Practical 1- part 1

Be familiar with working with MySQL in Linux environment using command prompt

Last update: 27 July 2022

Learning objectives

1. Initiating working with a MySQL database in Linux environment using commands.
2. Be familiar with the workflow to create SQL commands in a text file and use them in the Terminal.
3. Create simple tables in an existing database.
4. Getting information about a database.

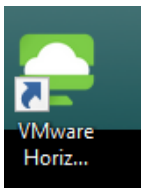
Tasks

In this practical, you will learn how to work with MySQL interactive environment. You will use MySQL on a Linux environment and use a Terminal to enter commands to work with MySQL server.

Note : "`>`" or "`mysql>`" are used in the lab sheets to indicate you are supposed to type commands in the terminal prompt or mysql prompt respectively. Do not type "`>`" or "`mysql>`" as a part of your commands.

1. Set up your working directory

In Curtin labs, log in to the computer using your Oasis username and password. Click on the 'VMWare Horizon client' icon on your desktop and provide your Oasis username and password again to connect to the virtual machine.



If 'VMWare Horizon client' is not available on your desktop, search for it and you will find it.

Once you connected to the virtual machine, you will see 'mydesktop.curtin.edu.au' icon. Click on that icon.

Then select 'Computer Science Linux Lab' icon and you will be connected to the Linux environment.

Now, make sure you go to your home drive before starting your work. You will see a short cut to access your home drive (`<student ID> Home Drive`) on your desktop.

In your Home Drive, get a Terminal (press CTRL+ALT+T)

Type > pwd to verify whether you are at the correct directory.

You will see your directory is /mnt/home/<your student id>

In your home drive, create a directory to do your DBS work (> mkdir DBS)

Go inside DBS directory (> cd DBS) and create a directory for each week

(> mkdir Prac01 Prac02)

In your Home Drive you should have followed directory structure

DBS/

- Prac01
- Prac02
- ..
- Prac10

Now, go to Prac01 directory to start your work (>cd Prac01)

If you are using your own PC and accessing a Linux environment, still make sure you have created a DBS directory and sub-directories to do your practicals. If you organize your work now, it will be much easier in later lab sessions.

2. Use terminal and starting MySQL server

You will use Terminal and run commands to access MySQL server and working with databases.

You can know whether MySQL is available and the version information using the following command. In your Terminal prompt type the following and press enter key:

We are going to connect to MySQL server now. In your Terminal prompt type

```
> mysql -V
```

the following and press enter key:

You will be prompted to give the password for the server.

```
> mysql –u me -p
```

Your username for Curtin Computer lab's MySQL server is 'me'. Password is: ' myUserPassword'. Type this password and press Enter key.

You will be connected to the MySQL server and the prompt will be changed to:

You are now in the MySQL interactive environment.

```
mysql>
```

To close the connection, type exit.

Again, connect to the MySQL server using the required command.

```
mysql> exit
```

3. See the available databases and creating tables

It is a good practice to type your commands in a text editor and then copy and paste the command in the Terminal prompt. If you have saved your commands in a text file, you can re-run them and get the required output again easily.

Open a file in Vim (`> vim Prac01Part1Commands`) or any text editor you wish to use (Make sure you are in your Home Drive, and the DBS/Prac01 directory). Type and save all your commands in Prac01Part1Commands txt file . You can keep the Terminal and the Text editor side by side so that work can be done easily.

1. To see the databases already in the server, use the following command.
2. Type the same command as above in simple letters and observe the

```
mysql> SHOW DATABASES;
```

   output.
3. You will create tables in the 'dswork' database in following tasks.

   To use the 'dswork' database type:

```
mysql> USE dswork;
```

   You will see a message, 'Database changed', indicating the working database is 'dswork'

4. Student, Unit and Enrolment are three relations to be created in the 'dswork' database.

Consider the following definition for Student relation.

Student relation

COL NAME	TYPE	SIZE	DESCRIPTION
sno	CHAR	8	Student number
firstname	VARCHAR	12	First name
lastname	VARCHAR	15	Last name
phoneno	CHAR	10	Student's telephone number

To create the 'Student' relation using SQL, first type the following command in your text file. Then copy and paste the command to mysql prompt to create the 'Studen' table.

```
Mysql> CREATE TABLE Student(
sno  CHAR(8),
firstname VARCHAR(12),
lastname  VARCHAR(15),
phoneno  CHAR(10)
);
```

5. Write down the command to create the following table in the text editor(Hint : use CREATE TABLE statement). Copy and paste the command you have written to create Unit table in 'dswork' database.

