  
SELECT A.CODIGO, A.NOME, B.ID, B.VALOR  
FROM TABELA_A A  
INNER JOIN TABELA_B B ON A.CODIGO = B.CODIGO;
```

```
SELECT A.CODIGO, A.NOME, B.ID, B.VALOR  
FROM TABELA_A A  
LEFT OUTER JOIN TABELA_B B ON A.CODIGO = B.CODIGO;
```

```
SELECT A.CODIGO, A.NOME, B.ID, B.VALOR  
FROM TABELA_A A  
RIGHT OUTER JOIN TABELA_B B ON A.CODIGO = B.CODIGO;
```

```
SELECT A.CODIGO, A.NOME, B.ID, B.VALOR  
FROM TABELA_A A  
FULL OUTER JOIN TABELA_B B ON A.CODIGO = B.CODIGO;
```

```
-- 2 registros para a empresa 2 (que existe na tabela A);  
SELECT A.CODIGO, A.NOME, B.ID, B.VALOR  
FROM TABELA_A A  
INNER JOIN TABELA_B B ON A.CODIGO = B.CODIGO;
```

```
SELECT A.CODIGO, A.NOME, B.ID, B.VALOR
FROM TABELA_A A
LEFT OUTER JOIN TABELA_B B ON A.CODIGO = B.CODIGO;

SELECT A.CODIGO, A.NOME, B.ID, B.VALOR
FROM TABELA_A A
RIGHT OUTER JOIN TABELA_B B ON A.CODIGO = B.CODIGO;

SELECT A.CODIGO, A.NOME, B.ID, B.VALOR
FROM TABELA_A A
FULL OUTER JOIN TABELA_B B ON A.CODIGO = B.CODIGO;

-- 1 registros para a empresa 3 (que existe na tabela A);
SELECT A.CODIGO, A.NOME, B.ID, B.VALOR
FROM TABELA_A A
INNER JOIN TABELA_B B ON A.CODIGO = B.CODIGO;

SELECT A.CODIGO, A.NOME, B.ID, B.VALOR
FROM TABELA_A A
LEFT OUTER JOIN TABELA_B B ON A.CODIGO = B.CODIGO;

SELECT A.CODIGO, A.NOME, B.ID, B.VALOR
FROM TABELA_A A
RIGHT OUTER JOIN TABELA_B B ON A.CODIGO = B.CODIGO;

SELECT A.CODIGO, A.NOME, B.ID, B.VALOR
FROM TABELA_A A
FULL OUTER JOIN TABELA_B B ON A.CODIGO = B.CODIGO;

-- 0 registros para a empresa 4 (que existe na tabela A);
SELECT A.CODIGO, A.NOME, B.ID, B.VALOR
FROM TABELA_A A
INNER JOIN TABELA_B B ON A.CODIGO = B.CODIGO;

SELECT A.CODIGO, A.NOME, B.ID, B.VALOR
FROM TABELA_A A
LEFT OUTER JOIN TABELA_B B ON A.CODIGO = B.CODIGO;

SELECT A.CODIGO, A.NOME, B.ID, B.VALOR
FROM TABELA_A A
RIGHT OUTER JOIN TABELA_B B ON A.CODIGO = B.CODIGO;

SELECT A.CODIGO, A.NOME, B.ID, B.VALOR
```

```

FROM TABELA_A A
FULL OUTER JOIN TABELA_B B ON A.CODIGO = B.CODIGO;
-- 1 registros para a empresa 5 (que existe na tabela A);

SELECT A.CODIGO, A.NOME, B.ID, B.VALOR
FROM TABELA_A A
INNER JOIN TABELA_B B ON A.CODIGO = B.CODIGO;

SELECT A.CODIGO, A.NOME, B.ID, B.VALOR
FROM TABELA_A A
LEFT OUTER JOIN TABELA_B B ON A.CODIGO = B.CODIGO;

SELECT A.CODIGO, A.NOME, B.ID, B.VALOR
FROM TABELA_A A
RIGHT OUTER JOIN TABELA_B B ON A.CODIGO = B.CODIGO;

SELECT A.CODIGO, A.NOME, B.ID, B.VALOR
FROM TABELA_A A
FULL OUTER JOIN TABELA_B B ON A.CODIGO = B.CODIGO;

