Database Systems

SQL

Utah Tech University—Department of Computing

Spring 2024

Anatomy of a SQL query

Consider this query to find any student who has been in a class with Shankar:

```
SELECT a.name, b.name
FROM student AS a
JOIN takes AS a_takes ON a.ID = a_takes.ID
JOIN section ON
    a_takes.course_id = section.course_id AND
    a_takes.sec_id    = section.sec_id AND
    a_takes.semester  = section.semester AND
    a_takes.year      = section.year
JOIN takes AS b_takes ON
    b_takes.course_id = section.course_id AND
    b_takes.sec_id    = section.sec_id AND
    b_takes.semester  = section.semester AND
    b_takes.year      = section.year
JOIN student AS b ON b.ID = b_takes.ID
WHERE a.name = 'Shankar' AND a.ID <> b.ID
GROUP BY a.ID, b.ID, a.name, b.name
ORDER BY b.name
```

General strategy

- Expand the result set to capture all information you need. Focus on the structure of the data at this point.
- Filter down to the tuples you need. This does not change the structure, it just removes tuples.
- Aggregate to merge groups of tuples you want to consider as a single entry.
- Pick out the columns you want returned and consider sorting, truncating, etc.

More concretely

- Start with `FROM`, and use `JOIN`s to expand and link until all the data you need is included
- Use `WHERE` to filter it down to the rows you are interested in
- Use `GROUP BY` to combine rows and/or compute aggregate values
- Use `ORDER BY`, `LIMIT`, and `OFFSET` to control how results are returned
- Use `SELECT` to pick the columns you need

Schema

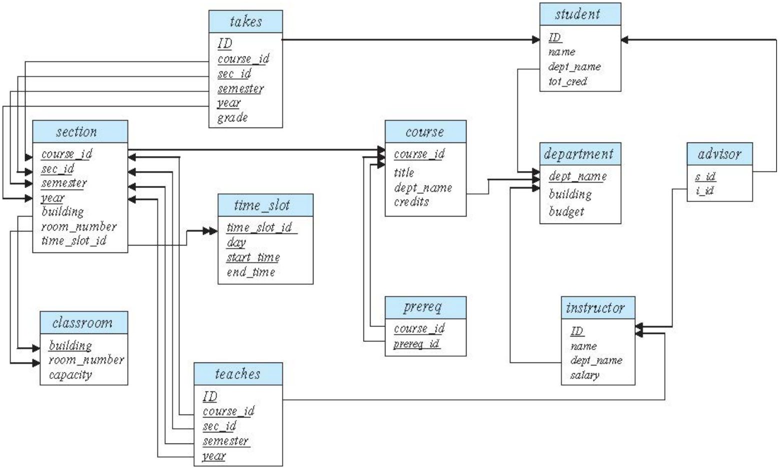

Figure 1: University dataset schema

Book chapter 3