Department of Computer Science and Engineering

Lab 03 : Introduction to Bank DB, SQL Joins and Constraints

Name: Nabonita Saha

Student ID : 22301645

LAB Section: 08

Theory Section: 11

Create the "Bank" database and then create all necessary tables below

```
CREATE DATABASE Bank_22301645;
USE Bank_22301645;

create table customer (
customer_id varchar(10) not null,
customer_name varchar(20) not null,
customer_street varchar(30),
customer_city varchar(30),
primary key (customer_id));

create table branch (
branch_name varchar(15),
branch_city varchar(30),
assets int,
primary key (branch_name),
check (assets >= 0));

create table account (
branch_name varchar(15),
account_number varchar(10) not null,
balance int,
primary key (account_number),
check (balance >= 0));

create table loan (
loan_number varchar(10) not null,
branch_name varchar(15),
```

```
amount int,
primary key (loan_number));

create table depositor (
customer_id varchar(10) not null,
account_number varchar(10) not null,
primary key (customer_id,account_number),
foreign key (customer_id) references customer(customer_id),
foreign key (account_number) references account(account_number));

create table borrower (
customer_id varchar(10) not null,
loan_number varchar(10) not null,
primary key (customer_id, loan_number),
foreign key (customer_id) references customer(customer_id),
foreign key (loan_number) references loan(loan_number));
```

Task 2

Once all your tables have been created, you should insert the data below. The insertion code has been provided for you. After insertion, check that data has been correctly inserted in all tables using the "Select" query

```
insert into customer values
('C-101','Jones', 'Main', 'Harrison'),
('C-201','Smith', 'North', 'Rye'),
('C-211','Hayes', 'Main', 'Harrison'),
('C-212','Curry', 'North', 'Rye'),
('C-215','Lindsay', 'Park', 'Pittsfield'),
('C-220','Turner', 'Putnam', 'Stamford'),
('C-222','Williams', 'Nassau', 'Princeton'),
('C-225','Adams', 'Spring', 'Pittsfield'),
('C-226','Johnson', 'Alma', 'Palo Alto'),
('C-233','Glenn', 'Sand Hill', 'Woodside'),
('C-234','Brooks', 'Senator', 'Brooklyn'),
('C-255','Green', 'Walnut', 'Stamford');

insert into branch values
('Downtown', 'Brooklyn',9000000),
('Redwood', 'Palo Alto',2100000),
('Perryridge', 'Horseneck',1700000),
('Mianus', 'Horseneck',400000),
('Round Hill', 'Horseneck',8000000),
```

('Pownal', 'Bennington',300000),
('North Town', 'Rye',3700000),
('Brighton', 'Brooklyn',7100000);

insert into account values
('Downtown','A-101',500),
('Mianus','A-215',700) ,
('Perryridge','A-102',400),
('Round Hill','A-305',350),
('Brighton','A-201',900),
('Redwood','A-222',700),
('Brighton','A-217',750);

insert into loan values
('L-17', 'Downtown', 1000),
('L-23', 'Redwood', 2000),
('L-15', 'Perryridge', 1500),
('L-14', 'Downtown', 1500),
('L-93', 'Mianus', 500),
('L-11', 'Round Hill', 900),
('L-16', 'Perryridge', 1300);

insert into depositor values
('C-226', 'A-101'),
('C-201', 'A-215'),
('C-211', 'A-102'),
('C-220', 'A-305'),
('C-226', 'A-201'),
('C-101', 'A-217'),
('C-215', 'A-222');

insert into borrower values
('C-101', 'L-17'),
('C-201', 'L-23'),
('C-211', 'L-15'),
('C-226', 'L-14'),
('C-212', 'L-93'),
('C-201', 'L-11'),
('C-222', 'L-17'),
('C-225', 'L-16');

Assignment Tasks :

TASK 01:

Find the name and loan number of all customers having a loan at the Downtown branch. [2]

MariaDB [Bank_22301645]> select t1.customer_name, t3.loan_number from customer t1, borrower t2, loan t3 where t1.customer_id=t2.customer_id and t2.loan_number=t3.loan_number and t3.branch_name="downtown";

```
MariaDB [Bank_22301645]> select t1.customer_name, t3.loan_number from customer t1, borrower t2, loan t3 where t1.custome
r_id=t2.customer_id and t2.loan_number=t3.loan_number and t3.branch_name="downtown";
+---------------+-------------+
| customer_name | loan_number |
+---------------+-------------+
| Johnson       | L-14        |
| Jones         | L-17        |
| Williams      | L-17        |
+---------------+-------------+
3 rows in set (0.001 sec)
```

TASK 02:

Find all the possible pairs of customers who are from the same city. show in the format Customer1, Customer2, City. [2]

MariaDB [Bank_22301645]> select t1.customer_name as Customer1, t2.customer_name as Customer2 , t1.customer_city as city from customer t1, customer t2 where t1.customer_city=t2.customer_city and t1.customer_name!=t2.customer_name group by t1.customer_city ;

```
MariaDB [Bank_22301645]> select t1.customer_name as Customer1, t2.customer_name as Customer2 , t1.customer_city as city
from customer t1, customer t2 where t1.customer_city=t2.customer_city and t1.customer_name!=t2.customer_name group by t1
.customer_city ;
+-----------+-----------+------------+
| Customer1 | Customer2 | city       |
+-----------+-----------+------------+
| Hayes     | Jones     | Harrison   |
| Adams     | Lindsay   | Pittsfield |
| Curry     | Smith     | Rye        |
| Green     | Turner    | Stamford   |
+-----------+-----------+------------+
4 rows in set (0.001 sec)
```