-- 1 registros para a empresa 8 (que NÃO existe na tabela A)
SELECT A.CODIGO, A.NOME, B.ID, B.VALOR
FROM TABELA_A A
INNER JOIN TABELA_B B ON A.CODIGO = B.CODIGO;

SELECT A.CODIGO, A.NOME, B.ID, B.VALOR
FROM TABELA_A A
LEFT OUTER JOIN TABELA_B B ON A.CODIGO = B.CODIGO;

SELECT A.CODIGO, A.NOME, B.ID, B.VALOR
FROM TABELA_A A
RIGHT OUTER JOIN TABELA_B B ON A.CODIGO = B.CODIGO;

SELECT A.CODIGO, A.NOME, B.ID, B.VALOR
FROM TABELA_A A
FULL OUTER JOIN TABELA_B B ON A.CODIGO = B.CODIGO;

-- 4) Operações CRUD (Complementar) Suponha que cada uma das seguintes
operações de atualização seja aplicada
-- diretamente ao estado do banco de dados "empresa", apresentado na Atividade
1. Escreva o comando DML adequado que:

-- a) Encontre todos os empregados cujo supervisor esteja alocado em um
departamento

```

```
-- diferente. Mostre o número de segurança social e o nome completo do
empregado.
SELECT e.ssn, e.pnome, e.inicialm, e.unome
FROM EMPREGADO e
JOIN EMPREGADO s ON e.superssn = s.ssn
WHERE e.dno <> s.dno; -- Filtra os empregados cujo departamento é diferente
do departamento do supervisor
```

```
4 --3 A|
5 SELECT e.ssn, e.pnome, e.inicialm, e.unome
6 FROM EMPREGADO e
7 JOIN EMPREGADO s ON e.superssn = s.ssn
8 WHERE e.dno <> s.dno;
```

Results Chart Export ▾

ssn	pnome	inicialm	unome
"333445555"	"Fabio"	"T"	"Will"
"987654321"	"Jennifer"	"S"	"Wallace"
"999887777"	"Alice"	"J"	"Zebra"

```
-- b) Mostre o número de segurança social do empregado, o nome do dependente o
e
-- parentesco, ordenado por ssn ascendente e parentesco descendente.
SELECT e.ssn, d.nomedep, d.parentesco
FROM EMPREGADO e
JOIN DEPENDENTE d ON e.ssn = d.essn
ORDER BY e.ssn ASC, d.parentesco DESC;
```

```

4  --3 B
5  SELECT e.ssn, d.nomedep, d.parentesco
6  FROM EMPREGADO e
7  JOIN DEPENDENTE d ON e.ssn = d.essn
8  ORDER BY e.ssn ASC, d.parentesco DESC;

```

Results Chart Export ▾

ssn	nomedep	parentesco
"123456789"	"Michel"	"FILHO"
"123456789"	"Alice"	"FILHA"
"123456789"	"Elizabete"	"CONJUGE"
"333445555"	"Teodoro"	"FILHO"
"333445555"	"Alice"	"FILHA"
"333445555"	"Joana"	"CONJUGE"
"453453453"	"Joao"	"CONJUGE"
"987654321"	"Abdala"	"CONJUGE"

-- c) Encontre os nomes de todos os empregados que são diretamente supervisionados por "Joaquim E Brito".

```

SELECT e.pnome, e.inicialm, e.unome
FROM EMPREGADO e
JOIN EMPREGADO s ON e.superssn = s.ssn
WHERE s.pnome = 'Joaquim' AND s.unome = 'Brito';

```

```

4  --3 C
5  SELECT e.pnome, e.inicialm, e.unome
6  FROM EMPREGADO e
7  JOIN EMPREGADO s ON e.superssn = s.ssn
8  WHERE s.pnome = 'Joaquim' AND s.unome = 'Brito';

```

Results Chart Export ▾

pnome	inicialm	unome
"Fabio"	"T"	"Will"
"Jennifer"	"S"	"Wallace"
"Roberto"	"F"	"Silva"

-- d) Faça uma lista dos números e nomes de projetos que envolvam um empregado cujo último nome seja 'Will', mesmo que esse trabalhador seja o gerente do departamento que controla o projeto.

