University of Toronto
CSC343, Winter 2024

Assignment 2 Warmup

Due: Wednesday 12th February by 4:00 p.m.

Learning Goals

This assignment aims to help you learn to:

- run SQL code on a remote server
- work with provided SQL starter code files
- write queries in SQL

Please read this assignment thoroughly before you proceed. Failure to follow instructions can affect your grade.

We will be testing your code in the **CS Teaching Labs environment** using PostgreSQL. It is your responsibility to make sure your code runs in this environment before the deadline! **Code which works on your machine but not on the CS Teaching Labs will not receive credit**.

This assignment is to be done *individually*.

The Domain

In this assignment, we will work with a database to support airline travel, using a schema similar (but not exactly identical) to the one you used in Assignment 1. Keep in mind that your code for this assignment must work on *any* database instance (including ones with empty tables) that satisfies the schema.

All values of type time, timestamp etc. in the dataset provided are on a 24-hour clock.

Begin by getting familiar with the schema that we have provided in `a2warmup_airtravel_schema.ddl`.

Ask yourself questions like the ones laid out for you in Assignment 1. You will work with a very similar schema on Assignment 2.

What to do

General requirements

Since your code must run correctly on the Teaching Labs (dbsrv1, specifically), you will need to test your code there before you submit. We also suggest you to work on the Teaching Labs, rather than installing PostgreSQL locally. To do that, you will need to have a process for working on a remote server. If you are not familiar with tools like `ssh` or `scp`, and other Unix-based command line tools, we recommend giving yourself some extra time to figure those out. Being able to use a command line text editor (e.g. vim, emacs, nano) will also be helpful. If you don't know how to use one, we recommend choosing the one that the people around you use, so you can ask them questions and observe what they do.

This is an **individual** assignment, which means you should be able to work on a remote server yourself, and write and debug your own SQL queries. If you need help with using the command line and working on a remote server, it is OK to ask your classmates or friends for help with that, as long as you can do it yourself by the time you submit the assignment. If you need help with the queries, please ask the CSC343 course staff. There will be additional office hours added in advance of the assignment deadline. We expect that the queries are your own individual work, and no one else has helped you write or debug them.

The schema and data

Included in the starter files is a file called `a2warmup_airtravel_schema.ddl`, which defines the schema for this assignment. We have also provided files in the `data` folder to populate the tables with data.

In order to be able to work on your queries, you will need to copy these files to dbsrv1 and use `\i` to import first the schema DDL file and then the data (`populate_data.sql`) into psql. You do NOT need to reload the data every time you open PostgreSQL; just remember to set your search path. The data is still there.

The queries

To ensure that your query results match the form expected by the auto-tester (attribute types and order, for instance), we are providing a schema for the result of each query. These can be found in files `wu1.sql`, `wu2.sql`, and `wu3.sql`. You must add your solution code for each query to the corresponding file. Make sure that each file is entirely self-contained, and not dependent on any other files; each will be run separately on a fresh database instance, and so (for example) any views you create in `wu1.sql` will not be accessible in `wu3.sql`.

Write SQL queries for each of the following:

1. Find all pairs of passengers such that they have the same last name and booked a seat in the same flight within 1 hour of each other. Report their firstnames and lastname. Duplicates are fine for this question as some passengers might have done it multiple times.
2. Find the total number of operational airports per country. We define an airport to be operational if atleast one flight departs from or arrives at that airport. Only give countries which have atleast two airports. Report the country name and the count.
3. For each airline, get a count of the number of planes that are small, medium, and large. We'll define a *large* plane as one that has all three seat classes and atleast 45 rows of seating, and a *medium* plane as one that has atleast two seat classes and a minimum of 30 rows of seating. All other planes are considered *small*.

   As part of this query, you will need to take a look at the PostgreSQL documentation to learn about the command `CAST`. We have provided a type for you to use to represent the size of a plane.

Submission instructions

For this assignment, you will hand in your three .sql files to MarkUs. Check that you have submitted the correct version of your file by downloading it from MarkUs. New files will not be accepted after the due date. Make sure that the files you submit run. This assignment will be autotested, and there will be no remarks.