DBMS – LAB -05

NAME : RAHUL VARMA

ROLL NO: S20200010212

SECTION: C

TASK: (LAB EXERCISES)

SQL COMMANDS

1. FUNCTIONS

2. JOINS

QUESTIONS:

Assignment Questions:

1. Find all the bank customers having a loan, an account or both at the bank
2. Find those customers who are borrowers from the bank and who appear in the list of account holders (i.e present in depositor table)
3. Find all the customers who have loan at the bank ,but do not have an account at the bank
4. Find the names of all branches that have assets greater than those of at least one branch located in Brooklyn (without using subquery)
5. Find the names of all branches that have assets greater than those of at least one branch located in Brooklyn (using subquery)
6. Find the branch that has the highest average balance
7. Find all the customers who have both an account and a loan at the bank,by a subquery using "exists" keyword
8. Perform natural join between tables loan and borrower

9. Perform inner join between tables loan and borrower,with loan_number as joining condition
10. Perform natural right outer join between tables loan and borrower
11. Perform right outer join between tables loan and borrower,with loan_number as joining condition
12. Perform natural left outer join between tables loan and borrower
13. Perform left outer join between tables loan and borrower,with loan_number as joining condition
14. Perform full outer between tables loan and borrower

PRACTISE QUESTION :

1. create table small_customers(id smallint,name varchar(10),age smallint,address varchar(15),salary int);
2. create table small_customers2(id smallint,name varchar(10),age smallint,address varchar(15),salary int);
3. create table orders (oid int,date datetime,customer_id smallint,amount int);
4. LOAD DATA LOCAL INFILE 'small_customers.csv' INTO table small_customers COLUMNS TERMINATED BY ',';
5. LOAD DATA LOCAL INFILE 'orders.csv' INTO table orders COLUMNS TERMINATED BY ',';

Using small_customers and Orders schema
I created small_customers and Orders table and loaded data into it.

```
Database changed
mysql> create table small_customers(id smallint,name varchar(10),age smallint,address varchar(15),salary int);
Query OK, 0 rows affected (0.04 sec)

mysql> create table small_customers2(id smallint,name varchar(10),age smallint,address varchar(15),salary int);
Query OK, 0 rows affected (0.02 sec)

mysql> create table orders (oid int,date datetime,customer_id smallint,amount int);
Query OK, 0 rows affected (0.05 sec)
```

CONTENT IN THE TABLE :

```
mysql> select * from small_customers;
+------+----------+------+-----------+--------+
| id   | name     | age  | address   | salary |
+------+----------+------+-----------+--------+
|    1 | Ramesh   |   35 | Ahmedabad |    125 |
|    2 | Khilan   |   25 | Delhi     |   1500 |
|    3 | kaushik  |   23 | Kota      |   2000 |
|    4 | Chaitali |   25 | Mumbai    |   6500 |
|    5 | Hardik   |   27 | Bhopal    |   2125 |
|    6 | Komal    |   22 | MP        |   4500 |
|    7 | Muffy    |   24 | Indore    |  10000 |
+------+----------+------+-----------+--------+
7 rows in set (0.00 sec)

mysql> select *from orders;
+------+---------------------+-------------+--------+
| oid  | date                | customer_id | amount |
+------+---------------------+-------------+--------+
|  102 | 2009-10-08 00:00:00 |           3 |   3000 |
|  100 | 2009-10-08 00:00:00 |           3 |   1500 |
|  101 | 2009-11-20 00:00:00 |           2 |   1560 |
|  103 | 2008-05-20 00:00:00 |           4 |   2060 |
+------+---------------------+-------------+--------+
4 rows in set (0.00 sec)
```

Using Bank Database Schema:

1.Find all the bank customers having a loan, an account or both at the bank

```
mysql> (select customer_name from depositor)
    -> union
    -> (select customer_name from borrower);
+---------------+
| customer_name |
+---------------+
| Johnson       |
| Hayes         |
| Smith         |
| Jones         |
| Lindsay       |
| Turner        |
| Adams         |
| Williams      |
| Curry         |
+---------------+
9 rows in set (0.01 sec)
```

