Enrolment No:202203103510303

Practical No.1

Aim: To implement Basic SQL commands and to access & modify Data using SQL. Create and populate database using Data Definition Language (DDL) and DML Commands

Theory:

Implementing basic SQL commands involves utilizing Data Definition Language (DDL) and Data Manipulation Language (DML) to create, access, and modify a database. DDL commands like CREATE DATABASE and CREATE TABLE establish the database's structure. For instance, "CREATE DATABASE Library;" generates a new database named "Library". Tables are designed using CREATE TABLE, defining columns like BookID, Title, and AuthorID. Foreign keys ensure data consistency and relationships.

With the structure in place, DML commands enable data interactions. INSERT statements add data; "INSERT INTO Books (BookID, Title, AuthorID) VALUES (1, 'SQL Basics', 1);" populates the "Books" table. SELECT queries retrieve data; "SELECT * FROM Books;" fetches all book records. UPDATE statements modify data; "UPDATE Books SET Title = 'SQL Fundamentals' WHERE BookID = 1;" changes the book's title.

INSERT introduces new records, as in "INSERT INTO Books (BookID, Title, AuthorID) VALUES (2, 'Database Design', 2);". DELETE commands remove data; "DELETE FROM Books WHERE BookID = 2;" deletes the book with ID 2. However, cautious usage is advised to prevent accidental data loss.

In conclusion, SQL proficiency is essential for managing data. DDL creates databases and tables, while DML provides the means to insert, retrieve, modify, and delete data. Balancing these actions while prioritizing data integrity and security ensures efficient and safe data management.

1)Change the price of „plate‟ from 1500 to 2000.

```
mysql> SELECT*FROM SUPPLIER;
+-------------+-----------+-----------+
| SUPPLIER_NO | NAME      | ADDRESS   |
+-------------+-----------+-----------+
|        1001 | MICHAEL   | BASILDON  |
|        1002 | RINGWORLD | GERMANY   |
|        1003 | BABYLON   | LONDON    |
|        1004 | JOHN      | BASILDON  |
|        1005 | SMITH     | GERMANY   |
+-------------+-----------+-----------+
5 rows in set (0.00 sec)
```

```
mysql> UPDATE PRODUCT SET PRICE=2000 WHERE DESCRIPTION='PLATE';
Query OK, 1 row affected (0.06 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT*FROM PRODUCT;
+------------+---------------+-------+-------------+-----------------+---------------+
| PRODUCT_NO | DESCRIPTION   | PRICE | SUPPLIER_NO | MARKETING_REP_NO | SUPPLY_DEPT_NO |
+------------+---------------+-------+-------------+-----------------+---------------+
|        120 | REDUCER       |  1200 |        1005 |               5 |              6 |
|        121 | PLATE         |  2000 |        1004 |               3 |              1 |
|        122 | HANDLE        |   700 |        1003 |               2 |              4 |
|        124 | WIDGET REMOVER |   900 |        1005 |               4 |              2 |
|        136 | SIZE WIDGET   |  1000 |        1001 |               1 |              5 |
|        137 | SIZE WIDGET   | 15000 |        1002 |               2 |             16 |
+------------+---------------+-------+-------------+-----------------+---------------+
6 rows in set (0.00 sec)
```

2) Modify the credit limit to 8000 for those customers who live in „grange“.

```
mysql> SELECT*FROM CUSTOMER;
+-------------+--------------+---------------+----------+--------------+
| CUSTOMER_NO | NAME         | ADDRESS       | DEPOT_NO | CREDIT_LIMIT |
+-------------+--------------+---------------+----------+--------------+
|          10 | GARRY SMITH  | BRIXTON       |        6 |         1000 |
|          20 | PATEL        | GRANGE        |        1 |         4000 |
|          30 | DRAKE        | BRIXTON       |        4 |         7000 |
|          40 | BOB SMITH    | LONDON        |        2 |        10000 |
|          50 | JAMES        | GRANGE        |        3 |         5000 |
|          60 | NORTON       | SAN FRANSISCO |        5 |        17000 |
|          70 | JOHN MICHAEL | EUROPE        |       16 |         8000 |
+-------------+--------------+---------------+----------+--------------+
7 rows in set (0.00 sec)
```

```
mysql> UPDATE CUSTOMER SET CREDIT_LIMIT = 8000 WHERE ADDRESS = "GRANGE";
Query OK, 2 rows affected (0.04 sec)
Rows matched: 2  Changed: 2  Warnings: 0

mysql> SELECT*FROM CUSTOMER;
+-------------+--------------+---------------+----------+--------------+
| CUSTOMER_NO | NAME         | ADDRESS       | DEPOT_NO | CREDIT_LIMIT |
+-------------+--------------+---------------+----------+--------------+
|          10 | GARRY SMITH  | BRIXTON       |        6 |         1000 |
|          20 | PATEL        | GRANGE        |        1 |         8000 |
|          30 | DRAKE        | BRIXTON       |        4 |         7000 |
|          40 | BOB SMITH    | LONDON        |        2 |        10000 |
|          50 | JAMES        | GRANGE        |        3 |         8000 |
|          60 | NORTON       | SAN FRANSISCO |        5 |        17000 |
|          70 | JOHN MICHAEL | EUROPE        |       16 |         8000 |
+-------------+--------------+---------------+----------+--------------+
7 rows in set (0.00 sec)
```

