FUNDAÇÃO EDSON QUEIROZ
UNIVERSIDADE DE FORTALEZA - UNIFOR
CENTRO DE TECNOLOGIA E ENGENHARIA
CURSO MBA EM ENGENHARIA DE DADOS

TRABALHO FINAL PARA DISCIPLINA: LINGUAGEM DE PROGRAMAÇÃO PARA ENGENHARIA DE DADOS

FELIPE ALVES DA SILVA

Fortaleza-CE
2024

FELIPE ALVES DA SILVA

TRABALHO FINAL PARA DISCIPLINA: LINGUAGEM DE PROGRAMAÇÃO PARA ENGENHARIA DE DADOS

Trabalho final apresentado ao curso de MBA em engenharia de dados da Universidade de Fortaleza, para obtenção de nota final para a disciplina de linguagem de programação para engenharia de dados lecionado pelo professor Thiago Bluhm.

Fortaleza-CE

2024

SUMÁRIO

SUMÁRIO 3
1 DIFICULDADES E PONTOS A FAVOR 4
2 EVIDÊNCIAS DE PERSISTÊNCIA DE DADOS NO POSTGRESQL 5
3 FONTES DE PESQUISA 7

LINK PARA NOTEBOOK NO GITHUB:
HTTPS://GITHUB.COM/FELIPEALVSS/PROJETO_DADOS_IBGE/BLOB/MAIN/DESAFIO%20IBGE.IPYNB

1 DIFICULDADES E PONTOS A FAVOR

O primeiro ponto de dificuldade que deparei nesse trabalho foi buscar substitutos para o *Pandas*, pois era a única biblioteca que até o momento conhecia para uso de diversos dados, porém com o uso do *CSV* e do *OS* pude implementar a lógica de carga de vários arquivos de forma ágil. Em seguida, a outra problemática percebida foi entender como manipular o dado de forma competente sem que eu estourasse a memória da minha máquina pessoal. Através de sites e (na maior parte) de diversas tentativas práticas, pude finalmente entender como aplicar uma carga incremental através de lotes de dados de forma que a inserção dos dados funcionou de forma competente.

Como ponto a favor da estratégia que escolhi, percebi que escolher utilizar a função *session.bulk_insert_mappings()* (*sqlalchemy*) juntamente com a divisão dos dados em lote se tornou muito mais rápida do que a inserção de dados utilizando a função *session.add(),* a qual mostrou ótima performance utilizando apenas um arquivo como parâmetro para carga (cerca de 950mil linhas), porém se mostrou extremamente onerosa com grande volume de dados.

Outro ponto positivo foi ter a necessidade de entender como realizar limpezas das variáveis e a manter um controle de memória com o *garbage collector* do python, foi nítido como o uso dessas boas práticas tornam o processamento mais leve para execução em hardwares não tão potentes (caso do meu computador pessoal).

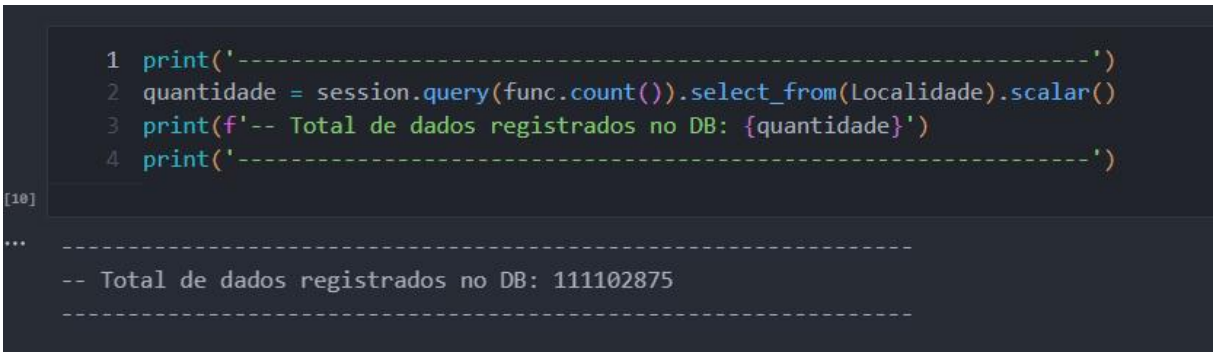
Ao buscar por soluções presentes de mercado, pude perceber que muito se indica o uso do *Pandas* e *NumPy* juntamente com o *Dask*, permitindo aplicação de escalabilidade e utilizando a sintaxe padrão do Python. Outro que é muito comentado é o *Spark*, o qual funciona em cluster e consegue manipular de forma ágil e simples grandes volumes de dados. Há também a presença de ferramentas de ETL que podem realizar estes serviços, tais com o Pentaho Data integration, o Talend Open Studio e até mesmo o SSIS (Microsoft SQL Server Integration Services).

