

## Exercício Floricultura

Uma floricultura deseja informatizar suas operações.

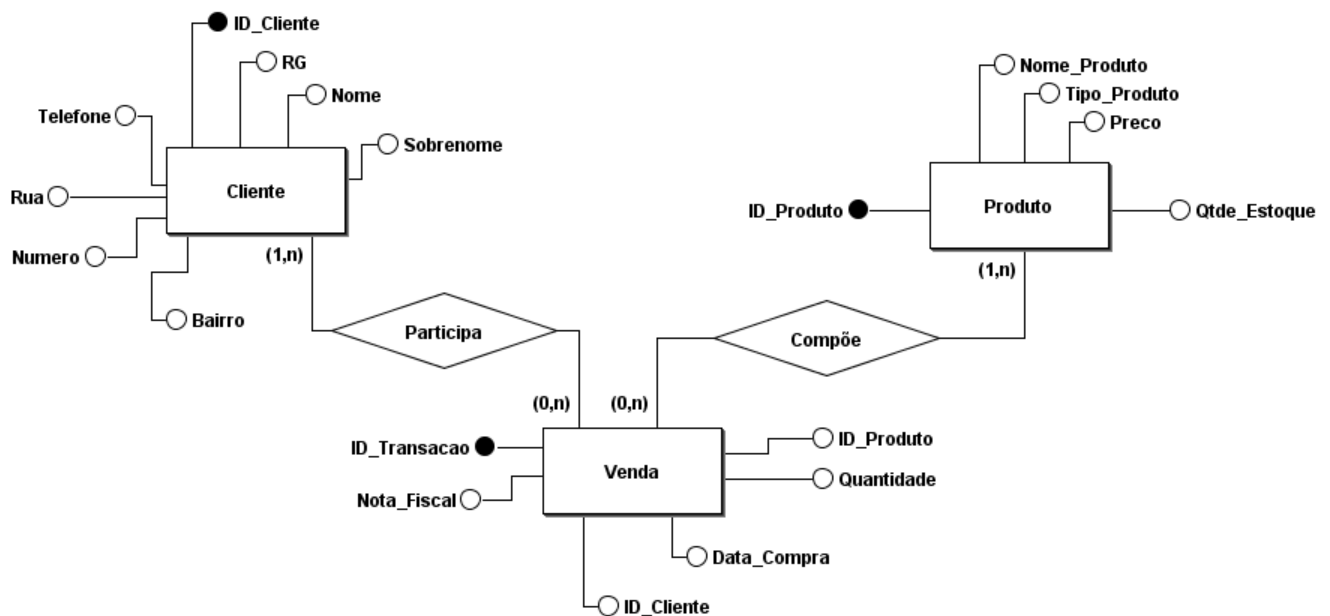
Inicialmente, deseja manter um cadastro de todos os seus clientes, mantendo informações como: RG, nome, sobrenome, telefone e endereço completo.

Deseja também manter um cadastro contendo informações sobre os produtos que vende, tais como: nome do produto, tipo (flor, vaso, planta,...), preço e quantidade em estoque.

