Experiment 1

TITLE: DDL (Data Definition Language) commands

Objective: To understand the concept of designing issue related to the database with creating, populating the tables.

1. **Create the tables described below:**

Table name: CLIENT_MASTER
Description: used to store client information.

Column name	data type	Size
CLIENTNO	Varchar	6
NAME	Varchar	20
ADDRESS 1	Varchar	30
ADDRESS 2	Varchar	30
CITY	Varchar	15
PINCODE	Integer	
STATE	Varchar	15
BALDUE	decimal	10,2

Commands:

```
CREATE TABLE CLIENT_MASTER (
   CLIENTNO VARCHAR(6),
   NAME VARCHAR(20),
   ADDRESS1 VARCHAR(30),
   ADDRESS2 VARCHAR(30),
   CITY VARCHAR(15),
   PINCODE INTEGER,
   STATE VARCHAR(15),
   BALDUE
 DECIMAL(10, 2)
);
```

Output:

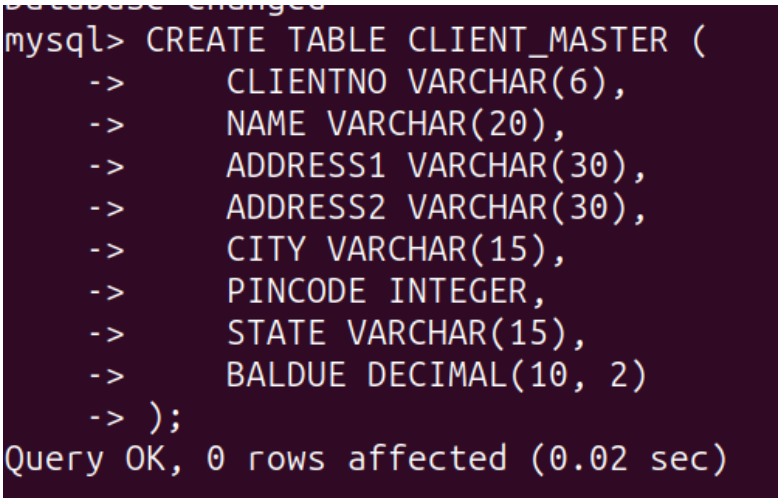


Table Name: PRODUCT_MASTER
Description: used to store product information

Column name	data type	Size
PRODUCTNO	Varchar	6
DESCRIPTION	Varchar	15
PROFITPERCENT	Decimal	4,2
UNIT MEASURE	Varchar	10
QTYONHAND	Integer	
REORDERL VL	Integer	
SELLPRICE	Decimal	8,2
COSTPRICE	Decimal	8,2

Commands:

```
CREATE TABLE PRODUCT_MASTER (
   PRODUCTNO VARCHAR(6),
   DESCRIPTION VARCHAR(15),
   PROFITPERCENT DECIMAL(4, 2),
   UNIT_MEASURE VARCHAR(10),
   QTYONHAND INTEGER,
   REORDERLVL INTEGER,
   SELLPRICE DECIMAL(8, 2),
   COSTPRICE DECIMAL(8, 2)
);
```

Output:

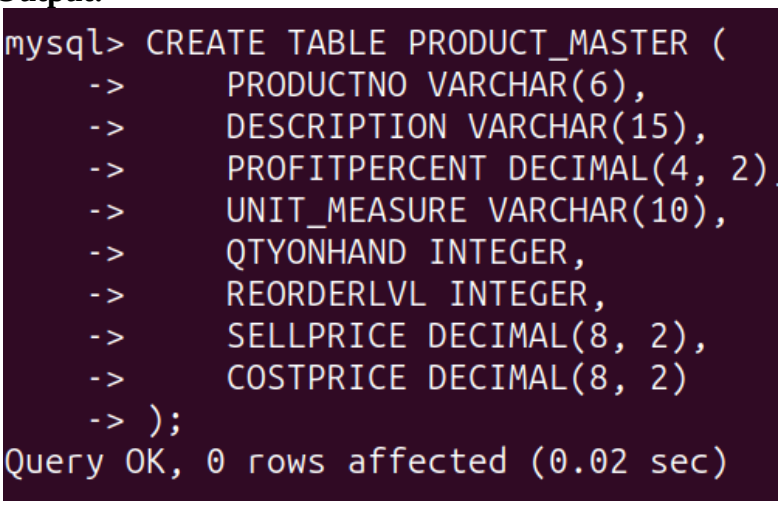


Table Name: SALESMAN_MASTER
Description: Used to store salesman information working for the company.

Column name	data type	Size
SALESMANNO	Varchar	6
SALESMANNAME	Varchar	20

ADDRESS 1	Varchar	30
ADDRESS 2	Varchar	30
CITY	Varchar	20
PINCODE	Integer	
STATE	Varchar	20
SALAMT	Real	
TGTTOGET	Decimal	
YTDSALES	Double	6,2
REMARKS	Varchar	60

Commands:

```
CREATE TABLE SALESMAN_MASTER (
   SALESMANNO VARCHAR(6),
   SALESMANNAME VARCHAR(20),
   ADDRESS1 VARCHAR(30),
   ADDRESS2 VARCHAR(30),
   CITY VARCHAR(20),
   PINCODE INTEGER,
   STATE VARCHAR(20),
   SALAMT REAL,
   TGTTOGET DECIMAL(10, 0),
   YTDSALES DOUBLE(6, 2),
   REMARKS VARCHAR(60)
);
```

Output:

```
mysql> CREATE TABLE SALESMAN_MASTER (
    ->     SALESMANNO VARCHAR(6),
    ->     SALESMANNAME VARCHAR(20),
    ->     ADDRESS1 VARCHAR(30),
    ->     ADDRESS2 VARCHAR(30),
    ->     CITY VARCHAR(20),
    ->     PINCODE INTEGER,
    ->     STATE VARCHAR(20),
    ->     SALAMT REAL,
    ->     TGTTOGET DECIMAL(10, 0),
    ->     YTDSALES DOUBLE(6, 2),
    ->     REMARKS VARCHAR(60)
    -> );
Query OK, 0 rows affected, 1 warning (0.02 sec)
```

```
mysql> show tables
    -> ;
+-----------------+
| Tables_in_exp1  |
+-----------------+
| CLIENT_MASTER   |
| PRODUCT_MASTER  |
| SALESMAN_MASTER |
+-----------------+
3 rows in set (0.00 sec)
```

2. **Insert the following data into their respective tables:**

a) Data for **CLIENT_MASTER** table:

Client no	Name	city	Pincode	state	BalDue
C00001	Ivan bayross	Mumbai	400054	Maharashtra	15000
C00002	Mamta muzumdar	Madras	780001	Tamil nadu	0
C00003	Chhaya bankar	Mumbai	400057	Maharashtra	5000
C00004	Ashwini joshi	Bangalore	560001	Karnataka	0
C00005	Hansel colaco	Mumbai	400060	Maharashtra	2000
C00006	Deepak sharma	Mangalore	560050	Karnataka	0

Commands:

a.

```
INSERT INTO CLIENT_MASTER (CLIENTNO, NAME, CITY, PINCODE, STATE, BALDUE) VALUES
('C00001', 'Ivan Bayross', 'Mumbai', 400054, 'Maharashtra', 15000),
('C00002', 'Mamta Muzumdar', 'Madras', 780001, 'Tamil Nadu', 0),
('C00003', 'Chhaya Bankar', 'Mumbai', 400057, 'Maharashtra', 5000),
('C00004', 'Ashwini Joshi', 'Bangalore', 560001, 'Karnataka', 0),
('C00005', 'Hansel Colaco', 'Mumbai', 400060, 'Maharashtra', 2000),
('C00006', 'Deepak Sharma', 'Mangalore', 560050, 'Karnataka', 0);
```

Output:

```
mysql> INSERT INTO CLIENT_MASTER (CLIENTNO, NAME, CITY, PINCODE, STATE, BALDUE) VALUES
    -> ('C00001', 'Ivan Bayross', 'Mumbai', 400054, 'Maharashtra', 15000),
    -> ('C00002', 'Mamta Muzumdar', 'Madras', 780001, 'Tamil Nadu', 0),
    -> ('C00003', 'Chhaya Bankar', 'Mumbai', 400057, 'Maharashtra', 5000),
    -> ('C00004', 'Ashwini Joshi', 'Bangalore', 560001, 'Karnataka', 0),
    -> ('C00005', 'Hansel Colaco', 'Mumbai', 400060, 'Maharashtra', 2000),
    -> ('C00006', 'Deepak Sharma', 'Mangalore', 560050, 'Karnataka', 0);
Query OK, 6 rows affected (0.02 sec)
Records: 6  Duplicates: 0  Warnings: 0
```

```
mysql> select * from CLIENT_MASTER;
+----------+----------------+----------+----------+-----------+---------+-------------+----------+
| CLIENTNO | NAME           | ADDRESS1 | ADDRESS2 | CITY      | PINCODE | STATE       | BALDUE   |
+----------+----------------+----------+----------+-----------+---------+-------------+----------+
| C00001   | Ivan Bayross   | NULL     | NULL     | Mumbai    |  400054 | Maharashtra | 15000.00 |
| C00002   | Mamta Muzumdar | NULL     | NULL     | Madras    |  780001 | Tamil Nadu  |     0.00 |
| C00003   | Chhaya Bankar  | NULL     | NULL     | Mumbai    |  400057 | Maharashtra |  5000.00 |
| C00004   | Ashwini Joshi  | NULL     | NULL     | Bangalore |  560001 | Karnataka   |     0.00 |
| C00005   | Hansel Colaco  | NULL     | NULL     | Mumbai    |  400060 | Maharashtra |  2000.00 |
| C00006   | Deepak Sharma  | NULL     | NULL     | Mangalore |  560050 | Karnataka   |     0.00 |
+----------+----------------+----------+----------+-----------+---------+-------------+----------+
6 rows in set (0.00 sec)
```

b) Data for **PRODUCT_MASTER** table:

ProductNo	Description	Profit percent	Unit measure	Qtyonhand	RecorderLvl	SellPrice	CostPrice

P00001	T-Shirt	5	Piece	200	50	350	250
P0345	Shirts	6	Piece	150	50	500	350
P06734	Cotton jeans	5	Piece	100	20	600	450
P07865	Jeans	5	Piece	100	20	750	500
P07868	Trousers	2	Piece	150	50	850	550
P07885	Pull Overs	2.5	Piece	80	30	700	450
P07965	Denim jeans	4	Piece	100	40	350	250
P07975	Lycra tops	5	Piece	70	30	300	175
P08865	Skirts	5	Piece	75	30	450	300

Commands:

INSERT INTO PRODUCT_MASTER (PRODUCTNO, DESCRIPTION, PROFITPERCENT, UNIT_MEASURE, QTYONHAND, REORDERLVL, SELLPRICE, COSTPRICE) VALUES
('P00001', 'T-Shirt', 5, 'Piece', 200, 50, 350, 250),
('P0345', 'Shirts', 6, 'Piece', 150, 50, 500, 350),
('P06734', 'Cotton jeans', 5, 'Piece', 100, 20, 600, 450),
('P07865', 'Jeans', 5, 'Piece', 100, 20, 750, 500),
('P07868', 'Trousers', 2, 'Piece', 150, 50, 850, 550),
('P07885', 'Pull Overs', 2.5, 'Piece', 80, 30, 700, 450),
('P07965', 'Denim jeans', 4, 'Piece', 100, 40, 350, 250),
('P07975', 'Lycra tops', 5, 'Piece', 70, 30, 300, 175),
('P08865', 'Skirts', 5, 'Piece', 75, 30, 450, 300);

