CSLR51 – Database Management Systems Laboratory

#Session: 06 || Date: 05/09/2024

Viva Due: 1. Queries a – i (05/09/2024) and j - w (12/09/2024)
Moodle Due: 10/09/2024 at 11 PM

1. Relational Database Design – University Schema

a. Find the titles of courses in the CSE department that have 3 credits.

```
mysql> select title from course where dept_name='CSE' and credits=3;
+----------------+
| title          |
+----------------+
| Data structure |
+----------------+
1 row in set (0.00 sec)
```

b. Find the highest salary of any professor.

```
mysql> select max(salary) from professor;
+-------------+
| max(salary) |
+-------------+
|       70000 |
+-------------+
1 row in set (0.01 sec)
```

c. Find all professors earning the highest salary (there may be more than one with the same salary).

```
mysql> select name from professor where salary=(select max(salary) from professor);
+--------+
| name   |
+--------+
| Kiran  |
| Kadhir |
+--------+
2 rows in set (0.00 sec)
```

d. Find the maximum enrollment, across all sections, in Fall 2020.

```
mysql> select sec_id,count(*) from student natural join takes where year=2020 and
semester='fall' g
roup by sec_id;
+--------+----------+
| sec_id | count(*) |
+--------+----------+
| CSEA   |        2 |
| CSEB   |        1 |
+--------+----------+
2 rows in set (0.00 sec)
mysql> select sec_id,count(*) as Enrollment from student natural join takes where year=2020
and semester='fall' group by sec_id having count(*)=(select count(*) from student natural join
takes where year=2020 and semester='fall' group by sec_id limit 1);
+--------+------------+
| sec_id | Enrollment |
+--------+------------+
| CSEA   |          2 |
+--------+------------+
1 row in set (0.00 sec)
```

e. Find the enrollment of each section that was offered in Spring 2019.

```
mysql> select sec_id,count(*) from takes natural join course where semester='spring' and
year=2
019  group by sec_id;
+--------+----------+
| sec_id | count(*) |
+--------+----------+
| CSEA   |        1 |
+--------+----------+
1 row in set (0.01 sec)
```

f. Find the IDs and names of all students who have not taken any course offering before Spring
2013.

```
mysql> select sID,name from student where sID not in(select sID from takes where year<2013)
and sID not in(select sID from takes where year=2013 and semester='autumn');
+-----+-------+
| sID | name  |
+-----+-------+
| 1C  | Anu   |
| 1E  | Lal   |
| 2I  | Sunny |
+-----+-------+
3 rows in set (0.00 sec)
```

g. Find the lowest, across all departments, of the per-department maximum salary computed
by the preceding query.

```
mysql> select min(sal) from (select max(salary) as sal,dept_name from professor group by
dept_n
ame) AS A;
+----------+
| min(sal) |
+----------+
|    60000 |
+----------+
1 row in set (0.00 sec)
```

h. Create a new course “CS-001”, titled “Weekly Seminar”, with 1 credit.

```
mysql> insert into course values('CS-001','Weekly Seminar','CSE',1);
Query OK, 1 row affected (0.01 sec)
```

i. Delete the course CS-001. What will happen if you run this delete statement without first deleting offerings (sections) of this course.

```
mysql> delete from course where course_id='CS_001';
Query OK, 0 rows affected (0.00 sec)
```

If this record is deleted then the course begin offered to a section becomes meaningless.
So this show lack of integrity constraints in our relations.

j. Display the list of all course sections offered in Spring 2022, along with the names of the professors teaching the section. If a section has more than one professor, it should appear as many times in the result as it has professor. If it does not have any professors, it should still appear in the result with the professor name set to “-”.

```
mysql> select
course_id,sec_id,semester,year,building,room_number,time_slot_id,p,IFNULL(name,'-') as
p_name from ((select
section.course_id,section.sec_id,section.semester,section.year,section.building,section.room_n
umber,section.time_slot_id,IFNULL(pID,'-') as p from (section left outer join teaches on
section.course_id=teaches.course_id and section.sec_id=teaches.sec_id and
section.year=teaches.year and section.semester=teaches.semester) where section.year=2022
and section.semester='spring') B left outer join professor on p=Professor.pID) ;
+-----------+--------+----------+------+----------+-------------+--------------+----+--------+
| course_id | sec_id | semester | year | building | room_number | time_slot_id | p  | p_name |
+-----------+--------+----------+------+----------+-------------+--------------+----+--------+
| CS1       | CSEA   | spring   | 2022 | logos    |          10 | slot1        | C1 | Kiran  |
| CS1       | CSEB   | spring   | 2022 | orion    |          16 | slot1        | -  | -      |
| CS2       | CSEB   | spring   | 2022 | orion    |          15 | slot2        | C2 | Suman  |
+-----------+--------+----------+------+----------+-------------+--------------+----+--------+
3 rows in set (0.00 sec)
```

