DBMS LAB-11

NAME:- ABHISHIKTH BODA

ROLL NUMBER:- S20210010044

DATE:- 16-11-2022

TABLES:-

Classroom:-

```
mysql> select * from classroom;
+----------+-------------+----------+
| building | room_number | capacity |
+----------+-------------+----------+
| Packard  | 101         |      500 |
| Painter  | 514         |       10 |
| Taylor   | 3128        |       70 |
| Watson   | 100         |       30 |
| Watson   | 120         |       50 |
+----------+-------------+----------+
5 rows in set (0.00 sec)
```

Department:-

```
mysql> select * from department;
+------------+----------+-----------+
| dept_name  | building | budget    |
+------------+----------+-----------+
| Biology    | Watson   |  90000.00 |
| Comp. Sci. | Taylor   | 100000.00 |
| Elec. Eng. | Taylor   |  85000.00 |
| Finance    | Painter  | 120000.00 |
| History    | Painter  |  50000.00 |
| Music      | Packard  |  80000.00 |
| Physics    | Watson   |  70000.00 |
+------------+----------+-----------+
7 rows in set (0.00 sec)
```

Course:-

```
mysql> select * from course;
+-----------+----------------------------+------------+---------+
| course_id | title                      | dept_name  | credits |
+-----------+----------------------------+------------+---------+
| BIO-101   | Intro. to Biology          | Biology    |       4 |
| BIO-301   | Genetics                   | Biology    |       4 |
| BIO-399   | Computational Biology      | Biology    |       3 |
| CS-101    | Intro. to Computer Science | Comp. Sci. |       4 |
| CS-190    | Game Design                | Comp. Sci. |       4 |
| CS-315    | Robotics                   | Comp. Sci. |       3 |
| CS-319    | Image Processing           | Comp. Sci. |       3 |
| CS-347    | Database System Concepts   | Comp. Sci. |       3 |
| EE-181    | Intro. to Digital Systems  | Elec. Eng. |       3 |
| FIN-201   | Investment Banking         | Finance    |       3 |
| HIS-351   | World History              | History    |       3 |
| MU-199    | Music Video Production     | Music      |       3 |
| PHY-101   | Physical Principles        | Physics    |       4 |
+-----------+----------------------------+------------+---------+
13 rows in set (0.00 sec)
```

Instructor:-

```
mysql> select * from instructor;
+-------+------------+------------+----------+
| ID    | name       | dept_name  | salary   |
+-------+------------+------------+----------+
| 10101 | Srinivasan | Comp. Sci. | 65000.00 |
| 12121 | Wu         | Finance    | 90000.00 |
| 15151 | Mozart     | Music      | 40000.00 |
| 22222 | Einstein   | Physics    | 95000.00 |
| 32343 | El Said    | History    | 60000.00 |
| 33456 | Gold       | Physics    | 87000.00 |
| 45565 | Katz       | Comp. Sci. | 75000.00 |
| 58583 | Califieri  | History    | 62000.00 |
| 76543 | Singh      | Finance    | 80000.00 |
| 76766 | Crick      | Biology    | 72000.00 |
| 83821 | Brandt     | Comp. Sci. | 92000.00 |
| 98345 | Kim        | Elec. Eng. | 80000.00 |
+-------+------------+------------+----------+
12 rows in set (0.00 sec)
```

Section:-

```
mysql> select * from section;
+-----------+--------+----------+------+----------+-------------+--------------+
| course_id | sec_id | semester | year | building | room_number | time_slot_id |
+-----------+--------+----------+------+----------+-------------+--------------+
| BIO-101   | 1      | Summer   | 2009 | Painter  | 514         | B            |
| BIO-301   | 1      | Summer   | 2010 | Painter  | 514         | A            |
| CS-101    | 1      | Fall     | 2009 | Packard  | 101         | H            |
| CS-101    | 1      | Spring   | 2010 | Packard  | 101         | F            |
| CS-190    | 1      | Spring   | 2009 | Taylor   | 3128        | E            |
| CS-190    | 2      | Spring   | 2009 | Taylor   | 3128        | A            |
| CS-315    | 1      | Spring   | 2010 | Watson   | 120         | D            |
| CS-319    | 1      | Spring   | 2010 | Watson   | 100         | B            |
| CS-319    | 2      | Spring   | 2010 | Taylor   | 3128        | C            |
| CS-347    | 1      | Fall     | 2009 | Taylor   | 3128        | A            |
| EE-181    | 1      | Spring   | 2009 | Taylor   | 3128        | C            |
| FIN-201   | 1      | Spring   | 2010 | Packard  | 101         | B            |
| HIS-351   | 1      | Spring   | 2010 | Painter  | 514         | C            |
| MU-199    | 1      | Spring   | 2010 | Packard  | 101         | D            |
| PHY-101   | 1      | Fall     | 2009 | Watson   | 100         | A            |
+-----------+--------+----------+------+----------+-------------+--------------+
15 rows in set (0.00 sec)
```

