ORDER BY, GROUP BY, UNION, INTERSECT

SQL

TABELAS

```
CREATE TABLE Departamento (
    id INTEGER AUTO_INCREMENT,
    nome VARCHAR(100) NOT NULL,
    data_inicio DATE,
    PRIMARY KEY(id));

CREATE TABLE Empregado (
    matricula INTEGER,
    nome VARCHAR(100) NOT NULL,
    endereco VARCHAR(100),
    salario DECIMAL(10,2),
    supervisor INTEGER,
    id_depto INTEGER,
    PRIMARY KEY (matricula),
    FOREIGN KEY (supervisor) REFERENCES Empregado(matricula),
    FOREIGN KEY (id_depto) REFERENCES Departamento(id));
```

TABELAS

```
CREATE TABLE Gerencia (
    id_depto INTEGER,
    matricula_emp INTEGER,
    PRIMARY KEY(id_depto),
    FOREIGN KEY (matricula_emp) REFERENCES Empregado(matricula),
    FOREIGN KEY (id_depto) REFERENCES Departamento(id));

CREATE TABLE Projeto (
    id INTEGER AUTO_INCREMENT,
    nome VARCHAR(100) NOT NULL,
    local VARCHAR(100) NOT NULL,
    id_depto INTEGER,
    PRIMARY KEY(id),
    FOREIGN KEY (id_depto) REFERENCES Departamento(id));
```

TABELAS

```
CREATE TABLE Alocacao (
    id_projeto INTEGER,
    matricula_emp INTEGER,
    horas INTEGER,
    PRIMARY KEY(id_projeto, matricula_emp),
    FOREIGN KEY (matricula_emp) REFERENCES Empregado(matricula),
    FOREIGN KEY (id_projeto) REFERENCES Projeto(id));

CREATE TABLE Dependente (
    id INTEGER AUTO_INCREMENT,
    nome VARCHAR(100) NOT NULL,
    sexo CHAR(1) NOT NULL,
    matricula_responsavel INTEGER,
    PRIMARY KEY(id),
    FOREIGN KEY (matricula_responsavel) REFERENCES Empregado(matricula));
```

DADOS

```
INSERT INTO Departamento (nome, data_inicio) VALUES ("Pesquisa", "2019-11-09");
INSERT INTO Departamento (nome, data_inicio) VALUES ("Cozinha", "2009-08-05");
INSERT INTO Departamento (nome, data_inicio) VALUES ("Estudo", "2020-03-17");
INSERT INTO Departamento (nome, data_inicio) VALUES ("Ciencia", "2010-03-09");
INSERT INTO Departamento (nome, data_inicio) VALUES ("Compras", "2012-09-09");
```

```
INSERT INTO Empregado (matricula, nome, endereco, salario, id_depto) VALUES
(273729, "Carla", "Rua A", 1000.20, 3);
INSERT INTO Empregado VALUES (777746, "Danilo", "Rua B", 2000.20, 273729, 3);
INSERT INTO Empregado VALUES (988716, "Daniel", "Rua C", 100.20, 273729, 2);
INSERT INTO Empregado VALUES (483292, "Alice", "Rua D", 2000.20, 777746, 2);
INSERT INTO Empregado VALUES (553211, "Picachu", "Rua E", 20.20, 777746, 2);
```

```
INSERT INTO Gerencia VALUES (1, 273729);
INSERT INTO Gerencia VALUES (2, 777746);
INSERT INTO Gerencia VALUES (3, 273729);
```

DADOS

```
INSERT INTO Projeto (nome, local, id_depto) VALUES ("Projeto Alpha", "Natal", 1);
INSERT INTO Projeto (nome, local, id_depto) VALUES ("Projeto Beta", "Recife", 1);
INSERT INTO Projeto (nome, local, id_depto) VALUES ("Projeto Delta", "Natal", 1);
INSERT INTO Projeto (nome, local, id_depto) VALUES ("Projeto Gama", "Natal", 2);
```

```
INSERT INTO Alocacao VALUES (1, 273729, 40);
INSERT INTO Alocacao VALUES (2, 273729, 20);
INSERT INTO Alocacao VALUES (3, 777746, 60);
INSERT INTO Alocacao VALUES (3, 988716, 40);
INSERT INTO Alocacao VALUES (3, 483292, 40);
```

```
INSERT INTO Dependente (nome, sexo, matricula_responsavel) VALUES ("Filho 1", 'M', 273729);
INSERT INTO Dependente (nome, sexo, matricula_responsavel) VALUES ("Filha 2", 'F', 273729);
```

01

ORDER BY

Ordenação do resultado de uma busca

ORDER BY

```
SELECT lista_atributos
FROM tabelas
[ORDER BY lista_atributos [ASC, DESC]]
```

