Projeto de Bancos de Dados:

Consumo Cliente – Sistema de Varejo para Eletrônicos.

Arthur Santos Bascoy, Luís Fernando Cardoso Silva Sousa, Luine De Paula Freitas, Marilda Batista de Souza Marcelino.

Orientadora Luciana Mara Freitas Diniz

PUC Minas
Curso de Tecnologia em Banco de Dados
abascoy@sga.pucminas.br; lfcssousa@sga.pulmonar.br; luine.freitas@sga.pucminas.br, marilda.marcelino@sga.pucminas.br

lmfdiniz@sga.pucminas.br.

Resumo. O presente relatório apresenta o projeto do banco de dados SVE (Sistema de Varejo para Eletrônicos). Especificamente, o relatório apresenta o resultado das atividades de especificação do minimundo, análise de requisitos, projeto conceitual, projeto lógico e projeto físico do SVE. As informações necessárias para a realização das atividades de modelagem foram coletadas a partir de especificações textuais. O banco de dados SVE foi concebido dentro do paradigma relacional utilizando como base o modelo relacional, sendo constituído por um conjunto de entidades. Fisicamente, o banco de dados é composto por seis arquivos indexados com índice em cada um dos campos correspondentes às chaves primárias das tabelas de origem.

1 Introdução

Empresas de varejo eletrônico precisam manter seu banco de dados atualizado a fim de manter os dados e dia e organizados para visualização de indicadores. O objetivo do atual trabalho é criar um banco de dados para consumo de lojas de varejo de eletrônicos, empresa de médio porte, com aproximadamente 499 funcionários, onde serão coletados dados referentes às compras. Propõe-se uma especificação de minimundo, análise de requisitos, projeto conceitual, projeto lógico e projeto físico do banco de dados SVE (Sistema de Varejo para Eletrônicos), que em sua versão 1.0 será utilizado como base o modelo relacional e podendo ser implementado em sistemas gerenciadores de banco de dados (SGBD) relacionais comerciais.

2 Especificação do Minimundo

Essa seção apresenta a descrição textual de minimundo do SVE (v1.0), um banco de dados para um sistema loja de varejo de eletrônicos que gerencia produtos e clientes. Os produtos eletrônicos comercializados são identificados por códigos, custos, cor, categoria especificando o tipo de item e os clientes identificados por seu CPF, email, data de nascimento e código. Cada cliente pode comprar diversos produtos, sendo que cada produto pode ser comprado por diversos clientes, mas devemos conhecer as informações de cada compra feita.

2.1 Requisitos Funcionais

Os variados grupos de clientes demandarão diferentes operações de manipulação de dados sobre diferentes porções do banco de dados. O grupo Gerência demandará atualização e recuperação de dados sobre praticamente todas os elementos do banco de dados, uma vez que esse grupo será o responsável por manter os dados atualizados, dando suporte aos outros grupos. O grupo vendedor demandará consultas de recuperação de dados. O grupo cliente demandará consultas para atualização de seus dados cadastrais e para manipulação de dados sobre suas compras. O grupo geral demandará recuperação de dados sobre as compras, vendas, custos, cor, categoria e tipo de produto eletrônico. A Tabela 1 apresenta as principais consultas que cada grupo de usuários demandará ao sistema de banco de dados, bem como a frequência esperada de submissão (A para alta, M para média e B para baixa.

Tabela 1: Frequência esperada de consultas por grupo de usuário

Consulta		Grupo	Frequência
Q001	Listar todas as lojas cadastradas no sistema	Gerência	M
Q002	Encontrar a loja com o maior número de vendas	Geral	B
Q003	Listar todos os vendedores cadastrados no sistema	Gerência	B
Q004	Encontrar o vendedor que mais vendeu no último mês	Gerência	M
Q005	Listar todas as vendas realizadas no sistema	Vendedor	A
Q006	Encontrar as vendas realizadas entre duas datas específicas	Vendedor	A
Q007	Inserir um novo produto na tabela de produtos	Gerência	A
Q008	Listar todos os fornecedores cadastrados no sistema	Gerência	M
Q009	Inserir um novo fornecedor na tabela de fornecedores	Gerência	A
Q010	Encontrar a receita total de uma loja	Gerência	M
Q011	Listar vendas por vendedor (qual vendedor realizou a venda)	Vendedor	A
Q012	Inserir um novo cliente no banco de dados	Geral	B
Q013	Listar clientes por vendedor	Vendedor	B
Q014	Encontra os produtos que nunca foram vendidos	Vendedor	M
Q015	Listar todos os clientes cadastrados no sistema	Vendedor	M
Q016	Contar quantas compra foram efetuadas por cliente	Vendedor	M
Q017	Visualizar compras realizadas por cliente	Cliente	M
Q018	Atualizar dados cadastrais do cliente	Cliente	M

3. Projeto Conceitual

Aqui, através da Figura 1, apresentamos o projeto conceitual do banco de dados do SVE, através do diagrama de entidade-relacionamento (DER).

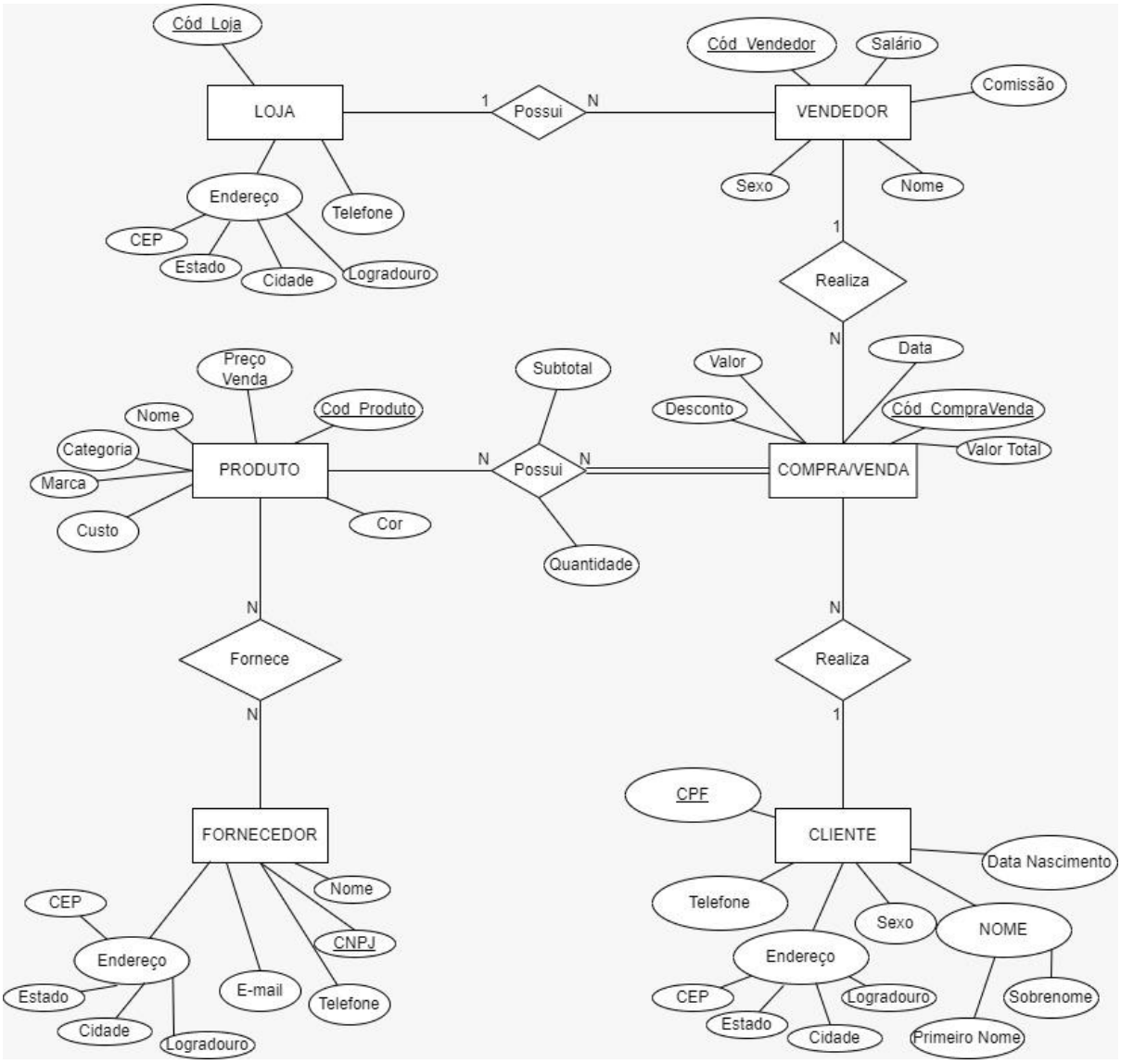

Figura 1: Diagrama Entidade-Relacionamento do SVE.

Temos também, a tabela 2, com o detalhamento das entidades, relacionamentos e atributos, que foram apresentados no diagrama da Figura 1, de entidade de relacionamento da SVE.

Tabela 2: Elementos do modelo conceitual do SVE

Tipo	Subtipo	ID	Rótulo	Referência	Descrição
Entidade	Forte	E001	PRODUTO		
Entidade	Forte	E002	CLIENTE		
Entidade	Forte	E003	LOJA		
Entidade	Forte	E004	VENDEDOR		

Entidade	Forte	E005	COMPRA/VENDA	
Entidade	Forte	E006	FORNECEDOR	
Atributo	Chave	A001	Cód_Produto	E001
Atributo	Chave	A002	CPF	E002
Atributo	Chave	A003	Cód_Loja	E003
Atributo	Chave	A004	Cód_Vendedor	E004
Atributo	Chave	A005	Cód_Venda	E005
Atributo	Simples	A006	Nome	E001, E006
Atributo	Simples	A007	Categoria	E001
Atributo	Simples	A008	Marca	E001
Atributo	Simples	A009	Custo	E001
Atributo	Simples	A010	Preço Venda	E001
Atributo	Simples	A011	Cor	E001
Atributo	Composto	A012	Fornecedor	E001, E003
Atributo	Simples	A013	Telefone	E002. E003, E006
Atributo	Composto	A014	Endereço	E002, E006
Atributo	Simples	A015	Sexo	E002, E004
Atributo	Composto	A016	Nome	E002, A024, A012
Atributo	Simples	A017	Data Nascimento	E002
Atributo	Simples	A018	CPF Cliente	E005
Atributo	Simples	A019	Cod Vendedor	E005
Atributo	Simples	A020	Cod Loja	E005
Atributo	Simples	A021	Cod Produto	E005
Atributo	Simples	A022	Desconto	E005
Atributo	Simples	A023	Valor	E005
Atributo	Composto	A024	Funcionário	E003
Atributo	Simples	A025	Código	A024
Atributo	Simples	A026	CNPJ	E006
Atributo	Simples	A027	Primeiro Nome	A006
Atributo	Simples	A028	Sobrenome	A006
Atributo	Simples	A029	Salário	E004
Atributo	Simples	A030	Comissão	E004
Atributo	Simples	A031	Loja	E004
Relacionamento	Forte	R001	Possui	E003, E004
Relacionamento	Forte	R002	Realiza	E004, E005
Relacionamento	Forte	R003	Compra	E001, E002

Relacionamento	Forte	R004	Possui	E001, E005
Relacionamento	Forte	R005	Possui	E001, E006

4. Projeto Lógico

Aqui é apresentado o projeto lógico do banco de dados do SVE, com as descrições das principais estruturas e restrições lógicas, baseadas no modelo de implementação relacional. Temos a figura 2 demonstrando o diagrama do modelo de implementação relacional do SVE, mapeados a partir do modelo conceitual descrito na seção do projeto conceitual.

Figura 2: Diagrama do modelo de implementação relacional do SVE.

Na figura 2, podemos observar o mapeamento de seis relações, com relações que partem de 6 atributos por relação e vão até 9 atributos por relação. Esse diagrama é uma maneira fácil e simples de visualizarmos as relações, de forma rápida e compacta, observando os atributos chave de cada relação, assim como as chaves estrangeiras ligadas às mesmas.

Além da representação feita pela figura 2, é importante ressaltar no projeto lógico do banco de dados a escolha, abordagem e solução a serem adotadas para a

implementação do SVE. Será adotada a abordagem baseada em SGBD relacional e a solução comercial em MySQL.

A figura 3 representa o EER do modelo de implementação relacional do SVE, incluindo restrições de chave, representadas como uma figura amarela de chave ao lado esquerdo do rótulo do atributo, tipo, apresentada ao lado direito do rótulo do atributo, nulidade, representada como um losango ao lado esquerdo do rótulo do atributo (losango azul para NOT NULL), e integridade referencial, com losango vermelho representado chaves estrangeiras.

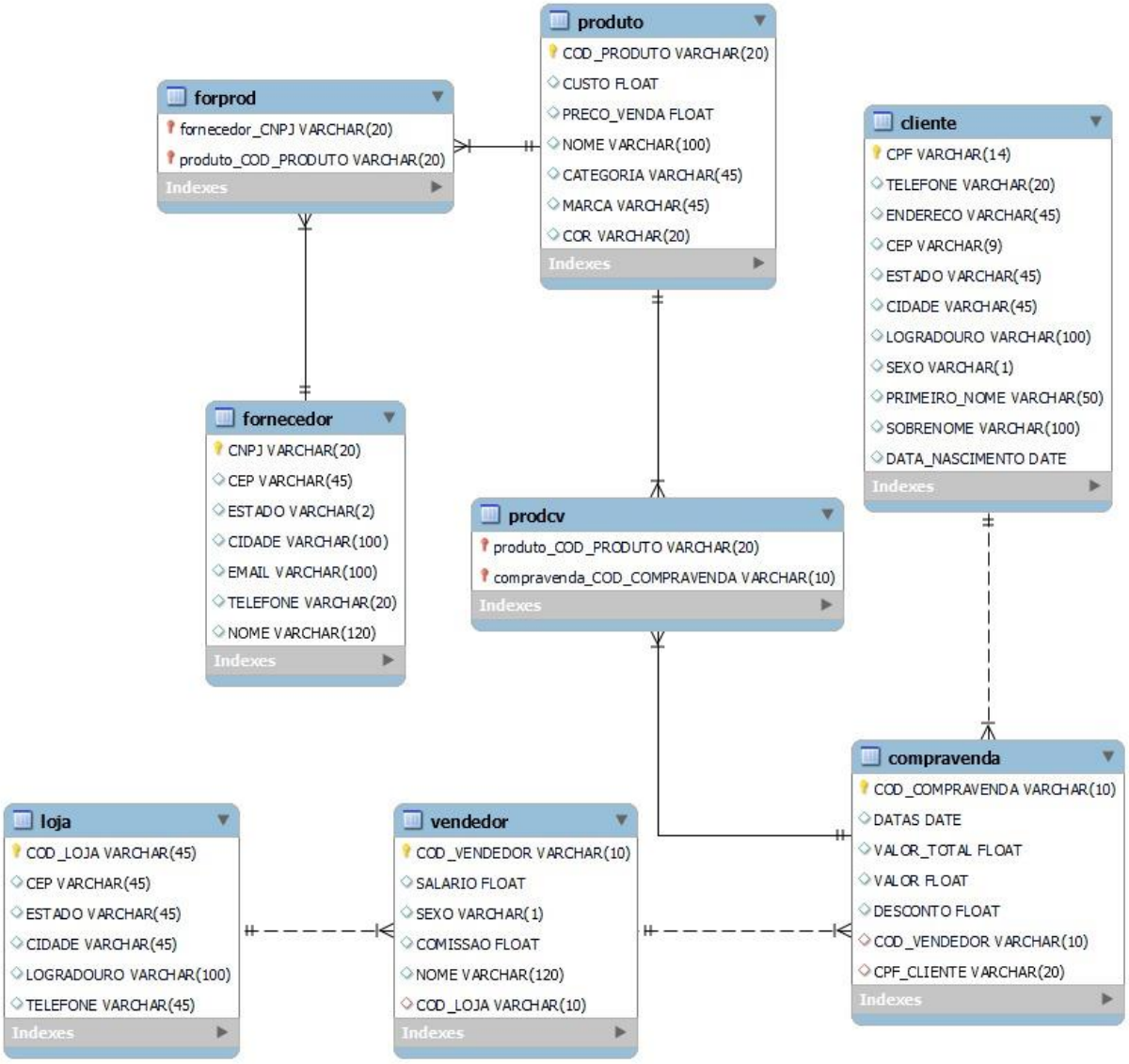

Figura 3: EER do modelo de implementação relacional do SVE.

5 Conclusão

O presente relatório apresentou o projeto do banco de dados SVE para um sistema de Varejo para Eletrônicos, em sua versão 1.0, pode ser utilizado por diversas lojas de varejo eletrônico para gerenciar suas vendas e compras com fornecedor. Especificamente, propusemos uma especificação de minimundo e apresentamos os requisitos funcionais e operacionais, o projeto conceitual, lógico e físico do banco de dados, concebido no paradigma relacional e projetado para ser implementado em um SGBD relacional comercial. Por fim, fizemos a transformação do banco de dados que estava em SQL, para a linguagem de programação Python.

6 Anexos: Scripts de Banco de Dados

Os arquivos SQL estão disponíveis no drive através DESSE LINK.

6.1. Script de criação do banco de dados e tabelas:

```
DROP SCHEMA IF EXISTS SVE;

CREATE SCHEMA IF NOT EXISTS SVE DEFAULT CHARACTER SET utf8;

DROP TABLE COMPRAVENDA;
DROP TABLE VENDEDOR;
DROP TABLE LOJA;
DROP TABLE FORNECEDOR;
DROP TABLE PRODUTO;
DROP TABLE CLIENTE;
DROP TABLE FORPROD;
DROP TABLE CVPROD;

CREATE TABLE IF NOT EXISTS CLIENTE(
  CPF VARCHAR(14) PRIMARY KEY,
  TELEFONE VARCHAR(20),
  ENDERECO VARCHAR(45),
  CEP VARCHAR(9),
  ESTADO VARCHAR(45),
  CIDADE VARCHAR(45),
```

```
  LOGRADOURO VARCHAR(100),
  SEXO VARCHAR(1),
  PRIMEIRO_NOME VARCHAR(50),
  SOBRENOME VARCHAR(100),
  DATA_NASCIMENTO DATE
);

CREATE TABLE IF NOT EXISTS LOJA(
  COD_LOJA VARCHAR(45) PRIMARY KEY,
  CEP VARCHAR(45),
  ESTADO VARCHAR(45),
  CIDADE VARCHAR(45),
  LOGRADOURO VARCHAR(100),
  TELEFONE VARCHAR(45)
);

CREATE TABLE IF NOT EXISTS VENDEDOR(
  COD_VENDEDOR VARCHAR(10) PRIMARY KEY,
  SALARIO FLOAT,
  SEXO VARCHAR(1),
  COMISSAO FLOAT,
  NOME VARCHAR(120),
  COD_LOJA VARCHAR(10),
  FOREIGN KEY (COD_LOJA) REFERENCES LOJA(COD_LOJA)
);

CREATE TABLE IF NOT EXISTS PRODUTO(
  COD_PRODUTO VARCHAR(20) PRIMARY KEY,
  CUSTO FLOAT,
  PRECO_VENDA FLOAT,
  NOME VARCHAR(100),
  CATEGORIA VARCHAR(45),
  MARCA VARCHAR(45),
  COR VARCHAR(20)
);
```

```
CREATE TABLE IF NOT EXISTS COMPRAVENDA(
  COD_COMPRAVENDA VARCHAR(10) PRIMARY KEY,
  DATAS DATE,
  VALOR_TOTAL FLOAT,
  VALOR FLOAT,
  DESCONTO FLOAT,
  COD_VENDEDOR VARCHAR(10),
  CPF_CLIENTE VARCHAR(20),
  COD_LOJA VARCHAR(10),
  COD_PRODUTO VARCHAR(45),
  FOREIGN KEY (COD_VENDEDOR) REFERENCES VENDEDOR(COD_VENDEDOR),
  FOREIGN KEY (CPF_CLIENTE) REFERENCES CLIENTE(CPF),
  FOREIGN KEY (COD_PRODUTO) REFERENCES PRODUTO(COD_PRODUTO),
  FOREIGN KEY (COD_LOJA) REFERENCES LOJA(COD_LOJA)
);

CREATE TABLE IF NOT EXISTS FORNECEDOR(
  CNPJ VARCHAR(20) PRIMARY KEY,
  CEP VARCHAR(45),
  ESTADO VARCHAR(2),
  CIDADE VARCHAR(100),
  EMAIL VARCHAR(100),
  TELEFONE VARCHAR(20),
  NOME VARCHAR(120),
  COD_PRODUTO VARCHAR(100),
  FOREIGN KEY (COD_PRODUTO) REFERENCES PRODUTO(COD_PRODUTO)
);

CREATE TABLE IF NOT EXISTS FORPROD(
CNPJ VARCHAR(20), COD_PRODUTO VARCHAR(20) PRIMARY KEY,
FOREIGN KEY (CNPJ) REFERENCES FORNECEDOR(CNPJ),
FOREIGN KEY (COD_PRODUTO) REFERENCES PRODUTO(COD_PRODUTO)
);

CREATE TABLE IF NOT EXISTS CVPROD(
```

```
COD_COMPRAVENDA VARCHAR(20), COD_PRODUTO VARCHAR(20) PRIMARY KEY,
FOREIGN KEY (COD_COMPRAVENDA) REFERENCES COMPRAVENDA(COD_COMPRAVENDA),
FOREIGN KEY (COD_PRODUTO) REFERENCES PRODUTO(COD_PRODUTO)
);
```

6.2. Script de inserção de informação nas tabelas:

```
-- queries de inserção em CLIENTE

INSERT INTO CLIENTE (CPF, TELEFONE, ENDERECO, CEP, ESTADO, CIDADE,
LOGRADOURO, SEXO, PRIMEIRO_NOME, SOBRENOME,
DATA_NASCIMENTO)
VALUES ('123.456.789-00', '(11) 1234-5678', 'Rua A, 123', '12345-678', 'São Paulo',
'São Paulo', 'Residencial ABC', 'M', 'João', 'Silva', '1990-01-01');

INSERT INTO CLIENTE (CPF, TELEFONE, ENDERECO, CEP, ESTADO, CIDADE,
LOGRADOURO, SEXO, PRIMEIRO_NOME, SOBRENOME,
DATA_NASCIMENTO)
VALUES ('987.654.321-00', '(22) 9876-5432', 'Avenida B, 456', '98765-432', 'Rio de
Janeiro', 'Rio de Janeiro', 'Apartamento XYZ', 'F', 'Maria', 'Souza', '1985-05-10');

INSERT INTO CLIENTE (CPF, TELEFONE, ENDERECO, CEP, ESTADO, CIDADE,
LOGRADOURO, SEXO, PRIMEIRO_NOME, SOBRENOME,
DATA_NASCIMENTO)
VALUES ('111.222.333-44', '(33) 1111-2222', 'Rua C, 789', '54321-098', 'Minas Gerais',
'Belo Horizonte', 'Casa 123', 'M', 'Pedro', 'Santos', '1982-12-25');

INSERT INTO CLIENTE (CPF, TELEFONE, ENDERECO, CEP, ESTADO, CIDADE,
LOGRADOURO, SEXO, PRIMEIRO_NOME, SOBRENOME,
DATA_NASCIMENTO)
VALUES ('555.444.333-22', '(44) 5555-4444', 'Avenida D, 987', '76543-210', 'Bahia',
'Salvador', 'Bloco ABCD', 'F', 'Ana', 'Oliveira', '1995-09-15');

INSERT INTO CLIENTE (CPF, TELEFONE, ENDERECO, CEP, ESTADO, CIDADE,
LOGRADOURO, SEXO, PRIMEIRO_NOME, SOBRENOME,
DATA_NASCIMENTO)
VALUES ('777.888.999-00', '(55) 7777-8888', 'Rua E, 321', '01234-567', 'São Paulo',
'São Paulo', 'Edifício EFGH', 'M', 'Lucas', 'Ferreira', '1988-07-20');

-- queries de inserção em LOJA

INSERT INTO LOJA (COD_LOJA, CEP, ESTADO, CIDADE, LOGRADOURO,
TELEFONE)
VALUES ('LJ001', '12345-678', 'São Paulo', 'São Paulo', 'Rua A, 123', '(11) 1234-5678');
```

```
INSERT INTO LOJA (COD_LOJA, CEP, ESTADO, CIDADE, LOGRADOURO,
TELEFONE)
VALUES ('LJ002', '98765-432', 'Rio de Janeiro', 'Rio de Janeiro', 'Avenida B, 456', '(21)
9876-5432');

INSERT INTO LOJA (COD_LOJA, CEP, ESTADO, CIDADE, LOGRADOURO,
TELEFONE)
VALUES ('LJ003', '54321-876', 'Minas Gerais', 'Belo Horizonte', 'Rua C, 789', '(31)
5432-1876');

INSERT INTO LOJA (COD_LOJA, CEP, ESTADO, CIDADE, LOGRADOURO,
TELEFONE)
VALUES ('LJ004', '76543-210', 'Bahia', 'Salvador', 'Avenida D, 987', '(71) 7654-3210');

INSERT INTO LOJA (COD_LOJA, CEP, ESTADO, CIDADE, LOGRADOURO,
TELEFONE)
VALUES ('LJ005', '32109-876', 'Espírito Santo', 'Vitória', 'Rua E, 654', '(27) 3210-
9876');

-- queries de inserção em PRODUTO

INSERT INTO PRODUTO (COD_PRODUTO, CUSTO, PRECO_VENDA, NOME,
CATEGORIA, MARCA, COR)
VALUES ('P001', 700.00, 999.00, 'iPhone 13', 'Eletrônicos', 'Apple', 'Preto');

INSERT INTO PRODUTO (COD_PRODUTO, CUSTO, PRECO_VENDA, NOME,
CATEGORIA, MARCA, COR)
VALUES ('P002', 800.00, 1099.00, 'Smart TV 4K', 'Eletrônicos', 'Samsung', 'Prata');

INSERT INTO PRODUTO (COD_PRODUTO, CUSTO, PRECO_VENDA, NOME,
CATEGORIA, MARCA, COR)
VALUES ('P003', 50.00, 79.00, 'Fone de Ouvido Bluetooth', 'Eletrônicos', 'Sony',
'Vermelho');

INSERT INTO PRODUTO (COD_PRODUTO, CUSTO, PRECO_VENDA, NOME,
CATEGORIA, MARCA, COR)
VALUES ('P004', 200.00, 299.00, 'Tablet Galaxy Tab A', 'Eletrônicos', 'Samsung',
'Preto');

INSERT INTO PRODUTO (COD_PRODUTO, CUSTO, PRECO_VENDA, NOME,
CATEGORIA, MARCA, COR)
VALUES ('P005', 300.00, 449.00, 'Câmera Digital', 'Eletrônicos', 'Canon', 'Branco');

-- queries de inserção em VENDEDOR

INSERT INTO VENDEDOR (COD_VENDEDOR, SALARIO, SEXO, COMISSAO,
NOME, COD_LOJA)
```

```
VALUES ('VD001', 3000.00, 'M', 0.00, 'João Silva', 'LJ001');

INSERT INTO VENDEDOR (COD_VENDEDOR, SALARIO, SEXO, COMISSAO,
NOME, COD_LOJA)
VALUES ('VD002', 2500.00, 'F', 0.05, 'Maria Santos', 'LJ002');

INSERT INTO VENDEDOR (COD_VENDEDOR, SALARIO, SEXO, COMISSAO,
NOME, COD_LOJA)
VALUES ('VD003', 4000.00, 'M', 0.00, 'Pedro Oliveira', 'LJ003');

INSERT INTO VENDEDOR (COD_VENDEDOR, SALARIO, SEXO, COMISSAO,
NOME, COD_LOJA)
VALUES ('VD004', 2800.00, 'F', 0.03, 'Ana Souza', 'LJ001');

INSERT INTO VENDEDOR (COD_VENDEDOR, SALARIO, SEXO, COMISSAO,
NOME, COD_LOJA)
VALUES ('VD005', 3500.00, 'M', 0.02, 'Carlos Mendes', 'LJ002');

-- queries de inserção em COMPRAVENDA

INSERT INTO COMPRAVENDA (COD_COMPRAVENDA, DATAS,
VALOR_TOTAL, VALOR, DESCONTO, COD_VENDEDOR, CPF_CLIENTE,
COD_LOJA, COD_PRODUTO)
VALUES ('CV001', '2023-05-17', 150.99, 129.99, 21.00, 'VD001', '123.456.789-00',
'LJ001', 'P001');

INSERT INTO COMPRAVENDA (COD_COMPRAVENDA, DATAS,
VALOR_TOTAL, VALOR, DESCONTO, COD_VENDEDOR, CPF_CLIENTE,
COD_LOJA, COD_PRODUTO)
VALUES ('CV002', '2023-05-18', 75.50, 75.50, 0.00, 'VD002', '987.654.321-00',
'LJ002', 'P002');

INSERT INTO COMPRAVENDA (COD_COMPRAVENDA, DATAS,
VALOR_TOTAL, VALOR, DESCONTO, COD_VENDEDOR, CPF_CLIENTE,
COD_LOJA, COD_PRODUTO)
VALUES ('CV003', '2023-05-19', 200.00, 200.00, 0.00, 'VD003', '987.654.321-00',
'LJ001', 'P003');

INSERT INTO COMPRAVENDA (COD_COMPRAVENDA, DATAS,
VALOR_TOTAL, VALOR, DESCONTO, COD_VENDEDOR, CPF_CLIENTE,
COD_LOJA, COD_PRODUTO)
VALUES ('CV004', '2023-05-20', 500.00, 400.00, 100.00, 'VD002', '111.222.333-44',
'LJ002', 'P004');

INSERT INTO COMPRAVENDA (COD_COMPRAVENDA, DATAS,
VALOR_TOTAL, VALOR, DESCONTO, COD_VENDEDOR, CPF_CLIENTE,
COD_LOJA, COD_PRODUTO)
```

```
VALUES ('CV005', '2023-05-21', 1000.00, 950.00, 50.00, 'VD001', '555.444.333-22',
'LJ003', 'P005');

-- queries de inserção em FORNECEDOR

INSERT INTO FORNECEDOR (CNPJ, CEP, ESTADO, CIDADE, EMAIL,
TELEFONE, NOME, COD_PRODUTO)
VALUES ('12345678900001', '12345-678', 'SP', 'São Paulo',
'fornecedor1@example.com', '1111111111', 'Fornecedor 1', 'P001');

INSERT INTO FORNECEDOR (CNPJ, CEP, ESTADO, CIDADE, EMAIL,
TELEFONE, NOME, COD_PRODUTO)
VALUES ('98765432100002', '98765-432', 'RJ', 'Rio de Janeiro',
'fornecedor2@example.com', '2222222222', 'Fornecedor 2', 'P002');

INSERT INTO FORNECEDOR (CNPJ, CEP, ESTADO, CIDADE, EMAIL,
TELEFONE, NOME, COD_PRODUTO)
VALUES ('45678912300003', '54321-876', 'MG', 'Belo Horizonte',
'fornecedor3@example.com', '3333333333', 'Fornecedor 3', 'P003');

INSERT INTO FORNECEDOR (CNPJ, CEP, ESTADO, CIDADE, EMAIL,
TELEFONE, NOME, COD_PRODUTO)
VALUES ('78912345600004', '76543-210', 'RS', 'Porto Alegre',
'fornecedor4@example.com', '4444444444', 'Fornecedor 4', 'P004');

INSERT INTO FORNECEDOR (CNPJ, CEP, ESTADO, CIDADE, EMAIL,
TELEFONE, NOME, COD_PRODUTO)
VALUES ('32165498700005', '23456-789', 'SC', 'Florianópolis',
'fornecedor5@example.com', '5555555555', 'Fornecedor 5', 'P005');
```

6.3. Script das queries:

```
-- Q001  Listar todas as lojas cadastradas no sistema
SELECT COD_LOJA FROM LOJA;


-- Q002 Encontrar a loja com o maior número de vendas
SELECT COUNT(A.VALOR), B.COD_LOJA FROM COMPRAVENDA A
INNER JOIN LOJA B ON A.COD_LOJA = B.COD_LOJA
GROUP BY B.COD_LOJA
LIMIT 1;


-- Q003 Listar todos os vendedores cadastrados no sistema
SELECT DISTINCT NOME FROM VENDEDOR;
```

```
-- Q004 Encontrar o vendedor que mais vendeu
SELECT SUM(A.VALOR), B.COD_VENDEDOR FROM COMPRAVENDA A
INNER JOIN VENDEDOR B ON A.COD_VENDEDOR = B.COD_VENDEDOR
WHERE
MONTH(DATAS) = MONTH(CURRENT_DATE)
GROUP BY B.COD_VENDEDOR
ORDER BY SUM(A.VALOR) DESC
LIMIT 1;

-- Q005 Listar todas as vendas realizadas no sistema
SELECT * FROM COMPRAVENDA;

-- Q006 Encontrar as vendas realizadas entre duas datas específicas
SELECT * FROM COMPRAVENDA
WHERE DATAS BETWEEN '2023-05-18' AND '2023-05-20';

-- Q007 NOVA: Inserir um novo produto na tabela proutos
INSERT INTO PRODUTO (COD_PRODUTO, CUSTO, PRECO_VENDA, NOME, CATEGORIA, MARCA, COR)
VALUES ('P010', 900.00, 1200.00, 'iPhone 14', 'Eletrônicos', 'Apple', 'Cinza');

-- Q008 Listar todos os fornecedores cadastrados no sistema
SELECT * FROM FORNECEDOR;

-- Q009 NOVA: Inserir um novo fornecedor
INSERT INTO FORNECEDOR (CNPJ, CEP, ESTADO, CIDADE, EMAIL, TELEFONE, NOME, COD_PRODUTO)
VALUES ('32165498701015', '12345-000', 'AM', 'Manaus', 'fornecedor6@example.com', '9999855555', 'Fornecedor 6', 'P005');

-- Q010 Encontrar a receita total de uma loja
SELECT SUM(A.VALOR), B.COD_LOJA FROM COMPRAVENDA A
INNER JOIN LOJA B ON A.COD_LOJA = B.COD_LOJA
WHERE B.COD_LOJA = 'LJ003'
GROUP BY B.COD_LOJA;
```

```
-- Q011 Listar vendas por vendedor (qual vendedor realizou a venda)
SELECT A.COD_COMPRAVENDA, B.COD_VENDEDOR FROM COMPRAVENDA A
INNER JOIN VENDEDOR B ON A.COD_VENDEDOR = B.COD_VENDEDOR
GROUP BY A.COD_COMPRAVENDA;

-- Q012 NOVA: Inserir um novo cliente no banco de dados
INSERT INTO CLIENTE (CPF, TELEFONE, ENDERECO, CEP, ESTADO, CIDADE, LOGRADOURO, SEXO, PRIMEIRO_NOME, SOBRENOME, DATA_NASCIMENTO)
VALUES ('121.232.333-44', '(33) 4321-1234', 'Rua XYZ, 32', '43256-020', 'Belo Horizonte', 'BH', 'Residencial TPNB', 'H', 'Marcius', 'Cavalcante', '1950-02-12');

-- Q013 Listar clientes por vendedor
SELECT A.NOME AS NOME_VENDEDOR, B.PRIMEIRO_NOME AS NOME_CLIENTE, B.SOBRENOME AS SOBRENOME_CLIENTE FROM VENDEDOR A
INNER JOIN COMPRAVENDA C ON A.COD_VENDEDOR = C.COD_VENDEDOR
INNER JOIN CLIENTE B ON C.CPF_CLIENTE = B.CPF
ORDER BY A.NOME;

-- Q014 Encontrar os produtos que nunca foram vendidos
SELECT A.NOME AS PRODUTOS, B.COD_COMPRAVENDA FROM PRODUTO A
LEFT JOIN COMPRAVENDA B ON A.COD_PRODUTO = B.COD_PRODUTO
WHERE B.COD_COMPRAVENDA IS NULL
ORDER BY B.COD_COMPRAVENDA;

-- Q015 Listar todos os clientes cadastrados no sistema
SELECT * FROM CLIENTE;

-- Q016 NOVA: Contar quantas compras foram efetudadas por cliente
SELECT A.PRIMEIRO_NOME, COUNT(*) FROM CLIENTE A
INNER JOIN COMPRAVENDA B ON A.CPF = B.CPF_CLIENTE
GROUP BY B.CPF_CLIENTE;

-- Q017 NOVA: Visualizar compras realizadas por cliente
SELECT B.COD_COMPRAVENDA AS COMPRA, A.PRIMEIRO_NOME, A.SOBRENOME, C.NOME AS PRODUTO, B.VALOR_TOTAL FROM CLIENTE A
```

```
INNER JOIN COMPRAVENDA B ON A.CPF = B.CPF_CLIENTE
INNER JOIN PRODUTO C ON B.COD_PRODUTO = C.COD_PRODUTO
GROUP BY B.COD_COMPRAVENDA;

-- Q018 NOVA: Atualizar dados cadastrais do cliente
UPDATE CLIENTE SET SOBRENOME = 'Filho'
WHERE CPF = '123.456.789-00';
```

