

# **DBMS ASSIGNMENT 2**

## **Aggregates and grouping and ordering**

**Name:** Bhaskar Subhash Pardeshi

**MIS:** 111703041

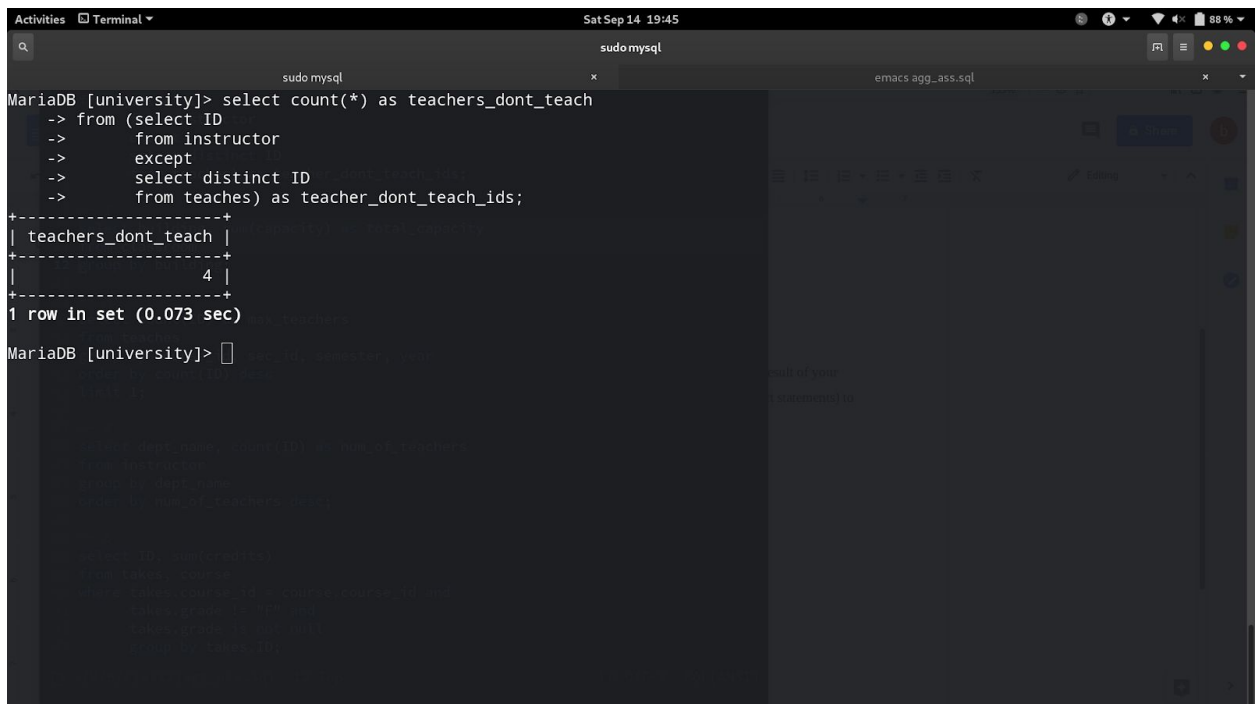
**Batch:** T3

**Year:** TY

- 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.

**Query:** select count(\*) as teachers\_dont\_teach  
from (select ID  
from instructor  
except  
select distinct ID  
from teaches) as teacher\_dont\_teach\_ids;

**Result:**



The screenshot shows a terminal window with a MySQL prompt. The user enters a query to find the number of instructors who have never taught any course. The result shows 1 row in the set, indicating that there is 1 instructor who has never taught any course.

