

SOUL **CODE**

MIGRAÇÃO E INTEGRAÇÃO DE BASE DE DADOS

ALUNO: JUM SAHEKI

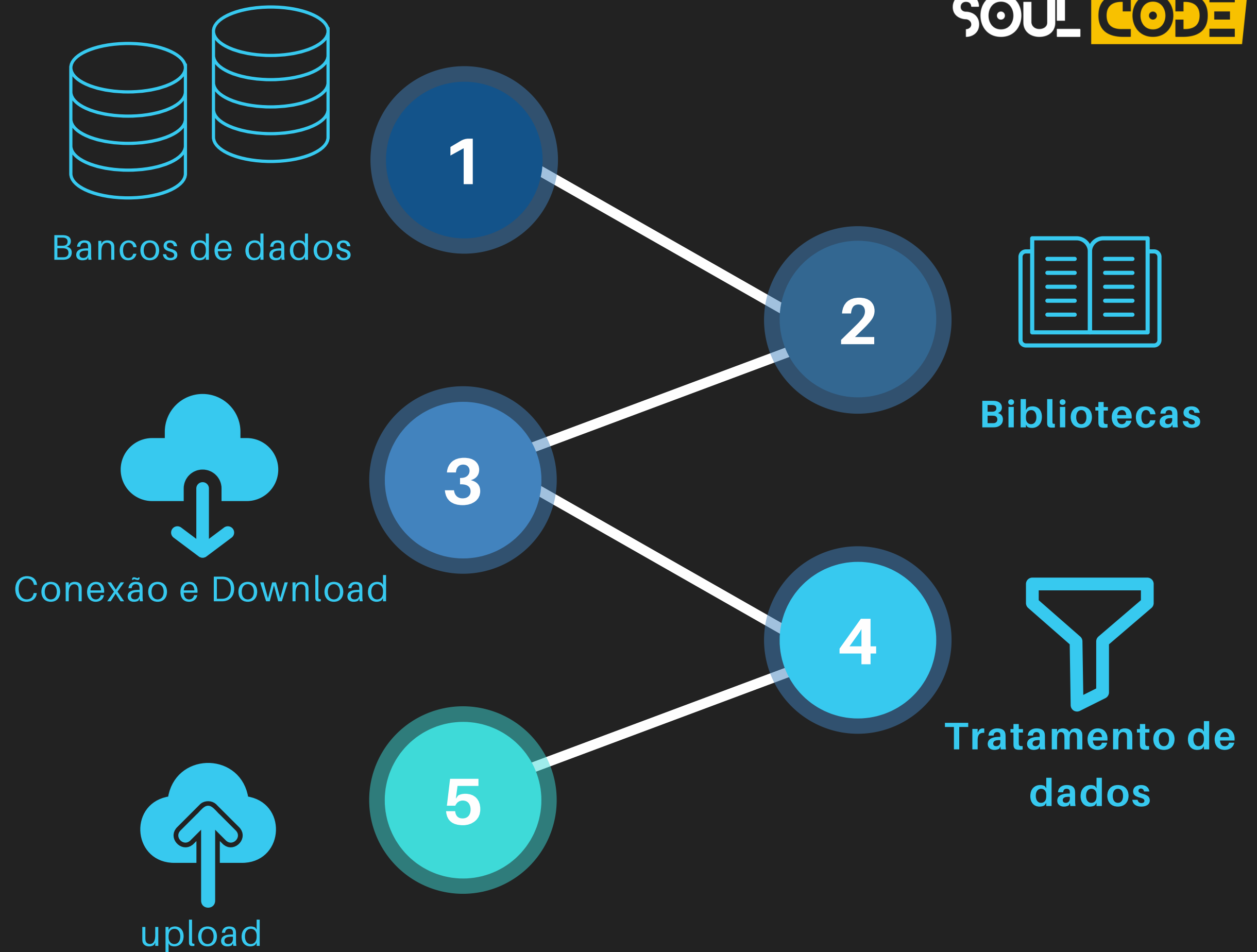
TURMA: BC17 - ENGENHARIA DE DADOS

Enunciado

Atividade 20

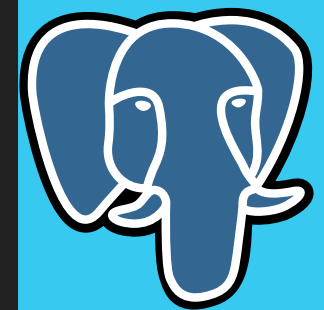
A SoulCode Academy utiliza dois sistemas distintos para cadastro de vendas, um em sua matriz e outro em suas filiais. O sistema da matriz utiliza um db noSQL enquanto os das filiais ainda utilizam um SQL. As tabelas a seguir são amostras de dados dos dois tipos de db enviadas para auxiliar seu desenvolvimento. Crie uma db em SQL e uma noSQL, e adicione essas amostras para criar um ambiente simulando o problema real. A seguir crie um código em Python que leia a db SQL, corrija, padronize e insira os dados na db noSQL.

