from pyspark.sql import SparkSession

from pyspark.sql.functions import col, to\_timestamp, hour, dayofweek, percentile\_approx, coalesce

spark = SparkSession.builder.appName("OpenAQBatch").getOrCreate()

spark.sparkContext.setLogLevel("WARN")

# 1) Cargar

df\_raw = (spark.read

.option("header", True)

.csv("data/air\_quality\_openaq.csv"))

# 2) Filtrar PM2.5 y tipificar

# - "datetime" es tu columna de tiempo

# - por si el formato varía, probamos varios patrones con coalesce

ts = coalesce(

to\_timestamp(col("datetime")),

to\_timestamp(col("datetime"), "yyyy-MM-dd'T'HH:mm:ss'Z'"),

to\_timestamp(col("datetime"), "yyyy-MM-dd HH:mm:ss"),

to\_timestamp(col("datetime"), "yyyy/MM/dd HH:mm:ss")

)

df = (df\_raw

.filter(col("parameter").isin("pm25", "PM2.5", "pm2.5"))

.withColumn("station\_id", col("location"))

.withColumn("timestamp", ts)

.withColumn("pm25", col("value").cast("double"))

.filter(col("timestamp").isNotNull() & col("station\_id").isNotNull())

.filter((col("pm25") >= 0) & (col("pm25") <= 1000))

.withColumn("hour", hour(col("timestamp")))

.withColumn("dow", dayofweek(col("timestamp")))

)

# 3) Salidas

df.write.mode("overwrite").parquet("output/cleansed")

hourly = (df.groupBy("station\_id", "hour").agg({"pm25":"avg"}))

hourly.write.mode("overwrite").parquet("output/aggregates\_hourly")

p90 = (df.groupBy("station\_id")

.agg(percentile\_approx("pm25", 0.90).alias("pm25\_p90")))

p90.write.mode("overwrite").parquet("output/aggregates\_percentiles")

spark.stop()

oliver

ahi esta el codigo

esta cargado

el video usted me dice cuando

ese es el archivo python