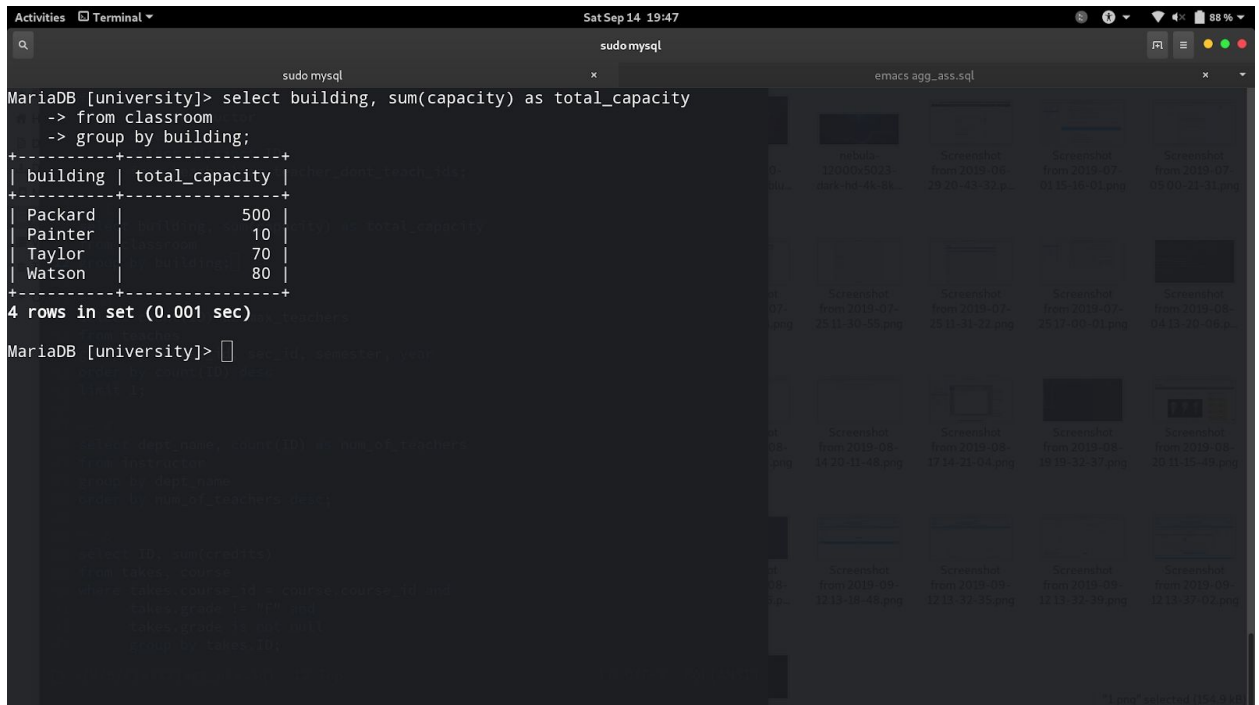
```
Activities Terminal Sat Sep 14 19:45
sudo mysql
MariaDB [university]> select count(*) as teachers_dont_teach
-> from (select ID
-> from instructor
-> except
-> select distinct ID
-> from teaches) as teacher_dont_teach_ids;
+-----+
| teachers_dont_teach |
+-----+
| 4 |
+-----+
1 row in set (0.073 sec)

MariaDB [university]>
```

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

**Query:** select building, sum(capacity) as total\_capacity  
from classroom  
group by building;

**Result:**



The screenshot shows a terminal window with a dark background. The title bar indicates 'Activities', 'Terminal', and the date 'Sat Sep 14 19:47'. The terminal prompt is 'MariaDB [university]>'. The user has entered the following SQL query:

```
select building, sum(capacity) as total_capacity
from classroom
group by building;
```

The result is displayed as a table with two columns: 'building' and 'total\_capacity'. The data is as follows:

building	total_capacity
Packard	500
Painter	10
Taylor	70
Watson	80

Below the table, the terminal shows '4 rows in set (0.001 sec)'. The prompt 'MariaDB [university]>' is visible again. In the background, several window thumbnails are visible, including one titled 'emacs agg\_ass.sql'.

- Find the maximum number of teachers for any single course section. Your output should be a single number. For example, if CS-101 section 1 in Spring 2012 had 3 instructors teaching the course, and no other section had more instructors teaching the section, your answer would be 3.

