from cassandra.cluster import Cluster

from cassandra.auth import PlainTextAuthProvider

# Path to your Secure Connect Bundle

secure\_connect\_bundle\_path = < your Secure Connect Bundle>

# Your application token

application\_token = <Your application token>

# Setup authentication provider

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

# Connect to the Cassandra database using the secure connect bundle

cluster = Cluster(

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

auth\_provider=auth\_provider

)

session = cluster.connect()

# Define keyspace

keyspace = "catalog"

v\_dimension = 5

# Set the keyspace

session.set\_keyspace(keyspace)

# Verify connection by querying the system.local table

rows = session.execute("SELECT release\_version FROM system.local")

for row in rows:

print(f"Connected to Cassandra, release version: {row.release\_version}")

# Print the current keyspace

current\_keyspace = session.execute("SELECT keyspace\_name FROM system\_schema.keyspaces WHERE keyspace\_name = %s", [keyspace])

for row in current\_keyspace:

print(f"Connected to keyspace: {row.keyspace\_name}")

print("Connected to AstraDB and keyspace successfully!")

session.execute((

"CREATE TABLE IF NOT EXISTS {keyspace}.ProductImageVectors (ProductId INT PRIMARY KEY, ProductDesc TEXT, ImageURL text, ProductImageVector VECTOR<FLOAT,{v\_dimension}>);"

).format(keyspace=keyspace, v\_dimension=v\_dimension))

session.execute((

"CREATE CUSTOM INDEX IF NOT EXISTS idx\_ProductImageVectors "

"ON {keyspace}.ProductImageVectors "

"(ProductImageVector) USING 'StorageAttachedIndex' WITH OPTIONS = "

"{{'similarity\_function' : 'cosine'}};"

).format(keyspace=keyspace))

text\_blocks = [

(1, "Under colors of Benetton Men White Boxer Trunks","UndercolorsofBenetton-Men-White-Boxer\_b4ef04538840c0020e4829ecc042ead1\_images.jpg", [-0.0711570307612419, 0.0490173473954201, -0.0348679609596729, -0.0208837632089853, 0.0250527486205101]

),

(2, "Turtle Men Check Red Shirt","Turtle-Men-Check-Red-Shirt\_4982b2b1a76a85a85c9adc8b4b2d523a\_images.jpg" ,[-0.0678209140896797, 0.0918413251638412, 0.0087888557463884, -0.0005505480221473, 0.0586152337491512]),

(3, "United Colors of Benetton Men White Check Shirt","United-Colors-of-Benetton-Men-White-Check-Shirt\_13cfaff26872c298112a8e7da15c1e1d\_images.jpg" ,[-0.0697127357125282, 0.0486216545104980, -0.0169006455689669, -0.0160229168832302, 0.0137890130281448]

),

(4, "United Colors of Benetton Men Check White Shirts","UnitedColorsofBenetton-Men-Check-White-Shirts\_5bd8cae4fc61052a6f00cfcd69c4a936\_images.jpg" ,[-0.0499644242227077, 0.0566278323531151, -0.0294290613383055, -0.0070271748118103, 0.0289674568921328]

),

(5, "Wrangler-Men-Broad-Blue-Shirt","Wrangler-Men-Broad-Blue-Shirt\_8211520250143786-1.jpg" ,[-0.0581886917352676, 0.0378338471055031, 0.0425588376820087, -0.0423909239470959, 0.0186673272401094]

)

]

for block in text\_blocks:

id, text, text,vector = block

session.execute(

f"INSERT INTO {keyspace}.ProductImageVectors(ProductId, ProductDesc, ImageURL,ProductImageVector) VALUES (%s, %s,%s, %s)",

(id, text, text,vector)

)

ann\_query = (

f"SELECT ProductDesc, ImageURL,similarity\_cosine(ProductImageVector, [0.15, 0.1, 0.1, 0.35, 0.55]) as similarity FROM {keyspace}.ProductImageVectors "

"ORDER BY ProductImageVector ANN OF [0.15, 0.1, 0.1, 0.35, 0.55] LIMIT 2"

)

for row in session.execute(ann\_query):

print(f"[{row.productdesc}\" (sim: {row.similarity:.4f})")

# Print success message

print("Data with semantic match.")

ann\_query\_matching = (

f"SELECT ProductDesc, ImageURL,similarity\_cosine(ProductImageVector, [-0.0499644242227077, 0.0566278323531151, -0.0294290613383055, -0.0070271748118103, 0.0289674568921328]) as similarity FROM {keyspace}.ProductImageVectors "

"ORDER BY ProductImageVector ANN OF [-0.0499644242227077, 0.0566278323531151, -0.0294290613383055, -0.0070271748118103, 0.0289674568921328] LIMIT 2"

)

for row in session.execute(ann\_query\_matching):

print(f"[{row.productdesc}\" (sim: {row.similarity:.4f})")

print("Data with similar match.")

