

## <Cláusula WHERE>

A cláusula **WHERE** é a responsável por filtrar resultados, utilizando parâmetros comparativos como igual, diferente, maior, menor e entre outros. Além disso, podemos usar diversos filtros em nossas consultas, inclusive adicionar parênteses para priorizar consultas dentro de escopos.

## <Comparativos>

Operador comparativo “<>” (diferente)

Operador comparativo “>” (maior que)

Operador comparativo “<” (menor que)

## <Comparativo 'LIKE'>

Para buscar itens que contenham trechos de um texto, podemos utilizar o comparativo LIKE. O comparativo LIKE vem acompanhado de uma máscara de filtro. Essa máscara pode alterar de um SGBD para o outro, nesse exemplo utilizaremos a máscara “%”.

```
SELECT * FROM CLIENTES  
  
WHERE RAMO LIKE '%info%';
```

## <Juntando comparativos>

### Operador lógico “AND”

O operador “AND” verifica as condições como um todo e retorna os resultados que atendem a todas as condições.

```
SELECT * FROM clientes
```

```
where ramo = 'TI' AND nr_funcionarios > 10
```

## <Juntando comparativos>

### Operador lógico “OR”

O operador “OR” verifica separadamente as condições e retorna os resultados que atendem pelo menos uma das condições.

```
SELECT * FROM clientes
```

```
where ramo = 'TI' OR nr_funcionarios = 10;
```

## <Uso de parênteses>

Operador lógico “OR”

Os parênteses são utilizados para dar prioridade a uma condição dentro de uma consulta.

```
SELECT * FROM clientes
```

```
where nr_funcionarios > 10
```

```
AND (nome = 'johnny' OR 'joão');
```

# <Cláusula ORDER BY>

## Cláusula ORDER BY

Hora de ordenar os nossos resultados com a cláusula ORDER BY. Inicialmente, vamos ordenar de modo **ascendente (ASC)** e depois de forma **descendente (DESC)**

## <Cláusula ORDER BY>

```
SELECT * FROM produto  
ORDER BY descricao ASC;
```

```
SELECT * FROM produto  
ORDER BY descricao DESC;
```



## <Funções de Agregação>

Funções de agregação são funções SQL que permitem executar uma operação aritmética nos valores de uma coluna em todos os registros de uma tabela.

Retornam um valor simples baseado em um conjunto de valores de entrada.

## <Funções de Agregação>

As principais funções de agregação (mais comuns) em MySQL são as seguintes:

**MIN** = Valor Mínimo de um conjunto de valores

**MAX** = Valor Máximo de um conjunto de valores

**AVG** = Média Aritmética de um conjunto de valores

**SUM** = Total (Soma) de um conjunto de valores

**COUNT** = Contar quantidade total de itens


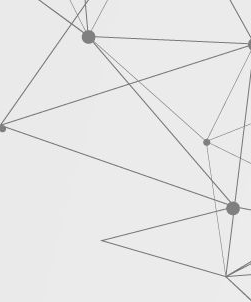
## <Funções de Agregação>

A funções **SUM** e **AVG** somente aceitam como entrada um conjunto de números; já as demais funções podem operar também com outros tipos de dados não-numéricos, como por exemplo strings (caracteres) ou datas.

