

Aula 10

Trabalho para aprender o agrupamento de tabelas em MySQL

Tabela utilizada:

```
create table medicos(  
  idmed          int          not null,  
  medico         varchar(40) not null,  
  especialidade  varchar(20) not null,  
  salario        decimal(8,2) not null,  
  cidade         varchar(20) not null,  
  idade          int          null,  
  dtadm          datetime    not null,  
  primary key(idmed)  
);
```

Exemplo de valores inseridos/validos:

```
INSERT INTO medicos (idmed, medico, especialidade, salario, cidade, idade, dtadm) VALUES  
(1, 'Ana Beatriz Lima', 'Cardiologia', 15800.00, 'Bauru', 45, '2015-03-12'),  
(2, 'Carlos Eduardo Santos', 'Pediatría', 14250.50, 'Agudos', 38, '2018-05-09'),  
(3, 'Fernanda Souza', 'Dermatologia', 13300.75, 'Piratinga', 40, '2016-01-15'),  
(4, 'João Pedro Mendes', 'Neurologia', 18000.00, 'Pederneiras', 50, '2012-07-30'),  
(5, 'Juliana Rocha', 'Ortopedia', 12000.00, 'Lençóis Paulista', 36, '2019-04-20'),  
(6, 'Marcos Vinícius Alves', 'Cardiologia', 15800.00, 'Arealva', 41, '2017-11-10'),  
(7, 'Paula Regina Silva', 'Ginecologia', 14000.00, 'Avaí', 37, '2020-06-01'),  
(8, 'Ricardo Nunes', 'Psiquiatria', 16500.00, 'Duartina', 48, '2013-02-14'),  
(9, 'Sabrina Freitas', 'Oftalmologia', 15000.00, 'Macatuba', 39, '2018-08-22'),  
(10, 'Thiago Martins', 'Clínica Geral', 12500.00, 'Lucianópolis', 34, '2021-09-18');
```

1- Mostrar a média salarial dos médicos agrupados por cidade.

Aula 10

```
SELECT cidade, avg(salario) as 'Média salarial por cidade'
FROM `medicos`
group by cidade;
```

cidade	Media salarial por cidade
Bauru	15800.000000
Agudos	14250.500000
Piratininga	13300.750000
Pederneiras	18000.000000
Lençóis Paulista	12000.000000
Arealva	15800.000000
Avai	14000.000000
Duartina	16500.000000
Macatuba	15000.000000
Lucianópolis	12500.000000

2 - Mostrar a média salarial dos médicos agrupados por especialidade

```
SELECT especialidade, avg(salario) as 'Média salarial por
especialidade'
FROM `medicos`
group by especialidade;
```

especialidade	Media salarial por especialidade
Cardiologia	15800.000000
Pediatria	14250.500000
Dermatologia	13300.750000
Neurologia	18000.000000
Ortopedia	12000.000000
Ginecologia	14000.000000
Psiquiatria	16500.000000
Oftalmologia	15000.000000
Clínica Geral	12500.000000

3 - Mostrar o maior salário dos médicos agrupados por cidade e especialidade.

Aula 10

SELECT cidade, especialidade, max(salario) as
'maior salário'

FROM `medicos`

group by cidade, especialidade;

Agudos	Ginecologia	14000.00
Agudos	Pediatria	14250.50
Arealva	Cardiologia	15900.00
Arealva	Clínica Geral	12700.00
Arealva	Dermatologia	13500.00
Arealva	Ginecologia	14100.00
Arealva	Neurologia	16900.00
Arealva	Pediatria	15000.00
Avaí	Cardiologia	15800.00
Avaí	Clínica Geral	12800.00
Avaí	Endocrinologia	15500.00
Avaí	Ginecologia	14000.00
Avaí	Ortopedia	13300.00
Avaí	Psiquiatria	16400.00
Bauru	Cardiologia	15800.00
Bauru	Ginecologia	13800.00

Obs. Vai agrupar primeiro pelas cidades, e depois agrupa (dentro de cada grupo de cidade) a especialidade.

