Modern Concepts in Python: Spring 2026

by Eric Rying

Module 3: Database Queries Assignment Experiment

Feb 2026

Written Assignment Requirements

Having carried out this assignment, please write two paragraphs about the inherent limitations of carrying out analytics over anonymously submitted data items. Did the analytic responses surprise you? How does this different from standards? For example, the average GRE quantitative reasoning score was

157 for 2023-2023 and was nearly 165 for grad school entries submitted (see sample output). Why do you think that is? What might cause this to occur? Please place your essay into a file called limitations.pdf

Response:

Limitations of the GradCafe Data and Scraping Pipeline

Analytics performed on anonymously submitted data are inherently constrained by issues of data quality, completeness, and representativeness. Because participation is voluntary and unverified, users choose what information to share, how to format it, and whether to report sensitive metrics such as GPA or GRE scores. This leads to inconsistent coverage across fields, missing values, and occasional exaggeration or misreporting. The anonymity of platforms like GradCafe also introduces self-selection bias: applicants who post tend to be more anxious, more stats-focused, or more competitive than the general applicant pool. As a result, the dataset reflects the behavior and characteristics of people who choose to self-report online, not the broader population of graduate applicants. These limitations mean that any statistical patterns observed should be interpreted as informal community signals rather than rigorous or generalizable findings.

The analytic results did not surprise me once these biases were considered, but they differ sharply from standardized national metrics. For example, the average GRE Quantitative score in my scraped dataset was close to 165, far above the ETS national mean of roughly 157. This inflation is expected: applicants with unusually strong scores are more likely to post them publicly, while those with weaker scores often omit them or choose not to post at all. Additionally, many programs have made the GRE optional, so the only people reporting GRE scores on GradCafe tend to be those who performed well and feel confident sharing them. This combination of self-selection, survivorship bias, and incomplete reporting naturally pushes the averages upward. These patterns illustrate why analytics over anonymously submitted data must be interpreted cautiously: the dataset reflects who chooses to contribute to GradCafe, not the full population.

Technical Limitations and Field Extraction Challenges

A significant technical limitation emerged during data collection: GradCafe does not provide a functional API for retrieving detailed applicant information. Although a /api/result/{id} endpoint is documented online, all attempts to access it returned HTML error pages rather than JSON, indicating that the API is deprecated or inaccessible. To obtain detail fields such as GPA, GRE scores, citizenship, and application term, the scraper was forced to rely on HTML parsing and regex extraction from individual result pages. This approach is inherently fragile because the HTML structure varies across entries, and many users omit these fields entirely.

The initial scraping implementation achieved highly variable coverage across different fields when processing 35,000 entries:

- Citizenship: 100% coverage (34,995 of 34,995 entries)
- Degree level: ~99% coverage
- GPA: 1.2% coverage (403 of 34,995 entries)
- GRE Verbal: 6% coverage (2,230 of 34,995 entries)
- GRE Total: 0.02% coverage (only 6 entries)
- Application term: 0.3% coverage (110 of 34,995 entries)
- Comments: 0% coverage (extraction not a key focus or priority)

Root Cause Analysis

This unexpectedly low coverage for GPA, GRE, and term data resulted from multiple technical and implementation issues:

1. **Aggressive Scraping Strategy:** The scraper used 15 parallel threads with only 100ms delay between requests, resulting in approximately 150 requests per second. This aggressive approach likely triggered rate-limiting on GradCafe's servers, causing many requests to timeout or return incomplete HTML.

2. **Generic HTML Parsing:** The extraction code searched through generic , <p>, <li>, and <dd> tags across entire pages rather than targeting GradCafe's specific <dl> (definition list) structure where detail information is consistently formatted. This resulted in the parser finding irrelevant text or missing the data entirely.

3. **No Retry Logic**: When requests failed due to timeouts or rate-limiting, the scraper simply returned empty data with no retry attempts. There was no logging mechanism to track extraction failures or diagnose issues during the scraping process.

4. **Incomplete Implementation**: Comment extraction code was never implemented, resulting in 0% coverage for this field despite it being a required component.

Further testing with an improved parser that correctly targets GradCafe's <dl> structure confirmed that comments can be extracted successfully when present. However, manual inspection of sample detail pages revealed that most users simply do not provide GPA, GRE, or detailed term information—the fields are optional on GradCafe and commonly left blank. This represents a fundamental data availability issue beyond extraction failures.

Impact on Analysis

The term field's near-complete absence (0.3% coverage) prevents term-specific filtering and causes several assignment queries to return limited or N/A results:

- Q1 (Fall 2026 applicant count): Only 9 entries found
- Q4 (Average GPA of American students in Fall 2026): N/A (no data)
- Q5 (Fall 2026 acceptance rate): Limited to 9 entries
- Q6 (Average GPA of Fall 2026 acceptances): N/A (no data)
- Q8-Q9 (CS PhD acceptances in 2026): 0 results

However, queries that do not depend on these sparse fields remain robust:

- Q2 (International student percentage): Reliable with 100% citizenship coverage
- Q7 (JHU CS Masters applicants): 29 entries found
- Q11 (Acceptance rates by degree type): Strong results with ~34,000 entries

Lessons Learned

This experience demonstrates that data quality depends on both extraction strategy and source data availability. Even with improved parsing targeting the correct HTML structure, coverage would remain limited because users voluntarily omit most optional fields. The combination of implementation issues (aggressive scraping, generic parsing) and fundamental data sparsity (users not filling in fields) resulted in a dataset suitable for exploratory analysis but insufficient for robust statistical inference on metrics like GPA and GRE scores.

The successful elements—100% citizenship coverage, effective LLM-based standardization with CUDA acceleration (277 tokens/sec), and robust degree-level analysis—demonstrate that when data exists and extraction is properly implemented, the pipeline functions correctly. This underscores a critical lesson in data engineering: technical capability must be matched with realistic expectations about source data quality and respectful scraping practices to achieve reliable results.