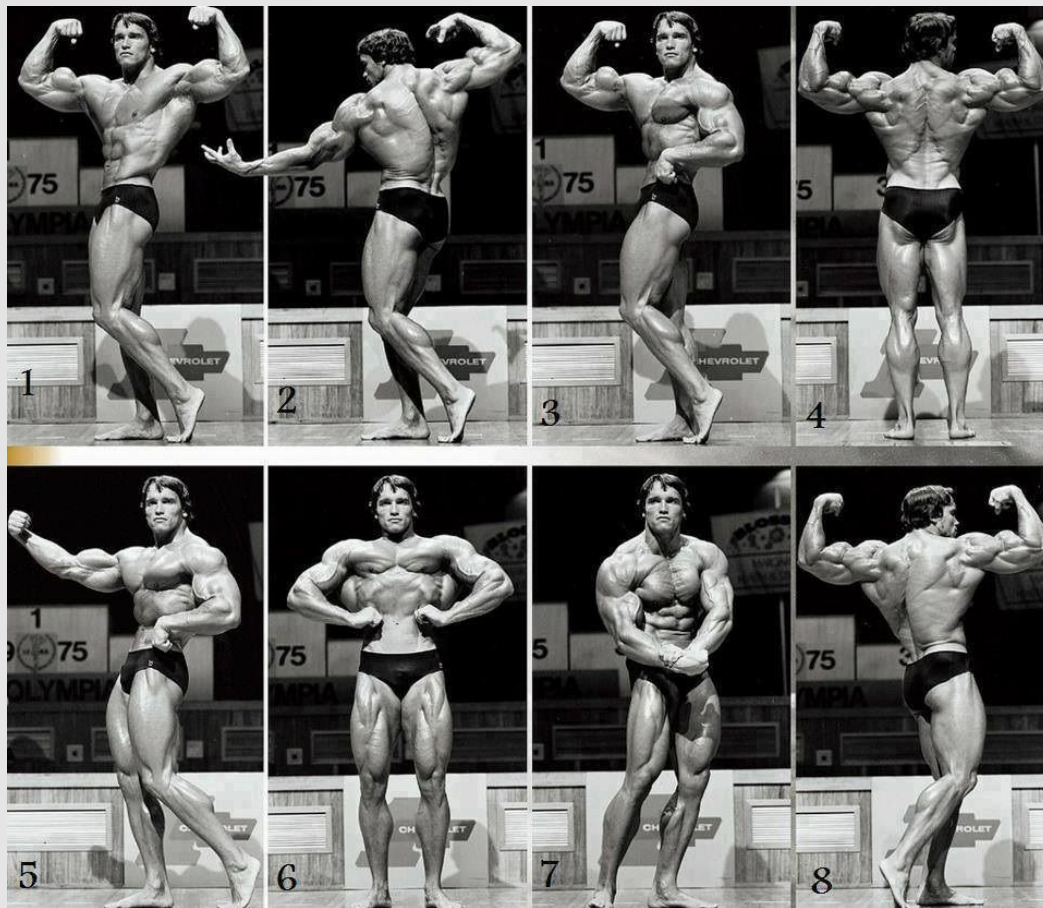


## <Funções de Agregação>

**DISTINCT** – Usa apenas valores distintos (sem repetição) ao avaliar a função.



# <Exercicios>



## <Exercícios>

1) retornar os produtos da Classificação 003 e que a unidade de medida não seja 'UN'

R: 139 registros;

2) Retornar os produtos da Classificação 003, com a unidade de medida 'UN' em que a quantidade seja entre 5 e 7 com o valor menor que 10;

R: 27 registros;



## <Exercícios>



3) Valor total dos 'biscoito' da base de dados;  
R: 3021;

---

4) Validar se existe algum 'martelo' que não pertença a classificação material de Construção;

## <Exercícios>

5) Retornar os produtos da classificação EPI que estejam em menos de 5 caixas;

R: 2 registros;

6) Retornar os produtos da Classificação EPI que NÃO ESTEJA em caixas e sua quantidade esteja em 10 e 50;

R:9 registros;



## <Exercícios>

7) Retornar todos registros da classificação UNIFORMES com o nome 'camiseta' e todos os produtos da classificação MATERIAL ESPORTIVO e com nome 'bola'

R: 11 registros;

8) Retornar a média do valor dos produtos que a quantidade esteja entre 2 e 4, com valor inferior a 50, que não seja material de construção e que não seja um 'copo';

R: 18.8688

## <Exercícios>

9) Retornar o quantidade total de pacotes ( PCT) dos produtos alimenticios

R: 1165;

10) Retornar apenas o numero total de produtos cadastrados com unidade pacote e que seja da classificação de alimentos

R: 23 produtos;

## <Exercícios>

11) Retornar qual é o maior valor de um produto do estoque, este deve ser o produto que sua quantidade \* valor seja o maior

R: 1134870;

12) Retornar o menor valor de um produto que a quantidade seja maior que 0 e que a unidade seja 'UN' e classificação alimentos

R: 1;

## <Exercicios>

13) Retornar é o valor total dos produtos da categoria 'Material Hospitalares'

R: 406355;

14) Retornar TODOS os valores totais por categoria e ordenar por categoria

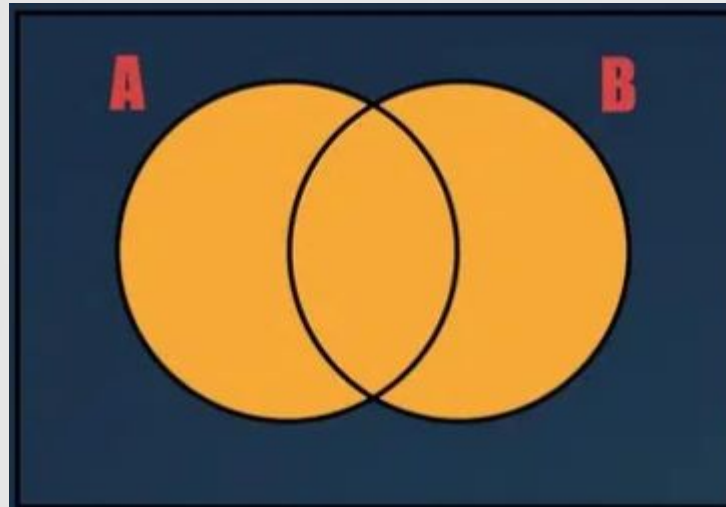
## <Exercícios>

15) Retornar todos os tipos de 'UNIDADE' da classificação Veterinária  
R: 12;

