RAFIQA

MAHARANI PUTRI SIREGAR

L 200 154 010
X class

Database Management Systems

ASSIGNMENT 2

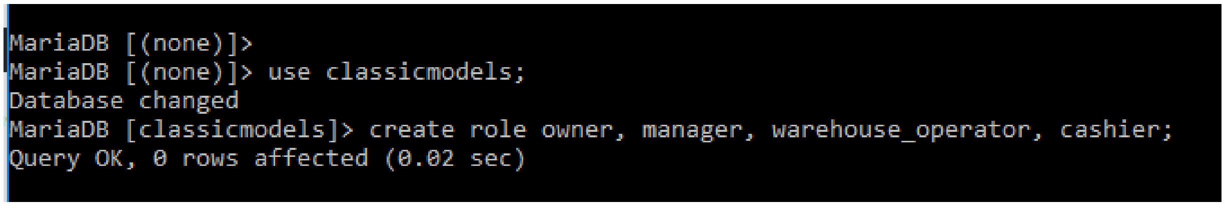
Rafiqa Maharani Putri Siregar – L 200 154 010
Database Management System
X Class

ASSIGNMENT 2

Create role based on ‘classicmodels’ database:

At least 3 group: owner, manager, operator.

1. Go to database ‘classicmodels’ and create roles. In this case there are 4 roles: **owner, manager, warehouse_operator, cashier**.

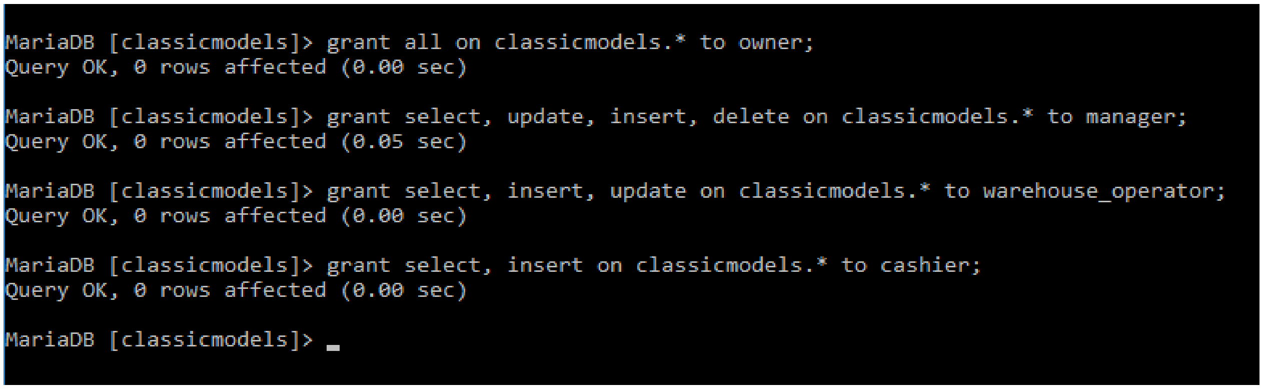


```
MariaDB [(none)]>
MariaDB [(none)]> use classicmodels;
Database changed
MariaDB [classicmodels]> create role owner, manager, warehouse_operator, cashier;
Query OK, 0 rows affected (0.02 sec)
```

Query:

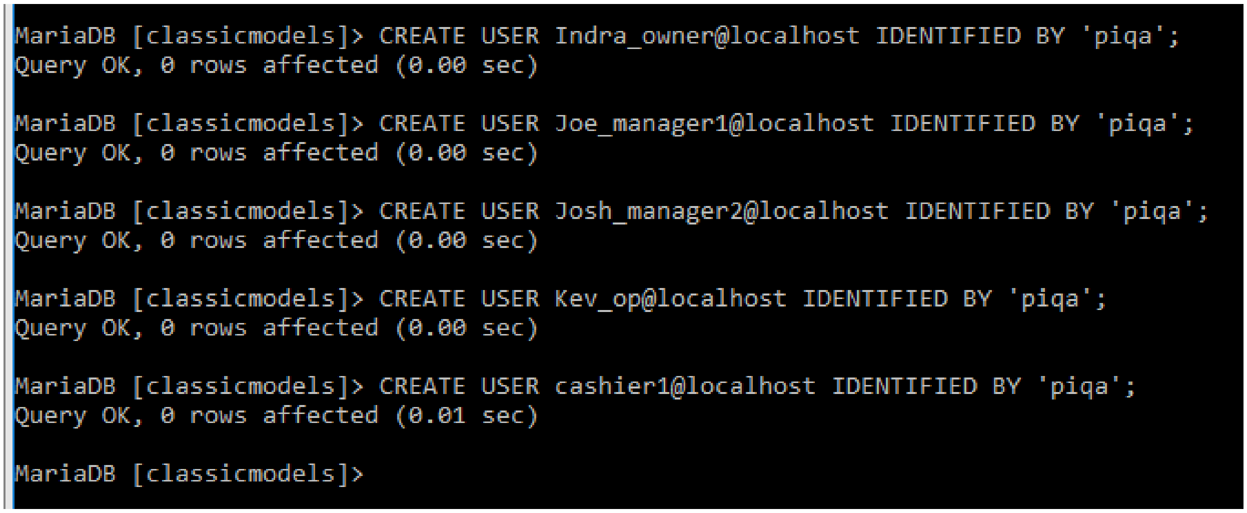
```
CREATE ROLE owner, manager, warehouse_operator, cashier;
```

2. Give GRANT to each role that has been made before, each role has different GRANT.
   - owner get all grant
   - manager only can do SELECT, UPDATE, INSERT, and DELETE
   - warehouse_operator only can do SELECT, INSERT, and UPDATE
   - cashier only can do SELECT and INSERT



```
MariaDB [classicmodels]> grant all on classicmodels.* to owner;
Query OK, 0 rows affected (0.00 sec)

MariaDB [classicmodels]> grant select, update, insert, delete on classicmodels.* to manager;
Query OK, 0 rows affected (0.05 sec)

MariaDB [classicmodels]> grant select, insert, update on classicmodels.* to warehouse_operator;
Query OK, 0 rows affected (0.00 sec)

MariaDB [classicmodels]> grant select, insert on classicmodels.* to cashier;
Query OK, 0 rows affected (0.00 sec)

MariaDB [classicmodels]> _
```

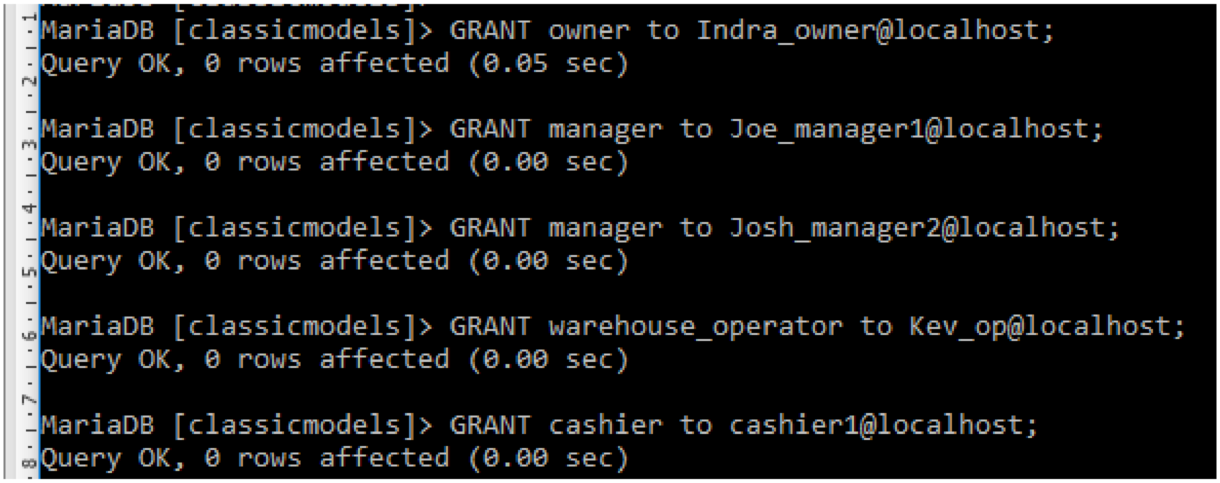
3. The next thing to do is make new user. Using CREATE USER syntax.



```
MariaDB [classicmodels]> CREATE USER Indra_owner@localhost IDENTIFIED BY 'piqa';
Query OK, 0 rows affected (0.00 sec)

MariaDB [classicmodels]> CREATE USER Joe_manager1@localhost IDENTIFIED BY 'piqa';
Query OK, 0 rows affected (0.00 sec)

MariaDB [classicmodels]> CREATE USER Josh_manager2@localhost IDENTIFIED BY 'piqa';
Query OK, 0 rows affected (0.00 sec)

MariaDB [classicmodels]> CREATE USER Kev_op@localhost IDENTIFIED BY 'piqa';
Query OK, 0 rows affected (0.00 sec)

MariaDB [classicmodels]> CREATE USER cashier1@localhost IDENTIFIED BY 'piqa';
Query OK, 0 rows affected (0.01 sec)

MariaDB [classicmodels]>
```

