# Module 3 - Database Queries Assignment Experiment

Start Assignment

- Due Sunday by 11:59pm
- Points 100
- Submitting a text entry box or a file upload

# Introduction

This assignment will introduce you to querying relational databases in SQL. In module 1, we scraped the site **grad cafe** **(https://www.thegradcafe.com/)** . Now, in module 3, we're going to use that data to learn more about the data entries submitted to grad café.

This assignment will require you to load your scraped and cleaned grad café into a PostgreSQL database, query the data to answer provided questions, create our first dynamic webpage to display our queries and results, and finally, to think about the potential limitations of web scraping free-submission, anonymous data.

**Skills**: SQL, Data Cleaning, PostgreSQL, Flask, json, HTML, CSS

THE GRADCAFE

CV
IT'S NOT EXACTLY WHAT I MEANT WHEN I SAID I NEED AN SQL NINJA...
Dataedo /cartoon
Piotr@Dataedo

# Assignment Overview

In this assignment you will:

1. Load collected data into a PostgreSQL database using psycopg
2. Carry out data analysis using SQL queries to answer questions about submission entries to grad café.
3. Display your analysis results on a dynamic flask webpage (our first!)
4. Write two paragraphs about the inherent limitations caused by using grad café (and similar self-submit sites) as data sources.

After this assignment, you will have a postgreSQL relational database containing scraped data from module 2 and python-based SQL code to query the database.

# Load data into a SQL Database

In this portion of the assignment, you will create a file called load_data.py that takes your cleaned applicant grad café entries from module 2 and loads it into a PostgreSQL database.

You can follow directions for either replit database setup (which will require a $20 replit subscription): https://docs.replit.com/hosting/databases/postgresql-on-replit

Or you can follow the directions displayed in the lecture slides on how to set up a PostgreSQL database locally in either windows or linux/unix.

Your database must include a single table of applicants. The table should include the following columns:

| Column Name | Data Type | Description |
|---|---|---|
| p_id | integer | Unique identifier |
| program | text | University and Department |
| comments | text | Comments |
| date_added | date | Date Added |
| url | text | Link to Post on Grad Café |
| status | text | Admission Status |
| term | text | Start Term |
| us_or_international | text | Student nationality |
| gpa | float | Student GPA |
| gre | float | Student GRE Quant |
| gre_v | float | Student GRE Verbal |
| gre_aw | float | Student Average Writing |
| degree | text | Student Program Degree Type |
| llm_generated_program | text | LLM Generated Department / Program |
| llm_generated_university | text | LLM Generated University |

## Programming Assignment Requirements

Once you have loaded in your PostgreSQL data, please answer the following questions:

1. How many entries do you have in your database who have applied for Fall 2026?
2. What percentage of entries are from international students (not American or Other) (to two decimal places)?
3. What is the average GPA, GRE, GRE V, GRE AW of applicants who provide these metrics?
4. What is their average GPA of American students in Fall 2026?
5. What percent of entries for Fall 2025 are Acceptances (to two decimal places)?
6. What is the average GPA of applicants who applied for Fall 2026 who are Acceptances?
7. How many entries are from applicants who applied to JHU for a masters degrees in Computer Science?

8. How many entries from 2026 are acceptances from applicants who applied to Georgetown University, MIT, Stanford University, or Carnegie Mellon University for a PhD in Computer Science?
9. Do you numbers for question 8 change if you use LLM Generated Fields (rather than your downloaded fields)?