Teaches:-

```
mysql> select * from teaches;
+-------+-----------+--------+----------+------+
| ID    | course_id | sec_id | semester | year |
+-------+-----------+--------+----------+------+
| 76766 | BIO-101   | 1      | Summer   | 2009 |
| 76766 | BIO-301   | 1      | Summer   | 2010 |
| 10101 | CS-101    | 1      | Fall     | 2009 |
| 45565 | CS-101    | 1      | Spring   | 2010 |
| 83821 | CS-190    | 1      | Spring   | 2009 |
| 83821 | CS-190    | 2      | Spring   | 2009 |
| 10101 | CS-315    | 1      | Spring   | 2010 |
| 45565 | CS-319    | 1      | Spring   | 2010 |
| 83821 | CS-319    | 2      | Spring   | 2010 |
| 10101 | CS-347    | 1      | Fall     | 2009 |
| 98345 | EE-181    | 1      | Spring   | 2009 |
| 12121 | FIN-201   | 1      | Spring   | 2010 |
| 32343 | HIS-351   | 1      | Spring   | 2010 |
| 15151 | MU-199    | 1      | Spring   | 2010 |
| 22222 | PHY-101   | 1      | Fall     | 2009 |
+-------+-----------+--------+----------+------+
15 rows in set (0.00 sec)
```

Student:-

```
mysql> select * from student;
+-------+----------+------------+----------+
| ID    | name     | dept_name  | tot_cred |
+-------+----------+------------+----------+
| 00128 | Zhang    | Comp. Sci. |      102 |
| 12345 | Shankar  | Comp. Sci. |       32 |
| 19991 | Brandt   | History    |       80 |
| 23121 | Chavez   | Finance    |      110 |
| 44553 | Peltier  | Physics    |       56 |
| 45678 | Levy     | Physics    |       46 |
| 54321 | Williams | Comp. Sci. |       54 |
| 55739 | Sanchez  | Music      |       38 |
| 70557 | Snow     | Physics    |        0 |
| 76543 | Brown    | Comp. Sci. |       58 |
| 76653 | Aoi      | Elec. Eng. |       60 |
| 98765 | Bourikas | Elec. Eng. |       98 |
| 98988 | Tanaka   | Biology    |      120 |
+-------+----------+------------+----------+
13 rows in set (0.00 sec)
```

Takes:-

```
mysql> select * from takes;
+-------+-----------+--------+----------+------+-------+
| ID    | course_id | sec_id | semester | year | grade |
+-------+-----------+--------+----------+------+-------+
| 00128 | CS-101    | 1      | Fall     | 2009 | A     |
| 00128 | CS-347    | 1      | Fall     | 2009 | A-    |
| 12345 | CS-101    | 1      | Fall     | 2009 | C     |
| 12345 | CS-190    | 2      | Spring   | 2009 | A     |
| 12345 | CS-315    | 1      | Spring   | 2010 | A     |
| 12345 | CS-347    | 1      | Fall     | 2009 | A     |
| 19991 | HIS-351   | 1      | Spring   | 2010 | B     |
| 23121 | FIN-201   | 1      | Spring   | 2010 | C+    |
| 44553 | PHY-101   | 1      | Fall     | 2009 | B-    |
| 45678 | CS-101    | 1      | Fall     | 2009 | F     |
| 45678 | CS-101    | 1      | Spring   | 2010 | B+    |
| 45678 | CS-319    | 1      | Spring   | 2010 | B     |
| 54321 | CS-101    | 1      | Fall     | 2009 | A-    |
| 54321 | CS-190    | 2      | Spring   | 2009 | B+    |
| 55739 | MU-199    | 1      | Spring   | 2010 | A-    |
| 76543 | CS-101    | 1      | Fall     | 2009 | A     |
| 76543 | CS-319    | 2      | Spring   | 2010 | A     |
| 76653 | EE-181    | 1      | Spring   | 2009 | C     |
| 98765 | CS-101    | 1      | Fall     | 2009 | C-    |
| 98765 | CS-315    | 1      | Spring   | 2010 | B     |
| 98988 | BIO-101   | 1      | Summer   | 2009 | A     |
| 98988 | BIO-301   | 1      | Summer   | 2010 | NULL  |
+-------+-----------+--------+----------+------+-------+
22 rows in set (0.00 sec)
```

