UNIVERSITY OF PETROLEUM AND ENERGY STUDIES
DEHRADUN, UTTARAKHAND

Advanced Database Management System Lab

Lab File
(2023-2024)

for

4TH Semester

Submitted To:

Ms.Kalpana Rangra

Submitted By:

Eksha Malhotra

B. Tech. CSE [4th Semester]

R2142210300

500091532

Batch 1- DEVOPS

EXPERIMENT-1

1. Create the tables described below:

Table name: CLIENT_MASTER

Description: used to store client information.

```
mysql>  CREATE TABLE CLIENT_MASTER(CLIENTNO VARCHAR(6),NAME VARCHAR(20),ADDRESS1 VARCHAR(30),ADDRESS2 VARCHAR(30),CITY VARCHAR(15),PINCODE INT,STATE VARCHAR
(15),BALDUE FLOAT(10,2));
Query OK, 0 rows affected, 1 warning (0.01 sec)
```

Table Name: PRODUCT_MASTER

Description: used to store product information

```
mysql> CREATE TABLE PRODUCT_MASTER(PRODUCTNO VARCHAR(6),DESCRIPTION VARCHAR(15),PROFITPERCENT DECIMAL(4,2),UNITMEASURE VARCHAR(10),QTYONHAND INT,REORDERVL I
NT,SELLPRICE FLOAT(8,2),COSTPRICE FLOAT(8,2));
Query OK, 0 rows affected, 2 warnings (0.03 sec)
```

Table Name: SALESMAN_MASTER

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

```
mysql>  CREATE TABLE SALESMAN_MASTER(SALESMANO VARCHAR(6),SALESMANNAME VARCHAR(20),ADDRESS1 VARCHAR(20),ADDDRESS2 VARCHAR(30),CITY VARCHAR(20),P
INCODE INT,STATE VARCHAR(20),SALAMT REAL,TGTTOGET DECIMAL,YTDSALES DOUBLE(6,2),REMARKS VARCHAR(60));
Query OK, 0 rows affected, 1 warning (0.02 sec)
```

2. Insert the following data into their respective tables:

a) Data for CLIENT_MASTER table

```
mysql>  INSERT INTO CLIENT_MASTER VALUES("C00001","IVAN BANYROSS","BANDRA","KURLA COMPLEX","MUMBAI","400054","MAHARASHTRA",15000);
Query OK, 1 row affected (0.01 sec)

mysql>  INSERT INTO CLIENT_MASTER VALUES("C00002","MAMTA MUZUMDAR","NDC","MURAD NAGAR","MADRAS","780001","TAMIL NADU","0");
Query OK, 1 row affected (0.00 sec)

mysql>  INSERT INTO CLIENT_MASTER VALUES("C00003","CHHAYA BANKAR","BORIVALI","YOGI NAGAR","MUMBAI","400057","MAHARASHTRA",500);
Query OK, 1 row affected (0.00 sec)

mysql>  INSERT INTO CLIENT_MASTER VALUES("C00004","ASHWINI JOSHI","SECTOR 19","INDIRA NAGAR","BANGLORE","560001","KARNATAKA",0);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO CLIENT_MASTER VALUES("C00005","HANSAL COLACO","BANDRA","LINKING ROAD","MUMBAI","400060","MAHARASHTRA",200);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM CLIENT_MASTER;
+----------+---------------+-----------+---------------+----------+---------+-------------+----------+
| CLIENTNO | NAME          | ADDRESS1  | ADDRESS2      | CITY     | PINCODE | STATE       | BALDUE   |
+----------+---------------+-----------+---------------+----------+---------+-------------+----------+
| C00001   | IVAN BANYROSS | BANDRA    | KURLA COMPLEX | MUMBAI   |  400054 | MAHARASHTRA | 15000.00 |
| C00002   | MAMTA MUZUMDAR | NDC       | MURAD NAGAR   | MADRAS   |  780001 | TAMIL NADU  |     0.00 |
| C00003   | CHHAYA BANKAR | BORIVALI  | YOGI NAGAR    | MUMBAI   |  400057 | MAHARASHTRA |   500.00 |
| C00004   | ASHWINI JOSHI | SECTOR 19 | INDIRA NAGAR  | BANGLORE |  560001 | KARNATAKA   |     0.00 |
| C00005   | HANSAL COLACO | BANDRA    | LINKING ROAD  | MUMBAI   |  400060 | MAHARASHTRA |   200.00 |
+----------+---------------+-----------+---------------+----------+---------+-------------+----------+
5 rows in set (0.00 sec)
```

b) Data for PRODUCT_MASTER table:

```
mysql>  INSERT INTO PRODUCT_MASTER VALUES("P00001","T-SHIRT",5,"PIECE",200,50,350,250);
Query OK, 1 row affected (0.00 sec)

mysql>  INSERT INTO PRODUCT_MASTER VALUES("P0345","SHIRTS",6,"PIECE",150,50,500,350);
Query OK, 1 row affected (0.00 sec)

mysql>  INSERT INTO PRODUCT_MASTER VALUES("P06734","COTTON JEANS",5,"PIECE",100,20,600,450);
Query OK, 1 row affected (0.00 sec)

mysql>  INSERT INTO PRODUCT_MASTER VALUES("P07865", "JEANS",5,"PIECE",10,20,750,500);
Query OK, 1 row affected (0.00 sec)

mysql>  INSERT INTO PRODUCT_MASTER VALUES("P07868","TROUSERS",2,"PIECE",150,50,850,550);
Query OK, 1 row affected (0.00 sec)

mysql>  INSERT INTO PRODUCT_MASTER VALUES("P07885","PULL OVERS",2.5,"PIECE",80,30,700,450);
Query OK, 1 row affected (0.00 sec)

mysql>  INSERT INTO PRODUCT_MASTER VALUES("P07965","DENIM JEANS",4,"PIECE",100,40,350,250);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PRODUCT_MASTER VALUES("P07975","LYCRA TOPS",5,"PIECE",70,30,300,175);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PRODUCT_MASTER VALUES("P08865","SKIRTS",5,"PIECE",75,30,450,300);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM PRODUCT_MASTER;
+-----------+--------------+---------------+-------------+-----------+-----------+-----------+-----------+
| PRODUCTNO | DESCRIPTION  | PROFITPERCENT | UNITMEASURE | QTYONHAND | REORDERVL | 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       |        10 |        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:

```
mysql>  INSERT INTO SALESMAN_MASTER VALUES("S00001","AMAN","A/14","WORLI","MUMBAI",400002,"MAHARASHTRA",3000,100,50,"GOOD");
Query OK, 1 row affected (0.00 sec)

mysql>  INSERT INTO SALESMAN_MASTER VALUES("S00002","OMKAR","65","NAIRMAN","MUMBAI",400001,"MAHARASHTRA",3000,200,100,"GOOD");
Query OK, 1 row affected (0.00 sec)

mysql>  INSERT INTO SALESMAN_MASTER VALUES("S00003","RAJ","P-7","BANDRA","MUMBAI",400032,"MAHARASHTRA",3000,200,100,"GOOD");
Query OK, 1 row affected (0.01 sec)

mysql>  INSERT INTO SALESMAN_MASTER VALUES("S00004","ASHISH","A/5","JUHU","MUMBAI",400044,"MAHARASHTRA",3500,200,150,"GOOD");
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM SALESMAN_MASTER;
+-----------+-------------+----------+-----------+--------+---------+-------------+--------+----------+----------+---------+
| SALESMANO | SALESMANNAME | ADDRESS1 | ADDDRESS2 | CITY   | PINCODE | STATE       | SALAMT | TGTTOGET | YTDSALES | REMARKS |
+-----------+-------------+----------+-----------+--------+---------+-------------+--------+----------+----------+---------+
| S00001    | AMAN        | A/14     | WORLI     | MUMBAI |  400002 | MAHARASHTRA |   3000 |      100 |    50.00 | GOOD    |
| S00002    | OMKAR       | 65       | NAIRMAN   | 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)
```

3. Exercise on retrieving records from a table.

a. Find out the names of all the clients.

```
mysql> SELECT NAME FROM CLIENT_MASTER;
+----------------+
| NAME           |
+----------------+
| IVAN BANYROSS  |
| MAMTA MUZUMDAR |
| CHHAYA BANKAR  |
| ASHWINI JOSHI  |
| HANSAL COLACO  |
| DEEPAK SHARMA  |
+----------------+
6 rows in set (0.00 sec)
```

b. Retrieve the entire contents of the Client_Master table.

```
mysql> SELECT * FROM CLIENT_MASTER;
+---------+---------------+-------------+--------------+---------+---------+-------------+----------+
| CLIENTNO | NAME          | ADDRESS1    | ADDRESS2     | CITY    | PINCODE | STATE       | BALDUE   |
+---------+---------------+-------------+--------------+---------+---------+-------------+----------+
| C00001   | IVAN BANYROSS | BANDRA      | KURLA COMPLEX | MUMBAI  |  400054 | MAHARASHTRA | 15000.00 |
| C00002   | MAMTA MUZUMDAR | NDC        | MURAD NAGAR  | MADRAS  |  780001 | TAMIL NADU  |     0.00 |
| C00003   | CHHAYA BANKAR | BORIVALI    | YOGI NAGAR   | MUMBAI  |  400057 | MAHARASHTRA |   500.00 |
| C00004   | ASHWINI JOSHI | SECTOR 19   | INDIRA NAGAR | BANGLORE |  560001 | KARNATAKA   |     0.00 |
| C00005   | HANSAL COLACO | BANDRA      | LINKING ROAD | MUMBAI  |  400060 | MAHARASHTRA |   200.00 |
| C00006   | DEEPAK SHARMA | VASANT VIHAR | LANE 2       | BANGLORE |  560050 | KARNATAKA   |     0.00 |
+---------+---------------+-------------+--------------+---------+---------+-------------+----------+
6 rows in set (0.00 sec)
```

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

