

Yishu Yang

D-C1-2828-0

CISC 3000 - 001

### Assignment 3 (Chapter 4)

#### 4.2 Write the following queries in SQL:

- Display a list of all instructors, showing each instructor's ID and the number of sections taught. Make sure to show the number of sections as 0 for instructors who have not taught any section. Your query should use an outer join, and should not use subqueries.
- Write the same query as in part a, but using a scalar subquery and not using outer join.
- Display the list of all course sections offered in Spring 2018, along with the ID and name of each instructor teaching the section. If a section has more than one instructor, that section should appear as many times in the result as it has instructors. If a section does not have any instructor, it should still appear in the result with the instructor name set to "—".
- Display the list of all departments, with the total number of instructors in each department, without using subqueries. Make sure to show departments that have no instructors, and list those departments with an instructor count of zero.

4.3 Outer join expressions can be computed in SQL without using the SQL outer join operation. To illustrate this fact, show how to rewrite each of the following SQL queries without using the outer join expression.

- select \* from *student* **natural left outer join** *takes*
- select \* from *student* **natural full outer join** *takes*

4.2 Write the following queries in SQL:

a. Display a list of all instructors, showing each instructor's ID and the number of sections taught. Make sure to show the number of sections as 0 for instructors who have not taught any section. Your query should use an outer join, and should not use subqueries.

b. Write the same query as in part a, but using a scalar subquery and not using outer join.

c. Display the list of all course sections offered in Spring 2018, along with the ID and name of each instructor teaching the section. If a section has more than one instructor, that section should appear as many times in the result as it has instructors. If a section does not have any instructor, it should still appear in the result with the instructor name set to "—".

d. Display the list of all departments, with the total number of instructors in each department, without using subqueries. Make sure to show departments that have no instructors, and list those departments with an instructor count of zero.

Solution:

a. `SELECT i.ID, COUNT(t.sec_id) AS num_sections`

`FROM instructor i`

`LEFT OUTER JOIN teaches t ON i.ID = t.ID`

`GROUP BY i.ID;`

b. `SELECT i.ID, (SELECT COUNT(*)`

`FROM teaches t`

`WHERE t.ID = i.ID) AS num_sections`

`FROM instructor i;`

c. `SELECT s.course_id, s.sec_id, COALESCE(i.ID, '-') AS instructor_id, COALESCE(i.name, '-') AS instructor_name`

`FROM section s`

`LEFT OUTER JOIN teaches t ON s.course_id = t.course_id`

`AND s.sec_id = t.sec_id`

`AND s.semester = t.semester`

`AND s.year = t.year`

`LEFT OUTER JOIN instructor i ON t.ID = i.ID`

`WHERE s.semester = 'Spring' AND s.year = '2018';`

d. `SELECT d.dept_name, COUNT(i.ID) AS num_instructors`

`FROM department d`

`LEFT OUTER JOIN instructor i ON d.dept_name = i.dept_name`

`GROUP BY d.dept_name;`

4.3 Outer join expressions can be computed in SQL without using the SQL outer join operation. To illustrate this fact, show how to rewrite each of the following SQL queries without using the outer join expression.

a. `select * from student natural left outer join takes`

b. `select * from student natural full outer join takes`

Solution:

a. `SELECT s.*, t.course_id, t.sec_id, t.semester, t.year, t.grade`

`FROM student s`

`LEFT JOIN takes t ON s.ID = t.ID;`

b. `SELECT s.*, t.course_id, t.sec_id, t.semester, t.year, t.grade`

`FROM student s`

`LEFT JOIN takes t ON s.ID = t.ID`

`UNION`

`SELECT s.*, t.course_id, t.sec_id, t.semester, t.year, t.grade`

`FROM student s`

`RIGHT JOIN takes t ON s.ID = t.ID`

`WHERE s.ID IS NULL;`