

DBMS QUERIES

1) Create Database “Bank” and display the list of database

```
create database bank;  
show databases;
```

```
MariaDB [(none)]> create database bank;  
Query OK, 1 row affected (0.001 sec)  
  
MariaDB [(none)]> show databases;  
+-----+  
| Database |  
+-----+  
| bank |  
| database |  
| information_schema |  
| mysql |  
| performance_schema |  
| sys |  
| university |  
+-----+  
7 rows in set (0.041 sec)
```

2) Drop database “Bank”

```
drop database bank;
```

```
MariaDB [(none)]> drop database bank;  
Query OK, 0 rows affected (0.084 sec)  
  
MariaDB [(none)]> show databases;  
+-----+  
| Database |  
+-----+  
| database |  
| information_schema |  
| mysql |  
| performance_schema |  
| sys |  
| university |  
+-----+  
6 rows in set (0.001 sec)
```

3) Create tables of given schema

- branch(branch_number, branch_name, branch_city, assets)
- customer(customer_number, customer_name, customer_street, customer_city)
- loan(loan_number, branch_number, amount)
- borrower(customer_number, loan_number)
- account(account_number, branch_number, balance)
- depositor(customer_number, account_number)

```
-- Branch
create table branch(
branch_number int unique,
branch_name varchar(20),
branch_city varchar(20),
assets varchar(10),
primary key(branch_number,branch_name)) ;

-- Customer
create table customer(
customer_number int unique,
customer_name varchar(25),
customer_street varchar(20),
customer_city varchar(15),
primary key(customer_name, customer_number));

-- Loan
CREATE TABLE loan (
loan_number bigint unique,
branch_number int,
amount int,
PRIMARY KEY (loan_number),
constraint fk_const foreign key(branch_number)references
branch(branch_number) on delete cascade on update cascade);
```

```

-- Borrower
create table borrower(
customer_number int,
loan_number bigint,
constraint fk_const5 foreign key(customer_number) references
customer(customer_number) on delete cascade on update cascade,
constraint fk_loan4 foreign key(loan_number) references
loan(loan_number)
on delete cascade on update cascade);

-- Account
create table account(
account_number int unique,
branch_number int,
balance decimal(10,2),
primary key(account_number),
constraint fk_cons foreign key(branch_number)references
branch(branch_number) on delete cascade on update cascade);

-- Depositor
create table depositor(
customer_number int not null,
account_number int not null,
constraint fk_customer foreign key(customer_number)references
customer(customer_number) on delete cascade on update cascade,
constraint fk_acc foreign key(account_number) references
account(account_number)on delete cascade on update cascade);

```

```

MariaDB [bank]> describe branch; These lab tasks will guide you to prepare for SQL CRUD operations. You have to submit lab report of pre-defined tasks. Your lab report to be submitted should contain snapshot of your lab tasks.
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| branch_number | int(11) | NO | PRI | NULL |
| branch_name | varchar(20) | NO | PRI | NULL |
| branch_city | varchar(20) | YES | MUL | NULL |
| assets | varchar(10) | YES | MUL | NULL |
+-----+-----+-----+-----+
4 rows in set (0.002 sec)

3. Objective(s)
MariaDB [bank]> describe customer; 4. Related Theory
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| customer_number | int(11) | NO | UNI | NULL |
| customer_name | varchar(25) | NO | PRI | NULL |
| customer_street | varchar(20) | YES | MUL | NULL |
| customer_city | varchar(15) | YES | MUL | NULL |
+-----+-----+-----+-----+
4 rows in set (0.002 sec)

NOTE: Lab report should be Hand-written (not printed)

MariaDB [bank]> describe loan; Write SQL to solve the following queries. All students have to insert at least one record in each table. The tabulated data should be unique. SQL Server or anything else, it is up to you to implement the task.
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| loan_number | bigint(20) | NO | PRI | NULL |
| branch_number | int(11) | YES | MUL | NULL |
| amount | int(11) | YES | MUL | NULL |
+-----+-----+-----+-----+
3 rows in set (0.001 sec)

Objective: To be familiar with Data Definition Language [ ] corresponding next lab. Your lab report to be submitted should contain snapshot of your lab tasks.

MariaDB [bank]> describe borrower;
+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+
| customer_number | int(11) | YES | MUL | NULL |
| loan_number | bigint(20) | YES | MUL | NULL |
+-----+-----+-----+-----+
2 rows in set (0.002 sec)

4. Related Theory
MariaDB [bank]> describe account; 5. List all the records of the table
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| account_number | int(11) | NO | PRI | NULL |
| branch_number | int(11) | YES | MUL | NULL |
| balance | decimal(10,2) | YES | MUL | NULL |
+-----+-----+-----+-----+
3 rows in set (0.002 sec)

Write SQL to solve the following queries. All students have to insert at least one record in each relation of the given schema. The tabulated data should be unique. SQL Server or anything else, it is up to you to implement the task.
MariaDB [bank]> describe depositor;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| customer_number | int(11) | NO | MUL | NULL |
| account_number | int(11) | NO | MUL | NULL |
+-----+-----+-----+-----+
2 rows in set (0.002 sec)

Objective: To be familiar with Data Definition Language [ ]

```

4) List all the tables

```
show tables;
```

```
MariaDB [bank]>
MariaDB [bank]>
MariaDB [bank]> show tables;
+-----+
| Tables_in_bank |
+-----+
| account        |
| borrower       |
| branch         |
| customer       |
| depositor      |
| loan           |
+-----+
6 rows in set (0.001 sec)
```

- 5) List all the
- 6) Add column
- branch tab
- 7) Drop column

5) List all the columns of “branch” and “customer” tables

```
show columns from branch;
describe customer;
```

6) Add column “Branch_manger” in branch table and list all the columns of branch table

```
alter table branch add branch_manager varchar(20);
show columns from branch;
```

```
MariaDB [bank]> alter table branch add branch_manager varchar(20);
Query OK, 0 rows affected (0.410 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [bank]> describe branch;
+-----+-----+-----+-----+-----+
| Field      | Type       | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| branch_number | int(11)    | NO   | PRI | NULL    |       |
| branch_name   | varchar(20) | NO   | PRI | NULL    |       |
| branch_city    | varchar(20) | YES  |     | NULL    |       |
| assets        | varchar(10) | YES  |     | NULL    |       |
| branch_manager | varchar(20) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
5 rows in set (0.002 sec)
```

7) Drop column “Branch_manger” from branch table and list all the columns of branch table

```
alter table branch drop branch_manager;
describe branch;
```

```
MariaDB [bank]> alter table branch drop branch_manager;
Query OK, 0 rows affected (0.315 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [bank]> describe branch
-> ;
+-----+-----+-----+-----+-----+
| Field      | Type       | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| branch_number | int(11) | NO   | PRI | NULL    |       |
| branch_name   | varchar(20) | NO  | PRI | NULL    |       |
| branch_city   | varchar(20) | YES  |     | NULL    |       |
| assets        | varchar(10) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
4 rows in set (0.002 sec)
```

8) Change the data type of a column “balance” from decimal (12,2) to int in account table.

```
alter table account modify column balance int;
describe account;
```

```
MariaDB [bank]> describe account;
+-----+-----+-----+-----+-----+
| Field      | Type       | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| account_number | int(11) | NO   | PRI | NULL    |       |
| branch_number | int(11) | NO   | MUL | NULL    |       |
| branch_name   | varchar(20) | NO  |     | NULL    |       |
| balance        | decimal(12,2) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
4 rows in set (0.027 sec)

MariaDB [bank]> alter table account modify column balance int;
Query OK, 0 rows affected (0.302 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [bank]> describe account;
+-----+-----+-----+-----+-----+
| Field      | Type       | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| account_number | int(11) | NO   | PRI | NULL    |       |
| branch_number | int(11) | NO   | MUL | NULL    |       |
| branch_name   | varchar(20) | NO  |     | NULL    |       |
| balance        | int(11) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
4 rows in set (0.030 sec)
```

9) Modify the length of data type “customer_name” from varchar (30) to varchar (50) in a customer table.

```
alter table customer modify column customer_name varchar(50);
```

```
MariaDB [bank]> describe customer;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| customer_name | varchar(25) | NO   | PRI  | NULL    |       |
| customer_street | varchar(20) | YES  |      | NULL    |       |
| customer_city | varchar(15) | YES  |      | NULL    |       |
+-----+-----+-----+-----+-----+
3 rows in set (0.001 sec)

MariaDB [bank]> alter table customer modify column customer_name varchar(50);
Query OK, 0 rows affected (0.192 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [bank]>
MariaDB [bank]> describe customer;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| customer_name | varchar(50) | NO   | PRI  | NULL    |       |
| customer_street | varchar(20) | YES  |      | NULL    |       |
| customer_city | varchar(15) | YES  |      | NULL    |       |
+-----+-----+-----+-----+-----+
3 rows in set (0.001 sec)
```

- 7) Drop column
branch table
- 8) Change the a
account table
- 9) Modify the l
varchar (50)
- 10) Add a NOT
table.
- 11) Add default
list the colum
- 12) Drop the de
columns of a
- 13) Add Check
100,00,000 a
- 14) Drop the ch
- 15) Add Primary
on relational
- 16) Drop the pri

10) Add a NOT NULL constraint in the column “branch_city” of the branch table.

```
alter table branch modify column branch_name varchar(20) not null;
```

```
ERROR 1064 (42000): Unknown data type: not
MariaDB [bank]>
MariaDB [bank]> alter table branch modify column branch_name varchar(20) not null;
Query OK, 0 rows affected (0.475 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

**11) Add default constraint in “account” table and set default balance is 1000.
And list the columns of account table.**

```
alter table account alter balance set default 1000;
```

```
MariaDB [bank]>
MariaDB [bank]> alter table account alter balance set default 1000;
Query OK, 0 rows affected (0.165 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [bank]> describe account;
+-----+-----+-----+-----+-----+
| Field      | Type       | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| account_number | int(11) | NO   | PRI | NULL    | locations |
| branch_number | int(11) | NO   | MUL | NULL    |             |
| branch_name   | varchar(20) | NO  |     | NULL    |             |
| balance       | int(11) | YES  |     | 1000    |             |
+-----+-----+-----+-----+-----+
4 rows in set (0.002 sec)
```

12) Drop the default constraint of balance from the account table and list the columns of account table.

```
alter table account alter balance drop default;
```

```
MariaDB [bank]> alter table account alter balance drop default;
Query OK, 0 rows affected (0.292 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [bank]> describe account;
+-----+-----+-----+-----+-----+
| Field      | Type       | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| account_number | int(11)    | NO   | PRI | NULL    |       |
| branch_number  | int(11)    | NO   | MUL | NULL    |       |
| branch_name    | varchar(20) | NO   |      | NULL    |       |
| balance        | int(11)    | YES  |      | NULL    |       |
+-----+-----+-----+-----+-----+
4 rows in set (0.002 sec)
```

13) Add Check constraint in “branch” Table assets must be greater than 100,00,000 and department city must be either KTM or PKR or BRT.

```
alter table branch add constraint check_asset_city check
(assets>10000000 and branch_city in ('KTM', 'PKR', 'BRT'));
```

```
MariaDB [bank]> alter table branch
-> add constraint check_asset_city
-> check (assets>10000000 and branch_city in ('KTM', 'PKR', 'BRT'));
Query OK, 0 rows affected (0.491 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [bank]> describe branch
->
-> ;
+-----+-----+-----+-----+-----+
| Field      | Type       | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| branch_number | int(11)    | NO   | PRI | NULL    |       |
| branch_name  | varchar(20) | NO   | PRI | NULL    |       |
| assets        | int(11)    | YES  |      | NULL    |       |
| branch_city   | varchar(10) | YES  |      | NULL    |       |
+-----+-----+-----+-----+
4 rows in set (0.012 sec)
```

14) Drop the check constraints from the branch table.

```
alter table branch drop constraint check_asset_city;
```

```
MariaDB [bank]> drop constraint check_asset_city;
MariaDB [bank]> alter table branch drop constraint check_asset_city;
Query OK, 0 rows affected (0.174 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [bank]> branch
```

15) Add Primary Key constraint in table branch, customer, loan and account based on relational schema. List the columns of all tables.

