Pós-graduação

Engenharia de Dados / Big Data

Aluno: Charles de Araujo Melo

Infnet - Trabalho Infraestrutura Cassandra

1. Explique, com suas palavras, as diferenças entre bases de dados SQL e NoSQL, apresentando exemplos de uso para cada uma delas.

R. O grande diferencial dos bancos de dados NoSQL é que toda a informação é agrupada e guardada no mesmo registro. Já no SQL você precisa ter o relacionamento entre várias tabelas para ter a informação, informação esta disposta no modelo entidade e relacionamento.

As bases de dados SQL utilizam uma abordagem de escala vertical, o que significa que escalam adicionando mais poder ao servidor. As bases de dados NoSQL utilizam uma abordagem de escala horizontal, o que significa que são escaladas adicionando mais servidores.

O NoSQL é mais indicado para aqueles sistemas que tenham necessidades maiores de armazenamento e desempenho com dados não estruturados. Já o SQL é indicado quando necessitamos de maior consistência.

2. Escolha uma base de dados pública brasileira, que será o alvo para condução de seu Projeto de Disciplina. Esta base será carregada para uma infraestrutura Cassandra, para dar suporte à geração de um processo de análise de dados.

R. Base de dados escolida : food-preference
https://www.kaggle.com/datasets/naufaal98/food-preference

3. Defina pelo menos uma pergunta de negócio que será o objetivo de seu processo de análise a ser desenvolvido a partir dos dados armazenados na infraestrutura Cassandra.

R. Todos Clientes com idade maior que 30 anos e que comeram sobremesa.

R. Média da idade de todos os clientes do gênero masculino e que não comeram sobremesa.

1.1 Apresente um planejamento básico da infraestrutura necessária para comportar os dados que deseja importar.

1.2 Apresente a descrição das configurações feitas em um ambiente computacional para receber a infraestrutura Cassandra. Ao final deste passo, a máquina que hospedará o ambiente deverá estar pronta para receber o conjunto de dados escolhido para o trabalho.

1.3 Faça a carga de dados para o ambiente configurado e execute todas as atividades necessárias na base para que a infraestrutura fique pronta para a execução de operações com os dados. Apresente uma comprovação de que a carga de dados tenha sido corretamente executada.

Usarei o Docker para criar um container com a imagem do cassandra.

Instalação e configuração do Cassandra (via container Docker):

https://www.docker.com/products/docker-desktop/

Baixa a imagem do Cassandra no Docker Hub para a máquina local

1.1 Baixar a imagem:

docker pull cassandra

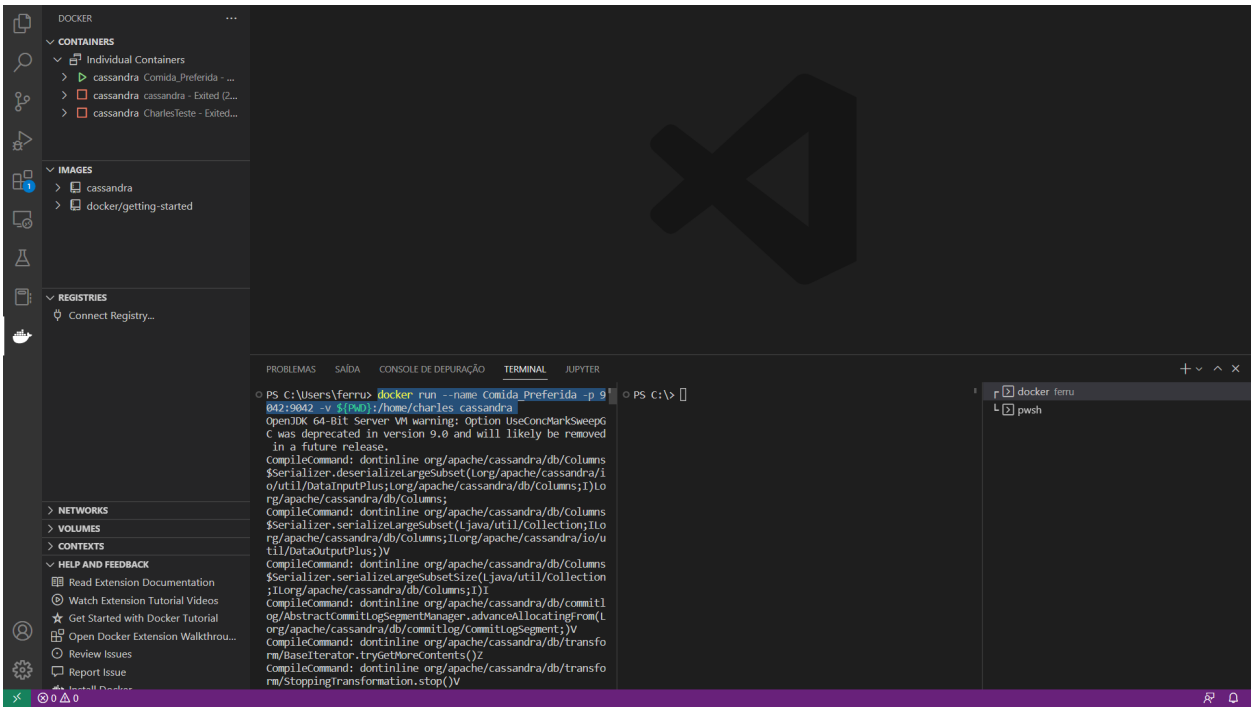
1.2 Visualizar as imagens existentes na máquina