Liste os empregados, seus departamentos e seus projetos, ordenando por nome do departamento.

```
mysql> SELECT d.nome, e.nome, p.nome
    -> FROM Departamento d, Empregado e, Projeto p, Alocacao a
    -> WHERE d.id = e.id_depto AND e.matricula = a.matricula_emp
    -> AND a.id_projeto = p.id
    -> ORDER BY d.nome, e.nome;
+---------+--------+---------------+
| nome    | nome   | nome          |
+---------+--------+---------------+
| Cozinha | Alice  | Projeto Delta |
| Cozinha | Daniel | Projeto Delta |
| Estudo  | Carla  | Projeto Alpha |
| Estudo  | Carla  | Projeto Beta  |
| Estudo  | Danilo | Projeto Delta |
+---------+--------+---------------+
5 rows in set (0.01 sec)
```

```
SELECT d.nome, e.nome, p.nome
FROM Departamento d, Empregado e, Projeto p, Alocacao a
WHERE d.id = e.id_depto AND e.matricula = a.matricula_emp
        AND a.id_projeto = p.id
ORDER BY d.nome, e.nome;
```

ORDER BY

```
SELECT lista_atributos
FROM tabelas
[ORDER BY lista_atributos [ASC, DESC]]
```

Liste os empregados, seus departamentos e seus projetos, ordenando por nome do departamento.

```
mysql> SELECT d.nome, e.nome, p.nome
    -> FROM Departamento d, Empregado e, Projeto p, Alocacao a
    -> WHERE d.id = e.id_depto AND e.matricula = a.matricula_emp
    -> AND a.id_projeto = p.id
    -> ORDER BY d.nome DESC, e.nome;
+---------+--------+---------------+
| nome    | nome   | nome          |
+---------+--------+---------------+
| Estudo  | Carla  | Projeto Alpha |
| Estudo  | Carla  | Projeto Beta  |
| Estudo  | Danilo | Projeto Delta |
| Cozinha | Alice  | Projeto Delta |
| Cozinha | Daniel | Projeto Delta |
+---------+--------+---------------+
5 rows in set (0.00 sec)
```

```
SELECT d.nome, e.nome, p.nome
FROM Departamento d, Empregado e, Projeto p, Alocacao a
WHERE d.id = e.id_depto AND e.matricula = a.matricula_emp
        AND a.id_projeto = p.id
ORDER BY d.nome DESC, e.nome;
```

02

GROUP BY

Operações agregadas

SUM, COUNT, MIN, MAX, AVG

GROUP BY

```
SELECT funcao(atributo)
FROM tabelas
[GROUP BY atributo]
[HAVING condicao_funcao]
```

Encontre o valor total pago de salário aos empregados na empresa, o maior e o menor salário, e a média salarial

```
SELECT SUM(salario), MAX(salario), MIN(salario), AVG(salario)
FROM Empregado;
```

```
mysql> SELECT SUM(salario), MAX(salario), MIN(salario), AVG(salario)
    -> FROM Empregado;
+--------------+--------------+--------------+--------------+
| SUM(salario) | MAX(salario) | MIN(salario) | AVG(salario) |
+--------------+--------------+--------------+--------------+
|      5121.00 |      2000.20 |        20.20 |  1024.200000 |
+--------------+--------------+--------------+--------------+
1 row in set (0.01 sec)
```

GROUP BY

```
SELECT funcao(atributo)
FROM tabelas
[GROUP BY atributo]
[HAVING condicao_funcao]
```

Encontre o valor total pago de salário aos empregados na empresa, o maior e o menor salário, e a média salarial

```
SELECT SUM(salario) AS Total, MAX(salario) AS Topo_salarial,
       MIN(salario) AS Piso_salarial, round(AVG(salario),2) AS Media_salarial
FROM Empregado;
```

```
mysql> SELECT SUM(salario) AS Total, MAX(salario) AS Topo_salarial,
    -> MIN(salario) AS Piso_salarial, round(AVG(salario),2) AS Media_salarial
    -> FROM Empregado;
+---------+---------------+---------------+----------------+
| Total   | Topo_salarial | Piso_salarial | Media_salarial |
+---------+---------------+---------------+----------------+
| 5121.00 |       2000.20 |         20.20 |        1024.20 |
+---------+---------------+---------------+----------------+
1 row in set (0.00 sec)
```

GROUP BY

Encontre o número de empregados da empresa

```
SELECT COUNT(*) AS Quantidade_empregados
FROM Empregado;
```

```
mysql> SELECT COUNT(*) AS Quantidade_empregados
    -> FROM Empregado;
+----------------------+
| Quantidade_empregados |
+----------------------+
|                    5 |
+----------------------+
1 row in set (0.01 sec)
```

Encontre o número de empregados do departamento Cozinha

```
SELECT COUNT(*) AS Quantidade_empregados
FROM Empregado e, Departamento d
WHERE e.id_depto = d.id
          AND d.nome = "Cozinha";
```

```
mysql> SELECT COUNT(*) AS Quantidade_empregados
    -> FROM Empregado e, Departamento d
    -> WHERE e.id_depto = d.id
    -> AND d.nome = "Cozinha";
+----------------------+
| Quantidade_empregados |
+----------------------+
|                    3 |
+----------------------+
1 row in set (0.00 sec)
```

