**Questão 01:** Inserção manual de dados

Foi escolhidos dados que simulem compras do mundo do agro:

*-- Inserção de clientes*

*INSERT INTO clientes (nome, data\_cadastro) VALUES*

*('Carlos Silva', '2020-01-01'),*

*('Maria Souza', '2020-06-15'),*

*('Luciano Castro', '2020-11-23'),*

*('João Pereira', '2021-03-10'),*

*('Ana Oliveira', '2021-09-25'),*

*('Patricia Nunes', '2021-12-18'),*

*('Pedro Santos', '2022-02-05'),*

*('Mariana Costa', '2022-07-20'),*

*('Ricardo Almeida', '2023-01-11'),*

*('Fernanda Lopes', '2023-06-30'),*

*('Gustavo Ribeiro', '2023-10-07'),*

*('Gabriel Rocha', '2024-04-09'),*

*('Isabela Martins', '2024-08-15'),*

*('Roberto Lima', '2025-01-10'),*

*('Juliana Mendes', '2025-02-15');*

*-- Inserção de produtos*

*INSERT INTO produtos (nome, categoria, preco) VALUES*

*-- Sementes*

*('Soja', 'Sementes', 150.00),*

*('Milho', 'Sementes', 120.00),*

*('Sorgo', 'Sementes', 130.00),*

*-- Fertilizantes*

*('Nitrogênio', 'Fertilizante', 200.00),*

*('Potássio', 'Fertilizante', 180.00),*

*('Colheita', 'Fertilizante', 220.00),*

*-- Herbicidas*

*('Glifosato', 'Herbicida', 250.00),*

*('Paraquat', 'Herbicida', 270.00),*

*('Glufosinato de amônio', 'Herbicida', 260.00);*

*– inserindo dados tabela pedidos*

*INSERT INTO pedidos (data\_pedido, valor\_total, id\_cliente) VALUES*

*('2020-01-01', 350, 1),*

*('2020-03-02', 250, 1),*

*('2020-06-15', 270, 2),*

*('2020-11-12', 550, 1),*

*('2020-11-23', 260, 3),*

*('2020-12-01', 710, 2),*

*('2021-02-02', 780, 3),*

*('2021-03-10', 520, 4),*

*('2021-03-20', 800, 4),*

*('2021-03-20', 790, 2),*

*('2021-08-14', 2500, 3),*

*('2021-09-25', 2420, 5),*

*('2021-09-28', 2160, 5),*

*('2021-12-18', 7250, 6),*

*('2022-01-05', 2640, 2),*

*('2022-01-10', 5040, 3),*

*('2022-02-05', 8100, 7),*

*('2022-06-10', 8010, 4),*

*('2022-07-20', 11700, 8),*

*('2022-11-10', 3000, 8),*

*('2022-12-01', 3360, 1),*

*('2023-01-04', 6370, 5),*

*('2023-01-11', 9030, 9),*

*('2023-02-20', 1300, 6),*

*('2023-04-15', 3940, 3),*

*('2023-06-30', 16750, 10),*

*('2023-08-06', 7250, 4),*

*('2023-10-07', 3660, 11),*

*('2023-11-28', 8010, 4),*

*('2023-12-28', 6370, 5),*

*('2024-01-11', 5040, 3),*

*('2024-01-12', 9030, 9),*

*('2024-02-16', 16300, 10),*

*('2024-03-22', 1300, 6),*

*('2024-03-30', 11700, 8),*

*('2024-04-09', 12020, 12),*

*('2024-05-21', 5260, 7),*

*('2024-06-29', 11700, 8),*

*('2024-08-15', 134400, 13),*

*('2024-09-08', 3660, 11),*

*('2024-10-08', 8010, 4),*

*('2024-11-12', 550, 1),*

*('2024-12-10', 6370, 5),*

*('2024-12-03', 9030, 9),*

*('2025-01-04', 5040, 3),*

*('2025-01-08', 1300, 6),*

*('2025-01-10', 21890, 14),*

*('2025-01-23', 58250, 13),*

*('2025-02-15', 59700, 15),*

*('2025-03-02', 39000, 12);*

*– Inserção tabela itens\_pedido:  
INSERT INTO itens\_pedido (id\_pedido, id\_produto, quantidade, preco\_unitario) VALUES*

