

Considere as seguintes relações em que as chaves primárias estão sublinhadas.

Tab_Departamentos (codigoDepto, nomeDepto, gerenteDepto)

Tab_Empregados (matriculaEmp, nomeEmp, salarioEmp, deptoEmp)

Tab_Dependentes (matriculaEmp, nomeDepend, dataNascDepend, GrauParentesco)

Tab_Projetos (numeroProj, localProj, deptoProj)

Tab_HorasAlocadas (projeto, empregado, horas)

Sobre essas relações foram definidas as seguintes restrições de integridade referenciais:

Tab_Empregados.deptoEmp	Referencia	Tab_Departamentos.codigoDepto
Tab_Departamentos.gerenteDepto	Referencia	Tab_Empregados.matriculaEmp
Tab_Dependentes.matriculaEmp	Referencia	Tab_Empregados.matriculaEmp
Tab_HorasAlocadas.empregado	Referencia	Tab_Empregados.matriculaEmp
Tab_HorasAlocadas.projeto	Referencia	Tab_Projetos.numeroProj
Tab_Projetos.deptoProj	Referencia	Tab_Departamentos.codigoDepto

Elabore as seguintes consultas usando SQL:

- 1) Obter uma relação contendo a matrícula e o nome dos empregados do departamento 10.

```
SELECT matriculaEmp, nomeEmp
FROM Tab_Empregados
WHERE deptoEmp = 10;
```

- 2) Obter uma relação contendo a matrícula e o nome dos empregados do departamento “Tecnologia da Informacao” que ganham menos do que 1000 reais.

```
SELECT matricEmp, nomeEMP
FROM Tab_Empregados JOIN Tab_Departamentos ON deptoEmp = codigoDepto
WHERE salarioEmp < 1000
      nomeDepto = ‘Tecnologia da Informacao’
```

Sintaxe alternativa colocando a condição de junção na cláusula WHERE

```
SELECT matricEmp, nomeEMP
FROM Tab_Empregados, Tab_Departamentos
WHERE deptoEmp = codigoDepto AND
      salarioEmp < 1000 AND
      nomeDepto = ‘Tecnologia da Informacao’
```

- 3) Obter uma relação contendo os números dos projetos em que o empregado “João da Silva Vieira” trabalha mais de 30 horas.

```
SELECT projeto
FROM Tab_Empregados, Tab_HorasAlocadas
WHERE nomeEmp = ‘João da Silva Vieira’ AND
      matriculaEMP = empregado AND
      horas > 30;
```

- 4) Obter uma relação contendo os nomes dos empregados que não têm dependentes.

```
SELECT nomeEmp
FROM Tab_Empregados
WHERE matriculaEmp NOT IN (SELECT DISTINCT matriculaEmp
                           FROM Tab_Dependentes);
```

```
SELECT nomeEmp
FROM Tab_Empregados
WHERE matriculaEmp IN ( SELECT matriculaEmp
                        EXCEPT
                        SELECT DISTINCT matriculaEmp
                        FROM Tab_Empregados e, Tab_Dependentes d
                        WHERE e.matriculaEmp = d.matriculaEmp);
```

- 5) Obter uma relação contendo os nomes dos empregados que são gerentes e que têm pelo menos um dependente.

```
SELECT nomeEmp
FROM Tab_Empregados e JOIN Tab_Departamentos d
                        ON e.matriculaEmp = d.gerenteDepto
JOIN Tab_Dependentes ON e.matriculaEmp = d.matriculaEmp
```

```
SELECT nomeEmp
FROM Tab_Empregados e, Tab_Departamentos, Tab_Dependentes d
WHERE e.matriculaEmp = d.gerenteDepto AND
      e.matriculaEmp = d.matriculaEmp
```

```
SELECT nomeEmp
FROM Tab_Empregados
WHERE matriculaEmp in (SELECT gerenteDepto FROM Tab_Departamentos) AND
      matriculaEmp in (SELECT matriculaEmp FROM Tab_Dependentes)
```

- 6) Obter uma relação contendo as matrículas e nomes dos empregados que trabalham em projetos da empresa e também dos empregados que gerenciam os departamentos responsáveis pelos projetos.

```
SELECT matriculaEmp, nomeEmp
FROM Tab_Empregados
WHERE matriculaEmp IN (SELECT empregado FROM Tab_HorasAlocadas)
UNION
SELECT matriculaEmp, nomeEmp
FROM Tab_Empregados
WHERE matriculaEmp IN
      (SELECT gerenteDepto
       FROM Tab_Departamentos, Tab_Projetos
       WHERE codigoDepto = deptoProj);
```

```

SELECT matriculaEmp, nomeEmp
FROM Tab_Empregados JOIN Tab_HorasAlocadas on matriculaEmp = empregado
UNION
SELECT matriculaEmp, nomeEmp
FROM Tab_Empregados JOIN Tab_Departamentos on matriculaEmp = gerenteDepto
JOIN Tab_Projetos on codigoDepto = deptoProj

```

```

SELECT matriculaEmp, nomeEmp
FROM Tab_Empregados
WHERE matriculaEmp IN (SELECT empregado FROM Tab_HorasAlocadas)
OR
matriculaEmp IN
(SELECT gerenteDepto
FROM Tab_Departamentos, Tab_Projetos
WHERE codigoDepto = deptoProj);

```

- 7) Obter uma relação contendo os departamentos com salário médio dos empregados maior do que 1000,00. A relação deve conter o código do departamento e o salário médio.

```

SELECT deptoEmp, avg(salarioEmp) "Salario Medio"
FROM Tab_Empregados
GROUP BY deptoEmp
HAVING avg(salarioEmp) > 1000

```

- 8) Obter uma relação contendo os empregados do departamento "Tecnologia da Informacao" que trabalham em mais de dois projetos. A relação deve conter a matrícula do empregado e a quantidade de projetos em que ele trabalha.

```

SELECT empregado, count(projeto) "Qtd projetos em que trabalha"
FROM Tab_Empregados, Tab_Departamentos, Tab_HorasAlocadas
WHERE nomeDepto = 'Tecnologia da Informacao' AND
codigoDepto = deptoEmp AND
matriculaEmp = empregado
GROUP BY empregado
HAVING count(projeto) > 2

```