Advance Database Management Systems
Lab File

Name – Mohak Bajaj

SAP ID – 500093079

Roll No. – R2142210493

Submitted to – Ms. Kalpana Rangra

GitHub Repo: GitHub Link

Experiment – 1: To understand DDL and DML commands

Objective: To understand the concept of designing issue related to the database with creating, populating the tables. Also familiarize students with different ways of manipulation in database.

Queries: [Github]

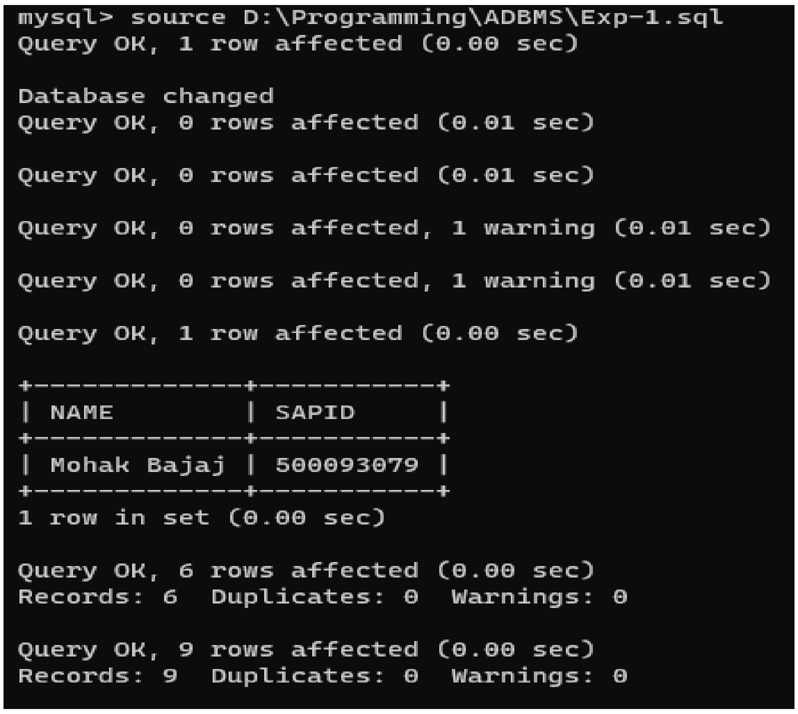
```sql
-- 1 Creation of database and tables
-- create database exp-1
CREATE DATABASE IF NOT EXISTS exp_1;
USE exp_1;
-- create table CLIENT_MASTER
CREATE TABLE IF NOT EXISTS CLIENT_MASTER (
    CLIENTNO VARCHAR(6),
    NAME VARCHAR(20),
    CITY VARCHAR(15),
    PINCODE INT,
    STATE VARCHAR(15),
    BALDUE DECIMAL(10, 2)
);
-- create table PRODUCT_MASTER
CREATE TABLE IF NOT EXISTS PRODUCT_MASTER (
    PRODUCTNO VARCHAR(6),
    DESCRIPTION VARCHAR(15),
    PROFITPERCENT DECIMAL(4, 2),
    UNIT_MEASURE VARCHAR(10),
    QTYONHAND INT,
    REORDERL_VL INT,
    SELLPRICE DECIMAL(8, 2),
    COSTPRICE DECIMAL(8, 2)
);
-- create table SALESMAN_MASTER
CREATE TABLE IF NOT EXISTS SALESMAN_MASTER (
    SALESMANNO VARCHAR(6),
    SALESMANNAME VARCHAR(20),
    ADDRESS_1 VARCHAR(30),
    ADDRESS_2 VARCHAR(30),
    CITY VARCHAR(20),
    PINCODE INT,
    STATE VARCHAR(20),
    SALAMT REAL,
    TGTTOGET DECIMAL,
    YTDSALES DOUBLE(6, 2),
    REMARKS VARCHAR(60)
);
-- Misc Author Record - Mohak Bajaj
CREATE TABLE IF NOT EXISTS AUTHOR (NAME VARCHAR(20), SAPID INT(9));
INSERT INTO AUTHOR
VALUES ('Mohak Bajaj', 500093079);
SELECT *
FROM AUTHOR;
-- 2 Inserting Data into tables
-- insert data into CLIENT_MASTER
INSERT INTO CLIENT_MASTER
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
    );
-- insert data into PRODUCT_MASTER
INSERT INTO PRODUCT_MASTER
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',
        100,
        30,
        450,
        300
    );
-- insert data into SALESMAN_MASTER
INSERT INTO SALESMAN_MASTER
VALUES (
        'S00001',
        'Aman',
        'A/14',
        'Worli',
        'Mumbai',
        400002,
        'Maharashtra',
        3000,
        50000,
        0,
        'Good'
    ),
    (
        'S00002',
        'Omkar',
        '65',
        'Nariman',
        'Mumbai',
        400001,
        'Maharashtra',
```

```sql
        3500,
        50000,
        0,
        'Good'
    ),
    (
        'S00003',
        'Raj',
        'P-7',
        'Bandra',
        'Mumbai',
        400032,
        'Maharashtra',
        3000,
        50000,
        0,
        'Good'
    ),
    (
        'S00004',
        'Ashish',
        'A/5',
        'Juhu',
        'Mumbai',
        400044,
        'Maharashtra',
        3500,
        50000,
        0,
        'Good'
    );
-- 3 Data Retrival
-- a. Find out the names of all the clients.
SELECT NAME
FROM CLIENT_MASTER;
-- b. Retrieve the entire contents of the Client_Master table.
SELECT *
FROM CLIENT_MASTER;
-- c. Retrieve the list of names, city and the state of all the
clients.
```

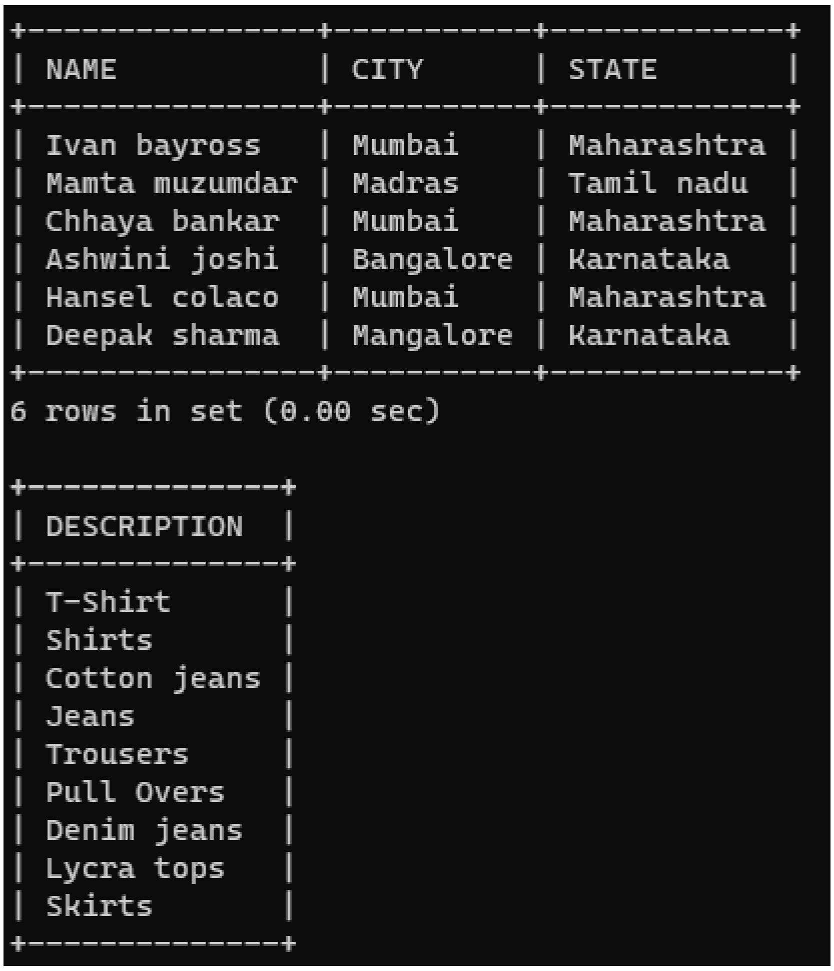
```sql
SELECT NAME,
    CITY,
    STATE
FROM CLIENT_MASTER;
-- d. List the various products available from the Product_Master
table.
SELECT DESCRIPTION
FROM PRODUCT_MASTER;
-- e. List all the clients who are located in Mumbai.
SELECT *
FROM CLIENT_MASTER
WHERE CITY = 'Mumbai';
-- f. Find the names of salesman who have a salary equal to Rs.3000.
SELECT SALESMANNAME
FROM SALESMAN_MASTER
WHERE SALAMT = 3000;
-- 4 Data Updatation
-- a. Change the city of ClientNo ‘C00005’ to ‘Bangalore’.
UPDATE CLIENT_MASTER
SET CITY = 'Bangalore'
WHERE CLIENTNO = 'C00005';
-- b. Change the BalDue of ClientNo ‘C00001’ to Rs.1000.
UPDATE CLIENT_MASTER
SET BALDUE = 1000
WHERE CLIENTNO = 'C00001';
-- c. Change the cost price of ‘Trousers’ to rs.950.00.
UPDATE PRODUCT_MASTER
SET COSTPRICE = 950
WHERE DESCRIPTION = 'Trousers';
-- d. Change the city of the salesman to Pune.
UPDATE SALESMAN_MASTER
SET CITY = 'Pune';
-- 5 Data Deletion
-- a. Delete all salesman from the Salesman_Master whose salaries are
equal to Rs.3500.
DELETE FROM SALESMAN_MASTER
WHERE SALAMT = 3500;
-- b. Delete all products from Product_Master where the quantity on
hand is equal to 100.
DELETE FROM PRODUCT_MASTER
```

```sql
WHERE QTYONHAND = 100;
-- c. Delete from Client_Master where the column state holds the value
‘Tamil Nadu’.
DELETE FROM CLIENT_MASTER
WHERE STATE = 'Tamil Nadu';
-- 6 Data Alteration
-- a. Add a column called ‘Telephone’ of data type integer to the
Client_Master table.
ALTER TABLE CLIENT_MASTER
ADD TELEPHONE INT;
-- b. Change the size off SellPrice column in Product _Master to 10,
2.
ALTER TABLE PRODUCT_MASTER
MODIFY SELLPRICE DECIMAL(10, 2);
-- 7 Deletion on Table Structure with its Data
-- a. Destroy the table Client_Master along with its data.
DROP TABLE CLIENT_MASTER;
-- 8 Rename Table
-- a. Change the name of the Salesman_Master to sman_mast.
RENAME TABLE SALESMAN_MASTER TO sman_mast;
```

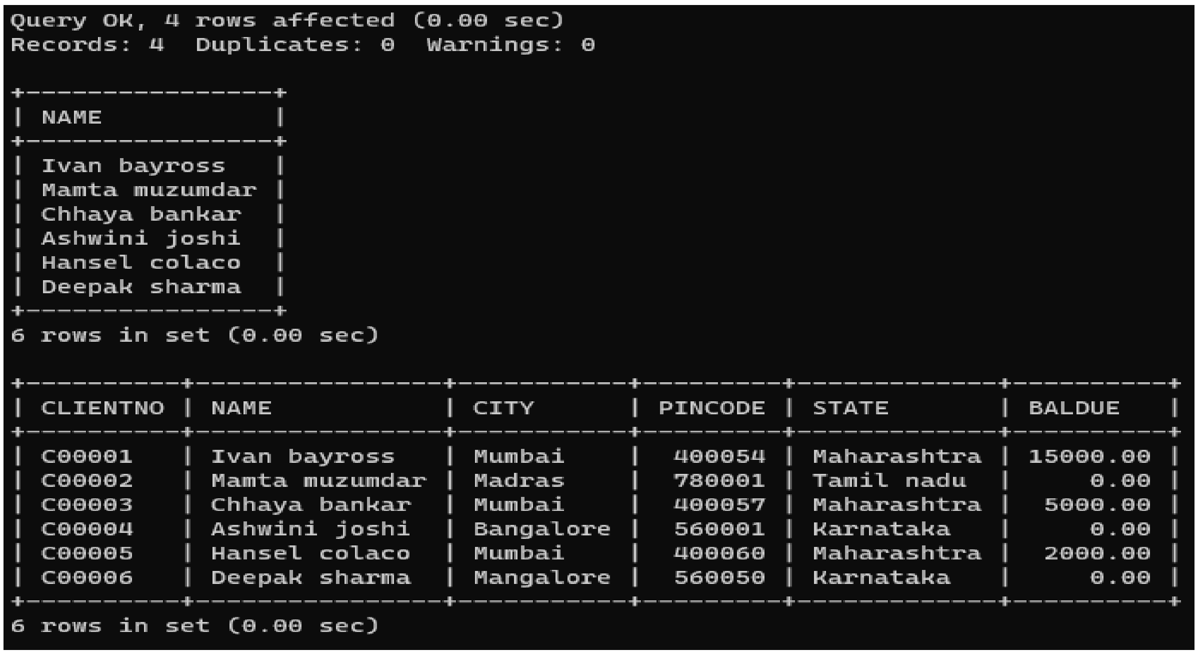
Output:



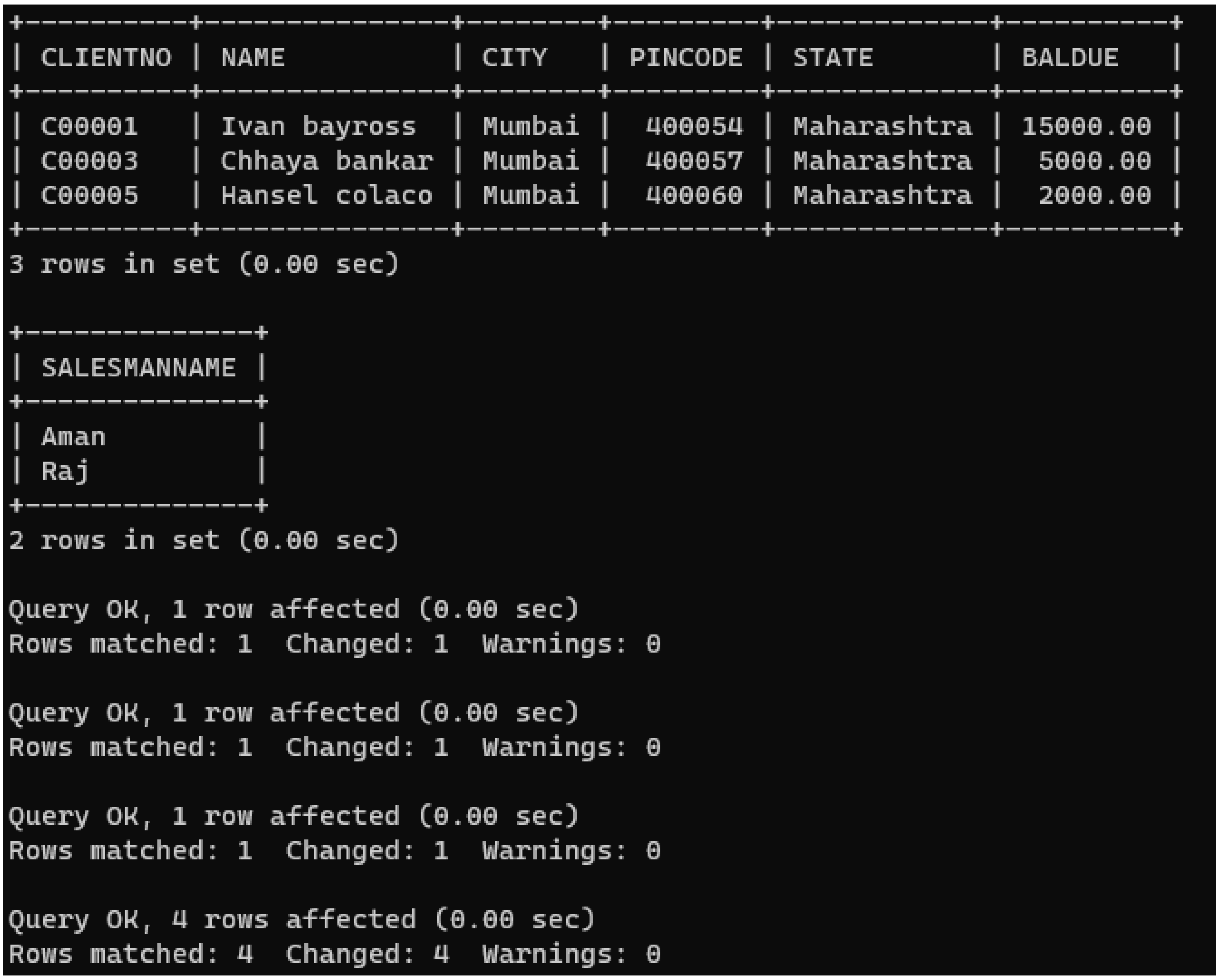
```
mysql> source D:\Programming\ADBMS\Exp-1.sql
Query OK, 1 row affected (0.00 sec)

Database changed
Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected, 1 warning (0.01 sec)

Query OK, 0 rows affected, 1 warning (0.01 sec)

Query OK, 1 row affected (0.00 sec)

+-------------+-----------+
| NAME        | SAPID     |
+-------------+-----------+
| Mohak Bajaj | 500093079 |
+-------------+-----------+
1 row in set (0.00 sec)

Query OK, 6 rows affected (0.00 sec)
Records: 6  Duplicates: 0  Warnings: 0

Query OK, 9 rows affected (0.00 sec)
Records: 9  Duplicates: 0  Warnings: 0
```