Output:

```
mysql> INSERT INTO PRODUCT_MASTER (PRODUCTNO, DESCRIPTION, PROFITPERCENT, UNIT_MEASURE, QTYONHAND, REORDERLVL, SELLPRICE, COSTPRICE) VALUES
    -> ('P00001', 'T-Shirt', 5, 'Piece', 200, 50, 350, 250),
    -> ('P0345', 'Shirts', 6, 'Piece', 150, 50, 500, 350),
    -> ('P06734', 'Cotton jeans', 5, 'Piece', 100, 20, 600, 450),
    -> ('P07865', 'Jeans', 5, 'Piece', 100, 20, 750, 500),
    -> ('P07868', 'Trousers', 2, 'Piece', 150, 50, 850, 550),
    -> ('P07885', 'Pull Overs', 2.5, 'Piece', 80, 30, 700, 450),
    -> ('P07965', 'Denim jeans', 4, 'Piece', 100, 40, 350, 250),
    -> ('P07975', 'Lycra tops', 5, 'Piece', 70, 30, 300, 175),
    -> ('P08865', 'Skirts', 5, 'Piece', 75, 30, 450, 300);
Query OK, 9 rows affected (0.01 sec)
Records: 9  Duplicates: 0  Warnings: 0
```

```
mysql> select * from PRODUCT_MASTER;
+-----------+--------------+---------------+--------------+-----------+------------+-----------+-----------+
| PRODUCTNO | DESCRIPTION  | PROFITPERCENT | UNIT_MEASURE | QTYONHAND | REORDERLVL | SELLPRICE | COSTPRICE |
+-----------+--------------+---------------+--------------+-----------+------------+-----------+-----------+
| P00001    | T-Shirt      |          5.00 | Piece        |       200 |         50 |    350.00 |    250.00 |
| P0345     | Shirts       |          6.00 | Piece        |       150 |         50 |    500.00 |    350.00 |
| P06734    | Cotton jeans |          5.00 | Piece        |       100 |         20 |    600.00 |    450.00 |
| P07865    | Jeans        |          5.00 | Piece        |       100 |         20 |    750.00 |    500.00 |
| P07868    | Trousers     |          2.00 | Piece        |       150 |         50 |    850.00 |    550.00 |
| P07885    | Pull Overs   |          2.50 | Piece        |        80 |         30 |    700.00 |    450.00 |
| P07965    | Denim jeans  |          4.00 | Piece        |       100 |         40 |    350.00 |    250.00 |
| P07975    | Lycra tops   |          5.00 | Piece        |        70 |         30 |    300.00 |    175.00 |
| P08865    | Skirts       |          5.00 | Piece        |        75 |         30 |    450.00 |    300.00 |
+-----------+--------------+---------------+--------------+-----------+------------+-----------+-----------+
9 rows in set (0.00 sec)
```

c) Data for **SALESMAN_MASTER** table:

SalesmanNo	Name	Address1	Address2	City	PinCode	State
S00001	Aman	A/14	Worli	Mumbai	400002	Maharashtra
S00002	Omkar	65	Nariman	Mumbai	400001	Maharashtra
S00003	Raj	P-7	Bandra	Mumbai	400032	Maharashtra

S00004	Ashish	A/5	Juhu	Mumbai	400044	Maharashtra

SalesmanNo	SalAmt	TgtToGet	YtdSales	Remarks
S00001	3000	100	50	Good
S00002	3000	200	100	Good
S00003	3000	200	100	Good
S00004	3500	200	150	Good

Commands:

INSERT INTO SALESMAN_MASTER (SALESMANNO, SALESMANNAME, ADDRESS1, ADDRESS2, CITY, PINCODE, STATE, SALAMT, TGTTOGET, YTDSALES, REMARKS) VALUES
('S00001', 'Aman', 'A/14', 'Worli', 'Mumbai', 400002, 'Maharashtra', 3000, 100, 50, 'Good'),
('S00002', 'Omkar', '65', 'Nariman', 'Mumbai', 400001, 'Maharashtra', 3000, 200, 100, 'Good'),
('S00003', 'Raj', 'P-7', 'Bandra', 'Mumbai', 400032, 'Maharashtra', 3000, 200, 100, 'Good'),
('S00004', 'Ashish', 'A/5', 'Juhu', 'Mumbai', 400044, 'Maharashtra', 3500, 200, 150, 'Good');

Output:

```
mysql> INSERT INTO SALESMAN_MASTER (SALESMANNO, SALESMANNAME, ADDRESS1, ADDRESS2, CITY, PINCODE, STATE, SALAMT, TGTTOGET, YTDSALES, REMARKS) VALUE
S
    -> ('S00001', 'Aman', 'A/14', 'Worli', 'Mumbai', 400002, 'Maharashtra', 3000, 100, 50, 'Good'),
    -> ('S00002', 'Omkar', '65', 'Nariman', 'Mumbai', 400001, 'Maharashtra', 3000, 200, 100, 'Good'),
    -> ('S00003', 'Raj', 'P-7', 'Bandra', 'Mumbai', 400032, 'Maharashtra', 3000, 200, 100, 'Good'),
    -> ('S00004', 'Ashish', 'A/5', 'Juhu', 'Mumbai', 400044, 'Maharashtra', 3500, 200, 150, 'Good');
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0
```

```
mysql> select * from SALESMAN_MASTER;
+------------+--------------+----------+----------+--------+---------+-------------+--------+----------+----------+---------+
| SALESMANNO | SALESMANNAME | ADDRESS1 | ADDRESS2 | CITY   | PINCODE | STATE       | SALAMT | TGTTOGET | YTDSALES | REMARKS |
+------------+--------------+----------+----------+--------+---------+-------------+--------+----------+----------+---------+
| S00001     | Aman         | A/14     | Worli    | Mumbai |  400002 | Maharashtra |   3000 |      100 |    50.00 | Good    |
| S00002     | Omkar        | 65       | Nariman  | Mumbai |  400001 | Maharashtra |   3000 |      200 |   100.00 | Good    |
| S00003     | Raj          | P-7      | Bandra   | Mumbai |  400032 | Maharashtra |   3000 |      200 |   100.00 | Good    |
| S00004     | Ashish       | A/5      | Juhu     | Mumbai |  400044 | Maharashtra |   3500 |      200 |   150.00 | Good    |
+------------+--------------+----------+----------+--------+---------+-------------+--------+----------+----------+---------+
4 rows in set (0.00 sec)
```

Experiment 2

Title: DML commands with constraints

Objective: - To understand the concept of different DML commands.

1. Exercise on retrieving records from a table.
   a. Find out the names of all the clients.

Commands:
SELECT NAME FROM CLIENT_MASTER;

Output:

```
mysql> SELECT NAME FROM CLIENT_MASTER;
+-----------------+
| NAME            |
+-----------------+
| Ivan Bayross    |
| Mamta Muzumdar  |
| Chhaya Bankar   |
| Ashwini Joshi   |
| Hansel Colaco   |
| Deepak Sharma   |
+-----------------+
6 rows in set (0.00 sec)
```

   b. Retrieve the entire contents of the Client_Master table.

Commands:
SELECT * FROM CLIENT_MASTER;

Output:

```
mysql> SELECT * FROM CLIENT_MASTER;
+----------+---------------+----------+----------+-----------+---------+-------------+----------+
| CLIENTNO | NAME          | ADDRESS1 | ADDRESS2 | CITY      | PINCODE | STATE       | BALDUE   |
+----------+---------------+----------+----------+-----------+---------+-------------+----------+
| C00001   | Ivan Bayross  | NULL     | NULL     | Mumbai    |  400054 | Maharashtra | 15000.00 |
| C00002   | Mamta Muzumdar| NULL     | NULL     | Madras    |  780001 | Tamil Nadu  |     0.00 |
| C00003   | Chhaya Bankar | NULL     | NULL     | Mumbai    |  400057 | Maharashtra |  5000.00 |
| C00004   | Ashwini Joshi | NULL     | NULL     | Bangalore |  560001 | Karnataka   |     0.00 |
| C00005   | Hansel Colaco | NULL     | NULL     | Mumbai    |  400060 | Maharashtra |  2000.00 |
| C00006   | Deepak Sharma | NULL     | NULL     | Mangalore |  560050 | Karnataka   |     0.00 |
+----------+---------------+----------+----------+-----------+---------+-------------+----------+
6 rows in set (0.01 sec)
```

   c. Retrieve the list of names,city and the state of all the clients.

Commands:
SELECT NAME, CITY, STATE FROM CLIENT_MASTER;

Output:

```
mysql> SELECT NAME, CITY, STATE FROM CLIENT_MASTER;
+----------------+-----------+-------------+
| NAME           | CITY      | STATE       |
+----------------+-----------+-------------+
| Ivan Bayross   | Mumbai    | Maharashtra |
| Mamta Muzumdar | Madras    | Tamil Nadu  |
| Chhaya Bankar  | Mumbai    | Maharashtra |
| Ashwini Joshi  | Bangalore | Karnataka   |
| Hansel Colaco  | Mumbai    | Maharashtra |
| Deepak Sharma  | Mangalore | Karnataka   |
+----------------+-----------+-------------+
6 rows in set (0.00 sec)
```

d. List the various products available from the Product_Master table.

Commands:
SELECT DESCRIPTION FROM PRODUCT_MASTER;

Output:

```
mysql> SELECT DESCRIPTION FROM PRODUCT_MASTER;
+--------------+
| DESCRIPTION  |
+--------------+
| T-Shirt      |
| Shirts       |
| Cotton jeans |
| Jeans        |
| Trousers     |
| Pull Overs   |
| Denim jeans  |
| Lycra tops   |
| Skirts       |
+--------------+
9 rows in set (0.00 sec)
```

e. List all the clients who are located in Mumbai.

Commands:
SELECT * FROM CLIENT_MASTER WHERE CITY = 'Mumbai';

Output:

```
mysql> SELECT * FROM CLIENT_MASTER WHERE CITY = 'Mumbai';
+----------+--------------+----------+----------+--------+---------+-------------+----------+
| CLIENTNO | NAME         | ADDRESS1 | ADDRESS2 | CITY   | PINCODE | STATE       | BALDUE   |
+----------+--------------+----------+----------+--------+---------+-------------+----------+
| C00001   | Ivan Bayross | NULL     | NULL     | Mumbai |  400054 | Maharashtra | 15000.00 |
| C00003   | Chhaya Bankar | NULL     | NULL     | Mumbai |  400057 | Maharashtra |  5000.00 |
| C00005   | Hansel Colaco | NULL     | NULL     | Mumbai |  400060 | Maharashtra |  2000.00 |
+----------+--------------+----------+----------+--------+---------+-------------+----------+
3 rows in set (0.00 sec)
```

f. Find the names of salesman who have a salary equal to Rs.3000.

