Sara Nasab
Snasab1@jh.edu
Module 3: Description of SQL Queries Used

Before describing the reasoning for each of my SQL queries (for the questions assigned in the homework). It is useful to understand the pattern of each of the query functions used to obtain the answers for each question. Each function involves two main components: (1) a parameterized SQL query and the (2) helper function `execute_and_fetch` in order to connect to the PostgreSQL database and then execute the prior defined SQL query with the remaining parameters needed to complete that query.

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

   This is a "count" type of query, so the query executed is simply done with one f-string that uses `SELECT COUNT` and filters by the placeholder `term = %s`, in which we use to filter the counts by `term = "Fall 2024"`.

2. What percentage of entries are from international students (to two decimal places)?

   The function uses two `SELECT COUNT(*)` queries. The first counts all entries in the applicants table to get the total number of applicants. The second counts entries where the `us_or_international` column is "International". Once again, the query utilizes `%s` in order to input specific parameters that filters from which students we are counting. In the case of counting the number of international students, we use `us_or_international = "International"`. The rest of the function (not a SQL query), calculates the percentage by dividing the international count by the total count and multiplying by 100.

3. What is the average GPA, GRE, GRE V, GRE AW of applicants who provide these metrics?

   For each score, we can directly retrieve the average using the SQL query. To ensure that we do not take that average across students who did not submit a (e.g., GPA, GRE) score, the SQL query uses the condition `WHERE [score] IS NOT NULL`. Each average score (GPA, GRE, GRE V, GRE AW) requires its own SQL query.

4. What is their average GPA of American students in Fall 2024?

   The SQL query retrieves the appropriate average GPA score directly and at the same time, filters the set of applicants (term, applicant type) by using a `WHERE` condition. Additionally, in order to not take the GPA score across only applicants whose GPA is provided, we add the condition `WHERE gpa IS NOT NULL`.

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

   The function consists of two queries. The first query counts the total applicants for "Fall 2024" using `SELECT COUNT(*)`. Then, the second query counts applicants for "Fall 2024" whose `status` column matches 'Accepted' (using `LIKE %s` with `'%Accepted%'`). Note that because the `status` values include if the applicant was Accepted/Rejected and the date in which that decision was determined, we use the wildcard `'%Accepted%'` (to count the number of acceptances. The rest of the function (not a SQL query) computes the

percentage by dividing the count of accepted applicants by the total applicants and multiplying by 100.

6. What is the average GPA of applicants who applied for Fall 2024 who are Acceptances?

   This is a very similar query to the previous question (but in some ways much simpler). This question only requires one query which is primarily used to retrieve the average GPA score among Fall 2024/Accepted applicants who submitted their GPA scores. Note that the query filters by term, status, and ensure that the GPA is provided. Since the value of the acceptance status is part of a larger string (which includes if the applicant was Accepted/Rejected and the date in which that decision was determined), we use the wildcard `'%Accepted%'`.

7. How many entries are from applicants who applied to JHU for a masters degrees in Computer Science?

   We use `SELECT COUNT(*)` to these entries and use the `WHERE` condition to filter by program (Johns Hopkins University Computer Science) and degree (Masters). Since the program value is submitted manually by the user (i.e., the applicant types out the university and program; this is not a drop-down menu), I used a wildcard `"%Johns%Hopkins%Comp%Sci%"` to allow for any typos. Similar to the degree type, I used the wildcard `%Mas%` to filter. These two wildcards together should successfully filter out the applicants of interest.