```
mysql> SELECT NAME,CITY,STATE FROM CLIENT_MASTER;
+----------------+----------+-------------+
| NAME           | CITY     | STATE       |
+----------------+----------+-------------+
| IVAN BANYROSS  | MUMBAI   | MAHARASHTRA |
| MAMTA MUZUMDAR | MADRAS   | TAMIL NADU  |
| CHHAYA BANKAR  | MUMBAI   | MAHARASHTRA |
| ASHWINI JOSHI  | BANGLORE | KARNATAKA   |
| HANSAL COLACO  | MUMBAI   | MAHARASHTRA |
| DEEPAK SHARMA  | BANGLORE | KARNATAKA   |
+----------------+----------+-------------+
6 rows in set (0.00 sec)
```

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

```
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.

```
mysql> SELECT * FROM CLIENT_MASTER WHERE CITY="MUMBAI";
+---------+-------------+---------+-------------+--------+--------+------------+---------+
| CLIENTNO | NAME        | ADDRESS1 | ADDRESS2    | CITY   | PINCODE | STATE      | BALDUE  |
+---------+-------------+---------+-------------+--------+--------+------------+---------+
| C00001  | IVAN BANYROSS | BANDRA  | KURLA COMPLEX | MUMBAI |  400054 | MAHARASHTRA | 15000.00 |
| C00003  | CHHAYA BANKAR | BORIVALI | YOGI NAGAR  | MUMBAI |  400057 | MAHARASHTRA |   500.00 |
| C00005  | HANSAL COLACO | BANDRA  | LINKING ROAD | MUMBAI |  400060 | MAHARASHTRA |   200.00 |
+---------+-------------+---------+-------------+--------+--------+------------+---------+
3 rows in set (0.00 sec)
```

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

```
mysql>  SELECT SALESMANNAME FROM SALESMAN_MASTER WHERE SALAMT="3000";
+--------------+
| SALESMANNAME |
+--------------+
| AMAN         |
| OMKAR        |
| RAJ          |
+--------------+
3 rows in set (0.00 sec)
```

4. Exercise on updating records in a table

a. Change the city of ClientNo ‘C00005’ to ‘Bangalore’.

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

mysql> SELECT * FROM CLIENT_MASTER;
+----------+---------------+-------------+---------------+----------+---------+-------------+----------+
| CLIENTNO | NAME          | ADDRESS1    | ADDRESS2      | CITY     | PINCODE | STATE       | BALDUE   |
+----------+---------------+-------------+---------------+----------+---------+-------------+----------+
| C00001   | IVAN BANYROSS | BANDRA      | KURLA COMPLEX | MUMBAI   |  400054 | MAHARASHTRA | 15000.00 |
| C00002   | MAMTA MUZUMDAR | NDC        | MURAD NAGAR   | MADRAS   |  780001 | TAMIL NADU  |     0.00 |
| C00003   | CHHAYA BANKAR | BORIVALI    | YOGI NAGAR    | MUMBAI   |  400057 | MAHARASHTRA |   500.00 |
| C00004   | ASHWINI JOSHI | SECTOR 19   | INDIRA NAGAR  | BANGLORE |  560001 | KARNATAKA   |     0.00 |
| C00005   | HANSAL COLACO | BANDRA      | LINKING ROAD  | BANGLORE |  400060 | MAHARASHTRA |   200.00 |
| C00006   | DEEPAK SHARMA | VASANT VIHAR | LANE 2       | BANGLORE |  560050 | KARNATAKA   |     0.00 |
+----------+---------------+-------------+---------------+----------+---------+-------------+----------+
6 rows in set (0.00 sec)
```

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

```
mysql>  UPDATE CLIENT_MASTER SET BALDUE=1000 WHERE CLIENTNO="C00001";
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM CLIENT_MASTER;
+----------+---------------+-------------+---------------+----------+---------+-------------+---------+
| CLIENTNO | NAME          | ADDRESS1    | ADDRESS2      | CITY     | PINCODE | STATE       | BALDUE  |
+----------+---------------+-------------+---------------+----------+---------+-------------+---------+
| C00001   | IVAN BANYROSS | BANDRA      | KURLA COMPLEX | MUMBAI   |  400054 | MAHARASHTRA | 1000.00 |
| C00002   | MAMTA MUZUMDAR | NDC        | MURAD NAGAR   | MADRAS   |  780001 | TAMIL NADU  |    0.00 |
| C00003   | CHHAYA BANKAR | BORIVALI    | YOGI NAGAR    | MUMBAI   |  400057 | MAHARASHTRA |  500.00 |
| C00004   | ASHWINI JOSHI | SECTOR 19   | INDIRA NAGAR  | BANGLORE |  560001 | KARNATAKA   |    0.00 |
| C00005   | HANSAL COLACO | BANDRA      | LINKING ROAD  | BANGLORE |  400060 | MAHARASHTRA |  200.00 |
| C00006   | DEEPAK SHARMA | VASANT VIHAR | LANE 2       | BANGLORE |  560050 | KARNATAKA   |    0.00 |
+----------+---------------+-------------+---------------+----------+---------+-------------+---------+
6 rows in set (0.00 sec)
```

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

```
mysql>  UPDATE PRODUCT_MASTER SET COSTPRICE=950 WHERE DESCRIPTION="TROUSERS";
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT* FROM PRODUCT_MASTER;
+-----------+--------------+--------------+-------------+-----------+-----------+-----------+-----------+
| PRODUCTNO | DESCRIPTION  | PROFITPERCENT | UNITMEASURE | QTYONHAND | REORDERVL | 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       |        10 |        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.

```
mysql> UPDATE SALESMAN_MASTER SET CITY = "PUNE";
Query OK, 4 rows affected (0.00 sec)
Rows matched: 4  Changed: 4  Warnings: 0
```

```
mysql> SELECT * FROM SALESMAN_MASTER;
+-----------+-------------+----------+-----------+------+---------+-------------+--------+----------+----------+---------+
| SALESMANO | SALESMANNAME | ADDRESS1 | ADDDRESS2 | CITY | PINCODE | STATE       | SALAMT | TGTTOGET | YTDSALES | REMARKS |
+-----------+-------------+----------+-----------+------+---------+-------------+--------+----------+----------+---------+
| S00001    | AMAN        | A/14     | WORLI     | PUNE |  400002 | MAHARASHTRA |   3000 |      100 |    50.00 | GOOD    |
| S00002    | OMKAR       | 65       | NAIRMAN   | 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)
```

5. Exercise on deleting records in a table.

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

```
mysql> DELETE FROM SALESMAN_MASTER WHERE SALAMT=3500;
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM SALESMAN_MASTER;
+-----------+-------------+----------+-----------+------+---------+-------------+--------+----------+----------+---------+
| SALESMANO | SALESMANNAME | ADDRESS1 | ADDDRESS2 | CITY | PINCODE | STATE       | SALAMT | TGTTOGET | YTDSALES | REMARKS |
+-----------+-------------+----------+-----------+------+---------+-------------+--------+----------+----------+---------+
| S00001    | AMAN        | A/14     | WORLI     | PUNE |  400002 | MAHARASHTRA |   3000 |      100 |    50.00 | GOOD    |
| S00002    | OMKAR       | 65       | NAIRMAN   | 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..

```
mysql> DELETE FROM PRODUCT_MASTER WHERE QTYONHAND=100;
Query OK, 2 rows affected (0.01 sec)

mysql> SELECT * FROM PRODUCT_MASTER;
+-----------+-------------+---------------+-------------+-----------+-----------+-----------+-----------+
| PRODUCTNO | DESCRIPTION | PROFITPERCENT | UNITMEASURE | QTYONHAND | REORDERVL | 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 |
| P07865    | JEANS       |          5.00 | PIECE       |        10 |        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 |
| P07975    | LYCRA TOPS  |          5.00 | PIECE       |        70 |        30 |    300.00 |    175.00 |
| P08865    | SKIRTS      |          5.00 | PIECE       |        75 |        30 |    450.00 |    300.00 |
+-----------+-------------+---------------+-------------+-----------+-----------+-----------+-----------+
7 rows in set (0.00 sec)
```

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

```
mysql> DELETE FROM CLIENT_MASTER WHERE STATE="TAMIL NADU";
Query OK, 1 row affected (0.00 sec)
```

```
mysql> SELECT * FROM CLIENT_MASTER;
+----------+--------------+-------------+---------------+----------+---------+-------------+---------+
| CLIENTNO | NAME         | ADDRESS1    | ADDRESS2      | CITY     | PINCODE | STATE       | BALDUE  |
+----------+--------------+-------------+---------------+----------+---------+-------------+---------+
| C00001   | IVAN BANYROSS | BANDRA      | KURLA COMPLEX | MUMBAI   |  400054 | MAHARASHTRA | 1000.00 |
| C00003   | CHHAYA BANKAR | BORIVALI    | YOGI NAGAR    | MUMBAI   |  400057 | MAHARASHTRA |  500.00 |
| C00004   | ASHWINI JOSHI | SECTOR 19   | INDIRA NAGAR  | BANGLORE |  560001 | KARNATAKA   |    0.00 |
| C00005   | HANSAL COLACO | BANDRA      | LINKING ROAD  | BANGLORE |  400060 | MAHARASHTRA |  200.00 |
| C00006   | DEEPAK SHARMA | VASANT VIHAR | LANE 2        | BANGLORE |  560050 | KARNATAKA   |    0.00 |
+----------+--------------+-------------+---------------+----------+---------+-------------+---------+
5 rows in set (0.00 sec)
```

6. Exercise on altering the table structure .

a. Add a column called ‘Telephone’ of data type integer to the Client_Master table.

