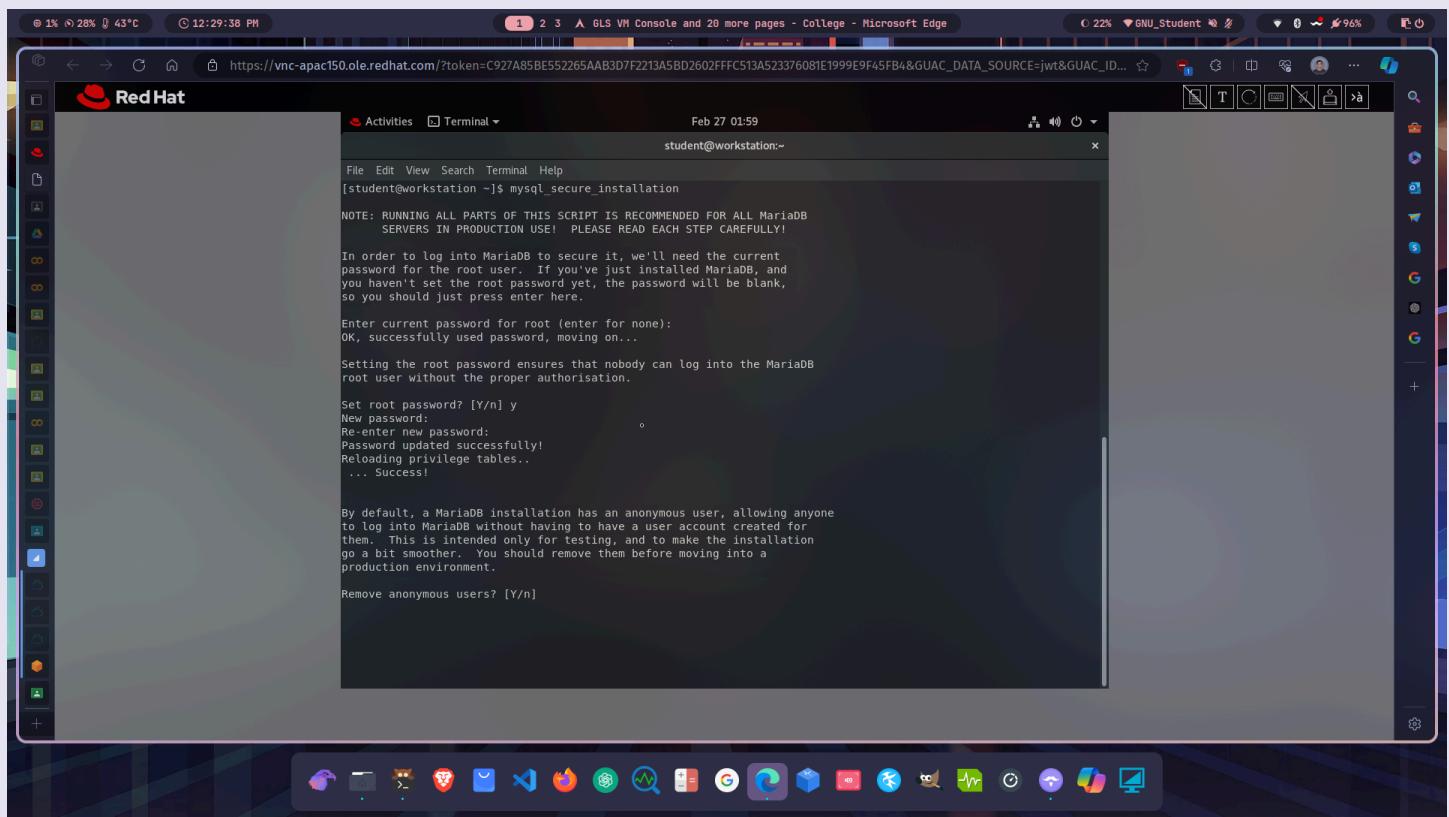


Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 12

1) Configure MariaDB server on server.example.com with below conditions:

- a) Set the root password as "access" and block root access from remote hosts.
- b) Create a user "yourname" by password "password"
- c) Create another user with password as "@yourname"
- d) Only local hosts should have access to MariaDB server.



The screenshot shows a Red Hat Linux desktop environment with a terminal window open. The terminal window title is "Activities Terminal" and the command being run is "[student@workstation ~]\$ mysql_secure_installation". The output of the script is displayed in the terminal, showing the steps to secure the MySQL root password and remove anonymous users.

```
[student@workstation ~]$ mysql_secure_installation
NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

Set root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables...
... Success!

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n]
```

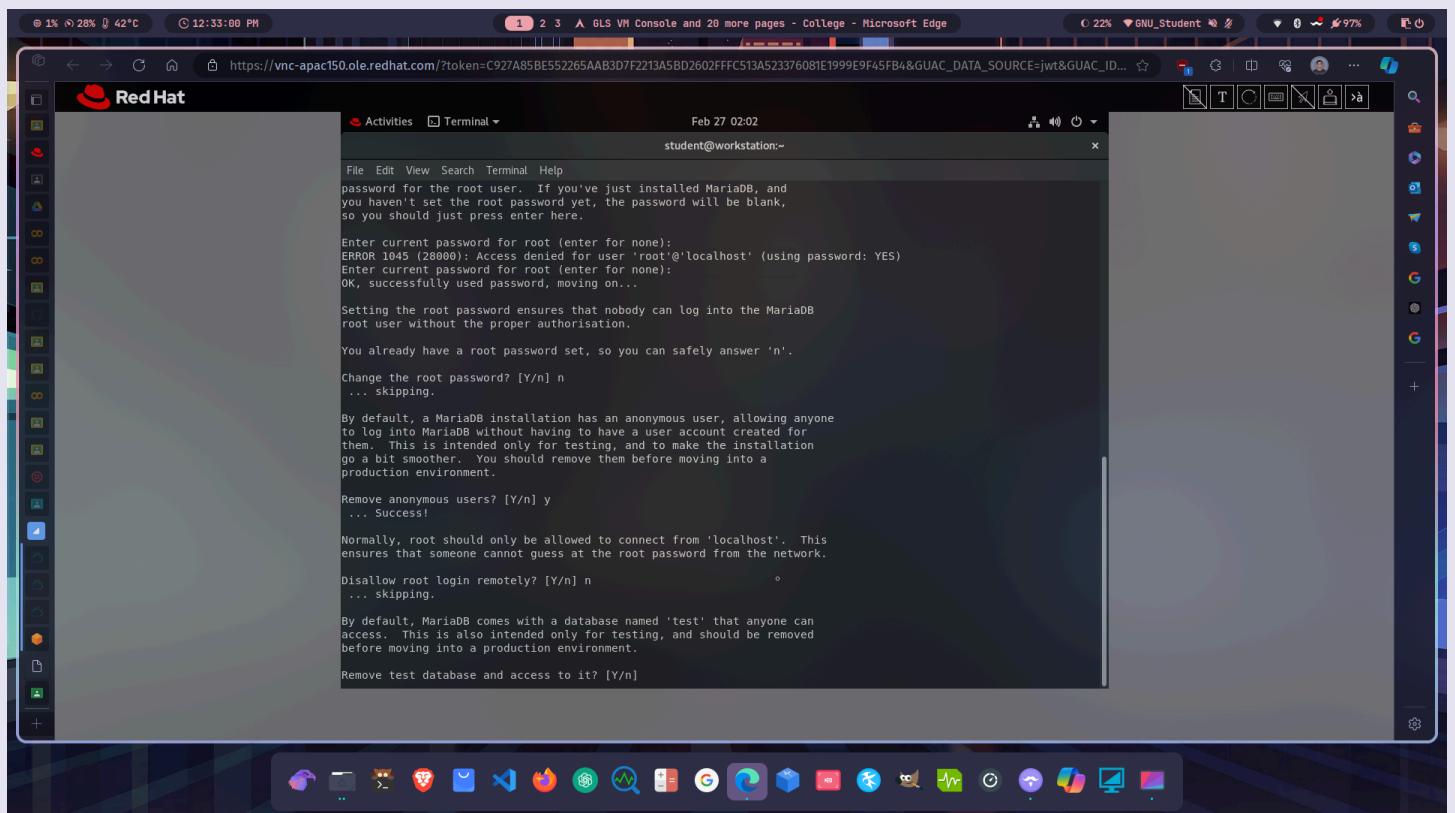
Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 61

