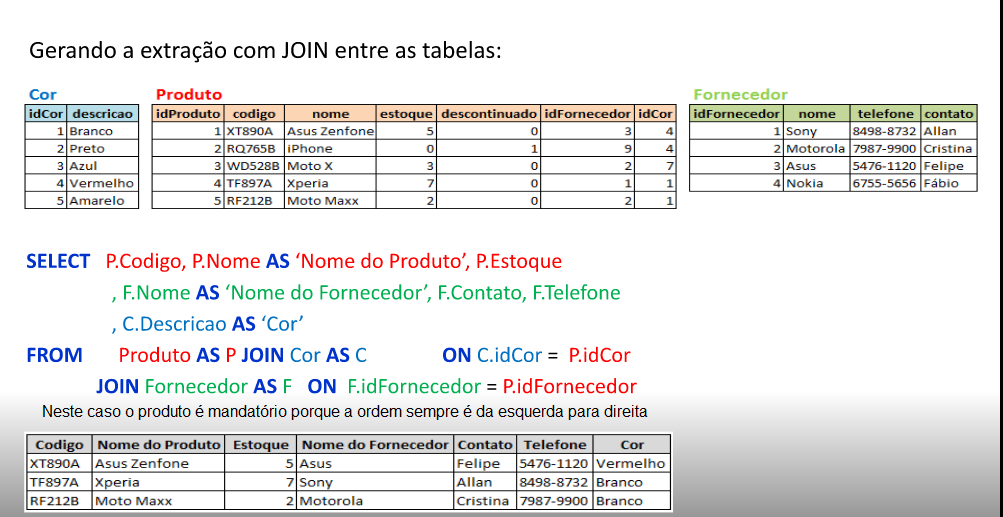
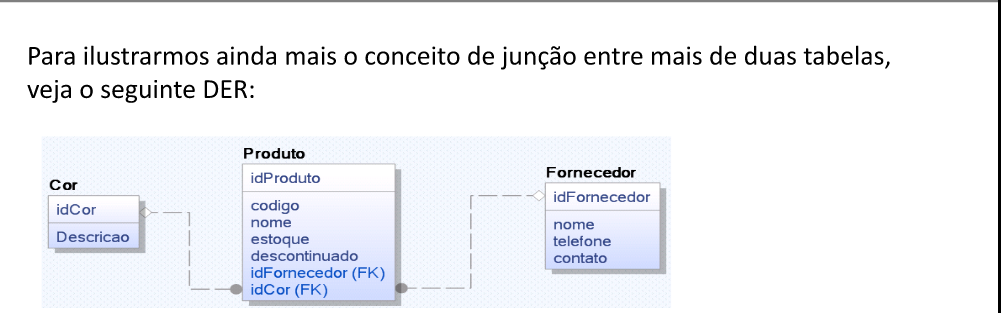
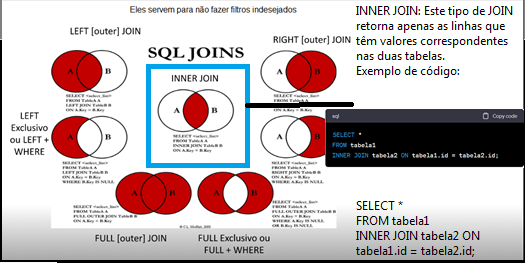
**Relatório da aula 10**

**DQL (Data Query Language) Exemplos de scripts das aulas anteriores**

****

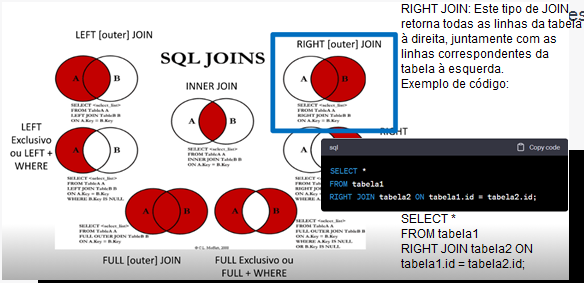
**Minha digitação:** SELECT p.codigo, p.nome AS’Nome do Produto’, P.Estoque,

f.nome AS’Nome do Fornecedor’, f.Contato, f,Telefone,

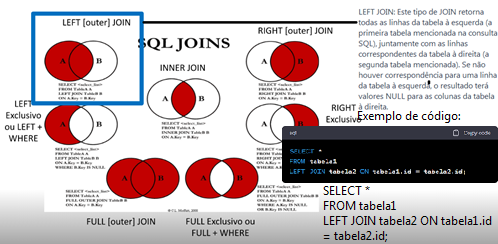
c.descrição AS ‘Cor’