```
mysql> ALTER TABLE CLIENT_MASTER ADD TELEPHONE INT;
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM CLIENT_MASTER;
+----------+--------------+-------------+---------------+----------+---------+-------------+---------+-----------+
| CLIENTNO | NAME         | ADDRESS1    | ADDRESS2      | CITY     | PINCODE | STATE       | BALDUE  | TELEPHONE |
+----------+--------------+-------------+---------------+----------+---------+-------------+---------+-----------+
| C00001   | IVAN BANYROSS | BANDRA      | KURLA COMPLEX | MUMBAI   |  400054 | MAHARASHTRA | 1000.00 |      NULL |
| C00003   | CHHAYA BANKAR | BORIVALI    | YOGI NAGAR    | MUMBAI   |  400057 | MAHARASHTRA |  500.00 |      NULL |
| C00004   | ASHWINI JOSHI | SECTOR 19   | INDIRA NAGAR  | BANGLORE |  560001 | KARNATAKA   |    0.00 |      NULL |
| C00005   | HANSAL COLACO | BANDRA      | LINKING ROAD  | BANGLORE |  400060 | MAHARASHTRA |  200.00 |      NULL |
| C00006   | DEEPAK SHARMA | VASANT VIHAR | LANE 2        | BANGLORE |  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.

```
mysql> ALTER TABLE PRODUCT_MASTER MODIFY SELLPRICE FLOAT(10,2);
Query OK, 0 rows affected, 1 warning (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 1

mysql> SELECT * FROM PRODUCT_MASTER;
+-----------+-------------+--------------+-------------+-----------+-----------+-----------+-----------+
| PRODUCTNO | DESCRIPTION | PROFITPERCENT | UNITMEASURE | QTYONHAND | REORDERVL | 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 |
| P07865    | JEANS       |         5.00 | PIECE       |        10 |        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 |
| P07975    | LYCRA TOPS  |         5.00 | PIECE       |        70 |        30 |    300.00 |    175.00 |
| P08865    | SKIRTS      |         5.00 | PIECE       |        75 |        30 |    450.00 |    300.00 |
+-----------+-------------+--------------+-------------+-----------+-----------+-----------+-----------+
7 rows in set (0.00 sec)
```

7. Exercise on deleting the table structure along with the data .

a. Destroy the table Client_Master along with its data.

```
mysql> DROP TABLE CLIENT_MASTER;
Query OK, 0 rows affected (0.02 sec)
```

8. Exercise on renaming the table

a. Change the name of the Salesman_Master to sman_mast.

```
mysql> ALTER TABLE SALESMAN_MASTER RENAME TO SMAN_MAST;
Query OK, 0 rows affected (0.01 sec)

mysql> SELECT * FROM SMAN_MAST;
+-----------+--------------+----------+-----------+------+---------+-------------+--------+----------+----------+---------+
| SALESMANO | SALESMANNAME | ADDRESS1 | ADDDRESS2 | CITY | PINCODE | STATE       | SALAMT | TGTTOGET | YTDSALES | REMARKS |
+-----------+--------------+----------+-----------+------+---------+-------------+--------+----------+----------+---------+
| S00001    | AMAN         | A/14     | WORLI     | PUNE |  400002 | MAHARASHTRA |   3000 |      100 |    50.00 | GOOD    |
| S00002    | OMKAR        | 65       | NAIRMAN   | 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 (0.01 sec)
```

```
mysql> DESC CLIENT_MASTER_1;
+-----------+---------------+------+-----+---------+-------+
| Field     | Type          | Null | Key | Default | Extra |
+-----------+---------------+------+-----+---------+-------+
| CLIENTNO  | varchar(6)    | NO   | PRI | NULL    |       |
| NAME      | varchar(20)   | NO   |     | NULL    |       |
| ADDRESS_1 | varchar(30)   | YES  |     | NULL    |       |
| ADDRESS_2 | varchar(30)   | YES  |     | NULL    |       |
| CITY      | varchar(15)   | YES  |     | NULL    |       |
| PINCODE   | int           | YES  |     | NULL    |       |
| STATE     | varchar(15)   | YES  |     | NULL    |       |
| BALDUE    | decimal(10,2) | YES  |     | NULL    |       |
+-----------+---------------+------+-----+---------+-------+
8 rows in set (0.04 sec)
```

```
mysql> CREATE TABLE PRODUCT_MASTER_1(PRODUCTNO VARCHAR(6) PRIMARY KEY CHECK (PRODUCTNO='%P') ,DESCRIPTIOM VARCHAR(15) NOT NULL,PROFITPERCENT DECIMAL(4,2) NO
T NULL,UNIT_MEASURE VARCHAR(10) NOT NULL,QTYONHAND INT(8) NOT NULL,REORDER_VL INT(8) NOT NULL,SELLPRICE DECIMAL(8,2) NOT NULL,COSTPRICE DECIMAL(8,2)NOT NULL
);
Query OK, 0 rows affected, 2 warnings (0.02 sec)
```

```
mysql> CREATE TABLE SALESMAN_MASTER_1(SALESMANNO VARCHAR(6) PRIMARY KEY CHECK(SALESMANNO='%S'),SALESMANNAME VARCHAR(20) NOT NULL,ADDRESS_1 VARCHAR(30) NOT N
ULL,ADDRESS_2 VARCHAR(30),CITY VARCHAR(20),PINCODE INT(8),STATE VARCHAR(20),SALAMT REAL(8,2) NOT NULL ,TGTTOGET DECIMAL(6,2) NOT NULL,YTDSALES DOUBLE(6,2) N
OT NULL, REMARKS VARCHAR(60));
Query OK, 0 rows affected, 3 warnings (0.01 sec)
```

EXPERIMENT-2

Title: 2. To understand and apply the concept of 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 ForeignKey.

1. Create the tables described below:

Table name: CLIENT_MASTER_1

```
mysql> CREATE TABLE CLIENT_MASTER_1(CLIENTNO VARCHAR(6) CHECK(CLIENTNO LIKE 'C%') PRIMARY KEY,NAME VARCHAR(20) NOT NULL,ADDRESS1 VARCHAR(30),ADDRESS2 VARCHA
R(30),CITY VARCHAR(15),PINCODE INT,STATE VARCHAR(15),BALDUE FLOAT(10,2));
Query OK, 0 rows affected, 1 warning (0.01 sec)

mysql> DESC CLIENT_MASTER_1;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| CLIENTNO | varchar(6)  | NO   | PRI | NULL    |       |
| NAME     | varchar(20) | NO   |     | NULL    |       |
| ADDRESS1 | varchar(30) | YES  |     | NULL    |       |
| ADDRESS2 | varchar(30) | YES  |     | NULL    |       |
| CITY     | varchar(15) | YES  |     | NULL    |       |
| PINCODE  | int         | YES  |     | NULL    |       |
| STATE    | varchar(15) | YES  |     | NULL    |       |
| BALDUE   | float(10,2) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
8 rows in set (0.00 sec)
```

Table Name: PRODUCT_MASTER_1

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

mysql> DESC PRODUCT_MASTER_1;
+--------------+--------------+------+-----+---------+-------+
| Field        | Type         | Null | Key | Default | Extra |
+--------------+--------------+------+-----+---------+-------+
| PRODUCTNO    | varchar(6)   | NO   | PRI | NULL    |       |
| DESCRIPTION  | varchar(15)  | NO   |     | NULL    |       |
| PROFITPERCENT | decimal(4,2) | NO   |     | NULL    |       |
| UNITMEASURE  | varchar(10)  | NO   |     | NULL    |       |
| QTYONHAND    | int          | NO   |     | NULL    |       |
| REOERDERVL   | int          | NO   |     | NULL    |       |
| SELLPRICE    | float(8,2)   | NO   |     | NULL    |       |
| COSTPRICE    | float(8,2)   | NO   |     | NULL    |       |
+--------------+--------------+------+-----+---------+-------+
8 rows in set (0.00 sec)
```

Table Name: SALESMAN_MASTER _1

```
mysql> CREATE TABLE SALESMAN_MASTER_1(SALESMANO VARCHAR(6) CHECK (SALESMANO LIKE 'S%') PRIMARY KEY,SALESMANNAME VARCHAR(20) NOT NULL,ADDRESS1 VARCHAR(30) NO
T NULL,ADRESS2 VARCHAR(30),CITY VARCHAR(20),PINCODE INT,STATE VARCHAR(20),SALAMT DOUBLE CHECK(SALAMT!=0) NOT NULL,TGTTOGET DECIMAL(10,0) CHECK(TGTTOGET!=0)
NOT NULL,YTDSALES DOUBLE(6,2) NOT NULL,REMARKS VARCHAR(60));
Query OK, 0 rows affected, 1 warning (0.02 sec)

mysql> DESC SALESMAN_MASTER_1;
+--------------+---------------+------+-----+---------+-------+
| Field        | Type          | Null | Key | Default | Extra |
+--------------+---------------+------+-----+---------+-------+
| SALESMANO    | varchar(6)    | NO   | PRI | NULL    |       |
| SALESMANNAME | varchar(20)   | NO   |     | NULL    |       |
| ADDRESS1     | varchar(30)   | NO   |     | NULL    |       |
| ADRESS2      | varchar(30)   | YES  |     | NULL    |       |
| CITY         | varchar(20)   | YES  |     | NULL    |       |
| PINCODE      | int           | YES  |     | NULL    |       |
| STATE        | varchar(20)   | YES  |     | NULL    |       |
| SALAMT       | double        | NO   |     | NULL    |       |
| TGTTOGET     | decimal(10,0) | NO   |     | NULL    |       |
| YTDSALES     | double(6,2)   | NO   |     | NULL    |       |
| REMARKS      | varchar(60)   | YES  |     | NULL    |       |
+--------------+---------------+------+-----+---------+-------+
11 rows in set (0.00 sec)
```

2. **Reinsert the data in these two tables based upon Lab 1.**
3. **Display the contents of each table.**