TASK 03:

If the bank gives out 4% interest to all accounts, show the total interest across each branch. Print Branch_name, Total_Interest [2]

MariaDB [Bank_22301645]> select branch_name , sum(balance* 0.04) as total_interest from account group by branch_name;

```
MariaDB [Bank_22301645]> select branch_name , sum(balance* 0.04 ) as total_interest from account group by branch_name;
+-------------+----------------+
| branch_name | total_interest |
+-------------+----------------+
| Brighton    |          66.00 |
| Downtown    |          20.00 |
| Mianus      |          28.00 |
| Perryridge  |          16.00 |
| Redwood     |          28.00 |
| Round Hill  |          14.00 |
+-------------+----------------+
6 rows in set (0.001 sec)
```

TASK 04:

Find account numbers with the highest balances for each city in the database [2]

MariaDB [Bank_22301645]> select t3.account_number from customer t1, depositor t2, account t3 where t1.customer_id=t2.customer_id and t2.account_number=t3.account_number and t3.balance=(select max(ac.balance) from account ac, customer cu, depositor de where ac.account_number=de.account_number and cu.customer_id=de.customer_id and t1.customer_city=cu.customer_city);

```
MariaDB [Bank_22301645]> select t3.account_number from customer t1, depositor t2, account t3 where t1.customer_id=t2.cus
tomer_id and t2.account_number=t3.account_number and t3.balance=( select max(ac.balance) from account ac, customer cu, d
epositor de where ac.account_number=de.account_number and cu.customer_id=de.customer_id and t1.customer_city=cu.customer
_city);
+----------------+
| account_number |
+----------------+
| A-217          |
| A-215          |
| A-222          |
| A-305          |
| A-201          |
+----------------+
5 rows in set (0.001 sec)
```

TASK 05:

Show the loan number, loan amount, and name of customers with the top 5 highest loan amounts. The data should be sorted by increasing amounts, then decreasing loan numbers in case of the same loan amount. [Hint for top 5: Check the "limit" keyword in mysql] [2]

MariaDB [Bank_22301645]> select loan_number, loan_amount, customer_name from (select l1.loan_number, l1.amount as loan_amount, c1.customer_name from customer c1, loan l1, borrower b1 where c1.customer_id=b1.customer_id and b1.loan_number=l1.loan_number order by l1.amount desc , l1.loan_number desc limit 5) as subquery order by loan_amount asc, loan_number desc;

```
MariaDB [Bank_22301645]> select loan_number, loan_amount, customer_name from ( select l1.loan_number, l1.amount as loan_
amount, c1.customer_name from customer c1, loan l1, borrower b1 where c1.customer_id=b1.customer_id and b1.loan_number=l
1.loan_number order by l1.amount desc , l1.loan_number desc limit 5 ) as subquery order by loan_amount asc, loan_number
desc;
+-------------+-------------+---------------+
| loan_number | loan_amount | customer_name |
+-------------+-------------+---------------+
| L-17        |        1000 | Jones         |
| L-16        |        1300 | Adams         |
| L-15        |        1500 | Hayes         |
| L-14        |        1500 | Johnson       |
| L-23        |        2000 | Smith         |
+-------------+-------------+---------------+
5 rows in set (0.001 sec)
```

TASK 06:

Find the names of customers with an account and also a loan at the Perryridge branch. [2]

MariaDB [Bank_22301645]> select t1.customer_name from customer t1, borrower t2, loan t3 , depositor t4, account t5 where t1.customer_id=t2.customer_id and t2.loan_number=t3.loan_number and t1.customer_id=t4.customer_id and t4.account_number=t5.account_number and t3.branch_name="perryridge" and t5.branch_name="perryridge";

```
MariaDB [Bank_22301645]> select t1.customer_name from customer t1, borrower t2, loan t3 , depositor t4, account t5 where
 t1.customer_id=t2.customer_id and t2.loan_number=t3.loan_number and t1.customer_id=t4.customer_id and t4.account_number
=t5.account_number and t3.branch_name="perryridge" and t5.branch_name="perryridge";
+---------------+
| customer_name |
+---------------+
| Hayes         |
+---------------+
1 row in set (0.000 sec)
```

TASK 07:

Find the total loan amount of all customers having at least 2 loans from the bank. Show in format customer name, total_loan. [2]

MariaDB [Bank_22301645]> select t1.customer_name, sum(t3.amount) as total_loan from customer t1, borrower t2, loan t3 where t1.customer_id=t2.customer_id and t3.loan_number=t2.loan_number group by t1.customer_name having count(t2.loan_number)>=2;

```
MariaDB [Bank_22301645]> select t1.customer_name, sum(t3.amount) as total_loan  from customer t1, borrower t2, loan t3 w
here t1.customer_id=t2.customer_id and t3.loan_number=t2.loan_number group by t1.customer_name having count(t2.loan_numb
er)>=2;
+---------------+------------+
| customer_name | total_loan |
+---------------+------------+
| Smith         |       2900 |
+---------------+------------+
1 row in set (0.001 sec)
```