Name:Surendra Baskey(111601027)

use university database and do following.

1. [P] Create a variable to store the average of total credits of students from stud_to_cred table. Show name s of the students with total credit more than twice the average of total credits.

2. Create a function fun_sel_avg that when called will accept a threshold and return average budgets of those departments which have budget more than the threshold. Use 10000 as the threshold to verify your function.

```
drop function if exists fun_sel_avg;
DELIMITER $$
create function fun_sel_avg(threshold decimal(10,2))
returns decimal(10,2)
deterministic
begin
declare avg_budg decimal(10,2);
   select avg(budget) into avg_budg
   from department where budget>threshold;
return avg_budg;
end;$$
DELIMITER ;
```

```
MariaDB [university]> select fun_sel_avg(10000);
+---------------------+
| fun_sel_avg(10000) |
+---------------------+
|           85000.00 |
+---------------------+
1 row in set (0.001 sec)
```

3. Create a procedure sp_top_budg that when called will accept a number N and return budgets of top N departments. Use N=3 to verify your procedure.

```
drop procedure if exists sp_top_budg;
delimiter $$
create procedure sp_top_budg(in N int)
begin
select budget
from department
order by budget desc limit N;
end;$$
delimiter ;
```

```
MariaDB [university]> call sp_top_budg(3);
+-----------+
| budget    |
+-----------+
| 120000.00 |
| 100000.00 |
|  90000.00 |
+-----------+
3 rows in set (0.013 sec)

Query OK, 0 rows affected (0.013 sec)
```

4. Insert following into department: (‘Dance’,’Watson’,10000). Create a procedure which will calculate average budget of all department and then update budget of a department to half of the average budget if the budget of that department is less than half of the current average budget.

```
MariaDB [university]> insert into department values('Dance','Watson',10000);
Query OK, 1 row affected (0.064 sec)

MariaDB [university]> select *from department
    -> ;
+------------+----------+-----------+
| dept_name  | building | budget    |
+------------+----------+-----------+
| Biology    | Watson   |  90000.00 |
| Comp. Sci. | Taylor   | 100000.00 |
| Dance      | Watson   |  10000.00 |
| Elec. Eng. | Taylor   |  85000.00 |
| Finance    | Painter  | 120000.00 |
| History    | Painter  |  50000.00 |
| Music      | Packard  |  80000.00 |
| Physics    | Watson   |  70000.00 |
+------------+----------+-----------+
8 rows in set (0.001 sec)

MariaDB [university]>
```

a. Do it without calling any function of procedure.

```
drop procedure if exists update_budg;
delimiter $$
create procedure update_budg()
begin
declare avg_budg decimal(10,2);
select avg(budget)/2 from department into avg_budg;
update department
```

```
set budget=avg_budg where budget < avg_budg;
end; $$
delimiter ;
```

```
Database changed
MariaDB [university]> call update_budg();
Query OK, 2 rows affected (0.153 sec)

MariaDB [university]> select *from department;
+------------+----------+-----------+
| dept_name  | building | budget    |
+------------+----------+-----------+
| Biology    | Watson   |  90000.00 |
| Comp. Sci. | Taylor   | 100000.00 |
| Dance      | Watson   |  37812.50 |
| Elec. Eng. | Taylor   |  85000.00 |
| Finance    | Painter  | 120000.00 |
| History    | Painter  |  50000.00 |
| Music      | Packard  |  80000.00 |
| Physics    | Watson   |  70000.00 |
+------------+----------+-----------+
8 rows in set (0.001 sec)

MariaDB [university]>
```