**Query:** select count(ID) as max\_teachers

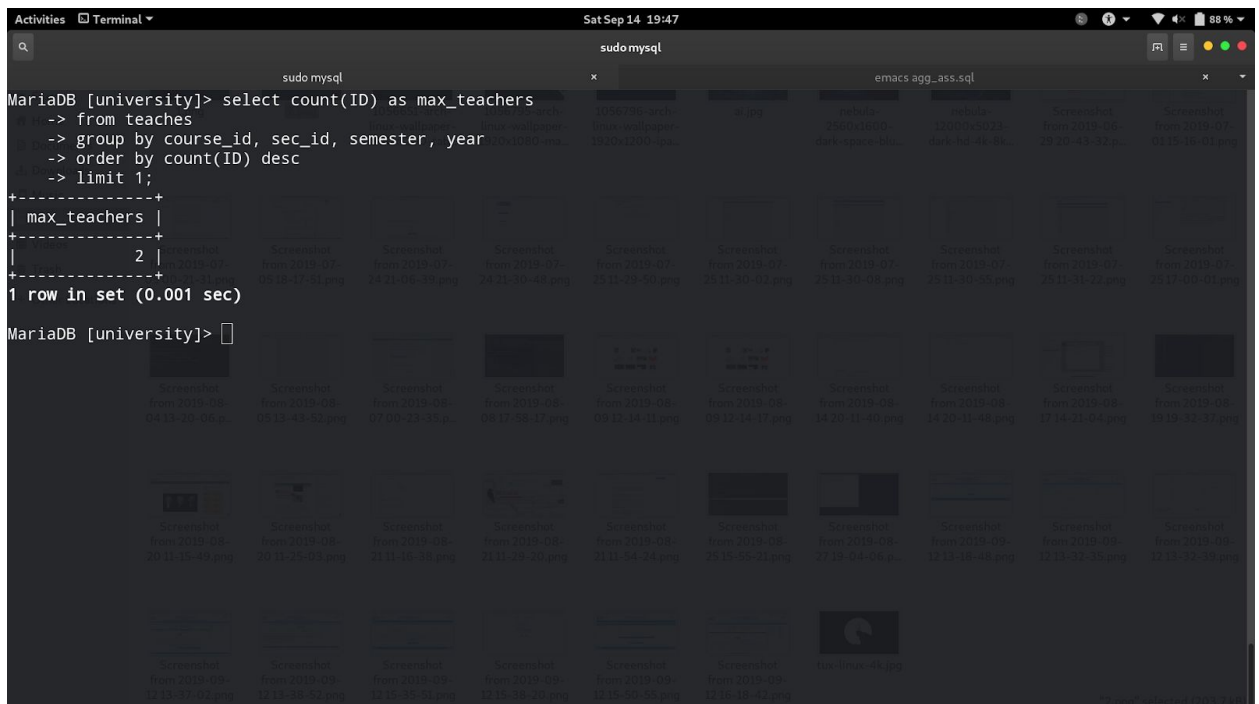
from teaches

group by course\_id, sec\_id, semester, year

order by count(ID) desc

limit 1;

**Result:**

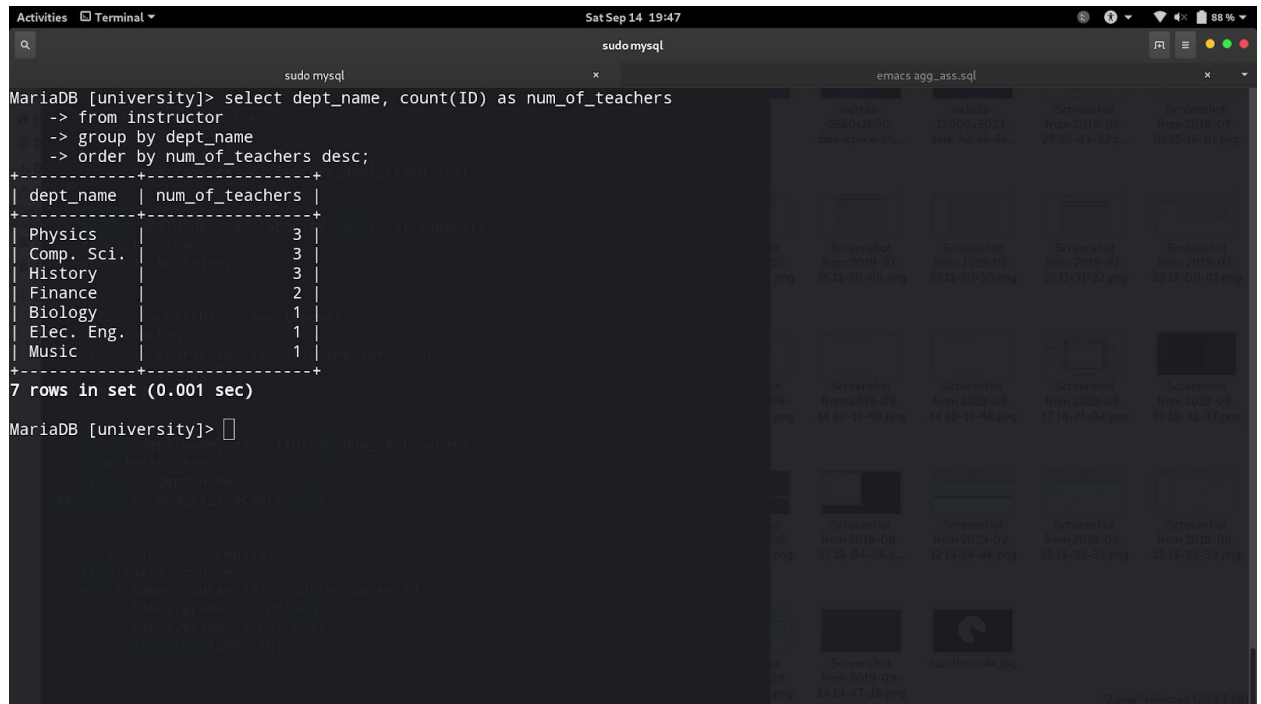


```
Activities Terminal Sat Sep 14 19:47
sudo mysql
MariaDB [university]> select count(ID) as max_teachers
-> from teaches
-> group by course_id, sec_id, semester, year
-> order by count(ID) desc
-> limit 1;
+-----+
| max_teachers |
+-----+
| 2 |
+-----+
1 row in set (0.001 sec)
MariaDB [university]>
```

- 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.

**Query:** select dept\_name, count(ID) as num\_of\_teachers  
from instructor  
group by dept\_name  
order by num\_of\_teachers desc;

**Result:**



The screenshot shows a terminal window with a MySQL prompt. The query executed is: select dept\_name, count(ID) as num\_of\_teachers from instructor group by dept\_name order by num\_of\_teachers desc;. The result is a table with 7 rows. The first three rows show departments with 3 teachers: Physics, Comp. Sci., and History. The next three rows show departments with 2 teachers: Finance, Biology, and Elec. Eng. The last row shows Music with 1 teacher. The terminal also shows the file manager in the background with various screenshots.

```
Activities Terminal Sat Sep 14 19:47
sudo mysql
MariaDB [university]> select dept_name, count(ID) as num_of_teachers
-> from instructor
-> group by dept_name
-> order by num_of_teachers desc;
+-----+-----+
| dept_name | num_of_teachers |
+-----+-----+
| Physics   | 3               |
| Comp. Sci. | 3               |
| History   | 3               |
| Finance   | 2               |
| Biology   | 1               |
| Elec. Eng. | 1               |
| Music     | 1               |
+-----+-----+
7 rows in set (0.001 sec)
MariaDB [university]>
```

- 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\_credits attribute of student.