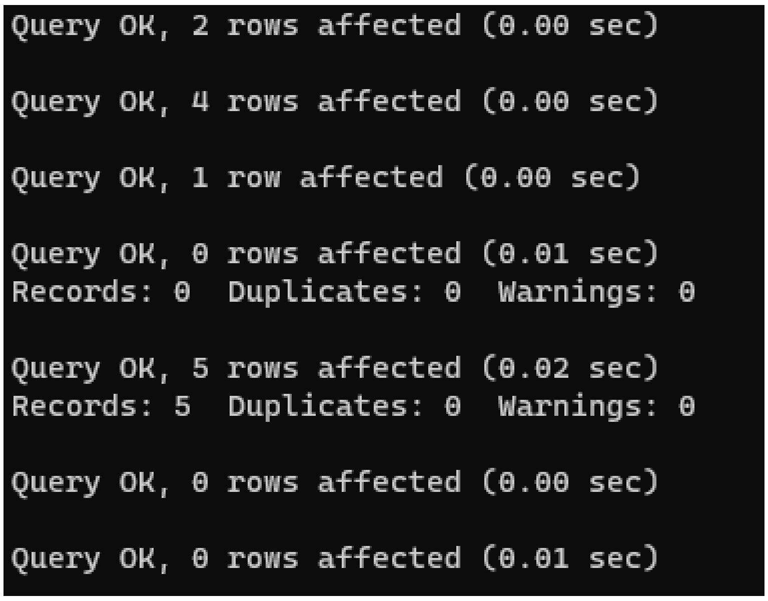
```
+----------------+-----------+-------------+
| NAME           | CITY      | STATE       |
+----------------+-----------+-------------+
| Ivan bayross   | Mumbai    | Maharashtra |
| Mamta muzumdar | Madras    | Tamil nadu  |
| Chhaya bankar  | Mumbai    | Maharashtra |
| Ashwini joshi  | Bangalore | Karnataka   |
| Hansel colaco  | Mumbai    | Maharashtra |
| Deepak sharma  | Mangalore | Karnataka   |
+----------------+-----------+-------------+
6 rows in set (0.00 sec)

+--------------+
| DESCRIPTION  |
+--------------+
| T-Shirt      |
| Shirts       |
| Cotton jeans |
| Jeans        |
| Trousers     |
| Pull Overs   |
| Denim jeans  |
| Lycra tops   |
| Skirts       |
+--------------+
```



```
Query OK, 4 rows affected (0.00 sec)
Records: 4  Duplicates: 0  Warnings: 0

+----------------+
| NAME           |
+----------------+
| Ivan bayross   |
| Mamta muzumdar |
| Chhaya bankar  |
| Ashwini joshi  |
| Hansel colaco  |
| Deepak sharma  |
+----------------+
6 rows in set (0.00 sec)

+----------+----------------+-----------+---------+-------------+----------+
| CLIENTNO | NAME           | CITY      | PINCODE | STATE       | BALDUE   |
+----------+----------------+-----------+---------+-------------+----------+
| C00001   | Ivan bayross   | Mumbai    |  400054 | Maharashtra | 15000.00 |
| C00002   | Mamta muzumdar | Madras    |  780001 | Tamil nadu  |     0.00 |
| C00003   | Chhaya bankar  | Mumbai    |  400057 | Maharashtra |  5000.00 |
| C00004   | Ashwini joshi  | Bangalore |  560001 | Karnataka   |     0.00 |
| C00005   | Hansel colaco  | Mumbai    |  400060 | Maharashtra |  2000.00 |
| C00006   | Deepak sharma  | Mangalore |  560050 | Karnataka   |     0.00 |
+----------+----------------+-----------+---------+-------------+----------+
6 rows in set (0.00 sec)
```



```
+----------+---------------+--------+---------+-------------+----------+
| CLIENTNO | NAME          | CITY   | PINCODE | STATE       | BALDUE   |
+----------+---------------+--------+---------+-------------+----------+
| C00001   | Ivan bayross  | Mumbai |  400054 | Maharashtra | 15000.00 |
| C00003   | Chhaya bankar | Mumbai |  400057 | Maharashtra |  5000.00 |
| C00005   | Hansel colaco | Mumbai |  400060 | Maharashtra |  2000.00 |
+----------+---------------+--------+---------+-------------+----------+
3 rows in set (0.00 sec)

+--------------+
| SALESMANNAME |
+--------------+
| Aman         |
| Raj          |
+--------------+
2 rows in set (0.00 sec)

Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

Query OK, 4 rows affected (0.00 sec)
Rows matched: 4  Changed: 4  Warnings: 0
```



```
Query OK, 2 rows affected (0.00 sec)

Query OK, 4 rows affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0

Query OK, 5 rows affected (0.02 sec)
Records: 5  Duplicates: 0  Warnings: 0

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
```

Experiment – 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 Foreign Key.

Queries: Github

```sql
-- 1. Creation of database and tables
-- create database exp-2
CREATE DATABASE IF NOT EXISTS exp_2;
USE exp_2;
-- Create the tables described below:
-- Client Master 1
CREATE TABLE IF NOT EXISTS CLIENT_MASTER_1 (
    CLIENTNO VARCHAR(6) CHECK (CLIENTNO LIKE 'C%'),
    NAME VARCHAR(20) NOT NULL,
    ADDRESS_1 VARCHAR(30) NOT NULL,
    ADDRESS_2 VARCHAR(30) NOT NULL,
    CITY VARCHAR(15) NOT NULL,
    PINCODE INT NOT NULL,
    STATE VARCHAR(15) NOT NULL,
    BALDUE DECIMAL(10, 2) NOT NULL,
    PRIMARY KEY (CLIENTNO)
);
-- Product Master 1
CREATE TABLE IF NOT EXISTS PRODUCT_MASTER_1 (
    PRODUCTNO VARCHAR(6) CHECK (PRODUCTNO LIKE 'P%'),
    DESCRIPTION VARCHAR(15) NOT NULL,
    PROFITPERCENT DECIMAL(4, 2) NOT NULL,
    UNIT_MEASURE VARCHAR(10) NOT NULL,
    QTYONHAND INT NOT NULL,
    REORDERL_VL INT NOT NULL,
    SELLPRICE DECIMAL(8, 2) NOT NULL,
    COSTPRICE DECIMAL(8, 2) NOT NULL,
    PRIMARY KEY (PRODUCTNO)
);
-- Salesman Master 1
CREATE TABLE IF NOT EXISTS SALESMAN_MASTER_1 (
    SALESMANNO VARCHAR(6) CHECK (SALESMANNO LIKE 'S%'),
    SALESMANNAME VARCHAR(20) NOT NULL,
    ADDRESS_1 VARCHAR(30) NOT NULL,
    ADDRESS_2 VARCHAR(30),
    CITY VARCHAR(20),
    PINCODE INT,
    STATE VARCHAR(20),
    SALAMT REAL NOT NULL CHECK (SALAMT > 0),
    TGTTOGET DECIMAL NOT NULL CHECK (TGTTOGET > 0),
    YTDSALES DOUBLE(6, 2) NOT NULL,
    REMARKS VARCHAR(60),
    PRIMARY KEY (SALESMANNO)
);
-- Misc Author Record - Mohak Bajaj
CREATE TABLE IF NOT EXISTS CREATOR (NAME VARCHAR(20), SAPID INT(9));
INSERT INTO CREATOR
VALUES ('Mohak Bajaj', 500093079);
SELECT *
FROM AUTHOR;
-- 2. populate the tables with data with random data
-- Client Master 1
INSERT INTO CLIENT_MASTER_1
VALUES (
        'C00001',
        'Mohak Bajaj',
        'Bangalore',
        '560001',
        'Karnataka',
        560001,
        'Karnataka',
        25000
    ),
    (
        'C00002',
        'Mohak Bajaj',
        'Bangalore',
        '560001',
        'Karnataka',
        560001,
        'Karnataka',
        25000
    ),
    (
        'C00003',
```

```sql
        'Mohak Bajaj',
        'Bangalore',
        '560001',
        'Karnataka',
        560001,
        'Karnataka',
        25000
    ),
    (
        'C00004',
        'Mohak Bajaj',
        'Bangalore',
        '560001',
        'Karnataka',
        560001,
        'Karnataka',
        25000
    ),
    (
        'C00005',
        'Mohak Bajaj',
        'Bangalore',
        '560001',
        'Karnataka',
        560001,
        'Karnataka',
        25000
    ),
    (
        'C00006',
        'Mohak Bajaj',
        'Bangalore',
        '560001',
        'Karnataka',
        560001,
        'Karnataka',
        25000
    );
-- Product Master 1
INSERT INTO PRODUCT_MASTER_1
```

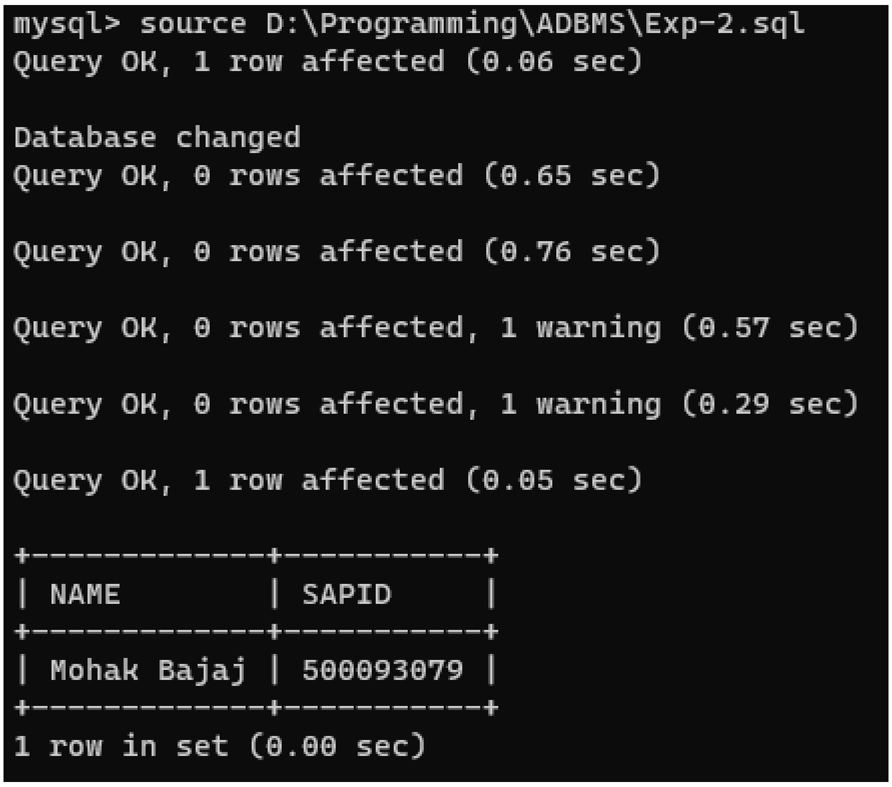
```
VALUES (
        'P00001',
        'Laptop',
        10,
        'Pcs',
        100,
        10,
        10000,
        9000
    ),
    (
        'P00002',
        'Mobile',
        10,
        'Pcs',
        100,
        10,
        10000,
        9000
    ),
    (
        'P00003',
        'Tablet',
        10,
        'Pcs',
        100,
        10,
        10000,
        9000
    ),
    (
        'P00004',
        'Laptop',
        10,
        'Pcs',
        100,
        10,
        10000,
        9000
    ),
```

```sql
    (
        'P00005',
        'Laptop',
        10,
        'Pcs',
        100,
        10,
        10000,
        9000
    ),
    (
        'P00006',
        'Laptop',
        10,
        'Pcs',
        100,
        10,
        10000,
        9000
    );
-- Salesman Master 1
INSERT INTO SALESMAN_MASTER_1
VALUES (
        'S00001',
        'Aman',
        'A/14',
        'Worli',
        'Mumbai',
        400002,
        'Maharashtra',
        3000,
        50000,
        0,
        'Good'
    ),
    (
        'S00002',
        'Omkar',
        '65',
        'Nariman',
        'Mumbai',
        400001,
        'Maharashtra',
        3500,
        50000,
        0,
        'Good'
    ),
    (
        'S00003',
        'Raj',
        'P-7',
        'Bandra',
        'Mumbai',
        400032,
        'Maharashtra',
        3000,
        50000,
        0,
        'Good'
    ),
    (
        'S00004',
        'Ashish',
        'A/5',
        'Juhu',
        'Mumbai',
        400044,
        'Maharashtra',
        3500,
        50000,
        0,
        'Good'
    );
-- 3. Display the content of each table
SELECT *
FROM CLIENT_MASTER_1;
SELECT *
FROM PRODUCT_MASTER_1;
SELECT *
```

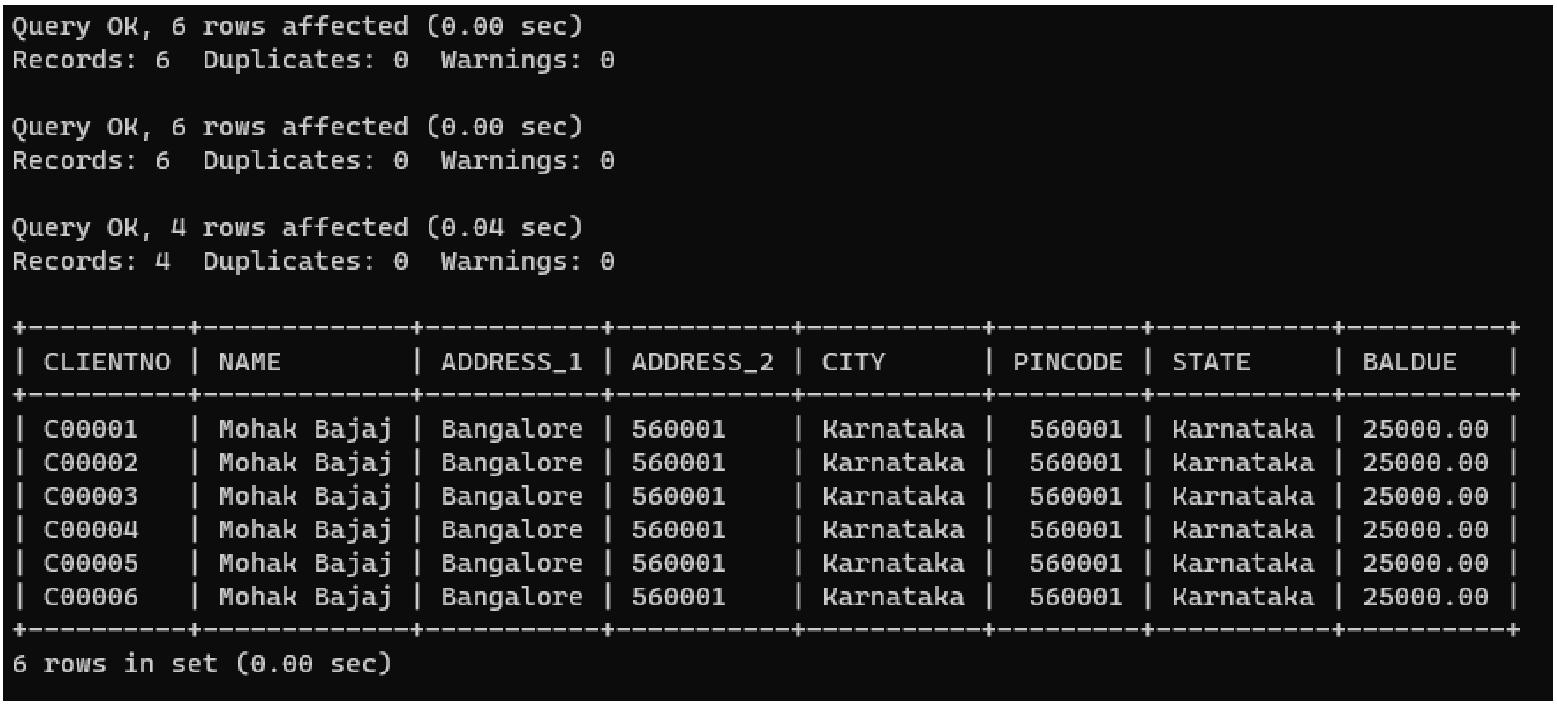
```sql
FROM SALESMAN_MASTER_1;
-- 4.Create table AUTHOR
CREATE TABLE IF NOT EXISTS AUTHOR (
    AUTHOR_ID VARCHAR(5),
    LASTNAME VARCHAR(15) NOT NULL,
    FIRSTNAME VARCHAR(15) NOT NULL,
    EMAIL VARCHAR(40),
    CITY VARCHAR(15),
    COUNTRY VARCHAR(15),
    PRIMARY KEY (AUTHOR_ID)
);
-- 5. Create Table BOOK
CREATE TABLE IF NOT EXISTS BOOK (
    BOOK_ID VARCHAR(5) CHECK (BOOK_ID like 'B%'),
    BOOK_TITLE VARCHAR(15) NOT NULL,
    COPIES INT CHECK (COPIES > 2),
    PRIMARY KEY (BOOK_ID)
);
-- 6. Create table AUTHOR_LIST
CREATE TABLE IF NOT EXISTS AUTHOR_LIST (
    AUTHOR_ID VARCHAR(5),
    BOOK_ID VARCHAR(5),
    ROLE VARCHAR(15),
    PRIMARY KEY (AUTHOR_ID, BOOK_ID),
    FOREIGN KEY (AUTHOR_ID) REFERENCES AUTHOR(AUTHOR_ID),
    FOREIGN KEY (BOOK_ID) REFERENCES BOOK(BOOK_ID)
);
-- 7. Add four records in each tables AUTHOR, BOOK, AUTHOR_LIST.
INSERT INTO AUTHOR
VALUES (
        'A001',
        'Bajaj',
        'Mohak',
        'mb@gmail.com',
        'delhi',
        'india'
    ),
    (
        'A002',
        'Bajaj',
        'Mohak',
        'xyz@gmail.com',
        'delhi',
        'india'
    ),
    (
        'A003',
        'Bajaj',
        'Mohak',
        'abc@gmail.com',
        'delhi',
        'india'
    ),
    (
        'A004',
        'Bajaj',
        'Mohak',
        'qqq@gmauil.com',
        'delhi',
        'india'
    );
INSERT INTO BOOK
VALUES ('B001', 'Book1', 10),
    ('B002', 'Book2', 10),
    ('B003', 'Book3', 10),
    ('B004', 'Book4', 10);
INSERT INTO AUTHOR_LIST
VALUES ('A001', 'B002', 'author'),
    ('A002', 'B003', 'co-author'),
    ('A003', 'B004', 'author'),
    ('A004', 'B001', 'author');
SELECT *
FROM AUTHOR;
SELECT *
FROM BOOK;
SELECT *
FROM AUTHOR_LIST;
-- 8.
Alter  structure  of  table  AUTHOR_LIST  add  the  field  Publisher
data  type  of  30 Character.
```

```sql
ALTER TABLE AUTHOR_LIST
ADD PUBLISHER VARCHAR(30);
```

