DBMS LAB EXPERIMENTS

Experiment 1:

You need to create new database and insert below data to three different tables.

Data Dictionary:

Create three tables with below schema:

Sailors (sid: integer, sname: string, rating: integer, age: real)

Boats (bid: integer, bname:string, color:string)

Reserves (sid: integer, bid: integer, day: date)

Source code:

Create database record;

use record;

create table sailors (sid integer not null, sname varchar(32), rating integer,age real, CONSTRAINT PK_sailors PRIMARY KEY (sid));

INSERT INTO sailors(sid, sname, rating, age) VALUES (22, 'Rahul', 10, 25.0), (29, 'Anand', 9, 26.0), (31, 'Niket', 7, 45.0), (32, 'Neha',5,56.6),

(58, 'Bijay', 4, 23.5), (64, 'Thomas', 8, 35.0), (71, 'Rusty', 8, 29.5), (74, 'Venky', 6, 63.5), (85, 'Alfred', 1, 42.5), (95, 'Vikky', 9, 24.5);

create table boats (bid integer not null, boatname varchar(45), color varchar(32), constraint PK_boats primary key (bid));

INSERT INTO boats (bid, boatname, color) VALUES (101, 'Waterking', 'Red'), (102, 'Waterking', 'Blue'), (103, 'Marine', 'Red'), (104, 'Seaway', 'Green');

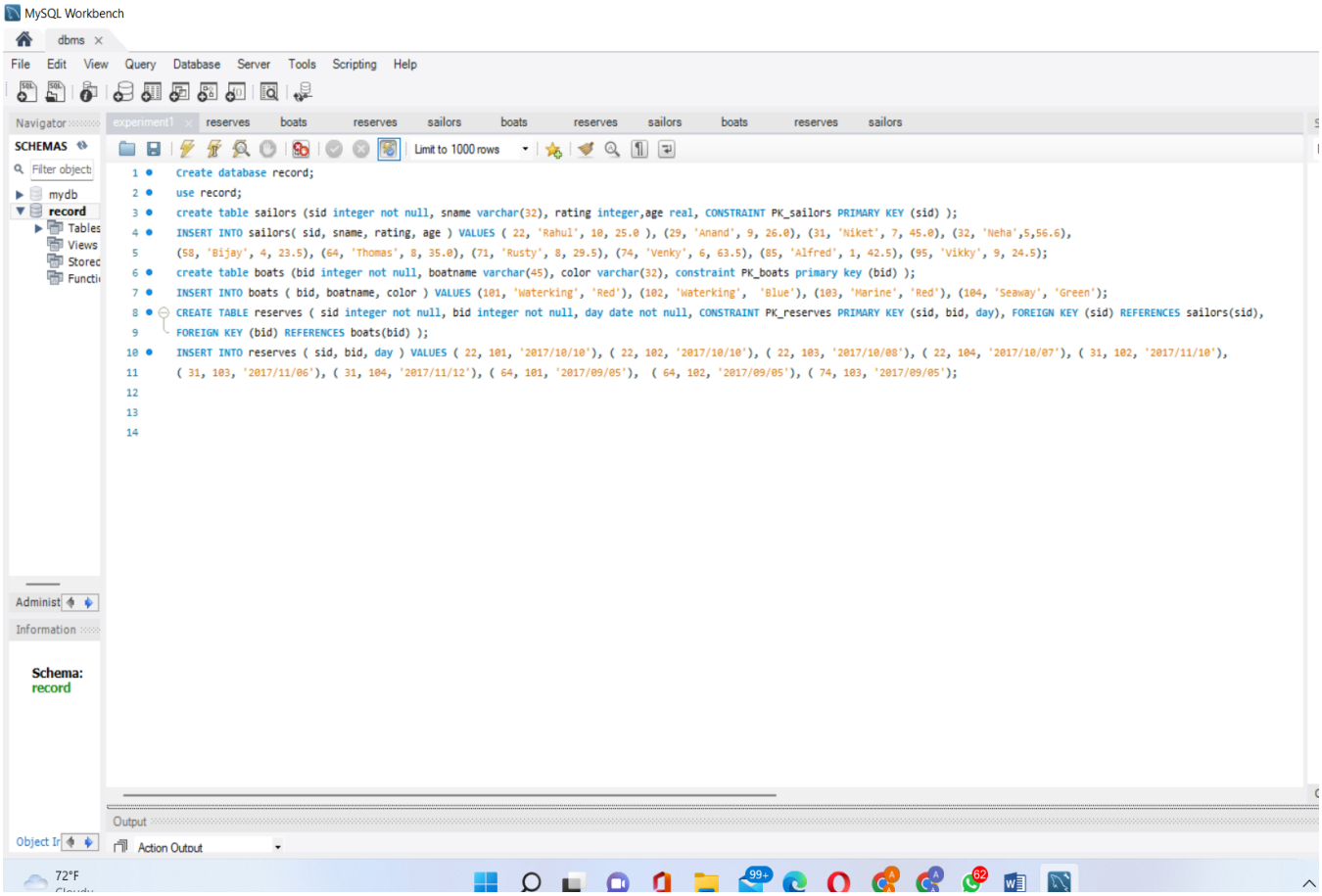
CREATE TABLE reserves (sid integer not null, bid integer not null, day date not null, CONSTRAINT PK_reserves PRIMARY KEY (sid, bid, day), FOREIGN KEY (sid) REFERENCES sailors(sid),

FOREIGN KEY (bid) REFERENCES boats(bid));

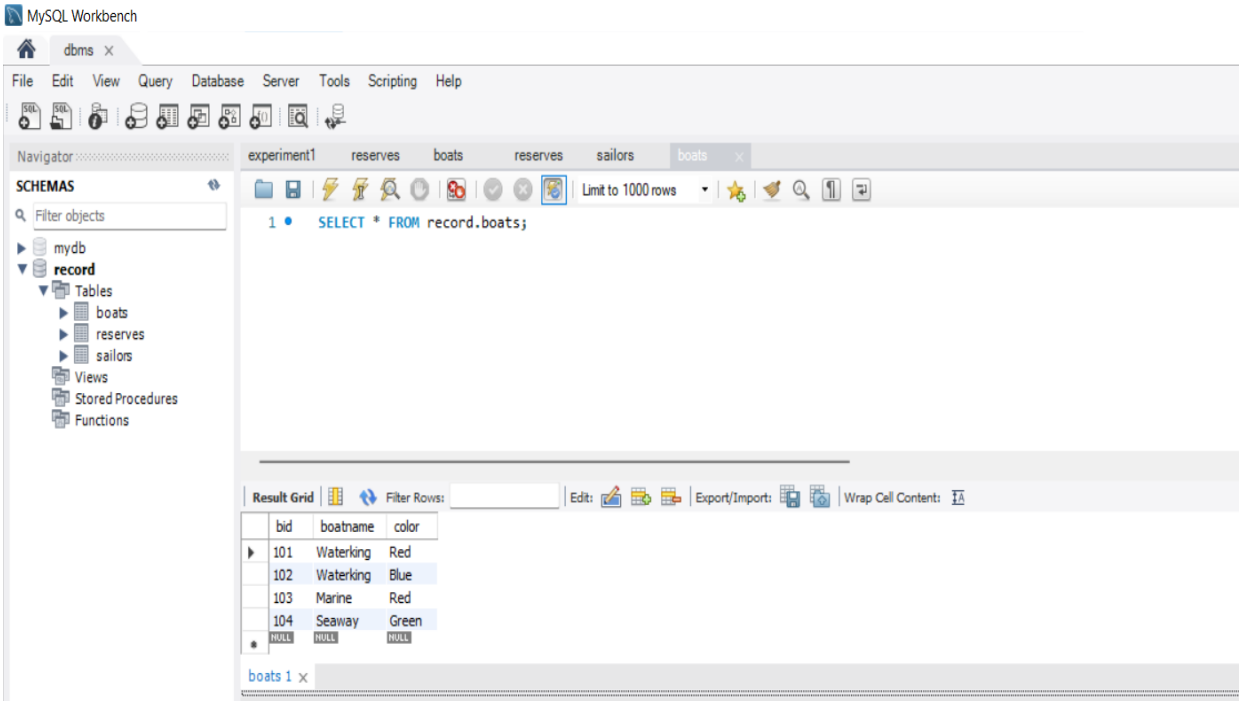
INSERT INTO reserves (sid, bid, day) VALUES (22, 101, '2017/10/10'), (22, 102, '2017/10/10'), (22, 103, '2017/10/08'), (22, 104, '2017/10/07'), (31, 102, '2017/11/10'),

(31, 103, '2017/11/06'), (31, 104, '2017/11/12'), (64, 101, '2017/09/05'), (64, 102, '2017/09/05'), (74, 103, '2017/09/05');

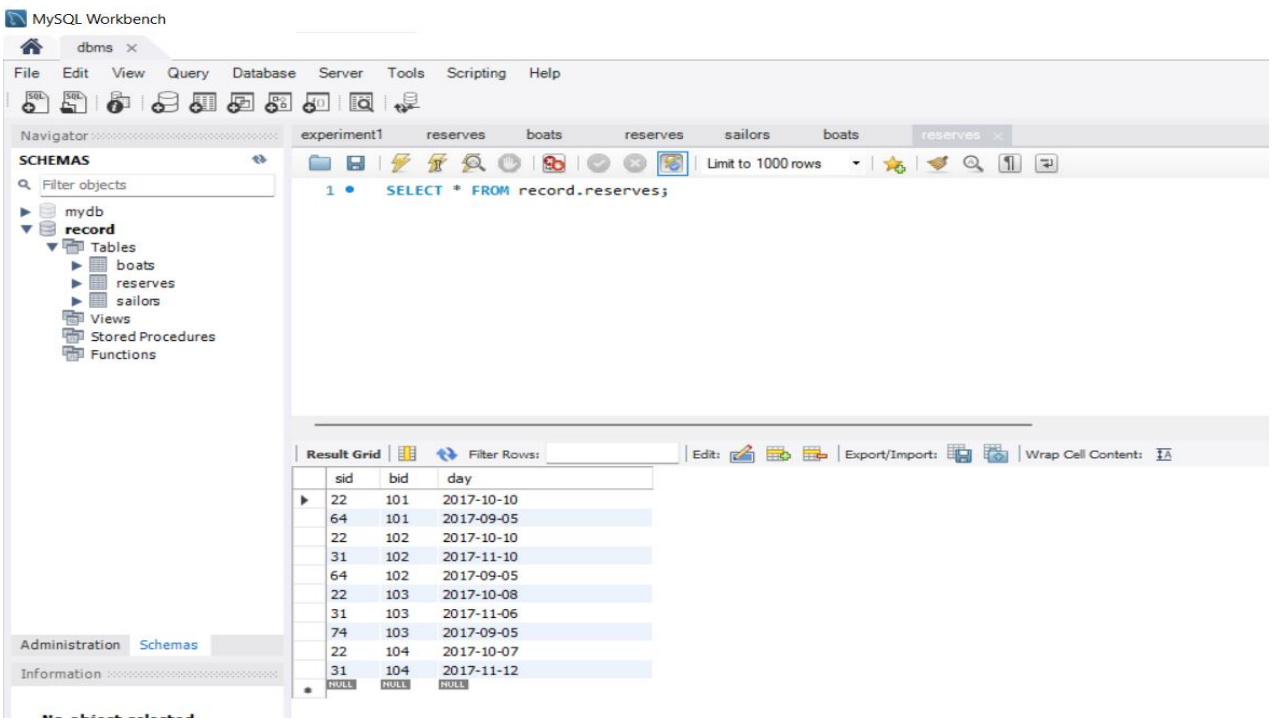
Output:



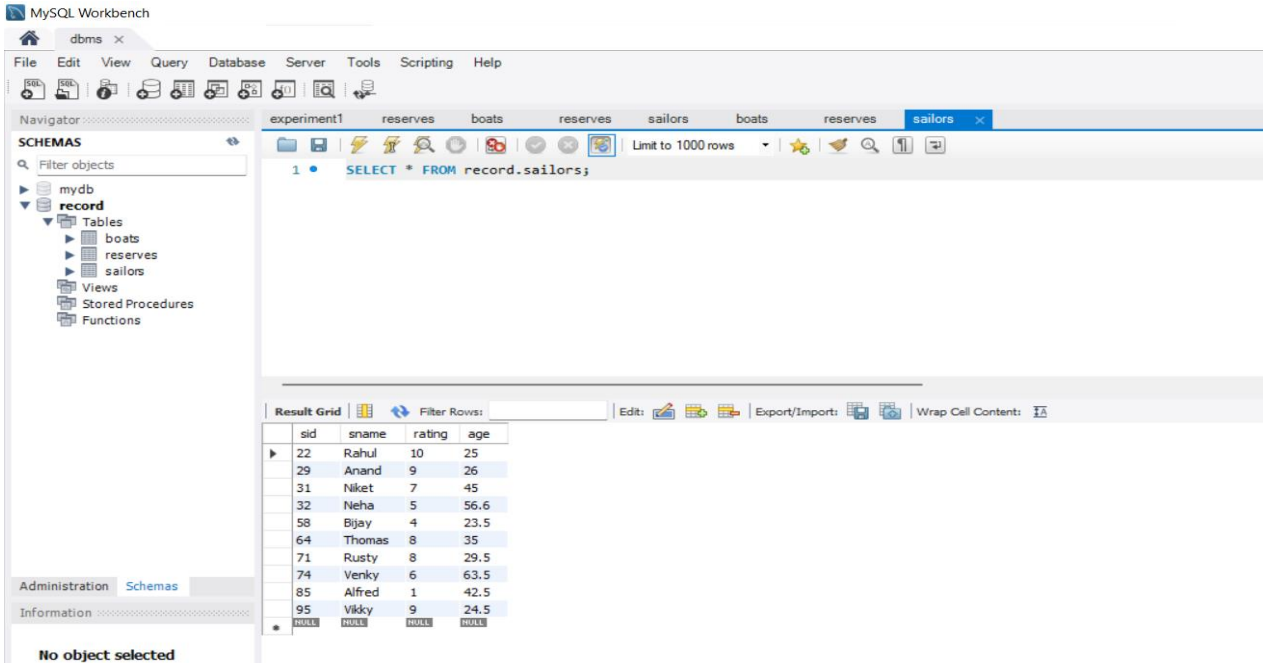
Boats Table:



Reserves Table:



Sailors table:

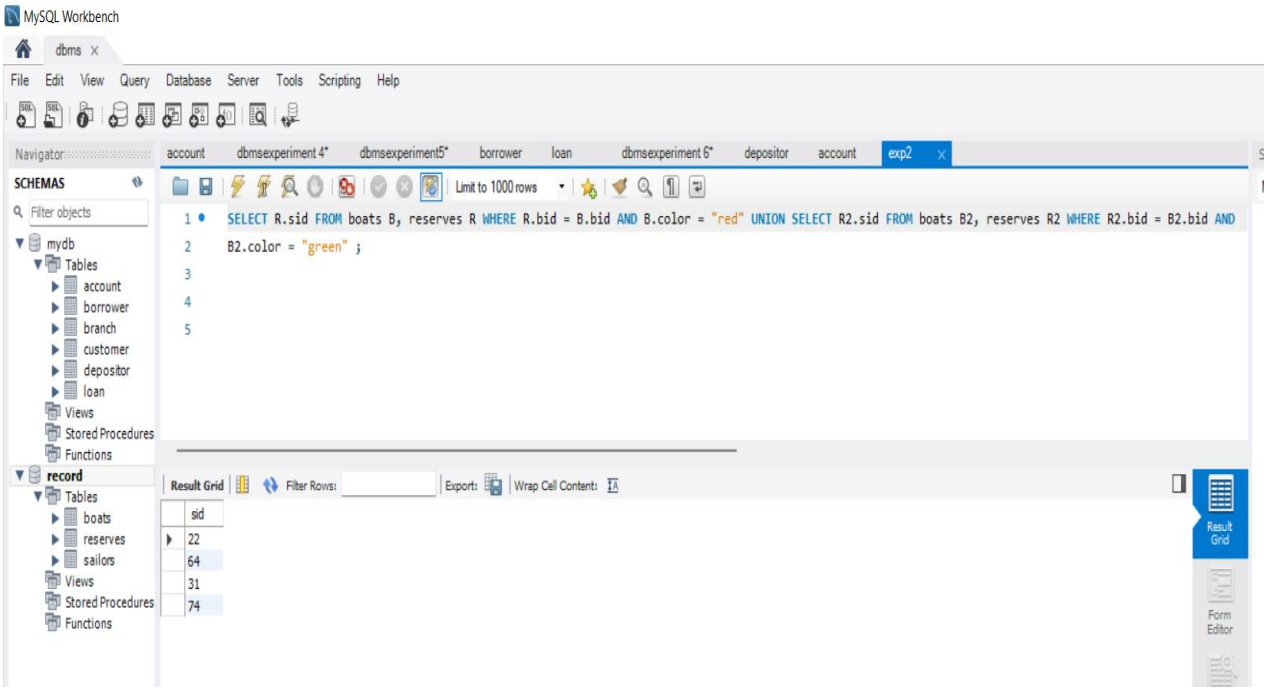


Lab Experiment 2:

You need to find solution of below questions based on Experiment 1.

1. Find the names and sids of sailors who have reserved a red or a Green boat?

SELECT R.sid FROM boats B, reserves R WHERE R.bid = B.bid AND B.color = "red" UNION SELECT R2.sid FROM boats B2, reserves R2 WHERE R2.bid = B2.bid AND B2.color = "green" ;



2. Find the names and sids of sailors who have reserved a red and a Green boat?

3. Find all sids of sailors who have a rating of 10 or reserved boat 104?

4. Find the names of sailors who have reserved boat 103?

5. Find the names of sailors who have reserved a red boat?

6. Find the names of sailors who have not reserved a red boat?

7. Find sailors whose rating is better than some sailor called Horatio?

8. Find the sailors with the highest rating?

9. Find the names of sailors who have reserved both a red and a green boat?

10. Find the names of sailors who have reserved boat no 103?

11. Find the names of sailors who have reserved all boats?

Lab Experiment 3:

Experiment 3 to Experiment 8 are related to each other. You need to follow same data and table details for the solution. You need create new database with below table details and have to insert data based on below tables.

Branch Schema <branch-name, branch-city, assets>

Customer Schema <customer-name, customer-street, customer-city>

Loan Schema <loan-number, branch-name, amount>

Borrower Schema <customer-name, loan-number>

Account Scheme <account-number, branch-name, balance>

Depositor Scheme <customer-name, account-number>

Source code:

```
create database mydb;

use mydb;
```

Create table branch (branch_name varchar(20), branch_city varchar(20), assets integer);

insert into branch(branch_name, branch_city, assets) values("Brighton", "Brooklyn", 7100000),("Downtown", "Brooklyn", 9000000),("Mianus", "Horseneck", 3700000),("NorthTown", "Rye", 400000),("Perryridge", "Horseneck", 1700000),("Pownal", "Bennington", 300000),("Redwood", "Palo Alto", 2100000),("Round Hill", "Horseneck", 800000);

create table customer(customer_name varchar(20), customer_street varchar(20), customer_city varchar(20));

insert into customer(customer_name, customer_street, customer_city) values("Adams", "Spring","Pittsfield"),("Brooks", "Senator","Brooklyn"),("Curry", "North","Rye"),("Glenn", "Sand Hill","Woodside"),("Green", "Walnut","stamford"),("Hayes", "Main","Harrison"),("Johnson", "Alma","Palo Alto"),("Jones", "Main","Harrison"),("Lindsay", "Park","Pittsfield"),("Turner", "Putnam","Stamford"),("Smith", "North","Rye"),("Williams", "Nassau","Princeton");

create table loan(loan_number varchar(20) primary key, branch_name varchar(20), amount integer);

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

create table borrower(customer_name varchar(20), loan_number varchar(20), foreign key (loan_number) References loan(loan_number));

insert into borrower(customer_name, loan_number) values("Adams", "L-16"), ("Curry", "L-93"), ("Hayes", "L-15"), ("Jackson", "L-14"), ("Jones", "L-17"), ("Smith", "L-11"), ("Smith", "L-23"), ("Williams", "L-17");