LINK PARA NOTEBOOK NO GITHUB:
HTTPS://GITHUB.COM/FELIPEALVSS/PROJETO_DADOS_IBGE/BLOB/MAIN/DESAFIO%20IBGE.IPYNB

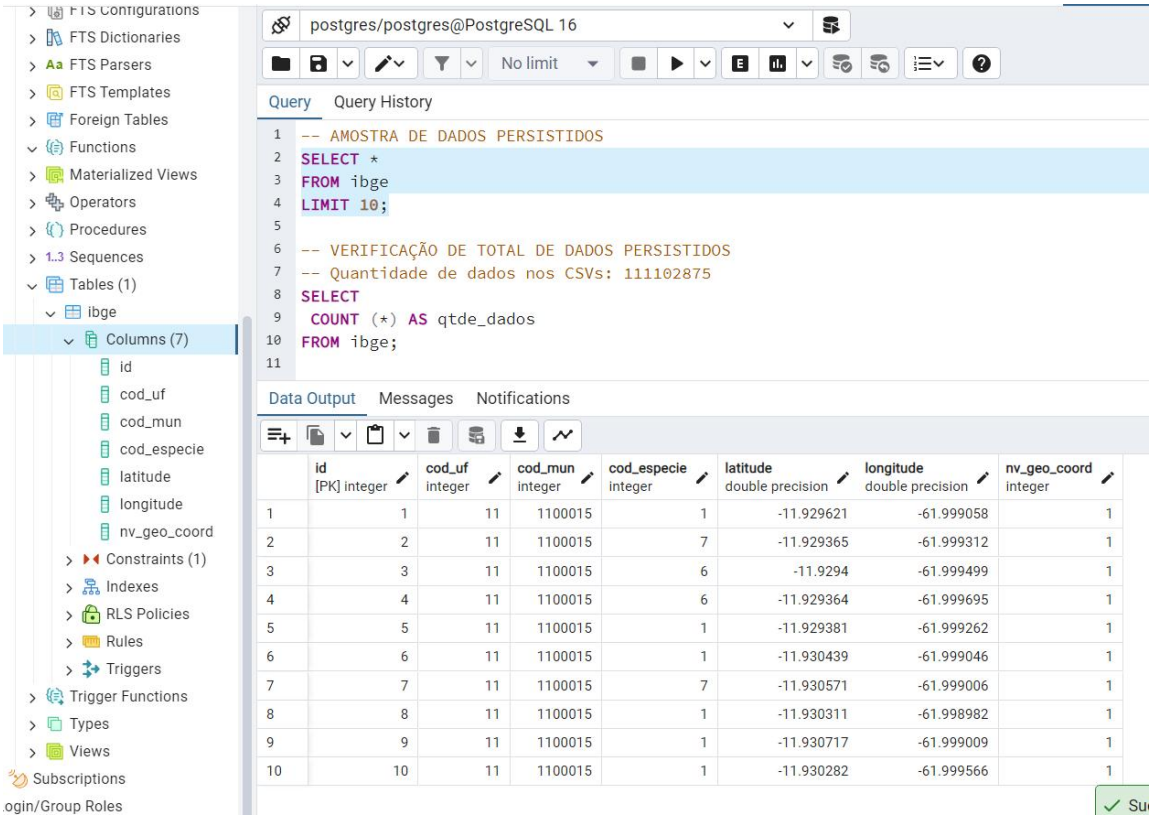
2 EVIDÊNCIAS DE PERSISTÊNCIA DE DADOS NO POSTGRESQL