ITIM Practical 12

2) Do the Security setting of the Maria DB and ensure that remote access of the db is enabled and anonymous user access is denied.



The screenshot shows a Red Hat Linux desktop environment with a terminal window open. The terminal window title is "Activities Terminal" and the command line shows a MySQL setup script. The script asks for the root password, which is entered as "YES". It then asks if the root password is set, and "n" is entered. It asks if the root password should be changed, and "n" is entered. It then asks about anonymous users, and "y" is entered, followed by "Success!". It asks if root should be allowed to connect from localhost, and "n" is entered. It asks if root login should be disallowed remotely, and "n" is entered. Finally, it asks if the test database should be removed, and "n" is entered. The terminal window is part of a desktop interface with a dock at the bottom containing various application icons.

```
password for the root user. If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

You already have a root password set, so you can safely answer 'n'.

Change the root password? [Y/n]
... skipping.

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] y
... Success!

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] n
... skipping.

By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] n
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 12

3) Create 2 database named - Batch61 and yourname_first4digitofyourenrollemtno. Each database has 3 tables present in it and tables must have atleast 5 entries.

The screenshot shows a dual-monitor setup. The left monitor displays a Google Classroom assignment titled 'Practical 12' for the course '[2024] ITIM - 2CSE605 CBA'. The assignment is due on March 5 and is worth 20 points. It contains 14 numbered steps related to MySQL database configuration and management. The right monitor displays a terminal window titled 'MariaDB 11.3.2' running on mycli 1.27.0. The terminal shows the creation of a database named 'Batch61', the selection of this database, the creation of a table named 'table1' with columns 'id' (auto-increment primary key) and 'name' (VARCHAR(50)), and the insertion of five entries ('Entry 1' through 'Entry 5') into the table. The terminal also shows a 'select * from table1' command which returns the five entries.

```
MariaDB 11.3.2
mycli 1.27.0
Home: http://mycli.net
Bug tracker: https://github.com/dbcli/mycli/issues
Thanks to the contributor - jbruno

MariaDB yash@haribol64 in (none) > CREATE DATABASE Batch61;
Query OK, 1 row affected
Time: 0.001s

MariaDB yash@haribol64 in (none) > USE Batch61;
You are now connected to database "Batch61" as user "yash"
Time: 0.000s

MariaDB yash@haribol64 in Batch61 > CREATE TABLE table1 (
    →     id INT AUTO_INCREMENT PRIMARY KEY,
    →     name VARCHAR(50)
    → );
Query OK, 0 rows affected
Time: 0.014s

MariaDB yash@haribol64 in Batch61 > INSERT INTO table1 (name) VALUES ('Entry 1'), ('Entry 2'), ('Entry 3'), ('Entry 4'), ('Entry 5');
Query OK, 5 rows affected
Time: 0.005s

MariaDB yash@haribol64 in Batch61 > select * from table1;
+----+-----+
| id | name  |
+----+-----+
| 1  | Entry 1 |
| 2  | Entry 2 |
| 3  | Entry 3 |
| 4  | Entry 4 |
| 5  | Entry 5 |
+----+-----+
5 rows in set
Time: 0.006s
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 12

08:21:28

Classroom > [2024] ITIM - 2CSE605
CBA

Practical 12

Neha Rajput • Feb 27
20 points
Due Mar 5

1) Configure MariaDB server on server.example.com with below conditions:
a) Set the root password as "access" and block root access from remote hosts.
b) Create a user "yourname" by password "password"
c) Create another user with password as "@yourname"
d) Only local hosts should have access to MariaDB server.

2) Do the Security setting of the Maria DB and ensure that remote access of the db is enabled and anonymous user access is denied.

3) Create 2 database named - Batch61 and yourname_first4digitofyourenrollemtnto. Each database has 3 tables present in it and tables must have atleast 5 entries.

4) One user must be having the access to both the database while another user should have the read access to one database and read and write access to another database.

5) Create a third user and that user should have only the access to two tables present in Batch61 database.

6) Demonstrate how to check the privilege details of users. (show the privileges assigned to all the users)

7) Demonstrate how to revoke a single permission of a user

8) Set up a default secure MariaDB database called "result" with a user "yourname"with all privileges. In this database, create one simple table with name "student" that allow to store names varchar(20) and their marks int(10). Enter two students with their marks.

9) Demonstrate how to update the records of the table using any user except root user.

10) Delete the create user

11) Backup the database with mysqldump to /root/result.dump.

12) Create another user and provide the privilege to just update the information.

MariaDB yash@haribol64 in Batch61 >
→ CREATE TABLE table2 (
→ → id INT AUTO_INCREMENT PRIMARY KEY,
→ → name VARCHAR(50)
→);
Query OK, 0 rows affected
Time: 0.012s

MariaDB yash@haribol64 in Batch61 > INSERT INTO table2 (name) VALUES ('Entry 1'), ('Entry 2'), ('Entry 3'), ('Entry 4'), ('Entry 5');
Query OK, 5 rows affected
Time: 0.005s

MariaDB yash@haribol64 in Batch61 > select * from table2;

id	name
1	Entry 1
2	Entry 2
3	Entry 3
4	Entry 4
5	Entry 5

5 rows in set
Time: 0.006s

MariaDB yash@haribol64 in Batch61 >

MariaDB yash@haribol64 in Batch61 >

08:22:01

Classroom > [2024] ITIM - 2CSE605
CBA

Practical 12

Neha Rajput • Feb 27
20 points
Due Mar 5

1) Configure MariaDB server on server.example.com with below conditions:
a) Set the root password as "access" and block root access from remote hosts.
b) Create a user "yourname" by password "password"
c) Create another user with password as "@yourname"
d) Only local hosts should have access to MariaDB server.