FROM Produto AS p JOIN Cor AS c ON c.idcor = p.idcor

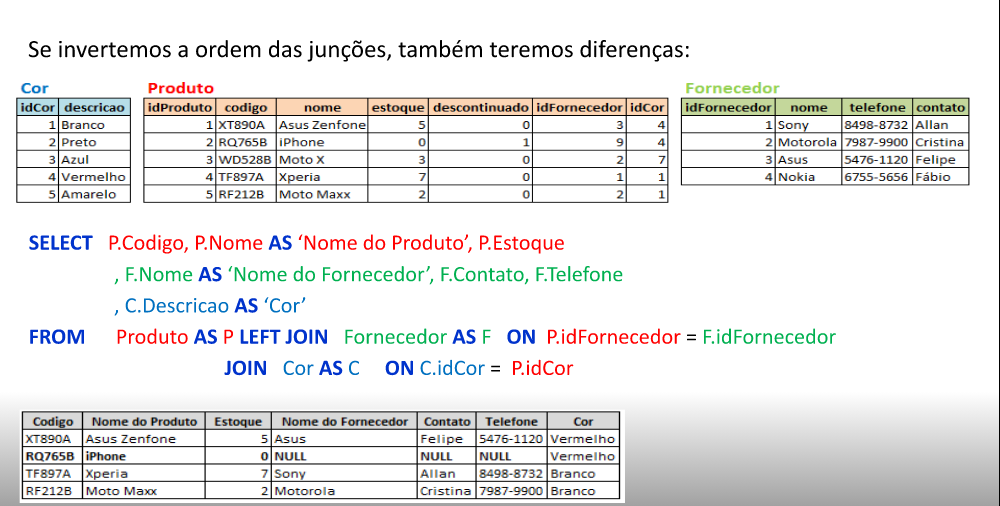
JOIN Fornecedor AS f on f.idFornecedor = p.idFornecedor;

