A.Y. 2020-2021
Class: SE-ITA/B, Semester: III

Subject: **Structured Query Lab**

Experiment – 8: Perform Authorization using Grant and Revoke.

1. **Aim:** To Implement DCL commands and Perform Authorization using Grant and Revoke.

2. **Objective:**
3. After performing the experiment, the students will be able to write DCL queries for authorization
4. **Outcome:** L303.4: To Formulate query using SQL commands.
5. **Prerequisite:** Understanding data control language commands and basic authorization concepts
6. **Requirements:** PC, Oracle 11g/SQL Server 2008 R2, Microsoft Word, Internet, MySQL, JDK Netbeans,

7. **Pre-Experiment Exercise:**
   Brief Theory :(To be hand written)
   Explain DCL commands with example

8. **Laboratory Exercise**
   A. Procedure:(Refer additional attachment for commands and details
   a. Create Database
   b. Create Tables
   c. Create user(enter username and password)
   d. Transfer privileges using Grant and Revoke
   e. For select, insert , delete

B. **Result/Observation/Program code:** Attach codes and query of commands that are executed

```
--Creating a database;
create database sample;
use sample;

--creating a student table;
create table student(id int primary key, name varchar(20), age int);
--inserting values in student table;
insert into student values(1,'ram',20);
insert into student values(2,'sam',20);
insert into student values(3,'seema',20);
insert into student values(4,'reema',20);
```

```
select * from student;
show tables;
```

MySQL 8.0 Command Line Client

```
mysql> create database sample;
Query OK, 1 row affected (0.03 sec)

mysql> use sample;
Database changed
mysql> create table student(id int primary key, name varchar(20), age int);
Query OK, 0 rows affected (0.14 sec)

mysql> insert into student values(1,'ram',20);
Query OK, 1 row affected (0.02 sec)

mysql> insert into student values(2,'sam',20);
Query OK, 1 row affected (0.01 sec)

mysql> insert into student values(3,'seema',20);
Query OK, 1 row affected (0.02 sec)

mysql> insert into student values(4,'reema',20);
Query OK, 1 row affected (0.02 sec)

mysql> select * from student;
+----+-------+------+
| id | name  | age  |
+----+-------+------+
|  1 | ram   |   20 |
|  2 | sam   |   20 |
|  3 | seema |   20 |
|  4 | reema |   20 |
+----+-------+------+
4 rows in set (0.00 sec)

mysql> show tables;
+------------------+
| Tables_in_sample |
+------------------+
| student          |
+------------------+
1 row in set (0.02 sec)
```

```
--creating user1 and granting access;
create user 'user1'@'localhost' identified by 'user1';
grant all privileges on *.* to 'user1'@'localhost';
```

Name:Yash Mahajan Class:SE IT B Roll No.:04 PID:191061

```sql
--creating user2 and granting insert access;
create user 'user2'@'localhost' identified by 'user2';
grant insert on sample.student to 'user2'@'localhost';
```

MySQL 8.0 Command Line Client

```
mysql> create user 'user1'@'localhost' identified by 'user1';
Query OK, 0 rows affected (0.04 sec)

mysql> grant all privileges on *.* to 'user1'@'localhost';
Query OK, 0 rows affected (0.03 sec)

mysql> create user 'user2'@'localhost' identified by 'user2';
Query OK, 0 rows affected (0.03 sec)

mysql> grant insert on sample.student to 'user2'@'localhost';
Query OK, 0 rows affected (0.04 sec)

mysql>
```

```sql
--showing grants of each user;
show grants for 'user1'@'localhost';
show grants for 'user2'@'localhost';
```

