Abishek S
EE18B001

CS3700:
Introduction to Database Systems
Assignment 2

October 30, 2021

1 Question 1

List the department(s) with at least two students who got 'S' grades in the 'International Practicum' course in the even semester of 2004, along with the count of such people.

Query

```
SELECT d.name as dname, COUNT(s.rollNo) AS numStudentsWithS
FROM department AS d, student AS s, enrollment AS e, course AS c
WHERE d.deptId = s.deptNo AND s.rollNo = e.rollNo AND c.courseID = e.courseID
        AND c.cname = 'International Practicum' AND e.grade = 'S'
        AND e.sem = 'even' AND e.year = 2004
GROUP BY d.name
HAVING COUNT(s.rollNo) >= 2;
```

Result Grid Filter Rows: Search Export:

dname	numStudentsWithS
Civil Eng.	2
Biology	2
Comp. Sci.	4
Physics	4
Athletics	2
Finance	5
Math	4
Geology	3
Marketing	2
Elec. Eng.	3
History	2
Mech. Eng.	3
Astronomy	3
Psychology	2

Figure 1: Question 1 query results

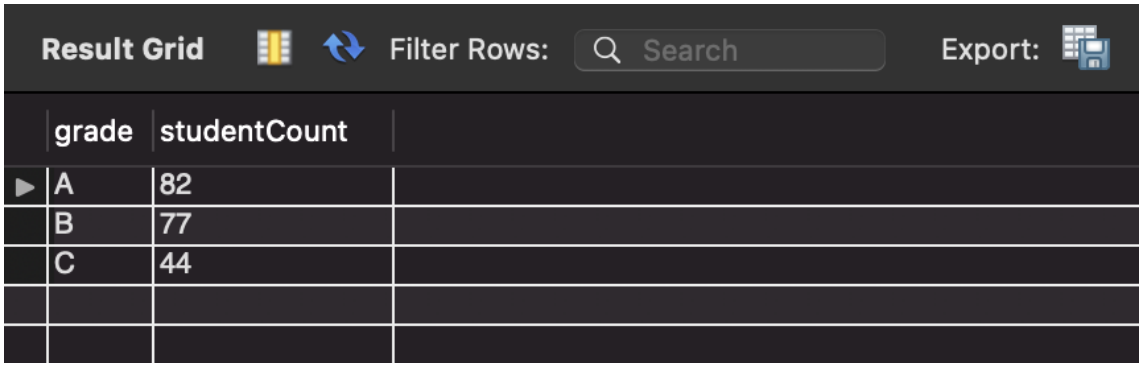
2 Question 2

Get the distribution of grades (number of students who got a grade) for the course 'Embedded Systems' taught in the odd semester of 2005 for the top 3 grades with the most number of people.

Query

```
SELECT grade, COUNT(rollNo) AS studentCount
FROM enrollment AS e, course AS c
WHERE e.courseId = c.courseId AND c.cname = 'Embedded Systems'
        AND e.year = 2005 AND e.sem = 'odd'
```

```
GROUP BY grade
ORDER BY studentCount DESC LIMIT 3;
```



grade	studentCount
A	82
B	77
C	44

Figure 2: Question 2 query results

3 Question 3

Find the average number of years of service for the professors in each department, where the professors considered have taught at least one course and gave less than 10 'S' grades in all course offering they taught (course offered in two different semesters or years are considered different).

Query

```
DROP VIEW IF EXISTS deptAndReqdProf;
CREATE VIEW deptAndReqdProf AS (
    SELECT deptNo, empId, startYear
    FROM professor
    WHERE empId NOT IN (
        SELECT p.empId
        FROM professor as p
        WHERE EXISTS (
            SELECT COUNT(rollNo) AS gradeCount
            FROM enrollment AS e, teaching AS t
            WHERE t.empId = p.empId AND e.courseId = t.courseId
                    AND e.year = t.year AND e.sem = t.sem AND e.grade = 'S'
            GROUP BY e.courseId, e.year, e.sem
            HAVING gradeCount >= 10
        ) OR NOT EXISTS (
            SELECT courseId
            FROM teaching t
            WHERE t.empId = p.empId
        )
    )
);

SELECT deptNo, AVG(2021-startYear) AS avgProfService
FROM deptAndReqdProf
GROUP BY deptNo;
```