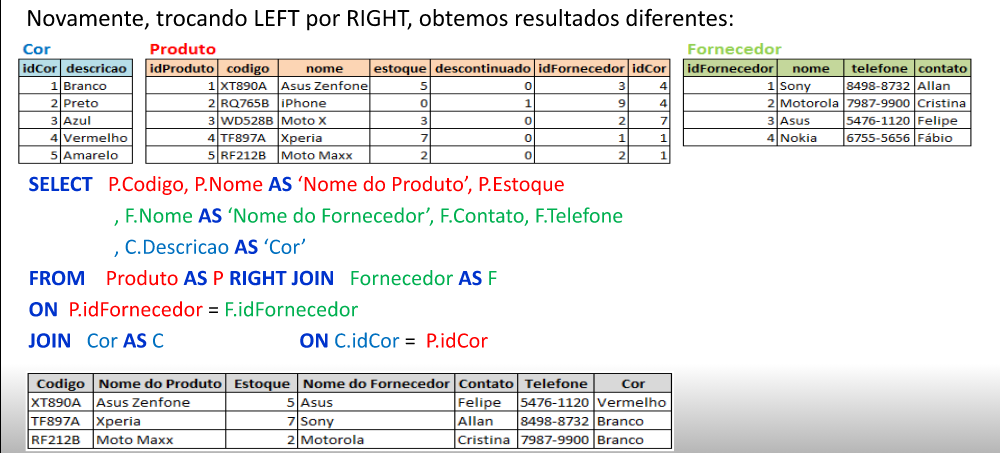
****



****

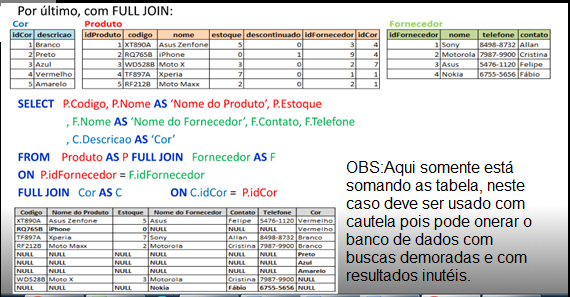


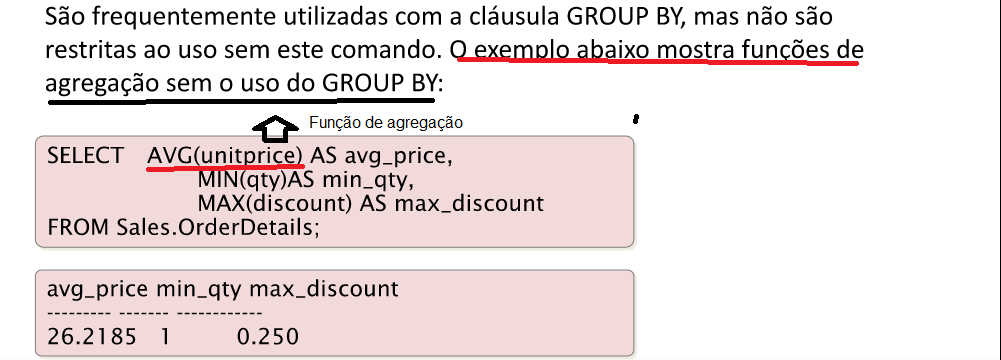
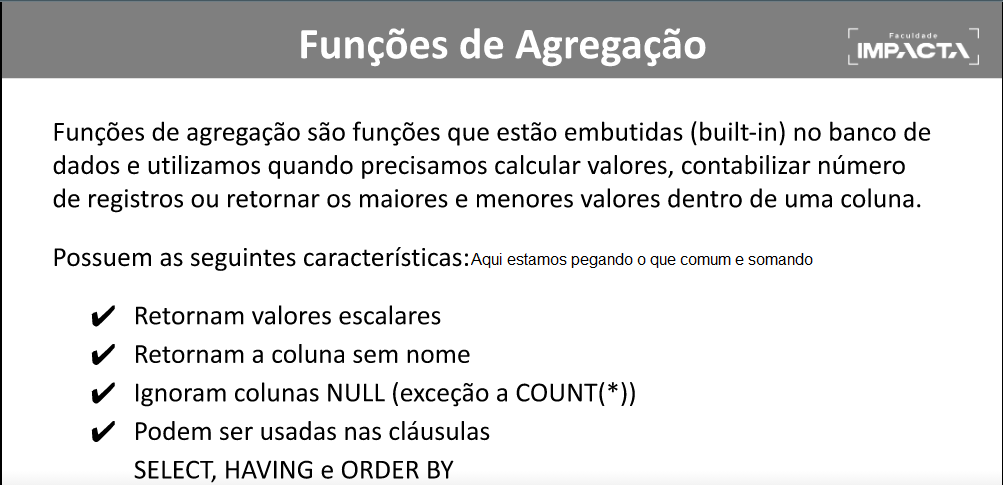








