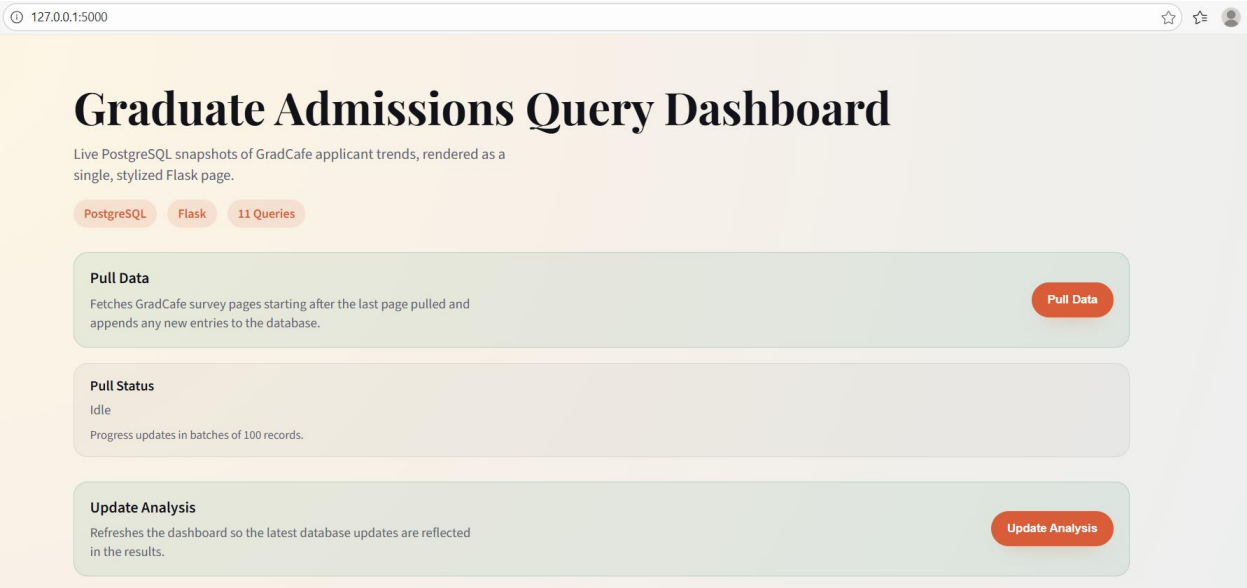
Module 3 - Database Queries Assignment Experiment

