

# Documentation Overview

---

This document describes the purpose of each file in our submission.

## Primary Work

- **ER Diagram.pdf**: The revised ER diagram for the final version of our database.
- **schema.pdf**: The revised relational schema for the final version of our database.
  - **schema.sql**: Same as **schema.pdf** but as a plaintext file. This can be run from an **SQL\*PLUS** terminal using the command `@schema.sql` to generate the tables for the database.
- **COVID-19 Database Overview PPT.pdf**: A PDF of the PowerPoints we use in our presentation. This also serves as an overview of our database and its design.
- **script.txt**: A script file of inserts, updates, deletes, and queries run on our database. The commands are run from the below two files.
  - **queries.sql**: An SQL file that can be used to run each of our queries in an **SQL\*PLUS** terminal using the command `@queries.sql`. It also contains additional insertion statements and updates to tables.
  - **dropTables.sql**: An SQL file to drop all tables used in our database (erases the database).

## Bonus Work

- **plot.sh**: A bash script that generates graphical plots of data from our database as JPEG images. The data being plotted can be specified by the user.
  - To run the script, first create and populate our database in **SQL\*PLUS** on the school server. Then, download the script into a new directory and run `bash plot.sh`. JPEG files will be generated in the current directory for each plot.
  - **NOTE 1**: This script has only been tested on the school server and relies on software that may not be present by default on other systems.
  - **NOTE 2**: This script relies on the default table / column names from our schema. It will not work if these names are changed (i.e. after running **queries.sql**).
- **Database.xlsx**: The raw data we used to construct our database. It was compiled from various datasets found on [https://data.ontario.ca/dataset?keywords\\_en=COVID-19](https://data.ontario.ca/dataset?keywords_en=COVID-19).
- **ExcelToSql.py**: A custom Python script we made that reads our **Database.xlsx** file and outputs **inserts.sql** to automatically populate our database.
  - **NOTE**: This script has **not** been tested on the school servers (it was used locally). The dependencies required in order to run the script are specified at the top of the file. They can be installed using `pip3 install pandas` and `pip3 install openpyxl` in a terminal.
- **inserts.sql**: An SQL file generated by **ExcelToSql.py** for populating our database. It contains an **INSERT** statement for each row of data from **Database.xlsx** into the corresponding table.
  - It can be run in an **SQL\*PLUS** terminal using the command `@inserts.sql` (assuming tables have already been created with **tables.sql** but the database has not been populated). It will take several minutes to run.

## Complex Queries

A description of each query in `queries.sql` is given in comments above each query. The queries are in no particular order. We consider queries 1-7 all to be non-trivial, however the following three may stand out as the most complex (in no particular order):

- **QUERY 3:** Involves user input, joins, grouping, and ordering to output the number of individuals of a given age group who have received their first and second dose of the vaccine. This gives a good overview of the vaccination data for each province.
- **QUERY 5:** Displays the percent change in COVID cases between months. As noted by the comment above the query, it is interesting to note that there was a staggering 500% increase in cases between the months of November and December, 2021. This corresponds with the first reported case of the Omicron variant in Canada on November 28th.
- **QUERY 7:** Involves user input to identify the smallest value each month in a column specified by the user. This can be used to identify outlier values in a specified column.