4. To make those new user has its own accessibility towards database, we can use our role. GRANT statement can be used to grant the use of one or more ROLES to one or more USERS.
Syntax:

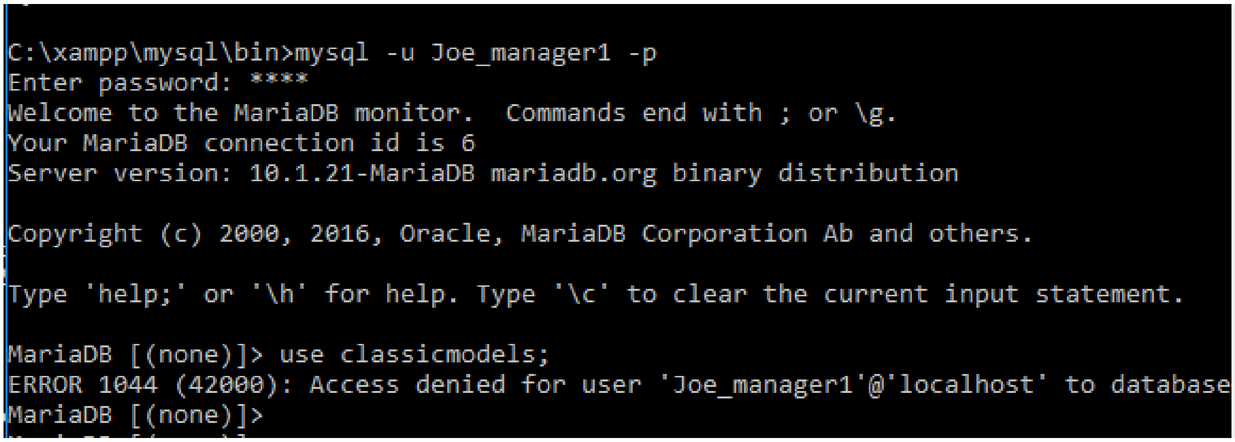
```
GRANT role TO grantee [, grantee2 ….]
```



```
MariaDB [classicmodels]> GRANT owner to Indra_owner@localhost;
Query OK, 0 rows affected (0.05 sec)

MariaDB [classicmodels]> GRANT manager to Joe_manager1@localhost;
Query OK, 0 rows affected (0.00 sec)

MariaDB [classicmodels]> GRANT manager to Josh_manager2@localhost;
Query OK, 0 rows affected (0.00 sec)

MariaDB [classicmodels]> GRANT warehouse_operator to Kev_op@localhost;
Query OK, 0 rows affected (0.00 sec)

MariaDB [classicmodels]> GRANT cashier to cashier1@localhost;
Query OK, 0 rows affected (0.00 sec)
```

5. However the users already been granted by role, the users do not automatically acquire all permissions associated with that role. The solution is by using SET ROLE to activates the role.

