DBMS
Semester IV
LAB Sheet - 2
Due Date for submitting report : 9th February 2022

1. Introduction:

MySQL is an open-source database management software that helps users store, organize, and later retrieve data. It has a variety of options to grant specific users nuanced permissions within the tables and databases—this tutorial will give a short overview of a few of the many options.

How to Create a New User:

While logging from root user MySQL, it would have full access to all of the databases. However, in cases where more restrictions may be required, there are ways to create users with custom permissions. Initially log in from root user and we would be creating a new-user from root. We can create user within the MySQL shell:

```
mysql> CREATE USER 'newuser'@'localhost' IDENTIFIED BY 'password';
```

At this point new-user has no permissions to do anything with the databases. In fact, even if new-user tries to login (with the password, password), they will not be able to reach the MySQL shell. Therefore, we would provide the user with access to the information they will need. A strong password should be kept including upper case/ lower case and special character.

```
mysql> GRANT ALL PRIVILEGES ON * . * TO 'newuser'@'localhost';
```

Once you have finalized the permissions that you want to set up for your new users, always be sure to reload all the privileges.

```
mysql> FLUSH PRIVILEGES;
```

1.2 How To Grant Different User Permissions

Here is a short list of other common possible permissions that users can enjoy.

1. ALL PRIVILEGES- as we saw previously, this would allow a MySQL user full access to a designated database (or if no database is selected, global access across the system)
2. CREATE- allows them to create new tables or databases
3. DROP- allows them to them to delete tables or databases
4. DELETE- allows them to delete rows from tables
5. INSERT- allows them to insert rows into tables
6. SELECT- allows them to use the SELECT command to read through databases
7. UPDATE- allow them to update table rows
8. GRANT OPTION- allows them to grant or remove other users' privileges

To provide a specific user with a permission, you can use this framework:

```
mysql> GRANT type_of_permission ON database_name.table_name TO
```

```
'username'@'localhost';
```

Note: Each time you update or change a permission be sure to use the Flush Privileges command.

If you need to revoke a permission, the structure is almost identical to granting it:

```
mysql> REVOKE type_of_permission ON database_name.table_name FROM 'username'@'localhost';
```

You can review a user's current permissions by running the following:

```
mysql> SHOW GRANTS FOR 'username'@'localhost';
```

Just as you can delete databases with DROP, you can use DROP to delete a user altogether:

```
mysql> DROP USER 'username'@'localhost';
```

To test out your new user, log out by typing:

```
mysql> quit
```

and log back in with this command in terminal:

mysql -u [username] -p

Note:

1. All MySQL commands end with a semicolon; if the phrase does not end with a semicolon, the command will not execute.
2. Also, although it is not required, MySQL commands are usually written in uppercase and databases, tables, usernames, or text are in lowercase to make them easier to distinguish. However, the MySQL command line is not case sensitive.

2. How to Create and Delete a MySQL Database

MySQL organizes its information into databases; each one can hold tables with specific data.

You can quickly check what databases are available by typing:

```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
+--------------------+
3 rows in set (0.01 sec)
```

Creating a database is very easy:

```
CREATE DATABASE database name;
```

How to Access a MySQL Database:

Once we have a new database, we can begin to fill it with information. The first step is to create a new table within the larger database. Let's open up the database we want to use:

```
USE events;
```

In the same way that you could check the available databases, you can also see an overview of the tables that the database contains.

```
SHOW tables;
```

Since this is a new database, MySQL has nothing to show, and you will get a message that says, "Empty set"

How to Create a MySQL Table

Let's imagine that we are planning a get together of friends. We can use MySQL to track the details of the event. Let's create a new MySQL table:

```
CREATE TABLE student (PRN INT NOT NULL PRIMARY KEY
AUTO_INCREMENT,
FirsName VARCHAR(20),
LastName VARCHAR(30),
Batch INT,
DoB DATE);
```

Show table:

```
mysql> SHOW TABLES;
+------------------+
| Tables_in_events |
+------------------+
| student          |
+------------------+
1 row in set (0.01 sec)
```

We can remind ourselves about the table's organization with this command:

```
mysql> DESCRIBE student;
+----------+-------------+------+-----+---------+----------------+
| Field    | Type        | Null | Key | Default | Extra          |
+----------+-------------+------+-----+---------+----------------+
| PRN      | int         | NO   | PRI | NULL    | auto_increment |
| FirsName | varchar(20) | YES  |     | NULL    |                |
| LastName | varchar(30) | YES  |     | NULL    |                |
| Batch    | int         | YES  |     | NULL    |                |
| DoB      | date        | YES  |     | NULL    |                |
+----------+-------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

3. Exercise :

Create the following tables (Don't specify any constraints):

A. Category_details (category_id numeric (2), category_name varchar (30))
B. Sub_category_details (sub_category_id numeric(2), category_id numeric(2),sub_category_name varchar(30)

C. Product_details (Product_id numeric (6), category_id numeric(2),sub_category_id numeric(2), product_name varchar(30))

Now perform the following operations:

1. Add a primary key constraint (without any constraint name) on column category_id of category_details table.
2. Add a primary key constraint with a constraint name on column sub_category_id of sub_category_details table.
3. Add a foreign key constraint with constraint name on column category_id of sub_category_details table referencing category_id of category_details table.
4. For product_details table add primary key constraint on product_id. Also add foreign key constraint on category_id and sub_category_id columns referencing category_details(category_id) and sub_category_details (sub_category_id). Give appropriate names for all constraints.
5. Add a new column (price numeric(2)) to product_details table
6. Modify the data type of price to numeric(6,2)
7. Insert four tuples in the table. (With valid data)
8. Drop the price column

Deliverables:

A report should be prepared with AIM, Experiments, results and conclusion. (Not more that 5 pages.

1. **Draw the Schema diagram for the problem statement.**
2. **Paste the snaps of each of the table and enter some details for respect tables.**