7 Anexos Python: Scripts de Banco de Dados na Linguagem de Programação Python.

Os arquivos em Python estão disponíveis no drive através DESSE LINK.


```
{
 "nbformat": 4,
 "nbformat_minor": 0,
 "metadata": {
  "colab": {
   "provenance": []
  },
  "kernelspec": {
   "name": "python3",
   "display_name": "Python 3"
  },
  "language_info": {
   "name": "python"
  }
 },
 "cells": [
  {
   "cell_type": "markdown",
   "source": [
    "## **Etapa 5 - Dataframes em Python**"
   ],
   "metadata": {
    "id": "iT-DAc7kqMZ7"
```

```
    }
   },
   {
    "cell_type": "markdown",
    "source": [
     "## Nesta etapa, você deverá converter as tabelas do banco de dados em dataframes para operações em Python, bem como as consultas (queries) presentes na especificação de requisitos funcionais.\n",
     "\n",
     "## Você deve entregar um notebook em Python, no formato .ipynb, contendo a codificação para fazer as conversões e relatar as dificuldades encontradas, bem como as alternativas adotadas para enfrentá-las."
    ],
    "metadata": {
     "id": "tLWXWvcVqOhO"
    }
   },
   {
    "cell_type": "code",
    "source": [
     "#Rodar as bibliotecas\n",
     "import sqlite3\n",
     "import pandas as pd\n"
    ],
    "metadata": {
     "id": "Qs0IQrOy_Fch"
    },
    "execution_count": null,
    "outputs": []
   },
   {
    "cell_type": "code",
    "source": [
     "from google.colab import files\n",
     "uploaded = files.upload()"
    ],
    "metadata": {
```

```
      "colab": {
        "base_uri": "https://localhost:8080/",
        "height": 75
      },
      "id": "pIVOtlyB3Bmo",
      "outputId": "1f9da909-0b63-4490-a37b-47d2cdadcf12"
    },
    "execution_count": 1,
    "outputs": [
      {
        "output_type": "display_data",
        "data": {
          "text/plain": [
            "<IPython.core.display.HTML object>"
          ],
          "text/html": [
            "\n",
            "      <input type=\"file\" id=\"files-7a356017-2dee-455c-9e65-9722557f5113\" name=\"files[]\" multiple disabled\n",
            "        style=\"border:none\" />\n",
            "     <output id=\"result-7a356017-2dee-455c-9e65-9722557f5113\">\n",
            "      Upload widget is only available when the cell has been executed in the\n",
            "      current browser session. Please rerun this cell to enable.\n",
            "      </output>\n",
            "      <script>// Copyright 2017 Google LLC\n",
            "//\n",
            "// Licensed under the Apache License, Version 2.0 (the \"License\");\n",
            "// you may not use this file except in compliance with the License.\n",
            "// You may obtain a copy of the License at\n",
            "//\n",
            "//      http://www.apache.org/licenses/LICENSE-2.0\n",
            "//\n",
            "// Unless required by applicable law or agreed to in writing, software\n",
            "// distributed under the License is distributed on an \"AS IS\" BASIS,\n",
            "// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.\n",
```

```
"// See the License for the specific language governing permissions and\n",
"// limitations under the License.\n",
"\n",
"/**\n",
" * @fileoverview Helpers for google.colab Python module.\n",
" */\n",
"(function(scope) {\n",
"function span(text, styleAttributes = {}) {\n",
"  const element = document.createElement('span');\n",
"  element.textContent = text;\n",
"  for (const key of Object.keys(styleAttributes)) {\n",
"    element.style[key] = styleAttributes[key];\n",
"  }\n",
"  return element;\n",
"}\n",
"\n",
"// Max number of bytes which will be uploaded at a time.\n",
"const MAX_PAYLOAD_SIZE = 100 * 1024;\n",
"\n",
"function _uploadFiles(inputId, outputId) {\n",
"  const steps = uploadFilesStep(inputId, outputId);\n",
"  const outputElement = document.getElementById(outputId);\n",
"  // Cache steps on the outputElement to make it available for the next call\n",
"  // to uploadFilesContinue from Python.\n",
"  outputElement.steps = steps;\n",
"\n",
"  return _uploadFilesContinue(outputId);\n",
"}\n",
"\n",
"// This is roughly an async generator (not supported in the browser yet),\n",
"// where there are multiple asynchronous steps and the Python side is going\n",
"// to poll for completion of each step.\n",
"// This uses a Promise to block the python side on completion of each step,\n",
"// then passes the result of the previous step as the input to the next step.\n",
"function _uploadFilesContinue(outputId) {\n",
"  const outputElement = document.getElementById(outputId);\n",
```

```
"  const steps = outputElement.steps;\n",
"\n",
"  const next = steps.next(outputElement.lastPromiseValue);\n",
"  return Promise.resolve(next.value.promise).then((value) => {\n",
"    // Cache the last promise value to make it available to the next\n",
"    // step of the generator.\n",
"    outputElement.lastPromiseValue = value;\n",
"    return next.value.response;\n",
"  });\n",
"}\n",
"\n",
"/**\n",
" * Generator function which is called between each async step of the upload\n",
" * process.\n",
" * @param {string} inputId Element ID of the input file picker element.\n",
" * @param {string} outputId Element ID of the output display.\n",
" * @return {!Iterable<!Object>} Iterable of next steps.\n",
" */\n",
"function* uploadFilesStep(inputId, outputId) {\n",
"  const inputElement = document.getElementById(inputId);\n",
"  inputElement.disabled = false;\n",
"\n",
"  const outputElement = document.getElementById(outputId);\n",
"  outputElement.innerHTML = '';\n",
"\n",
"  const pickedPromise = new Promise((resolve) => {\n",
"    inputElement.addEventListener('change', (e) => {\n",
"      resolve(e.target.files);\n",
"    });\n",
"  });\n",
"\n",
"  const cancel = document.createElement('button');\n",
"  inputElement.parentElement.appendChild(cancel);\n",
"  cancel.textContent = 'Cancel upload';\n",
"  const cancelPromise = new Promise((resolve) => {\n",
"    cancel.onclick = () => {\n",
```

```
"      resolve(null);\n",
"    };\n",
"  });\n",
"\n",
"  // Wait for the user to pick the files.\n",
"  const files = yield {\n",
"    promise: Promise.race([pickedPromise, cancelPromise]),\n",
"    response: {\n",
"      action: 'starting',\n",
"    }\n",
"  };\n",
"\n",
"  cancel.remove();\n",
"\n",
"  // Disable the input element since further picks are not allowed.\n",
"  inputElement.disabled = true;\n",
"\n",
"  if (!files) {\n",
"    return {\n",
"      response: {\n",
"        action: 'complete',\n",
"      }\n",
"    };\n",
"  }\n",
"\n",
"  for (const file of files) {\n",
"    const li = document.createElement('li');\n",
"    li.append(span(file.name, {fontWeight: 'bold'}));\n",
"    li.append(span(\n",
"        `(${file.type || 'n/a'}) - ${file.size} bytes, ` +\n",
"        `last modified: ${\n",
"            file.lastModifiedDate ? file.lastModifiedDate.toLocaleDateString() :\n",
"                                    'n/a'} - `));\n",
"    const percent = span('0% done');\n",
"    li.appendChild(percent);\n",
"\n",
```

```
"    outputElement.appendChild(li);\n",
"\n",
"    const fileDataPromise = new Promise((resolve) => {\n",
"      const reader = new FileReader();\n",
"      reader.onload = (e) => {\n",
"        resolve(e.target.result);\n",
"      };\n",
"      reader.readAsArrayBuffer(file);\n",
"    });\n",
"    // Wait for the data to be ready.\n",
"    let fileData = yield {\n",
"      promise: fileDataPromise,\n",
"      response: {\n",
"        action: 'continue',\n",
"      }\n",
"    };\n",
"\n",
"    // Use a chunked sending to avoid message size limits. See b/62115660.\n",
"    let position = 0;\n",
"    do {\n",
"                  const length = Math.min(fileData.byteLength - position,
MAX_PAYLOAD_SIZE);\n",
"      const chunk = new Uint8Array(fileData, position, length);\n",
"      position += length;\n",
"\n",
"      const base64 = btoa(String.fromCharCode.apply(null, chunk));\n",
"      yield {\n",
"        response: {\n",
"          action: 'append',\n",
"          file: file.name,\n",
"          data: base64,\n",
"        },\n",
"      };\n",
"\n",
"      let percentDone = fileData.byteLength === 0 ?\n",
"          100 :\n",
```

```
        "          Math.round((position / fileData.byteLength) * 100);\n",
        "      percent.textContent = `${percentDone}% done`;\n",
        "\n",
        "    } while (position < fileData.byteLength);\n",
        "  }\n",
        "\n",
        "  // All done.\n",
        "  yield {\n",
        "    response: {\n",
        "      action: 'complete',\n",
        "    }\n",
        "  };\n",
        "}\n",
        "\n",
        "scope.google = scope.google || {};\n",
        "scope.google.colab = scope.google.colab || {};\n",
        "scope.google.colab._files = {\n",
        "  _uploadFiles,\n",
        "  _uploadFilesContinue,\n",
        "};\n",
        "})(self);\n",
        "</script> "
       ]
      },
      "metadata": {}
    },
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "Saving trabalhosve.db to trabalhosve.db\n"
      ]
    }
  ]
},
{
```

```
"cell_type": "code",
"source": [
 "#Refiz o banco no proprio sqlite3, NAO PRECISA RODAR, SO DEIXEI PARA MOSTRAR COMO FIZEMOS\n",
 "\n",
 "import sqlite3\n",
 "\n",
 "# Conectar ao banco de dados (será criado se não existir)\n",
 "conn = sqlite3.connect('trabalhosve.db')\n",
 "\n",
 "# Criar um cursor para executar comandos SQL\n",
 "cursor = conn.cursor()\n",
 "\n",
 "# DROP SCHEMA\n",
 "cursor.execute(\"DROP TABLE IF EXISTS SVE;\")\n",
 "\n",
 "# CREATE SCHEMA\n",
 "cursor.execute(\"CREATE TABLE IF NOT EXISTS SVE (id INTEGER PRIMARY KEY AUTOINCREMENT);\")\n",
 "\n",
 "# DROP TABLES\n",
 "cursor.execute(\"DROP TABLE IF EXISTS COMPRAVENDA;\")\n",
 "cursor.execute(\"DROP TABLE IF EXISTS VENDEDOR;\")\n",
 "cursor.execute(\"DROP TABLE IF EXISTS LOJA;\")\n",
 "cursor.execute(\"DROP TABLE IF EXISTS FORNECEDOR;\")\n",
 "cursor.execute(\"DROP TABLE IF EXISTS PRODUTO;\")\n",
 "cursor.execute(\"DROP TABLE IF EXISTS CLIENTE;\")\n",
 "\n",
 "# Criar a tabela \"PRODUTO\" se ela não existir\n",
 "cursor.execute('''CREATE TABLE IF NOT EXISTS CLIENTE(\n",
 "  CPF VARCHAR(14) PRIMARY KEY,\n",
 "  TELEFONE VARCHAR(20),\n",
 "  ENDERECO VARCHAR(45),\n",
 "  CEP VARCHAR(9),\n",
 "  ESTADO VARCHAR(45),\n",
 "  CIDADE VARCHAR(45),\n",
```

```
"  LOGRADOURO VARCHAR(100),\n",
"  SEXO VARCHAR(1),\n",
"  PRIMEIRO_NOME VARCHAR(50),\n",
"  SOBRENOME VARCHAR(100),\n",
"  DATA_NASCIMENTO DATE\n",
")'')\n",
"\n",
"# Inserir os valores na tabela \"CLIENTE\"\n",
"cursor.execute(\"\"\"INSERT INTO CLIENTE (CPF, TELEFONE, ENDERECO, CEP, ESTADO, CIDADE, LOGRADOURO, SEXO, PRIMEIRO_NOME, SOBRENOME, DATA_NASCIMENTO)\n",
"            VALUES ('123.456.789-00', '(11) 1234-5678', 'Rua A, 123', '12345-678', 'São Paulo', 'São Paulo', 'Residencial ABC', 'M', 'João', 'Silva', '1990-01-01')\"\"\")\n",
"\n",
"cursor.execute(\"\"\"INSERT INTO CLIENTE (CPF, TELEFONE, ENDERECO, CEP, ESTADO, CIDADE, LOGRADOURO, SEXO, PRIMEIRO_NOME, SOBRENOME, DATA_NASCIMENTO)\n",
"            VALUES ('987.654.321-00', '(22) 9876-5432', 'Avenida B, 456', '98765-432', 'Rio de Janeiro', 'Rio de Janeiro', 'Apartamento XYZ', 'F', 'Maria', 'Souza', '1985-05-10')\"\"\")\n",
"\n",
"cursor.execute(\"\"\"INSERT INTO CLIENTE (CPF, TELEFONE, ENDERECO, CEP, ESTADO, CIDADE, LOGRADOURO, SEXO, PRIMEIRO_NOME, SOBRENOME, DATA_NASCIMENTO)\n",
"            VALUES ('111.222.333-44', '(33) 1111-2222', 'Rua C, 789', '54321-098', 'Minas Gerais', 'Belo Horizonte', 'Casa 123', 'M', 'Pedro', 'Santos', '1982-12-25')\"\"\")\n",
"\n",
"cursor.execute(\"\"\"INSERT INTO CLIENTE (CPF, TELEFONE, ENDERECO, CEP, ESTADO, CIDADE, LOGRADOURO, SEXO, PRIMEIRO_NOME, SOBRENOME, DATA_NASCIMENTO)\n",
"            VALUES ('555.444.333-22', '(44) 5555-4444', 'Avenida D, 987', '76543-210', 'Bahia', 'Salvador', 'Bloco ABCD', 'F', 'Ana', 'Oliveira', '1995-09-15')\"\"\")\n",
"\n",
"cursor.execute(\"\"\"INSERT INTO CLIENTE (CPF, TELEFONE, ENDERECO, CEP, ESTADO, CIDADE, LOGRADOURO, SEXO, PRIMEIRO_NOME, SOBRENOME, DATA_NASCIMENTO)\n",
"            VALUES ('777.888.999-00', '(55) 7777-8888', 'Rua E, 321', '01234-567', 'São Paulo', 'São Paulo', 'Edifício EFGH', 'M', 'Lucas', 'Ferreira', '1988-07-20')\"\"\")\n",
"\n",
"# Criar tabela LOJA se ela não existir\n",
```

```
"cursor.execute('''CREATE TABLE IF NOT EXISTS LOJA(\n",
"    COD_LOJA VARCHAR(45) PRIMARY KEY,\n",
"    CEP VARCHAR(45),\n",
"    ESTADO VARCHAR(45),\n",
"    CIDADE VARCHAR(45),\n",
"    LOGRADOURO VARCHAR(100),\n",
"    TELEFONE VARCHAR(45)\n",
")''')\n",
"\n",
"# Inserir registros na tabela LOJA\n",
"cursor.execute('''INSERT INTO LOJA (COD_LOJA, CEP, ESTADO, CIDADE, LOGRADOURO, TELEFONE)\n",
"VALUES ('LJ001', '12345-678', 'São Paulo', 'São Paulo', 'Rua A, 123', '(11) 1234-5678')''')\n",
"\n",
"cursor.execute('''INSERT INTO LOJA (COD_LOJA, CEP, ESTADO, CIDADE, LOGRADOURO, TELEFONE)\n",
"VALUES ('LJ002', '98765-432', 'Rio de Janeiro', 'Rio de Janeiro', 'Avenida B, 456', '(21) 9876-5432')''')\n",
"\n",
"cursor.execute('''INSERT INTO LOJA (COD_LOJA, CEP, ESTADO, CIDADE, LOGRADOURO, TELEFONE)\n",
"VALUES ('LJ003', '54321-876', 'Minas Gerais', 'Belo Horizonte', 'Rua C, 789', '(31) 5432-1876')''')\n",
"\n",
"cursor.execute('''INSERT INTO LOJA (COD_LOJA, CEP, ESTADO, CIDADE, LOGRADOURO, TELEFONE)\n",
"VALUES ('LJ004', '76543-210', 'Bahia', 'Salvador', 'Avenida D, 987', '(71) 7654-3210')''')\n",
"\n",
"cursor.execute('''INSERT INTO LOJA (COD_LOJA, CEP, ESTADO, CIDADE, LOGRADOURO, TELEFONE)\n",
"VALUES ('LJ005', '32109-876', 'Espírito Santo', 'Vitória', 'Rua E, 654', '(27) 3210-9876')''')\n",
"\n",
"\n",
"# Criar tabela PRODUTO se ela não existir\n",
"cursor.execute('''CREATE TABLE IF NOT EXISTS PRODUTO(\n",
"    COD_PRODUTO VARCHAR(20) PRIMARY KEY,\n",
```

```
"    CUSTO FLOAT,\n",
"    PRECO_VENDA FLOAT,\n",
"    NOME VARCHAR(100),\n",
"    CATEGORIA VARCHAR(45),\n",
"    MARCA VARCHAR(45),\n",
"    COR VARCHAR(20)\n",
")'')\n",
"\n",
"# Inserir registros na tabela PRODUTO\n",
"cursor.execute(''INSERT INTO PRODUTO (COD_PRODUTO, CUSTO, PRECO_VENDA, NOME, CATEGORIA, MARCA, COR)\n",
"VALUES ('P001', 700.00, 999.00, 'iPhone 13', 'Eletrônicos', 'Apple', 'Preto')'')\n",
"\n",
"cursor.execute(''INSERT INTO PRODUTO (COD_PRODUTO, CUSTO, PRECO_VENDA, NOME, CATEGORIA, MARCA, COR)\n",
"VALUES ('P002', 800.00, 1099.00, 'Smart TV 4K', 'Eletrônicos', 'Samsung', 'Prata')'')\n",
"\n",
"cursor.execute(''INSERT INTO PRODUTO (COD_PRODUTO, CUSTO, PRECO_VENDA, NOME, CATEGORIA, MARCA, COR)\n",
"VALUES ('P003', 50.00, 79.00, 'Fone de Ouvido Bluetooth', 'Eletrônicos', 'Sony', 'Vermelho')'')\n",
"\n",
"cursor.execute(''INSERT INTO PRODUTO (COD_PRODUTO, CUSTO, PRECO_VENDA, NOME, CATEGORIA, MARCA, COR)\n",
"VALUES ('P004', 200.00, 299.00, 'Tablet Galaxy Tab A', 'Eletrônicos', 'Samsung', 'Preto')'')\n",
"\n",
"cursor.execute(''INSERT INTO PRODUTO (COD_PRODUTO, CUSTO, PRECO_VENDA, NOME, CATEGORIA, MARCA, COR)\n",
"VALUES ('P005', 300.00, 449.00, 'Câmera Digital', 'Eletrônicos', 'Canon', 'Branco')'')\n",
"\n",
"\n",
"# Criar tabela VENDEDOR se ela não existir\n",
"cursor.execute(''CREATE TABLE IF NOT EXISTS VENDEDOR(\n",
"    COD_VENDEDOR VARCHAR(10) PRIMARY KEY,\n",
"    SALARIO FLOAT,\n",
```

```
"    SEXO VARCHAR(1),\n",
"    COMISSAO FLOAT,\n",
"    NOME VARCHAR(120),\n",
"    COD_LOJA VARCHAR(10),\n",
"    FOREIGN KEY (COD_LOJA) REFERENCES LOJA(COD_LOJA)\n",
")'')\n",
"\n",
"# Inserir registros na tabela VENDEDOR\n",
"cursor.execute(''INSERT INTO VENDEDOR (COD_VENDEDOR, SALARIO, SEXO, COMISSAO, NOME, COD_LOJA)\n",
"VALUES ('VD001', 3000.00, 'M', 0.00, 'João Silva', 'LJ001')'')\n",
"\n",
"cursor.execute(''INSERT INTO VENDEDOR (COD_VENDEDOR, SALARIO, SEXO, COMISSAO, NOME, COD_LOJA)\n",
"VALUES ('VD002', 2500.00, 'F', 0.05, 'Maria Santos', 'LJ002')'')\n",
"\n",
"cursor.execute(''INSERT INTO VENDEDOR (COD_VENDEDOR, SALARIO, SEXO, COMISSAO, NOME, COD_LOJA)\n",
"VALUES ('VD003', 4000.00, 'M', 0.00, 'Pedro Oliveira', 'LJ003')'')\n",
"\n",
"cursor.execute(''INSERT INTO VENDEDOR (COD_VENDEDOR, SALARIO, SEXO, COMISSAO, NOME, COD_LOJA)\n",
"VALUES ('VD004', 2800.00, 'F', 0.03, 'Ana Souza', 'LJ001')'')\n",
"\n",
"cursor.execute(''INSERT INTO VENDEDOR (COD_VENDEDOR, SALARIO, SEXO, COMISSAO, NOME, COD_LOJA)\n",
"VALUES ('VD005', 3500.00, 'M', 0.02, 'Carlos Mendes', 'LJ002')'')\n",
"\n",
"# Criar tabela COMPRAVENDA se ela não existir\n",
"cursor.execute(''CREATE TABLE IF NOT EXISTS COMPRAVENDA(\n",
"    COD_COMPRAVENDA VARCHAR(10) PRIMARY KEY,\n",
"    DATAS DATE,\n",
"    VALOR_TOTAL FLOAT,\n",
"    VALOR FLOAT,\n",
"    DESCONTO FLOAT,\n",
"    COD_VENDEDOR VARCHAR(10),\n",
"    CPF_CLIENTE VARCHAR(20),\n",
```

```
"    COD_LOJA VARCHAR(10),\n",
"    COD_PRODUTO VARCHAR(45),\n",
"                FOREIGN KEY (COD_VENDEDOR) REFERENCES VENDEDOR(COD_VENDEDOR),\n",
"    FOREIGN KEY (CPF_CLIENTE) REFERENCES CLIENTE(CPF),\n",
"    FOREIGN KEY (COD_LOJA) REFERENCES LOJA(COD_LOJA),\n",
"                FOREIGN KEY (COD_PRODUTO) REFERENCES PRODUTO(COD_PRODUTO) \n",
")'')\n",
"\n",
"# Inserir registros na tabela COMPRAVENDA\n",
"cursor.execute(''INSERT INTO COMPRAVENDA (COD_COMPRAVENDA, DATAS, VALOR_TOTAL, VALOR, DESCONTO, COD_VENDEDOR, CPF_CLIENTE, COD_LOJA, COD_PRODUTO)\n",
"VALUES ('CV001', '2023-05-17', 150.99, 129.99, 21.00, 'VD001', '123.456.789-00', 'LJ001', 'P001')'')\n",
"\n",
"cursor.execute(''INSERT INTO COMPRAVENDA (COD_COMPRAVENDA, DATAS, VALOR_TOTAL, VALOR, DESCONTO, COD_VENDEDOR, CPF_CLIENTE, COD_LOJA, COD_PRODUTO)\n",
"VALUES ('CV002', '2023-05-18', 75.50, 75.50, 0.00, 'VD002', '987.654.321-00', 'LJ002', 'P002')'')\n",
"\n",
"cursor.execute(''INSERT INTO COMPRAVENDA (COD_COMPRAVENDA, DATAS, VALOR_TOTAL, VALOR, DESCONTO, COD_VENDEDOR, CPF_CLIENTE, COD_LOJA, COD_PRODUTO)\n",
"VALUES ('CV003', '2023-05-19', 200.00, 200.00, 0.00, 'VD003', '987.654.321-00', 'LJ001', 'P003')'')\n",
"\n",
"cursor.execute(''INSERT INTO COMPRAVENDA (COD_COMPRAVENDA, DATAS, VALOR_TOTAL, VALOR, DESCONTO, COD_VENDEDOR, CPF_CLIENTE, COD_LOJA, COD_PRODUTO)\n",
"VALUES ('CV004', '2023-05-20', 500.00, 400.00, 100.00, 'VD002', '111.222.333-44', 'LJ002', 'P004')'')\n",
"\n",
"cursor.execute(''INSERT INTO COMPRAVENDA (COD_COMPRAVENDA, DATAS, VALOR_TOTAL, VALOR, DESCONTO, COD_VENDEDOR, CPF_CLIENTE, COD_LOJA, COD_PRODUTO)\n",
"VALUES ('CV005', '2023-05-21', 1000.00, 950.00, 50.00, 'VD001', '555.444.333-22', 'LJ003', 'P005')'')\n",
```

```
"\n",
"\n",
"# Criar tabela FORNECEDOR se ela não existir\n",
"cursor.execute('''CREATE TABLE IF NOT EXISTS FORNECEDOR(\n",
"    CNPJ VARCHAR(20) PRIMARY KEY,\n",
"    CEP VARCHAR(45),\n",
"    ESTADO VARCHAR(2),\n",
"    CIDADE VARCHAR(100),\n",
"    EMAIL VARCHAR(100),\n",
"    TELEFONE VARCHAR(20),\n",
"    NOME VARCHAR(120),\n",
"    COD_PRODUTO VARCHAR(100),\n",
"    FOREIGN KEY (COD_PRODUTO) REFERENCES PRODUTO(COD_PRODUTO)\n",
")''')\n",
"\n",
"# Inserir registros na tabela FORNECEDOR\n",
"cursor.execute('''INSERT INTO FORNECEDOR (CNPJ, CEP, ESTADO, CIDADE, EMAIL, TELEFONE, NOME, COD_PRODUTO)\n",
"VALUES ('12345678900001', '12345-678', 'SP', 'São Paulo', 'fornecedor1@example.com', '1111111111', 'Fornecedor 1', 'P001')''')\n",
"\n",
"cursor.execute('''INSERT INTO FORNECEDOR (CNPJ, CEP, ESTADO, CIDADE, EMAIL, TELEFONE, NOME, COD_PRODUTO)\n",
"VALUES ('98765432100002', '98765-432', 'RJ', 'Rio de Janeiro', 'fornecedor2@example.com', '2222222222', 'Fornecedor 2', 'P002')''')\n",
"\n",
"cursor.execute('''INSERT INTO FORNECEDOR (CNPJ, CEP, ESTADO, CIDADE, EMAIL, TELEFONE, NOME, COD_PRODUTO)\n",
"VALUES ('45678912300003', '54321-876', 'MG', 'Belo Horizonte', 'fornecedor3@example.com', '3333333333', 'Fornecedor 3', 'P003')''')\n",
"\n",
"cursor.execute('''INSERT INTO FORNECEDOR (CNPJ, CEP, ESTADO, CIDADE, EMAIL, TELEFONE, NOME, COD_PRODUTO)\n",
"VALUES ('78912345600004', '76543-210', 'RS', 'Porto Alegre', 'fornecedor4@example.com', '4444444444', 'Fornecedor 4', 'P004')''')\n",
"\n",
"cursor.execute('''INSERT INTO FORNECEDOR (CNPJ, CEP, ESTADO, CIDADE, EMAIL, TELEFONE, NOME, COD_PRODUTO)\n",
```

```
"VALUES ('32165498700005', '23456-789', 'SC', 'Florianópolis', 'fornecedor5@example.com', '5555555555', 'Fornecedor 5', 'P005')"')\n",
"\n",
"\n",
"# Criar a tabela \"FORPROD\" se ela não existir\n",
"cursor.execute('''CREATE TABLE IF NOT EXISTS FORPROD(\n",
"  CNPJ VARCHAR(20),\n",
"  COD_PRODUTO VARCHAR(20) PRIMARY KEY,\n",
"  FOREIGN KEY (CNPJ) REFERENCES FORNECEDOR(CNPJ),\n",
"  FOREIGN KEY (COD_PRODUTO) REFERENCES PRODUTO(COD_PRODUTO)\n",
")'''')\n",
"\n",
"# Criar a tabela \"CVPROD\" se ela não existir\n",
"cursor.execute('''CREATE TABLE IF NOT EXISTS CVPROD(\n",
"  COD_COMPRAVENDA VARCHAR(20),\n",
"  COD_PRODUTO VARCHAR(20) PRIMARY KEY,\n",
"  FOREIGN KEY (COD_COMPRAVENDA) REFERENCES COMPRAVENDA(COD_COMPRAVENDA),\n",
"  FOREIGN KEY (COD_PRODUTO) REFERENCES PRODUTO(COD_PRODUTO)\n",
")'''')\n",
"\n",
"# Salvar as alterações\n",
"conn.commit()\n",
"\n",
"# Fechar a conexão\n",
"conn.close()\n",
"\n",
"\n",
"\n"
],
"metadata": {
"id": "1mguSePoYHmE"
},
"execution_count": 2,
"outputs": []
```

```
},
{
 "cell_type": "code",
 "source": [
  "import pandas as pd\n",
  "import sqlite3\n",
  "\n",
  "# Estabelecer conexão com o banco de dados\n",
  "conn = sqlite3.connect('trabalhosve.db')\n",
  "\n",
  "# Consulta à tabela CLIENTE\n",
  "query_cliente = \"SELECT * FROM CLIENTE\"\n",
  "df_cliente = pd.read_sql_query(query_cliente, conn)\n",
  "\n",
  "# Consulta à tabela PRODUTO\n",
  "query_produto = \"SELECT * FROM PRODUTO\"\n",
  "df_produto = pd.read_sql_query(query_produto, conn)\n",
  "\n",
  "# Consulta à tabela LOJA\n",
  "query_loja = \"SELECT * FROM LOJA\"\n",
  "df_loja = pd.read_sql_query(query_loja, conn)\n",
  "\n",
  "# Consulta à tabela COMPRAVENDA\n",
  "query_compravenda = \"SELECT * FROM COMPRAVENDA\"\n",
  "df_compravenda = pd.read_sql_query(query_compravenda, conn)\n",
  "\n",
  "# Consulta à tabela VENDEDOR\n",
  "query_vendedor = \"SELECT * FROM VENDEDOR\"\n",
  "df_vendedor = pd.read_sql_query(query_vendedor, conn)\n",
  "\n",
  "# Consulta à tabela VENDEDOR\n",
  "query_fornecedor = \"SELECT * FROM VENDEDOR\"\n",
  "df_fornecedor = pd.read_sql_query(query_fornecedor, conn)\n",
  "\n",
  "#Consulta a tabela CVPROD\n",
  "query_cvprod = \"SELECT * FROM CVPROD\"\n",
```

```
"df_cvprod = pd.read_sql_query(query_cvprod, conn)\n",
"\n",
"#Consulta a tabela CVPROD\n",
"query_forprod = \"SELECT * FROM FORPROD\"\n",
"df_forprod = pd.read_sql_query(query_forprod, conn)\n",
"\n",
"# Fechar a conexão com o banco de dados\n",
"conn.close()\n",
"\n",
"# Imprimir os resultados\n",
"print(\"Tabela CLIENTE:\")\n",
"print(df_cliente)\n",
"print()\n",
"\n",
"print(\"Tabela PRODUTO:\")\n",
"print(df_produto)\n",
"print()\n",
"\n",
"print(\"Tabela VENDEDOR:\")\n",
"print(df_vendedor)\n",
"print()\n",
"\n",
"print(\"Tabela FORNECEDOR:\")\n",
"print(df_fornecedor)\n",
"print()\n",
"\n",
"\n",
"print(\"Tabela LOJA:\")\n",
"print(df_loja)\n",
"print()\n",
"\n",
"print(\"Tabela COMPRAVENDA:\")\n",
"print(df_compravenda)\n",
"print()\n",
"\n",
"print(\"Tabela VENDEDOR:\")\n",
```

```
      "print(df_vendedor)\n",
      "\n",
      "print(\"Tabela CVPROD:\")\n",
      "print(df_cvprod)\n",
      "\n",
      "print(\"Tabela FORPROD:\")\n",
      "print(df_forprod)\n"
     ],
     "metadata": {
      "colab": {
       "base_uri": "https://localhost:8080/"
      },
      "outputId": "083c66e2-ad8d-4203-fabd-0639501e5bca",
      "id": "Z2OxudhQczkO"
     },
     "execution_count": 2,
     "outputs": [
      {
       "output_type": "stream",
       "name": "stdout",
       "text": [
        "Tabela CLIENTE:\n",
        "          CPF       TELEFONE       ENDERECO      CEP       ESTADO \\\n",
        "0  123.456.789-00  (11) 1234-5678      Rua A, 123  12345-678      São Paulo \n",
        "1  987.654.321-00  (22) 9876-5432  Avenida B, 456  98765-432  Rio de Janeiro \n",
        "2  111.222.333-44  (33) 1111-2222      Rua C, 789  54321-098     Minas Gerais \n",
        "3  555.444.333-22  (44) 5555-4444  Avenida D, 987  76543-210           Bahia \n",
        "4  777.888.999-00  (55) 7777-8888      Rua E, 321  01234-567      São Paulo \n",
        "\n",
        "          CIDADE     LOGRADOURO SEXO PRIMEIRO_NOME SOBRENOME \\\n",
        "0      São Paulo  Residencial ABC    M          João     Silva   \n",
```

```
"1  Rio de Janeiro  Apartamento XYZ    F        Maria     Souza   \n",
"2  Belo Horizonte        Casa 123    M        Pedro    Santos   \n",
"3       Salvador       Bloco ABCD    F         Ana  Oliveira   \n",
"4      São Paulo    Edifício EFGH    M        Lucas  Ferreira   \n",
"\n",
"  DATA_NASCIMENTO  \n",
"0     1990-01-01  \n",
"1     1985-05-10  \n",
"2     1982-12-25  \n",
"3     1995-09-15  \n",
"4     1988-07-20  \n",
"\n",
"Tabela PRODUTO:\n",
"  COD_PRODUTO   CUSTO   PRECO_VENDA                        NOME CATEGORIA  \\\n",
"0       P001  700.0       999.0              iPhone 13  Eletrônicos   \n",
"1       P002  800.0      1099.0            Smart TV 4K  Eletrônicos   \n",
"2       P003   50.0        79.0  Fone de Ouvido Bluetooth  Eletrônicos   \n",
"3       P004  200.0       299.0       Tablet Galaxy Tab A  Eletrônicos   \n",
"4       P005  300.0       449.0          Câmera Digital  Eletrônicos   \n",
"\n",
"    MARCA      COR  \n",
"0   Apple     Preto  \n",
"1  Samsung    Prata  \n",
"2    Sony  Vermelho  \n",
"3  Samsung    Preto  \n",
"4   Canon   Branco  \n",
"\n",
"Tabela VENDEDOR:\n",
"  COD_VENDEDOR   SALARIO  SEXO   COMISSAO                NOME COD_LOJA\n",
"0       VD001   3000.0    M      0.00      João Silva    LJ001\n",
"1       VD002   2500.0    F      0.05    Maria Santos    LJ002\n",
"2       VD003   4000.0    M      0.00  Pedro Oliveira    LJ003\n",
"3       VD004   2800.0    F      0.03       Ana Souza    LJ001\n",
"4       VD005   3500.0    M      0.02   Carlos Mendes    LJ002\n",
```

```
        "\n",
        "Tabela FORNECEDOR:\n",
        "  COD_VENDEDOR  SALARIO  SEXO  COMISSAO          NOME COD_LOJA\n",
        "0       VD001   3000.0    M      0.00      João Silva    LJ001\n",
        "1       VD002   2500.0    F      0.05    Maria Santos    LJ002\n",
        "2       VD003   4000.0    M      0.00  Pedro Oliveira    LJ003\n",
        "3       VD004   2800.0    F      0.03       Ana Souza    LJ001\n",
        "4       VD005   3500.0    M      0.02   Carlos Mendes    LJ002\n",
        "\n",
        "Tabela LOJA:\n",
        "  COD_LOJA        CEP         ESTADO         CIDADE      LOGRADOURO \\\n",
        "0    LJ001  12345-678      São Paulo      São Paulo      Rua A, 123   \n",
        "1    LJ002  98765-432  Rio de Janeiro  Rio de Janeiro  Avenida B, 456   \n",
        "2    LJ003  54321-876   Minas Gerais  Belo Horizonte      Rua C, 789   \n",
        "3    LJ004  76543-210         Bahia       Salvador  Avenida D, 987   \n",
        "4    LJ005  32109-876  Espírito Santo        Vitória      Rua E, 654   \n",
        "\n",
        "         TELEFONE  \n",
        "0  (11) 1234-5678  \n",
        "1  (21) 9876-5432  \n",
        "2  (31) 5432-1876  \n",
        "3  (71) 7654-3210  \n",
        "4  (27) 3210-9876  \n",
        "\n",
        "Tabela COMPRAVENDA:\n",
        "  COD_COMPRAVENDA        DATAS  VALOR_TOTAL  VALOR DESCONTO COD_VENDEDOR  \n",
        "0          CV001  2023-05-17       150.99  129.99      21.0       VD001   \n",
        "1          CV002  2023-05-18        75.50   75.50       0.0       VD002   \n",
        "2          CV003  2023-05-19       200.00  200.00       0.0       VD003   \n",
        "3          CV004  2023-05-20       500.00  400.00     100.0       VD002   \n",
        "4          CV005  2023-05-21      1000.00  950.00      50.0       VD001   \n",
        "\n",
        "      CPF_CLIENTE COD_LOJA COD_PRODUTO  \n",
        "0  123.456.789-00    LJ001        P001  \n",
```

```
        "1  987.654.321-00    LJ002       P002  \n",
        "2  987.654.321-00    LJ001       P003  \n",
        "3  111.222.333-44    LJ002       P004  \n",
        "4  555.444.333-22    LJ003       P005  \n",
        "\n",
        "Tabela VENDEDOR:\n",
        "   COD_VENDEDOR   SALARIO  SEXO    COMISSAO                NOME COD_LOJA\n",
        "0       VD001   3000.0    M     0.00     João Silva    LJ001\n",
        "1       VD002   2500.0    F     0.05    Maria Santos    LJ002\n",
        "2       VD003   4000.0    M     0.00  Pedro Oliveira    LJ003\n",
        "3       VD004   2800.0    F     0.03      Ana Souza    LJ001\n",
        "4       VD005   3500.0    M     0.02   Carlos Mendes    LJ002\n",
        "Tabela CVPROD:\n",
        "Empty DataFrame\n",
        "Columns: [COD_COMPRAVENDA, COD_PRODUTO]\n",
        "Index: []\n",
        "Tabela FORPROD:\n",
        "Empty DataFrame\n",
        "Columns: [CNPJ, COD_PRODUTO]\n",
        "Index: []\n"
       ]
      }
     ]
    },
    {
     "cell_type": "code",
     "source": [
      "df_produtos.dtypes"
     ],
     "metadata": {
      "colab": {
       "base_uri": "https://localhost:8080/"
      },
      "id": "eDMQYegTxsI-",
      "outputId": "444e3927-dc98-4c00-9551-617474eabe81"
```

```
    },
    "execution_count": 6,
    "outputs": [
     {
      "output_type": "execute_result",
      "data": {
       "text/plain": [
        "COD_PRODUTO     object\n",
        "CUSTO          float64\n",
        "PRECO_VENDA    float64\n",
        "NOME            object\n",
        "CATEGORIA       object\n",
        "MARCA           object\n",
        "COR             object\n",
        "dtype: object"
       ]
      },
      "metadata": {},
      "execution_count": 6
     }
    ]
   },
   {
    "cell_type": "code",
    "source": [
     "#Convertendo tipos de dados\n",
     "df_produto['COD_PRODUTO'] = df_produto['COD_PRODUTO'].astype(str)\n",
     "df_produto['CUSTO'] = df_produto['CUSTO'].astype(float)\n",
     "df_produto['PRECO_VENDA'] = df_produto['PRECO_VENDA'].astype(float)\n",
     "df_produto['NOME'] = df_produto['NOME'].astype(str)\n",
     "df_produto['CATEGORIA'] = df_produto['CATEGORIA'].astype(str)\n",
     "df_produto['MARCA'] = df_produto['MARCA'].astype(str)\n",
     "df_produto['COR'] = df_produto['COR'].astype(str)\n"
    ],
    "metadata": {
```

```
    "id": "YgJ3kEp_1gbM"
   },
   "execution_count": 4,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
    "df_cliente.dtypes"
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "id": "xV43tDtDx25R",
    "outputId": "e57e6c43-9c41-456b-e1ba-2b70b14d2063"
   },
   "execution_count": 7,
   "outputs": [
    {
     "output_type": "execute_result",
     "data": {
      "text/plain": [
       "CPF               object\n",
       "TELEFONE          object\n",
       "ENDERECO          object\n",
       "CEP               object\n",
       "ESTADO            object\n",
       "CIDADE            object\n",
       "LOGRADOURO        object\n",
       "SEXO              object\n",
       "PRIMEIRO_NOME     object\n",
       "SOBRENOME         object\n",
       "DATA_NASCIMENTO   object\n",
       "dtype: object"
      ]
```

```
      },
      "metadata": {},
      "execution_count": 7
     }
    ]
   },
   {
    "cell_type": "code",
    "source": [
     "# Remover pontos e traço do CPF para nao dar erro\n",
     "df_cliente['CPF'] = df_cliente['CPF'].str.replace('.', '').str.replace('-', '')"
    ],
    "metadata": {
     "colab": {
      "base_uri": "https://localhost:8080/"
     },
     "id": "aHDuVsRg0oYB",
     "outputId": "126178c9-a436-4fa6-a70e-c9d4cd4737f4"
    },
    "execution_count": 5,
    "outputs": [
     {
      "output_type": "stream",
      "name": "stderr",
      "text": [
       "<ipython-input-5-dbd20c55266e>:2: FutureWarning: The default value of regex will change from True to False in a future version. In addition, single character regular expressions will *not* be treated as literal strings when regex=True.\n",
       "  df_cliente['CPF'] = df_cliente['CPF'].str.replace('.', '').str.replace('-', '')\n"
      ]
     }
    ]
   },
   {
    "cell_type": "code",
    "source": [
```

```
      "# Converter os tipos de dados das colunas\n",
      "df_cliente['CPF'] = df_cliente['CPF'].astype(str)\n",
      "df_cliente['TELEFONE'] = df_cliente['TELEFONE'].astype(str)\n",
      "df_cliente['CEP'] = df_cliente['CEP'].astype(str)\n",
      "df_cliente['ESTADO'] = df_cliente['ESTADO'].astype('category')\n",
      "df_cliente['DATA_NASCIMENTO'] = pd.to_datetime(df_cliente['DATA_NASCIMENTO'], format='%d/%m/%Y', errors='coerce')"
     ],
     "metadata": {
      "id": "W4dIi5qx0pyZ"
     },
     "execution_count": 6,
     "outputs": []
    },
    {
     "cell_type": "code",
     "source": [
      "#verificar numeros faltantes\n",
      "df_cliente.isna().sum()"
     ],
     "metadata": {
      "colab": {
       "base_uri": "https://localhost:8080/"
      },
      "id": "jAYwNptXrmi3",
      "outputId": "fefaf1f6-7ab8-4b56-a691-9f66ace58946"
     },
     "execution_count": 4,
     "outputs": [
      {
       "output_type": "execute_result",
       "data": {
        "text/plain": [
         "CPF             0\n",
         "TELEFONE        0\n",
```

```
        "ENDERECO          0\n",
        "CEP               0\n",
        "ESTADO            0\n",
        "CIDADE            0\n",
        "LOGRADOURO        0\n",
        "SEXO              0\n",
        "PRIMEIRO_NOME     0\n",
        "SOBRENOME         0\n",
        "DATA_NASCIMENTO   0\n",
        "dtype: int64"
       ]
      },
      "metadata": {},
      "execution_count": 4
    }
  ]
 },
 {
  "cell_type": "code",
  "source": [
   "df_vendedor.dtypes"
  ],
  "metadata": {
   "colab": {
    "base_uri": "https://localhost:8080/"
   },
   "id": "ezUooO8zx8IS",
   "outputId": "6d80144f-80f9-48cd-f3dd-fbaf0ba3e366"
  },
  "execution_count": 8,
  "outputs": [
   {
    "output_type": "execute_result",
    "data": {
     "text/plain": [
      "COD_VENDEDOR     object\n",
```

```
        "SALARIO        float64\n",
        "SEXO            object\n",
        "COMISSAO       float64\n",
        "NOME            object\n",
        "COD_LOJA        object\n",
        "dtype: object"
       ]
      },
      "metadata": {},
      "execution_count": 8
    }
  ]
},
{
  "cell_type": "code",
  "source": [
    "df_fornecedor.dtypes"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "RUHihXmhyCYZ",
    "outputId": "0f0617af-c773-4bcd-ad1d-c5d0894dbcd2"
  },
  "execution_count": 9,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "COD_VENDEDOR    object\n",
          "SALARIO        float64\n",
          "SEXO            object\n",
          "COMISSAO       float64\n",
          "NOME            object\n",
```

```
          "COD_LOJA         object\n",
          "dtype: object"
         ]
        },
        "metadata": {},
        "execution_count": 9
       }
      ]
     },
     {
      "cell_type": "code",
      "source": [
       "df_fornecedor['COD_VENDEDOR']                                                    =
df_fornecedor['COD_VENDEDOR'].astype(str)\n",
       "df_fornecedor['SALARIO'] = df_fornecedor['SALARIO'].astype(float)\n",
       "df_fornecedor['SEXO'] = df_fornecedor['SEXO'].astype(str)\n",
       "df_fornecedor['COMISSAO'] = df_fornecedor['COMISSAO'].astype(float)\n",
       "df_fornecedor['NOME'] = df_fornecedor['NOME'].astype(str)\n",
       "df_fornecedor['COD_LOJA'] = df_fornecedor['COD_LOJA'].astype(str)\n"
      ],
      "metadata": {
       "id": "l461wxWJ0STK"
      },
      "execution_count": 7,
      "outputs": []
     },
     {
      "cell_type": "code",
      "source": [
       "#Verificar se tem numeros faltantes\n",
       "df_fornecedor.isna().sum()"
      ],
      "metadata": {
       "colab": {
        "base_uri": "https://localhost:8080/"
       },
```

```
      "id": "9HOWPLJmr9UF",
      "outputId": "2bb52ce9-9d03-4553-a5de-79f0bb7da561"
     },
     "execution_count": 5,
     "outputs": [
      {
       "output_type": "execute_result",
       "data": {
        "text/plain": [
         "COD_VENDEDOR    0\n",
         "SALARIO         0\n",
         "SEXO            0\n",
         "COMISSAO        0\n",
         "NOME            0\n",
         "COD_LOJA        0\n",
         "dtype: int64"
        ]
       },
       "metadata": {},
       "execution_count": 5
      }
     ]
    },
    {
     "cell_type": "code",
     "source": [
      "df_fornecedor['COD_VENDEDOR'] = df_fornecedor['COD_VENDEDOR'].astype(str)\n",
      "df_fornecedor['SALARIO'] = df_fornecedor['SALARIO'].astype(float)\n",
      "df_fornecedor['SEXO'] = df_fornecedor['SEXO'].astype(str)\n",
      "df_fornecedor['COMISSAO'] = df_fornecedor['COMISSAO'].astype(float)\n",
      "df_fornecedor['NOME'] = df_fornecedor['NOME'].astype(str)\n",
      "df_fornecedor['COD_LOJA'] = df_fornecedor['COD_LOJA'].astype(str)\n"
     ],
     "metadata": {
      "id": "aRCaQTgjzviY"
```

```
  },
  "execution_count": null,
  "outputs": []
 },
 {
  "cell_type": "code",
  "source": [
   "df_loja.dtypes"
  ],
  "metadata": {
   "colab": {
    "base_uri": "https://localhost:8080/"
   },
   "id": "cKCfjmzoyMLZ",
   "outputId": "c51f7d08-6e25-47e9-e3ce-945d83b6ab04"