**Query:** select ID, sum(credits)

from takes, course

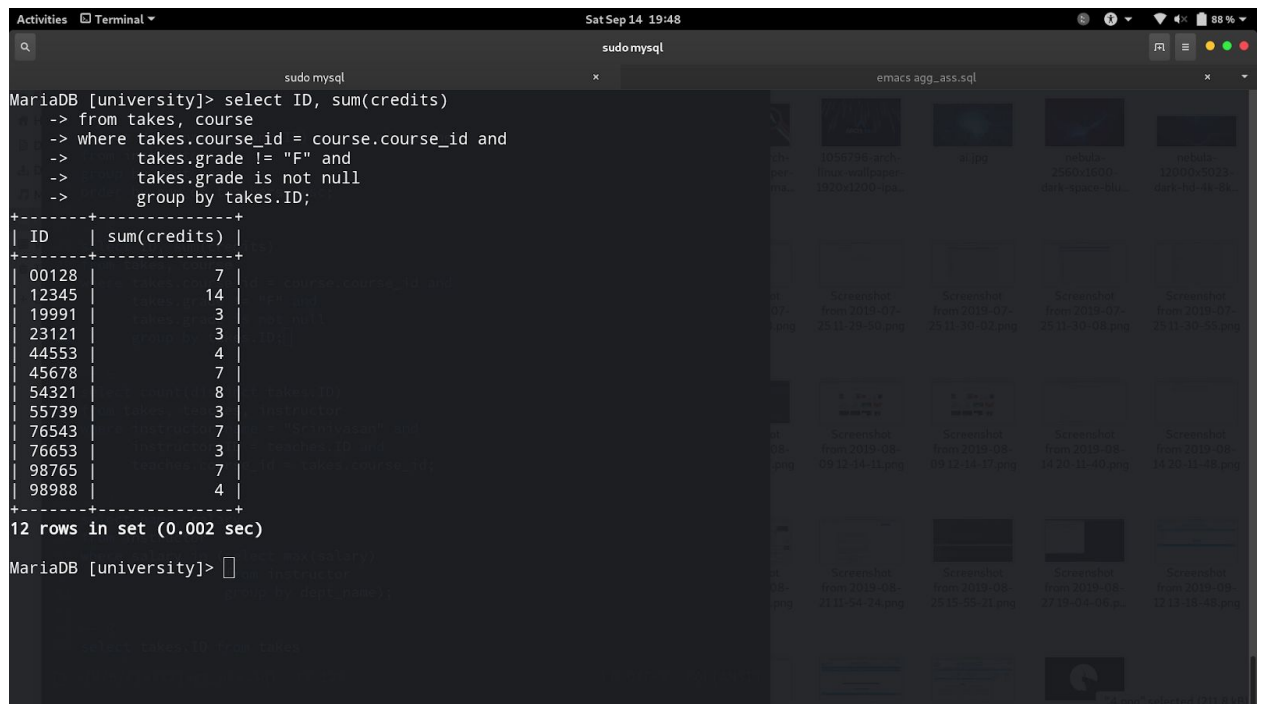
where takes.course\_id = course.course\_id and

takes.grade != "F" and

takes.grade is not null

group by takes.ID;

**Result:**



The screenshot shows a terminal window with a MySQL prompt. The query executed is: `select ID, sum(credits) from takes, course where takes.course_id = course.course_id and takes.grade != "F" and takes.grade is not null group by takes.ID;` The results are displayed in a table with 12 rows. The background shows a file manager window with various screenshot files.

```
MariaDB [university]> select ID, sum(credits)
-> from takes, course
-> where takes.course_id = course.course_id and
-> takes.grade != "F" and
-> takes.grade is not null
-> group by takes.ID;
+-----+-----+
| ID | sum(credits) |
+-----+-----+
| 00128 | 7 |
| 12345 | 14 |
| 19991 | 3 |
| 23121 | 3 |
| 44553 | 4 |
| 45678 | 7 |
| 54321 | 8 |
| 55739 | 3 |
| 76543 | 7 |
| 76653 | 3 |
| 98765 | 7 |
| 98988 | 4 |
+-----+-----+
12 rows in set (0.002 sec)

MariaDB [university]>
```

- Find the number of students who have been taught (at any time) by an instructor named 'Srinivasan'. Make sure you count a student only once even if the student has taken more than one course from Srinivasan.

