

1. Patrimônio Mobiliário Tombado Da Fiocruz

- Esse notebook é o relatório final do Trabalho Prático 2 da disciplina de Introdução a Banco de Dados da UFMG 2023/2.
- Instruções para utilização/execução:

Caso as consultas não estiverem previamente executadas basta acessar a aba "Ambiente de execução" do Colaboratory e clicar nas opções "Desconectar e excluir o ambiente de execução" e "Executar tudo", respectivamente.

2. Membros

- ARTHUR BUZELIN GALERY - 2022043230
- ARTHUR RODRIGUES CHAGAS - 2022069417
- CAIO SANTANA TRIGUEIRO - 2022043310
- YAN AQUINO AMORIM - 2022043221

3. Descrição dos dados

- Link do DataSet Escolhido: <https://dados.gov.br/dados/conjuntos-dados/patrimnio-mobilirio-tombado-da-fiocruz>
- Tratamento do DataSet disponibilizado:

Inicialmente, os dados — que estavam disponíveis em um único arquivo no formato .csv — foram processados por meio de scripts Python, para garantir a divisão da tabela original em outras sete tabelas normalizadas, também no formato .csv. Essa segmentação foi baseada no Diagrama Relacional apresentado no Tópico 5, garantindo a organização adequada dos dados.

A novas tabelas normalizadas foram, então, agrupadas e salvas em um GitHub, em um arquivo compactado no formato .zip. A partir daí, as bibliotecas Python "requests" e "zipfile" foram utilizadas para baixar os dados diretamente para o ambiente local do Notebook e armazená-los utilizando SQLite. Cada tabela recebeu um nome correspondente ao nome dos arquivos .csv originais.

Os códigos utilizados, assim como a pasta zip contendo os arquivos .csv originais podem ser visualizados no GitHub pelo link: <https://github.com/caio-santtt/IBD-2023-2/tree/main> (o código utilizado para desenvolver a apresentação dos dados de forma interativa pelo Streamlit em Python também está no GitHub).

- Informações qualitativas das tabelas:

A tabela mais importante do esquema é a tabela "Bem", que contém a maior quantidade de dados gerais sobre os bens adquiridos pela Fiocruz. As demais tabelas fazem referências à "Bem", contendo informações como a categoria na qual o bem é caracterizado, a localização de seu armazenamento, suas unidades, pertences e compras; todas — de certa forma — ligadas aos bens, que são o foco deste trabalho.