  },
  "execution_count": 10,
  "outputs": [
   {
    "output_type": "execute_result",
    "data": {
     "text/plain": [
      "COD_LOJA      object\n",
      "CEP           object\n",
      "ESTADO        object\n",
      "CIDADE        object\n",
      "LOGRADOURO    object\n",
      "TELEFONE      object\n",
      "dtype: object"
     ]
    },
    "metadata": {},
    "execution_count": 10
   }
  ]
 },
```

```
{
 "cell_type": "code",
 "source": [
  "df_loja['COD_LOJA'] = df_loja['COD_LOJA'].astype(str)\n",
  "df_loja['CEP'] = df_loja['CEP'].astype(str)\n",
  "df_loja['ESTADO'] = df_loja['ESTADO'].astype(str)\n",
  "df_loja['CIDADE'] = df_loja['CIDADE'].astype(str)\n",
  "df_loja['LOGRADOURO'] = df_loja['LOGRADOURO'].astype(str)\n",
  "df_loja['TELEFONE'] = df_loja['TELEFONE'].astype(str)\n"
 ],
 "metadata": {
  "id": "qdQT9oa9zfAz"
 },
 "execution_count": 8,
 "outputs": []
},
{
 "cell_type": "code",
 "source": [
  "df_loja.isna().sum()"
 ],
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/"
  },
  "id": "Emb2NlettB3H",
  "outputId": "e9515022-d8c5-4d83-abdc-0621d42b0ad9"
 },
 "execution_count": 6,
 "outputs": [
  {
   "output_type": "execute_result",
   "data": {
    "text/plain": [
     "COD_LOJA      0\n",
     "CEP           0\n",
```

```
          "ESTADO       0\n",
          "CIDADE       0\n",
          "LOGRADOURO    0\n",
          "TELEFONE      0\n",
          "dtype: int64"
        ]
      },
      "metadata": {},
      "execution_count": 6
    }
  ]
},
{
  "cell_type": "code",
  "source": [
    "df_compravenda.dtypes"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "yL-sp-HeyQSk",
    "outputId": "b30c953b-f117-46ea-ca6c-a24d0431a7a0"
  },
  "execution_count": 11,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "COD_COMPRAVENDA     object\n",
          "DATAS               object\n",
          "VALOR_TOTAL        float64\n",
          "VALOR              float64\n",
          "DESCONTO           float64\n",
          "COD_VENDEDOR        object\n",
```

```
          "CPF_CLIENTE        object\n",
          "COD_LOJA           object\n",
          "COD_PRODUTO        object\n",
          "dtype: object"
         ]
        },
        "metadata": {},
        "execution_count": 11
       }
     ]
    },
    {
     "cell_type": "code",
     "source": [
       "df_compravenda['COD_COMPRAVENDA'] = df_compravenda['COD_COMPRAVENDA'].astype(str)\n",
       "df_compravenda['DATAS'] = df_compravenda['DATAS'].astype(str)\n",
       "df_compravenda['VALOR_TOTAL'] = df_compravenda['VALOR_TOTAL'].astype(float)\n",
       "df_compravenda['VALOR'] = df_compravenda['VALOR'].astype(float)\n",
       "df_compravenda['DESCONTO'] = df_compravenda['DESCONTO'].astype(float)\n",
       "df_compravenda['COD_VENDEDOR'] = df_compravenda['COD_VENDEDOR'].astype(str)\n",
       "df_compravenda['CPF_CLIENTE'] = df_compravenda['CPF_CLIENTE'].astype(str)\n",
       "df_compravenda['COD_LOJA'] = df_compravenda['COD_LOJA'].astype(str)\n",
       "df_compravenda['COD_PRODUTO'] = df_compravenda['COD_PRODUTO'].astype(str)\n"
     ],
     "metadata": {
       "id": "e_yaDEp1zPPw"
     },
     "execution_count": 9,
     "outputs": []
    },
    {
```

```
 "cell_type": "code",
 "source": [
  "df_compravenda.isna().sum()"
 ],
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/"
  },
  "id": "5pmBtIFstG2T",
  "outputId": "bf19ccc9-e177-4d01-c945-b2030c6061c4"
 },
 "execution_count": 8,
 "outputs": [
  {
   "output_type": "execute_result",
   "data": {
    "text/plain": [
     "COD_COMPRAVENDA    0\n",
     "DATAS              0\n",
     "VALOR_TOTAL        0\n",
     "VALOR              0\n",
     "DESCONTO           0\n",
     "COD_VENDEDOR       0\n",
     "CPF_CLIENTE        0\n",
     "COD_LOJA           0\n",
     "COD_PRODUTO        0\n",
     "dtype: int64"
    ]
   },
   "metadata": {},
   "execution_count": 8
  }
 ]
},
{
 "cell_type": "markdown",
```

```
  "source": [
    "**Q001  Listar todas as lojas cadastradas no sistema**"
  ],
  "metadata": {
    "id": "kwzFfDGjBnO0"
  }
 },
 {
  "cell_type": "code",
  "source": [
    "# Listar todas as lojas cadastradas\n",
    "lojas_cadastradas = df_loja['COD_LOJA'].tolist()\n",
    "lojas_cadastradas"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "BGgetTA9v627",
    "outputId": "5a140d0d-3411-4b41-a223-db64578d44eb"
  },
  "execution_count": null,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "['LJ001', 'LJ002', 'LJ003', 'LJ004', 'LJ005']"
        ]
      },
      "metadata": {},
      "execution_count": 4
    }
  ]
 },
 {
```

```
    "cell_type": "markdown",
    "source": [
      "**Q002 Encontrar a loja com o maior número de vendas no último mês**"
    ],
    "metadata": {
      "id": "xlDjJejRDPAe"
    }
   },
   {
    "cell_type": "code",
    "source": [
      "import pandas as pd\n",
      "\n",
      "# Converter a coluna 'DATAS' para o tipo datetime\n",
      "df_compravenda['DATAS'] = pd.to_datetime(df_compravenda['DATAS'])\n",
      "\n",
      "# Filtrar os dados para o último mês\n",
      "ultimo_mes = df_compravenda[df_compravenda['DATAS'] >= df_compravenda['DATAS'].max() - pd.DateOffset(months=1)]\n",
      "\n",
      "# Calcular o número de vendas para cada loja\n",
      "vendas_por_loja = ultimo_mes['COD_LOJA'].value_counts()\n",
      "\n",
      "# Encontrar a loja com o maior número de vendas\n",
      "loja_mais_vendida = vendas_por_loja.idxmax()\n",
      "quantidade_vendas = vendas_por_loja.max()\n",
      "\n",
      "print(\"Loja mais vendida:\", loja_mais_vendida)\n",
      "print(\"Quantidade de vendas:\", quantidade_vendas)\n"
    ],
    "metadata": {
      "colab": {
        "base_uri": "https://localhost:8080/"
      },
      "id": "PgKel2ZuIR30",
      "outputId": "cfff8f76-7f1f-458f-8dab-f0f8b1141a71"
```

```
  },
  "execution_count": null,
  "outputs": [
   {
    "output_type": "stream",
    "name": "stdout",
    "text": [
     "Loja mais vendida: LJ001\n",
     "Quantidade de vendas: 2\n"
    ]
   }
  ]
 },
 {
  "cell_type": "code",
  "source": [
   "df_compravenda"
  ],
  "metadata": {
   "colab": {
    "base_uri": "https://localhost:8080/",
    "height": 250
   },
   "id": "LcflvkALEjos",
   "outputId": "5518237f-9880-40c9-d41c-031885e291e0"
  },
  "execution_count": null,
  "outputs": [
   {
    "output_type": "execute_result",
    "data": {
     "text/plain": [
      "  COD_COMPRAVENDA      DATAS  VALOR_TOTAL  VALOR DESCONTO COD_VENDEDOR  \\\n",
      "0          CV001 2023-05-17       150.99  129.99      21.0       VD001   \n",
      "1          CV002 2023-05-18        75.50   75.50       0.0       VD002   \n",
```

```
 "2          CV003  2023-05-19      200.00  200.00      0.0       VD003   \n",
 "3          CV004  2023-05-20      500.00  400.00     100.0       VD002   \n",
 "4          CV005  2023-05-21     1000.00  950.00      50.0       VD001   \n",
 "\n",
 "      CPF_CLIENTE COD_LOJA COD_PRODUTO  \n",
 "0  123.456.789-00    LJ001       P001  \n",
 "1  987.654.321-00    LJ002       P002  \n",
 "2  987.654.321-00    LJ001       P003  \n",
 "3  111.222.333-44    LJ002       P004  \n",
 "4  555.444.333-22    LJ003       P005  "
],
"text/html": [
 "\n",
 "  <div id=\"df-d4d44dc9-3b23-4ac5-b2e1-88eb28f873b3\">\n",
 "    <div class=\"colab-df-container\">\n",
 "      <div>\n",
 "<style scoped>\n",
 "    .dataframe tbody tr th:only-of-type {\n",
 "        vertical-align: middle;\n",
 "    }\n",
 "\n",
 "    .dataframe tbody tr th {\n",
 "        vertical-align: top;\n",
 "    }\n",
 "\n",
 "    .dataframe thead th {\n",
 "        text-align: right;\n",
 "    }\n",
 "</style>\n",
 "<table border=\"1\" class=\"dataframe\">\n",
 "  <thead>\n",
 "    <tr style=\"text-align: right;\">\n",
 "      <th></th>\n",
 "      <th>COD_COMPRAVENDA</th>\n",
 "      <th>DATAS</th>\n",
 "      <th>VALOR_TOTAL</th>\n",
```

```
"      <th>VALOR</th>\n",
"      <th>DESCONTO</th>\n",
"      <th>COD_VENDEDOR</th>\n",
"      <th>CPF_CLIENTE</th>\n",
"      <th>COD_LOJA</th>\n",
"      <th>COD_PRODUTO</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>CV001</td>\n",
"      <td>2023-05-17</td>\n",
"      <td>150.99</td>\n",
"      <td>129.99</td>\n",
"      <td>21.0</td>\n",
"      <td>VD001</td>\n",
"      <td>123.456.789-00</td>\n",
"      <td>LJ001</td>\n",
"      <td>P001</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>CV002</td>\n",
"      <td>2023-05-18</td>\n",
"      <td>75.50</td>\n",
"      <td>75.50</td>\n",
"      <td>0.0</td>\n",
"      <td>VD002</td>\n",
"      <td>987.654.321-00</td>\n",
"      <td>LJ002</td>\n",
"      <td>P002</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>CV003</td>\n",
```

```
"      <td>2023-05-19</td>\n",
"      <td>200.00</td>\n",
"      <td>200.00</td>\n",
"      <td>0.0</td>\n",
"      <td>VD003</td>\n",
"      <td>987.654.321-00</td>\n",
"      <td>LJ001</td>\n",
"      <td>P003</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>CV004</td>\n",
"      <td>2023-05-20</td>\n",
"      <td>500.00</td>\n",
"      <td>400.00</td>\n",
"      <td>100.0</td>\n",
"      <td>VD002</td>\n",
"      <td>111.222.333-44</td>\n",
"      <td>LJ002</td>\n",
"      <td>P004</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>4</th>\n",
"      <td>CV005</td>\n",
"      <td>2023-05-21</td>\n",
"      <td>1000.00</td>\n",
"      <td>950.00</td>\n",
"      <td>50.0</td>\n",
"      <td>VD001</td>\n",
"      <td>555.444.333-22</td>\n",
"      <td>LJ003</td>\n",
"      <td>P005</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
```

```
"          <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-d4d44dc9-3b23-4ac5-b2e1-88eb28f873b3')\"\n",
"            title=\"Convert this dataframe to an interactive table.\"\n",
"            style=\"display:none;\">\n",
"       \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",
"       width=\"24px\">\n",
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"      <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
"  </svg>\n",
"      </button>\n",
"      \n",
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
"\n",
"    .colab-df-convert:hover {\n",
```

```
"      background-color: #E2EBFA;\n",
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
"    }\n",
"  </style>\n",
"\n",
"      <script>\n",
"        const buttonEl =\n",
"          document.querySelector('#df-d4d44dc9-3b23-4ac5-b2e1-88eb28f873b3 button.colab-df-convert');\n",
"        buttonEl.style.display =\n",
"          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
"\n",
"        async function convertToInteractive(key) {\n",
"        const element = document.querySelector('#df-d4d44dc9-3b23-4ac5-b2e1-88eb28f873b3');\n",
"          const dataTable =\n",
"            await google.colab.kernel.invokeFunction('convertToInteractive',\n",
"                                                     [key], {});\n",
"          if (!dataTable) return;\n",
"\n",
"          const docLinkHtml = 'Like what you see? Visit the ' +\n",
```

```
          "                                                    '<a       target=\"_blank\"
href=https://colab.research.google.com/notebooks/data_table.ipynb>data                table
notebook</a>'\n",
          "           + ' to learn more about interactive tables.';\n",
          "         element.innerHTML = '';\n",
          "         dataTable['output_type'] = 'display_data';\n",
          "         await google.colab.output.renderOutput(dataTable, element);\n",
          "         const docLink = document.createElement('div');\n",
          "         docLink.innerHTML = docLinkHtml;\n",
          "         element.appendChild(docLink);\n",
          "       }\n",
          "     </script>\n",
          "   </div>\n",
          " </div>\n",
          " "
        ]
       },
       "metadata": {},
       "execution_count": 6
     }
    ]
   },
   {
    "cell_type": "markdown",
    "source": [
     "**Q003 Listar todos os vendedores cadastrados no sistema**"
    ],
    "metadata": {
     "id": "iVbUBtSBLHRf"
    }
   },
   {
    "cell_type": "code",
    "source": [
     "vendedores_cadastrados = df_vendedor['NOME'].head()\n",
     "vendedores_cadastrados"
```

```
 ],
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/"
  },
  "id": "xDl7FWWDLRg2",
  "outputId": "690c643e-9ef4-4709-e556-6f51bad846a3"
 },
 "execution_count": null,
 "outputs": [
  {
   "output_type": "execute_result",
   "data": {
    "text/plain": [
     "0       João Silva\n",
     "1      Maria Santos\n",
     "2    Pedro Oliveira\n",
     "3         Ana Souza\n",
     "4     Carlos Mendes\n",
     "Name: NOME, dtype: object"
    ]
   },
   "metadata": {},
   "execution_count": 16
  }
 ]
},
{
 "cell_type": "markdown",
 "source": [
  "**Q004 Encontrar o vendedor que mais vendeu no último mês**"
 ],
 "metadata": {
  "id": "S3GcM-XXLjsY"
 }
},
```

```
{
 "cell_type": "code",
 "source": [
  "maisvend = df_vendedor['COD_VENDEDOR'][0]\n",
  "maisvend"
 ],
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/",
   "height": 35
  },
  "id": "aI_BNOgvLipR",
  "outputId": "c68f2bd4-2a65-4f1b-c1cc-79f838b8146b"
 },
 "execution_count": null,
 "outputs": [
  {
   "output_type": "execute_result",
   "data": {
    "text/plain": [
     "'VD001'"
    ],
    "application/vnd.google.colaboratory.intrinsic+json": {
     "type": "string"
    }
   },
   "metadata": {},
   "execution_count": 22
  }
 ]
},
{
 "cell_type": "code",
 "source": [
  "df_vendedor"
 ],
```

```
  "metadata": {
   "colab": {
    "base_uri": "https://localhost:8080/",
    "height": 206
   },
   "id": "XbW4hef5L4Q7",
   "outputId": "f1c4aee6-d81e-41ed-fe37-98d0aebd05d1"
  },
  "execution_count": null,
  "outputs": [
   {
    "output_type": "execute_result",
    "data": {
     "text/plain": [
      "   COD_VENDEDOR   SALARIO  SEXO   COMISSAO                NOME COD_LOJA\n",
      "0       VD001   3000.0    M      0.00      João Silva    LJ001\n",
      "1       VD002   2500.0    F      0.05    Maria Santos    LJ002\n",
      "2       VD003   4000.0    M      0.00  Pedro Oliveira    LJ003\n",
      "3       VD004   2800.0    F      0.03       Ana Souza    LJ001\n",
      "4       VD005   3500.0    M      0.02   Carlos Mendes    LJ002"
     ],
     "text/html": [
      "\n",
      "  <div id=\"df-250cf88b-064c-4e2f-83d7-5dd043096049\">\n",
      "    <div class=\"colab-df-container\">\n",
      "      <div>\n",
      "<style scoped>\n",
      "    .dataframe tbody tr th:only-of-type {\n",
      "        vertical-align: middle;\n",
      "    }\n",
      "\n",
      "    .dataframe tbody tr th {\n",
      "        vertical-align: top;\n",
      "    }\n",
      "\n",
```

```
"    .dataframe thead th {\n",
"        text-align: right;\n",
"    }\n",
"</style>\n",
"<table border=\"1\" class=\"dataframe\">\n",
"  <thead>\n",
"    <tr style=\"text-align: right;\">\n",
"      <th></th>\n",
"      <th>COD_VENDEDOR</th>\n",
"      <th>SALARIO</th>\n",
"      <th>SEXO</th>\n",
"      <th>COMISSAO</th>\n",
"      <th>NOME</th>\n",
"      <th>COD_LOJA</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>VD001</td>\n",
"      <td>3000.0</td>\n",
"      <td>M</td>\n",
"      <td>0.00</td>\n",
"      <td>João Silva</td>\n",
"      <td>LJ001</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>VD002</td>\n",
"      <td>2500.0</td>\n",
"      <td>F</td>\n",
"      <td>0.05</td>\n",
"      <td>Maria Santos</td>\n",
"      <td>LJ002</td>\n",
"    </tr>\n",
"    <tr>\n",
```

```
"      <th>2</th>\n",
"      <td>VD003</td>\n",
"      <td>4000.0</td>\n",
"      <td>M</td>\n",
"      <td>0.00</td>\n",
"      <td>Pedro Oliveira</td>\n",
"      <td>LJ003</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>VD004</td>\n",
"      <td>2800.0</td>\n",
"      <td>F</td>\n",
"      <td>0.03</td>\n",
"      <td>Ana Souza</td>\n",
"      <td>LJ001</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>4</th>\n",
"      <td>VD005</td>\n",
"      <td>3500.0</td>\n",
"      <td>M</td>\n",
"      <td>0.02</td>\n",
"      <td>Carlos Mendes</td>\n",
"      <td>LJ002</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
"          <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-250cf88b-064c-4e2f-83d7-5dd043096049')\"\n",
"            title=\"Convert this dataframe to an interactive table.\"\n",
"            style=\"display:none;\">\n",
"        \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",
```

```
"       width=\"24px\">\n",
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"       <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
"  </svg>\n",
"      </button>\n",
"      \n",
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
"\n",
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
"    }\n",
"\n",
```

```
"    [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
"    }\n",
"  </style>\n",
"\n",
"      <script>\n",
"        const buttonEl =\n",
"          document.querySelector('#df-250cf88b-064c-4e2f-83d7-5dd043096049 button.colab-df-convert');\n",
"        buttonEl.style.display =\n",
"          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
"\n",
"        async function convertToInteractive(key) {\n",
"         const element = document.querySelector('#df-250cf88b-064c-4e2f-83d7-5dd043096049');\n",
"          const dataTable =\n",
"            await google.colab.kernel.invokeFunction('convertToInteractive',\n",
"                                                     [key], {});\n",
"          if (!dataTable) return;\n",
"\n",
"          const docLinkHtml = 'Like what you see? Visit the ' +\n",
"                              '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebook</a>'\n",
"            + ' to learn more about interactive tables.';\n",
"          element.innerHTML = '';\n",
"          dataTable['output_type'] = 'display_data';\n",
"          await google.colab.output.renderOutput(dataTable, element);\n",
"          const docLink = document.createElement('div');\n",
```

```
          "          docLink.innerHTML = docLinkHtml;\n",
          "          element.appendChild(docLink);\n",
          "        }\n",
          "      </script>\n",
          "    </div>\n",
          "  </div>\n",
          "  "
         ]
        },
        "metadata": {},
        "execution_count": 20
       }
      ]
     },
     {
      "cell_type": "markdown",
      "source": [
       "**Q005 Listar todas as vendas realizadas no sistema**"
      ],
      "metadata": {
       "id": "8isinGBwMMvv"
      }
     },
     {
      "cell_type": "code",
      "source": [
       "vendasreal = df_compravenda[['VALOR', 'COD_VENDEDOR','VALOR_TOTAL', 'DESCONTO']]\n",
       "vendasreal.head()\n"
      ],
      "metadata": {
       "colab": {
        "base_uri": "https://localhost:8080/",
        "height": 206
       },
       "id": "OSg4MeKpMTsb",
```

```
  "outputId": "497190fd-7d4a-4303-c314-30f36644f0b6"
},
"execution_count": null,
"outputs": [
 {
  "output_type": "execute_result",
  "data": {
   "text/plain": [
    "    VALOR COD_VENDEDOR  VALOR_TOTAL  DESCONTO\n",
    "0  129.99        VD001       150.99      21.0\n",
    "1   75.50        VD002        75.50       0.0\n",
    "2  200.00        VD003       200.00       0.0\n",
    "3  400.00        VD002       500.00     100.0\n",
    "4  950.00        VD001      1000.00      50.0"
   ],
   "text/html": [
    "\n",
    "  <div id=\"df-37216ef0-eaff-4c31-85ab-eba9fe569823\">\n",
    "    <div class=\"colab-df-container\">\n",
    "      <div>\n",
    "<style scoped>\n",
    "    .dataframe tbody tr th:only-of-type {\n",
    "        vertical-align: middle;\n",
    "    }\n",
    "\n",
    "    .dataframe tbody tr th {\n",
    "        vertical-align: top;\n",
    "    }\n",
    "\n",
    "    .dataframe thead th {\n",
    "        text-align: right;\n",
    "    }\n",
    "</style>\n",
    "<table border=\"1\" class=\"dataframe\">\n",
    "  <thead>\n",
    "    <tr style=\"text-align: right;\">\n",
```

```
"     <th></th>\n",
"     <th>VALOR</th>\n",
"     <th>COD_VENDEDOR</th>\n",
"     <th>VALOR_TOTAL</th>\n",
"     <th>DESCONTO</th>\n",
"   </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"   <tr>\n",
"     <th>0</th>\n",
"     <td>129.99</td>\n",
"     <td>VD001</td>\n",
"     <td>150.99</td>\n",
"     <td>21.0</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>1</th>\n",
"     <td>75.50</td>\n",
"     <td>VD002</td>\n",
"     <td>75.50</td>\n",
"     <td>0.0</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>2</th>\n",
"     <td>200.00</td>\n",
"     <td>VD003</td>\n",
"     <td>200.00</td>\n",
"     <td>0.0</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>3</th>\n",
"     <td>400.00</td>\n",
"     <td>VD002</td>\n",
"     <td>500.00</td>\n",
"     <td>100.0</td>\n",
"   </tr>\n",
```

```
"    <tr>\n",
"      <th>4</th>\n",
"      <td>950.00</td>\n",
"      <td>VD001</td>\n",
"      <td>1000.00</td>\n",
"      <td>50.0</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
"          <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-37216ef0-eaff-4c31-85ab-eba9fe569823')\"\n",
"            title=\"Convert this dataframe to an interactive table.\"\n",
"            style=\"display:none;\">\n",
"       \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",
"       width=\"24px\">\n",
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"      <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
"  </svg>\n",
"      </button>\n",
"      \n",
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
```

```
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
"\n",
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
"    }\n",
"  </style>\n",
"\n",
"      <script>\n",
"        const buttonEl =\n",
"            document.querySelector('#df-37216ef0-eaff-4c31-85ab-eba9fe569823 button.colab-df-convert');\n",
"        buttonEl.style.display =\n",
"          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
"\n",
```

```
          "        async function convertToInteractive(key) {\n",
          "          const element = document.querySelector('#df-37216ef0-eaff-4c31-85ab-eba9fe569823');\n",
          "          const dataTable =\n",
          "            await google.colab.kernel.invokeFunction('convertToInteractive',\n",
          "                                                     [key], {});\n",
          "          if (!dataTable) return;\n",
          "\n",
          "          const docLinkHtml = 'Like what you see? Visit the ' +\n",
          "                              '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebook</a>'\n",
          "            + ' to learn more about interactive tables.';\n",
          "          element.innerHTML = '';\n",
          "          dataTable['output_type'] = 'display_data';\n",
          "          await google.colab.output.renderOutput(dataTable, element);\n",
          "          const docLink = document.createElement('div');\n",
          "          docLink.innerHTML = docLinkHtml;\n",
          "          element.appendChild(docLink);\n",
          "        }\n",
          "      </script>\n",
          "    </div>\n",
          "  </div>\n",
          "  "
        ]
       },
       "metadata": {},
       "execution_count": 27
      }
     ]
    },
    {
     "cell_type": "markdown",
     "source": [
      "\n",
      "\n",
      "**Q006 Encontrar as vendas realizadas entre duas datas específicas**\n"
```

```
    ],
    "metadata": {
     "id": "C_7Cuz2HNYXS"
    }
   },
   {
    "cell_type": "code",
    "source": [
     "df_vendas = df_compravenda.loc[(df_compravenda['DATAS'] >= '2023-05-18') &
(df_compravenda['DATAS'] <= '2023-05-20')]"
    ],
    "metadata": {
     "id": "c_xgUgDxNay-"
    },
    "execution_count": null,
    "outputs": []
   },
   {
    "cell_type": "code",
    "source": [
     "df_vendas"
    ],
    "metadata": {
     "colab": {
      "base_uri": "https://localhost:8080/",
      "height": 187
     },
     "id": "F6mJYUyuNpZA",
     "outputId": "dd2a2822-b802-4644-abf8-7ab586daf0bd"
    },
    "execution_count": null,
    "outputs": [
     {
      "output_type": "execute_result",
      "data": {
       "text/plain": [
```

```
"    COD_COMPRAVENDA           DATAS    VALOR_TOTAL    VALOR DESCONTO COD_VENDEDOR  \\\n",
"1          CV002 2023-05-18        75.5   75.5       0.0        VD002   \n",
"2          CV003 2023-05-19       200.0  200.0       0.0        VD003   \n",
"3          CV004 2023-05-20       500.0  400.0     100.0        VD002   \n",
"\n",
"     CPF_CLIENTE COD_LOJA COD_PRODUTO  \n",
"1  987.654.321-00    LJ002       P002  \n",
"2  987.654.321-00    LJ001       P003  \n",
"3  111.222.333-44    LJ002       P004  "
],
"text/html": [
"\n",
"  <div id=\"df-c6873a65-9e31-4b76-87fa-562814367b8c\">\n",
"    <div class=\"colab-df-container\">\n",
"      <div>\n",
"<style scoped>\n",
"    .dataframe tbody tr th:only-of-type {\n",
"        vertical-align: middle;\n",
"    }\n",
"\n",
"    .dataframe tbody tr th {\n",
"        vertical-align: top;\n",
"    }\n",
"\n",
"    .dataframe thead th {\n",
"        text-align: right;\n",
"    }\n",
"</style>\n",
"<table border=\"1\" class=\"dataframe\">\n",
"  <thead>\n",
"    <tr style=\"text-align: right;\">\n",
"      <th></th>\n",
"      <th>COD_COMPRAVENDA</th>\n",
"      <th>DATAS</th>\n",
"      <th>VALOR_TOTAL</th>\n",
```

```
"     <th>VALOR</th>\n",
"     <th>DESCONTO</th>\n",
"     <th>COD_VENDEDOR</th>\n",
"     <th>CPF_CLIENTE</th>\n",
"     <th>COD_LOJA</th>\n",
"     <th>COD_PRODUTO</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"     <th>1</th>\n",
"     <td>CV002</td>\n",
"     <td>2023-05-18</td>\n",
"     <td>75.5</td>\n",
"     <td>75.5</td>\n",
"     <td>0.0</td>\n",
"     <td>VD002</td>\n",
"     <td>987.654.321-00</td>\n",
"     <td>LJ002</td>\n",
"     <td>P002</td>\n",
"    </tr>\n",
"    <tr>\n",
"     <th>2</th>\n",
"     <td>CV003</td>\n",
"     <td>2023-05-19</td>\n",
"     <td>200.0</td>\n",
"     <td>200.0</td>\n",
"     <td>0.0</td>\n",
"     <td>VD003</td>\n",
"     <td>987.654.321-00</td>\n",
"     <td>LJ001</td>\n",
"     <td>P003</td>\n",
"    </tr>\n",
"    <tr>\n",
"     <th>3</th>\n",
"     <td>CV004</td>\n",
```

```
"      <td>2023-05-20</td>\n",
"      <td>500.0</td>\n",
"      <td>400.0</td>\n",
"      <td>100.0</td>\n",
"      <td>VD002</td>\n",
"      <td>111.222.333-44</td>\n",
"      <td>LJ002</td>\n",
"      <td>P004</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
"          <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-c6873a65-9e31-4b76-87fa-562814367b8c')\"\n",
"            title=\"Convert this dataframe to an interactive table.\"\n",
"            style=\"display:none;\">\n",
"       \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",
"      width=\"24px\">\n",
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"      <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
"  </svg>\n",
"      </button>\n",
"      \n",
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
```

```
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
"\n",
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
"    }\n",
"  </style>\n",
"\n",
"      <script>\n",
"        const buttonEl =\n",
"          document.querySelector('#df-c6873a65-9e31-4b76-87fa-562814367b8c button.colab-df-convert');\n",
"        buttonEl.style.display =\n",
```

```
          "          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
          "\n",
          "        async function convertToInteractive(key) {\n",
          "         const element = document.querySelector('#df-c6873a65-9e31-4b76-87fa-562814367b8c');\n",
          "          const dataTable =\n",
          "            await google.colab.kernel.invokeFunction('convertToInteractive',\n",
          "                                                     [key], {});\n",
          "          if (!dataTable) return;\n",
          "\n",
          "          const docLinkHtml = 'Like what you see? Visit the ' +\n",
          "                                                    '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebook</a>'\n",
          "            + ' to learn more about interactive tables.';\n",
          "          element.innerHTML = '';\n",
          "          dataTable['output_type'] = 'display_data';\n",
          "          await google.colab.output.renderOutput(dataTable, element);\n",
          "          const docLink = document.createElement('div');\n",
          "          docLink.innerHTML = docLinkHtml;\n",
          "          element.appendChild(docLink);\n",
          "        }\n",
          "      </script>\n",
          "    </div>\n",
          "  </div>\n",
          "  "
        ]
      },
      "metadata": {},
      "execution_count": 31
    }
  ]
},
{
  "cell_type": "code",
  "source": [
    "df_produto"
```

```
    ],
    "metadata": {
     "colab": {
      "base_uri": "https://localhost:8080/",
      "height": 206
     },
     "id": "EC1cl9zBpOOp",
     "outputId": "e50db3f0-42bb-438b-fe97-5661c759663d"
    },
    "execution_count": null,
    "outputs": [
     {
      "output_type": "execute_result",
      "data": {
       "text/plain": [
        "   COD_PRODUTO   CUSTO   PRECO_VENDA                          NOME CATEGORIA  \\\n",
        "0       P001  700.0       999.0              iPhone 13  Eletrônicos   \n",
        "1       P002  800.0      1099.0            Smart TV 4K  Eletrônicos   \n",
        "2       P003   50.0        79.0  Fone de Ouvido Bluetooth  Eletrônicos   \n",
        "3       P004  200.0       299.0      Tablet Galaxy Tab A  Eletrônicos   \n",
        "4       P005  300.0       449.0         Câmera Digital  Eletrônicos   \n",
        "\n",
        "     MARCA       COR  \n",
        "0    Apple     Preto  \n",
        "1  Samsung     Prata  \n",
        "2     Sony  Vermelho  \n",
        "3  Samsung     Preto  \n",
        "4    Canon    Branco  "
       ],
       "text/html": [
        "\n",
        "  <div id=\"df-77332f07-1bae-4cfe-b6b1-a173a0ddebe7\">\n",
        "    <div class=\"colab-df-container\">\n",
        "      <div>\n",
        "<style scoped>\n",
```

```
"    .dataframe tbody tr th:only-of-type {\n",
"        vertical-align: middle;\n",
"    }\n",
"\n",
"    .dataframe tbody tr th {\n",
"        vertical-align: top;\n",
"    }\n",
"\n",
"    .dataframe thead th {\n",
"        text-align: right;\n",
"    }\n",
"</style>\n",
"<table border=\"1\" class=\"dataframe\">\n",
"  <thead>\n",
"    <tr style=\"text-align: right;\">\n",
"      <th></th>\n",
"      <th>COD_PRODUTO</th>\n",
"      <th>CUSTO</th>\n",
"      <th>PRECO_VENDA</th>\n",
"      <th>NOME</th>\n",
"      <th>CATEGORIA</th>\n",
"      <th>MARCA</th>\n",
"      <th>COR</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>P001</td>\n",
"      <td>700.0</td>\n",
"      <td>999.0</td>\n",
"      <td>iPhone 13</td>\n",
"      <td>Eletrônicos</td>\n",
"      <td>Apple</td>\n",
"      <td>Preto</td>\n",
"    </tr>\n",
```

```
"   <tr>\n",
"     <th>1</th>\n",
"     <td>P002</td>\n",
"     <td>800.0</td>\n",
"     <td>1099.0</td>\n",
"     <td>Smart TV 4K</td>\n",
"     <td>Eletrônicos</td>\n",
"     <td>Samsung</td>\n",
"     <td>Prata</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>2</th>\n",
"     <td>P003</td>\n",
"     <td>50.0</td>\n",
"     <td>79.0</td>\n",
"     <td>Fone de Ouvido Bluetooth</td>\n",
"     <td>Eletrônicos</td>\n",
"     <td>Sony</td>\n",
"     <td>Vermelho</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>3</th>\n",
"     <td>P004</td>\n",
"     <td>200.0</td>\n",
"     <td>299.0</td>\n",
"     <td>Tablet Galaxy Tab A</td>\n",
"     <td>Eletrônicos</td>\n",
"     <td>Samsung</td>\n",
"     <td>Preto</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>4</th>\n",
"     <td>P005</td>\n",
"     <td>300.0</td>\n",
"     <td>449.0</td>\n",
"     <td>Câmera Digital</td>\n",
```

```
"      <td>Eletrônicos</td>\n",
"      <td>Canon</td>\n",
"      <td>Branco</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
"          <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-77332f07-1bae-4cfe-b6b1-a173a0ddebe7')\"\n",
"            title=\"Convert this dataframe to an interactive table.\"\n",
"            style=\"display:none;\">\n",
"       \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",
"       width=\"24px\">\n",
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"       <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
"  </svg>\n",
"      </button>\n",
"      \n",
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
```

```
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
"\n",
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
"    }\n",
"  </style>\n",
"\n",
"      <script>\n",
"        const buttonEl =\n",
"          document.querySelector('#df-77332f07-1bae-4cfe-b6b1-a173a0ddebe7 button.colab-df-convert');\n",
"        buttonEl.style.display =\n",
"          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
"\n",
"        async function convertToInteractive(key) {\n",
"          const element = document.querySelector('#df-77332f07-1bae-4cfe-b6b1-a173a0ddebe7');\n",
"          const dataTable =\n",
```

```
          "           await google.colab.kernel.invokeFunction('convertToInteractive',\n",
          "                                                    [key], {});\n",
          "         if (!dataTable) return;\n",
          "\n",
          "         const docLinkHtml = 'Like what you see? Visit the ' +\n",
          "                                                    '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebook</a>'\n",
          "           + ' to learn more about interactive tables.';\n",
          "         element.innerHTML = '';\n",
          "         dataTable['output_type'] = 'display_data';\n",
          "         await google.colab.output.renderOutput(dataTable, element);\n",
          "         const docLink = document.createElement('div');\n",
          "         docLink.innerHTML = docLinkHtml;\n",
          "         element.appendChild(docLink);\n",
          "       }\n",
          "     </script>\n",
          "   </div>\n",
          " </div>\n",
          " "
        ]
       },
       "metadata": {},
       "execution_count": 5
      }
     ]
    },
    {
     "cell_type": "markdown",
     "source": [
      "**Q007 NOVA: Inserir um novo produto na tabela proutos**"
     ],
     "metadata": {
      "id": "dD1PHCzwjW95"
     }
    },
```

```
{
 "cell_type": "code",
 "source": [
  "#Testando outra maneira\n",
  "# Criar um dataframe vazio para armazenar os produtos\n",
  "df_produtos = pd.DataFrame(columns=['COD_PRODUTO', 'CUSTO', 'PRECO_VENDA', 'NOME', 'CATEGORIA', 'MARCA', 'COR'])\n",
  "\n",
  "# Solicitar as informações do novo produto ao usuário\n",
  "cod_produto = input(\"Informe o código do produto: \")\n",
  "custo = float(input(\"Informe o custo do produto: \"))\n",
  "preco_venda = float(input(\"Informe o preço de venda do produto: \"))\n",
  "nome = input(\"Informe o nome do produto: \")\n",
  "categoria = input(\"Informe a categoria do produto: \")\n",
  "marca = input(\"Informe a marca do produto: \")\n",
  "cor = input(\"Informe a cor do produto: \")\n",
  "\n",
  "# Adicionar o novo produto ao dataframe 'produtos'\n",
  "df_produtos.loc[len(df_produtos)] = [cod_produto, custo, preco_venda, nome, categoria, marca, cor]\n",
  "\n",
  "# Imprimir o dataframe 'produtos' atualizado\n",
  "df_produtos.head()\n"
 ],
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/",
   "height": 202
  },
  "id": "IwW2_1MVwk6D",
  "outputId": "76c933af-2cf0-494b-e09a-21e5aa384ef5"
 },
 "execution_count": 3,
 "outputs": [
  {
   "name": "stdout",
```

```
  "output_type": "stream",
  "text": [
   "Informe o código do produto: P010\n",
   "Informe o custo do produto: 900.00\n",
   "Informe o preço de venda do produto: 1200.00\n",
   "Informe o nome do produto: iPhone 14'\n",
   "Informe a categoria do produto: Eletrônicos\n",
   "Informe a marca do produto: Apple\n",
   "Informe a cor do produto: Cinza\n"
  ]
 },
 {
  "output_type": "execute_result",
  "data": {
   "text/plain": [
    "  COD_PRODUTO  CUSTO  PRECO_VENDA       NOME  CATEGORIA  MARCA   COR\n",
    "0       P010  900.0       1200.0  iPhone 14'  Eletrônicos  Apple  Cinza"
   ],
   "text/html": [
    "\n",
    "  <div id=\"df-f8230a81-5144-435a-9941-6a35535bc622\">\n",
    "    <div class=\"colab-df-container\">\n",
    "      <div>\n",
    "<style scoped>\n",
    "    .dataframe tbody tr th:only-of-type {\n",
    "        vertical-align: middle;\n",
    "    }\n",
    "\n",
    "    .dataframe tbody tr th {\n",
    "        vertical-align: top;\n",
    "    }\n",
    "\n",
    "    .dataframe thead th {\n",
    "        text-align: right;\n",
    "    }\n",
```

```
"</style>\n",
"<table border=\"1\" class=\"dataframe\">\n",
"  <thead>\n",
"    <tr style=\"text-align: right;\">\n",
"      <th></th>\n",
"      <th>COD_PRODUTO</th>\n",
"      <th>CUSTO</th>\n",
"      <th>PRECO_VENDA</th>\n",