**Query:** select count(distinct takes.ID)

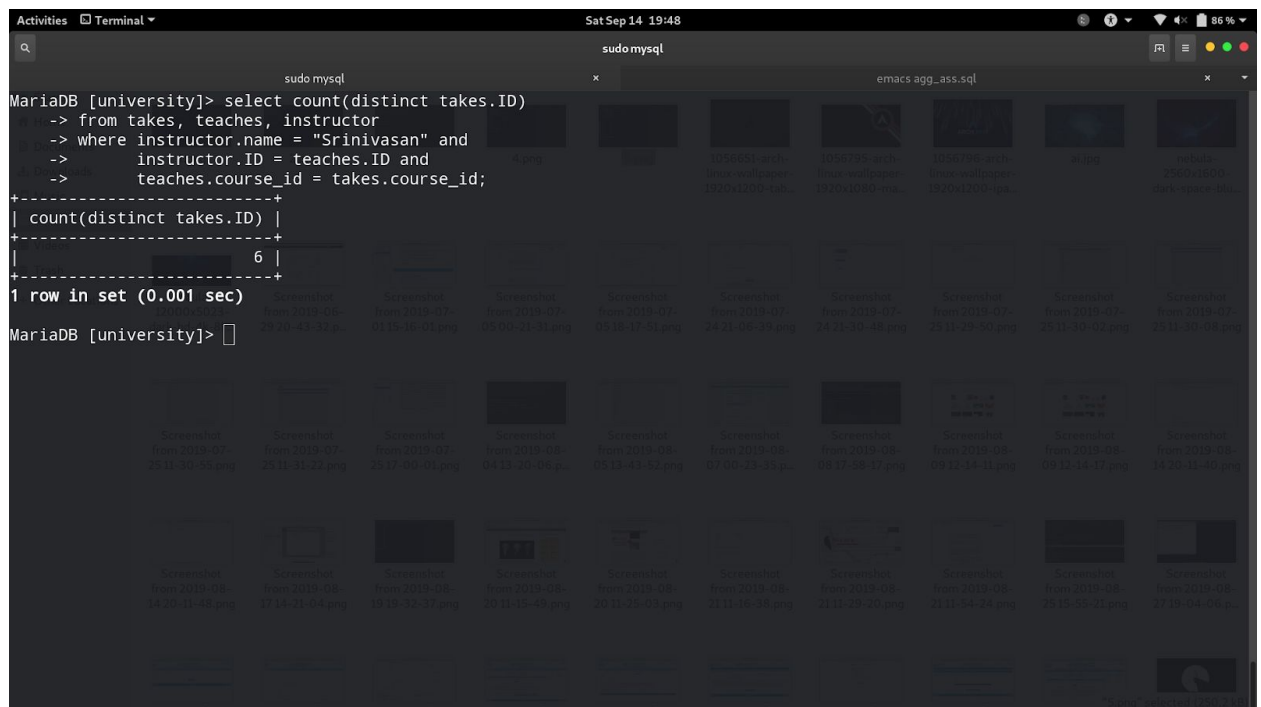
from takes, teaches, instructor

where instructor.name = "Srinivasan" and

instructor.ID = teaches.ID and

teaches.course\_id = takes.course\_id;

**Result:**



The screenshot shows a terminal window with the following content:

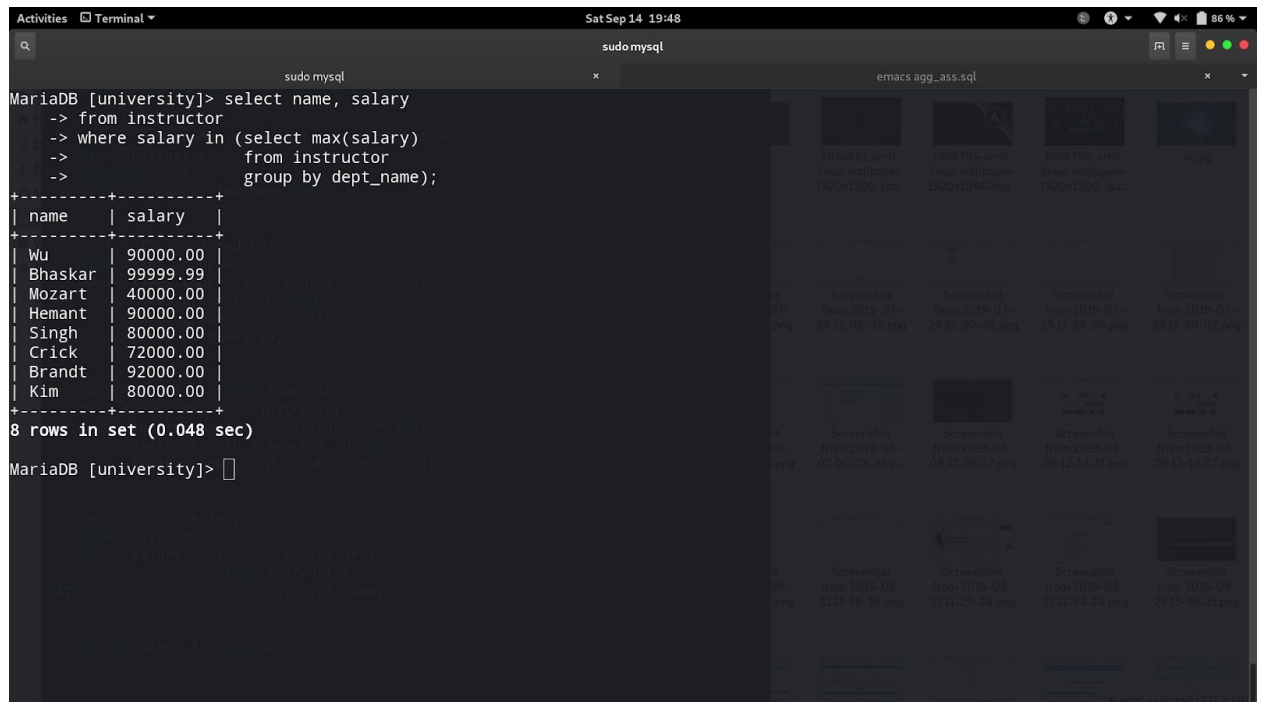
```
Sat Sep 14 19:48
sudo mysql
MariaDB [university]> select count(distinct takes.ID)
-> from takes, teaches, instructor
-> where instructor.name = "Srinivasan" and
-> instructor.ID = teaches.ID and
-> teaches.course_id = takes.course_id;
+-----+
| count(distinct takes.ID) |
+-----+
| 6 |
+-----+
1 row in set (0.001 sec)
MariaDB [university]>
```