```

SELECT p.pnumero, p.pjnome
FROM PROJETO p
JOIN TRABALHA t ON p.pnumero = t.pno
JOIN EMPREGADO e ON t.essn = e.ssn
WHERE e.unome = 'Will'
       OR p.dnum IN (SELECT dnumero FROM DEPARTAMENTO WHERE gerssn IN (SELECT ssn
FROM EMPREGADO WHERE unome = 'Will'));

```

```

4  --3 D
5  SELECT p.pnumero, p.pjnome
6  FROM PROJETO p
7  JOIN TRABALHA t ON p.pnumero = t.pno
8  JOIN EMPREGADO e ON t.essn = e.ssn
9  WHERE e.unome = 'Will'
10 | OR p.dnum IN (SELECT dnumero FROM DEPARTAMENTO WHERE gerssn IN (SELECT ssn FROM EMPREGADO WHERE unome = 'Will'));

```

Results Chart Export ▾

Source Prim

Success. No rows returned

-- e) Encontre o nome dos empregados que trabalham em algum dos projetos controlados pelo departamento número 5.

```

SELECT DISTINCT e.pnome, e.inicialm, e.unome

```



```

FROM EMPREGADO e
JOIN TRABALHA t ON e.ssn = t.essn
JOIN PROJETO p ON t.pno = p.pnumero
WHERE p.dnum::integer = 5;

```

```

4  --3 E
5  SELECT DISTINCT e.pnome, e.inicialm, e.unome
6  FROM EMPREGADO e
7  JOIN TRABALHA t ON e.ssn = t.essn
8  JOIN PROJETO p ON t.pno = p.pnumero
9  WHERE p.dnum::integer = 5;

```

Results Chart Export ▾

pnome	inicialm	unome	
"Joao"	"B"	"Souza"	
"Jussara"	"A"	"Pereira"	
"Ricardo"	"K"	"Nantes"	

-- f) Encontre o nome e o endereço de todos os empregados que trabalhem em pelo menos um projeto localizado em 'Curitiba', mas cujo departamento não se localiza em 'Curitiba'.

```

SELECT DISTINCT e.pnome, e.unome, e.endereco
FROM EMPREGADO e
JOIN TRABALHA t ON e.ssn = t.essn
JOIN PROJETO p ON t.pno = p.pnumero
JOIN DEPARTAMENTO d ON e.dno = d.dnumero
JOIN LOCALIZACAO l ON d.dnumero = l.dnum
WHERE p.plocal = 'Curitiba'
AND l.dlocalizacao <> 'Curitiba';

```

```

4  --3 F
5  SELECT DISTINCT e.pnome, e.unome, e.endereco
6  FROM EMPREGADO e
7  JOIN TRABALHA t ON e.ssn = t.essn
8  JOIN PROJETO p ON t.pno = p.pnumero
9  JOIN DEPARTAMENTO d ON e.dno = d.dnumero
10 JOIN LOCALIZACAO l ON d.dnumero = l.dnum
11 WHERE p.plocal = 'Curitiba'
12 | AND l.dlocalizacao <> 'Curitiba';

```

Results Chart Export ▾

pnome	unome	endereco
"Jennifer"	"Wallace"	"R. Berry, 291 - Colombo - PR"
"Ricardo"	"Nantes"	"Av. Figueira, 55 - Almirante Tamandare - PR"

-- 5) Funções agregadas e nativas - Utilize funções agregadas e/ou nativas SQL para responder as seguintes questões:

-- a) Recupere a média salarial de todos os empregados do sexo feminino.

```

SELECT AVG(salario) AS media_salarial
FROM EMPREGADO
WHERE sexo = 'F';

```

```

4  SELECT AVG(salario) AS media_salarial
5  FROM EMPREGADO
6  WHERE sexo = 'F';

```

Results Chart Export ▾

media_salarial
31000

-- b) Mostre o número de empregados por supervisor.

```

SELECT s.pnome AS nome_supervisor, COUNT(*) AS numero_empregados
FROM EMPREGADO e
JOIN EMPREGADO s ON e.superssn = s.ssn
GROUP BY s.pnome
ORDER BY numero_empregados DESC;

```

```

4 SELECT s.pnome AS nome_supervisor, COUNT(*) AS numero_employees
5 FROM EMPREGADO e
6 JOIN EMPREGADO s ON e.superssn = s.ssn
7 GROUP BY s.pnome
8 ORDER BY numero_employees DESC;
9

```

Results Chart Export ▾

nome_supervisor	numero_employees
"Joaquim"	3
"Fabio"	3
"Roberto"	1
"Jennifer"	1

-- c) Mostre o maior número de horas envolvido em projetos.