```
alter table branch add primary key(branch_number);
alter table customer add primary key(customer_name);
alter table loan add primary key(loan_number);
alter table account add primary key(account_number);
```

```
Key(customer_name), at line 1
MariaDB [bank]> alter table customer add primary key(customer_name);
Query OK, 0 rows affected (0.242 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [bank]> describe customer;
+-----+-----+-----+-----+-----+
| Field      | Type       | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| customer_name | varchar(50) | NO  | PRI | NULL    |       |
| customer_street | varchar(20) | YES |     | NULL    |       |
| customer_city   | varchar(15) | YES |     | NULL    |       |
+-----+-----+-----+-----+-----+
3 rows in set (0.002 sec)
```

```
MariaDB [bank]> alter table loan add primary key(loan_number);
Query OK, 0 rows affected (0.265 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
MariaDB [bank]> describe loan;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| loan_number | int(11) | NO | PRI | NULL | |
| branch_number | int(11) | NO | MUL | NULL | |
| branch_name | varchar(20) | YES | | NULL | |
| amount | int(11) | YES | | NULL | |
+-----+-----+-----+-----+-----+
4 rows in set (0.002 sec)
```

```
MariaDB [bank]> alter table account add primary key(account_number);
Query OK, 0 rows affected (0.275 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
MariaDB [bank]> describe borrower;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| customer_name | varchar(25) | NO | PRI | NULL | |
| loan_number | int(11) | NO | PRI | NULL | |
+-----+-----+-----+-----+-----+
2 rows in set (0.002 sec)
```

```
MariaDB [bank]> describe account;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| account_number | int(11) | NO | PRI | NULL | |
| branch_number | int(11) | NO | MUL | NULL | |
| branch_name | varchar(20) | NO | | NULL | |
| balance | int(11) | YES | | NULL | |
+-----+-----+-----+-----+
4 rows in set (0.002 sec)
```

16) Drop the primary key from the account table. List the columns of account table.

```
alter table account drop primary key;
```

```
MariaDB [bank]> alter table account drop primary key;
Query OK, 0 rows affected (0.292 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
MariaDB [bank]> describe account
```

```
-> ;
+-----+-----+-----+-----+-----+
| Field      | Type       | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| account_number | int(11)    | NO   |     | NULL    |       |
| branch_number | int(11)    | NO   | MUL | NULL    |       |
| branch_name   | varchar(20) | NO   |     | NULL    |       |
| balance       | int(11)    | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
4 rows in set (0.050 sec)
```

17) Add the foreign key constraints on the borrower and depositor table.

```
alter table borrower add constraint fk_const foreign
key(customer_name) references customer(customer_name);
alter table borrower add constraint fk_const1 foreign key(loan_number)
references loan(loan_number);
alter table depositor add constraint fk_acc_const foreign
key(account_number) references account(account_number);
alter table depositor add constraint cons foreign key(customer_name)
references customer(customer_name);
```

```
MariaDB [bank]> alter table borrower add constraint fk_const foreign key(customer_name)
      -> references customer(customer_name);
Query OK, 0 rows affected (0.159 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
MariaDB [bank]> alter table borrower add constraint fk_const foreign key(loan_number) ref
      erences loan(loan_number);
ERROR 1826 (HY000): Duplicate FOREIGN KEY constraint name 'bank/fk_const'
MariaDB [bank]> alter table borrower add constraint fk_const1 foreign key(loan_number) re
      ferences loan(loan_number);
Query OK, 0 rows affected (0.178 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
MariaDB [bank]> describe borrower
```

```
-> ;
+-----+-----+-----+-----+
| Field      | Type       | Null | Key | Default | Extra |
+-----+-----+-----+-----+
| customer_name | varchar(50) | NO   | PRI  | NULL    |       |
| loan_number   | int(11)     | NO   | PRI  | NULL    |       |
+-----+-----+-----+-----+
2 rows in set (0.002 sec)
```

```
MariaDB [bank]>
MariaDB [bank]>
MariaDB [bank]> alter table depositor
      -> add constraint cons foreign key(customer_name) references customer(customer_name);
Query OK, 0 rows affected (0.168 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
MariaDB [bank]> alter table depositor add constraint fk_acc_const foreign key(account_num
      ber) references account(account_number);
Query OK, 0 rows affected (0.167 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
MariaDB [bank]> describe depositor;
+-----+-----+-----+-----+
| Field      | Type       | Null | Key | Default | Extra |
+-----+-----+-----+-----+
| customer_name | varchar(25) | NO   | PRI  | NULL    |       |
| account_number | int(11)     | NO   | PRI  | NULL    |       |
+-----+-----+-----+-----+
2 rows in set (0.002 sec)
```

NOTE: Lab report should be Handwritten.

Write SQL to solve the following queries. All the queries must be handwritten in each relation of the given schema. The tabular representation of the relations is given below.

You can use MySQL/SQL Server or any other database system. The SQL must be screen shot or clipped part of your lab report.

Objective: To be familiar with Data Definition Language.

18) Drop the foreign key constraints from depositor table.

```
alter table depositor drop constraint fk_acc;
alter table depositor drop constraint fk_acc_const;
alter table depositor drop constraint fk_cus1;
```

```
MariaDB [bank]> alter table depositor drop constraint fk_acc;
Query OK, 0 rows affected (0.245 sec)
Records: 0  Duplicates: 0  Warnings: 0

Affected: 0  Duration: 0.167 sec

MariaDB [bank]> alter table depositor drop constraint fk_acc_const;
Query OK, 0 rows affected (0.174 sec)
Records: 0  Duplicates: 0  Warnings: 0

Affected: 0  Duration: 0.170 sec

MariaDB [bank]> alter table depositor drop constraint fk_cus1;
Query OK, 0 rows affected (0.170 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

19) Add unique constraints on “customer_name” column in the customer table.

```
alter table customer add constraint const2 unique(customer_name);
```

```
MariaDB [bank]> alter table customer add constraint const2 unique(customer_name);
Query OK, 0 rows affected (0.233 sec)
Records: 0  Duplicates: 0  Warnings: 0

Affected: 0  Duration: 0.161 sec

MariaDB [bank]> describe customer;
+-----+-----+-----+-----+-----+
| Field      | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| customer_name | varchar(50) | NO   | PRI | NULL    |       |
| customer_street | varchar(20) | YES  |     | NULL    |       |
| customer_city  | varchar(15) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
```

20)Drop the unique constraints from the customer table.

```
alter table customer drop index const2;
```

```
MariaDB [bank]> alter table customer drop index const2;
Query OK, 0 rows affected (0.166 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

21) Add primary key constraint in the account table. And also add the foreign key constraints on the depositor table

```
alter table depositor add constraint fkey_cons1 foreign
key(customer_name) references customer(customer_name);
alter table depositor add constraint fkey_cons2 foreign
key(account_number) references account(account_number);
```

```
MariaDB [bank]>
MariaDB [bank]> alter table account add constraint pkey_acc primary key(account_
number);
Query OK, 0 rows affected, 1 warning (0.260 sec)
Records: 0  Duplicates: 0  Warnings: 1 "branch" and "customer" tables
6) Add column "Branch_manger" in branch table and list all the co
MariaDB [bank]> describe account;
+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+
| account_number | int(11) | NO | PRI | NULL |
| branch_name | varchar(20) | NO | MUL | NULL |
| balance | decimal(12,2) | YES | | NULL |
+-----+-----+-----+-----+
3 rows in set (0.002 sec)

MariaDB [bank]> Add column "Branch_manger" in branch table and list all the co
MariaDB [bank]> alter table depositor add constraint fkey_cons1 foreign key(cust
omer_name) references customer(customer_name);
Query OK, 0 rows affected (0.297 sec)
Records: 0  Duplicates: 0  Warnings: 0
MariaDB [bank]> alter table depositor add constraint fkey_cons2 foreign key(acco
unt_number) references account(account_number);
Query OK, 0 rows affected (0.305 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

22) Insert at least 10 records in branch, customer, account, loan, borrower and depositor relations.

```
-- Branch
insert into branch values(6345, 'Shivam Tol', 'Kathmandu', 5000000);
insert into branch values(6344, 'Bagmara Marg', 'Kathmandu', 4500000);
insert into branch values(6325, 'Baneshwor', 'Kathmandu', 500000);
insert into branch values(8345, 'Bagar', 'Pokhara', 5000000);
insert into branch values(8245, 'Sabhagriha Chowk', 'Pokhara', 4000000);
insert into branch values(8145, 'Tinkune', 'Pokhara', 5340000);
insert into branch values(8344, 'Lakeside', 'Pokhara', 2000000);
insert into branch values(5345, 'Ichchhakamana', 'Narayanghat', 990000);
insert into branch values(5340, 'Sahid Chowk', 'Narayanghat', 1000000);
insert into branch values(0345, 'Maharshi Chowk', 'Dhamauli', 2000000);
insert into branch values(4537, 'Shukra Path', 'Biratnagar', 500000);
insert into branch values(2537, 'Bhrikuti Marg', 'Dharan', 1200000);
insert into branch values(4345, 'Sangam Chowk', 'Hetauda', 1000000);
insert into branch values(2345, 'Saptagandaki
Chowk', 'Bharatpur', 750000);
insert into branch values(3425, 'Ram Chowk', 'Janakput', 400000);
insert into branch values(2435, 'Kupandol', 'Lalitpur', 2500000);
insert into branch values(2430, 'Nakidot', 'Lalitpur', 2509000);
insert into branch values(4325, 'Birendra Path', 'Hetauda', 3000000);
insert into branch values(4320, 'School Road', 'Hetauda', 1010000);
insert into branch values(4324, 'Sital Mall', 'Hetauda', 1010000);
insert into branch values(2325, 'Amarsingh Chowk', 'Pokhara', 1200000);
insert into branch values(4343, 'Hospital Chowk', 'Pokhara', 1209900);

-- Customer
insert into customer values(1, 'Shyam Sharma', 'Shivam
Tol', 'Kathmandu');
insert into customer values(2, 'Hari Parsad Baral', 'Bagmara
Marg', 'Kathmandu');
insert into customer values(3, 'Jhon Cordoba', 'Baneshwor', 'Kathmandu');
insert into customer values(4, 'Wyane Rooney', 'Kupandol', 'Lalitpur');
insert into customer values(5, 'Ross Smith', 'Nakidot', 'Lalitpur');
insert into customer values(6, 'Ted Wilson', 'Bagar', 'Pokhara');
insert into customer values(7, 'Prajwol Khan', 'Sabhagriha
Chowk', 'Pokhara');
insert into customer values(8, 'Aagman Poudel', 'Tinkune', 'Pokhara');
insert into customer values(9, 'Sanskriti Giri', 'Lakeside', 'Pokhara');
```

```

insert into customer values(10, 'Roxane
Chalise', 'Ichchhakamana', 'Narayanghat');
insert into customer values(11, 'Riva Chalise', 'Sahid
Chowk', 'Narayanghat');
insert into customer values(12, 'Bhawana Baral', 'Maharshi
Chowk', 'Dhamauli');
insert into customer values(13, 'Rena Baral', 'Shukra
Path', 'Biratnagar');
insert into customer values(14, 'Robin Jhonson', 'Bhrikuti
Marg', 'Dharan');
insert into customer values(15, 'Barney Campbell', 'Sangam
Chowk', 'Hetauda');
insert into customer values(16, 'Charles Robinson', 'Saptagandaki
Chowk', 'Bharatpur');
insert into customer values(17, 'Thomas Cooper', 'Ram
Chowk', 'Janakpur');
insert into customer values(18, 'David Lae', 'Birendra Path', 'Hetauda');
insert into customer values(19, 'James Lewis', 'Sital Mall', 'Hetauda');
insert into customer values(20, 'Robert Allen', 'School
Road', 'Hetauda');
insert into customer values(21, 'Rishab Khan', 'Hospital
Chowk', 'Pokhara');
insert into customer values(22, 'Joey Miller', 'Amarsingh
Chowk', 'Pokhara');
insert into customer values(23, 'Joy Miller', 'main road', 'Pokhara');
insert into customer values(24, 'Burno Cordoba', 'New road', 'Pokhara');

-- Account
insert into account values(01012305478, 6345, 20000);
insert into account values(01212305468, 6344, 25000);
insert into account values(01012404586, 6325, 50000);
insert into account values(01012104586, 2435, 65000);
insert into account values(01112404586, 2430, 70000);
insert into account values(01012305481, 8345, 45000);
insert into account values(01612103586, 8245, 50000);
insert into account values(00112404863, 8145, 15000);
insert into account values(01012404562, 8344, 150000);
insert into account values(01012404526, 5345, 160000);
insert into account values(01212104580, 5340, 100000);
insert into account values(01012504576, 0345, 250000);

```

```

insert into account values(01112403497,4537,150000);
insert into account values(0112504526,2345,250000);
insert into account values(01012104526,0345,350000);
insert into account values(00112604080,2537,450000);
insert into account values(01012504550,8145,99000);
insert into account values(01012404500,4345,550000);
insert into account values(00112404050,4324,650000);
insert into account values(01012403521,8145,350000);
insert into account values(01012404001,4343,55000);
insert into account values(02015404080,4320,105000);
insert into account values(00112404007,2430,140000);
insert into account values(00112404017,4325,145000);
insert into account values(00112404117,2325,345000);
insert into account values(00112404217,3425,345000);
insert into account values(00112404317,2435,345000);

-- Loan
insert into loan values(05674321567,4345,100000);
insert into loan values(76528612384,4345,20000);
insert into loan values(10192357333,2430,200000);
insert into loan values(12012404280,8145,500000);
insert into loan values(79263994636,4343,65000);
insert into loan values(23763499521,8145,70000);
insert into loan values(82539451438,4324,400000);
insert into loan values(92734755222,2537,230000);
insert into loan values(09862343172,8344,105000);
insert into loan values(02934699255,4320,1500000);
insert into loan values(62348762342,0345,1600000);
insert into loan values(32487691465,4537,1000000);
insert into loan values(76342625343,5345,1500000);
insert into loan values(76342625333,5340,1500000);
insert into loan values(92348769444,8245,3500000);
insert into loan values(67523849211,8345,4050000);
insert into loan values(12359612875,2435,5500000);
insert into loan values(23487652982,6345,3550000);
insert into loan values(23487699172,6344,555000);
insert into loan values(18273497653,8145,1005000);
insert into loan values(23475628222,6325,140000);
insert into loan values(53487618722,2345,870000);