3) Change the size of the customer address to 30.

```
mysql> SELECT*FROM CUSTOMER;
+-------------+--------------+---------------+----------+--------------+
| CUSTOMER_NO | NAME         | ADDRESS       | DEPOT_NO | CREDIT_LIMIT |
+-------------+--------------+---------------+----------+--------------+
|          10 | GARRY SMITH  | BRIXTON       |        6 |         1000 |
|          20 | PATEL        | GRANGE        |        1 |         4000 |
|          30 | DRAKE        | BRIXTON       |        4 |         7000 |
|          40 | BOB SMITH    | LONDON        |        2 |        10000 |
|          50 | JAMES        | GRANGE        |        3 |         5000 |
|          60 | NORTON       | SAN FRANSISCO |        5 |        17000 |
|          70 | JOHN MICHAEL | EUROPE        |       16 |         8000 |
+-------------+--------------+---------------+----------+--------------+
7 rows in set (0.00 sec)
```

```
mysql> ALTER TABLE CUSTOMER MODIFY ADDRESS VARCHAR(30);
Query OK, 7 rows affected (1.22 sec)
Records: 7  Duplicates: 0  Warnings: 0

mysql> SELECT*FROM CUSTOMER;
+-------------+--------------+---------------+----------+--------------+
| CUSTOMER_NO | NAME         | ADDRESS       | DEPOT_NO | CREDIT_LIMIT |
+-------------+--------------+---------------+----------+--------------+
|          10 | GARRY SMITH  | BRIXTON       |        6 |         1000 |
|          20 | PATEL        | GRANGE        |        1 |         8000 |
|          30 | DRAKE        | BRIXTON       |        4 |         7000 |
|          40 | BOB SMITH    | LONDON        |        2 |        10000 |
|          50 | JAMES        | GRANGE        |        3 |         8000 |
|          60 | NORTON       | SAN FRANSISCO |        5 |        17000 |
|          70 | JOHN MICHAEL | EUROPE        |       16 |         8000 |
+-------------+--------------+---------------+----------+--------------+
7 rows in set (0.00 sec)
```

4) Create a table cust1 with the attributes and formats

Customer_no number (10)

Name varchar2 (20)

Address varchar2 (20)

Rep_no number (10)

```
mysql> CREATE TABLE CUST1(CUSTOMER_NO INT(10),NAME VARCHAR(20),ADDRESS VARCHAR(20),REP_NO INT(10));
Query OK, 0 rows affected (0.47 sec)

mysql> SELECT*FROM CUST1;
Empty set (0.00 sec)
```

5) Add a new field email id in the cust1 table.

```
mysql> ALTER TABLE CUST1 ADD EMAIL_ID VARCHAR(30);
Query OK, 0 rows affected (0.74 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> SELECT*FROM CUST1;
Empty set (0.00 sec)
```

6) Display the structure of the cust1 table.

```
mysql> DESC CUST1;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| CUSTOMER_NO | int(10)     | YES  |     | NULL    |       |
| NAME        | varchar(20) | YES  |     | NULL    |       |
| ADDRESS     | varchar(20) | YES  |     | NULL    |       |
| REP_NO      | int(10)     | YES  |     | NULL    |       |
| EMAIL_ID    | varchar(30) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

7) Display the content of the cust1 table.

```
mysql> INSERT INTO CUST1 VALUES(1,'ABC','SURAT',123,'ABCDEF'),(2,'SDF','SURAT',456,'QWEERTY');
Query OK, 2 rows affected (0.05 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> SELECT*FROM CUST1;
+-------------+------+---------+--------+----------+
| CUSTOMER_NO | NAME | ADDRESS | REP_NO | EMAIL_ID |
+-------------+------+---------+--------+----------+
|           1 | ABC  | SURAT   |    123 | ABCDEF   |
|           2 | SDF  | SURAT   |    456 | QWEERTY  |
+-------------+------+---------+--------+----------+
2 rows in set (0.00 sec)
```

8) Delete details of customer no 2 from cust1 table.

```
mysql> DELETE FROM CUST1 WHERE CUSTOMER_NO=2;
Query OK, 1 row affected (0.05 sec)

mysql> SELECT*FROM CUST1;
+-------------+------+---------+--------+----------+
| CUSTOMER_NO | NAME | ADDRESS | REP_NO | EMAIL_ID |
+-------------+------+---------+--------+----------+
|           1 | ABC  | SURAT   |    123 | ABCDEF   |
+-------------+------+---------+--------+----------+
1 row in set (0.00 sec)
```

9) Delete email id field from cust1 table.