```

SELECT MAX(horas) AS maior_numero_horas
FROM TRABALHA;

```

```

4 SELECT MAX(horas) AS maior_numero_horas
5 FROM TRABALHA;
6

```

Results Chart Export ▾

maior_numero_
40

-- d) Para cada projeto, liste o nome do projeto e o total de horas por semana (de todos os

-- empregados) gastas no projeto.

```

SELECT p.pjnome AS nome_projeto, SUM(t.horas) AS total_horas_semana
FROM PROJETO p
JOIN TRABALHA t ON p.pnumero = t.pno
GROUP BY p.pjnome;

```

```

4 SELECT p.pjnome AS nome_projeto, SUM(t.horas) AS total_horas_semana
5 FROM PROJETO p
6 JOIN TRABALHA t ON p.pnumero = t.pno
7 GROUP BY p.pjnome;
8
9

```

Results Chart Export ▾

nome_projeto	total_horas_s
"Automatizacao"	40
"ProdutoZ"	40
"ProdutoY"	27.5
"Novos Beneficios"	55
"ProdutoX"	52.5
"Reorganizacao"	15

-- e) Para cada departamento, recupere o nome do departamento e a média salarial de todos os empregados que trabalham nesse departamento.

```

SELECT d.dnome AS nome_departamento, AVG(e.salario) AS media_salarial
FROM DEPARTAMENTO d
JOIN EMPREGADO e ON d.dnumero = e.dno
GROUP BY d.dnome;

```

```

4 SELECT d.dnome AS nome_departamento, AVG(e.salario) AS media_salarial
5 FROM DEPARTAMENTO d
6 JOIN EMPREGADO e ON d.dnumero = e.dno
7 GROUP BY d.dnome;
8
9

```

Results Chart Export ▾

nome_departamento	media_salarial
"Administracao"	31000
"Pesquisa"	33250
"Sede Administrativa"	56500

```
-- f) Liste os nomes de todos os empregados com dois ou mais dependentes.
SELECT e.pnome, e.inicialm, e.unome
FROM EMPREGADO e
JOIN DEPENDENTE d ON e.ssn = d.essn
GROUP BY e.ssn, e.pnome, e.inicialm, e.unome
HAVING COUNT(*) >= 2;
```

```
4 SELECT e.pnome, e.inicialm, e.unome
5 FROM EMPREGADO e
6 JOIN DEPENDENTE d ON e.ssn = d.essn
7 GROUP BY e.ssn, e.pnome, e.inicialm, e.unome
8 HAVING COUNT(*) >= 2;
9
```

Results Chart Export ▾

pnome	inicialm	unome
"Joao"	"B"	"Souza"
"Fabio"	"T"	"Will"

```
-- g) Mostre o nome do departamento que possui o menor número de projetos associados.
```

```
SELECT d.dnome AS nome_departamento
FROM DEPARTAMENTO d
LEFT JOIN PROJETO p ON d.dnumero = p.dnum
GROUP BY d.dnome
ORDER BY COUNT(p.pnumero) ASC
LIMIT 1;
```

```

4  SELECT d.dnome AS nome_departamento
5  FROM DEPARTAMENTO d
6  LEFT JOIN PROJETO p ON d.dnumero = p.dnum
7  GROUP BY d.dnome
8  ORDER BY COUNT(p.pnumero) ASC
9  LIMIT 1;
10

```

Results   Chart   Export ▾

nome_departamento
"Sede Administrativa"

-- h) Escreva uma consulta que retorne do 10o ao 22o caractere do endereço do empregado.

```

SELECT SUBSTRING(endereco, 10, 13) AS trecho_endereco
FROM EMPREGADO;

```

```

4 SELECT SUBSTRING(endereco, 10, 13) AS trecho_endereco
5 FROM EMPREGADO;
6
7

```

Results Chart Export ▾

trecho_endereco
"maio, 1500 - "
"s, 325 - Curi"
" 291 - Colomb"
"ira, 55 - Alm"
"Lapa, 23 - Cu"
" 1245 - Curit"
" 450 - Curiti"
" - Araucária "
"a, 98 - Pinha"

-- i) Escreva uma consulta que retorne apenas o mês de nascimento de cada funcionário.