Advisor:-

```
mysql> select * from advisor;
+-------+-------+
| s_ID  | i_ID  |
+-------+-------+
| 12345 | 10101 |
| 44553 | 22222 |
| 45678 | 22222 |
| 00128 | 45565 |
| 76543 | 45565 |
| 23121 | 76543 |
| 98988 | 76766 |
| 76653 | 98345 |
| 98765 | 98345 |
+-------+-------+
9 rows in set (0.00 sec)
```

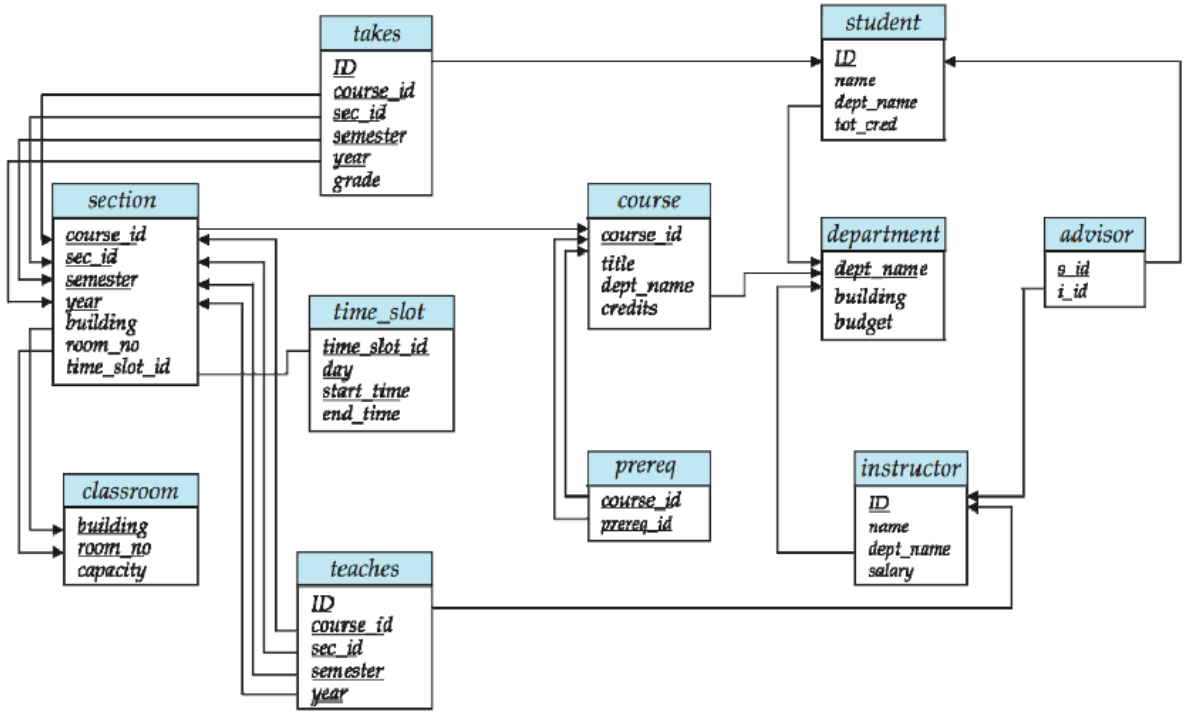
Time_slot:-

```
mysql> select * from time_slot;
+--------------+-----+----------+-----------+--------+---------+
| time_slot_id | day | start_hr | start_min | end_hr | end_min |
+--------------+-----+----------+-----------+--------+---------+
| A            | F   |        8 |         0 |      8 |      50 |
| A            | M   |        8 |         0 |      8 |      50 |
| A            | W   |        8 |         0 |      8 |      50 |
| B            | F   |        9 |         0 |      9 |      50 |
| B            | M   |        9 |         0 |      9 |      50 |
| B            | W   |        9 |         0 |      9 |      50 |
| C            | F   |       11 |         0 |     11 |      50 |
| C            | M   |       11 |         0 |     11 |      50 |
| C            | W   |       11 |         0 |     11 |      50 |
| D            | F   |       13 |         0 |     13 |      50 |
| D            | M   |       13 |         0 |     13 |      50 |
| D            | W   |       13 |         0 |     13 |      50 |
| E            | R   |       10 |        30 |     11 |      45 |
| E            | T   |       10 |        30 |     11 |      45 |
| F            | R   |       14 |        30 |     15 |      45 |
| F            | T   |       14 |        30 |     15 |      45 |
| G            | F   |       16 |         0 |     16 |      50 |
| G            | M   |       16 |         0 |     16 |      50 |
| G            | W   |       16 |         0 |     16 |      50 |
| H            | W   |       10 |         0 |     12 |      30 |
+--------------+-----+----------+-----------+--------+---------+
20 rows in set (0.00 sec)
```