k. Find the professor ID, name, dept name, and salary for professors whose salary is greater than 50,000.

```
mysql> select * from professor where salary>50000;
+-----+--------+-----------+--------+
| pID | name   | dept_name | salary |
+-----+--------+-----------+--------+
| C1  | Kiran  | CSE       |  70000 |
| C2  | Suman  | CSE       |  51000 |
| E3  | Karthi | ECE       |  60000 |
| I4  | Kadhir | ICE       |  70000 |
+-----+--------+-----------+--------+
4 rows in set (0.00 sec)
```

l. Find the names of all professors in the Chemical Engineering department together with the course id of all courses they teach.

```
mysql> select name,course_id from professor natural join teaches where dept_name='CHEM';
+--------+-----------+
| name   | course_id |
+--------+-----------+
| Shanti | CH1       |
| Arun   | CH1       |
| Arun   | CH2       |
+--------+-----------+
3 rows in set (0.00 sec)
```

m. Find the set of all courses taught in the Fall 2021 semester, the Spring 2021 semester, or both.

```
mysql> select distinct(course_id) from ((section natural join teaches) natural join takes) where
year=2021;
+-----------+
| course_id |
+-----------+
| CH1       |
| CS3       |
+-----------+
2 rows in set (0.00 sec)
```

n. Find the names of all professors whose department is in the ‘ORION’ building.

```
mysql> select name from department natural join professor where building='orion';
+--------+
| name   |
+--------+
| Shanti |
| Arun   |
+--------+
2 rows in set (0.00 sec)
```

o. Find the set of all courses taught in the Fall 2023 semester, or in the Spring 2022 semester, or both.

```
mysql> select * from ((section natural join teaches) natural join takes) where (semester='fall' and year=2023) or (semester='spring' and year=2022);
+-----------+--------+----------+------+----------+-------------+--------------+-----+-----+--------+
| course_id | sec_id | semester | year | building | room_number | time_slot_id | pID | sID | credit |
+-----------+--------+----------+------+----------+-------------+--------------+-----+-----+--------+
| CH1       | CHEMA  | spring   | 2022 | logos    |          10 | slot1        | Ch1 | 1CH |      3 |
| CS3       | CSEB   | fall     | 2023 | orion    |          15 | slot2        | C1  | 2C  |      3 |
+-----------+--------+----------+------+----------+-------------+--------------+-----+-----+--------+
2 rows in set (0.00 sec)
```

p. Find the set of all courses taught in the Fall 2021 semester, but not in the Spring 2019 semester.

```
mysql> select * from ((section natural join teaches) natural join takes) where (semester='fall' and year=2021) and course_id not in(select course_id from ((section natural join teaches) natural join takes) where semester='spring' and year=2019);
+-----------+--------+----------+------+----------+-------------+--------------+-----+-----+--------+
| course_id | sec_id | semester | year | building | room_number | time_slot_id | pID | sID | credit |
+-----------+--------+----------+------+----------+-------------+--------------+-----+-----+--------+
| CS3       | CSEB   | fall     | 2021 | orion    |          15 | slot2        | C1  | 2C  |      3 |
+-----------+--------+----------+------+----------+-------------+--------------+-----+-----+--------+
1 row in set (0.00 sec)
```