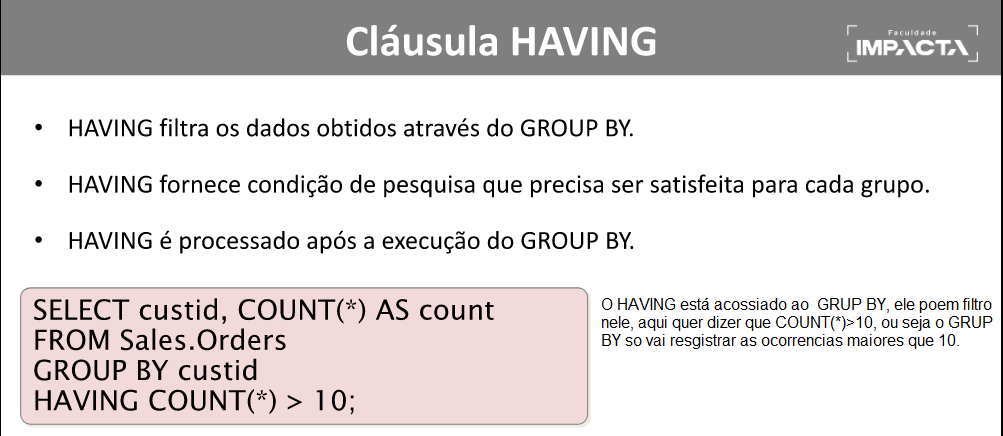
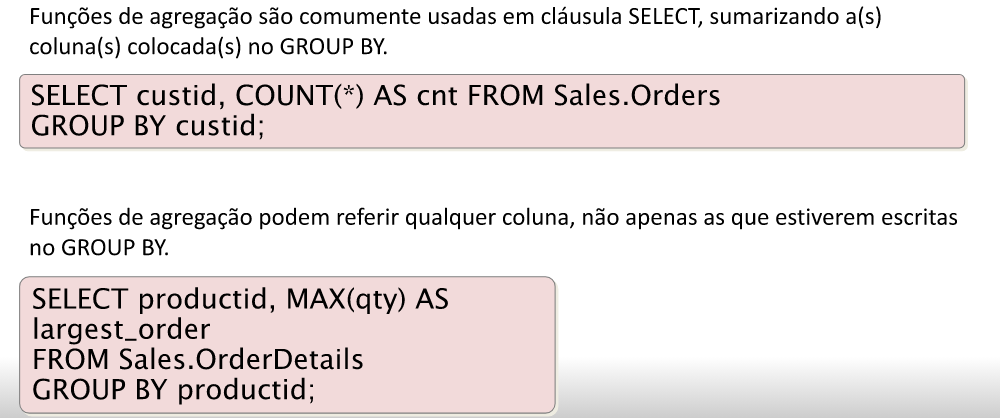
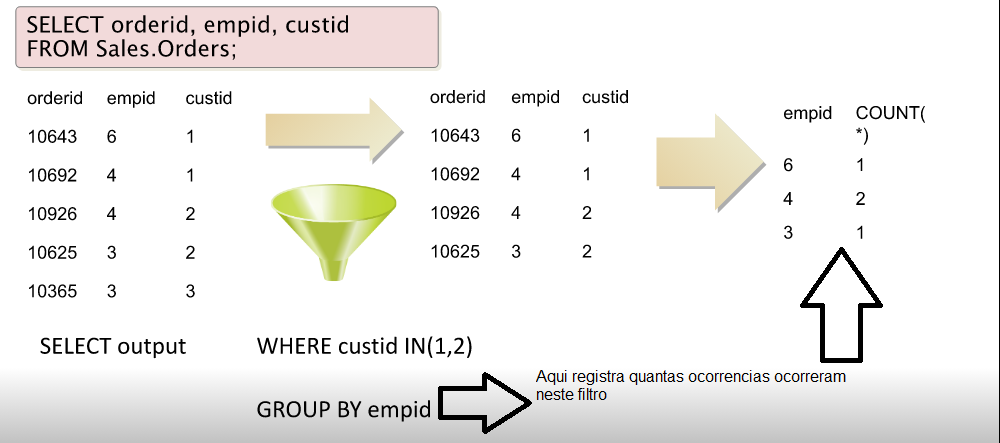
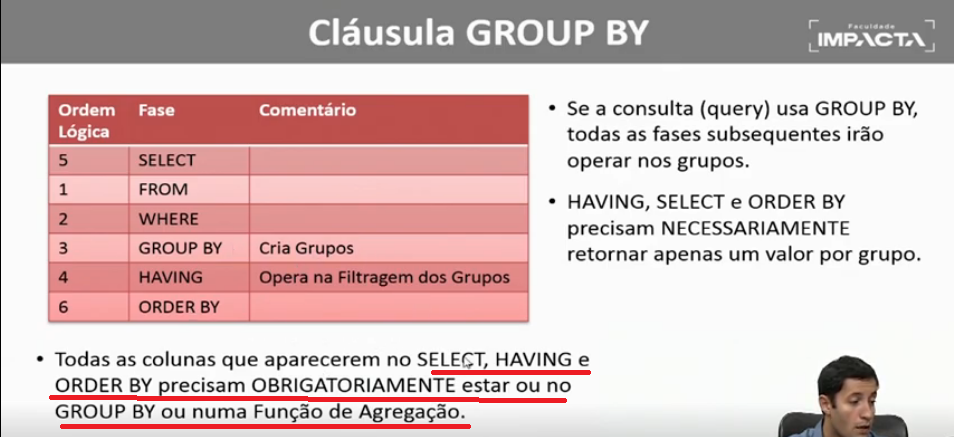
