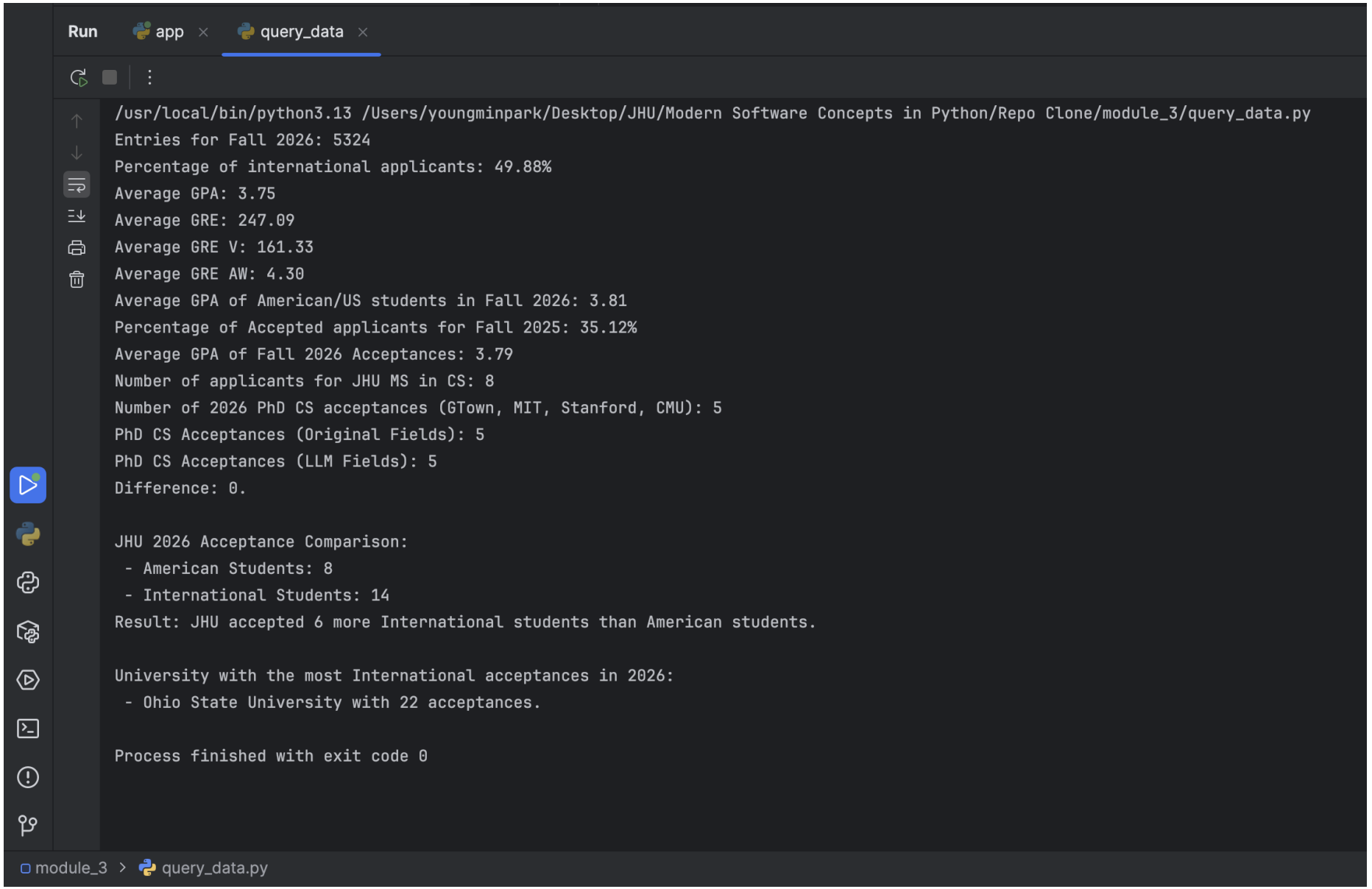
Screenshot query_data.py console output from initial 30,000 entry



Description and Logic of Queries

1. How many entries do you have in your database who have applied for Fall 2026?
    a. I used an OR condition to check both the "term" and "status" columns for the string "2026." There are instances in the JSON file where the application term was mentioned in the "status" field. To account for various ways in which "2026" can appear (i.e. Accepted 2026, Applied 2026, Rejected 2026, and various capitalizations of these phrases), I utilized a combination of the ILIKE (case insensitive pattern match keyword) and % (wildcard). When used together, it will search the string "2026" regardless of what comes before or after it thereby accounting for all variations of "2026."