Unit table

COL NAME	TYPE	SIZE	DESCRIPTION
unitindex	CHAR	8	Unit code
dept	CHAR	10	Department Code
unitname	VARCHAR	40	Unit Name

6. Similarly, create the Enrolments in 'dswork' database using the following definition for the Enrolement table.

Enrolment table			
COL NAME	**TYPE**	**SIZE**	**DESCRIPTION**
unit	CHAR	8	Unit code
student	CHAR	8	Student number
year	CHAR	4	Year enrolled
mark	INT	3	mark for unit
grade	CHAR	1	grade for unit

NOTE 1: Data types and size of them

- The CHAR and VARCHAR types are declared with a length that indicates the maximum number of characters you want to store. CHAR(8) can hold up to 8 characters. The length of a CHAR column is fixed to the length that you declare when you create the table. The length can be any value from 0 to 255.

- Values in VARCHAR columns are variable-length strings. The length can be specified as a value from 0 to 65,535. For integer data types, size here indicates the maximum display width, which is optional to state. The maximum display width is 255.

- Integer data types can be defined as INT([M]), where optional parameter M indicates the maximum display width. INT(3) specifies an INT with a display width of three digits. Display width is unrelated to the range of values a type can store and does not constrain the range of values that can be stored in the column. INT type is of size 4. bytes.

4. Getting information about your database

1. Display all the tables currently in your database using the following command:

```
SHOW TABLES;
```

2. To learn about the attributes of relation Student, issue the command:

Hint: The DESCRIBE command can be shortened to DESC.

```
DESCRIBE Student;
```

3. You can also use SHOW to see the contents of a table.

```
SHOW COLUMNS FROM Student;
```

5. Save the commands and reuse

Instead of copy and paste the commands, you can use the file name to run the SQL commands in a file. Type the following command and observe the output.

1. Create a copy of the 'Prac01Part1Commands' file as 'Pra01tables' (Use cp command)
2. Open the 'Prac01tables' file and delete any other commands except three commands to create three tables.
3. Change the table names on the command to 'TestStudent',' TestUnit' and 'TestEnrolment'. Save the Prac01tables file.
4. Type the following command and observe the output.

```
mysql> SOURCE Prac01tables;
```

5. Use a suitable command and see the tables in the databse. You will see three new tables are created.

After finishing your work, close your connection uisng exit command. Log off from vertual machine also.

Make sure you have saved all the commands you have created in the DBS/Prac01/Prac01Part1Commands file. We will be using the same commands in the part2 of this worksheet.

Check whether you have achieved learning objectives of this worksheet :

I am confident that I can,

Connect to a MySQL server in Linux environment using Terminal and close the connection after my work	✓
Create a simple table in an existing database using SQL commands	
Get information about a database using SQL commands	
Save SQL commands in a text file and reuse them by copy and paste/ reading from the file	

Please refer lecture slides, reading materials, and online resources, and attempt again, if all the objectives are not achieved. Ask your tutor and get help if you need any clarification.

It's always a good practise to try to finish the practical of a particular week, before attempting the next practical worksheet as your work will be building upon the previous week's tasks.

NOTE 2 : Remember to use a text file and save commands.

- Remember to first type the required commands in the text editor and then copy and paste it in the mysql > prompt.
- Save the text editor with the commands using suitable file name (e.g.: Prac01Part1) so that you will have all the commands of the practical saved in a text file.
- You may use # and add comments, brief description of what you are doing before the command.
- Initially you will use short commands but in few weeks you will start creating lengthy commands so you will see the value of using a text editor. Additionally, all the commands you created in the practical as answers to the tasks can be saved using the text editor.

NOTE 3: Missing semi-colon error.

- Every command in MySQL must be terminated with " ; ". If you forget this and press 'Enter', you'll receive a blank line instead of the usual prompt. When that happens, just type a semi-colon and Enter.

NOTE 4: naming conventions

- Note that the table name (Student) is capitalized while the attribute names (like sno) are not. This is a convention used in this unit to make it easy to tell them apart.
- Table names are case sensitive on the server being used, so take care to get this correct. Case doesn't matter for the attributes, but keeping them all in lower case helps differentiate them from table names and reserved words.
- We write reserved words in ALL-CAPS, again just to make them more obvious. MySQL will accept "Create Table", "CREATE TABLE" or "create table" equally (without the quotes).
- Another convention to make the table names singular (*i.e.*, Student rather than Students). Though a table contains many students, not just one, singular convention is easier when dealing with relationships.
- ***You can adopt any suitable naming convention but do not mix them!*** Attribute names can be written as "first_name" also to make it more readable rather than "first_name". Same can be adopted to table names.