```
mysql> INSERT INTO CLIENT_MASTER_1 VALUES("C00001","IVAN BANYROSS","BANDRA","KURLA COMPLEX","MUMBAI","400054","MAHARASHTRA",15000);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO CLIENT_MASTER_1 VALUES("C00002","MAMTA MUZUMDAR","NDC","MURAD NAGAR","MADRAS","780001","TAMIL NADU",0);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO CLIENT_MASTER_1 VALUES("C00003","CHHAYA BANKAR","BORIVALI","YOGI NAGAR","MUMBAI","400057","MAHARASHTRA",500);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO CLIENT_MASTER_1 VALUES("C00004","ASHWINI JOSHI","SECTOR 19","INDIRA NAGAR","BANGLORE","560001","KARNATAKA",0);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO CLIENT_MASTER_1 VALUES("C00005","HANSAL COLACO","BANDRA","LINKING ROAD","MUMBAI","400060","MAHARASHTRA",200);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM CLIENT_MASTER_1;
+----------+---------------+----------+---------------+----------+---------+-------------+----------+
| CLIENTNO | NAME          | ADDRESS1 | ADDRESS2      | CITY     | PINCODE | STATE       | BALDUE   |
+----------+---------------+----------+---------------+----------+---------+-------------+----------+
| C00001   | IVAN BANYROSS | BANDRA   | KURLA COMPLEX | MUMBAI   |  400054 | MAHARASHTRA | 15000.00 |
| C00002   | MAMTA MUZUMDAR | NDC     | MURAD NAGAR   | MADRAS   |  780001 | TAMIL NADU  |     0.00 |
| C00003   | CHHAYA BANKAR | BORIVALI | YOGI NAGAR    | MUMBAI   |  400057 | MAHARASHTRA |   500.00 |
| C00004   | ASHWINI JOSHI | SECTOR 19 | INDIRA NAGAR  | BANGLORE |  560001 | KARNATAKA   |     0.00 |
| C00005   | HANSAL COLACO | BANDRA   | LINKING ROAD  | MUMBAI   |  400060 | MAHARASHTRA |   200.00 |
+----------+---------------+----------+---------------+----------+---------+-------------+----------+
5 rows in set (0.00 sec)
```

```
mysql> INSERT INTO PRODUCT_MASTER_1 VALUES("P00001","T-SHIRT",5,"PIECE",200,50,350,250);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO PRODUCT_MASTER_1 VALUES("P0345","SHIRT",6,"PIECE",150,50,500,350);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PRODUCT_MASTER_1 VALUES("P06734","COTTON JEANS",5,"PIECE",100,20,600,450);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PRODUCT_MASTER_1 VALUES("P07865"," JEANS",5,"PIECE",10,20,750,500);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PRODUCT_MASTER_1 VALUES("P07868","TROUSERS",2,"PIECE",150,50,850,550);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PRODUCT_MASTER_1 VALUES("P07885","PULL OVERS",2.50,"PIECE",80,30,700,450);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PRODUCT_MASTER_1 VALUES("P07965","DENIM JEANS",4,"PIECE",100,40,350,250);;
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO PRODUCT_MASTER_1 VALUES("P07975","LYCRA TOPS",5,"PIECE",70,30,300,175);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PRODUCT_MASTER_1 VALUES("P08865","SKIRTS",5,"PIECE",75,30,450,300);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT* FROM PRODUCT_MASTER_1;
+-----------+--------------+---------------+-------------+-----------+------------+-----------+-----------+
| PRODUCTNO | DESCRIPTION  | PROFITPERCENT | UNITMEASURE | QTYONHAND | REOERDERVL | SELLPRICE | COSTPRICE |
+-----------+--------------+---------------+-------------+-----------+------------+-----------+-----------+
| P00001    | T-SHIRT      |          5.00 | PIECE       |       200 |         50 |    350.00 |    250.00 |
| P0345     | SHIRT        |          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       |        10 |         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)
```

```
mysql> INSERT INTO SALESMAN_MASTER_1 VALUES("S00001","AMAN","A/14","WORLI","MUMBAI",400002,"MAHARASHTRA",3000,100,50,"GOOD");
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO SALESMAN_MASTER_1 VALUES("S00002","OMKAR","65","NAIRMAN","MUMBAI",400001,"MAHARASHTRA",3000,200,100,"GOOD");
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO SALESMAN_MASTER_1 VALUES("S00003","RAJ","P-7","BANDRA","MUMBAI",400032,"MAHARASHTRA",3000,200,100,"GOOD");
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO SALESMAN_MASTER_1 VALUES("S00004","ASHISH","A/5","JUHU","MUMBAI",400044,"MAHARASHTRA",3500,200,150,"GOOD");
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM SALESMAN_MASTER_1;
+----------+-------------+----------+---------+--------+---------+-------------+--------+----------+----------+---------+
| SALESMANO | SALESMANNAME | ADDRESS1 | ADRESS2 | CITY   | PINCODE | STATE       | SALAMT | TGTTOGET | YTDSALES | REMARKS |
+----------+-------------+----------+---------+--------+---------+-------------+--------+----------+----------+---------+
| S00001    | AMAN         | A/14     | WORLI   | MUMBAI |  400002 | MAHARASHTRA |   3000 |      100 |    50.00 | GOOD    |
| S00002    | OMKAR        | 65       | NAIRMAN | 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)
```

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

```
mysql> CREATE TABLE AUTHOR(AUTHOR_ID VARCHAR(5) PRIMARY KEY,LASTNAME VARCHAR(15)
NOT NULL,FIRSTNAME VARCHAR(15) NOT NULL,EMAIL VARCHAR(40),CITY VARCHAR(15),COUNTR
Y VARCHAR(15));
Query OK, 0 rows affected (0.02 sec)

mysql> DESC AUTHOR;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| AUTHOR_ID | varchar(5)  | NO   | PRI | NULL    |       |
| LASTNAME  | varchar(15) | NO   |     | NULL    |       |
| FIRSTNAME | varchar(15) | NO   |     | NULL    |       |
| EMAIL     | varchar(40) | YES  |     | NULL    |       |
| CITY      | varchar(15) | YES  |     | NULL    |       |
| COUNTRY   | varchar(15) | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

4. **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

```
mysql> CREATE TABLE BOOK(BOOK_ID VARCHAR(5) CHECK(BOOK_ID LIKE 'B%') PRIMARY KEY,
BOOK_TITLE VARCHAR(50) NOT NULL,COPIES INT);
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> DESC BOOK;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| BOOK_ID    | varchar(5)  | NO   | PRI | NULL    |       |
| BOOK_TITLE | varchar(50) | NO   |     | NULL    |       |
| COPIES     | int         | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

5. **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

```
mysql> CREATE TABLE AUTHOR_LIST(AUTHOR_ID VARCHAR(15),PRIMARY KEY (AUTHOR_ID,BOOK_ID),BOOK_ID VARCHAR(15),ROLE VARCHAR(15));
Query OK, 0 rows affected (0.38 sec)

mysql> DESC AUTHOR_LIST;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| AUTHOR_ID | varchar(15) | NO   | PRI | NULL    |       |
| BOOK_ID   | varchar(15) | NO   | PRI | NULL    |       |
| ROLE      | varchar(15) | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
3 rows in set (0.10 sec)
```

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

```
mysql> INSERT INTO AUTHOR VALUES("A0001","BHAGAT","CHETAN","","","INDIA");
Query OK, 1 row affected (0.15 sec)

mysql> INSERT INTO AUTHOR VALUES("A0002","TAGORE","RABINDRANTH","","","INDIA");
Query OK, 1 row affected (0.02 sec)

mysql> INSERT INTO AUTHOR VALUES("A0003","BOND","RUSKIN","","","INDIA");
Query OK, 1 row affected (0.03 sec)

mysql> INSERT INTO AUTHOR VALUES("A0004","THAROOR","SHASHI","","","INDIA");
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM AUTHOR;
+-----------+----------+-------------+-------+------+---------+
| AUTHOR_ID | LASTNAME | FIRSTNAME   | EMAIL | CITY | COUNTRY |
+-----------+----------+-------------+-------+------+---------+
| A0001     | BHAGAT   | CHETAN      |       |      | INDIA   |
| A0002     | TAGORE   | RABINDRANTH |       |      | INDIA   |
| A0003     | BOND     | RUSKIN      |       |      | INDIA   |
| A0004     | THAROOR  | SHASHI      |       |      | INDIA   |
+-----------+----------+-------------+-------+------+---------+
4 rows in set (0.03 sec)
```

```
mysql> INSERT INTO BOOK VALUES("B0001","FIVE POINT SOMEONE",64);
Query OK, 1 row affected (0.14 sec)

mysql> INSERT INTO BOOK VALUES("B0002","GITANJALI",124);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO BOOK VALUES("B0003","THE ROOM ON THE ROOF",345);
Query OK, 1 row affected (0.10 sec)

mysql> INSERT INTO BOOK VALUES("B0004","WHY I AM A HINDU",174);
Query OK, 1 row affected (0.10 sec)

mysql> SELECT* FROM BOOK;
+---------+---------------------+--------+
| BOOK_ID | BOOK_TITLE          | COPIES |
+---------+---------------------+--------+
| B0001   | FIVE POINT SOMEONE  |     64 |
| B0002   | GITANJALI           |    124 |
| B0003   | THE ROOM ON THE ROOF |    345 |
| B0004   | WHY I AM A HINDU    |    174 |
+---------+---------------------+--------+
4 rows in set (0.03 sec)
```

```
mysql> INSERT INTO AUTHOR_LIST VALUES("A0001","B0001","AUTHOR");
Query OK, 1 row affected (0.30 sec)

mysql> INSERT INTO AUTHOR_LIST VALUES("A0002","B0002","AUTHOR");
Query OK, 1 row affected (0.04 sec)

mysql> INSERT INTO AUTHOR_LIST VALUES("A0003","B0003","AUTHOR");
Query OK, 1 row affected (0.04 sec)

mysql> INSERT INTO AUTHOR_LIST VALUES("A0004","B0004","AUTHOR");
Query OK, 1 row affected (0.02 sec)

mysql> SELECT * FROM AUTHOR_LIST;
+-----------+---------+--------+
| AUTHOR_ID | BOOK_ID | ROLE   |
+-----------+---------+--------+
| A0001     | B0001   | AUTHOR |
| A0002     | B0002   | AUTHOR |
| A0003     | B0003   | AUTHOR |
| A0004     | B0004   | AUTHOR |
+-----------+---------+--------+
4 rows in set (0.04 sec)
```

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

```
mysql> ALTER TABLE AUTHOR_LIST
    -> ADD PUBLISHER VARCHAR(30);
Query OK, 0 rows affected (0.21 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESC AUTHOR_LIST;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| AUTHOR_ID | varchar(15) | NO   | PRI | NULL    |       |
| BOOK_ID   | varchar(15) | NO   | PRI | NULL    |       |
| ROLE      | varchar(15) | YES  |     | NULL    |       |
| PUBLISHER | varchar(30) | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
4 rows in set (0.05 sec)
```

EXPERIMENT -03

TITLE: 3.TO UNDERSTAND AND USE SQL SUB-QUERY

OBJECTIVE: TO UNDERSTAND THE USE OF SQL SUB-QUERY

1. Create the following table.

Supplier-(scode,sname,scity,turnover)

Part-(pcode,weight,color,cost,sellingprice)

Supplier_Part-(scode,pcode,qty)