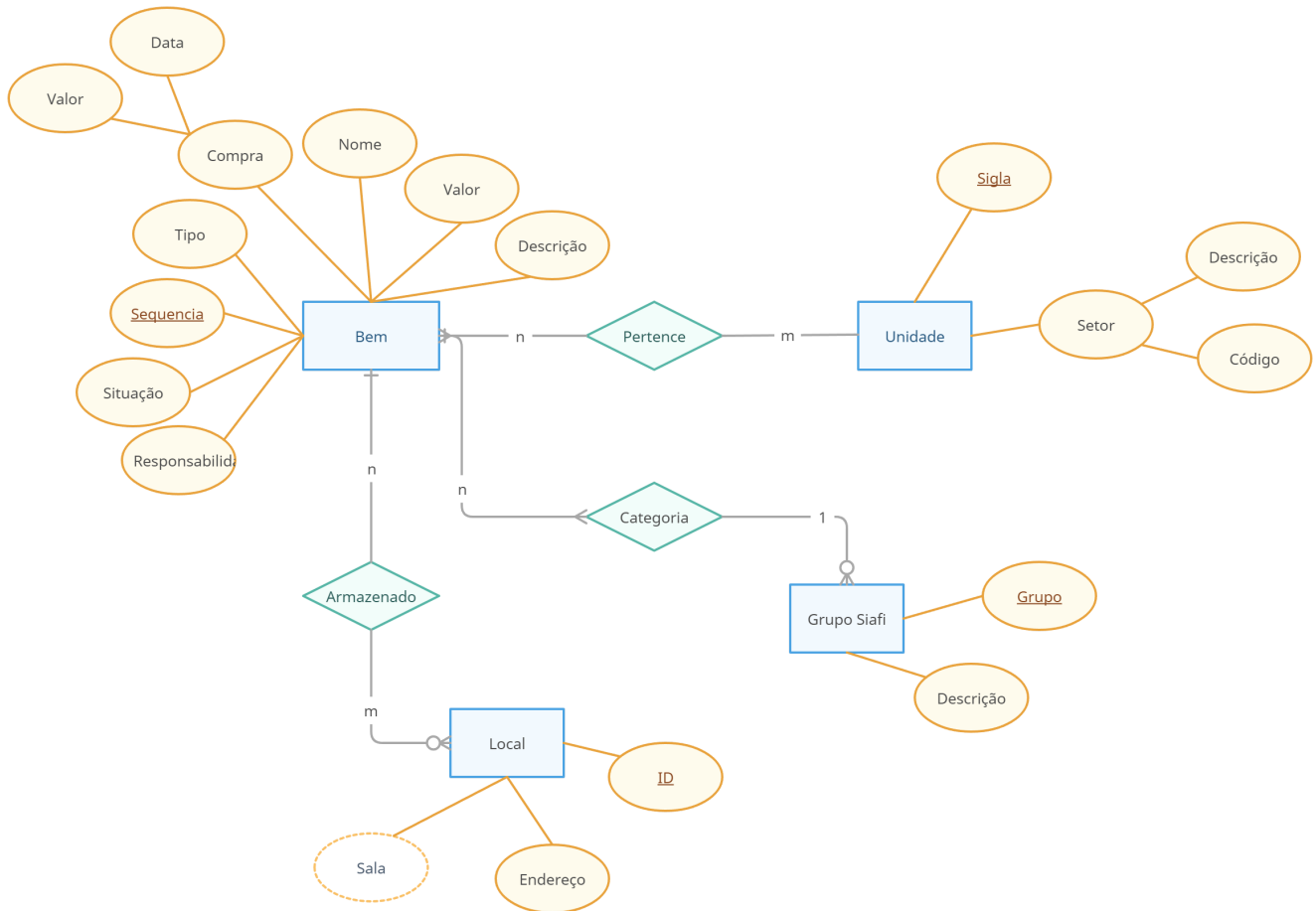
```
# Importar as Bibliotecas Necessárias
import io
import sqlite3
import pandas as pd
import requests
import zipfile
import matplotlib.pyplot as plt

# Download do Dump:
url = 'https://github.com/caio-santtt/IBD-2023-2/raw/main/tabelas.zip'
r = requests.get(url)
z = zipfile.ZipFile(io.BytesIO(r.content))
z.extractall("/tmp/dataset")

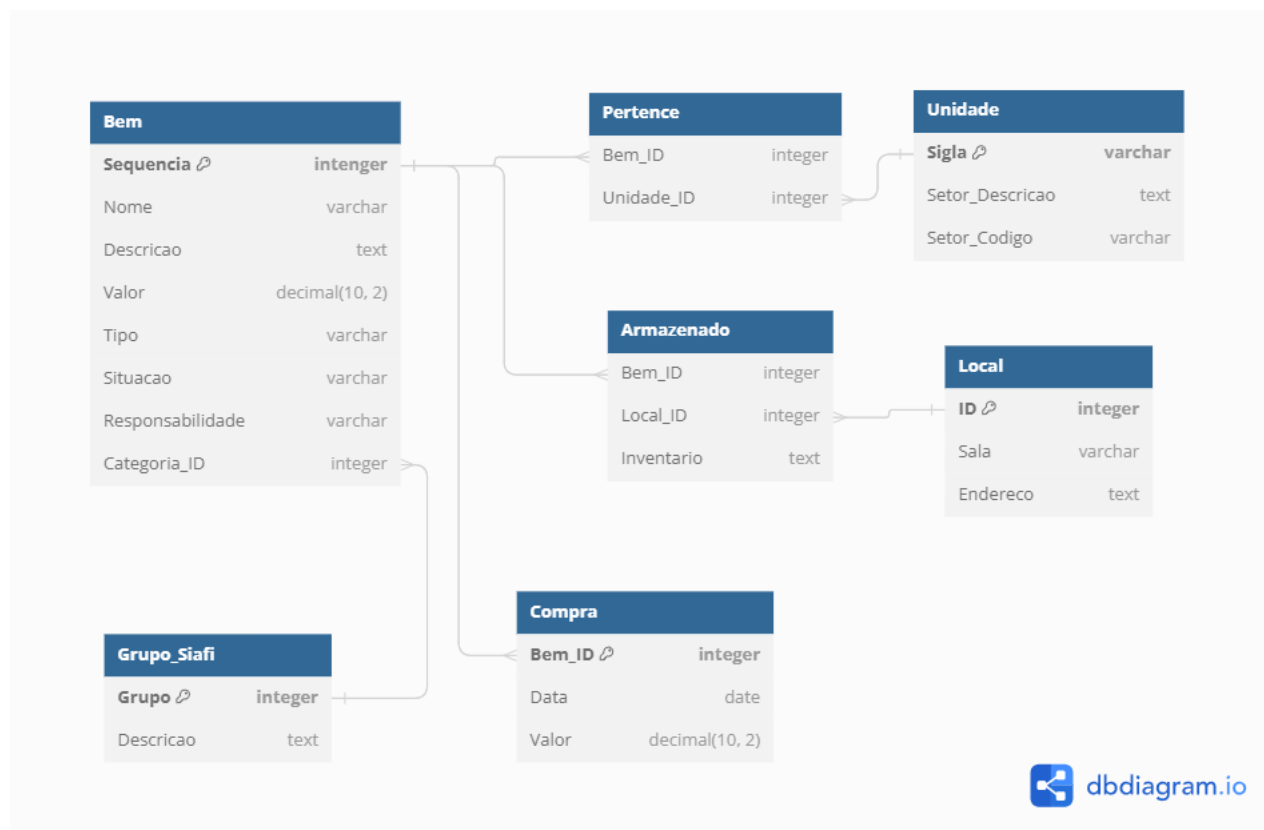
# Conectar ao Banco de Dados SQLite
conn = sqlite3.connect('/tmp/my_database.db')

# Carregar cada arquivo CSV no banco de dados SQLite
for filename in z.namelist():
    if filename.endswith('.csv'):
        df = pd.read_csv(f'/tmp/dataset/{filename}')
        table_name = filename.replace('.csv', '')
        df.to_sql(table_name, conn, if_exists='replace', index=False)
```

