

1. --Adicionar um novo cliente:

```
CREATE TABLE IF NOT EXISTS CLIENTES(  
  id SERIAL NOT NULL,  
  nome VARCHAR(120) NOT NULL,  
  email VARCHAR(120) UNIQUE NOT NULL,  
  telefone VARCHAR(11) NOT NULL,  
  idade INT NOT NULL,  
  CONSTRAINT CLIENTE_PK PRIMARY KEY(ID)  
);
```

INSERT INTO clientes (nome, email, telefone, idade)

VALUES ('Romulo Nogueira de Souza', 'nogueirainclusivedev@desenvolvedor.com', '9999-9999', 39);

2. --Adicionar um novo produto:

```
CREATE TABLE PRODUTOS (  
  id SERIAL PRIMARY KEY,  
  nome VARCHAR(255) NOT NULL,  
  preco NUMERIC(10, 2) NOT NULL,  
  quantidade INT NOT NULL  
);
```

INSERT INTO PRODUTOS (nome, preco, quantidade)

VALUES ('Vassoura de Curling', 100.00, 100);

3. --Adicionar um novo vendas:

```
CREATE TABLE VENDAS (  
    id SERIAL PRIMARY KEY,  
    produto_id INT NOT NULL,  
    cliente_id INT NOT NULL,  
    quantidade INT NOT NULL,  
    FOREIGN KEY (produto_id) REFERENCES produtos(id),  
    FOREIGN KEY (cliente_id) REFERENCES clientes(id)  
);
```

```
INSERT INTO VENDAS (produto_id, cliente_id, quantidade)  
VALUES (1, 1, 10);
```

4. --Atualizar o telefone de um cliente:

```
UPDATE CLIENTES SET telefone = '89123-7888' WHERE email = 'brabo1@clienteloucao.com';
```

5. --Atualizar o preço e a quantidade de um produto:

```
UPDATE PRODUTOS SET preco = 15.00, quantidade = 50 WHERE nome = 'Flor de Singapura';
```

6. -- Atualizar a quantidade de um item de venda:

```
UPDATE VENDAS SET quantidade = 5 WHERE id = 1;
```

7. -- Deletar um cliente específico:

```
DELETE FROM clientes WHERE email = 'romulonogueira929@sankhyasoulcode.com';
```

8. -- Deletar um produto específico:

```
DELETE FROM produtos WHERE nome = 'Vassoura de Curling ';
```

9. -- Deletar um item de venda específico:

```
DELETE FROM vendas WHERE id = 1;
```

10. -- Criar uma view que exibe todos os clientes com suas respectivas idades e telefones:

```
CREATE VIEW view_clientes AS SELECT nome, idade, telefone FROM clientes;
```

```
CREATE VIEW view_clientes AS
```

```
SELECT nome, idade, telefone
```

```
FROM clientes;
```

11. -- Criar uma view que exibe todos os produtos com seus preços e quantidades em estoque:

```
CREATE VIEW view_produtos AS SELECT nome, preco, quantidade FROM produtos;
```

```
CREATE VIEW view_produtos AS
```

```
SELECT nome, preco, quantidade
```

```
FROM produtos;
```

12. -- Criar uma view que exibe o total vendido por cada cliente:

```
CREATE VIEW view_total_vendas_por_cliente AS
```

```
SELECT c.nome, SUM(v.quantidade * p.preco) AS total_vendido
```

```
FROM vendas v
```

```
JOIN clientes c ON v.cliente_id = c.id
```

```
JOIN produtos p ON v.produto_id = p.id
```

```
GROUP BY c.nome;
```

13. -- Adicionar um novo cliente e uma nova compra para esse cliente:

```
INSERT INTO clientes (nome, email, telefone, idade)
```

```
VALUES ('Romulo Thiago Pedro', 'trabalhobd02@cliente.com', '97777-7777', 22);
```

```
INSERT INTO vendas (produto_id, cliente_id, quantidade)
```

```
VALUES (1, (SELECT id FROM clientes WHERE email='trabalhobd02@cliente.com'), 3);
```

14. -- Atualizar o nome e o email de um cliente específico:

```
UPDATE clientes
```

```
SET nome = 'O BRABO', email = 'TITIT@TIURANGO'
```

```
WHERE id = 1;
```

15. -- Deletar todos os itens de venda de um produto específico:

```
DELETE FROM vendas WHERE produto_id = (SELECT id FROM produtos WHERE  
nome='Capacete do Power Ranger Verde');
```

16. -- Adicionar múltiplos clientes:

```
INSERT INTO clientes (nome, email, telefone, idade)
```

```
VALUES
```

```
('Romulo Nogueira', 'romulobd@cliente.com', '91111-1111', 40),
```

```
('Thiago Ariça', 'thiagofk@cliente.com', '92222-2222', 33),
```

```
('Pedro Pantanoso', 'pedropk@cliente.com', '93333-3333', 20);
```

17. -- Adicionar múltiplos produtos:

```
INSERT INTO produtos (nome, preco, quantidade)
```

```
VALUES
```

```
('Armadura de ouro de Sagitário', 65040.00, 200),
```

```
('Kaneda's bike', 3530.00, 300);
```

18. -- Adicionar múltiplos itens de venda:

Adicione três novos itens de venda relacionando diferentes produtos e clientes existentes em uma única instrução `INSERT`.

```
INSERT INTO vendas (cliente_id, produto_id, quantidade)
```

```
VALUES
```

```
(1, 101, 2, '2024-06-18'),
```