```
mysql> CREATE TABLE SUPPLIER(SCODE VARCHAR(5) PRIMARY KEY,SNAME VARCHAR(30),
SCITY VARCHAR(30),TURNOVER INT);
Query OK, 0 rows affected (0.25 sec)
```

```
mysql> DESC SUPPLIER;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| SCODE    | varchar(5)  | NO   | PRI | NULL    |       |
| SNAME    | varchar(30) | YES  |     | NULL    |       |
| SCITY    | varchar(30) | YES  |     | NULL    |       |
| TURNOVER | int         | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
4 rows in set (0.10 sec)
```

```
mysql> CREATE TABLE PART(PCODE VARCHAR(5) PRIMARY KEY,WEIGHT INT,COLOR VARCH
AR(10),COST INT,SELLINGPRICE INT);
Query OK, 0 rows affected (0.20 sec)

mysql> DESC PART;
+-------------+------------+------+-----+---------+-------+
| Field       | Type       | Null | Key | Default | Extra |
+-------------+------------+------+-----+---------+-------+
| PCODE       | varchar(5) | NO   | PRI | NULL    |       |
| WEIGHT      | int        | YES  |     | NULL    |       |
| COLOR       | varchar(10)| YES  |     | NULL    |       |
| COST        | int        | YES  |     | NULL    |       |
| SELLINGPRICE| int        | YES  |     | NULL    |       |
+-------------+------------+------+-----+---------+-------+
5 rows in set (0.07 sec)
```

```
mysql> CREATE TABLE SUPPLIER_PART(SCODE VARCHAR(5) REFERENCES SUPPLIER(SCODE
),PCODE VARCHAR(5) REFERENCES PART(PCODE),QTY INT);
Query OK, 0 rows affected (0.14 sec)

mysql> DESC SUPPLIER_PART;
+-------+-----------+------+-----+---------+-------+
| Field | Type      | Null | Key | Default | Extra |
+-------+-----------+------+-----+---------+-------+
| SCODE | varchar(5)| YES  |     | NULL    |       |
| PCODE | varchar(5)| YES  |     | NULL    |       |
| QTY   | int       | YES  |     | NULL    |       |
+-------+-----------+------+-----+---------+-------+
3 rows in set (0.06 sec)
```

POPULATE THE TABLE

2 a)TABLE OF SUPPLIER

```
mysql> INSERT INTO SUPPLIER VALUES("S0001","ABC","DELHI",100);
Query OK, 1 row affected (0.16 sec)

mysql> INSERT INTO SUPPLIER VALUES("S0002","DEF","BOMBAY",120);
Query OK, 1 row affected (0.05 sec)

mysql> INSERT INTO SUPPLIER VALUES("S0003","GHI","CHENNAI",300);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO SUPPLIER VALUES("S0004","XYZ","KOLKATA",70);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO SUPPLIER VALUES("S0005","MNO","BOMBAY",30);
Query OK, 1 row affected (0.09 sec)

mysql> INSERT INTO SUPPLIER VALUES("S0006","JKL","BOMBAY",50);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO SUPPLIER VALUES("S0007","PQR","DELHI",NULL);
Query OK, 1 row affected (0.01 sec)

mysql> SELECT *FROM SUPPLIER;
+-------+-------+---------+----------+
| SCODE | SNAME | SCITY   | TURNOVER |
+-------+-------+---------+----------+
| S0001 | ABC   | DELHI   |      100 |
| S0002 | DEF   | BOMBAY  |      120 |
| S0003 | GHI   | CHENNAI |      300 |
| S0004 | XYZ   | KOLKATA |       70 |
| S0005 | MNO   | BOMBAY  |       30 |
| S0006 | JKL   | BOMBAY  |       50 |
| S0007 | PQR   | DELHI   |     NULL |
+-------+-------+---------+----------+
7 rows in set (0.04 sec)
```

2B)TABLE OF PART

```
mysql> INSERT INTO PART VALUES("P0001",25,"RED",20,25);
Query OK, 1 row affected (0.13 sec)

mysql> INSERT INTO PART VALUES("P0002",50,"BLUE",25,24);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PART VALUES("P0003",20,"RED",30,35);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PART VALUES("P0004",34,"GREEN",40,45);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM PART;
+-------+--------+-------+------+--------------+
| PCODE | WEIGHT | COLOR | COST | SELLINGPRICE |
+-------+--------+-------+------+--------------+
| P0001 |     25 | RED   |   20 |           25 |
| P0002 |     50 | BLUE  |   25 |           24 |
| P0003 |     20 | RED   |   30 |           35 |
| P0004 |     34 | GREEN |   40 |           45 |
+-------+--------+-------+------+--------------+
4 rows in set (0.03 sec)
```

2C)TABLE OF SUPPLIER_PART

```
mysql> INSERT INTO SUPPLIER_PART VALUES("S0001","P0002",120);
Query OK, 1 row affected (0.11 sec)

mysql> INSERT INTO SUPPLIER_PART VALUES("S0001","P0003",100);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO SUPPLIER_PART VALUES("S0002","P0002",200);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO SUPPLIER_PART VALUES("S0002","P0004",120);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO SUPPLIER_PART VALUES("S0003","P0001",50);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO SUPPLIER_PART VALUES("S0003","P0002",60);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO SUPPLIER_PART VALUES("S0004","P0002",40);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO SUPPLIER_PART VALUES("S0004","P0003",70);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO SUPPLIER_PART VALUES("S0005","P0003",90);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO SUPPLIER_PART VALUES("S0005","P0002",100);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO SUPPLIER_PART VALUES("S0006","P0004",100);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO SUPPLIER_PART VALUES("S0006","P0001",60);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO SUPPLIER_PART VALUES("S0007","P0004",90);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO SUPPLIER_PART VALUES("S0007","P0001",45);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM SUPPLIER_PART;
+-------+-------+------+
| SCODE | PCODE | QTY  |
+-------+-------+------+
| S0001 | P0002 |  120 |
| S0001 | P0003 |  100 |
| S0002 | P0002 |  200 |
| S0002 | P0004 |  120 |
| S0003 | P0001 |   50 |
| S0003 | P0002 |   60 |
| S0004 | P0002 |   40 |
| S0004 | P0003 |   70 |
| S0005 | P0003 |   90 |
| S0005 | P0002 |  100 |
| S0006 | P0004 |  100 |
| S0006 | P0001 |   60 |
| S0007 | P0004 |   90 |
| S0007 | P0001 |   45 |
+-------+-------+------+
14 rows in set (0.02 sec)
```

Q3. WRITE APPROPRIATE SQL STATEMENT FOR THE FOLLOWING.

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

```
mysql> SELECT SCODE,PCODE FROM SUPPLIER_PART ORDER BY SCODE ASC;
+-------+-------+
| SCODE | PCODE |
+-------+-------+
| S0001 | P0002 |
| S0001 | P0003 |
| S0002 | P0002 |
| S0002 | P0004 |
| S0003 | P0001 |
| S0003 | P0002 |
| S0004 | P0002 |
| S0004 | P0003 |
| S0005 | P0003 |
| S0005 | P0002 |
| S0006 | P0004 |
| S0006 | P0001 |
| S0007 | P0004 |
| S0007 | P0001 |
+-------+-------+
14 rows in set (0.09 sec)
```

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

```
mysql> SELECT * FROM SUPPLIER WHERE SCITY="BOMBAY" AND TURNOVER=50;
+-------+-------+--------+----------+
| SCODE | SNAME | SCITY  | TURNOVER |
+-------+-------+--------+----------+
| S0006 | JKL   | BOMBAY |       50 |
+-------+-------+--------+----------+
1 row in set (0.10 sec)
```

3.Get the total number of supplier.

```
mysql> SELECT COUNT(SCODE) FROM SUPPLIER;
+--------------+
| COUNT(SCODE) |
+--------------+
|            7 |
+--------------+
1 row in set (0.13 sec)
```

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

```
mysql> SELECT PCODE FROM PART WHERE WEIGHT BETWEEN 25 AND 35;
+-------+
| PCODE |
+-------+
| P0001 |
| P0004 |
+-------+
2 rows in set (0.09 sec)
```

5.Get the supplier number whose turnover is null.

```
mysql> SELECT SCODE FROM SUPPLIER WHERE TURNOVER IS NULL;
+-------+
| SCODE |
+-------+
| S0007 |
+-------+
1 row in set (0.08 sec)
```

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

```
mysql> SELECT PCODE FROM PART WHERE COST IN (20,30,40);
+-------+
| PCODE |
+-------+
| P0001 |
| P0003 |
| P0004 |
+-------+
3 rows in set (0.09 sec)
```

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

```
mysql> SELECT SUM(QTY) FROM SUPPLIER_PART WHERE PCODE="P0002";
+----------+
| SUM(QTY) |
+----------+
|      520 |
+----------+
1 row in set (0.10 sec)
```

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

```
mysql> SELECT SUPPLIER.SNAME FROM SUPPLIER INNER JOIN SUPPLIER_PART ON SUPPLIER.SCODE=SUPPLIER_PART.SCODE WHERE SUPPLIER_PART.PCODE
="P0002";
+-------+
| SNAME |
+-------+
| ABC   |
| DEF   |
| GHI   |
| XYZ   |
| MNO   |
+-------+
5 rows in set (0.10 sec)
```

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

```
mysql> SELECT PCODE FROM PART WHERE COST>(SELECT AVG(COST) FROM PART);
+-------+
| PCODE |
+-------+
| P0003 |
| P0004 |
+-------+
2 rows in set (0.08 sec)
```

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

```
mysql> SELECT SCODE,TURNOVER FROM SUPPLIER ORDER BY TURNOVER DESC;
+-------+----------+
| SCODE | TURNOVER |
+-------+----------+
| S0003 |      300 |
| S0002 |      120 |
| S0001 |      100 |
| S0004 |       70 |
| S0006 |       50 |
| S0005 |       30 |
| S0007 |     NULL |
+-------+----------+
7 rows in set (0.10 sec)
```

EXPERIMENT-04

TITLE: 4. USE OF INBUILT FUNCTIONS AND RELATIONAL ALGEBRA OPERATION.

OBJECTIVE: TO UNDERSTAND THE USE OF INBUILT FUNCTION AND RELATIONAL ALGEBRA WITH SQL QUERY.

1. CREATE THE FOLLOEING TWO TABLES (EMP AND DEPT).