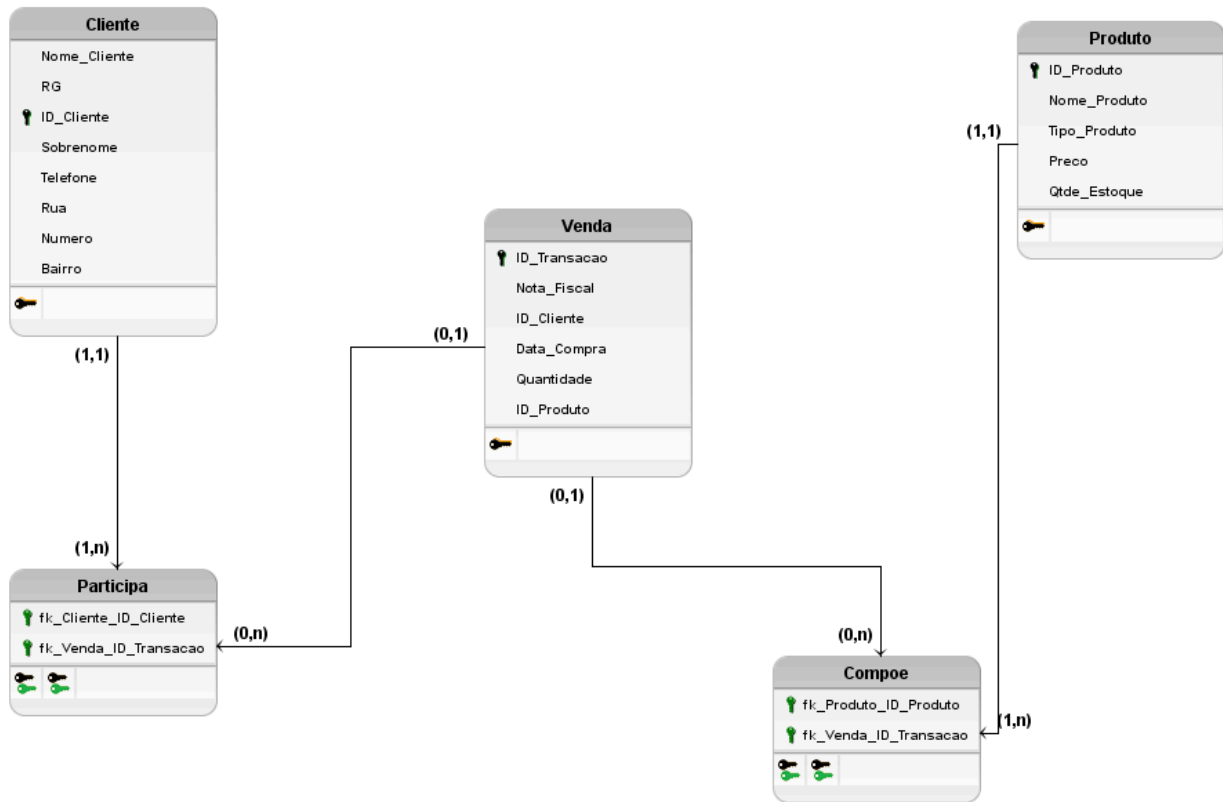
Quando um cliente faz uma compra, ela é armazenada, mantendo informação sobre o cliente que fez a compra, a data da compra, número da nota fiscal, um código de identificação da transação, quantidade de cada produto e os produtos comprados.

### DER

#### Modelo Físico



## Modelo Lógico



## Atividades - Parte I

1. Realizar a modelagem de dados para o exercício (MER, DER, DD, Normalização, etc.).
2. Criar o banco de dados usando Python + SQLite.
3. Criar as tabelas necessárias, com seus relacionamentos.
4. Inserir dados de testes nas tabelas - 10 registros em cada uma.
5. Teste inicial: verificar se os registros foram inseridos corretamente usando uma declaração `SELECT * FROM` para cada tabela.

CÓDIGO:

```
import sqlite3 as con
```

```
# Criar tabelas
```

```
sql_clientes = ""
```

```
CREATE TABLE IF NOT EXISTS Cliente (  
    ID_Cliente INTEGER PRIMARY KEY AUTOINCREMENT,  
    RG VARCHAR (12) NOT NULL,  
    Nome_Cliente VARCHAR(30) NOT NULL,
```

```
Sobrenome_Cliente VARCHAR(40),
Telefone VARCHAR(12),
Rua VARCHAR(40),
Numero VARCHAR(5),
Bairro VARCHAR(25)
);
'''
```

```
sql_produtos = '''
CREATE TABLE IF NOT EXISTS Produto (
ID_Produto INTEGER PRIMARY KEY AUTOINCREMENT,
Nome_Produto VARCHAR (30) NOT NULL,
Tipo_Produto VARCHAR (25) NOT NULL,
Preco DECIMAL(10,2) NOT NULL,
Qtde_Estoque SMALLINT NOT NULL
);
'''
```

```
sql_vendas = '''
CREATE TABLE IF NOT EXISTS Venda (
ID_Transacao INTEGER PRIMARY KEY AUTOINCREMENT,
Nota_Fiscal SMALLINT NOT NULL,
ID_Cliente INTEGER NOT NULL,
Data_Compra DATETIME,
ID_Produto INTEGER NOT NULL,
Quantidade SMALLINT NOT NULL,
FOREIGN KEY (ID_Cliente) REFERENCES Clientes(ID_Cliente),
FOREIGN KEY (ID_Produto) REFERENCES Produtos(ID_Produto)
);
'''
```

```
try:
    conexao = con.connect('floricultura.db')
    cursor = conexao.cursor()

    cursor.execute(sql_clientes)
    cursor.execute(sql_produtos)
    cursor.execute(sql_vendas)

    conexao.commit()
except con.DatabaseError as erro:
    print("Erro no banco de dados", erro)
finally:
```

```
if conexao:
    conexao.close()
```

