SQL Basics

Login to MySQL, CLI using the credentials,
Host: ip-172-31-13-154, User: insofeadmin and Password: MDQzZTgyYj

```
[rameshmelapu9416@ip-172-31-20-58 ~]$ mysql -h ip-172-31-13-154 -u insofeadmin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 157167
Server version: 5.6.30 MySQL Community Server (GPL)

Copyright (c) 2000, 2015, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]>
```

SHOW DATABASES: List all the existing databases.

```
MySQL [(none)]> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| insofe_emp_db      |
| insofe_employees   |
| insofe_employeesDB |
| insofe_naveen      |
| insofe_results     |
| insofe_stocks      |
| insofe_test        |
+--------------------+
8 rows in set (0.00 sec)
```

Let us begin with a simple example - a *product sales database*. A product sales database typically consists of many tables, e.g., products, customers, suppliers, orders, payments, employees, among others.

We shall begin with the first table called "products" with the following columns (having data types as indicated) and rows:

Database: insofe_prodsalesdb;
Table: products

productID INT	productCode CHAR(3)	name VARCHAR(30)	quantity INT	price DECIMAL(10,2)
1001	PEN	Pen Red	5000	1.23
1002	PEN	Pen Blue	8000	1.25
1003	PEN	Pen Black	2000	1.25
1004	PEC	Pencil 2B	10000	0.48
1005	PEC	Pencil 2H	8000	0.49

Creating and Deleting a Database - CREATE DATABASE and DROP DATABASE

```
MySQL [(none)]> CREATE DATABASE insofe_prodsalesdb;
Query OK, 1 row affected (0.01 sec)

MySQL [(none)]> DROP DATABASE insofe_prodsalesdb;
Query OK, 0 rows affected (0.01 sec)

MySQL [(none)]> CREATE DATABASE IF NOT EXISTS insofe_prodsalesdb;
Query OK, 1 row affected (0.00 sec)

MySQL [(none)]> DROP DATABASE IF EXISTS insofe_prodsalesdb;
Query OK, 0 rows affected (0.00 sec)

MySQL [(none)]> DROP DATABASE IF EXISTS insofe_prodsalesdb;
Query OK, 0 rows affected, 1 warning (0.00 sec)

MySQL [(none)]> DROP DATABASE insofe_prodsalesdb;
ERROR 1008 (HY000): Can't drop database 'insofe_prodsalesdb'; database doesn't exist
```

IMPORTANT: Use SQL DROP (and DELETE) commands with extreme care, as the deleted entities are irrecoverable. **THERE IS NO UNDO!!!**

```
MySQL [(none)]> CREATE DATABASE IF NOT EXISTS insofe_prodsalesdb;
Query OK, 1 row affected (0.00 sec)

MySQL [(none)]> SHOW CREATE DATABASE insofe_prodsalesdb;
+--------------------+-----------------------------------------------------------------------------------+
| Database           | Create Database                                                                   |
+--------------------+-----------------------------------------------------------------------------------+
| insofe_prodsalesdb | CREATE DATABASE `insofe_prodsalesdb` /*!40100 DEFAULT CHARACTER SET latin1 */ |
+--------------------+-----------------------------------------------------------------------------------+
1 row in set (0.01 sec)

MySQL [(none)]> SHOW CREATE DATABASE insofe_prodsalesdb \G;
*************************** 1. row ***************************
       Database: insofe_prodsalesdb
Create Database: CREATE DATABASE `insofe_prodsalesdb` /*!40100 DEFAULT CHARACTER SET latin1 */
1 row in set (0.00 sec)
```

-- Show all the databases in the server, to confirm that "insofe_prodsalesdb" database has been created.