16) Contar Quantos produtos são da categoria de Aviamentos por unidade. EX: (20 produtos - UN; 2 PRODUTOS - PCT)

## <Join>

INNER JOIN - A cláusula JOIN é uma forma de relacionar duas tabelas e gerar uma tabela resultante

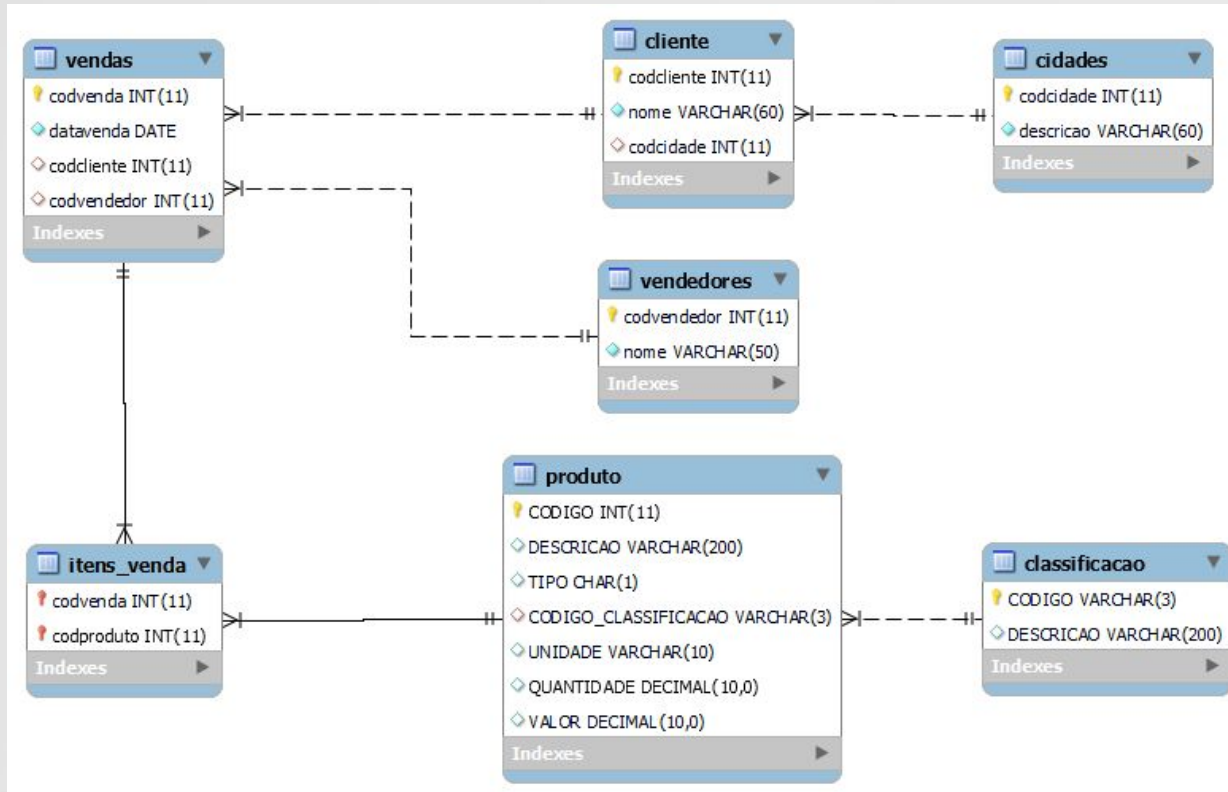


## <Join>

```
SELECT *  
FROM produto p  
  join classificacao c on p.CODIGO_CLASSIFICACAO = c.CODIGO
```

# <DER>

## Diagrama de entidade relacionamento







## <Join>



- 1) Qual Cliente foi a venda ?
  - 2) Qual Cidade é o cliente ?
  - 3) Quais produtos ele comprou ?
  - 4) Qual quantidade de cada produto ele comprou ?
  - 5) Qual é Classificação do produto ?
  - 6) Qual nome vendedor ?
  - 7) Qual valor total de cada produto ?
-