2. What percentage of entries are from international students (not American or Other) (to two decimal places)?
    a. The COUNT(*) and FILTER keywords were used to isolate international student entries and count them. ::DECIMAL was used to convert that count integer into a decimal to enable floating-point division, which will avoid a small decimal resulting in zero after the division. NULLIF keyword to avoid the return of a zero, which would throw an error due to dividing by zero. Lastly, the ROUND keyword was used to keep the resulting percentage to two decimal places.

3. What is the average GPA, GRE, GRE V, GRE AW of applicants who provide these metrics?
    a. The AVG() function was used to calculate the average GPA/GRE scores because it automatically excludes any entries with NULL and only uses actual scores that were provided. This is effective because there are many instances where scores were not provided and prevents the skewing of the calculation.

4. What is their average GPA of American students in Fall 2026?
    a. After selecting the average GPA, different ways of annotating American citizenship were accounted for by utilizing the combination of the ILIKE keyword and the % wildcard keyword; this accounted for entries that said "US" or "American." Then the AND keyword was used to combine American student entries with any variation of the string "Fall 2026 by again utilizing the ILIKE / % keyword combination.
5. What percent of entries for Fall 2025 are Acceptances (to two decimal places)?

    a. The COUNT(*) and FILTER keywords were used to count variations of the strings "Fall 2025" and "Accepted". The ILIKE/% combination was used for both to account for variations on how these strings appear. ::DECIMAL was used to convert the integer into a float to facilitate decimal division to avoid small decimals resulting in a zero and NULLIF was used to prevent dividing by zero. Lastly ROUND was used to limit to two decimal places.

6. What is the average GPA of applicants who applied for Fall 2026 who are Acceptances?
    a. After selecting AVG(GPA), both the "term" and "status" columns were searched for any variation of the string "Fall 2026" and "Accepted" respectively. The ILIKE/% combination was used for both the term and status column to account for variations of how the strings appeared in a sentence of how they were capitalized.

Description and Logic of Queries

7. How many entries are from applicants who applied to JHU for a masters degrees in Computer Science?
    a. The ILIKE/% combination was used again for variations in how “Johns Hopkins University” appeared (ILIKE accounts for case insensitivity and % accounts for misspellings and spacing variations). Also, the ILIKE/% was used in the case the acronym JHU was used. The OR keyword was used to account for both ways to entering Johns Hopkins University. The AND keyword was used to add the criteria of a Masters of Science degree and Computer Science. The ILIKE/% combination was used for both to account for variations.