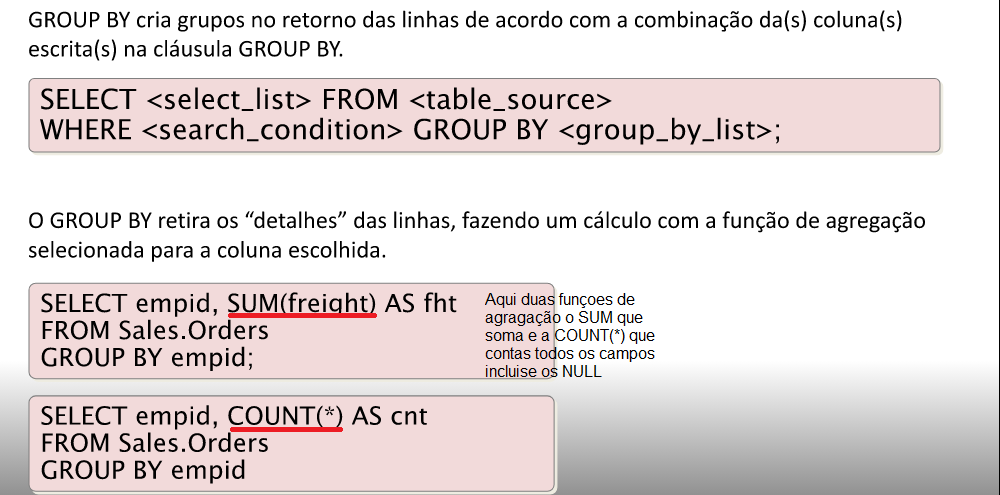
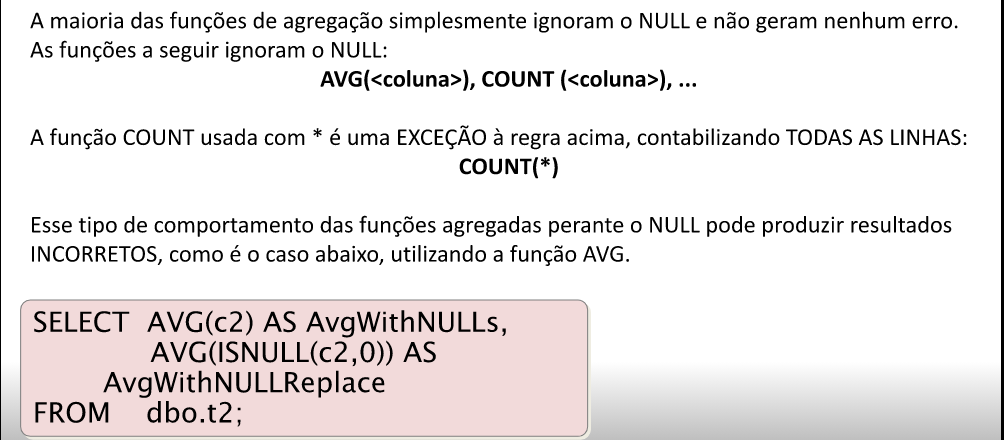
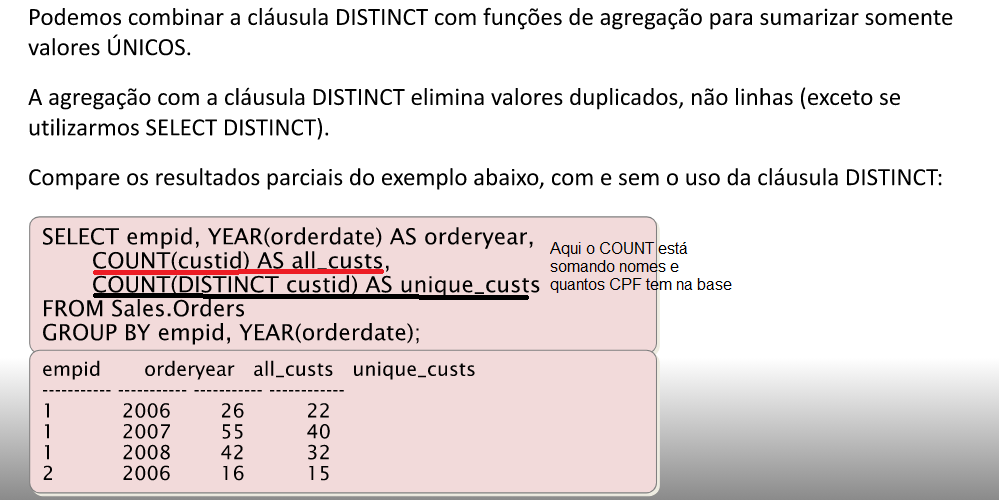
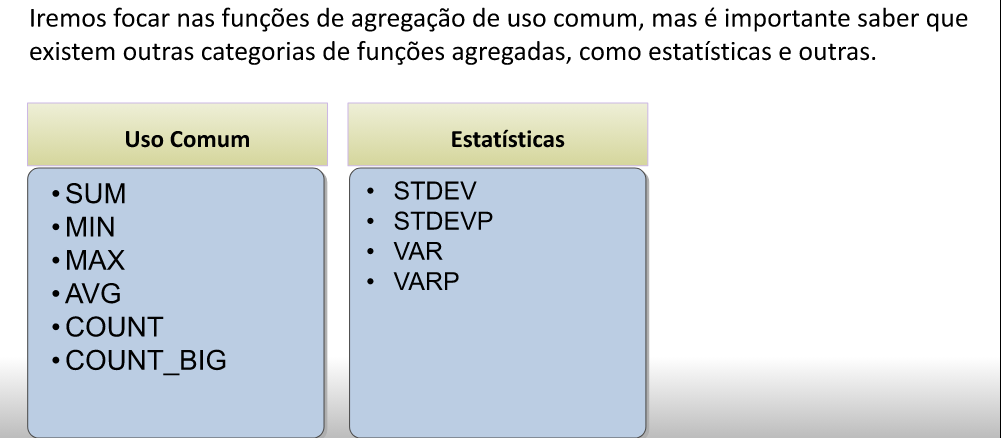