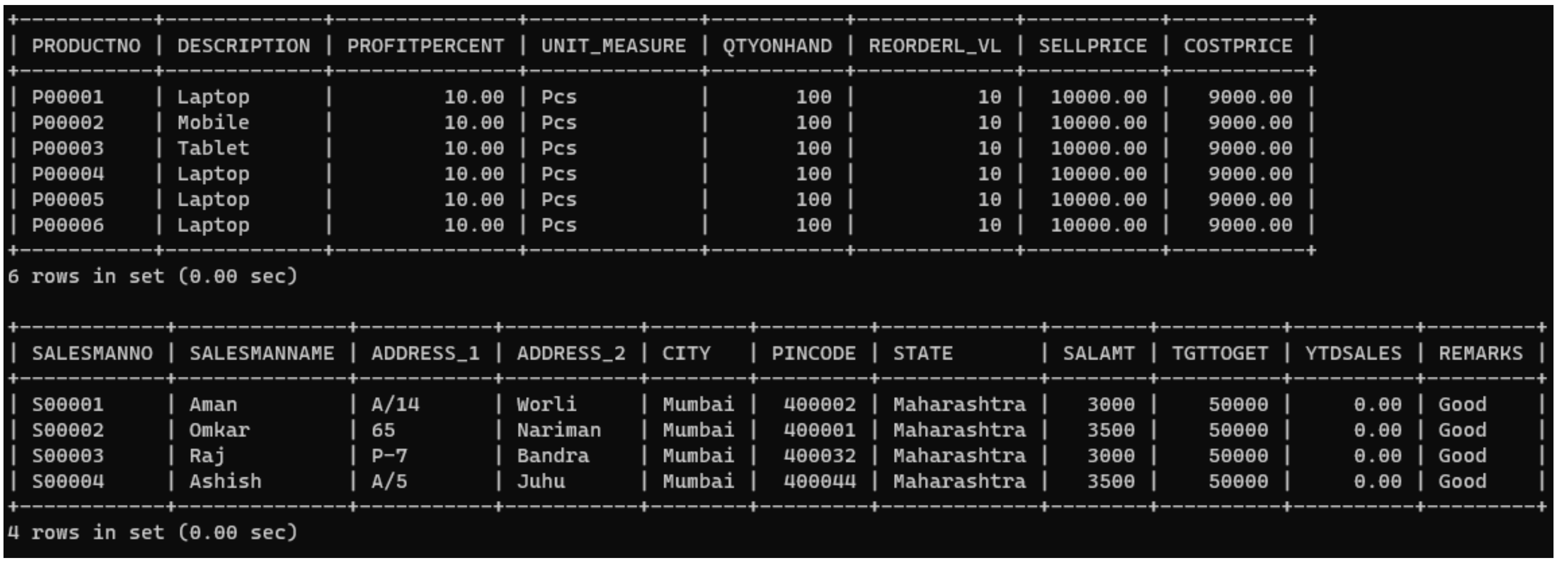
Output:



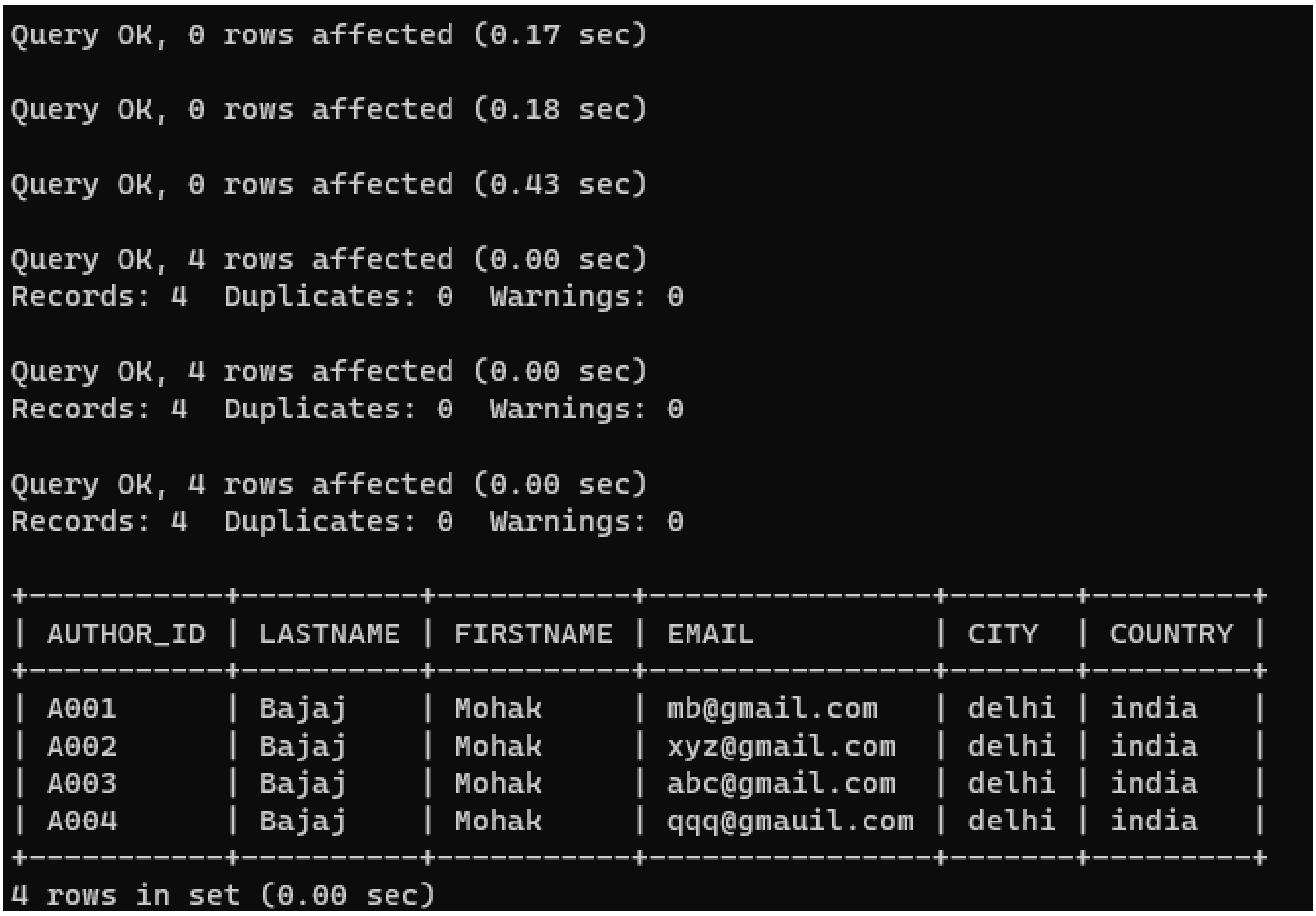
```
mysql> source D:\Programming\ADBMS\Exp-2.sql
Query OK, 1 row affected (0.06 sec)

Database changed
Query OK, 0 rows affected (0.65 sec)

Query OK, 0 rows affected (0.76 sec)

Query OK, 0 rows affected, 1 warning (0.57 sec)

Query OK, 0 rows affected, 1 warning (0.29 sec)

Query OK, 1 row affected (0.05 sec)

+-------------+-----------+
| NAME        | SAPID     |
+-------------+-----------+
| Mohak Bajaj | 500093079 |
+-------------+-----------+
1 row in set (0.00 sec)
```



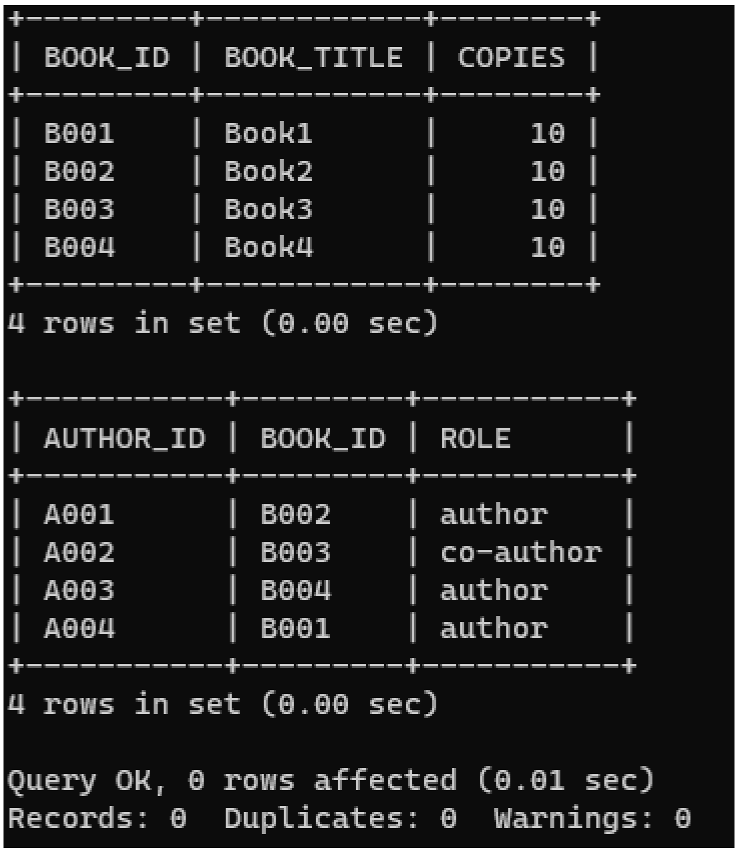
```
Query OK, 6 rows affected (0.00 sec)
Records: 6  Duplicates: 0  Warnings: 0

Query OK, 6 rows affected (0.00 sec)
Records: 6  Duplicates: 0  Warnings: 0

Query OK, 4 rows affected (0.04 sec)
Records: 4  Duplicates: 0  Warnings: 0

+----------+-------------+-----------+-----------+-----------+---------+-----------+----------+
| CLIENTNO | NAME        | ADDRESS_1 | ADDRESS_2 | CITY      | PINCODE | STATE     | BALDUE   |
+----------+-------------+-----------+-----------+-----------+---------+-----------+----------+
| C00001   | Mohak Bajaj | Bangalore | 560001    | Karnataka |  560001 | Karnataka | 25000.00 |
| C00002   | Mohak Bajaj | Bangalore | 560001    | Karnataka |  560001 | Karnataka | 25000.00 |
| C00003   | Mohak Bajaj | Bangalore | 560001    | Karnataka |  560001 | Karnataka | 25000.00 |
| C00004   | Mohak Bajaj | Bangalore | 560001    | Karnataka |  560001 | Karnataka | 25000.00 |
| C00005   | Mohak Bajaj | Bangalore | 560001    | Karnataka |  560001 | Karnataka | 25000.00 |
| C00006   | Mohak Bajaj | Bangalore | 560001    | Karnataka |  560001 | Karnataka | 25000.00 |
+----------+-------------+-----------+-----------+-----------+---------+-----------+----------+
6 rows in set (0.00 sec)
```



```
+-----------+-------------+---------------+--------------+-----------+------------+-----------+-----------+
| PRODUCTNO | DESCRIPTION | PROFITPERCENT | UNIT_MEASURE | QTYONHAND | REORDERL_VL | SELLPRICE | COSTPRICE |
+-----------+-------------+---------------+--------------+-----------+------------+-----------+-----------+
| P00001    | Laptop      |         10.00 | Pcs          |       100 |         10 |  10000.00 |   9000.00 |
| P00002    | Mobile      |         10.00 | Pcs          |       100 |         10 |  10000.00 |   9000.00 |
| P00003    | Tablet      |         10.00 | Pcs          |       100 |         10 |  10000.00 |   9000.00 |
| P00004    | Laptop      |         10.00 | Pcs          |       100 |         10 |  10000.00 |   9000.00 |
| P00005    | Laptop      |         10.00 | Pcs          |       100 |         10 |  10000.00 |   9000.00 |
| P00006    | Laptop      |         10.00 | Pcs          |       100 |         10 |  10000.00 |   9000.00 |
+-----------+-------------+---------------+--------------+-----------+------------+-----------+-----------+
6 rows in set (0.00 sec)

+------------+--------------+-----------+-----------+--------+---------+-------------+--------+----------+----------+---------+
| SALESMANNO | SALESMANNAME | ADDRESS_1 | ADDRESS_2 | CITY   | PINCODE | STATE       | SALAMT | TGTTOGET | YTDSALES | REMARKS |
+------------+--------------+-----------+-----------+--------+---------+-------------+--------+----------+----------+---------+
| S00001     | Aman         | A/14      | Worli     | Mumbai |  400002 | Maharashtra |   3000 |    50000 |     0.00 | Good    |
| S00002     | Omkar        | 65        | Nariman   | Mumbai |  400001 | Maharashtra |   3500 |    50000 |     0.00 | Good    |
| S00003     | Raj          | P-7       | Bandra    | Mumbai |  400032 | Maharashtra |   3000 |    50000 |     0.00 | Good    |
| S00004     | Ashish       | A/5       | Juhu      | Mumbai |  400044 | Maharashtra |   3500 |    50000 |     0.00 | Good    |
+------------+--------------+-----------+-----------+--------+---------+-------------+--------+----------+----------+---------+
4 rows in set (0.00 sec)
```



```
Query OK, 0 rows affected (0.17 sec)

Query OK, 0 rows affected (0.18 sec)

Query OK, 0 rows affected (0.43 sec)

Query OK, 4 rows affected (0.00 sec)
Records: 4  Duplicates: 0  Warnings: 0

Query OK, 4 rows affected (0.00 sec)
Records: 4  Duplicates: 0  Warnings: 0

Query OK, 4 rows affected (0.00 sec)
Records: 4  Duplicates: 0  Warnings: 0

+-----------+----------+-----------+-----------------+-------+---------+
| AUTHOR_ID | LASTNAME | FIRSTNAME | EMAIL           | CITY  | COUNTRY |
+-----------+----------+-----------+-----------------+-------+---------+
| A001      | Bajaj    | Mohak     | mb@gmail.com    | delhi | india   |
| A002      | Bajaj    | Mohak     | xyz@gmail.com   | delhi | india   |
| A003      | Bajaj    | Mohak     | abc@gmail.com   | delhi | india   |
| A004      | Bajaj    | Mohak     | qqq@gmauil.com  | delhi | india   |
+-----------+----------+-----------+-----------------+-------+---------+
4 rows in set (0.00 sec)
```



```
+---------+------------+--------+
| BOOK_ID | BOOK_TITLE | COPIES |
+---------+------------+--------+
| B001    | Book1      |     10 |
| B002    | Book2      |     10 |
| B003    | Book3      |     10 |
| B004    | Book4      |     10 |
+---------+------------+--------+
4 rows in set (0.00 sec)

+-----------+---------+-----------+
| AUTHOR_ID | BOOK_ID | ROLE      |
+-----------+---------+-----------+
| A001      | B002    | author    |
| A002      | B003    | co-author |
| A003      | B004    | author    |
| A004      | B001    | author    |
+-----------+---------+-----------+
4 rows in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Experiment – 3: To understand and use SQL Sub-Query

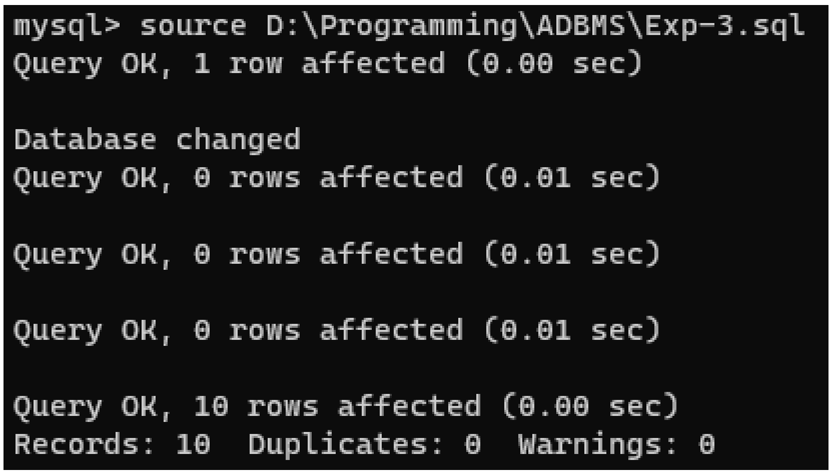
Objective: To understand the use of SQL subquery.

Queries: GitHub

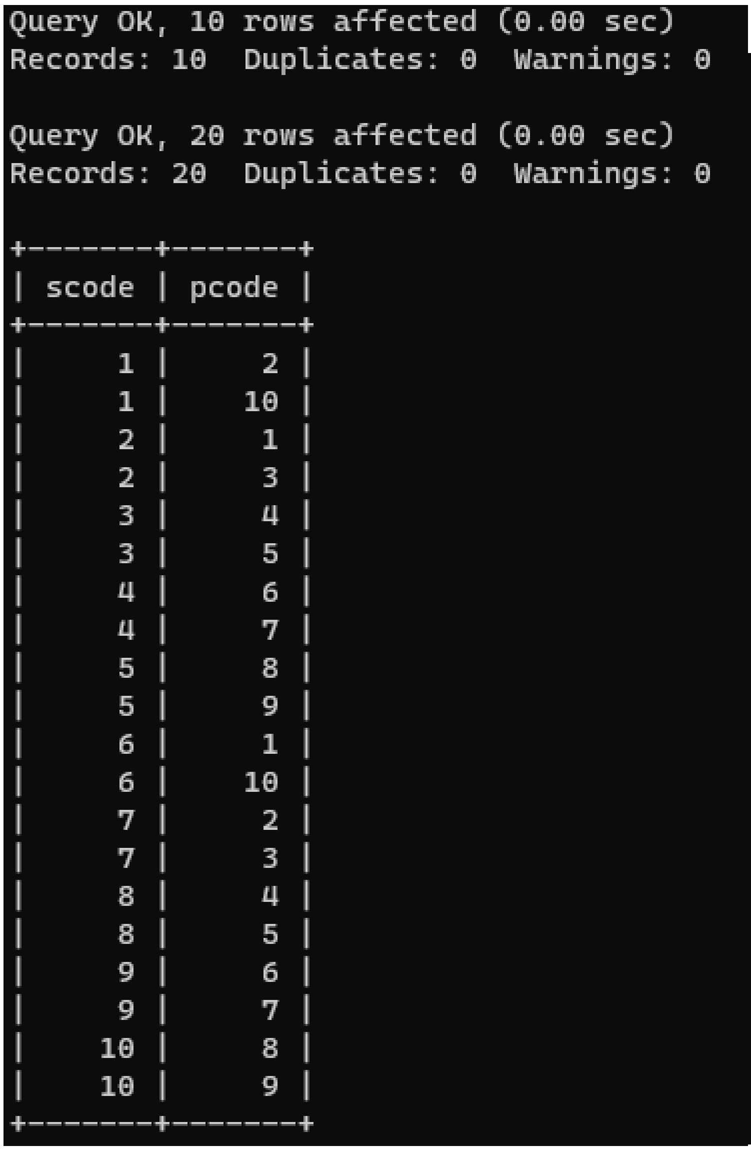
```sql
-- Experiment - 3
-- Objective: To understand the use of sql subquery.
-- 1. Create the following table.
--    Supplier-(scode,sname,scity,turnover)
--    Part-(pcode,weigh,color,cost,sellingprice)
--    Supplier_Part-(scode,pcode,qty)
-- 2. Populate the table
-- 3. Write appropriate SQL Statement for the following:
--      1. Get the supplier number and part number in ascending order of supplier number.
--      2. Get the details of supplier who operate from Bombay with turnover 50.
--      3. Get the total number of supplier.
--      4. Get the part number weighing between 25 and 35.
--      5. Get the supplier number whose turnover is null.
--      6. Get the part number that cost 20, 30 or 40 rupees.
--      7. Get the total quantity of part 2 that is supplied.
--      8. Get the name of supplier who supply part 2.
--      9. Get the part number whose cost is greater than the average cost.
--      10. Get the supplier number and turnover in descending order of turnover.
-- Initialize the Datbase
CREATE DATABASE IF NOT EXISTS exp_3;
USE exp_3;
-- Create the table
CREATE TABLE IF NOT EXISTS supplier(
    scode INT NOT NULL,
    sname VARCHAR(50) NOT NULL,
    scity VARCHAR(50) NOT NULL,
    turnover INT NOT NULL,
    PRIMARY KEY(scode)
);
CREATE TABLE IF NOT EXISTS part(
    pcode INT NOT NULL,
    weigh INT NOT NULL,
    color VARCHAR(50) NOT NULL,
    cost INT NOT NULL,
    sellingprice INT NOT NULL,
    PRIMARY KEY(pcode)
);
CREATE TABLE IF NOT EXISTS supplier_part(
    scode INT NOT NULL,
    pcode INT NOT NULL,
    qty INT NOT NULL,
    PRIMARY KEY(scode, pcode),
    FOREIGN KEY(scode) REFERENCES supplier(scode),
    FOREIGN KEY(pcode) REFERENCES part(pcode)
);
-- Populate the table with fake data
INSERT INTO supplier(scode, sname, scity, turnover)
VALUES (1, 'Supplier 1', 'Mumbai', 100),
    (2, 'Supplier 2', 'Delhi', 200),
    (3, 'Supplier 3', 'Mumbai', 300),
    (4, 'Supplier 4', 'Mumbai', 400),
    (5, 'Supplier 5', 'Delhi', 500),
    (6, 'Supplier 6', 'Mumbai', 600),
    (7, 'Supplier 7', 'Delhi', 700),
    (8, 'Supplier 8', 'Mumbai', 800),
    (9, 'Supplier 9', 'Delhi', 900),
    (10, 'Supplier 10', 'Mumbai', 1000);
INSERT INTO part(pcode, weigh, color, cost, sellingprice)
VALUES (1, 10, 'Red', 10, 20),
    (2, 20, 'Blue', 20, 30),
    (3, 30, 'Green', 30, 40),
    (4, 40, 'Yellow', 40, 50),
    (5, 50, 'Black', 50, 60),
    (6, 60, 'White', 60, 70),
    (7, 70, 'Pink', 70, 80),
    (8, 80, 'Orange', 80, 90),
    (9, 90, 'Purple', 90, 100),
    (10, 100, 'Brown', 100, 110);
INSERT INTO supplier_part(scode, pcode, qty)
VALUES (1, 10, 24),
    (1, 2, 23),
    (2, 3, 35),
    (2, 1, 32),
    (3, 4, 45),
    (3, 5, 43),
    (4, 6, 56),
    (4, 7, 54),
    (5, 8, 67),
    (5, 9, 65),
    (6, 10, 78),
    (6, 1, 76),
    (7, 2, 89),
    (7, 3, 87),
    (8, 4, 90),
    (8, 5, 98),
    (9, 6, 109),
    (9, 7, 107),
    (10, 8, 120),
    (10, 9, 118);
-- 1. Get the supplier number and part number in ascending order of
supplier number.
SELECT scode,
    pcode
FROM supplier_part
ORDER BY scode ASC;
-- 2. Get the details of supplier who operate from Bombay with
turnover 50.
SELECT *
FROM supplier
WHERE scity = 'Mumbai'
    AND turnover = 50;
-- 3. Get the total number of supplier.
SELECT COUNT(*)
FROM supplier;
-- 4. Get the part number weighing between 25 and 35.
SELECT pcode
FROM part
WHERE weigh BETWEEN 25 AND 35;
-- 5. Get the supplier number whose turnover is null.
SELECT scode
FROM supplier
WHERE turnover IS NULL;
```

