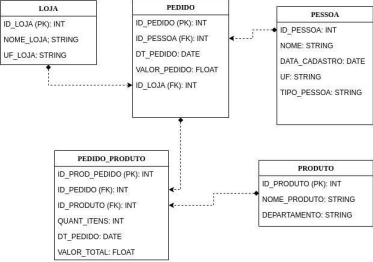
**Questões SQL**

*Diagrama para as questões 1 e 2*

**

Informações adicionais

Pessoa:

● UF: 27 unidades federativas

● TIPO\_PESSOA: PF ou PJ (Pessoa física ou Pessoa Jurídica)

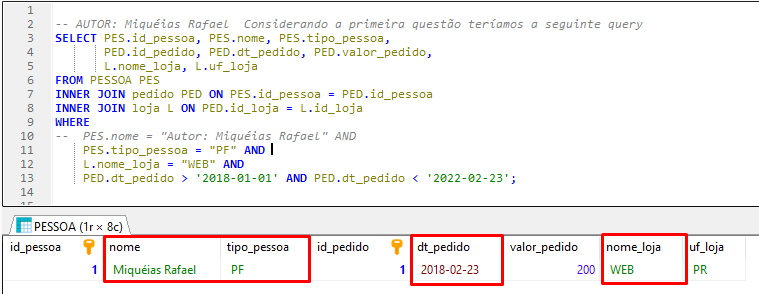
Pedido\_Produto:

● VALOR\_TOTAL: É o valor total já multiplicado pela QUANT\_ITENS Loja:

● Existem lojas em todos as UF do Brasil

A partir do modelo exemplificado acima, escreva em SQL as seguintes solicitações:

1. Montar uma consulta SQL para selecionar todas as pessoas físicas que compraram na loja de nome “WEB” e no período de 2018 até a data presente, as seguintes informações:



*-- AUTOR: Miquéias Rafael Considerando a primeira questão teríamos a seguinte query*

**SELECT** PES.id\_pessoa, PES.nome, PES.tipo\_pessoa,

PED.id\_pedido, PED.dt\_pedido, PED.valor\_pedido,

L.nome\_loja, L.uf\_loja

**FROM** PESSOA PES

**INNER** **JOIN** pedido PED **ON** PES.id\_pessoa = PED.id\_pessoa

**INNER** **JOIN** loja L **ON** PED.id\_loja = L.id\_loja

**WHERE**

*-- PES.nome = "Autor: Miquéias Rafael" AND*

PES.tipo\_pessoa = "PF" **AND**

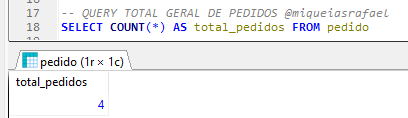
L.nome\_loja = "WEB" **AND**

PED.dt\_pedido > '2018-01-01' **AND** PED.dt\_pedido < '2022-02-23';

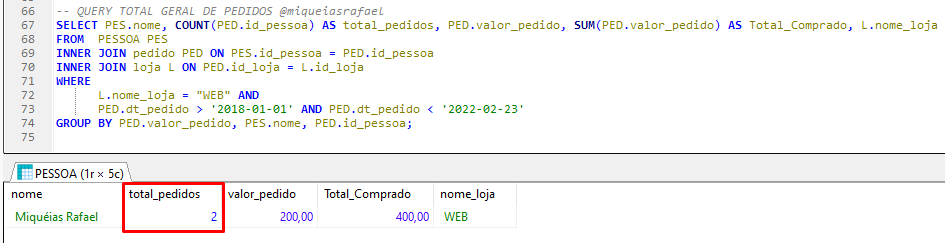
○ Total de pedidos;