As funções de agregação são usadas para executar operações em conjunto de dados, geralmente para produzir um resultado resumido. As funções de agregação mais comuns em SQL são:

1. COUNT(): Esta função é usada para contar o número de linhas em uma tabela ou o número de valores não nulos em uma coluna.
2. SUM(): Esta função é usada para calcular a soma dos valores em uma coluna.
3. AVG(): Esta função é usada para calcular a média dos valores em uma coluna.
4. MIN(): Esta função é usada para encontrar o valor mínimo em uma coluna.
5. MAX(): Esta função é usada para encontrar o valor máximo em uma coluna.

Todas essas funções podem ser usadas com a cláusula GROUP BY para agrupar os resultados por uma ou mais colunas. Por exemplo, se quisermos contar o número de clientes por país em uma tabela de clientes, podemos usar a função COUNT() com a cláusula GROUP BY no país.



Artigo :

<https://www.devmedia.com.br/desvendando-a-clausula-group-by-artigo-sql-magazine-47/8082>

**Pesquisa ChatGPT:**

Para escrever um script SQL que usa o comando JOIN, siga os seguintes passos:

1. Selecione as tabelas que você deseja unir. Certifique-se de que elas têm pelo menos uma coluna em comum, que será usada como a chave para a junção.

2. Escolha o tipo de junção que deseja usar. Os tipos mais comuns são INNER JOIN, LEFT JOIN, RIGHT JOIN e FULL OUTER JOIN. Cada tipo de junção tem um comportamento diferente, por isso escolha o que melhor se adapte às suas necessidades.

