School of Computer Sciences
Uwe Roehm
Sidath Randeni Kadutpitige, Chris Natoli

DATA3404: Scalable Data Management **1.Sem./2023**

Assignment 1: SQL in PostgreSQL vs Databricks

Group Assignment (10%) **17/04/2023**

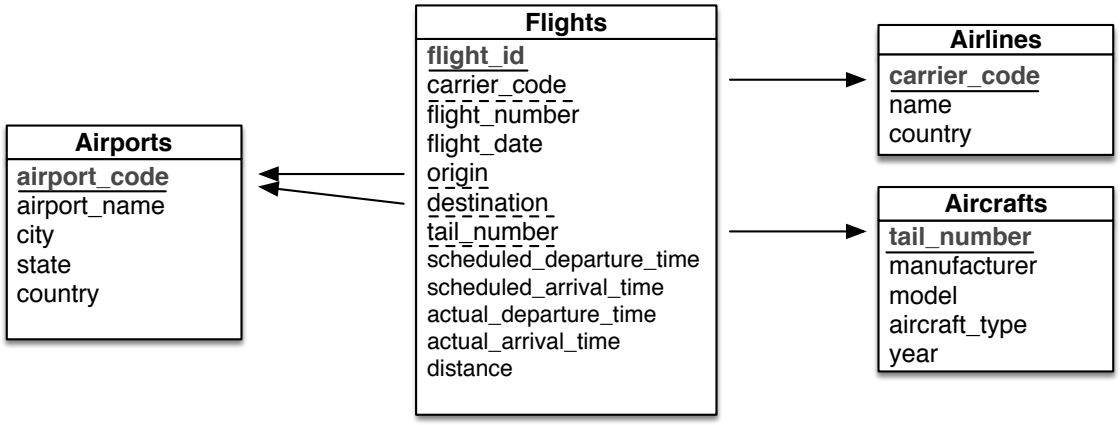
Introduction

This is the first part of the major practical assignment of DATA3404 in which you have to write a series of SQL statements and Databricks / Apache Spark queries to analyse an air traffic data set. We provide you with the schema and dataset. Your task is to implement some given data analysis queries on both platforms, and to write a short report on your results and the query plans.

You find links to online documentation, data, and hints on tools and schema needed for this assignment in the 'Assignment' section in Canvas modules.

Data Set Description

This assignment is based on an Aviation On-time data set which includes data about *airports*, *airlines*, *aircrafts*, and *flights*. This data set has the following structure (primary keys are underlined):



You will be provided with scaffold Jupyter notebook files for setting up your Databricks workspace similar to the introductory tutorial. Note that you can take a naïve approach to processing flights – if a flight has a row in the flights csv file, it is a flight – regardless of whether it was cancelled, or if it is somehow duplicated or any other data quality issues exist. In general, if there is missing data you need for a query, you can ignore that row.

Downloading Datasets

For the raw CSVs to use in SQL, links will be available in Canvas Modules.
For Databricks, use the Assignment Bootstrap notebook to handle this downloading into DBFS.

Questions

There are two question sections:

- In the first section, each student chooses an individual question they would like to answer;
- In the second section, each team answers **ALL** the questions in this section together.

Individual Questions - Choose one for each team member to complete (No need to complete all the questions)

Each team member chooses a single and distinct question to answer from the following:

1. **Determine the name of the 3 airlines with the most aircrafts.**

   Expected SQL output: Three row table with columns: name, count_of_aircrafts
   Execution time comparison: Compare the execution times of your query on PostgreSQL and on Spark/Databricks for the small and the medium datasets.

2. **Determine which airline departs late the most. (A flight is late to depart if and only if its actual departure time is after its scheduled departure time.)**

   Expected SQL output: One row table with columns: name, count_of_late_departures
   Execution time comparison: Compare the execution times of your query on PostgreSQL and on Spark/Databricks for the small and the medium datasets.

3. **Determine which model of aircraft has visited which state the most.**

   Expected SQL output: One row table with columns: aircraft_model, state, count_of_flights
   Execution time comparison: Compare the execution times of your query on PostgreSQL and on Spark/Databricks for the small and the medium datasets.

4. **Determine the top two airlines with the longest total distance flown.**

   Expected SQL output: Two row table with columns: airline, total_distance
   Execution time comparison: Compare the execution times of your query on PostgreSQL and on Spark/Databricks for the small and the medium datasets.

Team Questions - All must be answered

This section contains the group questions that **ALL** must be answered.

1. **Team Question 1: Determine the airline with the largest accumulated delay (arrival + departure), and which aircraft model of that airline contributed most (in percentage) to the total lateness of that airline.** Ignore NULL models.

