Use MYSQL and practice following commands using MYSQL command line client

1. Creating and Using a Database

- Use the SHOW statement to find out what databases currently exist on the server:
  mysql> SHOW DATABASES;

```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| test               |
+--------------------+
3 rows in set (0.00 sec)
```

- If the database exists, try to access it:
  mysql>USE database_name;

```
mysql> USE test;
Database changed
```

- If you want to create a new data base:
  mysql>CREATE DATABASE database_name;

```
mysql> CREATE DATABASE employees;
Query OK, 1 row affected (0.01 sec)
```

2. Creating a Table

- If you want to create new table in a selected database :
  mysql> CREATE TABLE table_name (column_name1 data_type, column_name2 data_type,....);

```
mysql> USE employees;
Database changed
mysql> CREATE TABLE employee(
    -> Id INT(2) PRIMARY KEY,
    -> firstName VARCHAR(10),
    -> lastName VARCHAR(10),
    -> district VARCHAR(10),
    -> salary INT(10)
    -> );
Query OK, 0 rows affected (0.20 sec)
```

- Use the SHOW statement to find out what tables currently exist on the database:
  mysql>SHOW TABLES;

```
mysql> SHOW TABLES;
+----------------------+
| Tables_in_employees  |
+----------------------+
| employee             |
+----------------------+
1 row in set (0.00 sec)
```

- To verify that your table was created the way you expected, use a DESCRIBE statement
  mysql>DESCRIBE table_name;

```
mysql> DESCRIBE employee;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| Id        | int(2)      | NO   | PRI | NULL    |       |
| firstName | varchar(10) | YES  |     | NULL    |       |
| lastName  | varchar(10) | YES  |     | NULL    |       |
| district  | varchar(10) | YES  |     | NULL    |       |
| salary    | int(10)     | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
5 rows in set (0.01 sec)
```

3. Inserting Data into a Table

- When you want to add new records one at a time, the INSERT statement is useful. In its simplest form, you can supply values for each column, in the order in which the columns were listed in the CREATE TABLE statement:
  mysql>INSERT INTO table_name (column1, coulmn2, ...) VALUES (value1, value2, ...);

```
mysql> INSERT INTO employee (Id, firstName,lastName, district, salary) VALUES (1
, 'Saman', 'Rathnayake','Kandy',25000);
Query OK, 1 row affected (0.07 sec)
```

4. Retrieving Information from a Table

- The SELECT statement is used to pull information from a table. The general form of the statement is:
  SELECT what_to_select
  FROM which_table
  WHERE conditions_to_satisfy;

  - ***Selecting all data***
    The simplest form of SELECT retrieves everything from a table:
    mysql>SELECT * FROM table_name;

```
mysql> SELECT * FROM employee;
+----+-----------+------------+------------+--------+
| Id | firstName | lastName   | district   | salary |
+----+-----------+------------+------------+--------+
|  1 | Saman     | Rathnayake | Kandy      |  25000 |
|  2 | Nimal     | De Silva   | Kurunegala |  25000 |
|  3 | Ruwan     | Perera     | Kurunegala |  25000 |
|  4 | Ruwani    | Perera     | Colombo    |  25000 |
+----+-----------+------------+------------+--------+
4 rows in set (0.00 sec)
```

- ***Selecting particular rows***
  You can select only particular rows from your table:
  mysql>SELECT * FROM table_name WHERE condition_to_satisfy

```
mysql> SELECT * FROM employee WHERE district='Kurunegala';
+----+-----------+----------+------------+--------+
| Id | firstName | lastName | district   | salary |
+----+-----------+----------+------------+--------+
|  2 | Nimal     | De Silva | Kurunegala |  25000 |
|  3 | Ruwan     | Perera   | Kurunegala |  25000 |
+----+-----------+----------+------------+--------+
2 rows in set (0.00 sec)
```