*(1, 1, 1, 150),*

*(1, 4, 1, 200),*

*(2, 7, 1, 250),*

*(3, 8, 1, 270),*

*(4, 2, 1, 120),*

*(4, 5, 1, 180),*

*(4, 7, 1, 250),*

*(5, 9, 1, 260),*

*(6, 2, 2, 120),*

*(6, 4, 1, 200),*

*(6, 8, 2, 270),*

*(7, 1, 2, 150),*

*(7, 6, 1, 220),*

*(7, 9, 1, 260),*

*(8, 3, 4, 130),*

*(9, 4, 4, 200),*

*(10, 3, 1, 130),*

*(10, 4, 2, 200),*

*(10, 9, 1, 260),*

*(11, 7, 10, 250),*

*(12, 9, 11, 260),*

*(13, 8, 8, 270),*

*(14, 1, 10, 150),*

*(14, 4, 10, 200),*

*(14, 7, 15, 250),*

*(15, 6, 12, 220),*

*(16, 2, 15, 120),*

*(16, 5, 18, 180),*

*(17, 3, 20, 130),*

*(17, 6, 25, 220),*

*(18, 1, 15, 150),*

*(18, 5, 10, 180),*

*(18, 5, 18, 200),*

*(19, 1, 20, 150),*

*(19, 2, 10, 120),*

*(19, 5, 18, 180),*

*(19, 7, 30, 250),*

*(20, 4, 4, 200),*

*(21, 2, 28, 120),*

*(22, 7, 13, 250),*

*(22, 9, 12, 260),*

*(23, 7, 17, 250),*

*(23, 8, 10, 270),*

*(23, 9, 8, 260),*

*(24, 2, 10, 120),*

*(25, 1, 15, 150),*

*(25, 2, 10, 120),*

*(26, 1, 25, 150),*

*(26, 2, 15, 120),*

*(26, 6, 30, 220),*

*(26, 4, 23, 200),*

*(27, 1, 10, 150),*

*(27, 4, 10, 200),*

*(27, 7, 10, 250),*

*(28, 3, 15, 130),*

*(28, 5, 5, 180),*

*(28, 8, 3, 270),*

*(29, 1, 15, 150),*

*(29, 5, 10, 180),*

*(29, 4, 18, 200),*

*(30, 7, 13, 250),*

*(30, 9, 12, 260),*

*(31, 2, 15, 120),*

*(31, 5, 18, 180),*

*(32, 7, 17, 250),*

*(32, 8, 10, 270),*

*(32, 9, 8, 260),*

*(33, 7, 30, 250),*

*(33, 6, 40, 220),*

*(34, 2, 10, 120),*

*(35, 1, 20, 150),*

*(35, 2, 10, 130),*

*(35, 5, 18, 180),*

*(35, 7, 30, 250),*

*(36, 1, 34, 150),*

*(36, 5, 9, 180),*

*(36, 4, 10, 200),*

*(36, 6, 15, 220),*

*(37, 3, 10, 130),*

*(37, 5, 7, 180),*

*(37, 8, 10, 270),*

*(38, 1, 20, 150),*

*(38, 2, 10, 120),*

*(38, 5, 18, 180),*

*(38, 7, 30, 250),*

*(39, 1, 100, 150),*

*(39, 2, 120, 120),*

*(39, 5, 250, 180),*

*(39, 4, 300, 200),*

*(40, 3, 15, 130),*

*(40, 5, 5, 180),*

*(40, 8, 3, 270),*

*(41, 1, 15, 150),*

*(41, 5, 10, 180),*

*(41, 4, 18, 200),*

*(42, 2, 1, 120),*

*(42, 5, 1, 180),*

*(42, 7, 1, 250),*

*(43, 7, 13, 250),*

*(43, 9, 12, 260),*

*(44, 7, 17, 250),*

*(44, 8, 10, 270),*

*(44, 9, 8, 260),*

*(45, 2, 15, 130),*

*(45, 5, 18, 180),*

*(46, 2, 10, 120),*

*(47, 7, 53, 250),*

*(47, 8, 32, 270),*

*(48, 7, 232, 250),*

*(49, 4, 123, 200),*

*(49, 8, 130, 270),*

*(50, 9, 150, 260);*

Os dados acima foram escolhidos para representar simular a vendas de uma revenda do agronegócio.