3. Escreva o comando JOIN, seguido do tipo de junção que escolheu. Em seguida, especifique as tabelas que deseja unir e as colunas em comum que serão usadas como chave para a junção. A sintaxe básica para um comando JOIN é a seguinte:

```

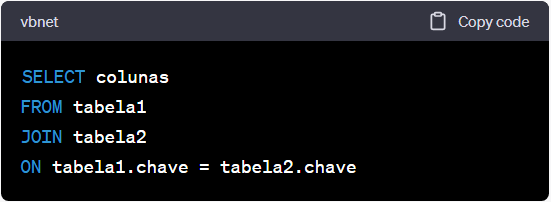
SELECT colunas

FROM tabela1

JOIN tabela2

ON tabela1.chave = tabela2.chave

```



4. Personalize o comando JOIN de acordo com suas necessidades. Você pode adicionar condições adicionais usando as cláusulas WHERE e HAVING, filtrar as colunas usando a cláusula SELECT e agrupar os resultados usando a cláusula GROUP BY.

Aqui está um exemplo de um comando JOIN usando a tabela "clientes" e a tabela "pedidos":

```

SELECT clientes.nome, pedidos.produto

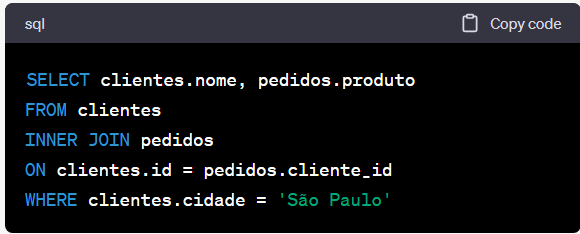
FROM clientes

INNER JOIN pedidos

ON clientes.id = pedidos.cliente\_id

WHERE clientes.cidade = 'São Paulo'

```



Neste exemplo, estamos usando um INNER JOIN para unir as tabelas "clientes" e "pedidos" usando a coluna "id" da tabela "clientes" e a coluna "cliente\_id" da tabela "pedidos". Em seguida, estamos selecionando apenas os clientes de São Paulo e o nome do produto de cada pedido.

**Exercicíos Para praticar:**

**Sim, aqui está um exercício com script SQL que usa o comando JOIN para você praticar:**

**Suponha que você tenha duas tabelas:**

**```**

**tabela1: alunos**

**+----+-----------+------------+**

**| id | nome | curso |**

**+----+-----------+------------+**

**| 1 | João | Engenharia |**

**| 2 | Maria | Medicina |**

**| 3 | Pedro | Direito |**

**| 4 | Ana | Engenharia |**

**| 5 | Luiza | Arquitetura|**

**+----+-----------+------------+**

**tabela2: notas**

**+----+-----------+-------+**

**| id | disciplina| nota |**

**+----+-----------+-------+**

**| 1 | Matemática| 8.5 |**

**| 1 | Física | 9.0 |**

**| 2 | Matemática| 7.0 |**

**| 2 | Física | 6.5 |**

**| 3 | Matemática| 10.0 |**

**| 3 | Física | 9.0 |**

**| 4 | Matemática| 6.0 |**

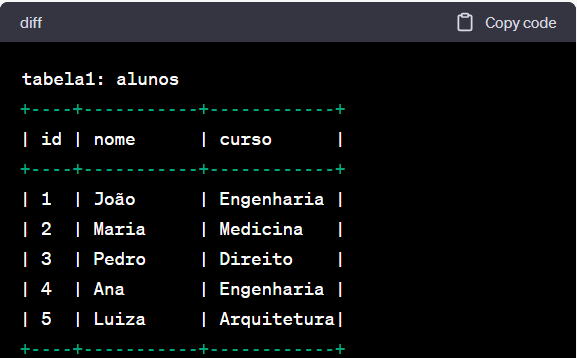
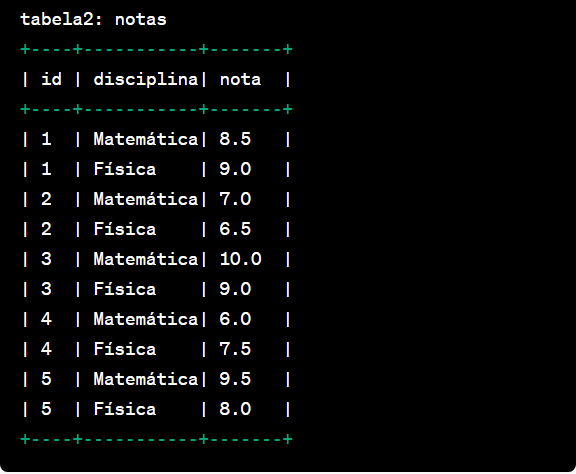
**| 4 | Física | 7.5 |**