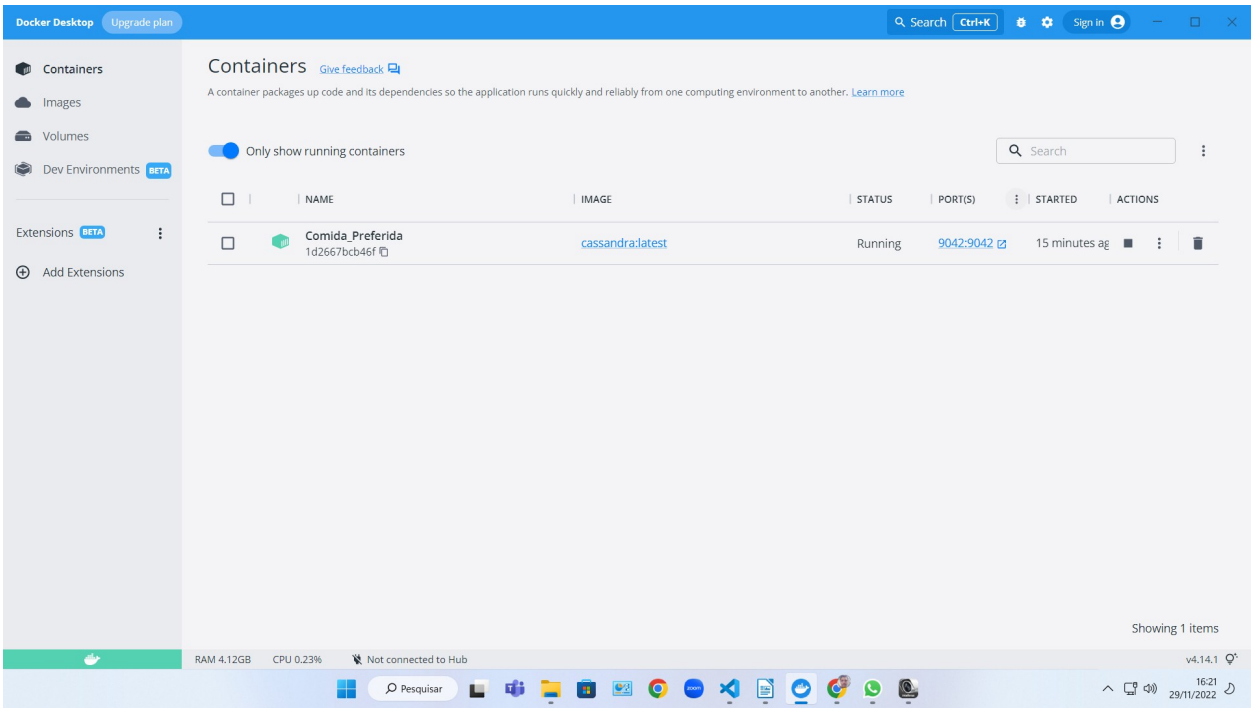
docker image ls

Utilizando o terminal do Visual Studio Code para criar meu container com espelhamento em meu diretório local para por meus arquivos para serem enxergados dentro do container.

Utilizei o seguinte comando:

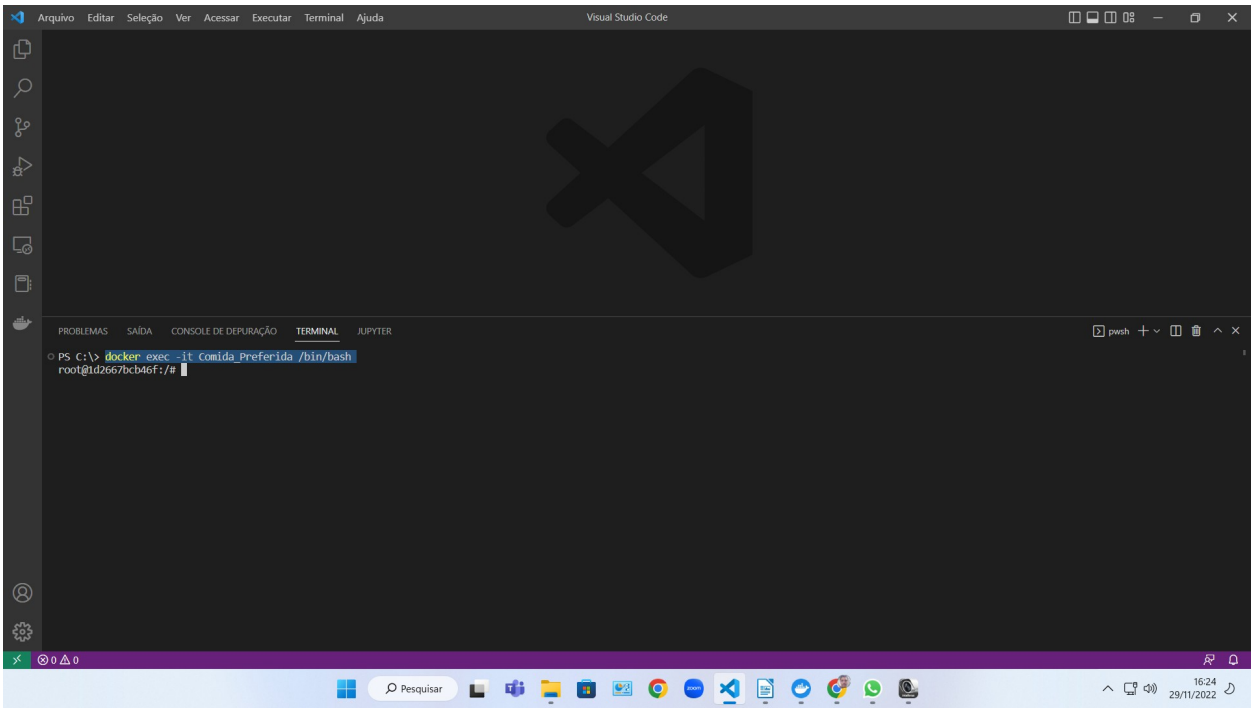
docker run --name Comida_Preferida -p 9042:9042 -v ${PWD}:/home/charles cassandra


DOCKER
CONTAINERS
Individual Containers
cassandra Comida_Preferida - ...
cassandra cassandra - Exited (2...
cassandra CharlesTeste - Exited...
IMAGES
cassandra
docker/getting-started
REGISTRIES
Connect Registry...
NETWORKS
VOLUMES
CONTEXTS
HELP AND FEEDBACK
Read Extension Documentation
Watch Extension Tutorial Videos
Get Started with Docker Tutorial
Open Docker Extension Walkthrou...
Review Issues
Report Issue
PROBLEMAS
SAÍDA
CONSOLE DE DEPURAÇÃO
TERMINAL
JUPYTER
PS C:\Users\ferru> docker run --name Comida_Preferida -p 9042:9042 -v ${PWD}:/home/charles_cassandra
OpenJDK 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprecated in version 9.0 and will likely be removed in a future release.
CompileCommand: dontinline org/apache/cassandra/db/Columns$Serializer.deserializeLargeSubset(Lorg/apache/cassandra/io/util/DataInputPlus;Lorg/apache/cassandra/db/Columns;I)Lorg/apache/cassandra/db/Columns;
CompileCommand: dontinline org/apache/cassandra/db/Columns$Serializer.serializeLargeSubset(Ljava/util/Collection;ILorg/apache/cassandra/db/Columns;ILorg/apache/cassandra/io/util/DataOutputPlus;)V
CompileCommand: dontinline org/apache/cassandra/db/Columns$Serializer.serializeLargeSubsetSize(Ljava/util/Collection;ILorg/apache/cassandra/db/Columns;I)I
CompileCommand: dontinline org/apache/cassandra/db/commitlog/AbstractCommitLogSegmentManager.advanceAllocatingFrom(Lorg/apache/cassandra/db/commitlog/CommitLogSegment;)V
CompileCommand: dontinline org/apache/cassandra/db/transform/BaseIterator.tryGetMoreContents()Z
CompileCommand: dontinline org/apache/cassandra/db/transform/StoppingTransformation.stop()V
PS C:\>
docker ferru
pwsh


Docker Desktop
Upgrade plan
Search Ctrl+K
Sign in
Containers
Images
Volumes
Dev Environments BETA
Extensions BETA
Add Extensions
Containers
Give feedback
A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. Learn more
Only show running containers
Search
NAME
IMAGE
STATUS
PORT(S)
STARTED
ACTIONS
Comida_Preferida
1d2667bcb46f
cassandra:latest
Running
9042:9042
15 minutes ag
Showing 1 items
RAM 4.12GB
CPU 0.23%
Not connected to Hub
v4.14.1
Pesquisar
16:21
29/11/2022