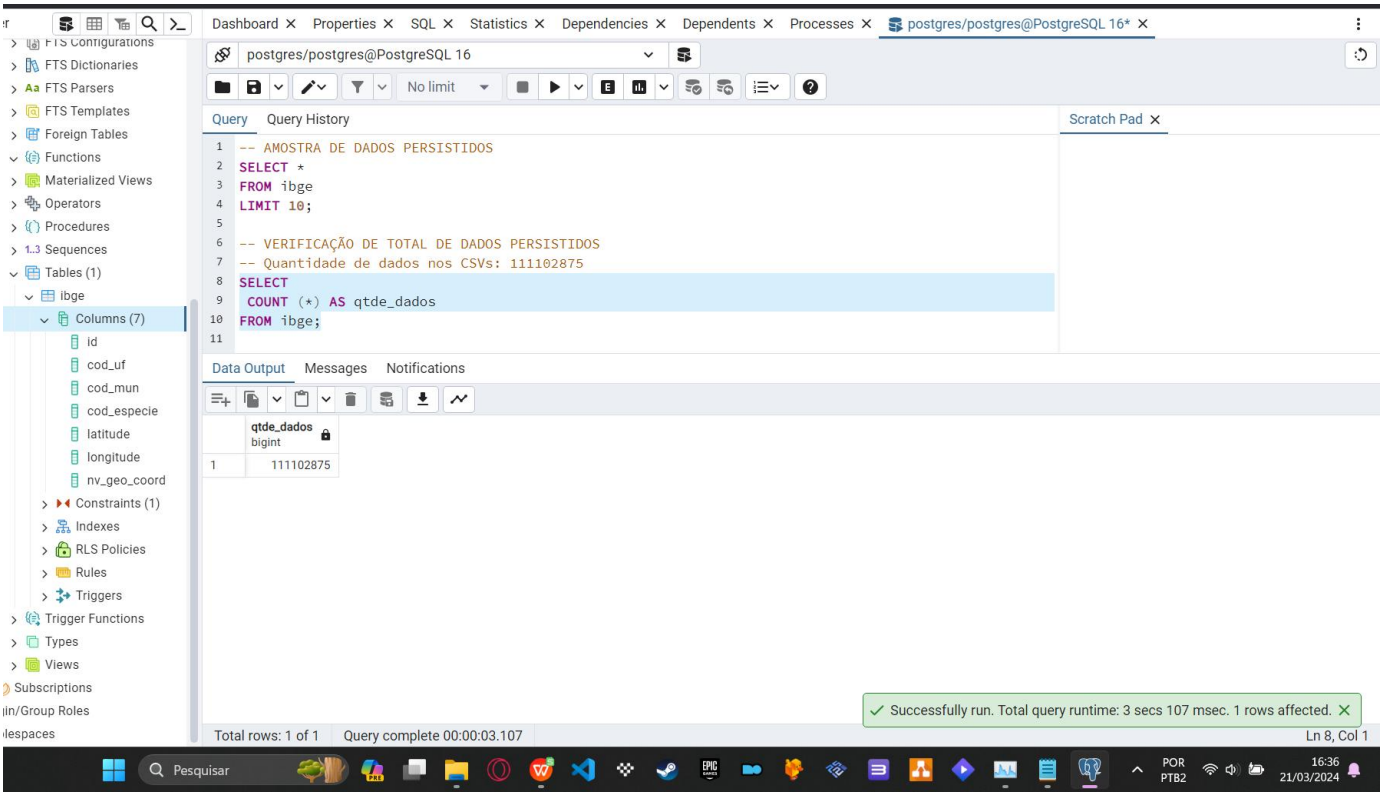
Seguem imagens com evidências que todos os dados foram persistidos no SGBD PostgreSQL:

1. Print da querry presente no fim do Notebook informando a quantidade total de dados persistidos no PostgreSQL:



2. Prints de querries executadas no próximo PostgreSQL exibindo uma amostra dos dados persistidos e executando uma contagem total dos dados persistidos na tabela:




Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Processes
postgres/postgres@PostgreSQL 16*
postgres/postgres@PostgreSQL 16
No limit
Query
Query History
Scratch Pad
-- AMOSTRA DE DADOS PERSISTIDOS
SELECT *
FROM ibge
LIMIT 10;
-- VERIFICAÇÃO DE TOTAL DE DADOS PERSISTIDOS
-- Quantidade de dados nos CSVs: 111102875
SELECT
COUNT (*) AS qtde_dados
FROM ibge;
Data Output
Messages
Notifications
qtde_dados
bigint
1
111102875
Successfully run. Total query runtime: 3 secs 107 msec. 1 rows affected.
Total rows: 1 of 1
Query complete 00:00:03.107
Ln 8, Col 1
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Operators
Procedures
Sequences
Tables (1)
ibge
Columns (7)
id
cod_uf
cod_mun
cod_especie
latitude
longitude
nv_geo_coord
Constraints (1)
Indexes
RLS Policies
Rules
Triggers
Trigger Functions
Types
Views
Subscriptions
Pesquisar
POR
PTB2
16:36
21/03/2024

LINK PARA NOTEBOOK NO GITHUB:
HTTPS://GITHUB.COM/FELIPEALVSS/PROJETO_DADOS_IBGE/BLOB/MAIN/DESAFIO%20IBGE.IPYNB

3 FONTES DE PESQUISA

https://coderpad.io/blog/development/sqlalchemy-with-postgresql/

https://docs.sqlalchemy.org/en/20/orm/mapping_styles.html

https://docs-sqlalchemy.readthedocs.io/ko/latest/orm/persistence_techniques.html

https://www.pythonforbeginners.com/basics/generator-comprehension-in-python

https://www.w3schools.com/python/python_dictionaries.asp

https://towardsdatascience.com/how-to-perform-bulk-inserts-with-sqlalchemy-efficiently-in-python-23044656b97d

https://realpython.com/python-timer/