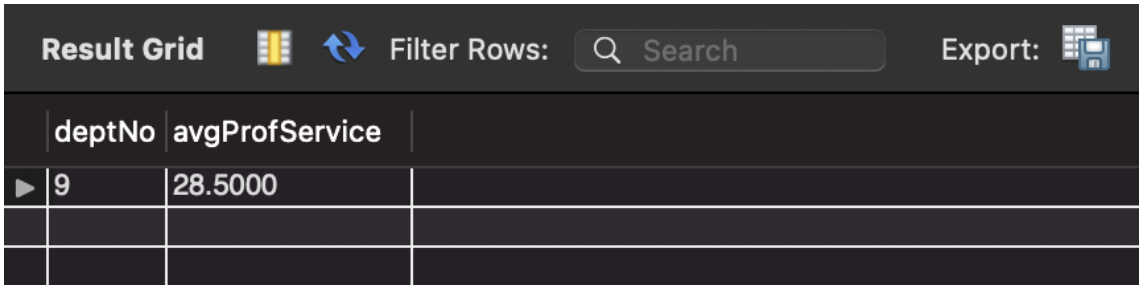


Figure 3: Question 3 query results

4 Question 4

Find the classrooms where more than 1 Computer Science course was taught between the years 2002 and 2018, both included (the same course taught in two different years or semesters are treated as different).

Query

```
SELECT classRoom
FROM teaching AS t, course AS c, department AS d
WHERE t.courseId = c.courseId AND c.deptNo = d.deptId AND d.name = 'Comp. Sci.'
        AND year >= 2002 AND year <= 2018
GROUP BY classRoom
HAVING COUNT(DISTINCT t.courseId, year, sem) > 1;
```

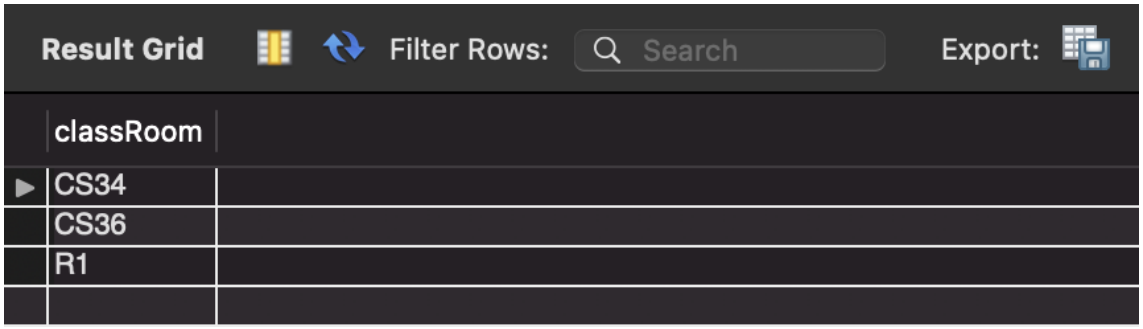


Figure 4: Question 4 query results

5 Question 5

List the course name, id, year, semester and the number of students enrolled for a course offered by the 'Comp. Sci.' department which had the highest number of enrolled students. Assume that there was at least one student enrolled in a course offered by the department.

Query

```
DROP VIEW IF EXISTS compSciCourseOfferingAndCount;
CREATE VIEW compSciCourseOfferingAndCount AS (
    SELECT c.courseId AS courseId, cname, year, sem, COUNT(rollNo) AS enrolledCount
    FROM enrollment AS e, course AS c, department AS d
    WHERE e.courseId = c.courseId AND c.deptNo = d.deptId AND d.name = 'Comp. Sci.'
    GROUP BY courseId, year, sem
);

SELECT courseId, cname, year, sem, enrolledCount
FROM compSciCourseOfferingAndCount
WHERE enrolledCount = ANY (
```

```
    SELECT MAX(enrolledCount)
    FROM compSciCourseOfferingAndCount
);
```