The terminal window also shows a grid of thumbnail images in the background, likely from a file manager or a web browser.

- Find the name of all instructors who get the highest salary in their department.

**Query:** select name, salary  
from instructor  
where salary in (select max(salary)  
from instructor  
group by dept\_name);

**Result:**



The screenshot shows a terminal window with a MySQL prompt. The query executed is: `select name, salary from instructor where salary in (select max(salary) from instructor group by dept_name);`. The results are displayed in a table with two columns: name and salary. The results are: Wu (90000.00), Bhaskar (99999.99), Mozart (40000.00), Hemant (90000.00), Singh (80000.00), Crick (72000.00), Brandt (92000.00), and Kim (80000.00). The terminal also shows the status '8 rows in set (0.048 sec)'.

```
MariaDB [university]> select name, salary
-> from instructor
-> where salary in (select max(salary)
->                  from instructor
->                  group by dept_name);
+-----+-----+
| name  | salary |
+-----+-----+
| Wu    | 90000.00 |
| Bhaskar | 99999.99 |
| Mozart | 40000.00 |
| Hemant | 90000.00 |
| Singh | 80000.00 |
| Crick  | 72000.00 |
| Brandt | 92000.00 |
| Kim    | 80000.00 |
+-----+-----+
8 rows in set (0.048 sec)

MariaDB [university]>
```



- Find all students who have taken all courses taken by instructor 'Srinivasan'. (This is the division operation of relational algebra.) You can implement it by counting the number of courses taught by Srinivasan, and for each student (i.e. group by student), find the number of courses taken by that student, which were taught by Srinivasan. Make sure to count each course ID only once.

**Query:** select takes.ID from takes

except

select students\_not\_in\_all\_srinivasan.ID from

(select student.ID, srinivasan\_courses.course\_id

from student, (select teaches.course\_id

from teaches, instructor

where instructor.name = "Srinivasan" and

instructor.ID = teaches.ID) as srinivasan\_courses

except

select takes.ID, takes.course\_id

from takes) as students\_not\_in\_all\_srinivasan;