**| 5 | Matemática| 9.5 |**

**| 5 | Física | 8.0 |**

**+----+-----------+-------+**

**```**

** **

**Escreva um script SQL que retorna o nome, curso e nota média dos alunos que cursam Engenharia, ordenados pelo nome do aluno.**

**Sintaxe básica:**

**```**

**SELECT colunas**

**FROM tabela1**

**JOIN tabela2**

**ON tabela1.chave = tabela2.chave**

**```**

SELECT alunos.nome, alunos.curso, AVG (media.notas) AS medias.notas

FROM alunos

INNER JOIN notas

ON alunos.id = notas.idalunos AND alunos.curso = ‘Engenharia’

GRUPO BY aluno.nome, aluno.curso

ORDER BT aluno,nome

**Para resolver esse exercício, precisamos unir as duas tabelas usando a coluna "id" e "curso". Em seguida, agrupamos os resultados por aluno para calcular a média de suas notas. Aqui está o script SQL que resolve o exercício:**

**```**

**SELECT alunos.nome, alunos.curso, AVG(notas.nota) AS media\_notas**

**FROM alunos**

**INNER JOIN notas**

**ON alunos.id = notas.id AND alunos.curso = 'Engenharia'**

**GROUP BY alunos.nome, alunos.curso**

**ORDER BY alunos.nome**

**```**

**Explicação:**

**- Na cláusula SELECT, selecionamos o nome e o curso do aluno, bem como a média de suas notas. Usamos a função AVG() para calcular a média.**

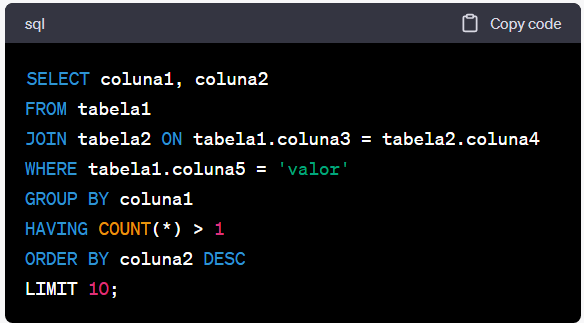
**- Na cláusula FROM, selecionamos as tabelas "alunos" e "notas".**

**- Na cláusula JOIN, unimos as tabelas "alunos" e "notas" usando as colunas "id" e "curso". Usamos um INNER JOIN porque queremos apenas os alunos que cursam Engenharia.**

**- Na cláusula GROUP BY, agrupamos os resultados por aluno.**

**- Na cláusula ORDER BY, ordenamos os resultados pelo nome do aluno.**

**Uma query com todos as clausulas**

****

**SELECT coluna1, coluna2**

**FROM tabela1**

**JOIN tabela2 ON tabela1.coluna3 = tabela2.coluna4**

**WHERE tabela1.coluna5 = 'valor'**

**GROUP BY coluna1**

**HAVING COUNT(\*) > 1**

**ORDER BY coluna2 DESC**

**LIMIT 10;**

Nesta query, estamos selecionando as colunas **coluna1** e **coluna2** das tabelas **tabela1** e **tabela2**, respectivamente. Estamos usando a cláusula **JOIN** para unir as duas tabelas, relacionando as colunas **coluna3** da **tabela1** com a **coluna4** da **tabela2**.

Em seguida, estamos usando a cláusula **WHERE** para filtrar os resultados onde a **coluna5** da **tabela1** é igual a um determinado valor.

Estamos então agrupando os resultados pela **coluna1** usando a cláusula **GROUP BY**. E estamos usando a cláusula **HAVING** para selecionar apenas as linhas onde há mais de uma ocorrência do valor da **coluna1**.

Finalmente, estamos ordenando os resultados em ordem decrescente da **coluna2** usando a cláusula **ORDER BY**, e limitando o número de resultados retornados para 10 usando a cláusula **LIMIT**.