```

```

-- Borrower
insert into borrower values(8, 18273497653);
insert into borrower values(15, 05674321567);
insert into borrower values(12, 62348762342);
insert into borrower values(16, 53487618722);
insert into borrower values(18, 76528612384);
insert into borrower values(2, 23487699172);
insert into borrower values(19, 82539451438);
insert into borrower values(3, 23475628222);
insert into borrower values(22, 09862343172);
insert into borrower values(7, 67523849211);
insert into borrower values(21, 32487691465);
insert into borrower values(11, 76342625333);
insert into borrower values(20, 02934699255);
insert into borrower values(14, 92734755222);
insert into borrower values(5, 10192357333);
insert into borrower values(10, 76342625343);
insert into borrower values(9, 79263994636);
insert into borrower values(1, 23487652982);
insert into borrower values(13, 92348769444);
insert into borrower values(6, 12012404280);
insert into borrower values(17, 23763499521);
insert into borrower values(4, 12359612875);

-- Depositor
insert into depositor values(8, 112404863);
insert into depositor values(15, 1012404500);
insert into depositor values(12, 1012504576);
insert into depositor values(16, 112504526);
insert into depositor values(18, 112404017);
insert into depositor values(2, 1212305468);
insert into depositor values(19, 112404050);
insert into depositor values(3, 1012404586 );
insert into depositor values(22, 112404117);
insert into depositor values(7, 1612103586);
insert into depositor values(13, 1112403497);
insert into depositor values(21, 1012404001);
insert into depositor values(11, 1212104580);
insert into depositor values(20, 2015404080);
insert into depositor values(14, 112604080);

```

```
insert into depositor values(5,1112404586);  
insert into depositor values(10,1012404526);  
insert into depositor values(9,1012404562);  
insert into depositor values(1,1012305478);  
insert into depositor values(6,1012305481);  
insert into depositor values(17,112404217);  
insert into depositor values(4,1012104586);
```

```

MariaDB [bank]> select * from branch;
+-----+-----+-----+-----+
| branch_number | branch_name      | branch_city | assets   |
+-----+-----+-----+-----+
|      345 | Maharshi Chowk    | Dhamauli     | 2000000  |
|     2325 | Amarsingh Chowk   | Pokhara      | 1200000  |
|     2345 | Saptagandaki Chowk | Bharatpur    | 750000   |
|     2430 | Nakidot           | Lalitpur     | 2509000  |
|     2435 | Kupandol          | Lalitpur     | 2500000  |
|     2537 | Bhrikuti Marg     | Dharan       | 1200000  |
|     3425 | Ram Chowk          | Janakput     | 400000   |
|     4320 | School Road        | Hetauda      | 1010000  |
|     4324 | Sital Mall         | Hetauda      | 1010000  |
|     4325 | Birendra Path     | Hetauda      | 3000000  |
|     4343 | Hospital Chowk    | Pokhara      | 1209900  |
|     4345 | Sangam Chowk      | Hetauda      | 1000000  |
|     4537 | Shukra Path        | Biratnagar   | 500000   |
|     5340 | Sahid Chowk        | Narayanghat  | 1000000  |
|     5345 | Ichchhakamana     | Narayanghat  | 990000   |
|     6325 | Baneshwor          | Kathmandu    | 500000   |
|     6344 | Bagmara Marg       | Kathmandu    | 4500000  |
|     6345 | Shivam Tol         | Kathmandu    | 5000000  |
|     8145 | Tinkune            | Pokhara      | 5340000  |
|     8245 | Sabhagriha Chowk  | Pokhara      | 4000000  |
|     8344 | Lakeside            | Pokhara      | 2000000  |
|     8345 | Bagar               | Pokhara      | 5000000  |
+-----+-----+-----+-----+
22 rows in set (0.001 sec)

```

```

MariaDB [bank]> select * from customer;
+-----+-----+-----+-----+
| customer_number | customer_name      | customer_street   | customer_city |
+-----+-----+-----+-----+
|          8 | Aagman Poudel     | Tinkune           | Pokhara        |
|         15 | Barney Campbell    | Sangam Chowk     | Hetauda        |
|         12 | Bhawana Baral     | Maharshi Chowk   | Dhamauli       |
|         16 | Charles Robinson   | Saptagandaki Chowk | Bharatpur      |
|         18 | David Lae          | Birendra Path     | Hetauda        |
|          2 | Hari Parsad Baral | Bagmara Marg      | Kathmandu      |
|         19 | James Lewis         | Sital Mall        | Hetauda        |
|          3 | Jhon Cordoba       | Baneshwor         | Kathmandu      |
|         22 | Joey Miller         | Amarsingh Chowk  | Pokhara        |
|          7 | Prajwol Khan       | Sabhagriha Chowk | Pokhara        |
|         13 | Rena Baral         | Shukra Path       | Biratnagar     |
|         21 | Rishab Khan         | Hospital Chowk   | Pokhara        |
|         11 | Riva Chalise       | Sahid Chowk       | Narayanghat   |
|         20 | Robert Allen        | School Road       | Hetauda        |
|         14 | Robin Jhonson      | Bhrikuti Marg     | Dharan         |
|          5 | Ross Smith          | Nakidot           | Lalitpur       |
|         10 | Roxane Chalise      | Ichchhakamana    | Narayanghat   |
|         9 | Sanskriti Giri      | Lakeside          | Pokhara        |
|        101 | Shyam Sharma        | Shivam Tol        | Kathmandu      |
|        101 | Ted Wilson          | Bagar             | Pokhara        |
|        300 | Thomas Cooper        | Ram Chowk         | Janakpur      |
|        120 | Wyane Rooney         | Kupandol          | Lalitpur       |
+-----+-----+-----+-----+
22 rows in set (0.001 sec)

```

```
MariaDB [bank]> select * from loan; Insert at least 10 records in branch, customer, account, loan
+-----+-----+-----+
| loan_number | branch_number | amount |
+-----+-----+-----+
| 2934699255 | 4320 | 1500000 | 7) Display top 5 records of depositor relations. [Every student inserts their own set of records.]
| 5674321567 | 4345 | 100000 | 8) Find the names of all branches in the branch relation.
| 9862343172 | 8344 | 3105000 | 9) Display all the records of customer relation.
| 10192357333 | 2430 | 4200000 | 10) Display all the records of loan relation.
| 12012404280 | 8145 | 500000 | 11) Display all the records of borrower relation.
| 12359612875 | 2435 | 5500000 | 12) Display all the records of account relation.
| 18273497653 | 8145 | 1005000 |
| 23475628222 | 6325 | 140000 |
| 23487652982 | 6345 | 3550000 |
| 23487699172 | 6344 | 5550000 | DBMS LAB SHEET BY: BAL KRISHNA NYAUPANE
| 23763499521 | 8145 | 70000 |
| 32487691465 | 4537 | 1000000 |
| 53487618722 | 2345 | 870000 |
| 62348762342 | 345 | 1600000 |
| 67523849211 | 8345 | 4050000 |
| 76342625333 | 5340 | 1500000 |
| 76342625343 | 5345 | 1500000 |
| 76528612384 | 4345 | 20000 |
| 79263994636 | 4343 | 65000 |
| 82539451438 | 4324 | 400000 |
| 92348769444 | 8245 | 3500000 |
| 92734755222 | 2537 | 230000 |
+-----+-----+-----+
22 rows in set (0.000 sec)
```

```
MariaDB [bank]> select * from borrower; Insert at least 10 records in branch, customer, account, loan
+-----+-----+
| customer_number | loan_number | depositor relations. [Every student inserts their own set of records.]
+-----+-----+
| 8 | 18273497653 | should be different from your friends.]
| 15 | 5674321567 | 1) Display all the records of branch relation.
| 12 | 62348762342 | 2) Display all the records of customer relation.
| 16 | 53487618722 | 3) Display all the records of loan relation.
| 18 | 76528612384 | 4) Display all the records of borrower relation.
| 2 | 23487699172 | 5) Display all the records of account relation.
| 19 | 82539451438 |
| 3 | 23475628222 |
| 22 | 9862343172 |
| 7 | 67523849211 |
| 21 | 32487691465 |
| 20 | 2934699255 |
| 14 | 92734755222 |
| 5 | 10192357333 |
| 10 | 76342625343 |
| 9 | 79263994636 |
| 1 | 23487652982 |
| 13 | 92348769444 |
| 6 | 12012404280 |
| 4 | 12359612875 |
| 11 | 76342625333 |
| 17 | 23763499521 |
+-----+-----+
22 rows in set (0.000 sec)
```

```
MariaDB [bank]> select * from account;  QL to achieve the following results:
+-----+-----+-----+
| account_number | branch_number | balance |
+-----+-----+-----+
| 112404007 | 2430 | 140000.00 | last 10 records in branch, customer, account, loan
| 112404017 | 4325 | 145000.00 | relations. [Every student inserts their own set of 10]
| 112404050 | 4324 | 650000.00 | different from your friends.]
| 112404117 | 2325 | 345000.00 | the records of branch relation.
| 112404217 | 3425 | 345000.00 | the records of customer relation.
| 112404317 | 2435 | 345000.00 | the records of loan relation.
| 112404863 | 8145 | 15000.00 | the records of borrower relation.
| 112504526 | 2345 | 250000.00 | the records of account relation.
| 112604080 | 2537 | 450000.00 | the records of account relation.
| 1012104526 | 345 | 350000.00 |
| 1012104586 | 2435 | 65000.00 |
| 1012305478 | 6345 | 20000.00 | BY: BAL KRISHNA NYAUPANE
| 1012305481 | 8345 | 45000.00 |
| 1012403521 | 8145 | 350000.00 |
| 1012404001 | 4343 | 55000.00 |
| 1012404500 | 4345 | 550000.00 |
| 1012404526 | 5345 | 160000.00 |
| 1012404562 | 8344 | 150000.00 |
| 1012404586 | 6325 | 50000.00 |
| 1012504550 | 8145 | 99000.00 |
| 1012504576 | 345 | 250000.00 |
| 1112403497 | 4537 | 150000.00 |
| 1112404586 | 2430 | 70000.00 |
| 1212104580 | 5340 | 100000.00 |
| 1212305468 | 6344 | 25000.00 | up to 5 records of depositor relation
| 1612103586 | 8245 | 50000.00 |
| 2015404080 | 4320 | 105000.00 | times of all branches in the loan relation (with or
+-----+-----+-----+
27 rows in set (0.001 sec)
```

```
MariaDB [bank]> select * from depositor;
+-----+-----+
| customer_number | account_number |
+-----+-----+
| 140000.00 | 8 | 112404863 |
| 145000.00 | 15 | 1012404500 |
| 650000.00 | 12 | 1012504576 |
| 345000.00 | 16 | 112504526 |
| 345000.00 | 18 | 112404017 |
| 345000.00 | 2 | 1212305468 |
| 15000.00 | 19 | 112404050 |
| 250000.00 | 3 | 1012404586 |
| 450000.00 | 22 | 112404117 |
| 350000.00 | 7 | 1612103586 |
| 65000.00 | 13 | 1112403497 |
| 20000.00 | 21 | 1012404001 |
| 45000.00 | 11 | 1212104580 |
| 350000.00 | 20 | 2015404080 |
| 55000.00 | 14 | 112604080 |
| 550000.00 | 5 | 1112404586 |
| 160000.00 | 10 | 1012404526 |
| 150000.00 | 9 | 1012404562 |
| 50000.00 | 1 | 1012305478 |
| 99000.00 | 6 | 1012305481 |
| 250000.00 | 17 | 112404217 |
| 150000.00 | 4 | 1012104586 |
+-----+-----+
22 rows in set (0.001 sec)
```

23) Display all the records

```
select * from branch;  
  
select * from customer;  
  
select * from loan;  
  
select * from borrower;  
  
select * from account;  
  
select * from depositor;
```

24) Display top 5 records of depositor relation

```
select * from depositor limit 5;
```