Prereq:-

```
mysql> select * from prereq;
+-----------+-----------+
| course_id | prereq_id |
+-----------+-----------+
| BIO-301   | BIO-101   |
| BIO-399   | BIO-101   |
| CS-190    | CS-101    |
| CS-315    | CS-101    |
| CS-319    | CS-101    |
| CS-347    | CS-101    |
| EE-181    | PHY-101   |
+-----------+-----------+
7 rows in set (0.00 sec)
```

Schema For University Database:-



Question 1:-

Write the following queries in SQL, using the university schema. Schema Attached separately

Question 1a:-

Find the titles of courses in the Comp. Sci. department that have 3 credits.

Solution:-

```
mysql> select title from course
    -> where credits = 3
    -> and
    -> dept_name = 'Comp. Sci.';
+--------------------------+
| title                    |
+--------------------------+
| Robotics                 |
| Image Processing         |
| Database System Concepts |
+--------------------------+
3 rows in set (0.01 sec)
```

Question 1b:-

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

Solution:-

```
mysql> select distinct ID from student natural join takes
    -> where course_id in (select course_id from
    -> instructor natural join teaches
    -> where name = 'Einstein');
+-------+
| ID    |
+-------+
| 44553 |
+-------+
1 row in set (0.02 sec)
```

Question 1c:-

Find the highest salary of any instructor.

Solution:-

```
mysql> select max(salary) from instructor;
+-------------+
| max(salary) |
+-------------+
|    95000.00 |
+-------------+
1 row in set (0.01 sec)
```

Question 1d:-

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

Solution:-

```
mysql> select * from instructor
    -> where salary in (select max(salary) from instructor);
+-------+----------+-----------+----------+
| ID    | name     | dept_name | salary   |
+-------+----------+-----------+----------+
| 22222 | Einstein | Physics   | 95000.00 |
+-------+----------+-----------+----------+
1 row in set (0.00 sec)
```

Question 1e:-

Find the enrollment of each section that was offered in Autumn 2009.

Solution:-

```
mysql> select count(course_id),sec_id from section
    -> where semester = 'Spring' and year = '2009'
    -> group by sec_id;
+------------------+--------+
| count(course_id) | sec_id |
+------------------+--------+
|                2 | 1      |
|                1 | 2      |
+------------------+--------+
2 rows in set (0.00 sec)
```

Question 1f:-

Find the maximum enrollment, across all sections, in Autumn 2009.

Solution:-

```
mysql> with sec_enrollment as (
    -> select course_id, sec_id, count(ID) as enrollment
    -> from section natural join takes
    -> where semester = 'Autumn'
    -> and year = 2009
    -> group by course_id, sec_id)
    -> select course_id, sec_id
    -> from sec_enrollment
    -> where enrollment = (select max(enrollment) from sec_enrollment);
Empty set (0.01 sec)
```

Question 1g:-

Find the sections that had the maximum enrollment in Autumn 2009.

Solution:-

```
mysql> select max(enrollment) from (select count(ID) as enrollment from takes t where t.sec_id in
(select s.sec_id from section s where year=2009 and semester="Autumn")) as dt;
+-----------------+
| max(enrollment) |
+-----------------+
|               0 |
+-----------------+
1 row in set (0.00 sec)
```

Question 2:-

Write the following inserts, deletes or updates in SQL, using the university schema.

Question 2a:-

Increase the salary of each instructor in the Comp. Sci. department by 10%.

Solution:-

```
mysql> update instructor
    -> set salary = salary * 1.10
    -> where dept_name = 'Comp. Sci.';
Query OK, 3 rows affected (0.03 sec)
Rows matched: 3  Changed: 3  Warnings: 0

mysql> select * from instructor;
+-------+------------+------------+-----------+
| ID    | name       | dept_name  | salary    |
+-------+------------+------------+-----------+
| 10101 | Srinivasan | Comp. Sci. |  71500.00 |
| 12121 | Wu         | Finance    |  90000.00 |
| 15151 | Mozart     | Music      |  40000.00 |
| 22222 | Einstein   | Physics    |  95000.00 |
| 32343 | El Said    | History    |  60000.00 |
| 33456 | Gold       | Physics    |  87000.00 |
| 45565 | Katz       | Comp. Sci. |  82500.00 |
| 58583 | Califieri  | History    |  62000.00 |
| 76543 | Singh      | Finance    |  80000.00 |
| 76766 | Crick      | Biology    |  72000.00 |
| 83821 | Brandt     | Comp. Sci. | 101200.00 |
| 98345 | Kim        | Elec. Eng. |  80000.00 |
+-------+------------+------------+-----------+
12 rows in set (0.00 sec)
```

Question 2b:-

Delete all courses that have never been offered (that is, do not occur in the section relation).