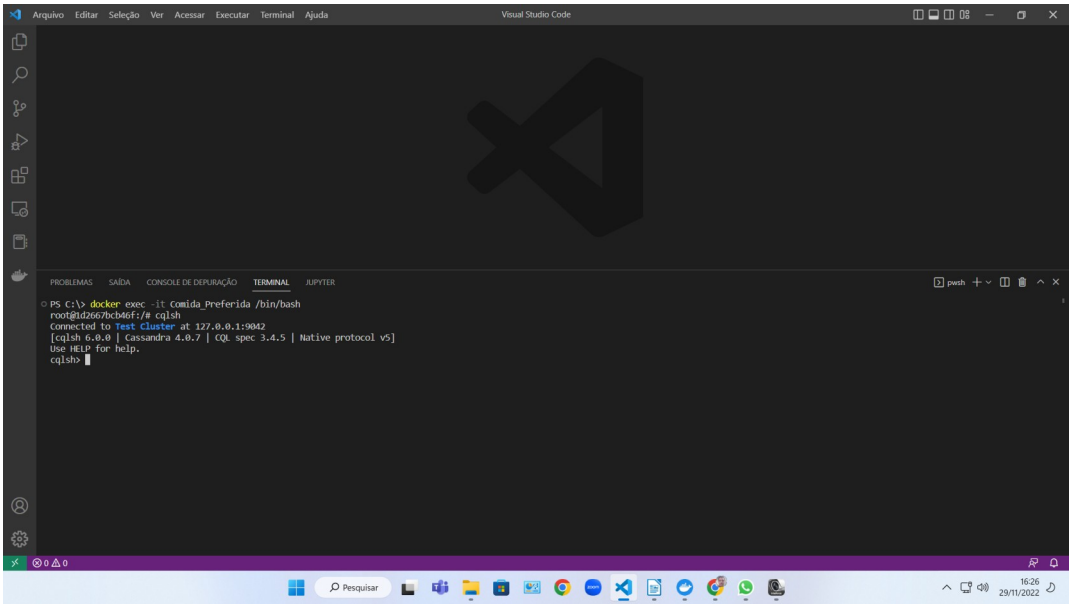
Iniciando meu container atráves de um terminal do (Visual Studio Code).

docker exec -it Comida_Preferida /bin/bash



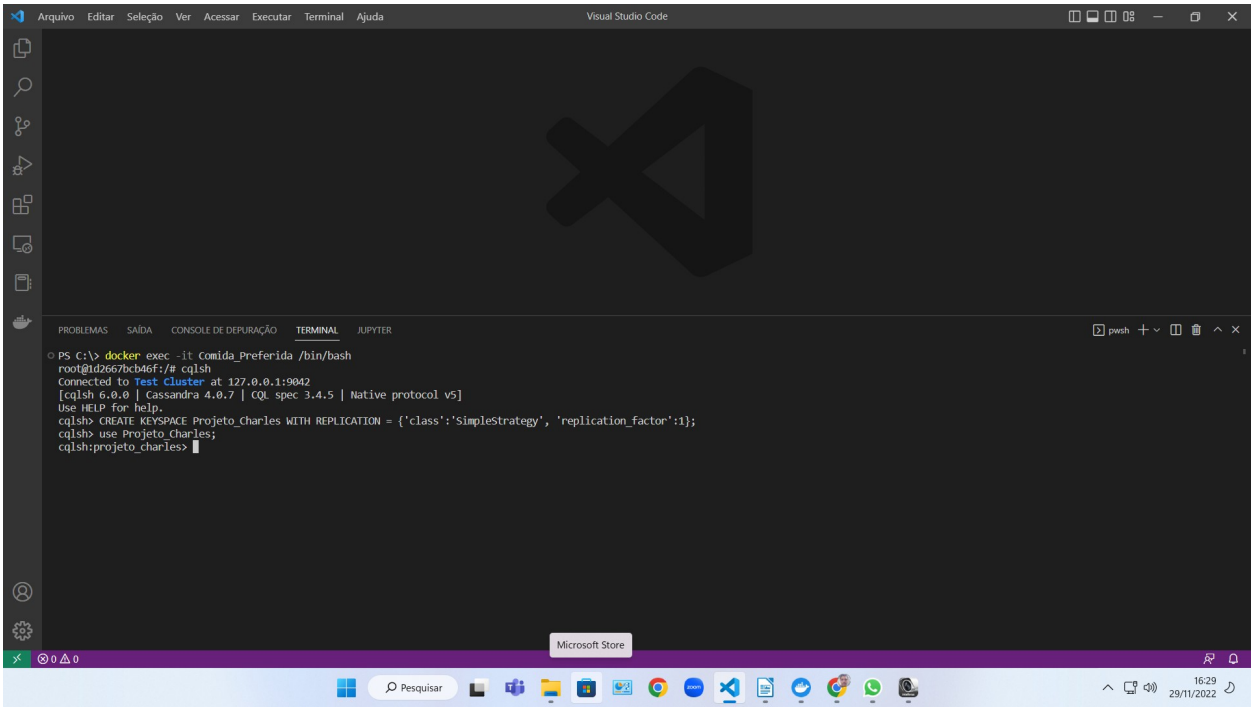
Linguagem de consulta Cassandra(CQL) é a principal linguagem para se comunicar com o Apache Cassandra.