GROUP BY

Encontre o maior e o menor salário do departamento de Cozinha

```
SELECT MAX(salario) AS salario_max, MIN(salario) as salario_min
FROM Empregado e, Departamento d
WHERE e.id_depto = d.id AND d.nome = "Cozinha";
```

```
mysql> SELECT MAX(salario) AS salario_max, MIN(salario) as salario_min
    -> FROM Empregado e, Departamento d
    -> WHERE e.id_depto = d.id AND d.nome = "Cozinha";
+-------------+-------------+
| salario_max | salario_min |
+-------------+-------------+
|     2000.20 |       20.20 |
+-------------+-------------+
1 row in set (0.00 sec)
```

Obtenha o número de salários distintos pagos pela empresa

```
SELECT COUNT(DISTINCT salario) AS salarios
FROM Empregado;
```

```
mysql> SELECT COUNT(DISTINCT salario) AS salarios
    -> FROM Empregado;
+----------+
| salarios |
+----------+
|        4 |
+----------+
1 row in set (0.01 sec)
```

E se eu quiser fazer a agregação, mas separado por grupos?

Encontre a média salarial, e o salário máximo, separado por departamentos.

GROUP BY

Para cada departamento, obter o código do departamento, o número de empregados e a média salarial

```
SELECT id_depto, COUNT(*), round(AVG(salario),2)
FROM Empregado
GROUP BY id_depto;
```

```
mysql> SELECT id_depto, COUNT(*), round(AVG(salario),2)
    -> FROM Empregado
    -> GROUP BY id_depto;
+-----------+----------+-----------------------+
| id_depto | COUNT(*) | round(AVG(salario),2) |
+-----------+----------+-----------------------+
|         2 |        3 |                706.87 |
|         3 |        2 |               1500.20 |
+-----------+----------+-----------------------+
2 rows in set (0.00 sec)
```

```
SELECT d.nome, COUNT(*) AS No_Empregados, round(AVG(salario),2) AS Media_salarial
FROM Empregado e, Departamento d
WHERE e.id_depto = d.id
GROUP BY id_depto
ORDER BY d.nome;
```

```
mysql> SELECT d.nome, COUNT(*) AS No_Empregados, round(AVG(salario),2) AS Media_salarial
    -> FROM Empregado e, Departamento d
    -> WHERE e.id_depto = d.id
    -> GROUP BY id_depto
    -> ORDER BY d.nome;
+---------+---------------+----------------+
| nome    | No_Empregados | Media_salarial |
+---------+---------------+----------------+
| Cozinha |             3 |         706.87 |
| Estudo  |             2 |        1500.20 |
+---------+---------------+----------------+
2 rows in set (0.00 sec)
```

Para cada projeto, obter o nome do projeto, seu identificador e o número de empregados que trabalham nele

GROUP BY

```
SELECT p.nome, p.id, COUNT(*) AS Total_participantes
FROM Empregado e, Projeto p, Alocacao a
WHERE a.id_projeto = p.id AND a.matricula_emp = e.matricula
GROUP BY p.id
ORDER BY p.nome;
```

```
mysql> SELECT p.nome, p.id, COUNT(*) AS Total_participantes
    -> FROM Empregado e, Projeto p, Alocacao a
    -> WHERE a.id_projeto = p.id AND a.matricula_emp = e.matricula
    -> GROUP BY p.id
    -> ORDER BY p.nome;
+---------------+----+---------------------+
| nome          | id | Total_participantes |
+---------------+----+---------------------+
| Projeto Alpha |  1 |                   1 |
| Projeto Beta  |  2 |                   1 |
| Projeto Delta |  3 |                   3 |
+---------------+----+---------------------+
3 rows in set (0.01 sec)
```

GROUP BY

Para cada projeto que possui mais do que 2 empregados alocados, obter o nome do projeto, seu identificador e o número de empregados que trabalham nele

```
SELECT p.nome, p.id, COUNT(*) AS Total_participantes
FROM Empregado e, Projeto p, Alocacao a
WHERE a.id_projeto = p.id AND a.matricula_emp = e.matricula
GROUP BY p.id
HAVING COUNT(*) > 2
ORDER BY p.nome;
```

```
mysql> SELECT p.nome, p.id, COUNT(*) AS Total_participantes
    -> FROM Empregado e, Projeto p, Alocacao a
    -> WHERE a.id_projeto = p.id AND a.matricula_emp = e.matricula
    -> GROUP BY p.id
    -> HAVING COUNT(*) > 2
    -> ORDER BY p.nome;
+---------------+----+---------------------+
| nome          | id | Total_participantes |
+---------------+----+---------------------+
| Projeto Delta |  3 |                   3 |
+---------------+----+---------------------+
1 row in set (0.00 sec)
```

GROUP BY

Identifique a quantidade de níveis salariais da empresa para os empregados de carteira assinada (empregados com salário maior ou igual a um salário mínimo R$1000)

```
SELECT COUNT(DISTINCT salario) AS Niveis_salariais
FROM Empregado
WHERE salario >= 1000;
```

```
mysql> SELECT COUNT(DISTINCT salario) AS Niveis_salariais
    -> FROM Empregado
    -> WHERE salario >= 1000;
+------------------+
| Niveis_salariais |
+------------------+
|                2 |
+------------------+
1 row in set (0.01 sec)
```