```
mysql> SELECT*FROM CUST1;
+-------------+------+---------+--------+----------+
| CUSTOMER_NO | NAME | ADDRESS | REP_NO | EMAIL_ID |
+-------------+------+---------+--------+----------+
|           1 | ABC  | SURAT   |    123 | ABCDEF   |
+-------------+------+---------+--------+----------+
1 row in set (0.00 sec)

mysql> ALTER TABLE CUST1 DROP COLUMN EMAIL_ID;
Query OK, 0 rows affected (0.77 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> SELECT*FROM CUST1;
+-------------+------+---------+--------+
| CUSTOMER_NO | NAME | ADDRESS | REP_NO |
+-------------+------+---------+--------+
|           1 | ABC  | SURAT   |    123 |
+-------------+------+---------+--------+
1 row in set (0.00 sec)
```

10) Delete all the data rows from cust1 and look at the contents again.

```
mysql> DELETE FROM CUST1;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT*FROM CUST1;
Empty set (0.00 sec)
```

11) Delete the table cust1 and then try to look at its contents again.

```
mysql> DROP TABLE CUST1;
Query OK, 0 rows affected (0.18 sec)

mysql> SELECT * FROM CUST1;
ERROR 1146 (42S02): Table 'TUTORIAL1.CUST1' doesn't exist
```

12) List the customer numbers (customer_no) and names (name) of all customers.

```
mysql> SELECT*FROM CUSTOMER;
+-------------+--------------+---------------+----------+--------------+
| CUSTOMER_NO | NAME         | ADDRESS       | DEPOT_NO | CREDIT_LIMIT |
+-------------+--------------+---------------+----------+--------------+
|          10 | GARRY SMITH  | BRIXTON       |        6 |         1000 |
|          20 | PATEL        | GRANGE        |        1 |         8000 |
|          30 | DRAKE        | BRIXTON       |        4 |         7000 |
|          40 | BOB SMITH    | LONDON        |        2 |        10000 |
|          50 | JAMES        | GRANGE        |        3 |         8000 |
|          60 | NORTON       | SAN FRANSISCO |        5 |        17000 |
|          70 | JOHN MICHAEL | EUROPE        |       16 |         8000 |
+-------------+--------------+---------------+----------+--------------+
7 rows in set (0.00 sec)

mysql> SELECT CUSTOMER_NO,NAME FROM CUSTOMER;
+-------------+--------------+
| CUSTOMER_NO | NAME         |
+-------------+--------------+
|          10 | GARRY SMITH  |
|          20 | PATEL        |
|          30 | DRAKE        |
|          40 | BOB SMITH    |
|          50 | JAMES        |
|          60 | NORTON       |
|          70 | JOHN MICHAEL |
+-------------+--------------+
7 rows in set (0.00 sec)
```

13) List all details of the product with a product number (product_no) of 121 and 136.

```
mysql> SELECT*FROM PRODUCT;
+------------+---------------+-------+-------------+-----------------+----------------+
| PRODUCT_NO | DESCRIPTION   | PRICE | SUPPLIER_NO | MARKETING_REP_NO | SUPPLY_DEPT_NO |
+------------+---------------+-------+-------------+-----------------+----------------+
|        120 | REDUCER       |  1200 |        1005 |                5 |              6 |
|        121 | PLATE         |  1500 |        1004 |                3 |              1 |
|        122 | HANDLE        |   700 |        1003 |                2 |              4 |
|        124 | WIDGET REMOVER |   900 |        1005 |                4 |              2 |
|        136 | SIZE WIDGET   |  1000 |        1001 |                1 |              5 |
|        137 | SIZE WIDGET   | 15000 |        1002 |                2 |             16 |
+------------+---------------+-------+-------------+-----------------+----------------+
6 rows in set (0.00 sec)
```

```
mysql> SELECT*FROM PRODUCT WHERE PRODUCT_NO=121 OR PRODUCT_NO=136;
+------------+-------------+-------+-------------+-----------------+----------------+
| PRODUCT_NO | DESCRIPTION | PRICE | SUPPLIER_NO | MARKETING_REP_NO | SUPPLY_DEPT_NO |
+------------+-------------+-------+-------------+-----------------+----------------+
|        121 | PLATE       |  2000 |        1004 |                3 |              1 |
|        136 | SIZE WIDGET |  1000 |        1001 |                1 |              5 |
+------------+-------------+-------+-------------+-----------------+----------------+
2 rows in set (0.00 sec)
```

14) List all details of depots with rep 5 as their rep(rep_no).