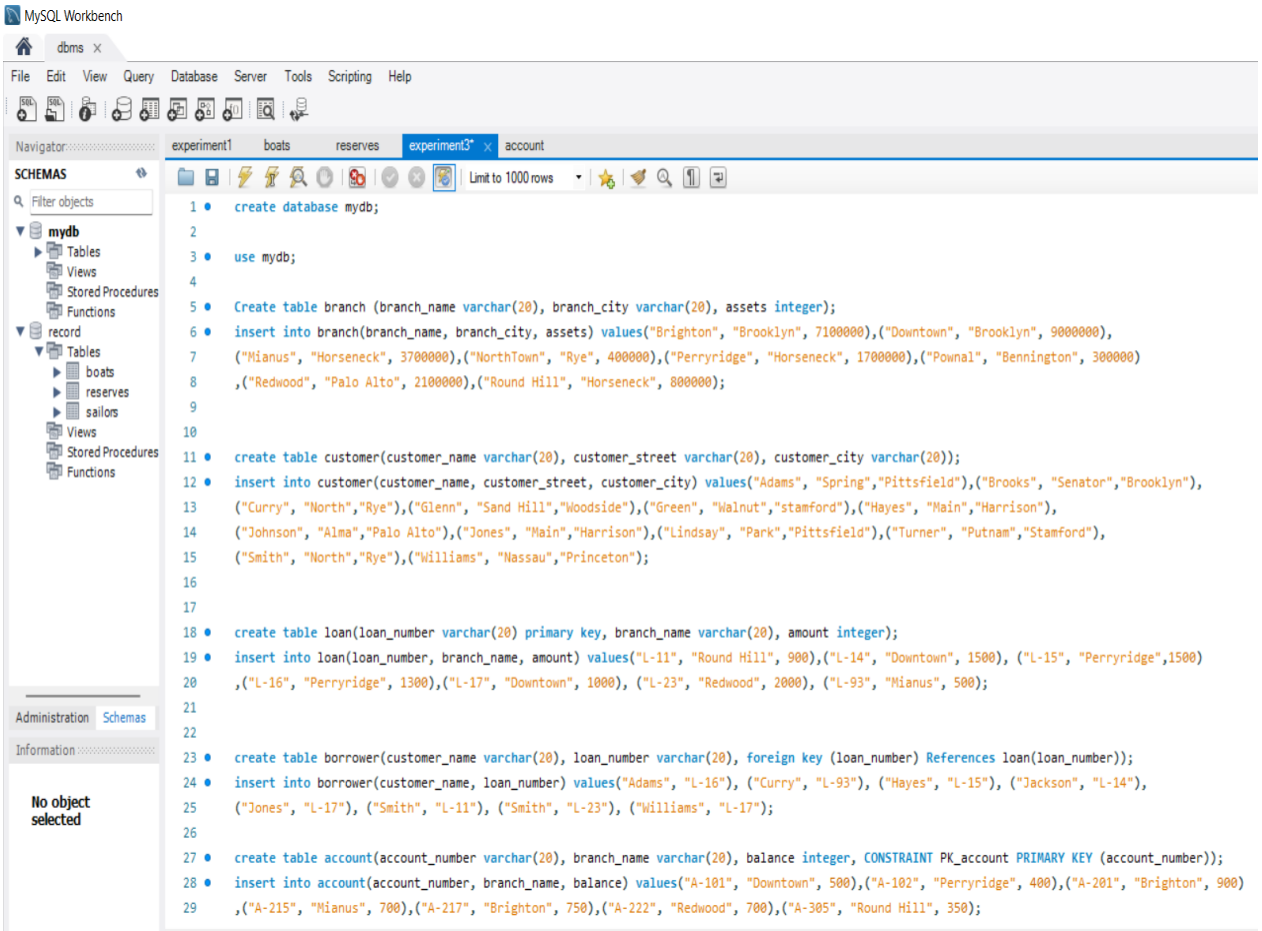
create table account(account_number varchar(20), branch_name varchar(20), balance integer, CONSTRAINT PK_account PRIMARY KEY (account_number));

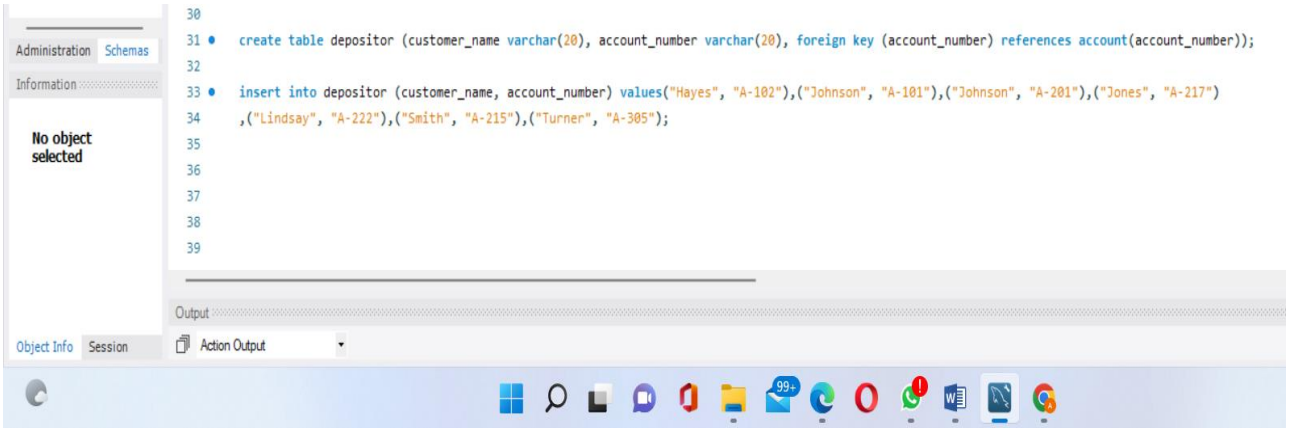
insert into account(account_number, branch_name, balance) values("A-101", "Downtown", 500),("A-102", "Perryridge", 400),("A-201", "Brighton", 900),("A-215", "Mianus", 700),("A-217", "Brighton", 750),("A-222", "Redwood", 700),("A-305", "Round Hill", 350);

create table depositor (customer_name varchar(20), account_number varchar(20), foreign key (account_number) references account(account_number));

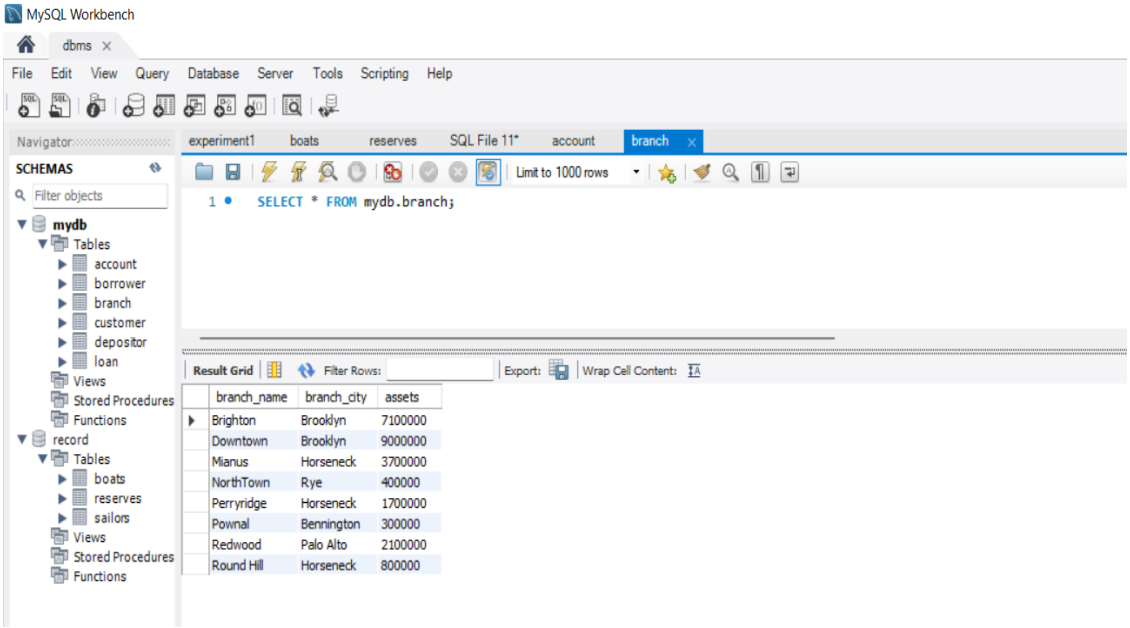
insert into depositor (customer_name, account_number) values("Hayes", "A-102"),("Johnson", "A-101"),("Johnson", "A-201"),("Jones", "A-217"),("Lindsay", "A-222"),("Smith", "A-215"),("Turner", "A-305");

Output:

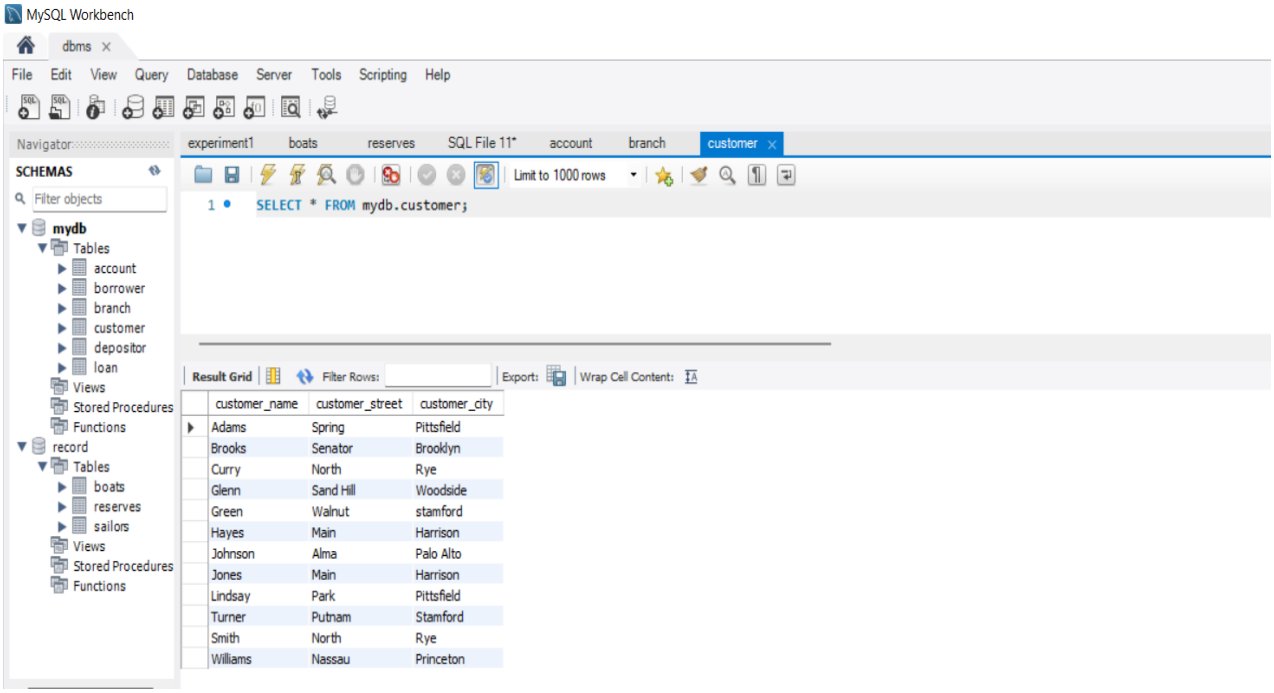




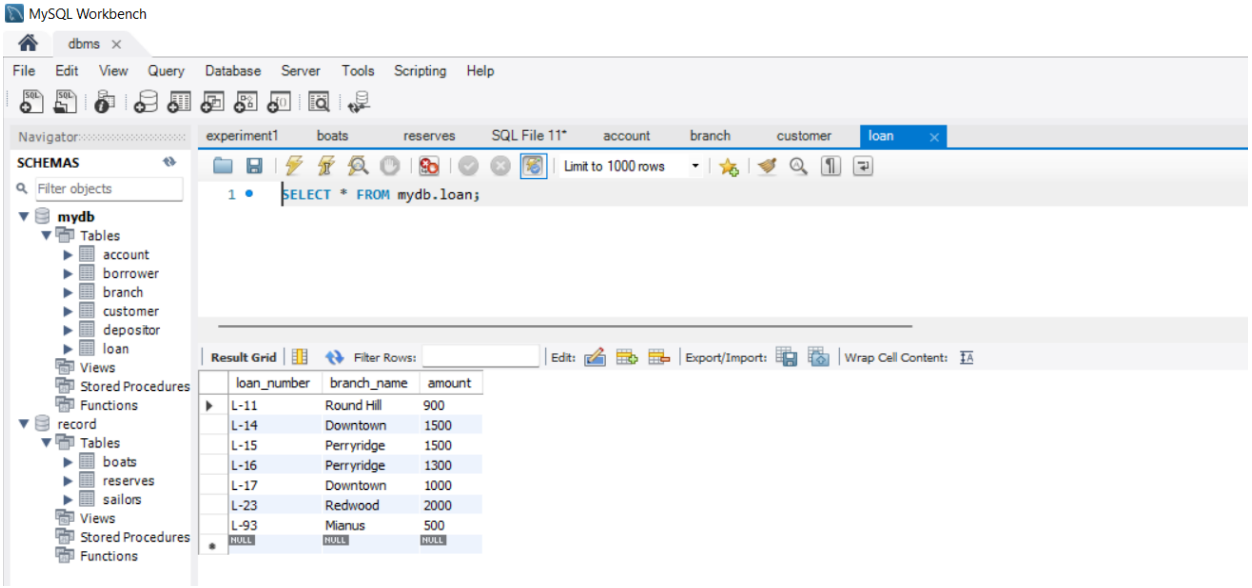
Branch table:



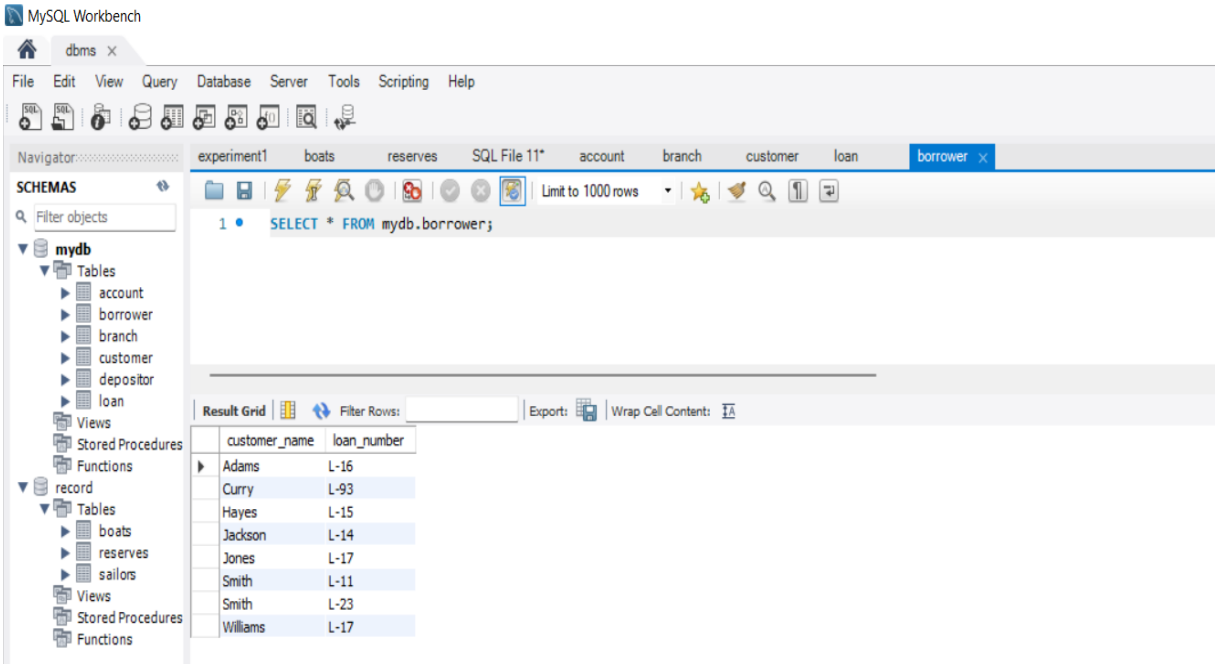
Customer table:



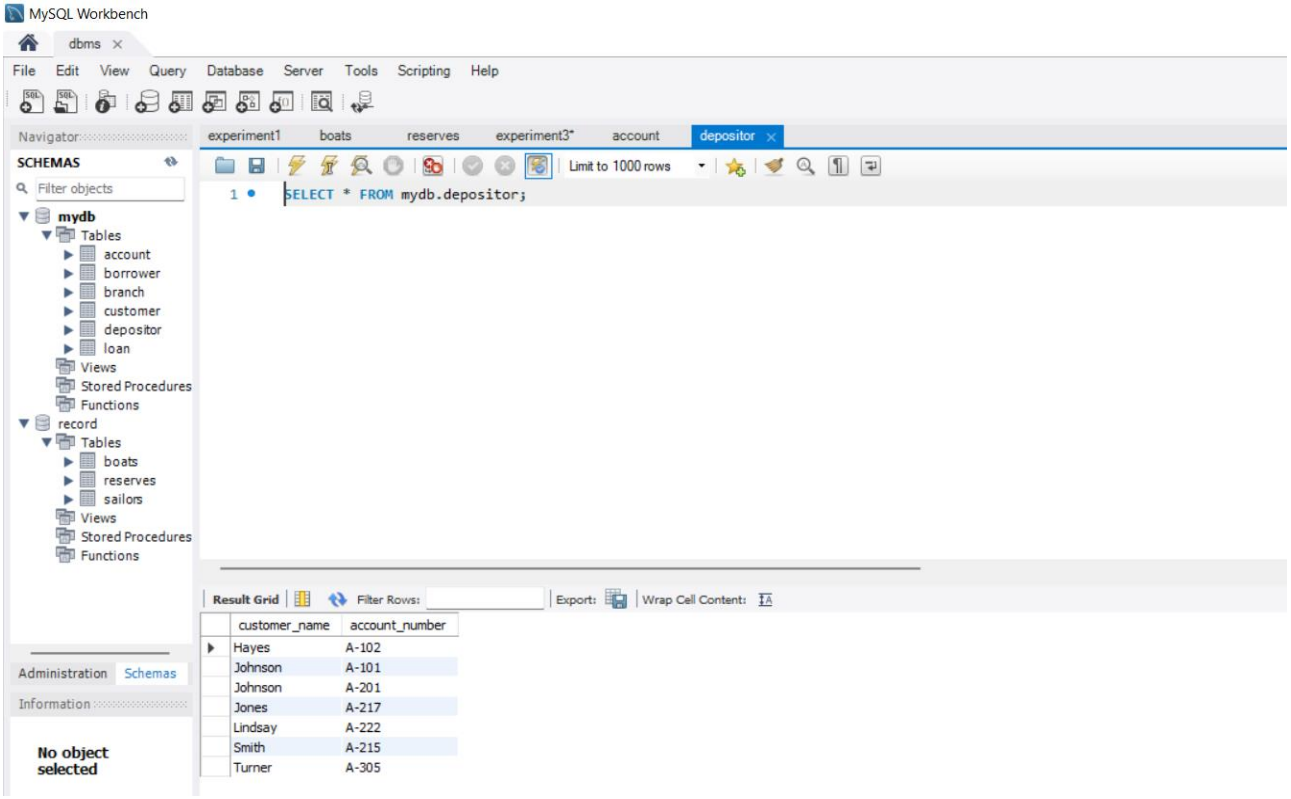
Loan Table:



Borrow Table:



Deposit Table:

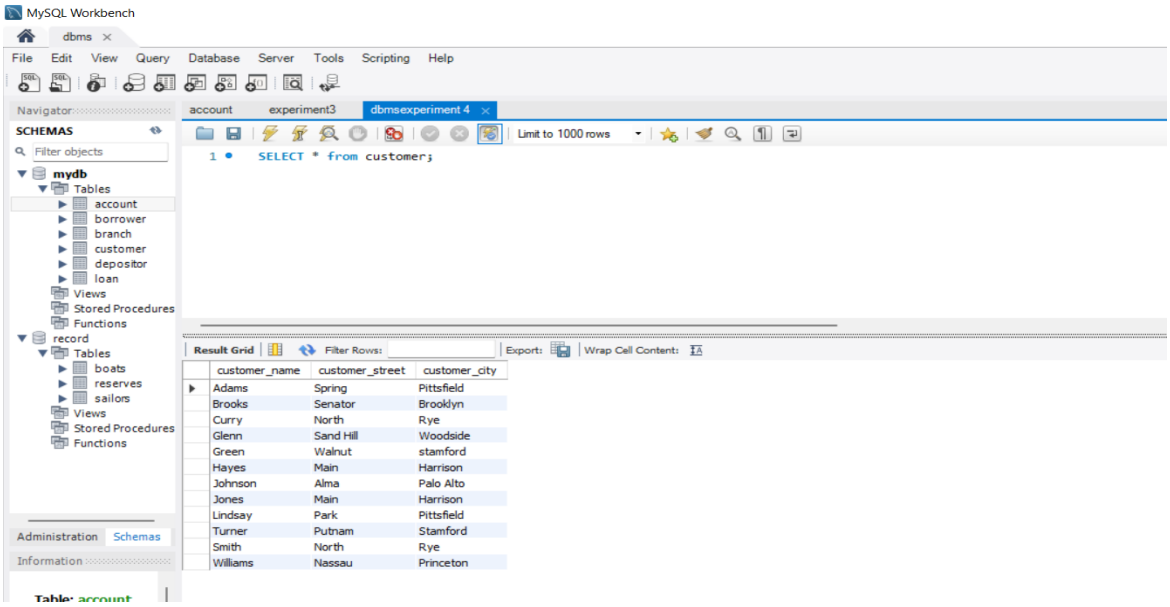


Lab Experiment 4:

You need to find solution of below questions based on tables using Experiment Number 03.