```

SELECT pnome, EXTRACT(MONTH FROM datanasc) AS mes_nascimento
FROM EMPREGADO;

```

```

3 SELECT pnome, EXTRACT(MONTH FROM datanasc) AS mes_nascimento
4 FROM EMPREGADO;
5
6

```

Results Chart Export ▾

pnome	mes_nascimento
"Joao"	"1"
"Fabio"	"12"
"Jennifer"	"6"
"Ricardo"	"9"
"Jussara"	"7"
"Alberto"	"3"
"Joaquim"	"11"
"Roberto"	"6"
"Alice"	"1"

-- j) Escreva uma consulta que retorne a idade (em anos) que o empregado tinha quando o

-- dependente de parentesco filhou ou filha nasceu.

```

SELECT
    e.pnome,
    e.unome,
    d.nomedep,
    d.parentesco,
    DATE_PART('year', AGE(d.datanascdep, e.datanasc)) AS idade_empregado
FROM
    EMPREGADO e
JOIN
    DEPENDENTE d ON e.ssn = d.ssn
WHERE
    d.parentesco IN ('FILHO', 'FILHA');

```



```

3 SELECT
4     e.pnome,
5     e.unome,
6     d.nomedep,
7     d.parentesco,
8     DATE_PART('year', AGE(d.datanascdep, e.datanasc)) AS idade_empregado
9 FROM
10     EMPREGADO e
11 JOIN
12     DEPENDENTE d ON e.ssn = d.ssn
13 WHERE
14     d.parentesco IN ('FILHO', 'FILHA');
15

```

Results Chart Export ▾

pnome	unome	nomedep	parentesco	idade_emprega
"Fabio"	"Will"	"Alice"	"FILHA"	30
"Fabio"	"Will"	"Teodoro"	"FILHO"	27
"Joao"	"Souza"	"Michel"	"FILHO"	22
"Joao"	"Souza"	"Alice"	"FILHA"	23

-- k) Escreva uma consulte que conte o número de dependentes por ano de nascimento.

```

SELECT EXTRACT(YEAR FROM datanascdep) AS ano_nascimento, COUNT(*) AS
numero_dependentes
FROM DEPENDENTE
GROUP BY ano_nascimento
ORDER BY ano_nascimento;

```

```

3 SELECT EXTRACT(YEAR FROM datanascdep) AS ano_nascimento, COUNT(*) AS numero_dependentes
4 FROM DEPENDENTE
5 GROUP BY ano_nascimento
6 ORDER BY ano_nascimento;
7

```

Results Chart Export ▾

ano_nascimento	numero_dependentes
"1942"	1
"1958"	1
"1967"	1
"1970"	1
"1983"	1
"1986"	1
"1988"	2

-- 1) Escreva uma consulta que mostre o nome de empregados supervisor que tenham 2 ou mais supervisionados.

```

SELECT s.pnome, s.unome
FROM EMPREGADO e
JOIN EMPREGADO s ON e.superssn = s.ssn
GROUP BY s.ssn, s.pnome, s.unome
HAVING COUNT(*) >= 2;

```

```

3 SELECT s.pnome, s.unome
4 FROM EMPREGADO e
5 JOIN EMPREGADO s ON e.superssn = s.ssn
6 GROUP BY s.ssn, s.pnome, s.unome
7 HAVING COUNT(*) >= 2;
8

```

Results Chart Export ▾

pnome	unome
"Joaquim"	"Brito"
"Fabio"	"Will"

-- m) Escreva uma consulta que mostre o valor mensal a ser pago por projeto (considere que a

```
-- coluna 'salário' de empregado é mensal).
SELECT p.pjnome AS nome_projeto, SUM(e.salario) AS valor_mensal_projeto
FROM EMPREGADO e
JOIN TRABALHA t ON e.ssn = t.essn
JOIN PROJETO p ON t.pno = p.pnumero
GROUP BY p.pjnome;
```

```
2 SELECT p.pjnome AS nome_projeto, SUM(e.salario) AS valor_mensal_projeto
3 FROM EMPREGADO e
4 JOIN TRABALHA t ON e.ssn = t.essn
5 JOIN PROJETO p ON t.pno = p.pnumero
6 GROUP BY p.pjnome;
7
```

Results Chart Export ▾

nome_projeto	valor_mensal_
"Automatizacao"	50000
"ProdutoZ"	38000
"ProdutoY"	55000
"Novos Beneficios"	93000
"ProdutoX"	55000
"Reorganizacao"	98000

-- 6) - Subconsultas - Escreva subconsultas SQL para as questes:

-- a) Recupere nome (pnome e unome) de cada um dos empregados que tenham um dependente

-- cujo primeiro nome e sexo sejam o mesmo do empregado em questão.

