Data 101/CS 187: Data Engineering Fall 2026 Discussion 01 – September 4

Review: SQL

Instructor: Aditya Parameswaran *Covers Lecture(s): Review*

I Execution Order

1. Warmup: Add a number indicating the order of execution of the below query.

SELECT	S	☐
FROM	$R_1,\ R_2,\ \ldots$	☐
WHERE	C_1	☐
GROUP BY	$A_1,\ A_2,\ \ldots$	☐
HAVING	C_2	☐
ORDER BY	S	☐
LIMIT	c	☐

II SQL Review

Suppose a full database schema is as follows:

- `books (book_id, book_name, year, genre, author_id, times_trending)`
- `authors (author_id, author_name, publishing_company, debut_year)`
- `awards (award_id, book_id, award_name)`

Write a SQL query to accomplish each task below:

2. Select only the book names and years from the `books` table.

3. Find the names of all authors whose debut year is in the 21st century (defined as after 2000).

4. Find the names of all authors who released a *sci-fi* genre book in 1974 that won an award.

III SQL Aggregation Review

In some cases, we may need to accumulate results by combining multiple rows via *aggregation*. We use the same schema as in Section 2.

5. Write a query that calculates the total number of books written by author Agatha Christie.

6. The *group-by* construct lets us compute aggregates for different chunks of a relation. Find the total number of books released per genre. The output should contain two columns: genre and the total count in that genre. Don't include genres with a count less than 2. (Hint: `HAVING`)

IV Optional Questions

7. Find the names of the 5 books that trended the least, ordered from least to most. Break ties by book name in alphabetical order.

8. **Challenge:** Write a query that computes the total times each author has had a trending book. The output should have two columns: author name and the total times. (Hint: you will need to use a join for this as well.)