Solution:-

```
mysql> delete from course
    -> where course_id not in
    -> (select course_id from section);
Query OK, 1 row affected (0.01 sec)

mysql> select * from course;
+-----------+----------------------------+------------+---------+
| course_id | title                      | dept_name  | credits |
+-----------+----------------------------+------------+---------+
| BIO-101   | Intro. to Biology          | Biology    |       4 |
| BIO-301   | Genetics                   | Biology    |       4 |
| CS-101    | Intro. to Computer Science | Comp. Sci. |       4 |
| CS-190    | Game Design                | Comp. Sci. |       4 |
| CS-315    | Robotics                   | Comp. Sci. |       3 |
| CS-319    | Image Processing           | Comp. Sci. |       3 |
| CS-347    | Database System Concepts   | Comp. Sci. |       3 |
| EE-181    | Intro. to Digital Systems  | Elec. Eng. |       3 |
| FIN-201   | Investment Banking         | Finance    |       3 |
| HIS-351   | World History              | History    |       3 |
| MU-199    | Music Video Production     | Music      |       3 |
| PHY-101   | Physical Principles        | Physics    |       4 |
+-----------+----------------------------+------------+---------+
12 rows in set (0.00 sec)
```

Question 2c:-

Insert every student whose tot_cred attribute is greater than 100 as an instructor in the same department, with a salary of $30,000.

Solution:-

```
mysql> insert into instructor
    -> (select ID, name, dept_name, 30000
    -> from student
    -> where tot_cred > 100);
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from instructor;
+-------+------------+------------+-----------+
| ID    | name       | dept_name  | salary    |
+-------+------------+------------+-----------+
| 00128 | Zhang      | Comp. Sci. |  30000.00 |
| 10101 | Srinivasan | Comp. Sci. |  71500.00 |
| 12121 | Wu         | Finance    |  90000.00 |
| 15151 | Mozart     | Music      |  40000.00 |
| 22222 | Einstein   | Physics    |  95000.00 |
| 23121 | Chavez     | Finance    |  30000.00 |
| 32343 | El Said    | History    |  60000.00 |
| 33456 | Gold       | Physics    |  87000.00 |
| 45565 | Katz       | Comp. Sci. |  82500.00 |
| 58583 | Califieri  | History    |  62000.00 |
| 76543 | Singh      | Finance    |  80000.00 |
| 76766 | Crick      | Biology    |  72000.00 |
| 83821 | Brandt     | Comp. Sci. | 101200.00 |
| 98345 | Kim        | Elec. Eng. |  80000.00 |
| 98988 | Tanaka     | Biology    |  30000.00 |
+-------+------------+------------+-----------+
15 rows in set (0.00 sec)
```

Question 3:-

The SQL like operator is case sensitive, but the lower () function on strings can be used to perform case insensitive matching. To show how, write a query that finds departments whose names contain the string “sci” as a substring, regardless of the case.

Solution:-

```
mysql> select dept_name
    -> from department
    -> where lower(dept_name) like '%sci%';
+------------+
| dept_name  |
+------------+
| Comp. Sci. |
+------------+
1 row in set (0.01 sec)
```

Question 4:-

Write the following queries in SQL using university schema:

Question 4a:-

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

Solution:-

```
mysql> select ID,name,count(sec_id) as 'Number of sections' from instructor natural
 left outer join teaches group by ID,name;
+-------+------------+--------------------+
| ID    | name       | Number of sections |
+-------+------------+--------------------+
| 00128 | Zhang      |                  0 |
| 10101 | Srinivasan |                  3 |
| 12121 | Wu         |                  1 |
| 15151 | Mozart     |                  1 |
| 22222 | Einstein   |                  1 |
| 23121 | Chavez     |                  0 |
| 32343 | El Said    |                  1 |
| 33456 | Gold       |                  0 |
| 45565 | Katz       |                  2 |
| 58583 | Califieri  |                  0 |
| 76543 | Singh      |                  0 |
| 76766 | Crick      |                  2 |
| 83821 | Brandt     |                  3 |
| 98345 | Kim        |                  1 |
| 98988 | Tanaka     |                  0 |
+-------+------------+--------------------+
15 rows in set (0.00 sec)
```