25) Find the names of all branches in the loan relation (with or without duplicates).

```
select branch_number,branch_name from loan;  
select distinct branch_number,branch_name from loan;
```

```
MariaDB [bank]> select branch_number,branch_name from loan;
+-----+-----+
| branch_number | branch_name |
+-----+-----+
| 345 | Maharshi Chowk |
| 2345 | Saptagandaki Chowk |
| 2430 | Nakidot |
| 2435 | Kupandol |
| 2537 | Bhrikuti Marg |
| 4320 | School Road |
| 4324 | Sital Mall |
| 4343 | Hospital Chowk |
| 4345 | Sangam Chowk |
| 4345 | Sangam Chowk |
| 4537 | Shukra Path |
| 5340 | Sahid Chowk |
| 5345 | Ichchhakamana |
| 6325 | Baneshwor |
| 6344 | Bagmara Marg |
| 6345 | Shivam Tol |
| 8145 | Tinkune |
| 8145 | Tinkune |
| 8145 | Tinkune |
| 8245 | Sabhagriha Chowk |
| 8344 | Lakeside |
| 8345 | Bagar |
+-----+
22 rows in set (0.000 sec)
```

```
MariaDB [bank]> select distinct branch_number,branch_name from loan;
+-----+-----+
| branch_number | branch_name |
+-----+-----+
| 345 | Maharshi Chowk |
| 2345 | Saptagandaki Chowk |
| 2430 | Nakidot |
| 2435 | Kupandol |
| 2537 | Bhrikuti Marg |
| 4320 | School Road |
| 4324 | Sital Mall |
| 4343 | Hospital Chowk |
| 4345 | Sangam Chowk |
| 4537 | Shukra Path |
| 5340 | Sahid Chowk |
| 5345 | Ichchhakamana |
| 6325 | Baneshwor |
| 6344 | Bagmara Marg |
| 6345 | Shivam Tol |
| 8145 | Tinkune |
| 8245 | Sabhagriha Chowk |
| 8344 | Lakeside |
| 8345 | Bagar |
+-----+
19 rows in set (0.001 sec)
```

26) Find all loan numbers for loans made at the Tinkune branch with loan amounts greater than 200000.

```
select loan_number, l.branch_number, b.branch_name from loan as
l,branch as b where l.branch_number=b.branch_number and
b.branch_name='Tinkune' and amount>200000;
```

```
MariaDB [bank]> select loan_number, branch_number, amount from loan where branch_name='Tinkune' and
amount>200000;
+-----+-----+-----+
| loan_number | branch_number | amount |
+-----+-----+-----+
| 12012404280 | 8145 | 500000 |
| 18273497653 | 8145 | 1005000 |
+-----+-----+-----+
2 rows in set (0.001 sec)
```

27) Find the loan number of those loans with loan amounts between 9,00,000 and 10,00,000.

```
select loan_number,amount from loan where amount between 900000 and 1000000;
```

```
MariaDB [bank]> select loan_number,amount from loan where amount between 900000 and 1000000;
+-----+-----+
| loan_number | amount |
+-----+-----+
| 32487691465 | 1000000 |
+-----+-----+
1 row in set (0.000 sec)
```

28) For all customers who have a loan from the bank, find their names, loan numbers, and loan amount.

```
select c.customer_name, l.loan_number, l.amount from loan as l,
customer as c , borrower as b where b.loan_number=l.loan_number and
c.customer_number=b.customer_number;
```

```
MariaDB [bank]> select c.customer_name, l.loan_number, l.amount from loan as l, customer as c , borrower as b where b.loan_number=l.loan_number and c.customer_number=b.customer_number;
+-----+-----+-----+
| customer_name | loan_number | amount |
+-----+-----+-----+
| Aagman Poudel | 18273497653 | 1005000 |
| Barney Campbell | 5674321567 | 1000000 |
| Bhawana Baral | 62348762342 | 1600000 |
| Charles Robinson | 53487618722 | 870000 |
| David Lae | 76528612384 | 20000 |
| Hari Parsad Baral | 23487699172 | 555000 |
| James Lewis | 82539451438 | 400000 |
| Jhon Cordoba | 23475628222 | 140000 |
| Joey Miller | 9862343172 | 105000 |
| Prajwol Khan | 67523849211 | 4050000 |
| Rishab Khan | 32487691465 | 1000000 |
| Robert Allen | 2934699255 | 1500000 |
| Robin Jhonsen | 92734755222 | 230000 |
| Ross Smith | 10192357333 | 200000 |
| Roxane Chalise | 76342625343 | 1500000 |
| Sanskriti Giri | 79263994636 | 65000 |
| Shyam Sharma | 23487652982 | 3550000 |
| Rena Baral | 92348769444 | 3500000 |
| Ted Wilson | 12012404280 | 500000 |
| Wyane Rooney | 12359612875 | 5500000 |
| Riva Chalise | 76342625333 | 1500000 |
| Thomas Cooper | 23763499521 | 70000 |
+-----+-----+-----+
22 rows in set (0.001 sec)
```

29) Find the customer names, loan numbers, and loan amounts for all loans at the Tinkune branch.

```
select c.customer_name,l.loan_number,l.amount
from loan as l,customer as c, borrower as b, branch as bh
where l.loan_number=b.loan_number and
c.customer_number=b.customer_number and
l.branch_number=bh.branch_number and bh.branch_name='Tinkune';
```

```
MariaDB [bank]> select c.customer_name,l.loan_number,l.amount
-> from loan as l,customer as c, borrower as b, branch as bh
-> where l.loan_number=b.loan_number and c.customer_number=b.customer_number and l.branch_number=bh.branch_number and bh.branch_name='Tinkune';
+-----+-----+
| customer_name | loan_number | amount |
+-----+-----+
| Ted Wilson   | 12012404280 | 500000 |
| Aagman Poudel | 18273497653 | 1005000 |
| Thomas Cooper | 23763499521 | 70000 |
+-----+-----+
3 rows in set (0.001 sec)
```

30) Find the names of all branches that have assets greater than at least one branch located in Baneshwor.

```
select branch_name,assets from branch where assets > some (select
assets from branch where branch_name='Baneshwor');
```

```
MariaDB [bank]> select branch_name,assets from branch where assets > some (select assets from branch where branch_name
='Baneshwor');
+-----+
| branch_name | assets |
+-----+
| Saptagandaki Chowk | 750000 |
| Nakidot | 2509000 |
| Kupandol | 2500000 |
| Ram Chowk | 400000 |
| Birendra Path | 3000000 |
| Shukra Path | 500000 |
| Ichchhakamana | 990000 |
| Baneshwor | 500000 |
| Bagmara Marg | 4500000 |
| Shivam Tol | 5000000 |
| Tinkune | 5340000 |
| Sabhgriha Chowk | 4000000 |
| Bagar | 5000000 |
+-----+
13 rows in set (0.001 sec)
```

7) Display top 5 records of depositor relation
8) Find the names of all branches in the loan relation (with or without duplicates).
9) Find all loan numbers for loans made at the **Tinkune** branch with loan amounts greater than 200000.
10) Find the loan number of those loans with loan amounts between 9,00,000 and 10,00,000.
11) For all customers who have a loan from the bank, find their names, loan numbers, and loan amount.

31) Find the names of all customers whose street address includes the substring 'main'.

```
select customer_name from customer where customer_street like '%main%';
```

```
MariaDB [bank]> select customer_name from customer where customer_street like '%main%';
+-----+
| customer_name |
+-----+
| Joy Miller    |
+-----+
1 row in set (0.001 sec)
```

- 7) Display top 5 records of depositor relation
8) Find the names of all branches in the loan relation (no duplicates).

32) Find the names of all customers whose name contains at least four characters.

```
select customer_name from customer where customer_name like '____%';
```

```
MariaDB [bank]> select customer_name from customer where customer_name like '____%';
+-----+
| customer_name |
+-----+
| Shyam Sharma   |
| Hari Parsad Baral |
| Jhon Cordoba   |
| Wyane Rooney   |
| Ross Smith     |
| Ted Wilson     |
| Prajwol Khan   |
| Aagman Poudel  |
| Sanskriti Giri |
| Roxane Chalise  |
| Riva Chalise   |
| Bhawana Baral  |
| Rena Baral     |
| Robin Jhonsone |
| Barney Campbell |
| Charles Robinson|
| Thomas Cooper   |
| David Lae       |
| James Lewis     |
| Robert Allen    |
| Rishab Khan     |
| Joey Miller     |
| Joy Miller      |
+-----+
23 rows in set (0.000 sec)
```

- [Should be different from your friends.]
2) Display all the records of branch relation.
3) Display all the records of customer relation.
4) Display all the records of loan relation.
5) Display all the records of borrower relation.
6) Display all the records of account relation.

DBMS LAB SHEET BY: BAL KRISHNA NYAUPANE

- 7) Display top 5 records of depositor relation
8) Find the names of all branches in the loan relation (no duplicates).

33) Find the names of all customers whose name start with 'b' and end with 'a'.

```
select customer_name from customer where customer_name like 'b%a';
```

```
MariaDB [bank]> select customer_name from customer where customer_name like 'b%a';
+-----+
| customer_name |
+-----+
| Burno Cordoba |
+-----+
1 row in set (0.001 sec)
```

8) Find the names of all branches in the loan relation (no duplicates).

9) Find all loan numbers for loans made at the **Tinkune** branch with amounts greater than 200000.

34) List all the customer's name in alphabetic order who have a loan at the Tinkune branch.

```
select c.customer_name, b.branch_name from customer as c, branch as b,
loan as l, borrower as br where c.customer_number=br.customer_number
and br.loan_number=l.loan_number and l.branch_number=b.branch_number
and b.branch_name='Tinkune' order by c.customer_name;
```

```
MariaDB [bank]> select c.customer_name, b.branch_name from customer as c, branch as b, loan as l, borrower as br where
c.customer_number=br.customer_number and br.loan_number=l.loan_number and l.branch_number=b.branch_number and b.branch_name='Tinkune' order by c.customer_name;
+-----+-----+
| customer_name | branch_name |
+-----+-----+
| Aagman Poudel | Tinkune      | 11) For all customers who have a loan from the bank, find their names, loan
| Ted Wilson    | Tinkune      | numbers, and loan amount.
| Thomas Cooper | Tinkune      | 12) Find the customer names, loan numbers, and loan amounts for all loans at
|                |              | the Tinkune branch.
+-----+-----+
3 rows in set (0.001 sec)
```

13) Find the names of all branches that have assets greater than one million.

35) List the entire loan relation in descending order of amount. If several loans have the same amount, then order them in ascending order by loan number.

```
select * from loan order by amount desc, loan_number asc;
```

```
MariaDB [bank]> select * from loan order by amount desc, loan_number asc;
+-----+-----+-----+
| loan_number | branch_number | amount |
+-----+-----+-----+
| 12359612875 | 2435 | 5500000 |
| 67523849211 | 8345 | 4050000 |
| 23487652982 | 6345 | 3550000 |
| 92348769444 | 8245 | 3500000 |
| 62348762342 | 345 | 1600000 |
| 2934699255 | 4320 | 1500000 |
| 76342625333 | 5340 | 1500000 |
| 76342625343 | 5345 | 1500000 |
| 18273497653 | 8145 | 1005000 |
| 32487691465 | 4537 | 1000000 |
| 53487618722 | 2345 | 870000 |
| 23487699172 | 6344 | 555000 |
| 12012404280 | 8145 | 500000 |
| 82539451438 | 4324 | 400000 |
| 92734755222 | 2537 | 230000 |
| 10192357333 | 2430 | 200000 |
| 23475628222 | 6325 | 140000 |
| 9862343172 | 8344 | 105000 |
| 5674321567 | 4345 | 100000 |
| 23763499521 | 8145 | 70000 |
| 79263994636 | 4343 | 65000 |
| 76528612384 | 4345 | 20000 |
+-----+-----+-----+
22 rows in set (0.001 sec)
```

36) Find all the bank customers having a loan, an account, or both at the bank (with or without duplicates).

```
-- without duplicate
(select c.customer_name from customer as c, loan as l ,borrower as b
where c.customer_number=b.customer_number and
b.loan_number=l.loan_number)union(select c.customer_name from customer
as c, account as a, depositor as d where
c.customer_number=d.customer_number and
d.account_number=a.account_number)

-- with duplicate
(select c.customer_name from customer as c, loan as l ,borrower as b
where c.customer_number=b.customer_number and
b.loan_number=l.loan_number)union all(select c.customer_name from
customer as c, account as a, depositor as d where
c.customer_number=d.customer_number and
d.account_number=a.account_number)
```

```
MariaDB [bank]> (select c.customer_name from customer as c, loan as l ,borrower as b where c.customer_number=b.customer_number
and b.loan_number=l.loan_number) union (select c.customer_name from customer as c, account as a, depositor as d where c.custo
mer_number=d.customer_number and d.account_number=a.account_number);
+-----+
| customer_name |
+-----+
| Aagman Poudel
| Barney Campbell
| Bhawana Baral
| Charles Robinson
| David Lae
| Hari Parsad Baral
| James Lewis
| Jhon Cordoba
| Joey Miller
| Prajwol Khan
| Rishab Khan
| Robert Allen
| Robin Jhonsen
| Ross Smith
| Roxane Chalise
| Sanskriti Giri
| Shyam Sharma
| Rena Baral
| Ted Wilson
| Wyane Rooney
| Riva Chalise
| Thomas Cooper
| Joy Miller
+-----+
23 rows in set (0.001 sec)
```