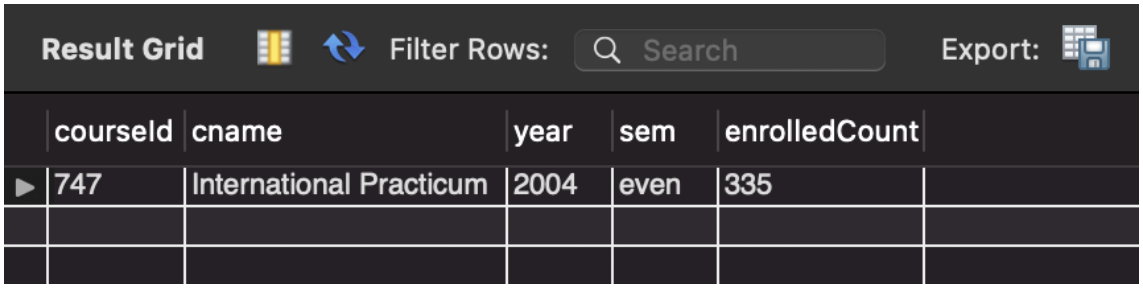


Figure 5: Question 5 query results

6 Question 6

List the number of credits completed (assume that obtaining a non 'U' grade is considered pass) by each student against his roll number and name.

Query

```
SELECT s.rollNo AS rollNo, s.name AS sname, SUM(c.credits) AS totalCredits
FROM (student AS s
        LEFT OUTER JOIN enrollment AS e ON s.rollNo = e.rollNo
        INNER JOIN course AS c ON e.courseID = c.courseID)
GROUP BY s.rollNo;
```

Result Grid | Filter Rows: Search | Export:

rollNo	sname	totalCredits
1000	Manber	29
10033	Zelty	22
10076	Duan	24
1018	Colin	46
10204	Mediratta	21
10267	Rzecz	20
10269	Hilberg	10
10454	Ugarte	27
10481	Grosch	37
10527	Kieras	24
10556	Reed	7
10663	Okaf	25
10693	Zabary	19
107	Shabuno	44
10705	Terauchi	19
10727	Allard	21
10736	Veselov...	17
108	Dhav	24

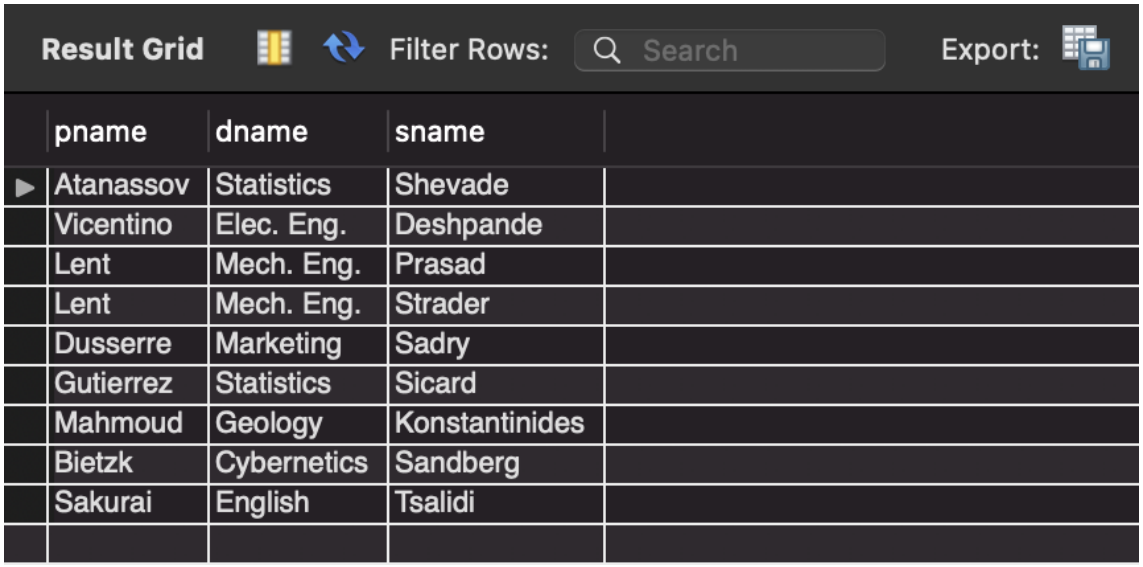
Figure 6: Question 6 query results (truncated as there are many rows in output)

7 Question 7

List the name and department name of female professors along with the name of students that contain 'sad' as a subsequence (case-insensitive) in their names and are/were advised by that professor.

