**CSc 484 – Assignment #2 – AKUJOBI2**

**Due: 02-22-24 @ 9:30 AM**

**Team**:

* John Akujobi
* Izak Benitez-Lopez
* Frezer Berhanu

a.

-- a. List the IDs of all students who are advised by the instructor named Sullivan

SELECT student.ID as student\_ID

FROM student

JOIN advisor ON student.ID = advisor.s\_ID

JOIN instructor ON advisor.i\_ID = instructor.ID

WHERE instructor.name = 'Sullivan'

LIMIT 10;

A screenshot of a computer

Description automatically generated

--Count

SELECT COUNT(student.ID) as Total\_students

FROM student

JOIN advisor ON student.ID = advisor.s\_ID

JOIN instructor ON advisor.i\_ID = instructor.ID

WHERE instructor.name = 'Sullivan';

A black rectangular object with a white border

Description automatically generated

B.

-- b. List all instructors in alphabetical order by department name, and then by instructor’s name

SELECT id, name, dept\_name

FROM instructor

ORDER BY dept\_name, name

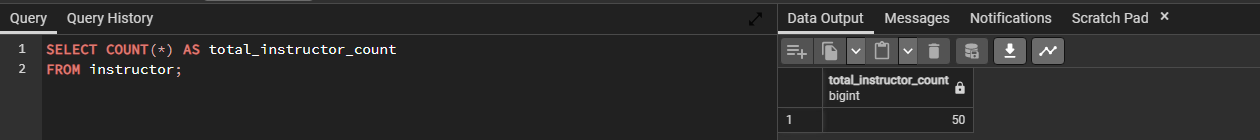
LIMIT 10;

A screenshot of a computer

Description automatically generated

SELECT COUNT(\*) AS total\_instructor\_count

FROM instructor;



C.

-- c. List all instructors with the lowest salary

--     There may be more than one instructor with the same salary

-- This one checks accross the board

SELECT \*

FROM instructor

WHERE salary = (

    SELECT MIN(salary)

    FROM instructor );

A black screen with green and white text

Description automatically generated

SELECT COUNT(\*) AS total\_instructors\_with\_min\_salary

FROM instructor

WHERE salary = (SELECT MIN(salary) FROM instructor);

A black rectangle with white text

Description automatically generated

D.

-- d. List the number of students in each department

SELECT dept\_name, COUNT(ID) AS student\_count

FROM student

GROUP BY dept\_name

ORDER BY dept\_name

LIMIT 10;

A screenshot of a computer

Description automatically generated

SELECT COUNT(DISTINCT ID) AS total\_students

FROM student;

A black rectangular object with a white border

Description automatically generated

E.

-- e. List the ID, name, salary of all instructors whose salary is greater than every average salary of every department

SELECT id, name, salary, dept\_name

FROM instructor

WHERE salary > ALL(SELECT AVG(salary) FROM instructor GROUP BY dept\_name)

LIMIT 10;

A screenshot of a computer

Description automatically generated

SELECT COUNT(\*) AS total\_instructors\_above\_avg\_salary

FROM instructor

WHERE salary > ALL (

    SELECT AVG(salary) FROM instructor GROUP BY dept\_name

);

A screenshot of a computer

Description automatically generated

F.

-- f. List the department name and the number of instructors for all departments with 2 or more instructors

SELECT dept\_name, COUNT(ID) AS instructor\_count

FROM instructor

GROUP BY dept\_name

HAVING COUNT(ID) >= 2

LIMIT 10;

A screenshot of a computer

Description automatically generated

G.

-- g. List the ID, name, and course\_id of all students enrolled in Fall 2004

SELECT student.ID, student.name, takes.course\_id

FROM student

JOIN takes ON student.ID = takes.ID

JOIN section ON takes.course\_id = section.course\_id AND takes.sec\_id = section.sec\_id

WHERE takes.semester = 'Fall' AND takes.year = 2004

LIMIT 10;

A screenshot of a computer program

Description automatically generated

SELECT COUNT(student.ID)

FROM student

JOIN takes ON student.ID = takes.ID

JOIN section ON takes.course\_id = section.course\_id AND takes.sec\_id = section.sec\_id

WHERE takes.semester = 'Fall' AND takes.year = 2004;

A computer screen with colorful text

Description automatically generated

H.

-- h. List the course\_id and the number of students enrolled of each course offered in Spring 2010

--     Courses with different sec\_id values are the same course

SELECT section.course\_id, COUNT(takes.ID) AS enrolled\_students

FROM section

JOIN takes ON section.course\_id = takes.course\_id AND section.sec\_id = takes.sec\_id

WHERE section.semester = 'Spring' AND section.year = 2010

GROUP BY section.course\_id

ORDER BY section.course\_id

LIMIT 10;

A screen shot of a computer

Description automatically generated

SELECT COUNT(DISTINCT section.course\_id) AS total\_courses\_spring\_2010

FROM section

JOIN takes ON section.course\_id = takes.course\_id AND section.sec\_id = takes.sec\_id

WHERE section.semester = 'Spring' AND section.year = 2010;

A screen shot of a computer

Description automatically generated

I.