Exemplo: tenho vários médicos na cidade Bauru e Agudos, e cada médico pode ter uma especialidade distinta do outro, independente da cidade. Com o SELECT acima, irá primeiro agrupar todos os médicos pela cidade. Então teríamos dois grupos, o grupo da cidade de Bauru e o grupo da cidade de Agudos.

Depois dentro desses dois grupos, irá ser criados os subgrupos separados pela especialidade. Como exemplo teríamos Bauru - cardiologia, e também poderíamos ter Bauru - dermatologia.

No caso acima que está filtrando pelo maior campo de salário, ele vai verificar dentro do grupo cidades, em cada subgrupo, qual tem o maior salário. Por isso que a especialidade se repete, somente uma vez em cada cidade, e cada cidade se repete de acordo a quantidade de especialidades diferentes que possui.

4 – Mostrar o menor salário dos médicos agrupados por especialidade e cidade.

SELECT especialidade, cidade, max(salario) as 'maior
salário'

FROM `medicos`

group by especialidade, cidade

ORDER BY `medicos`.`especialidade` ASC;

Cardiologia	Lençóis Paulista	15800.00
Cardiologia	Lucianópolis	15700.00
Clínica Geral	Arealva	12700.00
Clínica Geral	Avaí	12800.00
Clínica Geral	Lucianópolis	12500.00
Clínica Geral	Macatuba	12900.00
Clínica Geral	Piratininga	12500.00
Dermatologia	Agudos	13500.00
Dermatologia	Arealva	13500.00
Dermatologia	Lençóis Paulista	13400.00
Dermatologia	Macatuba	13600.00
Dermatologia	Piratininga	13300.75
Endocrinologia	Avaí	15500.00

Obs. Utilizei o order by , pois no MySQL não ordena automaticamente pelo group by

5 – Mostrar a soma dos salários dos médicos agrupados por cidade, mas somente dos médicos que morem em Bauru ou Agudos.

```
SELECT cidade, sum(salario) as 'soma dos salários'
FROM `medicos`
group by cidade
HAVING (upper(cidade) = 'BAURU' or upper(cidade)='AGUDOS');
```

cidade	soma dos salário
Bauru	134900.00
Agudos	141252.50

6 – Mostrar a soma dos salários dos médicos agrupados por cidade, mas somente dos salários entre 15000 e 30000.

```
SELECT cidade, sum(salario) as 'soma dos salários'
FROM `medicos`
where (salario >= 15000 and salario <=30000)
group by cidade;
```

cidade	soma dos salário
Bauru	95000.00
Pederneiras	138900.00
Arealva	126800.00
Duartina	129650.00
Macatuba	75000.00
Agudos	15800.00
Piratininga	49600.00
Lençóis Paulista	46400.00
Avaí	47700.00
Lucianópolis	79700.00

7 – Mostrar a soma dos salários dos médicos agrupados por cidade, mas somente os médicos que morem em Bauru ou agudos e também dos médicos com salário entre 15000 e 30000.

Aula 10

```
SELECT cidade, sum(salario) as 'soma dos salários'
FROM `medicos`
where (salario >= 15000 and salario <=30000)
group by cidade
HAVING (upper(cidade) = 'BAURU' or upper(cidade)='AGUDOS');
```

cidade	soma dos salário
Bauru	95000.00
Agudos	15800.00

8 – Mostrar a quantidade de médicos agrupados por especialidade.

```
SELECT especialidade, count(medico) as 'total medicos'
FROM `medicos`
group by especialidade;
```

especialidade	total medicos
Cardiologia	16
Clínica Geral	10
Dermatologia	11
Endocrinologia	5
Ginecologia	10
Neurologia	10
Oftalmologia	9
Ortopedia	9
Pediatria	10
Psiquiatria	10

9 – Mostrar a quantidade de médicos agrupados por cidade com salário entre 15000 e 30000 e que morem em jau, bauru ou agudos.

```
SELECT cidade, count(medico) as 'total medicos'
FROM `medicos`
where (salario >= 15000 and salario <=30000)
group by cidade
HAVING (upper(cidade) = 'BAURU' or upper(cidade)='AGUDOS' or upper(cidade)='JAU')
ORDER BY `medicos`.`cidade` ASC;
```

cidade	total medicos
Agudos	1
Bauru	6

Aula 10

10 – Utilizar todas as funções de agregação em um único script de agrupamento de dados.