```
mysql> SELECT*FROM DEPOT;
+----------+-------------+---------+--------+
| DEPOT_NO | LOCATION    | ADDRESS | REP_NO |
+----------+-------------+---------+--------+
|        1 | NORTH       | UK      |      1 |
|        2 | SOUTH       | USA     |      2 |
|        3 | LONDON WEST | USA     |      3 |
|        4 | EAST        | NZ      |      4 |
|        5 | WALES       | UK      |      5 |
|        6 | NORTH       | KENYA   |      6 |
|       16 | SOUTH       | UK      |      2 |
+----------+-------------+---------+--------+
7 rows in set (0.00 sec)

mysql> SELECT*FROM DEPOT WHERE REP_NO=5;
+----------+----------+---------+--------+
| DEPOT_NO | LOCATION | ADDRESS | REP_NO |
+----------+----------+---------+--------+
|        5 | WALES    | UK      |      5 |
+----------+----------+---------+--------+
1 row in set (0.00 sec)
```

15) List the product number (product_no) and description only of all products from supplier number 1005 (supplier_no).

```
mysql> SELECT*FROM PRODUCT;
+------------+----------------+-------+-------------+-----------------+----------------+
| PRODUCT_NO | DESCRIPTION    | PRICE | SUPPLIER_NO | MARKETING_REP_NO | SUPPLY_DEPT_NO |
+------------+----------------+-------+-------------+-----------------+----------------+
|        120 | REDUCER        |  1200 |        1005 |               5 |              6 |
|        121 | PLATE          |  2000 |        1004 |               3 |              1 |
|        122 | HANDLE         |   700 |        1003 |               2 |              4 |
|        124 | WIDGET REMOVER |   900 |        1005 |               4 |              2 |
|        136 | SIZE WIDGET    |  1000 |        1001 |               1 |              5 |
|        137 | SIZE WIDGET    | 15000 |        1002 |               2 |             16 |
+------------+----------------+-------+-------------+-----------------+----------------+
6 rows in set (0.00 sec)

mysql> SELECT PRODUCT_NO,DESCRIPTION FROM PRODUCT WHERE SUPPLIER_NO=1005;
+------------+----------------+
| PRODUCT_NO | DESCRIPTION    |
+------------+----------------+
|        120 | REDUCER        |
|        124 | WIDGET REMOVER |
+------------+----------------+
2 rows in set (0.00 sec)
```

16)List the sales rep number (rep_no), depot number and address for depots located at NORTH and address is UK.

```
mysql> SELECT*FROM DEPOT;
+----------+-------------+---------+--------+
| DEPOT_NO | LOCATION    | ADDRESS | REP_NO |
+----------+-------------+---------+--------+
|        1 | NORTH       | UK      |      1 |
|        2 | SOUTH       | USA     |      2 |
|        3 | LONDON WEST | USA     |      3 |
|        4 | EAST        | NZ      |      4 |
|        5 | WALES       | UK      |      5 |
|        6 | NORTH       | KENYA   |      6 |
|       16 | SOUTH       | UK      |      2 |
+----------+-------------+---------+--------+
7 rows in set (0.00 sec)

mysql> SELECT DEPOT_NO,REP_NO FROM DEPOT WHERE LOCATION='NORTH' AND ADDRESS='UK';
+----------+--------+
| DEPOT_NO | REP_NO |
+----------+--------+
|        1 |      1 |
+----------+--------+
1 row in set (0.00 sec)
```

Conclusion:

Mastering basic SQL commands enables creating, accessing, and modifying databases. DDL crafts the structure, DML handles data. Balancing efficacy with data integrity and security ensures successful management.

Practical No.2

Aim: The aim of this practical exercise is to develop practical skills in querying a relational database. Through this practical we will gain hands-on experience in retrieving specific information from a database using SQL queries.

Theory:

The theory behind this practical exercise is to understand and apply Structured Query Language (SQL) for data retrieval. Participants will learn how to write SQL queries to extract specific data from relational databases, focusing on SELECT statements, filtering conditions, and pattern matching using SQL.

1) List the customer numbers (customer_no) and names (name) of all customers.

```
mysql> /* 202203103510303 */
mysql> SELECT * FROM CUSTOMER;
+-------------+--------------+---------------+----------+--------------+
| CUSTOMER_NO | NAME         | ADDRESS       | DEPOT_NO | CREDIT_LIMIT |
+-------------+--------------+---------------+----------+--------------+
|          10 | GARRY SMITH  | BRIXTON       |        6 |         1000 |
|          20 | PATEL        | GRANGE        |        1 |         8000 |
|          30 | DRAKE        | BRIXTON       |        4 |         7000 |
|          40 | BOB SMITH    | LONDON        |        2 |        10000 |
|          50 | JAMES        | GRANGE        |        3 |         8000 |
|          60 | NORTON       | SAN FRANSISCO |        5 |        17000 |
|          70 | JOHN MICHAEL | EUROPE        |       16 |         8000 |
+-------------+--------------+---------------+----------+--------------+
7 rows in set (0.00 sec)

mysql> SELECT CUSTOMER_NO , NAME FROM CUSTOMER;
+-------------+--------------+
| CUSTOMER_NO | NAME         |
+-------------+--------------+
|          10 | GARRY SMITH  |
|          20 | PATEL        |
|          30 | DRAKE        |
|          40 | BOB SMITH    |
|          50 | JAMES        |
|          60 | NORTON       |
|          70 | JOHN MICHAEL |
+-------------+--------------+
7 rows in set (0.00 sec)
```

2) List all details of the product with a product number (product_no) of 121 and 136.