- 10) Find the loan number of those loans with loan amounts between 9,00,000 and 10,00,000.
- 11) For all customers who have a loan from the bank, find their names, loan numbers, and loan amount.
- 12) Find the customer names, loan numbers, and loan amounts for all loans at the **Tinkune** branch.
- 13) Find the names of all branches that have assets greater than at least one branch located in **Baneshwor**.
- 14) Find the names of all customers whose street address includes the substring 'main'.
- 15) Find the names of all customers whose name contains *at least four characters*.
- 16) Find the names of all customers whose name start with 'b' and end with 'a'.
- 17) List all the customer's name in alphabetic order who have a loan at the **Tinkune** branch.
- 18) List the entire loan relation in descending order of amount. If several loans have the same amount, then order them in ascending order by loan number.
- 19) Find all the bank customers having a loan, an account, or both at the bank (with or without duplicates).

```
MariaDB [bank]> (select c.customer_name from customer as c, loan as l ,borrower as b where c.customer_number=b.customer_number and b.loan_number=l.loan_number) union all(s
elect c.customer_name from customer as c, account as a, depositor as d where c.customer_number=d.customer_number and d.account_number=a.account_number);
```

customer_name	customer_number	branch	loan_number	amount
Aagman Poudel	1010000000000000000	Tinkune	1010000000000000000	4250
Barney Campbell	1010000000000000001	Tinkune	1010000000000000001	250
Bhawana Baral	1010000000000000002	Tinkune	1010000000000000002	300
Charles Robinson	1010000000000000003	Tinkune	1010000000000000003	300
David Lae	1010000000000000004	Tinkune	1010000000000000004	300
Hari Parsad Baral	1010000000000000005	Tinkune	1010000000000000005	300
James Lewis	1010000000000000006	Tinkune	1010000000000000006	300
Jhon Cordoba	1010000000000000007	Tinkune	1010000000000000007	300
Joey Miller	1010000000000000008	Tinkune	1010000000000000008	300
Prajwol Khan	1010000000000000009	Tinkune	1010000000000000009	300
Rishab Khan	1010000000000000010	Tinkune	1010000000000000010	300
Robert Allen	1010000000000000011	Tinkune	1010000000000000011	300
Robin Jhonsen	1010000000000000012	Tinkune	1010000000000000012	300
Ross Smith	1010000000000000013	Tinkune	1010000000000000013	300
Roxane Chalise	1010000000000000014	Tinkune	1010000000000000014	300
Sanskriti Giri	1010000000000000015	Tinkune	1010000000000000015	300
Shyam Sharma	1010000000000000016	Tinkune	1010000000000000016	300
Rena Baral	1010000000000000017	Tinkune	1010000000000000017	300
Ted Wilson	1010000000000000018	Tinkune	1010000000000000018	300
Wyane Rooney	1010000000000000019	Tinkune	1010000000000000019	300
Riva Chalise	1010000000000000020	Tinkune	1010000000000000020	300
Thomas Cooper	1010000000000000021	Tinkune	1010000000000000021	300
Joy Miller	1010000000000000022	Tinkune	1010000000000000022	300
Aagman Poudel	1010000000000000023	Baneshwor	1010000000000000023	4250
Barney Campbell	1010000000000000024	Baneshwor	1010000000000000024	250
Bhawana Baral	1010000000000000025	Baneshwor	1010000000000000025	300
Charles Robinson	1010000000000000026	Baneshwor	1010000000000000026	300
David Lae	1010000000000000027	Baneshwor	1010000000000000027	300
Hari Parsad Baral	1010000000000000028	Baneshwor	1010000000000000028	300
James Lewis	1010000000000000029	Baneshwor	1010000000000000029	300
Jhon Cordoba	1010000000000000030	Baneshwor	1010000000000000030	300
Joey Miller	1010000000000000031	Baneshwor	1010000000000000031	300
Prajwol Khan	1010000000000000032	Baneshwor	1010000000000000032	300
Rishab Khan	1010000000000000033	Baneshwor	1010000000000000033	300
Rena Baral	1010000000000000034	Baneshwor	1010000000000000034	300
Ted Wilson	1010000000000000035	Baneshwor	1010000000000000035	300
Wyane Rooney	1010000000000000036	Baneshwor	1010000000000000036	300
Riva Chalise	1010000000000000037	Baneshwor	1010000000000000037	300
Thomas Cooper	1010000000000000038	Baneshwor	1010000000000000038	300
Joy Miller	1010000000000000039	Baneshwor	1010000000000000039	300

Q. Find all the bank customers having a loan (with or without duplicates).

37) Find all customers who have both a loan and an account at the bank (with or without duplicates).

```
-- without duplicate
(select c.customer_name from customer as c, loan as l ,borrower as b
where c.customer_number=b.customer_number and
b.loan_number=l.loan_number) intersect (select c.customer_name from
customer as c, account as a, depositor as d where
c.customer_number=d.customer_number and
d.account_number=a.account_number);

-- with duplicate
(select c.customer_name from customer as c, loan as l ,borrower as b
where c.customer_number=b.customer_number and
b.loan_number=l.loan_number) intersect all(select c.customer_name from
customer as c, account as a, depositor as d where
c.customer_number=d.customer_number and
d.account_number=a.account_number);
```

MariaDB [bank]> (select c.customer_name from customer as c, loan as l ,borrower as b where c.customer_number=b.customer_number and b.loan_number=l.loan_number) intersect (select c.customer_name from customer as c, account as a, depositor as d where c.customer_number=d.customer_number and d.account_number=a.account_number);

customer_name
Aagman Poudel
Barney Campbell
Bhawana Baral
Charles Robinson
David Lee
Hari Parsad Baral
James Lewis
Jhon Cordoba
Joey Miller
Prajwol Khan
Rishab Khan
Robert Allen
Robin Jhonsnon
Ross Smith
Roxane Chalise
Sanskriti Giri
Shyam Sharma
Rena Baral
Ted Wilson
Wayne Rooney
Riva Chalise
Thomas Cooper

22 rows in set (0.001 sec)

9) Find all loan numbers for loans made at the *Tinkana* branch with loan amounts greater than 200000.
10) Find the loan number of those loans with loan amounts between 9,00,000 and 10,00,000.
11) For all customers who have a loan from the bank, find their names, loan numbers, and loan amount.
12) Find the customer names, loan numbers, and loan amounts for all loans at the *Tinkana* branch.
13) Find the names of all branches that have assets greater than at least one branch located in *Hanshwar*.
14) Find the names of all customers whose street address includes the substring 'main'.
15) Find the names of all customers whose name contains *at least four characters*.
16) Find the names of all customers whose name start with 'b' and end with 'a'.
17) List all the customer's name in alphabetic order who have a loan at the *Tinkana* branch.
18) List the entire loan relation in descending order of amount. If several loans

38) Find all customers who have an account but no loan at the bank (with or without duplicates).

```
-- Without duplicate
(select c.customer_name from customer as c, depositor as d, account as a where c.customer_number=d.customer_number and d.account_number=a.account_number) except(select cu.customer_name from customer as cu, loan as l, borrower as br where cu.customer_number=br.customer_number and br.loan_number=l.loan_number);

-- with duplicate
(select c.customer_name from customer as c, depositor as d, account as a where c.customer_number=d.customer_number and d.account_number=a.account_number) except all(select cu.customer_name from customer as cu, loan as l, borrower as br where cu.customer_number=br.customer_number and br.loan_number=l.loan_number);
```

```
MariaDB [bank]> (select c.customer_name from customer as c, depositor as d, account as a where c.customer_number=d.customer_number and d.account_number=a.account_number) except(select cu.customer_name from customer as cu, loan as l, borrower as br where cu.customer_number=br.customer_number and br.loan_number=l.loan_number);
+-----+
| customer_name |
+-----+
| Joy Miller    |
+-----+
1 row in set (0.001 sec)
```

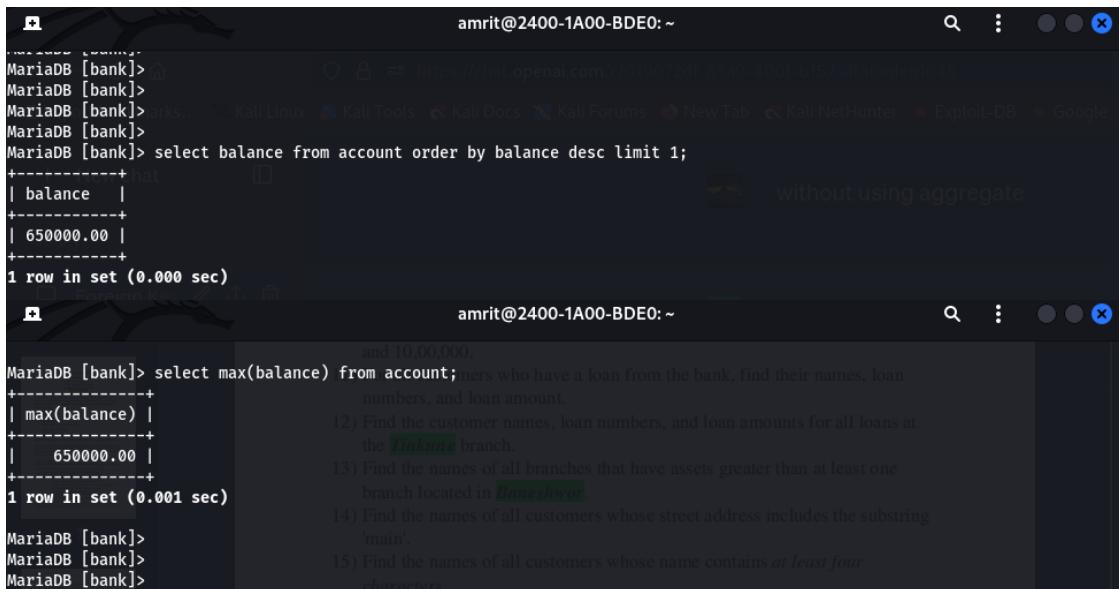
the *Dakine* branch.

13) Find the names of all branches that have assets greater than at least one branch located in *Baneshwor*.

14) Find the names of all customers whose street address includes the substring 'main'.

39) Find the largest account balance in the bank [with or without aggregate function].

```
-- without aggregate
select balance from account order by desc limit 1;
-- with aggregate
select max(balance) from account;
```



```
MariaDB [bank]> select balance from account order by balance desc limit 1;
+-----+
| balance |
+-----+
| 650000.00 |
+-----+
1 row in set (0.000 sec)

MariaDB [bank]> select max(balance) from account;
+-----+
| max(balance) |
+-----+
| 650000.00 |
+-----+
1 row in set (0.001 sec)
```

40) Find the names of all customers who have an account in Tinkune branch or Baneshwor branch, or both.

```
(select c.customer_name from customer as c,branch as b,borrower as br,loan as l where c.customer_number=br.customer_number and br.loan_number = l.loan_number and l.branch_number=b.branch_number and b.branch_name='Tinkune')

union

(select c.customer_name from customer as c,branch as b,borrower as br,loan as l where c.customer_number=br.customer_number and br.loan_number = l.loan_number and l.branch_number=b.branch_number and b.branch_name='Baneshwor');
```

```

MariaDB [bank]> (select c.customer_name from customer as c,branch as b,borrower as br,loan as l where c.customer_number=br.customer_number and br.loan_number = l.loan_number and l.branch_number=b.branch_number and b.branch_name='Tinkune') union (select c.customer_name from customer as c,branch as b,borrower as br,loan as l where c.customer_number=br.customer_number and br.loan_number = l.loan_number and l.branch_number=b.branch_number and b.branch_name='Baneshwor');
+-----+
| customer_name |
+-----+
| Aagman Poudel |
| Ted Wilson |
| Thomas Cooper |
| Jhon Cordoba |
+-----+
4 rows in set (0.001 sec)

          (with or without duplicates).
20) Find all customers who have both a loan and an account at the bank (with or without duplicates).
21) Find all customers who have an account but no loan at the bank (with or without duplicates).
22) Find the largest account balance in the bank [with or without aggregate function].
23) Find the names of all customers who have an account in Tinkune branch or

```

41) Find the average account balance at the Tinkune branch.

```

select a.account_number, a.balance,b.branch_name from account as a,
branch as b where a.branch_number=b.branch_number and
b.branch_name='Tinkune';

```

```

MariaDB [bank]> select a.account_number, a.balance,b.branch_name from account as a, branch as b where a.branch_number=b.branch_number and b.branch_name='Tinkune'; the names of all customers who have an account in Tinkune branch or
+-----+-----+-----+
| account_number | balance | branch_name |
+-----+-----+-----+
| 112404863 | 15000.00 | Tinkune |
| 112424863 | 150000.00 | Tinkune |
| 1012403521 | 350000.00 | Tinkune |
| 1012504550 | 99000.00 | Tinkune |
+-----+-----+-----+
4 rows in set (0.001 sec)

          29) Find the average balance for all account.

MariaDB [bank]>

```

42) Find the number of tuples in the customer relation.

```

select count(*) from customer;

```

```

MariaDB [bank]> select count(*) as tuple_count from customer;
+-----+
| tuple_count |
+-----+
|      24 |
+-----+
1 row in set (0.000 sec)

```

Executing this query will give you

what is column and row called in?