Identifique a média salarial de cada departamento que tem pelo menos 3 empregados.

```
SELECT id_depto, ROUND(AVG(salario),2) AS media
FROM Empregado
GROUP BY id_depto;
```

```
mysql> SELECT id_depto, ROUND(AVG(salario),2) AS media
    -> FROM Empregado
    -> GROUP BY id_depto;
+----------+---------+
| id_depto | media   |
+----------+---------+
|        2 |  706.87 |
|        3 | 1500.20 |
+----------+---------+
2 rows in set (0.00 sec)
```

GROUP BY

Identifique a quantidade de níveis salariais da empresa para os empregados de carteira assinada (empregados com salário maior ou igual a um salário mínimo R$1000)

```
SELECT COUNT(DISTINCT salario) AS Niveis_salariais
FROM Empregado
WHERE salario >= 1000;
```

```
mysql> SELECT COUNT(DISTINCT salario) AS Niveis_salariais
    -> FROM Empregado
    -> WHERE salario >= 1000;
+------------------+
| Niveis_salariais |
+------------------+
|                2 |
+------------------+
1 row in set (0.01 sec)
```

Identifique a média salarial de cada departamento que tem pelo menos 3 empregados.

```
SELECT id_depto, ROUND(AVG(salario),2) AS media
FROM Empregado
GROUP BY id_depto
HAVING COUNT(*) >= 3;
```

```
mysql> SELECT id_depto, ROUND(AVG(salario),2) AS media
    -> FROM Empregado
    -> GROUP BY id_depto
    -> HAVING COUNT(*) >= 3;
+----------+--------+
| id_depto | media  |
+----------+--------+
|        2 | 706.87 |
+----------+--------+
1 row in set (0.00 sec)
```

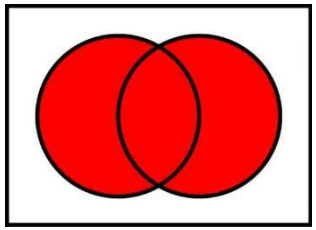
03

UNION

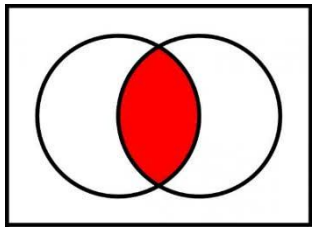
O que acontece quando precisamos de dados de duas buscas diferentes?

UNION

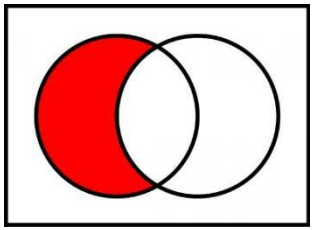
```
(SELECT …
FROM …
WHERE …)
UNION
(SELECT …
FROM …
WHERE …)
```



```
(SELECT …
FROM …
WHERE …)
INTERSECT
(SELECT …
FROM …
WHERE …)
```



```
(SELECT …
FROM …
WHERE …)
EXCEPT
(SELECT …
FROM …
WHERE …)
```



UNIÃO ENTRE QUERYS

Identifique os departamentos responsáveis por cada projeto em que Carla trabalha

```
SELECT d.nome
FROM Departamento d, Empregado e,
     Projeto p, Alocacao a
WHERE a.id_projeto = p.id AND
     a.matricula_emp = e.matricula AND
     p.id_depto = d.id AND
     e.nome = "Carla";
```

```
mysql> SELECT d.nome
    -> FROM Departamento d, Empregado e,
    -> Projeto p, Alocacao a
    -> WHERE a.id_projeto = p.id AND
    -> a.matricula_emp = e.matricula AND
    -> p.id_depto = d.id AND
    -> e.nome = "Carla";
+----------+
| nome     |
+----------+
| Pesquisa |
| Pesquisa |
+----------+
2 rows in set (0.00 sec)
```

Identifique os departamentos que Carla gerencia

```
SELECT d.nome
FROM Departamento d, Empregado e,
     Gerencia g
WHERE g.id_depto = d.id AND
     g.matricula_emp = e.matricula AND
     e.nome = "Carla";
```

```
mysql> SELECT d.nome
    -> FROM Departamento d, Empregado e,
    -> Gerencia g
    -> WHERE g.id_depto = d.id AND
    -> g.matricula_emp = e.matricula AND
    -> e.nome = "Carla";
+----------+
| nome     |
+----------+
| Pesquisa |
| Estudo   |
+----------+
2 rows in set (0.00 sec)
```

UNION

Identifique os departamentos associados a Carla, seja responsáveis pelos projetos em que ela trabalha, ou os departamentos que ela gerencia

```
(SELECT d.nome
FROM Departamento d, Empregado e,
     Projeto p, Alocacao a
WHERE a.id_projeto = p.id AND
     a.matricula_emp = e.matricula AND
     p.id_depto = d.id AND
     e.nome = "Carla")
UNION
(SELECT d.nome
FROM Departamento d, Empregado e,
     Gerencia g
WHERE g.id_depto = d.id AND
     g.matricula_emp = e.matricula AND
     e.nome = "Carla");
```

```
mysql> (SELECT d.nome
    -> FROM Departamento d, Empregado e,
    -> Projeto p, Alocacao a
    -> WHERE a.id_projeto = p.id AND
    -> a.matricula_emp = e.matricula AND
    -> p.id_depto = d.id AND
    -> e.nome = "Carla")
    -> UNION
    -> (SELECT d.nome
    -> FROM Departamento d, Empregado e,
    -> Gerencia g
    -> WHERE g.id_depto = d.id AND
    -> g.matricula_emp = e.matricula AND
    -> e.nome = "Carla");
+----------+
| nome     |
+----------+
| Pesquisa |
| Estudo   |
+----------+
2 rows in set (0.01 sec)
```