```sql
-- 6. Get the part number that cost 20, 30 or 40 rupees.
SELECT pcode
FROM part
WHERE cost IN (20, 30, 40);
-- 7. Get the total quantity of part 2 that is supplied.
SELECT SUM(qty)
FROM supplier_part
WHERE pcode = 2;
-- 8. Get the name of supplier who supply part 2.
SELECT sname
FROM supplier
WHERE scode IN (
        SELECT scode
        FROM supplier_part
        WHERE pcode = 2
    );
-- 9. Get the part number whose cost is greater than the average cost.
SELECT pcode
FROM part
WHERE cost > (
        SELECT AVG(cost)
        FROM part
    );
-- 10. Get the supplier number and turnover in descending order of
turnover.
SELECT scode,
    turnover
FROM supplier
ORDER BY turnover DESC;
```

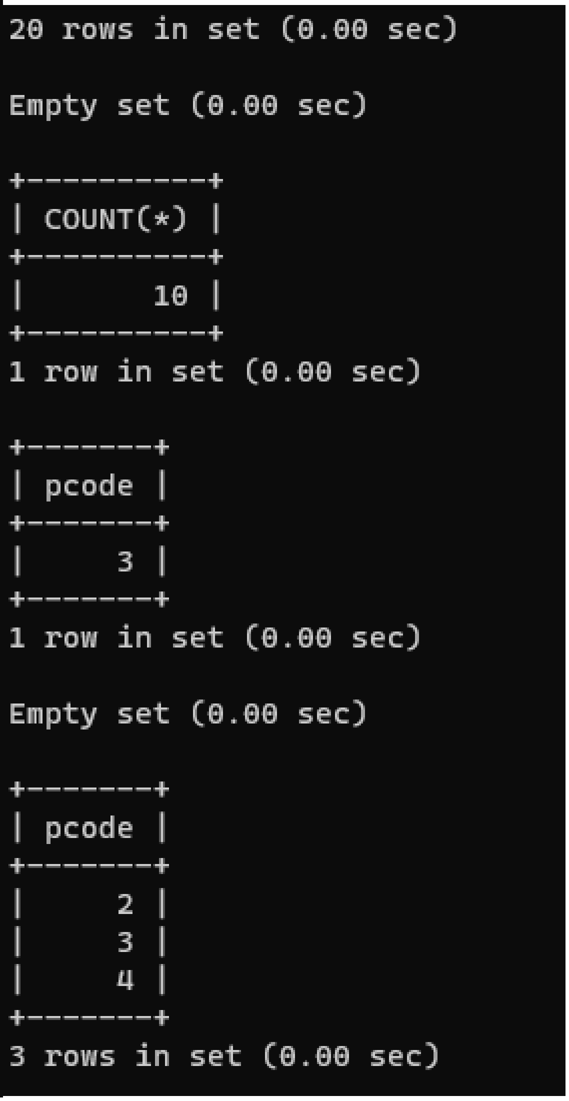
Output:



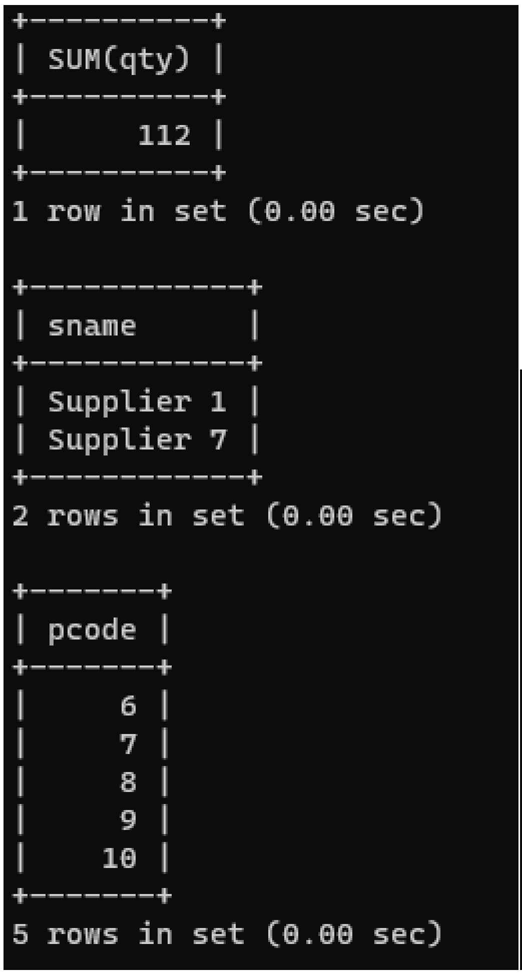
```
mysql> source D:\Programming\ADBMS\Exp-3.sql
Query OK, 1 row affected (0.00 sec)

Database changed
Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 10 rows affected (0.00 sec)
Records: 10  Duplicates: 0  Warnings: 0
```



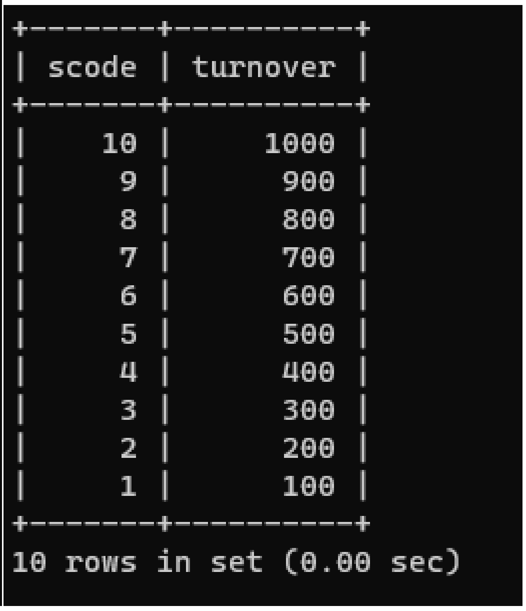
```
Query OK, 10 rows affected (0.00 sec)
Records: 10  Duplicates: 0  Warnings: 0

Query OK, 20 rows affected (0.00 sec)
Records: 20  Duplicates: 0  Warnings: 0

+-------+-------+
| scode | pcode |
+-------+-------+
|     1 |     2 |
|     1 |    10 |
|     2 |     1 |
|     2 |     3 |
|     3 |     4 |
|     3 |     5 |
|     4 |     6 |
|     4 |     7 |
|     5 |     8 |
|     5 |     9 |
|     6 |     1 |
|     6 |    10 |
|     7 |     2 |
|     7 |     3 |
|     8 |     4 |
|     8 |     5 |
|     9 |     6 |
|     9 |     7 |
|    10 |     8 |
|    10 |     9 |
+-------+-------+
```



```
20 rows in set (0.00 sec)

Empty set (0.00 sec)

+----------+
| COUNT(*) |
+----------+
|       10 |
+----------+
1 row in set (0.00 sec)

+-------+
| pcode |
+-------+
|     3 |
+-------+
1 row in set (0.00 sec)

Empty set (0.00 sec)

+-------+
| pcode |
+-------+
|     2 |
|     3 |
|     4 |
+-------+
3 rows in set (0.00 sec)
```



```
+----------+
| SUM(qty) |
+----------+
|      112 |
+----------+
1 row in set (0.00 sec)

+------------+
| sname      |
+------------+
| Supplier 1 |
| Supplier 7 |
+------------+
2 rows in set (0.00 sec)

+-------+
| pcode |
+-------+
|     6 |
|     7 |
|     8 |
|     9 |
|    10 |
+-------+
5 rows in set (0.00 sec)
```



```
+-------+----------+
| scode | turnover |
+-------+----------+
|    10 |     1000 |
|     9 |      900 |
|     8 |      800 |
|     7 |      700 |
|     6 |      600 |
|     5 |      500 |
|     4 |      400 |
|     3 |      300 |
|     2 |      200 |
|     1 |      100 |
+-------+----------+
10 rows in set (0.00 sec)
```

Experiment – 4: Use of Inbuilt functions and relational algebra operation

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

Queries: Github

```sql
-- Experiment - 4
-- Objective: To understand the use of inbuilt function and relational
algebra with sql query
-- 1. Create the following two tables (EMP and DEPT)
CREATE DATABASE IF NOT EXISTS exp_4;
USE exp_4;
-- Create the table
CREATE TABLE IF NOT EXISTS dept(
    deptno INT NOT NULL,
    dname VARCHAR(50) NOT NULL,
    loc VARCHAR(50) NOT NULL,
    PRIMARY KEY(deptno)
);
CREATE TABLE IF NOT EXISTS emp(
    empno INT NOT NULL,
    ename VARCHAR(50) NOT NULL,
    job VARCHAR(50) NOT NULL,
    mgr INT,
    hiredate DATE NOT NULL,
    sal INT NOT NULL,
    comm INT,
    deptno INT NOT NULL,
    PRIMARY KEY(empno),
    FOREIGN KEY(deptno) REFERENCES dept(deptno)
);
-- Insert the data
INSERT INTO dept
VALUES (10, 'ACCOUNTING', 'NEW YORK'),
    (20, 'RESEARCH', 'DALLAS'),
    (30, 'SALES', 'CHICAGO'),
    (40, 'OPERATIONS', 'BOSTON');
INSERT INTO emp
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,
```

```sql
        '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,
```

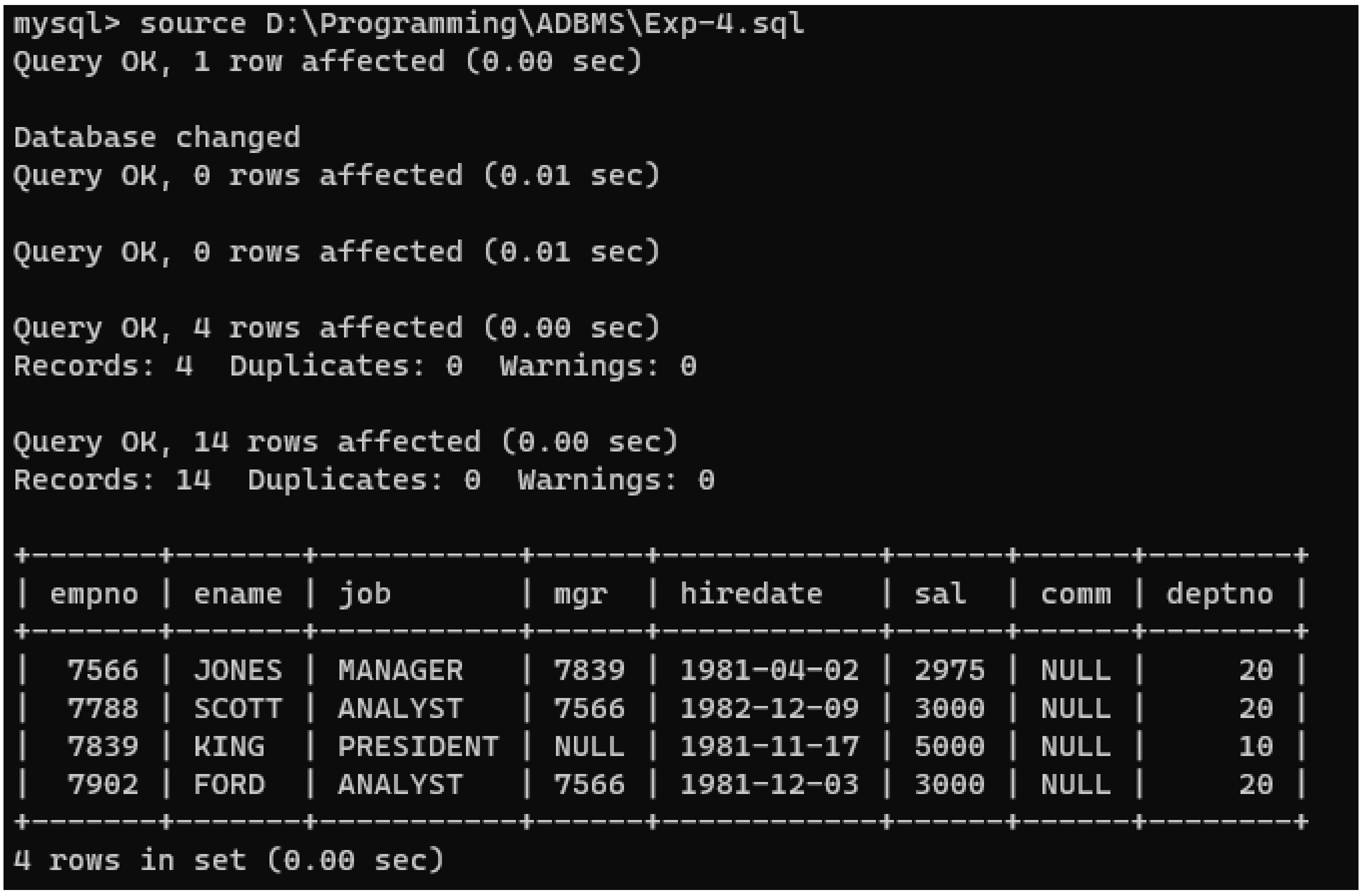
```sql
        'FORD',
        'ANALYST',
        7566,
        '1981-12-03',
        3000,
        NULL,
        20
    ),
    (
        7934,
        'MILLER',
        'CLERK',
        7782,
        '1982-01-23',
        1300,
        NULL,
        10
    );
-- Write the Nested Queries for the following queries.
-- 1. List the details of the emps whose Salaries more than the
employee BLAKE.
SELECT *
FROM emp
WHERE sal > (
        SELECT sal
        FROM emp
        WHERE ename = 'BLAKE'
    );
-- 2. List the emps whose Jobs are same as ALLEN.
SELECT *
FROM emp
WHERE job = (
        SELECT job
        FROM emp
        WHERE ename = 'ALLEN'
    );
-- 3. List the Emps whose Sal is same as FORD or SMITH in desc order
of Names.
SELECT *
FROM emp
```