43) Find the average account balance at each branch.

```
select avg(a.balance),b.branch_name from account as a, branch as b  
where a.branch_number = b.branch_number group by b.branch_name;
```

```
MariaDB [bank]> select avg(a.balance),b.branch_name from account as a, branch as b where a.branch_number = b.branch_number group by b.branch_name;  
+-----+-----+  
| avg(a.balance) | branch_name |  
+-----+-----+  
| 345000.000000 | Amarsingh Chowk |  
| 45000.000000 | Bagar |  
| 25000.000000 | Bagmara Marg |  
| 50000.000000 | Baneshwor |  
| 450000.000000 | Bhrikuti Marg |  
| 145000.000000 | Birendra Path |  
| 55000.000000 | Hospital Chowk |  
| 160000.000000 | Ichchhakamana |  
| 205000.000000 | Kupandol |  
| 150000.000000 | Lakeside |  
| 300000.000000 | Maharshi Chowk |  
| 105000.000000 | Nakidot |  
| 345000.000000 | Ram Chowk |  
| 50000.000000 | Sabhagriha Chowk |  
| 100000.000000 | Sahid Chowk |  
| 550000.000000 | Sangam Chowk |  
| 250000.000000 | Saptagandaki Chowk |  
| 105000.000000 | School Road |  
| 20000.000000 | Shivam Tol |  
| 150000.000000 | Shukra Path |  
| 650000.000000 | Sital Mall |  
| 153500.000000 | Tinkune |  
+-----+-----+  
22 rows in set (0.002 sec)
```

44) Find the number of depositors for each branch.

```
select count(customer_number), b.branch_name, b.branch_number from
depositor as d, branch as b, account as a where
a.account_number=d.account_number and a.branch_number =
b.branch_number group by branch_name;
```

count(customer_number)	branch_name	branch_number
1	Amarsingh Chowk	3235
1	Bagar	8345
1	Bagnara Marg	6344
1	Baneshwor	6325
1	Bhrikuti Marg	2537
1	Birendra Path	4325
1	Hospital Chowk	4343
1	Ichchhakamana	5345
1	Kupandol	2435
1	Lakeside	customer 8344 depositor as d
1	Maharshi Chowk	345 number =
1	Nakidot	2430
1	Ram Chowk	3425
1	SabhaGriha Chowk	8245
1	Sahid Chowk	5340
1	Sangam Chowk	4345
1	Saptgandaki Chowk	2345
1	School Road	4320
1	Shivam Tol	6345
1	Shukra Path	4537
1	Sital Mall	4324
2	Tinkune	8145

45) Find the name of branches where the average account balance is more than 12,00,000.

```
select avg(balance), b.branch_name from account as a, branch as b
where b.branch_number=a.branch_number and a.balance>1200000;
```

avg(balance)	branch_name
195678.571429	Nakidot

46) Find the average balance for all account.

```
select avg(balance) from account group by account_number;
```

```
MariaDB [bank]> select avg(balance) from account group by account_number;
+-----+
| avg(balance) |
+-----+
| 140000.000000 |
| 145000.000000 |
| 650000.000000 |
| 345000.000000 |
| 345000.000000 |
| 345000.000000 |
| 15000.000000 |
| 150000.000000 |
| 250000.000000 |
| 450000.000000 |
| 350000.000000 |
| 650000.000000 |
| 20000.000000 |
| 45000.000000 |
| 350000.000000 |
| 55000.000000 |
| 55000.000000 |
| 160000.000000 |
| 150000.000000 |
| 50000.000000 |
| 99000.000000 |
| 250000.000000 |
| 150000.000000 |
| 70000.000000 |
| 100000.000000 |
| 25000.000000 |
| 50000.000000 |
| 105000.000000 |
+-----+
28 rows in set (0.001 sec)
```

```
+-----+
| branch_name |
+-----+
| Ichchhakamana |
| Kupandali |
| Lakeside |
| Maharsi Chowk |
| Nakidot |
| Ram Chowk |
| Sabhgriha Ch |
| Sahid Chowk |
| Sangam Chowk |
| Saptagandaki |
| School Road |
| Shilvam Tel |
| Shukra Path |
| Sital Mall |
| Tinkune |
+-----+
22 rows in set (0.001 sec)
```

Q 28) Find the name of branches w

```
MariaDB [bank]> select avg(balance), b.branch_name
    from account a
    inner join branch b
    on a.branch_number = b.branch_id
    where a.balance > 12000;
+-----+-----+
| avg(balance) | branch_name |
+-----+-----+
| 195678.571429 | Nakidot |
+-----+-----+
1 row in set (0.001 sec)
```

47) Find the average balance for each customer who lives in Pokhara and has at least three accounts.

```
select avg(a.balance), b.branch_city from account as a, branch as b, customer as c, depositor as d where a.branch_number=b.branch_number and a.account_number=d.account_number and d.customer_number = c.customer_number and c.customer_city = 'Pokhara';
```

```
MariaDB [bank]> select avg(a.balance), b.branch_city from account as a, branch as b, customer as c, depositor as d where a.branch_number=b.branch_number and a.account_number=d.account_number and d.customer_number = c.customer_number and c.customer_city = 'Pokhara';  
+-----+-----+  
| avg(a.balance) | branch_city |  
+-----+-----+  
| 115714.285714 | Pokhara |  
+-----+-----+  
1 row in set (0.001 sec)
```

48) Find all customers who have both an account and a loan at the Tinkune branch. [use set membership]

```
select distinct c.customer_name from customer as c, branch as b where c.customer_name in(select customer_name from customer as c, branch as b, loan as l, account as a, borrower as br , depositor as d where b.branch_name = 'Tinkune' and a.branch_number=b.branch_number and a.account_number=d.account_number and d.customer_number=c.customer_number and br.loan_number=l.loan_number and br.customer_number = c.customer_number);
```

```
MariaDB [bank]> select distinct c.customer_name from customer as c, branch as b where c.customer_name in(select customer_name from customer as c, branch as b, loan as l, account as a, borrower as br , depositor as d where b.branch_name = 'Tinkune' and a.branch_number=b.branch_number and a.account_number=d.account_number and d.customer_number=c.customer_number and br.loan_number=l.loan_number and br.customer_number = c.customer_number);  
+-----+  
| customer_name |  
+-----+  
| Aagman Poudel |  
+-----+  
1 row in set (0.018 sec)
```

49) Find all customers who do have a loan at the bank, but do not have an account at the bank. [use set membership]

```
select distinct c.customer_name from customer as c, branch as b where
c.customer_name in(select customer_name from customer as c, loan as l,
account as a, borrower as br, depositor as d where );
```

```
3 rows in set (0.001 sec)

MariaDB [bank]> select customer_name from customer where customer_name in(select c.customer_name fro
m customer as c, loan as l, borrower as b where l.loan_number=b.loan_number and b.customer_number=c.
customer_number) and customer_name not in (select c.customer_name from customer as c, account as a,
depositor as d where d.account_number=a.account_number and c.customer_number=d.customer_number);
Empty set (0.001 sec)

MariaDB [bank]>
```

50) Find out the total balance of the bank.

```
select balance, b.branch_name from account as a, branch as b where
a.branch_number=b.branch_number;
```

```
MariaDB [bank]> select balance, b.branch_name from account as a, branch as b where a.branch_number=
b.branch_number;
+-----+-----+
| balance | branch_name |
+-----+-----+
| 140000.00 | Nakidot |
| 145000.00 | Birendra Path |
| 650000.00 | Sital Mall |
| 345000.00 | Amarsingh Chowk |
| 345000.00 | Ram Chowk |
| 345000.00 | Kupandol |
| 15000.00 | Tinkune |
| 150000.00 | Tinkune |
| 250000.00 | Saptagandaki Chowk |
| 450000.00 | Bhrikuti Marg |
| 350000.00 | Maharsi Chowk |
| 65000.00 | Kupandol |
| 20000.00 | Shivam Tol |
| 45000.00 | Bagar |
| 350000.00 | Tinkune |
| 55000.00 | Hospital Chowk |
| 550000.00 | Sangam Chowk |
| 160000.00 | Ichchhakama |
| 150000.00 | Lakeside |
| 50000.00 | Baneshwor |
| 99000.00 | Tinkune |
| 250000.00 | Maharsi Chowk |
| 150000.00 | Shukra Path |
| 70000.00 | Nakidot |
| 100000.00 | Sahid Chowk |
| 25000.00 | Bagmara Marg |
| 50000.00 | Sabhagriha Chowk |
| 105000.00 | School Road |
+-----+-----+
28 rows in set (0.001 sec)
```

Q 30) Find the average balance for each customer with their branch name.

```
MariaDB [bank]> select avg(a.balance), b.branch_name
from depositor as d where a.branch_number=b.branch_number
and d.customer_number = c.customer_number and c.customer_city
= 'Tinkune';
+-----+-----+
| avg(a.balance) | branch_name |
+-----+-----+
| 119714.285714 | Pokhara |
+-----+-----+
1 row in set (0.001 sec)
```

Q 31) Find all customers who have both an account and a depositor.

```
MariaDB [bank]> select distinct c.customer_name from
c in(select customer_name from customer as c, branch
as b, depositor as d where b.branch_name = 'Tinkune' and
d.account_number and d.customer_number=c.customer_number);
+-----+
| customer_name |
+-----+
| Arpana Poudel |
+-----+
1 row in set (0.018 sec)
```

Q 32) Find all customers who do have a loan at the bank, but do not have an account at the bank. [use set membership]

```
MariaDB [bank]>
```

Q 33) Find out the total balance of the bank.

51) Find the number of branches appearing in the account relation.

```
select count(distinct branch_number) as count from account;
```

```
MariaDB [bank]> select count(distinct branch_number) as count from account;
+-----+
| count |
+-----+
| 22 |
+-----+
1 row in set (0.001 sec)
```

52) Find the total balance of each branch of the bank.

```
select balance, b.branch_name from account as a, branch as b where
a.branch_number=b.branch_number group by b.branch_name;
```

```
MariaDB [bank]> select balance, b.branch_name from account as a, branch as b where a.branch_number=
b.branch_number group by b.branch_name;
+-----+-----+
| balance | branch_name |
+-----+-----+
| 345000.00 | Amarsingh Chowk |
| 45000.00 | Bagar |
| 25000.00 | Bagmara Marg |
| 50000.00 | Baneshwor |
| 450000.00 | Bhrikuti Marg |
| 145000.00 | Birendra Path |
| 55000.00 | Hospital Chowk |
| 160000.00 | Ichchhakamana |
| 345000.00 | Kupandol |
| 150000.00 | Lakeside |
| 350000.00 | Maharshi Chowk |
| 140000.00 | Nakidot |
| 345000.00 | Ram Chowk |
| 50000.00 | Sabhagriha Chowk |
| 100000.00 | Sahid Chowk |
| 550000.00 | Sangam Chowk |
| 250000.00 | Saptagandaki Chowk |
| 105000.00 | School Road |
| 20000.00 | Shivam Tol |
| 150000.00 | Shukra Path |
| 650000.00 | Sital Mall |
| 15000.00 | Tinkune |
+-----+-----+
22 rows in set (0.001 sec)
```



```
MariaDB [bank]> select count(distinct branch_number) as count from account;
+-----+
| count |
+-----+
| 22 |
+-----+
1 row in set (0.001 sec)
```

53) Find the maximum balance at each branch and sum of the balance of each branch. Rename your output attributes.

```
select max(a.balance) as max_balance, sum(a.balance), a.branch_number
as sum_balance from account as a group by a.branch_number;
```

```
MariaDB [bank]> select max(a.balance) as max_balance, sum(a.balance), a.branch_number as sum_balance
from acco
+-----+-----+-----+
| max_balance | sum(a.balance) | sum_balance |
+-----+-----+-----+
| 350000.00 | 600000.00 | 345 |
| 345000.00 | 345000.00 | 2325 |
| 250000.00 | 250000.00 | 2345 |
| 140000.00 | 210000.00 | 2430 |
| 345000.00 | 410000.00 | 2435 |
| 450000.00 | 450000.00 | 2537 |
| 345000.00 | 345000.00 | 3425 |
| 105000.00 | 105000.00 | 4320 |
| 650000.00 | 650000.00 | 4324 |
| 145000.00 | 145000.00 | 4325 |
| 55000.00 | 55000.00 | 4343 |
| 550000.00 | 550000.00 | 4345 |
| 150000.00 | 150000.00 | 4537 |
| 100000.00 | 100000.00 | 5340 |
| 160000.00 | 160000.00 | 5345 |
| 50000.00 | 50000.00 | 6325 |
| 25000.00 | 25000.00 | 6344 |
| 20000.00 | 20000.00 | 6345 |
| 350000.00 | 614000.00 | 8145 |
| 50000.00 | 50000.00 | 8245 |
| 150000.00 | 150000.00 | 8344 |
| 45000.00 | 45000.00 | 8345 |
+-----+-----+-----+
22 rows in set (0.001 sec)
```