b. Do it by calling function fun_sel_avg.

```
drop procedure if exists update_fun_avg_sel;
delimiter $$
create procedure update_fun_sel_avg()
begin
declare avg_budg decimal(10,2);
set avg_budg=fun_sel_avg()/2;
update department set budget=avg_budg where budget < avg_budg;
end; $$
delimiter ;
```

```
Database changed
MariaDB [university]> call update_budg();
Query OK, 2 rows affected (0.153 sec)

MariaDB [university]> select *from department;
+------------+----------+-----------+
| dept_name  | building | budget    |
+------------+----------+-----------+
| Biology    | Watson   |  90000.00 |
| Comp. Sci. | Taylor   | 100000.00 |
| Dance      | Watson   |  37812.50 |
| Elec. Eng. | Taylor   |  85000.00 |
| Finance    | Painter  | 120000.00 |
| History    | Painter  |  50000.00 |
| Music      | Packard  |  80000.00 |
| Physics    | Watson   |  70000.00 |
+------------+----------+-----------+
8 rows in set (0.001 sec)

MariaDB [university]>
```

c. Do it by calling procedure sp_top_avg.

5. Create a function fun_sq that squares a number. Use f un_sq to define a procedure sp_sel_avg_sq that will accept a threshold budget and return square of the average of the budgets of those departments which have budget more than the threshold. Use 10000 as the threshold to verify your procedure.

```
drop function if exists fun_sq;
delimiter $$
CREATE function fun_sq(num decimal)
returns decimal
deterministic
BEGIN
return num*num;
end; $$
delimiter ;



drop procedure if exists sp_sel_avg_sq;
delimiter $$
CREATE procedure sp_sel_avg_sq(in threshold decimal,out ret_value decimal)
BEGIN
declare sp decimal;
```

```
select avg(budget) from department WHERE budget>threshold into sp;
set ret_value=fun_sq(sp);
end; $$
delimiter ;
```

```
MariaDB [university]> call sp_sel_avg_sq(10000,@sq);
Query OK, 1 row affected, 1 warning (0.001 sec)

MariaDB [university]> select @sq;
+------------+
| @sq        |
+------------+
| 6257126404 |
+------------+
1 row in set (0.000 sec)

MariaDB [university]>
```

6. Create a procedure chek_avg that will accept a threshold and check if the average budget is greater than the threshold, then chek_avg will return 1 otherwise will return 0.Event Use your university database.

```
drop procedure if exists chek_avg;
delimiter $$
create procedure chek_avg(in threshold decimal(10,0))
begin
declare chk_budget decimal(10,0);
select avg(budget) into chk_budget
from department;
if chk_budget > threshold then
select 1;
else
select 0;
end if;
end;$$
delimiter ;
```

```
MariaDB [university]> select avg(budget) from department;
+--------------+
| avg(budget)  |
+--------------+
| 79101.562500 |
+--------------+
1 row in set (0.000 sec)

MariaDB [university]> call chek_avg(900000);
+---+
| 0 |
+---+
| 0 |
+---+
1 row in set (0.001 sec)

Query OK, 1 row affected, 1 warning (0.001 sec)

MariaDB [university]> call chek_avg(9000);
+---+
| 1 |
+---+
| 1 |
+---+
1 row in set (0.001 sec)

Query OK, 1 row affected, 1 warning (0.001 sec)
```

1. Check if the event scheduler is ON. If not on, then set it ON.

```
MariaDB [university]> SHOW VARIABLES
    -> WHERE VARIABLE_NAME = 'event_scheduler';
+-----------------+-------+
| Variable_name   | Value |
+-----------------+-------+
| event_scheduler | OFF   |
+-----------------+-------+
1 row in set (0.151 sec)

MariaDB [university]>
```

```
MariaDB [university]> set GLOBAL  event_scheduler='ON';
Query OK, 0 rows affected (0.018 sec)

MariaDB [university]> SHOW VARIABLES
    -> WHERE VARIABLE_NAME='event_scheduler';
+-----------------+-------+
| Variable_name   | Value |
+-----------------+-------+
| event_scheduler | ON    |
+-----------------+-------+
1 row in set (0.001 sec)

MariaDB [university]>
```

