1.Introduction to basic SQL queries

Instructor: Bidur Devkota
7th Sept 2020

MySQL installation for windows:
https://dev.mysql.com/doc/mysql-shell/8.0/en/mysql-shell-install-windows-quick.html
MySQL installation for Linux:
https://dev.mysql.com/doc/mysql-shell/8.0/en/mysql-shell-install-linux-quick.html

1. login to MySQL shell for user root (ubuntu)

```
mysql -u root -p
```

For windows , lofin for user root

```
mysql.exe –uroot –p
```

Try some MySQL Shell Commands

https://dev.mysql.com/doc/mysql-shell/8.0/en/mysql-shell-commands.html

Command	Alias/ Shortcut	Description
\help	\h or \?	Print help about MySQL Shell, or search the online help.
\quit	\q or \exit	Exit MySQL Shell.
\		In SQL mode, begin multiple-line mode. Code is cached and executed when an empty line is entered.
\status	\s	Show the current MySQL Shell status.
\js		Switch execution mode to JavaScript.
\py		Switch execution mode to Python.
\sql		Switch execution mode to SQL.
\connect	\c	Connect to a MySQL Server.
\reconnect		Reconnect to the same MySQL Server.
\use	\u	Specify the schema to use.
\source	\. or source	Execute a script file using the active language.

Command	Alias/ Shortcut	Description
	(no backslash)	
\warnings	\W	Show any warnings generated by a statement.
\ nowarnings	\w	Do not show any warnings generated by a statement.
\history		View and edit command line history.
\rehash		Manually update the autocomplete name cache.
\option		Query and change MySQL Shell configuration options.
\show		Run the specified report using the provided options and arguments.
\watch		Run the specified report using the provided options and arguments, and refresh the results at regular intervals.
\edit	\e	Open a command in the default system editor then present it in MySQL Shell.
\pager	\P	Configure the pager which MySQL Shell uses to display text.
\nopager		Disable any pager which MySQL Shell was configured to use.
\system	\!	Run the specified operating system command and display the results in MySQL Shell.

2. Playing with simple queries:

SHOW DATABASES;

CREATE DATABASE *test_db*;

SHOW DATABASES;

DROP DATABASE *test_db*;

```
          USE test_db;

SHOW tables;


          CREATE TABLE contacts
       (
       id INT(11) ,
         last_name VARCHAR(30) NOT NULL,
         first_name VARCHAR(30),
         address VARCHAR(30)
       );

          drop table  contacts;

INSERT INTO `contacts` (`id`,`last_name`,`first_name`,`address`) VALUES (1, "Sharma",
"Shyam","Pokhara");

select * from contacts;


INSERT INTO `contacts` (`id`,`last_name`,`first_name`,`address`) VALUES (2, "Hanks",
"Tom","Ktm"),(3, "Sharma", "Hari","Pokhara");

          select * from contacts;

          select  `id`,`last_name`,`first_name`,`address` from contacts;

          select  `id`,`first_name`,`address` from contacts;

          select  `id`,`first_name`,`address` from contacts where last_name = “Sharma”;

          select * from contacts where address="Pokhara";

          select * from contacts where address < "PokharA";
          select * from contacts where address != "PokharA";

          select * from contacts where address LIKE "P%";

          select * from contacts where address LIKE "%m";
```

Task:

1. Create table course with course id, name, total students, instructor.
2. Insert >10 rows into the table with different values.
3. Select courses with minimum 20 students;
4. select courses whose instructors name starts with a letter “A”.
5. Select courses whose id is even;

Submission:

1. Title: 1. Introduction to basic SQL queries.
2. Theory: Explain about the related queries(syntax, Descriptions)
3. Observations: **Queries** you run and their **screenshot**.
4. Conclusion

Note:

1. Submit a file with name <**1_SQL_Intro_ROLL_NAME.pdf**>

For example student ‘**Ram**’ with roll number ‘**100**’ should send a file named:

1_SQL_Intro_100_Ram.pdf

2. Copying will be marked ZERO.
3. Send as email attachment to bidur.devkota@informaxcollege.edu.np
4. Email Subject: **1.Introduction to basic SQL queries**