2) Do the Security setting of the Maria DB and ensure that remote access of the db is enabled and anonymous user access is denied.

3) Create 2 database named - Batch61 and yourname_first4digitofyourenrollemtnto. Each database has 3 tables present in it and tables must have atleast 5 entries.

4) One user must be having the access to both the database while another user should have the read access to one database and read and write access to another database.

5) Create a third user and that user should have only the access to two tables present in Batch61 database.

6) Demonstrate how to check the privilege details of users. (show the privileges assigned to all the users)

7) Demonstrate how to revoke a single permission of a user

8) Set up a default secure MariaDB database called "result" with a user "yourname"with all privileges. In this database, create one simple table with name "student" that allow to store names varchar(20) and their marks int(10). Enter two students with their marks.

9) Demonstrate how to update the records of the table using any user except root user.

10) Delete the create user

11) Backup the database with mysqldump to /root/result.dump.

12) Create another user and provide the privilege to just update the information.

MariaDB yash@haribol64 in Batch61 >
→ CREATE TABLE table3 (
→ → id INT AUTO_INCREMENT PRIMARY KEY,
→ → name VARCHAR(50)
→);
Query OK, 0 rows affected
Time: 0.012s

MariaDB yash@haribol64 in Batch61 > INSERT INTO table3 (name) VALUES ('Entry 1'), ('Entry 2'), ('Entry 3'), ('Entry 4'), ('Entry 5');
Query OK, 5 rows affected
Time: 0.005s

MariaDB yash@haribol64 in Batch61 > select * from table3;

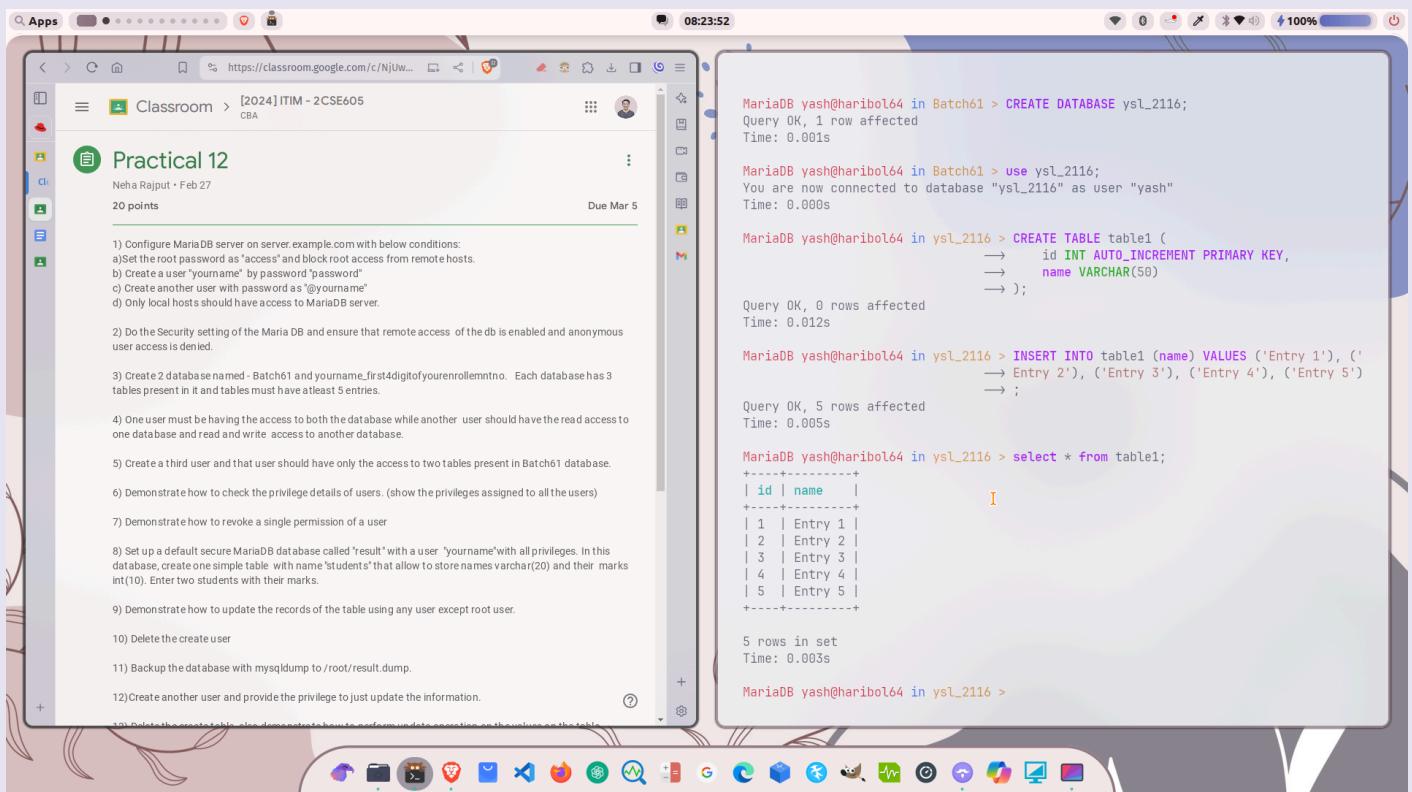
id	name
1	Entry 1
2	Entry 2
3	Entry 3
4	Entry 4
5	Entry 5