54) List the names of customers who have a loan at the bank, and whose names are neither Shyam nor Hari.

```
select c.customer_name,l.loan_number from customer as c, loan as l,
borrower as b where c.customer_number=b.customer_number and
b.loan_number=l.loan_number and c.customer_name not in('Aagman
Poudel','Ross Smith');
```

```
MariaDB [bank]> select c.customer_name,l.loan_number from customer as c, loan as l, borrower as b where c.customer_number=b.customer_number and b.loan_number=l.loan_number and c.customer_name <> 'Aagman Poudel' and c.customer_name<>'Ross Smith';
+-----+-----+
| customer_name | loan_number |
+-----+-----+
| Barney Campbell | 5674321567 |
| Bhawana Baral | 62348762342 |
| Charles Robinson | 53487618722 |
| David Lae | 76528612384 |
| Hari Parsad Baral | 23487699172 |
| James Lewis | 82539451438 |
| Jhon Cordoba | 23475628222 |
| Joey Miller | 9862343172 |
| Prajwol Khan | 67523849211 |
| Rishab Khan | 32487691465 |
| Robert Allen | 2934699255 |
| Robin Jhonson | 92734755222 |
| Roxane Chalise | 76342625343 |
| Sanskriti Giri | 79263994636 |
| Shyam Sharma | 23487652982 |
| Rena Baral | 92348769444 |
| Ted Wilson | 12012404280 |
| Wyane Rooney | 12359612875 |
| Riva Chalise | 76342625333 |
| Thomas Cooper | 23763499521 |
+-----+-----+
20 rows in set (0.001 sec)
```

55) Find the name of all branches that have assets greater than those of at least one branch located in Baneshwor. [with or without using Set Comparison]

```
select branch_name,assets from branch where assets >some (select assets from branch where branch_name = 'Baneshwor');
```

```
MariaDB [bank]> select c.customer_name,l.loan_number from customer as c, loan as l, borrower as b where c.customer_number=b.customer_number and b.loan_number=l.loan_number and c.customer_name not in('Aagman Poudel','Ross Smith');
+-----+-----+
| customer_name | loan_number |
+-----+-----+
| Bhawana Baral | 62348762342 |
| Charles Robinson | 53487618722 |
| Hari Parsad Baral | 23487699172 |
| James Lewis | 82539451438 |
| Prajwol Khan | 67523849211 |
| Rishab Khan | 32487691465 |
| Robert Allen | 2934699255 |
| Robin Jhonsen | 92734755222 |
| Roxane Chalise | 76342625343 |
| Shyam Sharma | 23487652982 |
| Rena Baral | 92348769444 |
| Ted Wilson | 12012404280 |
| Wayne Rooney | 12359612875 |
| Riva Chalise | 76342625333 |
+-----+-----+
14 rows in set (0.001 sec)
```

56) Find the names of all branches that have an asset value greater than that of each branch in Baneshwor.

```
select branch_name,assets from branch where assets >all (select assets from branch where branch_name = 'Baneshwor');
```

```
MariaDB [bank]> select branch_name,assets from branch where assets >all (select assets from branch where branch_name = 'Baneshwor');
+-----+-----+
| branch_name | assets |
+-----+-----+
| Saptagandaki Chowk | 750000 |
| Ichchhakamana | 990000 |
| Shivam Tol | 5000000 |
| Tinkune | 5340000 |
| Bagar | 5000000 |
+-----+-----+
5 rows in set (0.002 sec)
```

57) Delete all account tuples in the Tinkune branch.

```
delete from account where branch_number in(select branch_number from branch where branch_name='Tinkune');
```

```
MariaDB [bank]> delete from account where branch_number in(select branch_number from branch where branch_name='Tinkune');
Query OK, 4 rows affected (0.020 sec)

MariaDB [bank]> select * from account;
+-----+-----+-----+
| account_number | branch_number | balance |
+-----+-----+-----+
| 112404007 | 4320 | 140000.00 |
| 112404017 | 4325 | 145000.00 |
| 112404050 | 4324 | 650000.00 |
| 112404117 | 2325 | 345000.00 |
| 112404217 | 3425 | 345000.00 |
| 112404317 | 2435 | 345000.00 |
| 112504526 | 2345 | 250000.00 |
| 112604080 | 2537 | 450000.00 |
| 1012104526 | 345 | 350000.00 |
| 1012104586 | 2435 | 65000.00 |
| 1012305478 | 6345 | 20000.00 |
| 1012305481 | 8345 | 45000.00 |
| 1012404001 | 4343 | 55000.00 |
| 1012404500 | 4345 | 550000.00 |
| 1012404526 | 5345 | 160000.00 |
| 1012404562 | 8344 | 150000.00 |
| 1012404586 | 6325 | 50000.00 |
| 1012504576 | 345 | 250000.00 |
| 1112403497 | 4537 | 150000.00 |
| 1112404586 | 2430 | 70000.00 |
| 1212104580 | 5340 | 100000.00 |
| 1212305468 | 6344 | 25000.00 |
| 1612103586 | 8245 | 50000.00 |
| 2015404080 | 4320 | 105000.00 |
+-----+-----+-----+
24 rows in set (0.001 sec)
```

58) Delete all loans with loan amounts between 13000 and 150000.

```
delete from loan where amount between 13000 and 150000;
```

```
22 rows in set (0.000 sec)          where clause.  
that of each branch in Baneshwor.  
MariaDB [bank]> delete from loan where amount between 13000 and 150000;  
Query OK, 6 rows affected (0.032 sec)  of who works in the Finance department.  
  
MariaDB [bank]> select * from loan;  
+-----+-----+-----+  
| loan_number | branch_number | amount |  
+-----+-----+-----+  
| 2934699255 |        4320 | 1500000 |  
| 10192357333 |       2430 | 200000 |  
| 12012404280 |       8145 | 500000 |  
| 12359612875 |       2435 | 5500000 |  
| 18273497653 |       8145 | 1005000 |  
| 23487652982 |       6345 | 3550000 |  
| 23487699172 |       6344 | 555000 |  
| 32487691465 |       4537 | 1000000 |  
| 53487618722 |       2345 | 870000 |  
| 62348762342 |       345 | 1600000 |  
| 67523849211 |       8345 | 4050000 |  
| 76342625333 |       5340 | 1500000 |  
| 76342625343 |       5345 | 1500000 |  
| 82539451438 |       4324 | 400000 |  
| 92348769444 |       8245 | 3500000 |  
| 92734755222 |       2537 | 230000 |  
+-----+-----+-----+  
16 rows in set (0.001 sec)
```

59) Delete all account tuples at every branch located in Baneshwor.

```
delete from account where branch_number in(select branch_number from branch where branch_name = 'Baneshwor');
```

```
MariaDB [bank]> delete from account where branch_number in(select branch_number from branch where branch_name = 'Baneshwor');
Query OK, 1 row affected (0.075 sec)

MariaDB [bank]> select * from account;
+-----+-----+-----+
| account_number | branch_number | balance |
+-----+-----+-----+
| 112404007 | 2430 | 140000.00 |
| 112404017 | 4325 | 145000.00 |
| 112404050 | 4324 | 650000.00 |
| 112404117 | 2325 | 345000.00 |
| 112404217 | 3425 | 345000.00 |
| 112404317 | 2435 | 345000.00 |
| 112504526 | 2345 | 250000.00 |
| 112604080 | 2537 | 450000.00 |
| 1012104526 | 345 | 350000.00 |
| 1012104586 | 2435 | 65000.00 |
| 1012305478 | 6345 | 20000.00 |
| 1012305481 | 8345 | 45000.00 |
| 1012404001 | 4343 | 55000.00 |
| 1012404500 | 4345 | 550000.00 |
| 1012404526 | 5345 | 160000.00 |
| 1012404562 | 8344 | 150000.00 |
| 1012504576 | 345 | 250000.00 |
| 1112403497 | 4537 | 150000.00 |
| 1112404586 | 2430 | 70000.00 |
| 1212104580 | 5340 | 100000.00 |
| 1212305468 | 6344 | 25000.00 |
| 1612103586 | 8245 | 50000.00 |
| 2015404080 | 4320 | 105000.00 |
+-----+-----+-----+
23 rows in set (0.000 sec)
```

60) Delete the records of all accounts with balances below the average at the bank.

```
delete from account where balance < (select avg(balance) from account);
```

```
MariaDB [bank]> delete from account where balance < (select avg(balance) from account);
Query OK, 14 rows affected (0.080 sec)

MariaDB [bank]> select * from account;
+-----+-----+-----+
| account_number | branch_number | balance |
+-----+-----+-----+
| 112404050 | 4324 | 650000.00 |
| 112404117 | 2325 | 345000.00 |
| 112404217 | 3425 | 345000.00 |
| 112404317 | 2435 | 345000.00 |
| 112504526 | 2345 | 250000.00 |
| 112604080 | 2537 | 450000.00 |
| 1012104526 | 345 | 350000.00 |
| 1012404500 | 4345 | 550000.00 |
| 1012504576 | 345 | 250000.00 |
+-----+-----+-----+
9 rows in set (0.001 sec)
```

61) Increase all balance by 5 percent.

```
update account set balance = balance * 1.05;
```

```
MariaDB [bank]> update account set balance = balance * 1.05;
Query OK, 9 rows affected (0.050 sec)
Rows matched: 9  Changed: 9  Warnings: 0

MariaDB [bank]> select * from account;
+-----+-----+-----+-----+
| account_number | branch_number | balance | credits, dept name)
+-----+-----+-----+-----+
| 112404050 | 4324 | 682500.00 | 37, 4, 'Comp. Sci.')
| 112404117 | 2325 | 362250.00 |
| 112404217 | 3425 | 362250.00 |
| 112404317 | 2435 | 362250.00 |
| 112504526 | 2345 | 262500.00 |
| 112604080 | 2537 | 472500.00 |
| 1012104526 | 345 | 367500.00 |
| 1012404500 | 4345 | 577500.00 |
| 1012504576 | 345 | 262500.00 |
+-----+-----+-----+-----+
9 rows in set (0.001 sec)
```

62) Increase balance only to accounts with a balance of 10000 or more by 5 percent.

```
update account set balance = balance*1.05 where balance>=300000;
```

```
MariaDB [bank]> update account set balance = balance*1.05 where balance>=300000;
Query OK, 7 rows affected (0.010 sec)
Rows matched: 7  Changed: 7  Warnings: 0

MariaDB [bank]> select * from account;
+-----+-----+-----+
| account_number | branch_number | balance |
+-----+-----+-----+
| 112404050 | 4324 | 716625.00 |
| 112404117 | 2325 | 380362.50 |
| 112404217 | 3425 | 380362.50 |
| 112404317 | 2435 | 380362.50 |
| 112504526 | 2345 | 262500.00 |
| 112604080 | 2537 | 496125.00 |
| 1012104526 | 345 | 385875.00 |
| 1012404500 | done | 606375.00 |
| 1012504576 | 345 | 262500.00 |
+-----+-----+-----+
9 rows in set (0.001 sec) completed with 3 local
```

Q 49) List the records of view all_customer.

Q 50) Drop all_customer view.

Q 51) Perform the Join operation (Natural join, Left join, Right join, and

63) Pay 5 percent interest on accounts whose balance is greater than average.

```
update account set balance = balance*1.05 where balance > (select
avg(balance) from account);
```

```
MariaDB [bank]> update account set balance = balance*1.05 where balance > (select avg(balance) from account);
Query OK, 3 rows affected (0.729 sec)
Rows matched: 3  Changed: 3  Warnings: 0

MariaDB [bank]> select * from account;
+-----+-----+-----+
| account_number | branch_number | balance |
+-----+-----+-----+
| 112404050 | 4324 | 752456.25 |
| 112404117 | 2325 | 380362.50 |
| 112404217 | 3425 | 380362.50 |
| 112404317 | 2435 | 380362.50 |
| 112504526 | 2345 | 262500.00 |
| 112604080 | 2537 | 520931.25 |
| 1012104526 | 345 | 385875.00 |
| 1012404500 | done | 636693.75 |
| 1012504576 | 345 | 262500.00 |
+-----+-----+-----+
9 rows in set (0.001 sec) completed with 3 local
```

Q 46) Pay 5 percent interest on accounts whose balance is greater than av

Q 47) Update all accounts with balances over 10,00,000 receive 6 percent

64) Update all accounts with balances over 10,00,000 receive 6 percent interest, whereas all others receive 5 percent.

```
update account set balance=case when balance>650000 then balance*1.06
else balance*1.05 end;
```

```

MariaDB [bank]> update account set balance=case when balance>650000 then balance*1.06 else balance*1.05 end;
Query OK, 9 rows affected, 6 warnings (0.055 sec)
Rows matched: 9  Changed: 9  Warnings: 6

MariaDB [bank]> select * from account;
+-----+-----+-----+
| account_number | branch_number | balance |
+-----+-----+-----+
| 112404050 | 4324 | 797603.63 |
| 112404117 | 2325 | 399380.63 |
| 112404217 | 3425 | 399380.63 |
| 112404317 | 2435 | 399380.63 |
| 112504526 | 2345 | 275625.00 |
| 112604080 | 2537 | 546977.81 |
| 1012104526 | 345 | 405168.75 |
| 1012404500 | 4345 | 668528.44 |
| 1012504576 | 345 | 275625.00 |
+-----+-----+-----+
9 rows in set (0.000 sec)

Q 47) Update all accounts with balances over 10,00,000 receive 6 percent interest, receive 5 percent.

Q 48) Create a view called all_customer consisting of branches and their customer's name.

Q 49) List the records of view all_customer.

Q 50) Drop all_customer view.

