```
"q1": """
    SELECT COUNT(*)
    FROM applicants
    WHERE term = 'Fall 2026';
  """,
```

For Question 1: Since the question is asking for all the entries that have applied for Fall 2026, I queried for where the term is Fall 2026, and counted all the entries.

```
  "q2": """
    SELECT ROUND(
      100.0 * SUM(CASE WHEN us_or_international NOT IN ('American', 'Other') THEN 1
ELSE 0 END) / COUNT(*),
      2
    )
    FROM applicants;
  """,
```

For Question 2: The question is asking the percentage of entries that are international students, therefore I used a SUM and a CASE expression such that it would create a counter that counts the number of entries that the us_or_international is neither American nor Other. This was then divided by the total number of entries to get a fraction and that reaction is multiplied by 100 to get the percentage. I ultimately used the round function to round the answer to 2 decimal points.

```
  "q3": """
    SELECT
      ROUND(AVG(gpa)::numeric, 2),
      ROUND(AVG(gre)::numeric, 2),
      ROUND(AVG(gre_v)::numeric, 2),
      ROUND(AVG(gre_aw)::numeric, 2)
    FROM applicants;
  """,
```

For Question 3: The question asks for the overall averages of GPA, GRE total, GRE verbal, and GRE analytical writing across all applicants. I used the AVG function on each of these columns and cast the results to numeric so that I could apply the ROUND function. Each value is rounded to two decimal places to make the results easier to interpret and consistent with typical reporting standards.

```
  "q4": """
```

```
    SELECT ROUND(AVG(gpa)::numeric, 2)
    FROM applicants
    WHERE us_or_international = 'American' AND term = 'Fall 2026';
  """,
```

For Question 4: This query calculates the average GPA specifically for American applicants who applied for Fall 2026. I filtered the dataset using a WHERE clause to include only rows where us_or_international is equal to 'American' and the term is 'Fall 2026'. I then computed the average GPA and rounded it to two decimal places.

```
  "q5": """
    SELECT ROUND(
      100.0 * SUM(CASE WHEN status ILIKE 'Accepted on%' THEN 1 ELSE 0 END) /
COUNT(*),
      2
    )
    FROM applicants
    WHERE term = 'Fall 2026';
  """,
```

For Question 5: I used a CASE statement inside a SUM to count how many applicants have a status that begins with 'Accepted on'. This count is divided by the total number of Fall 2026 applicants to obtain a fraction, which is then multiplied by 100 to convert it into a percentage. The final result is rounded to two decimal places.

```
  "q6": """
    SELECT ROUND(AVG(gpa)::numeric, 2)
    FROM applicants
    WHERE term = 'Fall 2026' AND status ILIKE 'Accepted on%';
  """,
```

For Question 6: This query finds the average GPA of applicants who were accepted for Fall 2026. I filtered the data to include only entries from Fall 2026 with a status indicating acceptance. The AVG function was then applied to the GPA column, and the result was rounded to two decimal places.

```
  "q7": """
    SELECT COUNT(*)
    FROM applicants
    WHERE degree = 'Masters' AND program = 'Computer Science' AND university = 'Johns
Hopkins University';
  """,
```

For Question 7: The purpose of this query is to count how many applicants applied to the Master's program in Computer Science at Johns Hopkins University. I used multiple

conditions in the WHERE clause to ensure that only entries matching the specified university, degree type, and program were included. The total number of matching rows was then returned using COUNT(*).

```
  "q8": """
    SELECT COUNT(*)
    FROM applicants
    WHERE term ILIKE '%2026%'
    AND (
       university ILIKE 'Georgetown University'
       OR university ILIKE '%Massachusetts Institute of Technology%'
       OR university ILIKE 'Stanford%'
       OR university ILIKE 'Carnegie Mellon University'
       )
    AND degree = 'PhD'
    AND program ILIKE '%Computer Science%'
    AND status ILIKE 'Accepted on%';
  """,
```

For Question 8: This question focuses on the number of PhD applicants in Computer Science who were accepted in 2026 at a selected group of universities. I filtered the dataset to include only terms containing "2026," limited the universities using multiple OR conditions, restricted the degree to PhD, and required the program to contain "Computer Science." I also filtered for accepted applicants. The final output is the count of all matching records.

```
  "q9": """
    SELECT COUNT(*)
    FROM applicants
    WHERE term ILIKE '%2026%'
      AND (
       llm_generated_university ILIKE 'Georgetown University'
       OR llm_generated_university ILIKE '%Massachusetts Institute of Technology%'
       OR llm_generated_university ILIKE 'Stanford%'
       OR llm_generated_university ILIKE 'Carnegie Mellon University'
       )
      AND degree = 'PhD'
      AND llm_generated_program ILIKE '%Computer Science%'
      AND status ILIKE 'Accepted on%';
  """,
```

For Question 9: This query is similar to Question 8, but instead of using the original university and program fields, it relies on the LLM-generated university and program

columns. This allows for a comparison between structured input data and normalized LLM-processed fields, while applying the same filters for year, degree, program, university, and acceptance status.

```
"q10": """
    SELECT COUNT(*)
    FROM applicants
    WHERE university = 'Johns Hopkins University'
      AND status ILIKE 'Accepted on%'
      AND us_or_international NOT IN ('American', 'Other');
""",
```

For Question 10: I filtered for accepted applicants at Johns Hopkins and excluded applicants whose us_or_international status was 'American' or 'Other'. The result is a count of accepted international students at that institution.

```
"q11": """
    SELECT ROUND(AVG(gpa)::numeric, 2)
    FROM applicants
    WHERE university = 'Harvard University'
      AND program = 'Computer Science'
      AND status ILIKE 'Accepted on%';
""",
```

For Question 11: This query calculates the average GPA of accepted Computer Science applicants at Harvard University. I filtered the data by university, program, and acceptance status, then used the AVG function to compute the mean GPA. The result is rounded to two decimal places for clarity.