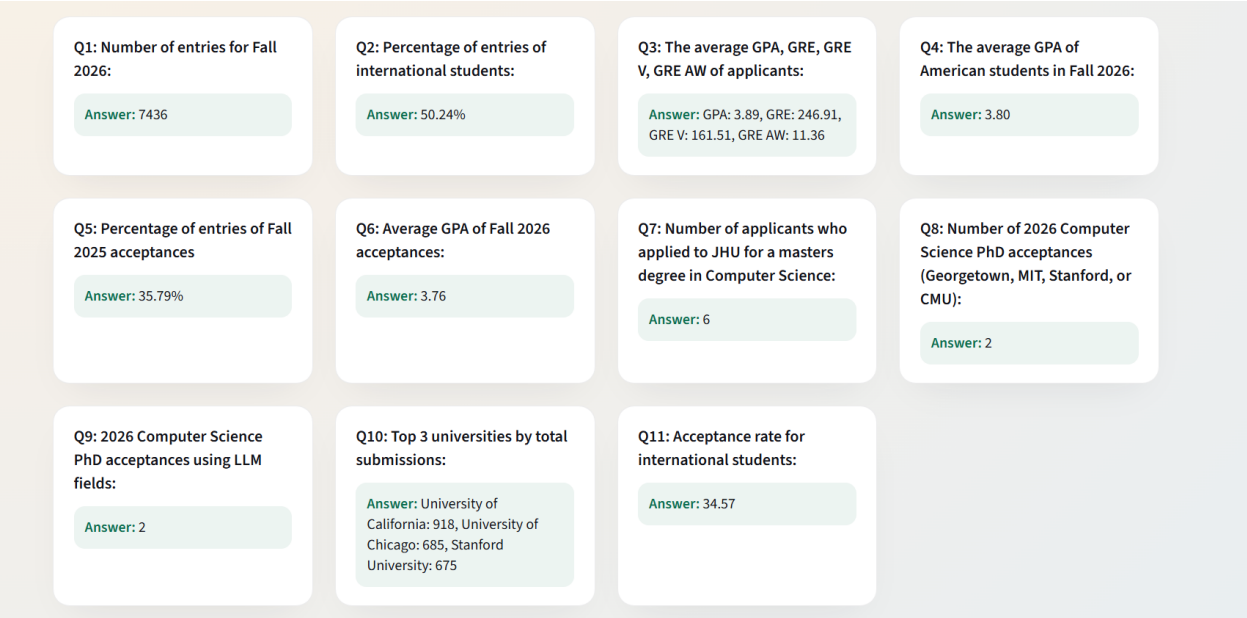
Console Output:

```
Q1: Number of entries for Fall 2026:
Answer: 7428

Q2: Percentage of entries of international students:
Answer: 50.24%

Q3: The average GPA, GRE, GRE V, GRE AW of applicants:
Answer: GPA: 3.89, GRE: 246.91, GRE V: 161.51, GRE AW: 11.36

Q4: The average GPA of American students in Fall 2026:
Answer: 3.80

Q5: Percentage of entries of Fall 2025 acceptances
Answer: 35.79%
```

```
Q6: Average GPA of Fall 2026 acceptances:
Answer: 3.76

Q7: Number of applicants who applied to JHU for a masters degree in Computer Science:
Answer: 6

Q8: Number of 2026 Computer Science PhD acceptances (Georgetown, MIT, Stanford, or CMU):
Answer: 2

Q9: 2026 Computer Science PhD acceptances using LLM fields:
Answer: 2

Q10: Top 3 universities by total submissions:
Answer: University of California: 918, University of Chicago: 685, Stanford University: 675

Q11: Acceptance rate for international students:
Answer: 34.56
```

Webpage:


127.0.0.1:5000
Graduate Admissions Query Dashboard
Live PostgreSQL snapshots of GradCafe applicant trends, rendered as a single, stylized Flask page.
PostgreSQL
Flask
11 Queries
Pull Data
Fetches GradCafe survey pages starting after the last page pulled and appends any new entries to the database.
Pull Data
Pull Status
Idle
Progress updates in batches of 100 records.
Update Analysis
Refreshes the dashboard so the latest database updates are reflected in the results.
Update Analysis


Q1: Number of entries for Fall 2026:
Answer: 7436
Q2: Percentage of entries of international students:
Answer: 50.24%
Q3: The average GPA, GRE, GRE V, GRE AW of applicants:
Answer: GPA: 3.89, GRE: 246.91, GRE V: 161.51, GRE AW: 11.36
Q4: The average GPA of American students in Fall 2026:
Answer: 3.80
Q5: Percentage of entries of Fall 2025 acceptances
Answer: 35.79%
Q6: Average GPA of Fall 2026 acceptances:
Answer: 3.76
Q7: Number of applicants who applied to JHU for a masters degree in Computer Science:
Answer: 6
Q8: Number of 2026 Computer Science PhD acceptances (Georgetown, MIT, Stanford, or CMU):
Answer: 2
Q9: 2026 Computer Science PhD acceptances using LLM fields:
Answer: 2
Q10: Top 3 universities by total submissions:
Answer: University of California: 918, University of Chicago: 685, Stanford University: 675
Q11: Acceptance rate for international students:
Answer: 34.57

Queries:

1. For question 1 “How many entries do you have in your database who have applied for Fall 2026?“, the query used is:

```
SELECT COUNT(*)
FROM applicants
WHERE term = 'Fall 2026';
```

The WHERE term = 'Fall 2026' filters to that term, and COUNT(*) returns the total number of matching rows.

2. For question 2 “What percentage of entries are from international students (not American or Other) (to two decimal places)?”, the query used is:

```
SELECT ROUND(
  100.0 * SUM(CASE WHEN LOWER(us_or_international) NOT IN ('american', 'other') THEN 1 ELSE 0 END)
  / NULLIF(COUNT(*), 0),
  2
) AS percent_international
FROM applicants
WHERE us_or_international IS NOT NULL AND us_or_international <> '';
```

WHERE filters rows that are not American/Other to find the number of international students. The SUM keeps track of the total number of rows that meets the filtered condition. NULLIF(COUNT(*), 0) avoids division by zero. Divides by the total number of rows with a non-empty nationality to get a percentage, and ROUND(..., 2) formats the percent to two decimals.