TÓPICOS



Banco de dados

SQL

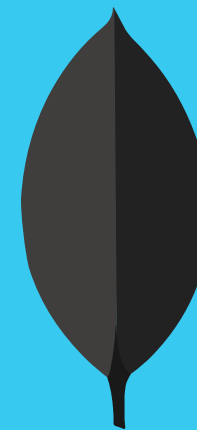


PostgreSQL



heroku

noSQL



mongoDB

BIBLIOTECAS

pandas

pymongo

psycopg2

drive

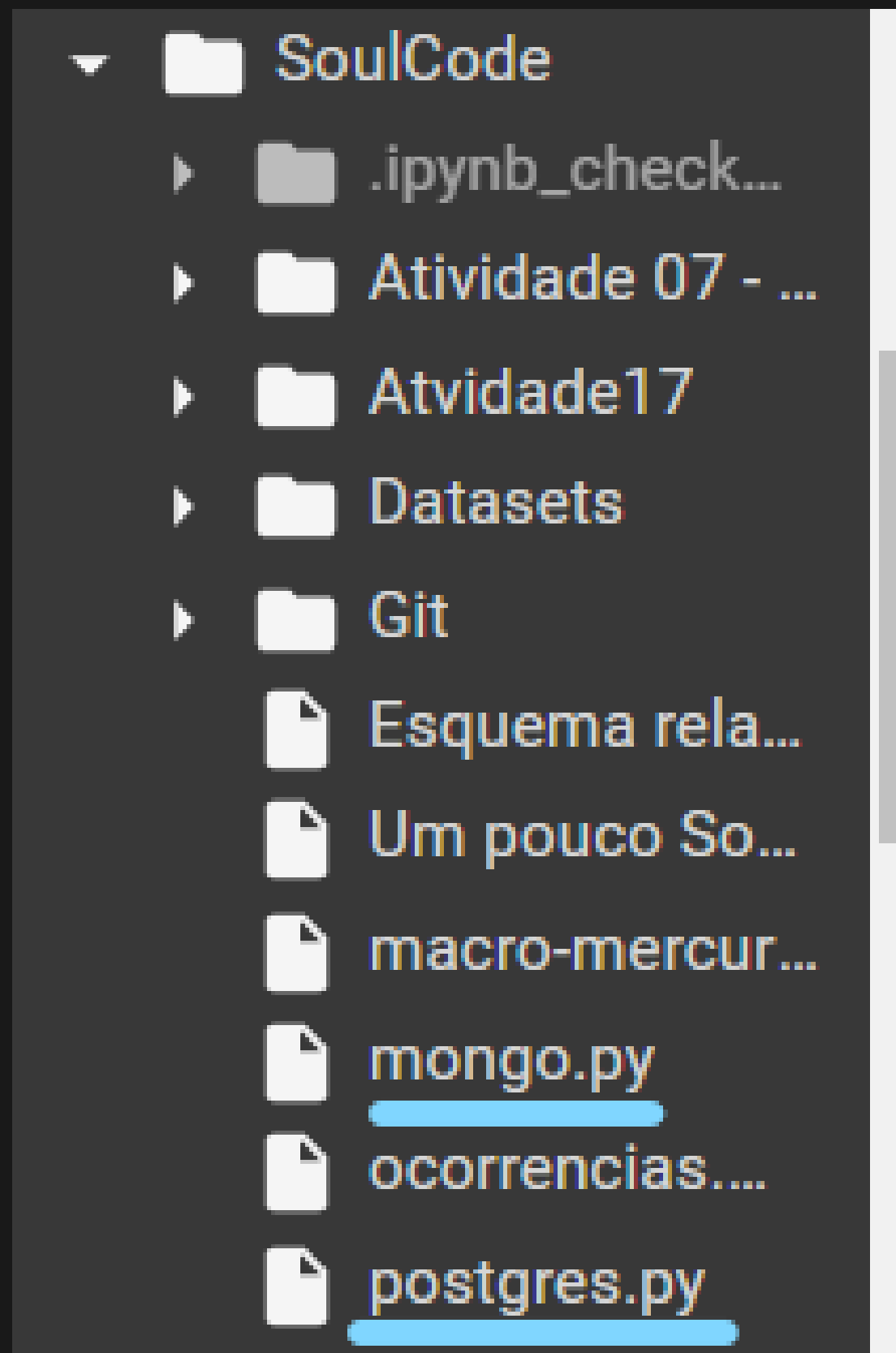
```
[ ] # pandas
import pandas as pd
from pandas.io.json import json_normalize
```

```
[ ] # pymongo
!pip install pymongo[srv]
from pymongo import MongoClient
!cp /content/drive/MyDrive/SoulCode/mongo.py .
from mongo import get_mongo_access
```

```
[ ] # psycopg2
!cp /content/drive/MyDrive/SoulCode/postgres.py .
from postgres import Conector_postgres, get_postgres_access
```

```
[ ] # Google drive
from google.colab import drive
```