```

65) Create a view called all_customer consisting of branches and their customer's name.

```

create view all_customer as select
c.customer_number,c.customer_name,b.branch_name from customer as c,
branch as b, account as a, depositor as d where
c.customer_number=d.customer_number and
d.account_number=a.account_number and a.branch_number=b.branch_number;

```

```

MariaDB [bank]> create view all_customer as select c.customer_number,c.customer_name,b.branch_name from customer as c, branch as b, account as a, depositor as d where c.customer_number=d.customer_number and d.account_number=a.account_number and a.branch_number=b.branch_number;
Query OK, 0 rows affected (0.583 sec)

MariaDB [bank]> show tables;
+-----+
| Tables_in_bank |
+-----+
| account |
| all_customer |
| borrower |
| branch |
| customer |
| depositor |
| loan |
+-----+
7 rows in set (0.002 sec)

Q 49) List the records of view all_customer.

Q 50) Drop all_customer view.

```

66) List the records of view all_customer.

```

select * from all_customer;

```

```
MariaDB [bank]> select * from all_customer;
+-----+-----+-----+
| customer_number | customer_name | branch_name |
+-----+-----+-----+
| 15 | Barney Campbell | Sangam Chowk |
| 12 | Bhawana Baral | Maharshi Chowk |
| 16 | Charles Robinson | Saptagandaki Chowk |
| 19 | James Lewis | Sital Mall |
| 22 | Joey Miller | Amarsingh Chowk |
| 14 | Robin Jhonson | Bhrikuti Marg |
| 17 | Thomas Cooper | Ram Chowk |
+-----+-----+-----+
7 rows in set (0.177 sec)
```

67) Drop all_customer view.

```
drop view all_customer;
```

```
MariaDB [bank]> drop view all_customer;
Query OK, 0 rows affected (0.092 sec)

MariaDB [bank]> show tables;
+-----+
| Tables_in_bank |
+-----+
| account |
| borrower |
| branch |
| customer |
| depositor |
| loan |
+-----+
6 rows in set (0.001 sec)
```

Q 50) Drop all_customer view.
Q 51) Perform the Join operation (Natural join, Left join, Right join, and Full join) in the tables account and customer tables.

68) Perform the Join operation (Natural join, Left join, Right join, and Full join) in the tables account and customer tables.

```
-- Natural Join
select * from account natural join depositor natural join customer;

-- left join
select * from account left join depositor on
account.account_number=depositor.account_number left join customer on
depositor.customer_number=customer.customer_number;

-- Right Join
select * from account right join depositor on
account.account_number=depositor.account_number right join customer on
depositor.customer_number=customer.customer_number;

-- Full Join
```

```

select * from account full join depositor on
account.account_number=depositor.account_number full join customer on
depositor.customer_number=customer.customer_number;

-- "One that doesnot support full join"
(select * from account left join depositor on
account.account_number=depositor.account_number left join customer on
depositor.customer_number=customer.customer_number) union (select *
from account right join depositor on
account.account_number=depositor.account_number right join customer on
depositor.customer_number=customer.customer_number);

```

```

MariaDB [bank]> SELECT * FROM account
--> NATURAL JOIN depositor
--> NATURAL JOIN customer;
+-----+-----+-----+-----+-----+-----+-----+
| customer_number | account_number | branch_number | balance | customer_name | customer_street | customer_city |
+-----+-----+-----+-----+-----+-----+-----+
|      15 | 1012404500 |          4345 | 668528.44 | Barney Campbell | Sangam Chowk | Hetauda
|      12 | 1012504576 |          345 | 275625.00 | Bhawana Baral | Maharshi Chowk | Dhamauli
|      16 | 112504526 |          2345 | 275625.00 | Charles Robinson | Saptagandaki Chowk | Bharatpur
|      19 | 112404050 |          4324 | 797603.63 | James Lewis | Sital Mall | Hetauda
|      22 | 112404117 |          2325 | 399380.63 | Joey Miller | Amarsingh Chowk | Pokhara
|      14 | 112604080 |          2537 | 546977.81 | Robin Jhonson | Bhrikuti Marg | Dharan
|      17 | 112404217 |          3425 | 399380.63 | Thomas Cooper | Ram Chowk | Janakpur
+-----+-----+-----+-----+-----+-----+-----+
7 rows in set (0.001 sec)

drop view all_customers;

MariaDB [bank]> select * from account left join depositor on account.account_number=depositor.account_number left join customer on depositor.customer_number=customer.customer_number;
+-----+-----+-----+-----+-----+-----+-----+
| account_number | branch_number | balance | customer_number | account_number | customer_number | customer_name | customer_street | customer_city |
+-----+-----+-----+-----+-----+-----+-----+
| 112404050 |          4324 | 797603.63 |         19 | 112404050 |          19 | James Lewis | Sital Mall | Hetauda
| 112404117 |          2325 | 399380.63 |         22 | 112404117 |          22 | Joey Miller | Amarsingh Chowk | Pokhara
| 112404217 |          3425 | 399380.63 |         17 | 112404217 |          17 | Thomas Cooper | Ram Chowk | Janakpur
| 112404317 |          2435 | 399380.63 |        NULL | 112504526 |          16 | Charles Robinson | Saptagandaki Chowk | Bharatpur
| 112504526 |          2345 | 275625.00 |        NULL | 112504526 |          16 | (Q:5) Perform NULL | NULL | NULL
| 112604080 |          2537 | 546977.81 |         14 | 112604080 |          14 | Robin Jhonson | Bhrikuti Marg | Dharan
| 1012104526 |          345 | 405168.75 |        NULL | 1012404500 |          15 | Barney Campbell | Sangam Chowk | Hetauda
| 1012404500 |          4345 | 668528.44 |        NULL | 1012404500 |          15 | Barney Campbell | Sangam Chowk | Hetauda
| 1012504576 |          345 | 275625.00 |        NULL | 1012504576 |          12 | Bhawana Baral | Maharshi Chowk | Dhamauli
+-----+-----+-----+-----+-----+-----+-----+
9 rows in set (0.019 sec)

```

MariaDB [bank]> select * from account right join depositor on account.account_number=depositor.account_number right join customer on depositor.customer_number=customer.customer_number;										
account_number	branch_number	balance	customer_number	account_number	customer_number	customer_name	customer_street	customer_city	customer_id	customer_type
NULL	NULL	NULL	NULL	NULL	8	Aagnam Poudel	Tinkunk	Pokhara	101244500	Businessman
101244500	4345	668528.44	15	101244500	15	Barney Campbell	Sangam Chowk	Hetauda	101244570	Businessman
1012584570	215	275625.90	12	1012584570	12	Bhawana Baral	Maharishi Chowk	Dhamaili	NULL	Businessman
NULL	NULL	NULL	NULL	NULL	24	Burno Cordoba	Shrestha	Pokhara	112584526	Businessman
112584526	2345	275625.00	16	112584526	16	Charles Robinson	Saptagandaki Chowk	Bhatarpur	NULL	Businessman
NULL	NULL	NULL	NULL	NULL	19	David Lewis	Birendra Path	Hetauda	NULL	Businessman
NULL	NULL	NULL	NULL	NULL	2	Hari Prasad Baral	Bageshwar Marg	Kathmandu	112484050	Businessman
112484050	4324	797603.63	19	112484050	19	Jones Lewis	Sital Marg	Kathmandu	NULL	Businessman
NULL	NULL	NULL	NULL	NULL	3	Jhon Cordoba	Baneshwor	Kathmandu	112484117	Businessman
112484117	2325	399380.63	22	112484117	22	Joey Miller	Amarasingh Chowk	Pokhara	NULL	Businessman
NULL	NULL	NULL	NULL	NULL	23	Joy Miller	mai road	Pokhara	NULL	Businessman
NULL	NULL	NULL	NULL	NULL	7	Prajwol Khan	Sabaghriha Chowk	Pokhara	NULL	Businessman
NULL	NULL	NULL	NULL	NULL	13	Rena Baral	Shukra Path	Biratnagar	NULL	Businessman
NULL	NULL	NULL	NULL	NULL	21	Rishab Khan	Hospital Chowk	Pokhara	NULL	Businessman
NULL	NULL	NULL	NULL	NULL	11	Riva Chalise	Sahid Chowk	Narayanghat	NULL	Businessman
NULL	NULL	NULL	NULL	NULL	20	Robert Allen	School Road	Hetauda	112684080	Businessman
112684080	2537	546977.81	14	112684080	14	Robin Jhonson	Bhrikuti Marg	Dharan	NULL	Businessman
NULL	NULL	NULL	NULL	NULL	5	Ross Smith	Nakidot	Lalitpur	NULL	Businessman
NULL	NULL	NULL	NULL	NULL	10	Roxane Chalise	Ichchhakamana	Narayanghat	NULL	Businessman
NULL	NULL	NULL	NULL	NULL	9	Sanskriti Giri	Lakeside	Pokhara	NULL	Businessman
NULL	NULL	NULL	NULL	NULL	1	Shyam Sharma	Shivam Tol	Kathmandu	NULL	Businessman
NULL	NULL	NULL	NULL	NULL	6	Ted Wilson	Bagar	Pokhara	112484217	Businessman
112484217	3425	399380.63	17	112484217	17	Thomas Cooper	Ram Chowk	Janakpur	NULL	Businessman
NULL	NULL	NULL	NULL	NULL	4	Wayne Rooney	Kupandol	Lalitpur	NULL	Businessman

account_number	branch_number	balance	customer_number	account_number	customer_number	customer_name	customer_street	customer_city
112404050	4324	797683.63	19	112404050	19	James Lewis	Sital Mall	Hetauda
112404117	2325	399380.63	22	112404117	22	Joey Miller	Amarsingh Chowk	Pokhara
112404217	3425	399380.63	17	112404217	17	Thomas Cooper	Ram Chowk	Janakpur
112404317	2435	399380.63	NULL	NULL	NULL	NULL	NULL	NULL
112504526	2345	275625.00	16	112504526	16	Charles Robinson	Saptagandaki Chowk	Bharatpur
112604080	2537	546977.81	14	112604080	14	Robin Jhonson	Bhrikuti Marg	Dharam
1012104526	345	405168.75	NULL	NULL	NULL	NULL	NULL	NULL
1012404500	4345	668528.44	15	1012404500	15	Barney Campbell	Sangam Chowk	Hetauda
1012504576	345	275625.00	32	1012504576	12	Bhawana Baral	Maharshi Chowk	Dhamauli
NULL	NULL	NULL	NULL	NULL	8	Aagman Poudel	Tinkune	Pokhara
NULL	NULL	NULL	NULL	NULL	24	Burno Cordoba	New road	Pokhara
NULL	NULL	NULL	NULL	NULL	18	David Lae	Birendra Path	Hetauda
NULL	NULL	NULL	NULL	NULL	2	Hari Parsad Baral	Bagmara Marg	Kathmandu
NULL	NULL	NULL	NULL	NULL	3	Jhon Cordoba	Baneshwor	Kathmandu
NULL	NULL	NULL	NULL	NULL	23	Joy Miller	main road	Pokhara
NULL	NULL	NULL	NULL	NULL	7	Prajwol Khan	Sabaghina Chowk	Pokhara
NULL	NULL	NULL	NULL	NULL	13	Reena Baral	Shukra Path	Biratnagar
NULL	NULL	NULL	NULL	NULL	21	Rishab Khan	Hospital Chowk	Pokhara
NULL	NULL	NULL	NULL	NULL	11	Riva Chalise	Sahid Chowk	Narayanghat
NULL	NULL	NULL	NULL	NULL	20	Robert Allen	School Road	Hetauda
NULL	NULL	NULL	NULL	NULL	5	Ross Smith	Nakidot	Lalitpur
NULL	NULL	NULL	NULL	NULL	10	Roxane Chalise	Ichchhakamanan	Narayanghat
NULL	NULL	NULL	NULL	NULL	9	Sanskriti Giri	Lakeside	Pokhara
NULL	NULL	NULL	NULL	NULL	1	Shyan Sharma	Shivam Tol	Kathmandu
NULL	NULL	NULL	NULL	NULL	6	Ted Wilson	Bagar	Pokhara
NULL	NULL	NULL	NULL	NULL	4	Wynae Rooney	Kupandol	Lalitpur