```
mysql> /* 202203103510303 */
mysql> SELECT * FROM PRODUCT;
+------------+---------------+-------+------------+----------------+----------------+
| PRODUCT_NO | DESCRIPTION   | PRICE | SUPPLIER_NO | MARKETING_REP_NO | SUPPLY_DEPOT_NO |
+------------+---------------+-------+------------+----------------+----------------+
|        120 | REDUCER       |  1200 |       1005 |              5 |              6 |
|        121 | PLATE         |  2000 |       1004 |              3 |              1 |
|        122 | HANDEL        |   700 |       1003 |              2 |              4 |
|        124 | WIDGET REMOVER |   900 |       1005 |              4 |              2 |
|        136 | SIZE WIDGET   |  1000 |       1001 |              1 |              5 |
|        137 | SIZE WIDGET   | 15000 |       1002 |              2 |             16 |
+------------+---------------+-------+------------+----------------+----------------+
6 rows in set (0.00 sec)

mysql> SELECT * FROM PRODUCT WHERE PRODUCT_NO > 120 AND PRODUCT_NO < 137;
+------------+---------------+-------+------------+----------------+----------------+
| PRODUCT_NO | DESCRIPTION   | PRICE | SUPPLIER_NO | MARKETING_REP_NO | SUPPLY_DEPOT_NO |
+------------+---------------+-------+------------+----------------+----------------+
|        121 | PLATE         |  2000 |       1004 |              3 |              1 |
|        122 | HANDEL        |   700 |       1003 |              2 |              4 |
|        124 | WIDGET REMOVER |   900 |       1005 |              4 |              2 |
|        136 | SIZE WIDGET   |  1000 |       1001 |              1 |              5 |
+------------+---------------+-------+------------+----------------+----------------+
4 rows in set (0.02 sec)
```

3) List all details of depots with rep 5 as their rep(rep_no).

```
mysql> /* 202203103510303 */
mysql> SELECT * FROM DEPOT;
+----------+-------------+---------+---------+
| DEPOT_NO | LOCATION    | ADDRESS | REP_NO |
+----------+-------------+---------+---------+
|        1 | NORTH       | UK      |       1 |
|        2 | SOUTH       | USA     |       2 |
|        3 | LONDON WEST | USA     |       3 |
|        4 | EAST        | NZ      |       4 |
|        5 | WALES       | UK      |       5 |
|        6 | NORTH       | KENYA   |       6 |
|       16 | SOUTH       | UK      |       2 |
+----------+-------------+---------+---------+
7 rows in set (0.00 sec)

mysql> SELECT * FROM DEPOT WHERE REP_NO=5 ;
+----------+----------+---------+---------+
| DEPOT_NO | LOCATION | ADDRESS | REP_NO |
+----------+----------+---------+---------+
|        5 | WALES    | UK      |       5 |
+----------+----------+---------+---------+
1 row in set (0.00 sec)
```

4) List the product number (product_no) and description only of all products from supplier number 1005 (supplier_no).

```
mysql> /* 202203103510303 */
mysql> SELECT * FROM PRODUCT;
+------------+----------------+-------+------------+-----------------+-----------------+
| PRODUCT_NO | DESCRIPTION    | PRICE | SUPPLIER_NO | MARKETING_REP_NO | SUPPLY_DEPOT_NO |
+------------+----------------+-------+------------+-----------------+-----------------+
|        120 | REDUCER        |  1200 |        1005 |               5 |               6 |
|        121 | PLATE          |  2000 |        1004 |               3 |               1 |
|        122 | HANDEL         |   700 |        1003 |               2 |               4 |
|        124 | WIDGET REMOVER |   900 |        1005 |               4 |               2 |
|        136 | SIZE WIDGET    |  1000 |        1001 |               1 |               5 |
|        137 | SIZE WIDGET    | 15000 |        1002 |               2 |              16 |
+------------+----------------+-------+------------+-----------------+-----------------+
6 rows in set (0.00 sec)

mysql> SELECT PRODUCT_NO, DESCRIPTION FROM PRODUCT WHERE SUPPLIER_NO = 1005;
+------------+----------------+
| PRODUCT_NO | DESCRIPTION    |
+------------+----------------+
|        120 | REDUCER        |
|        124 | WIDGET REMOVER |
+------------+----------------+
2 rows in set (0.00 sec)
```

5) List all details for all customers with names (name) starting from ga followed by 2 character followed by y followed by anything.