-- Set " **insofe_prodsalesdb**" as the default database so as to reference its table directly.

```
MySQL [(none)]> USE insofe_prodsalesdb;
Database changed
MySQL [insofe_prodsalesdb]>
```

-- Show the current (default) database.

```
MySQL [insofe_prodsalesdb]> SELECT DATABASE();
+--------------------+
| DATABASE()         |
+--------------------+
| insofe_prodsalesdb |
+--------------------+
1 row in set (0.00 sec)
```

-- Show all the tables in the current database.
-- " **insofe_prodsalesdb** " has no table (empty set).

```
MySQL [insofe_prodsalesdb]> SHOW TABLES;
Empty set (0.00 sec)
```

-- Create the table "products".

```
MySQL [insofe_prodsalesdb]> CREATE TABLE IF NOT EXISTS products (
    -> productID INT UNSIGNED NOT NULL AUTO_INCREMENT,
    -> productCode CHAR(3) NOT NULL DEFAULT '',
    -> name VARCHAR(30) NOT NULL DEFAULT '',
    -> quantity INT UNSIGNED NOT NULL DEFAULT 0,
    -> price DECIMAL(7,2) NOT NULL DEFAULT 99999.99,
    -> PRIMARY KEY(productID)
    -> );
Query OK, 0 rows affected (0.01 sec)
```

-- Show all the tables to confirm that the "products" table has been created.

```
MySQL [insofe_prodsalesdb]> SHOW TABLES;
+------------------------------+
| Tables_in_insofe_prodsalesdb |
+------------------------------+
| products                     |
+------------------------------+
1 row in set (0.00 sec)
```

```
-- Describe the fields (columns) of the "products" table.
```

```
MySQL [insofe_prodsalesdb]> DESCRIBE products;
+-------------+------------------+------+-----+----------+----------------+
| Field       | Type             | Null | Key | Default  | Extra          |
+-------------+------------------+------+-----+----------+----------------+
| productID   | int(10) unsigned | NO   | PRI | NULL     | auto_increment |
| productCode | char(3)          | NO   |     |          |                |
| name        | varchar(30)      | NO   |     |          |                |
| quantity    | int(10) unsigned | NO   |     | 0        |                |
| price       | decimal(7,2)     | NO   |     | 99999.99 |                |
+-------------+------------------+------+-----+----------+----------------+
5 rows in set (0.00 sec)
```

Inserting Rows - INSERT INTO

INSERT INTO Syntax

```
-- All columns
     INSERT INTO tableName VALUES (firstColumnValue, ..., lastColumnValue)

     -- Insert Multiple rows at a time.
     INSERT INTO tableName VALUES
          (row1FirstColumnValue, ..., row1lastColumnValue),
          (row2FirstColumnValue, ..., row2lastColumnValue),
          ...

-- Insert single record with selected columns
     INSERT INTO tableName (column1Name, ..., columnNName) VALUES
     (column1Value, ..., columnNValue)

-- Alternately, use SET to set the values
     INSERT INTO tableName SET column1=value1, column2=value2, ...

-- Insert multiple records
     INSERT INTO tableName
        (column1Name, ..., columnNName)
     VALUES
        (row1column1Value, ..., row2ColumnNValue),
        (row2column1Value, ..., row2ColumnNValue),
        ...
```

```
-- Insert a row with all the column values
mysql>  INSERT INTO products VALUES (1001, 'PEN', 'Pen Red', 5000, 1.23);
Query OK, 1 row affected (0.00 sec)

-- Insert multiple rows in one command
-- Inserting NULL to the auto_increment column results in max_value + 1
mysql> INSERT INTO products VALUES
    ->          (NULL, 'PEN', 'Pen Blue',  8000, 1.25),
    ->          (NULL, 'PEN', 'Pen Black', 2000, 1.25);
Query OK, 2 rows affected (0.00 sec)
Records: 2  Duplicates: 0  Warnings: 0


-- Insert value to selected columns
-- Missing value for the auto_increment column also results in max_value
+ 1
mysql> INSERT INTO products (productCode, name, quantity, price) VALUES
    ->          ('PEC', 'Pencil 2B', 10000, 0.48),
    ->          ('PEC', 'Pencil 2H', 8000, 0.49);
Query OK, 2 rows affected (0.00 sec)
Records: 2  Duplicates: 0  Warnings: 0


-- Missing columns get their default values
mysql> INSERT INTO products (productCode, name) VALUES ('PEC', 'Pencil HB');
Query OK, 1 row affected (0.00 sec)


-- 2nd column (productCode) is defined to be NOT NULL
mysql> INSERT INTO products values (NULL, NULL, NULL, NULL, NULL);
ERROR 1048 (23000): Column 'productCode' cannot be null
```