Question 4b:-

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

Solution:-

```
mysql> select ID, name,
    -> (select count(*) as 'Number of sections'
    -> from teaches T where T.id = I.id) as No_of_sec
    -> from instructor I;
+-------+------------+-----------+
| ID    | name       | No_of_sec |
+-------+------------+-----------+
| 00128 | Zhang      |         0 |
| 10101 | Srinivasan |         3 |
| 12121 | Wu         |         1 |
| 15151 | Mozart     |         1 |
| 22222 | Einstein   |         1 |
| 23121 | Chavez     |         0 |
| 32343 | El Said    |         1 |
| 33456 | Gold       |         0 |
| 45565 | Katz       |         2 |
| 58583 | Califieri  |         0 |
| 76543 | Singh      |         0 |
| 76766 | Crick      |         2 |
| 83821 | Brandt     |         3 |
| 98345 | Kim        |         1 |
| 98988 | Tanaka     |         0 |
+-------+------------+-----------+
15 rows in set (0.00 sec)
```

Question 4c:-

Display the list of all course sections offered in Spring 2010, along with the names of the instructors teaching the section. If a section has more than one instructor, it should appear as many times in the result as it has instructors. If it does not have any instructor, it should still appear in the result with the instructor name set to "—".

Solution:-

```
mysql> select course_id, sec_id, ID,
    -> coalesce(name, '-')
    -> from (section natural left outer join teaches)
    -> natural left outer join instructor
    -> where semester='Spring' and year= 2010;
+-----------+--------+-------+---------------------+
| course_id | sec_id | ID    | coalesce(name, '?') |
+-----------+--------+-------+---------------------+
| CS-101    | 1      | 45565 | Katz                |
| FIN-201   | 1      | 12121 | Wu                  |
| MU-199    | 1      | 15151 | Mozart              |
| HIS-351   | 1      | 32343 | El Said             |
| CS-319    | 2      | 83821 | Brandt              |
| CS-319    | 1      | 45565 | Katz                |
| CS-315    | 1      | 10101 | Srinivasan          |
+-----------+--------+-------+---------------------+
7 rows in set (0.00 sec)
```

Question 4d:-

Display the list of all departments, with the total number of instructors in each department, without using scalar subqueries. Make sure to correctly handle departments with no instructors.

Solution:-

```
mysql> select dept_name, count(ID)
    -> from department natural left outer join instructor
    -> group by dept_name;
+------------+-----------+
| dept_name  | count(ID) |
+------------+-----------+
| Biology    |         2 |
| Comp. Sci. |         4 |
| Elec. Eng. |         1 |
| Finance    |         3 |
| History    |         2 |
| Music      |         1 |
| Physics    |         2 |
+------------+-----------+
7 rows in set (0.00 sec)
```

Question 5:-

Outer join expressions can be computed in SQL without using the SQL outer join operation. To illustrate this fact, show how to rewrite each of the following SQL queries without using the outer join expression considering university Schema.