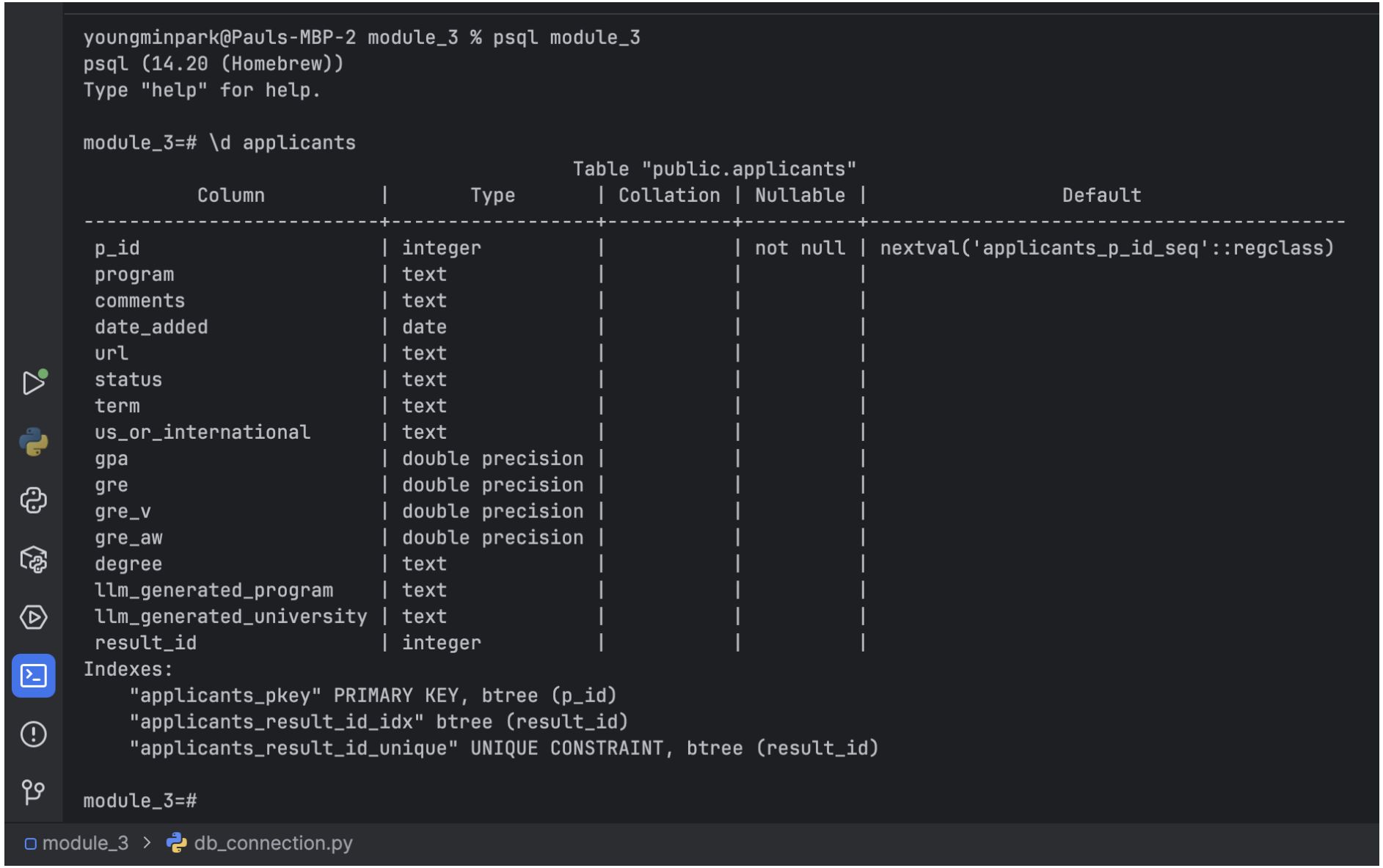
8. How many entries from 2026 are acceptances from applicants who applied to Georgetown University, MIT, Stanford University, or Carnegie Mellon University for a PhD in Computer Science?
    a. The AND keyword was used to search the criteria of being accepted in 2026 and the OR keyword was used several times to search through all the listed colleges. The ILIKE/% combination was used for all fields to account for variations and typos.

9. Do your numbers for question 8 change if you use LLM Generated Fields (rather than your downloaded fields)?
    a. The comparison logic here is to count using the “program field,” which is from the original scraped data before running it through the LLM. Then count using the LLM generated data. The two results were subtracted to see if there was a difference (zero in this case). As in the previous query a combination of AND and OR keywords to used to count all instances of the prescribed criteria and the ILIKE/% combination was used to account for variations and typos.

10. How many American students versus International Students were accepted to JHU in 2026?
    a. The COUNT(*) and FILTER keyword combination, setting WHERE to Johns Hopkins University, and searching the “status” column for any variation of “Accepted 2026” were used to get the total count of American Students and International Students that were accepted to Johns Hopkins University in 2026. The ILIKE/% combination was used to account for variations and typos for all these fields. Then a comparison was done by seeing which value (American and International) was larger to determine which population had a higher acceptance.

11. Which university accepted the most International Students in 2026?
    a. After setting the criteria to search for international students being accepted in 2026, the GROUP BY phrase was used to consolidate the number of international students by university. Then a combination of ORDER BY and DESC were used to determine the university that accepted the most international students.

