DBMS Lab
2021

Final Exam

Yogesh R, 111801047

Part-I: Use MariaDB and university database [18]

1. Show the name of the departments which have more number instructors than number of students. [**2**]

```
MariaDB [university_endsem]> select instrutor_count.dept_name
    -> from (
    ->  select dept_name, count(ID) as students
    ->  from student
    ->  group by dept_name
    -> ) as student_count,
    -> (
    ->     select dept_name, count(ID) as instructors
    ->     from instructor
    ->     group by dept_name
    ->  ) as instrutor_count
    ->
    -> where (student_count.dept_name = instrutor_count.dept_name) and
    ->       (student_count.students < instrutor_count.instructors);
+------------+
| dept_name  |
+------------+
| Finance    |
| History    |
+------------+
2 rows in set (0.002 sec)
```

2. Show the names of all students who have taken any course from the Comp. Sci.department. [**2**]

```
MariaDB [university_endsem]> select ID, name
    -> from student
    -> inner join (
    ->     select distinct ID
    ->     from takes
    ->                 inner join (
    ->         select *
    ->         from course
    ->         where dept_name='Comp. Sci.'
    ->     ) as cs using (course_id)
    -> ) as cs_students using(ID);
+-------+-----------+
| ID    | name      |
+-------+-----------+
| 00128 | Zhang     |
| 12345 | Shankar   |
| 45678 | Levy      |
| 54321 | Williams  |
| 76543 | Brown     |
| 98765 | Bourikas  |
+-------+-----------+
6 rows in set (0.001 sec)
```

3. Show the name of the instructors and the courses they have taught if the number of courses taught by the instructor is more than 1. [**3**]

```
MariaDB [university_endsem]> select ID, name, course_id, title
    -> from teaches
    -> inner join course using (course_id)
    -> inner join (
    ->     select ID, name
    ->     from instructor
    ->              inner join (
    ->         select ID
    ->         from teaches
    ->         group by ID
    ->         having count(course_id) > 1
    ->     ) as more_courses using (ID)
    -> ) as more_teaching_inst using(ID);
+-------+------------+-----------+--------------------------------+
| ID    | name       | course_id | title                          |
+-------+------------+-----------+--------------------------------+
| 10101 | Srinivasan | CS-101    | Intro. to Computer Science |
| 10101 | Srinivasan | CS-315    | Robotics                       |
| 10101 | Srinivasan | CS-347    | Database System Concepts   |
| 45565 | Katz       | CS-101    | Intro. to Computer Science |
| 45565 | Katz       | CS-319    | Image Processing           |
| 76766 | Crick      | BIO-101   | Intro. to Biology          |
| 76766 | Crick      | BIO-301   | Genetics                   |
| 83821 | Brandt     | CS-190    | Game Design                |
| 83821 | Brandt     | CS-190    | Game Design                |
| 83821 | Brandt     | CS-319    | Image Processing           |
+-------+------------+-----------+--------------------------------+
10 rows in set (0.001 sec)
```

4. Show the names of the students who have taken courses from departments which are different from the department of their enrolment. Show the names of the students, names of the courses, departments of the courses, and departments of enrolments. [**3**]

```
MariaDB [university_endsem]> select ID, name, student.dept_name as student_dept,
    ->        course_id, title, course_dept.dept_name as course_dept
    -> from student
    -> inner join (
    ->     select distinct ID, course_id, title, dept_name
    ->     from takes
    ->     inner join course using (course_id)
    -> ) as course_dept using (ID)
    -> where course_dept.dept_name<>student.dept_name;
+-------+----------+--------------+-----------+--------------------------------+-------------+
| ID    | name     | student_dept | course_id | title                          | course_dept |
+-------+----------+--------------+-----------+--------------------------------+-------------+
| 45678 | Levy     | Physics      | CS-319    | Image Processing           | Comp. Sci.  |
| 45678 | Levy     | Physics      | CS-101    | Intro. to Computer Science | Comp. Sci.  |
| 98765 | Bourikas | Elec. Eng.   | CS-315    | Robotics                       | Comp. Sci.  |
| 98765 | Bourikas | Elec. Eng.   | CS-101    | Intro. to Computer Science | Comp. Sci.  |
+-------+----------+--------------+-----------+--------------------------------+-------------+
4 rows in set (0.001 sec)
```