"      <th>NOME</th>\n",
"      <th>CATEGORIA</th>\n",
"      <th>MARCA</th>\n",
"      <th>COR</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>P010</td>\n",
"      <td>900.0</td>\n",
"      <td>1200.0</td>\n",
"      <td>iPhone 14'</td>\n",
"      <td>Eletrônicos</td>\n",
"      <td>Apple</td>\n",
"      <td>Cinza</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
"          <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-
f8230a81-5144-435a-9941-6a35535bc622')\"\n",
"            title=\"Convert this dataframe to an interactive table.\"\n",
"            style=\"display:none;\">\n",
"       \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0
24 24\"\n",
"      width=\"24px\">\n",
```

```
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"      <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
"  </svg>\n",
"    </button>\n",
"    \n",
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
"\n",
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert {\n",
```

```
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
"    }\n",
"  </style>\n",
"\n",
"      <script>\n",
"        const buttonEl =\n",
"          document.querySelector('#df-f8230a81-5144-435a-9941-6a35535bc622 button.colab-df-convert');\n",
"        buttonEl.style.display =\n",
"          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
"\n",
"        async function convertToInteractive(key) {\n",
"          const element = document.querySelector('#df-f8230a81-5144-435a-9941-6a35535bc622');\n",
"          const dataTable =\n",
"            await google.colab.kernel.invokeFunction('convertToInteractive',\n",
"                                                     [key], {});\n",
"          if (!dataTable) return;\n",
"\n",
"          const docLinkHtml = 'Like what you see? Visit the ' +\n",
"                              '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebook</a>'\n",
"            + ' to learn more about interactive tables.';\n",
"          element.innerHTML = '';\n",
"          dataTable['output_type'] = 'display_data';\n",
"          await google.colab.output.renderOutput(dataTable, element);\n",
"          const docLink = document.createElement('div');\n",
"          docLink.innerHTML = docLinkHtml;\n",
```

```
          "          element.appendChild(docLink);\n",
          "        }\n",
          "      </script>\n",
          "    </div>\n",
          "  </div>\n",
          "  "
         ]
        },
        "metadata": {},
        "execution_count": 3
       }
      ]
     },
     {
      "cell_type": "markdown",
      "source": [
       "insira esses itens: \n",
       "P010 \t\n",
       "900.0 \n",
       "1200.0 \t\n",
       "iPhone 14 \n",
       "Eletrônicos \n",
       "Apple \t\n",
       "Cinza"
      ],
      "metadata": {
       "id": "urZxEJ8skApO"
      }
     },
     {
      "cell_type": "code",
      "source": [
       "# Realizar o merge das tabelas depois de inserir as informações, para guardar as informações juntas\n",
       "df_merged = pd.concat([df_produto, df_produtos], ignore_index=True)"
      ],
```

```
   "metadata": {
    "id": "O7iSBmJ5wNa2"
   },
   "execution_count": 4,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
    "df_merged "
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/",
     "height": 238
    },
    "id": "pzZmx3OixYk4",
    "outputId": "66150343-d13b-4149-ae1b-5cb2f95588fa"
   },
   "execution_count": 5,
   "outputs": [
    {
     "output_type": "execute_result",
     "data": {
      "text/plain": [
       "   COD_PRODUTO   CUSTO   PRECO_VENDA                          NOME
CATEGORIA  \\\n",
       "0       P001  700.0       999.0              iPhone 13  Eletrônicos   \n",
       "1       P002  800.0      1099.0            Smart TV 4K  Eletrônicos   \n",
       "2       P003   50.0        79.0  Fone de Ouvido Bluetooth  Eletrônicos   \n",
       "3       P004  200.0       299.0      Tablet Galaxy Tab A  Eletrônicos   \n",
       "4       P005  300.0       449.0          Câmera Digital  Eletrônicos   \n",
       "5       P010  900.0      1200.0             iPhone 14'  Eletrônicos   \n",
       "\n",
       "    MARCA       COR  \n",
       "0    Apple     Preto  \n",
```

```
  "1  Samsung     Prata  \n",
  "2     Sony  Vermelho  \n",
  "3  Samsung     Preto  \n",
  "4    Canon    Branco  \n",
  "5    Apple     Cinza  "
 ],
 "text/html": [
  "\n",
  "  <div id=\"df-4e325fcb-f132-44c0-977c-77f3c82573c6\">\n",
  "    <div class=\"colab-df-container\">\n",
  "      <div>\n",
  "<style scoped>\n",
  "    .dataframe tbody tr th:only-of-type {\n",
  "        vertical-align: middle;\n",
  "    }\n",
  "\n",
  "    .dataframe tbody tr th {\n",
  "        vertical-align: top;\n",
  "    }\n",
  "\n",
  "    .dataframe thead th {\n",
  "        text-align: right;\n",
  "    }\n",
  "</style>\n",
  "<table border=\"1\" class=\"dataframe\">\n",
  "  <thead>\n",
  "    <tr style=\"text-align: right;\">\n",
  "      <th></th>\n",
  "      <th>COD_PRODUTO</th>\n",
  "      <th>CUSTO</th>\n",
  "      <th>PRECO_VENDA</th>\n",
  "      <th>NOME</th>\n",
  "      <th>CATEGORIA</th>\n",
  "      <th>MARCA</th>\n",
  "      <th>COR</th>\n",
  "    </tr>\n",
```

```
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>P001</td>\n",
"      <td>700.0</td>\n",
"      <td>999.0</td>\n",
"      <td>iPhone 13</td>\n",
"      <td>Eletrônicos</td>\n",
"      <td>Apple</td>\n",
"      <td>Preto</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>P002</td>\n",
"      <td>800.0</td>\n",
"      <td>1099.0</td>\n",
"      <td>Smart TV 4K</td>\n",
"      <td>Eletrônicos</td>\n",
"      <td>Samsung</td>\n",
"      <td>Prata</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>P003</td>\n",
"      <td>50.0</td>\n",
"      <td>79.0</td>\n",
"      <td>Fone de Ouvido Bluetooth</td>\n",
"      <td>Eletrônicos</td>\n",
"      <td>Sony</td>\n",
"      <td>Vermelho</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>P004</td>\n",
"      <td>200.0</td>\n",
```

```
"     <td>299.0</td>\n",
"     <td>Tablet Galaxy Tab A</td>\n",
"     <td>Eletrônicos</td>\n",
"     <td>Samsung</td>\n",
"     <td>Preto</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>4</th>\n",
"     <td>P005</td>\n",
"     <td>300.0</td>\n",
"     <td>449.0</td>\n",
"     <td>Câmera Digital</td>\n",
"     <td>Eletrônicos</td>\n",
"     <td>Canon</td>\n",
"     <td>Branco</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>5</th>\n",
"     <td>P010</td>\n",
"     <td>900.0</td>\n",
"     <td>1200.0</td>\n",
"     <td>iPhone 14'</td>\n",
"     <td>Eletrônicos</td>\n",
"     <td>Apple</td>\n",
"     <td>Cinza</td>\n",
"   </tr>\n",
" </tbody>\n",
"</table>\n",
"</div>\n",
"          <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-4e325fcb-f132-44c0-977c-77f3c82573c6')\"\n",
"            title=\"Convert this dataframe to an interactive table.\"\n",
"            style=\"display:none;\">\n",
"       \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",
```

```
"       width=\"24px\">\n",
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"       <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
"  </svg>\n",
"      </button>\n",
"      \n",
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
"\n",
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
"    }\n",
"\n",
```

```
"    [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
"    }\n",
"  </style>\n",
"\n",
"      <script>\n",
"        const buttonEl =\n",
"          document.querySelector('#df-4e325fcb-f132-44c0-977c-77f3c82573c6 button.colab-df-convert');\n",
"        buttonEl.style.display =\n",
"          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
"\n",
"        async function convertToInteractive(key) {\n",
"          const element = document.querySelector('#df-4e325fcb-f132-44c0-977c-77f3c82573c6');\n",
"          const dataTable =\n",
"            await google.colab.kernel.invokeFunction('convertToInteractive',\n",
"                                                     [key], {});\n",
"          if (!dataTable) return;\n",
"\n",
"          const docLinkHtml = 'Like what you see? Visit the ' +\n",
"                              '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebook</a>'\n",
"            + ' to learn more about interactive tables.';\n",
"          element.innerHTML = '';\n",
"          dataTable['output_type'] = 'display_data';\n",
"          await google.colab.output.renderOutput(dataTable, element);\n",
"          const docLink = document.createElement('div');\n",
```

```
          "          docLink.innerHTML = docLinkHtml;\n",
          "          element.appendChild(docLink);\n",
          "        }\n",
          "      </script>\n",
          "    </div>\n",
          "  </div>\n",
          "  "
        ]
      },
      "metadata": {},
      "execution_count": 5
    }
  ]
},
{
  "cell_type": "markdown",
  "source": [
    "**Q008 Listar todos os fornecedores cadastrados no sistema**"
  ],
  "metadata": {
    "id": "Op2raiJPPj-p"
  }
},
{
  "cell_type": "code",
  "source": [
    "fornecedores_cad = df_fornecedor['NOME'].head()\n",
    "fornecedores_cad "
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "QJFi8XDePyfc",
    "outputId": "a9f63dc6-76d2-401c-888d-56316fef8240"
  },
```

```
   "execution_count": null,
   "outputs": [
    {
     "output_type": "execute_result",
     "data": {
      "text/plain": [
       "0       João Silva\n",
       "1      Maria Santos\n",
       "2    Pedro Oliveira\n",
       "3        Ana Souza\n",
       "4     Carlos Mendes\n",
       "Name: NOME, dtype: object"
      ]
     },
     "metadata": {},
     "execution_count": 22
    }
   ]
  },
  {
   "cell_type": "markdown",
   "source": [
    "**Q010 Encontrar a receita total de uma loja**"
   ],
   "metadata": {
    "id": "ZlnxqtQMug9k"
   }
  },
  {
   "cell_type": "code",
   "source": [
    "receita_total = df_compravenda[['VALOR', 'COD_LOJA']].groupby(['COD_LOJA']).sum().reset_index()"
   ],
   "metadata": {
    "id": "gXRDKvL3ue9h"
```

```
  },
  "execution_count": null,
  "outputs": []
 },
 {
  "cell_type": "code",
  "source": [
   "receita_total.head(5)"
  ],
  "metadata": {
   "colab": {
    "base_uri": "https://localhost:8080/",
    "height": 143
   },
   "id": "BW1m_CiAuwh4",
   "outputId": "0c63a05c-44fa-4c46-8ced-f6aa07426ed0"
  },
  "execution_count": null,
  "outputs": [
   {
    "output_type": "execute_result",
    "data": {
     "text/plain": [
      "  COD_LOJA    VALOR\n",
      "0    LJ001  329.99\n",
      "1    LJ002  475.50\n",
      "2    LJ003  950.00"
     ],
     "text/html": [
      "\n",
      "  <div id=\"df-22180660-8775-49ca-8f9f-1fdc1444cd37\">\n",
      "    <div class=\"colab-df-container\">\n",
      "      <div>\n",
      "<style scoped>\n",
      "    .dataframe tbody tr th:only-of-type {\n",
      "        vertical-align: middle;\n",
```

```
"    }\n",
"\n",
"    .dataframe tbody tr th {\n",
"        vertical-align: top;\n",
"    }\n",
"\n",
"    .dataframe thead th {\n",
"        text-align: right;\n",
"    }\n",
"</style>\n",
"<table border=\"1\" class=\"dataframe\">\n",
"  <thead>\n",
"    <tr style=\"text-align: right;\">\n",
"      <th></th>\n",
"      <th>COD_LOJA</th>\n",
"      <th>VALOR</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>LJ001</td>\n",
"      <td>329.99</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>LJ002</td>\n",
"      <td>475.50</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>LJ003</td>\n",
"      <td>950.00</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
```

```
"</div>\n",
"          <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-22180660-8775-49ca-8f9f-1fdc1444cd37')\"\n",
"            title=\"Convert this dataframe to an interactive table.\"\n",
"            style=\"display:none;\">\n",
"       \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",
"       width=\"24px\">\n",
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"      <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
"  </svg>\n",
"      </button>\n",
"      \n",
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
"\n",
```

```
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
"    }\n",
"  </style>\n",
"\n",
"      <script>\n",
"        const buttonEl =\n",
"          document.querySelector('#df-22180660-8775-49ca-8f9f-1fdc1444cd37 button.colab-df-convert');\n",
"        buttonEl.style.display =\n",
"          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
"\n",
"        async function convertToInteractive(key) {\n",
"          const element = document.querySelector('#df-22180660-8775-49ca-8f9f-1fdc1444cd37');\n",
"          const dataTable =\n",
"            await google.colab.kernel.invokeFunction('convertToInteractive',\n",
"                                                     [key], {});\n",
"          if (!dataTable) return;\n",
"\n",
"          const docLinkHtml = 'Like what you see? Visit the ' +\n",
```

```
          "                                                    '<a target=\"_blank\"
href=https://colab.research.google.com/notebooks/data_table.ipynb>data table
notebook</a>'\n",
          "           + ' to learn more about interactive tables.';\n",
          "         element.innerHTML = '';\n",
          "         dataTable['output_type'] = 'display_data';\n",
          "         await google.colab.output.renderOutput(dataTable, element);\n",
          "         const docLink = document.createElement('div');\n",
          "         docLink.innerHTML = docLinkHtml;\n",
          "         element.appendChild(docLink);\n",
          "       }\n",
          "     </script>\n",
          "   </div>\n",
          " </div>\n",
          " "
        ]
       },
       "metadata": {},
       "execution_count": 33
      }
    ]
   },
   {
    "cell_type": "markdown",
    "source": [
     "**Q011 Listar vendas por vendedor**"
    ],
    "metadata": {
     "id": "E48dehFmxgoQ"
    }
   },
   {
    "cell_type": "code",
    "source": [
     "vendas_porv = df_compravenda[['COD_COMPRAVENDA',
'COD_VENDEDOR']].groupby('COD_VENDEDOR').count().reset_index()"
    ],
```

```
 "metadata": {
  "id": "13abliYCxfmO"
 },
 "execution_count": null,
 "outputs": []
},
{
 "cell_type": "code",
 "source": [
  "vendas_porv.head(5)"
 ],
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/",
   "height": 143
  },
  "id": "OJHTy4bMx9D0",
  "outputId": "3fdc1304-b93c-4649-b1eb-46f1c7879639"
 },
 "execution_count": null,
 "outputs": [
  {
   "output_type": "execute_result",
   "data": {
    "text/plain": [
     "  COD_VENDEDOR  COD_COMPRAVENDA\n",
     "0       VD001              2\n",
     "1       VD002              2\n",
     "2       VD003              1"
    ],
    "text/html": [
     "\n",
     "  <div id=\"df-c4312ac1-96ef-4f03-a621-422443c40150\">\n",
     "    <div class=\"colab-df-container\">\n",
     "      <div>\n",
     "<style scoped>\n",
```

```
"    .dataframe tbody tr th:only-of-type {\n",
"        vertical-align: middle;\n",
"    }\n",
"\n",
"    .dataframe tbody tr th {\n",
"        vertical-align: top;\n",
"    }\n",
"\n",
"    .dataframe thead th {\n",
"        text-align: right;\n",
"    }\n",
"</style>\n",
"<table border=\"1\" class=\"dataframe\">\n",
"  <thead>\n",
"    <tr style=\"text-align: right;\">\n",
"      <th></th>\n",
"      <th>COD_VENDEDOR</th>\n",
"      <th>COD_COMPRAVENDA</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>VD001</td>\n",
"      <td>2</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>VD002</td>\n",
"      <td>2</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>VD003</td>\n",
"      <td>1</td>\n",
"    </tr>\n",
```

```
"  </tbody>\n",
"</table>\n",
"</div>\n",
"          <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-c4312ac1-96ef-4f03-a621-422443c40150')\"\n",
"            title=\"Convert this dataframe to an interactive table.\"\n",
"            style=\"display:none;\">\n",
"       \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",
"      width=\"24px\">\n",
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"      <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
"  </svg>\n",
"    </button>\n",
"    \n",
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
```

```
"    }\n",
"\n",
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
"    }\n",
"  </style>\n",
"\n",
"      <script>\n",
"        const buttonEl =\n",
"          document.querySelector('#df-c4312ac1-96ef-4f03-a621-422443c40150 button.colab-df-convert');\n",
"        buttonEl.style.display =\n",
"          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
"\n",
"        async function convertToInteractive(key) {\n",
"          const element = document.querySelector('#df-c4312ac1-96ef-4f03-a621-422443c40150');\n",
"          const dataTable =\n",
"            await google.colab.kernel.invokeFunction('convertToInteractive',\n",
"                                                     [key], {});\n",
"          if (!dataTable) return;\n",
"\n",
```

```
          "          const docLinkHtml = 'Like what you see? Visit the ' +\n",
          "                                                    '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebook</a>'\n",
          "            + ' to learn more about interactive tables.';\n",
          "          element.innerHTML = '';\n",
          "          dataTable['output_type'] = 'display_data';\n",
          "          await google.colab.output.renderOutput(dataTable, element);\n",
          "          const docLink = document.createElement('div');\n",
          "          docLink.innerHTML = docLinkHtml;\n",
          "          element.appendChild(docLink);\n",
          "        }\n",
          "      </script>\n",
          "    </div>\n",
          "  </div>\n",
          "  "
         ]
        },
        "metadata": {},
        "execution_count": 36
      }
    ]
  },
  {
   "cell_type": "markdown",
   "source": [
    "**Q012 NOVA: Inserir um novo cliente no banco de dados**"
   ],
   "metadata": {
    "id": "o1kXLwX8yfMi"
   }
  },
  {
   "cell_type": "code",
   "source": [
    "df_novo_cliente = pd.DataFrame({\n",
```

```
    "    'CPF': [cpf],\n",
    "    'TELEFONE': [telefone],\n",
    "    'ENDERECO': [endereco],\n",
    "    'CEP': [cep],\n",
    "    'ESTADO': [estado],\n",
    "    'CIDADE': [cidade],\n",
    "    'LOGRADOURO': [logradouro],\n",
    "    'SEXO': [sexo],\n",
    "    'PRIMEIRO_NOME': [primeiro_nome],\n",
    "    'SOBRENOME': [sobrenome],\n",
    "    'DATA_NASCIMENTO': [data_nascimento]\n",
    "})\n",
    "\n",
    "# Inserir o novo cliente na tabela cliente\n",
    "df_cliente = df_cliente.append(df_novo_cliente, ignore_index=True)\n",
    "\n",
    "# Ver se deu certo...\n",
    "df_cliente.head()"
   ],
   "metadata": {
    "id": "yaeP4KB1yu59"
   },
   "execution_count": null,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
    "df_novo_cliente = pd.DataFrame(columns=['CPF', 'TELEFONE', 'ENDERECO', 'CEP', 'ESTADO', 'CIDADE', 'LOGRADOURO','SEXO', 'PRIMEIRO_NOME', 'SOBRENOME', 'DATA_NASCIMENTO' ])\n",
    "\n",
    "# Solicitar as informações do novo cliente ao usuário\n",
    "cpf = input(\"Informe o CPF do cliente: \")\n",
    "telefone = input(\"Informe o telefone do cliente: \")\n",
    "endereco = input(\"Informe o endereço do cliente: \")\n",
```

```
"cep = input(\"Informe o CEP do cliente: \")\n",
"estado = input(\"Informe o estado do cliente: \")\n",
"cidade = input(\"Informe a cidade do cliente: \")\n",
"logradouro = input(\"Informe o logradouro do cliente: \")\n",
"sexo = input(\"Informe o sexo do cliente: \")\n",
"primeiro_nome = input(\"Informe o primeiro nome do cliente: \")\n",
"sobrenome = input(\"Informe o sobrenome do cliente: \")\n",
"data_nascimento = input(\"Informe a data de nascimento do cliente: \")\n",
"\n",
"# Adicionar o novo cliente ao dataframe 'df_cliente'\n",
"#df_cliente.loc[len(df_cliente)] = [cpf, telefone, endereco, cep, estado, cidade, logradouro, sexo, primeiro_nome, sobrenome, data_nascimento]\n",
"\n",
"# Verificar se o novo cliente foi adicionado corretamente\n",
"print(df_cliente.head())"
],
"metadata": {
"id": "FhkF2VwFufK8"
},
"execution_count": null,
"outputs": []
},
{
"cell_type": "markdown",
"source": [
"insira uma informações qualquer e ele ira guardar:\n",
"\n",
"\n",
"121.232.333-44,\n",
"(33) 4321-1234,\n",
"Rua XYZ,\n",
"32,\n",
"43256-020,\n",
"Belo Horizonte,\n",
"BH,\n",
"Residencial TPNB,\n",
```

```
      "H,\n",
      "Marcius,\n",
      "Cavalcante,\n",
      "1950-02-12"
     ],
     "metadata": {
      "id": "_VRUQ481kZ-_"
     }
    },
    {
     "cell_type": "code",
     "source": [
      "# Realizar o merge das tabelas depois de inserir as informações, para guardar as informações juntas\n",
      "df_cliente = pd.concat([df_cliente, df_novo_cliente], ignore_index=True)"
     ],
     "metadata": {
      "id": "kC5D6vISnsYb"
     },
     "execution_count": null,
     "outputs": []
    },
    {
     "cell_type": "code",
     "source": [
      "df_cliente"
     ],
     "metadata": {
      "colab": {
       "base_uri": "https://localhost:8080/",
       "height": 250
      },
      "id": "xNl8CJTeu3L9",
      "outputId": "93667346-6b86-45a7-ae11-35d5488523a2"
     },
     "execution_count": 9,
```

```
  "outputs": [
   {
    "output_type": "execute_result",
    "data": {
     "text/plain": [
      "            CPF        TELEFONE        ENDERECO        CEP        ESTADO \\\n",
      "0  123.456.789-00  (11) 1234-5678      Rua A, 123  12345-678       São Paulo \n",
      "1  987.654.321-00  (22) 9876-5432  Avenida B, 456  98765-432  Rio de Janeiro \n",
      "2  111.222.333-44  (33) 1111-2222      Rua C, 789  54321-098    Minas Gerais \n",
      "3  555.444.333-22  (44) 5555-4444  Avenida D, 987  76543-210          Bahia \n",
      "4  777.888.999-00  (55) 7777-8888      Rua E, 321  01234-567       São Paulo \n",
      "\n",
      "            CIDADE          LOGRADOURO  SEXO  PRIMEIRO_NOME  SOBRENOME  \\\n",
      "0       São Paulo  Residencial ABC     M          João     Silva   \n",
      "1  Rio de Janeiro  Apartamento XYZ     F         Maria     Souza   \n",
      "2  Belo Horizonte         Casa 123     M         Pedro    Santos   \n",
      "3        Salvador       Bloco ABCD     F          Ana  Oliveira   \n",
      "4       São Paulo     Edifício EFGH     M         Lucas  Ferreira   \n",
      "\n",
      "  DATA_NASCIMENTO  \n",
      "0      1990-01-01  \n",
      "1      1985-05-10  \n",
      "2      1982-12-25  \n",
      "3      1995-09-15  \n",
      "4      1988-07-20  "
     ],
     "text/html": [
      "\n",
      "  <div id=\"df-129d1eef-9130-4503-be98-afcf42a8885e\">\n",
      "    <div class=\"colab-df-container\">\n",
      "      <div>\n",
```

```
"<style scoped>\n",
"    .dataframe tbody tr th:only-of-type {\n",
"        vertical-align: middle;\n",
"    }\n",
"\n",
"    .dataframe tbody tr th {\n",
"        vertical-align: top;\n",
"    }\n",
"\n",
"    .dataframe thead th {\n",
"        text-align: right;\n",
"    }\n",
"</style>\n",
"<table border=\"1\" class=\"dataframe\">\n",
"  <thead>\n",
"    <tr style=\"text-align: right;\">\n",
"      <th></th>\n",
"      <th>CPF</th>\n",
"      <th>TELEFONE</th>\n",
"      <th>ENDERECO</th>\n",
"      <th>CEP</th>\n",
"      <th>ESTADO</th>\n",
"      <th>CIDADE</th>\n",
"      <th>LOGRADOURO</th>\n",
"      <th>SEXO</th>\n",
"      <th>PRIMEIRO_NOME</th>\n",
"      <th>SOBRENOME</th>\n",
"      <th>DATA_NASCIMENTO</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>123.456.789-00</td>\n",
"      <td>(11) 1234-5678</td>\n",
"      <td>Rua A, 123</td>\n",
```

```
"     <td>12345-678</td>\n",
"     <td>São Paulo</td>\n",
"     <td>São Paulo</td>\n",
"     <td>Residencial ABC</td>\n",
"     <td>M</td>\n",
"     <td>João</td>\n",
"     <td>Silva</td>\n",
"     <td>1990-01-01</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>1</th>\n",
"     <td>987.654.321-00</td>\n",
"     <td>(22) 9876-5432</td>\n",
"     <td>Avenida B, 456</td>\n",
"     <td>98765-432</td>\n",
"     <td>Rio de Janeiro</td>\n",
"     <td>Rio de Janeiro</td>\n",
"     <td>Apartamento XYZ</td>\n",
"     <td>F</td>\n",
"     <td>Maria</td>\n",
"     <td>Souza</td>\n",
"     <td>1985-05-10</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>2</th>\n",
"     <td>111.222.333-44</td>\n",
"     <td>(33) 1111-2222</td>\n",
"     <td>Rua C, 789</td>\n",
"     <td>54321-098</td>\n",
"     <td>Minas Gerais</td>\n",
"     <td>Belo Horizonte</td>\n",
"     <td>Casa 123</td>\n",
"     <td>M</td>\n",
"     <td>Pedro</td>\n",
"     <td>Santos</td>\n",
"     <td>1982-12-25</td>\n",
```

```
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>555.444.333-22</td>\n",
"      <td>(44) 5555-4444</td>\n",
"      <td>Avenida D, 987</td>\n",
"      <td>76543-210</td>\n",
"      <td>Bahia</td>\n",
"      <td>Salvador</td>\n",
"      <td>Bloco ABCD</td>\n",
"      <td>F</td>\n",
"      <td>Ana</td>\n",
"      <td>Oliveira</td>\n",
"      <td>1995-09-15</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>4</th>\n",
"      <td>777.888.999-00</td>\n",
"      <td>(55) 7777-8888</td>\n",
"      <td>Rua E, 321</td>\n",
"      <td>01234-567</td>\n",
"      <td>São Paulo</td>\n",
"      <td>São Paulo</td>\n",
"      <td>Edifício EFGH</td>\n",
"      <td>M</td>\n",
"      <td>Lucas</td>\n",
"      <td>Ferreira</td>\n",
"      <td>1988-07-20</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
"          <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-
129d1eef-9130-4503-be98-afcf42a8885e')\"\n",
"            title=\"Convert this dataframe to an interactive table.\"\n",
"            style=\"display:none;\">\n",
```

```
"       \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",
"       width=\"24px\">\n",
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"       <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
"  </svg>\n",
"     </button>\n",
"     \n",
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
"\n",
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
"       box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
```

```
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
"    }\n",
"  </style>\n",
"\n",
"      <script>\n",
"        const buttonEl =\n",
"          document.querySelector('#df-129d1eef-9130-4503-be98-afcf42a8885e button.colab-df-convert');\n",
"        buttonEl.style.display =\n",
"          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
"\n",
"        async function convertToInteractive(key) {\n",
"          const element = document.querySelector('#df-129d1eef-9130-4503-be98-afcf42a8885e');\n",
"          const dataTable =\n",
"            await google.colab.kernel.invokeFunction('convertToInteractive',\n",
"                                                     [key], {});\n",
"          if (!dataTable) return;\n",
"\n",
"          const docLinkHtml = 'Like what you see? Visit the ' +\n",
"                              '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebook</a>'\n",
"            + ' to learn more about interactive tables.';\n",
"          element.innerHTML = '';\n",
"          dataTable['output_type'] = 'display_data';\n",
```

```
          "          await google.colab.output.renderOutput(dataTable, element);\n",
          "          const docLink = document.createElement('div');\n",
          "          docLink.innerHTML = docLinkHtml;\n",
          "          element.appendChild(docLink);\n",
          "        }\n",
          "      </script>\n",
          "    </div>\n",
          "  </div>\n",
          "  "
         ]
        },
        "metadata": {},
        "execution_count": 9
       }
      ]
     },
     {
      "cell_type": "markdown",
      "source": [
       "**Q013 Listar clientes por vendedor**"
      ],
      "metadata": {
       "id": "asPB8VSXy7Op"
      }
     },
     {
      "cell_type": "code",
      "source": [
       "df_resultado = df_vendedor.merge(df_compravenda, on='COD_VENDEDOR')\n",
       "df_resultado = df_resultado.merge(df_cliente, left_on='CPF_CLIENTE', right_on='CPF')\n",
       "df_resultado = df_resultado[['NOME', 'PRIMEIRO_NOME', 'SOBRENOME']]\n",
       "df_resultado = df_resultado.rename(columns={'PRIMEIRO_NOME': 'NOME_CLIENTE', 'SOBRENOME': 'SOBRENOME_CLIENTE'})\n",
       "df_resultado = df_resultado.sort_values(by='NOME')"
      ],
```

```
 "metadata": {
  "id": "2mJtCtiDyyEF"
 },
 "execution_count": null,
 "outputs": []
},
{
 "cell_type": "code",
 "source": [
  "df_resultado"
 ],
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/",
   "height": 206
  },
  "id": "WAltvt1r0JtJ",
  "outputId": "46ce36e1-65cb-4c23-a283-0b986cad5d3b"
 },
 "execution_count": null,
 "outputs": [
  {
   "output_type": "execute_result",
   "data": {
    "text/plain": [
     "            NOME NOME_CLIENTE SOBRENOME_CLIENTE\n",
     "0     João Silva        João           Silva\n",
     "1     João Silva         Ana        Oliveira\n",
     "2    Maria Santos       Maria          Souza\n",
     "4    Maria Santos       Pedro         Santos\n",
     "3  Pedro Oliveira       Maria          Souza"
    ],
    "text/html": [
     "\n",
     "  <div id=\"df-66b238f4-e91c-4698-9f84-7abbd697a25e\">\n",
     "    <div class=\"colab-df-container\">\n",
```

```
"      <div>\n",
"<style scoped>\n",
"    .dataframe tbody tr th:only-of-type {\n",
"        vertical-align: middle;\n",
"    }\n",
"\n",
"    .dataframe tbody tr th {\n",
"        vertical-align: top;\n",
"    }\n",
"\n",
"    .dataframe thead th {\n",
"        text-align: right;\n",
"    }\n",
"</style>\n",
"<table border=\"1\" class=\"dataframe\">\n",
"  <thead>\n",
"    <tr style=\"text-align: right;\">\n",
"      <th></th>\n",
"      <th>NOME</th>\n",
"      <th>NOME_CLIENTE</th>\n",
"      <th>SOBRENOME_CLIENTE</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>João Silva</td>\n",
"      <td>João</td>\n",
"      <td>Silva</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>João Silva</td>\n",
"      <td>Ana</td>\n",
"      <td>Oliveira</td>\n",
"    </tr>\n",
```

```
"    <tr>\n",
"      <th>2</th>\n",
"      <td>Maria Santos</td>\n",
"      <td>Maria</td>\n",
"      <td>Souza</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>4</th>\n",
"      <td>Maria Santos</td>\n",
"      <td>Pedro</td>\n",
"      <td>Santos</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>Pedro Oliveira</td>\n",
"      <td>Maria</td>\n",
"      <td>Souza</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
"          <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-66b238f4-e91c-4698-9f84-7abbd697a25e')\"\n",
"            title=\"Convert this dataframe to an interactive table.\"\n",
"            style=\"display:none;\">\n",
"       \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",
"      width=\"24px\">\n",
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"      <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
"  </svg>\n",
"      </button>\n",
```

```
"      \n",
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
"\n",
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
```

```
"      fill: #FFFFFF;\n",
"    }\n",
"  </style>\n",
"\n",
"      <script>\n",
"        const buttonEl =\n",
"          document.querySelector('#df-66b238f4-e91c-4698-9f84-7abbd697a25e button.colab-df-convert');\n",
"        buttonEl.style.display =\n",
"          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
"\n",
"        async function convertToInteractive(key) {\n",
"         const element = document.querySelector('#df-66b238f4-e91c-4698-9f84-7abbd697a25e');\n",
"          const dataTable =\n",
"            await google.colab.kernel.invokeFunction('convertToInteractive',\n",
"                                                     [key], {});\n",
"          if (!dataTable) return;\n",
"\n",
"          const docLinkHtml = 'Like what you see? Visit the ' +\n",
"                              '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebook</a>'\n",
"            + ' to learn more about interactive tables.';\n",
"          element.innerHTML = '';\n",
"          dataTable['output_type'] = 'display_data';\n",
"          await google.colab.output.renderOutput(dataTable, element);\n",
"          const docLink = document.createElement('div');\n",
"          docLink.innerHTML = docLinkHtml;\n",
"          element.appendChild(docLink);\n",
"        }\n",
"      </script>\n",
"    </div>\n",
"  </div>\n",
"  "
    ]
  },
```

```
      "metadata": {},
      "execution_count": 46
     }
    ]
   },
   {
    "cell_type": "markdown",
    "source": [
     "\n",
     "\n",
     "**Q014 Encontrar os produtos que nunca foram vendidos**\n"
    ],
    "metadata": {
     "id": "MNCiRgOz0Vz9"
    }
   },
   {
    "cell_type": "code",
    "source": [
     "# vamos combinar os dataframes com merge\n",
     "df_resultado = pd.merge(df_produto, df_compravenda, on='COD_PRODUTO', how='left', indicator=True)\n",
     "#produtos que nunca foram vendidos\n",
     "df_nao_vendidos = df_resultado[df_resultado['_merge'] == 'left_only']\n",
     "df_nao_vendidos = df_nao_vendidos[['NOME', 'COD_COMPRAVENDA']]\n",
     "df_nao_vendidos = df_nao_vendidos.sort_values(by='COD_COMPRAVENDA')\n",
     "\n",
     "df_nao_vendidos.head()"
    ],
    "metadata": {
     "colab": {
      "base_uri": "https://localhost:8080/",
      "height": 81
     },
     "id": "zQ4bkOVN0UMP",
```

```
  "outputId": "176712c6-cf70-48c6-b171-5e9f3de96f86"
},
"execution_count": null,
"outputs": [
  {
    "output_type": "execute_result",
    "data": {
      "text/plain": [
        "        NOME COD_COMPRAVENDA\n",
        "5  iPhone 14             NaN"
      ],
      "text/html": [
        "\n",
        "  <div id=\"df-5466ae21-a8bf-4919-ae41-124e28a98c92\">\n",
        "    <div class=\"colab-df-container\">\n",
        "      <div>\n",
        "<style scoped>\n",
        "    .dataframe tbody tr th:only-of-type {\n",
        "        vertical-align: middle;\n",
        "    }\n",
        "\n",
        "    .dataframe tbody tr th {\n",
        "        vertical-align: top;\n",
        "    }\n",
        "\n",
        "    .dataframe thead th {\n",
        "        text-align: right;\n",
        "    }\n",
        "</style>\n",
        "<table border=\"1\" class=\"dataframe\">\n",
        "  <thead>\n",
        "    <tr style=\"text-align: right;\">\n",
        "      <th></th>\n",
        "      <th>NOME</th>\n",
        "      <th>COD_COMPRAVENDA</th>\n",
        "    </tr>\n",
```

```
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>5</th>\n",
"      <td>iPhone 14</td>\n",
"      <td>NaN</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
"          <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-5466ae21-a8bf-4919-ae41-124e28a98c92')\"\n",
"            title=\"Convert this dataframe to an interactive table.\"\n",
"            style=\"display:none;\">\n",
"       \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",
"       width=\"24px\">\n",
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"       <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
"  </svg>\n",
"      </button>\n",
"      \n",
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
```

```
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
"\n",
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
"    }\n",
"  </style>\n",
"\n",
"      <script>\n",
"        const buttonEl =\n",
"          document.querySelector('#df-5466ae21-a8bf-4919-ae41-124e28a98c92 button.colab-df-convert');\n",
"        buttonEl.style.display =\n",
"          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
"\n",
```

```
          "        async function convertToInteractive(key) {\n",
          "         const element = document.querySelector('#df-5466ae21-a8bf-4919-ae41-124e28a98c92');\n",
          "          const dataTable =\n",
          "            await google.colab.kernel.invokeFunction('convertToInteractive',\n",
          "                                                     [key], {});\n",
          "          if (!dataTable) return;\n",
          "\n",
          "          const docLinkHtml = 'Like what you see? Visit the ' +\n",
          "                              '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebook</a>'\n",
          "            + ' to learn more about interactive tables.';\n",
          "          element.innerHTML = '';\n",
          "          dataTable['output_type'] = 'display_data';\n",
          "          await google.colab.output.renderOutput(dataTable, element);\n",
          "          const docLink = document.createElement('div');\n",
          "          docLink.innerHTML = docLinkHtml;\n",
          "          element.appendChild(docLink);\n",
          "        }\n",
          "      </script>\n",
          "    </div>\n",
          "  </div>\n",
          "  "
         ]
        },
        "metadata": {},
        "execution_count": 47
       }
      ]
     },
     {
