

Atividade 01

INNER JOIN

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

Mostra apenas os registros que existem em ambas as tabelas (1, 2, 3, 5).

LEFT OUTER JOIN

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

Todos os registros de A aparecem, o número 8 não aparece por não estar em B

RIGHT OUTER JOIN

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

Todos os registros de B aparecem, o número 4 não aparece por não estar em A

FULL OUTER JOIN

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

Todos os registros de ambas as tabelas aparecem

Atividade 02

- a) SELECT AVG(salario) AS media_salarial_feminino
FROM EMPREGADO
WHERE sexo = 'F';

Resultado: 31000

- b) SELECT superssn AS supervisor_ssn, COUNT(*) AS num_employees
FROM EMPREGADO
WHERE superssn IS NOT NULL
GROUP BY superssn;

Resultado: 333445555 (Fabio Will): 3 empregados
987654321 (Jennifer Wallace): 2 empregados
888665555 (Joaquim Brito): 2 empregados

- c) SELECT MAX(horas) AS max_horas
FROM TRABALHA;

Resultado: 40

- d) SELECT P.pjnome, SUM(T.horas) AS total_horas
FROM PROJETO P
JOIN TRABALHA T ON P.pnumero = T.pno
GROUP BY P.pjnome;

Resultado:

ProdutoX =	52.5
ProdutoY =	37.5
ProdutoZ =	50.0
Automatizacao =	55.0
Reorganizacao =	25.0
Novos Beneficios =	35.0

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

Resultado:

Pesquisa =	33333.33
Administracao =	31000.00
Sede Administrativa =	55000.00

f) SELECT E.pnome, E.unome
 FROM EMPREGADO E
 JOIN DEPENDENTE D ON E.ssn = D.essn
 GROUP BY E.ssn, E.pnome, E.unome
 HAVING COUNT(*) >= 2;

Resultado:

pnome	unome
Joao	Souza
Fabio	Will

g) SELECT D.dnome
 FROM DEPARTAMENTO D
 JOIN PROJETO P ON D.dnumero = P.dnum
 GROUP BY D.dnome
 ORDER BY COUNT(*) ASC
 LIMIT 1;

Resultado: Sede administrativa

h) SELECT pnome, unome, SUBSTRING(endereco FROM 10 FOR 13) AS
 trecho_endereco
 FROM EMPREGADO;

"R. 24 de maio, 1500 - Curitiba - PR" virou Joao Souza maio, 1500 - C

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

Resultado:

Joao Souza	1
Fabio Will	12
Alice Zebra	1
Jennifer Wallace	6
Ricardo Nantes	9
Jussara Pereira	7
Alberto Medeiros	3
Joaquim Brito	11

j) SELECT E.pnome, E.unome, D.nomedep,
 EXTRACT(YEAR FROM D.datanascdep) - EXTRACT(YEAR FROM E.datanasc)
 AS idade_quando_nasceu

```
FROM EMPREGADO E
JOIN DEPENDENTE D ON E.ssn = D.essn
WHERE UPPER(D.parentesco) IN ('FILHO', 'FILHA');
```

Resultado:

```
Joao Souza Michel 23
Joao Souza Alice 23
Fabio Will Teodoro 28
Fabio Will Alice 31
```

k)

```
SELECT EXTRACT(YEAR FROM datanascdep) AS ano_nasc,
      COUNT(*) AS num_dependentes
FROM DEPENDENTE
GROUP BY EXTRACT(YEAR FROM datanascdep)
ORDER BY ano_nasc;
```

Resultado:

```
1942 | 1
1958 | 1
1967 | 1
1983 | 1
1986 | 1
1988 | 2
```

i)

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

Resultado:

```
Fabio Will
Jennifer Wallace
Joaquim Brito
```

m)

```
SELECT P.pjnome,
      ROUND(SUM(E.salario * T.horas / 160), 2) AS valor_mensal_pago
FROM PROJETO P
JOIN TRABALHA T ON P.pnumero = T.pno
JOIN EMPREGADO E ON E.ssn = T.essn
GROUP BY P.pjnome;
```

Resultado:

ProdutoX	13125.00
ProdutoY	10187.50
ProdutoZ	12225.00
Automatizacao	14062.50
Reorganizacao	12343.75
Novos Beneficios	8525.00

Atividade 03

a)

```
SELECT E.pnome, E.unome
FROM EMPREGADO E
WHERE EXISTS (
    SELECT 1
    FROM DEPENDENTE D
    WHERE D.essn = E.ssn
        AND D.nome_dependente = E.pnome
        AND D.sexo = E.sexo
);
```

b)

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

c)

```
SELECT DISTINCT T1.essn
FROM TRABALHA T1
WHERE (T1.pno, T1.horas) IN (
    SELECT T2.pno, T2.horas
    FROM TRABALHA T2
    WHERE T2.essn = '333445555'
)
AND T1.essn <> '333445555';
```

d)

```
SELECT pnome, unome
FROM EMPREGADO E
WHERE NOT EXISTS (
    SELECT 1
    FROM TRABALHA T
    WHERE T.essn = E.ssn
);
```

e)

```
SELECT pnome, unome
FROM EMPREGADO E
WHERE NOT EXISTS (
    SELECT 1
    FROM DEPENDENTE D
    WHERE D.essn = E.ssn
);
```

f)

```
SELECT E.unome
FROM EMPREGADO E
JOIN DEPARTAMENTO D ON D.cpf_gerente = E.ssn
WHERE NOT EXISTS (
    SELECT 1
    FROM DEPENDENTE D2
    WHERE D2.essn = E.ssn
);
```

g)

```
SELECT E.pnome, E.unome
FROM EMPREGADO E
JOIN DEPARTAMENTO D ON D.cpf_gerente = E.ssn
WHERE EXISTS (
    SELECT 1
    FROM DEPENDENTE D2
    WHERE D2.essn = E.ssn
);
```

Atividade 04

a)

```
CREATE VIEW TRABALHA_EM AS
SELECT E.pnome, E.unome, P.pjnome, T.horas
FROM EMPREGADO E
JOIN TRABALHA T ON E.ssn = T.ssn
JOIN PROJETO P ON P.pnumero = T.pno;
```

b)

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

c)

```
DROP VIEW TRABALHA_EM;
```

d)

```
CREATE MATERIALIZED VIEW DEPTO_INFO AS
SELECT D.dnome,
       COUNT(E.ssn) AS total_employees,
       SUM(E.salario) AS soma_salarios
FROM DEPARTAMENTO D
LEFT JOIN EMPREGADO E ON D.dnumero = E.dno
GROUP BY D.dnome;
```

e)

```
SELECT *
FROM DEPTO_INFO
ORDER BY soma_salarios DESC;
```

f)

A diferença entre as visões TRABALHA_EM e DEPTO_INFO é que a TRABALHA_EM é dinâmica, ou seja, sempre mostra os dados mais recentes automaticamente. Já a DEPTO_INFO é uma visão materializada, que só atualiza os dados quando a gente roda o comando REFRESH MATERIALIZED VIEW manualmente. Quando você faz INSERT, UPDATE ou DELETE, a visão materializada não muda sozinha — precisa desta atualização manual para refletir as mudanças.

g)

```
DROP MATERIALIZED VIEW DEPTO_INFO;
```