```
mysql> CREATE TABLE EMP(EMPNO INT,ENAME VARCHAR(15),JOB VARCHAR(20),MGR INT,HIREDATE DATE,SAL NUMERIC,COMM NUMERIC
,DEPTNO INT);
Query OK, 0 rows affected (0.14 sec)
```

EMP TABLE

```
mysql> INSERT INTO EMP VALUES(7369,"SMITH","CLERK",7902,'1989-12-17',500,800,20);
Query OK, 1 row affected (0.04 sec)

mysql> INSERT INTO EMP VALUES(7499,"ALLEN","SALESMAN",7968,'1981-02-20',1600,300,30);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO EMP VALUES(7521,"WARD","SALESMAN",7698,'1981-02-22',1250,500,30);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO EMP VALUES(7566,"JONES","MANAGER",7839,'1981-04-02',2975,NULL,20);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO EMP VALUES(7654,"MARTIN","SALESMAN",7698,'1981-09-28',1250,1400,30);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO EMP VALUES(7698,"BLAKE","MANAGER",7839,'1981-05-01',2850,NULL,30);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO EMP VALUES(7782,"CLARK","MANAGER",7839,'1981-06-09',2450,NULL,10);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO EMP VALUES(7788,"SCOTT","ANALYST",7566,'1982-12-09',3000,NULL,20);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO EMP VALUES(7839,"KING","PRESIDENT",NULL,'1981-11-17',5000,NULL,10);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO EMP VALUES(7844,"TURNER","SALESMAN",7698,'1981-09-08',1500,0,30);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO EMP VALUES(7876,"ADBMS","CLERK",7788,'1983-01-12',1100,NULL,20);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO EMP VALUES(7900,"JAMES","CLERK",7698,'1981-12-03',950,NULL,30);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO EMP VALUES(7902,"FORD","ANALYST",7566,'1981-12-03',3000,NULL,20);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO EMP VALUES(7934,"MILLER","CLERK",7782,'1982-12-23',1300,NULL,10);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM EMP;
+-------+--------+-----------+------+------------+------+------+--------+
| EMPNO | ENAME  | JOB       | MGR  | HIREDATE   | SAL  | COMM | DEPTNO |
+-------+--------+-----------+------+------------+------+------+--------+
|  7369 | SMITH  | CLERK     | 7902 | 1989-12-17 |  500 |  800 |     20 |
|  7499 | ALLEN  | SALESMAN  | 7968 | 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 | ADBMS  | 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 | 1982-12-23 | 1300 | NULL |     10 |
+-------+--------+-----------+------+------------+------+------+--------+
14 rows in set (0.01 sec)
```

DEPT TABLE

```
mysql> CREATE TABLE DEPT(DEPTNO NUMERIC,DNAME VARCHAR(15),LOC VARCHAR(20));
Query OK, 0 rows affected (0.03 sec)

mysql> INSERT INTO DEPT VALUES(10,"ACCOUNTING","NEW YORK");
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO DEPT VALUES(20,"RESEARCH","DALLAS");
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO DEPT VALUES(30,"SALES","CHICAGO");
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO DEPT VALUES(40,"OPERATIONS","BOSTON");
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM DEPT;
+--------+------------+----------+
| DEPTNO | DNAME      | LOC      |
+--------+------------+----------+
|     10 | ACCOUNTING | NEW YORK |
|     20 | RESEARCH   | DALLAS   |
|     30 | SALES      | CHICAGO  |
|     40 | OPERATIONS | BOSTON   |
+--------+------------+----------+
4 rows in set (0.00 sec)
```

WRITE THE NESTED QUERIES FOR THE FOLLOWING QUERIES

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

```
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 (0.02 sec)
```

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

```
mysql> SELECT * FROM EMP WHERE JOB=(SELECT JOB FROM EMP WHERE ENAME="ALLEN");
+-------+--------+----------+------+------------+------+------+--------+
| EMPNO | ENAME  | JOB      | MGR  | HIREDATE   | SAL  | COMM | DEPTNO |
+-------+--------+----------+------+------------+------+------+--------+
|  7499 | ALLEN  | SALESMAN | 7968 | 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.

```
mysql> SELECT * FROM EMP WHERE SAL IN(SELECT SAL FROM EMP WHERE ENAME="FORD" OR ENAME="SMITH") ORDER BY ENAME DESC;
+-------+-------+---------+------+------------+------+------+--------+
| EMPNO | ENAME | JOB     | MGR  | HIREDATE   | SAL  | COMM | DEPTNO |
+-------+-------+---------+------+------------+------+------+--------+
|  7369 | SMITH | CLERK   | 7902 | 1989-12-17 |  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 (0.02 sec)
```

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

```
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 | 1989-12-17 |  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 | ADBMS  | 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 | 1982-12-23 | 1300 | NULL |     10 |
+-------+--------+-----------+------+------------+------+------+--------+
10 rows in set (0.00 sec)
```

5)Find the highest paid employee of sales department.

```
mysql> SELECT * FROM EMP WHERE SAL=(SELECT MAX(SAL) FROM EMP);
+-------+-------+-----------+------+------------+------+------+--------+
| EMPNO | ENAME | JOB       | MGR  | HIREDATE   | SAL  | COMM | DEPTNO |
+-------+-------+-----------+------+------------+------+------+--------+
|  7839 | KING  | PRESIDENT | NULL | 1981-11-17 | 5000 | NULL |     10 |
+-------+-------+-----------+------+------------+------+------+--------+
1 row in set (0.01 sec)
```

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

```
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 |
+-------+-------+----------+------+------------+------+------+--------+
|  7499 | ALLEN | SALESMAN | 7968 | 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 |
+-------+-------+----------+------+------------+------+------+--------+
4 rows in set (0.00 sec)
```

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

```
mysql> SELECT * FROM EMP WHERE SAL IN (SELECT MAX(SAL) FROM EMP GROUP BY DEPTNO);
+-------+-------+-----------+------+------------+------+------+--------+
| EMPNO | ENAME | JOB       | MGR  | HIREDATE   | SAL  | COMM | DEPTNO |
+-------+-------+-----------+------+------------+------+------+--------+
|  7698 | BLAKE | MANAGER   | 7839 | 1981-05-01 | 2850 | NULL |     30 |
|  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)
```

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

```
mysql> SELECT * FROM EMP WHERE SAL=(SELECT (MAX(SAL)+MIN(SAL))/2 FROM EMP);
Empty set (0.01 sec)
```

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

```
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.

```
mysql> SELECT * FROM EMP E WHERE HIREDATE <(SELECT HIREDATE FROM EMP WHERE EMPNO=E.MGR);
+-------+-------+----------+------+------------+------+------+--------+
| EMPNO | ENAME | JOB      | MGR  | HIREDATE   | SAL  | COMM | DEPTNO |
+-------+-------+----------+------+------------+------+------+--------+
|  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 |
+-------+-------+----------+------+------------+------+------+--------+
4 rows in set (0.00 sec)
```

EXPERIMENT- 05

TITLE : 5.USE DIFFERENT SQL CLAUSES AND JOIN

OBJECTIVE : TO UNDERSTAND THE UE OF GROUP BY AND HAVING CLAUSE AND EXECUTE THE SQL COMMANDS USING JOIN.

Q1.WRITE THE SQL QUERIES FOR THE FOLLOWING QUERIES (USE EMP_TABLE AND DEPT_TABLE OF EXPERIMENT 4.

1. List the Deptno where there are no emps.

2. List the No.of emp's and Avg salary within each department for each job.

```
mysql> SELECT COUNT(*),AVG(SAL),DEPTNO,JOB FROM EMP GROUP BY DEPTNO,JOB;
+----------+-----------+--------+-----------+
| COUNT(*) | AVG(SAL)  | DEPTNO | JOB       |
+----------+-----------+--------+-----------+
|        2 |  800.0000 |     20 | CLERK     |
|        4 | 1400.0000 |     30 | SALESMAN  |
|        1 | 2975.0000 |     20 | MANAGER   |
|        1 | 2850.0000 |     30 | MANAGER   |
|        1 | 2450.0000 |     10 | MANAGER   |
|        2 | 3000.0000 |     20 | ANALYST   |
|        1 | 5000.0000 |     10 | PRESIDENT |
|        1 |  950.0000 |     30 | CLERK     |
|        1 | 1300.0000 |     10 | CLERK     |
+----------+-----------+--------+-----------+
9 rows in set (0.01 sec)
```

3. Find the maximum average salary drawn for each job except for 'President'.

```
mysql> SELECT MAX(sAL) FROM EMP WHERE SAL IN (SELECT AVG (SAL) FROM EMP WHERE JOB <>"PRESIDENT" GROUP BY JOB);
+----------+
| MAX(sAL) |
+----------+
|     3000 |
+----------+
1 row in set (0.00 sec)
```

4. List the department details where at least two emps are working.

```
mysql> SELECT * FROM DEPT WHERE DEPTNO IN (SELECT DEPTNO FROM EMP GROUP BY DEPTNO HAVING COUNT(*) >=2);
+--------+------------+----------+
| DEPTNO | DNAME      | LOC      |
+--------+------------+----------+
|     10 | ACCOUNTING | NEW YORK |
|     20 | RESEARCH   | DALLAS   |
|     30 | SALES      | CHICAGO  |
+--------+------------+----------+
3 rows in set (0.00 sec)
```

5. List the no. of emps in each department where the no. is more than 3.