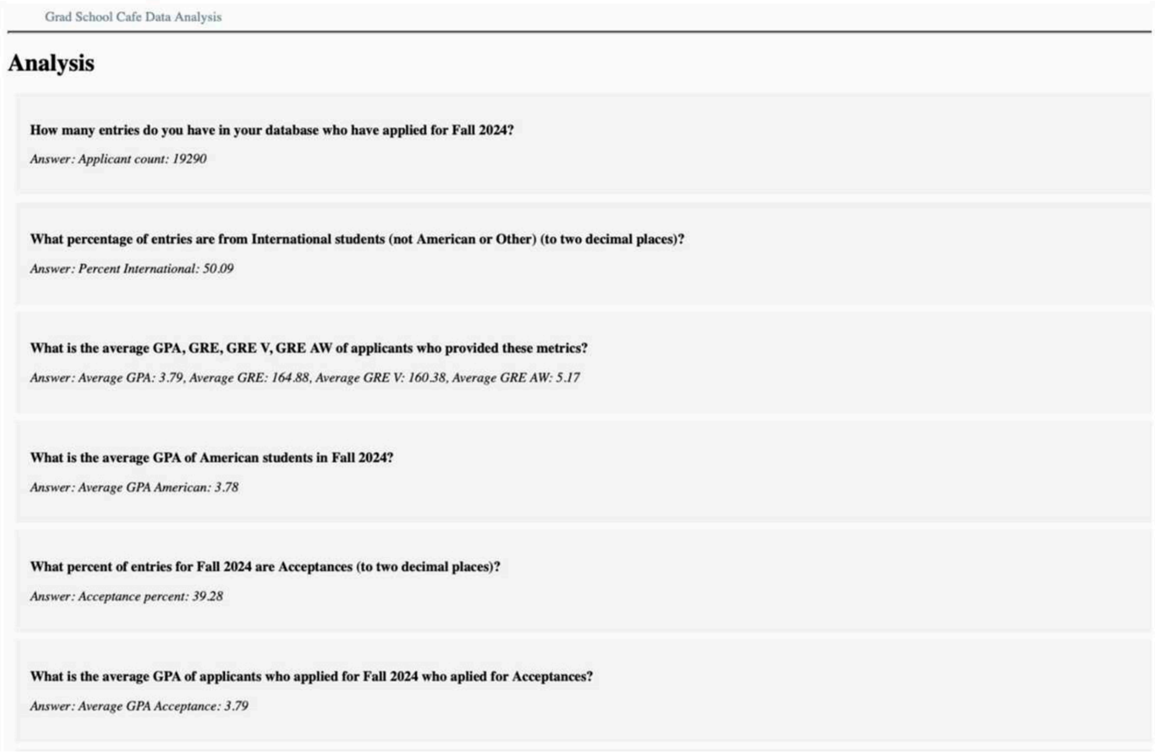
Next, come up with 2 additional questions that you are curious to answer — formulate those questions in words, and then write SQL code to answer.

Finally, provide your answers within a PDF file with a description of the query you used (and why). Please provide the code used to create the queries within a file called query_data.py.

# Flask Webpage

## Part A:

In class, we learned how to connect Flask Webpage frontends to PostgreSQL databases! Now is our chance to put that experience into practice. Please create a single, stylized (CSS) Flask page that displays the results of your PostgreSQL queries. A sample format will be shared in class, and you may use the same template structure. Your output should look like the following:



Grad School Cafe Data Analysis

**Analysis**

**How many entries do you have in your database who have applied for Fall 2024?**

*Answer: Applicant count: 19290*

**What percentage of entries are from International students (not American or Other) (to two decimal places)?**

*Answer: Percent International: 50.09*

**What is the average GPA, GRE, GRE V, GRE AW of applicants who provided these metrics?**

*Answer: Average GPA: 3.79, Average GRE: 164.88, Average GRE V: 160.38, Average GRE AW: 5.17*

**What is the average GPA of American students in Fall 2024?**

*Answer: Average GPA American: 3.78*

**What percent of entries for Fall 2024 are Acceptances (to two decimal places)?**

*Answer: Acceptance percent: 39.28*

**What is the average GPA of applicants who applied for Fall 2024 who aplied for Acceptances?**

*Answer: Average GPA Acceptance: 3.79*

Note: Your numbers (and questions) may vary from those listed above.

## Part B:

**To complete Part B, it is strongly encouraged that you copy over your module_2 code into your module_3 folder.**

1. Add a button to your Analysis webpage that upon request calls your code from module 2 to scrape any (new) data available on grad cafe and adds it to your database.
    - Name the button “Pull Data” (and include information explaining to users what the button accomplishes in a user friendly way).
2. Add a second button to your Analysis webpage (top right) that, upon request (and as long as a current request to scrape new data has not been running), updates analysis (such that the newest results are included where relevant).

- This button should be called “Update Analysis” (and include information explaining to users what the button accomplishes in a user friendly way).
- The button should do nothing if a request to pull data is currently running (which you should let your users know). Otherwise (if no request is running for more data), the button should refresh the page with the most up-to-date information.