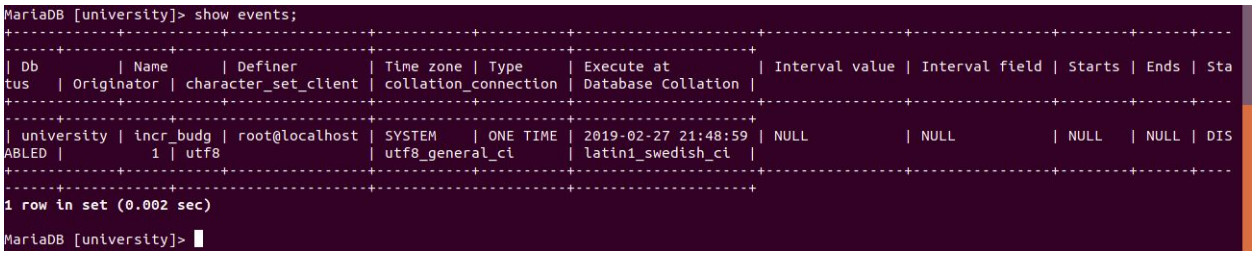
2. Create an event to increment budget of all departments by 5% after 1 minute.

```
CREATE EVENT IF NOT EXISTS incr_budg
ON SCHEDULE AT CURRENT_TIMESTAMP + INTERVAL 1 MINUTE
ON COMPLETION PRESERVE
DO
  UPDATE department
  SET budget=1.05*budget;
```

```
MariaDB [university]> select *from department;
+------------+----------+-----------+
| dept_name  | building | budget    |
+------------+----------+-----------+
| Biology    | Watson   |  90000.00 |
| Comp. Sci. | Taylor   | 100000.00 |
| Dance      | Watson   |  37812.50 |
| Elec. Eng. | Taylor   |  85000.00 |
| Finance    | Painter  | 120000.00 |
| History    | Painter  |  50000.00 |
| Music      | Packard  |  80000.00 |
| Physics    | Watson   |  70000.00 |
+------------+----------+-----------+
8 rows in set (0.000 sec)

MariaDB [university]> select *from department;
+------------+----------+-----------+
| dept_name  | building | budget    |
+------------+----------+-----------+
| Biology    | Watson   |  94500.00 |
| Comp. Sci. | Taylor   | 105000.00 |
| Dance      | Watson   |  39703.13 |
| Elec. Eng. | Taylor   |  89250.00 |
| Finance    | Painter  | 126000.00 |
| History    | Painter  |  52500.00 |
| Music      | Packard  |  84000.00 |
| Physics    | Watson   |  73500.00 |
+------------+----------+-----------+
8 rows in set (0.001 sec)
```

3. Show the details of the events in an easy to read format.



```
MariaDB [university]> show events;
+------------+----------+---------------+----------+---------+---------------------+----------------+----------------+---------+------+---
---+-----------+--------------------+----------------------+--------------------+
| Db         | Name     | Definer       | Time zone | Type    | Execute at          | Interval value | Interval field | Starts | Ends | Sta
tus   | Originator | character_set_client | collation_connection | Database Collation |
+------------+----------+---------------+----------+---------+---------------------+----------------+----------------+---------+------+---
---+-----------+--------------------+----------------------+--------------------+
| university | incr_budg | root@localhost | SYSTEM    | ONE TIME | 2019-02-27 21:48:59 | NULL           | NULL           | NULL    | NULL | DIS
ABLED |          1 | utf8                 | utf8_general_ci      | latin1_swedish_ci  |
+------------+----------+---------------+----------+---------+---------------------+----------------+----------------+---------+------+---
---+-----------+--------------------+----------------------+--------------------+
1 row in set (0.002 sec)

MariaDB [university]>
```