- You can combine conditions, for example to find employees who have last name=Perera and district=Kurunegala

```
mysql> SELECT * FROM employee WHERE lastName='Perera' AND district='Kurunegala';

+----+-----------+----------+------------+--------+
| Id | firstName | lastName | district   | salary |
+----+-----------+----------+------------+--------+
|  3 | Ruwan     | Perera   | Kurunegala |  25000 |
+----+-----------+----------+------------+--------+
1 row in set (0.00 sec)
```

- The preceding query uses the AND logical operator. There is also an OR operator, for example to find employees who have last name=Perera or district=Kurunegala

```
mysql> SELECT * FROM employee WHERE lastName='Perera' OR district='Kurunegala';
+----+-----------+----------+------------+--------+
| Id | firstName | lastName | district   | salary |
+----+-----------+----------+------------+--------+
|  2 | Nimal     | De Silva | Kurunegala |  25000 |
|  3 | Ruwan     | Perera   | Kurunegala |  25000 |
|  4 | Ruwani    | Perera   | Colombo    |  25000 |
+----+-----------+----------+------------+--------+
3 rows in set (0.00 sec)
```

- AND and OR may be intermixed, although AND has higher precedence than OR. If you use both operators, it is a good idea to use parentheses to indicate explicitly how conditions should be grouped:

```
mysql> SELECT * FROM employee WHERE (lastName='Perera' AND salary='25000') OR
(district='Kandy');
+----+-----------+------------+------------+--------+
| Id | firstName | lastName   | district   | salary |
+----+-----------+------------+------------+--------+
|  1 | Saman     | Rathnayake | Kandy      |  25000 |
|  3 | Ruwan     | Perera     | Kurunegala |  25000 |
|  4 | Ruwani    | Perera     | Colombo    |  25000 |
+----+-----------+------------+------------+--------+
3 rows in set (0.00 sec)
```

5. **<u>Selecting Particular Columns</u>**
   If you do not want to see entire rows from your table, just name the columns in which you are interested, separated by commas.

- For example, if you want to know the name of the employees, select the firstName and lastName columns.

```
mysql> SELECT firstName, lastName FROM employee;
+-----------+------------+
| firstName | lastName   |
+-----------+------------+
| Saman     | Rathnayake |
| Nimal     | De Silva   |
| Ruwan     | Perera     |
| Ruwani    | Perera     |
+-----------+------------+
4 rows in set (0.00 sec)
```

- Notice that the following query simply retrieves the lastName column from each record and some of them appear more than once. To minimize the output, retrieve each unique output record just once by adding the keyword DISTINCT:

```
mysql> SELECT lastName FROM employee;
+------------+
| lastName   |
+------------+
| Rathnayake |
| De Silva   |
| Perera     |
| Perera     |
+------------+
4 rows in set (0.00 sec)
```

```
mysql> SELECT DISTINCT lastName FROM employee;
+------------+
| lastName   |
+------------+
| Rathnayake |
| De Silva   |
| Perera     |
+------------+
3 rows in set (0.04 sec)
```

- You can use a WHERE clause to combine row selection with column selection. For example, to get district of Ruwani use this query:

```
mysql> SELECT district FROM employee WHERE firstName='Ruwani';
+----------+
| district |
+----------+
| Colombo  |
+----------+
1 row in set (0.00 sec)
```

6. Update Values in a Table

The UPDATE statement is used to update existing records in a table

```
UPDATE table_name
SET coulmn1=value ,...
WHERE some_column=some_value,...
```

```
mysql> UPDATE employee SET firstName='Nuwani', lastName='Herath' WHERE firstName
='Ruwani' AND lastName='Perera';
Query OK, 1 row affected (0.60 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM employee;
+----+-----------+------------+------------+--------+
| Id | firstName | lastName   | district   | salary |
+----+-----------+------------+------------+--------+
|  1 | Saman     | Rathnayake | Kandy      |  25000 |
|  2 | Nimal     | De Silva   | Kurunegala |  25000 |
|  3 | Ruwan     | Perera     | Kurunegala |  25000 |
|  4 | Nuwani    | Herath     | Colombo    |  25000 |
+----+-----------+------------+------------+--------+
4 rows in set (0.00 sec)
```