      "cell_type": "markdown",
      "source": [
       "**Q015 Listar todos os clientes cadastrados no sistema**"
      ],
      "metadata": {
```

```
    "id": "_0JAcVo_0tVq"
   }
  },
  {
   "cell_type": "code",
   "source": [
    "df_cliente['PRIMEIRO_NOME'].head()\n"
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "id": "bCJEDB3fzq27",
    "outputId": "bfe65f7e-8185-405a-c4b0-82f35eda2384"
   },
   "execution_count": null,
   "outputs": [
    {
     "output_type": "execute_result",
     "data": {
      "text/plain": [
       "0     João\n",
       "1    Maria\n",
       "2    Pedro\n",
       "3      Ana\n",
       "4    Lucas\n",
       "Name: PRIMEIRO_NOME, dtype: object"
      ]
     },
     "metadata": {},
     "execution_count": 50
    }
   ]
  },
  {
   "cell_type": "markdown",
```

```
    "source": [
     "**Q016 NOVA: Contar quantas compras foram efetudadas por cliente**"
    ],
    "metadata": {
     "id": "ucMDEIwv1JtY"
    }
   },
   {
    "cell_type": "code",
    "source": [
     "compras_por_cliente = df_cliente.merge(df_compravenda, left_on='CPF', right_on='CPF_CLIENTE')\n",
     "contagem_compras = compras_por_cliente['PRIMEIRO_NOME'].value_counts().reset_index()\n",
     "contagem_compras.columns = ['PRIMEIRO_NOME', 'COUNT']\n",
     "contagem_compras.head(4)"
    ],
    "metadata": {
     "colab": {
      "base_uri": "https://localhost:8080/",
      "height": 175
     },
     "id": "WsDuc2xj1RXi",
     "outputId": "1b8ffd09-c3da-4a61-9ef4-6c09a99026c7"
    },
    "execution_count": null,
    "outputs": [
     {
      "output_type": "execute_result",
      "data": {
       "text/plain": [
        "  PRIMEIRO_NOME  COUNT\n",
        "0         Maria      2\n",
        "1          João      1\n",
        "2         Pedro      1\n",
        "3           Ana      1"
```

```
],
"text/html": [
 "\n",
 "  <div id=\"df-a4e82c85-66e5-45fe-bce1-4d6bb9cdfc74\">\n",
 "    <div class=\"colab-df-container\">\n",
 "      <div>\n",
 "<style scoped>\n",
 "    .dataframe tbody tr th:only-of-type {\n",
 "        vertical-align: middle;\n",
 "    }\n",
 "\n",
 "    .dataframe tbody tr th {\n",
 "        vertical-align: top;\n",
 "    }\n",
 "\n",
 "    .dataframe thead th {\n",
 "        text-align: right;\n",
 "    }\n",
 "</style>\n",
 "<table border=\"1\" class=\"dataframe\">\n",
 "  <thead>\n",
 "    <tr style=\"text-align: right;\">\n",
 "      <th></th>\n",
 "      <th>PRIMEIRO_NOME</th>\n",
 "      <th>COUNT</th>\n",
 "    </tr>\n",
 "  </thead>\n",
 "  <tbody>\n",
 "    <tr>\n",
 "      <th>0</th>\n",
 "      <td>Maria</td>\n",
 "      <td>2</td>\n",
 "    </tr>\n",
 "    <tr>\n",
 "      <th>1</th>\n",
 "      <td>João</td>\n",
```

```
"      <td>1</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>Pedro</td>\n",
"      <td>1</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>Ana</td>\n",
"      <td>1</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
"          <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-a4e82c85-66e5-45fe-bce1-4d6bb9cdfc74')\"\n",
"            title=\"Convert this dataframe to an interactive table.\"\n",
"            style=\"display:none;\">\n",
"       \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",
"      width=\"24px\">\n",
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"      <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
"  </svg>\n",
"      </button>\n",
"      \n",
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
```

```
"    }\n",
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
"\n",
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
"    }\n",
"  </style>\n",
"\n",
"      <script>\n",
"        const buttonEl =\n",
```

```
          "           document.querySelector('#df-a4e82c85-66e5-45fe-bce1-4d6bb9cdfc74
button.colab-df-convert');\n",
          "       buttonEl.style.display =\n",
          "         google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
          "\n",
          "       async function convertToInteractive(key) {\n",
          "        const element = document.querySelector('#df-a4e82c85-66e5-45fe-bce1-
4d6bb9cdfc74');\n",
          "         const dataTable =\n",
          "           await google.colab.kernel.invokeFunction('convertToInteractive',\n",
          "                                                    [key], {});\n",
          "         if (!dataTable) return;\n",
          "\n",
          "         const docLinkHtml = 'Like what you see? Visit the ' +\n",
          "                                                              '<a       target=\"_blank\"
href=https://colab.research.google.com/notebooks/data_table.ipynb>data                table
notebook</a>'\n",
          "           + ' to learn more about interactive tables.';\n",
          "         element.innerHTML = '';\n",
          "         dataTable['output_type'] = 'display_data';\n",
          "         await google.colab.output.renderOutput(dataTable, element);\n",
          "         const docLink = document.createElement('div');\n",
          "         docLink.innerHTML = docLinkHtml;\n",
          "         element.appendChild(docLink);\n",
          "       }\n",
          "     </script>\n",
          "   </div>\n",
          " </div>\n",
          " "
        ]
       },
       "metadata": {},
       "execution_count": 56
      }
     ]
    },
    {
```

```
    "cell_type": "markdown",
    "source": [
      "**Q017 NOVA: Visualizar compras realizadas por cliente**"
    ],
    "metadata": {
      "id": "BjIh4kV21yin"
    }
   },
   {
    "cell_type": "code",
    "source": [
      "# Juntar com inner join\n",
      "df = pd.merge(df_cliente, df_compravenda, left_on='CPF', right_on='CPF_CLIENTE')\n",
      "df = pd.merge(df, df_produto, on='COD_PRODUTO')\n",
      "df_grouped = df.groupby('COD_COMPRAVENDA').first().reset_index()\n",
      "df_result = df_grouped[['COD_COMPRAVENDA', 'PRIMEIRO_NOME', 'SOBRENOME', 'NOME', 'VALOR_TOTAL']]\n",
      "\n",
      "df_result.head()"
    ],
    "metadata": {
      "colab": {
        "base_uri": "https://localhost:8080/",
        "height": 206
      },
      "id": "DQKFkUGn1xMO",
      "outputId": "602a619e-fa23-43aa-edea-d5434423ec88"
    },
    "execution_count": null,
    "outputs": [
      {
        "output_type": "execute_result",
        "data": {
          "text/plain": [
            "   COD_COMPRAVENDA PRIMEIRO_NOME SOBRENOME NOME  \\\n",
```

```
 "0          CV001         João     Silva               iPhone 13   \n",
 "1          CV002         Maria     Souza              Smart TV 4K   \n",
 "2          CV003         Maria     Souza  Fone de Ouvido Bluetooth   \n",
 "3          CV004         Pedro    Santos       Tablet Galaxy Tab A   \n",
 "4          CV005          Ana  Oliveira           Câmera Digital   \n",
 "\n",
 "   VALOR_TOTAL  \n",
 "0       150.99  \n",
 "1        75.50  \n",
 "2       200.00  \n",
 "3       500.00  \n",
 "4      1000.00  "
],
"text/html": [
 "\n",
 "  <div id=\"df-fed169c3-75a7-4a8c-a458-95c327da3982\">\n",
 "    <div class=\"colab-df-container\">\n",
 "      <div>\n",
 "<style scoped>\n",
 "    .dataframe tbody tr th:only-of-type {\n",
 "        vertical-align: middle;\n",
 "    }\n",
 "\n",
 "    .dataframe tbody tr th {\n",
 "        vertical-align: top;\n",
 "    }\n",
 "\n",
 "    .dataframe thead th {\n",
 "        text-align: right;\n",
 "    }\n",
 "</style>\n",
 "<table border=\"1\" class=\"dataframe\">\n",
 "  <thead>\n",
 "    <tr style=\"text-align: right;\">\n",
 "      <th></th>\n",
 "      <th>COD_COMPRAVENDA</th>\n",
```

```
"      <th>PRIMEIRO_NOME</th>\n",
"      <th>SOBRENOME</th>\n",
"      <th>NOME</th>\n",
"      <th>VALOR_TOTAL</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>CV001</td>\n",
"      <td>João</td>\n",
"      <td>Silva</td>\n",
"      <td>iPhone 13</td>\n",
"      <td>150.99</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>CV002</td>\n",
"      <td>Maria</td>\n",
"      <td>Souza</td>\n",
"      <td>Smart TV 4K</td>\n",
"      <td>75.50</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>CV003</td>\n",
"      <td>Maria</td>\n",
"      <td>Souza</td>\n",
"      <td>Fone de Ouvido Bluetooth</td>\n",
"      <td>200.00</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>CV004</td>\n",
"      <td>Pedro</td>\n",
"      <td>Santos</td>\n",
```

```
"      <td>Tablet Galaxy Tab A</td>\n",
"      <td>500.00</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>4</th>\n",
"      <td>CV005</td>\n",
"      <td>Ana</td>\n",
"      <td>Oliveira</td>\n",
"      <td>Câmera Digital</td>\n",
"      <td>1000.00</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
"          <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-fed169c3-75a7-4a8c-a458-95c327da3982')\"\n",
"            title=\"Convert this dataframe to an interactive table.\"\n",
"            style=\"display:none;\">\n",
"       \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",
"      width=\"24px\">\n",
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"      <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94-2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94-2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
"  </svg>\n",
"      </button>\n",
"      \n",
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
"    }\n",
```

```
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
"\n",
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
"    }\n",
"  </style>\n",
"\n",
"      <script>\n",
"        const buttonEl =\n",
```

```
          "          document.querySelector('#df-fed169c3-75a7-4a8c-a458-95c327da3982
button.colab-df-convert');\n",
          "       buttonEl.style.display =\n",
          "         google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
          "\n",
          "       async function convertToInteractive(key) {\n",
          "        const element = document.querySelector('#df-fed169c3-75a7-4a8c-a458-
95c327da3982');\n",
          "         const dataTable =\n",
          "           await google.colab.kernel.invokeFunction('convertToInteractive',\n",
          "                                                    [key], {});\n",
          "         if (!dataTable) return;\n",
          "\n",
          "         const docLinkHtml = 'Like what you see? Visit the ' +\n",
          "                                                              '<a       target=\"_blank\"
href=https://colab.research.google.com/notebooks/data_table.ipynb>data                table
notebook</a>'\n",
          "           + ' to learn more about interactive tables.';\n",
          "         element.innerHTML = '';\n",
          "         dataTable['output_type'] = 'display_data';\n",
          "         await google.colab.output.renderOutput(dataTable, element);\n",
          "         const docLink = document.createElement('div');\n",
          "         docLink.innerHTML = docLinkHtml;\n",
          "         element.appendChild(docLink);\n",
          "       }\n",
          "     </script>\n",
          "   </div>\n",
          "  </div>\n",
          "  "
        ]
      },
      "metadata": {},
      "execution_count": 57
    }
  ]
},
{
```

```
  "cell_type": "markdown",
  "source": [
    "**Q018 NOVA: Atualizar dados cadastrais do cliente**"
  ],
  "metadata": {
    "id": "uu5LW1D5ZZFY"
  }
},
{
  "cell_type": "code",
  "source": [
    "import pandas as pd\n",
    "\n",
    "# Create a DataFrame with the new data to be inserted\n",
    "new_data = {\n",
    "    'CPF': ['123.456.789-00'],\n",
    "    'SOBRENOME': ['Filho'],\n",
    "    'NOME': ['João']\n",
    "}\n",
    "df = pd.DataFrame(new_data)\n",
    "\n",
    "\n"
  ],
  "metadata": {
    "id": "FtFGdZbxk7S7"
  },
  "execution_count": 10,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "# Merge the new data with the CLIENTE DataFrame\n",
    "df_cliente = pd.merge(df_cliente, df, on='CPF', how='outer')"
  ],
  "metadata": {
```

```
      "id": "9YykBuSRlEN-"
     },
     "execution_count": 11,
     "outputs": []
    },
    {
     "cell_type": "code",
     "source": [
      "df_cliente.head()"
     ],
     "metadata": {
      "colab": {
       "base_uri": "https://localhost:8080/",
       "height": 337
      },
      "id": "3PdZRQGdlJ7X",
      "outputId": "eb69e3c1-134f-4c82-ba84-f12f00fa99ed"
     },
     "execution_count": 12,
     "outputs": [
      {
       "output_type": "execute_result",
       "data": {
        "text/plain": [
         "       CPF       TELEFONE       ENDERECO       CEP       ESTADO  \\\n",
         "0  12345678900  (11) 1234-5678     Rua A, 123  12345-678      São Paulo  \n",
         "1  98765432100  (22) 9876-5432  Avenida B, 456  98765-432  Rio de Janeiro \n",
         "2  11122233344  (33) 1111-2222      Rua C, 789  54321-098     Minas Gerais \n",
         "3  55544433322  (44) 5555-4444  Avenida D, 987  76543-210         Bahia  \n",
         "4  77788899900  (55) 7777-8888     Rua E, 321  01234-567      São Paulo  \n",
         "\n",
         "            CIDADE          LOGRADOURO SEXO PRIMEIRO_NOME SOBRENOME_x  \\n",
         "0       São Paulo  Residencial ABC     M          João       Silva   \n",
         "1  Rio de Janeiro  Apartamento XYZ     F         Maria       Souza   \n",
```

```
 "2  Belo Horizonte        Casa 123    M        Pedro      Santos   \n",
 "3       Salvador       Bloco ABCD    F         Ana    Oliveira   \n",
 "4      São Paulo    Edifício EFGH    M        Lucas    Ferreira   \n",
 "\n",
 "  DATA_NASCIMENTO SOBRENOME_y NOME  \n",
 "0             NaT        NaN  NaN  \n",
 "1             NaT        NaN  NaN  \n",
 "2             NaT        NaN  NaN  \n",
 "3             NaT        NaN  NaN  \n",
 "4             NaT        NaN  NaN  "
],
"text/html": [
 "\n",
 "  <div id=\"df-a6c54e52-82d7-4fba-93ff-b22f86486361\">\n",
 "    <div class=\"colab-df-container\">\n",
 "      <div>\n",
 "<style scoped>\n",
 "    .dataframe tbody tr th:only-of-type {\n",
 "        vertical-align: middle;\n",
 "    }\n",
 "\n",
 "    .dataframe tbody tr th {\n",
 "        vertical-align: top;\n",
 "    }\n",
 "\n",
 "    .dataframe thead th {\n",
 "        text-align: right;\n",
 "    }\n",
 "</style>\n",
 "<table border=\"1\" class=\"dataframe\">\n",
 "  <thead>\n",
 "    <tr style=\"text-align: right;\">\n",
 "      <th></th>\n",
 "      <th>CPF</th>\n",
 "      <th>TELEFONE</th>\n",
 "      <th>ENDERECO</th>\n",
```

```
"     <th>CEP</th>\n",
"     <th>ESTADO</th>\n",
"     <th>CIDADE</th>\n",
"     <th>LOGRADOURO</th>\n",
"     <th>SEXO</th>\n",
"     <th>PRIMEIRO_NOME</th>\n",
"     <th>SOBRENOME_x</th>\n",
"     <th>DATA_NASCIMENTO</th>\n",
"     <th>SOBRENOME_y</th>\n",
"     <th>NOME</th>\n",
"   </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"   <tr>\n",
"     <th>0</th>\n",
"     <td>12345678900</td>\n",
"     <td>(11) 1234-5678</td>\n",
"     <td>Rua A, 123</td>\n",
"     <td>12345-678</td>\n",
"     <td>São Paulo</td>\n",
"     <td>São Paulo</td>\n",
"     <td>Residencial ABC</td>\n",
"     <td>M</td>\n",
"     <td>João</td>\n",
"     <td>Silva</td>\n",
"     <td>NaT</td>\n",
"     <td>NaN</td>\n",
"     <td>NaN</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>1</th>\n",
"     <td>98765432100</td>\n",
"     <td>(22) 9876-5432</td>\n",
"     <td>Avenida B, 456</td>\n",
"     <td>98765-432</td>\n",
"     <td>Rio de Janeiro</td>\n",
```

```
"     <td>Rio de Janeiro</td>\n",
"     <td>Apartamento XYZ</td>\n",
"     <td>F</td>\n",
"     <td>Maria</td>\n",
"     <td>Souza</td>\n",
"     <td>NaT</td>\n",
"     <td>NaN</td>\n",
"     <td>NaN</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>2</th>\n",
"     <td>11122233344</td>\n",
"     <td>(33) 1111-2222</td>\n",
"     <td>Rua C, 789</td>\n",
"     <td>54321-098</td>\n",
"     <td>Minas Gerais</td>\n",
"     <td>Belo Horizonte</td>\n",
"     <td>Casa 123</td>\n",
"     <td>M</td>\n",
"     <td>Pedro</td>\n",
"     <td>Santos</td>\n",
"     <td>NaT</td>\n",
"     <td>NaN</td>\n",
"     <td>NaN</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>3</th>\n",
"     <td>55544433322</td>\n",
"     <td>(44) 5555-4444</td>\n",
"     <td>Avenida D, 987</td>\n",
"     <td>76543-210</td>\n",
"     <td>Bahia</td>\n",
"     <td>Salvador</td>\n",
"     <td>Bloco ABCD</td>\n",
"     <td>F</td>\n",
"     <td>Ana</td>\n",
```

```
"      <td>Oliveira</td>\n",
"      <td>NaT</td>\n",
"      <td>NaN</td>\n",
"      <td>NaN</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>4</th>\n",
"      <td>77788899900</td>\n",
"      <td>(55) 7777-8888</td>\n",
"      <td>Rua E, 321</td>\n",
"      <td>01234-567</td>\n",
"      <td>São Paulo</td>\n",
"      <td>São Paulo</td>\n",
"      <td>Edifício EFGH</td>\n",
"      <td>M</td>\n",
"      <td>Lucas</td>\n",
"      <td>Ferreira</td>\n",
"      <td>NaT</td>\n",
"      <td>NaN</td>\n",
"      <td>NaN</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
"          <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-a6c54e52-82d7-4fba-93ff-b22f86486361')\"\n",
"            title=\"Convert this dataframe to an interactive table.\"\n",
"            style=\"display:none;\">\n",
"       \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",
"      width=\"24px\">\n",
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"      <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-
```

```
.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
"  </svg>\n",
"      </button>\n",
"      \n",
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
"\n",
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
```

```
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
"    }\n",
"  </style>\n",
"\n",
"      <script>\n",
"        const buttonEl =\n",
"          document.querySelector('#df-a6c54e52-82d7-4fba-93ff-b22f86486361 button.colab-df-convert');\n",
"        buttonEl.style.display =\n",
"          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
"\n",
"        async function convertToInteractive(key) {\n",
"          const element = document.querySelector('#df-a6c54e52-82d7-4fba-93ff-b22f86486361');\n",
"          const dataTable =\n",
"            await google.colab.kernel.invokeFunction('convertToInteractive',\n",
"                                                     [key], {});\n",
"          if (!dataTable) return;\n",
"\n",
"          const docLinkHtml = 'Like what you see? Visit the ' +\n",
"                              '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebook</a>'\n",
"            + ' to learn more about interactive tables.';\n",
"          element.innerHTML = '';\n",
"          dataTable['output_type'] = 'display_data';\n",
"          await google.colab.output.renderOutput(dataTable, element);\n",
"          const docLink = document.createElement('div');\n",
"          docLink.innerHTML = docLinkHtml;\n",
"          element.appendChild(docLink);\n",
"        }\n",
"      </script>\n",
"    </div>\n",
```

```
            "  </div>\n",
            "  "
           ]
          },
          "metadata": {},
          "execution_count": 12
        }
      ]
    }
  ]
}{
  "nbformat": 4,
  "nbformat_minor": 0,
  "metadata": {
    "colab": {
      "provenance": []
    },
    "kernelspec": {
      "name": "python3",
      "display_name": "Python 3"
    },
    "language_info": {
      "name": "python"
    }
  },
  "cells": [
    {
      "cell_type": "markdown",
      "source": [
        "## **Etapa 5 - Dataframes em Python**"
      ],
      "metadata": {
        "id": "iT-DAc7kqMZ7"
      }
    },
    {
```

```
    "cell_type": "markdown",
    "source": [
      "## Nesta etapa, você deverá converter as tabelas do banco de dados em dataframes para operações em Python, bem como as consultas (queries) presentes na especificação de requisitos funcionais.\n",
      "\n",
      "## Você deve entregar um notebook em Python, no formato .ipynb, contendo a codificação para fazer as conversões e relatar as dificuldades encontradas, bem como as alternativas adotadas para enfrentá-las."
    ],
    "metadata": {
      "id": "tLWXWvcVqOhO"
    }
  },
  {
    "cell_type": "code",
    "source": [
      "#Rodar as bibliotecas\n",
      "import sqlite3\n",
      "import pandas as pd\n"
    ],
    "metadata": {
      "id": "Qs0IQrOy_Fch"
    },
    "execution_count": null,
    "outputs": []
  },
  {
    "cell_type": "code",
    "source": [
      "from google.colab import files\n",
      "uploaded = files.upload()"
    ],
    "metadata": {
      "colab": {
        "base_uri": "https://localhost:8080/",
        "height": 75
```

```
    },
    "id": "pIVOtlyB3Bmo",
    "outputId": "1f9da909-0b63-4490-a37b-47d2cdadcf12"
   },
   "execution_count": 1,
   "outputs": [
    {
     "output_type": "display_data",
     "data": {
      "text/plain": [
       "<IPython.core.display.HTML object>"
      ],
      "text/html": [
       "\n",
       "      <input type=\"file\" id=\"files-7a356017-2dee-455c-9e65-9722557f5113\" name=\"files[]\" multiple disabled\n",
       "        style=\"border:none\" />\n",
       "     <output id=\"result-7a356017-2dee-455c-9e65-9722557f5113\">\n",
       "      Upload widget is only available when the cell has been executed in the\n",
       "      current browser session. Please rerun this cell to enable.\n",
       "      </output>\n",
       "      <script>// Copyright 2017 Google LLC\n",
       "//\n",
       "// Licensed under the Apache License, Version 2.0 (the \"License\");\n",
       "// you may not use this file except in compliance with the License.\n",
       "// You may obtain a copy of the License at\n",
       "//\n",
       "//      http://www.apache.org/licenses/LICENSE-2.0\n",
       "//\n",
       "// Unless required by applicable law or agreed to in writing, software\n",
       "// distributed under the License is distributed on an \"AS IS\" BASIS,\n",
       "// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.\n",
       "// See the License for the specific language governing permissions and\n",
       "// limitations under the License.\n",
       "\n",
```

```
"/**\n",
" * @fileoverview Helpers for google.colab Python module.\n",
" */\n",
"(function(scope) {\n",
"function span(text, styleAttributes = {}) {\n",
"  const element = document.createElement('span');\n",
"  element.textContent = text;\n",
"  for (const key of Object.keys(styleAttributes)) {\n",
"    element.style[key] = styleAttributes[key];\n",
"  }\n",
"  return element;\n",
"}\n",
"\n",
"// Max number of bytes which will be uploaded at a time.\n",
"const MAX_PAYLOAD_SIZE = 100 * 1024;\n",
"\n",
"function _uploadFiles(inputId, outputId) {\n",
"  const steps = uploadFilesStep(inputId, outputId);\n",
"  const outputElement = document.getElementById(outputId);\n",
"  // Cache steps on the outputElement to make it available for the next call\n",
"  // to uploadFilesContinue from Python.\n",
"  outputElement.steps = steps;\n",
"\n",
"  return _uploadFilesContinue(outputId);\n",
"}\n",
"\n",
"// This is roughly an async generator (not supported in the browser yet),\n",
"// where there are multiple asynchronous steps and the Python side is going\n",
"// to poll for completion of each step.\n",
"// This uses a Promise to block the python side on completion of each step,\n",
"// then passes the result of the previous step as the input to the next step.\n",
"function _uploadFilesContinue(outputId) {\n",
"  const outputElement = document.getElementById(outputId);\n",
"  const steps = outputElement.steps;\n",
"\n",
"  const next = steps.next(outputElement.lastPromiseValue);\n",
```

```
"  return Promise.resolve(next.value.promise).then((value) => {\n",
"    // Cache the last promise value to make it available to the next\n",
"    // step of the generator.\n",
"    outputElement.lastPromiseValue = value;\n",
"    return next.value.response;\n",
"  });\n",
"}\n",
"\n",
"/**\n",
" * Generator function which is called between each async step of the upload\n",
" * process.\n",
" * @param {string} inputId Element ID of the input file picker element.\n",
" * @param {string} outputId Element ID of the output display.\n",
" * @return {!Iterable<!Object>} Iterable of next steps.\n",
" */\n",
"function* uploadFilesStep(inputId, outputId) {\n",
"  const inputElement = document.getElementById(inputId);\n",
"  inputElement.disabled = false;\n",
"\n",
"  const outputElement = document.getElementById(outputId);\n",
"  outputElement.innerHTML = '';\n",
"\n",
"  const pickedPromise = new Promise((resolve) => {\n",
"    inputElement.addEventListener('change', (e) => {\n",
"      resolve(e.target.files);\n",
"    });\n",
"  });\n",
"\n",
"  const cancel = document.createElement('button');\n",
"  inputElement.parentElement.appendChild(cancel);\n",
"  cancel.textContent = 'Cancel upload';\n",
"  const cancelPromise = new Promise((resolve) => {\n",
"    cancel.onclick = () => {\n",
"      resolve(null);\n",
"    };\n",
"  });\n",
```

```
"\n",
"  // Wait for the user to pick the files.\n",
"  const files = yield {\n",
"    promise: Promise.race([pickedPromise, cancelPromise]),\n",
"    response: {\n",
"      action: 'starting',\n",
"    }\n",
"  };\n",
"\n",
"  cancel.remove();\n",
"\n",
"  // Disable the input element since further picks are not allowed.\n",
"  inputElement.disabled = true;\n",
"\n",
"  if (!files) {\n",
"    return {\n",
"      response: {\n",
"        action: 'complete',\n",
"      }\n",
"    };\n",
"  }\n",
"\n",
"  for (const file of files) {\n",
"    const li = document.createElement('li');\n",
"    li.append(span(file.name, {fontWeight: 'bold'}));\n",
"    li.append(span(\n",
"        `(${file.type || 'n/a'}) - ${file.size} bytes, ` +\n",
"        `last modified: ${\n",
"            file.lastModifiedDate ? file.lastModifiedDate.toLocaleDateString() :\n",
"                                    'n/a'} - `));\n",
"    const percent = span('0% done');\n",
"    li.appendChild(percent);\n",
"\n",
"    outputElement.appendChild(li);\n",
"\n",
"    const fileDataPromise = new Promise((resolve) => {\n",
```

```
"      const reader = new FileReader();\n",
"      reader.onload = (e) => {\n",
"        resolve(e.target.result);\n",
"      };\n",
"      reader.readAsArrayBuffer(file);\n",
"    });\n",
"    // Wait for the data to be ready.\n",
"    let fileData = yield {\n",
"      promise: fileDataPromise,\n",
"      response: {\n",
"        action: 'continue',\n",
"      }\n",
"    };\n",
"\n",
"    // Use a chunked sending to avoid message size limits. See b/62115660.\n",
"    let position = 0;\n",
"    do {\n",
"                  const length = Math.min(fileData.byteLength - position,
MAX_PAYLOAD_SIZE);\n",
"      const chunk = new Uint8Array(fileData, position, length);\n",
"      position += length;\n",
"\n",
"      const base64 = btoa(String.fromCharCode.apply(null, chunk));\n",
"      yield {\n",
"        response: {\n",
"          action: 'append',\n",
"          file: file.name,\n",
"          data: base64,\n",
"        },\n",
"      };\n",
"\n",
"      let percentDone = fileData.byteLength === 0 ?\n",
"          100 :\n",
"          Math.round((position / fileData.byteLength) * 100);\n",
"      percent.textContent = `${percentDone}% done`;\n",
"\n",
```

```
        "    } while (position < fileData.byteLength);\n",
        "  }\n",
        "\n",
        "  // All done.\n",
        "  yield {\n",
        "    response: {\n",
        "      action: 'complete',\n",
        "    }\n",
        "  };\n",
        "}\n",
        "\n",
        "scope.google = scope.google || {};\n",
        "scope.google.colab = scope.google.colab || {};\n",
        "scope.google.colab._files = {\n",
        "  _uploadFiles,\n",
        "  _uploadFilesContinue,\n",
        "};\n",
        "})(self);\n",
        "</script> "
       ]
      },
      "metadata": {}
    },
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "Saving trabalhosve.db to trabalhosve.db\n"
      ]
    }
  ]
 },
 {
  "cell_type": "code",
  "source": [
```

```
"#Refiz o banco no proprio sqlite3, NAO PRECISA RODAR, SO DEIXEI PARA MOSTRAR COMO FIZEMOS\n",
"\n",
"import sqlite3\n",
"\n",
"# Conectar ao banco de dados (será criado se não existir)\n",
"conn = sqlite3.connect('trabalhosve.db')\n",
"\n",
"# Criar um cursor para executar comandos SQL\n",
"cursor = conn.cursor()\n",
"\n",
"# DROP SCHEMA\n",
"cursor.execute(\"DROP TABLE IF EXISTS SVE;\")\n",
"\n",
"# CREATE SCHEMA\n",
"cursor.execute(\"CREATE TABLE IF NOT EXISTS SVE (id INTEGER PRIMARY KEY AUTOINCREMENT);\")\n",
"\n",
"# DROP TABLES\n",
"cursor.execute(\"DROP TABLE IF EXISTS COMPRAVENDA;\")\n",
"cursor.execute(\"DROP TABLE IF EXISTS VENDEDOR;\")\n",
"cursor.execute(\"DROP TABLE IF EXISTS LOJA;\")\n",
"cursor.execute(\"DROP TABLE IF EXISTS FORNECEDOR;\")\n",
"cursor.execute(\"DROP TABLE IF EXISTS PRODUTO;\")\n",
"cursor.execute(\"DROP TABLE IF EXISTS CLIENTE;\")\n",
"\n",
"# Criar a tabela \"PRODUTO\" se ela não existir\n",
"cursor.execute('''CREATE TABLE IF NOT EXISTS CLIENTE(\n",
"  CPF VARCHAR(14) PRIMARY KEY,\n",
"  TELEFONE VARCHAR(20),\n",
"  ENDERECO VARCHAR(45),\n",
"  CEP VARCHAR(9),\n",
"  ESTADO VARCHAR(45),\n",
"  CIDADE VARCHAR(45),\n",
"  LOGRADOURO VARCHAR(100),\n",
"  SEXO VARCHAR(1),\n",
```

```
"  PRIMEIRO_NOME VARCHAR(50),\n",
"  SOBRENOME VARCHAR(100),\n",
"  DATA_NASCIMENTO DATE\n",
")'')\n",
"\n",
"# Inserir os valores na tabela \"CLIENTE\"\n",
"cursor.execute(\"\"\"INSERT INTO CLIENTE (CPF, TELEFONE, ENDERECO, CEP, ESTADO, CIDADE, LOGRADOURO, SEXO, PRIMEIRO_NOME, SOBRENOME, DATA_NASCIMENTO)\n",
"            VALUES ('123.456.789-00', '(11) 1234-5678', 'Rua A, 123', '12345-678', 'São Paulo', 'São Paulo', 'Residencial ABC', 'M', 'João', 'Silva', '1990-01-01')\"\"\")\n",
"\n",
"cursor.execute(\"\"\"INSERT INTO CLIENTE (CPF, TELEFONE, ENDERECO, CEP, ESTADO, CIDADE, LOGRADOURO, SEXO, PRIMEIRO_NOME, SOBRENOME, DATA_NASCIMENTO)\n",
"            VALUES ('987.654.321-00', '(22) 9876-5432', 'Avenida B, 456', '98765-432', 'Rio de Janeiro', 'Rio de Janeiro', 'Apartamento XYZ', 'F', 'Maria', 'Souza', '1985-05-10')\"\"\")\n",
"\n",
"cursor.execute(\"\"\"INSERT INTO CLIENTE (CPF, TELEFONE, ENDERECO, CEP, ESTADO, CIDADE, LOGRADOURO, SEXO, PRIMEIRO_NOME, SOBRENOME, DATA_NASCIMENTO)\n",
"            VALUES ('111.222.333-44', '(33) 1111-2222', 'Rua C, 789', '54321-098', 'Minas Gerais', 'Belo Horizonte', 'Casa 123', 'M', 'Pedro', 'Santos', '1982-12-25')\"\"\")\n",
"\n",
"cursor.execute(\"\"\"INSERT INTO CLIENTE (CPF, TELEFONE, ENDERECO, CEP, ESTADO, CIDADE, LOGRADOURO, SEXO, PRIMEIRO_NOME, SOBRENOME, DATA_NASCIMENTO)\n",
"            VALUES ('555.444.333-22', '(44) 5555-4444', 'Avenida D, 987', '76543-210', 'Bahia', 'Salvador', 'Bloco ABCD', 'F', 'Ana', 'Oliveira', '1995-09-15')\"\"\")\n",
"\n",
"cursor.execute(\"\"\"INSERT INTO CLIENTE (CPF, TELEFONE, ENDERECO, CEP, ESTADO, CIDADE, LOGRADOURO, SEXO, PRIMEIRO_NOME, SOBRENOME, DATA_NASCIMENTO)\n",
"            VALUES ('777.888.999-00', '(55) 7777-8888', 'Rua E, 321', '01234-567', 'São Paulo', 'São Paulo', 'Edifício EFGH', 'M', 'Lucas', 'Ferreira', '1988-07-20')\"\"\")\n",
"\n",
"# Criar tabela LOJA se ela não existir\n",
"cursor.execute('''CREATE TABLE IF NOT EXISTS LOJA(\n",
"    COD_LOJA VARCHAR(45) PRIMARY KEY,\n",
```

```
"    CEP VARCHAR(45),\n",
"    ESTADO VARCHAR(45),\n",
"    CIDADE VARCHAR(45),\n",
"    LOGRADOURO VARCHAR(100),\n",
"    TELEFONE VARCHAR(45)\n",
")'")\n",
"\n",
"# Inserir registros na tabela LOJA\n",
"cursor.execute('''INSERT INTO LOJA (COD_LOJA, CEP, ESTADO, CIDADE, LOGRADOURO, TELEFONE)\n",
"VALUES ('LJ001', '12345-678', 'São Paulo', 'São Paulo', 'Rua A, 123', '(11) 1234-5678')''')\n",
"\n",
"cursor.execute('''INSERT INTO LOJA (COD_LOJA, CEP, ESTADO, CIDADE, LOGRADOURO, TELEFONE)\n",
"VALUES ('LJ002', '98765-432', 'Rio de Janeiro', 'Rio de Janeiro', 'Avenida B, 456', '(21) 9876-5432')''')\n",
"\n",
"cursor.execute('''INSERT INTO LOJA (COD_LOJA, CEP, ESTADO, CIDADE, LOGRADOURO, TELEFONE)\n",
"VALUES ('LJ003', '54321-876', 'Minas Gerais', 'Belo Horizonte', 'Rua C, 789', '(31) 5432-1876')''')\n",
"\n",
"cursor.execute('''INSERT INTO LOJA (COD_LOJA, CEP, ESTADO, CIDADE, LOGRADOURO, TELEFONE)\n",
"VALUES ('LJ004', '76543-210', 'Bahia', 'Salvador', 'Avenida D, 987', '(71) 7654-3210')''')\n",
"\n",
"cursor.execute('''INSERT INTO LOJA (COD_LOJA, CEP, ESTADO, CIDADE, LOGRADOURO, TELEFONE)\n",
"VALUES ('LJ005', '32109-876', 'Espírito Santo', 'Vitória', 'Rua E, 654', '(27) 3210-9876')''')\n",
"\n",
"\n",
"# Criar tabela PRODUTO se ela não existir\n",
"cursor.execute('''CREATE TABLE IF NOT EXISTS PRODUTO(\n",
"    COD_PRODUTO VARCHAR(20) PRIMARY KEY,\n",
"    CUSTO FLOAT,\n",
"    PRECO_VENDA FLOAT,\n",
```

```
"    NOME VARCHAR(100),\n",
"    CATEGORIA VARCHAR(45),\n",
"    MARCA VARCHAR(45),\n",
"    COR VARCHAR(20)\n",
")'')\n",
"\n",
"# Inserir registros na tabela PRODUTO\n",
"cursor.execute(''INSERT INTO PRODUTO (COD_PRODUTO, CUSTO, PRECO_VENDA, NOME, CATEGORIA, MARCA, COR)\n",
"VALUES ('P001', 700.00, 999.00, 'iPhone 13', 'Eletrônicos', 'Apple', 'Preto')'')\n",
"\n",
"cursor.execute(''INSERT INTO PRODUTO (COD_PRODUTO, CUSTO, PRECO_VENDA, NOME, CATEGORIA, MARCA, COR)\n",
"VALUES ('P002', 800.00, 1099.00, 'Smart TV 4K', 'Eletrônicos', 'Samsung', 'Prata')'')\n",
"\n",
"cursor.execute(''INSERT INTO PRODUTO (COD_PRODUTO, CUSTO, PRECO_VENDA, NOME, CATEGORIA, MARCA, COR)\n",
"VALUES ('P003', 50.00, 79.00, 'Fone de Ouvido Bluetooth', 'Eletrônicos', 'Sony', 'Vermelho')'')\n",
"\n",
"cursor.execute(''INSERT INTO PRODUTO (COD_PRODUTO, CUSTO, PRECO_VENDA, NOME, CATEGORIA, MARCA, COR)\n",
"VALUES ('P004', 200.00, 299.00, 'Tablet Galaxy Tab A', 'Eletrônicos', 'Samsung', 'Preto')'')\n",
"\n",
"cursor.execute(''INSERT INTO PRODUTO (COD_PRODUTO, CUSTO, PRECO_VENDA, NOME, CATEGORIA, MARCA, COR)\n",
"VALUES ('P005', 300.00, 449.00, 'Câmera Digital', 'Eletrônicos', 'Canon', 'Branco')'')\n",
"\n",
"\n",
"# Criar tabela VENDEDOR se ela não existir\n",
"cursor.execute(''CREATE TABLE IF NOT EXISTS VENDEDOR(\n",
"    COD_VENDEDOR VARCHAR(10) PRIMARY KEY,\n",
"    SALARIO FLOAT,\n",
"    SEXO VARCHAR(1),\n",
"    COMISSAO FLOAT,\n",
```

```
"    NOME VARCHAR(120),\n",
"    COD_LOJA VARCHAR(10),\n",
"    FOREIGN KEY (COD_LOJA) REFERENCES LOJA(COD_LOJA)\n",
")'')\n",
"\n",
"# Inserir registros na tabela VENDEDOR\n",
"cursor.execute(''INSERT INTO VENDEDOR (COD_VENDEDOR, SALARIO, SEXO, COMISSAO, NOME, COD_LOJA)\n",
"VALUES ('VD001', 3000.00, 'M', 0.00, 'João Silva', 'LJ001')'')\n",
"\n",
"cursor.execute(''INSERT INTO VENDEDOR (COD_VENDEDOR, SALARIO, SEXO, COMISSAO, NOME, COD_LOJA)\n",
"VALUES ('VD002', 2500.00, 'F', 0.05, 'Maria Santos', 'LJ002')'')\n",
"\n",
"cursor.execute(''INSERT INTO VENDEDOR (COD_VENDEDOR, SALARIO, SEXO, COMISSAO, NOME, COD_LOJA)\n",
"VALUES ('VD003', 4000.00, 'M', 0.00, 'Pedro Oliveira', 'LJ003')'')\n",
"\n",
"cursor.execute(''INSERT INTO VENDEDOR (COD_VENDEDOR, SALARIO, SEXO, COMISSAO, NOME, COD_LOJA)\n",
"VALUES ('VD004', 2800.00, 'F', 0.03, 'Ana Souza', 'LJ001')'')\n",
"\n",
"cursor.execute(''INSERT INTO VENDEDOR (COD_VENDEDOR, SALARIO, SEXO, COMISSAO, NOME, COD_LOJA)\n",
"VALUES ('VD005', 3500.00, 'M', 0.02, 'Carlos Mendes', 'LJ002')'')\n",
"\n",
"# Criar tabela COMPRAVENDA se ela não existir\n",
"cursor.execute(''CREATE TABLE IF NOT EXISTS COMPRAVENDA(\n",
"    COD_COMPRAVENDA VARCHAR(10) PRIMARY KEY,\n",
"    DATAS DATE,\n",
"    VALOR_TOTAL FLOAT,\n",
"    VALOR FLOAT,\n",
"    DESCONTO FLOAT,\n",
"    COD_VENDEDOR VARCHAR(10),\n",
"    CPF_CLIENTE VARCHAR(20),\n",
"    COD_LOJA VARCHAR(10),\n",
"    COD_PRODUTO VARCHAR(45),\n",
```

```
" FOREIGN KEY (COD_VENDEDOR) REFERENCES VENDEDOR(COD_VENDEDOR),\n",
" FOREIGN KEY (CPF_CLIENTE) REFERENCES CLIENTE(CPF),\n",
" FOREIGN KEY (COD_LOJA) REFERENCES LOJA(COD_LOJA),\n",
" FOREIGN KEY (COD_PRODUTO) REFERENCES PRODUTO(COD_PRODUTO) \n",
")'')\n",
"\n",
"# Inserir registros na tabela COMPRAVENDA\n",
"cursor.execute(''INSERT INTO COMPRAVENDA (COD_COMPRAVENDA, DATAS, VALOR_TOTAL, VALOR, DESCONTO, COD_VENDEDOR, CPF_CLIENTE, COD_LOJA, COD_PRODUTO)\n",
"VALUES ('CV001', '2023-05-17', 150.99, 129.99, 21.00, 'VD001', '123.456.789-00', 'LJ001', 'P001')'')\n",
"\n",
"cursor.execute(''INSERT INTO COMPRAVENDA (COD_COMPRAVENDA, DATAS, VALOR_TOTAL, VALOR, DESCONTO, COD_VENDEDOR, CPF_CLIENTE, COD_LOJA, COD_PRODUTO)\n",
"VALUES ('CV002', '2023-05-18', 75.50, 75.50, 0.00, 'VD002', '987.654.321-00', 'LJ002', 'P002')'')\n",
"\n",
"cursor.execute(''INSERT INTO COMPRAVENDA (COD_COMPRAVENDA, DATAS, VALOR_TOTAL, VALOR, DESCONTO, COD_VENDEDOR, CPF_CLIENTE, COD_LOJA, COD_PRODUTO)\n",
"VALUES ('CV003', '2023-05-19', 200.00, 200.00, 0.00, 'VD003', '987.654.321-00', 'LJ001', 'P003')'')\n",
"\n",
"cursor.execute(''INSERT INTO COMPRAVENDA (COD_COMPRAVENDA, DATAS, VALOR_TOTAL, VALOR, DESCONTO, COD_VENDEDOR, CPF_CLIENTE, COD_LOJA, COD_PRODUTO)\n",
"VALUES ('CV004', '2023-05-20', 500.00, 400.00, 100.00, 'VD002', '111.222.333-44', 'LJ002', 'P004')'')\n",
"\n",
"cursor.execute(''INSERT INTO COMPRAVENDA (COD_COMPRAVENDA, DATAS, VALOR_TOTAL, VALOR, DESCONTO, COD_VENDEDOR, CPF_CLIENTE, COD_LOJA, COD_PRODUTO)\n",