```sql
WHERE sal = (
        SELECT sal
        FROM emp
        WHERE ename = 'FORD'
    )
    OR sal = (
        SELECT sal
        FROM emp
        WHERE ename = 'SMITH'
    )
ORDER BY ename DESC;
-- 4. List the emps Whose Jobs are same as MILLER or Sal is more than
ALLEN.
SELECT *
FROM emp
WHERE job = (
        SELECT job
        FROM emp
        WHERE ename = 'MILLER'
    )
    OR sal > (
        SELECT sal
        FROM emp
        WHERE ename = 'ALLEN'
    );
-- 5. Find the highest paid employee of sales department.
SELECT *
FROM emp
WHERE sal = (
        SELECT MAX(sal)
        FROM emp
        WHERE deptno = (
                SELECT deptno
                FROM dept
                WHERE dname = 'SALES'
            )
    );
-- 6. List the employees who are senior to most recently hired
employee working under king.
SELECT *
```

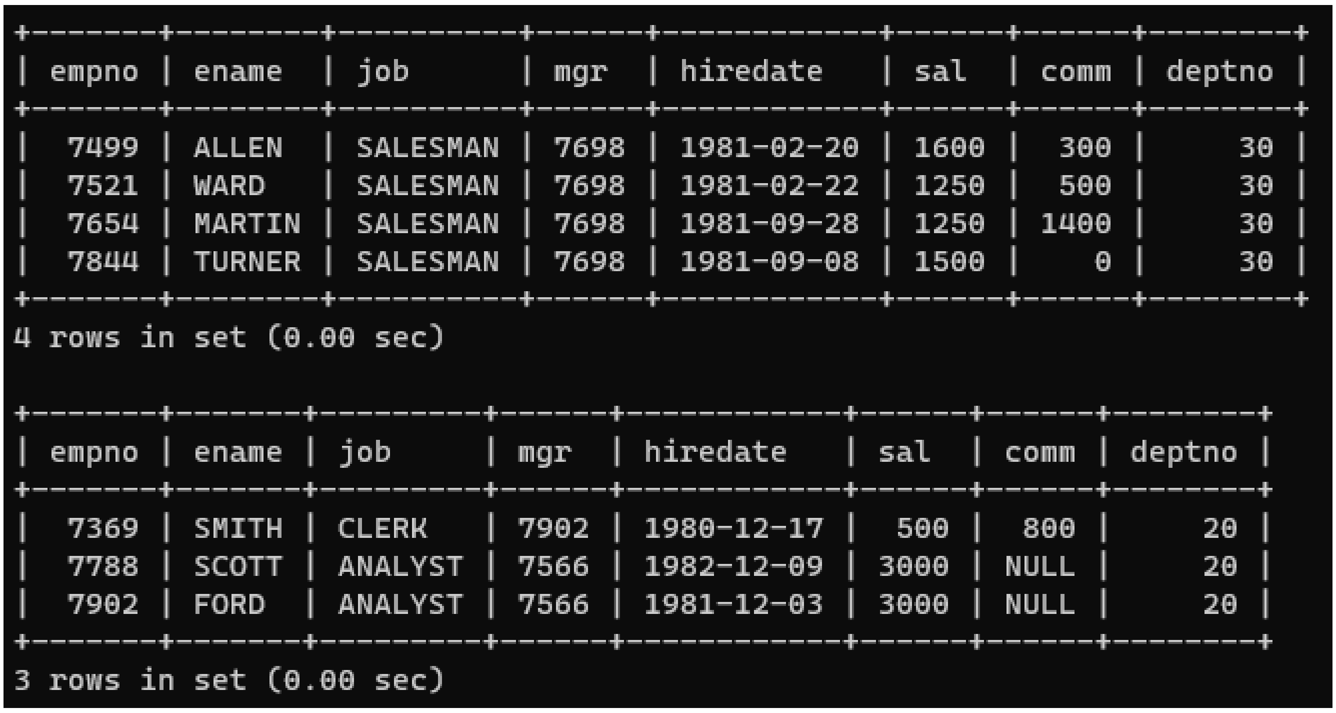
```
FROM emp
WHERE hiredate < (
        SELECT MAX(hiredate)
        FROM emp
        WHERE mgr = (
                SELECT empno
                FROM emp
                WHERE ename = 'KING'
            )
    );
-- 7. List the names of the emps who are getting the highest sal dept
wise.
SELECT ename
FROM emp
WHERE sal IN (
        SELECT MAX(sal)
        FROM emp
        GROUP BY deptno
    );
-- 8. List the emps whose sal is equal to the average of max and
minimum
SELECT *
FROM emp
WHERE sal = (
        SELECT AVG(max_min_sal)
        FROM (
                SELECT MAX(sal) + MIN(sal) AS max_min_sal
                FROM emp
            ) as temp
    );
-- 9. List the emps who joined in the company on the same date.
SELECT *
FROM emp
WHERE hiredate = (
        SELECT hiredate
        FROM emp
        GROUP BY hiredate
        HAVING COUNT(hiredate) > 1
    );
```

```sql
-- 10. Find out the emps who joined in the company before their
Managers.
SELECT *
FROM emp e
WHERE hiredate < (
        SELECT hiredate
        FROM emp
        WHERE empno = e.mgr
    );
```

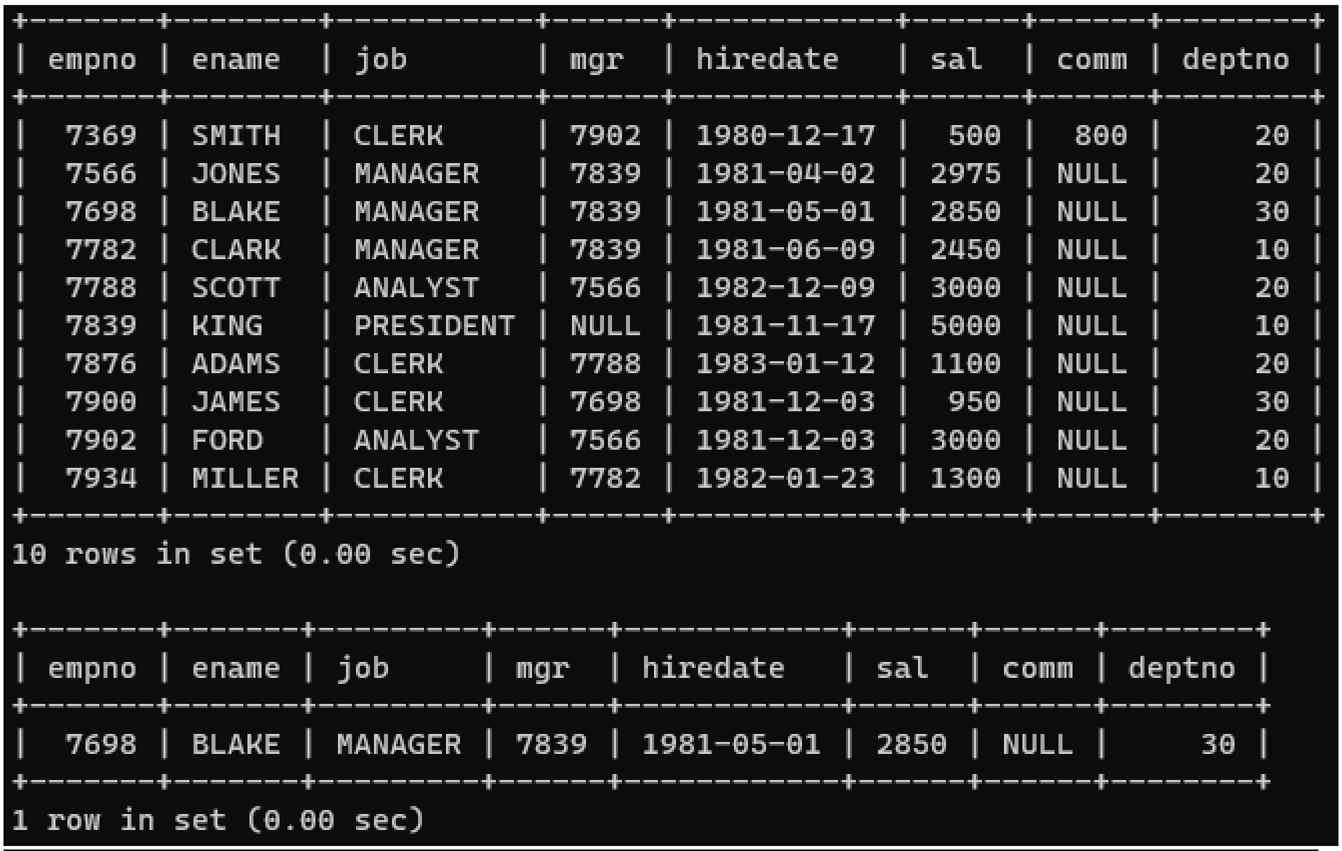
Output:



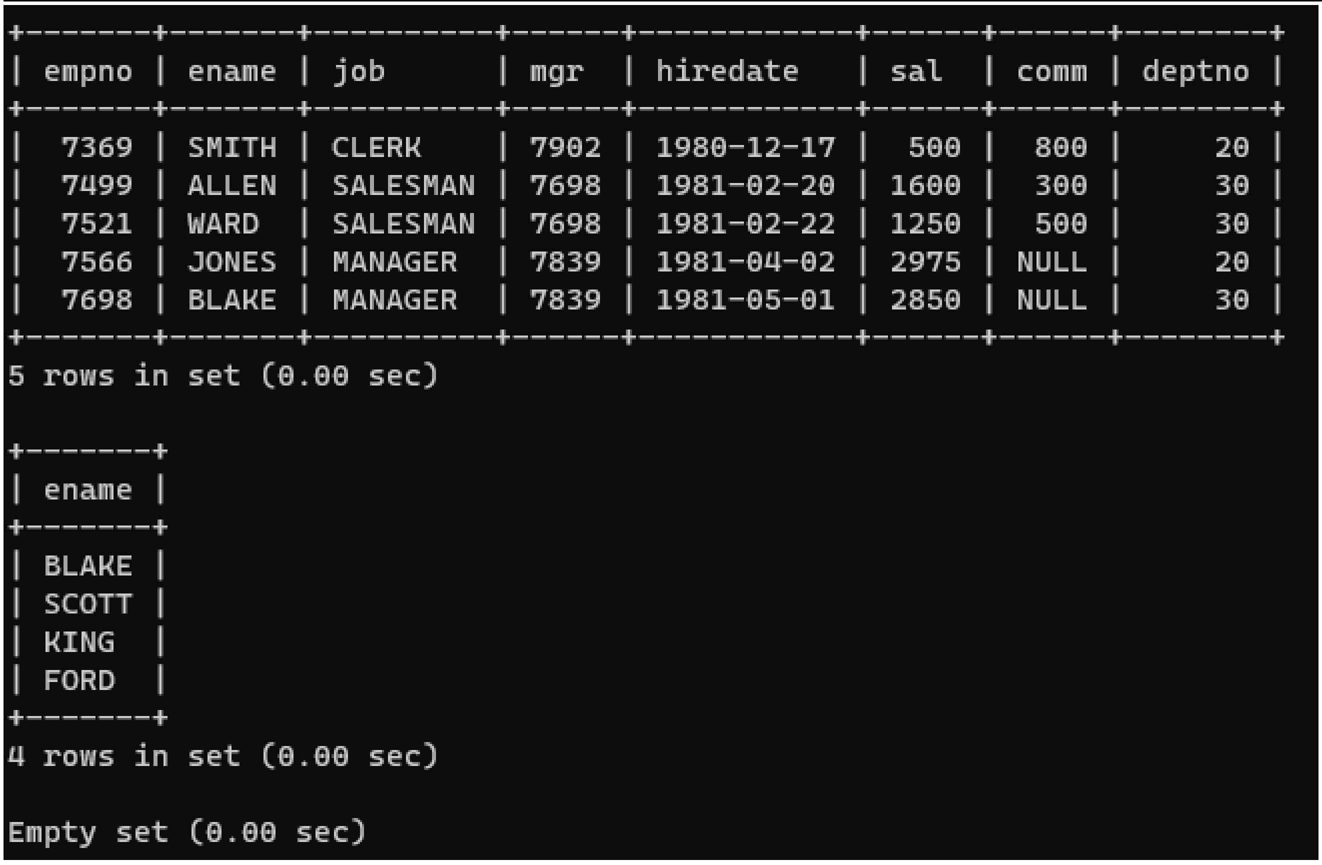
```
mysql> source D:\Programming\ADBMS\Exp-4.sql
Query OK, 1 row affected (0.00 sec)

Database changed
Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 4 rows affected (0.00 sec)
Records: 4  Duplicates: 0  Warnings: 0

Query OK, 14 rows affected (0.00 sec)
Records: 14  Duplicates: 0  Warnings: 0

+-------+-------+-----------+------+------------+------+------+--------+
| 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)
```



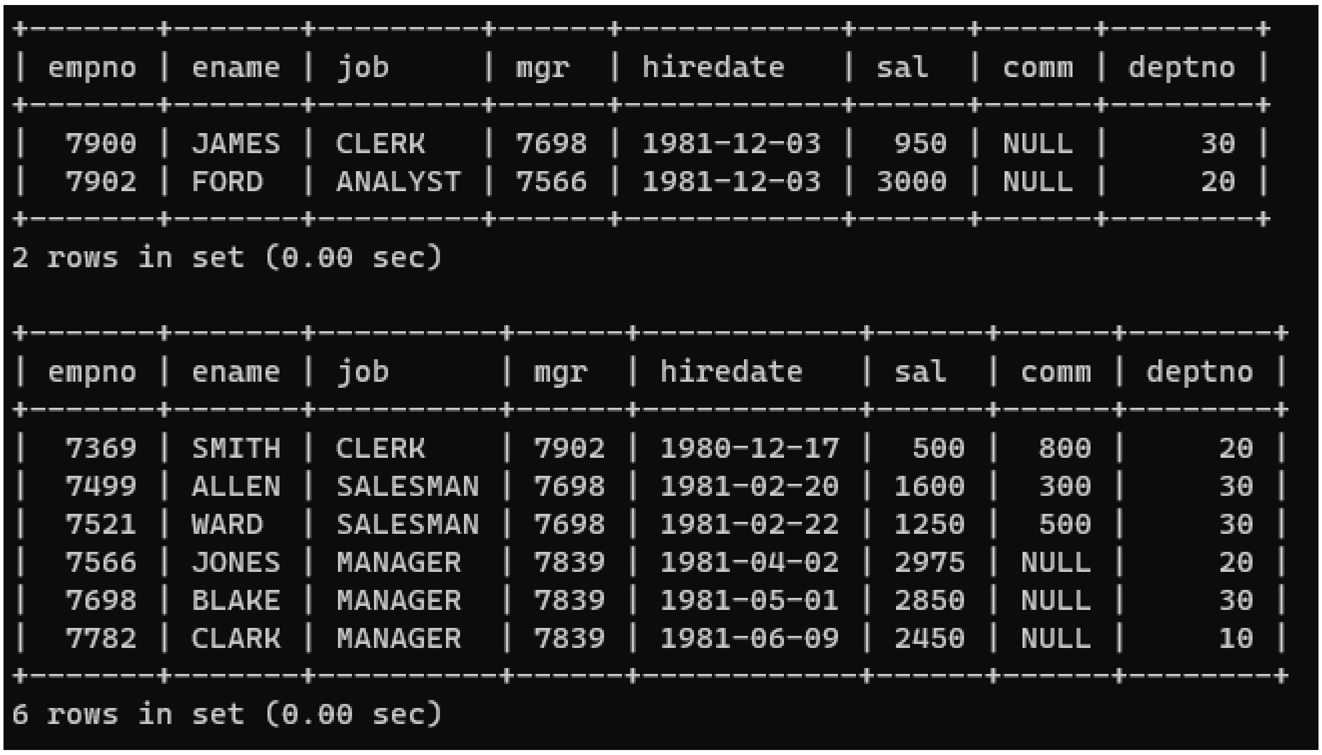
```
+-------+--------+----------+------+------------+------+------+--------+
| empno | ename  | job      | mgr  | hiredate   | sal  | comm | deptno |
+-------+--------+----------+------+------------+------+------+--------+
|  7499 | ALLEN  | SALESMAN | 7698 | 1981-02-20 | 1600 |  300 |     30 |
|  7521 | 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)

+-------+-------+---------+------+------------+------+------+--------+
| 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)
```



```
+-------+--------+-----------+------+------------+------+------+--------+
| 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)

+-------+-------+---------+------+------------+------+------+--------+
| 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)
```



```
+-------+-------+----------+------+------------+------+------+--------+
| 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)

+-------+
| ename |
+-------+
| BLAKE |
| SCOTT |
| KING  |
| FORD  |
+-------+
4 rows in set (0.00 sec)

Empty set (0.00 sec)
```


```
+-------+-------+---------+------+------------+------+------+--------+
| 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)

