Homework Assignment #0
Database Systems – Fall 2023/2024

This assignment consists of two parts:

part A - This part is technical, and you need to follow the instructions in order to set up the environment on which you'll run the database.

Part B - Contains five easy questions that you are required to answer and submit on Moodle.

Note: This assignment will **not** be graded, but is still mandatory and you have to submit it.

Part A - Installation

In this part, you will learn how to set up the MySQL environment in a docker container, and work with it.

Installing docker

1. Install Docker Desktop from here https://www.docker.com/products/docker-desktop/ .
2. Open the Docker Desktop program to start the docker engine.

Running MySQL

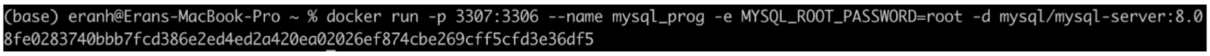
3. Run the following command, to create a mysql image and container:

   docker run -p 3307:3306 --name mysql_prog -e MYSQL_ROOT_PASSWORD=root -d mysql/mysql-server:8.0

   And your container should be up and running.
   To make sure, run the command

   docker ps

   this command shows the running containers. You should see this result:



```
(base) eranh@Erans-MacBook-Pro ~ % docker run -p 3307:3306 --name mysql_prog -e MYSQL_ROOT_PASSWORD=root -d mysql/mysql-server:8.0
8fe0283740bbb7fcd386e2ed4ed2a420ea02026ef874cbe269cff5cfd3e36df5
```

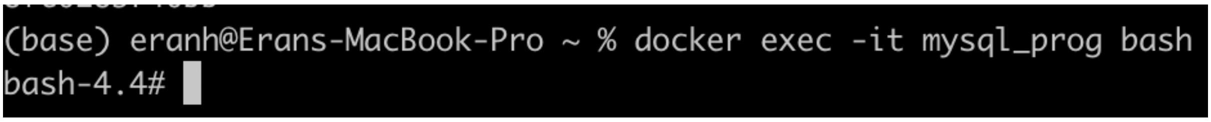
Connecting to MySQL CLI

Some commands need to be run on the MySQL CLI.

4. To connect to the CLI, run the following command to access the container's terminal:

docker exec -it mysql_prog bash

Now, You should have access to the terminal, which looks like this:

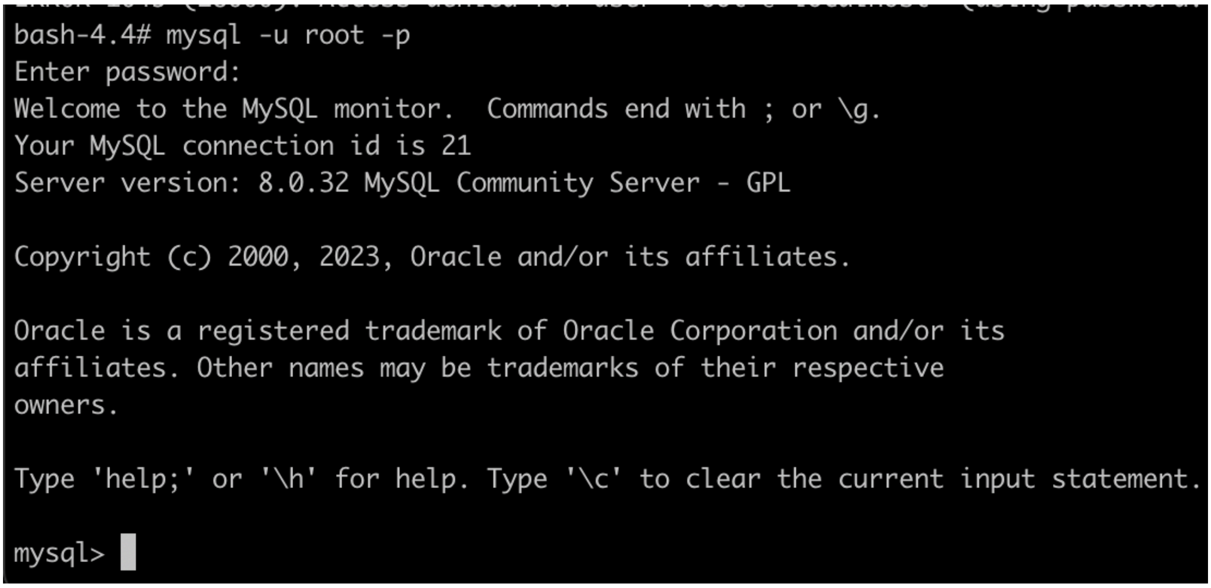


```
(base) eranh@Erans-MacBook-Pro ~ % docker exec -it mysql_prog bash
bash-4.4# 
```

5. Open the MySql CLI using the following command:

   mysql --local-infile=1 -uroot -p

   And enter the password. The password is root. This should look like this:



```
bash-4.4# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 21
Server version: 8.0.32 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> 
```

Configurations

You should change the settings to allow connections from your machine and to allow importing local files.