2. Find those customers who are borrowers from the bank and who appear in the list of account holders (i.e present in depositor table)

```
mysql> select distinct customer_name
    -> from borrower
    -> where customer_name in (select customer_name from depositor);
+---------------+
| customer_name |
+---------------+
| Smith         |
| Hayes         |
| Jones         |
+---------------+
3 rows in set (0.00 sec)
```

3. Find all the customers who have loan at the bank ,but do not have an account at the bank

```
mysql> select distinct customer_name
    -> from borrower
    -> where customer_name not in (select customer_name from depositor);
+---------------+
| customer_name |
+---------------+
| Adams         |
| Williams      |
| Curry         |
+---------------+
3 rows in set (0.00 sec)
```

4.Find the names of all branches that have assets greater than those of at least one branch located in Brooklyn (without using subquery)

```
mysql> select distinct s.branch_name
    -> from branch as s, branch as r
    -> where s.assets > r.assets and r.branch_city = 'Brooklyn';
+-------------+
| branch_name |
+-------------+
| Downtown    |
| Round Hill  |
+-------------+
2 rows in set (0.01 sec)
```

5. Find the names of all branches that have assets greater than those of at least one branch located in Brooklyn (using subquery)

```
mysql> select branch_name
    -> from branch
    -> where assets>some ( select assets from branch where branch_city ='Brooklyn');
+-------------+
| branch_name |
+-------------+
| Downtown    |
| Round Hill  |
+-------------+
2 rows in set (0.00 sec)
```

6. Find the branch that has the highest average balance

```
mysql> select branch_name
    -> from  account
    -> group by branch_name
    -> having avg(balance) >= all (select avg(balance) from account group by branch_name);
+-------------+
| branch_name |
+-------------+
| Brighton    |
+-------------+
1 row in set (0.01 sec)
```

7. Find all the customers who have both an account and a loan at the bank,by a subquery using “exists” keyword

```
mysql> select distinct s.customer_name
    -> from borrower s
    -> where exists (select * from depositor r where r.customer_name = s.customer_name);
+---------------+
| customer_name |
+---------------+
| Smith         |
| Hayes         |
| Jones         |
+---------------+
3 rows in set (0.00 sec)
```

8. Perform natural join between tables loan and borrower

```
mysql> select * from loan natural join borrower;
+-------------+-------------+--------+---------------+
| loan_number | branch_name | amount | customer_name |
+-------------+-------------+--------+---------------+
| L-11        | Round Hill  |    900 | Smith         |
| L-15        | Perryridge  |   1500 | Hayes         |
| L-16        | Perryridge  |   1300 | Adams         |
| L-17        | Downtown    |   1000 | Jones         |
| L-17        | Downtown    |   1000 | Williams      |
| L-23        | Redwood     |   2000 | Smith         |
| L-93        | Mianus      |    500 | Curry         |
+-------------+-------------+--------+---------------+
7 rows in set (0.00 sec)
```

9. Perform inner join between tables loan and borrower,with loan_number as joining condition

```
mysql> select *
    ->  from loan inner join borrower
    -> on loan.loan_number=borrower.loan_number;
+-------------+-------------+--------+---------------+-------------+
| loan_number | branch_name | amount | customer_name | loan_number |
+-------------+-------------+--------+---------------+-------------+
| L-11        | Round Hill  |    900 | Smith         | L-11        |
| L-15        | Perryridge  |   1500 | Hayes         | L-15        |
| L-16        | Perryridge  |   1300 | Adams         | L-16        |
| L-17        | Downtown    |   1000 | Jones         | L-17        |
| L-17        | Downtown    |   1000 | Williams      | L-17        |
| L-23        | Redwood     |   2000 | Smith         | L-23        |
| L-93        | Mianus      |    500 | Curry         | L-93        |
+-------------+-------------+--------+---------------+-------------+
7 rows in set (0.00 sec)
```

10. Perform natural right outer join between tables loan and borrower

```
mysql> select *
    -> from loan natural right outer join borrower;
+-------------+---------------+-------------+--------+
| loan_number | customer_name | branch_name | amount |
+-------------+---------------+-------------+--------+
| L-11        | Smith         | Round Hill  |    900 |
| L-15        | Hayes         | Perryridge  |   1500 |
| L-16        | Adams         | Perryridge  |   1300 |
| L-17        | Jones         | Downtown    |   1000 |
| L-17        | Williams      | Downtown    |   1000 |
| L-23        | Smith         | Redwood     |   2000 |
| L-93        | Curry         | Mianus      |    500 |
+-------------+---------------+-------------+--------+
7 rows in set (0.00 sec)
```

