

QUESTION 1: How many Fall 2025 entries are there?

SQL:

```
SELECT COUNT(*) FROM applicants WHERE term = 'Fall 2025';
```

Explanation:

COUNT(*) counts every row in the applicants table. The WHERE clause filters to only rows where the term column is exactly 'Fall 2025'. This gives us the total number of applicant records for the Fall 2025 admissions cycle.

Answer: 19,814

QUESTION 2: What percentage of applicants are international students?

SQL:

```
SELECT ROUND(
    (COUNT(*) FILTER (WHERE us_or_international NOT IN ('American', 'Other'))
    * 100.0) / COUNT(*),
    2) FROM applicants;
```

Explanation:

This uses PostgreSQL's FILTER clause to do a conditional count. The numerator counts rows where us_or_international is neither 'American' nor 'Other' -- i.e., international students. That count is multiplied by 100.0 (to force decimal division instead of integer division) and divided by the total row count. ROUND(..., 2) rounds the result to two decimal places.

Answer: 50.24%

QUESTION 3: What are the average GPA, GRE Quant, GRE Verbal, and GRE AW scores?

SQL:

```
SELECT AVG(gpa), AVG(gre), AVG(gre_v), AVG(gre_aw) FROM applicants;
```

Explanation:

AVG() computes the arithmetic mean of each column across all rows. NULL values are automatically excluded by AVG(), so only applicants who reported each score contribute to that score's average. No WHERE clause means all applicants in the table are included.

Answer:

```
Avg GPA:      3.8591
Avg GRE Q:    245.06
Avg GRE V:    161.61
Avg GRE AW:   11.83
```

QUESTION 4: What is the average GPA of American students in Fall 2025?

SQL:

```
SELECT AVG(gpa) FROM applicants
WHERE us_or_international = 'American' AND term = 'Fall 2025';
```

Explanation:

AVG(gpa) computes the mean GPA, but the WHERE clause applies two filters

simultaneously using AND: the student must be flagged as 'American' in the us_or_international column, and the term must be 'Fall 2025'. Only rows matching both conditions contribute to the average.

Answer: 3.7720

QUESTION 5: What is the acceptance rate for Fall 2025?

SQL:

```
SELECT ROUND(
    (COUNT(*) FILTER (WHERE status ILIKE '%Accepted%' AND term = 'Fall 2025')
    * 100.0) /
    NULLIF(COUNT(*) FILTER (WHERE term = 'Fall 2025'), 0),
    2) FROM applicants;
```

Explanation:

The numerator counts Fall 2025 rows whose status contains 'Accepted' (case-insensitive via ILIKE, with % wildcards on both sides to match substrings like "Accepted via Email"). The denominator counts all Fall 2025 rows. NULLIF(..., 0) prevents a division-by-zero error -- if there were no Fall 2025 entries, it would return NULL instead of crashing. The result is multiplied by 100 for a percentage and rounded to two decimal places.

Answer: 35.77%

QUESTION 6: What is the average GPA of accepted students in Fall 2025?

SQL:

```
SELECT AVG(gpa) FROM applicants
WHERE term = 'Fall 2025' AND status ILIKE '%Accepted%';
```

Explanation:

Same filtering logic as Question 5 (Fall 2025 + status contains 'Accepted'), but instead of counting rows, AVG(gpa) computes the mean GPA of those accepted applicants. ILIKE with wildcards ensures all acceptance status variations are captured regardless of case or surrounding text.

Answer: 3.7476

QUESTION 7: How many JHU MS Computer Science entries are there?

SQL:

```
SELECT COUNT(*) FROM applicants
WHERE (university ILIKE '%Johns Hopkins%' OR university ILIKE '%JHU%')
AND (program ILIKE '%Computer Science%'
    OR llm_generated_program ILIKE '%Computer Science%')
AND degree ILIKE '%Masters%';
```

Explanation:

This query uses multiple ILIKE pattern matches combined with OR/AND logic:

- University must contain 'Johns Hopkins' or 'JHU' (handles name variations).
- Program must contain 'Computer Science' in either the original user-entered program field or the llm_generated_program field (the LLM-standardized version). This catches entries where the original program name might be abbreviated (e.g., "MSCS") but the LLM normalized it.
- Degree must contain 'Masters' to exclude PhD entries.