```
mysql> /* 202203103510303 */
mysql> SELECT * FROM CUSTOMER;
+-------------+--------------+---------------+----------+--------------+
| CUSTOMER_NO | NAME         | ADDRESS       | DEPOT_NO | CREDIT_LIMIT |
+-------------+--------------+---------------+----------+--------------+
|          10 | GARRY SMITH  | BRIXTON       |        6 |         1000 |
|          20 | PATEL        | GRANGE        |        1 |         8000 |
|          30 | DRAKE        | BRIXTON       |        4 |         7000 |
|          40 | BOB SMITH    | LONDON        |        2 |        10000 |
|          50 | JAMES        | GRANGE        |        3 |         8000 |
|          60 | NORTON       | SAN FRANSISCO |        5 |        17000 |
|          70 | JOHN MICHAEL | EUROPE        |       16 |         8000 |
+-------------+--------------+---------------+----------+--------------+
7 rows in set (0.00 sec)

mysql> SELECT * FROM CUSTOMER WHERE NAME LIKE 'GA__Y%';
+-------------+-------------+---------+----------+--------------+
| CUSTOMER_NO | NAME        | ADDRESS | DEPOT_NO | CREDIT_LIMIT |
+-------------+-------------+---------+----------+--------------+
|          10 | GARRY SMITH | BRIXTON |        6 |         1000 |
+-------------+-------------+---------+----------+--------------+
1 row in set (0.00 sec)
```

6) List all details for all orders with date_placed from 01-jan-1993 to 31-mar-1996.

```
mysql> /* 202203103510303 */
mysql> SELECT * FROM CORDER;
+-----------+-------------+-------------+----------------+
| CORDER_NO | CUSTOMER_NO | DATE_PLACED | DATE_DELIVERED |
+-----------+-------------+-------------+----------------+
|       200 |          20 | 01-JAN-1993 | 04-JAN-1993    |
|       201 |          40 | 17-JAN-1993 | 20-JAN-1993    |
|       202 |          20 | 01-JAN-1993 | 04-JAN-1993    |
|       203 |          30 | 02-FEB-1995 | 05-FEB-1995    |
|       204 |          10 | 13-MAR-1996 | 16-MAR-1996    |
|       205 |          70 | 31-JAN-1993 | 03-FEB-1993    |
|       206 |          40 | 01-JAN-1994 | 04-JAN-1994    |
|       207 |          20 | 02-AUG-1994 | 05-AUG-1994    |
+-----------+-------------+-------------+----------------+
8 rows in set (0.00 sec)

mysql> SELECT * FROM CORDER WHERE DATE_PLACED BETWEEN '01-JAN-1993' AND '13-MAR-1996';
+-----------+-------------+-------------+----------------+
| CORDER_NO | CUSTOMER_NO | DATE_PLACED | DATE_DELIVERED |
+-----------+-------------+-------------+----------------+
|       200 |          20 | 01-JAN-1993 | 04-JAN-1993    |
|       202 |          20 | 01-JAN-1993 | 04-JAN-1993    |
|       203 |          30 | 02-FEB-1995 | 05-FEB-1995    |
|       204 |          10 | 13-MAR-1996 | 16-MAR-1996    |
|       206 |          40 | 01-JAN-1994 | 04-JAN-1994    |
|       207 |          20 | 02-AUG-1994 | 05-AUG-1994    |
+-----------+-------------+-------------+----------------+
6 rows in set (0.00 sec)
```

7) List the sales rep number (rep_no), depot number and address for depots located at NORTH and address is UK.

```
mysql> /* 202203103510303 */
mysql> SELECT * FROM DEPOT;
+----------+------------+---------+--------+
| DEPOT_NO | LOCATION   | ADDRESS | REP_NO |
+----------+------------+---------+--------+
|        1 | NORTH      | UK      |      1 |
|        2 | SOUTH      | USA     |      2 |
|        3 | LONDON WEST | USA     |      3 |
|        4 | EAST       | NZ      |      4 |
|        5 | WALES      | UK      |      5 |
|        6 | NORTH      | KENYA   |      6 |
|       16 | SOUTH      | UK      |      2 |
+----------+------------+---------+--------+
7 rows in set (0.00 sec)

mysql> SELECT REP_NO , DEPOT_NO ,ADDRESS FROM DEPOT WHERE LOCATION='NORTH' AND ADDRESS = 'UK';
+--------+----------+---------+
| REP_NO | DEPOT_NO | ADDRESS |
+--------+----------+---------+
|      1 |        1 | UK      |
+--------+----------+---------+
1 row in set (0.00 sec)
```

8) Give the total number of items (quantity) in stock in all depots.

```
mysql> /* 202203103510303 */
mysql> SELECT * FROM STOCK;
+----------+------------+----------+------+--------+
| DEPOT_NO | PRODUCT_NO | QUANTITY | RACK | BIN_NO |
+----------+------------+----------+------+--------+
|        1 |        120 |       50 |    1 |      1 |
|        2 |        137 |      100 |   10 |      2 |
|        3 |        136 |       40 |    2 |      3 |
|        4 |        120 |       60 |    7 |      1 |
|        5 |        121 |       90 |    5 |      4 |
|        6 |        124 |      120 |    4 |      7 |
|       16 |        122 |       80 |   10 |      8 |
+----------+------------+----------+------+--------+
7 rows in set (0.00 sec)

mysql> SELECT SUM(QUANTITY) FROM STOCK;
+---------------+
| SUM(QUANTITY) |
+---------------+
|           540 |
+---------------+
1 row in set (0.03 sec)
```

9) Give the total number of items (order line quantity) which have been ordered with corder_no 200.