+-------+-------+----------+------+------------+------+------+--------+
| 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 (0.00 sec)
```

Experiment – 5: Use of different SQL clauses and join

Objective: To understand the use of group by and having clause and execute the SQL commands using JOIN

Queries: GitHub

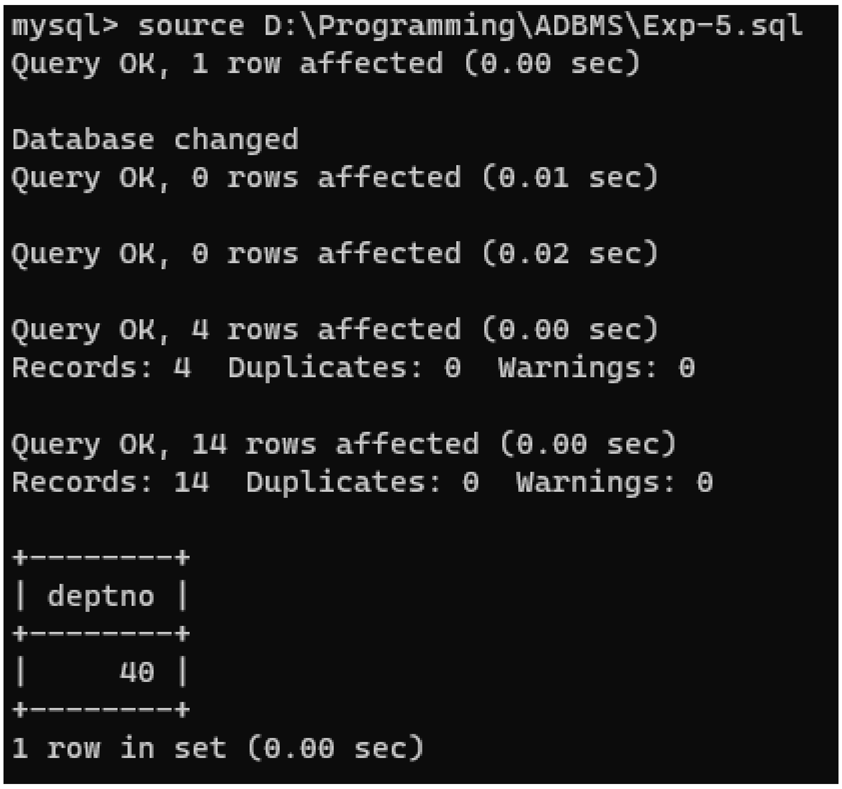
```sql
-- Experiment - 5
-- Objective: To understand the use of group by and having clause and
execute the SQL commands using JOIN
CREATE DATABASE IF NOT EXISTS exp_5;
USE exp_5;
-- Create the table
CREATE TABLE IF NOT EXISTS dept(
    deptno INT NOT NULL,
    dname VARCHAR(50) NOT NULL,
    loc VARCHAR(50) NOT NULL,
    PRIMARY KEY(deptno)
);
CREATE TABLE IF NOT EXISTS emp(
    empno INT NOT NULL,
    ename VARCHAR(50) NOT NULL,
    job VARCHAR(50) NOT NULL,
    mgr INT,
    hiredate DATE NOT NULL,
    sal INT NOT NULL,
    comm INT,
    deptno INT NOT NULL,
    PRIMARY KEY(empno),
    FOREIGN KEY(deptno) REFERENCES dept(deptno)
);
-- Insert the data
INSERT INTO dept
VALUES (10, 'ACCOUNTING', 'NEW YORK'),
    (20, 'RESEARCH', 'DALLAS'),
    (30, 'SALES', 'CHICAGO'),
    (40, 'OPERATIONS', 'BOSTON');
INSERT INTO emp
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',
```

```sql
        7566,
        '1981-12-03',
        3000,
        NULL,
        20
    ),
    (
        7934,
        'MILLER',
        'CLERK',
        7782,
        '1982-01-23',
        1300,
        NULL,
        10
    );
-- 1. 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.
SELECT deptno
FROM dept
WHERE deptno NOT IN (
        SELECT deptno
        FROM emp
    );
-- 2. List the No.of emp’s and Avg salary within each department for
each job.
SELECT deptno,
    job,
    COUNT(empno),
    AVG(sal)
FROM emp
GROUP BY deptno,
    job;
-- 3. Find the maximum average salary drawn for each job except for
‘President’.
SELECT MAX(sal)
FROM emp
WHERE sal IN (
        SELECT AVG(sal)
        FROM emp
        WHERE job <> 'PRESIDENT'
        GROUP BY job
    );
-- 4. List the department details where at least two emps are working.
SELECT *
FROM dept
WHERE deptno IN (
        SELECT deptno
        FROM emp
        GROUP BY deptno
        HAVING COUNT(empno) >= 2
    );
-- 5. List the no. of emps in each department where the no. is more
than 3.
SELECT deptno,
    COUNT(*) AS No_of_emp
FROM emp
GROUP BY deptno
HAVING COUNT(*) > 3;
-- 6. List the names of the emps who are getting the highest sal dept
wise.
SELECT deptno,
    ename,
    sal
FROM emp e
WHERE sal IN (
        SELECT MAX(sal)
        FROM emp
        GROUP BY deptno
    );
-- 7. List the Deptno and their average salaries for dept with the
average salary less than the averages for all departments.
SELECT deptno,
    AVG(sal)
FROM emp
GROUP BY deptno
HAVING AVG(sal) < (
        SELECT AVG(sal)
        FROM emp
    );
-- 2. Execute the experiment 4 using sql join.
-- 1. List the details of the emps whose Salaries more than the
employee BLAKE.
SELECT *
FROM emp
    RIGHT JOIN dept ON emp.deptno = dept.deptno
WHERE sal > (
        SELECT sal
        FROM emp
        WHERE ename = 'BLAKE'
    );
-- 2. List the emps whose Jobs are same as ALLEN.
SELECT *
FROM emp
    RIGHT JOIN dept ON emp.deptno = dept.deptno
WHERE job = (
        SELECT job
        FROM emp
        WHERE ename = 'ALLEN'
    );
-- 3. List the Emps whose Sal is same as FORD or SMITH in desc order
of Names.
SELECT *
FROM emp
    RIGHT JOIN dept ON emp.deptno = dept.deptno
WHERE sal = (
        SELECT sal
        FROM emp
        WHERE ename = 'FORD'
    )
    OR sal = (
        SELECT sal
        FROM emp
        WHERE ename = 'SMITH'
    )
ORDER BY ename DESC;
-- 4. List the emps Whose Jobs are same as MILLER or Sal is more than
ALLEN.
```

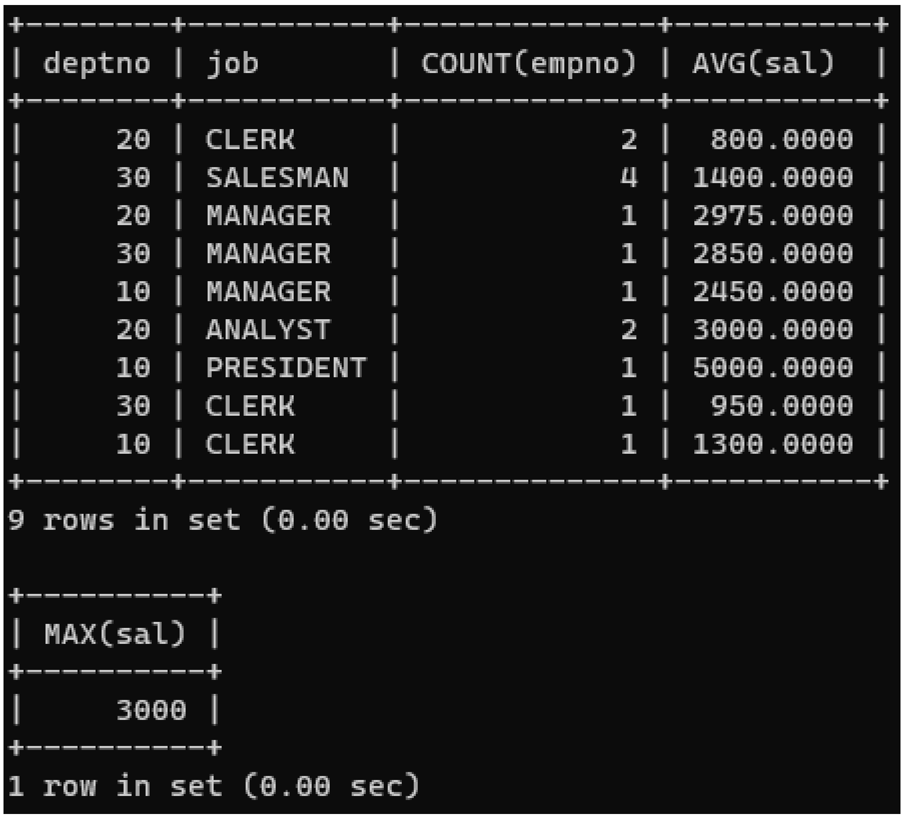
```sql
SELECT *
FROM emp
    RIGHT JOIN dept ON emp.deptno = dept.deptno
WHERE job = (
        SELECT job
        FROM emp
        WHERE ename = 'MILLER'
    )
    OR sal > (
        SELECT sal
        FROM emp
        WHERE ename = 'ALLEN'
    );
-- 5. Find the highest paid employee of sales department.
SELECT *
FROM emp
    RIGHT JOIN dept ON emp.deptno = dept.deptno
WHERE dept.dname = 'SALES'
    AND sal = (
        SELECT MAX(sal)
        FROM emp
            RIGHT JOIN dept ON emp.deptno = dept.deptno
        WHERE dept.dname = 'SALES'
    );
-- 6. List the employees who are senior to most recently hired
employee working under king.
SELECT *
FROM emp
    RIGHT JOIN dept ON emp.deptno = dept.deptno
WHERE hiredate < (
        SELECT MAX(hiredate)
        FROM emp
            RIGHT JOIN dept ON emp.deptno = dept.deptno
        WHERE mgr = (
                SELECT empno
                FROM emp
                    RIGHT JOIN dept ON emp.deptno = dept.deptno
                WHERE ename = 'KING'
            )
    );
```

```sql
-- 7. List the names of the emps who are getting the highest sal dept
wise.
SELECT *
FROM emp
    RIGHT JOIN dept ON emp.deptno = dept.deptno
WHERE sal IN (
        SELECT MAX(sal)
        FROM emp
        GROUP BY deptno
    );
-- 8. List the emps whose sal is equal to the average of max and
minimum
SELECT *
FROM emp
    RIGHT Join dept ON emp.deptno = dept.deptno
WHERE sal = (
        SELECT (MAX(sal) + MIN(sal)) / 2
        FROM emp
    );
-- 9. List the emps who joined in the company on the same date.
SELECT *
FROM emp
    RIGHT JOIN dept ON emp.deptno = dept.deptno
WHERE hiredate IN (
        SELECT hiredate
        FROM emp
        GROUP BY hiredate
        HAVING COUNT(hiredate) > 1
    );
-- 10. Find out the emps who joined in the company before their
Managers
SELECT *
FROM emp e
    RIGHT JOIN dept ON e.deptno = dept.deptno
WHERE hiredate < (
        SELECT hiredate
        FROM emp
        WHERE empno = e.mgr
    );
```

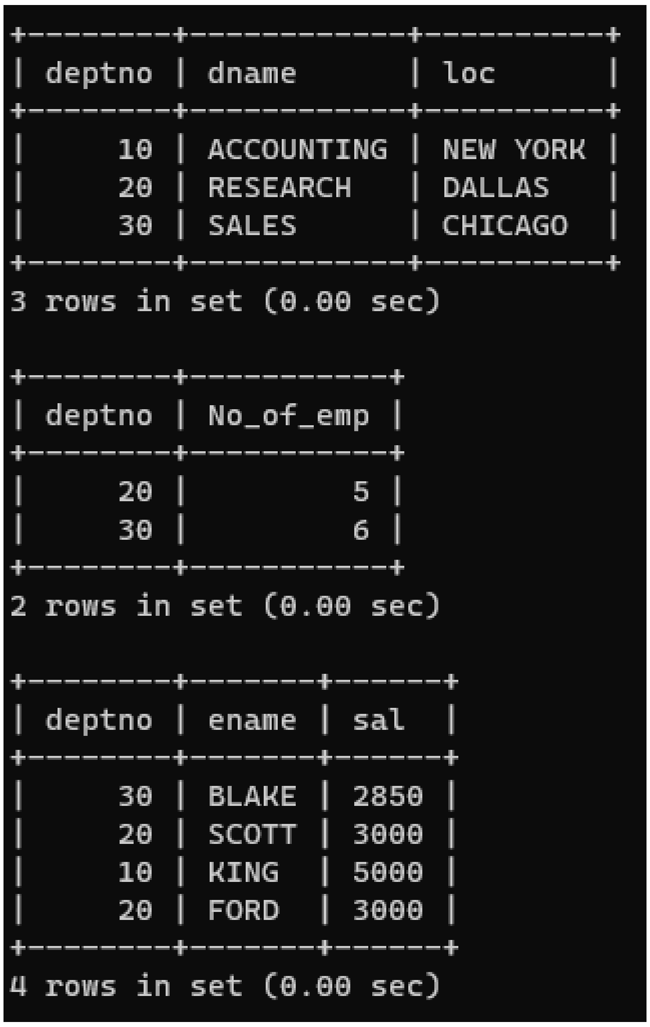
Output:



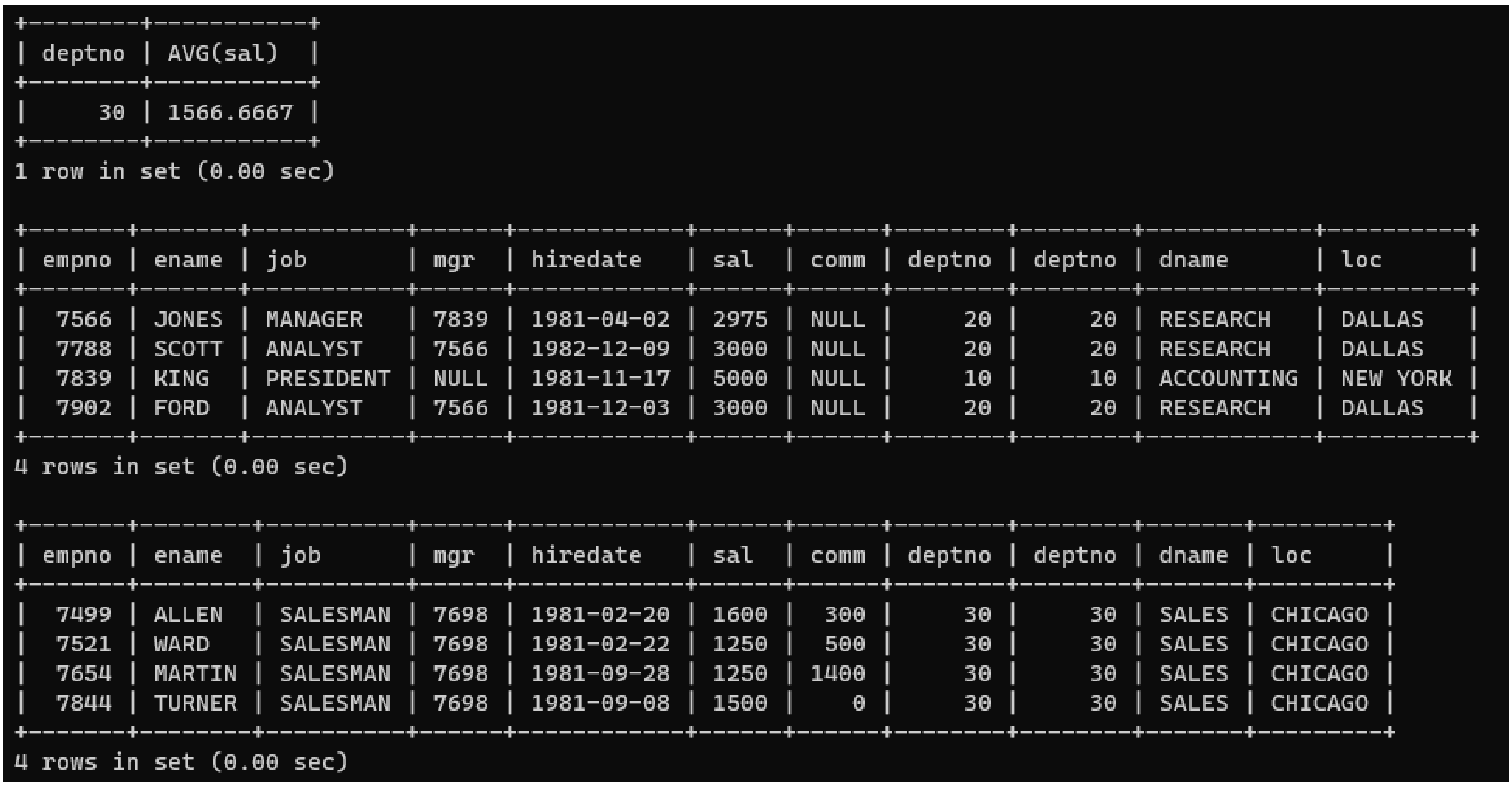
```
mysql> source D:\Programming\ADBMS\Exp-5.sql
Query OK, 1 row affected (0.00 sec)

Database changed
Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 4 rows affected (0.00 sec)
Records: 4  Duplicates: 0  Warnings: 0

Query OK, 14 rows affected (0.00 sec)
Records: 14  Duplicates: 0  Warnings: 0

+--------+
| deptno |
+--------+
|     40 |
+--------+
1 row in set (0.00 sec)
```



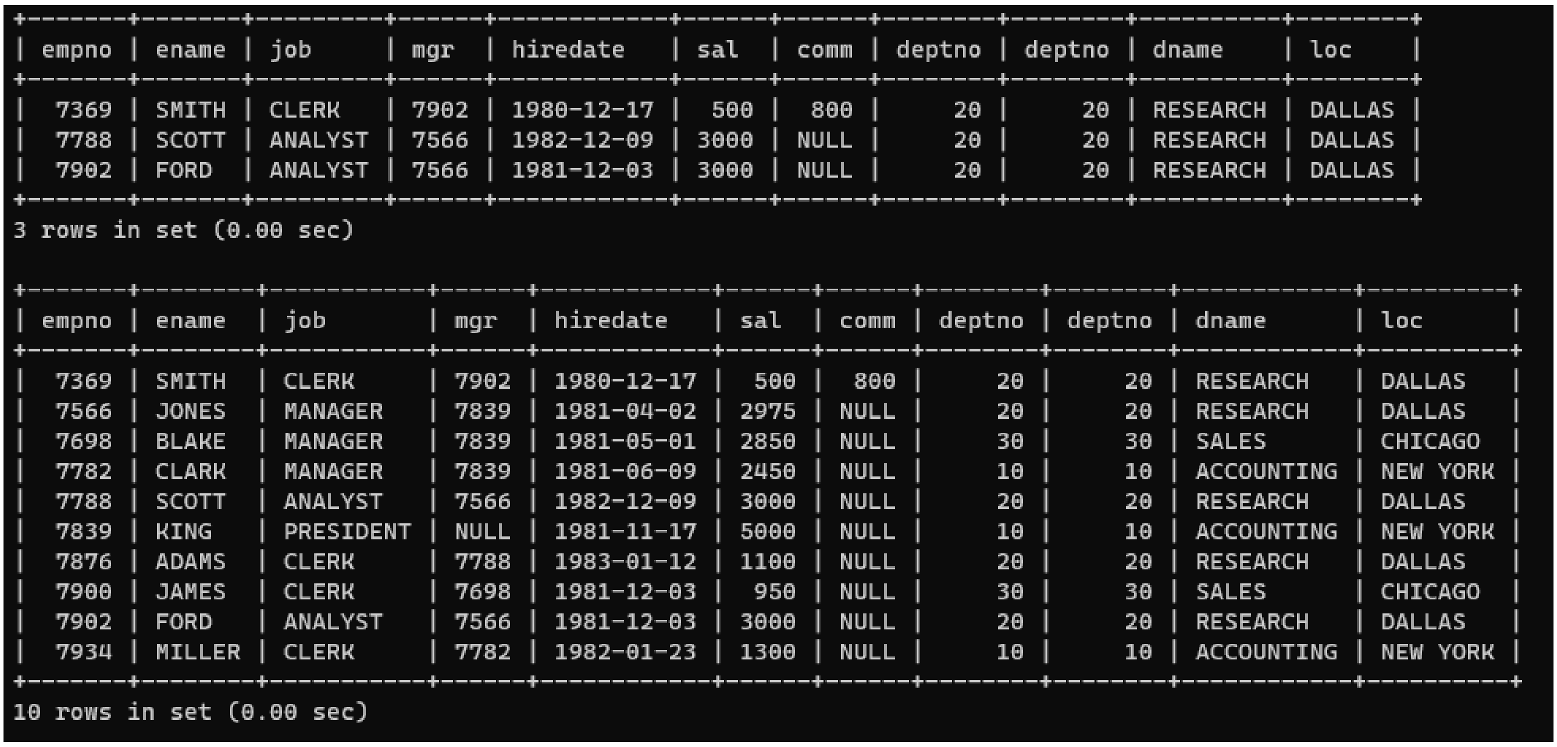
```
+--------+-----------+--------------+-----------+
| deptno | job       | COUNT(empno) | AVG(sal)  |
+--------+-----------+--------------+-----------+
|     20 | CLERK     |            2 |  800.0000 |
|     30 | SALESMAN  |            4 | 1400.0000 |
|     20 | MANAGER   |            1 | 2975.0000 |
|     30 | MANAGER   |            1 | 2850.0000 |
|     10 | MANAGER   |            1 | 2450.0000 |
|     20 | ANALYST   |            2 | 3000.0000 |
|     10 | PRESIDENT |            1 | 5000.0000 |
|     30 | CLERK     |            1 |  950.0000 |
|     10 | CLERK     |            1 | 1300.0000 |
+--------+-----------+--------------+-----------+
9 rows in set (0.00 sec)