Query

```
SELECT p.name AS pname, d.name AS dname, s.name as sname
FROM student AS s, professor AS p, department AS d
WHERE s.advisor = p.empId AND p.deptNo = d.deptId AND p.sex = 'female'
        AND UPPER(s.name) LIKE '%S%A%D%';
```



Result Grid | Filter Rows: Search | Export:

pname	dname	sname
Atanassov	Statistics	Shevade
Vicentino	Elec. Eng.	Deshpande
Lent	Mech. Eng.	Prasad
Lent	Mech. Eng.	Strader
Dusserre	Marketing	Sadry
Gutierrez	Statistics	Sicard
Mahmoud	Geology	Konstantinides
Bietzk	Cybernetics	Sandberg
Sakurai	English	Tsalidi

Figure 7: Question 7 query results

8 Question 8

List the name and employees ID of professors who taught one of the courses 'How to Groom your Cat' or 'Sanitary Engineering' and all its corresponding immediate (first level) pre-requisites at some point in time and in any order.

Query

```
(
    SELECT empId, name
    FROM professor AS p
    WHERE EXISTS (
        SELECT t.courseId
        FROM teaching AS t, course AS c
        WHERE t.courseId = c.courseId AND c.cname = 'How to Groom your Cat'
                AND t.empId = p.empId
    ) AND NOT EXISTS (
        SELECT preReqCourse
        FROM preRequisite, course AS c
        WHERE c.courseId = preRequisite.courseId AND c.cname = 'How to Groom your Cat'
                AND NOT EXISTS (
                    SELECT t.courseId
                    FROM teaching AS t
                    WHERE t.courseId = preReqCourse AND t.empId = p.empId
                )
    )
)
UNION
(
    SELECT empId, name
    FROM professor AS p
    WHERE EXISTS (
        SELECT t.courseId
        FROM teaching AS t, course AS c
        WHERE t.courseId = c.courseId AND c.cname = 'Sanitary Engineering'
                AND t.empId = p.empId
    ) AND NOT EXISTS (
        SELECT preReqCourse
        FROM preRequisite, course AS c
```

```
        WHERE c.courseId = preRequisite.courseId AND c.cname = 'Sanitary Engineering'
                AND NOT EXISTS (
                    SELECT t.courseId
                    FROM teaching AS t
                    WHERE t.courseId = preReqCourse AND t.empId = p.empId
                )
    )
);
```

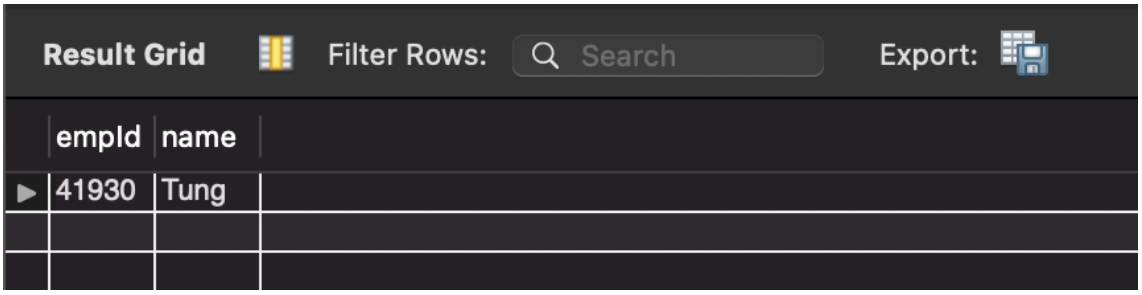


Figure 8: Question 8 query results