"VALUES ('CV005', '2023-05-21', 1000.00, 950.00, 50.00, 'VD001', '555.444.333-22', 'LJ003', 'P005')'')\n",
"\n",
"\n",
```

```
"# Criar tabela FORNECEDOR se ela não existir\n",
"cursor.execute('''CREATE TABLE IF NOT EXISTS FORNECEDOR(\n",
"    CNPJ VARCHAR(20) PRIMARY KEY,\n",
"    CEP VARCHAR(45),\n",
"    ESTADO VARCHAR(2),\n",
"    CIDADE VARCHAR(100),\n",
"    EMAIL VARCHAR(100),\n",
"    TELEFONE VARCHAR(20),\n",
"    NOME VARCHAR(120),\n",
"    COD_PRODUTO VARCHAR(100),\n",
"    FOREIGN KEY (COD_PRODUTO) REFERENCES PRODUTO(COD_PRODUTO)\n",
")''')\n",
"\n",
"# Inserir registros na tabela FORNECEDOR\n",
"cursor.execute('''INSERT INTO FORNECEDOR (CNPJ, CEP, ESTADO, CIDADE, EMAIL, TELEFONE, NOME, COD_PRODUTO)\n",
"VALUES ('12345678900001', '12345-678', 'SP', 'São Paulo', 'fornecedor1@example.com', '1111111111', 'Fornecedor 1', 'P001')''')\n",
"\n",
"cursor.execute('''INSERT INTO FORNECEDOR (CNPJ, CEP, ESTADO, CIDADE, EMAIL, TELEFONE, NOME, COD_PRODUTO)\n",
"VALUES ('98765432100002', '98765-432', 'RJ', 'Rio de Janeiro', 'fornecedor2@example.com', '2222222222', 'Fornecedor 2', 'P002')''')\n",
"\n",
"cursor.execute('''INSERT INTO FORNECEDOR (CNPJ, CEP, ESTADO, CIDADE, EMAIL, TELEFONE, NOME, COD_PRODUTO)\n",
"VALUES ('45678912300003', '54321-876', 'MG', 'Belo Horizonte', 'fornecedor3@example.com', '3333333333', 'Fornecedor 3', 'P003')''')\n",
"\n",
"cursor.execute('''INSERT INTO FORNECEDOR (CNPJ, CEP, ESTADO, CIDADE, EMAIL, TELEFONE, NOME, COD_PRODUTO)\n",
"VALUES ('78912345600004', '76543-210', 'RS', 'Porto Alegre', 'fornecedor4@example.com', '4444444444', 'Fornecedor 4', 'P004')''')\n",
"\n",
"cursor.execute('''INSERT INTO FORNECEDOR (CNPJ, CEP, ESTADO, CIDADE, EMAIL, TELEFONE, NOME, COD_PRODUTO)\n",
"VALUES ('32165498700005', '23456-789', 'SC', 'Florianópolis', 'fornecedor5@example.com', '5555555555', 'Fornecedor 5', 'P005')''')\n",
```

```
      "\n",
      "\n",
      "# Criar a tabela \"FORPROD\" se ela não existir\n",
      "cursor.execute('''CREATE TABLE IF NOT EXISTS FORPROD(\n",
      "  CNPJ VARCHAR(20),\n",
      "  COD_PRODUTO VARCHAR(20) PRIMARY KEY,\n",
      "  FOREIGN KEY (CNPJ) REFERENCES FORNECEDOR(CNPJ),\n",
      "  FOREIGN KEY (COD_PRODUTO) REFERENCES PRODUTO(COD_PRODUTO)\n",
      ")''')\n",
      "\n",
      "# Criar a tabela \"CVPROD\" se ela não existir\n",
      "cursor.execute('''CREATE TABLE IF NOT EXISTS CVPROD(\n",
      "  COD_COMPRAVENDA VARCHAR(20),\n",
      "  COD_PRODUTO VARCHAR(20) PRIMARY KEY,\n",
      "  FOREIGN KEY (COD_COMPRAVENDA) REFERENCES COMPRAVENDA(COD_COMPRAVENDA),\n",
      "  FOREIGN KEY (COD_PRODUTO) REFERENCES PRODUTO(COD_PRODUTO)\n",
      ")''')\n",
      "\n",
      "# Salvar as alterações\n",
      "conn.commit()\n",
      "\n",
      "# Fechar a conexão\n",
      "conn.close()\n",
      "\n",
      "\n",
      "\n"
    ],
    "metadata": {
      "id": "1mguSePoYHmE"
    },
    "execution_count": 2,
    "outputs": []
  },
  {
```

```
"cell_type": "code",
"source": [
  "import pandas as pd\n",
  "import sqlite3\n",
  "\n",
  "# Estabelecer conexão com o banco de dados\n",
  "conn = sqlite3.connect('trabalhosve.db')\n",
  "\n",
  "# Consulta à tabela CLIENTE\n",
  "query_cliente = \"SELECT * FROM CLIENTE\"\n",
  "df_cliente = pd.read_sql_query(query_cliente, conn)\n",
  "\n",
  "# Consulta à tabela PRODUTO\n",
  "query_produto = \"SELECT * FROM PRODUTO\"\n",
  "df_produto = pd.read_sql_query(query_produto, conn)\n",
  "\n",
  "# Consulta à tabela LOJA\n",
  "query_loja = \"SELECT * FROM LOJA\"\n",
  "df_loja = pd.read_sql_query(query_loja, conn)\n",
  "\n",
  "# Consulta à tabela COMPRAVENDA\n",
  "query_compravenda = \"SELECT * FROM COMPRAVENDA\"\n",
  "df_compravenda = pd.read_sql_query(query_compravenda, conn)\n",
  "\n",
  "# Consulta à tabela VENDEDOR\n",
  "query_vendedor = \"SELECT * FROM VENDEDOR\"\n",
  "df_vendedor = pd.read_sql_query(query_vendedor, conn)\n",
  "\n",
  "# Consulta à tabela VENDEDOR\n",
  "query_fornecedor = \"SELECT * FROM VENDEDOR\"\n",
  "df_fornecedor = pd.read_sql_query(query_fornecedor, conn)\n",
  "\n",
  "#Consulta a tabela CVPROD\n",
  "query_cvprod = \"SELECT * FROM CVPROD\"\n",
  "df_cvprod = pd.read_sql_query(query_cvprod, conn)\n",
  "\n",
```

```
"#Consulta a tabela CVPROD\n",
"query_forprod = \"SELECT * FROM FORPROD\"\n",
"df_forprod = pd.read_sql_query(query_forprod, conn)\n",
"\n",
"# Fechar a conexão com o banco de dados\n",
"conn.close()\n",
"\n",
"# Imprimir os resultados\n",
"print(\"Tabela CLIENTE:\")\n",
"print(df_cliente)\n",
"print()\n",
"\n",
"print(\"Tabela PRODUTO:\")\n",
"print(df_produto)\n",
"print()\n",
"\n",
"print(\"Tabela VENDEDOR:\")\n",
"print(df_vendedor)\n",
"print()\n",
"\n",
"print(\"Tabela FORNECEDOR:\")\n",
"print(df_fornecedor)\n",
"print()\n",
"\n",
"\n",
"print(\"Tabela LOJA:\")\n",
"print(df_loja)\n",
"print()\n",
"\n",
"print(\"Tabela COMPRAVENDA:\")\n",
"print(df_compravenda)\n",
"print()\n",
"\n",
"print(\"Tabela VENDEDOR:\")\n",
"print(df_vendedor)\n",
"\n",
```

```
    "print(\"Tabela CVPROD:\")\n",
    "print(df_cvprod)\n",
    "\n",
    "print(\"Tabela FORPROD:\")\n",
    "print(df_forprod)\n"
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "outputId": "083c66e2-ad8d-4203-fabd-0639501e5bca",
    "id": "Z2OxudhQczkO"
   },
   "execution_count": 2,
   "outputs": [
    {
     "output_type": "stream",
     "name": "stdout",
     "text": [
      "Tabela CLIENTE:\n",
      "            CPF       TELEFONE       ENDERECO      CEP       ESTADO \\\n",
      "0  123.456.789-00  (11) 1234-5678      Rua A, 123  12345-678      São Paulo \n",
      "1  987.654.321-00  (22) 9876-5432  Avenida B, 456  98765-432  Rio de Janeiro \n",
      "2  111.222.333-44  (33) 1111-2222      Rua C, 789  54321-098    Minas Gerais \n",
      "3  555.444.333-22  (44) 5555-4444  Avenida D, 987  76543-210          Bahia \n",
      "4  777.888.999-00  (55) 7777-8888      Rua E, 321  01234-567      São Paulo \n",
      "\n",
      "          CIDADE     LOGRADOURO SEXO PRIMEIRO_NOME SOBRENOME \\\n",
      "0      São Paulo  Residencial ABC    M          João     Silva   \n",
      "1  Rio de Janeiro  Apartamento XYZ    F         Maria     Souza   \n",
      "2  Belo Horizonte         Casa 123    M         Pedro    Santos   \n",
```

```
        "3       Salvador       Bloco ABCD    F          Ana  Oliveira   \n",
        "4      São Paulo    Edifício EFGH    M        Lucas  Ferreira   \n",
        "\n",
        "  DATA_NASCIMENTO  \n",
        "0      1990-01-01  \n",
        "1      1985-05-10  \n",
        "2      1982-12-25  \n",
        "3      1995-09-15  \n",
        "4      1988-07-20  \n",
        "\n",
        "Tabela PRODUTO:\n",
        "  COD_PRODUTO   CUSTO   PRECO_VENDA                          NOME
CATEGORIA  \\\n",
        "0       P001  700.0       999.0              iPhone 13  Eletrônicos   \n",
        "1       P002  800.0      1099.0             Smart TV 4K  Eletrônicos   \n",
        "2       P003   50.0        79.0  Fone de Ouvido Bluetooth  Eletrônicos   \n",
        "3       P004  200.0       299.0       Tablet Galaxy Tab A  Eletrônicos   \n",
        "4       P005  300.0       449.0           Câmera Digital  Eletrônicos   \n",
        "\n",
        "    MARCA       COR  \n",
        "0    Apple     Preto  \n",
        "1  Samsung     Prata  \n",
        "2     Sony  Vermelho  \n",
        "3  Samsung     Preto  \n",
        "4    Canon    Branco  \n",
        "\n",
        "Tabela VENDEDOR:\n",
        "  COD_VENDEDOR   SALARIO  SEXO    COMISSAO                NOME
COD_LOJA\n",
        "0       VD001   3000.0    M      0.00      João Silva    LJ001\n",
        "1       VD002   2500.0    F      0.05    Maria Santos    LJ002\n",
        "2       VD003   4000.0    M      0.00  Pedro Oliveira    LJ003\n",
        "3       VD004   2800.0    F      0.03       Ana Souza    LJ001\n",
        "4       VD005   3500.0    M      0.02   Carlos Mendes    LJ002\n",
        "\n",
        "Tabela FORNECEDOR:\n",
```

```
" COD_VENDEDOR SALARIO SEXO COMISSAO NOME COD_LOJA\n",
"0 VD001 3000.0 M 0.00 João Silva LJ001\n",
"1 VD002 2500.0 F 0.05 Maria Santos LJ002\n",
"2 VD003 4000.0 M 0.00 Pedro Oliveira LJ003\n",
"3 VD004 2800.0 F 0.03 Ana Souza LJ001\n",
"4 VD005 3500.0 M 0.02 Carlos Mendes LJ002\n",
"\n",
"Tabela LOJA:\n",
" COD_LOJA CEP ESTADO CIDADE LOGRADOURO \\\n",
"0 LJ001 12345-678 São Paulo São Paulo Rua A, 123 \n",
"1 LJ002 98765-432 Rio de Janeiro Rio de Janeiro Avenida B, 456 \n",
"2 LJ003 54321-876 Minas Gerais Belo Horizonte Rua C, 789 \n",
"3 LJ004 76543-210 Bahia Salvador Avenida D, 987 \n",
"4 LJ005 32109-876 Espírito Santo Vitória Rua E, 654 \n",
"\n",
" TELEFONE \n",
"0 (11) 1234-5678 \n",
"1 (21) 9876-5432 \n",
"2 (31) 5432-1876 \n",
"3 (71) 7654-3210 \n",
"4 (27) 3210-9876 \n",
"\n",
"Tabela COMPRAVENDA:\n",
" COD_COMPRAVENDA DATAS VALOR_TOTAL VALOR DESCONTO COD_VENDEDOR \\\n",
"0 CV001 2023-05-17 150.99 129.99 21.0 VD001 \n",
"1 CV002 2023-05-18 75.50 75.50 0.0 VD002 \n",
"2 CV003 2023-05-19 200.00 200.00 0.0 VD003 \n",
"3 CV004 2023-05-20 500.00 400.00 100.0 VD002 \n",
"4 CV005 2023-05-21 1000.00 950.00 50.0 VD001 \n",
"\n",
" CPF_CLIENTE COD_LOJA COD_PRODUTO \n",
"0 123.456.789-00 LJ001 P001 \n",
"1 987.654.321-00 LJ002 P002 \n",
"2 987.654.321-00 LJ001 P003 \n",
```

```
        "3  111.222.333-44    LJ002       P004  \n",
        "4  555.444.333-22    LJ003       P005  \n",
        "\n",
        "Tabela VENDEDOR:\n",
        "   COD_VENDEDOR   SALARIO  SEXO    COMISSAO                NOME COD_LOJA\n",
        "0       VD001   3000.0    M      0.00      João Silva    LJ001\n",
        "1       VD002   2500.0    F      0.05    Maria Santos    LJ002\n",
        "2       VD003   4000.0    M      0.00  Pedro Oliveira    LJ003\n",
        "3       VD004   2800.0    F      0.03      Ana Souza    LJ001\n",
        "4       VD005   3500.0    M      0.02   Carlos Mendes    LJ002\n",
        "Tabela CVPROD:\n",
        "Empty DataFrame\n",
        "Columns: [COD_COMPRAVENDA, COD_PRODUTO]\n",
        "Index: []\n",
        "Tabela FORPROD:\n",
        "Empty DataFrame\n",
        "Columns: [CNPJ, COD_PRODUTO]\n",
        "Index: []\n"
       ]
      }
     ]
    },
    {
     "cell_type": "code",
     "source": [
      "df_produtos.dtypes"
     ],
     "metadata": {
      "colab": {
       "base_uri": "https://localhost:8080/"
      },
      "id": "eDMQYegTxsI-",
      "outputId": "444e3927-dc98-4c00-9551-617474eabe81"
     },
    "execution_count": 6,
```

```
    "outputs": [
     {
      "output_type": "execute_result",
      "data": {
       "text/plain": [
        "COD_PRODUTO     object\n",
        "CUSTO          float64\n",
        "PRECO_VENDA    float64\n",
        "NOME           object\n",
        "CATEGORIA      object\n",
        "MARCA          object\n",
        "COR            object\n",
        "dtype: object"
       ]
      },
      "metadata": {},
      "execution_count": 6
     }
    ]
   },
   {
    "cell_type": "code",
    "source": [
     "#Convertendo tipos de dados\n",
     "df_produto['COD_PRODUTO'] = df_produto['COD_PRODUTO'].astype(str)\n",
     "df_produto['CUSTO'] = df_produto['CUSTO'].astype(float)\n",
     "df_produto['PRECO_VENDA']                                                    =
df_produto['PRECO_VENDA'].astype(float)\n",
     "df_produto['NOME'] = df_produto['NOME'].astype(str)\n",
     "df_produto['CATEGORIA'] = df_produto['CATEGORIA'].astype(str)\n",
     "df_produto['MARCA'] = df_produto['MARCA'].astype(str)\n",
     "df_produto['COR'] = df_produto['COR'].astype(str)\n"
    ],
    "metadata": {
     "id": "YgJ3kEp_1gbM"
    },
```

```
 "execution_count": 4,
 "outputs": []
},
{
 "cell_type": "code",
 "source": [
  "df_cliente.dtypes"
 ],
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/"
  },
  "id": "xV43tDtDx25R",
  "outputId": "e57e6c43-9c41-456b-e1ba-2b70b14d2063"
 },
 "execution_count": 7,
 "outputs": [
  {
   "output_type": "execute_result",
   "data": {
    "text/plain": [
     "CPF               object\n",
     "TELEFONE          object\n",
     "ENDERECO          object\n",
     "CEP               object\n",
     "ESTADO            object\n",
     "CIDADE            object\n",
     "LOGRADOURO        object\n",
     "SEXO              object\n",
     "PRIMEIRO_NOME     object\n",
     "SOBRENOME         object\n",
     "DATA_NASCIMENTO   object\n",
     "dtype: object"
    ]
   },
   "metadata": {},
```

```
      "execution_count": 7
     }
    ]
   },
   {
    "cell_type": "code",
    "source": [
     "# Remover pontos e traço do CPF para nao dar erro\n",
     "df_cliente['CPF'] = df_cliente['CPF'].str.replace('.', '').str.replace('-', '')"
    ],
    "metadata": {
     "colab": {
      "base_uri": "https://localhost:8080/"
     },
     "id": "aHDuVsRg0oYB",
     "outputId": "126178c9-a436-4fa6-a70e-c9d4cd4737f4"
    },
    "execution_count": 5,
    "outputs": [
     {
      "output_type": "stream",
      "name": "stderr",
      "text": [
       "<ipython-input-5-dbd20c55266e>:2: FutureWarning: The default value of regex will change from True to False in a future version. In addition, single character regular expressions will *not* be treated as literal strings when regex=True.\n",
       "  df_cliente['CPF'] = df_cliente['CPF'].str.replace('.', '').str.replace('-', '')\n"
      ]
     }
    ]
   },
   {
    "cell_type": "code",
    "source": [
     "# Converter os tipos de dados das colunas\n",
     "df_cliente['CPF'] = df_cliente['CPF'].astype(str)\n",
```

```
      "df_cliente['TELEFONE'] = df_cliente['TELEFONE'].astype(str)\n",
      "df_cliente['CEP'] = df_cliente['CEP'].astype(str)\n",
      "df_cliente['ESTADO'] = df_cliente['ESTADO'].astype('category')\n",
      "df_cliente['DATA_NASCIMENTO'] = pd.to_datetime(df_cliente['DATA_NASCIMENTO'], format='%d/%m/%Y', errors='coerce')"
     ],
     "metadata": {
      "id": "W4dIi5qx0pyZ"
     },
     "execution_count": 6,
     "outputs": []
    },
    {
     "cell_type": "code",
     "source": [
      "#verificar numeros faltantes\n",
      "df_cliente.isna().sum()"
     ],
     "metadata": {
      "colab": {
       "base_uri": "https://localhost:8080/"
      },
      "id": "jAYwNptXrmi3",
      "outputId": "fefaf1f6-7ab8-4b56-a691-9f66ace58946"
     },
     "execution_count": 4,
     "outputs": [
      {
       "output_type": "execute_result",
       "data": {
        "text/plain": [
         "CPF             0\n",
         "TELEFONE        0\n",
         "ENDERECO        0\n",
         "CEP             0\n",
```

```
        "ESTADO            0\n",
        "CIDADE            0\n",
        "LOGRADOURO        0\n",
        "SEXO              0\n",
        "PRIMEIRO_NOME     0\n",
        "SOBRENOME         0\n",
        "DATA_NASCIMENTO   0\n",
        "dtype: int64"
       ]
      },
      "metadata": {},
      "execution_count": 4
    }
  ]
},
{
  "cell_type": "code",
  "source": [
    "df_vendedor.dtypes"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "ezUooO8zx8IS",
    "outputId": "6d80144f-80f9-48cd-f3dd-fbaf0ba3e366"
  },
  "execution_count": 8,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "COD_VENDEDOR     object\n",
          "SALARIO         float64\n",
          "SEXO             object\n",
```

```
        "COMISSAO       float64\n",
        "NOME            object\n",
        "COD_LOJA        object\n",
        "dtype: object"
       ]
      },
      "metadata": {},
      "execution_count": 8
    }
  ]
},
{
  "cell_type": "code",
  "source": [
    "df_fornecedor.dtypes"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "RUHihXmhyCYZ",
    "outputId": "0f0617af-c773-4bcd-ad1d-c5d0894dbcd2"
  },
  "execution_count": 9,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "COD_VENDEDOR     object\n",
          "SALARIO         float64\n",
          "SEXO            object\n",
          "COMISSAO        float64\n",
          "NOME            object\n",
          "COD_LOJA        object\n",
          "dtype: object"
```

```
        ]
       },
       "metadata": {},
       "execution_count": 9
      }
     ]
    },
    {
     "cell_type": "code",
     "source": [
      "df_fornecedor['COD_VENDEDOR'] = df_fornecedor['COD_VENDEDOR'].astype(str)\n",
      "df_fornecedor['SALARIO'] = df_fornecedor['SALARIO'].astype(float)\n",
      "df_fornecedor['SEXO'] = df_fornecedor['SEXO'].astype(str)\n",
      "df_fornecedor['COMISSAO'] = df_fornecedor['COMISSAO'].astype(float)\n",
      "df_fornecedor['NOME'] = df_fornecedor['NOME'].astype(str)\n",
      "df_fornecedor['COD_LOJA'] = df_fornecedor['COD_LOJA'].astype(str)\n"
     ],
     "metadata": {
      "id": "l461wxWJ0STK"
     },
     "execution_count": 7,
     "outputs": []
    },
    {
     "cell_type": "code",
     "source": [
      "#Verificar se tem numeros faltantes\n",
      "df_fornecedor.isna().sum()"
     ],
     "metadata": {
      "colab": {
       "base_uri": "https://localhost:8080/"
      },
      "id": "9HOWPLJmr9UF",
      "outputId": "2bb52ce9-9d03-4553-a5de-79f0bb7da561"
```

```
    },
    "execution_count": 5,
    "outputs": [
     {
      "output_type": "execute_result",
      "data": {
       "text/plain": [
        "COD_VENDEDOR    0\n",
        "SALARIO         0\n",
        "SEXO            0\n",
        "COMISSAO        0\n",
        "NOME            0\n",
        "COD_LOJA        0\n",
        "dtype: int64"
       ]
      },
      "metadata": {},
      "execution_count": 5
     }
    ]
   },
   {
    "cell_type": "code",
    "source": [
     "df_fornecedor['COD_VENDEDOR'] = df_fornecedor['COD_VENDEDOR'].astype(str)\n",
     "df_fornecedor['SALARIO'] = df_fornecedor['SALARIO'].astype(float)\n",
     "df_fornecedor['SEXO'] = df_fornecedor['SEXO'].astype(str)\n",
     "df_fornecedor['COMISSAO'] = df_fornecedor['COMISSAO'].astype(float)\n",
     "df_fornecedor['NOME'] = df_fornecedor['NOME'].astype(str)\n",
     "df_fornecedor['COD_LOJA'] = df_fornecedor['COD_LOJA'].astype(str)\n"
    ],
    "metadata": {
     "id": "aRCaQTgjzviY"
    },
    "execution_count": null,
```

```
  "outputs": []
 },
 {
  "cell_type": "code",
  "source": [
   "df_loja.dtypes"
  ],
  "metadata": {
   "colab": {
    "base_uri": "https://localhost:8080/"
   },
   "id": "cKCfjmzoyMLZ",
   "outputId": "c51f7d08-6e25-47e9-e3ce-945d83b6ab04"
  },
  "execution_count": 10,
  "outputs": [
   {
    "output_type": "execute_result",
    "data": {
     "text/plain": [
      "COD_LOJA      object\n",
      "CEP           object\n",
      "ESTADO        object\n",
      "CIDADE        object\n",
      "LOGRADOURO    object\n",
      "TELEFONE      object\n",
      "dtype: object"
     ]
    },
    "metadata": {},
    "execution_count": 10
   }
  ]
 },
 {
  "cell_type": "code",
```

```
  "source": [
   "df_loja['COD_LOJA'] = df_loja['COD_LOJA'].astype(str)\n",
   "df_loja['CEP'] = df_loja['CEP'].astype(str)\n",
   "df_loja['ESTADO'] = df_loja['ESTADO'].astype(str)\n",
   "df_loja['CIDADE'] = df_loja['CIDADE'].astype(str)\n",
   "df_loja['LOGRADOURO'] = df_loja['LOGRADOURO'].astype(str)\n",
   "df_loja['TELEFONE'] = df_loja['TELEFONE'].astype(str)\n"
  ],
  "metadata": {
   "id": "qdQT9oa9zfAz"
  },
  "execution_count": 8,
  "outputs": []
 },
 {
  "cell_type": "code",
  "source": [
   "df_loja.isna().sum()"
  ],
  "metadata": {
   "colab": {
    "base_uri": "https://localhost:8080/"
   },
   "id": "Emb2NlettB3H",
   "outputId": "e9515022-d8c5-4d83-abdc-0621d42b0ad9"
  },
  "execution_count": 6,
  "outputs": [
   {
    "output_type": "execute_result",
    "data": {
     "text/plain": [
      "COD_LOJA      0\n",
      "CEP           0\n",
      "ESTADO        0\n",
      "CIDADE        0\n",
```

```
        "LOGRADOURO    0\n",
        "TELEFONE      0\n",
        "dtype: int64"
       ]
      },
      "metadata": {},
      "execution_count": 6
    }
  ]
},
{
  "cell_type": "code",
  "source": [
    "df_compravenda.dtypes"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "yL-sp-HeyQSk",
    "outputId": "b30c953b-f117-46ea-ca6c-a24d0431a7a0"
  },
  "execution_count": 11,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "COD_COMPRAVENDA     object\n",
          "DATAS               object\n",
          "VALOR_TOTAL        float64\n",
          "VALOR              float64\n",
          "DESCONTO           float64\n",
          "COD_VENDEDOR        object\n",
          "CPF_CLIENTE         object\n",
          "COD_LOJA            object\n",
```

```
          "COD_PRODUTO         object\n",
          "dtype: object"
         ]
        },
        "metadata": {},
        "execution_count": 11
       }
     ]
    },
    {
     "cell_type": "code",
     "source": [
       "df_compravenda['COD_COMPRAVENDA'] = df_compravenda['COD_COMPRAVENDA'].astype(str)\n",
       "df_compravenda['DATAS'] = df_compravenda['DATAS'].astype(str)\n",
       "df_compravenda['VALOR_TOTAL'] = df_compravenda['VALOR_TOTAL'].astype(float)\n",
       "df_compravenda['VALOR'] = df_compravenda['VALOR'].astype(float)\n",
       "df_compravenda['DESCONTO'] = df_compravenda['DESCONTO'].astype(float)\n",
       "df_compravenda['COD_VENDEDOR'] = df_compravenda['COD_VENDEDOR'].astype(str)\n",
       "df_compravenda['CPF_CLIENTE'] = df_compravenda['CPF_CLIENTE'].astype(str)\n",
       "df_compravenda['COD_LOJA'] = df_compravenda['COD_LOJA'].astype(str)\n",
       "df_compravenda['COD_PRODUTO'] = df_compravenda['COD_PRODUTO'].astype(str)\n"
     ],
     "metadata": {
       "id": "e_yaDEp1zPPw"
     },
     "execution_count": 9,
     "outputs": []
    },
    {
     "cell_type": "code",
     "source": [
```

```
    "df_compravenda.isna().sum()"
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "id": "5pmBtIFstG2T",
    "outputId": "bf19ccc9-e177-4d01-c945-b2030c6061c4"
   },
   "execution_count": 8,
   "outputs": [
    {
     "output_type": "execute_result",
     "data": {
      "text/plain": [
       "COD_COMPRAVENDA    0\n",
       "DATAS              0\n",
       "VALOR_TOTAL        0\n",
       "VALOR              0\n",
       "DESCONTO           0\n",
       "COD_VENDEDOR       0\n",
       "CPF_CLIENTE        0\n",
       "COD_LOJA           0\n",
       "COD_PRODUTO        0\n",
       "dtype: int64"
      ]
     },
     "metadata": {},
     "execution_count": 8
    }
   ]
  },
  {
   "cell_type": "markdown",
   "source": [
    "**Q001  Listar todas as lojas cadastradas no sistema**"
```

```
 ],
 "metadata": {
  "id": "kwzFfDGjBnO0"
 }
},
{
 "cell_type": "code",
 "source": [
  "# Listar todas as lojas cadastradas\n",
  "lojas_cadastradas = df_loja['COD_LOJA'].tolist()\n",
  "lojas_cadastradas"
 ],
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/"
  },
  "id": "BGgetTA9v627",
  "outputId": "5a140d0d-3411-4b41-a223-db64578d44eb"
 },
 "execution_count": null,
 "outputs": [
  {
   "output_type": "execute_result",
   "data": {
    "text/plain": [
     "['LJ001', 'LJ002', 'LJ003', 'LJ004', 'LJ005']"
    ]
   },
   "metadata": {},
   "execution_count": 4
  }
 ]
},
{
 "cell_type": "markdown",
 "source": [
```

```
      "**Q002 Encontrar a loja com o maior número de vendas no último mês**"
     ],
     "metadata": {
      "id": "xlDjJejRDPAe"
     }
    },
    {
     "cell_type": "code",
     "source": [
      "import pandas as pd\n",
      "\n",
      "# Converter a coluna 'DATAS' para o tipo datetime\n",
      "df_compravenda['DATAS'] = pd.to_datetime(df_compravenda['DATAS'])\n",
      "\n",
      "# Filtrar os dados para o último mês\n",
      "ultimo_mes = df_compravenda[df_compravenda['DATAS'] >= df_compravenda['DATAS'].max() - pd.DateOffset(months=1)]\n",
      "\n",
      "# Calcular o número de vendas para cada loja\n",
      "vendas_por_loja = ultimo_mes['COD_LOJA'].value_counts()\n",
      "\n",
      "# Encontrar a loja com o maior número de vendas\n",
      "loja_mais_vendida = vendas_por_loja.idxmax()\n",
      "quantidade_vendas = vendas_por_loja.max()\n",
      "\n",
      "print(\"Loja mais vendida:\", loja_mais_vendida)\n",
      "print(\"Quantidade de vendas:\", quantidade_vendas)\n"
     ],
     "metadata": {
      "colab": {
       "base_uri": "https://localhost:8080/"
      },
      "id": "PgKel2ZuIR30",
      "outputId": "cfff8f76-7f1f-458f-8dab-f0f8b1141a71"
     },
     "execution_count": null,
```

```
  "outputs": [
   {
    "output_type": "stream",
    "name": "stdout",
    "text": [
     "Loja mais vendida: LJ001\n",
     "Quantidade de vendas: 2\n"
    ]
   }
  ]
 },
 {
  "cell_type": "code",
  "source": [
   "df_compravenda"
  ],
  "metadata": {
   "colab": {
    "base_uri": "https://localhost:8080/",
    "height": 250
   },
   "id": "LcflvkALEjos",
   "outputId": "5518237f-9880-40c9-d41c-031885e291e0"
  },
  "execution_count": null,
  "outputs": [
   {
    "output_type": "execute_result",
    "data": {
     "text/plain": [
      "    COD_COMPRAVENDA       DATAS  VALOR_TOTAL  VALOR DESCONTO COD_VENDEDOR  \\\n",
      "0          CV001  2023-05-17       150.99  129.99      21.0       VD001   \n",
      "1          CV002  2023-05-18        75.50   75.50       0.0       VD002   \n",
      "2          CV003  2023-05-19       200.00  200.00       0.0       VD003   \n",
      "3          CV004  2023-05-20       500.00  400.00     100.0       VD002   \n",
```

```
  "4          CV005  2023-05-21      1000.00  950.00      50.0        VD001   \n",
  "\n",
  "      CPF_CLIENTE COD_LOJA COD_PRODUTO  \n",
  "0  123.456.789-00    LJ001        P001  \n",
  "1  987.654.321-00    LJ002        P002  \n",
  "2  987.654.321-00    LJ001        P003  \n",
  "3  111.222.333-44    LJ002        P004  \n",
  "4  555.444.333-22    LJ003        P005  "
 ],
 "text/html": [
  "\n",
  "  <div id=\"df-d4d44dc9-3b23-4ac5-b2e1-88eb28f873b3\">\n",
  "    <div class=\"colab-df-container\">\n",
  "      <div>\n",
  "<style scoped>\n",
  "    .dataframe tbody tr th:only-of-type {\n",
  "        vertical-align: middle;\n",
  "    }\n",
  "\n",
  "    .dataframe tbody tr th {\n",
  "        vertical-align: top;\n",
  "    }\n",
  "\n",
  "    .dataframe thead th {\n",
  "        text-align: right;\n",
  "    }\n",
  "</style>\n",
  "<table border=\"1\" class=\"dataframe\">\n",
  "  <thead>\n",
  "    <tr style=\"text-align: right;\">\n",
  "      <th></th>\n",
  "      <th>COD_COMPRAVENDA</th>\n",
  "      <th>DATAS</th>\n",
  "      <th>VALOR_TOTAL</th>\n",
  "      <th>VALOR</th>\n",
  "      <th>DESCONTO</th>\n",
```

```
"      <th>COD_VENDEDOR</th>\n",
"      <th>CPF_CLIENTE</th>\n",
"      <th>COD_LOJA</th>\n",
"      <th>COD_PRODUTO</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>CV001</td>\n",
"      <td>2023-05-17</td>\n",
"      <td>150.99</td>\n",
"      <td>129.99</td>\n",
"      <td>21.0</td>\n",
"      <td>VD001</td>\n",
"      <td>123.456.789-00</td>\n",
"      <td>LJ001</td>\n",
"      <td>P001</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>CV002</td>\n",
"      <td>2023-05-18</td>\n",
"      <td>75.50</td>\n",
"      <td>75.50</td>\n",
"      <td>0.0</td>\n",
"      <td>VD002</td>\n",
"      <td>987.654.321-00</td>\n",
"      <td>LJ002</td>\n",
"      <td>P002</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>CV003</td>\n",
"      <td>2023-05-19</td>\n",
"      <td>200.00</td>\n",
```

```
"      <td>200.00</td>\n",
"      <td>0.0</td>\n",
"      <td>VD003</td>\n",
"      <td>987.654.321-00</td>\n",
"      <td>LJ001</td>\n",
"      <td>P003</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>CV004</td>\n",
"      <td>2023-05-20</td>\n",
"      <td>500.00</td>\n",
"      <td>400.00</td>\n",
"      <td>100.0</td>\n",
"      <td>VD002</td>\n",
"      <td>111.222.333-44</td>\n",
"      <td>LJ002</td>\n",
"      <td>P004</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>4</th>\n",
"      <td>CV005</td>\n",
"      <td>2023-05-21</td>\n",
"      <td>1000.00</td>\n",
"      <td>950.00</td>\n",
"      <td>50.0</td>\n",
"      <td>VD001</td>\n",
"      <td>555.444.333-22</td>\n",
"      <td>LJ003</td>\n",
"      <td>P005</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
"          <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-d4d44dc9-3b23-4ac5-b2e1-88eb28f873b3')\"\n",
```

```
"            title=\"Convert this dataframe to an interactive table.\"\n",
"            style=\"display:none;\">\n",
"       \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",
"       width=\"24px\">\n",
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"      <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
"  </svg>\n",
"     </button>\n",
"     \n",
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
"\n",
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
```

```
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
"    }\n",
"  </style>\n",
"\n",
"      <script>\n",
"        const buttonEl =\n",
"          document.querySelector('#df-d4d44dc9-3b23-4ac5-b2e1-88eb28f873b3 button.colab-df-convert');\n",
"        buttonEl.style.display =\n",
"          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
"\n",
"        async function convertToInteractive(key) {\n",
"        const element = document.querySelector('#df-d4d44dc9-3b23-4ac5-b2e1-88eb28f873b3');\n",
"          const dataTable =\n",
"            await google.colab.kernel.invokeFunction('convertToInteractive',\n",
"                                                     [key], {});\n",
"          if (!dataTable) return;\n",
"\n",
"          const docLinkHtml = 'Like what you see? Visit the ' +\n",
"                              '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebook</a>'\n",
```

```
        "           + ' to learn more about interactive tables.';\n",
        "          element.innerHTML = '';\n",
        "          dataTable['output_type'] = 'display_data';\n",
        "          await google.colab.output.renderOutput(dataTable, element);\n",
        "          const docLink = document.createElement('div');\n",
        "          docLink.innerHTML = docLinkHtml;\n",
        "          element.appendChild(docLink);\n",
        "        }\n",
        "      </script>\n",
        "    </div>\n",
        "  </div>\n",
        "  "
       ]
      },
      "metadata": {},
      "execution_count": 6
    }
   ]
  },
  {
   "cell_type": "markdown",
   "source": [
    "**Q003 Listar todos os vendedores cadastrados no sistema**"
   ],
   "metadata": {
    "id": "iVbUBtSBLHRf"
   }
  },
  {
   "cell_type": "code",
   "source": [
    "vendedores_cadastrados = df_vendedor['NOME'].head()\n",
    "vendedores_cadastrados"
   ],
   "metadata": {
    "colab": {
```

```
      "base_uri": "https://localhost:8080/"
    },
    "id": "xDl7FWWDLRg2",
    "outputId": "690c643e-9ef4-4709-e556-6f51bad846a3"
   },
   "execution_count": null,
   "outputs": [
    {
     "output_type": "execute_result",
     "data": {
      "text/plain": [
       "0        João Silva\n",
       "1      Maria Santos\n",
       "2    Pedro Oliveira\n",
       "3         Ana Souza\n",
       "4     Carlos Mendes\n",
       "Name: NOME, dtype: object"
      ]
     },
     "metadata": {},
     "execution_count": 16
    }
   ]
  },
  {
   "cell_type": "markdown",
   "source": [
    "**Q004 Encontrar o vendedor que mais vendeu no último mês**"
   ],
   "metadata": {
    "id": "S3GcM-XXLjsY"
   }
  },
  {
   "cell_type": "code",
   "source": [
```

```
    "maisvend = df_vendedor['COD_VENDEDOR'][0]\n",
    "maisvend"
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/",
     "height": 35
    },
    "id": "aI_BNOgvLipR",
    "outputId": "c68f2bd4-2a65-4f1b-c1cc-79f838b8146b"
   },
   "execution_count": null,
   "outputs": [
    {
     "output_type": "execute_result",
     "data": {
      "text/plain": [
       "'VD001'"
      ],
      "application/vnd.google.colaboratory.intrinsic+json": {
       "type": "string"
      }
     },
     "metadata": {},
     "execution_count": 22
    }
   ]
  },
  {
   "cell_type": "code",
   "source": [
    "df_vendedor"
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/",
```

```
      "height": 206
     },
     "id": "XbW4hef5L4Q7",
     "outputId": "f1c4aee6-d81e-41ed-fe37-98d0aebd05d1"
    },
    "execution_count": null,
    "outputs": [
     {
      "output_type": "execute_result",
      "data": {
       "text/plain": [
        "   COD_VENDEDOR   SALARIO  SEXO   COMISSAO              NOME COD_LOJA\n",
        "0       VD001   3000.0    M      0.00      João Silva    LJ001\n",
        "1       VD002   2500.0    F      0.05    Maria Santos    LJ002\n",
        "2       VD003   4000.0    M      0.00  Pedro Oliveira    LJ003\n",
        "3       VD004   2800.0    F      0.03       Ana Souza    LJ001\n",
        "4       VD005   3500.0    M      0.02   Carlos Mendes    LJ002"
       ],
       "text/html": [
        "\n",
        "  <div id=\"df-250cf88b-064c-4e2f-83d7-5dd043096049\">\n",
        "    <div class=\"colab-df-container\">\n",
        "      <div>\n",
        "<style scoped>\n",
        "    .dataframe tbody tr th:only-of-type {\n",
        "        vertical-align: middle;\n",
        "    }\n",
        "\n",
        "    .dataframe tbody tr th {\n",
        "        vertical-align: top;\n",
        "    }\n",
        "\n",
        "    .dataframe thead th {\n",
        "        text-align: right;\n",
        "    }\n",
```

```
"</style>\n",
"<table border=\"1\" class=\"dataframe\">\n",
"  <thead>\n",
"    <tr style=\"text-align: right;\">\n",
"      <th></th>\n",
"      <th>COD_VENDEDOR</th>\n",
"      <th>SALARIO</th>\n",
"      <th>SEXO</th>\n",
"      <th>COMISSAO</th>\n",
"      <th>NOME</th>\n",
"      <th>COD_LOJA</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>VD001</td>\n",
"      <td>3000.0</td>\n",
"      <td>M</td>\n",
"      <td>0.00</td>\n",
"      <td>João Silva</td>\n",
"      <td>LJ001</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>VD002</td>\n",
"      <td>2500.0</td>\n",
"      <td>F</td>\n",
"      <td>0.05</td>\n",
"      <td>Maria Santos</td>\n",
"      <td>LJ002</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>VD003</td>\n",
"      <td>4000.0</td>\n",
```

```
"      <td>M</td>\n",
"      <td>0.00</td>\n",
"      <td>Pedro Oliveira</td>\n",
"      <td>LJ003</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>VD004</td>\n",
"      <td>2800.0</td>\n",
"      <td>F</td>\n",
"      <td>0.03</td>\n",
"      <td>Ana Souza</td>\n",
"      <td>LJ001</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>4</th>\n",
"      <td>VD005</td>\n",
"      <td>3500.0</td>\n",
"      <td>M</td>\n",
"      <td>0.02</td>\n",
"      <td>Carlos Mendes</td>\n",
"      <td>LJ002</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
"          <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-250cf88b-064c-4e2f-83d7-5dd043096049')\"\n",
"            title=\"Convert this dataframe to an interactive table.\"\n",
"            style=\"display:none;\">\n",
"       \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",
"      width=\"24px\">\n",
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"      <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10
```

```
10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
"  </svg>\n",
"    </button>\n",
"    \n",
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
"\n",
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
```

```
"\n",
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
"    }\n",
"  </style>\n",
"\n",
"      <script>\n",
"        const buttonEl =\n",
"          document.querySelector('#df-250cf88b-064c-4e2f-83d7-5dd043096049 button.colab-df-convert');\n",
"        buttonEl.style.display =\n",
"          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
"\n",
"        async function convertToInteractive(key) {\n",
"         const element = document.querySelector('#df-250cf88b-064c-4e2f-83d7-5dd043096049');\n",
"          const dataTable =\n",
"            await google.colab.kernel.invokeFunction('convertToInteractive',\n",
"                                                     [key], {});\n",
"          if (!dataTable) return;\n",
"\n",
"          const docLinkHtml = 'Like what you see? Visit the ' +\n",
"                              '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebook</a>'\n",
"            + ' to learn more about interactive tables.';\n",
"          element.innerHTML = '';\n",
"          dataTable['output_type'] = 'display_data';\n",
"          await google.colab.output.renderOutput(dataTable, element);\n",
"          const docLink = document.createElement('div');\n",
"          docLink.innerHTML = docLinkHtml;\n",
"          element.appendChild(docLink);\n",
"        }\n",
"      </script>\n",
```

```
            "    </div>\n",
            "  </div>\n",
            "  "
          ]
        },
        "metadata": {},
        "execution_count": 20
      }
    ]
  },
  {
    "cell_type": "markdown",
    "source": [
      "**Q005 Listar todas as vendas realizadas no sistema**"
    ],
    "metadata": {
      "id": "8isinGBwMMvv"
    }
  },
  {
    "cell_type": "code",
    "source": [