5 rows in set
Time: 0.003s

MariaDB yash@haribol64 in Batch61 >

(Semi-colon [] will end the line) [F3] Multiline: ON

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 12



```
MariaDB yash@haribol64 in Batch61 > CREATE DATABASE ysl_2116;
Query OK, 1 row affected
Time: 0.001s

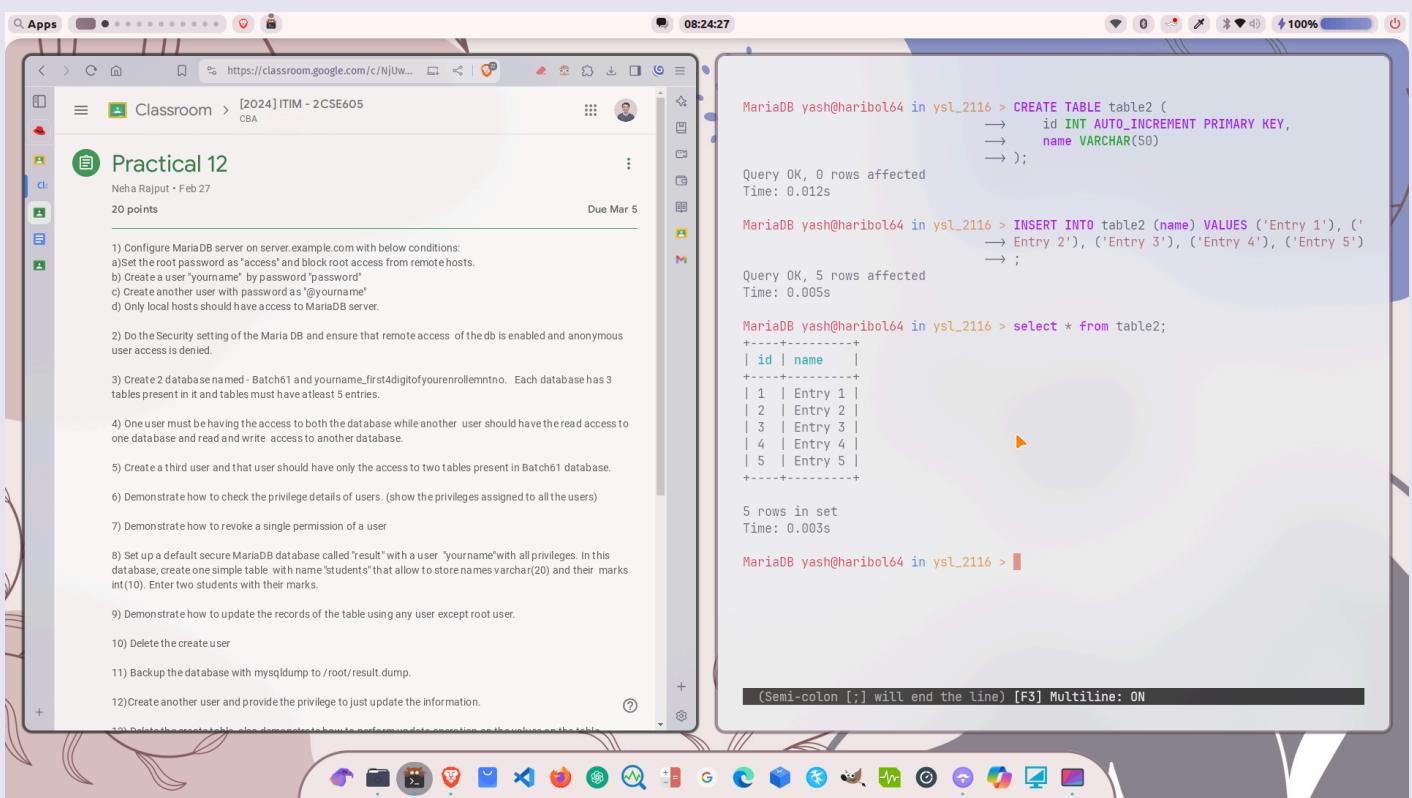
MariaDB yash@haribol64 in Batch61 > use ysl_2116;
You are now connected to database "ysl_2116" as user "yash"
Time: 0.000s

MariaDB yash@haribol64 in ysl_2116 > CREATE TABLE table1 (
    →     id INT AUTO_INCREMENT PRIMARY KEY,
    →     name VARCHAR(50)
    → );
Query OK, 0 rows affected
Time: 0.012s

MariaDB yash@haribol64 in ysl_2116 > INSERT INTO table1 (name) VALUES ('Entry 1'), ('Entry 2'), ('Entry 3'), ('Entry 4'), ('Entry 5')
Query OK, 5 rows affected
Time: 0.005s

MariaDB yash@haribol64 in ysl_2116 > select * from table1;
+----+-----+
| id | name  |
+----+-----+
| 1  | Entry 1 |
| 2  | Entry 2 |
| 3  | Entry 3 |
| 4  | Entry 4 |
| 5  | Entry 5 |
+----+-----+
5 rows in set
Time: 0.003s

MariaDB yash@haribol64 in ysl_2116 >
```



```
MariaDB yash@haribol64 in ysl_2116 > CREATE TABLE table2 (
    →     id INT AUTO_INCREMENT PRIMARY KEY,
    →     name VARCHAR(50)
    → );
Query OK, 0 rows affected
Time: 0.012s

MariaDB yash@haribol64 in ysl_2116 > INSERT INTO table2 (name) VALUES ('Entry 1'), ('Entry 2'), ('Entry 3'), ('Entry 4'), ('Entry 5')
Query OK, 5 rows affected
Time: 0.005s

MariaDB yash@haribol64 in ysl_2116 > select * from table2;
+----+-----+
| id | name  |
+----+-----+
| 1  | Entry 1 |
| 2  | Entry 2 |
| 3  | Entry 3 |
| 4  | Entry 4 |
| 5  | Entry 5 |
+----+-----+
5 rows in set
Time: 0.003s

MariaDB yash@haribol64 in ysl_2116 >
```

(Semi-colon [] will end the line) [F3] Multiline: ON

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 12

The screenshot shows a dual-monitor setup. The left monitor displays a Google Classroom assignment titled 'Practical 12' for the course '[2024] ITIM - 2CSE605'. The assignment is due on March 5 and has 20 points available. It contains 12 numbered tasks related to MySQL database management. The right monitor displays a terminal window with MariaDB commands and their output. The commands include creating a table named 'table3' with an auto-incrementing primary key 'id' and a column 'name' of type VARCHAR(50). Then, five entries ('Entry 1' through 'Entry 5') are inserted into the table. Finally, a 'select * from table3' query is run, showing the five entries in a tabular format.

```
MariaDB yash@haribol64 in ysl_2116 > CREATE TABLE table3 (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(50)
);

Query OK, 0 rows affected
Time: 0.012s

MariaDB yash@haribol64 in ysl_2116 > INSERT INTO table3 (name) VALUES ('Entry 1'), ('Entry 2'), ('Entry 3'), ('Entry 4'), ('Entry 5');

Query OK, 5 rows affected
Time: 0.005s

MariaDB yash@haribol64 in ysl_2116 > select * from table3;
+----+-----+
| id | name  |
+----+-----+
| 1  | Entry 1 |
| 2  | Entry 2 |
| 3  | Entry 3 |
| 4  | Entry 4 |
| 5  | Entry 5 |
+----+-----+

5 rows in set
Time: 0.003s

MariaDB yash@haribol64 in ysl_2116 > 
```

