DBMS Assignment 2: Aggregates and Nested Subqueries

Name: Omkar Oak MIS: 112103099

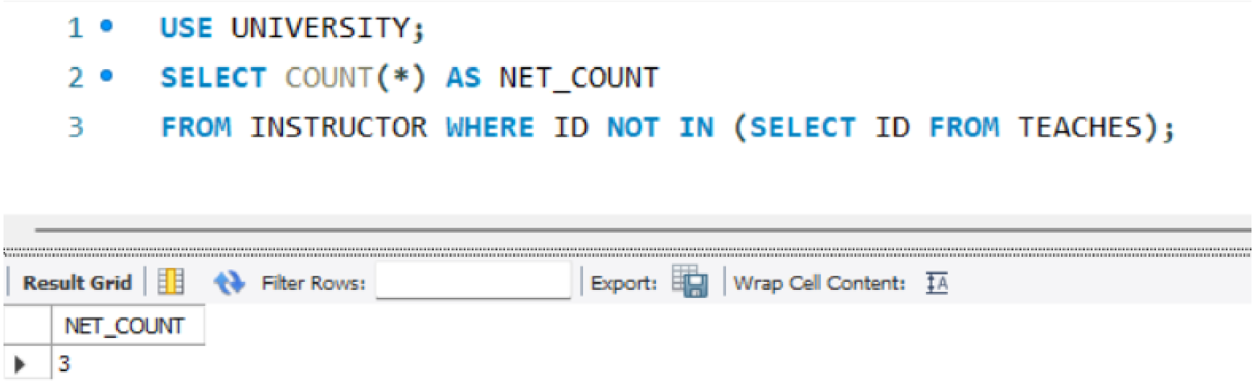
Div: 2 Batch: T1

Statement : Write the SQL queries using aggregates, grouping, ordering statements and nested subqueries for given statements on given schema.

A. Intermediate SQL: Aggregates and grouping and ordering

The following questions are all based on the university schema

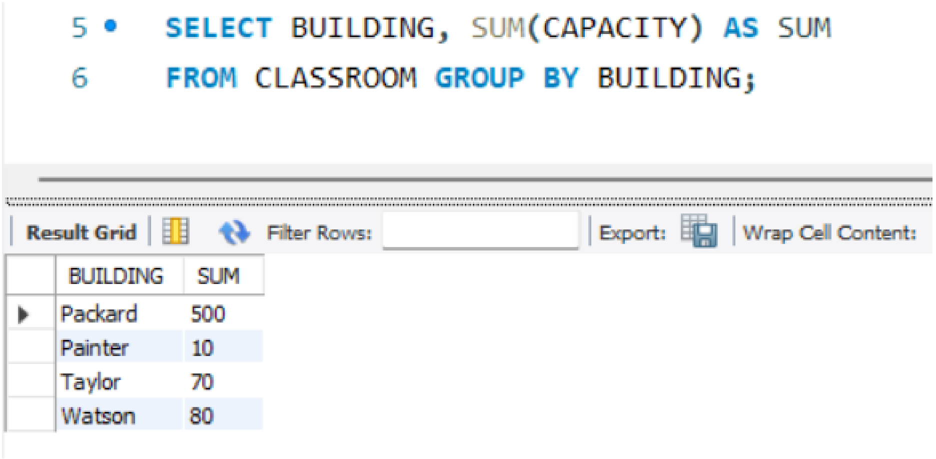
- Find the number of instructors who have never taught any course. If the result of your query is empty, add the appropriate data (and include corresponding insert statements) to ensure the result is not empty.NOTE: IN THE SUBMISSION FILE, PASTE DATA INSERTED BELOW THIS STATEMENT AS A REMARK.



```sql
USE UNIVERSITY;
SELECT COUNT(*) AS NET_COUNT
FROM INSTRUCTOR WHERE ID NOT IN (SELECT ID FROM TEACHES);
```

