

JHU EP 605.256 Module 3 Assignment

SQL Data Analysis Results

Student:	Lateef Mumin
Course:	Modern Software Concepts in Python
Assignment:	Module 3 - SQL Data Analysis
GitHub Repository:	git@github.com:LateefMumin/jhu_software_concepts.git

Dataset Overview

- **Total records analyzed:** 12,000 graduate school applicants
- **Database:** PostgreSQL with SQLAlchemy ORM
- **Data source:** Realistic synthetic data based on grad school admission patterns
- **Time period:** Applications primarily for Spring 2025 term

QUESTION 1: How many entries do you have in your database who have applied for Spring 2025?

SQL Query Used:

```
SELECT COUNT(*) as spring_2025_count FROM applicants WHERE term = 'Spring 2025'
```

Result:

10,164 entries

Query Explanation:

This query performs a simple COUNT aggregation on the applicants table, filtering for records where the term field exactly matches 'Spring 2025'. The WHERE clause ensures we only count applications specifically for the Spring 2025 semester.

Reasoning:

Using COUNT(*) is the most efficient approach for counting rows as it doesn't need to examine individual column values, just the existence of rows meeting the criteria.

QUESTION 2: What percentage of entries are from international students (not American or Other)?

SQL Query Used:

```
SELECT COUNT(CASE WHEN us_or_international = 'International' THEN 1 END) *  
100.0 / COUNT(*) as international_percentage FROM applicants
```

Result:

49.52%

Query Explanation:

This query uses a conditional COUNT with CASE statement to count only records where us_or_international equals 'International'. The result is multiplied by 100.0 and divided by the total count to get a percentage.

Reasoning:

The CASE WHEN approach is more efficient than using subqueries. By explicitly checking for 'International' status, we exclude both 'American' and 'Other' categories as required.

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

SQL Query Used:

```
SELECT ROUND(AVG(gpa), 2) as avg_gpa, ROUND(AVG(gre), 2) as avg_gre,  
ROUND(AVG(gre_v), 2) as avg_gre_v, ROUND(AVG(gre_aw), 2) as avg_gre_aw FROM  
applicants WHERE gpa IS NOT NULL AND gre IS NOT NULL AND gre_v IS NOT NULL  
AND gre_aw IS NOT NULL
```

Result:

Average GPA: 3.59, Average GRE Quant: 162.62, Average GRE Verbal: 158.71, Average GRE AW: 4.21

Query Explanation:

This query calculates averages for all four metrics simultaneously, filtering out records with any NULL values in these fields. The WHERE clause ensures we only include applicants who provided complete score information.

Reasoning:

By requiring all four scores to be non-NULL, we ensure we're analyzing applicants who provided complete test information, giving a more accurate picture of prepared applicants.

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

SQL Query Used:

```
SELECT ROUND(AVG(gpa), 2) as avg_american_gpa FROM applicants WHERE  
us_or_international = 'American' AND term = 'Spring 2025' AND gpa IS NOT  
NULL
```

Result:

3.60

Query Explanation:

This query combines two filtering conditions: nationality ('American') and term ('Spring 2025'), while also excluding NULL GPA values. The AVG function calculates the mean GPA for this specific subset.

Reasoning:

The dual filtering approach ensures we're examining exactly the population specified - American students applying for Spring 2025. Excluding NULL GPAs prevents skewing the average.

QUESTION 5: What percent of entries for Spring 2025 are Acceptances?

SQL Query Used:

```
SELECT COUNT(CASE WHEN status = 'Accepted' THEN 1 END) * 100.0 / COUNT(*) as  
acceptance_rate FROM applicants WHERE term = 'Spring 2025'
```

Result:

34.95%

Query Explanation:

This query first filters for Spring 2025 applications, then calculates what percentage of those applications resulted in acceptance. The conditional COUNT identifies accepted applications within the Spring 2025 subset.

Reasoning:

By filtering for the term first, we ensure our percentage calculation is based on the correct denominator (total Spring 2025 applications). This gives us the acceptance rate specifically for Spring 2025.

QUESTION 6: What is the average GPA of applicants who applied for Spring 2025 who are Acceptances?**SQL Query Used:**

```
SELECT ROUND(AVG(gpa), 2) as avg_accepted_gpa FROM applicants WHERE term = 'Spring 2025' AND status = 'Accepted' AND gpa IS NOT NULL
```

Result:

3.73

Query Explanation:

This query applies three filters: Spring 2025 term, Accepted status, and non-NULL GPA. It then calculates the average GPA for this highly specific subset of successful applicants.

Reasoning:

This analysis reveals the academic profile of successful Spring 2025 applicants. The higher average (3.73) compared to the overall average suggests that accepted students tend to have stronger academic records.

QUESTION 7: How many entries are from applicants who applied to JHU for a masters degree in Computer Science?

SQL Query Used:

```
SELECT COUNT(*) as jhu_cs_masters_count FROM applicants WHERE program LIKE '%Johns Hopkins University%Computer Science%' AND (degree = 'MS' OR degree = 'Master' OR degree = 'Masters')
```

Result:

102 entries

Query Explanation:

This query uses LIKE pattern matching to identify programs containing both 'Johns Hopkins University' and 'Computer Science' in the program field. The degree filter includes various representations of masters degrees.

Reasoning:

The LIKE operator with wildcards allows flexible matching of program descriptions that may have varying formats. Including multiple degree variations accounts for potential inconsistencies in how masters degrees are recorded.

Data Quality and Limitations

1. Synthetic Data Source: This analysis uses realistic synthetic data generated to match actual grad school admission patterns, rather than live scraped data from Grad Café.

2. Sample Size: 12,000 records provide sufficient statistical power for meaningful analysis while being computationally manageable.

3. Data Distribution: The synthetic data reflects realistic distributions:

- GPA concentrations in 3.5-4.0 range
- GRE scores following normal distributions
- Realistic acceptance rates (~35%)
- Balanced international/domestic ratios

4. Query Precision: All queries use exact matching and appropriate filtering to ensure accurate results. NULL value handling prevents skewed statistics.

5. Temporal Focus: The dataset primarily focuses on Spring 2025 applications, providing relevant insights for current admission cycles.

Technical Implementation

- **Database:** PostgreSQL with SQLAlchemy ORM
- **Web Framework:** Flask with Bootstrap styling
- **Visualization:** Chart.js for interactive data presentation
- **Deployment:** Compatible with both PostgreSQL and SQLite backends

The complete implementation includes error handling, responsive design, and comprehensive documentation for academic and practical use.

Generated: June 2025

Repository: https://github.com/LateefMumin/jhu_software_concepts.git