UNION

Identifique os departamentos associados a Carla, seja responsáveis pelos projetos em que ela trabalha, ou os departamentos que ela gerencia

```
(SELECT d.nome
FROM Departamento d, Empregado e,
     Projeto p, Alocacao a
WHERE a.id_projeto = p.id AND
     a.matricula_emp = e.matricula AND
     p.id_depto = d.id AND
     e.nome = "Carla")
UNION ALL
(SELECT d.nome
FROM Departamento d, Empregado e,
     Gerencia g
WHERE g.id_depto = d.id AND
     g.matricula_emp = e.matricula AND
     e.nome = "Carla");
```

```
mysql> (SELECT d.nome
    -> FROM Departamento d, Empregado e,
    -> Projeto p, Alocacao a
    -> WHERE a.id_projeto = p.id AND
    -> a.matricula_emp = e.matricula AND
    -> p.id_depto = d.id AND
    -> e.nome = "Carla")
    -> UNION ALL
    -> (SELECT d.nome
    -> FROM Departamento d, Empregado e,
    -> Gerencia g
    -> WHERE g.id_depto = d.id AND
    -> g.matricula_emp = e.matricula AND
    -> e.nome = "Carla");
+----------+
| nome     |
+----------+
| Pesquisa |
| Pesquisa |
| Pesquisa |
| Estudo   |
+----------+
4 rows in set (0.00 sec)
```

INTERSECT

Identifique os departamentos associados a Carla, responsáveis pelos projetos em que ela trabalha, e que ela gerencia

```
SELECT DISTINCT d.nome
FROM Departamento d, Empregado e,
     Projeto p, Alocacao a
WHERE a.id_projeto = p.id AND
     a.matricula_emp = e.matricula AND
     p.id_depto = d.id AND
     e.nome = "Carla" AND
     d.id IN
          (SELECT d.id
          FROM Departamento d, Empregado e,
               Gerencia g
          WHERE g.id_depto = d.id AND
               g.matricula_emp = e.matricula AND
               e.nome = "Carla");
```

```
mysql> SELECT DISTINCT d.nome
    -> FROM Departamento d, Empregado e,
    -> Projeto p, Alocacao a
    -> WHERE a.id_projeto = p.id AND
    -> a.matricula_emp = e.matricula AND
    -> p.id_depto = d.id AND
    -> e.nome = "Carla" AND
    -> d.id IN
    -> (SELECT d.id
    -> FROM Departamento d, Empregado e,
    -> Gerencia g
    -> WHERE g.id_depto = d.id AND
    -> g.matricula_emp = e.matricula AND
    -> e.nome = "Carla");
+----------+
| nome     |
+----------+
| Pesquisa |
+----------+
1 row in set (0.01 sec)
```

EXCEPT

Identifique os departamentos associados a Carla, responsáveis pelos projetos em que ela trabalha, mas que ela não gerencia

```
SELECT DISTINCT d.nome
FROM Departamento d, Empregado e,
     Projeto p, Alocacao a
WHERE a.id_projeto = p.id AND
     a.matricula_emp = e.matricula AND
     p.id_depto = d.id AND
     e.nome = "Carla" AND
     d.id NOT IN
          (SELECT d.id
          FROM Departamento d, Empregado e,
               Gerencia g
          WHERE g.id_depto = d.id AND
               g.matricula_emp = e.matricula AND
               e.nome = "Carla");
```

```
mysql> SELECT DISTINCT d.nome
    -> FROM Departamento d, Empregado e,
    -> Projeto p, Alocacao a
    -> WHERE a.id_projeto = p.id AND
    -> a.matricula_emp = e.matricula AND
    -> p.id_depto = d.id AND
    -> e.nome = "Carla" AND
    -> d.id NOT IN
    -> (SELECT d.id
    -> FROM Departamento d, Empregado e,
    -> Gerencia g
    -> WHERE g.id_depto = d.id AND
    -> g.matricula_emp = e.matricula AND
    -> e.nome = "Carla");
Empty set (0.00 sec)
```

