# Loading data from sources to staging tables

## Load files to staging tables from Azure object/blob store

## If it already exists, then it skips

## It then stores into deleta table in parquet format

%%pyspark
  
  
# Define the storage account and container
  
storage\_account\_name = "distributedanalytics"
  
container\_name = "healthdata-fs"
  
directory\_path = "data"
  
  
# Define the full path in which raw CSVs are placed
  
full\_path = f"abfss://{container\_name}@{storage\_account\_name}.dfs.core.windows.net/{directory\_path}"
  
  
  
fs = sc.\_jvm.org.apache.hadoop.fs.FileSystem.get(sc.\_jsc.hadoopConfiguration())
  
  
# List files
  
path = sc.\_jvm.org.apache.hadoop.fs.Path(full\_path)
  
files = fs.listStatus(path)
  
# Display the files
  
for file in files:
  
 file\_name = file.getPath().getName()
  
 if file\_name.endswith(".csv"):
  
 # Read the CSV file into a DataFrame
  
 print(f"the file is {full\_path}/{file\_name}")
  
 df = spark.read.option("header", "true").csv(f"{full\_path}/{file\_name}")
  
  
  
 # Define the Delta Lake table path
  
 delta\_table\_path = f"abfss://{container\_name}@{storage\_account\_name}.dfs.core.windows.net/delta/{file\_name.replace('.csv', '')}"
  
 # Check if the Delta table already exists
  
 if not fs.exists(sc.\_jvm.org.apache.hadoop.fs.Path(delta\_table\_path)):
  
 # Read the CSV file into a DataFrame
  
 df = spark.read.option("header", "true").csv(f"{full\_path}/{file\_name}")
  
 df.write.format("delta").mode("overwrite").option("delta.columnMapping.mode", "name").save(delta\_table\_path)
  
 delta\_df = spark.read.format("delta").load(delta\_table\_path)
  
  
  
 table\_name = file\_name.replace('.csv', '')
  
 # Write data to Lake database table
  
 delta\_df.write.format("parquet").saveAsTable(f"default.{table\_name}")
  
  
  
 print(f"Delta Lake table created for {file\_name} at {delta\_table\_path}")
  
 else:
  
 print(f"Delta Lake table already exists for {file\_name}, skipping...")

{"execution\_finish\_time":"2025-03-17T16:24:23.258661Z","execution\_start\_time":"2025-03-17T16:23:46.772071Z","livy\_statement\_state":"available","normalized\_state":"finished","parent\_msg\_id":"9ef06496-ffc9-4138-9f00-54ce9d6ca4f8","queued\_time":"2025-03-17T16:18:40.2815915Z","session\_id":"34","session\_start\_time":"2025-03-17T16:18:40.2829824Z","spark\_jobs":null,"spark\_pool":"ghparkcluster","state":"finished","statement\_id":2,"statement\_ids":[2]}