Commands:
SELECT SALESMANNAME FROM SALESMAN_MASTER WHERE SALAMT = 3000;

Output:

```
mysql> SELECT SALESMANNAME FROM SALESMAN_MASTER WHERE SALAMT = 3000;
+--------------+
| SALESMANNAME |
+--------------+
| Aman         |
| Omkar        |
| Raj          |
+--------------+
3 rows in set (0.00 sec)
```

2. Exercise on updating records in a table
   a. Change the city of ClientNo ‘C00005’ to ‘Bangalore’.

Commands:
UPDATE CLIENT_MASTER SET CITY = 'Bangalore' WHERE CLIENTNO = 'C00005';

Output:

```
mysql> UPDATE CLIENT_MASTER SET CITY = 'Bangalore' WHERE CLIENTNO = 'C00005';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

b. Change the BalDue of ClientNo ‘C00001’ to Rs.1000.

Commands:
UPDATE CLIENT_MASTER SET BALDUE = 1000 WHERE CLIENTNO = 'C00001';

Output:

```
mysql> UPDATE CLIENT_MASTER SET CITY = 'Bangalore' WHERE CLIENTNO = 'C00005';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> UPDATE CLIENT_MASTER SET BALDUE = 1000 WHERE CLIENTNO = 'C00001';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from CLIENT_MASTER;
+----------+---------------+----------+----------+-----------+---------+-------------+---------+
| CLIENTNO | NAME          | ADDRESS1 | ADDRESS2 | CITY      | PINCODE | STATE       | BALDUE  |
+----------+---------------+----------+----------+-----------+---------+-------------+---------+
| C00001   | Ivan Bayross  | NULL     | NULL     | Mumbai    |  400054 | Maharashtra | 1000.00 |
| C00002   | Mamta Muzumdar| NULL     | NULL     | Madras    |  780001 | Tamil Nadu  |    0.00 |
| C00003   | Chhaya Bankar | NULL     | NULL     | Mumbai    |  400057 | Maharashtra | 5000.00 |
| C00004   | Ashwini Joshi | NULL     | NULL     | Bangalore |  560001 | Karnataka   |    0.00 |
| C00005   | Hansel Colaco | NULL     | NULL     | Bangalore |  400060 | Maharashtra | 2000.00 |
| C00006   | Deepak Sharma | NULL     | NULL     | Mangalore |  560050 | Karnataka   |    0.00 |
+----------+---------------+----------+----------+-----------+---------+-------------+---------+
6 rows in set (0.00 sec)
```

c. Change the cost price of ‘Trousers’ to rs.950.00.

Commands:

UPDATE PRODUCT_MASTER SET COSTPRICE = 950.00 WHERE DESCRIPTION = 'Trousers';

Output:

```
mysql> select * from PRODUCT_MASTER;
+-----------+--------------+---------------+--------------+-----------+------------+-----------+-----------+
| PRODUCTNO | DESCRIPTION  | PROFITPERCENT | UNIT_MEASURE | QTYONHAND | REORDERLVL | SELLPRICE | COSTPRICE |
+-----------+--------------+---------------+--------------+-----------+------------+-----------+-----------+
| P00001    | T-Shirt      |          5.00 | Piece        |       200 |         50 |    350.00 |    250.00 |
| P0345     | Shirts       |          6.00 | Piece        |       150 |         50 |    500.00 |    350.00 |
| P06734    | Cotton jeans |          5.00 | Piece        |       100 |         20 |    600.00 |    450.00 |
| P07865    | Jeans        |          5.00 | Piece        |       100 |         20 |    750.00 |    500.00 |
| P07868    | Trousers     |          2.00 | Piece        |       150 |         50 |    850.00 |    950.00 |
| P07885    | Pull Overs   |          2.50 | Piece        |        80 |         30 |    700.00 |    450.00 |
| P07965    | Denim jeans  |          4.00 | Piece        |       100 |         40 |    350.00 |    250.00 |
| P07975    | Lycra tops   |          5.00 | Piece        |        70 |         30 |    300.00 |    175.00 |
| P08865    | Skirts       |          5.00 | Piece        |        75 |         30 |    450.00 |    300.00 |
+-----------+--------------+---------------+--------------+-----------+------------+-----------+-----------+
9 rows in set (0.00 sec)
```

d. Change the city of the salesman to Pune.

Commands:

UPDATE SALESMAN_MASTER SET CITY = 'Pune';

Output:

```
mysql> select * from SALESMAN_MASTER;
+------------+-------------+----------+----------+------+---------+-------------+--------+----------+----------+---------+
| SALESMANNO | SALESMANNAME | ADDRESS1 | ADDRESS2 | CITY | PINCODE | STATE       | SALAMT | TGTTOGET | YTDSALES | REMARKS |
+------------+-------------+----------+----------+------+---------+-------------+--------+----------+----------+---------+
| S00001     | Aman        | A/14     | Worli    | Pune |  400002 | Maharashtra |   3000 |      100 |    50.00 | Good    |
| S00002     | Omkar       | 65       | Nariman  | Pune |  400001 | Maharashtra |   3000 |      200 |   100.00 | Good    |
| S00003     | Raj         | P-7      | Bandra   | Pune |  400032 | Maharashtra |   3000 |      200 |   100.00 | Good    |
| S00004     | Ashish      | A/5      | Juhu     | Pune |  400044 | Maharashtra |   3500 |      200 |   150.00 | Good    |
+------------+-------------+----------+----------+------+---------+-------------+--------+----------+----------+---------+
4 rows in set (0.00 sec)
```

3. Exercise on deleting records in a table

a. Delete all salesman from the Salesman_Master whose salaries are equal to Rs.3500.

Commands:

DELETE FROM SALESMAN_MASTER WHERE SALAMT = 3500;

Output:

```
mysql> select * from SALESMAN_MASTER;
+------------+-------------+----------+----------+------+---------+-------------+--------+----------+----------+---------+
| SALESMANNO | SALESMANNAME | ADDRESS1 | ADDRESS2 | CITY | PINCODE | STATE       | SALAMT | TGTTOGET | YTDSALES | REMARKS |
+------------+-------------+----------+----------+------+---------+-------------+--------+----------+----------+---------+
| S00001     | Aman         | A/14     | Worli    | Pune |  400002 | Maharashtra |   3000 |      100 |    50.00 | Good    |
| S00002     | Omkar        | 65       | Nariman  | Pune |  400001 | Maharashtra |   3000 |      200 |   100.00 | Good    |
| S00003     | Raj          | P-7      | Bandra   | Pune |  400032 | Maharashtra |   3000 |      200 |   100.00 | Good    |
+------------+-------------+----------+----------+------+---------+-------------+--------+----------+----------+---------+
3 rows in set (0.00 sec)
```

b. Delete all products from Product_Master where the quantity on hand is equal to 100.

Commands:
DELETE FROM PRODUCT_MASTER WHERE QTYONHAND = 100;

Output:

```
mysql> select * from PRODUCT_MASTER;
+-----------+-------------+--------------+--------------+-----------+------------+-----------+-----------+
| PRODUCTNO | DESCRIPTION | PROFITPERCENT | UNIT_MEASURE | QTYONHAND | REORDERLVL | SELLPRICE | COSTPRICE |
+-----------+-------------+--------------+--------------+-----------+------------+-----------+-----------+
| P00001    | T-Shirt     |         5.00 | Piece        |       200 |         50 |    350.00 |    250.00 |
| P0345     | Shirts      |         6.00 | Piece        |       150 |         50 |    500.00 |    350.00 |
| P07868    | Trousers    |         2.00 | Piece        |       150 |         50 |    850.00 |    950.00 |
| P07885    | Pull Overs  |         2.50 | Piece        |        80 |         30 |    700.00 |    450.00 |
| P07975    | Lycra tops  |         5.00 | Piece        |        70 |         30 |    300.00 |    175.00 |
| P08865    | Skirts      |         5.00 | Piece        |        75 |         30 |    450.00 |    300.00 |
+-----------+-------------+--------------+--------------+-----------+------------+-----------+-----------+
6 rows in set (0.00 sec)
```

c. Delete from Client_Master where the column state holds the value ‘Tamil Nadu’.

Commands:
DELETE FROM CLIENT_MASTER WHERE STATE = 'Tamil Nadu';

Output:

```
mysql> select * from CLIENT_MASTER;
+----------+---------------+----------+----------+-----------+---------+-------------+---------+
| CLIENTNO | NAME          | ADDRESS1 | ADDRESS2 | CITY      | PINCODE | STATE       | BALDUE  |
+----------+---------------+----------+----------+-----------+---------+-------------+---------+
| C00001   | Ivan Bayross  | NULL     | NULL     | Mumbai    |  400054 | Maharashtra | 1000.00 |
| C00003   | Chhaya Bankar | NULL     | NULL     | Mumbai    |  400057 | Maharashtra | 5000.00 |
| C00004   | Ashwini Joshi | NULL     | NULL     | Bangalore |  560001 | Karnataka   |    0.00 |
| C00005   | Hansel Colaco | NULL     | NULL     | Bangalore |  400060 | Maharashtra | 2000.00 |
| C00006   | Deepak Sharma | NULL     | NULL     | Mangalore |  560050 | Karnataka   |    0.00 |
+----------+---------------+----------+----------+-----------+---------+-------------+---------+
5 rows in set (0.00 sec)
```

4. Exercise on altering the table structure
   a. Add a column called ‘Telephone’ of data type integer to the Client_Master table.

Commands:
ALTER TABLE CLIENT_MASTER ADD Telephone INTEGER;

Output:

```
mysql> select * from CLIENT_MASTER;
+----------+---------------+----------+----------+-----------+---------+-------------+---------+-----------+
| CLIENTNO | NAME          | ADDRESS1 | ADDRESS2 | CITY      | PINCODE | STATE       | BALDUE  | Telephone |
+----------+---------------+----------+----------+-----------+---------+-------------+---------+-----------+
| C00001   | Ivan Bayross  | NULL     | NULL     | Mumbai    |  400054 | Maharashtra | 1000.00 |      NULL |
| C00003   | Chhaya Bankar | NULL     | NULL     | Mumbai    |  400057 | Maharashtra | 5000.00 |      NULL |
| C00004   | Ashwini Joshi | NULL     | NULL     | Bangalore |  560001 | Karnataka   |    0.00 |      NULL |
| C00005   | Hansel Colaco | NULL     | NULL     | Bangalore |  400060 | Maharashtra | 2000.00 |      NULL |
| C00006   | Deepak Sharma | NULL     | NULL     | Mangalore |  560050 | Karnataka   |    0.00 |      NULL |
+----------+---------------+----------+----------+-----------+---------+-------------+---------+-----------+
5 rows in set (0.00 sec)
```

b. Change the size off SellPrice column in Product _Master to 10, 2.

Commands:
ALTER TABLE PRODUCT_MASTER MODIFY SellPrice DECIMAL(10, 2);

Output:

```
mysql> select * from PRODUCT_MASTER;
+-----------+-------------+--------------+--------------+-----------+------------+-----------+-----------+
| PRODUCTNO | DESCRIPTION | PROFITPERCENT | UNIT_MEASURE | QTYONHAND | REORDERLVL | SellPrice | COSTPRICE |
+-----------+-------------+--------------+--------------+-----------+------------+-----------+-----------+
| P00001    | T-Shirt     |         5.00 | Piece        |       200 |         50 |    350.00 |    250.00 |
| P0345     | Shirts      |         6.00 | Piece        |       150 |         50 |    500.00 |    350.00 |
| P07868    | Trousers    |         2.00 | Piece        |       150 |         50 |    850.00 |    950.00 |
| P07885    | Pull Overs  |         2.50 | Piece        |        80 |         30 |    700.00 |    450.00 |
| P07975    | Lycra tops  |         5.00 | Piece        |        70 |         30 |    300.00 |    175.00 |
| P08865    | Skirts      |         5.00 | Piece        |        75 |         30 |    450.00 |    300.00 |
+-----------+-------------+--------------+--------------+-----------+------------+-----------+-----------+
6 rows in set (0.00 sec)
```

5. Exercise on deleting the table structure along with the data
   a. Destroy the table Client_Master along with its data.

Commands:
DROP TABLE CLIENT_MASTER;

Output:

```
mysql> DROP TABLE CLIENT_MASTER;
Query OK, 0 rows affected (0.01 sec)

mysql> show tables;
+------------------+
| Tables_in_exp1   |
+------------------+
| PRODUCT_MASTER   |
| SALESMAN_MASTER  |
+------------------+
2 rows in set (0.00 sec)
```

6. Exercise on renaming the table
   a. Change the name of the Salesman_Master to sman_mast.

Commands:
ALTER TABLE SALESMAN_MASTER RENAME TO sman_mast;

Output:

```
mysql> show tables;
+-----------------+
| Tables_in_exp1  |
+-----------------+
| PRODUCT_MASTER  |
| sman_mast       |
+-----------------+
2 rows in set (0.01 sec)
```

EXPERIMENT-3

TITLE: DDL (Data Definition Language) commands with Data Constraints

Objective: To understand the concept of data constraints that is enforced on data being stored in the table. Focus on Primary Key and the Foreign Key

Create the tables described below:

Table name: CLIENT_MASTER_1
Description: used to store client information.

Column name	data type	Size	Constraints
CLIENTNO	Varchar	6	Primary key / first letter must start with 'C'
NAME	Varchar	20	Not Null
ADDRESS 1	Varchar	30	
ADDRESS 2	Varchar	30	
CITY	Varchar	15	
PINCODE	Integer	8	
STATE	Varchar	15	
BALDUE	Decimal	10,2	

Commands:

```
CREATE TABLE CLIENT_MASTER_1 (
CLIENTNO VARCHAR(6) PRIMARY KEY CHECK (CLIENTNO LIKE 'C%'),
NAME VARCHAR(20) NOT NULL,
ADDRESS1 VARCHAR(30),
ADDRESS2 VARCHAR(30),
CITY VARCHAR(15),
PINCODE INTEGER,
STATE VARCHAR(15),
BALDUE DECIMAL(10, 2)
);
```

Output:

```
mysql> CREATE TABLE CLIENT_MASTER_1 (
    ->     CLIENTNO VARCHAR(6) PRIMARY KEY CHECK (CLIENTNO LIKE 'C%'),
    ->     NAME VARCHAR(20) NOT NULL,
    ->     ADDRESS1 VARCHAR(30),
    ->     ADDRESS2 VARCHAR(30),
    ->     CITY VARCHAR(15),
    ->     PINCODE INTEGER,
    ->     STATE VARCHAR(15),
    ->     BALDUE DECIMAL(10, 2)
    -> );
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> show tables;
+-----------------+
| Tables_in_exp1  |
+-----------------+
| CLIENT_MASTER_1 |
| PRODUCT_MASTER  |
| sman_mast       |
+-----------------+
3 rows in set (0.01 sec)
```

Table Name: **PRODUCT_MASTER_1**
Description: used to store product information

Column name	data type	Size	Attributes
PRODUCTNO	Varchar	6	Primary Key/ first letter must start with ‘P’
DESCRIPTION	Varchar	15	Not Null
PROFITPERCENT	Decimal	4,2	Not Null
UNIT MEASURE	Varchar	10	Not Null
QTYONHAND	Integer	8	Not Null
REORDERL VL	Integer	8	Not Null
SELLPRICE	Decimal	8,2	Not Null
COSTPRICE	Decimal	8,2	Not Null

Commands:

```
CREATE TABLE PRODUCT_MASTER_1 (
PRODUCTNO VARCHAR(6) PRIMARY KEY CHECK (PRODUCTNO LIKE 'P%'),
DESCRIPTION VARCHAR(15) NOT NULL,
PROFITPERCENT DECIMAL(4, 2) NOT NULL,
UNIT_MEASURE VARCHAR(10) NOT NULL,
QTYONHAND INTEGER NOT NULL,
REORDERLVL INTEGER NOT NULL,
SELLPRICE DECIMAL(8, 2) NOT NULL,
COSTPRICE DECIMAL(8, 2) NOT NULL
);
```

Output:

```
mysql> CREATE TABLE PRODUCT_MASTER_1 (
    ->     PRODUCTNO VARCHAR(6) PRIMARY KEY CHECK (PRODUCTNO LIKE 'P%'),
    ->     DESCRIPTION VARCHAR(15) NOT NULL,
    ->     PROFITPERCENT DECIMAL(4, 2) NOT NULL,
    ->     UNIT_MEASURE VARCHAR(10) NOT NULL,
    ->     QTYONHAND INTEGER NOT NULL,
    ->     REORDERLVL INTEGER NOT NULL,
    ->     SELLPRICE DECIMAL(8, 2) NOT NULL,
    ->     COSTPRICE DECIMAL(8, 2) NOT NULL
    -> );
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> show tables;
+------------------+
| Tables_in_exp1   |
+------------------+
| CLIENT_MASTER_1  |
| PRODUCT_MASTER   |
| PRODUCT_MASTER_1 |
| sman_mast        |
+------------------+
4 rows in set (0.00 sec)
```

Table Name: **SALESMAN_MASTER _1**
Description: used to store salesman information working for the company.

Column name	data type	Size	Attributes
SALESMANNO	Varchar	6	Primary Key/ first letter must start with ‘S’
SALESMANNAME	Varchar	20	Not Null
ADDRESS 1	Varchar	30	Not Null
ADDRESS 2	Varchar	30	
CITY	Varchar	20	
PINCODE	Integer	8	
STATE	Varchar	20	
SALAMT	Real	8,2	Not Null , Cannot be 0
TGTTOGET	Decimal	6,2	Not Null , Cannot be 0
YTDSALES	Double	6,2	Not Null
REMARKS	Varchar	60	

Commands:

```
CREATE TABLE SALESMAN_MASTER_1 (
SALESMANNO VARCHAR(6) PRIMARY KEY CHECK (SALESMANNO LIKE 'S%'),
SALESMANNAME VARCHAR(20) NOT NULL,
ADDRESS1 VARCHAR(30) NOT NULL,
ADDRESS2 VARCHAR(30),
CITY VARCHAR(20),
PINCODE INTEGER,
```

STATE VARCHAR(20),
SALAMT REAL NOT NULL CHECK (SALAMT <> 0),
TGTTOGET DECIMAL(6, 2) NOT NULL CHECK (TGTTOGET <> 0),
YTDSALES DOUBLE(6, 2) NOT NULL,
REMARKS VARCHAR(60)
);

Output:

```
mysql> CREATE TABLE SALESMAN_MASTER_1 (
    ->     SALESMANNO VARCHAR(6) PRIMARY KEY CHECK (SALESMANNO LIKE 'S%'),
    ->     SALESMANNAME VARCHAR(20) NOT NULL,
    ->     ADDRESS1 VARCHAR(30) NOT NULL,
    ->     ADDRESS2 VARCHAR(30),
    ->     CITY VARCHAR(20),
    ->     PINCODE INTEGER,
    ->     STATE VARCHAR(20),
    ->     SALAMT REAL NOT NULL CHECK (SALAMT <> 0),
    ->     TGTTOGET DECIMAL(6, 2) NOT NULL CHECK (TGTTOGET <> 0),
    ->     YTDSALES DOUBLE(6, 2) NOT NULL,
    ->     REMARKS VARCHAR(60)
    -> );
Query OK, 0 rows affected, 1 warning (0.02 sec)
```

```
mysql> show tables;
+--------------------+
| Tables_in_exp1     |
+--------------------+
| CLIENT_MASTER_1    |
| PRODUCT_MASTER     |
| PRODUCT_MASTER_1   |
| SALESMAN_MASTER_1  |
| sman_mast          |
+--------------------+
5 rows in set (0.00 sec)
```

1. **Reinsert the data in these two tables based upon Lab 2.**

Commands:
INSERT INTO CLIENT_MASTER_1 (CLIENTNO, NAME, ADDRESS1, ADDRESS2, CITY, PINCODE, STATE, BALDUE) VALUES
('C00001', 'Ivan Bayross', 'Mumbai', NULL, NULL, 400054, 'Maharashtra', 15000),
('C00002', 'Mamta Muzumdar', 'Madras', NULL, NULL, 780001, 'Tamil Nadu', 0),
('C00003', 'Chhaya Bankar', 'Mumbai', NULL, NULL, 400057, 'Maharashtra', 5000),
('C00004', 'Ashwini Joshi', 'Bangalore', NULL, NULL, 560001, 'Karnataka', 0),
('C00005', 'Hansel Colaco', 'Mumbai', NULL, NULL, 400060, 'Maharashtra', 2000),
('C00006', 'Deepak Sharma', 'Mangalore', NULL, NULL, 560050, 'Karnataka', 0);

Output:

```
mysql> INSERT INTO CLIENT_MASTER_1 (CLIENTNO, NAME, ADDRESS1, ADDRESS2, CITY, PINCODE, STATE, BALDUE) VALUES
    -> ('C00001', 'Ivan Bayross', 'Mumbai', NULL, NULL, 400054, 'Maharashtra', 15000),
    -> ('C00002', 'Mamta Muzumdar', 'Madras', NULL, NULL, 780001, 'Tamil Nadu', 0),
    -> ('C00003', 'Chhaya Bankar', 'Mumbai', NULL, NULL, 400057, 'Maharashtra', 5000),
    -> ('C00004', 'Ashwini Joshi', 'Bangalore', NULL, NULL, 560001, 'Karnataka', 0),
    -> ('C00005', 'Hansel Colaco', 'Mumbai', NULL, NULL, 400060, 'Maharashtra', 2000),
    -> ('C00006', 'Deepak Sharma', 'Mangalore', NULL, NULL, 560050, 'Karnataka', 0);
Query OK, 6 rows affected (0.01 sec)
Records: 6  Duplicates: 0  Warnings: 0
```