A tabela possui um total de 4 pedidos de todos os usuários

**SELECT** **COUNT**(\*) **AS** total\_pedidos **FROM** pedido

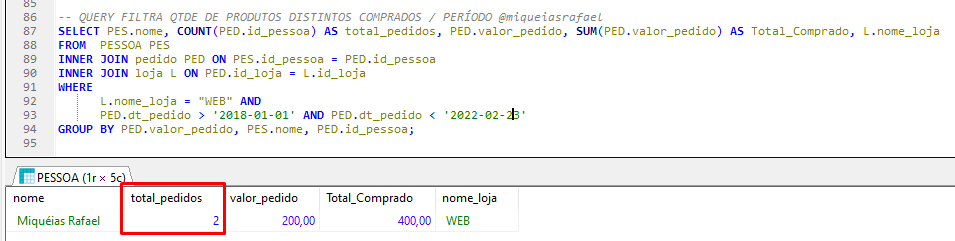


Porém os usuários filtrados do período de 2018 até o dia 23/02/2022

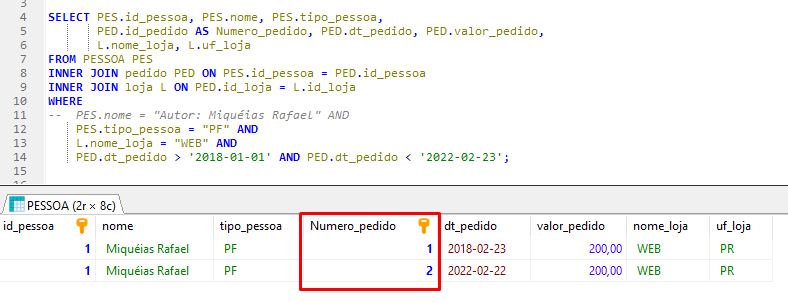


○ Quantidade de Produtos distintos comprados;

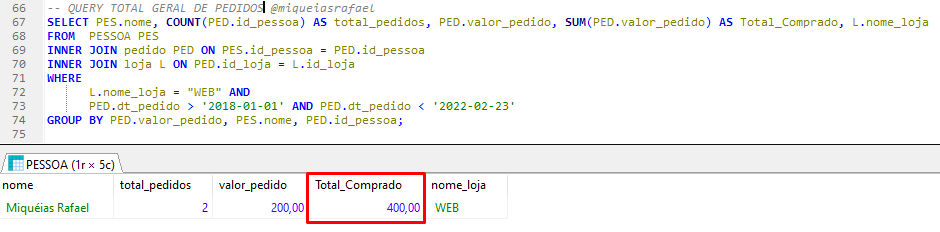
Informando apenas o valor da quantidade de produtos distintos comprados foram 2



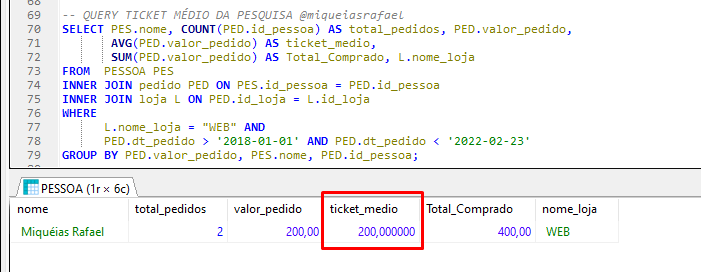
Informando o número do pedido



○ Valor total comprado;



○ Ticket médio (Valor / Pedidos);