4. Diagrama ER



5. Diagrama relacional



6. Consultas

6.1 Duas consultas envolvendo seleção e projeção

6.1.1 Consulta 1

Descrição da Consulta 01: Essa consulta retorna todos os bens (ID, Endereço e Valor de compra) comprados apenas para os laboratorios da fiocruz.

```
query = """
-- C01

SELECT DISTINCT Armazenado.Bem_ID, Local.Endereco, Compra.Valor
FROM Armazenado
JOIN Local ON Armazenado.Local_ID = Local.ID
JOIN Compra ON Armazenado.Bem_ID = Compra.Bem_ID
WHERE TRIM(Local.Endereco) LIKE 'Laboratorio'
ORDER BY Valor DESC;

"""
```

```
df = pd.read_sql_query(query, conn)
df
```

	Bem_ID	Endereco	Valor
0	22075	LABORATORIO ...	995,00
1	22076	LABORATORIO ...	995,00
2	22077	LABORATORIO ...	995,00
3	22078	LABORATORIO ...	995,00
4	22079	LABORATORIO ...	995,00
...
3857	8957	LABORATORIO ...	0,00
3858	8997	LABORATORIO ...	0,00
3859	9124	LABORATORIO ...	0,00
3860	9174	LABORATORIO ...	0,00
3861	9212	LABORATORIO ...	0,00

3862 rows × 3 columns

6.1.2 Consulta 2

Descrição da Consulta 02: Essa consulta retorna todos os bens comprados e suas datas que custam mais de 1000 reais ordenados por ano de aquisicao.

```
query = """
-- C02

SELECT strftime('%Y', Data) AS AnoAquisicao, CAST(REPLACE(Valor, ',', '.') AS DECIMAL(10, 2)) AS ValorConvertido
FROM Compra
WHERE CAST(REPLACE(Valor, ',', '.') AS DECIMAL(10, 2)) > 1000
ORDER BY AnoAquisicao ASC;

"""

df = pd.read_sql_query(query, conn)
df
```

	AnoAquisicao	ValorConvertido
0	1753	2124.57
1	1753	1350.00
2	1983	1242.55
3	1984	1156.03
4	1984	1277.80
...