7. **DELETE**

The DELETE statement is used to delete existing records in a table

DELETE FROM table_name

WHERE some_column=some_value

```
mysql> DELETE FROM employee WHERE district='Kurunegala';
Query OK, 2 rows affected (0.06 sec)

mysql> SELECT * FROM employee;
+----+-----------+------------+----------+--------+
| Id | firstName | lastName   | district | salary |
+----+-----------+------------+----------+--------+
|  1 | Saman     | Rathnayake | Kandy    |  25000 |
|  4 | Nuwani    | Herath     | Colombo  |  25000 |
+----+-----------+------------+----------+--------+
2 rows in set (0.00 sec)
```

8. **DROP**

- If you need to delete a table

  DROP TABLE table_name;

```
mysql> DROP TABLE employee;
Query OK, 0 rows affected (0.08 sec)

mysql> SHOW TABLES;
Empty set (0.00 sec)
```

- If you need to delete a database

  DROP DATABASE database_name;

```
mysql> DROP DATABASE employees;
Query OK, 0 rows affected (0.04 sec)

mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| test               |
+--------------------+
3 rows in set (0.00 sec)
```

9. **TRUNCATE**

- If you only want to delete the data inside the table and not the table itself
  TRUNCATE TABLE table_name;

```
mysql> TRUNCATE TABLE employee;
Query OK, 0 rows affected (0.03 sec)

mysql> SELECT * FROM employee;
Empty set (0.00 sec)
```

10. **ALTER TABLE**

The ALTER TABLE statement is used to add, delete or modify columns in an existing table.

- To add a column in a table, use the following syntax:
  ALTER TABLE table_name
  ADD coumn_name datatype;

```
mysql> ALTER TABLE employee ADD birthday Date;
Query OK, 0 rows affected (0.24 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESCRIBE employee;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| Id        | int(2)      | NO   | PRI | NULL    |       |
| firstName | varchar(10) | YES  |     | NULL    |       |
| lastName  | varchar(10) | YES  |     | NULL    |       |
| district  | varchar(10) | YES  |     | NULL    |       |
| salary    | int(10)     | YES  |     | NULL    |       |
| birthday  | date        | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
6 rows in set (0.02 sec)
```

- To delete a column in a table, use the following syntax:
  ALTER TABLE table_name
  DROP COLUMN column_name;

```
mysql> ALTER TABLE employee DROP COLUMN birthday;
Query OK, 0 rows affected (0.81 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESCRIBE employee;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| Id        | int(2)      | NO   | PRI | NULL    |       |
| firstName | varchar(10) | YES  |     | NULL    |       |
| lastName  | varchar(10) | YES  |     | NULL    |       |
| district  | varchar(10) | YES  |     | NULL    |       |
| salary    | int(10)     | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
5 rows in set (0.02 sec)
```

- To change the name or the data type of a column in a table, use the following syntax:
  ALTER TABLE table_name

CHANGE old_coulmn_name new_column_name datatype;

```
mysql> ALTER TABLE employee CHANGE Id IDNumber int(2);
Query OK, 0 rows affected (0.26 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESCRIBE employee;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| IDNumber  | int(2)      | NO   | PRI | 0       |       |
| firstName | varchar(10) | YES  |     | NULL    |       |
| lastName  | varchar(10) | YES  |     | NULL    |       |
| district  | varchar(10) | YES  |     | NULL    |       |
| salary    | int(10)     | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
5 rows in set (0.02 sec)
```

```
mysql> ALTER TABLE employee CHANGE IDNumber IdNumber VARCHAR(2);
Query OK, 0 rows affected (0.30 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESCRIBE employee;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| IdNumber  | varchar(2)  | NO   | PRI |         |       |
| firstName | varchar(10) | YES  |     | NULL    |       |
| lastName  | varchar(10) | YES  |     | NULL    |       |
| district  | varchar(10) | YES  |     | NULL    |       |
| salary    | int(10)     | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
5 rows in set (0.01 sec)
```

References:

1. http://www.w3schools.com/sql/default.asp (SQL Tutorial)
2. http://dev.mysql.com/downloads/installer/5.5.html (MySQL Installation)