**[I would recommend subprocesses. (https://www.geeksforgeeks.org/python/python-subprocess-module/)](https://www.geeksforgeeks.org/python/python-subprocess-module/)**

## Written Assignment Requirements

Having carried out this assignment, please write two paragraphs about the inherent limitations of carrying out analytics over anonymously submitted data items. Did the analytic responses surprise you? How does this different from standards? For example, the average GRE quantitative reasoning score was

157 for 2023-2023 and was nearly 165 for grad school entries submitted (see sample output). Why do you think that is? What might cause this to occur? Please place your essay into a file called limitations.pdf

# Deliverables

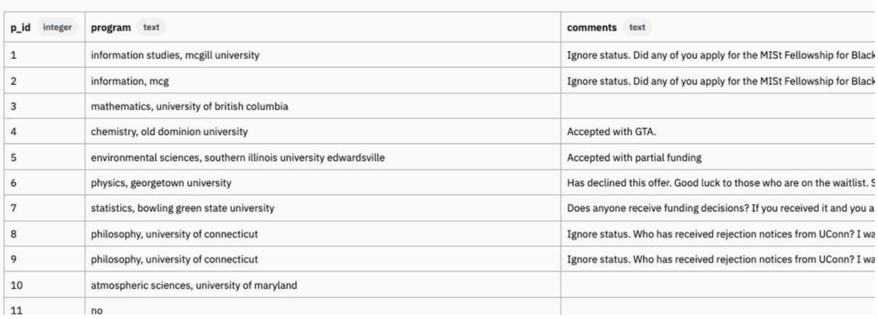
1. The SSH URL to your GitHub repository
2. load_data.py under module_3
3. query_data.py under module_3
4. All other code used to create and run your webpage + pull in new data under module_3
5. limitations.pdf under module_3
6. Screenshots of your console question output and running webpage under module_3
7. README under module_3
8. requirements.txt under module_3

Please remember to submit to both canvas and commit to your private github!

Please let us know if you have any questions via Teams or email!

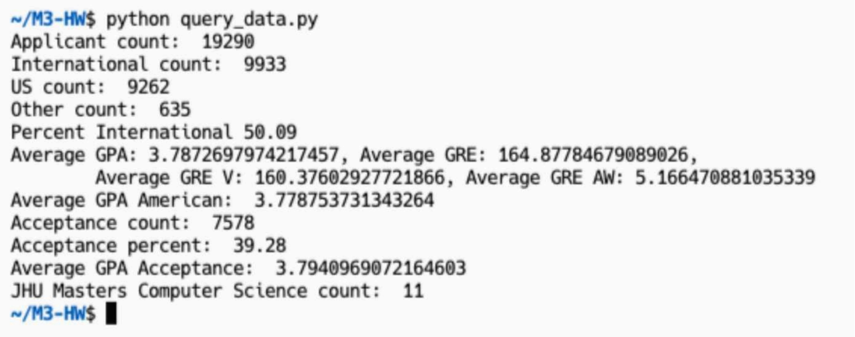
# Sample Cleaned Data Output

Start of the database:



| p_id integer | program text | comments text |
|---|---|---|
| 1 | information studies, mcgill university | Ignore status. Did any of you apply for the MISt Fellowship for Black |
| 2 | information, mcg | Ignore status. Did any of you apply for the MISt Fellowship for Black |
| 3 | mathematics, university of british columbia | |
| 4 | chemistry, old dominion university | Accepted with GTA. |
| 5 | environmental sciences, southern illinois university edwardsville | Accepted with partial funding |
| 6 | physics, georgetown university | Has declined this offer. Good luck to those who are on the waitlist. S |
| 7 | statistics, bowling green state university | Does anyone receive funding decisions? If you received it and you a |
| 8 | philosophy, university of connecticut | Ignore status. Who has received rejection notices from UConn? I wa |
| 9 | philosophy, university of connecticut | Ignore status. Who has received rejection notices from UConn? I wa |
| 10 | atmospheric sciences, university of maryland | |
| 11 | no | |

Question answers (yours will look different, because you will construct your database at a different / later date / your questions are slightly different):



```
~/M3-HW$ python query_data.py
Applicant count:  19290
International count:  9933
US count:  9262
Other count:  635
Percent International 50.09
Average GPA: 3.7872697974217457, Average GRE: 164.87784679089026,
        Average GRE V: 160.37602927721866, Average GRE AW: 5.166470881035339
Average GPA American:  3.778753731343264
Acceptance count:  7578
Acceptance percent:  39.28
Average GPA Acceptance:  3.7940969072164603
JHU Masters Computer Science count:  11
~/M3-HW$ 
```