q. Find the IDs of all students who were taught by an professor named Tejaswi; make sure there are no duplicates in the result.

```
mysql> select * from (teaches natural join takes)natural join student where pID in (select pID from professor where name='Tejaswi');
+-----+-----------+--------+----------+------+-----+--------+---------+-----------+------------+
| sID | course_id | sec_id | semester | year | pID | credit | name    | dept_name | tot_credit |
+-----+-----------+--------+----------+------+-----+--------+---------+-----------+------------+
| 1CH | CH1       | CHEMA  | spring   | 2021 | Ch1 |      3 | Divakar | CHEM      |         67 |
+-----+-----------+--------+----------+------+-----+--------+---------+-----------+------------+
1 row in set (0.00 sec)
mysql> select distinct(sID) from (teaches natural join takes)natural join student where pID in (select pID from professo
r where name='Tejaswi');
+-----+
| sID |
+-----+
| 1CH |
+-----+
1 row in set (0.00 sec)
```

r. Find the names of all students who have taken at least one Computer Science course; make sure there are no duplicate names in the result.

```
mysql> select distinct(name) from student where sID in ( select sID from takes where course_id
like'CS_');
+-------+
| name  |
+-------+
| Anu   |
| Raj   |
| Daisy |
+-------+
3 rows in set (0.00 sec)
```

s. For each department, find the maximum salary of professors in that department. You may assume that every department has at least one professor.

```
mysql> select max(salary),dept_name from professor group by dept_name;
+-------------+-----------+
| max(salary) | dept_name |
+-------------+-----------+
|       77000 | CHEM      |
|       70000 | CSE       |
|       60000 | ECE       |
|       70000 | ICE       |
+-------------+-----------+
4 rows in set (0.01 sec)
```

t. Display a list of all professors, showing their ID, name, and the number of sections that they
have taught. Make sure to show the number of sections as 0 for professors who have not taught any section. Your query should use an outerjoin, and should not use scalar subqueries.

```
mysql> select name,professor.pid,IFNULL(A,0) from professor left outer join (select pID as
p,count(distinct(sec_id)) as A from teaches group by p) B on
p=professor.pID;
+---------+-----+-------------+
| name    | pid | IFNULL(A,0) |
+---------+-----+-------------+
| Kiran   | C1  |           2 |
| Suman   | C2  |           1 |
| Tejaswi | Ch1 |           1 |
| Arun    | Ch2 |           2 |
| Karthi  | E3  |           0 |
| Kadhir  | I4  |           0 |
+---------+-----+-------------+
6 rows in set (0.00 sec)
```

u. Write the same query as above, but using a scalar subquery, without outerjoin.

```
mysql> select proff.name,proff.pID,IFNULL((select IFNULL(count,0) from (select
count(distinct(a.sec_id)) as count,a.pID,(select b.name from professor b wher
e b.pID=a.pID) as name1 from teaches a group by a.pID) B where proff.name=name1),0) as
section_count from professor proff;
+---------+-----+---------------+
| name    | pID | section_count |
+---------+-----+---------------+
| Kiran   | C1  |             2 |
| Suman   | C2  |             1 |
| Tejaswi | Ch1 |             1 |
| Arun    | Ch2 |             2 |
| Karthi  | E3  |             0 |
| Kadhir  | I4  |             0 |
+---------+-----+---------------+
6 rows in set (0.00 sec)
```

v. Find all students who have taken all courses offered in the Biology department.

```
mysql> select name from student where sID in ( select distinct(T1.sID) from (select * from takes
where course_id like 'BIO_') T1 where NOT EXISTS( ( select course_id from course where
dept_name='BIO') except (select T2.course_id from takes T2 where T2.sID =T1.sID and
course_id like 'BIO_')));
+-------+
| name  |
+-------+
| Sunny |
+-------+
1 row in set (0.00 sec)
```

w. Create your own query: define what you want to do in English, then write the query in SQL. Make it as difficult as you wish, the harder the better.