Comando: CQLSH

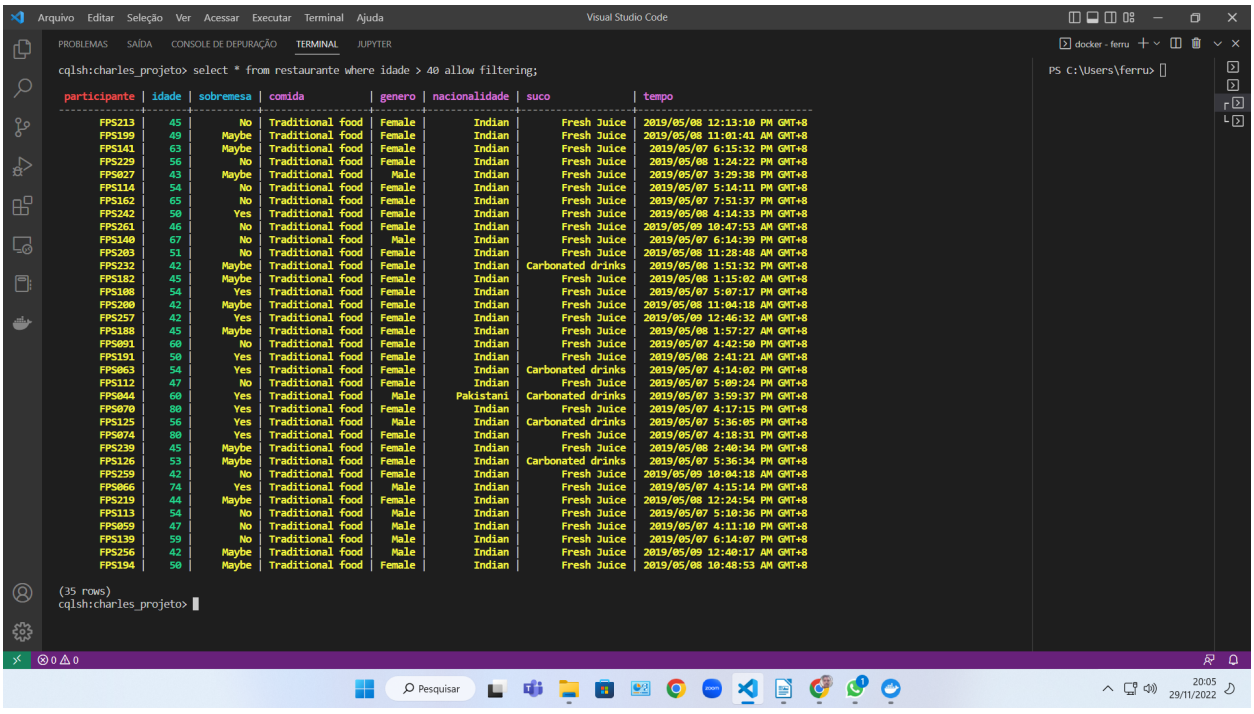


Criando a Keyspace e acessando:

CREATE KEYSPACE Projeto_Charles WITH REPLICATION = {'class':'SimpleStrategy', 'replication_factor':1};



1.4 Apresente o código e uma comprovação de sucesso de execução de uma consulta de leitura na base de dados carregada para a infraestrutura Cassandra.



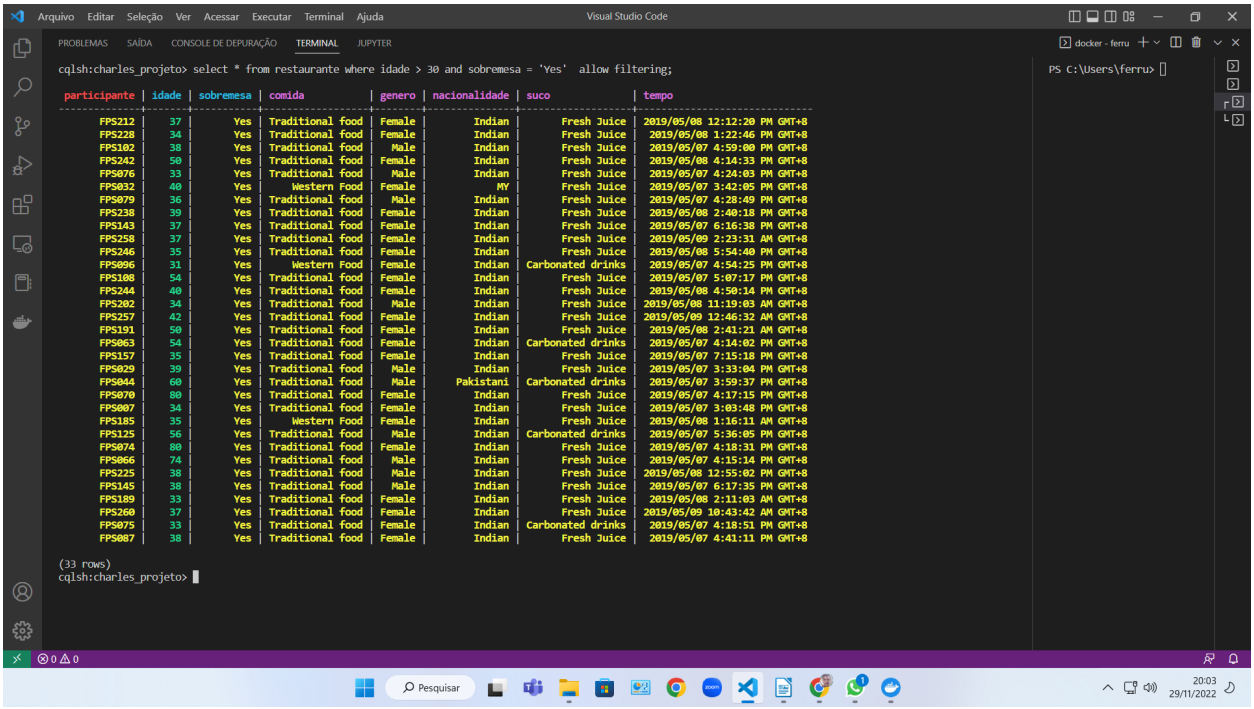
1.5 Utilizando linguagem Python, extraia o conjunto de informações necessário para dar prosseguimento ao processo de resposta à(s) pergunta(s) de negócio estabelecida(s) na questão 3. Apresente o código criado para a extração e uma comprovação de sucesso da operação.