the file is abfss://healthdata-fs@distributedanalytics.dfs.core.windows.net/data/CHEAspercentGDP.csv  
Delta Lake table already exists for CHEAspercentGDP.csv, skipping...  
the file is abfss://healthdata-fs@distributedanalytics.dfs.core.windows.net/data/CHEInMillionConstant2022USD.csv  
Delta Lake table already exists for CHEInMillionConstant2022USD.csv, skipping...  
the file is abfss://healthdata-fs@distributedanalytics.dfs.core.windows.net/data/CHEPerCapitaUSD.csv  
Delta Lake table already exists for CHEPerCapitaUSD.csv, skipping...  
the file is abfss://healthdata-fs@distributedanalytics.dfs.core.windows.net/data/DoctorsUK.csv  
Delta Lake table already exists for DoctorsUK.csv, skipping...  
the file is abfss://healthdata-fs@distributedanalytics.dfs.core.windows.net/data/GCEInmillionconstant2022USD.csv  
Delta Lake table already exists for GCEInmillionconstant2022USD.csv, skipping...  
the file is abfss://healthdata-fs@distributedanalytics.dfs.core.windows.net/data/LifeExpectancyBirthAnd60.csv  
Delta Lake table already exists for LifeExpectancyBirthAnd60.csv, skipping...  
the file is abfss://healthdata-fs@distributedanalytics.dfs.core.windows.net/data/LifeExpetancyAtBirth.csv  
Delta Lake table already exists for LifeExpetancyAtBirth.csv, skipping...  
the file is abfss://healthdata-fs@distributedanalytics.dfs.core.windows.net/data/MonthlyDiagnostics2024\_13CLR.csv  
Delta Lake table already exists for MonthlyDiagnostics2024\_13CLR.csv, skipping...  
the file is abfss://healthdata-fs@distributedanalytics.dfs.core.windows.net/data/NHAIndicators.csv  
Delta Lake table already exists for NHAIndicators.csv, skipping...  
the file is abfss://healthdata-fs@distributedanalytics.dfs.core.windows.net/data/PHCExpenditureTrends.csv  
Delta Lake table already exists for PHCExpenditureTrends.csv, skipping...  
the file is abfss://healthdata-fs@distributedanalytics.dfs.core.windows.net/data/UKGDPPriceindex2022is100.csv  
Delta Lake table already exists for UKGDPPriceindex2022is100.csv, skipping...  
the file is abfss://healthdata-fs@distributedanalytics.dfs.core.windows.net/data/WorldHealthData.csv  
Delta Lake table already exists for WorldHealthData.csv, skipping...  
the file is abfss://healthdata-fs@distributedanalytics.dfs.core.windows.net/data/gdppercapitaworldbank.csv  
Delta Lake table already exists for gdppercapitaworldbank.csv, skipping...  
the file is abfss://healthdata-fs@distributedanalytics.dfs.core.windows.net/data/ons\_nhs\_expenditure.csv  
Delta Lake table already exists for ons\_nhs\_expenditure.csv, skipping...  
the file is abfss://healthdata-fs@distributedanalytics.dfs.core.windows.net/data/uknetmigration.csv  
Delta Lake table already exists for uknetmigration.csv, skipping...  
the file is abfss://healthdata-fs@distributedanalytics.dfs.core.windows.net/data/ukpopulationbyageONS.csv  
Delta Lake table already exists for ukpopulationbyageONS.csv, skipping...

# Listing of all tables

from delta.tables import DeltaTable
  
  
all\_tables = spark.sql("SHOW TABLES").collect()
  
delta\_tables = []
  
  
for table in all\_tables:
  
 table\_full\_name = f"{table.tableName}"
  
 print(table\_full\_name)
  
 if DeltaTable.isDeltaTable(spark, table\_full\_name):
  
 delta\_tables.append(table\_full\_name)
  
  
for delta\_table in delta\_tables:
  
 print(delta\_table)

{"execution\_finish\_time":"2025-03-17T16:24:47.4511524Z","execution\_start\_time":"2025-03-17T16:24:23.2710293Z","livy\_statement\_state":"available","normalized\_state":"finished","parent\_msg\_id":"56426761-9eb6-4ae4-a445-781cb12e0145","queued\_time":"2025-03-17T16:18:40.2823692Z","session\_id":"34","session\_start\_time":null,"spark\_jobs":null,"spark\_pool":"ghparkcluster","state":"finished","statement\_id":3,"statement\_ids":[3]}

chepercapitausd  
doctorsuk  
lifeexpectancybirthand60  
nhaindicators  
phcexpendituretrends  
worldhealthdata  
cheaspercentgdp  
cheinmillionconstant2022usd  
gceinmillionconstant2022usd  
monthlydiagnostics2024\_13clr  
ukgdppriceindex2022is100  
gdppercapitaworldbank  
uknetmigration  
ukpopulationbyageons  
lifeexpetancyatbirth  
ons\_nhs\_expenditure