3. For question 3 “What is the average GPA, GRE, GRE V, GRE AW of applicants who provide these metrics?”, the query used is:

```
SELECT
  ROUND(AVG(gpa)::numeric, 2) AS avg_gpa,
  ROUND(AVG(gre)::numeric, 2) AS avg_gre,
  ROUND(AVG(gre_v)::numeric, 2) AS avg_gre_v,
  ROUND(AVG(gre_aw)::numeric, 2) AS avg_gre_aw
FROM applicants
WHERE gpa IS NOT NULL OR gre IS NOT NULL OR gre_v IS NOT NULL OR gre_aw IS NOT NULL;
```

AVG(...) computes the average for each column: gpa, gre, gre_v, and gre_aw and rounds to 2 decimals. The WHERE clause excludes rows where all four metrics are NULL, so the averages are based on actual data rather than empty rows.

4. For question 4 “What is their average GPA of American students in Fall 2026?”, the query used is:

```
SELECT ROUND(AVG(gpa)::numeric, 2) AS avg_gpa_american_fall_2026
FROM applicants
WHERE term = 'Fall 2026'
AND LOWER(us_or_international) = 'american'
AND gpa IS NOT NULL
AND gpa < 5.0;
```

WHERE filters to “Fall 2026”, to American students, excludes missing/invalid GPAs with gpa IS NOT NULL and gpa < 5.0. AVG computes the average GPA and ROUND rounds it to two decimals

5. For question 5, “What percent of entries for Fall 2025 are Acceptances (to two decimal places)?”, the query used is:

```
SELECT ROUND(
  100.0 * SUM(CASE WHEN LOWER(status) = 'accepted' THEN 1 ELSE 0 END)
  / NULLIF(COUNT(*), 0),
  2
) AS percent_accept_fall_2025
FROM applicants
WHERE term = 'Fall 2025';
```

WHERE filters rows to term = 'Fall 2025'. SUM calculate the total counts of those have status = 'accepted'. Divides by the total number of Fall 2025 rows to get a percentage. Uses NULLIF(COUNT(*), 0) to avoid division by zero. ROUND rounds to 2 decimals.

6. For question 6, “What is the average GPA of applicants who applied for Fall 2026 who are Acceptances?”, the query used is:

```
SELECT ROUND(AVG(gpa)::numeric, 2) AS avg_gpa_fall_2026_accepts
FROM applicants
WHERE term = 'Fall 2026'
 AND LOWER(status) = 'accepted'
 AND gpa IS NOT NULL
 AND gpa < 5.0;
```

WHERE filters to term 'Fall 2026', to accepted applicants with LOWER(status) = 'accepted', and excludes missing/invalid GPAs with gpa IS NOT NULL and gpa < 5.0. AVG computes the average GPA, and ROUND rounds it to two decimals.

7. For question 7, “How many entries are from applicants who applied to JHU for a masters degrees in Computer Science?”, the query used is:

```
SELECT COUNT(*)
FROM applicants
WHERE program ILIKE %s
  AND program ILIKE %s
  AND degree ILIKE %s;
```

("%Johns Hopkins%", "%Computer Science%", "Master%")

Uses program ILIKE with %Johns Hopkins% and %Computer Science% to match the program text. Uses degree ILIKE 'Master%' to restrict to master’s programs. WHERE filters to these text. COUNT counts matching rows.