```
mysql> select * from CLIENT_MASTER_1;
+----------+----------------+-----------+----------+------+---------+-------------+----------+
| CLIENTNO | NAME           | ADDRESS1  | ADDRESS2 | CITY | PINCODE | STATE       | BALDUE   |
+----------+----------------+-----------+----------+------+---------+-------------+----------+
| C00001   | Ivan Bayross   | Mumbai    | NULL     | NULL |  400054 | Maharashtra | 15000.00 |
| C00002   | Mamta Muzumdar | Madras    | NULL     | NULL |  780001 | Tamil Nadu  |     0.00 |
| C00003   | Chhaya Bankar  | Mumbai    | NULL     | NULL |  400057 | Maharashtra |  5000.00 |
| C00004   | Ashwini Joshi  | Bangalore | NULL     | NULL |  560001 | Karnataka   |     0.00 |
| C00005   | Hansel Colaco  | Mumbai    | NULL     | NULL |  400060 | Maharashtra |  2000.00 |
| C00006   | Deepak Sharma  | Mangalore | NULL     | NULL |  560050 | Karnataka   |     0.00 |
+----------+----------------+-----------+----------+------+---------+-------------+----------+
6 rows in set (0.00 sec)
```

2. **Display the contents of each table.**

Commands:

INSERT INTO PRODUCT_MASTER_1 (PRODUCTNO, DESCRIPTION, PROFITPERCENT, UNIT_MEASURE, QTYONHAND, REORDERLVL, SELLPRICE, COSTPRICE) VALUES
('P00001', 'T-Shirt', 5, 'Piece', 200, 50, 350, 250),
('P0345', 'Shirts', 6, 'Piece', 150, 50, 500, 350),
('P06734', 'Cotton jeans', 5, 'Piece', 100, 20, 600, 450),
('P07865', 'Jeans', 5, 'Piece', 100, 20, 750, 500),
('P07868', 'Trousers', 2, 'Piece', 150, 50, 850, 550),
('P07885', 'Pull Overs', 2.5, 'Piece', 80, 30, 700, 450),
('P07965', 'Denim jeans', 4, 'Piece', 100, 40, 350, 250),
('P07975', 'Lycra tops', 5, 'Piece', 70, 30, 300, 175),
('P08865', 'Skirts', 5, 'Piece', 75, 30, 450, 300);

Output:

```
mysql> INSERT INTO PRODUCT_MASTER_1 (PRODUCTNO, DESCRIPTION, PROFITPERCENT, UNIT_MEASURE, QTYONHAND, REORDERLVL, SELLPRICE, COSTPRICE) VALUES
    -> ('P00001', 'T-Shirt', 5, 'Piece', 200, 50, 350, 250),
    -> ('P0345', 'Shirts', 6, 'Piece', 150, 50, 500, 350),
    -> ('P06734', 'Cotton jeans', 5, 'Piece', 100, 20, 600, 450),
    -> ('P07865', 'Jeans', 5, 'Piece', 100, 20, 750, 500),
    -> ('P07868', 'Trousers', 2, 'Piece', 150, 50, 850, 550),
    -> ('P07885', 'Pull Overs', 2.5, 'Piece', 80, 30, 700, 450),
    -> ('P07965', 'Denim jeans', 4, 'Piece', 100, 40, 350, 250),
    -> ('P07975', 'Lycra tops', 5, 'Piece', 70, 30, 300, 175),
    -> ('P08865', 'Skirts', 5, 'Piece', 75, 30, 450, 300);
Query OK, 9 rows affected (0.01 sec)
Records: 9  Duplicates: 0  Warnings: 0
```

```
mysql> select* from PRODUCT_MASTER_1;
+-----------+--------------+---------------+--------------+-----------+------------+-----------+-----------+
| PRODUCTNO | DESCRIPTION  | PROFITPERCENT | UNIT_MEASURE | QTYONHAND | REORDERLVL | SELLPRICE | COSTPRICE |
+-----------+--------------+---------------+--------------+-----------+------------+-----------+-----------+
| P00001    | T-Shirt      |          5.00 | Piece        |       200 |         50 |    350.00 |    250.00 |
| P0345     | Shirts       |          6.00 | Piece        |       150 |         50 |    500.00 |    350.00 |
| P06734    | Cotton jeans |          5.00 | Piece        |       100 |         20 |    600.00 |    450.00 |
| P07865    | Jeans        |          5.00 | Piece        |       100 |         20 |    750.00 |    500.00 |
| P07868    | Trousers     |          2.00 | Piece        |       150 |         50 |    850.00 |    550.00 |
| P07885    | Pull Overs   |          2.50 | Piece        |        80 |         30 |    700.00 |    450.00 |
| P07965    | Denim jeans  |          4.00 | Piece        |       100 |         40 |    350.00 |    250.00 |
| P07975    | Lycra tops   |          5.00 | Piece        |        70 |         30 |    300.00 |    175.00 |
| P08865    | Skirts       |          5.00 | Piece        |        75 |         30 |    450.00 |    300.00 |
+-----------+--------------+---------------+--------------+-----------+------------+-----------+-----------+
9 rows in set (0.00 sec)
```

Commands:

INSERT INTO SALESMAN_MASTER_1 (SALESMANNO, SALESMANNAME, ADDRESS1, ADDRESS2, CITY, PINCODE, STATE, SALAMT, TGTTOGET, YTDSALES, REMARKS) VALUES
('S00001', 'Aman', 'A/14', 'Worli', 'Mumbai', 400002, 'Maharashtra', 3000, 100, 50, 'Good'),
('S00002', 'Omkar', '65', 'Nariman', 'Mumbai', 400001, 'Maharashtra', 3000, 200, 100, ' Good'),
('S00003', 'Raj', 'P-7', 'Bandra', 'Mumbai', 400032, 'Maharashtra', 3000, 200, 100, 'Good'),
('S00004', 'Ashish', 'A/5', 'Juhu', 'Mumbai', 400044, 'Maharashtra', 3500, 200, 150, 'Good');

Output:

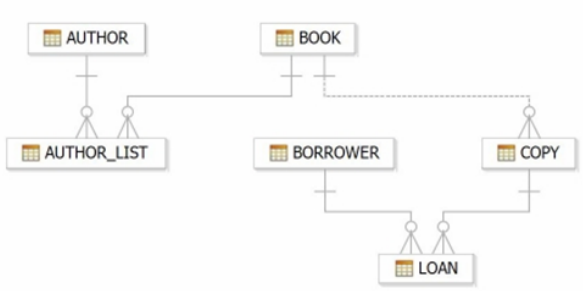
```
mysql> INSERT INTO SALESMAN_MASTER_1 (SALESMANNO, SALESMANNAME, ADDRESS1, ADDRESS2, CITY, PINCODE, STATE, SALAMT, TGTTOGET, YTDSALES, REMARKS) VAL
UES
    -> ('S00001', 'Aman', 'A/14', 'Worli', 'Mumbai', 400002, 'Maharashtra', 3000, 100, 50, 'Good'),
    -> ('S00002', 'Omkar', '65', 'Nariman', 'Mumbai', 400001, 'Maharashtra', 3000, 200, 100, 'Good'),
    -> ('S00003', 'Raj', 'P-7', 'Bandra', 'Mumbai', 400032, 'Maharashtra', 3000, 200, 100, 'Good'),
    -> ('S00004', 'Ashish', 'A/5', 'Juhu', 'Mumbai', 400044, 'Maharashtra', 3500, 200, 150, 'Good');
Query OK, 4 rows affected (0.02 sec)
Records: 4  Duplicates: 0  Warnings: 0
```

```
mysql> select * from SALESMAN_MASTER_1;
+------------+--------------+----------+----------+--------+---------+-------------+--------+----------+----------+---------+
| SALESMANNO | SALESMANNAME | ADDRESS1 | ADDRESS2 | CITY   | PINCODE | STATE       | SALAMT | TGTTOGET | YTDSALES | REMARKS |
+------------+--------------+----------+----------+--------+---------+-------------+--------+----------+----------+---------+
| S00001     | Aman         | A/14     | Worli    | Mumbai |  400002 | Maharashtra |   3000 |   100.00 |    50.00 | Good    |
| S00002     | Omkar        | 65       | Nariman  | Mumbai |  400001 | Maharashtra |   3000 |   200.00 |   100.00 | Good    |
| S00003     | Raj          | P-7      | Bandra   | Mumbai |  400032 | Maharashtra |   3000 |   200.00 |   100.00 | Good    |
| S00004     | Ashish       | A/5      | Juhu     | Mumbai |  400044 | Maharashtra |   3500 |   200.00 |   150.00 | Good    |
+------------+--------------+----------+----------+--------+---------+-------------+--------+----------+----------+---------+
4 rows in set (0.00 sec)
```

EXPERIMENT-4

TITLE: DDL (Data Definition Language) commands with Data Constraints

Objective: To understand the concept of data constraints that is enforced on data being stored in the table. Focus on Primary Key, The Foreign Key and constraints.
Review this diagram



1. Create table AUTHOR = {Author_ID , Lastname, Firstname, Email, City, Country}
Where:
Author_ID – text data type, 5 characters, primary key
Lastname – text data type, 15 characters, not null
Firstname – text data type, 15 characters, not null
Email – text data type, 40 characters,
City – text data type, 15 characters,
Country – text data type, 15 characters,

Commands:

```
CREATE TABLE AUTHOR (
   Author_ID VARCHAR(5) PRIMARY KEY CHECK (Author_ID LIKE 'A%'),
   Lastname VARCHAR(15) NOT NULL,
   Firstname VARCHAR(15) NOT NULL,
   Email VARCHAR(40),
   City VARCHAR(15),
   Country VARCHAR(15)
);
```

Output:

```
mysql> CREATE TABLE AUTHOR (
    ->     Author_ID VARCHAR(5) PRIMARY KEY CHECK (Author_ID LIKE 'A%'),
    ->     Lastname VARCHAR(15) NOT NULL,
    ->     Firstname VARCHAR(15) NOT NULL,
    ->     Email VARCHAR(40),
    ->     City VARCHAR(15),
    ->     Country VARCHAR(15)
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql> show tables;
+-------------------+
| Tables_in_exp1    |
+-------------------+
| AUTHOR            |
| CLIENT_MASTER_1   |
| PRODUCT_MASTER    |
| PRODUCT_MASTER_1  |
| SALESMAN_MASTER_1 |
| sman_mast         |
+-------------------+
6 rows in set (0.00 sec)
```

2. Create Table BOOK={ Book_ID, Book_Title, Copies)
Where :
Book_ID – text data type, 5 characters Primary Key Start With Character **B**
Book_Title - Text data Type Not Null
Copies- No.of copies Data Type int always greater the 2

Commands:

```
CREATE TABLE BOOK (
Book_ID VARCHAR(5) PRIMARY KEY CHECK (Book_ID LIKE 'B%'),
Book_Title TEXT NOT NULL,
Copies INT CHECK (Copies > 2)
);
```

Output:

```
mysql> CREATE TABLE BOOK (
    ->     Book_ID VARCHAR(5) PRIMARY KEY CHECK (Book_ID LIKE 'B%'),
    ->     Book_Title TEXT NOT NULL,
    ->     Copies INT CHECK (Copies > 2)
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql> show tables;
+--------------------+
| Tables_in_exp1     |
+--------------------+
| AUTHOR             |
| BOOK               |
| CLIENT_MASTER_1    |
| PRODUCT_MASTER     |
| PRODUCT_MASTER_1   |
| SALESMAN_MASTER_1  |
| sman_mast          |
+--------------------+
7 rows in set (0.01 sec)
```

3. Create table AUTHOR_LIST = {Author_ID , Book_ID , Role}
Where:
Author_ID – text data type, 5 characters, referenced by Author_ID from AUTHOR table
Book_ID – text data type, 5 characters
Role – text data type, 15 characters
and primary key is: Author_ID, Book_ID

Commands:

```
CREATE TABLE AUTHOR_LIST (
Author_ID VARCHAR(5),
Book_ID VARCHAR(5),
Role VARCHAR(15),
Publisher VARCHAR(30),
PRIMARY KEY (Author_ID, Book_ID),
```

FOREIGN KEY (Author_ID) REFERENCES AUTHOR(Author_ID),
FOREIGN KEY (Book_ID) REFERENCES BOOK(Book_ID)
);

Output:

```
mysql> CREATE TABLE AUTHOR_LIST (
    ->     Author_ID VARCHAR(5),
    ->     Book_ID VARCHAR(5),
    ->     Role VARCHAR(15),
    ->     Publisher VARCHAR(30),
    ->     PRIMARY KEY (Author_ID, Book_ID),
    ->     FOREIGN KEY (Author_ID) REFERENCES AUTHOR(Author_ID),
    ->     FOREIGN KEY (Book_ID) REFERENCES BOOK(Book_ID)
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> show tables;
+-------------------+
| Tables_in_exp1    |
+-------------------+
| AUTHOR            |
| AUTHOR_LIST       |
| BOOK              |
| CLIENT_MASTER_1   |
| PRODUCT_MASTER    |
| PRODUCT_MASTER_1  |
| SALESMAN_MASTER_1 |
| sman_mast         |
+-------------------+
8 rows in set (0.00 sec)
```

4. Add four records in each tables AUTHOR, BOOK, BOOK_LIST.

Commands:

1.

INSERT INTO AUTHOR (Author_ID, Lastname, Firstname, Email, City, Country) VALUES
('A0001', 'Smith', 'John', 'john@example.com', 'New York', 'USA'),
('A0002', 'Doe', 'Jane', 'jane@example.com', 'Los Angeles', 'USA'),
('A0003', 'Brown', 'Michael', 'michael@example.com', 'London', 'UK'),
('A0004', 'Johnson', 'Emily', 'emily@example.com', 'Sydney', 'Australia');

2.

INSERT INTO BOOK (Book_ID, Book_Title, Copies) VALUES
('B0001', 'Introduction to SQL', 10),
('B0002', 'Python Programming', 15),
('B0003', 'Data Structures and Algorithms', 20),
('B0004', 'Machine Learning Basics', 12);

3.

INSERT INTO AUTHOR_LIST (Author_ID, Book_ID, Role, Publisher) VALUES

('A0001', 'B0001', 'Author', 'ABC Publications'),
('A0002', 'B0001', 'Co-Author', 'ABC Publications'),
('A0003', 'B0002', 'Author', 'XYZ Publications'),
('A0004', 'B0003', 'Author', 'DEF Publications');

Output:

```
mysql> INSERT INTO AUTHOR (Author_ID, Lastname, Firstname, Email, City, Country) VALUES
    -> ('A0001', 'Smith', 'John', 'john@example.com', 'New York', 'USA'),
    -> ('A0002', 'Doe', 'Jane', 'jane@example.com', 'Los Angeles', 'USA'),
    -> ('A0003', 'Brown', 'Michael', 'michael@example.com', 'London', 'UK'),
    -> ('A0004', 'Johnson', 'Emily', 'emily@example.com', 'Sydney', 'Australia');
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> select * from AUTHOR;
+-----------+----------+-----------+--------------------+-------------+-----------+
| Author_ID | Lastname | Firstname | Email              | City        | Country   |
+-----------+----------+-----------+--------------------+-------------+-----------+
| A0001     | Smith    | John      | john@example.com   | New York    | USA       |
| A0002     | Doe      | Jane      | jane@example.com   | Los Angeles | USA       |
| A0003     | Brown    | Michael   | michael@example.com | London     | UK        |
| A0004     | Johnson  | Emily     | emily@example.com  | Sydney      | Australia |
+-----------+----------+-----------+--------------------+-------------+-----------+
4 rows in set (0.00 sec)
```

```
mysql> INSERT INTO BOOK (Book_ID, Book_Title, Copies) VALUES
    -> ('B0001', 'Introduction to SQL', 10),
    -> ('B0002', 'Python Programming', 15),
    -> ('B0003', 'Data Structures and Algorithms', 20),
    -> ('B0004', 'Machine Learning Basics', 12);
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0
```

```
mysql> select * from BOOK;
+---------+---------------------------------+--------+
| Book_ID | Book_Title                      | Copies |
+---------+---------------------------------+--------+
| B0001   | Introduction to SQL             |     10 |
| B0002   | Python Programming              |     15 |
| B0003   | Data Structures and Algorithms  |     20 |
| B0004   | Machine Learning Basics         |     12 |
+---------+---------------------------------+--------+
4 rows in set (0.00 sec)
```

```
mysql> INSERT INTO AUTHOR_LIST (Author_ID, Book_ID, Role, Publisher) VALUES
    -> ('A0001', 'B0001', 'Author', 'ABC Publications'),
    -> ('A0002', 'B0001', 'Co-Author', 'ABC Publications'),
    -> ('A0003', 'B0002', 'Author', 'XYZ Publications'),
    -> ('A0004', 'B0003', 'Author', 'DEF Publications');
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0
```

```
mysql> select * from AUTHOR_LIST;
+-----------+---------+-----------+------------------+
| Author_ID | Book_ID | Role      | Publisher        |
+-----------+---------+-----------+------------------+
| A0001     | B0001   | Author    | ABC Publications |
| A0002     | B0001   | Co-Author | ABC Publications |
| A0003     | B0002   | Author    | XYZ Publications |
| A0004     | B0003   | Author    | DEF Publications |
+-----------+---------+-----------+------------------+
4 rows in set (0.00 sec)
```

5. Alter structure of table AUTHOR_LIST add the field Publisher data type of 30 Character.

Commands:
ALTER TABLE AUTHOR_LIST ADD Publisher VARCHAR(30);

Output:

```
mysql> DESCRIBE AUTHOR_LIST;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| Author_ID | varchar(5)  | NO   | PRI | NULL    |       |
| Book_ID   | varchar(5)  | NO   | PRI | NULL    |       |
| Role      | varchar(15) | YES  |     | NULL    |       |
| Publisher | varchar(30) | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
4 rows in set (0.01 sec)
```

EXPERIMENT- 5,6

Title: Use of Inbuilt functions and relational algebra operation

Objective: To understand the use of inbuilt function and relational algebra with sql query.

1. Consider the following table structure and attempt.
Supplier-(scode,sname,scity,turnover)
Part-(pcode,weigh,color,cost,sellingprice)
Supplier_Part-(scode,pcode,qty)

a) Create tables

Commands:

a.
```
CREATE TABLE Supplier (
    scode INT PRIMARY KEY,
    sname VARCHAR(50),
    scity VARCHAR(50),
    turnover DECIMAL(10, 2)
);
```

b.
```
CREATE TABLE Part (
    pcode INT PRIMARY KEY,
    weigh DECIMAL(10, 2),
    color VARCHAR(20),
    cost DECIMAL(10, 2),
    sellingprice DECIMAL(10, 2)
);
```

c.
```
CREATE TABLE Supplier_Part (
    scode INT,
    pcode INT,
    qty INT,
    FOREIGN KEY (scode) REFERENCES Supplier(scode),
    FOREIGN KEY (pcode) REFERENCES Part(pcode),
    PRIMARY KEY (scode, pcode)
);
```

Output:

```
mysql> CREATE TABLE Supplier (
    ->     scode INT PRIMARY KEY,
    ->     sname VARCHAR(50),
    ->     scity VARCHAR(50),
    ->     turnover DECIMAL(10, 2)
    -> );
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE TABLE Part (
    ->     pcode INT PRIMARY KEY,
    ->     weigh DECIMAL(10, 2),
    ->     color VARCHAR(20),
    ->     cost DECIMAL(10, 2),
    ->     sellingprice DECIMAL(10, 2)
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE Supplier_Part (
    ->     scode INT,
    ->     pcode INT,
    ->     qty INT,
    ->     FOREIGN KEY (scode) REFERENCES Supplier(scode),
    ->     FOREIGN KEY (pcode) REFERENCES Part(pcode),
    ->     PRIMARY KEY (scode, pcode)
    -> );
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> show tables;
+------------------+
| Tables_in_exp1   |
+------------------+
| AUTHOR           |
| AUTHOR_LIST      |
| BOOK             |
| CLIENT_MASTER_1  |
| PRODUCT_MASTER   |
| PRODUCT_MASTER_1 |
| Part             |
| SALESMAN_MASTER_1|
| Supplier         |
| Supplier_Part    |
| sman_mast        |
+------------------+
11 rows in set (0.00 sec)
```

b) Populate the table.

Commands:

a.

```
INSERT INTO Supplier (scode, sname, scity, turnover) VALUES
(1, 'ABC Corporation', 'Bombay', 100),
(2, 'XYZ Enterprises', 'Delhi', 150),
(3, 'PQR Industries', 'Mumbai', 200),
(4, 'LMN Limited', 'Bangalore', 120),
(5, 'DEF Corporation', 'Chennai', 80);
```

b.

```
INSERT INTO Part (pcode, weigh, color, cost, sellingprice) VALUES
(1, 30, 'Red', 25, 50),
(2, 35, 'Blue', 30, 55),
(3, 40, 'Green', 40, 60),
(4, 25, 'Yellow', 20, 45),
(5, 20, 'Black', 35, 65);
```

c.