Obs: os csv estão no link do github

**Questão 02:** Análise RFM:

*WITH load\_pedidos AS (*

*SELECT*

*id\_cliente,*

*data\_pedido,*

*valor\_total,*

*ROW\_NUMBER() OVER (PARTITION BY id\_cliente ORDER BY data\_pedido DESC) AS classificao\_pedido,*

*COUNT(\*) OVER (PARTITION BY id\_cliente) AS total\_pedidos,*

*SUM(valor\_total) OVER (PARTITION BY id\_cliente) / COUNT(\*) OVER (PARTITION BY id\_cliente) AS ticket\_medio*

*FROM pedidos*

*),*

*pedidos\_cte AS (*

*SELECT*

*\**

*FROM load\_pedidos*

*WHERE classificao\_pedido = 1*

*)*

*SELECT*

*id\_cliente,*

*COALESCE(CURRENT\_DATE - data\_pedido, 0) AS dias\_desde\_ultimo\_pedido,*

*total\_pedidos,*

*ticket\_medio*

*FROM pedidos\_cte*

*ORDER BY id\_cliente;*

**Questão 03:** Permitir que um pedido tenha mais que um cliente

Para permitir que um pedido possa ter mais de um cliente é preciso criar uma tabela intermediária entre a tabela cliente e a tabela pedidos e para isso ocorrer devemos remover a chave estrangeira da tabela pedidos da coluna id\_cliente e criar uma nova tabela pedido\_clientes que irá possuir as seguintes colunas: id\_pedido e id\_cliente:

Código SQL:

*ALTER TABLE pedidos DROP CONSTRAINT pedidos\_id\_cliente\_fkey;*

*ALTER TABLE pedidos DROP COLUMN id\_cliente;*

CREATE TABLE pedido\_clientes (

id\_pedido INT REFERENCES pedidos(id\_pedido),

id\_cliente INT REFERENCES clientes(id\_cliente),

PRIMARY KEY (id\_pedido, id\_cliente)

);

**Questão 04:** Top 05 produtos mais rentáveis  
  
*WITH vendas\_por\_produto AS (*

*SELECT*

*p.id\_produto,*

*p.nome,*

*SUM(ip.quantidade \* ip.preco\_unitario) AS total\_vendas*

*FROM itens\_pedido ip*

*LEFT JOIN pedidos pe ON ip.id\_pedido = pe.id\_pedido*

*LEFT JOIN produtos p ON ip.id\_produto = p.id\_produto*

*WHERE pe.data\_pedido >= (CURRENT\_DATE - INTERVAL '1 year')*

*GROUP BY p.id\_produto, p.nome*

*)*

*SELECT*

*id\_produto,*

*nome,*

*total\_vendas*

*FROM vendas\_por\_produto*

*ORDER BY total\_vendas DESC*

*LIMIT 5;*

**Questão 05:***WITH vendas\_mensais AS (*

*SELECT*

*TO\_CHAR(data\_pedido, 'YYYY-MM') AS mes\_ano,*

*SUM(valor\_total) AS total\_vendas*

*FROM pedidos*

*GROUP BY TO\_CHAR(data\_pedido, 'YYYY-MM')*

*)*

*SELECT*

*mes\_ano,*

*total\_vendas,*

*ROUND(*

*(total\_vendas - LAG(total\_vendas) OVER (ORDER BY mes\_ano)) /*

*NULLIF(LAG(total\_vendas) OVER (ORDER BY mes\_ano), 0) \* 100, 2*

*) AS crescimento\_percentual*

*FROM vendas\_mensais*

*ORDER BY mes\_ano;*

**Questão 06:**

*WITH ultima\_compra AS (*

*SELECT*

*id\_cliente,*

*MAX(data\_pedido) AS ultima\_data\_pedido*

*FROM pedidos*

*GROUP BY id\_cliente*

*)*

*SELECT*

*c.id\_cliente,*

*c.nome,*

*u.ultima\_data\_pedido*

*FROM clientes c*

*LEFT JOIN ultima\_compra u ON c.id\_cliente = u.id\_cliente*

*WHERE u.ultima\_data\_pedido IS NULL*

*OR u.ultima\_data\_pedido < CURRENT\_DATE - INTERVAL '6 months'*

*ORDER BY u.ultima\_data\_pedido;*

**Questão 07:**

*WITH valor\_calculado\_cte AS (*

*SELECT*

*p.id\_pedido,*

*p.valor\_total AS valor\_total\_registrado,*

*SUM(ip.quantidade \* ip.preco\_unitario) AS valor\_calculado*

*FROM pedidos p*

*JOIN itens\_pedido ip ON p.id\_pedido = ip.id\_pedido*

*GROUP BY p.id\_pedido, p.valor\_total*

*)*

*SELECT*

*id\_pedido,*

*valor\_total\_registrado,*

*valor\_calculado*

*FROM valor\_calculado\_cte*

*WHERE valor\_total\_registrado <> valor\_calculado;*