from cassandra.cluster import Cluster

from cassandra.auth import PlainTextAuthProvider

def connect\_to\_astra\_db(secure\_connect\_bundle\_path, application\_token):

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

cluster = Cluster(

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

auth\_provider=auth\_provider

)

session = cluster.connect()

return session

def create\_keyspace(session, keyspace):

session.execute(f"CREATE KEYSPACE IF NOT EXISTS {keyspace} WITH REPLICATION = {{ 'class' : 'SimpleStrategy', 'replication\_factor' : 1 }}")

session.set\_keyspace(keyspace)

def create\_product\_image\_vectors\_table(session, keyspace, v\_dimension):

session.execute(f"CREATE TABLE IF NOT EXISTS {keyspace}.ProductImageVectors (ProductId INT PRIMARY KEY, ProductDesc TEXT, ImageURL text, ProductImageVector VECTOR<FLOAT,{v\_dimension}>);")

session.execute(f"CREATE CUSTOM INDEX IF NOT EXISTS idx\_ProductImageVectors ON {keyspace}.ProductImageVectors (ProductImageVector) USING 'StorageAttachedIndex' WITH OPTIONS = {{'similarity\_function' : 'cosine'}};")

def insert\_product\_image\_vectors(session, keyspace, text\_blocks):

for block in text\_blocks:

id, text, image\_url, vector = block

session.execute(

f"INSERT INTO {keyspace}.ProductImageVectors(ProductId, ProductDesc, ImageURL, ProductImageVector) VALUES (%s, %s, %s, %s)",

(id, text, image\_url, vector)

)

def query\_product\_image\_vectors(session, keyspace, vector, limit):

ann\_query = (

f"SELECT ProductDesc, ImageURL, similarity\_cosine(ProductImageVector, {vector}) as similarity FROM {keyspace}.ProductImageVectors "

f"ORDER BY ProductImageVector ANN OF {vector} LIMIT {limit}"

)

return session.execute(ann\_query)

def print\_product\_image\_vectors(rows):

for row in rows:

print(f"[{row.productdesc}\" (sim: {row.similarity:.4f})")

secure\_connect\_bundle\_path = 'D:/TaskL/ecommerce/fashion/FashionRecommenderApp/astra-db-recommendations-starter/SCB/secure-connect-fashion.zip'

application\_token = 'AstraCS:ZlyMyCobxWSafNzEERRFGkJt:b8190d55067628fcadb3e4224afbfb92327cb33590d8ddb955f0718d86b4d1bf'

keyspace = "catalog"

v\_dimension = 5

text\_blocks = [

(1, "Under colors of Benetton Men White Boxer Trunks", "UndercolorsofBenetton-Men-White-Boxer\_b4ef04538840c0020e4829ecc042ead1\_images.jpg", [-0.0711570307612419, 0.0490173473954201, -0.0348679609596729, -0.0208837632089853, 0.0250527486205101]),

(2, "Turtle Men Check Red Shirt", "Turtle-Men-Check-Red-Shirt\_4982b2b1a76a85a85c9adc8b4b2d523a\_images.jpg", [-0.0678209140896797, 0.0918413251638412, 0.0087888557463884, -0.0005505480221473, 0.0586152337491512]),

(3, "United Colors of Benetton Men White Check Shirt", "United-Colors-of-Benetton-Men-White-Check-Shirt\_13cfaff26872c298112a8e7da15c1e1d\_images.jpg", [-0.0697127357125282, 0.0486216545104980, -0.0169006455689669, -0.0160229168832302, 0.0137890130281448]),

(4, "United Colors of Benetton Men Check White Shirts", "UnitedColorsofBenetton-Men-Check-White-Shirts\_5bd8cae4fc61052a6f00cfcd69c4a936\_images.jpg", [-0.0499644242227077, 0.0566278323531151, -0.0294290613383055, -0.0070271748118103, 0.0289674568921328]),

(5, "Wrangler-Men-Broad-Blue-Shirt", "Wrangler-Men-Broad-Blue-Shirt\_8211520250143786-1.jpg", [-0.0581886917352676, 0.0378338471055031, 0.0425588376820087, -0.0423909239470959, 0.0186673272401094])

]

session = connect\_to\_astra\_db(secure\_connect\_bundle\_path, application\_token)

create\_keyspace(session, keyspace)

create\_product\_image\_vectors\_table(session, keyspace, v\_dimension)

insert\_product\_image\_vectors(session, keyspace, text\_blocks)

vector = [0.15, 0.1, 0.1, 0.35, 0.55]

limit = 2

rows = query\_product\_image\_vectors(session, keyspace, vector, limit)

print\_product\_image\_vectors(rows)

print("Data with semantic match.")

vector\_matching = [-0.0499644242227077, 0.0566278323531151, -0.0294290613383055, -0.0070271748118103, 0.0289674568921328]

rows\_matching = query\_product\_image\_vectors(session, keyspace, vector\_matching, limit)

print\_product\_image\_vectors(rows\_matching)

print("Data with similar match.")