Querying the Database - SELECT

```
-- List all rows for the specified columns
     SELECT name, price FROM products;
-- List all rows of ALL the columns. The wildcard * denotes ALL columns
     SELECT * FROM products;

-- Comparison Operations
     SELECT name, price FROM products WHERE price < 1.0;

     SELECT name, quantity FROM products WHERE quantity <= 2000;

     SELECT name, price FROM products WHERE productCode = 'PEN'; --
     String values are quoted

String Pattern Matching - LIKE and NOT LIKE
-- "name" begins with 'PENCIL'
     SELECT name, price FROM products WHERE name LIKE 'PENCIL%';

-- "name" begins with 'P', followed by any two characters,
--   followed by space, followed by zero or more characters
     SELECT name, price FROM products WHERE name LIKE 'P__ %';

Logical Operators - AND, OR, NOT, XOR
     SELECT * FROM products WHERE quantity >= 5000 AND name LIKE 'Pen
     %';

     SELECT * FROM products WHERE quantity >= 5000 AND price < 1.24 AND
     name LIKE 'Pen %';

     SELECT * FROM products WHERE NOT (quantity >= 5000 AND name LIKE
     'Pen %');

IS NULL, IS NOT NULL
     SELECT * FROM products WHERE productCode IS NULL;

     SELECT * FROM products WHERE productCode = NULL; -- This is a
     common mistake.
     NULL cannot be compared.
```

ORDER BY Clause

SYNTAX:

```
SELECT ... FROM tableName
WHERE criteria
ORDER BY columnA ASC|DESC, columnB ASC|DESC, ...
```

```
-- Order the results by price in descending order
    SELECT * FROM products WHERE name LIKE 'Pen %' ORDER BY price DESC;
-- Order by price in descending order, followed by quantity in ascending (default) order
    SELECT * FROM products WHERE name LIKE 'Pen %' ORDER BY price DESC, quantity;
```

LIMIT Clause

```
-- Display the first two rows
    SELECT * FROM products ORDER BY price LIMIT 2;
-- Skip the first two rows and display the next 1 row
    SELECT * FROM products ORDER BY price LIMIT 2, 1;
```

DISTINCT

```
-- Without DISTINCT
    SELECT price FROM products;
-- With DISTINCT on price
    SELECT DISTINCT price AS `Distinct Price` FROM products;
-- DISTINCT combination of price and name
    SELECT DISTINCT price, name FROM products;
```

GROUP BY Clause

The GROUP BY clause allows you to *collapse* multiple records with a common value into groups.

```
SELECT * FROM products GROUP BY productCode; -- Only first record in each group is shown.
```

Modifying Data - UPDATE

To modify existing data, use UPDATE ... SET command, with the following syntax:

```
      SYNTAX:
      UPDATE tableName SET columnName = {value|NULL|DEFAULT}, ... WHERE
      criteria

-- Increase the price by 10% for all products
      UPDATE products SET price = price * 1.1;
      SELECT * FROM products;

-- Modify selected rows
      UPDATE products SET quantity = quantity - 100 WHERE name = 'Pen
      Red';
      SELECT * FROM products WHERE name = 'Pen Red';

-- Modify more than one values
      UPDATE products SET quantity = quantity + 50, price = 1.23 WHERE
      name = 'Pen Red';
      SELECT * FROM products WHERE name = 'Pen Red';
```

Deleting Rows - DELETE FROM

```
-- Delete all rows from the table. Use with extreme care! Records are
NOT recoverable!!!
      SYNTAX:
      DELETE FROM tableName
-- Delete only row(s) that meets the criteria
      SYNTAX:
      DELETE FROM tableName WHERE criteria

      DELETE FROM products WHERE name LIKE 'Pencil%';
      SELECT * FROM products;

-- Use this with extreme care, as the deleted records are irrecoverable!
      DELETE FROM products;
      SELECT * FROM products;
```

https://www.ntu.edu.sg/home/ehchua/programming/sql/MySQL_Beginner.html