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> /*202203103510124*/
mysql> UPDATE PRODUCT
    -> SET PRICE = 2000
    -> WHERE DESCRIPTION = 'PLATE';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from product;
+------------+----------------+----------+-------------+------------------+-----------------+
| PRODUCT_NO | DESCRIPTION    | PRICE    | SUPPLIER_NO | MARKETING_REP_NO | SUPPLY_DEPOT_NO |
+------------+----------------+----------+-------------+------------------+-----------------+
|        120 | REDUCER        |  1200.00 |        1005 |                5 |               6 |
|        121 | PLATE          |  2000.00 |        1004 |                3 |               1 |
|        122 | HANDLE         |   700.00 |        1003 |                2 |               4 |
|        124 | WIDGET REMOVER |   900.00 |        1005 |                4 |               2 |
|        136 | SIZE WIDGET    |  1000.00 |        1001 |                1 |               5 |
|        137 | SIZE WIDGET    | 15000.00 |        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> /*202203103510124*/
mysql> UPDATE CUSTOMER
    -> SET CREDIT_LIMIT = 8000
    -> WHERE ADDRESS = 'GRANGE';
Query OK, 2 rows affected (0.00 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 |      1000.00 |
|          20 | PATEL        | GRANGE        |        1 |      8000.00 |
|          30 | DRAKE        | BRIXTON       |        4 |      7000.00 |
|          40 | BOB SMITH    | LONDON        |        2 |     10000.00 |
|          50 | JAMES        | GRANGE        |        3 |      8000.00 |
|          60 | NORTON       | SAN FRANCISCO |        5 |     17000.00 |
|          70 | JOHN MICHAEL | EUROPE        |       16 |      8000.00 |
+-------------+--------------+---------------+----------+--------------+
7 rows in set (0.00 sec)
```

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

```
mysql> /*202203103510124*/
mysql> ALTER TABLE CUSTOMER
    -> MODIFY ADDRESS VARCHAR(30);
Query OK, 7 rows affected (0.02 sec)
Records: 7  Duplicates: 0  Warnings: 0
```

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> /*202203103510124*/
mysql> CREATE TABLE CUST1 (
    ->     CUSTOMER_NO int(10),
    ->     NAME VARCHAR(20),
    ->     ADDRESS VARCHAR(20),
    ->     REP_NO int(10)
    -> );
Query OK, 0 rows affected, 2 warnings (0.01 sec)
```

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

```
mysql> /*202203103510124*/
mysql> ALTER TABLE CUST1
    -> ADD EMAIL_ID VARCHAR(50);
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

6) Display the structure of the cust1 table.

```
mysql> /*202203103510124*/
mysql> DESC CUST1;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| CUSTOMER_NO | int         | YES  |     | NULL    |       |
| NAME        | varchar(20) | YES  |     | NULL    |       |
| ADDRESS     | varchar(20) | YES  |     | NULL    |       |
| REP_NO      | int         | YES  |     | NULL    |       |
| EMAIL_ID    | varchar(50) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
5 rows in set (0.01 sec)
```

7) Display the content of the cust1 table.

```
mysql> /*202203103510124*/
mysql> SELECT * FROM CUST1;
Empty set (0.00 sec)
```

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

```
mysql> /*202203103510124*/
mysql> DELETE FROM CUST1
    -> WHERE CUSTOMER_NO = 2;
Query OK, 0 rows affected (0.00 sec)
```

9) Delete email id field from cust1 table.

```
mysql> /*202203103510124*/
mysql> ALTER TABLE CUST1
    -> DROP COLUMN EMAIL_ID;
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

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

```
mysql> /*202203103510124*/
mysql> DELETE FROM CUST1;
Query OK, 0 rows affected (0.00 sec)
```

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

```
mysql> /*202203103510124*/
mysql> DROP TABLE CUST1;
Query OK, 0 rows affected (0.01 sec)
```

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

```
mysql> /*202203103510124*/
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.(use Or).

```
mysql> /*202203103510124*/
mysql> SELECT *
    -> FROM PRODUCT
    -> WHERE PRODUCT_NO IN (121, 136);
+------------+-------------+---------+-------------+------------------+-----------------+
| PRODUCT_NO | DESCRIPTION | PRICE   | SUPPLIER_NO | MARKETING_REP_NO | SUPPLY_DEPOT_NO |
+------------+-------------+---------+-------------+------------------+-----------------+
|        121 | PLATE       | 2000.00 |        1004 |                3 |               1 |
|        136 | SIZE WIDGET | 1000.00 |        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> /*202203103510124*/
mysql> SELECT *
    -> FROM DEPOT
    -> WHERE REP_NO = 5;
+----------+----------+---------+--------+
| DEPOT_NO | LOCATION | ADDRESS | REP_NO |
+----------+----------+---------+--------+
|        5 | WALES UK | 5       |      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> /*202203103510124*/
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> /*202203103510124*/
mysql> SELECT REP_NO, DEPOT_NO, ADDRESS
    -> FROM DEPOT
    -> WHERE LOCATION = 'NORTH' AND ADDRESS = 'UK';
Empty 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.