1.6 Execute quaisquer operações de manipulação, ajuste e criação de variáveis necessárias para dar apoio à resposta do problema de negócio em estudo. Apresente códigos em Python de pelo menos uma operação de manipulação de dados e uma operação de criação de novas variáveis.

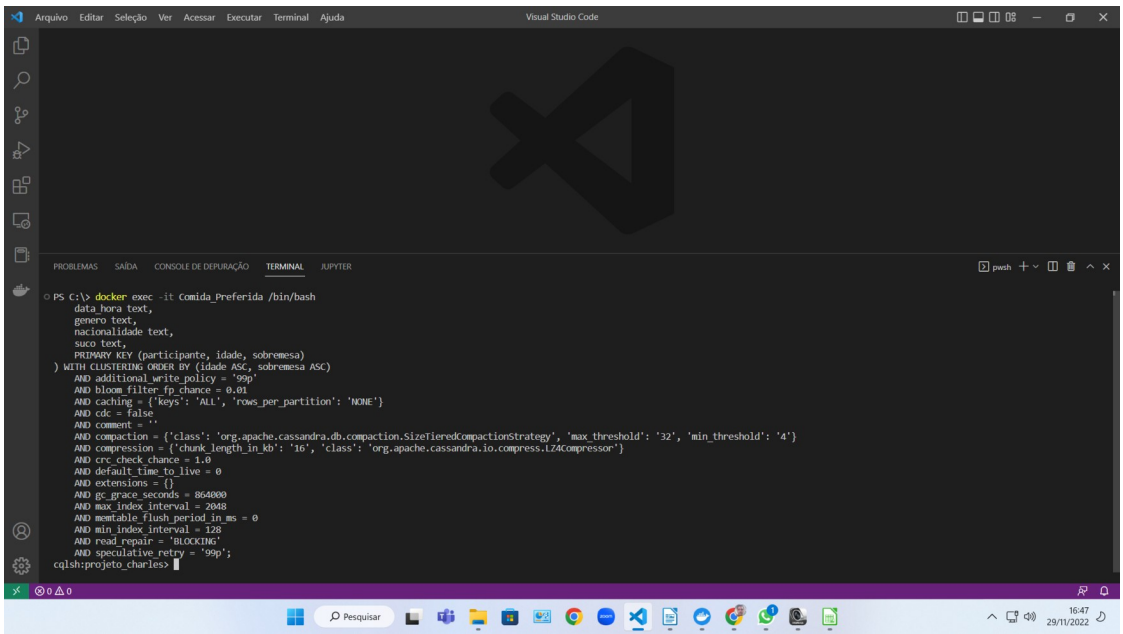
R. 1.1 Todos clientes com idade maior que 30 anos e que comeram sobremesa.