*--QUERY TICKET MÉDIO DA PESQUISA @miqueiasrafael*

**SELECT** PES.nome, **COUNT**(PED.id\_pessoa) **AS** total\_pedidos, PED.valor\_pedido,

**AVG**(PED.valor\_pedido) **AS** ticket\_medio,

**SUM**(PED.valor\_pedido) **AS** Total\_Comprado, L.nome\_loja

**FROM** PESSOA PES

**INNER** **JOIN** pedido PED **ON** PES.id\_pessoa = PED.id\_pessoa

**INNER** **JOIN** loja L **ON** PED.id\_loja = L.id\_loja

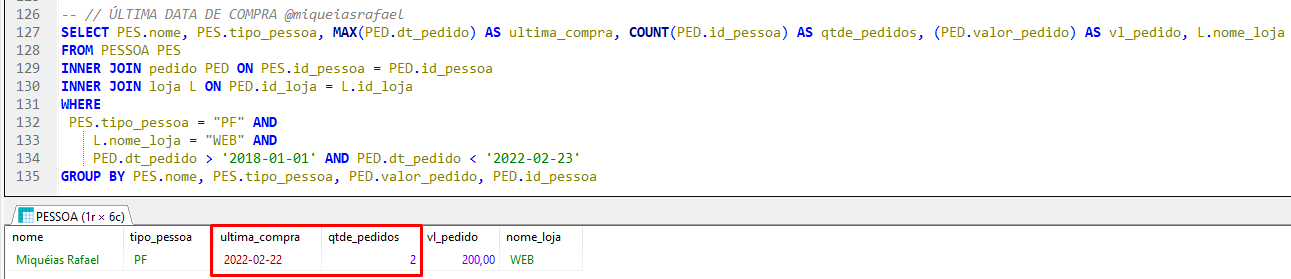
**WHERE**

L.nome\_loja = "WEB" **AND**

PED.dt\_pedido > '2018-01-01' **AND** PED.dt\_pedido < '2022-02-23'

**GROUP** **BY** PED.valor\_pedido, PES.nome, PED.id\_pessoa;

○ Última data de compra;



*-- // ÚLTIMA DATA DE COMPRA @miqueiasrafael*

**SELECT** PES.nome, PES.tipo\_pessoa, **MAX**(PED.dt\_pedido) **AS** ultima\_compra, **COUNT**(PED.id\_pessoa) **AS** qtde\_pedidos, (PED.valor\_pedido) **AS** vl\_pedido, L.nome\_loja

**FROM** PESSOA PES

**INNER** **JOIN** pedido PED **ON** PES.id\_pessoa = PED.id\_pessoa

**INNER** **JOIN** loja L **ON** PED.id\_loja = L.id\_loja

**WHERE**

PES.tipo\_pessoa = "PF" **AND**

L.nome\_loja = "WEB" **AND**

PED.dt\_pedido > '2018-01-01' **AND** PED.dt\_pedido < '2022-02-23'

**GROUP** **BY** PES.nome, PES.tipo\_pessoa, PED.valor\_pedido, PED.id\_pessoa;

Exemplo de retorno:

|  |  |  |
| --- | --- | --- |
| **PK\_ID\_PESSOA** | **TOTAL\_PEDIDOS** | **PRODUTOS\_DISTINTOS VALOR\_TOTAL TICKET\_MEDIO ÚLTIMA\_COMPRA** |
| 83735 | 3 | 3 1500,00 500,00 05/02/2019 |
| 196698 | 2 | 4 5024,90 2512,45 24/03/2020 |
| 309661 | 1 | 3 520,42 520,42 19/11/2019 |
| 422624 | 2 | 2 955,95 477,98 05/07/2018 |
| 535587 | 1 | 1 248,60 248,60 30/09/2019 |
| 648550 | 7 | 14 22542,55 3220,36 01/02/2020 |
| 761513 | 2 | 3 3058,93 1529,47 01/05/2020 |

2. Montar uma consulta para trazer agrupado **TODAS** as UF de quem comprou nos departamentos:

○ 4 - Móveis de Cozinha

○ 7 - Móveis de Sala de Estar

O valor total desses produtos deve ser maior que R$1000,00.

Para cada UF de Loja, independente se teve dados de venda ou não, traga as seguintes informações:

○ Valor Total;

○ Valor médio;

○ Última compra

Exemplo de retorno:

|  |  |
| --- | --- |
| **UF\_LOJA** | **VALOR\_TOTAL** |
| AC | NULL |
| AL | 1559578,7 |
| AP | NULL |
| AM | NULL |
| BA | 2595729,48 |
| CE | 1874245,32 |
| DF | 5098242,42 |
| ES | 984242,52 |
| GO | 742523,52 |
| MA | 2235498,99 |

