Task #1

**import org.apache.spark.sql.functions.\_**

**val flights = spark.read.option("header",true)**

**.csv("data/flights.csv")**

**val result = flights.groupBy("ORIGIN\_AIRPORT")**

**.agg(sum("DEPARTURE\_DELAY")**

**.as("TOTAL\_DELAY"))**

**.orderBy(desc("TOTAL\_DELAY"))**

**result.show(numRows=1)**

**flights.createOrReplaceTempView("flights")**

**val resultSQL = spark.sql("""SELECT ORIGIN\_AIRPORT, SUM(DEPARTURE\_DELAY) as TOTAL\_DELAY**

**FROM flights**

**GROUP BY ORIGIN\_AIRPORT**

**ORDER BY TOTAL\_DELAY DESC""")**

**resultSQL.show(numRows=1)**

Text

Description automatically generated with medium confidence

TASK#2

**val result = spark.sql("""**

**WITH temp AS(SELECT**

**CASE WHEN ISNULL(departure\_time) THEN "2015-01-01"**

**ELSE TO\_DATE(CONCAT (YEAR,"-",MONTH,"-",DAY))**

**END as departure\_date,**

**coalesce(DATE\_FORMAT(TO\_TIMESTAMP(departure\_time, "HHmm"), "HH:mm:ss"), "00:00:00") as departure\_time,**

**airline, flight\_number**

**FROM flights**

**)**

**SELECT departure\_date, departure\_time, airline, flight\_number,**

**COUNT(\*) OVER(PARTITION BY departure\_date, airline) AS airline\_daily\_flights\_count,**

**ROW\_NUMBER() OVER(PARTITION BY departure\_date, airline ORDER BY departure\_time) AS daily\_flight\_serial\_number,**

**(int(to\_timestamp(departure\_time)) - int(to\_timestamp(lag(departure\_time) OVER(PARTITION BY departure\_date, airline ORDER BY departure\_time))))/(60) as time\_since\_previous\_departure**

**FROM temp**

**""")**

**result.show(numRows=5)**

**Table

Description automatically generated**

**var golden1 = spark.read.parquet("query1")**

**golden1.createOrReplaceTempView("goldenData")**

**result.createOrReplaceTempView("kaggleData")**

**val diff = spark.sql("""**

**SELECT \* FROM(**

**SELECT \* FROM kaggleData**

**UNION ALL**

**SELECT \* FROM goldenData) a**

**GROUP BY departure\_date,**

**departure\_time,**

**airline,**

**flight\_number,**

**airline\_daily\_flights\_count,**

**daily\_flight\_serial\_number,**

**time\_since\_previous\_departure**

**having count(\*) >= 1**

**""")**

**diff.show()**

**Table

Description automatically generated**

**diff.toDF()**

**diff.write.mode("overwrite").csv("logCSV/task2")**

**diff.write.mode("overwrite").parquet("logPARQ/task2")**

**TASK2 DF API**