```
SELECT DISTINCT e.pnome, e.unome
FROM EMPREGADO e
JOIN DEPENDENTE d ON e.ssn = d.essn
WHERE e.pnome = d.nomedep
AND e.sexo = d.sexodep;
```

```

3 SELECT DISTINCT e.pnome, e.unome
4 FROM EMPREGADO e
5 JOIN DEPENDENTE d ON e.ssn = d.essn
6 WHERE e.pnome = d.nomedep
7 AND e.sexo = d.sexodep;

```

Results Chart Export ▾

Success. No rows returned

-- b) Recupere os nomes dos empregados (pnome e unome) cujos salrios so maiores que a

-- media dos salrios dos empregados do departamento 5.

```

SELECT e.pnome, e.unome
FROM EMPREGADO e
WHERE e.salario > (
    SELECT AVG(salario)
    FROM EMPREGADO
    WHERE dno = '5'
);

```

```

3 SELECT e.pnome, e.unome
4 FROM EMPREGADO e
5 WHERE e.salario > (
6     SELECT AVG(salario)
7     FROM EMPREGADO
8     WHERE dno = '5'
9 );

```

Results Chart Export ▾

pnome	unome
"Fabio"	"Will"
"Jennifer"	"Wallace"
"Ricardo"	"Nantes"
"Joaquim"	"Brito"
"Roberto"	"Silva"

```
-- c) Retorne o nmero do seguro social (SSN) de todos os empregado que
trabalham com a mesma
-- combinao (projeto, horas) em algum dos projetos em que o empregado 'Fabio
Will' (SSN= 333445555) trabalhe.
```

```
SELECT DISTINCT e.ssn
FROM EMPREGADO e
JOIN TRABALHA t ON e.ssn = t.essn
WHERE (t.pno, t.horas) IN (
    SELECT t2.pno, t2.horas
    FROM TRABALHA t2
    WHERE t2.essn = '333445555' -- SSN do 'Fabio Will'
)
AND e.ssn <> '333445555'; -- Exclui o próprio 'Fabio Will'
```

```
1  SELECT DISTINCT e.ssn
2  FROM EMPREGADO e
3  JOIN TRABALHA t ON e.ssn = t.essn
4  WHERE (t.pno, t.horas) IN (
5      SELECT t2.pno, t2.horas
6      FROM TRABALHA t2
7      WHERE t2.essn = '333445555' -- SSN do 'Fabio Will'
8  )
9  AND e.ssn <> '333445555'; -- Exclui o próprio 'Fabio Will'
```

Results Chart Export ▾

Success. No rows returned

```
-- d) Recupere os nomes de todos os empregados que no trabalham em nenhum
projeto.
```

```
SELECT pnome, unome
FROM EMPREGADO
WHERE ssn NOT IN (SELECT essn FROM TRABALHA);
```

```

1 SELECT pnome, unome
2 FROM EMPREGADO
3 WHERE ssn NOT IN (SELECT essn FROM TRABALHA);
4

```

Results Chart Export ▾

pnome	unome
"Fabio"	"Will"
"Roberto"	"Silva"

-- e) Recupere o nome de empregados que no tenham dependentes.

```

SELECT pnome, unome
FROM EMPREGADO
WHERE ssn NOT IN (SELECT essn FROM DEPENDENTE);

```

```

1 SELECT pnome, unome
2 FROM EMPREGADO
3 WHERE ssn NOT IN (SELECT essn FROM DEPENDENTE);
4
5

```

Results Chart Export ▾

pnome	unome
"Ricardo"	"Nantes"
"Alberto"	"Medeiros"
"Joaquim"	"Brito"
"Roberto"	"Silva"
"Alice"	"Zebra"

-- f) Liste o ultimo nome de todos os gerentes de departamento que no tenham dependentes.