GOOGLE DRIVE



CONEXÃO E DOWNLOAD

POSTGRES

```
[ ] # Configurações da conexão  
pg_host, pg_db, pg_user, pg_pwd = get_postgres_access()  
banco_pg = Conector_postgres(pg_host, pg_db, pg_user, pg_pwd)  
colunas = ['nota_fiscal', 'vendedor', 'total']
```

```
[ ] # Download  
dados = banco_pg.selecionar('select * from sistema_a')  
df_pg = pd.DataFrame(dados, columns = colunas)
```

MONGO ATLAS

```
[ ] # Configurações de conexão
    !curl ipecho.net/plain
    conn, db_name, collection_name = get_mongo_access()
    client = MongoClient(conn)
    db = client[db_name]
    collection = db[collection_name]
```

```
[ ] # Download
    df_m = pd.DataFrame(list(collection.find()))
```

TRATAMENTO DE DADOS

NULOS

DUPLICATAS

TIPO DE
DADOS

CARACTERES
ESTRANHOS

BACKUP

```
✓ [15] df = df_pg.copy()  
0s
```

PRÉ-ANALISE

✓
0s



```
df.head()
```



	nota_fiscal	vendedor	total
--	-------------	----------	-------



0	1	Clarabelle	159.38
1	2	Berk	144.3
2	3	Kiri	302.49
3	4	Brigitta	381.61
4	5	Barret	277.63

NULOS

✓
0s

```
[16] df.isnull().sum()
```

```
nota_fiscal    0  
vendedor       31  
total          0  
dtype: int64
```

```
[ ] df.fillna('não informado', inplace=True)
```

DUPLICATAS

✓
0s



```
df['nota_fiscal'].is_unique
```

True

TIPOS DE DADOS

✓
0s

```
[18] df.dtypes
```

```
nota_fiscal    int64  
vendedor       object  
total          object  
dtype: object
```



```
df['total'] = pd.to_numeric(df['total'])
```

