Q1

- **Question:** How many entries … applied for Fall 2026?
- **Answer:** 6990
- **SQL:** `SELECT COUNT(*) FROM applicants WHERE term = %s;`
- **Why:** `COUNT(*)` counts rows (entries). Filtering on `term = 'Fall 2026'` restricts to the requested application term.

Q2

- **Question:** What percentage are international students (not American or Other)?
- **Answer:** 44.36%
- **SQL:** `COUNT(*) FILTER (WHERE us_or_international = 'International')` and `COUNT(*)`
- **Why:** We need a numerator (international entries) and denominator (all entries). The FILTER clause counts only rows meeting the condition.

Q3

- **Question:** Average GPA, GRE, GRE V, GRE AW of applicants who provide these metrics?
- **Answer:** GPA 3.81, GRE 204.89, GRE-V 160.42, GRE-AW 8.51
- **SQL:** `AVG(...) FILTER (WHERE ... IS NOT NULL)`
- **Why:** AVG ignores NULLs, but FILTER makes “provided the metric” explicit and consistent per column.

Q4

- **Question:** Average GPA of American students in Fall 2026?
- **Answer:** 4.07
- **SQL:** `SELECT AVG(gpa) FROM applicants WHERE term = %s AND us_or_international = %s AND gpa IS NOT NULL;`
- **Why:** We filter to the correct group (American + Fall 2026) and exclude NULL GPAs.

Q5

- **Question:** What percent of entries for Fall 2026 are Acceptances?
- **Answer:** 24.32%
- **SQL:** counts accepted within Fall 2026 divided by total Fall 2026
- **Why:** Percent requires accepted_count / total_count for the same term.

Q6

- **Question:** Average GPA of applicants who applied for Fall 2026 who are Acceptances?
- **Answer:** 3.80
- **SQL:** `AVG(gpa)` with `WHERE term='Fall 2026' AND status='Accepted' AND gpa IS NOT NULL`
- **Why:** Restricts to the exact subgroup and averages only available GPAs.

Q7

- **Question:** How many entries are from applicants who applied to JHU for a masters degree in CS?
- **Answer:** 8
- **SQL/Python logic:** Candidate pull in SQL + normalization matching in Python
- **Why:** The dataset contains inconsistent spellings/formatting (e.g., “CS”, “C.S.”, “computer sci”, “M.S.”). Normalization (lowercase + punctuation removal + token checks) improves robustness without relying on exact string matches.

Q8

- **Question:** How many entries from 2026 are acceptances … (Georgetown/MIT/Stanford/CMU) for a PhD in CS?
- **Answer:** 2
- **SQL/Python logic:** SQL filters to `date_added` in 2026 + `status='Accepted'` + `degree='PhD'`, then match university + CS on the program text
- **Why:** Filtering in SQL reduces the search space and makes the logic efficient; matching handles university and program identification.

Q9

- **Question:** Does Q8 change if you use LLM generated fields?
- **Answer:** No. Downloaded = 2, LLM = 2, Difference = 0
- **SQL/Python logic:** Same filters, but match on `llm_generated_university` and `llm_generated_program`
- **Why:** LLM fields normalize the extracted university/program; comparing counts tests whether normalization changes the result.

Additional Question 1

- **Question:** In Fall 2026, do American and International applicants report different average GPAs?
- **Answer:** American 4.07, International 3.86

- **SQL:** `GROUP BY us_or_international` with `WHERE term='Fall 2026' AND ... IN ('American','International')`
- **Why:** Grouping computes separate averages per cohort for comparison.

Additional Question 2

- **Question:** In Fall 2026, how do acceptance rates differ between PhD and Masters?
- **Answer:** Masters 72.53%, PhD 18.18%
- **SQL:** `COUNT(*) FILTER (WHERE status='Accepted')` grouped by `degree`
- **Why:** This computes a subgroup acceptance rate using accepted_count / total_count per degree type.