```
SELECT e.unome
FROM EMPREGADO e
JOIN DEPARTAMENTO d ON e.ssn = d.gerssn
WHERE e.ssn NOT IN (SELECT essn FROM DEPENDENTE);
```

```
1  SELECT e.unome
2  FROM EMPREGADO e
3  JOIN DEPARTAMENTO d ON e.ssn = d.gerssn
4  WHERE e.ssn NOT IN (SELECT essn FROM DEPENDENTE);
```

Results   Chart   Export ▾

unome
"Brito"

-- g) Liste os nomes dos gerentes que tenham, pelo menos, um dependente

```
SELECT DISTINCT e.pnome, e.unome
FROM EMPREGADO e
JOIN DEPARTAMENTO d ON e.ssn = d.gerssn
WHERE EXISTS (
    SELECT 1
    FROM DEPENDENTE dep
    WHERE dep.essn = e.ssn
);
```

```

1 SELECT DISTINCT e.pnome, e.unome
2 FROM EMPREGADO e
3 JOIN DEPARTAMENTO d ON e.ssn = d.gerssn
4 WHERE EXISTS (
5     SELECT 1
6     FROM DEPENDENTE dep
7     WHERE dep.essn = e.ssn
8 );

```

Results Chart Export ▾

pnome	unome
"Jennifer"	"Wallace"
"Joao"	"Souza"

-- 7) Índices - Escreva comandos SQL para responder as seguintes questes:

-- a) Crie a tabela tempEmpregado contendo dos dados dos atributos ssn, pnome, sexo, endereco  
 -- e datanasc da tabela empregado. Realizar a copia dos dados durante o processo de criaao da nova tabela.

```

CREATE TABLE tempEmpregado AS
SELECT ssn, pnome, sexo, endereco, datanasc
FROM empregado;

```















```

1 CREATE TABLE tempEmpregado AS
2 SELECT ssn, pnome, sexo, endereco, datanasc
3 FROM empregado;

```

Results Chart Export ▾

Success. No rows returned

-  departamento 
-  dependente 
-  empregado 
-  localizacao 
-  projeto 
-  tempempregado 
-  trabalha 

-- b) Utilize o comando EXPLAIN ANALYZE e anote o tempo de execuao da consulta  
 SELECT \*FROM TEMPEMPREGADO.

EXPLAIN ANALYZE



```
SELECT * FROM TEMPEMPREGADO;
```

```
4 EXPLAIN ANALYZE
5 SELECT * FROM TEMPEMPREGADO;
```

Results Chart Export ▾

#### QUERY PLAN

"Seq Scan on tempempregado (cost=0.00..12.90 rows=290 width=248) (actual time=0.012..0.014 rows=9 loops=1)"

"Planning Time: 0.153 ms"

"Execution Time: 0.077 ms"

-- c) Crie um índice com o campo ssn que no permite dados duplicados para a tabela

-- tempEmpregado, utilizando o algoritmo de ordenao BTREE.

```
CREATE UNIQUE INDEX idx_ssn_unique ON tempEmpregado (ssn);
```

--o tipo de índice B-tree é o padrão, portanto, você não precisa especificar USING BTREE explicitamente.

```
3 CREATE UNIQUE INDEX idx_ssn_unique ON tempEmpregado (ssn);
```

Results Chart Export ▾

Success. No rows returned

-- d) Crie um índice com o campo pnome para a tabela tempEmpregado, em ordem decrescente

-- e que utilize o algoritmo de ordenao HASH.

```
CREATE INDEX idx_pnome ON tempEmpregado (pnome DESC) USING HASH;
```

```
3 CREATE INDEX idx_pnome_hash ON tempEmpregado USING HASH (pnome);
```

Results Chart Export ▾

Success. No rows returned

-- e) Utilize o comando EXPLAIN ANALYSE, anote e compare com o tempo de execucao da

-- consulta SELECT \* FROM TEMPEMPREGADO. O que achou do resultado?