6.2 Três consultas envolvendo junção de duas relações

39874	2023	1195.00
-------	------	---------

6.2.1 Consulta 3

39876	2023	35148.95
-------	------	----------

Descrição da Consulta 03: Consulta os IDs distintos dos locais nos quais os Bens com valor mais alto estão armazenados, assim como o nome do Bem e seu valor.

```
query = """
-- C03

SELECT DISTINCT
    A.Local_ID, B.Nome, B. Valor
FROM
    Bem AS B
    Join Armazenado as A
    ON B.Sequencia = A.BEM_ID
WHERE
    B.Valor = (SELECT MAX(Valor) FROM Bem)
ORDER BY A.Local_ID ASC

"""
```

```
df = pd.read_sql_query(query, conn)
df
```

	Local_ID	NOME	VALOR
0	2535	CAMERA MICROSCOPIA ...	9998,00
1	13766	CAMERA MICROSCOPIA ...	9998,00
2	20421	CAMERA MICROSCOPIA ...	9998,00
3	26732	CAMERA MICROSCOPIA ...	9998,00

6.2.2 Consulta 4

Descrição da Consulta 04: Consulta o nome e a data de compra dos 20 Bens distintos comprados há mais tempo, que estão em situação "NORMAL"

```
query = """
-- C04

SELECT DISTINCT
    B.Nome, C.Data, B.Situacao
FROM
    Bem B
    JOIN Compra C
    ON B.Sequencia = C.Bem_ID
WHERE
    TRIM(B.Situacao) = 'NORMAL'
ORDER BY
    C.Data ASC
LIMIT 20

"""
```

```
df = pd.read_sql_query(query, conn)
df
```

	NOME	Data	SITUACAO
0	LIVROS ...	1753-01-01	NORMAL
1	TELEFONE ...	1753-01-01	NORMAL
2	GAVETEIRO ...	1753-01-01	NORMAL
3	BANCO/BANQUETA DE FERRO ...	1753-01-01	NORMAL
4	CADEIRA ...	1753-01-01	NORMAL
5	ESTANTE ...	1753-01-01	NORMAL
6	ARMARIO ...	1948-11-30	NORMAL
7	CADEIRA ...	1948-11-30	NORMAL
8	BANCO/BANQUETA DE FERRO ...	1948-11-30	NORMAL
9	CAMERA FOTOGRAFICA ...	1948-11-30	NORMAL
10	CAPELA/CABINE DE SEGURANÇA BIOLOGICA ...	1948-11-30	NORMAL
11	CADEIRA ...	1949-11-30	NORMAL
12	CAMARAS ...	1949-11-30	NORMAL
13	APAR.HOMOGENIZADOR ...	1949-11-30	NORMAL
14	...	1949-11-30	NORMAL
15	AGITADOR ...	1951-11-30	NORMAL
16	AR CONDICIONADO ...	1951-11-30	NORMAL
17	ARMADILHA LUMINOSA PARA CAPTURAR DE FLEBOTOMIN...	1951-11-30	NORMAL
18	BANCADA ...	1951-11-30	NORMAL

6.2.3 Consulta 5

Descrição da Consulta 05: Consulta o nome do responsável e a data de compra dos Bens comprados há mais tempo.

```
query = """
-- C05

SELECT DISTINCT
    B.Responsavel, C.Data
FROM
    Bem B
JOIN Compra C
    ON B.Sequencia = C.Bem_ID
ORDER BY
    C.Data ASC
LIMIT 20;

"""

df = pd.read_sql_query(query, conn)
df
```

	RESPONSAVEL	Data
0	...	1753-01-01
1	HELENA MARIA MAUÉS GUEDES COUTINHO ...	1753-01-01
2	MARIZE PEREIRA MIAGOSTOVICH ...	1753-01-01
3	ISABELA PENNA CERAVOLO ...	1753-01-01
4	FABIOLA LOPES CAETANO MACHADO ...	1753-01-01
5	ANDRE IVAN LOPES DE OLIVEIRA ...	1753-01-01