Screenshot of PostgreSQL Applicant Database

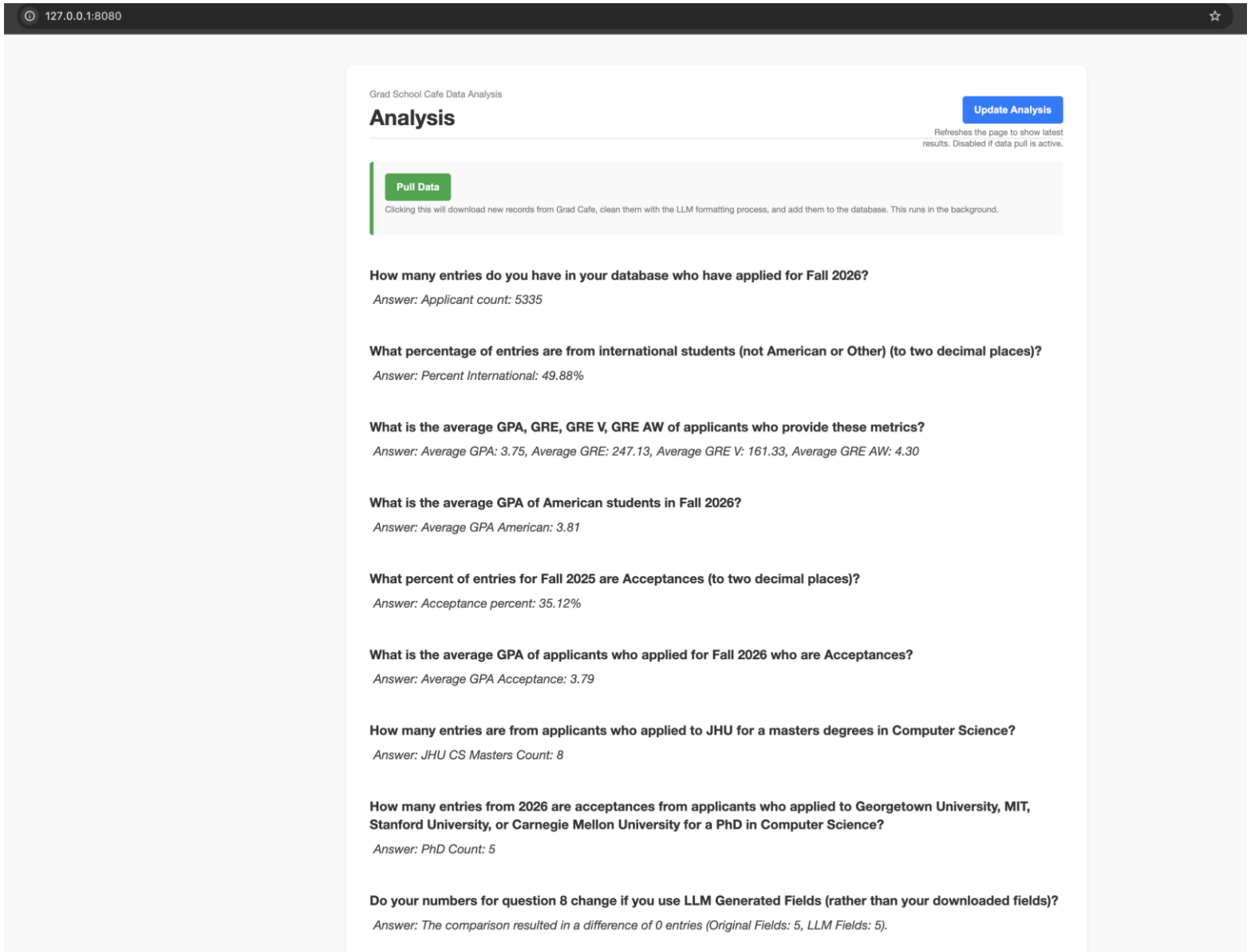


```
youngminpark@Pauls-MBP-2 module_3 % psql module_3
psql (14.20 (Homebrew))
Type "help" for help.

module_3=# \d applicants
                                    Table "public.applicants"
          Column          |       Type       | Collation | Nullable |                  Default
--------------------------+------------------+-----------+----------+-----------------------------------------
 p_id                     | integer          |           | not null | nextval('applicants_p_id_seq'::regclass)
 program                  | text             |           |          |
 comments                 | text             |           |          |
 date_added               | date             |           |          |
 url                      | text             |           |          |
 status                   | text             |           |          |
 term                     | text             |           |          |
 us_or_international      | text             |           |          |
 gpa                      | double precision |           |          |
 gre                      | double precision |           |          |
 gre_v                    | double precision |           |          |
 gre_aw                   | double precision |           |          |
 degree                   | text             |           |          |
 llm_generated_program    | text             |           |          |
 llm_generated_university | text             |           |          |
 result_id                | integer          |           |          |
Indexes:
    "applicants_pkey" PRIMARY KEY, btree (p_id)
    "applicants_result_id_idx" btree (result_id)
    "applicants_result_id_unique" UNIQUE CONSTRAINT, btree (result_id)

module_3=#
```

Screenshot of PostgreSQL Applicant Database (PG1)



Screenshot of PostgreSQL Applicant Database (PG2)