EXCEPT

Identifique os departamentos que Carla gerencia, mas que ela não tem nenhum projeto nesse departamento

```
SELECT d.nome
FROM Departamento d, Empregado e,
     Gerencia g
WHERE g.id_depto = d.id AND
     g.matricula_emp = e.matricula AND
     e.nome = "Carla" AND
     d.id NOT IN
          (SELECT d.id
          FROM Departamento d, Empregado e,
               Projeto p, Alocacao a
          WHERE a.id_projeto = p.id AND
               a.matricula_emp = e.matricula AND
               p.id_depto = d.id AND
               e.nome = "Carla");
```

```
mysql> SELECT d.nome
    -> FROM Departamento d, Empregado e,
    -> Gerencia g
    -> WHERE g.id_depto = d.id AND
    -> g.matricula_emp = e.matricula AND
    -> e.nome = "Carla" AND
    -> d.id NOT IN
    -> (SELECT d.id
    -> FROM Departamento d, Empregado e,
    -> Projeto p, Alocacao a
    -> WHERE a.id_projeto = p.id AND
    -> a.matricula_emp = e.matricula AND
    -> p.id_depto = d.id AND
    -> e.nome = "Carla");
+--------+
| nome   |
+--------+
| Estudo |
+--------+
1 row in set (0.01 sec)
```

04

FILTROS

Como fazemos filtros em um sistema?

RIACHUELO

Bem-vindo à Riachuelo

patrulha canina

NOVIDADES | FEMININO | MASCULINO | INFANTIL | MODA ÍNTIMA | CALÇADOS | ACESSÓRIOS E RELÓGIOS | BELEZA E PERFUME | CASA RIACHUELO | ELETRÔNICOS | PERSONAGENS | SALDOS