```
EXPLAIN ANALYZE SELECT * FROM tempEmpregado;
```

```
3 EXPLAIN ANALYZE SELECT * FROM tempEmpregado;
```

Results Chart Export ▾

#### QUERY PLAN

"Seq Scan on tempempregado (cost=0.00..1.09 rows=9 width=248) (actual time=0.008..0.009 rows=9 loops=1)"

"Planning Time: 0.317 ms"

"Execution Time: 0.078 ms"

```
-- f) Remova os indices criados nos itens c e d.  
--Remova os indices criados nos itens o campo ssn que no permite dados  
duplicados para a tabela tempEmpregado, e indice com o campo pnome para a  
tabela tempEmpregado
```

```
DROP INDEX idx_ssn ;
```

```
DROP INDEX idx_pnome;
```

```
3 DROP INDEX idx_ssn_unique;  
4 DROP INDEX idx_pnome_hash ;
```







Results Chart Export ▾

Success. No rows returned

```
-- g) Remova a tabela temporaria criada no item a.
```

```
DROP TABLE tempEmpregado;
```

```
3 DROP TABLE tempEmpregado;
```

departamento   
dependente   
empregado   
localizacao   
projeto   
trabalha 

Results Chart Export ▾

Success. No rows returned

-- 8) Visões - Escreva comandos SQL para responder as seguintes questes

-- a) Crie a viso chamada TRABALHA\_EM que dever conter os campos pnome e unome da

-- tabela empregado, o campo pjnome da tabela projeto e o campo horas da tabela trabalha

```
CREATE VIEW TRABALHA_EM AS  
SELECT e.pnome, e.unome, p.pjnome, t.horas  
FROM empregado e  
JOIN trabalha t ON e.ssn = t.essn  
JOIN projeto p ON t.pno = p.pnumero;
```

```
3 CREATE VIEW TRABALHA_EM AS  
4 SELECT e.pnome, e.unome, p.pjnome, t.horas  
5 FROM empregado e  
6 JOIN trabalha t ON e.ssn = t.essn  
7 JOIN projeto p ON t.pno = p.pnumero;
```

Results Chart Export ▾

Success. No rows returned

-- b) Crie uma consulta SQL na viso implementada no item a que retorne o ltimo e o primeiro

-- nome de todos os empregados que trabalham no 'ProdutoX'.

```
SELECT unome, pnome  
FROM TRABALHA_EM  
WHERE pjnome = 'ProdutoX';
```

```
3 SELECT unome, pnome
4 FROM TRABALHA_EM
5 WHERE pjnome = 'ProdutoX';
```

Results Chart Export ▾

unome	pnome
"Souza"	"Joao"
"Pereira"	"Jussara"

-- c) Exclua a visão criada no item a.

```
DROP VIEW TRABALHA_EM;
```

```
4
5 DROP VIEW TRABALHA_EM;
```

Results Chart Export ▾

Success. No rows returned

-- d) Crie uma visão chamada DEPTO\_INFO que deverá conter os campos dnome da tabela

-- departamento, e o total de empregados e somatório dos salários dos empregados da tabela

-- empregado por departamento

```
CREATE VIEW DEPTO_INFO AS
SELECT d.dnome,
       COUNT(e.ssn) AS total_empregados,
       SUM(e.salario) AS soma_salarios
FROM departamento d
LEFT JOIN empregado e ON d.dnumero = e.dno
GROUP BY d.dnome;
```

```

4
5 CREATE VIEW DEPTO_INFO AS
6 SELECT d.dnome,
7         COUNT(e.ssn) AS total_employees,
8         SUM(e.salario) AS soma_salarios
9 FROM departamento d
10 LEFT JOIN empregado e ON d.dnumero = e.dno
11 GROUP BY d.dnome;
12

```

Results Chart Export ▾

Success. No rows returned

-- e) Crie uma consulta SQL na visão implementada no item c que retorne a lista de informações  
-- por departamentos ordenados pelo somatório dos salários.

```

SELECT *
FROM DEPTO_INFO
ORDER BY soma_salarios DESC;

```

```

5 SELECT *
6 FROM DEPTO_INFO
7 ORDER BY soma_salarios DESC;
8
9

```

Results Chart Export ▾

dnome	total_employees	soma_salarios
"Pesquisa"	4	133000
"Sede Administrativa"	2	113000
"Administracao"	3	93000

-- f) Exclua as visões criadas nos itens a.  
DROP VIEW DEPTO\_INFO;

```

4
5 DROP VIEW DEPTO_INFO;
6
7

```

Results Chart Export ▾

Success. No rows returned