11. Perform right outer join between tables loan and borrower,with loan_number as joining condition

```
mysql> select *
    -> from loan right outer join borrower
    -> on loan.loan_number=borrower.loan_number;
+-------------+-------------+--------+---------------+-------------+
| loan_number | branch_name | amount | customer_name | loan_number |
+-------------+-------------+--------+---------------+-------------+
| L-11        | Round Hill  |    900 | Smith         | L-11        |
| L-15        | Perryridge  |   1500 | Hayes         | L-15        |
| L-16        | Perryridge  |   1300 | Adams         | L-16        |
| L-17        | Downtown    |   1000 | Jones         | L-17        |
| L-17        | Downtown    |   1000 | Williams      | L-17        |
| L-23        | Redwood     |   2000 | Smith         | L-23        |
| L-93        | Mianus      |    500 | Curry         | L-93        |
+-------------+-------------+--------+---------------+-------------+
7 rows in set (0.00 sec)
```

12. Perform natural left outer join between tables loan and borrower

```
mysql> select *
    ->  from loan natural left outer join borrower;
+-------------+-------------+--------+---------------+
| loan_number | branch_name | amount | customer_name |
+-------------+-------------+--------+---------------+
| L-11        | Round Hill  |    900 | Smith         |
| L-14        | Downtown    |   1500 | NULL          |
| L-15        | Perryridge  |   1500 | Hayes         |
| L-16        | Perryridge  |   1300 | Adams         |
| L-17        | Downtown    |   1000 | Jones         |
| L-17        | Downtown    |   1000 | Williams      |
| L-23        | Redwood     |   2000 | Smith         |
| L-93        | Mianus      |    500 | Curry         |
+-------------+-------------+--------+---------------+
8 rows in set (0.00 sec)
```

13. Perform left outer join between tables loan and borrower,with loan_number as joining condition

```
mysql> select *
    -> from loan left outer join borrower
    -> on loan.loan_number=borrower.loan_number;
+-------------+-------------+--------+---------------+-------------+
| loan_number | branch_name | amount | customer_name | loan_number |
+-------------+-------------+--------+---------------+-------------+
| L-11        | Round Hill  |    900 | Smith         | L-11        |
| L-14        | Downtown    |   1500 | NULL          | NULL        |
| L-15        | Perryridge  |   1500 | Hayes         | L-15        |
| L-16        | Perryridge  |   1300 | Adams         | L-16        |
| L-17        | Downtown    |   1000 | Jones         | L-17        |
| L-17        | Downtown    |   1000 | Williams      | L-17        |
| L-23        | Redwood     |   2000 | Smith         | L-23        |
| L-93        | Mianus      |    500 | Curry         | L-93        |
+-------------+-------------+--------+---------------+-------------+
8 rows in set (0.00 sec)
```

14. Perform full outer between tables loan and borrower

```
mysql> select *
    ->  from loan left join borrower
    -> on loan.loan_number=borrower.loan_number
    -> union all
    -> select *
    -> from loan right join borrower
    -> on loan.loan_number=borrower.loan_number where loan.loan_number is NULL;
+-------------+-------------+--------+---------------+-------------+
| loan_number | branch_name | amount | customer_name | loan_number |
+-------------+-------------+--------+---------------+-------------+
| L-11        | Round Hill  |    900 | Smith         | L-11        |
| L-14        | Downtown    |   1500 | NULL          | NULL        |
| L-15        | Perryridge  |   1500 | Hayes         | L-15        |
| L-16        | Perryridge  |   1300 | Adams         | L-16        |
| L-17        | Downtown    |   1000 | Jones         | L-17        |
| L-17        | Downtown    |   1000 | Williams      | L-17        |
| L-23        | Redwood     |   2000 | Smith         | L-23        |
| L-93        | Mianus      |    500 | Curry         | L-93        |
+-------------+-------------+--------+---------------+-------------+
8 rows in set (0.00 sec)
```

THANK YOU