All three groups must be true (AND) for a row to be counted.

Answer: 6

QUESTION 8: How many PhD CS acceptances in 2025 at Georgetown, MIT, Stanford, or Carnegie Mellon? (Using original/standard fields)

SQL:

```
SELECT COUNT(*) FROM applicants
WHERE term LIKE '%2025%'
AND status ILIKE '%Accepted%'
AND (university ILIKE '%Georgetown%' OR university ILIKE '%MIT%'
     OR university ILIKE '%Stanford%' OR university ILIKE '%Carnegie Mellon%')
AND program ILIKE '%Computer Science%'
AND degree ILIKE '%PhD%';
```

Explanation:

Counts rows matching ALL of these conditions:

- Term contains '2025' (covers Spring 2025, Fall 2025, etc.).
- Status contains 'Accepted' (case-insensitive).
- University matches one of the four target schools via ILIKE substring matching on the original user-entered university field.
- Program contains 'Computer Science' in the original field.
- Degree contains 'PhD'.

This uses the raw, user-entered data fields which may contain variations and abbreviations.

Answer: 18

QUESTION 9: Same as Q8 but using LLM-generated (standardized) fields

SQL:

```
SELECT COUNT(*) FROM applicants
WHERE term LIKE '%2025%'
AND status ILIKE '%Accepted%'
AND (llm_generated_university IN (
    'Georgetown University',
    'Massachusetts Institute of Technology',
    'Stanford University',
    'Carnegie Mellon University'))
AND llm_generated_program ILIKE '%Computer Science%'
AND degree ILIKE '%PhD%';
```

Explanation:

Same logic as Question 8, but uses the LLM-standardized fields instead. Because the LLM has normalized university names to canonical forms, this query can use exact IN(...) matching instead of fuzzy ILIKE patterns. For example, 'MIT' entries have been standardized to 'Massachusetts Institute of Technology'. The fact that Q8 and Q9 return the same count (18) validates that the LLM normalization is consistent with the original data for these schools.

Answer: 18

--- Curiosity Questions ---

QUESTION 10: What is the average GRE Quant score for MIT applicants?

SQL:

```
SELECT AVG(gre) FROM applicants
WHERE university ILIKE '%MIT%'
OR llm_generated_university = 'Massachusetts Institute of Technology';
```

Explanation:

AVG(gre) calculates the mean GRE Quantitative score. The WHERE clause uses OR to capture MIT applicants from both data sources: the original university field (via ILIKE substring match for 'MIT') and the LLM-standardized field (via exact match). This ensures no MIT applicants are missed regardless of how they were originally entered.

Answer: 229.02

QUESTION 11: Which university has the highest average GPA among accepted students?

SQL:

```
SELECT university, AVG(gpa) as avg_gpa
FROM applicants
WHERE status ILIKE '%Accepted%' AND gpa IS NOT NULL
GROUP BY university
ORDER BY avg_gpa DESC
LIMIT 1;
```

Explanation:

This is an aggregation query that:

1. Filters to accepted students with a non-null GPA (WHERE clause).
2. Groups all remaining rows by university (GROUP BY).
3. Calculates the average GPA for each university group (AVG(gpa)).
4. Sorts results in descending order by average GPA (ORDER BY ... DESC).
5. Returns only the top result (LIMIT 1).

Note: The result ('University of Potsdam', 73.0) suggests a data quality issue -- a GPA of 73.0 likely uses a different grading scale (e.g., a percentage-based system) rather than the standard US 4.0 scale.

Answer: University of Potsdam, with an average GPA of 73.0