(Semi-colon [;] will end the line) [F3] Multiline: ON

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 12

4) One user must be having the access to both the database while another user should have the read access to one database and read and write access to another database.

```
MariaDB 11.3.2
mycli 1.27.0
Home: http://mycli.net
Bug tracker: https://github.com/dbcli/mycli/issues
Thanks to the contributor - bitkeen

MariaDB yash@haribol64 in (none) > CREATE USER 'user1'@'localhost' IDENTIFIED BY 'psw'
→ d';
Query OK, 0 rows affected
Time: 0.005s

MariaDB yash@haribol64 in (none) > GRANT ALL PRIVILEGES ON Batch61.* TO 'user1'@'localhost';
(1044, "Access denied for user 'yash'@'localhost' to database 'Batch61'")

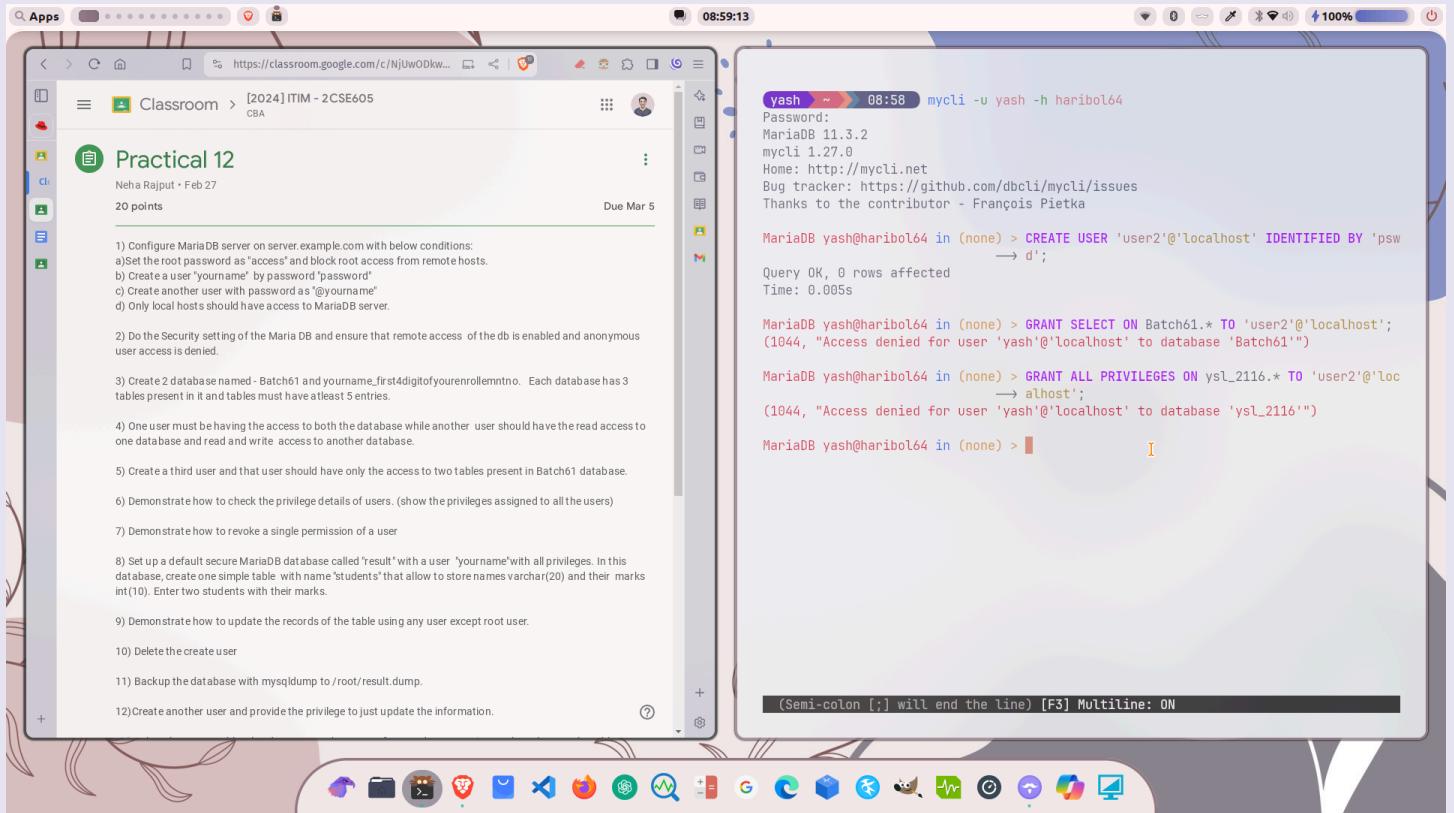
MariaDB yash@haribol64 in (none) >

MariaDB yash@haribol64 in (none) > GRANT ALL PRIVILEGES ON ysl_2116.* TO 'user1'@'localhost';
(1044, "Access denied for user 'yash'@'localhost' to database 'ysl_2116'")

MariaDB yash@haribol64 in (none) > 
```

(Semi-colon [;] will end the line) [F3] Multiline: ON

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 12



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 12

5) Create a third user and that user should have only the access to two tables present in Batch61 database.

The screenshot shows a Mac desktop with a browser window open to a Google Classroom assignment titled "Practical 12". The assignment details 12 tasks related to MySQL configuration and user management. To the right of the browser is a terminal window displaying MySQL command-line session logs. The logs show the creation of three users: "user2", "user2", and "user3", and the granting of specific privileges (SELECT) on two tables ("Batch61.table1" and "Batch61.table2") to the "user3" user.

```
MariaDB yash@haribol64 in (none) > CREATE USER 'user2'@'localhost' IDENTIFIED BY 'psw' → d';  
Query OK, 0 rows affected  
Time: 0.005s  
  
MariaDB yash@haribol64 in (none) > GRANT SELECT ON Batch61.* TO 'user2'@'localhost';  
(1044, "Access denied for user 'yash'@'localhost' to database 'Batch61'")  
  
MariaDB yash@haribol64 in (none) > GRANT ALL PRIVILEGES ON ysl_2116.* TO 'user2'@'loc  
→ alhost';  
(1044, "Access denied for user 'yash'@'localhost' to database 'ysl_2116'")  
  
MariaDB yash@haribol64 in (none) > CREATE USER 'user3'@'localhost' IDENTIFIED BY 'psw'  
→ d';  
Query OK, 0 rows affected  
Time: 0.005s  
  
MariaDB yash@haribol64 in (none) > GRANT SELECT ON Batch61.table1 TO 'user3'@'localho  
→ st';  
(1142, "GRANT command denied to user 'yash'@'localhost' for table 'Batch61`.`table1`")  
  
MariaDB yash@haribol64 in (none) > GRANT SELECT ON Batch61.table2 TO 'user3'@'localho  
→ st';  
(1142, "GRANT command denied to user 'yash'@'localhost' for table 'Batch61`.`table2`")  
  
MariaDB yash@haribol64 in (none) > ;  
  
(Semi-colon [;] will end the line) [F3] Multiline: ON
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 12

6) Demonstrate how to check the privilege details of users. (show the privileges assigned to all the users)

The screenshot shows a Mac desktop with a browser window open to a Google Classroom assignment titled "Practical 12". The assignment details 14 tasks related to MySQL configuration and management. To the right of the browser is a terminal window displaying three separate MySQL grant commands for users "user1", "user2", and "user3" on the localhost. Each command shows the GRANT USAGE privilege being granted to the respective user.