```
mysql> SELECT DEPTNO,COUNT(*) AS 'NO.OF EMP' FROM EMP GROUP BY DEPTNO HAVING COUNT(*) >=3;
+--------+-----------+
| DEPTNO | NO.OF EMP |
+--------+-----------+
|     20 |         5 |
|     30 |         6 |
|     10 |         3 |
+--------+-----------+
3 rows in set (0.00 sec)
```

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

```
mysql> SELECT E.DEPTNO,E.ENAME,E.SAL FROM EMP E WHERE E.SAL IN (SELECT MAX(SAL) FROM EMP GROUP BY DEPTNO);
+--------+-------+------+
| DEPTNO | ENAME | SAL  |
+--------+-------+------+
|     30 | BLAKE | 2850 |
|     20 | SCOTT | 3000 |
|     10 | KING  | 5000 |
|     20 | FORD  | 3000 |
+--------+-------+------+
4 rows in set (0.00 sec)
```

7.List the Deptno and their average salaries for dept with the average salary less than the averages for all departments.

```
mysql> SELECT E.DEPTNO,AVG(E.SAL) FROM EMP E GROUP BY E.DEPTNO HAVING AVG(SAL)< (SELECT AVG(SAL) FROM EMP);
+--------+------------+
| DEPTNO | AVG(E.SAL) |
+--------+------------+
|     30 |  1566.6667 |
+--------+------------+
1 row in set (0.00 sec)
```

2. EXECUTE THE EXPERIMENT 4 USING SQL JOIN.

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

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

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

```
mysql> SELECT * FROM EMP RIGHT JOIN DEPT ON EMP.DEPTNO=DEPT.DEPTNO WHERE JOB IN (SELECT JOB FROM EMP WHERE ENAME="ALLEN");
+-------+--------+----------+------+------------+------+------+--------+--------+-------+---------+
| EMPNO | ENAME  | JOB      | MGR  | HIREDATE   | SAL  | COMM | DEPTNO | DEPTNO | DNAME | LOC     |
+-------+--------+----------+------+------------+------+------+--------+--------+-------+---------+
|  7844 | TURNER | SALESMAN | 7698 | 1981-09-08 | 1500 |    0 |     30 |     30 | SALES | CHICAGO |
|  7654 | MARTIN | SALESMAN | 7698 | 1981-09-28 | 1250 | 1400 |     30 |     30 | SALES | CHICAGO |
|  7521 | WARD   | SALESMAN | 7698 | 1981-02-22 | 1250 |  500 |     30 |     30 | SALES | CHICAGO |
|  7499 | ALLEN  | SALESMAN | 7968 | 1981-02-20 | 1600 |  300 |     30 |     30 | SALES | CHICAGO |
+-------+--------+----------+------+------------+------+------+--------+--------+-------+---------+
4 rows in set (0.00 sec)
```

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

```
mysql> SELECT * FROM EMP RIGHT JOIN DEPT ON EMP.DEPTNO=DEPT.DEPTNO WHERE SAL IN (SELECT SAL FROM EMP WHERE ENAME="FORD" OR ENAME="S
MITH") ORDER BY ENAME DESC;
+-------+-------+---------+------+------------+------+------+--------+--------+----------+--------+
| EMPNO | ENAME | JOB     | MGR  | HIREDATE   | SAL  | COMM | DEPTNO | DEPTNO | DNAME    | LOC    |
+-------+-------+---------+------+------------+------+------+--------+--------+----------+--------+
|  7369 | SMITH | CLERK   | 7902 | 1989-12-17 |  500 |  800 |     20 |     20 | RESEARCH | DALLAS |
|  7788 | SCOTT | ANALYST | 7566 | 1982-12-09 | 3000 | NULL |     20 |     20 | RESEARCH | DALLAS |
|  7902 | FORD  | ANALYST | 7566 | 1981-12-03 | 3000 | NULL |     20 |     20 | RESEARCH | DALLAS |
+-------+-------+---------+------+------------+------+------+--------+--------+----------+--------+
3 rows in set (0.00 sec)
```

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

```
mysql> SELECT * FROM EMP RIGHT JOIN DEPT ON EMP.DEPTNO=DEPT.DEPTNO WHERE JOB=(SELECT JOB FROM EMP WHERE ENAME="MILLER") OR SAL> (SE
LECT SAL FROM EMP WHERE ENAME="ALLEN");
+-------+--------+-----------+------+------------+------+------+--------+--------+------------+----------+
| EMPNO | ENAME  | JOB       | MGR  | HIREDATE   | SAL  | COMM | DEPTNO | DEPTNO | DNAME      | LOC      |
+-------+--------+-----------+------+------------+------+------+--------+--------+------------+----------+
|  7369 | SMITH  | CLERK     | 7902 | 1989-12-17 |  500 |  800 |     20 |     20 | RESEARCH   | DALLAS   |
|  7566 | JONES  | MANAGER   | 7839 | 1981-04-02 | 2975 | NULL |     20 |     20 | RESEARCH   | DALLAS   |
|  7698 | BLAKE  | MANAGER   | 7839 | 1981-05-01 | 2850 | NULL |     30 |     30 | SALES      | CHICAGO  |
|  7782 | CLARK  | MANAGER   | 7839 | 1981-06-09 | 2450 | NULL |     10 |     10 | ACCOUNTING | NEW YORK |
|  7788 | SCOTT  | ANALYST   | 7566 | 1982-12-09 | 3000 | NULL |     20 |     20 | RESEARCH   | DALLAS   |
|  7839 | KING   | PRESIDENT | NULL | 1981-11-17 | 5000 | NULL |     10 |     10 | ACCOUNTING | NEW YORK |
|  7876 | ADBMS  | CLERK     | 7788 | 1983-01-12 | 1100 | NULL |     20 |     20 | RESEARCH   | DALLAS   |
|  7900 | JAMES  | CLERK     | 7698 | 1981-12-03 |  950 | NULL |     30 |     30 | SALES      | CHICAGO  |
|  7902 | FORD   | ANALYST   | 7566 | 1981-12-03 | 3000 | NULL |     20 |     20 | RESEARCH   | DALLAS   |
|  7934 | MILLER | CLERK     | 7782 | 1982-12-23 | 1300 | NULL |     10 |     10 | ACCOUNTING | NEW YORK |
+-------+--------+-----------+------+------------+------+------+--------+--------+------------+----------+
10 rows in set (0.00 sec)
```

5)Find the highest paid employee of sales department.

```
mysql> SELECT * FROM EMP RIGHT JOIN DEPT ON EMP.DEPTNO =DEPT.DEPTNO WHERE SAL =(SELECT MAX(SAL) FROM EMP);
+-------+-------+-----------+------+------------+------+------+--------+--------+------------+----------+
| EMPNO | ENAME | JOB       | MGR  | HIREDATE   | SAL  | COMM | DEPTNO | DEPTNO | DNAME      | LOC      |
+-------+-------+-----------+------+------------+------+------+--------+--------+------------+----------+
|  7839 | KING  | PRESIDENT | NULL | 1981-11-17 | 5000 | NULL |     10 |     10 | ACCOUNTING | NEW YORK |
+-------+-------+-----------+------+------------+------+------+--------+--------+------------+----------+
1 row in set (0.00 sec)
```

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

```
mysql> SELECT * FROM EMP RIGHT JOIN DEPT ON EMP.DEPTNO=DEPT.DEPTNO WHERE HIREDATE < (SELECT MAX(HIREDATE) FROM EMP WHERE MGR=(SELEC
T EMPNO FROM EMP WHERE ENAME="KING"));
+-------+-------+----------+------+------------+------+------+--------+--------+----------+---------+
| EMPNO | ENAME | JOB      | MGR  | HIREDATE   | SAL  | COMM | DEPTNO | DEPTNO | DNAME    | LOC     |
+-------+-------+----------+------+------------+------+------+--------+--------+----------+---------+
|  7499 | ALLEN | SALESMAN | 7968 | 1981-02-20 | 1600 |  300 |     30 |     30 | SALES    | CHICAGO |
|  7521 | WARD  | SALESMAN | 7698 | 1981-02-22 | 1250 |  500 |     30 |     30 | SALES    | CHICAGO |
|  7566 | JONES | MANAGER  | 7839 | 1981-04-02 | 2975 | NULL |     20 |     20 | RESEARCH | DALLAS  |
|  7698 | BLAKE | MANAGER  | 7839 | 1981-05-01 | 2850 | NULL |     30 |     30 | SALES    | CHICAGO |
+-------+-------+----------+------+------------+------+------+--------+--------+----------+---------+
4 rows in set (0.00 sec)
```

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

```
mysql> SELECT * FROM EMP RIGHT JOIN DEPT ON EMP.DEPTNO=DEPT.DEPTNO WHERE SAL IN (SELECT MAX(SAL) FROM EMP GROUP BY DEPTNO);
+-------+-------+-----------+------+------------+------+------+--------+--------+------------+----------+
| EMPNO | ENAME | JOB       | MGR  | HIREDATE   | SAL  | COMM | DEPTNO | DEPTNO | DNAME      | LOC      |
+-------+-------+-----------+------+------------+------+------+--------+--------+------------+----------+
|  7698 | BLAKE | MANAGER   | 7839 | 1981-05-01 | 2850 | NULL |     30 |     30 | SALES      | CHICAGO  |
|  7788 | SCOTT | ANALYST   | 7566 | 1982-12-09 | 3000 | NULL |     20 |     20 | RESEARCH   | DALLAS   |
|  7839 | KING  | PRESIDENT | NULL | 1981-11-17 | 5000 | NULL |     10 |     10 | ACCOUNTING | NEW YORK |
|  7902 | FORD  | ANALYST   | 7566 | 1981-12-03 | 3000 | NULL |     20 |     20 | RESEARCH   | DALLAS   |
+-------+-------+-----------+------+------------+------+------+--------+--------+------------+----------+
4 rows in set (0.00 sec)
```

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