   - **Expected SQL Output**: One row table with columns: airline_name, total_airline_delay, manufacturer, model, cumulative_lateness_of_model, percentage_of_total_lateness_for_airline
   - **Formatting:** ignore NULL models; give delay in minutes; percentage rounded to one decimal
   - Compare the **query execution plans** between the two systems, i.e. of both PostgreSQL and Spark/Databricks for *either* the small *or* the medium data size.

2. **Team Question 2:**
   We call any flight with a duration longer than the average flight time of all flights a long flight. Determine the top 5 airports which have the most long flights arriving. For each of those airports, determine the airline with the most long flights arriving there.

   - **Expected SQL Output**: One row table with columns: airport_code, airport_name, number_of_longflight_arrivals, average_longflight_duration, airline_name_with_most_longflight_arrivals, number_of_longflight_arrivals_of_airline
   - Compare the **query execution plans** for different data sizes, i.e. of *either* PostgreSQL *or* Spark/Databricks on the small *and* the medium (or large) dataset.

Deliverables and Submission Details

There are three deliverables per group:

1. a brief **report/documentation** outlining your outputs; and a
2. **source code - SQL** as a single SQL file that answers the chosen individual and all the team questions; and a
3. **source code - Jupyter notebook** as a single .DBC archive or SQL source file that answers the chosen individual and all the team questions.

Here are the specifics associated with each deliverable item.

Report

Filename recommendation: data3404_y23s1_assignment_task1_tutgroupname_assignmentgroupnum.pdf

- Your group name, including tutorial code and group number
- The answers (output) you receive for each question when executing against the small and medium datasets for SQL/Postgres, and against the small, medium, and large datasets for SparkDatabricks.
- For the individual queries, compare the execution times between PostgreSQL and Databricks.
- For the two team queries, compare the query execution plans using `EXPLAIN` on the small and medium datasets for both PostgreSQL and Databricks. Include a short paragraph describing the query plan differences between the two platforms and why that might be the case.
- A short explanation of 'what is going on' in the general sense for each SQL statement. **Note** that this does not need to go into technical detail (that will come in Assignment 2) - instead you should essentially explain what each SQL clause or function is there for; why you used it and what effect you expect it to have. A short paragraph for each question is plenty.
- A group contribution statement with the following headings:
  - Your group members' names and SIDs
  - Which Question 1 subquestion they completed

  - Whether they contributed meaningfully to questions 2 and 3, and the report (yes or no)

- This does not have a strict page limit, but you should keep it relevant to ensure feedback can be useful. In particular:
  - Do not include large code dumps. You are already submitting all of your code. Use 1 or 2 line snippets if you absolutely must. Remember that including your code is not explaining your code.
  - Do not keep writing more in the hope that some of what you include will be correct. You are more likely to run into issues including incorrect information than you are to gain by including correct information.

SQL File

Filename recommendation: data3404_y23s1_assignment_task1_tutgroupname_assignmentgroupnum.sql
A single .SQL file that contains all of your completed questions. Use /*comments*/ to indicate where each question starts and ends, and importantly for question 1 the name and SID of the group member who completed that sub-question. This .SQL file must be able to be run against a Postgres installation that has the datasets imported with the appropriate schema (i.e., the tables have been created and populated already) and return the correct results as indicated in your report. Try to answer each question with just a single (complex) SQL query, though the creation of utility functions, e.g. for the delay computation, are allowed.

Jupyter DBC Archive or SQL Source File

Filename recommendation:data3404_y23s1_assignment_task1_tutgroupname_assignmentgroupnum.dbc
A single Jupyter SQL source file (or .DBC archive) that contains all of your completed questions. Use markdown cells to indicate the same information as for the SQL file. This file must be able to be run attached to a Databricks Community cluster that has had the Assignment Bootstrap notebook run on it, and no other configurations made.

Due Date: All deliverables are due in Week 10, no later than **Sunday 7th May**. Late submission penalty: -5% of the marks per day late. The marking rubric is in Canvas.

Students must retain electronic copies of their submitted assignment files and databases, as the unit coordinator may request to inspect these files before marking of an assignment is completed. If these assignment files are not made available to the unit coordinator when requested, the marking of this assignment may not proceed.

All the best!

Group member participation

This is a group assignment. The mark awarded for your assignment is conditional on you being able to explain any of your answers to your tutor or the lecturers if asked.

If your group is experiencing difficulties with the content, you should ask on Ed (use a private post if you need to discuss code or report writing directly).

Level of contribution	Proportion of final grade received
No participation	0%
Did individual task but not group task (or vice-versa)	50%
Major contributor to the group's submission.	100%