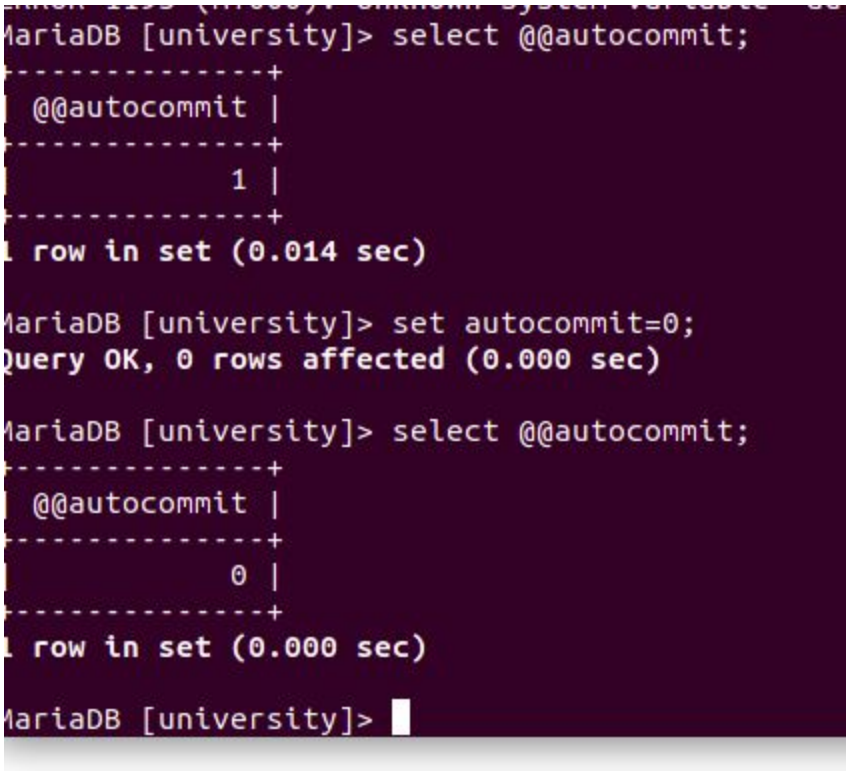
4. Modify the above event so that, it will increment budget of all departments by 100 in every minute for next 5 minutes.

```
CREATE EVENT IF NOT EXISTS incr_budg
ON SCHEDULE EVERY 1 MINUTE
STARTS CURRENT_TIMESTAMP
ENDS CURRENT_TIMESTAMP + INTERVAL 5 MINUTE
DO
```

```
UPDATE department
SET budget=budget + 100;
```

Commit and rollback

1. Check if autocommit is ON
2. Set autocommit OFF



3. Update department to add 0.01 to the budget of Physics department. Rollback, and see. Again Update department to add 0.01 to the budget of Physics department. Commit and See.

```
MariaDB [university]> update department
    -> set budget=budget+0.01 where dept_name='Physics';
Query OK, 1 row affected (0.099 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [university]> select *From department;
+------------+----------+-----------+
| dept_name  | building | budget    |
+------------+----------+-----------+
| Biology    | Watson   |  94900.00 |
| Comp. Sci. | Taylor   | 105400.00 |
| Dance      | Watson   |  40103.13 |
| Elec. Eng. | Taylor   |  89650.00 |
| Finance    | Painter  | 126400.00 |
| History    | Painter  |  52900.00 |
| Music      | Packard  |  84400.00 |
| Physics    | Watson   |  74000.01 |
+------------+----------+-----------+
8 rows in set (0.001 sec)

MariaDB [university]> rollback;
\Query OK, 0 rows affected (0.134 sec)

MariaDB [university]> select *From department;
+------------+----------+-----------+
| dept_name  | building | budget    |
+------------+----------+-----------+
| Biology    | Watson   |  95000.00 |
| Comp. Sci. | Taylor   | 105500.00 |
| Dance      | Watson   |  40203.13 |
| Elec. Eng. | Taylor   |  89750.00 |
| Finance    | Painter  | 126500.00 |
| History    | Painter  |  53000.00 |
| Music      | Packard  |  84500.00 |
| Physics    | Watson   |  74000.00 |
+------------+----------+-----------+
```

```
MariaDB [university]> update department set budget=budget+0.01 where dept_name='Physics';
Query OK, 1 row affected (0.001 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [university]> commit;
Query OK, 0 rows affected (0.028 sec)

MariaDB [university]> select *From department;
+------------+----------+-----------+
| dept_name  | building | budget    |
+------------+----------+-----------+
| Biology    | Watson   |  95000.00 |
| Comp. Sci. | Taylor   | 105500.00 |
| Dance      | Watson   |  40203.13 |
| Elec. Eng. | Taylor   |  89750.00 |
| Finance    | Painter  | 126500.00 |
| History    | Painter  |  53000.00 |
| Music      | Packard  |  84500.00 |
| Physics    | Watson   |  74000.01 |
+------------+----------+-----------+
8 rows in set (0.001 sec)

MariaDB [university]>
```