**Result:**

The screenshot shows a terminal window with a MySQL prompt. The query entered is a complex SQL statement using the EXCEPT operator to find students who have taken all courses taught by the instructor 'Srinivasan'. The result shows a single row with the student ID 12345.

```

MariaDB [university]> select takes.ID from takes
-> except
-> select students_not_in_all_srinivasan.ID from (select student.ID, srinivasan_courses.course_id
-> from student, (select teaches.course_id
-> from teaches, instructor
-> where instructor.name = "Srinivasan" and
-> instructor.ID = teaches.ID) as srinivasan_courses
-> except
-> select takes.ID, takes.course_id
-> from takes) as students_not_in_all_srinivasan;
+-----+
| ID |
+-----+
| 12345 |
+-----+
1 row in set (0.002 sec)

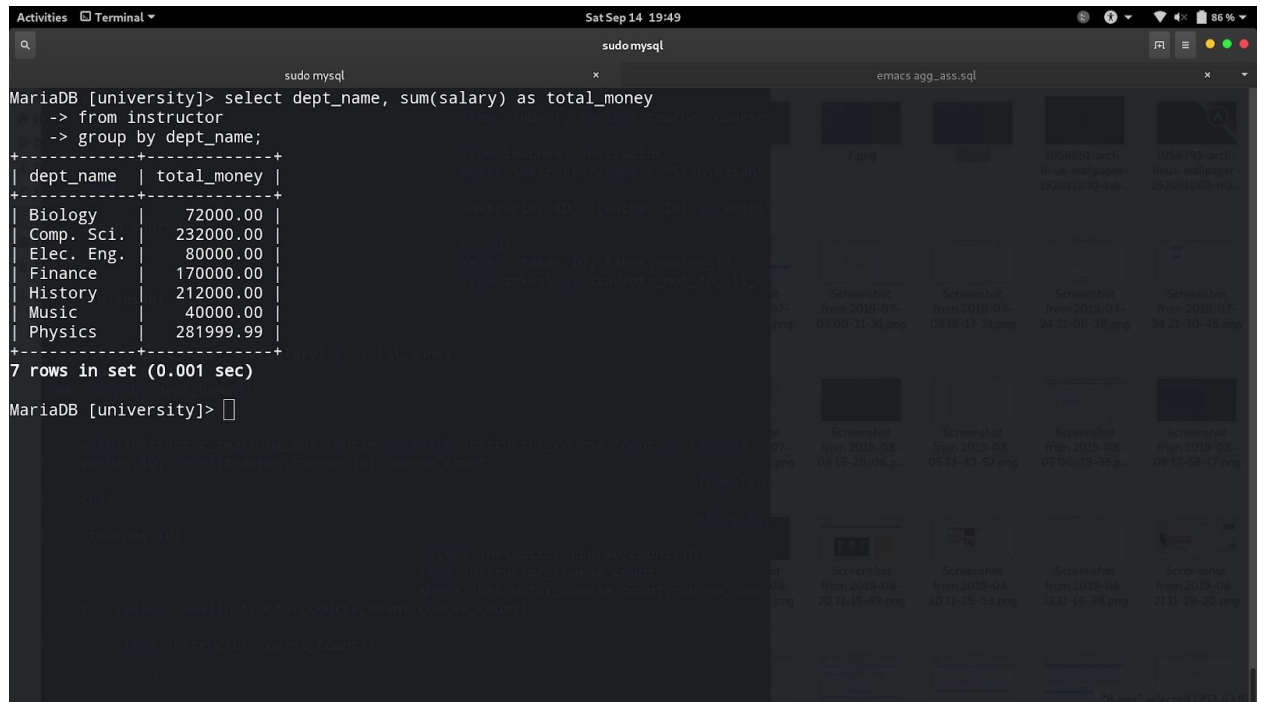
MariaDB [university]> 

```

- Find the total money spent by each department for salaries of instructors of that department.

**Query:** select dept\_name, sum(salary) as total\_money  
from instructor  
group by dept\_name;

**Result:**



The screenshot shows a terminal window with the following content:

```
Activities Terminal Sat Sep 14 19:49
sudo mysql
MariaDB [university]> select dept_name, sum(salary) as total_money
-> from instructor
-> group by dept_name;
+-----+-----+
| dept_name | total_money |
+-----+-----+
| Biology   | 72000.00    |
| Comp. Sci. | 232000.00   |
| Elec. Eng. | 80000.00    |
| Finance   | 170000.00   |
| History   | 212000.00   |
| Music     | 40000.00    |
| Physics   | 281999.99   |
+-----+-----+
7 rows in set (0.001 sec)

MariaDB [university]>
```

The terminal window also shows several open files in the background, including '7.png', '1056631-arch...', '1056795-arch...', 'Screenshot from 2019-07-07-05:00:21.png', 'Screenshot from 2019-07-07-05:18:17.png', 'Screenshot from 2019-07-24-21:46:38.png', 'Screenshot from 2019-07-24-21:50:42.png', 'Screenshot from 2019-08-04-14:20:06.png', 'Screenshot from 2019-08-05-13:43:52.png', 'Screenshot from 2019-08-07-00:23:15.png', 'Screenshot from 2019-08-08-17:56:17.png', 'Screenshot from 2019-08-20-11:35:48.png', 'Screenshot from 2019-08-20-11:25:03.png', 'Screenshot from 2019-08-22-11:46:08.png', and 'Screenshot from 2019-08-21-11:29:00.png'.

- Find the names of all students whose advisor has taught the maximum number of courses (multiple offerings of a course count as only 1).

(Note: this is a complex query, break it into parts by using the with clause.)

### Query:

```
with instructor_teaching_max_course as (
    with instructor_course_count as (
        select teaches.ID, count(teaches.course_id) course_count
        from teaches
        group by teaches.ID)
    select instructor_course_count.ID
    from instructor_course_count
    where instructor_course_count.course_count = (
        select max(instructor_course_count.course_count)
        from instructor_course_count))

select student.name
from advisor, instructor_teaching_max_course, student
where advisor.i_ID = instructor_teaching_max_course.ID and student.ID = advisor.s_ID;
```