(2, 102, 1, '2024-06-18'),

(3, 103, 5, '2024-06-18');

19. --Atualizar a idade de todos os clientes acima de 30 anos:

UPDATE CLIENTES

SET idade = idade + 1

WHERE idade > 30;

20. --Atualizar o preço de todos os produtos com quantidade menor que 10:

UPDATE produtos

SET preco = preco * 1.10

WHERE quantidade < 10;

21. --Atualizar o nome de todos os produtos para incluir uma palavra-chave:

UPDATE produtos

SET nome = 'Promoção ' || nome;

22. --Deletar todos os clientes com idade menor que 18 anos

DELETE FROM clientes

WHERE idade < 18;

23. --Deletar todos os produtos com preço maior que 50:

DELETE FROM PRODUTOS

WHERE preco > 50;

24. --Deletar todos os itens de venda de clientes específicos (substitua 'email@exemplo.com' pelo email específico):

```
DELETE FROM itens_venda  
WHERE cliente_id IN (  
SELECT id FROM clientes WHERE email = 'grupodeamigos@exemplo.com'  
);
```

25. --Criar uma view que exibe o estoque total de produtos vendidos:

```
CREATE VIEW estoque_total_vendido AS  
SELECT p.nome, SUM(iv.quantidade) AS total_vendido  
FROM produtos p  
JOIN itens_venda iv ON p.id = iv.produto_id  
GROUP BY p.nome;
```

26 Criar uma view que exibe os clientes que não fizeram compras**:

- Crie uma view que mostra o nome e email dos clientes que não têm itens de venda associados.

R

```
CREATE VIEW ClientesSemCompras AS  
SELECT nome, email  
FROM Clientes  
WHERE id_cliente NOT IN (SELECT DISTINCT id_cliente FROM Vendas);
```

27. --Criar uma view que exibe os produtos não vendidos:

```
CREATE VIEW produtos_ao_vendidos AS  
SELECT nome, preco FROM produtos
```

```
WHERE id NOT IN (SELECT DISTINCT produto_id FROM itens_venda);
```

27. --Criar uma view que exibe os produtos não vendidos:

```
CREATE VIEW produtos_ao_vendidos AS  
SELECT nome, preco FROM produtos  
WHERE id NOT IN (SELECT DISTINCT produto_id FROM itens_venda);
```

28. --Adicionar uma nova coluna em uma tabela e atualizar dados:

```
ALTER TABLE clientes ADD COLUMN data_de_cadastro DATE;  
UPDATE clientes SET data_de_cadastro = CURRENT_DATE;
```

29. --Atualizar o preço dos produtos com base em uma condição:

```
UPDATE produtos SET preco = preco * 1.05 WHERE quantidade < 20;
```

30. --Deletar registros duplicados em uma tabela (mantendo o registro com o menor id):

```
DELETE FROM CLIENTES  
WHERE id NOT IN (  
SELECT MIN(id) FROM clientes GROUP BY email  
);
```

31. --Criar uma view que exibe a média de vendas por produto:

```
CREATE VIEW media_vendas_por_produto AS  
SELECT p.nome, AVG(iv.quantidade) AS media_vendas  
FROM produtos p  
JOIN itens_venda iv ON p.id = iv.produto_id  
GROUP BY p.nome;
```

31. --Criar uma view que exibe a média de vendas por produto:

```
CREATE VIEW media_vendas_por_produto AS  
SELECT p.nome, AVG(iv.quantidade) AS media_vendas  
FROM produtos p
```

```
JOIN itens_venda iv ON p.id = iv.produto_id
```

```
GROUP BY p.nome;
```

32. --Adicionar um novo cliente e realizar múltiplas compras para esse cliente (substitua 'dados_ficticios' pelos dados reais do cliente):

```
INSERT INTO clientes (nome, email, dados_ficticios) VALUES ('Nome Cliente',  
'questao32db02@cliente.com', 'outros_dados');
```

```
INSERT INTO itens_venda (produto_id, cliente_id, quantidade) VALUES (1, 10, 5);
```

```
INSERT INTO itens_venda (produto_id, cliente_id, quantidade) VALUES (2, 10, 3);
```

```
INSERT INTO itens_venda (produto_id, cliente_id, quantidade) VALUES (3, 10, 2);
```

33. --Atualizar o email de todos os clientes para um novo domínio:

```
UPDATE clientes SET email = REPLACE(email,  
'romulonogueira@yagloghotmail.com','thiagoarica@uolbolibestig.com');
```

34. -- Deletar produtos que não foram vendidos nos últimos 6 meses:

```
DELETE FROM produtos WHERE id NOT IN (  
    SELECT DISTINCT produto_id FROM VENDAS  
    WHERE vendas >= CURRENT_DATE - INTERVAL '6 months'  
);
```

35. --Criar uma view que exibe a proporção de vendas por cliente:

```
CREATE OR REPLACE VIEW proporcao_vendas_cliente AS SELECT c.nome_cliente,  
TO_CHAR((SUM(i.quantidade * p.preco)::decimal * 100 / ( SELECT SUM(i2.quantidade *  
p2.preco) FROM itens i2 JOIN produtos p2 ON i2.codigo_produto = p2.codigo )),  
'99.99%') AS proporcao_vendas FROM clientes c JOIN itens i ON c.id = i.id_cliente JOIN  
produt
```