import psycopg2

from faker import Faker

import random

from datetime import datetime, timedelta

# Database connection setup

DB\_NAME = "26sqlll"

DB\_USER = "postgres"

DB\_PASSWORD = "postgres"

DB\_HOST = "localhost"

DB\_PORT = "5432"

fake = Faker()

try:

# Connect to PostgreSQL

conn = psycopg2.connect(

dbname=DB\_NAME,

user=DB\_USER,

password=DB\_PASSWORD,

host=DB\_HOST,

port=DB\_PORT

)

cur = conn.cursor()

print("Connected to the database!")

# Clear tables if needed (optional safety)

cur.execute("DELETE FROM analysis\_logs")

cur.execute("DELETE FROM Dataset")

cur.execute("DELETE FROM Users")

# Insert into Users table

roles = ['Analyst', 'Admin']

for user\_id in range(1, 201):

name = fake.name()

role = random.choice(roles)

cur.execute("""

INSERT INTO Users (user\_id, name, role)

VALUES (%s, %s, %s)

""", (user\_id, name, role))

# Insert into Dataset table

for record\_id in range(1, 201):

category = random.choice(['Sales', 'Marketing', 'Support', 'Finance', 'Logistics'])

value = round(random.uniform(100.0, 10000.0), 2)

timestamp = fake.date\_time\_between(start\_date='-6M', end\_date='now')

label = random.choice(['High', 'Medium', 'Low', 'Critical'])

cur.execute("""

INSERT INTO Dataset (record\_id, category, value, timestamp, label)

VALUES (%s, %s, %s, %s, %s)

""", (record\_id, category, value, timestamp, label))

# Insert into Analysis Logs table

for log\_id in range(1, 201):

user\_id = random.randint(1, 200) # assuming user\_id 1–200 exist

operation = random.choice([

'Average Value Computation',

'Trend Analysis',

'Outlier Detection',

'Category Summary',

'Label Distribution'

])

log\_time = fake.date\_time\_between(start\_date='-3M', end\_date='now')

cur.execute("""

INSERT INTO analysis\_logs (log\_id, user\_id, operation, log\_time)

VALUES (%s, %s, %s, %s)

""", (log\_id, user\_id, operation, log\_time))

# Commit changes

conn.commit()

print("✅ Successfully inserted 200 records into all tables.")

except Exception as e:

print("❌ Error:", e)

finally:

if conn:

cur.close()

conn.close()

print("Database connection closed.")