Lab2-Task3: Guided solution for data Ingestion with PySpark: FlightsRaw

Overview

read data from a CSV file named **flights_raw.csv**, transform the DataFrame to adhere to relational database norms, and write it back to S3 in the Parquet file format.

Step 1: Initialize the Spark Session

Create a Python script named *data_parser_flights_raw.py* in a folder called *exercises_two*.

```
# You initiate a Spark session to use Spark SQL's DataFrame API.
# A Spark Session is a combined entry point of Spark Context and SQL Context

from pyspark.sql import SparkSession
spark =SparkSession.builder\
.master("local")\
.appName ('ex2_flights')\
.getOrCreate()
```

Step 2: Read the Raw Flights Data

```
# Read raw flight data from S3 into a Spark DataFrame.

flights_raw_df = spark.read.csv('s3a://spark/data/raw/flights_raw/', header=True)
```

Step 3: Perform Data Transformation

```
from pyspark.sql import functions as F
from pyspark.sql import types as T

# Perform transformations on the raw DataFrame to make it relational:
# Cast relevant columns to IntegerType for numeric processing.
# Rename columns for better readability and understanding.
flight_df = flights_raw_df.select(
    F.col('DayofMonth').cast(T.IntegerType()).alias('day_of_month'),
    F.col('DayOfWeek').cast(T.IntegerType()).alias('day_of_week'),
    F.col('Carrier').alias('carrier'),
    F.col('OriginAirportID').cast(T.IntegerType()).alias('origin_airport_id'),
    F.col('DestAirportID').cast(T.IntegerType()).alias('dest_airport_id'),
    F.col('DepDelay').cast(T.IntegerType()).alias('dep_delay'),
    F.col('ArrDelay').cast(T.IntegerType()).alias('arr_delay'))
```

Step 4: Save the Transformed Data to S3

```
# Save the transformed DataFrame back to S3 in the Parquet format.
# If the directory already exists, the 'overwrite' mode will replace it.
flight_df.write.parquet('s3a://spark/data/source/flights_raw/', mode='overwrite')
```

Step 5: Stop the Spark Session

```
# Terminate the Spark session to release its resources.
spark.stop()
```

By completing these steps, you'll be able to read raw flight data, transform it into a structured and relational format, and save it back to S3 as a Parquet file. This exercise allows you to practice essential skills in data engineering and ETL processes.

Step 6 - Full Code Solution

Data Parsing - flightsRaw
Folder Name: exercises_two
File Name: data_parser_flights_raw.py

```
from pyspark.sql import SparkSession
from pyspark.sql import functions as F
from pyspark.sql import types as T

spark = SparkSession.builder.master("local").appName('ex2_flights').getOrCreate()

flights_raw_df = spark.read.csv('s3a://spark/data/raw/flights_raw/', header=True)


flight_df = flights_raw_df.select(
    F.col('DayofMonth').cast(T.IntegerType()).alias('day_of_month'),
    F.col('DayOfWeek').cast(T.IntegerType()).alias('day_of_week'),
    F.col('Carrier').alias('carrier'),
    F.col('OriginAirportID').cast(T.IntegerType()).alias('origin_airport_id'),
    F.col('DestAirportID').cast(T.IntegerType()).alias('dest_airport_id'),
    F.col('DepDelay').cast(T.IntegerType()).alias('dep_delay'),
    F.col('ArrDelay').cast(T.IntegerType()).alias('arr_delay'))


flight_df.write.parquet('s3a://spark/data/source/flights_raw/', mode='overwrite')

spark.stop()
```