---

```
# Verificar se tabelas foram criadas
import sqlite3 as con
```

```
try:
    conexao = con.connect('floricultura.db')
    cursor = conexao.cursor()

    # Verificar se tabelas foram criadas
    res = cursor.execute("SELECT name FROM sqlite_master")
    print(res.fetchall())
except con.DatabaseError as erro:
    print("Erro no banco de dados", erro)
finally:
    if conexao:
        conexao.close()
```

---

```
# Inserir dados na tabela Clientes
import sqlite3 as con
```

```
insere_cliente = """
    INSERT INTO Cliente (RG, Nome_Cliente, Sobrenome_Cliente, Telefone, Rua, Numero,
    Bairro)
    VALUES
    ('265356325', 'Fábio', 'dos Reis', '1156326356', 'Rua do Orfanato', '235', 'Vila Prudente'),
    ('268653215', 'João', 'Cavutto', '1178563214', 'Rua do Oratório', '1957', 'Mooca'),
    ('289632457', 'Julia', 'Tamashiro', '1196323654', 'Rua Lorenzo da Viterbo', '365', 'Vila
    Moraes');
    """
```

```
try:
    conexao = con.connect('floricultura.db')
    cursor = conexao.cursor()

    cursor.execute(insere_cliente)

    conexao.commit()
```

```

except con.DatabaseError as erro:
    print("Erro no banco de dados", erro)
else:
    # Verificar se dados foram cadastrados
    res = cursor.execute("SELECT * FROM Cliente;")
    print(res.fetchall())
finally:
    if conexao:
        conexao.close()

```

# Inserir dados na tabela Produtos

import sqlite3 as con

```

insere_produto = """
INSERT INTO Produto (Nome_Produto, Tipo_Produto, Preço, Qtde_Estoque)
VALUES
('Orquídea', 'Flor', 55.50, 25),
('Azaléa', 'Flor', 35.63, 15),
('Terra Vegetal', 'Insumo', 72.50, 20),
('Jardineira', 'Vaso', 15.50, 18);
"""

```

```

try:
    conexao = con.connect('floricultura.db')
    cursor = conexao.cursor()

    cursor.execute(insere_produto)
    conexao.commit()
except con.DatabaseError as erro:
    print("Erro no banco de dados", erro)
else:
    # Verificar se dados foram cadastrados
    res = cursor.execute("SELECT * FROM Produto;")
    print(res.fetchall())
finally:
    if conexao:
        conexao.close()

```

# Inserir dados na tabela Vendas

```

import sqlite3 as con

insere_venda = """
INSERT INTO Venda (Nota_Fiscal, ID_Cliente, Data_Compra, ID_Produto, Quantidade)
VALUES
(123, 1, '2024-04-04', 1, 3),
(123, 1, '2024-04-04', 2, 2),
(124, 3, '2024-03-28', 2, 5),
(124, 3, '2024-04-03', 3, 2),
(124, 3, '2024-03-30', 4, 3);
"""

try:
    conexao = con.connect('floricultura.db')
    cursor = conexao.cursor()

    cursor.execute(insere_venda)
    conexao.commit()
except con.DatabaseError as erro:
    print("Erro no banco de dados", erro)
else:
    # Verificar se dados foram cadastrados
    res = cursor.execute("SELECT * FROM Venda;")
    print(res.fetchall())
finally:
    if conexao:
        conexao.close()

```

## TESTES DE SOFTWARE

Executar as consultas a seguir no banco de dados Floricultura, usando scripts em Python com sqlite3:

# 0. Exibir os nomes dos clientes cadastrados.

```

import sqlite3 as con

conexao = con.connect('floricultura.db')
cursor = conexao.cursor()

conexao = con.connect('floricultura.db')
cursor = conexao.cursor()

```

```

consulta = ""
SELECT Nome_Cliente
FROM Cliente;
""

res = cursor.execute(consulta)
print("Cliente:\n")
for linha in res.fetchall():
    print(f'{linha[0]}')

conexao.close()

```

# 1. Exibir os nomes dos clientes e todos os produtos comprados por eles.  
import sqlite3 as con

```

conexao = con.connect('floricultura.db')
cursor = conexao.cursor()

consulta = ""
SELECT C.Nome_Cliente, P.Nome_Produto
FROM VENDA V
JOIN Cliente C
ON C.ID_Cliente = V.ID_Cliente
JOIN Produto P
ON P.ID_Produto = V.ID_Produto;
""

res = cursor.execute(consulta)
print("Cliente\tProduto")
for linha in res.fetchall():
    print(f'{linha[0]}\t{linha[1]}')

conexao.close()

