

# Consultas SQL: Avançando no SQL com MySQL

- LIKE
  - Pega tudo que contem %nome%, que começa com nome%, ou que termina com %nome

```
SELECT NOME FROM tabela_de_clientes WHERE NOME LIKE '%Carvalho%'; // lista os clientes que tem carvalho no nome
```

- Quantos clientes possuem o último sobrenome Mattos?

```
SELECT COUNT(*) FROM tabela_de_clientes WHERE NOME LIKE '%Mattos%'
```

- 
- DISTINCT
    - Não repete coluna

```
SELECT DISTINCT SABOR FROM tabela_de_produtos; //lista só os sabores sem repetir nenhum
```

- Quais são os bairros da cidade do Rio de Janeiro que possuem clientes?

```
SELECT DISTINCT BAIRRO FROM tabela_de_clientes WHERE CIDADE = 'Rio de Janeiro'
```

- 
- LIMIT
    - LIMIT 2,3 pega a partir da linha 2, os 3 proximos; LIMIT 2: pega os 2 primeiros

```
SELECT DISTINCT SABOR FROM tabela_de_produtos LIMIT 1,4
```

- Queremos obter as 10 primeiras vendas do dia 01/01/2017. Qual seria o comando SQL para obter este resultado?

```
SELECT * FROM notas_fiscais WHERE DATA_VENDA = '2017-01-01' LIMIT 10
```

- 
- ORDER BY
    - ASC, DESC

```
SELECT * from tabela_de_clientes WHERE IDADE > 20 HAVING SEXO = 'M'; //deixa em ordem alfabética
```

---

- GROUP BY
  - SUM
  - MAX
  - MIN
  - AVG
  - COUNT

```
SELECT COUNT(*), IDADE FROM tabela_de_clientes GROUP BY IDADE; //mostra a quantidade de pessoas com determinada idade
```

- 
- HAVING
    - Igual WHERE, só que depois do GROUP BY

```
SELECT * from tabela_de_clientes WHERE IDADE > 20 HAVING SEXO = 'M'
```

- 
- CASE
    - Vem "dentro" do SELECT, antes do from
    - WHEN
    - THEN
    - END

```
SELECT NOME, IDADE,  
CASE WHEN IDADE < 20 THEN 'NOVO'  
WHEN IDADE >=20 AND IDADE <30 THEN 'JOVEM'  
ELSE 'VELHO'  
END AS 'CLASSIFICAÇÃO ETÁRIA'  
FROM tabela_de_clientes  
ORDER BY IDADE
```

- 
- JOIN
    - INNER JOIN - pega só os que têm correspondência
    - LEFT JOIN - pega todos da esquerda e os que têm correspondência na direita
    - RIGHT JOIN - contrário de left
    - FULL JOIN - pega todos das 2 tabelas
    - CROSS JOIN - produto carteziano (nao escreve cross join, só coloca vírgula)

```
SELECT YEAR(DATA_VENDA), SUM(QUANTIDADE * PRECO) AS FATURAMENTO  
FROM notas_fiscais NF INNER JOIN itens_notas_fiscais INF  
ON NF.NUMERO = INF.NUMERO
```

```
GROUP BY YEAR(DATA_VENDA)
```

//AGRUPAR POR CLASSIFICAÇÃO ETÁRIA E MOSTRAR QUANTIDADE DE CADA UMA:

```
SELECT COUNT(*),  
CASE WHEN IDADE < 20 THEN 'NOVO'  
WHEN IDADE >=20 AND IDADE <30 THEN 'JOVEM'  
ELSE 'VELHO'  
END AS 'CLASSIFICACAO'  
FROM tabela_de_clientes  
GROUP BY CLASSIFICACAO
```

- 
- UNION
    - UNION ALL (considera repetidos)
- 

- SUB CONSULTAS
- 

- VIEW
    - Cria uma tabela nova com nome diferente
- 

- Usando algumas funções
- Queremos construir um SQL cujo resultado seja, para cada cliente: "O cliente João da Silva faturou 120000 no ano de 2016". Somente para o ano de 2016.

```
SELECT CONCAT('O cliente ', TDC.NOME, ' faturou ', SUM(INF.QUANTIDADE * INF.PRECO), '  
no ano de 2016')  
FROM notas_fiscais NF  
INNER JOIN itens_notas_fiscais INF ON NF.NUMERO = INF.NUMERO  
INNER JOIN tabela_de_clientes TDC ON NF.CPF = TDC.CPF  
WHERE YEAR(DATA_VENDA) = 2016  
GROUP BY NOME;
```