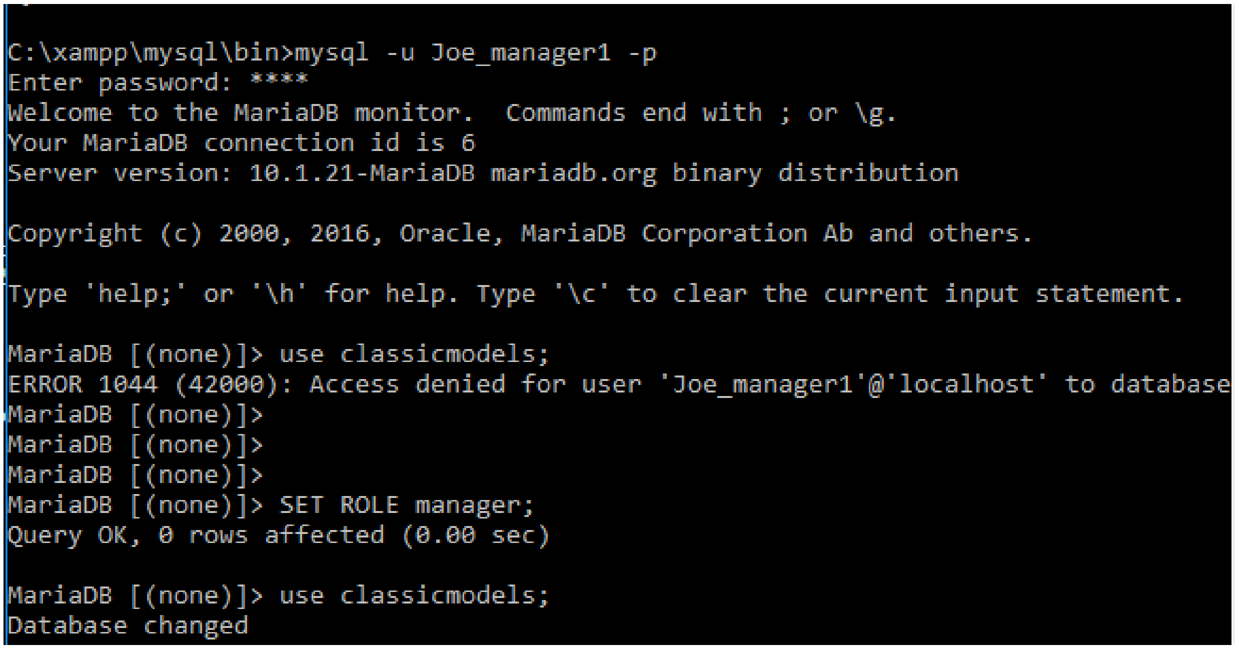
   Take a look to this picture below, user Joe_manager1 can not access database 'classicmodels' because the role is not activated yet.



```
C:\xampp\mysql\bin>mysql -u Joe_manager1 -p
Enter password: ****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 6
Server version: 10.1.21-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use classicmodels;
ERROR 1044 (42000): Access denied for user 'Joe_manager1'@'localhost' to database
MariaDB [(none)]>
```