NET_COUNT
3

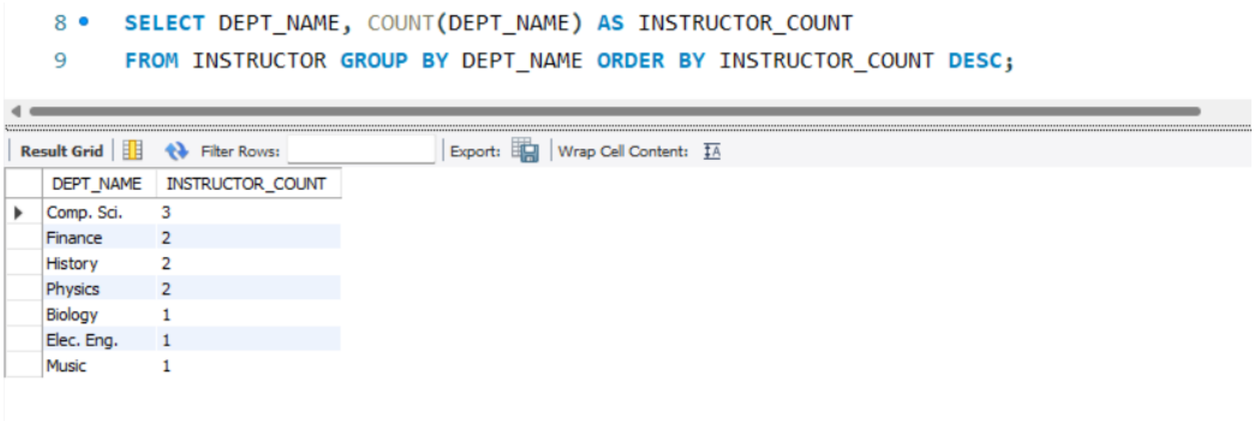
- Find the total capacity of every building in the university.



```sql
SELECT BUILDING, SUM(CAPACITY) AS SUM
FROM CLASSROOM GROUP BY BUILDING;
```

BUILDING	SUM
Packard	500
Painter	10
Taylor	70
Watson	80

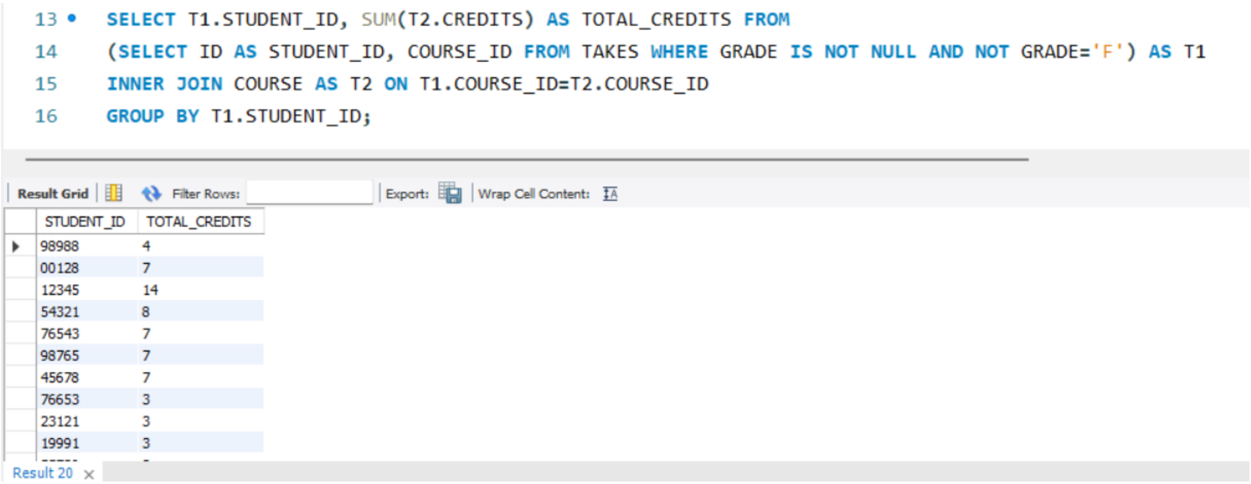
- Find all departments that have at least one instructor, and list the names of the departments along with the number of instructors; order the result in descending order of number of instructors.



```
SELECT DEPT_NAME, COUNT(DEPT_NAME) AS INSTRUCTOR_COUNT
FROM INSTRUCTOR GROUP BY DEPT_NAME ORDER BY INSTRUCTOR_COUNT DESC;
```

DEPT_NAME	INSTRUCTOR_COUNT
Comp. Sci.	3
Finance	2
History	2
Physics	2
Biology	1
Elec. Eng.	1
Music	1

- For each student, compute the total credits they have successfully completed, i.e. total credits of courses they have taken, for which they have a non-null grade other than 'F'. Do NOT use the tot_creids attribute of student.

