

Query Questions

1) How many entries do you have in your database who have applied for Fall 2026?

a) 6,978

b) `SELECT COUNT(*)`

`FROM applicants`

`WHERE term = 'Fall 2026';`

I used the `COUNT(*)` query to get the number of the applicants, and I used the `WHERE` term to filter rows for when the term equals "Fall 2026".

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

a) 44.36%

b) `SELECT`

`ROUND(100.0 * SUM(CASE WHEN us_or_international = 'International' THEN 1 ELSE 0
END) / COUNT(*), 2)`

`FROM applicants;`

This query calculates the percent of international applicants by using a conditional `SUM` to only count international students and then divide by the total count of records. I used `ROUND` to limit the percentage to two decimal points.

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

a) For GPA: 3.81, and Not Available for everything else

b) `SELECT`

`COUNT(gpa),`

`COUNT(gre),`

`COUNT(gre_v),`

`COUNT(gre_aw),`

`ROUND(AVG(gpa)::numeric, 2),`

```
ROUND(AVG(gre)::numeric, 2),  
ROUND(AVG(gre_v)::numeric, 2),  
ROUND(AVG(gre_aw)::numeric, 2)  
FROM applicants;
```

The query uses AVG to get the average for each metric, however SQL ignores all NULL values, and since GRE was not frequently submitted they could not be computed.

4) What is their average GPA of American students in Fall 2026?

a) 4.07

b)

```
SELECT ROUND(AVG(gpa)::numeric, 2)  
FROM applicants  
WHERE term = 'Fall 2026'  
AND us_or_international = 'American'  
AND gpa IS NOT NULL;
```

This query filters the applicants by both “Fall of 2026” and “American”, gets the GPA (if available), and rounds it. I made sure to use AND so that the overlap of the two metrics is counted.

5) What percent of entries for Fall 2026 are Acceptances (to two decimal places)?

a) 24.32%

b)

```
SELECT  
ROUND(100.0 * SUM(CASE WHEN status ILIKE 'accept%' THEN 1 ELSE 0 END)  
/COUNT(*), 2 )  
FROM applicants  
WHERE term = 'Fall 2026';
```

This query will filter for an accepted admission status limited within Fall 2026. Using ROUND again to round to 2 decimal places.

6) What is the average GPA of applicants who applied for Fall 2026 who are Acceptances?

a) 3.80

b) `SELECT ROUND(AVG(gpa)::numeric, 2)`

`FROM applicants`

`WHERE term = 'Fall 2026'`

`AND status ILIKE 'accept%'`

`AND gpa IS NOT NULL;`

This query averages the submitted GPA from accepted Fall 2026 applicants.

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

a) 40

b) `SELECT COUNT(*)`

`FROM applicants`

`WHERE degree ILIKE 'master%'`

`AND (program ILIKE '%johns hopkins%' OR program ILIKE '%jhu%')`

`AND (program ILIKE '%computer science%' OR program ILIKE '%cs%');`

This query filters applicants with a masters and it's from JHU and it's in computer science. I asked Google Gemini how to count and reference inconsistent names, like JHU/ jhu, or Johns Hopkins/johns Hopkins, and it recommended Insensitive LIKE, which treats uppercase and lowercase letters as identical. The '%txt%' as used above, will look for the text anywhere in the strng.

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) 9

b) `SELECT COUNT(*)`

`FROM applicants`

`WHERE date_added >= '2026-01-01'`

`AND date_added < '2027-01-01'`

`AND status ILIKE 'accept%'`

AND degree ILIKE 'phd%'

AND program ILIKE '%computer science%'

AND (program ILIKE '%mit%' OR program ILIKE '%stanford%' OR program ILIKE '%carnegie mellon%' OR program ILIKE '%georgetown%');

Similar to the code for the other questions, I filtered the applicants using WHERE and ILIKE to find applicants throughout 2026, to the mentioned schools, and for a PhD in computer science.

9) Do your numbers for question 8 change if you use LLM Generated Fields (rather than your downloaded fields)?

a) Yes, using LLM Generated Fields gave 0 matching entries

b) SELECT COUNT(*)

FROM applicants

WHERE date_added >= '2026-01-01'

AND date_added < '2027-01-01'

AND status ILIKE 'accept%'

AND degree ILIKE 'phd%'

AND llm_generated_program ILIKE '%computer science%'

AND llm_generated_university IN ('Georgetown University', 'Massachusetts Institute of Technology', 'Stanford University', 'Carnegie Mellon University');

This question was handled similarly to question 8 except it only references LLM generated university and program fields. I believe this shows that AI can group things together well but it can skip over valid entries due to strict matching of text.

10) Are international applicants more likely to be accepted than American applicants for Fall 2026?

a) Yes. The international acceptance rate was 27.73% and the American acceptance rate was 22.02%.

b) SELECT

us_or_international,

ROUND(

100.0 * SUM(CASE WHEN status ILIKE 'accept%' THEN 1 ELSE 0 END)

/ COUNT(*),

2

```
) AS acceptance_rate  
FROM applicants  
WHERE term = 'Fall 2026'  
GROUP BY us_or_international  
ORDER BY us_or_international;
```

This query filters the table to only Fall 2026 applicants. I then used GROUP BY to group applicants by the us_or_international category, creating two groups, U.S. applicants and international applicants. Using GROUP BY allows me to compute an acceptance percentage for each group. I used GROUP BY after looking at a SQL cheat sheet from www.DataCamp.com. I also added ORDER BY which orders output by alphabetical order, so international group comes first then American.

11) Does GPA differ between accepted and not accepted applicants for Fall 2026?

a) Yes, the accepted average GPA for Fall 2026 is 3.80 while the not accepted average GPA for Fall 2026 is 4.07.

b) SELECT
CASE
WHEN status ILIKE 'accept%' THEN 'Accepted'
ELSE 'Not Accepted'
END AS decision_group,
ROUND(AVG(gpa)::numeric, 2) AS avg_gpa
FROM applicants
WHERE term = 'Fall 2026'
AND gpa IS NOT NULL
GROUP BY decision_group
ORDER BY decision_group;

For this query I used CASE, which follows if-else logic, to create a temporary column called decision_group where applicants are sorted between accepted and everything else. WHERE filters the applicants for Fall 2026 and ignores any missing data, and I used GROUP BY to calculate the average GPA for each group.