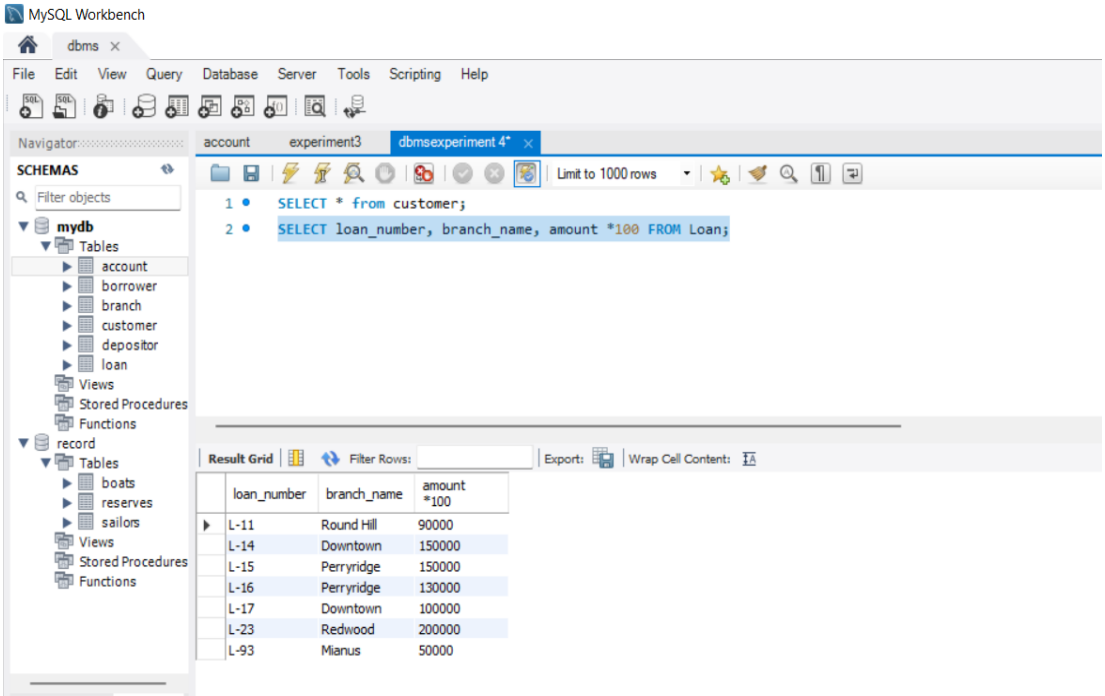
1.To list all the fields from the table Customer.

SELECT * from customer;



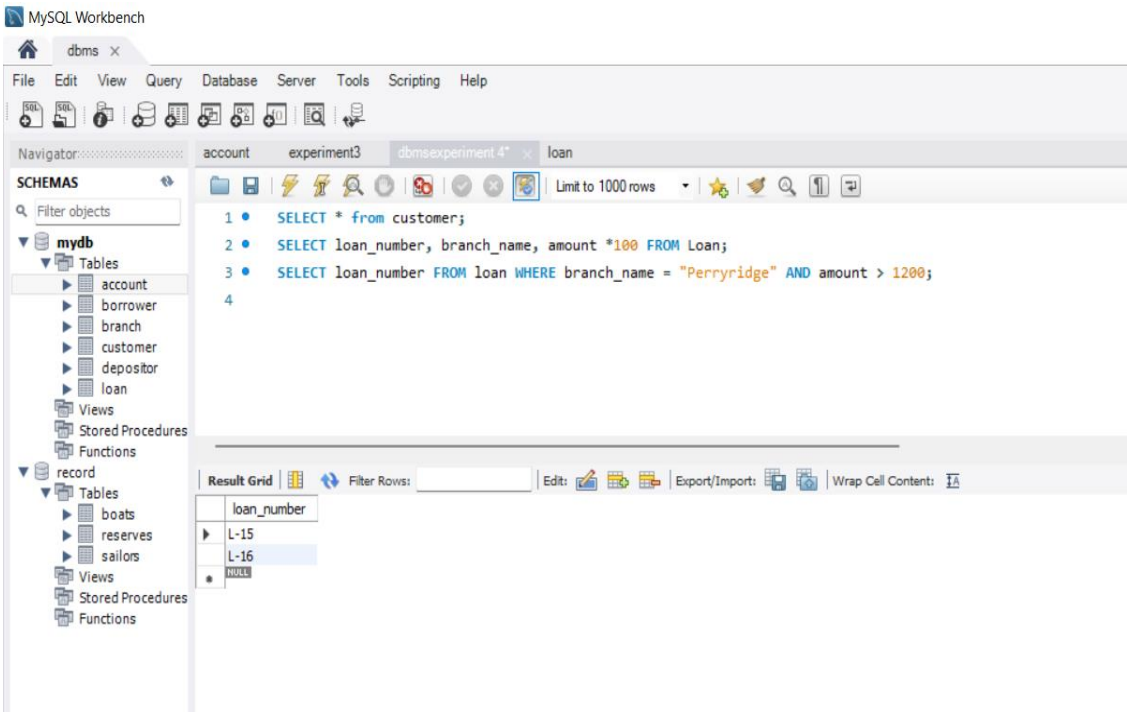
2.To list all the fields after applying arithmetic operations on column amount (amount*100).

SELECT loan_number, branch_name, amount *100 FROM Loan;



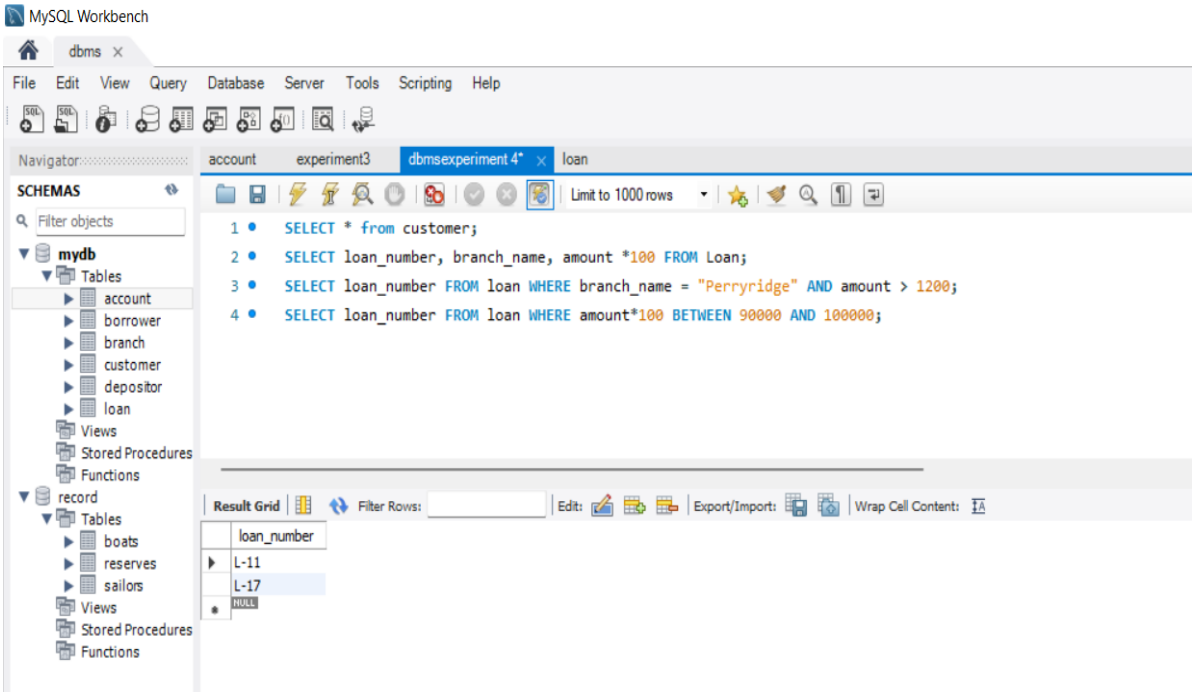
3.Find all loan numbers for loans made at the Perryridge branch withloan amounts greater than Rs1200.

SELECT loan_number FROM loan WHERE branch_name = "Perryridge" AND amount > 1200;



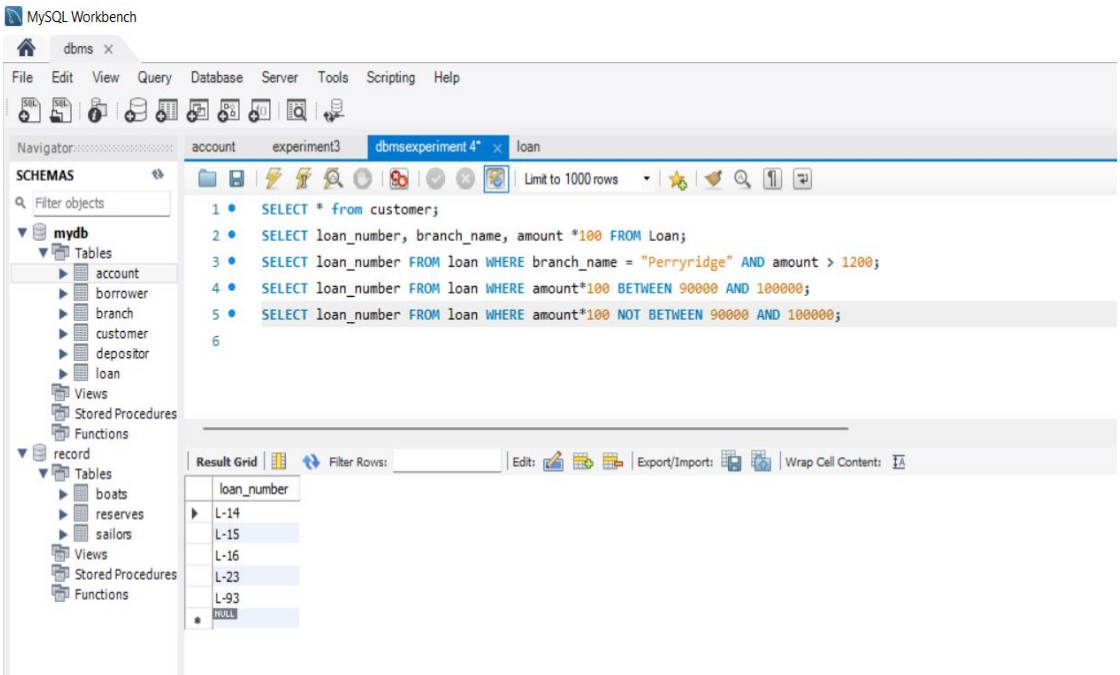
4. Find all loan numbers for loans with loan amounts between Rs90,000and Rs100,000.(amount*100).