```
MariaDB yash@haribol64 in (none) > SHOW GRANTS FOR 'user1'@'localhost';
+-----+
| Grants for user1@localhost
+-----+
| GRANT USAGE ON *.* TO 'user1'@'localhost' IDENTIFIED BY PASSWORD '*924DEB3C0C6DBA65
C8D67DB4F5DA7B4074AF518E' |
+-----+
1 row in set
Time: 0.006s

MariaDB yash@haribol64 in (none) > SHOW GRANTS FOR 'user2'@'localhost';
+-----+
| Grants for user2@localhost
+-----+
| GRANT USAGE ON *.* TO 'user2'@'localhost' IDENTIFIED BY PASSWORD '*924DEB3C0C6DBA65
C8D67DB4F5DA7B4074AF518E' |
+-----+
1 row in set
Time: 0.003s

MariaDB yash@haribol64 in (none) > SHOW GRANTS FOR 'user3'@'localhost';
+-----+
| Grants for user3@localhost
+-----+
| GRANT USAGE ON *.* TO 'user3'@'localhost' IDENTIFIED BY PASSWORD '*924DEB3C0C6DBA65
C8D67DB4F5DA7B4074AF518E' |
+-----+
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 12

7) Demonstrate how to revoke a single permission of a user

The screenshot shows a dual-monitor setup. The left monitor displays a Google Classroom assignment titled "Batch61 Assignment". The assignment contains 14 numbered tasks related to MySQL database management. Task 7 asks to demonstrate how to revoke a single permission of a user. Below the assignment is a "Your work" section with a "Missing" status and a "Mark as done" button. The right monitor displays a MySQL terminal window titled "MariaDB yash@haribol64 in (none)". The terminal shows the command "SHOW GRANTS FOR 'user3'@'localhost';" followed by the output of grants for user3. Then, the command "REVOKE INSERT ON ysl_2116.* FROM 'user1'@'localhost';" is run, resulting in an "Access denied" error message. The MySQL prompt is visible at the bottom of the terminal window.

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 61

ITIM Practical 12

8) Set up a default secure MariaDB database called "result" with a user "yourname"with all privileges. In this database, create one simple table with name "students" that allow to store names varchar(20) and their marks int(10). Enter two students with their marks.

The screenshot shows a Mac desktop with two windows open. On the left is a browser window for Google Classroom with a list of 14 tasks. On the right is a terminal window showing the MySQL command-line interface (CLI).

Tasks from Google Classroom:

- 2) Do the Security setting of the Maria DB and ensure that remote access of the db is enabled and anonymous user access is denied.
- 3) Create 2 database named - Batch61 and yourname_first4digitofyourenrollemtno. Each database has 3 tables present in it and tables must have atleast 5 entries.
- 4) One user must be having the access to both the database while another user should have the read access to one database and read and write access to another database.
- 5) Create a third user and that user should have only the access to two tables present in Batch61 database.
- 6) Demonstrate how to check the privilege details of users. (show the privileges assigned to all the users)
- 7) Demonstrate how to revoke a single permission of a user
- 8) Set up a default secure MariaDB database called "result" with a user "yourname"with all privileges. In this database, create one simple table with name "students" that allow to store names varchar(20) and their marks int(10). Enter two students with their marks.
- 9) Demonstrate how to update the records of the table using any user except root user.
- 10) Delete the create user
- 11) Backup the database with mysqldump to /root/result.dump.
- 12) Create another user and provide the privilege to just update the information.
- 13) Delete the create table, also demonstrate how to perform update operation on the values on the table.
- 14) Demonstrate how to delete the database along with the table present in it.

MySQL CLI Session:

```
MariaDB yash@haribol64 in (none) > REVOKE INSERT ON ysl_2116.* FROM 'user1'@'localhost';
MariaDB yash@haribol64 in (none) > CREATE DATABASE IF NOT EXISTS result;
Query OK, 1 row affected
Time: 0.001s

MariaDB yash@haribol64 in (none) > use result;
You are now connected to database "result" as user "yash"
Time: 0.000s

MariaDB yash@haribol64 in result > CREATE TABLE students (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(20),
    marks INT(10)
);
Query OK, 0 rows affected
Time: 0.012s

MariaDB yash@haribol64 in result > INSERT INTO students (name, marks) VALUES ('John', 85), ('Alice', 92);
Query OK, 2 rows affected
Time: 0.005s

MariaDB yash@haribol64 in result >
```

(Semi-colon [;] will end the line) [F3] Multiline: ON

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 61

ITIM Practical 12

The screenshot shows a Linux desktop environment with a terminal window and a web browser window.

Terminal Window:

```
(1044, "Access denied for user 'yash'@'localhost' to database 'ysl_2116'")  
MariaDB yash@haribol64 in (none) > CREATE DATABASE IF NOT EXISTS result;  
Query OK, 1 row affected  
Time: 0.001s  
  
MariaDB yash@haribol64 in (none) > use result;  
You are now connected to database "result" as user "yash"  
Time: 0.000s  
  
MariaDB yash@haribol64 in result > CREATE TABLE students (  
    → id INT AUTO_INCREMENT PRIMARY KEY,  
    → name VARCHAR(20),  
    → marks INT(10)  
    → );  
Query OK, 0 rows affected  
Time: 0.012s  
  
MariaDB yash@haribol64 in result > INSERT INTO students (name, marks) VALUES ('John',  
    → 85), ('Alice', 92);  
Query OK, 2 rows affected  
Time: 0.005s  
  
MariaDB yash@haribol64 in result > GRANT ALL PRIVILEGES ON result.* TO 'yash'@'haribo  
    → l64';  
(1044, "Access denied for user 'yash'@'localhost' to database 'result'")  
MariaDB yash@haribol64 in result > |  
  
(Semi-colon [;] will end the line) [F3] Multiline: ON
```

Google Classroom Assignment:

Your work

+ Add or create

Mark as done

Private comments

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 12

9) Demonstrate how to update the records of the table using any user except root user.