### Result:

```

Activities Terminal Sat Sep 14 20:25
sudo mysql
MariaDB [university]> with instructor_teaching_max_course as (
->   with instructor_course_count as (
->     select teaches.ID, count(teaches.course_id) course_count
->     from teaches
->     group by teaches.ID)
->   select instructor_course_count.ID
->   from instructor_course_count
->   where instructor_course_count.course_count = (
->     select max(instructor_course_count.course_count)
->     from instructor_course_count))
-> select student.name
-> from advisor, instructor_teaching_max_course, student
-> where advisor.i_ID = instructor_teaching_max_course.ID and student.ID = advisor.s_ID;
+-----+
| name |
+-----+
| Shankar |
+-----+
1 row in set (0.001 sec)
MariaDB [university]>

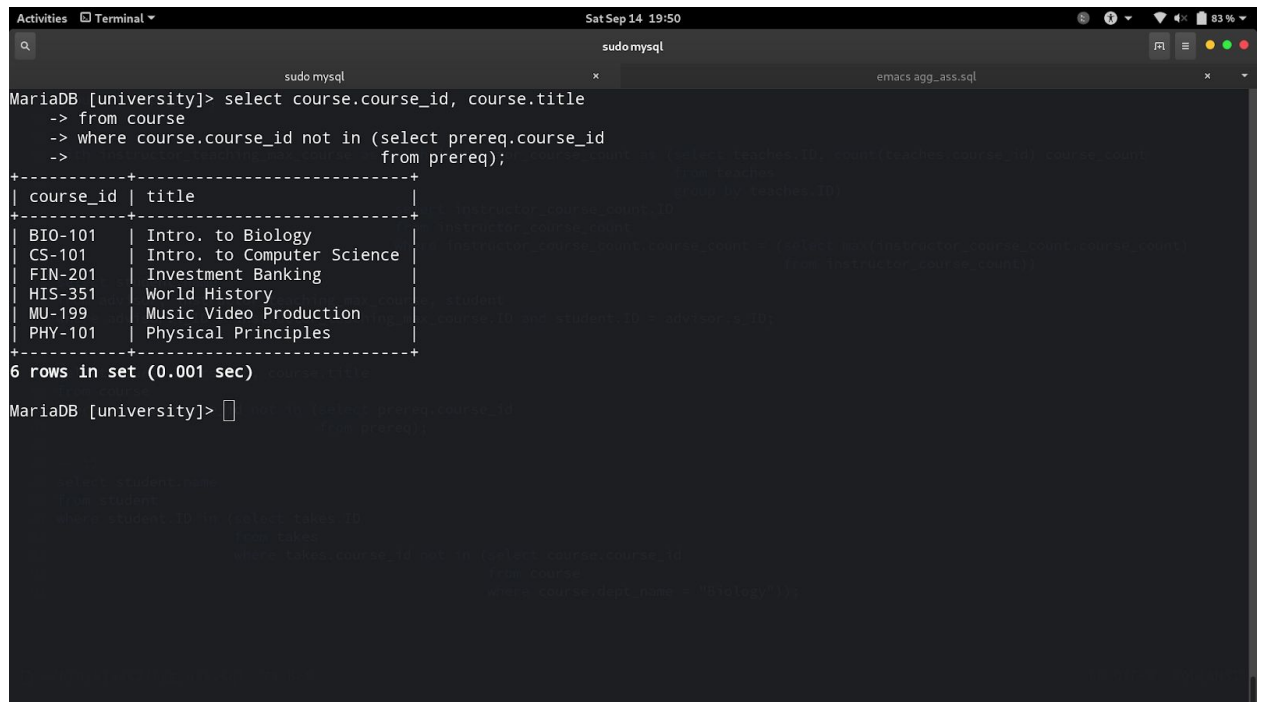
```

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

### Query:

```
select course.course_id, course.title
from course
where course.course_id not in (select prereq.course_id
                               from prereq);
```

### Result:



The screenshot shows a terminal window with a MySQL prompt. The query executed is: `select course.course_id, course.title from course where course.course_id not in (select prereq.course_id from prereq);`. The results are displayed in a table with 6 rows. The first row is the header: `course_id | title`. The subsequent rows are: `BIO-101 | Intro. to Biology`, `CS-101 | Intro. to Computer Science`, `FIN-201 | Investment Banking`, `HIS-351 | World History`, `MU-199 | Music Video Production`, and `PHY-101 | Physical Principles`. The terminal also shows the status `6 rows in set (0.001 sec)`.

```
Activities Terminal Sat Sep 14 19:50
sudo mysql
MariaDB [university]> select course.course_id, course.title
-> from course
-> where course.course_id not in (select prereq.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 |
+-----+-----+
6 rows in set (0.001 sec)
MariaDB [university]>
```

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

### Query:

```
select student.name
from student
where student.ID in (select takes.ID
                     from takes
                     where takes.course_id not in (select course.course_id
                                                  from course
                                                  where course.dept_name = "Biology"));
```

### Result:

```

Activities  Terminal
Sat Sep 14 19:51
sudo mysql
MariaDB [university]> select student.name
-> from student
-> where student.ID in (select takes.ID
-> from takes
-> where takes.course_id not in (select course.course_id
-> from course
-> where course.dept_name = "Biology"));
+-----+
| name |
+-----+
| Zhang |
| Shankar |
| Brandt |
| Chavez |
| Peltier |
| Levy |
| Williams |
| Sanchez |
| Brown |
| Aoi |
| Bourikas |
+-----+
11 rows in set (0.001 sec)

MariaDB [university]>

```