```
mysql> /* 202203103510303 */
mysql> SELECT * FROM OLINE;
+-----------+------------+----------+
| CORDER_NO | PRODUCT_NO | QUANTITY |
+-----------+------------+----------+
|       200 |        120 |        5 |
|       201 |        121 |       10 |
|       202 |        120 |        5 |
|       203 |        122 |       20 |
|       204 |        136 |       30 |
|       205 |        124 |       15 |
|       206 |        136 |       30 |
+-----------+------------+----------+
7 rows in set (0.00 sec)

mysql> SELECT SUM(QUANTITY) AS TOTAL_ITEMS_ORDERED FROM OLINE WHERE CORDER_NO = 200;
+---------------------+
| TOTAL_ITEMS_ORDERED |
+---------------------+
|                   5 |
+---------------------+
1 row in set (0.00 sec)
```

10) List product descriptions in reverse alphabetical order.

```
mysql> /* 202203103510303 */
mysql> SELECT * FROM PRODUCT;
+------------+---------------+-------+-------------+-----------------+-----------------+
| PRODUCT_NO | DESCRIPTION   | PRICE | SUPPLIER_NO | MARKETING_REP_NO | SUPPLY_DEPOT_NO |
+------------+---------------+-------+-------------+-----------------+-----------------+
|        120 | REDUCER       |  1200 |        1005 |               5 |               6 |
|        121 | PLATE         |  2000 |        1004 |               3 |               1 |
|        122 | HANDEL        |   700 |        1003 |               2 |               4 |
|        124 | WIDGET REMOVER |   900 |        1005 |               4 |               2 |
|        136 | SIZE WIDGET   |  1000 |        1001 |               1 |               5 |
|        137 | SIZE WIDGET   | 15000 |        1002 |               2 |              16 |
+------------+---------------+-------+-------------+-----------------+-----------------+
6 rows in set (0.00 sec)

mysql> SELECT DESCRIPTION FROM PRODUCT ORDER BY DESCRIPTION DESC;
+---------------+
| DESCRIPTION   |
+---------------+
| WIDGET REMOVER |
| SIZE WIDGET   |
| SIZE WIDGET   |
| REDUCER       |
| PLATE         |
| HANDEL        |
+---------------+
6 rows in set (0.00 sec)
```

11) List the customer details with the name ending with N.

```
mysql> /* 202203103510303 */
mysql> SELECT * FROM CUSTOMER;
+-------------+--------------+---------------+----------+--------------+
| CUSTOMER_NO | NAME         | ADDRESS       | DEPOT_NO | CREDIT_LIMIT |
+-------------+--------------+---------------+----------+--------------+
|          10 | GARRY SMITH  | BRIXTON       |        6 |         1000 |
|          20 | PATEL        | GRANGE        |        1 |         8000 |
|          30 | DRAKE        | BRIXTON       |        4 |         7000 |
|          40 | BOB SMITH    | LONDON        |        2 |        10000 |
|          50 | JAMES        | GRANGE        |        3 |         8000 |
|          60 | NORTON       | SAN FRANSISCO |        5 |        17000 |
|          70 | JOHN MICHAEL | EUROPE        |       16 |         8000 |
+-------------+--------------+---------------+----------+--------------+
7 rows in set (0.00 sec)

mysql> SELECT * FROM CUSTOMER WHERE NAME LIKE '%N';
+-------------+--------+---------------+----------+--------------+
| CUSTOMER_NO | NAME   | ADDRESS       | DEPOT_NO | CREDIT_LIMIT |
+-------------+--------+---------------+----------+--------------+
|          60 | NORTON | SAN FRANSISCO |        5 |        17000 |
+-------------+--------+---------------+----------+--------------+
1 row in set (0.00 sec)
```

12) List the customers details with a CustomerName that have “r” in the second position:

```
mysql> /* 202203103510303 */
mysql> SELECT * FROM CUSTOMER;
+-------------+--------------+---------------+----------+--------------+
| CUSTOMER_NO | NAME         | ADDRESS       | DEPOT_NO | CREDIT_LIMIT |
+-------------+--------------+---------------+----------+--------------+
|          10 | GARRY SMITH  | BRIXTON       |        6 |         1000 |
|          20 | PATEL        | GRANGE        |        1 |         8000 |
|          30 | DRAKE        | BRIXTON       |        4 |         7000 |
|          40 | BOB SMITH    | LONDON        |        2 |        10000 |
|          50 | JAMES        | GRANGE        |        3 |         8000 |
|          60 | NORTON       | SAN FRANSISCO |        5 |        17000 |
|          70 | JOHN MICHAEL | EUROPE        |       16 |         8000 |
+-------------+--------------+---------------+----------+--------------+
7 rows in set (0.00 sec)

mysql> SELECT * FROM CUSTOMER WHERE SUBSTRING(NAME,2,1)='R';
+-------------+-------+---------+----------+--------------+
| CUSTOMER_NO | NAME  | ADDRESS | DEPOT_NO | CREDIT_LIMIT |
+-------------+-------+---------+----------+--------------+
|          30 | DRAKE | BRIXTON |        4 |         7000 |
+-------------+-------+---------+----------+--------------+
1 row in set (0.00 sec)
```

13) List the customers with a Customer Name that starts with “N” and is at least 4 characters in length.