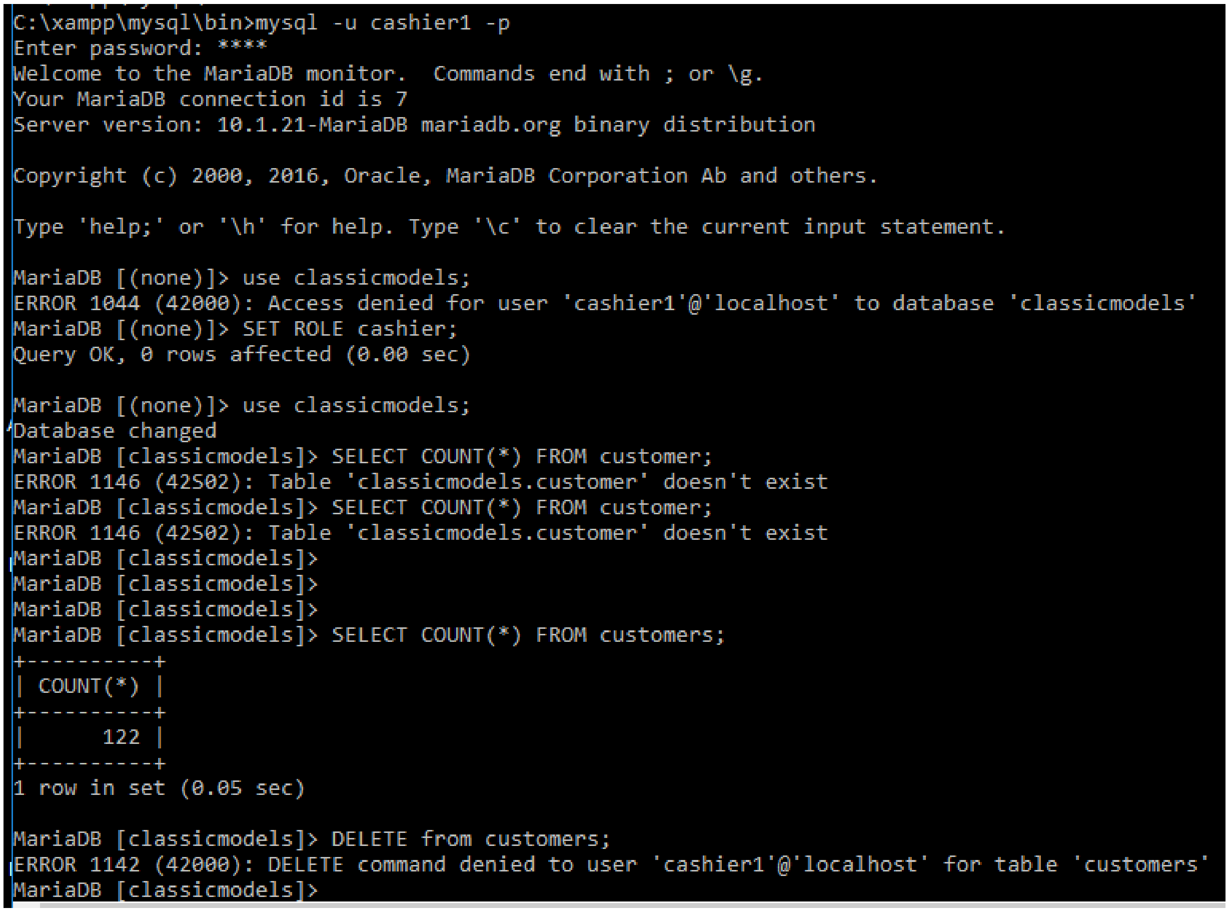
SO, we activate the role, use **SET ROLE name_role;** statement

And user Joe_manager1 can access the database and do any command that be allowed by the role privileges.