6.3 Três consultas envolvendo junção de três ou mais relações

6.3.1 Consulta 6

46 AV. AUGUSTO DE LIMA- Nº1715 - BELO HORIZONTE 1040-11-00

Descrição da Consulta 06: Consulta quantos Bens armazenados no endereço "AV. AUGUSTO DE LIMA- Nº1715 - BELO HORIZONTE" estão com situação "ALIENADO POR LEILÃO"

```
query = """
-- C06

SELECT
    COUNT(B.Sequencia)
FROM
    Bem B
    JOIN Armazenado A
        ON B.Sequencia = A.Bem_ID
    JOIN Local L
        ON A.Local_ID = L.ID
WHERE
    TRIM(B.Situacao) = 'ALIENADO POR LEILÃO'
    AND TRIM(L.Endereco) = 'AV. AUGUSTO DE LIMA- Nº1715 - BELO HORIZONTE'

"""
```

```
df = pd.read_sql_query(query, conn)
df
```



	COUNT(B.Sequencia)
0	463

6.3.2 Consulta 7

Descrição da Consulta 07: Consulta os 20 Bens distintos pertencentes ao Grupo Siafi "VEICULOS EM GERAL" com data de compra mais antiga, além da data de compra

```
query = """
-- C07

SELECT DISTINCT
    B.Nome, C.Data
FROM
    Bem B
    JOIN Grupo_Siafi G
        ON B.Categoria_ID = G.Grupo
    JOIN Compra C
        ON B.Sequencia = C.Bem_ID
WHERE
    TRIM(G.Descricao) = 'VEICULOS EM GERAL'
ORDER BY
    C.Data ASC
LIMIT 20

"""
```

```
df = pd.read_sql_query(query, conn)
df
```

	NOME	Data
0	CARRO P/TRANSP.MAT.EM ACO ...	1951-11-30
1	CARRINHO MAO/TRANSP.MATERIAL ...	1967-11-30
2	CARRINHO MAO/TRANSP.MATERIAL ...	1976-11-16
3	CARRINHO MAO/TRANSP.MATERIAL ...	1977-03-23
4	CARRINHO MAO/TRANSP.MATERIAL ...	1978-11-30
5	CARRINHO MAO/TRANSP.MATERIAL ...	1979-11-30
6	DIVERSOS ...	1979-11-30
7	BICICLETAS ...	1979-11-30
8	CARRO P/TRANSP.MAT.EM ACO ...	1979-11-30
9	EMPILHADEIRA/CARRO PLATAFORMA ...	1979-11-30
10	CARRINHO DE EMERGENCIA HOSPITALAR ...	1979-11-30
11	CARRO TUBULAR ...	1979-11-30
12	CARRO PARA LIMPEZA ...	1979-11-30
13	CARRO P/CILINDRO ...	1979-11-30
14	CARRO P/TRANSP.MAT.EM ACO ...	1983-05-09
15	CARRINHO MAO/TRANSP.MATERIAL ...	1983-05-10
16	CARRINHO MAO/TRANSP.MATERIAL ...	1983-12-22
17	CARRINHO MAO/TRANSP.MATERIAL ...	1984-01-17

6.3.3 Consulta 8

Descrição da Consulta 08: Consulta o nome e a situação de todos os Bens pertencentes a Unidade cuja descrição é: "FARMACIA POPULAR MG-BELO HORIZONTE"

```
query = """
-- C08

SELECT
    B.Nome, B.Situacao
FROM
    Bem B
JOIN Pertence P
    ON B.Sequencia = P.Bem_ID
JOIN Unidade U
    ON P.Unidade_ID = U.Sigla
WHERE
    TRIM(U.Setor_Descricao) = 'FARMACIA POPULAR MG-BELO HORIZONTE'

"""
```

```
df = pd.read_sql_query(query, conn)
df
```

	NOME	SITUACAO
0	ARQUIVO ... BAIXA BEM NAO LOCALIZADO/APURACAO	
1	ARQUIVO ... EM PROCESSO DE APURACAO	
2	ARQUIVO ...	NORMAL
3	BANHO SECO ...	NORMAL
4	BEM A ALIENAR ... ALIENADO POR LEILAO	
...
102212	BANCADA ...	NORMAL
102213	BANCADA ...	NORMAL
102214	CENTRIFUGA/MICRO CENTRIFUGA/ULTRA ...	NORMAL
102215	CENTRIFUGA/MICRO CENTRIFUGA/ULTRA ...	NORMAL
102216	CENTRIFUGA/MICRO CENTRIFUGA/ULTRA ...	NORMAL