CARACTERES ESTRANHOS

✓
0s



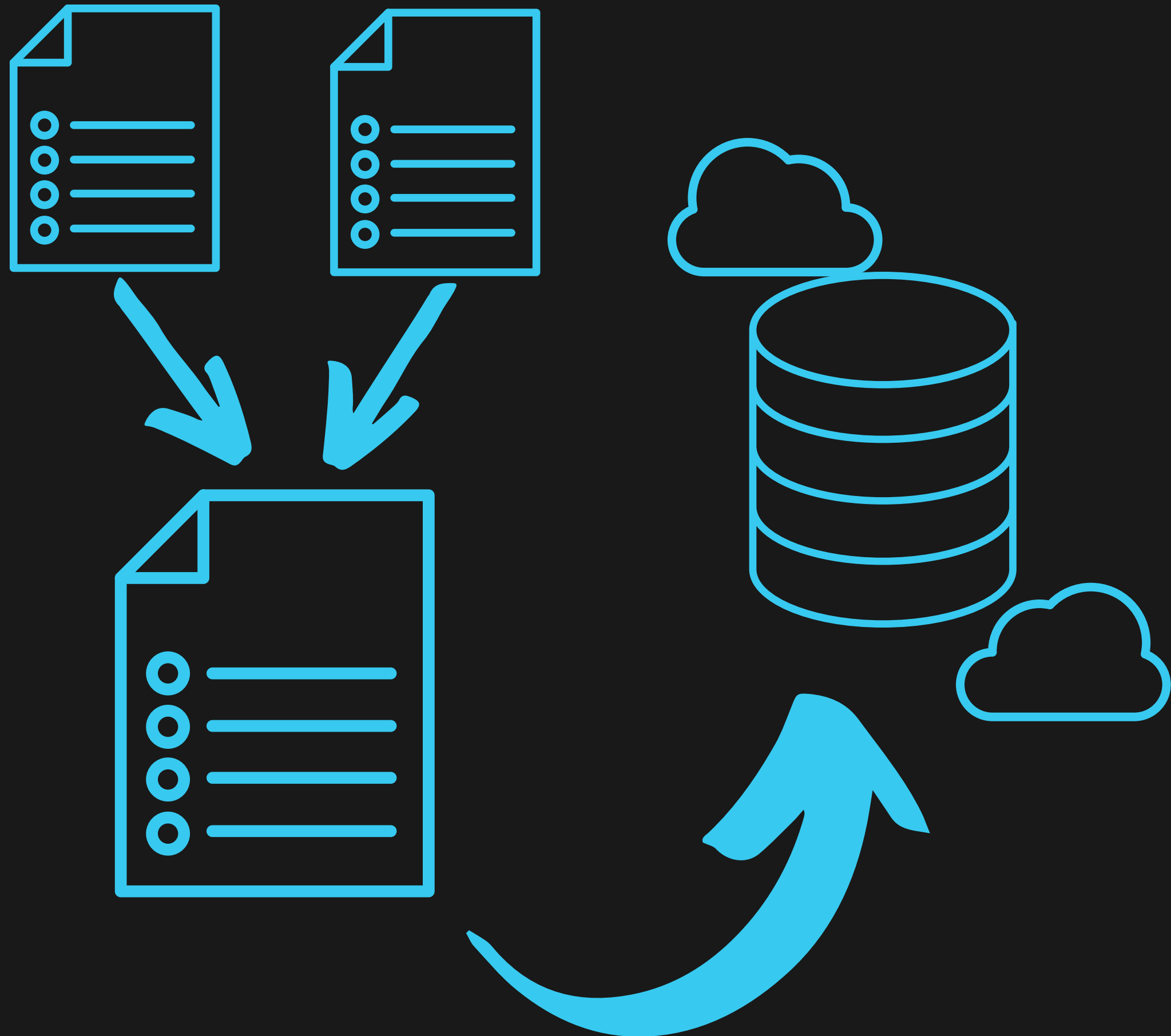
df.dtypes

```
nota_fiscal    int64
vendedor       object
total          float64
dtype: object
```

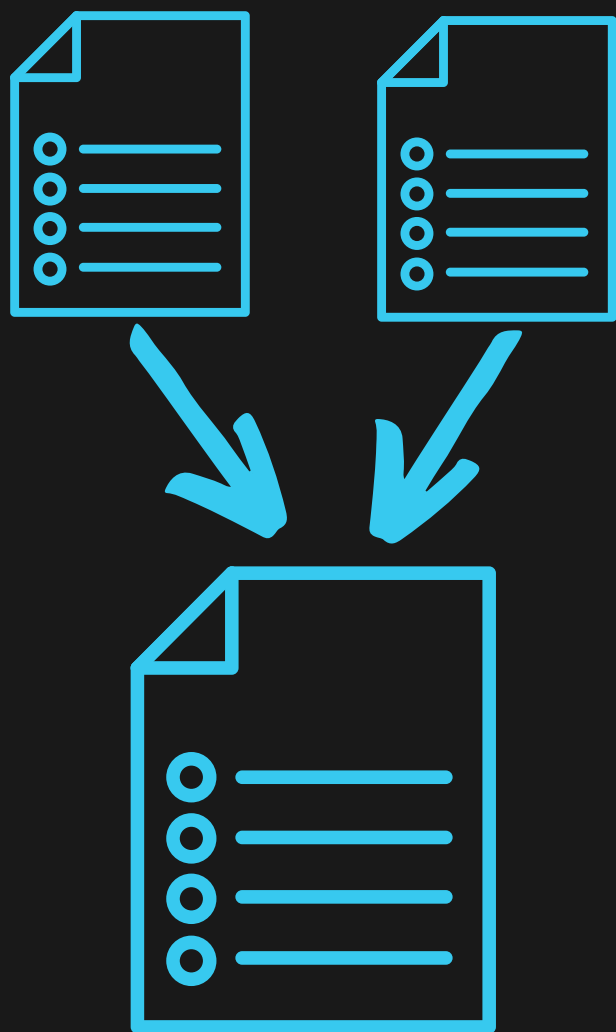
```
[ ] sorted(pd.unique(df['vendedores']))
```

```
[ ] sorted(pd.unique(df['total']))
```