The screenshot shows a Mac desktop with two windows open. On the left is a browser window displaying a Google Classroom assignment titled 'Batch61'. The assignment contains 14 numbered steps related to MySQL database management. Step 9 asks to demonstrate updating records using a user other than root. On the right is a terminal window titled 'Terminal' with the command-line interface for MariaDB. The user 'yash@haribol64' is logged in. The terminal shows the following session:

```
→ 85, ('Alice', 92);
Query OK, 2 rows affected
Time: 0.005s

MariaDB yash@haribol64 in result > GRANT ALL PRIVILEGES ON result.* TO 'yash'@'haribo
→ l64';
(1044, "Access denied for user 'yash'@'localhost' to database 'result'")

MariaDB yash@haribol64 in result >
MariaDB yash@haribol64 in result >
MariaDB yash@haribol64 in result > UPDATE students SET marks = 90 WHERE name = 'John'
→ ;
Query OK, 1 row affected
Time: 0.002s

MariaDB yash@haribol64 in result > select * from students;
+---+---+
| id | name | marks |
+---+---+
| 1 | John | 90 |
| 2 | Alice | 92 |
+---+---+
2 rows in set
Time: 0.003s

MariaDB yash@haribol64 in result > I
I

(Semi-colon [;] will end the line) [F3] Multiline: ON
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 12

10) Delete the create user

The screenshot shows a dual-pane desktop environment. On the left, a terminal window displays a MySQL session with the user 'yash@haribol64'. The user runs three consecutive 'DROP USER' commands to delete 'user3', 'user2', and 'user1' from the 'localhost' host. Each command is preceded by a warning about being destructive and asking for confirmation ('y/n'). The responses are 'y' for yes. The MySQL session ends with a prompt 'MariaDB yash@haribol64 in result >'. On the right, a Google Classroom assignment titled 'Practical 12' is open. The assignment details 14 tasks related to MySQL database management. Task 10 specifically asks to 'Delete the create user'. Below the task list is a 'Your work' section with a 'Missing' status, a 'Mark as done' button, and a note: '(Semi-colon [;] will end the line) [F3] Multiline: ON'.

2 rows in set
Time: 0.003s
MariaDB yash@haribol64 in result >

MariaDB yash@haribol64 in result > **DROP USER 'user3'@'localhost';**
You're about to run a destructive command.
Do you want to proceed? (y/n): y
Your call!
Query OK, 0 rows affected
Time: 0.005s

MariaDB yash@haribol64 in result > **DROP USER 'user2'@'localhost';**
You're about to run a destructive command.
Do you want to proceed? (y/n): y
Your call!
Query OK, 0 rows affected
Time: 0.023s

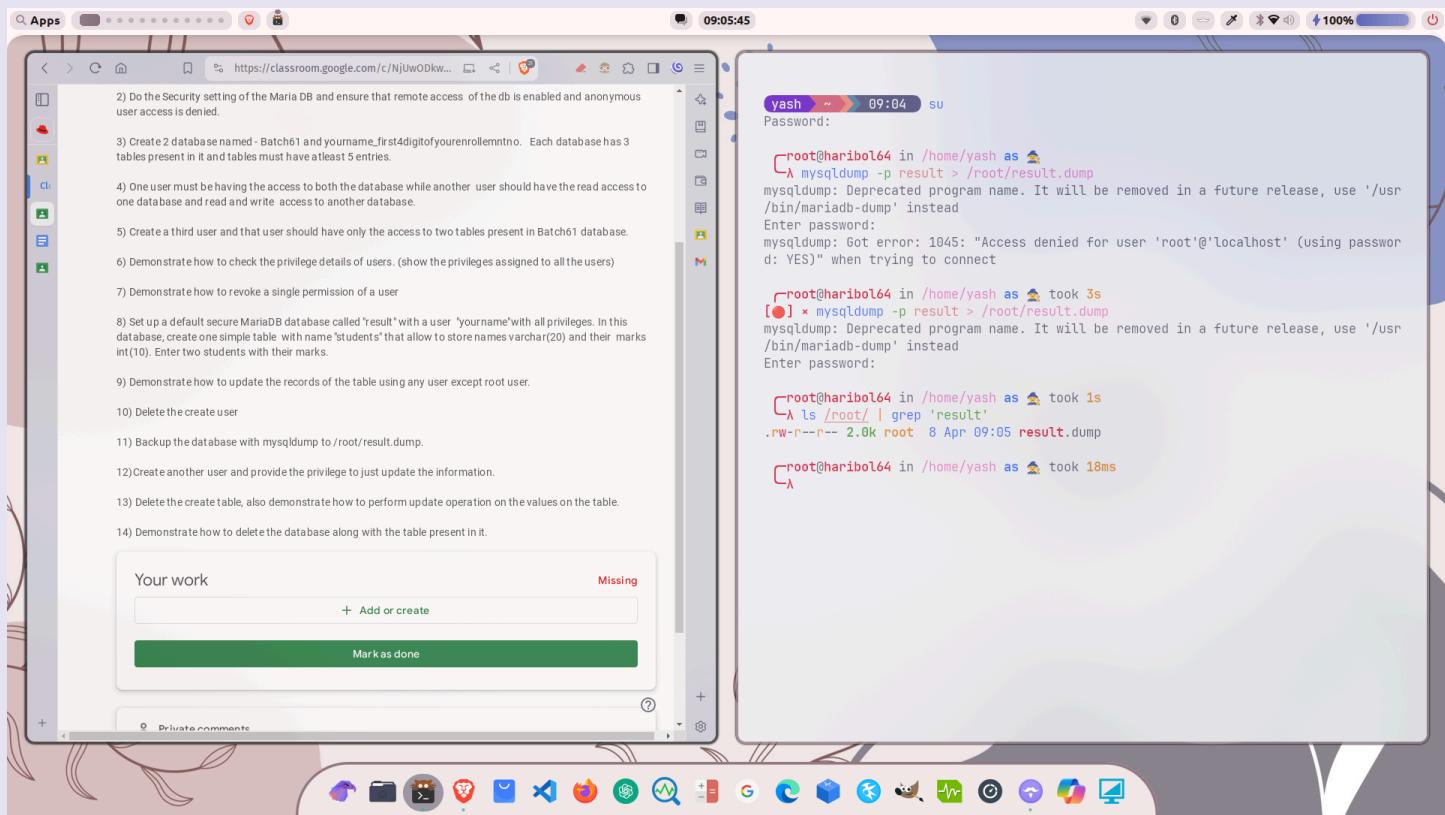
MariaDB yash@haribol64 in result > **DROP USER 'user1'@'localhost';**
You're about to run a destructive command.
Do you want to proceed? (y/n): y
Your call!
Query OK, 0 rows affected
Time: 0.002s

MariaDB yash@haribol64 in result > **[REDACTED]**

(Semi-colon [;] will end the line) [F3] Multiline: ON

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 12

11) Backup the database with mysqldump to /root/result.dump.



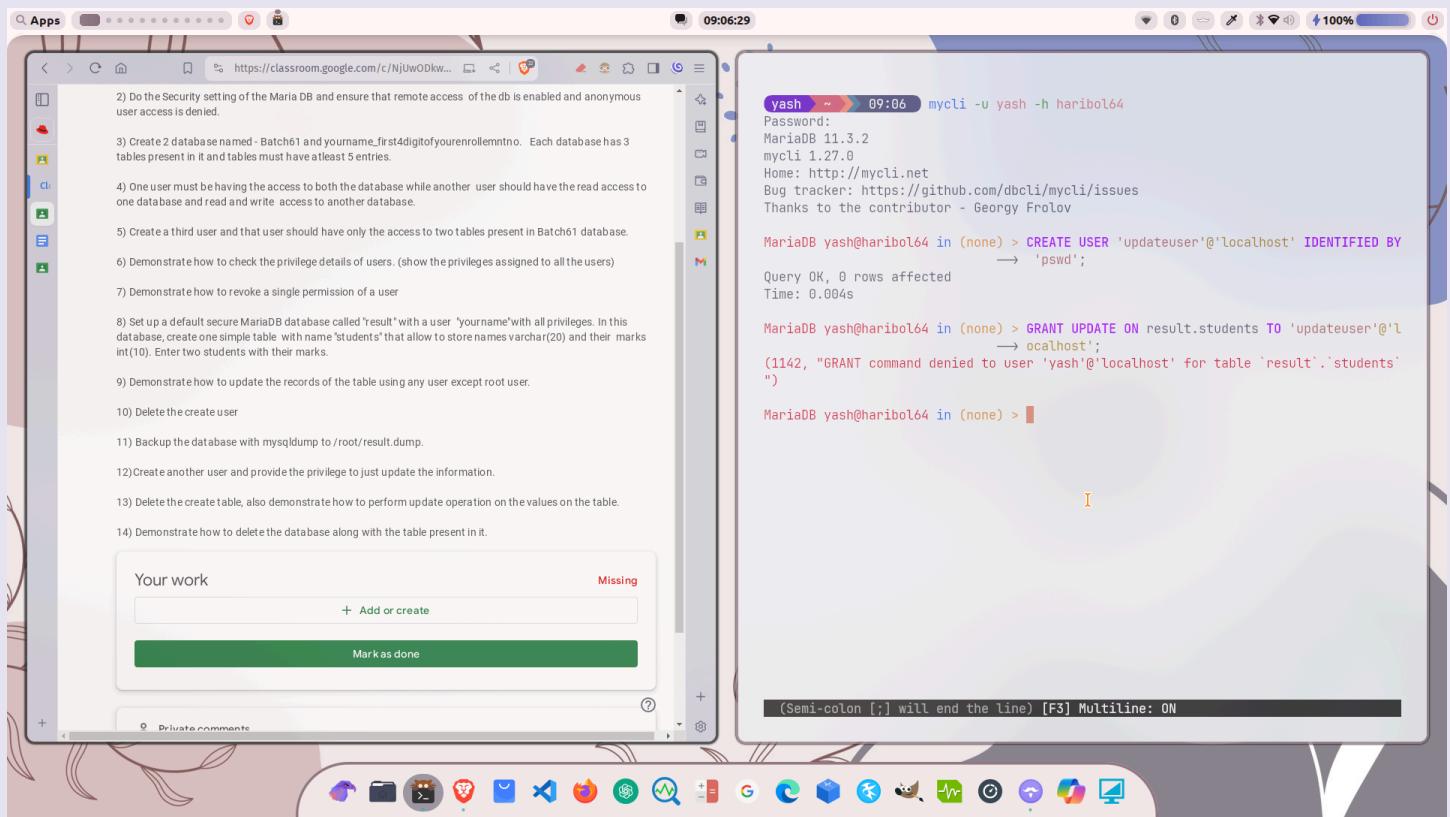
Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 61

ITIM Practical 12

12) Create another user and provide the privilege to just update the information.



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 12

13) Delete the create table, also demonstrate how to perform update operation on the values on the table.

The screenshot shows a Mac desktop with two windows open. On the left is a Google Classroom assignment titled 'Batch61' with a list of 14 tasks. Tasks 13 and 14 are highlighted in red. Task 13 says 'Delete the create table, also demonstrate how to perform update operation on the values on the table.' Task 14 says 'Demonstrate how to delete the database along with the table present in it.' On the right is a terminal window showing MySQL queries. The user has created a user 'updateuser' with password 'pswd'. They then grant update privileges on the 'result.students' table to this user. In the next few lines, they drop the 'result.students' table and the 'result' database. The terminal prompt at the bottom indicates 'Multiline: ON'.