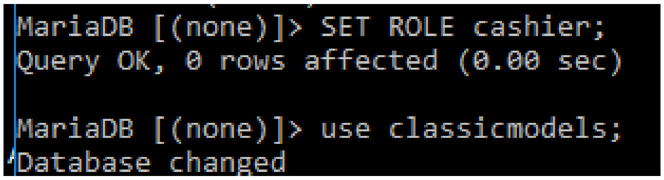
```
C:\xampp\mysql\bin>mysql -u Joe_manager1 -p
Enter password: ****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 6
Server version: 10.1.21-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use classicmodels;
ERROR 1044 (42000): Access denied for user 'Joe_manager1'@'localhost' to database
MariaDB [(none)]>
MariaDB [(none)]>
MariaDB [(none)]>
MariaDB [(none)]> SET ROLE manager;
Query OK, 0 rows affected (0.00 sec)

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

6. Check another user, let use cashier1.



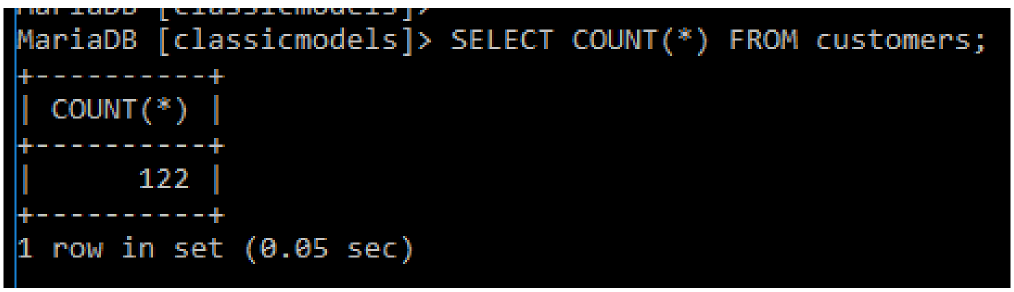
```
C:\xampp\mysql\bin>mysql -u cashier1 -p
Enter password: ****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 7
Server version: 10.1.21-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use classicmodels;
ERROR 1044 (42000): Access denied for user 'cashier1'@'localhost' to database 'classicmodels'
MariaDB [(none)]> SET ROLE cashier;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> use classicmodels;
Database changed
MariaDB [classicmodels]> SELECT COUNT(*) FROM customer;
ERROR 1146 (42S02): Table 'classicmodels.customer' doesn't exist
MariaDB [classicmodels]> SELECT COUNT(*) FROM customer;
ERROR 1146 (42S02): Table 'classicmodels.customer' doesn't exist
MariaDB [classicmodels]>
MariaDB [classicmodels]>
MariaDB [classicmodels]>
MariaDB [classicmodels]> SELECT COUNT(*) FROM customers;
+----------+
| COUNT(*) |
+----------+
|      122 |
+----------+
1 row in set (0.05 sec)

MariaDB [classicmodels]> DELETE from customers;
ERROR 1142 (42000): DELETE command denied to user 'cashier1'@'localhost' for table 'customers'
MariaDB [classicmodels]>
```

7. Type statement **SET ROLE cashier;** to activates the role **cashier**
8. After that, user cashier1 can access database classicmodels.



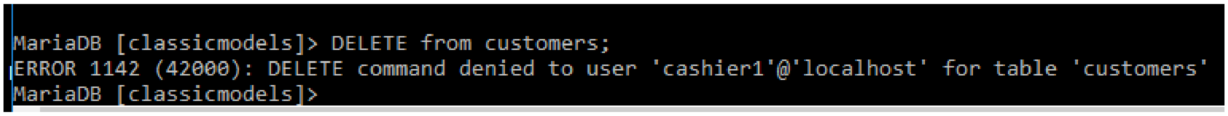
```
MariaDB [(none)]> SET ROLE cashier;
Query OK, 0 rows affected (0.00 sec)

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

9. Check privileges, we already make role cashier only can use SELECT and INSERT.
   So, first check whether the user cashier1 can use SELECT statement.
10. We can check it with **SELECT COUNT(*) FROM customers;** statement
11. The user **cashier1** can use SELECT statement.



```
MariaDB [classicmodels]> SELECT COUNT(*) FROM customers;
+----------+
| COUNT(*) |
+----------+
|      122 |
+----------+
1 row in set (0.05 sec)
```

12. Then check whether cahier1 can use DELETE statement or not. It should be not.
13. And, then yes. Cashier1 can not use DELETE statement.



```
MariaDB [classicmodels]> DELETE from customers;
ERROR 1142 (42000): DELETE command denied to user 'cashier1'@'localhost' for table 'customers'
MariaDB [classicmodels]>
```

This is how make roles and give those roles to some particular user.