UPLOAD

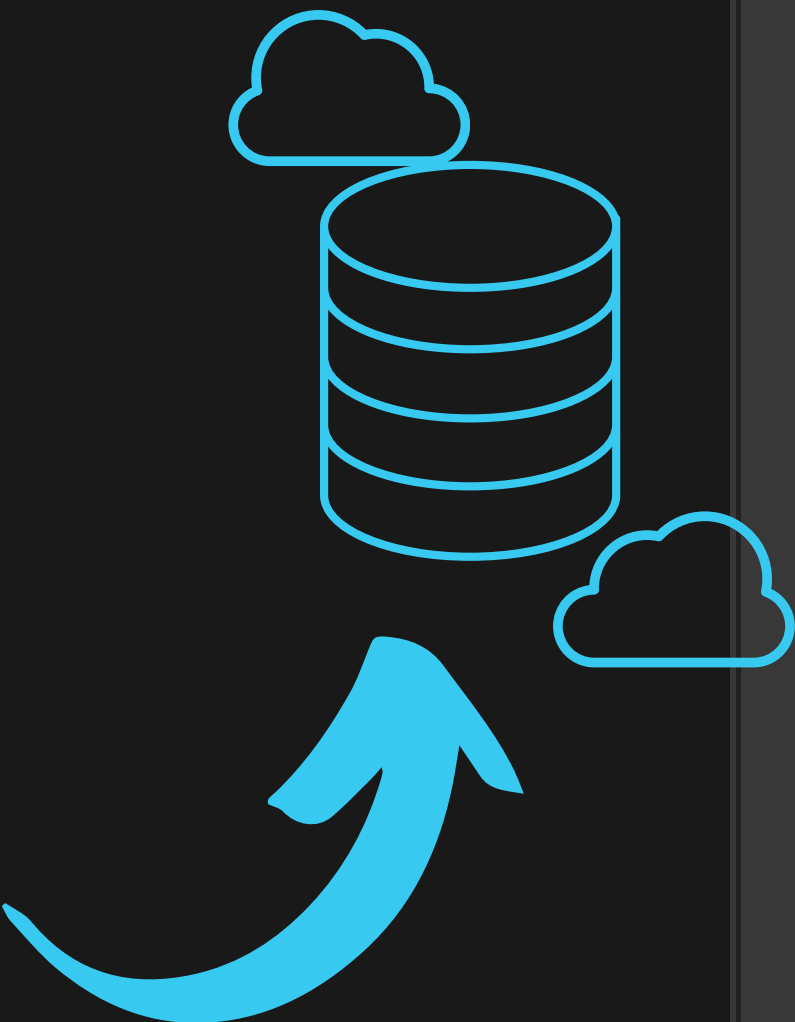


CONCATENANDO OS DATAFRAMES



```
[ ] df = pd.concat([df_pg2,df_m2], ignore_index=True)
```

UPLOAD



```
[ ] df_dict = df.to_dict('records')  
  
db.create_collection('venda')  
collection_in = db['venda']  
  
collection_in.insert_many(df_dict)
```

Link para a atividade:

[https://colab.research.google.com/drive/1U9yMlxLhV6j4p2U1wnHP-ON5EVC0tdXW?
usp=sharing](https://colab.research.google.com/drive/1U9yMlxLhV6j4p2U1wnHP-ON5EVC0tdXW?usp=sharing)

Link para a Apresentação:

[https://www.canva.com/design/DAFBec3c3hE/IsVb9JDHIwp9ErbuGfGs5g/edit?
utm_content=DAFBec3c3hE&utm_campaign=designshare&utm_medium=link2&utm_sour
ce=sharebutton](https://www.canva.com/design/DAFBec3c3hE/IsVb9JDHIwp9ErbuGfGs5g/edit?utm_content=DAFBec3c3hE&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton)

Muito Obrigado!