

# Assignment 10

Yogesh R, 111801047

## A. Do following tasks using University Database: - (1.5)

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

```
MariaDB [(none)]> SELECT @@event_scheduler;
+-----+
| @@event_scheduler |
+-----+
| OFF               |
+-----+
1 row in set (0.000 sec)
```

Since the event scheduler is off, we turn it on.

```
MariaDB [university]> set global event_scheduler = on;
Query OK, 0 rows affected (0.000 sec)
```

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

```
MariaDB [university]> create event test_event
→ on schedule every 1 minute
→ do update department set budget=budget*1.05;
Query OK, 0 rows affected (0.009 sec)
```

```
MariaDB [university]> select curtime();
+-----+
| curtime() |
+-----+
| 11:14:54   |
+-----+
1 row in set (0.001 sec)

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

```
MariaDB [university]> select curtime();
+-----+
| curtime() |
+-----+
| 11:16:00   |
+-----+
1 row in set (0.000 sec)

MariaDB [university]>
MariaDB [university]> select * from department;
+-----+-----+-----+
| dept_name | building | budget |
+-----+-----+-----+
| Biology   | Watson   | 99225.00 |
| Comp. Sci. | Taylor   | 110250.00 |
| Elec. Eng. | Taylor   | 93712.50 |
| Finance    | Painter   | 132300.00 |
| History    | Painter   | 55125.00 |
| Music      | Packard   | 88200.00 |
| Physics    | Watson    | 77175.00 |
+-----+-----+-----+
7 rows in set (0.001 sec)
```

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

```
MariaDB [university]> SHOW EVENTS\G;
***** 1. row *****
      Db: university
      Name: test_event
      Definer: root@localhost
      Time zone: SYSTEM
      Type: RECURRING
      Execute at: NULL
      Interval value: 1
      Interval field: MINUTE
      Starts: 2021-04-30 11:46:24
      Ends: NULL
      Status: ENABLED
      Originator: 1
character_set_client: cp850
collation_connection: cp850_general_ci
      Database Collation: latin1_swedish_ci
1 row in set (0.009 sec)
```

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

```
MariaDB [university]> alter event test_event
→      on schedule every 1 minute
→      ends current_timestamp + interval 5 minute
→      do update department set budget=budget+100;
Query OK, 0 rows affected (0.001 sec)
```

```
MariaDB [university]> SHOW EVENTS\G;
***** 1. row *****
      Db: university
      Name: test_event
      Definer: root@localhost
      Time zone: SYSTEM
      Type: RECURRING
      Execute at: NULL
      Interval value: 1
      Interval field: MINUTE
      Starts: 2021-04-30 11:47:49
      Ends: 2021-04-30 11:52:49
      Status: ENABLED
      Originator: 1
character_set_client: cp850
collation_connection: cp850_general_ci
      Database Collation: latin1_swedish_ci
1 row in set (0.009 sec)
```

As we can see the Ends is now present (unlike the previous case), with 5 min interval from the starts.

## B. Do Following Tasks: - (1.5)

### a. Set profiling ON

```
MariaDB [(none)]> SET profiling = 1;  
Query OK, 0 rows affected (0.000 sec)
```

### b. Show list of processes running in your DB.

```
MariaDB [(none)]> show processlist ;
```

Id	User	Host	db	Command	Time	State	Info	Progress
4	root	localhost:49685	university	Sleep	839		NULL	0.000
8	event_scheduler	localhost	NULL	Daemon	2032	Waiting on empty queue	NULL	0.000
16	root	localhost:52814	NULL	Query	0	starting	show processlist	0.000

```
3 rows in set (0.000 sec)
```

We can see the default processes and the event scheduler added in the previous question.

### c. Import database from “largeRelationsInsertFile.sql” and execute following queries and point out the bottlenecks (most costly task in terms of space and time).

#### i. select all departments having budget greater than 50000.