+----------+
| MAX(sal) |
+----------+
|     3000 |
+----------+
1 row in set (0.00 sec)
```



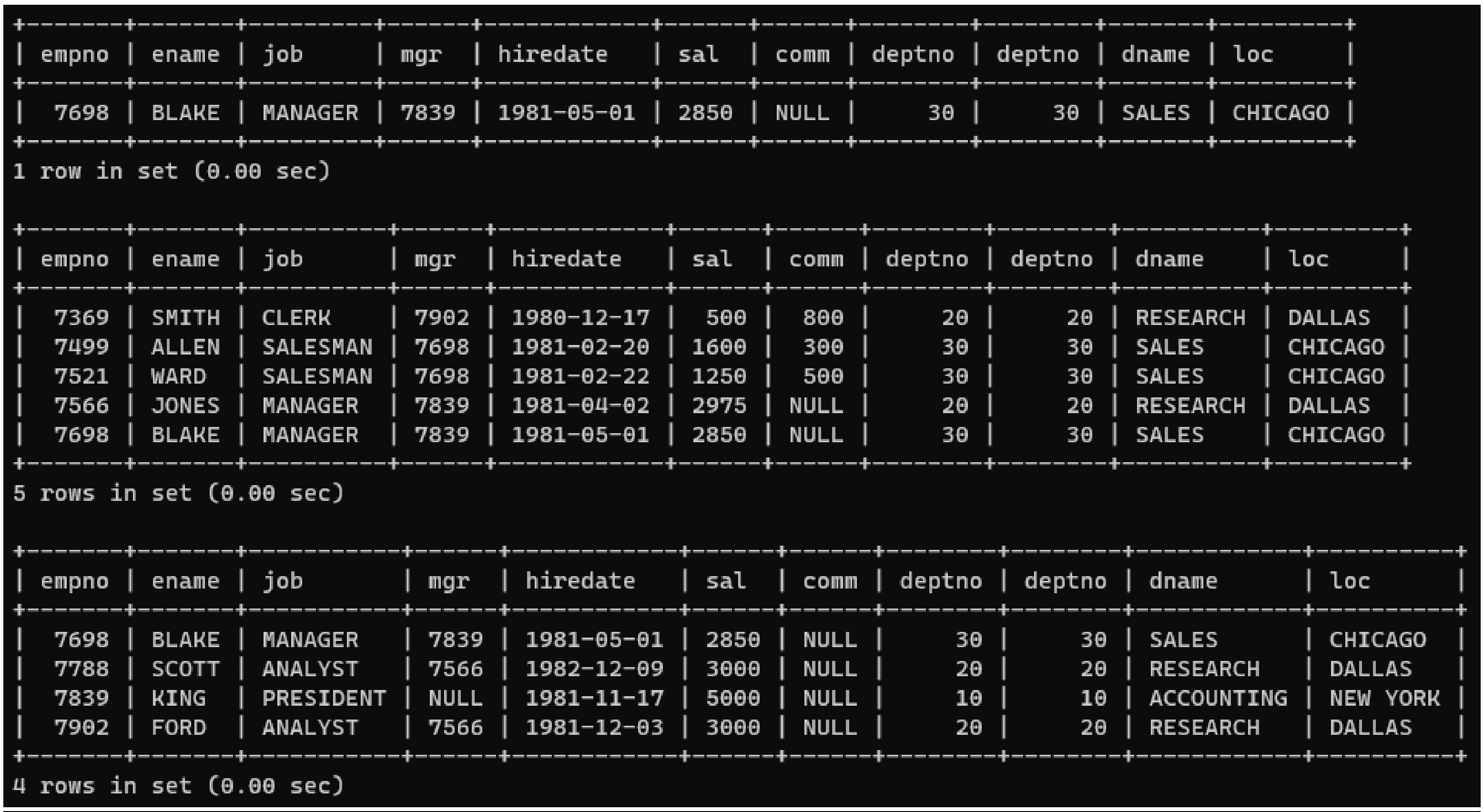
```
+--------+------------+----------+
| deptno | dname      | loc      |
+--------+------------+----------+
|     10 | ACCOUNTING | NEW YORK |
|     20 | RESEARCH   | DALLAS   |
|     30 | SALES      | CHICAGO  |
+--------+------------+----------+
3 rows in set (0.00 sec)

+--------+-----------+
| deptno | No_of_emp |
+--------+-----------+
|     20 |         5 |
|     30 |         6 |
+--------+-----------+
2 rows in set (0.00 sec)

+--------+-------+------+
| deptno | ename | sal  |
+--------+-------+------+
|     30 | BLAKE | 2850 |
|     20 | SCOTT | 3000 |
|     10 | KING  | 5000 |
|     20 | FORD  | 3000 |
+--------+-------+------+
4 rows in set (0.00 sec)
```



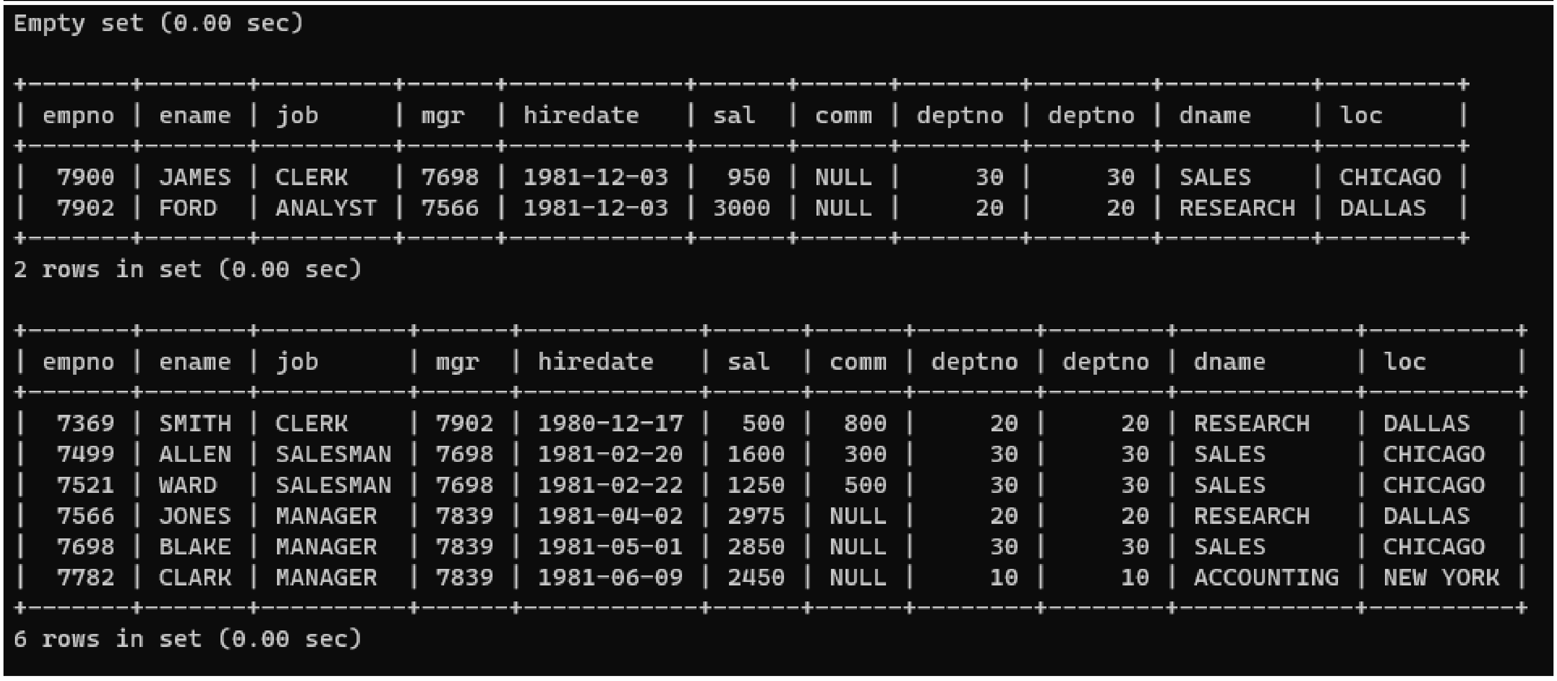
```
+--------+-----------+
| deptno | AVG(sal)  |
+--------+-----------+
|     30 | 1566.6667 |
+--------+-----------+
1 row in set (0.00 sec)

+-------+-------+-----------+------+------------+------+------+--------+--------+------------+----------+
| 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)

+-------+--------+----------+------+------------+------+------+--------+--------+-------+---------+
| empno | ename  | job      | mgr  | hiredate   | sal  | comm | deptno | deptno | dname | loc     |
+-------+--------+----------+------+------------+------+------+--------+--------+-------+---------+
|  7499 | ALLEN  | SALESMAN | 7698 | 1981-02-20 | 1600 |  300 |     30 |     30 | SALES | CHICAGO |
|  7521 | WARD   | SALESMAN | 7698 | 1981-02-22 | 1250 |  500 |     30 |     30 | SALES | CHICAGO |
|  7654 | MARTIN | SALESMAN | 7698 | 1981-09-28 | 1250 | 1400 |     30 |     30 | SALES | CHICAGO |
|  7844 | TURNER | SALESMAN | 7698 | 1981-09-08 | 1500 |    0 |     30 |     30 | SALES | CHICAGO |
+-------+--------+----------+------+------------+------+------+--------+--------+-------+---------+
4 rows in set (0.00 sec)
```



```
+-------+-------+---------+------+------------+------+------+--------+--------+----------+--------+
| empno | ename | job     | mgr  | hiredate   | sal  | comm | deptno | deptno | dname    | loc    |
+-------+-------+---------+------+------------+------+------+--------+--------+----------+--------+
|  7369 | SMITH | CLERK   | 7902 | 1980-12-17 |  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)

+-------+--------+-----------+------+------------+------+------+--------+--------+------------+----------+
| empno | ename  | job       | mgr  | hiredate   | sal  | comm | deptno | deptno | dname      | loc      |
+-------+--------+-----------+------+------------+------+------+--------+--------+------------+----------+
|  7369 | SMITH  | CLERK     | 7902 | 1980-12-17 |  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 | ADAMS  | 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 | 1982-01-23 | 1300 | NULL |     10 |     10 | ACCOUNTING | NEW YORK |
+-------+--------+-----------+------+------------+------+------+--------+--------+------------+----------+
10 rows in set (0.00 sec)
```



```
+-------+-------+---------+------+------------+------+------+--------+--------+-------+---------+
| empno | ename | job     | mgr  | hiredate   | sal  | comm | deptno | deptno | dname | loc     |
+-------+-------+---------+------+------------+------+------+--------+--------+-------+---------+
|  7698 | BLAKE | MANAGER | 7839 | 1981-05-01 | 2850 | NULL |     30 |     30 | SALES | CHICAGO |
+-------+-------+---------+------+------------+------+------+--------+--------+-------+---------+
1 row in set (0.00 sec)

+-------+-------+----------+------+------------+------+------+--------+--------+----------+---------+
| empno | ename | job      | mgr  | hiredate   | sal  | comm | deptno | deptno | dname    | loc     |
+-------+-------+----------+------+------------+------+------+--------+--------+----------+---------+
|  7369 | SMITH | CLERK    | 7902 | 1980-12-17 |  500 |  800 |     20 |     20 | RESEARCH | DALLAS  |
|  7499 | ALLEN | SALESMAN | 7698 | 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 |
+-------+-------+----------+------+------------+------+------+--------+--------+----------+---------+
5 rows in set (0.00 sec)

+-------+-------+-----------+------+------------+------+------+--------+--------+------------+----------+
| 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)
```



```
Empty set (0.00 sec)

+-------+-------+---------+------+------------+------+------+--------+--------+----------+---------+
| 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)

+-------+-------+----------+------+------------+------+------+--------+--------+------------+----------+
| empno | ename | job      | mgr  | hiredate   | sal  | comm | deptno | deptno | dname      | loc      |
+-------+-------+----------+------+------------+------+------+--------+--------+------------+----------+
|  7369 | SMITH | CLERK    | 7902 | 1980-12-17 |  500 |  800 |     20 |     20 | RESEARCH   | DALLAS   |
|  7499 | ALLEN | SALESMAN | 7698 | 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  |
|  7782 | CLARK | MANAGER  | 7839 | 1981-06-09 | 2450 | NULL |     10 |     10 | ACCOUNTING | NEW YORK |
+-------+-------+----------+------+------------+------+------+--------+--------+------------+----------+
6 rows in set (0.00 sec)
```

Experiment – 6: To understand the concepts of Views.

Objective: Students will be able to implement the concept of views.

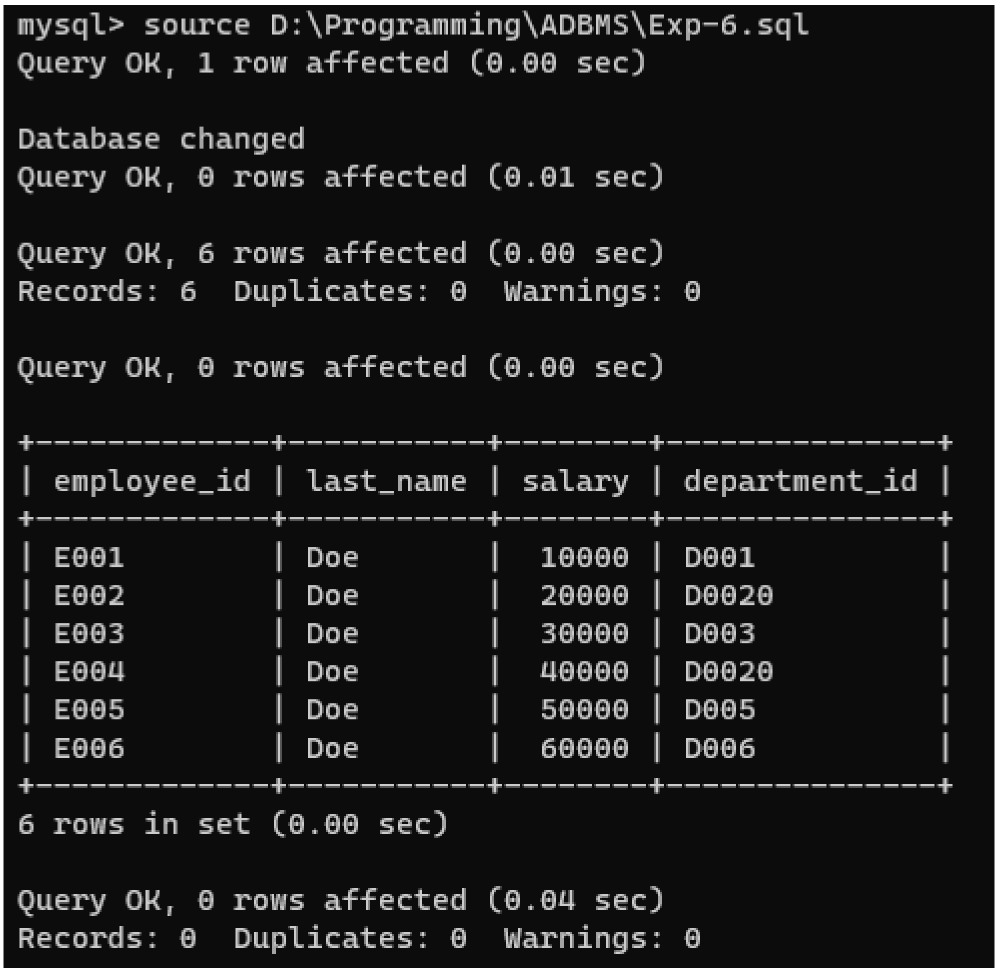
Queries: Github

