```
# How many entries do you have in your database who have applied for Fall 2025?
   query1 = """
   SELECT COUNT(*)
   FROM applicants
   WHERE term LIKE '%Fall 2025%'
   """
```

I selected all entries in the table that had “Fall 2025” as their term.

```
# What percentage of entries are from international students (not American or Other)
(to two decimal places)?
   query2 = """
   SELECT CAST(
       (COUNT(CASE WHEN us_or_international = 'International' THEN 1 END) * 100.0 /
COUNT(*)) AS DECIMAL(4,2)
   )
   FROM applicants
   """
```

I selected all entries with the “International” value as their us_or_international. I took the count of those number of occasions, divided by the total count of entries, multiplied by 100 in order to get a percentage, then rounded

```
# What is the average GPA, GRE, GRE V, GRE AW of applicants who provide these metrics?
   query3a = """
   SELECT AVG(gpa)
   FROM applicants
   WHERE gpa IS NOT NULL
   """
   execute_query("Average GPA", query3a)

   query3b = """
   SELECT AVG(gre)
   FROM applicants
   WHERE gre IS NOT NULL
   """
   execute_query("Average GRE", query3b)

   query3c = """
   SELECT AVG(gre_v)
   FROM applicants
   WHERE gre_v IS NOT NULL
   """
```

```
    execute_query("Average GRE V", query3c)

    query3d = """
    SELECT AVG(gre_aw)
    FROM applicants
    WHERE gre_aw IS NOT NULL
    """
```

For each score, I checked if their value was not empty then took the average of the scores.

```
# What is their average GPA of American students in Fall 2025?
    query4 = """
    SELECT AVG(gpa)
    FROM applicants
    WHERE us_or_international = 'American'
    AND term LIKE '%Fall 2025%'
    AND gpa IS NOT NULL
    """
```

I checked upon each entry whether they were “American”, had the term as “Fall 2025”, and had a non-empty GPA. I took the average of the GPA for those entries that met these conditions.

```
# What percent of entries for Fall 2025 are Acceptances (to two decimal places)?
    query5 = """
    SELECT CAST(
        (COUNT(CASE WHEN status LIKE '%Accepted%' THEN 1 END) * 100.0 / COUNT(*)) AS
DECIMAL(5,2)
    )
    FROM applicants
    WHERE term LIKE '%Fall 2025%'
    """
```

In entries that had term as “Fall 2025”, I counted the number of those entries that were accepted, divided by the total number of entries, then multiplied by 100 to get the percentage.

```
# What is the average GPA of applicants who applied for Fall 2025 who are Acceptances?
   query6 = """
   SELECT AVG(gpa)
   FROM applicants
   WHERE term LIKE '%Fall 2025%'
   AND status LIKE '%Accepted%'
   AND gpa IS NOT NULL
   """
```

I checked for entries with term “Fall 2025”, status of “Accepted”, a non-empty gpa, then took te average of the gpa of entries that met these conditions.

```
# How many entries are from applicants who applied to JHU for a masters degree in
Computer Science?
   query7 = """
   SELECT COUNT(*)
   FROM applicants
   WHERE llm_generated_program LIKE '%Computer Science'
   AND llm_generated_university LIKE '%Johns Hopkins%'
   """
```

I took the count of entries that had llm_generated_program “Computer Science” and llm_generated_university “Johns Hopkins”.

```
query8 = """
   SELECT COUNT(*)
   FROM applicants
   WHERE date_added >= '2025-01-01'
   AND date_added < '2026-01-01'
   AND status LIKE '%Accepted%'
   AND llm_generated_program LIKE '%Computer Science'
   AND llm_generated_university LIKE '%Georgetown%'
   """
```

I took the total count of entries that were added in the year 2025, were accepted, and had llm_generated_program of “Computer Science” and llm_generated_university of “Georgetown”.

My Queries:

```
# What is the acceptance rate by degree level?
   query9 = """
   SELECT
       degree,
       CAST(
           (COUNT(CASE WHEN status LIKE '%Accepted%' THEN 1 END) * 100.0 / COUNT(*))
AS float
       ) as acceptance_rate
   FROM applicants
   WHERE degree IS NOT NULL
   GROUP BY degree
   ORDER BY acceptance_rate DESC
   """
```

For each degree present in the table, I checked how many were accepted as a percentage and displayed them in ascending order.

```
# Which universities have the lowest acceptance rate?
   query10 = """
   SELECT
       llm_generated_university,
       CAST(
           (COUNT(CASE WHEN status LIKE '%Accepted%' THEN 1 END) * 100.0 / COUNT(*))
AS float
       ) as acceptance_rate
   FROM applicants
   WHERE llm_generated_university IS NOT NULL
   GROUP BY llm_generated_university
   HAVING COUNT(*) >= 25
   ORDER BY acceptance_rate ASC
   LIMIT 10
   """
```

For each university in the table, I checked for the non-null ones, then took the ratio of accepted to overall applications as a percentage and displayed them in ascending order. Initially, I did not have a HAVING clause but it displayed my schools with 0% acceptance rate. I came to the conclusion that these schools either were very difficult to get into or had a few applications that all got rejected, but I think the latter is true. So I just took universities with over 25 applications to make it more fair.