University of Toronto
CSC343, Fall 2024

# Assignment 2 Warmup

*Due: Wednesday, October 9, before 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 for a veterinary clinic, with a similar but not identical schema to the one you saw 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 `vet_warmup_schema.ddl`. 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 do your 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 `vet_warmup_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 (`import_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 the procedure(s) that have been done by the most staff. Report the procedure ID(s), and the number of unique staff who have done that procedure.

2. A "high-needs" patient is one that has had more than one and a half times the average number of appointments in the last calendar year (i.e., 2023). (Yes, the patient is included in the average.) Find the IDs of all high-needs patients and the number of appointments they have scheduled in 2024.

3. A patient is considered "young" if they are under 3 years old, "senior" if they are over 8 years old, and "adult" otherwise. Their age should be calculated as of the current date.

   For each month of 2024, report the number of patients in each age category who had an appointment where they saw a vet (DVM). Include only those patients who saw a vet in 2024 in the results.

   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 age category of a patient. You will also have to look at the documentation to learn how to work with dates in PostgreSQL. In particular, we expect `EXTRACT` will be helpful to you.

## 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. We will provide a checker for you to run that will confirm we can run your code, but it does not do thorough testing. You are responsible for testing your own code.