```sql
-- Experiment - 6
-- Objective: Students will be able to implement the concept of views.
CREATE DATABASE IF NOT EXISTS exp_6;
USE exp_6;
-- 1. 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
CREATE TABLE IF NOT EXISTS employee(
    employee_id CHAR(10) NOT NULL,
    first_name CHAR(30) NOT NULL,
    last_name CHAR(30) NOT NULL,
    dob DATE,
    salary INT NOT NULL,
    department_id CHAR(10) NOT NULL,
    PRIMARY KEY(employee_id)
);
-- add 6 rows
INSERT INTO employee
VALUES (
        'E001',
        'John',
        'Doe',
        '1990-01-01',
        10000,
        'D001'
    ),
    (
        'E002',
        'Jane',
        'Doe',
        '1991-02-02',
        20000,
        'D0020'
    ),
    (
        'E003',
        'John',
        'Doe',
        '1992-03-03',
        30000,
        'D003'
    ),
    (
        'E004',
        'John',
        'Doe',
        '1993-04-04',
        40000,
        'D0020'
    ),
    (
        'E005',
        'John',
        'Doe',
        '1994-05-05',
        50000,
        'D005'
    ),
    (
        'E006',
        'John',
        'Doe',
        '1995-06-06',
        60000,
        'D006'
    );
-- 2. 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.:
```

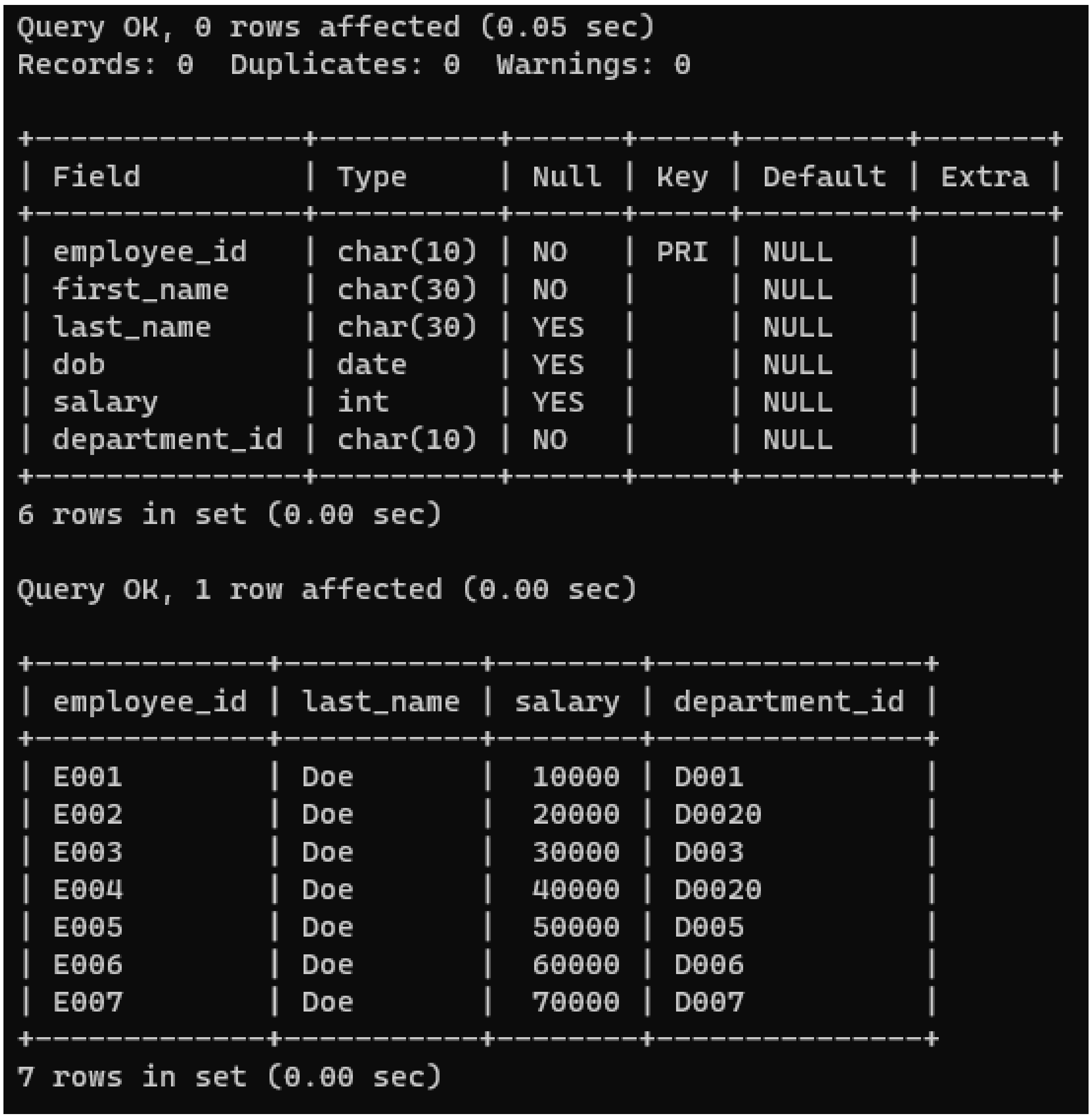
```sql
CREATE VIEW emp_view AS
SELECT employee_id,
    last_name,
    salary,
    department_id
FROM employee;
SELECT *
FROM emp_view;
-- 2) Insert values into view(remove the NOT NULL constraint and then
insert values):
ALTER TABLE employee
Modify salary INT;
ALTER TABLE employee
Modify last_name CHAR(30);
DESC employee;
INSERT INTO employee
VALUES (
        'E007',
        'John',
        'Doe',
        '1996-07-07',
        70000,
        'D007'
    );
SELECT *
FROM emp_view;
-- 3) Modify, delete and drop operations are performed on view.:
UPDATE emp_view
SET Department_id = 'D0020'
WHERE Employee_id = 'E004';
DELETE FROM emp_view
WHERE Last_Name = 'Doe'
    AND salary = 10000;
SELECT *
FROM emp_view;
DROP VIEW emp_view;
-- 4) Creates a view named salary_view. The view shows the employees
in department 20 and
-- their annual salary.
CREATE VIEW salary_view AS
```

```sql
SELECT employee_id,
    first_name,
    last_name,
    salary * 12 AS annual_salary
FROM employee
WHERE department_id = 'D0020';
SELECT *
FROM salary_view;
```

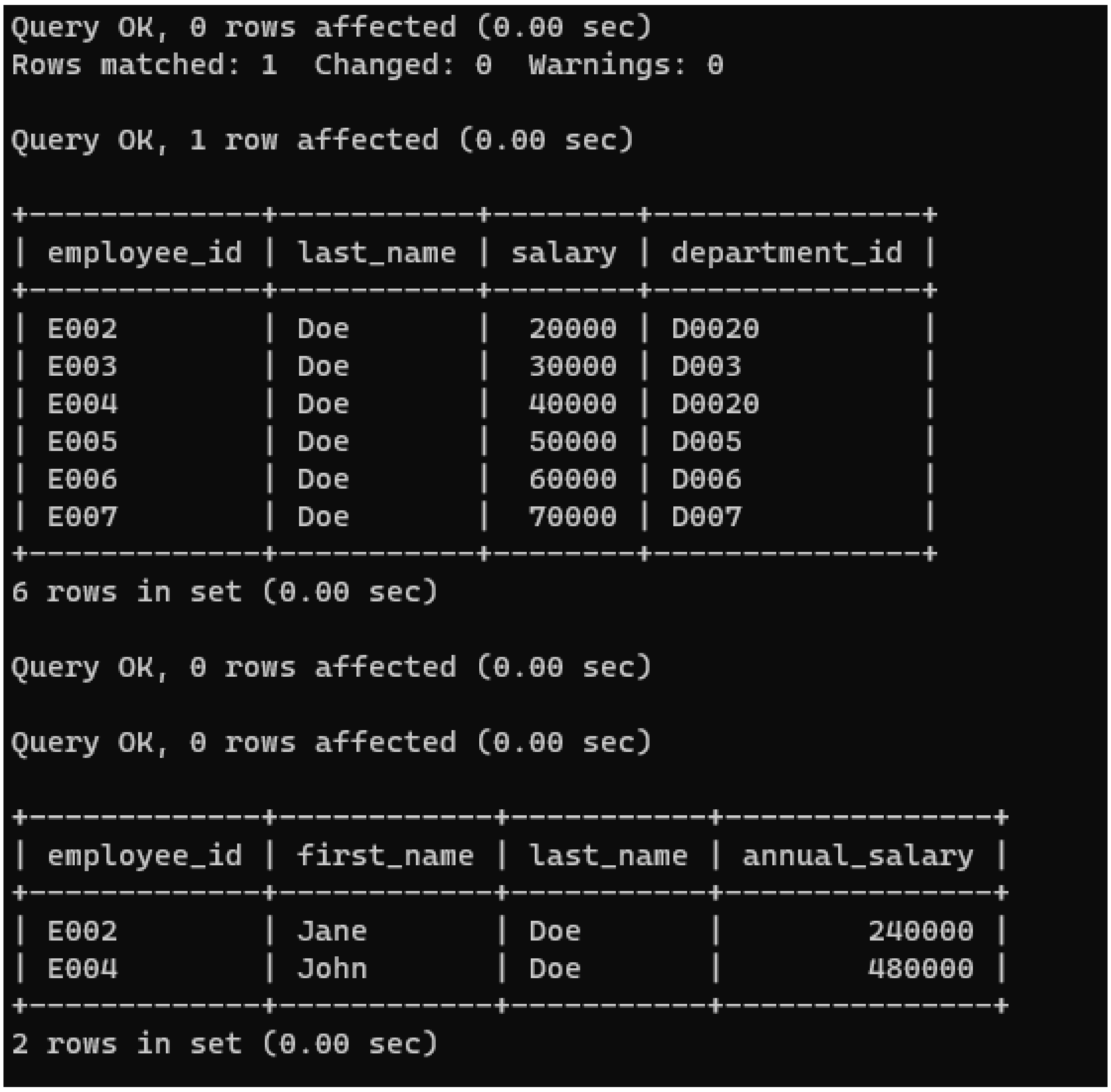
Output:



```
mysql> source D:\Programming\ADBMS\Exp-6.sql
Query OK, 1 row affected (0.00 sec)

Database changed
Query OK, 0 rows affected (0.01 sec)

Query OK, 6 rows affected (0.00 sec)
Records: 6  Duplicates: 0  Warnings: 0

Query OK, 0 rows affected (0.00 sec)

+-------------+-----------+--------+---------------+
| employee_id | last_name | salary | department_id |
+-------------+-----------+--------+---------------+
| E001        | Doe       |  10000 | D001          |
| E002        | Doe       |  20000 | D0020         |
| E003        | Doe       |  30000 | D003          |
| E004        | Doe       |  40000 | D0020         |
| E005        | Doe       |  50000 | D005          |
| E006        | Doe       |  60000 | D006          |
+-------------+-----------+--------+---------------+
6 rows in set (0.00 sec)

Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0
```



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

+---------------+----------+------+-----+---------+-------+
| Field         | Type     | Null | Key | Default | Extra |
+---------------+----------+------+-----+---------+-------+
| employee_id   | char(10) | NO   | PRI | NULL    |       |
| first_name    | char(30) | NO   |     | NULL    |       |
| last_name     | char(30) | YES  |     | NULL    |       |
| dob           | date     | YES  |     | NULL    |       |
| salary        | int      | YES  |     | NULL    |       |
| department_id | char(10) | NO   |     | NULL    |       |
+---------------+----------+------+-----+---------+-------+
6 rows in set (0.00 sec)

Query OK, 1 row affected (0.00 sec)

+-------------+-----------+--------+---------------+
| employee_id | last_name | salary | department_id |
+-------------+-----------+--------+---------------+
| E001        | Doe       |  10000 | D001          |
| E002        | Doe       |  20000 | D0020         |
| E003        | Doe       |  30000 | D003          |
| E004        | Doe       |  40000 | D0020         |
| E005        | Doe       |  50000 | D005          |
| E006        | Doe       |  60000 | D006          |
| E007        | Doe       |  70000 | D007          |
+-------------+-----------+--------+---------------+
7 rows in set (0.00 sec)
```



```
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1  Changed: 0  Warnings: 0

Query OK, 1 row affected (0.00 sec)

+-------------+-----------+--------+---------------+
| employee_id | last_name | salary | department_id |
+-------------+-----------+--------+---------------+
| E002        | Doe       |  20000 | D0020         |
| E003        | Doe       |  30000 | D003          |
| E004        | Doe       |  40000 | D0020         |
| E005        | Doe       |  50000 | D005          |
| E006        | Doe       |  60000 | D006          |
| E007        | Doe       |  70000 | D007          |
+-------------+-----------+--------+---------------+
6 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

+-------------+------------+-----------+---------------+
| employee_id | first_name | last_name | annual_salary |
+-------------+------------+-----------+---------------+
| E002        | Jane       | Doe       |        240000 |
| E004        | John       | Doe       |        480000 |
+-------------+------------+-----------+---------------+
2 rows in set (0.00 sec)
```

Experiment: 7. To understand the concepts of Index.

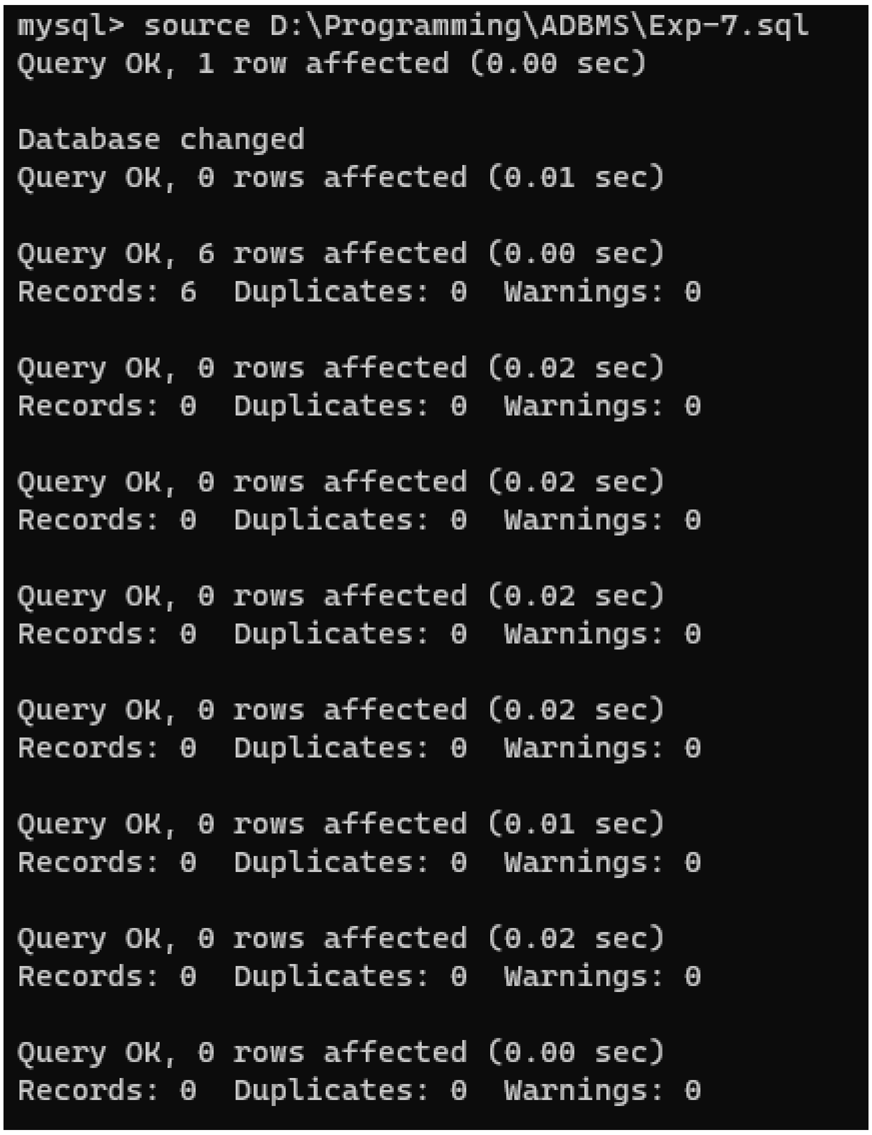
Objective: Students will be able to implement the concept of index.

Queries: GitHub

```sql
-- Experiment: 7. To understand the concepts of Index.
-- Objective: Students will be able to implement the concept of index.
-- Setup Database
CREATE DATABASE exp_7;
USE exp_7;
-- Table
CREATE TABLE IF NOT EXISTS employee(
    employee_id CHAR(10) NOT NULL,
    first_name CHAR(30) NOT NULL,
    last_name CHAR(30) NOT NULL,
    dob DATE,
    salary INT NOT NULL,
    department_id CHAR(10) NOT NULL,
    PRIMARY KEY(employee_id)
);
-- add 6 rows
INSERT INTO employee
VALUES (
        'E001',
        'John',
        'Doe',
        '1990-01-01',
        10000,
        'D001'
    ),
    (
        'E002',
        'Jane',
        'Doe',
        '1991-02-02',
        20000,
        'D0020'
    ),
    (
        'E003',
        'John',
        'Doe',
        '1992-03-03',
        30000,
        'D003'
    ),
    (
        'E004',
        'John',
        'Doe',
        '1993-04-04',
        40000,
        'D0020'
    ),
    (
        'E005',
        'John',
        'Doe',
        '1994-05-05',
        50000,
        'D005'
    ),
    (
        'E006',
        'John',
        'Doe',
        '1995-06-06',
        60000,
        'D006'
    );
-- 1. Execute the following index related queries:
--
1)Create  an  index  of  name  employee_idx  on  EMPLOYEES  with  colu
mn  Last_Name, Department_id
CREATE INDEX employee_idx ON employee (last_name, department_id);
-- 2)Find the ROWID for the above table and create a unique index on
employee_id column of the EMPLOYEES.
CREATE UNIQUE INDEX unique_employee_id ON employee (employee_id);
-- 3)Create a reverse index on employee_id column of the EMPLOYEES.
CREATE INDEX reverse_employee_id ON employee (employee_id DESC);
```

```sql
-- 4)Create a unique and composite index on employee_id and check
whether there is duplicity of tuples or not.
CREATE UNIQUE INDEX unique_employee_id_composite ON employee
(employee_id, last_name);
-- 5)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).
ALTER TABLE employee
ADD COLUMN last_name_lower CHAR(30) GENERATED ALWAYS AS
(LOWER(last_name)) VIRTUAL;
CREATE INDEX employee_lower_last_name ON employee (last_name_lower);
-- 6)Drop the function based index on column Last_Name
DROP INDEX employee_lower_last_name ON employee;
```

Output:



```
mysql> source D:\Programming\ADBMS\Exp-7.sql
Query OK, 1 row affected (0.00 sec)

Database changed
Query OK, 0 rows affected (0.01 sec)

Query OK, 6 rows affected (0.00 sec)
Records: 6  Duplicates: 0  Warnings: 0

Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0

Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

Query OK, 0 rows affected (0.00 sec)
Records: 0  Duplicates: 0  Warnings: 0
```