```
mysql> /* 202203103510303 */
mysql> SELECT * FROM CUSTOMER;
+-------------+--------------+---------------+----------+--------------+
| CUSTOMER_NO | NAME         | ADDRESS       | DEPOT_NO | CREDIT_LIMIT |
+-------------+--------------+---------------+----------+--------------+
|          10 | GARRY SMITH  | BRIXTON       |        6 |         1000 |
|          20 | PATEL        | GRANGE        |        1 |         8000 |
|          30 | DRAKE        | BRIXTON       |        4 |         7000 |
|          40 | BOB SMITH    | LONDON        |        2 |        10000 |
|          50 | JAMES        | GRANGE        |        3 |         8000 |
|          60 | NORTON       | SAN FRANSISCO |        5 |        17000 |
|          70 | JOHN MICHAEL | EUROPE        |       16 |         8000 |
+-------------+--------------+---------------+----------+--------------+
7 rows in set (0.00 sec)

mysql> SELECT * FROM CUSTOMER WHERE NAME LIKE '%N' AND LENGTH(NAME) >= 4;
+-------------+--------+---------------+----------+--------------+
| CUSTOMER_NO | NAME   | ADDRESS       | DEPOT_NO | CREDIT_LIMIT |
+-------------+--------+---------------+----------+--------------+
|          60 | NORTON | SAN FRANSISCO |        5 |        17000 |
+-------------+--------+---------------+----------+--------------+
1 row in set (0.00 sec)
```

14) Find all suppliers with a City containing the pattern “ny”.

```
mysql> /* 202203103510303 */
mysql> SELECT * FROM SUPPLIER;
+-------------+-----------+----------+
| SUPPLIER_NO | NAME      | ADDRESS  |
+-------------+-----------+----------+
|        1001 | MICHAEL   | BASILDON |
|        1002 | RINGWORLD | GERMANY  |
|        1003 | BABYLON   | LONDON   |
|        1004 | JOHN      | BASILDON |
|        1005 | SMITH     | GERMANY  |
+-------------+-----------+----------+
5 rows in set (0.00 sec)

mysql> SELECT * FROM SUPPLIER WHERE UPPER(ADDRESS) LIKE '%NY';
+-------------+-----------+---------+
| SUPPLIER_NO | NAME      | ADDRESS |
+-------------+-----------+---------+
|        1002 | RINGWORLD | GERMANY |
|        1005 | SMITH     | GERMANY |
+-------------+-----------+---------+
2 rows in set (0.00 sec)
```

15) selects all customers with a City starting with “L”, followed by any character, followed by “n”, followed by 2 character, followed by “n”:

```
mysql> /* 202203103510303 */
mysql> SELECT * FROM CUSTOMER;
+-------------+--------------+---------------+----------+--------------+
| CUSTOMER_NO | NAME         | ADDRESS       | DEPOT_NO | CREDIT_LIMIT |
+-------------+--------------+---------------+----------+--------------+
|          10 | GARRY SMITH  | BRIXTON       |        6 |         1000 |
|          20 | PATEL        | GRANGE        |        1 |         8000 |
|          30 | DRAKE        | BRIXTON       |        4 |         7000 |
|          40 | BOB SMITH    | LONDON        |        2 |        10000 |
|          50 | JAMES        | GRANGE        |        3 |         8000 |
|          60 | NORTON       | SAN FRANSISCO |        5 |        17000 |
|          70 | JOHN MICHAEL | EUROPE        |       16 |         8000 |
+-------------+--------------+---------------+----------+--------------+
7 rows in set (0.00 sec)

mysql> SELECT * FROM CUSTOMER WHERE ADDRESS LIKE 'L_N__N';
+-------------+-----------+---------+----------+--------------+
| CUSTOMER_NO | NAME      | ADDRESS | DEPOT_NO | CREDIT_LIMIT |
+-------------+-----------+---------+----------+--------------+
|          40 | BOB SMITH | LONDON  |        2 |        10000 |
+-------------+-----------+---------+----------+--------------+
1 row in set (0.00 sec)
```

Conclusion:

In conclusion, this practical exercise provides valuable experience in using SQL for data retrieval. Through this practical we have successfully practiced writing SQL queries to retrieve data from the given database, demonstrating their ability to select and filter data based on specific criteria.