6. Open the MySQL CLI, as described in steps 4-5.
7. Change the root user access setting and save, by running these 2 commands:

   update mysql.user set host='%' where user='root';

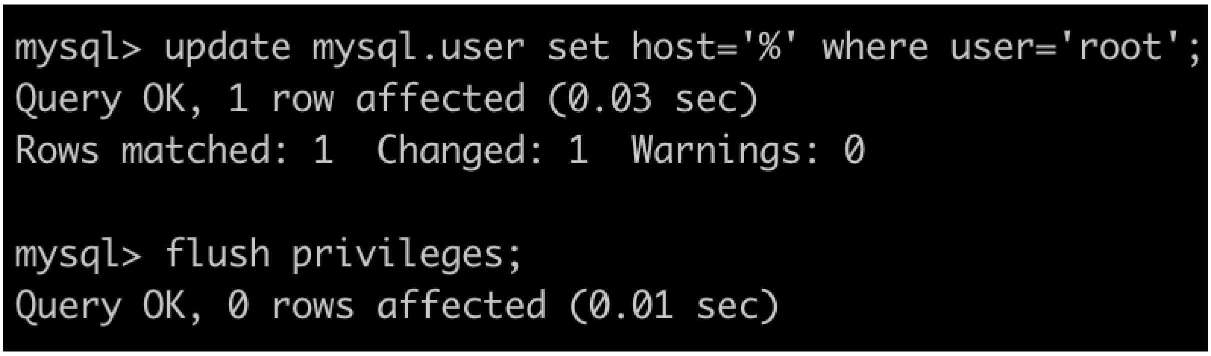
   flush privileges;

8. Allow importing local files:

   SET GLOBAL local_infile=1;

9. Exit the CLI by running the following exit command:

   ```
   exit
   ```



```
mysql> update mysql.user set host='%' where user='root';
Query OK, 1 row affected (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> flush privileges;
Query OK, 0 rows affected (0.01 sec)
```

Adding the database files

In this exercise, we are working with the covid database. The files are in the moodle. Now we will import the database csv files into our MySQL database.

10. Create a folder called “Ass0_DB”, and download the **.csv** files there.
11. From that folder, open CMD, PowerShell or Terminal.
12. Copy the csv files into the container with the following commands:

    ```
    docker cp covid_deaths.csv mysql_prog:covid_deaths.csv

    docker cp covid_vaccination.csv mysql_prog:covid_vaccination.csv
    ```

13. Open the mysql CLI inside the docker container like in steps #4 and #5.

14. Run the following commands:

    creating the database:

    ```
    create database covid_db;
    ```

    creating the deaths table:

    ```
    CREATE TABLE covid_db.covid_deaths (
    iso_code TEXT,
    continent TEXT,
    location TEXT,
    date TEXT,
    population BIGINT,
    total_cases BIGINT,
    new_cases BIGINT,
    new_cases_smoothed FLOAT,
    total_deaths BIGINT,
    new_deaths BIGINT,
    new_deaths_smoothed FLOAT,
    ```

```
total_cases_per_million BIGINT,
new_cases_per_million BIGINT,
new_cases_smoothed_per_million BIGINT,
total_deaths_per_million BIGINT,
new_deaths_per_million BIGINT,
new_deaths_smoothed_per_million FLOAT,
reproduction_rate BIGINT,
icu_patients BIGINT,
icu_patients_per_million BIGINT,
hosp_patients BIGINT,
hosp_patients_per_million BIGINT,
weekly_icu_admissions BIGINT,
weekly_icu_admissions_per_million BIGINT,
weekly_hosp_admissions BIGINT,
weekly_hosp_admissions_per_million BIGINT
);
```

creating the vaccinations table:

```
CREATE TABLE covid_db.covid_vaccination (
iso_code TEXT,
continent TEXT,
location TEXT,
date TEXT,
new_tests BIGINT,
total_tests BIGINT,
total_tests_per_thousand BIGINT,
new_tests_per_thousand BIGINT,
new_tests_smoothed BIGINT,
new_tests_smoothed_per_thousand FLOAT,
positive_rate FLOAT,
tests_per_case FLOAT,
tests_units BIGINT,
total_vaccinations BIGINT,
people_vaccinated BIGINT,
people_fully_vaccinated BIGINT,
new_vaccinations BIGINT,
new_vaccinations_smoothed BIGINT,
total_vaccinations_per_hundred FLOAT,
people_vaccinated_per_hundred FLOAT,
people_fully_vaccinated_per_hundred FLOAT,
new_vaccinations_smoothed_per_million BIGINT,
stringency_index FLOAT,
population_density FLOAT,
median_age FLOAT,
aged_65_older FLOAT,
aged_70_older FLOAT,
gdp_per_capita FLOAT,
extreme_poverty FLOAT,
cardiovasc_death_rate FLOAT,
diabetes_prevalence FLOAT,
female_smokers FLOAT,
male_smokers FLOAT,
handwashing_facilities FLOAT,
hospital_beds_per_thousand FLOAT,
life_expectancy FLOAT,
human_development_index FLOAT,
excess_mortality FLOAT
);
```