102217 rows × 2 columns

6.4 Duas consultas envolvendo agregação sobre junção de duas ou mais relações

6.4.1 Consulta 9

Descrição da Consulta 09: Consulta quais são os 7 Grupos Siafi cuja a soma do valor de compra de seus bens é a maior. E retorna o código de tal grupo, assim como todas suas descrições atribuídas, e o valor dessa soma.

```
query = """
-- C09

SELECT gs.Grupo, gs.Descricao, SUM(c.Valor) AS SomaValores
FROM Grupo_Siafi gs
JOIN Bem b ON gs.Grupo = b.Categoria_ID
LEFT JOIN Compra c ON b.Sequencia = c.Bem_Id
WHERE gs.Grupo IN (
    SELECT DISTINCT gs2.Grupo
    FROM (
        SELECT gs.Grupo, SUM(c2.Valor) AS SomaValores
        FROM Grupo_Siafi gs
        JOIN Bem b2 ON gs.Grupo = b2.Categoria_ID
        LEFT JOIN Compra c2 ON b2.Sequencia = c2.Bem_Id
        GROUP BY gs.Grupo
        ORDER BY SomaValores DESC
        LIMIT 7
    ) AS gs2
)
GROUP BY gs.Grupo, gs.Descricao
ORDER BY SomaValores DESC;
```

"""

```
df = pd.read_sql_query(query, conn)
df
```

	Grupo	Descricao	SomaValores
0	12.311.03.03	BENS MOVEIS A ALIENAR	625517951.0
1	12.311.03.03	MOBILIARIO EM GERAL	625517951.0
2	12.311.01.03		337015607.0
3	12.311.01.03	APAR.EQUIP.UTENS.MED.,ODONT,LABOR.HOSPIT.	337015607.0
4	12.311.01.03	EQUIP.UTENS.MED.ODONT.LABOR.HOSPITALAR	337015607.0
5	12.311.01.25	BENS MOVEIS A ALIENAR	85507117.0
6	12.311.01.25	OUTRAS MAQUINAS,EQUIPAMENTOS E FERRAMENTAS	85507117.0
7	12.311.02.01	EQUIPAMENTOS DE PROCESSAMENTO DE DADOS	74962492.0
8	12.311.02.01	EQUIPAMENTOS DE TIC - ATIVOS DE REDE	74962492.0
9	12.311.02.01	EQUIPAMENTOS DE TIC - COMPUTADORES	74962492.0
10	12.311.02.01	MATERIAL DE TIC (PERMANENTE)	74962492.0
11	12.311.99.01	BENS MOVEIS A ALIENAR	74454357.0
12	12.311.99.01	BENS MOVEIS A ALIENAR	74454357.0
13	12.311.99.01	EQUIPAMENTOS DE PROCESSAMENTO DE DADOS	74454357.0
14	12.311.99.01	MAQUINAS E EQUIPAMENTOS ENERGETICOS	74454357.0
15	12.311.01.01		40065815.0
16	12.311.01.01	APARELHOS DE MEDICAO E ORIENTACAO	40065815.0
17	12.311.04.02		30632219.0
18	12.311.04.02	COLECOES E MATERIAIS BIBLIOGRAFICOS	30632219.0