Obs.: round = quantas casas decimais depois da virgula

```
select cidade,  
       sum(salario) as 'Total gasto por cidade',  
       round(avg(salario), 2) as 'media gasta por profissional',  
       count(medico) as 'total de especialidade disponíveis',  
       date(max(dtadm)) as 'contratado mais recente',  
       min(especialidade) as 'especialidade do último contratado'  
from medicos  
group by cidade;
```

cidade	Total gasto por cidade	media gasta por profissional	total de especialidade disponíveis	contratado mais recente	especialidade do ultimo contratado
Bauru	134900.00	14988.89	9	2022-01-25	Cardiologia
Agudos	141252.50	14125.25	10	2021-06-22	Cardiologia
Piratininga	143103.75	14310.38	10	2023-02-03	Clínica Geral
Pederneiras	166550.00	16655.00	10	2020-10-15	Endocrinologia
Lençóis Paulista	133800.00	13380.00	10	2023-06-17	Cardiologia
Arealva	167100.00	15190.91	11	2022-11-19	Cardiologia
Avaí	143800.00	14380.00	10	2023-02-27	Cardiologia
Duartina	158450.00	15845.00	10	2023-04-05	Cardiologia
Macatuba	142800.00	14280.00	10	2021-12-01	Clínica Geral
Lucianópolis	142200.00	14220.00	10	2022-08-14	Cardiologia

Aula 10

11 – Usar todas as funções de agregação em um único script de agrupamento de dados utilizando where e having.

```
select cidade, salario, especialidade,  
       sum(salario)as 'Valor total gasto salarial',  
       count(medico) as 'total de funcionarios',  
       min(dtadm) as 'ultima contratação',  
       max(idade) as 'Medico mais velho' ,  
       avg(idade) as 'idade media'  
from medicos  
where year(dtadm) >= 2022 and idade >18  
group by cidade, salario, especialidade  
HAVING  salario BETWEEN 1000.00 and 30000.00 and  
        especialidade not like 'Clínica Geral' and  
        cidade like 'bauru' or cidade like 'agudos';
```

cidade	salario	especialidade	Valor total gasto salarial	total de funcionarios	ultima contratação	Medico mais velho	idade media
Bauru	13800.00	Ginecologia	13800.00	1	2022-01-25 00:00:00	37	37.0000

Obs. Filtros por data em MySQL :

`date(dtadm) = '2025-05-21' ;` -> filtra pela data exata

`dtadm BETWEEN '2025-05-01' AND '2025-05-21';` -> pesquisa na faixa de tempo

`WHERE MONTH(dtadm) = 5 AND YEAR(dtadm) = 2025;` -> filtrar por um mês e ano específico

Aula 10

12 – Criar uma tabela com grupo, subgrupo, fornecedor, produtos e preço. Inserir pelo menos 3 produtos para cada situação (grupo, subgrupo e fornecedor). Obs. Não sei se entendi, mas está aqui :->

```
CREATE TABLE Fornecedores (  
    id INT AUTO_INCREMENT,  
    nome VARCHAR(30) NOT NULL,  
    cidade varchar(30),  
    telefone varchar(20) not null,  
    PRIMARY key (id)  
);  
  
INSERT INTO Fornecedores (nome, cidade, telefone) VALUES  
( 'TechMaster Ltda', 'Bauru', '14988123456'),  
( 'InfoParts BR', 'Agudos', '14988234567'),  
( 'CompuHouse', 'Jaú', '14988345678'),  
( 'DeltaTec', 'Lençóis Paulista', '14988456789'),  
( 'NextGen Solutions', 'Pederneiras', '14988567890'),  
( 'MegaChip', 'Bariri', '14988678901'),  
( 'Platinum Eletrônicos', 'Arealva', '14988789012'),  
( 'GigaHardware', 'Macatuba', '14988890123'),  
( 'SmartSupply', 'Igarapu do Tietê', '14988901234'),  
( 'ConnectPro', 'Avaí', '14988012345');
```

```
create table Produtos(  
    id            int            AUTO_INCREMENT /*identity(1,1)*/ unique,  
    fornecedor    int            not null,  
    preco         float          not null check(preco > 1.00),  
    entrada       datetime       DEFAULT CURRENT_TIMESTAMP /*getdate()*/,  
    validade      date           not null,  
    primary key (id),
```