Question 5a:-

select * from student natural left outer join takes

Solution:-

```
mysql> select * from student natural join takes
    -> union
    -> select ID, name, dept_name, tot_cred, NULL, NULL, NULL, NULL, NULL
    -> from student S1 where not exists
    -> (select ID from takes T1 where T1.id = S1.id);
+-------+----------+------------+----------+-----------+--------+----------+------+-------+
| ID    | name     | dept_name  | tot_cred | course_id | sec_id | semester | year | grade |
+-------+----------+------------+----------+-----------+--------+----------+------+-------+
| 00128 | Zhang    | Comp. Sci. |      102 | CS-101    | 1      | Fall     | 2009 | A     |
| 00128 | Zhang    | Comp. Sci. |      102 | CS-347    | 1      | Fall     | 2009 | A-    |
| 12345 | Shankar  | Comp. Sci. |       32 | CS-101    | 1      | Fall     | 2009 | C     |
| 12345 | Shankar  | Comp. Sci. |       32 | CS-190    | 2      | Spring   | 2009 | A     |
| 12345 | Shankar  | Comp. Sci. |       32 | CS-315    | 1      | Spring   | 2010 | A     |
| 12345 | Shankar  | Comp. Sci. |       32 | CS-347    | 1      | Fall     | 2009 | A     |
| 19991 | Brandt   | History    |       80 | HIS-351   | 1      | Spring   | 2010 | B     |
| 23121 | Chavez   | Finance    |      110 | FIN-201   | 1      | Spring   | 2010 | C+    |
| 44553 | Peltier  | Physics    |       56 | PHY-101   | 1      | Fall     | 2009 | B-    |
| 45678 | Levy     | Physics    |       46 | CS-101    | 1      | Fall     | 2009 | F     |
| 45678 | Levy     | Physics    |       46 | CS-101    | 1      | Spring   | 2010 | B+    |
| 45678 | Levy     | Physics    |       46 | CS-319    | 1      | Spring   | 2010 | B     |
| 54321 | Williams | Comp. Sci. |       54 | CS-101    | 1      | Fall     | 2009 | A-    |
| 54321 | Williams | Comp. Sci. |       54 | CS-190    | 2      | Spring   | 2009 | B+    |
| 55739 | Sanchez  | Music      |       38 | MU-199    | 1      | Spring   | 2010 | A-    |
| 76543 | Brown    | Comp. Sci. |       58 | CS-101    | 1      | Fall     | 2009 | A     |
| 76543 | Brown    | Comp. Sci. |       58 | CS-319    | 2      | Spring   | 2010 | A     |
| 76653 | Aoi      | Elec. Eng. |       60 | EE-181    | 1      | Spring   | 2009 | C     |
| 98765 | Bourikas | Elec. Eng. |       98 | CS-101    | 1      | Fall     | 2009 | C-    |
| 98765 | Bourikas | Elec. Eng. |       98 | CS-315    | 1      | Spring   | 2010 | B     |
| 98988 | Tanaka   | Biology    |      120 | BIO-101   | 1      | Summer   | 2009 | A     |
| 98988 | Tanaka   | Biology    |      120 | BIO-301   | 1      | Summer   | 2010 | NULL  |
| 70557 | Snow     | Physics    |        0 | NULL      | NULL   | NULL     | NULL | NULL  |
+-------+----------+------------+----------+-----------+--------+----------+------+-------+
23 rows in set (0.01 sec)
```

Question 5b:-

select * from student natural full outer join takes

Solution:-

```
mysql> (select * from student natural join takes)
    -> union
    -> (select ID, name, dept_name, tot_cred, NULL, NULL, NULL, NULL, NULL
    -> from student S1
    -> where not exists
    -> (select ID from takes T1 where T1.id = S1.id))
    -> union
    -> (select ID, NULL, NULL, NULL, course_id, sec_id, semester, year, grade
    -> from takes T1
    -> where not exists
    -> (select ID from student S1 where T1.id = S1.id))
    -> ;
+-------+----------+------------+----------+-----------+--------+----------+------+-------+
| ID    | name     | dept_name  | tot_cred | course_id | sec_id | semester | year | grade |
+-------+----------+------------+----------+-----------+--------+----------+------+-------+
| 00128 | Zhang    | Comp. Sci. |      102 | CS-101    | 1      | Fall     | 2009 | A     |
| 00128 | Zhang    | Comp. Sci. |      102 | CS-347    | 1      | Fall     | 2009 | A-    |
| 12345 | Shankar  | Comp. Sci. |       32 | CS-101    | 1      | Fall     | 2009 | C     |
| 12345 | Shankar  | Comp. Sci. |       32 | CS-190    | 2      | Spring   | 2009 | A     |
| 12345 | Shankar  | Comp. Sci. |       32 | CS-315    | 1      | Spring   | 2010 | A     |
| 12345 | Shankar  | Comp. Sci. |       32 | CS-347    | 1      | Fall     | 2009 | A     |
| 19991 | Brandt   | History    |       80 | HIS-351   | 1      | Spring   | 2010 | B     |
| 23121 | Chavez   | Finance    |      110 | FIN-201   | 1      | Spring   | 2010 | C+    |
| 44553 | Peltier  | Physics    |       56 | PHY-101   | 1      | Fall     | 2009 | B-    |
| 45678 | Levy     | Physics    |       46 | CS-101    | 1      | Fall     | 2009 | F     |
| 45678 | Levy     | Physics    |       46 | CS-101    | 1      | Spring   | 2010 | B+    |
| 45678 | Levy     | Physics    |       46 | CS-319    | 1      | Spring   | 2010 | B     |
| 54321 | Williams | Comp. Sci. |       54 | CS-101    | 1      | Fall     | 2009 | A-    |
| 54321 | Williams | Comp. Sci. |       54 | CS-190    | 2      | Spring   | 2009 | B+    |
| 55739 | Sanchez  | Music      |       38 | MU-199    | 1      | Spring   | 2010 | A-    |
| 76543 | Brown    | Comp. Sci. |       58 | CS-101    | 1      | Fall     | 2009 | A     |
| 76543 | Brown    | Comp. Sci. |       58 | CS-319    | 2      | Spring   | 2010 | A     |
| 76653 | Aoi      | Elec. Eng. |       60 | EE-181    | 1      | Spring   | 2009 | C     |
| 98765 | Bourikas | Elec. Eng. |       98 | CS-101    | 1      | Fall     | 2009 | C-    |
| 98765 | Bourikas | Elec. Eng. |       98 | CS-315    | 1      | Spring   | 2010 | B     |
| 98988 | Tanaka   | Biology    |      120 | BIO-101   | 1      | Summer   | 2009 | A     |
| 98988 | Tanaka   | Biology    |      120 | BIO-301   | 1      | Summer   | 2010 | NULL  |
| 70557 | Snow     | Physics    |        0 | NULL      | NULL   | NULL     | NULL | NULL  |
+-------+----------+------------+----------+-----------+--------+----------+------+-------+
23 rows in set (0.01 sec)
```