```
INSERT INTO Supplier_Part (scode, pcode, qty) VALUES
(1, 1, 100),
(1, 2, 150),
(2, 2, 200),
(3, 3, 120),
(4, 4, 80),
(5, 5, 90);
```

Output:

```
mysql> INSERT INTO Supplier (scode, sname, scity, turnover) VALUES
    -> (1, 'ABC Corporation', 'Bombay', 100),
    -> (2, 'XYZ Enterprises', 'Delhi', 150),
    -> (3, 'PQR Industries', 'Mumbai', 200),
    -> (4, 'LMN Limited', 'Bangalore', 120),
    -> (5, 'DEF Corporation', 'Chennai', 80);
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> INSERT INTO Part (pcode, weigh, color, cost, sellingprice) VALUES
    -> (1, 30, 'Red', 25, 50),
    -> (2, 35, 'Blue', 30, 55),
    -> (3, 40, 'Green', 40, 60),
    -> (4, 25, 'Yellow', 20, 45),
    -> (5, 20, 'Black', 35, 65);
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> INSERT INTO Supplier_Part (scode, pcode, qty) VALUES
    -> (1, 1, 100),
    -> (1, 2, 150),
    -> (2, 2, 200),
    -> (3, 3, 120),
    -> (4, 4, 80),
    -> (5, 5, 90);
Query OK, 6 rows affected (0.01 sec)
Records: 6  Duplicates: 0  Warnings: 0
```

2. Write appropriate SQL Statement for the following:

    1. Get the supplier number and part number in ascending order of supplier number.

    Commands:
    SELECT Supplier.scode, Part.pcode
    FROM Supplier
    JOIN Supplier_Part ON Supplier.scode = Supplier_Part.scode
    JOIN Part ON Supplier_Part.pcode = Part.pcode
    ORDER BY Supplier.scode;

    Output:

```
mysql> SELECT Supplier.scode, Part.pcode
    -> FROM Supplier
    -> JOIN Supplier_Part ON Supplier.scode = Supplier_Part.scode
    -> JOIN Part ON Supplier_Part.pcode = Part.pcode
    -> ORDER BY Supplier.scode;
+-------+-------+
| scode | pcode |
+-------+-------+
|     1 |     1 |
|     1 |     2 |
|     2 |     2 |
|     3 |     3 |
|     4 |     4 |
|     5 |     5 |
+-------+-------+
6 rows in set (0.00 sec)
```

2. Get the details of supplier who operate from Bombay with turnover 50.

Commands:
SELECT *
FROM Supplier
WHERE scity = 'Bombay' AND turnover = 50;

Output:

```
mysql> SELECT *
    -> FROM Supplier
    -> WHERE scity = 'Bombay' AND turnover = 50;
Empty set (0.00 sec)
```

3. Get the total number of supplier.

Commands:
SELECT COUNT(*) AS total_suppliers
FROM Supplier;

Output:

```
mysql> SELECT COUNT(*) AS total_suppliers
    -> FROM Supplier;
+-----------------+
| total_suppliers |
+-----------------+
|               5 |
+-----------------+
1 row in set (0.00 sec)
```

4. Get the part number weighing between 25 and 35.

Commands:
SELECT pcode
FROM Part
WHERE weigh BETWEEN 25 AND 35;
Output:

```
mysql> SELECT pcode
    -> FROM Part
    -> WHERE weigh BETWEEN 25 AND 35;
+-------+
| pcode |
+-------+
|     1 |
|     2 |
|     4 |
+-------+
3 rows in set (0.00 sec)
```

5. Get the supplier number whose turnover is null.

Commands:
SELECT scode
FROM Supplier
WHERE turnover IS NULL;

Output:

```
mysql> SELECT scode
    -> FROM Supplier
    -> WHERE turnover IS NULL;
Empty set (0.00 sec)
```

6. Get the part number that cost 20, 30 or 40 rupees.

Commands:
SELECT pcode
FROM Part
WHERE cost IN (20, 30, 40);

Output:

```
mysql> SELECT pcode
    -> FROM Part
    -> WHERE cost IN (20, 30, 40);
+-------+
| pcode |
+-------+
|     2 |
|     3 |
|     4 |
+-------+
3 rows in set (0.00 sec)
```

7. Get the total quantity of part 2 that is supplied.

Commands:
SELECT SUM(qty) AS total_quantity
FROM Supplier_Part
WHERE pcode = 2;

Output:

```
mysql> SELECT SUM(qty) AS total_quantity
    -> FROM Supplier_Part
    -> WHERE pcode = 2;
+----------------+
| total_quantity |
+----------------+
|            350 |
+----------------+
1 row in set (0.00 sec)
```

8. Get the name of supplier who supply part 2.

Commands:
SELECT Supplier.sname
FROM Supplier
JOIN Supplier_Part ON Supplier.scode = Supplier_Part.scode
WHERE Supplier_Part.pcode = 2;

Output:

```
mysql> SELECT Supplier.sname
    -> FROM Supplier
    -> JOIN Supplier_Part ON Supplier.scode = Supplier_Part.scode
    -> WHERE Supplier_Part.pcode = 2;
+-----------------+
| sname           |
+-----------------+
| ABC Corporation |
| XYZ Enterprises |
+-----------------+
2 rows in set (0.00 sec)
```

9. Get the part number whose cost is greater than the average cost.

Commands:
SELECT pcode
FROM Part
WHERE cost > (SELECT AVG(cost) FROM Part);

Output:

```
mysql> SELECT pcode
    -> FROM Part
    -> WHERE cost > (SELECT AVG(cost) FROM Part);
+-------+
| pcode |
+-------+
|     3 |
|     5 |
+-------+
2 rows in set (0.00 sec)
```

10. Get the supplier number and turnover in descending order of turnover.

Command:
SELECT scode, turnover
FROM Supplier
ORDER BY turnover DESC;

Output:

```
mysql> SELECT scode, turnover
    -> FROM Supplier
    -> ORDER BY turnover DESC;
+-------+----------+
| scode | turnover |
+-------+----------+
|     3 |   200.00 |
|     2 |   150.00 |
|     4 |   120.00 |
|     1 |   100.00 |
|     5 |    80.00 |
+-------+----------+
5 rows in set (0.00 sec)
```

EXPERIMENT-7,8

TITLE: Nested sql queries or Subquries

Objective: To understand the use **SQL Subquery**

1.Create the following two tables (EMP and DEPT)

EMP TABLE

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	500	800	20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7788	SCOTT	ANALYST	7566	09-DEC-82	3000		20
7839	KING	PRESIDENT		17-NOV-81	5000		10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	7788	12-JAN-83	1100		20
7900	JAMES	CLERK	7698	03-DEC-81	950		30

7902	FORD	ANALYST	7566	03-DEC-81	3000	20
7934	MILLER	CLERK	7782	23-JAN-82	1300	10

DEPT TABLE

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

Commands:

```
CREATE TABLE EMP (
    EMPNO INT PRIMARY KEY,
    ENAME VARCHAR(50),
    JOB VARCHAR(50),
    MGR INT,
    HIREDATE DATE,
    SAL INT,
    COMM INT,
    DEPTNO INT
);

CREATE TABLE DEPT (
    DEPTNO INT PRIMARY KEY,
    DNAME VARCHAR(50),
    LOC VARCHAR(50)
);

INSERT INTO EMP (EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO)
VALUES
(7369, 'SMITH', 'CLERK', 7902, '1980-12-17', 500, 800, 20),
(7499, 'ALLEN', 'SALESMAN', 7698, '1981-02-20', 1600, 300, 30),
(7521, 'WARD', 'SALESMAN', 7698, '1981-02-22', 1250, 500, 30),
(7566, 'JONES', 'MANAGER', 7839, '1981-04-02', 2975, NULL, 20),
(7654, 'MARTIN', 'SALESMAN', 7698, '1981-09-28', 1250, 1400, 30),
(7698, 'BLAKE', 'MANAGER', 7839, '1981-05-01', 2850, NULL, 30),
(7782, 'CLARK', 'MANAGER', 7839, '1981-06-09', 2450, NULL, 10),
(7788, 'SCOTT', 'ANALYST', 7566, '1982-12-09', 3000, NULL, 20),
(7839, 'KING', 'PRESIDENT', NULL, '1981-11-17', 5000, NULL, 10),
(7844, 'TURNER', 'SALESMAN', 7698, '1981-09-08', 1500, 0, 30),
(7876, 'ADAMS', 'CLERK', 7788, '1983-01-12', 1100, NULL, 20),
```

(7900, 'JAMES', 'CLERK', 7698, '1981-12-03', 950, NULL, 30),
(7902, 'FORD', 'ANALYST', 7566, '1981-12-03', 3000, NULL, 20),
(7934, 'MILLER', 'CLERK', 7782, '1982-01-23', 1300, NULL, 10);

INSERT INTO DEPT (DEPTNO, DNAME, LOC) VALUES
(10, 'ACCOUNTING', 'NEW YORK'),
(20, 'RESEARCH', 'DALLAS'),
(30, 'SALES', 'CHICAGO'),
(40, 'OPERATIONS', 'BOSTON');

Outputs:

```
mysql> SELECT * FROM DEPT;
+--------+------------+----------+
| DEPTNO | DNAME      | LOC      |
+--------+------------+----------+
|     10 | ACCOUNTING | NEW YORK |
|     20 | RESEARCH   | DALLAS   |
|     30 | SALES      | CHICAGO  |
|     40 | OPERATIONS | BOSTON   |
+--------+------------+----------+
4 rows in set (0.01 sec)
```

```
mysql> SELECT * FROM EMP;
+-------+--------+-----------+------+------------+------+------+--------+
| EMPNO | ENAME  | JOB       | MGR  | HIREDATE   | SAL  | COMM | DEPTNO |
+-------+--------+-----------+------+------------+------+------+--------+
|  7369 | SMITH  | CLERK     | 7902 | 1980-12-17 |  500 |  800 |     20 |
|  7499 | ALLEN  | SALESMAN  | 7698 | 1981-02-20 | 1600 |  300 |     30 |
|  7521 | WARD   | SALESMAN  | 7698 | 1981-02-22 | 1250 |  500 |     30 |
|  7566 | JONES  | MANAGER   | 7839 | 1981-04-02 | 2975 | NULL |     20 |
|  7654 | MARTIN | SALESMAN  | 7698 | 1981-09-28 | 1250 | 1400 |     30 |
|  7698 | BLAKE  | MANAGER   | 7839 | 1981-05-01 | 2850 | NULL |     30 |
|  7782 | CLARK  | MANAGER   | 7839 | 1981-06-09 | 2450 | NULL |     10 |
|  7788 | SCOTT  | ANALYST   | 7566 | 1982-12-09 | 3000 | NULL |     20 |
|  7839 | KING   | PRESIDENT | NULL | 1981-11-17 | 5000 | NULL |     10 |
|  7844 | TURNER | SALESMAN  | 7698 | 1981-09-08 | 1500 |    0 |     30 |
|  7876 | ADAMS  | CLERK     | 7788 | 1983-01-12 | 1100 | NULL |     20 |
|  7900 | JAMES  | CLERK     | 7698 | 1981-12-03 |  950 | NULL |     30 |
|  7902 | FORD   | ANALYST   | 7566 | 1981-12-03 | 3000 | NULL |     20 |
|  7934 | MILLER | CLERK     | 7782 | 1982-01-23 | 1300 | NULL |     10 |
+-------+--------+-----------+------+------------+------+------+--------+
14 rows in set (0.00 sec)
```

Write the Nested Queries for the following queries.

1. List the details of the emps whose Salaries more than the employee BLAKE.

   Commands:

   SELECT *

   FROM EMP

   WHERE SAL > (SELECT SAL

FROM EMP

WHERE ENAME = 'BLAKE');

Outputs:

```
mysql> SELECT *
    -> FROM EMP
    -> WHERE SAL > (SELECT SAL
    ->              FROM EMP
    ->              WHERE ENAME = 'BLAKE');
+-------+-------+-----------+------+------------+------+------+--------+
| EMPNO | ENAME | JOB       | MGR  | HIREDATE   | SAL  | COMM | DEPTNO |
+-------+-------+-----------+------+------------+------+------+--------+
|  7566 | JONES | MANAGER   | 7839 | 1981-04-02 | 2975 | NULL |     20 |
|  7788 | SCOTT | ANALYST   | 7566 | 1982-12-09 | 3000 | NULL |     20 |
|  7839 | KING  | PRESIDENT | NULL | 1981-11-17 | 5000 | NULL |     10 |
|  7902 | FORD  | ANALYST   | 7566 | 1981-12-03 | 3000 | NULL |     20 |
+-------+-------+-----------+------+------------+------+------+--------+
4 rows in set (0.00 sec)
```

2. List the emps whose Jobs are same as ALLEN.

   Commands:

   SELECT *

   FROM EMP

   WHERE JOB = (SELECT JOB

   FROM EMP

   WHERE ENAME = 'ALLEN');

   Outputs:

```
mysql> SELECT *
    -> FROM EMP
    -> WHERE JOB = (SELECT JOB
    ->              FROM EMP
    ->              WHERE ENAME = 'ALLEN');
+-------+--------+----------+------+------------+------+------+--------+
| EMPNO | ENAME  | JOB      | MGR  | HIREDATE   | SAL  | COMM | DEPTNO |
+-------+--------+----------+------+------------+------+------+--------+
|  7499 | ALLEN  | SALESMAN | 7698 | 1981-02-20 | 1600 |  300 |     30 |
|  7521 | WARD   | SALESMAN | 7698 | 1981-02-22 | 1250 |  500 |     30 |
|  7654 | MARTIN | SALESMAN | 7698 | 1981-09-28 | 1250 | 1400 |     30 |
|  7844 | TURNER | SALESMAN | 7698 | 1981-09-08 | 1500 |    0 |     30 |
+-------+--------+----------+------+------------+------+------+--------+
4 rows in set (0.00 sec)
```

3. List the Emps whose Sal is same as FORD or SMITH in desc order of Names.

   Commands:

   SELECT *

FROM EMP

WHERE SAL IN (SELECT SAL

FROM EMP

WHERE ENAME IN ('FORD', 'SMITH'))

ORDER BY ENAME DESC;

Outputs:

```
mysql> SELECT *
    -> FROM EMP
    -> WHERE SAL IN (SELECT SAL
    ->               FROM EMP
    ->               WHERE ENAME IN ('FORD', 'SMITH'))
    -> ORDER BY ENAME DESC;
+-------+-------+---------+------+------------+------+------+--------+
| EMPNO | ENAME | JOB     | MGR  | HIREDATE   | SAL  | COMM | DEPTNO |
+-------+-------+---------+------+------------+------+------+--------+
|  7369 | SMITH | CLERK   | 7902 | 1980-12-17 |  500 |  800 |     20 |
|  7788 | SCOTT | ANALYST | 7566 | 1982-12-09 | 3000 | NULL |     20 |
|  7902 | FORD  | ANALYST | 7566 | 1981-12-03 | 3000 | NULL |     20 |
+-------+-------+---------+------+------------+------+------+--------+
3 rows in set (0.00 sec)
```

4. List the emps Whose Jobs are same as MILLER or Sal is more than ALLEN.

Commands:

SELECT *

FROM EMP

WHERE JOB = (SELECT JOB

FROM EMP

WHERE ENAME = 'MILLER')

OR SAL > (SELECT SAL

FROM EMP

WHERE ENAME = 'ALLEN');

Outputs:

```
mysql> SELECT *
    -> FROM EMP
    -> WHERE JOB = (SELECT JOB
    ->              FROM EMP
    ->              WHERE ENAME = 'MILLER')
    ->    OR SAL > (SELECT SAL
    ->              FROM EMP
    ->              WHERE ENAME = 'ALLEN');
+-------+--------+-----------+------+------------+------+------+--------+
| EMPNO | ENAME  | JOB       | MGR  | HIREDATE   | SAL  | COMM | DEPTNO |
+-------+--------+-----------+------+------------+------+------+--------+
|  7369 | SMITH  | CLERK     | 7902 | 1980-12-17 |  500 |  800 |     20 |
|  7566 | JONES  | MANAGER   | 7839 | 1981-04-02 | 2975 | NULL |     20 |
|  7698 | BLAKE  | MANAGER   | 7839 | 1981-05-01 | 2850 | NULL |     30 |
|  7782 | CLARK  | MANAGER   | 7839 | 1981-06-09 | 2450 | NULL |     10 |
|  7788 | SCOTT  | ANALYST   | 7566 | 1982-12-09 | 3000 | NULL |     20 |
|  7839 | KING   | PRESIDENT | NULL | 1981-11-17 | 5000 | NULL |     10 |
|  7876 | ADAMS  | CLERK     | 7788 | 1983-01-12 | 1100 | NULL |     20 |
|  7900 | JAMES  | CLERK     | 7698 | 1981-12-03 |  950 | NULL |     30 |
|  7902 | FORD   | ANALYST   | 7566 | 1981-12-03 | 3000 | NULL |     20 |
|  7934 | MILLER | CLERK     | 7782 | 1982-01-23 | 1300 | NULL |     10 |
+-------+--------+-----------+------+------------+------+------+--------+
10 rows in set (0.00 sec)
```

5. Find the highest paid employee of sales department.

   Commands:

   SELECT *

   FROM EMP

   WHERE DEPTNO = (SELECT DEPTNO

   FROM DEPT

   WHERE DNAME = 'SALES')

   ORDER BY SAL DESC

   LIMIT 1;

   Outputs:

```
mysql> SELECT *
    -> FROM EMP
    -> WHERE DEPTNO = (SELECT DEPTNO
    ->                 FROM DEPT
    ->                 WHERE DNAME = 'SALES')
    -> ORDER BY SAL DESC
    -> LIMIT 1;
+-------+-------+---------+------+------------+------+------+--------+
| EMPNO | ENAME | JOB     | MGR  | HIREDATE   | SAL  | COMM | DEPTNO |
+-------+-------+---------+------+------------+------+------+--------+
|  7698 | BLAKE | MANAGER | 7839 | 1981-05-01 | 2850 | NULL |     30 |
+-------+-------+---------+------+------------+------+------+--------+
1 row in set (0.00 sec)
```

6. List the employees who are senior to most recently hired employee working under king.

   Commands:

   SELECT *

   FROM EMP

   WHERE HIREDATE < (SELECT MAX(HIREDATE)

   FROM EMP

   WHERE MGR = (SELECT EMPNO

   FROM EMP

   WHERE ENAME = 'KING'));

   Outputs:

```
mysql> SELECT *
    -> FROM EMP
    -> WHERE HIREDATE < (SELECT MAX(HIREDATE)
    ->                   FROM EMP
    ->                   WHERE MGR = (SELECT EMPNO
    ->                                FROM EMP
    ->                                WHERE ENAME = 'KING'));
+-------+-------+----------+------+------------+------+------+--------+
| EMPNO | ENAME | JOB      | MGR  | HIREDATE   | SAL  | COMM | DEPTNO |
+-------+-------+----------+------+------------+------+------+--------+
|  7369 | SMITH | CLERK    | 7902 | 1980-12-17 |  500 |  800 |     20 |
|  7499 | ALLEN | SALESMAN | 7698 | 1981-02-20 | 1600 |  300 |     30 |
|  7521 | WARD  | SALESMAN | 7698 | 1981-02-22 | 1250 |  500 |     30 |
|  7566 | JONES | MANAGER  | 7839 | 1981-04-02 | 2975 | NULL |     20 |
|  7698 | BLAKE | MANAGER  | 7839 | 1981-05-01 | 2850 | NULL |     30 |
+-------+-------+----------+------+------------+------+------+--------+
5 rows in set (0.00 sec)
```

7. List the names of the emps who are getting the highest sal dept wise.

Commands:

```
SELECT E.ENAME, E.DEPTNO
FROM EMP E
WHERE E.SAL IN (SELECT MAX(SAL)
        FROM EMP
        GROUP BY DEPTNO);
```

Outputs:

```
mysql> SELECT E.ENAME, E.DEPTNO
    -> FROM EMP E
    -> WHERE E.SAL IN (SELECT MAX(SAL)
    ->                 FROM EMP
    ->                 GROUP BY DEPTNO);
+-------+--------+
| ENAME | DEPTNO |
+-------+--------+
| BLAKE |     30 |
| SCOTT |     20 |
| KING  |     10 |
| FORD  |     20 |
+-------+--------+
4 rows in set (0.00 sec)
```

8. List the emps whose sal is equal to the average of max and minimum

Commands:

```
SELECT *
FROM EMP
WHERE SAL = (SELECT (MAX(SAL) + MIN(SAL)) / 2
        FROM EMP);
```

Outputs:

```
mysql> SELECT *
    -> FROM EMP
    -> WHERE SAL = (SELECT (MAX(SAL) + MIN(SAL)) / 2
    ->              FROM EMP);
Empty set (0.00 sec)
```

9. List the emps who joined in the company on the same date.

Commands:

```
SELECT *
FROM EMP E
WHERE HIREDATE IN (SELECT HIREDATE
        FROM EMP
        WHERE E.EMPNO <> EMPNO);
```

Outputs:

```
mysql> SELECT *
    -> FROM EMP E
    -> WHERE HIREDATE IN (SELECT HIREDATE
    ->                    FROM EMP
    ->                    WHERE E.EMPNO <> EMPNO);
+-------+-------+---------+------+------------+------+------+--------+
| EMPNO | ENAME | JOB     | MGR  | HIREDATE   | SAL  | COMM | DEPTNO |
+-------+-------+---------+------+------------+------+------+--------+
|  7900 | JAMES | CLERK   | 7698 | 1981-12-03 |  950 | NULL |     30 |
|  7902 | FORD  | ANALYST | 7566 | 1981-12-03 | 3000 | NULL |     20 |
+-------+-------+---------+------+------------+------+------+--------+
2 rows in set (0.00 sec)
```

10. Find out the emps who joined in the company before their Managers.

Commands:

```
SELECT *
FROM EMP E
WHERE HIREDATE < (SELECT HIREDATE
        FROM EMP
        WHERE EMPNO = E.MGR);
```

Outputs:

```
mysql> SELECT *
    -> FROM EMP E
    -> WHERE HIREDATE < (SELECT HIREDATE
    ->                   FROM EMP
    ->                   WHERE EMPNO = E.MGR);
+-------+-------+----------+------+------------+------+------+--------+
| EMPNO | ENAME | JOB      | MGR  | HIREDATE   | SAL  | COMM | DEPTNO |
+-------+-------+----------+------+------------+------+------+--------+
|  7369 | SMITH | CLERK    | 7902 | 1980-12-17 |  500 |  800 |     20 |
|  7499 | ALLEN | SALESMAN | 7698 | 1981-02-20 | 1600 |  300 |     30 |
|  7521 | WARD  | SALESMAN | 7698 | 1981-02-22 | 1250 |  500 |     30 |
|  7566 | JONES | MANAGER  | 7839 | 1981-04-02 | 2975 | NULL |     20 |
|  7698 | BLAKE | MANAGER  | 7839 | 1981-05-01 | 2850 | NULL |     30 |
|  7782 | CLARK | MANAGER  | 7839 | 1981-06-09 | 2450 | NULL |     10 |
+-------+-------+----------+------+------------+------+------+--------+
6 rows in set (0.01 sec)
```