8. For question 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?”, the query used is:

```
SELECT COUNT(*)
FROM applicants
WHERE term LIKE %s
  AND LOWER(status) = 'accepted'
  AND degree ILIKE %s
  AND program ILIKE %s
  AND (
     program ILIKE %s
     OR program ILIKE %s
```

```
    OR program ILIKE %s
    OR program ILIKE %s
    OR program ILIKE %s
    OR program ILIKE %s
    OR program ILIKE %s
);
```

("%2026%", "PhD%", "%Computer Science%", "%George Town%", "%Georgetown%",
"%Massachusetts Institute of Technology%", "%MIT%", "%Stanford University%",
"%Carnegie Mellon University%", "%CMU%")

Uses program ILIKE with ("%2026%", "PhD%", "%Computer Science%", "%George Town%", "%Georgetown%", "%Massachusetts Institute of Technology%", "%MIT%", "%Stanford University%", "%Carnegie Mellon University%", "%CMU%") to match the program text. WHERE filters to accepted decisions with LOWER(status) = 'accepted', limits to 2026 terms with term LIKE '%2026%', filters to PhD with degree ILIKE 'PhD%', filters to Computer Science with program ILIKE '%Computer Science%', and limits the university to the specified schools (Georgetown, MIT, Stanford, CMU) using a set of OR conditions. Then COUNT(*) gives the number of matching rows.

9. For question 9, “Do you numbers for question 8 change if you use LLM Generated Fields (rather than your downloaded fields)?”, the query used is:

```
SELECT COUNT(*)
FROM applicants
WHERE term LIKE %s
  AND LOWER(status) = 'accepted'
  AND degree ILIKE %s
```

```
 AND llm_generated_program ILIKE %s
 AND (
    llm_generated_university ILIKE %s
    OR llm_generated_university ILIKE %s
    OR llm_generated_university ILIKE %s
    OR llm_generated_university ILIKE %s
    OR llm_generated_university ILIKE %s
    OR llm_generated_university ILIKE %s
 );
```

```
("%2026%", "PhD%", "%Computer Science%", "%George Town University%",
 "%Massachusetts Institute of Technology%", "%MIT%", "%Stanford University%",
 "%Carnegie Mellon University%", "%CMU%")
```

Uses program ILIKE with ("%2026%", "PhD%", "%Computer Science%", "%George Town University%", "%Massachusetts Institute of Technology%", "%MIT%", "%Stanford University%", "%Carnegie Mellon University%", "%CMU%") to match the program text. WHERE filters to 2026 terms, filters to accepted decisions, filters to PhD degrees, uses llm_generated_program for “Computer Science”, and uses llm_generated_university to match the target schools.

At first, my answer to question 8 and to question 9 were different. That was because question 8 used the raw program text, while question 9 uses the LLM-generated llm_generated_program/llm_generated_university. If the LLM normalization fixes messy or inconsistent text, question 9 may count more or fewer rows than question 8. If the LLM fields closely match the raw data, they might be the same. As I inspected the data in the database, I found that in the raw data Georgetown University is reported as “Georgetown University" but in the llm_generated_university, it is reported as "George Town University". The SQL query originally used for my question 9 search the llm_generated_university column for “Georgetown University" and returned no data.

After my inspection of the data, I change it to “George Town University", and the answer of question 9 matches the answer of question 8 at that point.

10. For question 10, “What are the top 3 universities in term of number of application submitted (using LLM generated university)”, the query used is:

```
SELECT llm_generated_university, COUNT(*) AS total_submissions
FROM applicants
WHERE llm_generated_university IS NOT NULL AND llm_generated_university <> ''
GROUP BY llm_generated_university
ORDER BY total_submissions DESC
LIMIT 3;
```

COUNT counts submissions per university. WHERE filters out empty llm_generated_university. GROUP groups submission by the university that they applied to. ORDER orders the groups by the count descending. LIMIT limits the result to the top 3 universities.

11. For question 11, “What is the acceptance rate for international students:”, the query used is:

```
SELECT ROUND(
  100.0 * SUM(CASE WHEN LOWER(status) = 'accepted' THEN 1 ELSE 0 END)
  / NULLIF(COUNT(*), 0),
  2
) AS percent_international_accepts
FROM applicants
WHERE LOWER(us_or_international) NOT IN ('american', 'other')
```

```
  AND us_or_international IS NOT NULL
  AND us_or_international <> '';
```

WHERE filters to international rows by excluding american and other, and removing blank values. COUNT counts how many of those have status = 'accepted'. SUM(CASE WHEN LOWER(status) = 'accepted' THEN 1 ELSE 0 END) keeps track of the total number of accepted international students. The sum is then divided by the total number of international rows to get a percentage. Uses NULLIF(COUNT(*), 0) to avoid divide-by-zero and rounds to two decimals.