5. Create a table oddeven that contains one integer field and one varchar (5) field. Create a procedure poe, that accepts two integers say a, b. Then poeinserts into table oddevenall integers between and including a and b. For the second field poeinserts ‘odd’ or ‘even’
   depending on the integer value in the first field. For example { {1,’odd’}, {2,’even’} }. [**4**]

```
MariaDB [university_endsem]> create table oddeven (
    ->   x int,
    ->   y varchar(5)
    -> );
Query OK, 0 rows affected (0.059 sec)
```

```
MariaDB [university_endsem]> delimiter //
MariaDB [university_endsem]>
MariaDB [university_endsem]> create procedure poe (in a int, in b int)
    -> begin
    ->     while a≤ b do
    ->         if a % 2 = 0 then
    ->             insert into oddeven values (a, 'even');
    ->         else
    ->             insert into oddeven values (a, 'odd');
    ->         end if;
    ->         set a = a + 1;
    ->     end while;
    -> end //
Query OK, 0 rows affected (0.014 sec)

MariaDB [university_endsem]> delimiter ;
```

```
MariaDB [university_endsem]> CALL poe(1, 4);
Query OK, 4 rows affected (0.009 sec)

MariaDB [university_endsem]> select *
    -> from oddeven;
+------+------+
| x    | y    |
+------+------+
|    1 | odd  |
|    2 | even |
|    3 | odd  |
|    4 | even |
+------+------+
4 rows in set (0.001 sec)
```

6. Create a function get_user which will return the username of the currently logged in user. Create a user Snow with password white. Demonstrate the function get_user using Snow. Use get_user in a query of your choice. [**4**]

```
MariaDB [university_endsem]> delimiter //
MariaDB [university_endsem]>
MariaDB [university_endsem]> create function get_user()
    -> returns varchar(32)
    -> begin
    -> return current_user();
    -> end//
Query OK, 0 rows affected (0.006 sec)

MariaDB [university_endsem]> delimiter ;
```

```
MariaDB [university_endsem]> create user 'Snow' identified by 'white';
Query OK, 0 rows affected (0.003 sec)

MariaDB [university_endsem]> GRANT all on university_endsem.* to 'Snow';
Query OK, 0 rows affected (0.006 sec)
```

As Snow

```
MariaDB [university_endsem]> select get_user();
+--------------+
| get_user()   |
+--------------+
| Snow@%       |
+--------------+
1 row in set (0.000 sec)
```

Part-II: MariaDB and canteen database [4]

1. Create a database canteen. Create a table menu with attributes id int not null, and name varchar(50) not null, type that can take value between ‘ healthy’ , and ‘ unhealthy’. Create another table customerorder with attribute idnot null , and countint not null. Create a table price that will contain id of a dish in the menu and amountfloat. [**1**]

```
MariaDB [(none)]> create database canteen;
Query OK, 1 row affected (0.007 sec)

MariaDB [(none)]> use canteen;
Database changed
```

```
MariaDB [canteen]> create table menu (
    ->     id int not null ,
    ->     name varchar(50) not null ,
    ->     type varchar(50),
    ->     check ( type in ('healthy', 'unhealthy') )
    -> );
Query OK, 0 rows affected (0.019 sec)

MariaDB [canteen]>
MariaDB [canteen]> create table customerorder (
    ->     id int not null,
    ->     count int not null
    -> );
Query OK, 0 rows affected (0.024 sec)

MariaDB [canteen]> create table price (
    ->     id int not null ,
    ->     amount float
    -> );
Query OK, 0 rows affected (0.012 sec)
```

2. Create a trigger init_price that will check the type of the newly added dish in the menu, and will automatically initialize a price in the correct table. For healthy foods price is 10, and for unhealthy foods price is 15. [**3**]

```
MariaDB [canteen]> delimiter //
MariaDB [canteen]> create trigger init_price
    ->     after insert on menu
    ->     for each row
    ->         if (NEW.type = 'healthy')
    ->         then
    ->             insert into price (id, amount) values (NEW.id, 10);
    ->         else
    ->             insert into price (id, amount) values (NEW.id, 15);
    ->         end if //
Query OK, 0 rows affected (0.029 sec)

MariaDB [canteen]> delimiter ;
```

```
MariaDB [canteen]> insert into menu values (1, 'rice', 'healthy');
Query OK, 1 row affected (0.017 sec)

MariaDB [canteen]> insert into menu values (2, 'burger', 'unhealthy');
Query OK, 1 row affected (0.003 sec)
```

```
MariaDB [canteen]> select *
    -> from menu;
+----+--------+-----------+
| id | name   | type      |
+----+--------+-----------+
|  1 | rice   | healthy   |
|  2 | burger | unhealthy |
+----+--------+-----------+
2 rows in set (0.018 sec)
```

```
MariaDB [canteen]> select *
    -> from price;
+----+--------+
| id | amount |
+----+--------+
|  1 |     10 |
|  2 |     15 |
+----+--------+
2 rows in set (0.002 sec)
```

Part-III: Use MongoDB and primer database [3]

1. Show the number of restaurants in Manhatttan for each cuisine in ascending order if the number is more than 100 and less than 500.
   [**3**]

   See 111801047_mongo.ipynb