Question 6:-

Consider a table with the schema BankCustomers (accNum, name and loan). Raise an exception when the customer initiates a loan amount above 10 lakhs.

Solution:-

```
mysql> create table BankCustomers(
    -> accNum varchar(35),name varchar(40),loan int);
Query OK, 0 rows affected (0.03 sec)

mysql> alter table BankCustomers add check(loan<=1000000);
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> insert into BankCustomers values("C109","Harry",50000);
Query OK, 1 row affected (0.01 sec)

mysql> insert into BankCustomers values("C119","Hari",1050000);
ERROR 3819 (HY000): Check constraint 'bankcustomers_chk_1' is violated.
mysql>
```

Question 7:-

Create an error handler that terminates the stored procedure whenever a duplicate key occurs and list out the number of customers who are majors.

Solution:-

```
use sql_revision;
drop procedure if exists duplicate;
delimiter //
create procedure duplicate (in id int, in name varchar(30), in dept varchar(40), in age int, in place varchar(40), in income int, in doj datetime)
begin
    declare count int;
    select count(*) into count from employee where emp_id = id;

    if (count > 0) then
        select * from employee where emp_age > 18;
        signal sqlstate "1444" set message_text = "Already Exists";
    end if;
end;

delimiter;

call duplicate (2590, "tonynflas", "sci", 56, "malibu", 999990, "26-04-1998");
```

Question 8:-

Write two triggers, one trigger will update the total asset for corresponding branch whenever an account is removed from the account table and another which will update the depositor table by deleting the entry for that account.

Solution:-

```
mysql> delimiter //
mysql> create trigger `update_branch` After delete on `account` for each row
    -> Begin
    -> update branch set branch.assets=branch.assets-old.balance where branch.branch_name=old.branch_name;
    -> end//
```

```
mysql> delimiter //
mysql> create trigger `update_depositor` After delete on `account` for each row
    -> Begin
    -> delete from depositor where depositor.account_number=old.account_number;
    -> end//
```

Question 9:-

Write a stored function to assign a position to each employee based on their experience i.e. if the experience is > 25 years then senior executive, >10 years then executive, >5 years then senior analyst and else as analyst. Create a separate table as EmployeePosition where you will have Emp_id, Emp_name, Position details.

Solution:-

```
mysql> create table Employee(emp_id int, emp_name varchar(20), experience int);
    -> create table Employee_Position (emp_id int, emp_name varchar(20),postion varchar(30));
    -> DELIMITER $$
Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.02 sec)
```

```
mysql> CREATE FUNCTION `employee_position`(experience int)
    -> RETURNS varchar(30)
    -> DETERMINISTIC
    -> BEGIN
    -> declare position varchar(30);
    -> if experience > 25 then
    -> set position = 'senior executive';
    -> elseif experience > 10 then
    -> set position = 'executive';
    -> elseif experience > 5 then
    -> set position = 'senior analyst';
    -> else
    -> set position = 'analyst';
    -> end if;
    -> return position;
    -> RETURN 1;
    -> END $$
Query OK, 0 rows affected (0.00 sec)
```

Question 10:-

For the above question, create a trigger so that whenever there is a new entry in the Employee table, it will automatically update the EmployeePosition table.

Solution:-

```
mysql> CREATE trigger autoUpdateEmpPosition
    -> after insert
    -> on Employee for each row
    -> insert into Employee_Position values
    -> (new.emp_id,new.emp_name,employee_position(new.experience))
    ->
    -> insert into Employee values(10,"ABC",6);
    -> select * from employee_position;$$
Query OK, 0 rows affected (0.01 sec)

Query OK, 1 row affected (0.03 sec)

+--------+----------+----------------+
| emp_id | emp_name | postion        |
+--------+----------+----------------+
|     10 | ABC      | senior analyst |
+--------+----------+----------------+
1 row in set (0.03 sec)
```

←THE END→