```
mysql> SELECT * FROM EMP RIGHT JOIN DEPT ON EMP.DEPTNO=DEPT.DEPTNO 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

```
mysql> SELECT * FROM EMP E RIGHT JOIN DEPT ON E.DEPTNO =DEPT.DEPTNO WHERE HIREDATE IN(SELECT HIREDATE FROM EMP WHERE E.EMPNO<> EMPN
O);
+-------+-------+---------+------+------------+------+------+--------+--------+----------+---------+
| EMPNO | ENAME | JOB     | MGR  | HIREDATE   | SAL  | COMM | DEPTNO | DEPTNO | DNAME    | LOC     |
+-------+-------+---------+------+------------+------+------+--------+--------+----------+---------+
|  7900 | JAMES | CLERK   | 7698 | 1981-12-03 |  950 | NULL |     30 |     30 | SALES    | CHICAGO |
|  7902 | FORD  | ANALYST | 7566 | 1981-12-03 | 3000 | NULL |     20 |     20 | RESEARCH | DALLAS  |
+-------+-------+---------+------+------------+------+------+--------+--------+----------+---------+
2 rows in set (0.00 sec)
```

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

```
mysql> SELECT * FROM EMP E RIGHT JOIN DEPT ON E.DEPTNO=DEPT.DEPTNO WHERE HIREDATE <( SELECT HIREDATE FROM EMP WHERE EMPNO=E.MGR);
+-------+-------+----------+------+------------+------+------+--------+--------+------------+----------+
| EMPNO | ENAME | JOB      | MGR  | HIREDATE   | SAL  | COMM | DEPTNO | DEPTNO | DNAME      | LOC      |
+-------+-------+----------+------+------------+------+------+--------+--------+------------+----------+
|  7521 | WARD  | SALESMAN | 7698 | 1981-02-22 | 1250 |  500 |     30 |     30 | SALES      | CHICAGO  |
|  7566 | JONES | MANAGER  | 7839 | 1981-04-02 | 2975 | NULL |     20 |     20 | RESEARCH   | DALLAS   |
|  7698 | BLAKE | MANAGER  | 7839 | 1981-05-01 | 2850 | NULL |     30 |     30 | SALES      | CHICAGO  |
|  7782 | CLARK | MANAGER  | 7839 | 1981-06-09 | 2450 | NULL |     10 |     10 | ACCOUNTING | NEW YORK |
+-------+-------+----------+------+------------+------+------+--------+--------+------------+----------+
4 rows in set (0.00 sec)
```

EXPERIMENT-06

TITLE:6.TO UNDERSTAND THE CONCEPTS OF VIEWS.

OBJECTIVE: STUDENTS WILL BE ABLE TO IMPLEMENT THE CONCEPTS OF VIEWS.

Q1)CREATE TABLE OF TABLE NAME: EMPLOYEES AND ADD 6 ROWS.

COLUMN NAME	DATA TYPE	WIDTH	ATTRIBUTES
EMPLOYEE_ID	CHARACTER	10	PK
FIRST_NAME	CHARACTER	30	NN
LAST_NAME	CHARACTER	30	NN
DOB	DATE		
SALARY	NUMBER	25	NN
DEPARTMENT_ID	CHARACTER	10	

```
mysql> CREATE TABLE EMPLOYEES(EMPLOYEE_ID VARCHAR(10) PRIMARY KEY,FIRST_NAME VARCHAR(30) NOT NULL,LAST_NAME VARCHAR(30) NOT NULL,DO
B DATE,SALARY NUMERIC(25) NOT NULL,DEPARTMENT_ID VARCHAR(10));
Query OK, 0 rows affected (0.05 sec)
```

```
mysql> INSERT INTO EMPLOYEES VALUES("E0001","JOHN","SMITH","2001-01-01",25000,"D0001");
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO EMPLOYEES VALUES("E0002","BARRY","ALLEN","1980-12-05",25000,"D0002");
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO EMPLOYEES VALUES("E0003","CLARK","KENT","1990-09-24",25000,"D0003");
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO EMPLOYEES VALUES("E0004","JAMES","MILLER","1990-11-15",25000,"D0004");
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM EMPLOYEES;
+-------------+------------+-----------+------------+--------+---------------+
| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | DOB        | SALARY | DEPARTMENT_ID |
+-------------+------------+-----------+------------+--------+---------------+
| E0001       | JOHN       | SMITH     | 2001-01-01 |  25000 | D0001         |
| E0002       | BARRY      | ALLEN     | 1980-12-05 |  25000 | D0002         |
| E0003       | CLARK      | KENT      | 1990-09-24 |  25000 | D0003         |
| E0004       | JAMES      | MILLER    | 1990-11-15 |  25000 | D0004         |
+-------------+------------+-----------+------------+--------+---------------+
4 rows in set (0.00 sec)
```

1.EXECUTE THE FOLLOWING VIEW RELATED QUERIES:

1) Create View of name emp_view and the column would be Employee_id, Last_Name, salary

and department_id only.:

```
mysql> CREATE VIEW EMP_VIEW AS SELECT EMPLOYEE_ID,LAST_NAME,SALARY,DEPARTMENT_ID FROM EMPLOYEES;
Query OK, 0 rows affected (0.02 sec)

mysql> SELECT * FROM EMP_VIEW;
+-------------+-----------+--------+---------------+
| EMPLOYEE_ID | LAST_NAME | SALARY | DEPARTMENT_ID |
+-------------+-----------+--------+---------------+
| E0001       | SMITH     |  25000 | D0001         |
| E0002       | ALLEN     |  25000 | D0002         |
| E0003       | KENT      |  25000 | D0003         |
| E0004       | MILLER    |  25000 | D0004         |
+-------------+-----------+--------+---------------+
4 rows in set (0.01 sec)
```

2) Insert values into view(remove the NOT NULL constraint and then insert values):

```
mysql> ALTER TABLE EMPLOYEES MODIFY LAST_NAME VARCHAR(30) NULL;ALTER TABLE EMPLOYEES MODIFY SALARY VARCHAR(30) NULL;
Query OK, 0 rows affected (0.10 sec)
Records: 0  Duplicates: 0  Warnings: 0

Query OK, 4 rows affected (0.05 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> DESC EMP_VIEW;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| EMPLOYEE_ID   | varchar(10) | NO   |     | NULL    |       |
| LAST_NAME     | varchar(30) | YES  |     | NULL    |       |
| SALARY        | varchar(30) | YES  |     | NULL    |       |
| DEPARTMENT_ID | varchar(10) | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
4 rows in set (0.04 sec)
```

```
mysql> INSERT INTO EMPLOYEES VALUES("E0005","PETER","PARKER","1970-08-21",NULL,"D0001");INSERT INTO EMPLOYEES VALUES("E0006","HOLDE
N",NULL,"1970--08-21",35000,"D0001");
Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected, 1 warning (0.00 sec)

mysql> SELECT * FROM EMP_VIEW;
+-------------+-----------+--------+---------------+
| EMPLOYEE_ID | LAST_NAME | SALARY | DEPARTMENT_ID |
+-------------+-----------+--------+---------------+
| E0001       | SMITH     | 25000  | D0001         |
| E0002       | ALLEN     | 25000  | D0002         |
| E0003       | KENT      | 25000  | D0003         |
| E0004       | MILLER    | 25000  | D0004         |
| E0005       | PARKER    | NULL   | D0001         |
| E0006       | NULL      | 35000  | D0001         |
+-------------+-----------+--------+---------------+
6 rows in set (0.00 sec)
```

3) Modify, delete and drop operations are performed on view.:

```
mysql> UPDATE EMP_VIEW SET DEPARTMENT_ID="D0020" WHERE EMPLOYEE_ID="E0004";
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> DELETE FROM EMP_VIEW WHERE LAST_NAME="PARKER";
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM EMP_VIEW;
+-------------+-----------+--------+---------------+
| EMPLOYEE_ID | LAST_NAME | SALARY | DEPARTMENT_ID |
+-------------+-----------+--------+---------------+
| E0001       | SMITH     | 25000  | D0001         |
| E0002       | ALLEN     | 25000  | D0002         |
| E0003       | KENT      | 25000  | D0003         |
| E0004       | MILLER    | 25000  | D0020         |
| E0006       | NULL      | 35000  | D0001         |
+-------------+-----------+--------+---------------+
5 rows in set (0.00 sec)

mysql> DROP VIEW EMP_VIEW2;
ERROR 1051 (42S02): Unknown table 'adbms1.emp_view2'
mysql> DROP VIEW EMP_VIEW;
Query OK, 0 rows affected (0.01 sec)
```

4) Creates a view named salary_view. The view shows the employees in department 20 and their annual salary.

```
mysql> CREATE VIEW SALARY_VIEW AS SELECT EMPLOYEE_ID,FIRST_NAME,LAST_NAME,SALARY * 12,DEPARTMENT_ID FROM EMPLOYEES WHERE DEPARTMENT
_ID="D0020";
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> SELECT * FROM SALARY_VIEW;
+-------------+------------+-----------+-------------+---------------+
| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | SALARY * 12 | DEPARTMENT_ID |
+-------------+------------+-----------+-------------+---------------+
| E0004       | JAMES      | MILLER    |      300000 | D0020         |
+-------------+------------+-----------+-------------+---------------+
1 row in set (0.00 sec)
```

EXPERIMENT 7

Title-To understand concept of INDEX

Create an index of name employee_idx on EMPLOYEES with column Last_Name, Department_id

```
mysql> CREATE INDEX EMPLOYEE_IDX on EMPLOYEE(LAST_NAME,DEPARTMENT_ID);
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Find the ROWID for the above table and create a unique index on employee_id column of the EMPLOYEES

```
mysql> CREATE UNIQUE INDEX ROWID ON EMPLOYEE(EMPLOYEE_ID);
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Create a reverse index on employee_id column of the EMPLOYEES

```
mysql> CREATE INDEX EMPLOYEE_ID_DESC_IDX ON EMPLOYEE(EMPLOYEE_ID DESC);
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Create a unique and composite index on employee_id and check whether there is duplicity of tuples or not

```
mysql> SELECT EMPLOYEE_ID, COUNT(*) FROM EMPLOYEE GROUP BY EMPLOYEE_ID HAVING COUNT(*)>1;
Empty set (0.01 sec)
```

Create Function-based indexes defined on the SQL functions UPPER(column_name) or LOWER(column_name) to facilitate case-insensitive searches(on column Last_Name)

```
mysql> CREATE INDEX EMPLOYEE_LAST_NAME_I ON EMPLOYEE(LAST_NAME);
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Drop the function based index on column Last_Name

```
mysql> DROP INDEX EMPLOYEE_LAST_NAME_I ON EMPLOYEE;
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0
```