```
MariaDB yash@haribol64 in (none) > CREATE USER 'updateuser'@'localhost' IDENTIFIED BY
                                         → 'pswd';
Query OK, 0 rows affected
Time: 0.004s

MariaDB yash@haribol64 in (none) > GRANT UPDATE ON result.students TO 'updateuser'@'localhost';
                                         →
(1142, "GRANT command denied to user 'yash'@'localhost' for table 'result'.`students`")
MariaDB yash@haribol64 in (none) >
MariaDB yash@haribol64 in (none) >

MariaDB yash@haribol64 in (none) > DROP TABLE result.students;
You're about to run a destructive command.
Do you want to proceed? (y/n): y
Your call!
Query OK, 0 rows affected
Time: 0.011s

MariaDB yash@haribol64 in (none) > DROP DATABASE ysl_2116;
You're about to run a destructive command.
Do you want to proceed? (y/n): y
Your call!
Query OK, 3 rows affected
Time: 0.023s

MariaDB yash@haribol64 in (none) > 
```

(Semi-colon [;] will end the line) [F3] Multiline: ON

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
ITIM Practical 12

14) Demonstrate how to delete the database along with the table present in it.

The screenshot shows a dual-monitor setup. The left monitor displays a Google Classroom assignment titled '14) Demonstrate how to delete the database along with the table present in it.' It lists 14 numbered steps related to MySQL database management. The right monitor shows a MySQL terminal window with the following session:

```
MariaDB yash@haribol64 in (none) > DROP TABLE result.students;
You're about to run a destructive command.
Do you want to proceed? (y/n): y
Your call!
Query OK, 0 rows affected
Time: 0.011s

MariaDB yash@haribol64 in (none) > DROP DATABASE ysl_2116;
You're about to run a destructive command.
Do you want to proceed? (y/n): y
Your call!
Query OK, 3 rows affected
Time: 0.023s

MariaDB yash@haribol64 in (none) > DROP DATABASE Batch61;
You're about to run a destructive command.
Do you want to proceed? (y/n): y
Your call!
Query OK, 3 rows affected
Time: 0.022s

MariaDB yash@haribol64 in (none) > DROP DATABASE result;
You're about to run a destructive command.
Do you want to proceed? (y/n): y
Your call!
Query OK, 0 rows affected
Time: 0.007s

MariaDB yash@haribol64 in (none) >
```

The MySQL terminal also includes a status bar at the bottom with the message '(Semi-colon [;] will end the line) [F3] Multiline: ON' and 'Refreshing completions.'