For each department, find the maximum salary of professors in that department along with the professor name. You may assume that every department has at least one professor.

```
mysql> select name,dept_name,salary from professor Q natural join department D where
salary=(select max(salary) from professor P where P.dept_name=Q.dept_name group by
P.dept_name);
+--------+-----------+--------+
| name   | dept_name | salary |
+--------+-----------+--------+
| Arun   | CHEM      |  77000 |
| Kiran  | CSE       |  70000 |
| Karthi | ECE       |  60000 |
| Kadhir | ICE       |  70000 |
+--------+-----------+--------+
4 rows in set (0.00 sec)
```

x. Use the DCL commands to perform the following operations.
i. Create a new user ‘testuser’ on the localhost.
ii. Grant all privileges for the testuser on the University database you have created.
iii. Revoke all the privileges given to testuser.

```
mysql> CREATE USER 'testuser'@'localhost' IDENTIFIED BY 'house000y';
Query OK, 0 rows affected (0.15 sec)

mysql> GRANT ALL PRIVILEGES ON University.* TO 'testuser'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql> system mysql -u testuser -p
Enter password: *********

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| performance_schema |
| university         |
+--------------------+
3 rows in set (0.01 sec)

mysql> REVOKE ALL PRIVILEGES ON University.* FROM 'testuser'@'localhost';
Query OK, 0 rows affected (0.02 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| performance_schema |
+--------------------+
2 rows in set (0.01 sec)

mysql> drop user 'testuser'@'localhost';
```

```
Query OK, 0 rows affected (0.01 sec)
```

y. Use the DCL command to revoke privilege to the user.
i. Create a new user ‘testuser1’ on the localhost.
ii. Grant only select privileges for the testuser1 on the Student table.
iii. Revoke the select privileges for the testuser1 on the Student table.

```
mysql> CREATE USER 'testuser1'@'localhost' IDENTIFIED BY 'house000y';
Query OK, 0 rows affected (0.02 sec)

mysql> GRANT INSERT ON University.Student TO 'testuser1'@'localhost';
Query OK, 0 rows affected (0.02 sec)

mysql> GRANT DELETE ON University.Student TO 'testuser1'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql> REVOKE INSERT ON University.Student FROM 'testuser1'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql> INSERT INTO STUDENT VALUES('EC5','1E','ECEA','spring',2022,4);
ERROR 1142 (42000): INSERT command denied to user 'testuser1'@'localhost' for table
'student'
```