R. 1.2 Média da idade de todos os clientes do gênero masculino e que não comeram sobremesa.

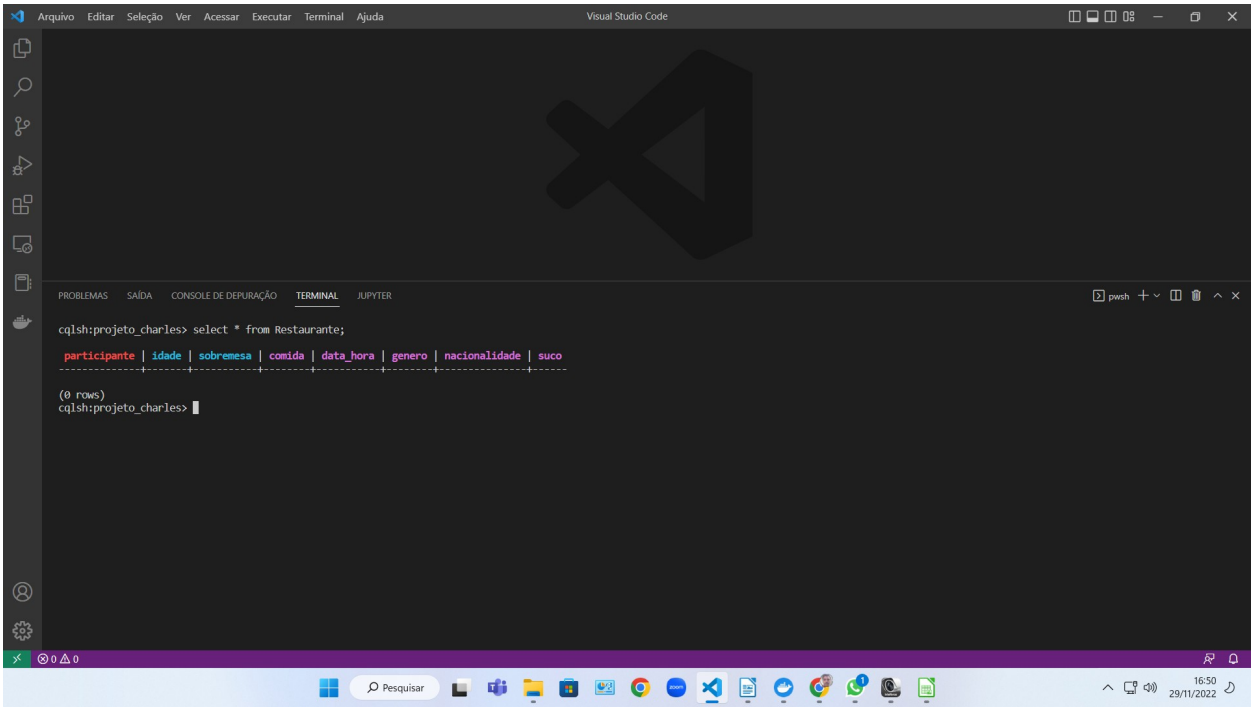
1.1



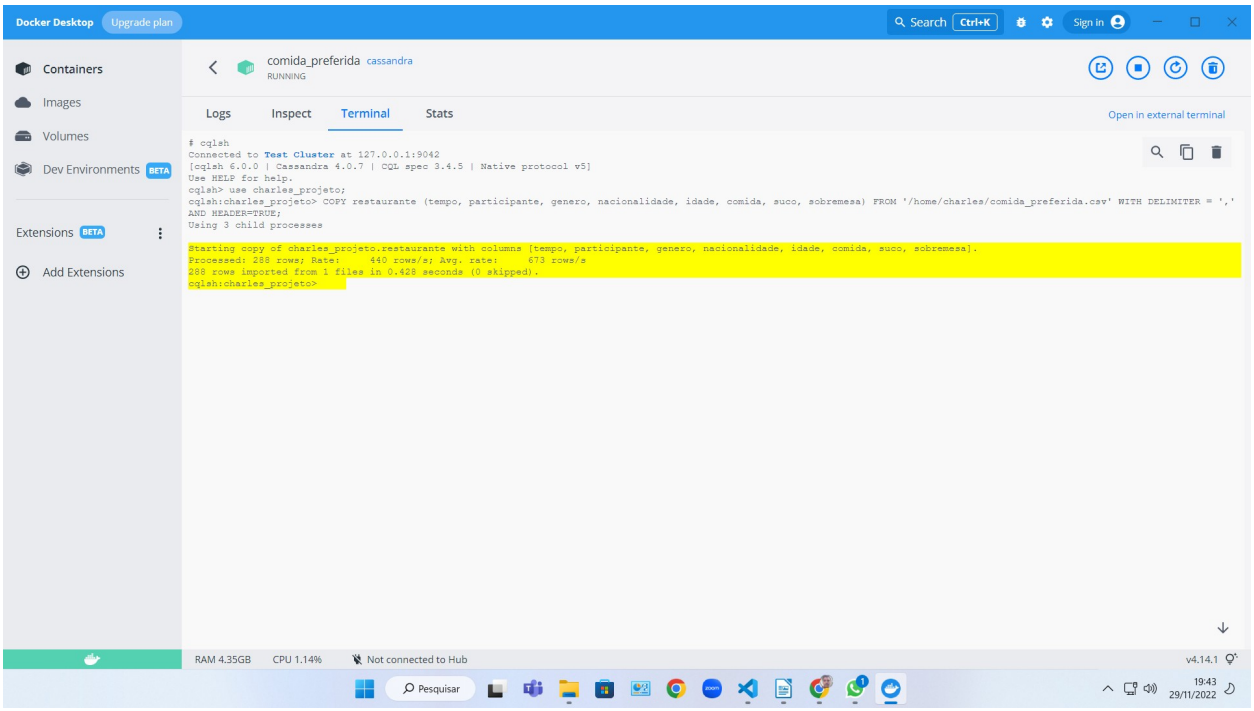
1.2



Select na tabela criada:



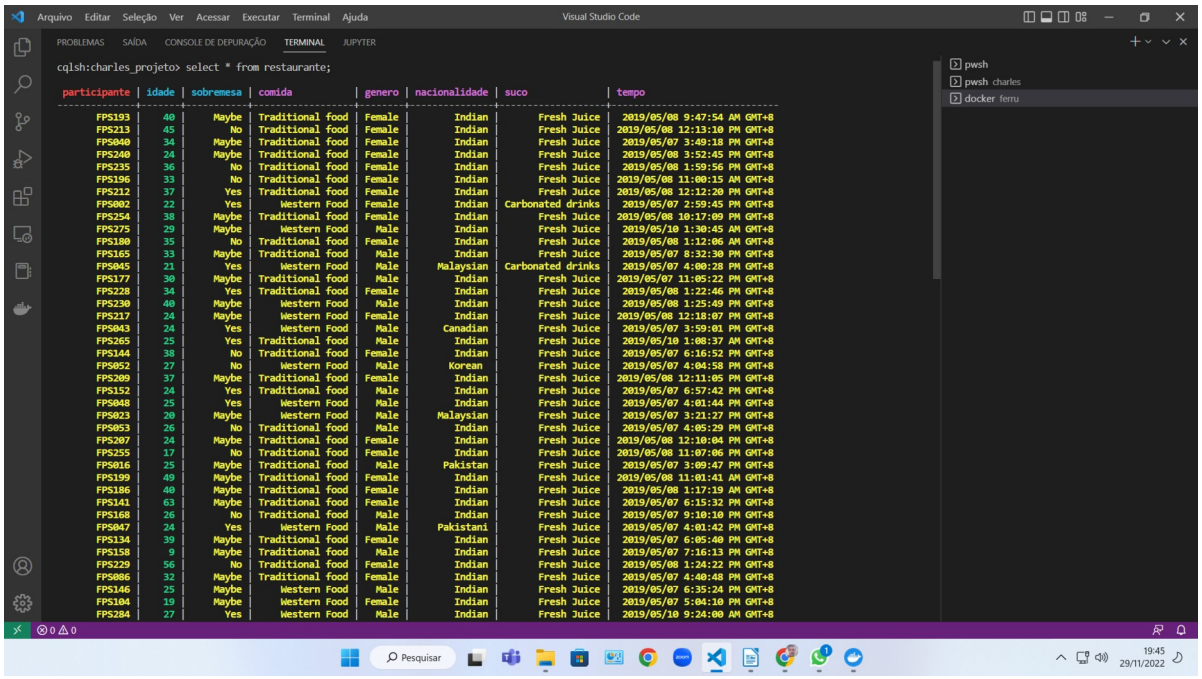
Importando os dados do arquivo comida_preferida.csv para minha tabela restaurante:



select * from restaurante where idade > 30 and sobremesa = 'Yes' allow filtering;

1.6 Exporte a base tratada final para formato CSV. Apresente o código em Python que demonstre a execução desta operação.

```
COPY restaurante (tempo, participante, genero, nacionalidade, idade, comida, suco, sobremesa) FROM
'/home/charles/comida_preferida.csv' WITH DELIMITER = ',' AND HEADER=TRUE;
```



1.7 Crie pelo menos uma visualização de dados que utilize um gráfico de barras e dê suporte à resposta para a(s) pergunta(s) de negócio estabelecida(s) na questão 3. Apresente a visualização gerada e como ela ajuda a responder a solução do problema proposto.

1.8 Crie pelo menos uma visualização de dados que utilize um gráfico de dispersão e dê suporte à resposta para a(s) pergunta(s) de negócio estabelecida(s) na questão 3. Apresente a visualização gerada e como ela ajuda a responder a solução do problema proposto.

Processando os dados e plotando alguns gráficos

Nesta etapa, vamos nos conectar as tabelas do Cassandra via Python

e faremos alguns processando nos dados com PySpark para respondermos

as perguntas iniciais sobre os dados. Também plotaremos alguns gráficos

para deixar a análise um pouco mais visual.

Instalando bibliotecas não existentes na máquina

!pip install findspark

Collecting findspark Downloading findspark-2.0.1-py2.py3-none-any.whl (4.4 kB) Installing collected packages: findspark Successfully installed findspark-2.0.1

Importando as bibliotecas

```
from pyspark import SparkContext, SparkConf

from pyspark.sql import *

from pyspark.sql.functions import *

import os

import sys

import findspark

import numpy

import pandas as pd

import matplotlib.pyplot as plt
```

Configurando o SparkConf

```
findspark.init()

os.environ['PYSPARK_SUBMIT_ARGS'] = '--packages com.datastax.spark:spark-cassandra’

conf = SparkConf()

conf.set('spark.cassandra.auth.username', 'cassandra')

conf.set('spark.cassandra.auth.password', 'cassandra')

conf.set('spark.cassandra.connection.host', 'localhost')

conf.set('spark.cassandra.connection.local_dc', 'dc1')
```

Criando o objeto SparkContext, através do objeto SparkConf configurado (conf)

```
sc = SparkContext().getOrCreate(conf) # Criando um objeto SQLContext do Spark também, que utiliza o objeto SparkContext sqlContext = SQLContext(sc)
```

```python
import matplotlib.pyplot as plt


x  = [42, 46, 47, 51, 54, 56, 59, 60, 65, 67]
y1 = [1, 1, 2, 1, 1, 2, 2, 1, 1, 1]
y2 = [1, 1, 2, 1, 1, 2, 2, 1, 1, 1]
plt.plot(x, y1, label="line L")
plt.plot(x, y2, label="line H")
plt.plot()

plt.xlabel("x axis")
plt.ylabel("y axis")
plt.title("Line Graph Example")
plt.legend()
plt.show()
```