```

# 2. Exibir o nome e todos os produtos comprados pelo cliente de ID 01.

```

import sqlite3 as con

conexao = con.connect('floricultura.db')
cursor = conexao.cursor()

consulta = ""
SELECT C.Nome_Cliente, P.Nome_Produto

```

```

FROM VENDA V
JOIN Cliente AS C
ON C.ID_Cliente = V.ID_Cliente
JOIN Produto P
ON P.ID_Produto = V.ID_Produto
WHERE C.ID_Cliente = 1;
'''

res = cursor.execute(consulta)
cliente = None
produtos = []
for linha in res.fetchall():
    cliente = linha[0]
    produtos.append(linha[1])

print(f'Cliente: {cliente}')
print("Produtos:")
for i, produto in enumerate(produtos, 1):
    print(f'{i}. {produto}')

conexao.close()

```

# 3. Exibir o valor total das compras realizadas pelo cliente de nome Julia..

```

import sqlite3 as con

conexao = con.connect('floricultura.db')
cursor = conexao.cursor()

consulta = '''
SELECT C.Nome_Cliente, SUM(P.Preco)
FROM VENDA V
INNER JOIN Cliente C
ON C.ID_Cliente = V.ID_Cliente
INNER JOIN Produto P
ON P.ID_Produto = V.ID_Produto
WHERE C.Nome_Cliente = 'Julia'
GROUP BY C.Nome_Cliente;
'''

res = cursor.execute(consulta)
print("Cliente\tValor Total")
linha = res.fetchone()
if linha:

```



```
print(f'{linha[0]}\tR$ {linha[1]:.2f}')
```

```
conexao.close()
```

# 4. Exibir o valor total de todas as vendas realizadas pela floricultura

```
import sqlite3 as con
```

```
conexao = con.connect('floricultura.db')
```

```
cursor = conexao.cursor()
```

```
consulta = ""
```

```
SELECT SUM(P.Preco * V.Quantidade)
```

```
FROM Venda V
```

```
JOIN Produto P
```

```
ON P.ID_Produto = V.ID_Produto
```

```
""
```

```
res = cursor.execute(consulta)
```

```
print("-----Total de Vendas da Floricultura-----")
```

```
linha = res.fetchone()
```

```
if linha:
```

```
    print(f'\t\tR$ {linha[0]:.2f}')
```

```
conexao.close()
```

# 5. Mostrar o valor total das vendas de vasos da floricultura.

```
import sqlite3 as con
```

```
conexao = con.connect('floricultura.db')
```

```
cursor = conexao.cursor()
```

```
consulta = ""
```

```
SELECT SUM(P.Preco * V.Quantidade)
```

```
FROM Venda V
```

```
JOIN Produto P ON P.ID_Produto = V.ID_Produto
```

```
WHERE P.Tipo_Produto = 'Vaso';
```

```
""
```

```
res = cursor.execute(consulta)
```

```
linha = res.fetchone()
```

```
if linha:
```

```

    valor_total = linha[0]
    print(f'Valor total das vendas de vasos: R$ {valor_total:.2f}')
else:
    print('Não foram vendidos vasos.')

```

```

conexao.close()

```

# 6. Mostrar o valor total de venda de cada produto da loja.

```

import sqlite3 as con

```

```

conexao = con.connect('floricultura.db')
cursor = conexao.cursor()

```

```

# Retornar o valor total de venda de cada produto da loja
consulta = """
SELECT P.Nome_Produto, SUM(P.Preco * V.Quantidade)
FROM Venda V
JOIN Produto P ON P.ID_Produto = V.ID_Produto
GROUP BY P.Nome_Produto;
"""

```

```

res = cursor.execute(consulta)
print("Produto\t\t\tValor Total")
for linha in res.fetchall():
    print(f'{linha[0]:<20}\tR$ {linha[1]:.2f}')

```

```

conexao.close()

```

Obs.: :<20 é uma especificação de formato para alinhar o texto à esquerda em um espaço de 20 caracteres. Cada parte tem o seguinte significado:

:: Indica que é uma especificação de formatação.

<: Indica alinhamento à esquerda.

20: Indica a largura total do campo, neste caso, 20 caracteres.

Assim, :<20 garante que o texto será alinhado à esquerda em um espaço de 20 caracteres.