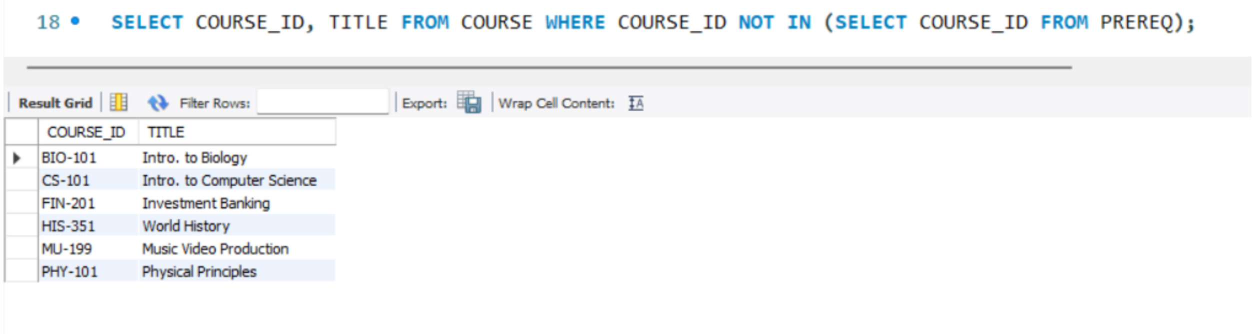


```
SELECT T1.STUDENT_ID, SUM(T2.CREDITS) AS TOTAL_CREDITS FROM
(SELECT ID AS STUDENT_ID, COURSE_ID FROM TAKES WHERE GRADE IS NOT NULL AND NOT GRADE='F') AS T1
INNER JOIN COURSE AS T2 ON T1.COURSE_ID=T2.COURSE_ID
GROUP BY T1.STUDENT_ID;
```

STUDENT_ID	TOTAL_CREDITS
98988	4
00128	7
12345	14
54321	8
76543	7
98765	7
45678	7
76653	3
23121	3
19991	3

B. **Nested Subqueries** : Write the following queries for university schema.

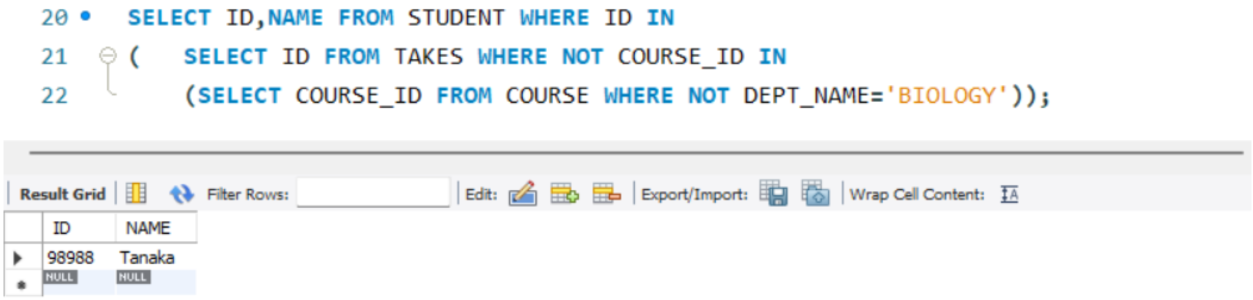
- Find the id and title of all courses which do not require any prerequisites.



```
SELECT COURSE_ID, TITLE FROM COURSE WHERE COURSE_ID NOT IN (SELECT COURSE_ID FROM PREREQ);
```

COURSE_ID	TITLE
BIO-101	Intro. to Biology
CS-101	Intro. to Computer Science
FIN-201	Investment Banking
HIS-351	World History
MU-199	Music Video Production
PHY-101	Physical Principles

- Find the names of students who have not taken any biology dept. courses.



```
SELECT ID,NAME FROM STUDENT WHERE ID IN
(    SELECT ID FROM TAKES WHERE NOT COURSE_ID IN
     (SELECT COURSE_ID FROM COURSE WHERE NOT DEPT_NAME='BIOLOGY'));
```

ID	NAME
98988	Tanaka
NULL	NULL