Aula 10

```
FOREIGN key (fornecedor) REFERENCES Fornecedores(id)
);
```

```
INSERT INTO Produtos (fornecedor, preco, entrada, validade) VALUES
(1, 25.90, '2025-05-01 14:20:00', '2025-12-31'),
(2, 13.49, '2025-04-20 09:15:00', '2025-11-15'),
(3, 45.75, '2025-03-18 11:30:00', '2025-10-20'),
(4, 60.00, '2025-01-10 10:00:00', '2026-01-10'),
(5, 22.10, '2025-02-05 16:40:00', '2025-09-05'),
(6, 31.25, '2025-05-10 08:25:00', '2025-08-18'),
(7, 18.99, '2025-05-15 13:15:00', '2025-12-01'),
(8, 50.00, '2025-04-28 17:55:00', '2026-03-22'),
(9, 19.90, '2025-03-21 15:45:00', '2025-07-30'),
(10, 27.30, '2025-05-03 12:00:00', '2025-06-25');
```

SELECT :

```
select f.cidade, f.nome, p.id,
       round(avg(p.preco),2) as 'Preço medio ',
       if (validade <= CURRENT_TIMESTAMP,'Vencido','P/ consumo') as 'Qualidade' ,
       count(entrada) as 'Quantidade de entrada'
from Fornecedores as f, Produtos as p
where f.id = p.fornecedor
group by f.cidade, p.fornecedor, p.id;
```

Opções extras

cidade	nome	id	Preço médio	Qualidade	Quantidade de entrada
Bauru	TechMaster Ltda	1	25.9	P/ consumo	1
Bauru	TechMaster Ltda	11	29.99	P/ consumo	1
Bauru	TechMaster Ltda	21	36.1	P/ consumo	1
Bauru	TechMaster Ltda	31	38.75	P/ consumo	1
Bauru	TechMaster Ltda	41	18.9	P/ consumo	1
Agudos	InfoParts BR	2	13.49	P/ consumo	1
Agudos	InfoParts BR	12	39.5	P/ consumo	1
Agudos	InfoParts BR	22	14.99	P/ consumo	1
Agudos	InfoParts BR	32	16.2	P/ consumo	1
Agudos	InfoParts BR	42	46.75	P/ consumo	1
Jaú	CompuHouse	3	45.75	P/ consumo	1
Jaú	CompuHouse	13	12.75	P/ consumo	1
Jaú	CompuHouse	23	49.5	P/ consumo	1
Jaú	CompuHouse	33	48.9	P/ consumo	1
Jaú	CompuHouse	43	23.3	P/ consumo	1
Lençóis Paulista	DeltaTec	4	60	P/ consumo	1
Lençóis Paulista	DeltaTec	14	44.9	P/ consumo	1
Lençóis Paulista	DeltaTec	24	25	P/ consumo	1
Lençóis Paulista	DeltaTec	34	22.1	P/ consumo	1
Lençóis Paulista	DeltaTec	44	59.99	P/ consumo	1
Pederneiras	NextGen Solutions	5	22.1	P/ consumo	1
Pederneiras	NextGen Solutions	15	28.6	P/ consumo	1
Pederneiras	NextGen Solutions	25	59.9	P/ consumo	1
Pederneiras	NextGen Solutions	35	41	P/ consumo	1
Pederneiras	NextGen Solutions	45	20	P/ consumo	1

Obs.: #1055 - Expression #1 of SELECT list is not in GROUP BY clause and contains nonaggregated column 'CCMarco.f.id' which is not functionally dependent on columns in GROUP BY clause; this is incompatible with sql_mode=only_full_group_by

O group by em MySQL exige ter pelo menos um campo de todas as tabelas que estiverem no from (ou seja que estejam sendo selecionadas)