# Big Data Infrastructures & Technology - Assignment 3

In this assignment your goal is to learn how to use SQL to manipulate data with Spark. In case you are completely unfamiliar with using SQL, here is a set of online exercises that will instruct you how to use SQL:
https://duckdb.org/docs/sql/tutorial

## Airline Information Scenario

In this assignment we will work with the ontime dataset, which contains information about flights in the United States. The schema of the table is provided below.

| column_name | column_type | column_name | column_type |
|---|---|---|---|
| year | BIGINT | deststatefips | VARCHAR |
| quarter | BIGINT | deststatename | VARCHAR |
| month | BIGINT | destwac | DOUBLE |
| dayofmonth | BIGINT | crsdeptime | DOUBLE |
| dayofweek | BIGINT | deptime | DOUBLE |
| flightdate | VARCHAR | depdelay | DOUBLE |
| uniquecarrier | VARCHAR | depdelayminutes | DOUBLE |
| airlineid | DOUBLE | depdel15 | DOUBLE |
| carrier | VARCHAR | departuredelaygroups | DOUBLE |
| flightnum | VARCHAR | deptimeblk | VARCHAR |
| originairportid | BIGINT | crsarrtime | DOUBLE |
| originairportseqid | BIGINT | arrtime | DOUBLE |
| origincitymarketid | BIGINT | arrdelay | DOUBLE |
| origin | VARCHAR | arrdelayminutes | DOUBLE |
| origincityname | VARCHAR | arrdel15 | DOUBLE |
| originstate | VARCHAR | arrivaldelaygroups | DOUBLE |
| originstatefips | VARCHAR | arrtimeblk | VARCHAR |
| originstatename | VARCHAR | cancelled | BIGINT |
| originwac | DOUBLE | diverted | BIGINT |
| destairportid | BIGINT | crselapsedtime | DOUBLE |
| destairportseqid | BIGINT | actualelapsedtime | DOUBLE |
| destcitymarketid | BIGINT | flights | DOUBLE |
| dest | VARCHAR | distance | DOUBLE |
| destcityname | VARCHAR | distancegroup | BIGINT |
| deststate | VARCHAR | | |

We start off by installing the packages we need and downloading the data.

**Setup [1]:**

```
!wget
https://github.com/Mytherin/NASAData/raw/master/ontime.parquet
!pip install pyspark plotly
```

We start off by initializing a Spark context again:

**In [1]:**

```
from pyspark.sql import SparkSession, Row
spark = SparkSession.builder.master("local").config("spark.ui.enabled",
"false").getOrCreate()
sc = spark.sparkContext
```

Afterwards, we load the tables from the Parquet files into Spark.

**In [2]:**

```
ontime = spark.read.parquet("ontime.parquet")
```

Then we register the loaded DataFrames into Spark as a view, which allows us to use it in subsequent SQL queries:

**In [3]:**

```
ontime.createOrReplaceTempView("ontime")
```

Now we can query the SQL DataFrames using the spark.sql function.

**In [4]:**

```
spark.sql("SELECT * FROM ontime LIMIT 5").show()
```

**Out [4]:**

```
+----+-------+-----+----------+---------+----------+
|year|quarter|month|dayofmonth|dayofweek|flightdate|...
+----+-------+-----+----------+---------+----------+...
|2017|      1|    2|        26|        7|2017-02-26|...
|2017|      1|    2|        26|        7|2017-02-26|...
|2017|      1|    2|        26|        7|2017-02-26|...
|2017|      1|    2|        26|        7|2017-02-26|...
|2017|      1|    2|        26|        7|2017-02-26|...
+----+-------+-----+----------+---------+----------+
```

## Multi-line Strings

You can use Python's multiline strings to make it nicer to write complex SQL statements. This can be done by using three quotes instead of the standard single quote. For example:

```
spark.sql('''
SELECT *
FROM ontime LIMIT 5
''').show();
```

## Assignment

Use SQL queries to find answers to the following questions:

1. What **date** has the **highest amount** of flights?
2. What are the **carrier**, **origin city** and **destination city** from the three flights with the **highest departure delay?** How high is that delay in minutes?
3. In the months of **June, July** and **August**, how many flights are there for each day of the week (i.e. how many flights are there on Monday, how many are there on Tuesday, etc)?

## Bonus Assignment

1. What are the three carriers that have the **highest percentage** of flights with a **departure delay** of 10 minutes or more, and how high is that percentage?
2. What are the three flights that **made up the most time lost time** during their flight (i.e. the arrival delay is lower than the departure delay)? What are the three flights that **lost the most time** during their flight?
3. What is the flight route that had the most **diverted** flights? What is the flight route that had the most **cancelled** flights?
4. What carrier had the most **cancelled flights** per kilometer their airplanes have travelled?