University Schema at the end:

```
mysql> select * from classroom;
+----------+-------------+----------+
| building | room_number | capacity |
+----------+-------------+----------+
| logos    |          10 |       50 |
| logos    |          11 |       50 |
| orion    |          15 |       70 |
| orion    |          16 |       70 |
+----------+-------------+----------+
4 rows in set (0.06 sec)
```

```
mysql> select  * from department;
+-----------+----------+---------+
| dept_name | building | budget  |
+-----------+----------+---------+
| BIO       | block3   | 1600000 |
| CHEM      | orion    | 1000000 |
| CSE       | block2   | 1500000 |
| ECE       | block3   | 2000000 |
| ICE       | block4   | 1800000 |
+-----------+----------+---------+
5 rows in set (0.02 sec)

mysql> select * from course;
+-----------+-------------------+-----------+---------+
| course_id | title             | dept_name | credits |
+-----------+-------------------+-----------+---------+
| BIO1      | Botany            | BIO       |       4 |
| BIO2      | Zoology           | BIO       |       3 |
| CH1       | Chemical analysis | CHEM      |       3 |
| CH2       | Equilibrium       | CHEM      |       2 |
| CS-001    | Weekly Seminar    | CSE       |       1 |
| CS1       | CS Essential      | CSE       |       4 |
| CS2       | Algorithm         | CSE       |       4 |
| CS3       | Data structure    | CSE       |       3 |
| EC4       | Circuit Theory    | ECE       |       4 |
| IC4       | Thermodynamics    | ICE       |       3 |
+-----------+-------------------+-----------+---------+
10 rows in set (0.02 sec)

mysql> select * from professor;
+-----+---------+-----------+--------+
| pID | name    | dept_name | salary |
+-----+---------+-----------+--------+
| C1  | Kiran   | CSE       |  70000 |
| C2  | Suman   | CSE       |  51000 |
| Ch1 | Tejaswi | CHEM      |  76000 |
| Ch2 | Arun    | CHEM      |  77000 |
| E3  | Karthi  | ECE       |  60000 |
| I4  | Kadhir  | ICE       |  70000 |
+-----+---------+-----------+--------+
6 rows in set (0.02 sec)
```

```
mysql> select * from section;
+-----------+--------+----------+------+----------+-------------+--------------+
| course_id | sec_id | semester | year | building | room_number | time_slot_id |
+-----------+--------+----------+------+----------+-------------+--------------+
| CH1       | CHEMA  | spring   | 2021 | logos    |          10 | slot1        |
| CS1       | CSEA   | spring   | 2022 | logos    |          10 | slot1        |
| CS1       | CSEB   | spring   | 2022 | orion    |          16 | slot1        |
| CS2       | CSEA   | fall     | 2020 | logos    |          11 | slot2        |
| CS2       | CSEB   | spring   | 2022 | orion    |          15 | slot2        |
| CS3       | CSEB   | fall     | 2019 | orion    |          15 | slot2        |
+-----------+--------+----------+------+----------+-------------+--------------+
6 rows in set (0.02 sec)

mysql> select * from teaches;
+-----+-----------+--------+----------+------+
| pID | course_id | sec_id | semester | year |
+-----+-----------+--------+----------+------+
| C1  | CS1       | CSEA   | spring   | 2022 |
| C1  | CS3       | CSEB   | fall     | 2019 |
| C2  | CS2       | CSEB   | spring   | 2022 |
| Ch1 | CH1       | CHEMA  | spring   | 2021 |
| Ch1 | CH2       | CHEMA  | fall     | 2023 |
| Ch2 | CH1       | CHEMB  | spring   | 2022 |
| Ch2 | CH2       | CHEMA  | fall     | 2022 |
+-----+-----------+--------+----------+------+
7 rows in set (0.02 sec)

mysql> select * from student;
+-----+---------+-----------+------------+
| sID | name    | dept_name | tot_credit |
+-----+---------+-----------+------------+
| 1C  | Anu     | CSE       |         50 |
| 1CH | Divakar | CHEM      |         67 |
| 1E  | Lal     | ECE       |         80 |
| 2C  | Raj     | CSE       |         70 |
| 2CH | Daisy   | CHEM      |         60 |
| 2I  | Sunny   | ICE       |         50 |
+-----+---------+-----------+------------+
6 rows in set (0.02 sec)
```

```
mysql> select * from takes;
+-----+-----------+--------+----------+------+--------+
| sID | course_id | sec_id | semester | year | credit |
+-----+-----------+--------+----------+------+--------+
| 1C  | CS1       | CSEA   | spring   | 2019 |      4 |
| 1C  | CS2       | CSEA   | fall     | 2020 |      4 |
| 1C  | CS3       | CSEA   | fall     | 2021 |      4 |
| 1CH | CH1       | CHEMA  | spring   | 2021 |      3 |
| 1CH | CH2       | CHEMA  | fall     | 2022 |      2 |
| 1E  | BIO2      | ECEB   | spring   | 2018 |      3 |
| 2C  | CS2       | CSEB   | spring   | 2012 |      4 |
| 2C  | CS3       | CSEB   | fall     | 2019 |      3 |
| 2CH | CH1       | CHEMB  | spring   | 2022 |      3 |
| 2CH | CS1       | CHEMB  | spring   | 2022 |      4 |
| 2I  | BIO1      | ICEA   | fall     | 2017 |      4 |
| 2I  | BIO2      | ICEA   | spring   | 2017 |      3 |
+-----+-----------+--------+----------+------+--------+
12 rows in set (0.02 sec)

mysql> select * from guide;
+-----+-----+
| sID | pID |
+-----+-----+
| 2C  | C1  |
| 1C  | C2  |
+-----+-----+
2 rows in set (0.02 sec)

mysql> select * from time_slot;
+--------------+--------+------------+----------+
| time_slot_id | day    | start_time | end_time |
+--------------+--------+------------+----------+
| slot1        | Monday | 09:20:00   | 10:10:00 |
| slot12       | Monday | 05:00:00   | 05:50:00 |
+--------------+--------+------------+----------+
2 rows in set (0.02 sec)

mysql> select * from prereq;
+-----------+----------+
| course_id | prere_id |
+-----------+----------+
| CS2       | CS1      |
| CS2       | CS3      |
+-----------+----------+
2 rows in set (0.02 sec)
```