4. Create a table budget_only and rollback.

Profiling

1. Set profiling ON

```
MariaDB [university]> select @@profiling;
+-------------+
| @@profiling |
+-------------+
|           0 |
+-------------+
1 row in set (0.000 sec)

MariaDB [university]> set profiling=1;
Query OK, 0 rows affected (0.000 sec)

MariaDB [university]> select @@profiling;
+-------------+
| @@profiling |
+-------------+
|           1 |
+-------------+
1 row in set (0.009 sec)

MariaDB [university]>
```

2. Use university database, and execute a query to select all departments having budget greater than 50000. Show the details of this query from the profiler.

```
MariaDB [university]> select *from department having budget>50000;
+------------+----------+-----------+
| dept_name  | building | budget    |
+------------+----------+-----------+
| Biology    | Watson   |  94900.00 |
| Comp. Sci. | Taylor   | 105400.00 |
| Elec. Eng. | Taylor   |  89650.00 |
| Finance    | Painter  | 126400.00 |
| History    | Painter  |  52900.00 |
| Music      | Packard  |  84400.00 |
| Physics    | Watson   |  73900.00 |
+------------+----------+-----------+
7 rows in set (0.048 sec)

MariaDB [university]> show profiles;
+----------+------------+-----------------------------------------------+
| Query_ID | Duration   | Query                                         |
+----------+------------+-----------------------------------------------+
|        1 | 0.00919416 | select @@profiling                            |
|        2 | 0.01862910 | select *from department                       |
|        3 | 0.00012235 | show profiling                                |
|        4 | 0.04756698 | select *from department having budget>50000 |
+----------+------------+-----------------------------------------------+
4 rows in set (0.000 sec)
```

```
MariaDB [university]> show profile for query 4
    -> ;
+----------------------------------+----------+
| Status                           | Duration |
+----------------------------------+----------+
| Starting                         | 0.000052 |
| Waiting for query cache lock     | 0.000007 |
| Init                             | 0.000006 |
| Checking query cache for query   | 0.016354 |
| Checking permissions             | 0.000036 |
| Opening tables                   | 0.000032 |
| After opening tables             | 0.000010 |
| System lock                      | 0.000008 |
| Table lock                       | 0.000015 |
| Waiting for query cache lock     | 0.000019 |
| Init                             | 0.016162 |
| Optimizing                       | 0.014471 |
| Statistics                       | 0.000055 |
| Preparing                        | 0.000036 |
| Executing                        | 0.000005 |
| Sending data                     | 0.000189 |
| End of update loop               | 0.000009 |
| Query end                        | 0.000006 |
| Commit                           | 0.000009 |
| Closing tables                   | 0.000008 |
| Unlocking tables                 | 0.000004 |
| Closing tables                   | 0.000010 |
| Starting cleanup                 | 0.000005 |
| Freeing items                    | 0.000012 |
| Updating status                  | 0.000039 |
| Reset for next command           | 0.000007 |
+----------------------------------+----------+
26 rows in set (0.001 sec)
```