importing the data:
deaths:

```
LOAD DATA LOCAL INFILE 'covid_deaths.csv' INTO TABLE covid_db.covid_deaths CHARACTER SET UTF8 FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\r\n' IGNORE 1 LINES;
```

vaccinations:

```
LOAD DATA LOCAL INFILE 'covid_vaccination.csv' INTO TABLE covid_db.covid_vaccination CHARACTER SET UTF8 FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\r\n' IGNORE 1 LINES;
```

Connecting with python

Now we are ready to connect with python from our local machine.

15. Install the mysql connector library:

```
pip install mysql-connector-python
```

16. In the “Ass0_DB” folder that we created earlier, create a new python file: ass0.py, with the following content:

    ```
    import mysql.connector

    mydb = mysql.connector.connect(
      host="localhost",
      user="root",
      password="root",
      database="mysql",
      port='3307'
    )

    cursor = mydb.cursor()
    cursor.execute("Show Databases;")
    res = cursor.fetchall()
    print(res)
    ```

17. Run the file. The desired output is:
    [('information_schema',), ('mysql',), ('performance_schema',), ('sys',)].

Part B - SQL Queries

- Your final submission should contain a zip file containing all submission requirements and a details.txt file that should contain the ID and name of each submitting student, such as:
  first_name last_name 123456789
  first_name last_name 987654321
- The zip file should be named with your ID, such as “123456789.zip”. If multiple students are submitting, then separate the ID numbers with underscore, such as “123456789_987654321.zip”.
- In this assignment, your zip file needs to contain 5 .py files, make sure you don’t add any other unnecessary files to the zip.
- Submission is through the course website and can be either individually or in pairs – if you submit in pairs only one of the two should submit and that student will also be the one to receive feedback. A submission for three people is not permitted, if you are unable to find a pair then you should submit by yourself (it is not necessary to ask for permissions to submit by yourself).

Objective

Understanding and experiencing writing and executing SQL queries.

Data

All the questions in this part will assume the data in part A.

Requirements

For each question you are required to submit a single file called **q{n}.py** (for example q1.py, q2.py etc.) that will include the following:

- The SQL query you used.
- Each of the python files **needs to be in the following format**. Do not change anything except where it says *## PUT YOUR QUERY HERE ##.*

```
import mysql.connector

if __name__ == '__main__':
   mydb = mysql.connector.connect(
       host="localhost",
       user="root",
       password="root",
       database="covid_db",
       port='3307',
   )
   cursor = mydb.cursor()
   cursor.execute("""
      ## PUT YOUR QUERY HERE ##
   """)
   print(', '.join(str(row) for row in cursor.fetchall()))
```

Important Notes

- The format is important because the testing is automatic. If the automatic script fails because you changed the format, you will get any points for the question.
- Your query must return the answer to the question exactly; no more and no less attributes or rows.
- Do not use commands that have not yet been learned, or will not be learned at all - a query that uses such commands, **all points will be deducted**.
- Please make sure that the queries are well formatted (use tabs and newlines, parentheses etc.) to make them readable (See the example format).

Example Format

```
SELECT      film_id, title
FROM        film
WHERE       length > 10
ORDER BY    title ASC
```

Questions

1. Write a query that returns a list of all locations included in the table(location). The result should only contain the location names, without duplicates.

2. Write a query that returns the dates(date) and amount of new cases in south america, where there have been more than 150,000 new cases there(south america). The result should be ordered by new cases in ascending order..

3. Write a query that returns a list of locations(location) included in the table where in some day, there were more new deaths(new_deaths) than new cases(new_cases). The result should only include the location names, without duplicates.

4. Find the top 20 days in terms of new covid cases in Israel(date, new_cases). In the result, include the date and the amount of new cases. The result should be ordered so that days with more cases are higher.

5. Write a query to find the dates(date), locations(location) and number of new cases(new_cases), where the number of new cases was equal to the number of weekly hospital admissions(weekly_hosp_admissions). This number cannot be 0. The higher the number, the higher the result in the order.

Code

Getting started

1. Follow the instructions in the first part of the assignment.
2. Run your queries in a python file, after connecting to the database which you created in the docker container.

Code requirements

- The code must be written in Python.
- Tip: when writing long queries, you can use three quotation marks to avoid writing the long query in a single line. For example:
    - query = “””
        SELECT *
        FROM film
      “””

Running python programs

To start the program, running the following line from the command line should work:

python q1.py

External python packages

Your code should run only with the *MySQL Connector/Python* package. If you want to use any other library, you first need to get approval by writing a message in the moodle course forum. Use only the moodle course forum, so we can approve or reject it once for all the students.