Tamanhos | Marcas | Categorias (3) | Preço | Ordenar

INFANTIL E TEEN | DISNEY | CAMISA E POLO | LIMPAR TUDO

Feedback

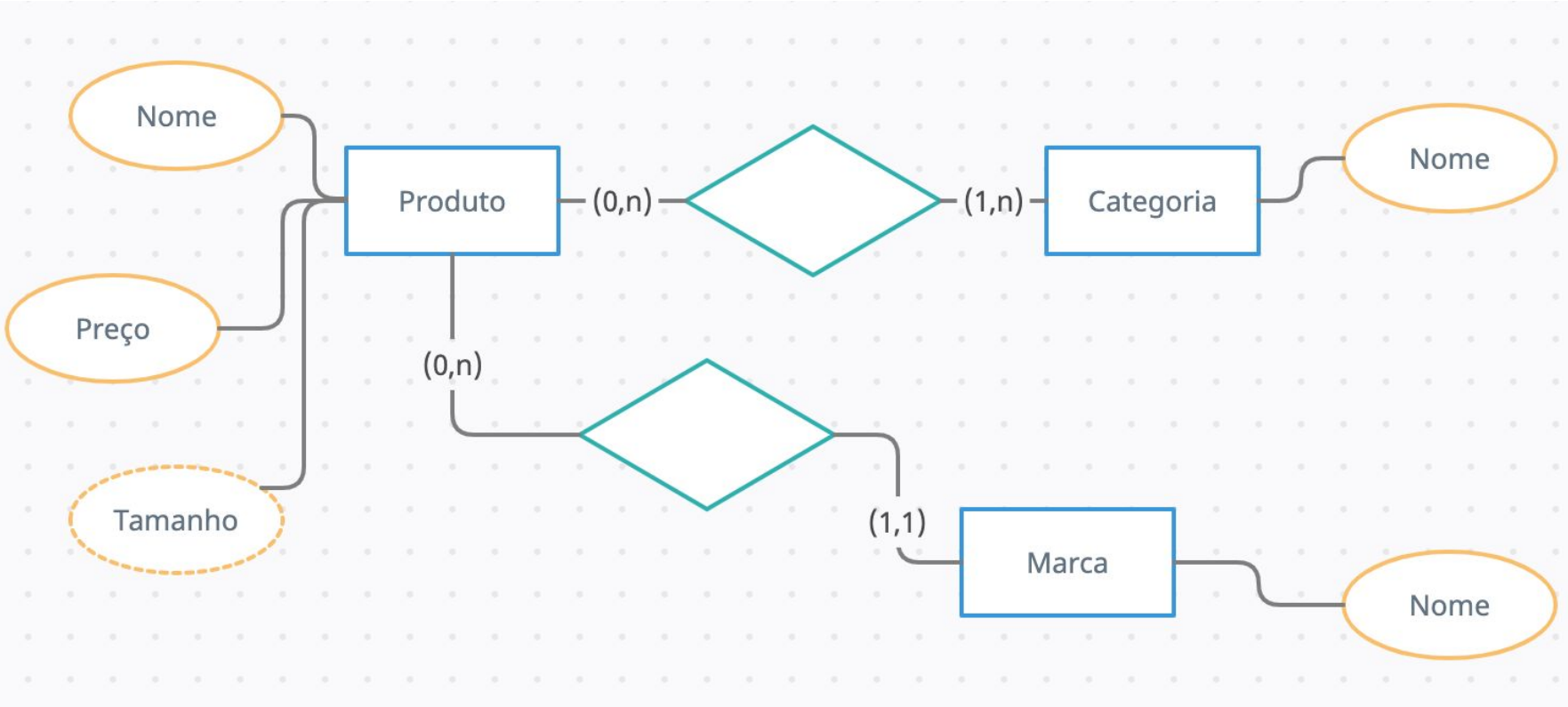
KIT 3 CUECAS PATRULHA CANINA

KIT CALCINHA 3 PEÇAS PATRULHA

CAMIS ATRULHA

CAMISETA MALHA ALGODÃO PATRULHA

Dúvidas? Clique aqui


Nome
Produto
(0,n)
(1,n)
Categoria
Nome
Preço
(0,n)
Tamanho
(1,1)
Marca
Nome

TABELAS

```
CREATE TABLE Marca (
     id INTEGER AUTO_INCREMENT,
     nome VARCHAR(100) NOT NULL,
     PRIMARY KEY(id));

CREATE TABLE Produto (
     id INTEGER AUTO_INCREMENT,
     nome VARCHAR(100) NOT NULL,
     preco DECIMAL(10,2),
     id_marca INTEGER,
     PRIMARY KEY (id),
     FOREIGN KEY (id_marca) REFERENCES Marca(id));

CREATE TABLE Tamanho (
     id_produto INTEGER,
     tamanho VARCHAR(100) NOT NULL,
     PRIMARY KEY (id_produto, tamanho),
     FOREIGN KEY (id_produto) REFERENCES Produto(id));
```

TABELAS

```
CREATE TABLE Categoria (
    id INTEGER AUTO_INCREMENT,
    nome VARCHAR(100) NOT NULL,
    PRIMARY KEY(id));
```

```
CREATE TABLE Categoria_Produto (
    id_produto INTEGER,
    id_categoria INTEGER,
    PRIMARY KEY (id_produto, id_categoria),
    FOREIGN KEY (id_produto) REFERENCES Produto(id),
    FOREIGN KEY (id_categoria) REFERENCES Categoria(id));
```

DADOS

```
INSERT INTO Marca (nome) VALUES ("Anne Frank");
INSERT INTO Marca (nome) VALUES ("Camisas legais");
INSERT INTO Marca (nome) VALUES ("Marca X");
INSERT INTO Marca (nome) VALUES ("Marca Y");
```

```
INSERT INTO Produto (nome, preco, id_marca) VALUES ("Blusa Skye", 49.90, 3);
INSERT INTO Produto (nome, preco, id_marca) VALUES ("Blusa Chase", 49.90, 3);
INSERT INTO Produto (nome, preco, id_marca) VALUES ("Blusa Rocky", 49.90, 3);
INSERT INTO Produto (nome, preco, id_marca) VALUES ("Bolsa tiracolo", 99.90, 1);
INSERT INTO Produto (nome, preco, id_marca) VALUES ("Bolsa barata", 49.90, 1);
INSERT INTO Produto (nome, preco, id_marca) VALUES ("Outra bolsa", 190.00, 1);
INSERT INTO Produto (nome, preco, id_marca) VALUES ("Blusa tiedye", 50.90, 2);
INSERT INTO Produto (nome, preco, id_marca) VALUES ("Blusa chique", 120.90, 2);
INSERT INTO Produto (nome, preco, id_marca) VALUES ("Blusa simples", 50.90, 2);
```

DADOS

```
INSERT INTO Tamanho VALUES (1, "P");
INSERT INTO Tamanho VALUES (1, "M");
INSERT INTO Tamanho VALUES (1, "G");
INSERT INTO Tamanho VALUES (2, "P");
INSERT INTO Tamanho VALUES (2, "M");
INSERT INTO Tamanho VALUES (3, "G");
INSERT INTO Tamanho VALUES (4, "U");
INSERT INTO Tamanho VALUES (5, "U");
INSERT INTO Tamanho VALUES (6, "U");
INSERT INTO Tamanho VALUES (7, "P");
INSERT INTO Tamanho VALUES (7, "M");
INSERT INTO Tamanho VALUES (7, "G");
INSERT INTO Tamanho VALUES (8, "P");
INSERT INTO Tamanho VALUES (9, "M");
INSERT INTO Tamanho VALUES (9, "G");
```

DADOS

```
INSERT INTO Categoria (nome) VALUES ("Blusa");
INSERT INTO Categoria (nome) VALUES ("Roupa");
INSERT INTO Categoria (nome) VALUES ("Bolsa");
INSERT INTO Categoria (nome) VALUES ("Feminino");
INSERT INTO Categoria (nome) VALUES ("Infantil");

INSERT INTO Categoria_Produto (id_produto, id_categoria) VALUES (1, 1);
INSERT INTO Categoria_Produto (id_produto, id_categoria) VALUES (1, 2);
INSERT INTO Categoria_Produto (id_produto, id_categoria) VALUES (1, 5);
INSERT INTO Categoria_Produto (id_produto, id_categoria) VALUES (2, 1);
INSERT INTO Categoria_Produto (id_produto, id_categoria) VALUES (2, 2);
INSERT INTO Categoria_Produto (id_produto, id_categoria) VALUES (2, 5);
INSERT INTO Categoria_Produto (id_produto, id_categoria) VALUES (3, 1);
INSERT INTO Categoria_Produto (id_produto, id_categoria) VALUES (3, 2);
INSERT INTO Categoria_Produto (id_produto, id_categoria) VALUES (3, 5);
INSERT INTO Categoria_Produto (id_produto, id_categoria) VALUES (4, 3);
INSERT INTO Categoria_Produto (id_produto, id_categoria) VALUES (4, 4);
INSERT INTO Categoria_Produto (id_produto, id_categoria) VALUES (5, 3);
INSERT INTO Categoria_Produto (id_produto, id_categoria) VALUES (5, 4);
```