SELECT loan_number FROM loan WHERE amount*100 BETWEEN 90000 AND 100000;



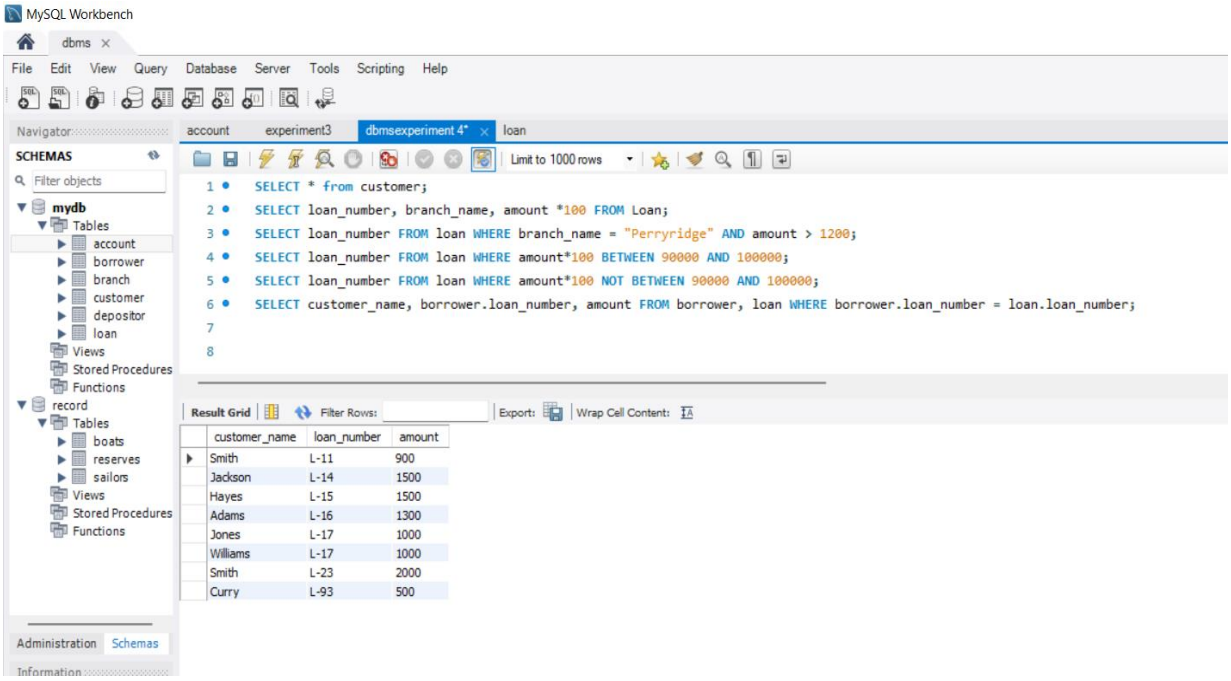
5. Find all loan numbers for loans with loan amounts not betweenRs90,000 and Rs100,000. (amount*100)

SELECT loan_number FROM loan WHERE amount*100 NOT BETWEEN 90000 AND 100000;



6. For all customers who have a loan from the bank, find their names, loan numbers and loan amounts.

SELECT customer_name, borrower.loan_number, amount FROM borrower, loan WHERE borrower.loan_number = loan.loan_number;

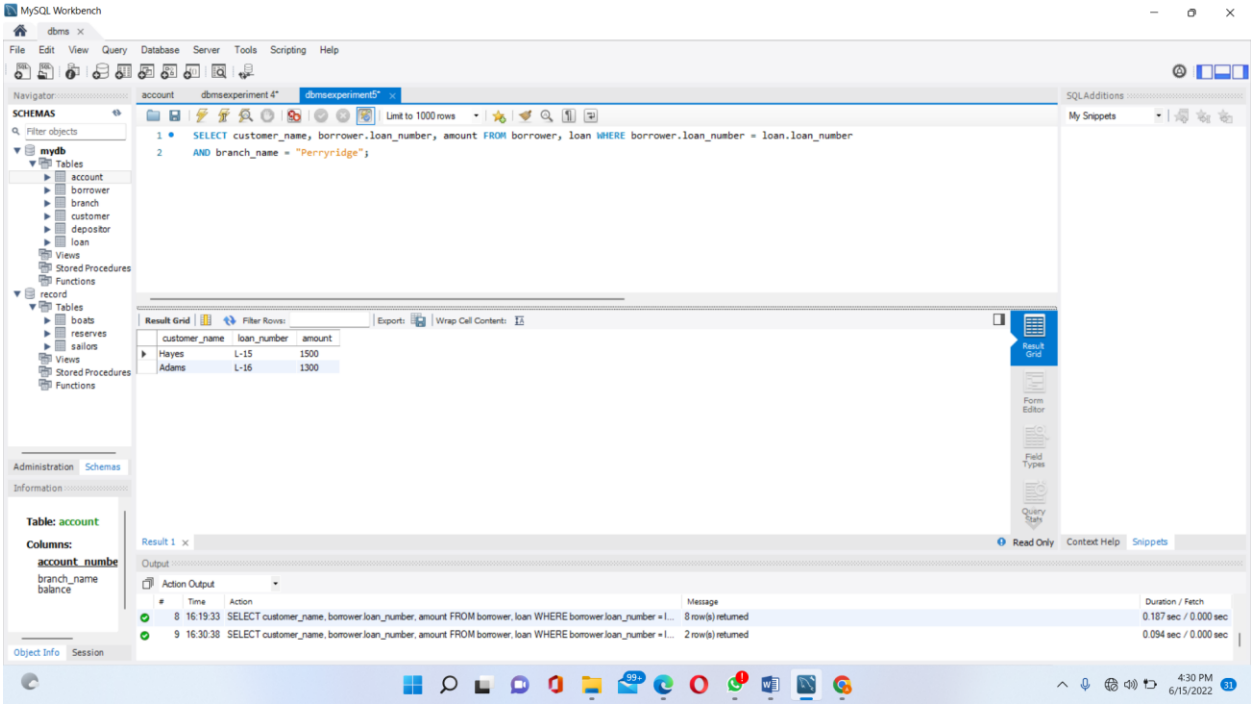


Lab Experiment 5:

You need to find solution of below questions based on tables using Experiment Number 03.

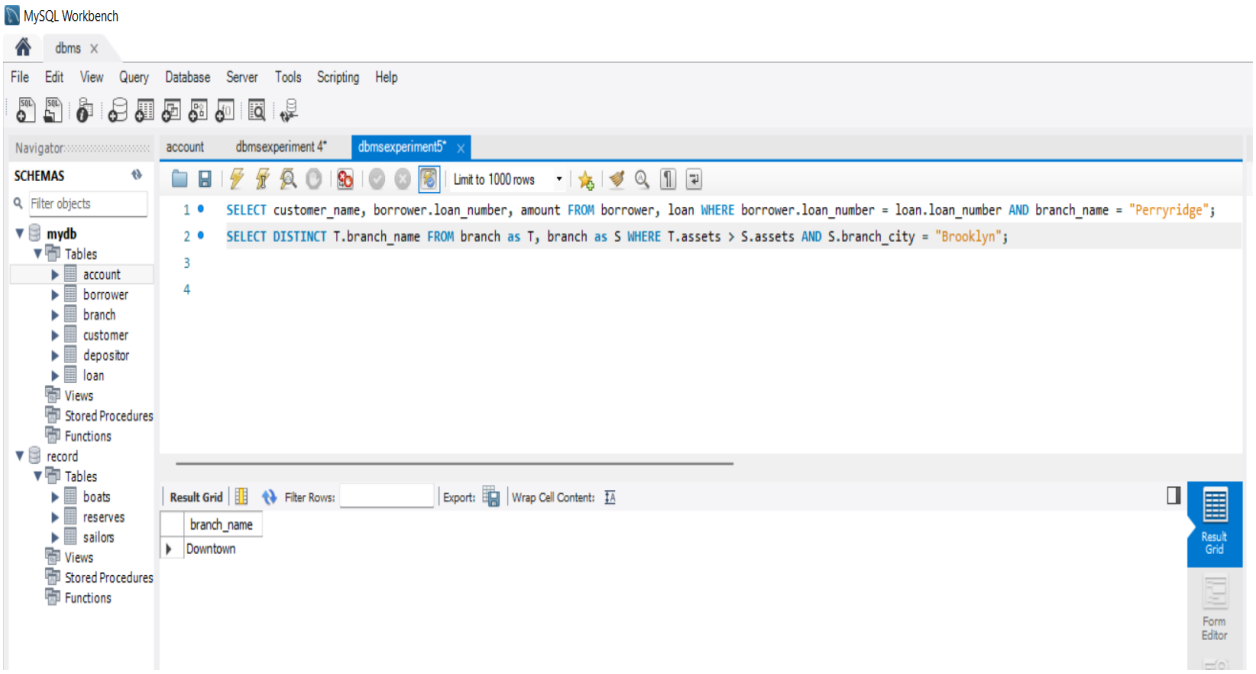
1. Find the customer names, loan numbers and loan amounts for all loans at the Perryridge branch.

SELECT customer_name, borrower.loan_number, amount FROM borrower, loan WHERE borrower.loan_number = loan.loan_number AND branch_name = "Perryridge";



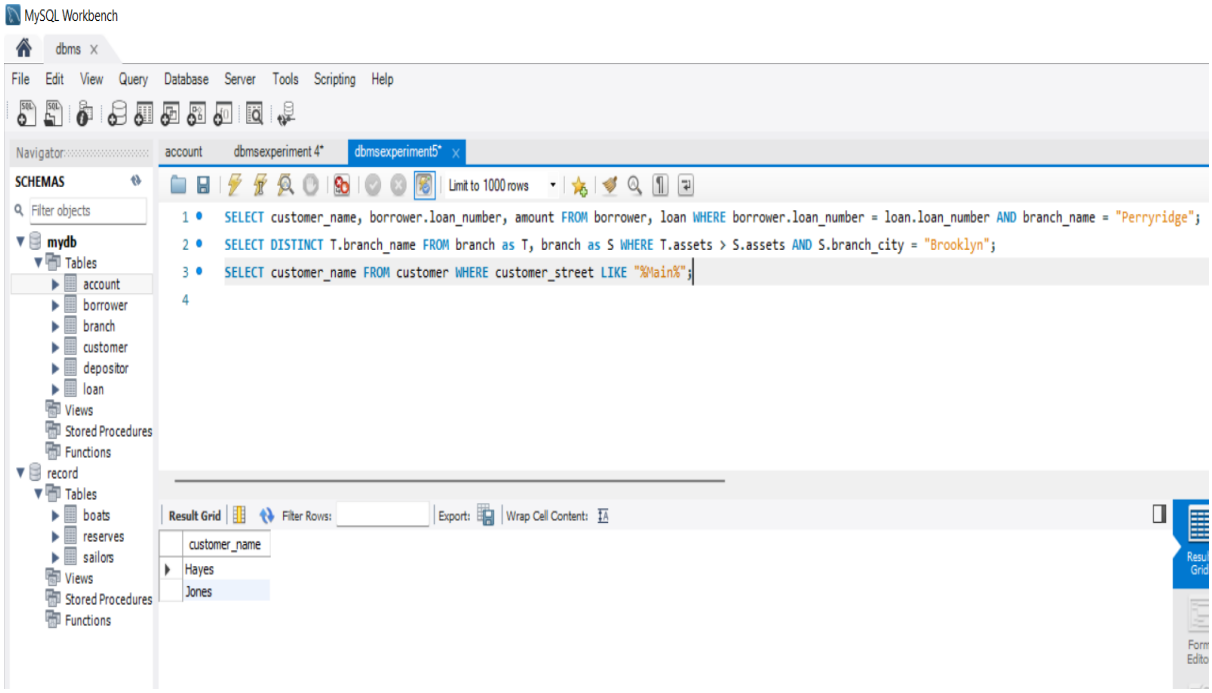
2. Find the names of all branches that have assets greater than atleast one branch located in Brooklyn.