**VALOR\_MEDIO ÚLTIMA\_COMPRA** NULL NULL

1220,55 10/05/2020

NULL NULL

NULL NULL

1150,45 11/05/2020

1098,43 09/05/2020

1842,42 15/05/2020

850,24 01/04/2020

1196,225 11/05/2020 1180,561 09/05/2020

MT 2074452,612 1164,897 15/05/2020 MS 2235498,99 1149,233 01/04/2020

|  |  |
| --- | --- |
| MG | 2074452,612 |
| PA | 1913406,234 |
| PB | 1752359,856 |
| PR | NULL |
| PE | NULL |
| PI | NULL |
| RJ | 8427562,32 |
| RN | 742523,52 |
| RS | 6273492,83 |
| RO | NULL |
| RR | NULL |
| SC | 5242525,2 |
| SP | 10098242,42 |
| SE | NULL |
| TO | NULL |

1133,569 11/05/2020 1117,905 09/05/2020 1102,241 15/05/2020 NULL NULL

NULL NULL

NULL NULL

2142,52 15/05/2020

934,23 28/04/2020

1723,63 12/05/2020

NULL NULL

NULL NULL

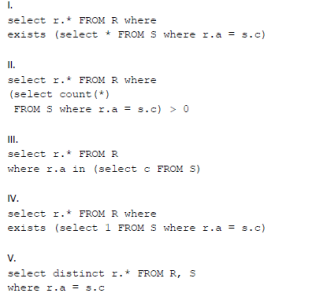
1642,5 14/05/2020

2314,23 15/05/2020

NULL NULL

NULL NULL

3 - Considere os comandos SQL a seguir.



Sabe-se que quatro desses comandos sempre produzem resultados com conteúdos idênticos, mesmo considerando-se diferentes instâncias de R e S.

O comando que NÃO faz parte desse grupo é:

1. I

2. II

3. III

4. IV

5. V

4 - Em uma instrução SQL, a cláusula GROUP BY, se utilizada juntamente com funções de grupo, deve estar em uma posição específica como

a) SELECT coluna1 [, coluna2, ... ] . função\_de\_grupo, [ função\_de\_grupo, ...] FROM tabela

ORDER BY coluna2

GROUP BY coluna4 [, coluna5, ...]

HAVING coluna5 > 10 WHERE condição;

b) SELECT coluna1 [, coluna2, ... ] . função\_de\_grupo, [ função\_de\_grupo, ...] FROM tabela

WHERE condição

ORDER BY coluna2

GROUP BY coluna4 [, coluna5, ...]

HAVING coluna5 > 10;

c) SELECT coluna1 [, coluna2, ... ] . função\_de\_grupo, [ função\_de\_grupo, ...] FROM tabela

WHERE condição

GROUP BY coluna4 [, coluna5, ...]

HAVING coluna5 > 10 ORDER BY coluna2;

d) SELECT coluna1 [, coluna2, ... ] . função\_de\_grupo, [ função\_de\_grupo, ...] FROM tabela

HAVING coluna5 > 10

ORDER BY coluna2

WHERE condição

GROUP BY coluna4 [, coluna5, ...] ;

e) SELECT coluna1 [, coluna2, ... ] . função\_de\_grupo, [ função\_de\_grupo, ...] FROM tabela

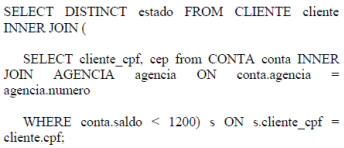
HAVING coluna5 > 10

WHERE condição

ORDER BY coluna2

GROUP BY coluna4 [, coluna5, ...];

5 - Visualize o select abaixo e responda a questão.



Considerando-se o modelo de dados das informações de referência, é correto afirmar que esse comando apresenta

a) CEP e CPF dos clientes que têm saldo menor que R$ 1.200,00.

b) os estados, sem repetir, onde moram clientes que têm saldo abaixo de R$ 1.200,00. c) os estados onde há agências em que os clientes têm saldo menor que R$ 1.200,00. d) os estados, com repetição, onde moram clientes que têm saldo positivo. e) os estados, sem repetir, onde moram clientes que têm saldo positivo.