DADOS

```
INSERT INTO Categoria_Produto (id_produto, id_categoria) VALUES (6, 3);
INSERT INTO Categoria_Produto (id_produto, id_categoria) VALUES (6, 4);
INSERT INTO Categoria_Produto (id_produto, id_categoria) VALUES (7, 1);
INSERT INTO Categoria_Produto (id_produto, id_categoria) VALUES (7, 2);
INSERT INTO Categoria_Produto (id_produto, id_categoria) VALUES (7, 4);
INSERT INTO Categoria_Produto (id_produto, id_categoria) VALUES (8, 1);
INSERT INTO Categoria_Produto (id_produto, id_categoria) VALUES (8, 2);
INSERT INTO Categoria_Produto (id_produto, id_categoria) VALUES (8, 4);
INSERT INTO Categoria_Produto (id_produto, id_categoria) VALUES (9, 1);
INSERT INTO Categoria_Produto (id_produto, id_categoria) VALUES (9, 2);
INSERT INTO Categoria_Produto (id_produto, id_categoria) VALUES (9, 4);
```

FILTRO

CATEGORIA Feminino e TAMANHO P

```
SELECT p.nome, p.preco
FROM Produto p, Tamanho t, Categoria c, Categoria_Produto cp
WHERE p.id = t.id_produto AND
      cp.id_produto = p.id AND cp.id_categoria = c.id AND
      c.nome = "Feminino" AND t.tamanho = "P";
```

```
mysql> SELECT p.nome, p.preco
    -> FROM Produto p, Tamanho t, Categoria c, Categoria_Produto cp
    -> WHERE p.id = t.id_produto AND
    -> cp.id_produto = p.id AND cp.id_categoria = c.id AND
    -> c.nome = "Feminino" AND t.tamanho = "P";
+--------------+--------+
| nome         | preco  |
+--------------+--------+
| Blusa tiedye |  50.90 |
| Blusa chique | 120.90 |
+--------------+--------+
2 rows in set (0.00 sec)
```

FILTRO

CATEGORIA Feminino e TAMANHO P

```
mysql> SELECT p.nome, p.preco
    -> FROM Produto p
    -> WHERE p.id IN (SELECT p.id
    ->                FROM Produto p, Tamanho t
    ->                WHERE t.id_produto = p.id AND t.tamanho = "P")
    ->           AND
    ->       p.id IN (SELECT p.id
    ->                FROM Produto p, Categoria c, Categoria_Produto cp
    ->                WHERE cp.id_produto = p.id AND cp.id_categoria = c.id AND
    ->                      c.nome = "Feminino");
+--------------+--------+
| nome         | preco  |
+--------------+--------+
| Blusa tiedye |  50.90 |
| Blusa chique | 120.90 |
+--------------+--------+
2 rows in set (0.00 sec)
```

```
SELECT p.nome, p.preco
FROM Produto p
WHERE p.id IN (SELECT p.id
               FROM Produto p, Tamanho t
               WHERE t.id_produto = p.id AND t.tamanho = "P")
          AND
      p.id IN (SELECT p.id
               FROM Produto p, Categoria c, Categoria_Produto cp
               WHERE cp.id_produto = p.id AND cp.id_categoria = c.id AND
                     c.nome = "Feminino");
```

CATEGORIA Bolsa, Blusa, Tamanho P, Preco > 100

FILTRO

```
mysql> SELECT p.nome, p.preco
    -> FROM Produto p
    -> WHERE p.preco > 100 AND
    ->       p.id IN (SELECT p.id
    ->                FROM Produto p, Tamanho t
    ->                WHERE t.id_produto = p.id AND t.tamanho = "P")
    ->           AND
    ->       p.id IN (SELECT p.id
    ->                FROM Produto p, Categoria c, Categoria_Produto cp
    ->                WHERE cp.id_produto = p.id AND cp.id_categoria = c.id AND
    ->                      (c.nome = "Bolsa" OR c.nome = "Blusa"));
+--------------+--------+
| nome         | preco  |
+--------------+--------+
| Blusa chique | 120.90 |
+--------------+--------+
1 row in set (0.00 sec)
```

```
SELECT p.nome, p.preco
FROM Produto p
WHERE p.preco > 100 AND
      p.id IN (SELECT p.id
               FROM Produto p, Tamanho t
               WHERE t.id_produto = p.id AND t.tamanho = "P")
          AND
      p.id IN (SELECT p.id
               FROM Produto p, Categoria c, Categoria_Produto cp
               WHERE cp.id_produto = p.id AND cp.id_categoria = c.id AND
                     (c.nome = "Bolsa" OR c.nome = "Blusa"));
```

DÚVIDAS?

Qualquer dúvida entrar em contato

Profª. Carla Fernandes Curvelo

carlafcf@gmail.com