MySQL 8.0 Command Line Client

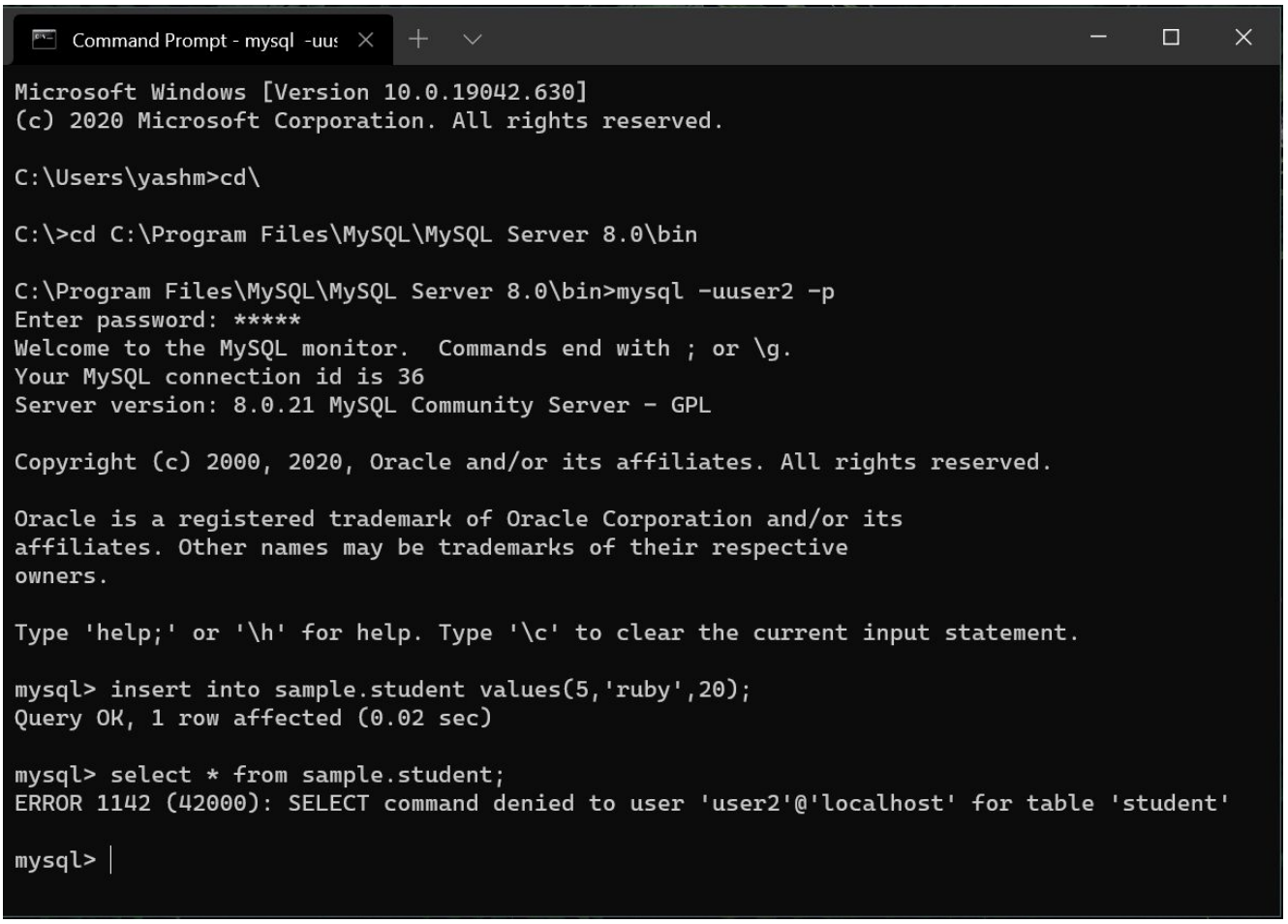
```
mysql> show grants for 'user1'@'localhost';
+-----------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------
----+
| Grants for user1@localhost

    |
+-----------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------
----+
| GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, SHUTDOWN, PROCESS, FILE, REFERENCES, INDEX, ALTER, SHOW DATABASES, SUPER, CREATE TEMPORARY TABLES, LOCK TABLES
, EXECUTE, REPLICATION SLAVE, REPLICATION CLIENT, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, CREATE USER, EVENT, TRIGGER, CREATE TABLESPACE, CREATE ROLE, DROP R
OLE ON *.* TO `user1`@`localhost`
    |
| GRANT APPLICATION_PASSWORD_ADMIN,AUDIT_ADMIN,BACKUP_ADMIN,BINLOG_ADMIN,BINLOG_ENCRYPTION_ADMIN,CLONE_ADMIN,CONNECTION_ADMIN,ENCRYPTION_KEY_ADMIN,GROUP_REPLICATION_ADMIN,I
NNODB_REDO_LOG_ARCHIVE,INNODB_REDO_LOG_ENABLE,PERSIST_RO_VARIABLES_ADMIN,REPLICATION_APPLIER,REPLICATION_SLAVE_ADMIN,RESOURCE_GROUP_ADMIN,RESOURCE_GROUP_USER,ROLE_ADMIN,SER
VICE_CONNECTION_ADMIN,SESSION_VARIABLES_ADMIN,SET_USER_ID,SHOW_ROUTINE,SYSTEM_USER,SYSTEM_VARIABLES_ADMIN,TABLE_ENCRYPTION_ADMIN,XA_RECOVER_ADMIN ON *.* TO `user1`@`localho
st` |
+-----------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------
----+
2 rows in set (0.00 sec)

mysql> show grants for 'user2'@'localhost';
+----------------------------------------------------------+
| Grants for user2@localhost                               |
+----------------------------------------------------------+
| GRANT USAGE ON *.* TO `user2`@`localhost`                |
| GRANT INSERT ON `sample`.`student` TO `user2`@`localhost` |
+----------------------------------------------------------+
2 rows in set (0.00 sec)

mysql>
```

```
--To test insert on student by user2;
insert into sample.student values(5,'ruby',20);
select * from sample.student;
```



```
select user from mysql.user;
select * from student;
```

