import os

import pandas as pd

from cassandra.cluster import Cluster

from cassandra.auth import PlainTextAuthProvider

# Read environment variables

# secure\_connect\_bundle\_path = os.getenv('ASTRA\_DB\_SECURE\_CONNECT')

# application\_token = os.getenv('ASTRA\_DB\_TOKEN')

# Read environment variables for secure connect bundle and application token

secure\_connect\_bundle\_path = os.getenv('ASTRA\_DB\_SECURE\_CONNECT')

application\_token = os.getenv('ASTRA\_DB\_TOKEN')

# Debug prints to check environment variables

print(f"Secure Connect Bundle Path: {secure\_connect\_bundle\_path}")

print(f"Application Token: {application\_token}")

# Check if the environment variables are loaded correctly

if not secure\_connect\_bundle\_path or not os.path.exists(secure\_connect\_bundle\_path):

raise FileNotFoundError(f"Secure connect bundle not found at path: {secure\_connect\_bundle\_path}")

if not application\_token:

raise ValueError("Application token not found in environment variables")

# Connect to the Cassandra database using the secure connect bundle

session = Cluster(

cloud={"secure\_connect\_bundle": secure\_connect\_bundle\_path},

auth\_provider=PlainTextAuthProvider("token", application\_token),

).connect()

# Use the keyspace

session.set\_keyspace('catalog')

# Create a table

session.execute("CREATE TABLE IF NOT EXISTS ProductImageVectors (ProductId int PRIMARY KEY, ProductDesc text, price int);")

# Insert data

session.execute("INSERT INTO ProductImageVectors (id, name, age) VALUES (1, 'Proline Men Cream-Coloured Polo T-Shirt', 3000);")

# Select data

results = session.execute("SELECT \* FROM ProductImageVectors WHERE id = 1;")

for row in results:

print(row)

df = pd.read\_sql\_query(results, session)

print(df)