6.4.2 Consulta 10

Descrição da Consulta 10: Consulta quais são os 7 Grupos Siafi cujo o valor médio por compra de bem é o maior. E retorna o código de tal grupo, assim como todas suas descrições atribuídas, e o valor dessa média.


```
query = ""
-- C10
```

```
SELECT gs.Grupo,
       gs.Descricao AS DescricaoGrupo,
       SUM(c.Valor) * 1.0 / COUNT(DISTINCT b.Sequencia) AS MediaValorCompraPorBem
FROM Grupo_Siafi gs
JOIN Bem b ON gs.Grupo = b.Categoria_ID
LEFT JOIN Compra c ON b.Sequencia = c.Bem_Id
WHERE gs.Grupo IN (
  SELECT DISTINCT gs2.Grupo
  FROM (
    SELECT gs3.Grupo, SUM(c2.Valor) AS SomaValores
    FROM Grupo_Siafi gs3
    JOIN Bem b2 ON gs3.Grupo = b2.Categoria_ID
    LEFT JOIN Compra c2 ON b2.Sequencia = c2.Bem_Id
    GROUP BY gs3.Grupo
    ORDER BY SomaValores DESC
    LIMIT 7
  ) AS gs2
)
GROUP BY gs.Grupo, gs.Descricao
ORDER BY MediaValorCompraPorBem DESC;
```

```
""
```

```
df = pd.read_sql_query(query, conn)
df
```

	Grupo	DescricaoGrupo	MediaValorCompraPorBem
0	12.311.01.03		27282.085890
1	12.311.01.03	APAR.EQUIP.UTENS.MED.,ODONT,LABOR.HOSPIT.	27282.085890
2	12.311.01.03	EQUIP.UTENS.MED.ODONT.LABOR.HOSPITALAR	27282.085890
3	12.311.01.01		17056.541081
4	12.311.01.01	APARELHOS DE MEDICAO E ORIENTACAO	17056.541081
5	12.311.03.03	BENS MOVEIS A ALIENAR	12138.908422
6	12.311.03.03	MOBILIARIO EM GERAL	12138.908422
7	12.311.04.02		9919.760039
8	12.311.04.02	COLECOES E MATERIAIS BIBLIOGRAFICOS	9919.760039
9	12.311.99.01	BENS MOVEIS A ALIENAR	9490.676482
10	12.311.99.01	BENS MOVEIS A ALIENAR	9490.676482
11	12.311.99.01	EQUIPAMENTOS DE PROCESSAMENTO DE DADOS	9490.676482
12	12.311.99.01	MAQUINAS E EQUIPAMENTOS ENERGETICOS	9490.676482
13	12.311.01.25	BENS MOVEIS A ALIENAR	8813.349516
14	12.311.01.25	OUTRAS MAQUINAS,EQUIPAMENTOS E FERRAMENTAS	8813.349516
15	12.311.02.01	EQUIPAMENTOS DE PROCESSAMENTO DE DADOS	8400.099955
16	12.311.02.01	EQUIPAMENTOS DE TIC - ATIVOS DE REDE	8400.099955
17	12.311.02.01	EQUIPAMENTOS DE TIC - COMPUTADORES	8400.099955
18	12.311.02.01	MATERIAL DE TIC (PERMANENTE)	8400.099955

6.5 Fechar a conexão com o Banco de Dados SQLite ao terminar as consultas

```
# FECHAR A CONEXÃO AO TERMINAR AS CONSULTAS
conn.close()
```

7. Autoavaliação dos membros

- ARTHUR BUZELIN GALERY:

Realizei a conversão da tabela única com uso de scripts para tabelas normalizadas, a normalização do diagrama ER para o Relacional, e as consultas 1 e 2. Também configurei o ambiente do Streamlit.

- ARTHUR RODRIGUES CHAGAS:

Na primeira etapa elaborei as possíveis investigações em alto nível a serem feitas a partir do banco de dados disponível. Já na segunda parte realizei a implementação do Streamlit via Web para análise iterativa dos dados e os slides a serem apresentados em sala de aula.

- CAIO SANTANA TRIGUEIRO:

Na primeira etapa do trabalho fiz parte da pesquisa para a escolha do dataset utilizado. Nesta segunda realizei a implementação do banco de dados no Notebook do Colab, implementação do ambiente do GitHub, o tópico 3 de descrição dos dados, formatação do notebook e as consultas 9 e 10.

- YAN AQUINO AMORIM:

Na primeira etapa do trabalho fiz parte da pesquisa para a escolha do dataset utilizado, corriji e renomeei as colunas no arquivo CSV original e elaborei o desenho do diagrama conforme o modelo Entidade-Relacionamento (ER). Posteriormente, na segunda etapa, realizei o tratamento das tabelas normalizadas, que precisaram ser modificadas e corrigidas, além das consultas 3, 4, 5, 6, 7 e 8.