MySQL 8.0 Command Line Client

```
mysql> select user from mysql.user;
+------------------+
| user             |
+------------------+
| mysql.infoschema |
| mysql.session    |
| mysql.sys        |
| root             |
| user1            |
| user2            |
+------------------+
6 rows in set (0.00 sec)

mysql> select * from student;
+----+-------+------+
| id | name  | age  |
+----+-------+------+
|  1 | ram   |   20 |
|  2 | sam   |   20 |
|  3 | seema |   20 |
|  4 | reema |   20 |
|  5 | ruby  |   20 |
+----+-------+------+
5 rows in set (0.00 sec)

mysql>
```

```
--Revoking grants on user2
revoke insert on sample.student from 'user2'@'localhost';
select user from mysql.user;
show grants for 'user2'@'localhost';

--deleting user2;
delete from mysql.user where user = 'user2';
select user from mysql.user;
```

```
MySQL 8.0 Command Line Client
mysql> revoke insert on sample.student from 'user2'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql> select user from mysql.user;
+------------------+
| user             |
+------------------+
| mysql.infoschema |
| mysql.session    |
| mysql.sys        |
| root             |
| user1            |
| user2            |
+------------------+
6 rows in set (0.00 sec)

mysql> show grants for 'user2'@'localhost';
+-----------------------------------------+
| Grants for user2@localhost              |
+-----------------------------------------+
| GRANT USAGE ON *.* TO `user2`@`localhost` |
+-----------------------------------------+
1 row in set (0.00 sec)

mysql> delete from mysql.user where user = 'user2';
Query OK, 1 row affected (0.02 sec)

mysql> select user from mysql.user;
+------------------+
| user             |
+------------------+
| mysql.infoschema |
| mysql.session    |
| mysql.sys        |
| root             |
| user1            |
+------------------+
5 rows in set (0.01 sec)

mysql>
```

9. Post Experimental Exercise-(To be hand written)

A. Questions:

1. Explain the term Access control in SQL
2. What is role based access control

B. Conclusion:

1. Write what was performed in the experiment
2. Mention few applications of what was studied.
3. Write the significance of the studied topic

C. References:

[1] Elmasri and Navathe, “Fundamentals of Database Systems”, 5th Edition, PEARSON Education.

[2] Korth, Silberchatz, Sudarshan, “Database System Concepts”, 6th Edition, McGraw – Hill

[3] https://www.w3schools.com/sql/sql_default.asp

Page No :

Date :

Yash Mahajan SE IT B 04

7. Pre Experiment Exercise :-

Brief Theory :-

i) Explain DCL Commands with example.

Data control language includes commands such as grant, revoke which mainly deals with the rights, permissions and other controls of database system.

(i) Grant command:

Grants a privilage to a user. It means that giving authority to other user by administrator. ~~then only you have~~ If you are the administrator then only you have the authority to grant privilage only if you have been granted that privilage.

Syntax :-

grant <Object Privilages> ON <Object Name> to <User Name> [WITH GRANT OPTION]

Eg :-

GRANT SELECT, UPDATE ON client_master TO 'user1';

-> Object Privilages.

A user can grant all privilages or grant only specific oject privilages.

Page No :
Date :

Yash Mahajan SE IT B 04

The list of object privileges is as follows :-

i) ALTER : Allows to change table definition using alter command.

ii) INDEX :- Allows to create an index

iii) Delete :- Allows to remove records.

(iv) INSERT :- Allow to add records.

v) SELECT :- Allows to query the table

(vi) UPDATE :- Allows to modify the records

Revoke Command.

The revoke command is used to deny the grant given on an object, Revokes a privilege from a user.

Syntax :-

REVOKE <Object Privilage> on <Object Name>
From <user name>

Eg :- Revoke DELETE ON Salesman_master
FROM 'user1'.

Page No :
Date :

Yash Mahajan SE ITB 04

9. Post Experimental Exercise

A. Questions :-

1) Explain the term access control in SQL.

The SQL access control model defines which authorization identifiers (users) can access a specific table.
SQL access control is based on privilages assigned to authorization identifiers to access objects. The creater of the object in a database is the owner and can perform any action on the object. By default no other user can access the objects unless the owner grants specific privilage to that user. The granting process assigns a privilage on an object to one or more users.
A user identifier represents a user of the DBMS and its is defined in an implementation dependent way SQL does not define how OS users are mapped to SQL users.

2) What is role based access control:
Role based access control (RBAC) is a method of restricting network access based on the roles of individual users within an enterprise. RBAC lets employees access the information only required to do their jobs and prevent them from

Attitude

accessing information that does not pretain to them.

In the role based access control data models, roles are based on several factors including authorization, responsibility and job ~~competion~~ competency. As such, companies can designate weather a user is an enduser an administrator or a specilist user. In addition, access to computer resources can be limited to specific tasks, such as the ~~abilt~~ ability to view, create or modify files

Companies that depend on RBAC are better able to secure their data and critical applications.

B) Conclusion :-

In this experiment we have studied and implemented various DCL commands on our database to create users and grant and revoke their privileges.

In SQL DCL Commands are used for assigning ~~the~~ authorization to a user and what type of authorization is ~~prov~~ assigned. They are also used to revoke any user privilege.

~~The~~ DCL Commands have an important role in providing access to the database to authorised users and prevent the users ~~to~~ from accessing the ~~information~~ information they don't pretain.