```
MariaDB [university]> select *  
→ from department  
→ where budget>50000;
```

dept_name	building	budget
Accounting	Saucon	441840.92
Astronomy	Taylor	617253.94
Athletics	Bronfman	734550.70
Biology	Candlestick	647610.55
Civil Eng.	Chandler	255041.46
Comp. Sci.	Lamberton	106378.69
Cybernetics	Mercer	794541.46
Elec. Eng.	Main	276527.61
English	Palmer	611042.66
Finance	Candlestick	866831.75
Geology	Palmer	406557.93
History	Taylor	699140.86
Languages	Linderman	601283.60
Marketing	Lambeau	210627.58
Math	Brodhead	777605.11
Mech. Eng.	Rauch	520350.65
Physics	Wrigley	942162.76
Pol. Sci.	Whitman	573745.09
Psychology	Thompson	848175.04
Statistics	Taylor	395051.74

```
20 rows in set (0.001 sec)
```

```

MariaDB [university]> show profiles;
+-----+-----+-----+
| Query_ID | Duration | Query |
+-----+-----+-----+
| 1 | 0.00019140 | SET profiling = 1 |
| 2 | 0.00037290 | select *
from department
where budget>50000 |
+-----+-----+-----+
2 rows in set (0.000 sec)

```

We can see that Query ID 2 corresponds to our recent query.

```

MariaDB [university]> show profile CPU for query 2;
+-----+-----+-----+-----+
| Status | Duration | CPU_user | CPU_system |
+-----+-----+-----+-----+
| Starting | 0.000070 | 0.000000 | 0.000000 |
| checking permissions | 0.000007 | 0.000000 | 0.000000 |
| Opening tables | 0.000031 | 0.000000 | 0.000000 |
| After opening tables | 0.000005 | 0.000000 | 0.000000 |
| System lock | 0.000005 | 0.000000 | 0.000000 |
| table lock | 0.000017 | 0.000000 | 0.000000 |
| init | 0.000021 | 0.000000 | 0.000000 |
| Optimizing | 0.000009 | 0.000000 | 0.000000 |
| Statistics | 0.000015 | 0.000000 | 0.000000 |
| Preparing | 0.000014 | 0.000000 | 0.000000 |
| Executing | 0.000004 | 0.000000 | 0.000000 |
| Sending data | 0.000068 | 0.000000 | 0.000000 |
| End of update loop | 0.000005 | 0.000000 | 0.000000 |
| Query end | 0.000009 | 0.000000 | 0.000000 |
| Commit | 0.000005 | 0.000000 | 0.000000 |
| closing tables | 0.000004 | 0.000000 | 0.000000 |
| Unlocking tables | 0.000003 | 0.000000 | 0.000000 |
| closing tables | 0.000006 | 0.000000 | 0.000000 |
| Starting cleanup | 0.000005 | 0.000000 | 0.000000 |
| Freeing items | 0.000006 | 0.000000 | 0.000000 |
| Updating status | 0.000060 | 0.000000 | 0.000000 |
| Reset for next command | 0.000005 | 0.000000 | 0.000000 |
+-----+-----+-----+-----+
22 rows in set (0.000 sec)

```

- ii. **fetch details of students from the student table whose name is 'wood'.**

```

MariaDB [university]> select *
→ from student
→ where name='wood';
+-----+-----+-----+-----+
| ID | name | dept_name | tot_cred |
+-----+-----+-----+-----+
| 33791 | Wood | Civil Eng. | 92 |
| 39876 | Wood | Accounting | 14 |
| 62054 | Wood | Mech. Eng. | 13 |
| 96085 | Wood | Accounting | 70 |
+-----+-----+-----+-----+
4 rows in set (0.003 sec)

```

```

MariaDB [university]> show profiles ;
+-----+-----+-----+
| Query_ID | Duration | Query |
+-----+-----+-----+
| 1 | 0.00019140 | SET profiling = 1 |
| 2 | 0.00037290 | select * |
| from department |
| where budget>50000 |
| 3 | 0.00277310 | select * |
| from student |
| where name='wood' |
+-----+-----+-----+
3 rows in set (0.000 sec)

```

We can see that Query ID 3 corresponds to our recent query.