      "vendasreal = df_compravenda[['VALOR', 'COD_VENDEDOR','VALOR_TOTAL', 'DESCONTO']]\n",
      "vendasreal.head()\n"
    ],
    "metadata": {
      "colab": {
        "base_uri": "https://localhost:8080/",
        "height": 206
      },
      "id": "OSg4MeKpMTsb",
      "outputId": "497190fd-7d4a-4303-c314-30f36644f0b6"
    },
    "execution_count": null,
    "outputs": [
```

```
{
 "output_type": "execute_result",
 "data": {
  "text/plain": [
   "    VALOR COD_VENDEDOR  VALOR_TOTAL  DESCONTO\n",
   "0  129.99        VD001       150.99      21.0\n",
   "1   75.50        VD002        75.50       0.0\n",
   "2  200.00        VD003       200.00       0.0\n",
   "3  400.00        VD002       500.00     100.0\n",
   "4  950.00        VD001      1000.00      50.0"
  ],
  "text/html": [
   "\n",
   "  <div id=\"df-37216ef0-eaff-4c31-85ab-eba9fe569823\">\n",
   "    <div class=\"colab-df-container\">\n",
   "      <div>\n",
   "<style scoped>\n",
   "    .dataframe tbody tr th:only-of-type {\n",
   "        vertical-align: middle;\n",
   "    }\n",
   "\n",
   "    .dataframe tbody tr th {\n",
   "        vertical-align: top;\n",
   "    }\n",
   "\n",
   "    .dataframe thead th {\n",
   "        text-align: right;\n",
   "    }\n",
   "</style>\n",
   "<table border=\"1\" class=\"dataframe\">\n",
   "  <thead>\n",
   "    <tr style=\"text-align: right;\">\n",
   "      <th></th>\n",
   "      <th>VALOR</th>\n",
   "      <th>COD_VENDEDOR</th>\n",
   "      <th>VALOR_TOTAL</th>\n",
```

```
"      <th>DESCONTO</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>129.99</td>\n",
"      <td>VD001</td>\n",
"      <td>150.99</td>\n",
"      <td>21.0</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>75.50</td>\n",
"      <td>VD002</td>\n",
"      <td>75.50</td>\n",
"      <td>0.0</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>200.00</td>\n",
"      <td>VD003</td>\n",
"      <td>200.00</td>\n",
"      <td>0.0</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>400.00</td>\n",
"      <td>VD002</td>\n",
"      <td>500.00</td>\n",
"      <td>100.0</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>4</th>\n",
"      <td>950.00</td>\n",
"      <td>VD001</td>\n",
```

```
"      <td>1000.00</td>\n",
"      <td>50.0</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
"          <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-37216ef0-eaff-4c31-85ab-eba9fe569823')\"\n",
"            title=\"Convert this dataframe to an interactive table.\"\n",
"            style=\"display:none;\">\n",
"       \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",
"      width=\"24px\">\n",
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"      <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
"  </svg>\n",
"      </button>\n",
"      \n",
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
```

```
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
"\n",
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
"    }\n",
"  </style>\n",
"\n",
"      <script>\n",
"        const buttonEl =\n",
"            document.querySelector('#df-37216ef0-eaff-4c31-85ab-eba9fe569823 button.colab-df-convert');\n",
"        buttonEl.style.display =\n",
"          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
"\n",
"        async function convertToInteractive(key) {\n",
"          const element = document.querySelector('#df-37216ef0-eaff-4c31-85ab-eba9fe569823');\n",
"          const dataTable =\n",
"            await google.colab.kernel.invokeFunction('convertToInteractive',\n",
```

```
          "                                        [key], {});\n",
          "          if (!dataTable) return;\n",
          "\n",
          "          const docLinkHtml = 'Like what you see? Visit the ' +\n",
          "                                                  '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebook</a>'\n",
          "            + ' to learn more about interactive tables.';\n",
          "          element.innerHTML = '';\n",
          "          dataTable['output_type'] = 'display_data';\n",
          "          await google.colab.output.renderOutput(dataTable, element);\n",
          "          const docLink = document.createElement('div');\n",
          "          docLink.innerHTML = docLinkHtml;\n",
          "          element.appendChild(docLink);\n",
          "        }\n",
          "      </script>\n",
          "    </div>\n",
          "  </div>\n",
          "  "
        ]
      },
      "metadata": {},
      "execution_count": 27
    }
  ]
},
{
  "cell_type": "markdown",
  "source": [
    "\n",
    "\n",
    "**Q006 Encontrar as vendas realizadas entre duas datas específicas**\n"
  ],
  "metadata": {
    "id": "C_7Cuz2HNYXS"
  }
```

```
  },
  {
   "cell_type": "code",
   "source": [
    "df_vendas = df_compravenda.loc[(df_compravenda['DATAS'] >= '2023-05-18') &
(df_compravenda['DATAS'] <= '2023-05-20')]"
   ],
   "metadata": {
    "id": "c_xgUgDxNay-"
   },
   "execution_count": null,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
    "df_vendas"
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/",
     "height": 187
    },
    "id": "F6mJYUyuNpZA",
    "outputId": "dd2a2822-b802-4644-abf8-7ab586daf0bd"
   },
   "execution_count": null,
   "outputs": [
    {
     "output_type": "execute_result",
     "data": {
      "text/plain": [
       "   COD_COMPRAVENDA      DATAS  VALOR_TOTAL  VALOR
DESCONTO COD_VENDEDOR  \\\n",
       "1          CV002 2023-05-18       75.5   75.5       0.0       VD002   \n",
       "2          CV003 2023-05-19      200.0  200.0       0.0       VD003   \n",
```

```
  "3          CV004 2023-05-20       500.0  400.0     100.0       VD002   \n",
  "\n",
  "     CPF_CLIENTE COD_LOJA COD_PRODUTO  \n",
  "1  987.654.321-00    LJ002       P002  \n",
  "2  987.654.321-00    LJ001       P003  \n",
  "3  111.222.333-44    LJ002       P004  "
 ],
 "text/html": [
  "\n",
  "  <div id=\"df-c6873a65-9e31-4b76-87fa-562814367b8c\">\n",
  "    <div class=\"colab-df-container\">\n",
  "      <div>\n",
  "<style scoped>\n",
  "    .dataframe tbody tr th:only-of-type {\n",
  "        vertical-align: middle;\n",
  "    }\n",
  "\n",
  "    .dataframe tbody tr th {\n",
  "        vertical-align: top;\n",
  "    }\n",
  "\n",
  "    .dataframe thead th {\n",
  "        text-align: right;\n",
  "    }\n",
  "</style>\n",
  "<table border=\"1\" class=\"dataframe\">\n",
  "  <thead>\n",
  "    <tr style=\"text-align: right;\">\n",
  "      <th></th>\n",
  "      <th>COD_COMPRAVENDA</th>\n",
  "      <th>DATAS</th>\n",
  "      <th>VALOR_TOTAL</th>\n",
  "      <th>VALOR</th>\n",
  "      <th>DESCONTO</th>\n",
  "      <th>COD_VENDEDOR</th>\n",
  "      <th>CPF_CLIENTE</th>\n",
```

```
"      <th>COD_LOJA</th>\n",
"      <th>COD_PRODUTO</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>CV002</td>\n",
"      <td>2023-05-18</td>\n",
"      <td>75.5</td>\n",
"      <td>75.5</td>\n",
"      <td>0.0</td>\n",
"      <td>VD002</td>\n",
"      <td>987.654.321-00</td>\n",
"      <td>LJ002</td>\n",
"      <td>P002</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>CV003</td>\n",
"      <td>2023-05-19</td>\n",
"      <td>200.0</td>\n",
"      <td>200.0</td>\n",
"      <td>0.0</td>\n",
"      <td>VD003</td>\n",
"      <td>987.654.321-00</td>\n",
"      <td>LJ001</td>\n",
"      <td>P003</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>CV004</td>\n",
"      <td>2023-05-20</td>\n",
"      <td>500.0</td>\n",
"      <td>400.0</td>\n",
"      <td>100.0</td>\n",
```

```
"      <td>VD002</td>\n",
"      <td>111.222.333-44</td>\n",
"      <td>LJ002</td>\n",
"      <td>P004</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
"          <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-c6873a65-9e31-4b76-87fa-562814367b8c')\"\n",
"            title=\"Convert this dataframe to an interactive table.\"\n",
"            style=\"display:none;\">\n",
"       \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",
"      width=\"24px\">\n",
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"      <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
"  </svg>\n",
"      </button>\n",
"      \n",
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
```

```
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
"\n",
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
"    }\n",
"  </style>\n",
"\n",
"      <script>\n",
"        const buttonEl =\n",
"          document.querySelector('#df-c6873a65-9e31-4b76-87fa-562814367b8c button.colab-df-convert');\n",
"        buttonEl.style.display =\n",
"          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
"\n",
"        async function convertToInteractive(key) {\n",
"          const element = document.querySelector('#df-c6873a65-9e31-4b76-87fa-562814367b8c');\n",
```

```
          "         const dataTable =\n",
          "           await google.colab.kernel.invokeFunction('convertToInteractive',\n",
          "                                                    [key], {});\n",
          "         if (!dataTable) return;\n",
          "\n",
          "         const docLinkHtml = 'Like what you see? Visit the ' +\n",
          "                                                              '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebook</a>'\n",
          "           + ' to learn more about interactive tables.';\n",
          "         element.innerHTML = '';\n",
          "         dataTable['output_type'] = 'display_data';\n",
          "         await google.colab.output.renderOutput(dataTable, element);\n",
          "         const docLink = document.createElement('div');\n",
          "         docLink.innerHTML = docLinkHtml;\n",
          "         element.appendChild(docLink);\n",
          "       }\n",
          "     </script>\n",
          "   </div>\n",
          " </div>\n",
          " "
        ]
       },
       "metadata": {},
       "execution_count": 31
      }
     ]
    },
    {
     "cell_type": "code",
     "source": [
      "df_produto"
     ],
     "metadata": {
      "colab": {
       "base_uri": "https://localhost:8080/",
```

```
      "height": 206
     },
     "id": "EC1cl9zBpOOp",
     "outputId": "e50db3f0-42bb-438b-fe97-5661c759663d"
    },
    "execution_count": null,
    "outputs": [
     {
      "output_type": "execute_result",
      "data": {
       "text/plain": [
        "   COD_PRODUTO   CUSTO   PRECO_VENDA                        NOME
CATEGORIA  \\\n",
        "0       P001  700.0       999.0              iPhone 13  Eletrônicos   \n",
        "1       P002  800.0      1099.0            Smart TV 4K  Eletrônicos   \n",
        "2       P003   50.0        79.0  Fone de Ouvido Bluetooth  Eletrônicos   \n",
        "3       P004  200.0       299.0       Tablet Galaxy Tab A  Eletrônicos   \n",
        "4       P005  300.0       449.0          Câmera Digital  Eletrônicos   \n",
        "\n",
        "     MARCA       COR  \n",
        "0    Apple     Preto  \n",
        "1  Samsung     Prata  \n",
        "2     Sony  Vermelho  \n",
        "3  Samsung     Preto  \n",
        "4    Canon    Branco  "
       ],
       "text/html": [
        "\n",
        "  <div id=\"df-77332f07-1bae-4cfe-b6b1-a173a0ddebe7\">\n",
        "    <div class=\"colab-df-container\">\n",
        "      <div>\n",
        "<style scoped>\n",
        "    .dataframe tbody tr th:only-of-type {\n",
        "        vertical-align: middle;\n",
        "    }\n",
        "\n",
```

```
"    .dataframe tbody tr th {\n",
"        vertical-align: top;\n",
"    }\n",
"\n",
"    .dataframe thead th {\n",
"        text-align: right;\n",
"    }\n",
"</style>\n",
"<table border=\"1\" class=\"dataframe\">\n",
"  <thead>\n",
"    <tr style=\"text-align: right;\">\n",
"      <th></th>\n",
"      <th>COD_PRODUTO</th>\n",
"      <th>CUSTO</th>\n",
"      <th>PRECO_VENDA</th>\n",
"      <th>NOME</th>\n",
"      <th>CATEGORIA</th>\n",
"      <th>MARCA</th>\n",
"      <th>COR</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>P001</td>\n",
"      <td>700.0</td>\n",
"      <td>999.0</td>\n",
"      <td>iPhone 13</td>\n",
"      <td>Eletrônicos</td>\n",
"      <td>Apple</td>\n",
"      <td>Preto</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>P002</td>\n",
"      <td>800.0</td>\n",
```

```
"    <td>1099.0</td>\n",
"    <td>Smart TV 4K</td>\n",
"    <td>Eletrônicos</td>\n",
"    <td>Samsung</td>\n",
"    <td>Prata</td>\n",
"  </tr>\n",
"  <tr>\n",
"    <th>2</th>\n",
"    <td>P003</td>\n",
"    <td>50.0</td>\n",
"    <td>79.0</td>\n",
"    <td>Fone de Ouvido Bluetooth</td>\n",
"    <td>Eletrônicos</td>\n",
"    <td>Sony</td>\n",
"    <td>Vermelho</td>\n",
"  </tr>\n",
"  <tr>\n",
"    <th>3</th>\n",
"    <td>P004</td>\n",
"    <td>200.0</td>\n",
"    <td>299.0</td>\n",
"    <td>Tablet Galaxy Tab A</td>\n",
"    <td>Eletrônicos</td>\n",
"    <td>Samsung</td>\n",
"    <td>Preto</td>\n",
"  </tr>\n",
"  <tr>\n",
"    <th>4</th>\n",
"    <td>P005</td>\n",
"    <td>300.0</td>\n",
"    <td>449.0</td>\n",
"    <td>Câmera Digital</td>\n",
"    <td>Eletrônicos</td>\n",
"    <td>Canon</td>\n",
"    <td>Branco</td>\n",
"  </tr>\n",
```

```
"  </tbody>\n",
"</table>\n",
"</div>\n",
"          <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-77332f07-1bae-4cfe-b6b1-a173a0ddebe7')\"\n",
"            title=\"Convert this dataframe to an interactive table.\"\n",
"            style=\"display:none;\">\n",
"       \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",
"      width=\"24px\">\n",
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"      <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
"  </svg>\n",
"      </button>\n",
"      \n",
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
```

```
"    }\n",
"\n",
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
"    }\n",
"  </style>\n",
"\n",
"      <script>\n",
"        const buttonEl =\n",
"          document.querySelector('#df-77332f07-1bae-4cfe-b6b1-a173a0ddebe7 button.colab-df-convert');\n",
"        buttonEl.style.display =\n",
"          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
"\n",
"        async function convertToInteractive(key) {\n",
"          const element = document.querySelector('#df-77332f07-1bae-4cfe-b6b1-a173a0ddebe7');\n",
"          const dataTable =\n",
"            await google.colab.kernel.invokeFunction('convertToInteractive',\n",
"                                                     [key], {});\n",
"          if (!dataTable) return;\n",
"\n",
```

```
          "          const docLinkHtml = 'Like what you see? Visit the ' +\n",
          "                                                  '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebook</a>'\n",
          "            + ' to learn more about interactive tables.';\n",
          "          element.innerHTML = '';\n",
          "          dataTable['output_type'] = 'display_data';\n",
          "          await google.colab.output.renderOutput(dataTable, element);\n",
          "          const docLink = document.createElement('div');\n",
          "          docLink.innerHTML = docLinkHtml;\n",
          "          element.appendChild(docLink);\n",
          "        }\n",
          "      </script>\n",
          "    </div>\n",
          "  </div>\n",
          "  "
         ]
        },
        "metadata": {},
        "execution_count": 5
      }
    ]
  },
  {
   "cell_type": "markdown",
   "source": [
    "**Q007 NOVA: Inserir um novo produto na tabela proutos**"
   ],
   "metadata": {
    "id": "dD1PHCzwjW95"
   }
  },
  {
   "cell_type": "code",
   "source": [
    "#Testando outra maneira\n",
```

```
"# Criar um dataframe vazio para armazenar os produtos\n",
"df_produtos = pd.DataFrame(columns=['COD_PRODUTO', 'CUSTO', 'PRECO_VENDA', 'NOME', 'CATEGORIA', 'MARCA', 'COR'])\n",
"\n",
"# Solicitar as informações do novo produto ao usuário\n",
"cod_produto = input(\"Informe o código do produto: \")\n",
"custo = float(input(\"Informe o custo do produto: \"))\n",
"preco_venda = float(input(\"Informe o preço de venda do produto: \"))\n",
"nome = input(\"Informe o nome do produto: \")\n",
"categoria = input(\"Informe a categoria do produto: \")\n",
"marca = input(\"Informe a marca do produto: \")\n",
"cor = input(\"Informe a cor do produto: \")\n",
"\n",
"# Adicionar o novo produto ao dataframe 'produtos'\n",
"df_produtos.loc[len(df_produtos)] = [cod_produto, custo, preco_venda, nome, categoria, marca, cor]\n",
"\n",
"# Imprimir o dataframe 'produtos' atualizado\n",
"df_produtos.head()\n"
],
"metadata": {
"colab": {
"base_uri": "https://localhost:8080/",
"height": 202
},
"id": "IwW2_1MVwk6D",
"outputId": "76c933af-2cf0-494b-e09a-21e5aa384ef5"
},
"execution_count": 3,
"outputs": [
{
"name": "stdout",
"output_type": "stream",
"text": [
"Informe o código do produto: P010\n",
"Informe o custo do produto: 900.00\n",
```

```
      "Informe o preço de venda do produto: 1200.00\n",
      "Informe o nome do produto: iPhone 14'\n",
      "Informe a categoria do produto: Eletrônicos\n",
      "Informe a marca do produto: Apple\n",
      "Informe a cor do produto: Cinza\n"
     ]
    },
    {
     "output_type": "execute_result",
     "data": {
      "text/plain": [
       "  COD_PRODUTO  CUSTO  PRECO_VENDA       NOME    CATEGORIA  MARCA    COR\n",
       "0       P010  900.0       1200.0  iPhone 14'  Eletrônicos  Apple  Cinza"
      ],
      "text/html": [
       "\n",
       "  <div id=\"df-f8230a81-5144-435a-9941-6a35535bc622\">\n",
       "    <div class=\"colab-df-container\">\n",
       "      <div>\n",
       "<style scoped>\n",
       "    .dataframe tbody tr th:only-of-type {\n",
       "        vertical-align: middle;\n",
       "    }\n",
       "\n",
       "    .dataframe tbody tr th {\n",
       "        vertical-align: top;\n",
       "    }\n",
       "\n",
       "    .dataframe thead th {\n",
       "        text-align: right;\n",
       "    }\n",
       "</style>\n",
       "<table border=\"1\" class=\"dataframe\">\n",
       "  <thead>\n",
       "    <tr style=\"text-align: right;\">\n",
```

```
"      <th></th>\n",
"      <th>COD_PRODUTO</th>\n",
"      <th>CUSTO</th>\n",
"      <th>PRECO_VENDA</th>\n",
"      <th>NOME</th>\n",
"      <th>CATEGORIA</th>\n",
"      <th>MARCA</th>\n",
"      <th>COR</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>P010</td>\n",
"      <td>900.0</td>\n",
"      <td>1200.0</td>\n",
"      <td>iPhone 14'</td>\n",
"      <td>Eletrônicos</td>\n",
"      <td>Apple</td>\n",
"      <td>Cinza</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
"          <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-
f8230a81-5144-435a-9941-6a35535bc622')\"\n",
"            title=\"Convert this dataframe to an interactive table.\"\n",
"            style=\"display:none;\">\n",
"       \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0
24 24\"\n",
"      width=\"24px\">\n",
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"      <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94
2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10
10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41
7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-
```

```
.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
"  </svg>\n",
"      </button>\n",
"      \n",
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
"\n",
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
```

```
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
"    }\n",
"  </style>\n",
"\n",
"      <script>\n",
"        const buttonEl =\n",
"          document.querySelector('#df-f8230a81-5144-435a-9941-6a35535bc622 button.colab-df-convert');\n",
"        buttonEl.style.display =\n",
"          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
"\n",
"        async function convertToInteractive(key) {\n",
"        const element = document.querySelector('#df-f8230a81-5144-435a-9941-6a35535bc622');\n",
"          const dataTable =\n",
"            await google.colab.kernel.invokeFunction('convertToInteractive',\n",
"                                                     [key], {});\n",
"          if (!dataTable) return;\n",
"\n",
"          const docLinkHtml = 'Like what you see? Visit the ' +\n",
"                              '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebook</a>'\n",
"            + ' to learn more about interactive tables.';\n",
"          element.innerHTML = '';\n",
"          dataTable['output_type'] = 'display_data';\n",
"          await google.colab.output.renderOutput(dataTable, element);\n",
"          const docLink = document.createElement('div');\n",
"          docLink.innerHTML = docLinkHtml;\n",
"          element.appendChild(docLink);\n",
"        }\n",
"      </script>\n",
"    </div>\n",
```

```
          "  </div>\n",
          "  "
         ]
        },
        "metadata": {},
        "execution_count": 3
       }
      ]
     },
     {
      "cell_type": "markdown",
      "source": [
       "insira esses itens: \n",
       "P010 \t\n",
       "900.0 \n",
       "1200.0 \t\n",
       "iPhone 14 \n",
       "Eletrônicos \n",
       "Apple \t\n",
       "Cinza"
      ],
      "metadata": {
       "id": "urZxEJ8skApO"
      }
     },
     {
      "cell_type": "code",
      "source": [
       "# Realizar o merge das tabelas depois de inserir as informações, para guardar as informações juntas\n",
       "df_merged = pd.concat([df_produto, df_produtos], ignore_index=True)"
      ],
      "metadata": {
       "id": "O7iSBmJ5wNa2"
      },
      "execution_count": 4,
```

```
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
    "df_merged "
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/",
     "height": 238
    },
    "id": "pzZmx3OixYk4",
    "outputId": "66150343-d13b-4149-ae1b-5cb2f95588fa"
   },
   "execution_count": 5,
   "outputs": [
    {
     "output_type": "execute_result",
     "data": {
      "text/plain": [
       "   COD_PRODUTO   CUSTO   PRECO_VENDA                          NOME
CATEGORIA  \\\n",
       "0       P001  700.0       999.0              iPhone 13  Eletrônicos   \n",
       "1       P002  800.0      1099.0            Smart TV 4K  Eletrônicos   \n",
       "2       P003   50.0        79.0  Fone de Ouvido Bluetooth  Eletrônicos   \n",
       "3       P004  200.0       299.0      Tablet Galaxy Tab A  Eletrônicos   \n",
       "4       P005  300.0       449.0         Câmera Digital  Eletrônicos   \n",
       "5       P010  900.0      1200.0             iPhone 14'  Eletrônicos   \n",
       "\n",
       "     MARCA       COR  \n",
       "0    Apple     Preto  \n",
       "1  Samsung     Prata  \n",
       "2     Sony  Vermelho  \n",
       "3  Samsung     Preto  \n",
       "4    Canon    Branco  \n",
```

```
  "5    Apple     Cinza  "
 ],
 "text/html": [
  "\n",
  "  <div id=\"df-4e325fcb-f132-44c0-977c-77f3c82573c6\">\n",
  "    <div class=\"colab-df-container\">\n",
  "      <div>\n",
  "<style scoped>\n",
  "    .dataframe tbody tr th:only-of-type {\n",
  "        vertical-align: middle;\n",
  "    }\n",
  "\n",
  "    .dataframe tbody tr th {\n",
  "        vertical-align: top;\n",
  "    }\n",
  "\n",
  "    .dataframe thead th {\n",
  "        text-align: right;\n",
  "    }\n",
  "</style>\n",
  "<table border=\"1\" class=\"dataframe\">\n",
  "  <thead>\n",
  "    <tr style=\"text-align: right;\">\n",
  "      <th></th>\n",
  "      <th>COD_PRODUTO</th>\n",
  "      <th>CUSTO</th>\n",
  "      <th>PRECO_VENDA</th>\n",
  "      <th>NOME</th>\n",
  "      <th>CATEGORIA</th>\n",
  "      <th>MARCA</th>\n",
  "      <th>COR</th>\n",
  "    </tr>\n",
  "  </thead>\n",
  "  <tbody>\n",
  "    <tr>\n",
  "      <th>0</th>\n",
```

```
"     <td>P001</td>\n",
"     <td>700.0</td>\n",
"     <td>999.0</td>\n",
"     <td>iPhone 13</td>\n",
"     <td>Eletrônicos</td>\n",
"     <td>Apple</td>\n",
"     <td>Preto</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>1</th>\n",
"     <td>P002</td>\n",
"     <td>800.0</td>\n",
"     <td>1099.0</td>\n",
"     <td>Smart TV 4K</td>\n",
"     <td>Eletrônicos</td>\n",
"     <td>Samsung</td>\n",
"     <td>Prata</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>2</th>\n",
"     <td>P003</td>\n",
"     <td>50.0</td>\n",
"     <td>79.0</td>\n",
"     <td>Fone de Ouvido Bluetooth</td>\n",
"     <td>Eletrônicos</td>\n",
"     <td>Sony</td>\n",
"     <td>Vermelho</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>3</th>\n",
"     <td>P004</td>\n",
"     <td>200.0</td>\n",
"     <td>299.0</td>\n",
"     <td>Tablet Galaxy Tab A</td>\n",
"     <td>Eletrônicos</td>\n",
"     <td>Samsung</td>\n",
```

```
"      <td>Preto</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>4</th>\n",
"      <td>P005</td>\n",
"      <td>300.0</td>\n",
"      <td>449.0</td>\n",
"      <td>Câmera Digital</td>\n",
"      <td>Eletrônicos</td>\n",
"      <td>Canon</td>\n",
"      <td>Branco</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>5</th>\n",
"      <td>P010</td>\n",
"      <td>900.0</td>\n",
"      <td>1200.0</td>\n",
"      <td>iPhone 14'</td>\n",
"      <td>Eletrônicos</td>\n",
"      <td>Apple</td>\n",
"      <td>Cinza</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
"          <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-4e325fcb-f132-44c0-977c-77f3c82573c6')\"\n",
"            title=\"Convert this dataframe to an interactive table.\"\n",
"            style=\"display:none;\">\n",
"       \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",
"      width=\"24px\">\n",
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"      <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41
```

```
7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
"  </svg>\n",
"    </button>\n",
"    \n",
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
"\n",
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
```

```
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
"    }\n",
"  </style>\n",
"\n",
"      <script>\n",
"        const buttonEl =\n",
"          document.querySelector('#df-4e325fcb-f132-44c0-977c-77f3c82573c6
button.colab-df-convert');\n",
"        buttonEl.style.display =\n",
"          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
"\n",
"        async function convertToInteractive(key) {\n",
"         const element = document.querySelector('#df-4e325fcb-f132-44c0-977c-
77f3c82573c6');\n",
"          const dataTable =\n",
"            await google.colab.kernel.invokeFunction('convertToInteractive',\n",
"                                                     [key], {});\n",
"          if (!dataTable) return;\n",
"\n",
"          const docLinkHtml = 'Like what you see? Visit the ' +\n",
"                                                              '<a       target=\"_blank\"
href=https://colab.research.google.com/notebooks/data_table.ipynb>data                table
notebook</a>'\n",
"            + ' to learn more about interactive tables.';\n",
"          element.innerHTML = '';\n",
"          dataTable['output_type'] = 'display_data';\n",
"          await google.colab.output.renderOutput(dataTable, element);\n",
"          const docLink = document.createElement('div');\n",
"          docLink.innerHTML = docLinkHtml;\n",
"          element.appendChild(docLink);\n",
"        }\n",
"      </script>\n",
"    </div>\n",
```

```
        "  </div>\n",
        "  "
       ]
      },
      "metadata": {},
      "execution_count": 5
    }
   ]
  },
  {
   "cell_type": "markdown",
   "source": [
    "**Q008 Listar todos os fornecedores cadastrados no sistema**"
   ],
   "metadata": {
    "id": "Op2raiJPPj-p"
   }
  },
  {
   "cell_type": "code",
   "source": [
    "fornecedores_cad = df_fornecedor['NOME'].head()\n",
    "fornecedores_cad "
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "id": "QJFi8XDePyfc",
    "outputId": "a9f63dc6-76d2-401c-888d-56316fef8240"
   },
   "execution_count": null,
   "outputs": [
    {
     "output_type": "execute_result",
     "data": {
```

```
          "text/plain": [
            "0       João Silva\n",
            "1      Maria Santos\n",
            "2    Pedro Oliveira\n",
            "3        Ana Souza\n",
            "4     Carlos Mendes\n",
            "Name: NOME, dtype: object"
          ]
        },
        "metadata": {},
        "execution_count": 22
      }
    ]
  },
  {
    "cell_type": "markdown",
    "source": [
      "**Q010 Encontrar a receita total de uma loja**"
    ],
    "metadata": {
      "id": "ZlnxqtQMug9k"
    }
  },
  {
    "cell_type": "code",
    "source": [
      "receita_total = df_compravenda[['VALOR', 'COD_LOJA']].groupby(['COD_LOJA']).sum().reset_index()"
    ],
    "metadata": {
      "id": "gXRDKvL3ue9h"
    },
    "execution_count": null,
    "outputs": []
  },
  {
```

```
"cell_type": "code",
"source": [
  "receita_total.head(5)"
],
"metadata": {
  "colab": {
    "base_uri": "https://localhost:8080/",
    "height": 143
  },
  "id": "BW1m_CiAuwh4",
  "outputId": "0c63a05c-44fa-4c46-8ced-f6aa07426ed0"
},
"execution_count": null,
"outputs": [
  {
    "output_type": "execute_result",
    "data": {
      "text/plain": [
        "  COD_LOJA    VALOR\n",
        "0    LJ001  329.99\n",
        "1    LJ002  475.50\n",
        "2    LJ003  950.00"
      ],
      "text/html": [
        "\n",
        "  <div id=\"df-22180660-8775-49ca-8f9f-1fdc1444cd37\">\n",
        "    <div class=\"colab-df-container\">\n",
        "      <div>\n",
        "<style scoped>\n",
        "    .dataframe tbody tr th:only-of-type {\n",
        "        vertical-align: middle;\n",
        "    }\n",
        "\n",
        "    .dataframe tbody tr th {\n",
        "        vertical-align: top;\n",
        "    }\n",
```

```
"\n",
"    .dataframe thead th {\n",
"        text-align: right;\n",
"    }\n",
"</style>\n",
"<table border=\"1\" class=\"dataframe\">\n",
"  <thead>\n",
"    <tr style=\"text-align: right;\">\n",
"      <th></th>\n",
"      <th>COD_LOJA</th>\n",
"      <th>VALOR</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>LJ001</td>\n",
"      <td>329.99</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>LJ002</td>\n",
"      <td>475.50</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>LJ003</td>\n",
"      <td>950.00</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
"          <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-22180660-8775-49ca-8f9f-1fdc1444cd37')\"\n",
"            title=\"Convert this dataframe to an interactive table.\"\n",
"            style=\"display:none;\">\n",
```

```
"       \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",
"       width=\"24px\">\n",
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"       <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
"  </svg>\n",
"      </button>\n",
"      \n",
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
"\n",
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
"       box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
```

```
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
"    }\n",
"  </style>\n",
"\n",
"      <script>\n",
"        const buttonEl =\n",
"          document.querySelector('#df-22180660-8775-49ca-8f9f-1fdc1444cd37
button.colab-df-convert');\n",
"        buttonEl.style.display =\n",
"          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
"\n",
"        async function convertToInteractive(key) {\n",
"         const element = document.querySelector('#df-22180660-8775-49ca-8f9f-
1fdc1444cd37');\n",
"          const dataTable =\n",
"            await google.colab.kernel.invokeFunction('convertToInteractive',\n",
"                                                     [key], {});\n",
"          if (!dataTable) return;\n",
"\n",
"          const docLinkHtml = 'Like what you see? Visit the ' +\n",
"                                                                '<a       target=\"_blank\"
href=https://colab.research.google.com/notebooks/data_table.ipynb>data                table
notebook</a>'\n",
"            + ' to learn more about interactive tables.';\n",
"          element.innerHTML = '';\n",
"          dataTable['output_type'] = 'display_data';\n",
```

```
          "          await google.colab.output.renderOutput(dataTable, element);\n",
          "          const docLink = document.createElement('div');\n",
          "          docLink.innerHTML = docLinkHtml;\n",
          "          element.appendChild(docLink);\n",
          "        }\n",
          "      </script>\n",
          "    </div>\n",
          "  </div>\n",
          "  "
         ]
        },
        "metadata": {},
        "execution_count": 33
       }
      ]
     },
     {
      "cell_type": "markdown",
      "source": [
       "**Q011 Listar vendas por vendedor**"
      ],
      "metadata": {
       "id": "E48dehFmxgoQ"
      }
     },
     {
      "cell_type": "code",
      "source": [
       "vendas_porv = df_compravenda[['COD_COMPRAVENDA', 'COD_VENDEDOR']].groupby('COD_VENDEDOR').count().reset_index()"
      ],
      "metadata": {
       "id": "13abliYCxfmO"
      },
      "execution_count": null,
      "outputs": []
```

```
},
{
 "cell_type": "code",
 "source": [
  "vendas_porv.head(5)"
 ],
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/",
   "height": 143
  },
  "id": "OJHTy4bMx9D0",
  "outputId": "3fdc1304-b93c-4649-b1eb-46f1c7879639"
 },
 "execution_count": null,
 "outputs": [
  {
   "output_type": "execute_result",
   "data": {
    "text/plain": [
     "  COD_VENDEDOR  COD_COMPRAVENDA\n",
     "0       VD001              2\n",
     "1       VD002              2\n",
     "2       VD003              1"
    ],
    "text/html": [
     "\n",
     "  <div id=\"df-c4312ac1-96ef-4f03-a621-422443c40150\">\n",
     "    <div class=\"colab-df-container\">\n",
     "      <div>\n",
     "<style scoped>\n",
     "    .dataframe tbody tr th:only-of-type {\n",
     "        vertical-align: middle;\n",
     "    }\n",
     "\n",
     "    .dataframe tbody tr th {\n",
```

```
"       vertical-align: top;\n",
"    }\n",
"\n",
"    .dataframe thead th {\n",
"       text-align: right;\n",
"    }\n",
"</style>\n",
"<table border=\"1\" class=\"dataframe\">\n",
"  <thead>\n",
"    <tr style=\"text-align: right;\">\n",
"      <th></th>\n",
"      <th>COD_VENDEDOR</th>\n",
"      <th>COD_COMPRAVENDA</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>VD001</td>\n",
"      <td>2</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>VD002</td>\n",
"      <td>2</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>VD003</td>\n",
"      <td>1</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
"          <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-
c4312ac1-96ef-4f03-a621-422443c40150')\"\n",
```

```
"            title=\"Convert this dataframe to an interactive table.\"\n",
"            style=\"display:none;\">\n",
"       \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",
"       width=\"24px\">\n",
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"      <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
"  </svg>\n",
"     </button>\n",
"     \n",
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
"\n",
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
```

```
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
"    }\n",
"  </style>\n",
"\n",
"      <script>\n",
"        const buttonEl =\n",
"          document.querySelector('#df-c4312ac1-96ef-4f03-a621-422443c40150 button.colab-df-convert');\n",
"        buttonEl.style.display =\n",
"          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
"\n",
"        async function convertToInteractive(key) {\n",
"          const element = document.querySelector('#df-c4312ac1-96ef-4f03-a621-422443c40150');\n",
"          const dataTable =\n",
"            await google.colab.kernel.invokeFunction('convertToInteractive',\n",
"                                                     [key], {});\n",
"          if (!dataTable) return;\n",
"\n",
"          const docLinkHtml = 'Like what you see? Visit the ' +\n",
"                              '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebook</a>'\n",
```

```
        "           + ' to learn more about interactive tables.';\n",
        "         element.innerHTML = '';\n",
        "         dataTable['output_type'] = 'display_data';\n",
        "         await google.colab.output.renderOutput(dataTable, element);\n",
        "         const docLink = document.createElement('div');\n",
        "         docLink.innerHTML = docLinkHtml;\n",
        "         element.appendChild(docLink);\n",
        "       }\n",
        "     </script>\n",
        "   </div>\n",
        " </div>\n",
        " "
       ]
      },
      "metadata": {},
      "execution_count": 36
    }
   ]
  },
  {
   "cell_type": "markdown",
   "source": [
    "**Q012 NOVA: Inserir um novo cliente no banco de dados**"
   ],
   "metadata": {
    "id": "o1kXLwX8yfMi"
   }
  },
  {
   "cell_type": "code",
   "source": [
    "df_novo_cliente = pd.DataFrame({\n",
    "    'CPF': [cpf],\n",