-- i. List all student’s names who have never received an A or A- grade in any course

SELECT s.name

FROM student s

WHERE NOT EXISTS ( --Learned from stack overflow

    SELECT 1

    FROM takes t

    WHERE t.ID = s.ID AND t.grade IN ('A', 'A-') )

LIMIT 10;

A screenshot of a computer

Description automatically generated

SELECT COUNT(\*) AS total\_students\_never\_A\_Aminus

FROM student s

WHERE NOT EXISTS (

    SELECT 1

    FROM takes t

    WHERE t.ID = s.ID AND t.grade IN ('A', 'A-')

);

A screenshot of a computer

Description automatically generated

J.

-- j. List all student IDs and names for students who have not taken any courses offered before 2005

SELECT DISTINCT s.ID, s.name

FROM student s

LEFT JOIN (takes t JOIN section sec ON t.course\_id = sec.course\_id AND t.sec\_id = sec.sec\_id)

ON s.ID = t.ID AND sec.year < 2005

WHERE t.ID IS NULL

ORDER BY s.name

LIMIT 10;

A black screen with colorful text

Description automatically generated

SELECT COUNT(DISTINCT s.ID) AS total\_students\_no\_pre2005\_courses

FROM student s

LEFT JOIN (takes t JOIN section sec ON t.course\_id = sec.course\_id AND t.sec\_id = sec.sec\_id)

ON s.ID = t.ID AND sec.year < 2005

WHERE t.ID IS NULL;

A black background with many colorful lights

Description automatically generated with medium confidence

K.

-- k. List the highest instructor salary for each department, except Marketing

SELECT dept\_name, MAX(salary) AS max\_salary

FROM instructor

WHERE dept\_name <> 'Marketing'

GROUP BY dept\_name

ORDER BY dept\_name

LIMIT 10;

A screenshot of a computer

Description automatically generated

SELECT COUNT(\*) AS total\_departments\_excl\_marketing

FROM (

    SELECT dept\_name, MAX(salary) AS max\_salary

    FROM instructor

    WHERE dept\_name <> 'Marketing'

    GROUP BY dept\_name

) AS subquery;

A screenshot of a computer

Description automatically generated

L.

-- l. List the ID, name and the number of courses taught for all instructors

--     The number of courses is 0 for instructors who have not taught any courses

SELECT i.ID, i.name, COUNT(t.course\_id) AS number\_of\_courses\_taught

FROM instructor i

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

GROUP BY i.ID, i.name

ORDER BY i.ID

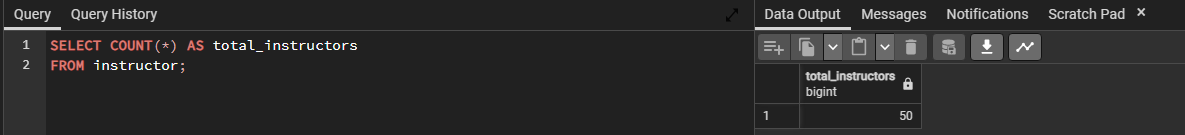
LIMIT 10;

A screenshot of a computer

Description automatically generated

SELECT COUNT(\*) AS total\_instructors

FROM instructor;



M.

-- m. List the name and the number of instructors assigned to each department

--     The number of instructors is 0 for departments that have no instructors

SELECT d.dept\_name, COUNT(i.ID) AS instructor\_count

FROM department d

LEFT JOIN instructor i ON d.dept\_name = i.dept\_name

GROUP BY d.dept\_name

ORDER BY d.dept\_name

LIMIT 10;

A screenshot of a computer program

Description automatically generated

SELECT COUNT(\*) AS total\_departments

FROM department;

A black rectangular object with white lines

Description automatically generated

N.

-- n. List the ID and name of all students who took courses in both Fall 2009 and Spring 2010

SELECT DISTINCT s.ID, s.name

FROM student s

JOIN takes t1 ON s.ID = t1.ID

JOIN takes t2 ON s.ID = t2.ID

WHERE t1.semester = 'Fall' AND t1.year = 2009

AND t2.semester = 'Spring' AND t2.year = 2010

ORDER BY s.name

LIMIT 10;

A screenshot of a computer

Description automatically generated

SELECT COUNT(DISTINCT s.ID) AS total\_students\_fall2009\_spring2010

FROM student s

JOIN takes t1 ON s.ID = t1.ID

JOIN takes t2 ON s.ID = t2.ID

WHERE t1.semester = 'Fall' AND t1.year = 2009

AND t2.semester = 'Spring' AND t2.year = 2010;

A screenshot of a computer

Description automatically generated

O.

-- o. List the ID and name of all students who have never taken a course at the university

SELECT DISTINCT s.ID, s.name

FROM student s

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

WHERE t.course\_id IS NULL

LIMIT 10;

A black screen with green text

Description automatically generated

SELECT COUNT(DISTINCT s.ID) AS total\_students\_never\_taken\_course

FROM student s

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

WHERE t.course\_id IS NULL;

A black screen with colorful text

Description automatically generated