```

MariaDB [university]> show profile CPU for query 3;
+-----+-----+-----+-----+
| Status | Duration | CPU_user | CPU_system |
+-----+-----+-----+-----+
| Starting | 0.000094 | 0.000000 | 0.000000 |
| checking permissions | 0.000013 | 0.000000 | 0.000000 |
| Opening tables | 0.000048 | 0.000000 | 0.000000 |
| After opening tables | 0.000010 | 0.000000 | 0.000000 |
| System lock | 0.000010 | 0.000000 | 0.000000 |
| table lock | 0.000015 | 0.000000 | 0.000000 |
| init | 0.000073 | 0.000000 | 0.000000 |
| Optimizing | 0.000041 | 0.000000 | 0.000000 |
| Statistics | 0.000073 | 0.000000 | 0.000000 |
| Preparing | 0.000043 | 0.000000 | 0.000000 |
| Executing | 0.000012 | 0.000000 | 0.000000 |
| Sending data | 0.002033 | 0.015625 | 0.000000 |
| End of update loop | 0.000027 | 0.000000 | 0.000000 |
| Query end | 0.000010 | 0.000000 | 0.000000 |
| Commit | 0.000014 | 0.000000 | 0.000000 |
| closing tables | 0.000011 | 0.000000 | 0.000000 |
| Unlocking tables | 0.000008 | 0.000000 | 0.000000 |
| closing tables | 0.000022 | 0.000000 | 0.000000 |
| Starting cleanup | 0.000012 | 0.000000 | 0.000000 |
| Freeing items | 0.000027 | 0.000000 | 0.000000 |
| Updating status | 0.000155 | 0.000000 | 0.000000 |
| Reset for next command | 0.000022 | 0.000000 | 0.000000 |
+-----+-----+-----+-----+
22 rows in set (0.001 sec)

```

We can see that query ID 3 (0.003 sec) took more than query ID 2 (0.001 sec) as the query ID 3 had some notable CPU usage as show in the profile, as the sending data step was quite expensive compare to the previous query.

From these observations, one can see that disk read and write is the main bottleneck as it is slow.

*The memory profile is not yet implemented in MariaDB.*

- C. Create a trigger that will not allow to enter any record into the takes table with a grade that is not used before in any record in the takes table. (2)

```
MariaDB [university]> delimiter //
MariaDB [university]> create trigger valid_grade
→ before insert on takes
→ for each row
→ begin
→ if NEW.grade not in (select distinct grade from takes where grade is not null )
→ then signal sqlstate '45000' set message_text = 'Not a valid grade';
→ end if;
→ end; //
Query OK, 0 rows affected (0.065 sec)

MariaDB [university]> delimiter ;
```

The if condition checks if the newly entering record is not in the existing grades of the takes table.

```
MariaDB [university]> insert into takes
→ values (12345, 'CS-101', 1, 'Spring', 2010, 'D');
ERROR 1644 (45000): Not a valid grade
```

Here since the grade D does not exist, it returns an error saying “not a valid grade”.

```
MariaDB [university]> select distinct grade from takes;
+-----+
| grade |
+-----+
| A      |
| A-     |
| C      |
| B      |
| C+     |
| B-     |
| F      |
| B+     |
| C-     |
| NULL   |
+-----+
10 rows in set (0.001 sec)
```

Here we can note that ‘D’ is not a possible grade.

```
MariaDB [university]> insert into takes
→ values (12345, 'CS-101', 1, 'Spring', 2010, 'A');
Query OK, 1 row affected (0.004 sec)
```

This worked as ‘A’ is a valid grade.