    "    'TELEFONE': [telefone],\n",
    "    'ENDERECO': [endereco],\n",
    "    'CEP': [cep],\n",
```

```
"    'ESTADO': [estado],\n",
"    'CIDADE': [cidade],\n",
"    'LOGRADOURO': [logradouro],\n",
"    'SEXO': [sexo],\n",
"    'PRIMEIRO_NOME': [primeiro_nome],\n",
"    'SOBRENOME': [sobrenome],\n",
"    'DATA_NASCIMENTO': [data_nascimento]\n",
"})\n",
"\n",
"# Inserir o novo cliente na tabela cliente\n",
"df_cliente = df_cliente.append(df_novo_cliente, ignore_index=True)\n",
"\n",
"# Ver se deu certo...\n",
"df_cliente.head()"
],
"metadata": {
"id": "yaeP4KB1yu59"
},
"execution_count": null,
"outputs": []
},
{
"cell_type": "code",
"source": [
"df_novo_cliente = pd.DataFrame(columns=['CPF', 'TELEFONE', 'ENDERECO', 'CEP', 'ESTADO', 'CIDADE', 'LOGRADOURO','SEXO', 'PRIMEIRO_NOME', 'SOBRENOME', 'DATA_NASCIMENTO' ])\n",
"\n",
"# Solicitar as informações do novo cliente ao usuário\n",
"cpf = input(\"Informe o CPF do cliente: \")\n",
"telefone = input(\"Informe o telefone do cliente: \")\n",
"endereco = input(\"Informe o endereço do cliente: \")\n",
"cep = input(\"Informe o CEP do cliente: \")\n",
"estado = input(\"Informe o estado do cliente: \")\n",
"cidade = input(\"Informe a cidade do cliente: \")\n",
"logradouro = input(\"Informe o logradouro do cliente: \")\n",
```

```
    "sexo = input(\"Informe o sexo do cliente: \")\n",
    "primeiro_nome = input(\"Informe o primeiro nome do cliente: \")\n",
    "sobrenome = input(\"Informe o sobrenome do cliente: \")\n",
    "data_nascimento = input(\"Informe a data de nascimento do cliente: \")\n",
    "\n",
    "# Adicionar o novo cliente ao dataframe 'df_cliente'\n",
    "#df_cliente.loc[len(df_cliente)] = [cpf, telefone, endereco, cep, estado, cidade, logradouro, sexo, primeiro_nome, sobrenome, data_nascimento]\n",
    "\n",
    "# Verificar se o novo cliente foi adicionado corretamente\n",
    "print(df_cliente.head())"
   ],
   "metadata": {
    "id": "FhkF2VwFufK8"
   },
   "execution_count": null,
   "outputs": []
  },
  {
   "cell_type": "markdown",
   "source": [
    "insira uma informações qualquer e ele ira guardar:\n",
    "\n",
    "\n",
    "121.232.333-44,\n",
    "(33) 4321-1234,\n",
    "Rua XYZ,\n",
    "32,\n",
    "43256-020,\n",
    "Belo Horizonte,\n",
    "BH,\n",
    "Residencial TPNB,\n",
    "H,\n",
    "Marcius,\n",
    "Cavalcante,\n",
    "1950-02-12"
```

```
    ],
    "metadata": {
     "id": "_VRUQ481kZ-_"
    }
   },
   {
    "cell_type": "code",
    "source": [
     "# Realizar o merge das tabelas depois de inserir as informações, para guardar as informações juntas\n",
     "df_cliente = pd.concat([df_cliente, df_novo_cliente], ignore_index=True)"
    ],
    "metadata": {
     "id": "kC5D6vISnsYb"
    },
    "execution_count": null,
    "outputs": []
   },
   {
    "cell_type": "code",
    "source": [
     "df_cliente"
    ],
    "metadata": {
     "colab": {
      "base_uri": "https://localhost:8080/",
      "height": 250
     },
     "id": "xNl8CJTeu3L9",
     "outputId": "93667346-6b86-45a7-ae11-35d5488523a2"
    },
    "execution_count": 9,
    "outputs": [
     {
      "output_type": "execute_result",
      "data": {
```

```
"text/plain": [
 "            CPF        TELEFONE        ENDERECO        CEP        ESTADO \\\n",
 "0  123.456.789-00  (11) 1234-5678      Rua A, 123  12345-678      São Paulo \n",
 "1  987.654.321-00  (22) 9876-5432  Avenida B, 456  98765-432  Rio de Janeiro \n",
 "2  111.222.333-44  (33) 1111-2222      Rua C, 789  54321-098    Minas Gerais \n",
 "3  555.444.333-22  (44) 5555-4444  Avenida D, 987  76543-210          Bahia \n",
 "4  777.888.999-00  (55) 7777-8888      Rua E, 321  01234-567      São Paulo \n",
 "\n",
 "                CIDADE          LOGRADOURO SEXO PRIMEIRO_NOME SOBRENOME  \\\n",
 "0       São Paulo  Residencial ABC    M         João     Silva   \n",
 "1  Rio de Janeiro  Apartamento XYZ    F        Maria     Souza   \n",
 "2  Belo Horizonte         Casa 123    M        Pedro    Santos   \n",
 "3        Salvador       Bloco ABCD    F         Ana  Oliveira   \n",
 "4       São Paulo    Edifício EFGH    M        Lucas  Ferreira   \n",
 "\n",
 "  DATA_NASCIMENTO  \n",
 "0      1990-01-01  \n",
 "1      1985-05-10  \n",
 "2      1982-12-25  \n",
 "3      1995-09-15  \n",
 "4      1988-07-20  "
],
"text/html": [
 "\n",
 "  <div id=\"df-129d1eef-9130-4503-be98-afcf42a8885e\">\n",
 "    <div class=\"colab-df-container\">\n",
 "      <div>\n",
 "<style scoped>\n",
 "    .dataframe tbody tr th:only-of-type {\n",
 "        vertical-align: middle;\n",
 "    }\n",
```

```
"\n",
"    .dataframe tbody tr th {\n",
"        vertical-align: top;\n",
"    }\n",
"\n",
"    .dataframe thead th {\n",
"        text-align: right;\n",
"    }\n",
"</style>\n",
"<table border=\"1\" class=\"dataframe\">\n",
"  <thead>\n",
"    <tr style=\"text-align: right;\">\n",
"      <th></th>\n",
"      <th>CPF</th>\n",
"      <th>TELEFONE</th>\n",
"      <th>ENDERECO</th>\n",
"      <th>CEP</th>\n",
"      <th>ESTADO</th>\n",
"      <th>CIDADE</th>\n",
"      <th>LOGRADOURO</th>\n",
"      <th>SEXO</th>\n",
"      <th>PRIMEIRO_NOME</th>\n",
"      <th>SOBRENOME</th>\n",
"      <th>DATA_NASCIMENTO</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>123.456.789-00</td>\n",
"      <td>(11) 1234-5678</td>\n",
"      <td>Rua A, 123</td>\n",
"      <td>12345-678</td>\n",
"      <td>São Paulo</td>\n",
"      <td>São Paulo</td>\n",
"      <td>Residencial ABC</td>\n",
```

```
"      <td>M</td>\n",
"      <td>João</td>\n",
"      <td>Silva</td>\n",
"      <td>1990-01-01</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>987.654.321-00</td>\n",
"      <td>(22) 9876-5432</td>\n",
"      <td>Avenida B, 456</td>\n",
"      <td>98765-432</td>\n",
"      <td>Rio de Janeiro</td>\n",
"      <td>Rio de Janeiro</td>\n",
"      <td>Apartamento XYZ</td>\n",
"      <td>F</td>\n",
"      <td>Maria</td>\n",
"      <td>Souza</td>\n",
"      <td>1985-05-10</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>111.222.333-44</td>\n",
"      <td>(33) 1111-2222</td>\n",
"      <td>Rua C, 789</td>\n",
"      <td>54321-098</td>\n",
"      <td>Minas Gerais</td>\n",
"      <td>Belo Horizonte</td>\n",
"      <td>Casa 123</td>\n",
"      <td>M</td>\n",
"      <td>Pedro</td>\n",
"      <td>Santos</td>\n",
"      <td>1982-12-25</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>555.444.333-22</td>\n",
```

```
"      <td>(44) 5555-4444</td>\n",
"      <td>Avenida D, 987</td>\n",
"      <td>76543-210</td>\n",
"      <td>Bahia</td>\n",
"      <td>Salvador</td>\n",
"      <td>Bloco ABCD</td>\n",
"      <td>F</td>\n",
"      <td>Ana</td>\n",
"      <td>Oliveira</td>\n",
"      <td>1995-09-15</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>4</th>\n",
"      <td>777.888.999-00</td>\n",
"      <td>(55) 7777-8888</td>\n",
"      <td>Rua E, 321</td>\n",
"      <td>01234-567</td>\n",
"      <td>São Paulo</td>\n",
"      <td>São Paulo</td>\n",
"      <td>Edifício EFGH</td>\n",
"      <td>M</td>\n",
"      <td>Lucas</td>\n",
"      <td>Ferreira</td>\n",
"      <td>1988-07-20</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
"          <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-
129d1eef-9130-4503-be98-afcf42a8885e')\"\n",
"            title=\"Convert this dataframe to an interactive table.\"\n",
"            style=\"display:none;\">\n",
"       \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0
24 24\"\n",
"       width=\"24px\">\n",
```

```
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"      <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
"  </svg>\n",
"    </button>\n",
"    \n",
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
"\n",
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert {\n",
```

```
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
"    }\n",
"  </style>\n",
"\n",
"      <script>\n",
"        const buttonEl =\n",
"          document.querySelector('#df-129d1eef-9130-4503-be98-afcf42a8885e button.colab-df-convert');\n",
"        buttonEl.style.display =\n",
"          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
"\n",
"        async function convertToInteractive(key) {\n",
"          const element = document.querySelector('#df-129d1eef-9130-4503-be98-afcf42a8885e');\n",
"          const dataTable =\n",
"            await google.colab.kernel.invokeFunction('convertToInteractive',\n",
"                                                     [key], {});\n",
"          if (!dataTable) return;\n",
"\n",
"          const docLinkHtml = 'Like what you see? Visit the ' +\n",
"                              '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebook</a>'\n",
"            + ' to learn more about interactive tables.';\n",
"          element.innerHTML = '';\n",
"          dataTable['output_type'] = 'display_data';\n",
"          await google.colab.output.renderOutput(dataTable, element);\n",
"          const docLink = document.createElement('div');\n",
"          docLink.innerHTML = docLinkHtml;\n",
```

```
          "          element.appendChild(docLink);\n",
          "        }\n",
          "      </script>\n",
          "    </div>\n",
          "  </div>\n",
          "  "
         ]
        },
        "metadata": {},
        "execution_count": 9
       }
      ]
     },
     {
      "cell_type": "markdown",
      "source": [
       "**Q013 Listar clientes por vendedor**"
      ],
      "metadata": {
       "id": "asPB8VSXy7Op"
      }
     },
     {
      "cell_type": "code",
      "source": [
       "df_resultado = df_vendedor.merge(df_compravenda, on='COD_VENDEDOR')\n",
       "df_resultado = df_resultado.merge(df_cliente, left_on='CPF_CLIENTE', right_on='CPF')\n",
       "df_resultado = df_resultado[['NOME', 'PRIMEIRO_NOME', 'SOBRENOME']]\n",
       "df_resultado = df_resultado.rename(columns={'PRIMEIRO_NOME': 'NOME_CLIENTE', 'SOBRENOME': 'SOBRENOME_CLIENTE'})\n",
       "df_resultado = df_resultado.sort_values(by='NOME')"
      ],
      "metadata": {
       "id": "2mJtCtiDyyEF"
      },
```

```
 "execution_count": null,
 "outputs": []
},
{
 "cell_type": "code",
 "source": [
  "df_resultado"
 ],
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/",
   "height": 206
  },
  "id": "WAltvt1r0JtJ",
  "outputId": "46ce36e1-65cb-4c23-a283-0b986cad5d3b"
 },
 "execution_count": null,
 "outputs": [
  {
   "output_type": "execute_result",
   "data": {
    "text/plain": [
     "            NOME NOME_CLIENTE SOBRENOME_CLIENTE\n",
     "0     João Silva        João           Silva\n",
     "1     João Silva         Ana         Oliveira\n",
     "2    Maria Santos       Maria          Souza\n",
     "4    Maria Santos       Pedro         Santos\n",
     "3  Pedro Oliveira       Maria          Souza"
    ],
    "text/html": [
     "\n",
     "  <div id=\"df-66b238f4-e91c-4698-9f84-7abbd697a25e\">\n",
     "    <div class=\"colab-df-container\">\n",
     "      <div>\n",
     "<style scoped>\n",
     "    .dataframe tbody tr th:only-of-type {\n",
```

```
"        vertical-align: middle;\n",
"    }\n",
"\n",
"    .dataframe tbody tr th {\n",
"        vertical-align: top;\n",
"    }\n",
"\n",
"    .dataframe thead th {\n",
"        text-align: right;\n",
"    }\n",
"</style>\n",
"<table border=\"1\" class=\"dataframe\">\n",
"  <thead>\n",
"    <tr style=\"text-align: right;\">\n",
"      <th></th>\n",
"      <th>NOME</th>\n",
"      <th>NOME_CLIENTE</th>\n",
"      <th>SOBRENOME_CLIENTE</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>João Silva</td>\n",
"      <td>João</td>\n",
"      <td>Silva</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>João Silva</td>\n",
"      <td>Ana</td>\n",
"      <td>Oliveira</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>Maria Santos</td>\n",
```

```
"      <td>Maria</td>\n",
"      <td>Souza</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>4</th>\n",
"      <td>Maria Santos</td>\n",
"      <td>Pedro</td>\n",
"      <td>Santos</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>Pedro Oliveira</td>\n",
"      <td>Maria</td>\n",
"      <td>Souza</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
"          <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-66b238f4-e91c-4698-9f84-7abbd697a25e')\"\n",
"            title=\"Convert this dataframe to an interactive table.\"\n",
"            style=\"display:none;\">\n",
"       \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",
"      width=\"24px\">\n",
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"      <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
"  </svg>\n",
"      </button>\n",
"      \n",
"  <style>\n",
"    .colab-df-container {\n",
```

```
"      display:flex;\n",
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
"\n",
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
"    }\n",
"  </style>\n",
```

```
        "\n",
        "      <script>\n",
        "        const buttonEl =\n",
        "          document.querySelector('#df-66b238f4-e91c-4698-9f84-7abbd697a25e button.colab-df-convert');\n",
        "        buttonEl.style.display =\n",
        "          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
        "\n",
        "        async function convertToInteractive(key) {\n",
        "          const element = document.querySelector('#df-66b238f4-e91c-4698-9f84-7abbd697a25e');\n",
        "          const dataTable =\n",
        "            await google.colab.kernel.invokeFunction('convertToInteractive',\n",
        "                                                     [key], {});\n",
        "          if (!dataTable) return;\n",
        "\n",
        "          const docLinkHtml = 'Like what you see? Visit the ' +\n",
        "                              '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebook</a>'\n",
        "            + ' to learn more about interactive tables.';\n",
        "          element.innerHTML = '';\n",
        "          dataTable['output_type'] = 'display_data';\n",
        "          await google.colab.output.renderOutput(dataTable, element);\n",
        "          const docLink = document.createElement('div');\n",
        "          docLink.innerHTML = docLinkHtml;\n",
        "          element.appendChild(docLink);\n",
        "        }\n",
        "      </script>\n",
        "    </div>\n",
        "  </div>\n",
        "  "
      ]
    },
    "metadata": {},
    "execution_count": 46
  }
```

```
   ]
  },
  {
   "cell_type": "markdown",
   "source": [
    "\n",
    "\n",
    "**Q014 Encontrar os produtos que nunca foram vendidos**\n"
   ],
   "metadata": {
    "id": "MNCiRgOz0Vz9"
   }
  },
  {
   "cell_type": "code",
   "source": [
    "# vamos combinar os dataframes com merge\n",
    "df_resultado = pd.merge(df_produto, df_compravenda, on='COD_PRODUTO', how='left', indicator=True)\n",
    "#produtos que nunca foram vendidos\n",
    "df_nao_vendidos = df_resultado[df_resultado['_merge'] == 'left_only']\n",
    "df_nao_vendidos = df_nao_vendidos[['NOME', 'COD_COMPRAVENDA']]\n",
    "df_nao_vendidos = df_nao_vendidos.sort_values(by='COD_COMPRAVENDA')\n",
    "\n",
    "df_nao_vendidos.head()"
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/",
     "height": 81
    },
    "id": "zQ4bkOVN0UMP",
    "outputId": "176712c6-cf70-48c6-b171-5e9f3de96f86"
   },
   "execution_count": null,
```

```
"outputs": [
 {
  "output_type": "execute_result",
  "data": {
   "text/plain": [
    "       NOME COD_COMPRAVENDA\n",
    "5  iPhone 14            NaN"
   ],
   "text/html": [
    "\n",
    "  <div id=\"df-5466ae21-a8bf-4919-ae41-124e28a98c92\">\n",
    "    <div class=\"colab-df-container\">\n",
    "      <div>\n",
    "<style scoped>\n",
    "    .dataframe tbody tr th:only-of-type {\n",
    "        vertical-align: middle;\n",
    "    }\n",
    "\n",
    "    .dataframe tbody tr th {\n",
    "        vertical-align: top;\n",
    "    }\n",
    "\n",
    "    .dataframe thead th {\n",
    "        text-align: right;\n",
    "    }\n",
    "</style>\n",
    "<table border=\"1\" class=\"dataframe\">\n",
    "  <thead>\n",
    "    <tr style=\"text-align: right;\">\n",
    "      <th></th>\n",
    "      <th>NOME</th>\n",
    "      <th>COD_COMPRAVENDA</th>\n",
    "    </tr>\n",
    "  </thead>\n",
    "  <tbody>\n",
    "    <tr>\n",
```

```
"      <th>5</th>\n",
"      <td>iPhone 14</td>\n",
"      <td>NaN</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
"          <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-5466ae21-a8bf-4919-ae41-124e28a98c92')\"\n",
"            title=\"Convert this dataframe to an interactive table.\"\n",
"            style=\"display:none;\">\n",
"       \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",
"       width=\"24px\">\n",
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"       <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
"  </svg>\n",
"      </button>\n",
"      \n",
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
```

```
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
"\n",
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
"    }\n",
"  </style>\n",
"\n",
"      <script>\n",
"        const buttonEl =\n",
"          document.querySelector('#df-5466ae21-a8bf-4919-ae41-124e28a98c92 button.colab-df-convert');\n",
"        buttonEl.style.display =\n",
"          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
"\n",
"        async function convertToInteractive(key) {\n",
"          const element = document.querySelector('#df-5466ae21-a8bf-4919-ae41-124e28a98c92');\n",
"          const dataTable =\n",
```

```
          "           await google.colab.kernel.invokeFunction('convertToInteractive',\n",
          "                                                    [key], {});\n",
          "         if (!dataTable) return;\n",
          "\n",
          "         const docLinkHtml = 'Like what you see? Visit the ' +\n",
          "                                                              '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebook</a>'\n",
          "           + ' to learn more about interactive tables.';\n",
          "         element.innerHTML = '';\n",
          "         dataTable['output_type'] = 'display_data';\n",
          "         await google.colab.output.renderOutput(dataTable, element);\n",
          "         const docLink = document.createElement('div');\n",
          "         docLink.innerHTML = docLinkHtml;\n",
          "         element.appendChild(docLink);\n",
          "       }\n",
          "     </script>\n",
          "   </div>\n",
          " </div>\n",
          " "
        ]
       },
       "metadata": {},
       "execution_count": 47
      }
     ]
    },
    {
     "cell_type": "markdown",
     "source": [
      "**Q015 Listar todos os clientes cadastrados no sistema**"
     ],
     "metadata": {
      "id": "_0JAcVo_0tVq"
     }
    },
```

```
{
 "cell_type": "code",
 "source": [
  "df_cliente['PRIMEIRO_NOME'].head()\n"
 ],
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/"
  },
  "id": "bCJEDB3fzq27",
  "outputId": "bfe65f7e-8185-405a-c4b0-82f35eda2384"
 },
 "execution_count": null,
 "outputs": [
  {
   "output_type": "execute_result",
   "data": {
    "text/plain": [
     "0     João\n",
     "1    Maria\n",
     "2    Pedro\n",
     "3      Ana\n",
     "4    Lucas\n",
     "Name: PRIMEIRO_NOME, dtype: object"
    ]
   },
   "metadata": {},
   "execution_count": 50
  }
 ]
},
{
 "cell_type": "markdown",
 "source": [
  "**Q016 NOVA: Contar quantas compras foram efetudadas por cliente**"
 ],
```

```
    "metadata": {
     "id": "ucMDEIwv1JtY"
    }
   },
   {
    "cell_type": "code",
    "source": [
     "compras_por_cliente = df_cliente.merge(df_compravenda, left_on='CPF', right_on='CPF_CLIENTE')\n",
     "contagem_compras = compras_por_cliente['PRIMEIRO_NOME'].value_counts().reset_index()\n",
     "contagem_compras.columns = ['PRIMEIRO_NOME', 'COUNT']\n",
     "contagem_compras.head(4)"
    ],
    "metadata": {
     "colab": {
      "base_uri": "https://localhost:8080/",
      "height": 175
     },
     "id": "WsDuc2xj1RXi",
     "outputId": "1b8ffd09-c3da-4a61-9ef4-6c09a99026c7"
    },
    "execution_count": null,
    "outputs": [
     {
      "output_type": "execute_result",
      "data": {
       "text/plain": [
        "  PRIMEIRO_NOME  COUNT\n",
        "0         Maria      2\n",
        "1          João      1\n",
        "2         Pedro      1\n",
        "3           Ana      1"
       ],
       "text/html": [
        "\n",
```

```
"  <div id=\"df-a4e82c85-66e5-45fe-bce1-4d6bb9cdfc74\">\n",
"    <div class=\"colab-df-container\">\n",
"      <div>\n",
"<style scoped>\n",
"    .dataframe tbody tr th:only-of-type {\n",
"        vertical-align: middle;\n",
"    }\n",
"\n",
"    .dataframe tbody tr th {\n",
"        vertical-align: top;\n",
"    }\n",
"\n",
"    .dataframe thead th {\n",
"        text-align: right;\n",
"    }\n",
"</style>\n",
"<table border=\"1\" class=\"dataframe\">\n",
"  <thead>\n",
"    <tr style=\"text-align: right;\">\n",
"      <th></th>\n",
"      <th>PRIMEIRO_NOME</th>\n",
"      <th>COUNT</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>Maria</td>\n",
"      <td>2</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>João</td>\n",
"      <td>1</td>\n",
"    </tr>\n",
"    <tr>\n",
```

```
"      <th>2</th>\n",
"      <td>Pedro</td>\n",
"      <td>1</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>Ana</td>\n",
"      <td>1</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
"          <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-a4e82c85-66e5-45fe-bce1-4d6bb9cdfc74')\"\n",
"            title=\"Convert this dataframe to an interactive table.\"\n",
"            style=\"display:none;\">\n",
"       \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",
"      width=\"24px\">\n",
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"      <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
"  </svg>\n",
"      </button>\n",
"      \n",
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
```

```
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
"\n",
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
"    }\n",
"  </style>\n",
"\n",
"      <script>\n",
"        const buttonEl =\n",
"            document.querySelector('#df-a4e82c85-66e5-45fe-bce1-4d6bb9cdfc74 button.colab-df-convert');\n",
"        buttonEl.style.display =\n",
```

```
          "          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
          "\n",
          "        async function convertToInteractive(key) {\n",
          "          const element = document.querySelector('#df-a4e82c85-66e5-45fe-bce1-4d6bb9cdfc74');\n",
          "          const dataTable =\n",
          "            await google.colab.kernel.invokeFunction('convertToInteractive',\n",
          "                                                     [key], {});\n",
          "          if (!dataTable) return;\n",
          "\n",
          "          const docLinkHtml = 'Like what you see? Visit the ' +\n",
          "                              '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebook</a>'\n",
          "            + ' to learn more about interactive tables.';\n",
          "          element.innerHTML = '';\n",
          "          dataTable['output_type'] = 'display_data';\n",
          "          await google.colab.output.renderOutput(dataTable, element);\n",
          "          const docLink = document.createElement('div');\n",
          "          docLink.innerHTML = docLinkHtml;\n",
          "          element.appendChild(docLink);\n",
          "        }\n",
          "      </script>\n",
          "    </div>\n",
          "  </div>\n",
          "  "
        ]
      },
      "metadata": {},
      "execution_count": 56
    }
  ]
},
{
  "cell_type": "markdown",
  "source": [
    "**Q017 NOVA: Visualizar compras realizadas por cliente**"
```

```
    ],
    "metadata": {
     "id": "BjIh4kV21yin"
    }
   },
   {
    "cell_type": "code",
    "source": [
     "# Juntar com inner join\n",
     "df = pd.merge(df_cliente, df_compravenda, left_on='CPF', right_on='CPF_CLIENTE')\n",
     "df = pd.merge(df, df_produto, on='COD_PRODUTO')\n",
     "df_grouped = df.groupby('COD_COMPRAVENDA').first().reset_index()\n",
     "df_result = df_grouped[['COD_COMPRAVENDA', 'PRIMEIRO_NOME', 'SOBRENOME', 'NOME', 'VALOR_TOTAL']]\n",
     "\n",
     "df_result.head()"
    ],
    "metadata": {
     "colab": {
      "base_uri": "https://localhost:8080/",
      "height": 206
     },
     "id": "DQKFkUGn1xMO",
     "outputId": "602a619e-fa23-43aa-edea-d5434423ec88"
    },
    "execution_count": null,
    "outputs": [
     {
      "output_type": "execute_result",
      "data": {
       "text/plain": [
        "  COD_COMPRAVENDA PRIMEIRO_NOME SOBRENOME NOME  \\\n",
        "0 CV001 João Silva iPhone 13  \n",
        "1 CV002 Maria Souza Smart TV 4K  \n",
        "2 CV003 Maria Souza Fone de Ouvido Bluetooth  \n",
```

```
  "3          CV004        Pedro    Santos       Tablet Galaxy Tab A   \n",
  "4          CV005          Ana  Oliveira          Câmera Digital   \n",
  "\n",
  "   VALOR_TOTAL  \n",
  "0       150.99  \n",
  "1        75.50  \n",
  "2       200.00  \n",
  "3       500.00  \n",
  "4      1000.00  "
 ],
 "text/html": [
  "\n",
  "  <div id=\"df-fed169c3-75a7-4a8c-a458-95c327da3982\">\n",
  "    <div class=\"colab-df-container\">\n",
  "      <div>\n",
  "<style scoped>\n",
  "    .dataframe tbody tr th:only-of-type {\n",
  "        vertical-align: middle;\n",
  "    }\n",
  "\n",
  "    .dataframe tbody tr th {\n",
  "        vertical-align: top;\n",
  "    }\n",
  "\n",
  "    .dataframe thead th {\n",
  "        text-align: right;\n",
  "    }\n",
  "</style>\n",
  "<table border=\"1\" class=\"dataframe\">\n",
  "  <thead>\n",
  "    <tr style=\"text-align: right;\">\n",
  "      <th></th>\n",
  "      <th>COD_COMPRAVENDA</th>\n",
  "      <th>PRIMEIRO_NOME</th>\n",
  "      <th>SOBRENOME</th>\n",
  "      <th>NOME</th>\n",
```

```
"      <th>VALOR_TOTAL</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>CV001</td>\n",
"      <td>João</td>\n",
"      <td>Silva</td>\n",
"      <td>iPhone 13</td>\n",
"      <td>150.99</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>CV002</td>\n",
"      <td>Maria</td>\n",
"      <td>Souza</td>\n",
"      <td>Smart TV 4K</td>\n",
"      <td>75.50</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>CV003</td>\n",
"      <td>Maria</td>\n",
"      <td>Souza</td>\n",
"      <td>Fone de Ouvido Bluetooth</td>\n",
"      <td>200.00</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>CV004</td>\n",
"      <td>Pedro</td>\n",
"      <td>Santos</td>\n",
"      <td>Tablet Galaxy Tab A</td>\n",
"      <td>500.00</td>\n",
"    </tr>\n",
```

```
"    <tr>\n",
"      <th>4</th>\n",
"      <td>CV005</td>\n",
"      <td>Ana</td>\n",
"      <td>Oliveira</td>\n",
"      <td>Câmera Digital</td>\n",
"      <td>1000.00</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
"          <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-fed169c3-75a7-4a8c-a458-95c327da3982')\"\n",
"            title=\"Convert this dataframe to an interactive table.\"\n",
"            style=\"display:none;\">\n",
"       \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",
"       width=\"24px\">\n",
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"       <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
"  </svg>\n",
"      </button>\n",
"      \n",
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
```

```
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
"\n",
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
"    }\n",
"  </style>\n",
"\n",
"      <script>\n",
"        const buttonEl =\n",
"          document.querySelector('#df-fed169c3-75a7-4a8c-a458-95c327da3982 button.colab-df-convert');\n",
"        buttonEl.style.display =\n",
"          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
```

```
          "\n",
          "        async function convertToInteractive(key) {\n",
          "         const element = document.querySelector('#df-fed169c3-75a7-4a8c-a458-95c327da3982');\n",
          "          const dataTable =\n",
          "            await google.colab.kernel.invokeFunction('convertToInteractive',\n",
          "                                                     [key], {});\n",
          "          if (!dataTable) return;\n",
          "\n",
          "          const docLinkHtml = 'Like what you see? Visit the ' +\n",
          "                                                    '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebook</a>'\n",
          "            + ' to learn more about interactive tables.';\n",
          "          element.innerHTML = '';\n",
          "          dataTable['output_type'] = 'display_data';\n",
          "          await google.colab.output.renderOutput(dataTable, element);\n",
          "          const docLink = document.createElement('div');\n",
          "          docLink.innerHTML = docLinkHtml;\n",
          "          element.appendChild(docLink);\n",
          "        }\n",
          "      </script>\n",
          "    </div>\n",
          "  </div>\n",
          "  "
        ]
       },
       "metadata": {},
       "execution_count": 57
      }
     ]
    },
    {
     "cell_type": "markdown",
     "source": [
      "**Q018 NOVA: Atualizar dados cadastrais do cliente**"
     ],
```

```
  "metadata": {
   "id": "uu5LW1D5ZZFY"
  }
 },
 {
  "cell_type": "code",
  "source": [
   "import pandas as pd\n",
   "\n",
   "# Create a DataFrame with the new data to be inserted\n",
   "new_data = {\n",
   "    'CPF': ['123.456.789-00'],\n",
   "    'SOBRENOME': ['Filho'],\n",
   "    'NOME': ['João']\n",
   "}\n",
   "df = pd.DataFrame(new_data)\n",
   "\n",
   "\n"
  ],
  "metadata": {
   "id": "FtFGdZbxk7S7"
  },
  "execution_count": 10,
  "outputs": []
 },
 {
  "cell_type": "code",
  "source": [
   "# Merge the new data with the CLIENTE DataFrame\n",
   "df_cliente = pd.merge(df_cliente, df, on='CPF', how='outer')"
  ],
  "metadata": {
   "id": "9YykBuSRlEN-"
  },
  "execution_count": 11,
  "outputs": []
```

```
  },
  {
   "cell_type": "code",
   "source": [
    "df_cliente.head()"
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/",
     "height": 337
    },
    "id": "3PdZRQGdlJ7X",
    "outputId": "eb69e3c1-134f-4c82-ba84-f12f00fa99ed"
   },
   "execution_count": 12,
   "outputs": [
    {
     "output_type": "execute_result",
     "data": {
      "text/plain": [
       "       CPF       TELEFONE       ENDERECO       CEP       ESTADO \\\n",
       "0  12345678900  (11) 1234-5678     Rua A, 123  12345-678      São Paulo  \n",
       "1  98765432100  (22) 9876-5432  Avenida B, 456  98765-432  Rio de Janeiro \n",
       "2  11122233344  (33) 1111-2222       Rua C, 789  54321-098     Minas Gerais \n",
       "3  55544433322  (44) 5555-4444  Avenida D, 987  76543-210         Bahia  \n",
       "4  77788899900  (55) 7777-8888     Rua E, 321  01234-567      São Paulo  \n",
       "\n",
       "            CIDADE           LOGRADOURO SEXO PRIMEIRO_NOME SOBRENOME_x  \\\n",
       "0       São Paulo  Residencial ABC    M          João       Silva  \n",
       "1  Rio de Janeiro  Apartamento XYZ    F         Maria       Souza  \n",
       "2  Belo Horizonte         Casa 123    M         Pedro      Santos  \n",
       "3       Salvador       Bloco ABCD    F          Ana     Oliveira  \n",
       "4       São Paulo    Edifício EFGH    M         Lucas    Ferreira  \n",
       "\n",
```

```
  "  DATA_NASCIMENTO SOBRENOME_y NOME  \n",
  "0             NaT        NaN  NaN  \n",
  "1             NaT        NaN  NaN  \n",
  "2             NaT        NaN  NaN  \n",
  "3             NaT        NaN  NaN  \n",
  "4             NaT        NaN  NaN  "
 ],
 "text/html": [
  "\n",
  "  <div id=\"df-a6c54e52-82d7-4fba-93ff-b22f86486361\">\n",
  "    <div class=\"colab-df-container\">\n",
  "      <div>\n",
  "<style scoped>\n",
  "    .dataframe tbody tr th:only-of-type {\n",
  "        vertical-align: middle;\n",
  "    }\n",
  "\n",
  "    .dataframe tbody tr th {\n",
  "        vertical-align: top;\n",
  "    }\n",
  "\n",
  "    .dataframe thead th {\n",
  "        text-align: right;\n",
  "    }\n",
  "</style>\n",
  "<table border=\"1\" class=\"dataframe\">\n",
  "  <thead>\n",
  "    <tr style=\"text-align: right;\">\n",
  "      <th></th>\n",
  "      <th>CPF</th>\n",
  "      <th>TELEFONE</th>\n",
  "      <th>ENDERECO</th>\n",
  "      <th>CEP</th>\n",
  "      <th>ESTADO</th>\n",
  "      <th>CIDADE</th>\n",
  "      <th>LOGRADOURO</th>\n",
```

```
"      <th>SEXO</th>\n",
"      <th>PRIMEIRO_NOME</th>\n",
"      <th>SOBRENOME_x</th>\n",
"      <th>DATA_NASCIMENTO</th>\n",
"      <th>SOBRENOME_y</th>\n",
"      <th>NOME</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>12345678900</td>\n",
"      <td>(11) 1234-5678</td>\n",
"      <td>Rua A, 123</td>\n",
"      <td>12345-678</td>\n",
"      <td>São Paulo</td>\n",
"      <td>São Paulo</td>\n",
"      <td>Residencial ABC</td>\n",
"      <td>M</td>\n",
"      <td>João</td>\n",
"      <td>Silva</td>\n",
"      <td>NaT</td>\n",
"      <td>NaN</td>\n",
"      <td>NaN</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>98765432100</td>\n",
"      <td>(22) 9876-5432</td>\n",
"      <td>Avenida B, 456</td>\n",
"      <td>98765-432</td>\n",
"      <td>Rio de Janeiro</td>\n",
"      <td>Rio de Janeiro</td>\n",
"      <td>Apartamento XYZ</td>\n",
"      <td>F</td>\n",
"      <td>Maria</td>\n",
```

```
"      <td>Souza</td>\n",
"      <td>NaT</td>\n",
"      <td>NaN</td>\n",
"      <td>NaN</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>11122233344</td>\n",
"      <td>(33) 1111-2222</td>\n",
"      <td>Rua C, 789</td>\n",
"      <td>54321-098</td>\n",
"      <td>Minas Gerais</td>\n",
"      <td>Belo Horizonte</td>\n",
"      <td>Casa 123</td>\n",
"      <td>M</td>\n",
"      <td>Pedro</td>\n",
"      <td>Santos</td>\n",
"      <td>NaT</td>\n",
"      <td>NaN</td>\n",
"      <td>NaN</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>55544433322</td>\n",
"      <td>(44) 5555-4444</td>\n",
"      <td>Avenida D, 987</td>\n",
"      <td>76543-210</td>\n",
"      <td>Bahia</td>\n",
"      <td>Salvador</td>\n",
"      <td>Bloco ABCD</td>\n",
"      <td>F</td>\n",
"      <td>Ana</td>\n",
"      <td>Oliveira</td>\n",
"      <td>NaT</td>\n",
"      <td>NaN</td>\n",
"      <td>NaN</td>\n",
```

```
"    </tr>\n",
"    <tr>\n",
"      <th>4</th>\n",
"      <td>77788899900</td>\n",
"      <td>(55) 7777-8888</td>\n",
"      <td>Rua E, 321</td>\n",
"      <td>01234-567</td>\n",
"      <td>São Paulo</td>\n",
"      <td>São Paulo</td>\n",
"      <td>Edifício EFGH</td>\n",
"      <td>M</td>\n",
"      <td>Lucas</td>\n",
"      <td>Ferreira</td>\n",
"      <td>NaT</td>\n",
"      <td>NaN</td>\n",
"      <td>NaN</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>\n",
"          <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-a6c54e52-82d7-4fba-93ff-b22f86486361')\"\n",
"            title=\"Convert this dataframe to an interactive table.\"\n",
"            style=\"display:none;\">\n",
"       \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",
"       width=\"24px\">\n",
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"      <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
"  </svg>\n",
"      </button>\n",
"      \n",
```

```
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
"\n",
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
```

```
"    }\n",
"  </style>\n",
"\n",
"      <script>\n",
"        const buttonEl =\n",
"            document.querySelector('#df-a6c54e52-82d7-4fba-93ff-b22f86486361 button.colab-df-convert');\n",
"        buttonEl.style.display =\n",
"          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
"\n",
"        async function convertToInteractive(key) {\n",
"          const element = document.querySelector('#df-a6c54e52-82d7-4fba-93ff-b22f86486361');\n",
"          const dataTable =\n",
"            await google.colab.kernel.invokeFunction('convertToInteractive',\n",
"                                                     [key], {});\n",
"          if (!dataTable) return;\n",
"\n",
"          const docLinkHtml = 'Like what you see? Visit the ' +\n",
"                              '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebook</a>'\n",
"            + ' to learn more about interactive tables.';\n",
"          element.innerHTML = '';\n",
"          dataTable['output_type'] = 'display_data';\n",
"          await google.colab.output.renderOutput(dataTable, element);\n",
"          const docLink = document.createElement('div');\n",
"          docLink.innerHTML = docLinkHtml;\n",
"          element.appendChild(docLink);\n",
"        }\n",
"      </script>\n",
"    </div>\n",
"  </div>\n",
"  "
    ]
  },
  "metadata": {},
```

```
        "execution_count": 12
       }
      ]
     }
    ]
}
```