SELECT DISTINCT T.branch_name FROM branch as T, branch as S WHERE T.assets > S.assets AND S.branch_city = "Brooklyn";



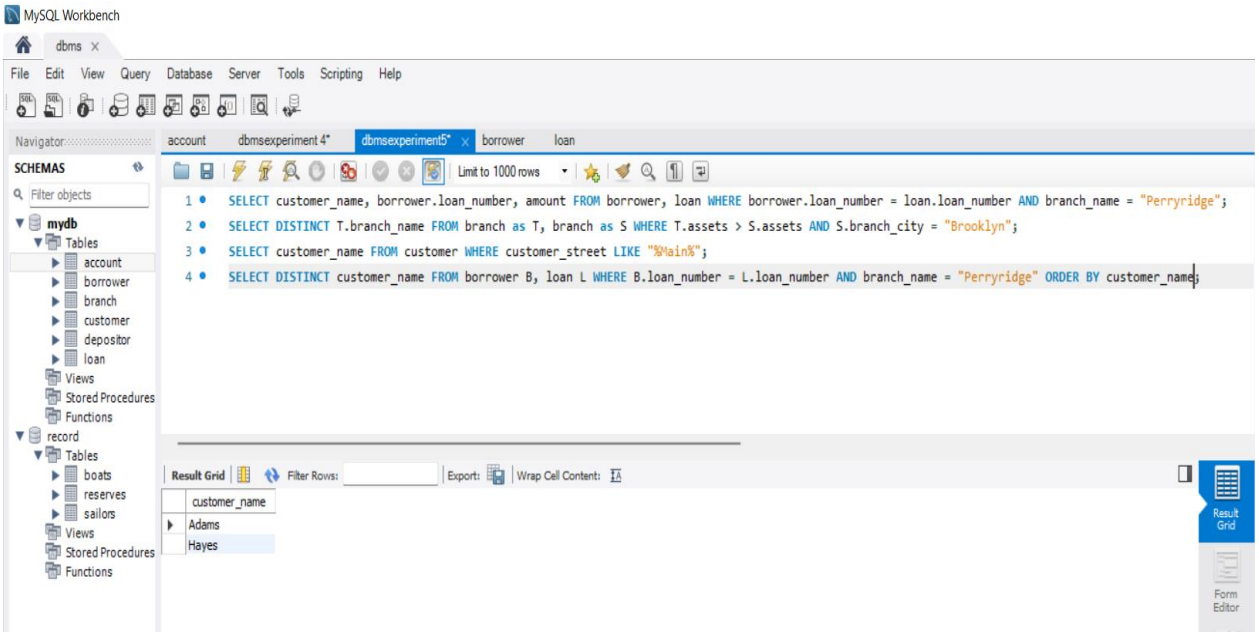
3. Find the names of all customers whose street address includes the substring ‘Main’.

SELECT customer_name FROM customer WHERE customer_street LIKE "%Main%";



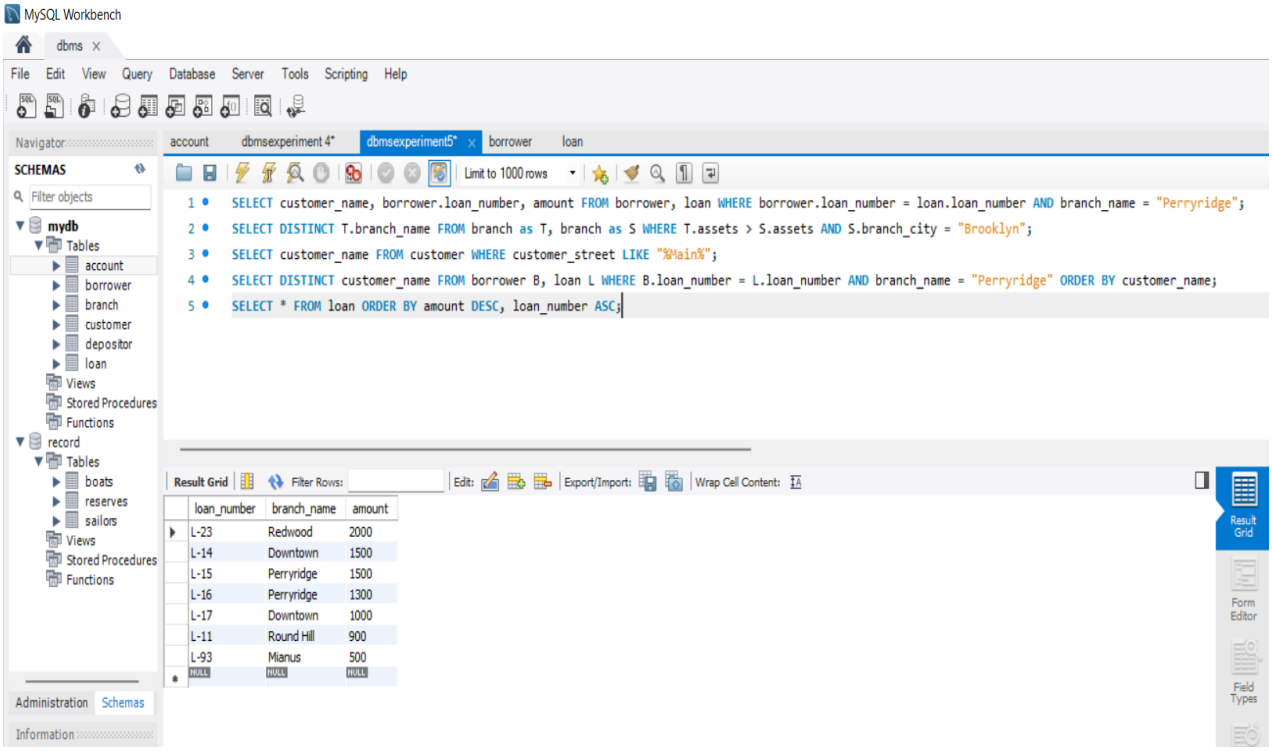
4. To list in alphabetic order all customers who have a loan at the Perryridge branch.

SELECT DISTINCT customer_name FROM borrower B, loan L WHERE B.loan_number = L.loan_number AND branch_name = "Perryridge" ORDER BY customer_name;



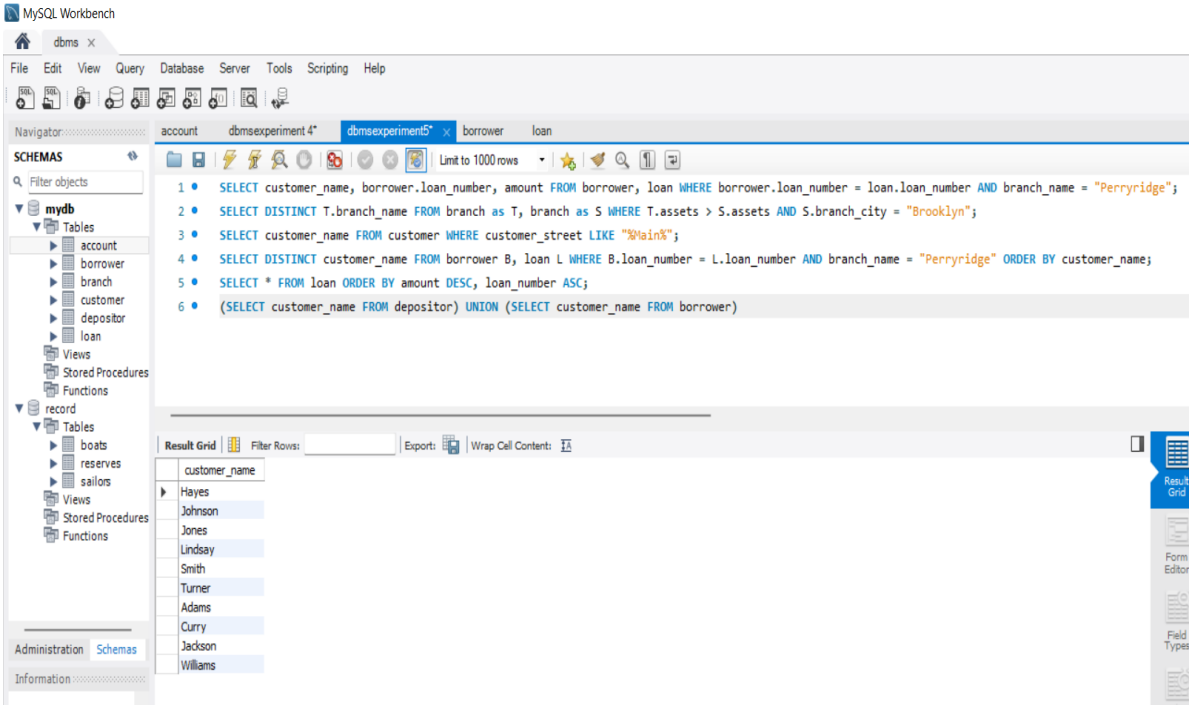
5. To list the entire loan info in descending order of amount.

SELECT * FROM loan ORDER BY amount DESC, loan_number ASC;



6. To find all customers having a loan, an account or both at the bank, without duplicates.

(SELECT customer_name FROM depositor) UNION (SELECT customer_name FROM borrower)

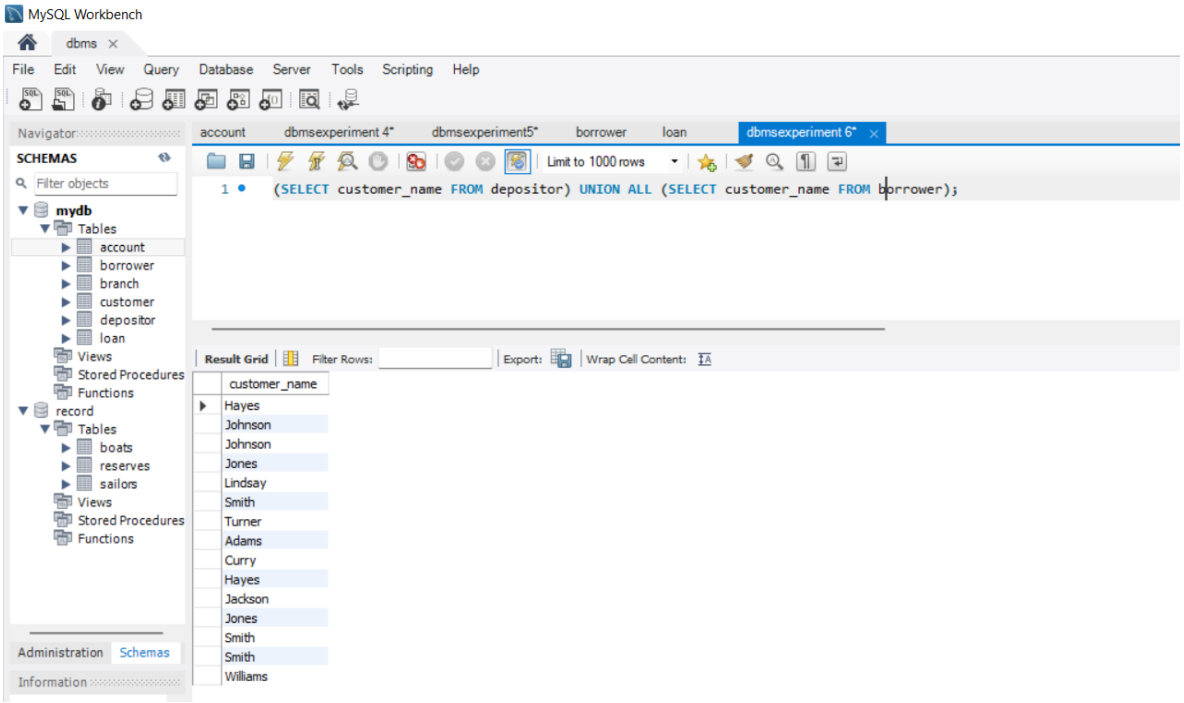


Lab Experiment 6:

You need to find solution of below questions based on tables using Experiment Number 03.

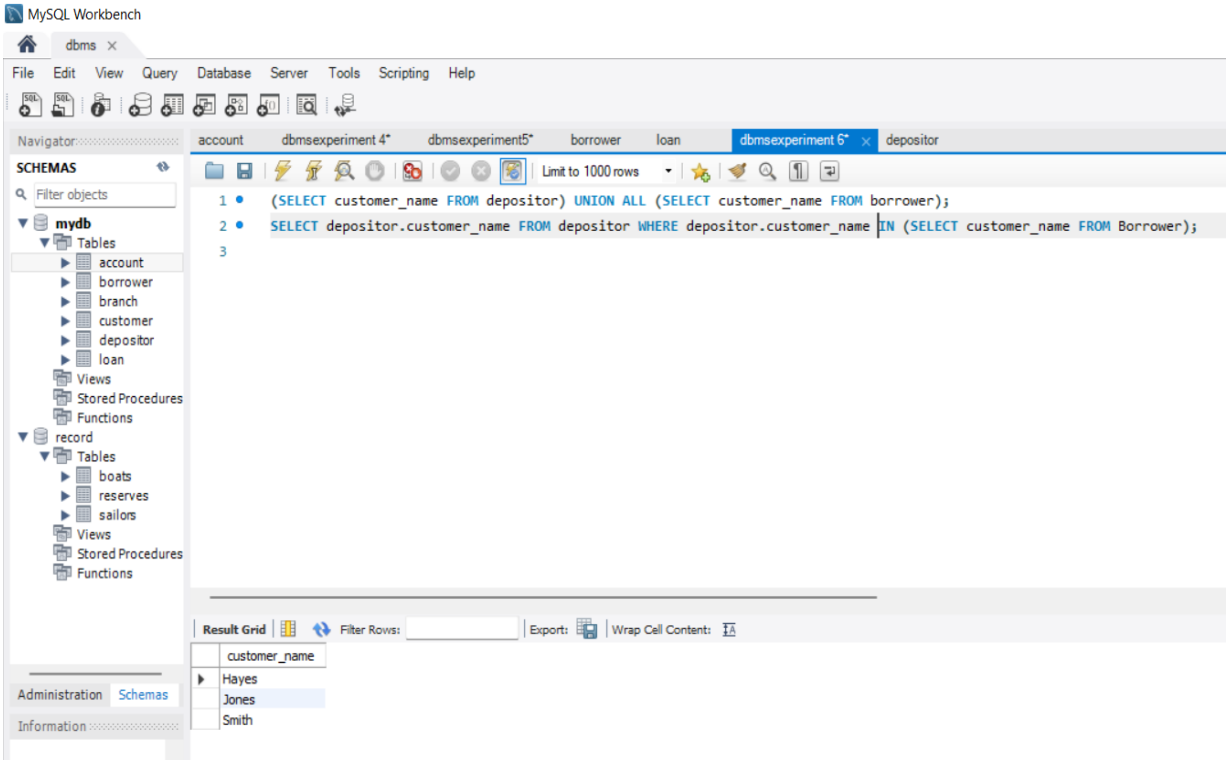
1. To find all customers having a loan, an account or both at the bank, with duplicates.

(SELECT customer_name FROM depositor) UNION ALL (SELECT customer_name FROM borrower);



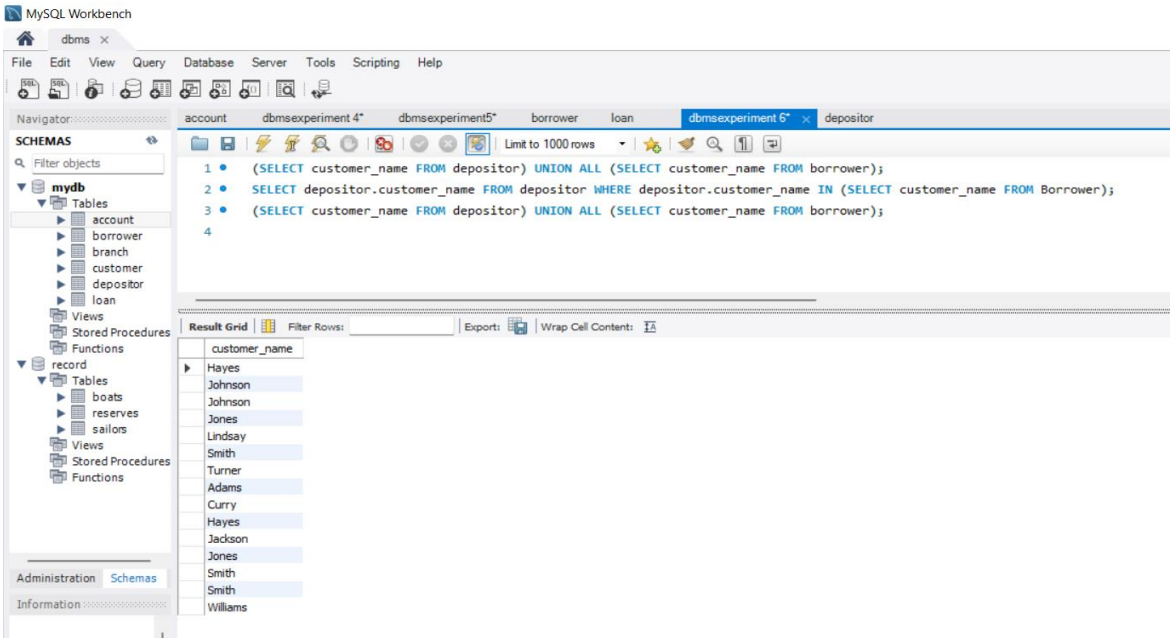
2. To find all customers having both a loan and an account at the bank, without duplicates.

SELECT depositor.customer_name FROM depositor WHERE depositor.customer_name IN (SELECT customer_name FROM Borrower);



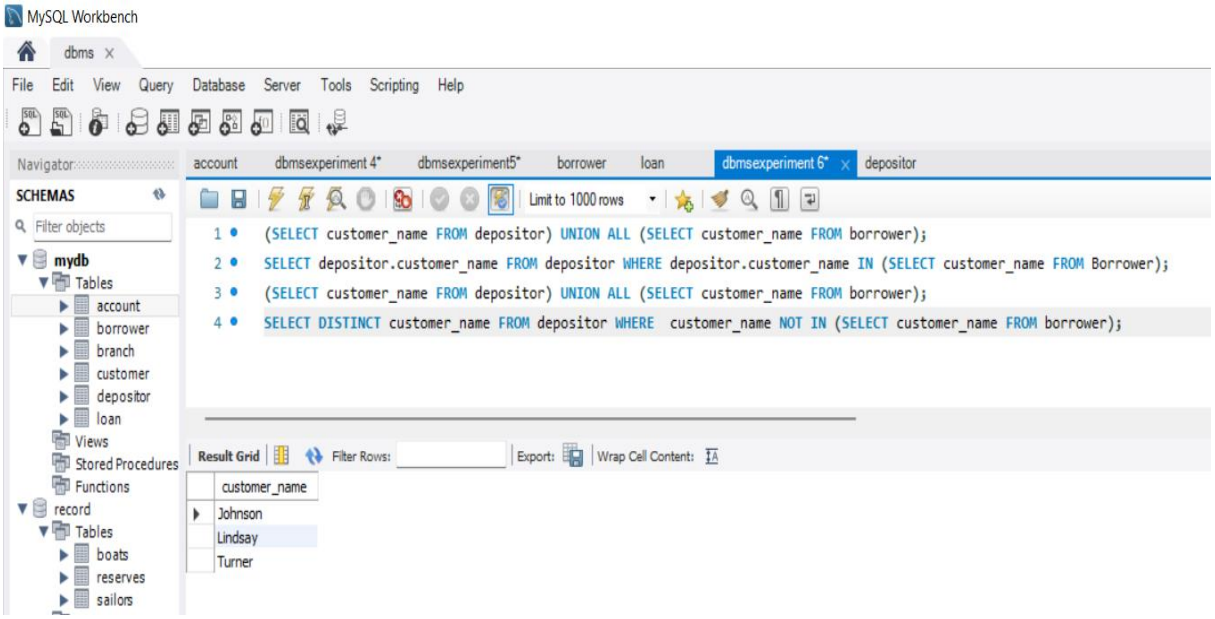
3. To find all customers having a loan, an account or both at the bank, with duplicates.

(SELECT customer_name FROM depositor) UNION ALL (SELECT customer_name FROM borrower);



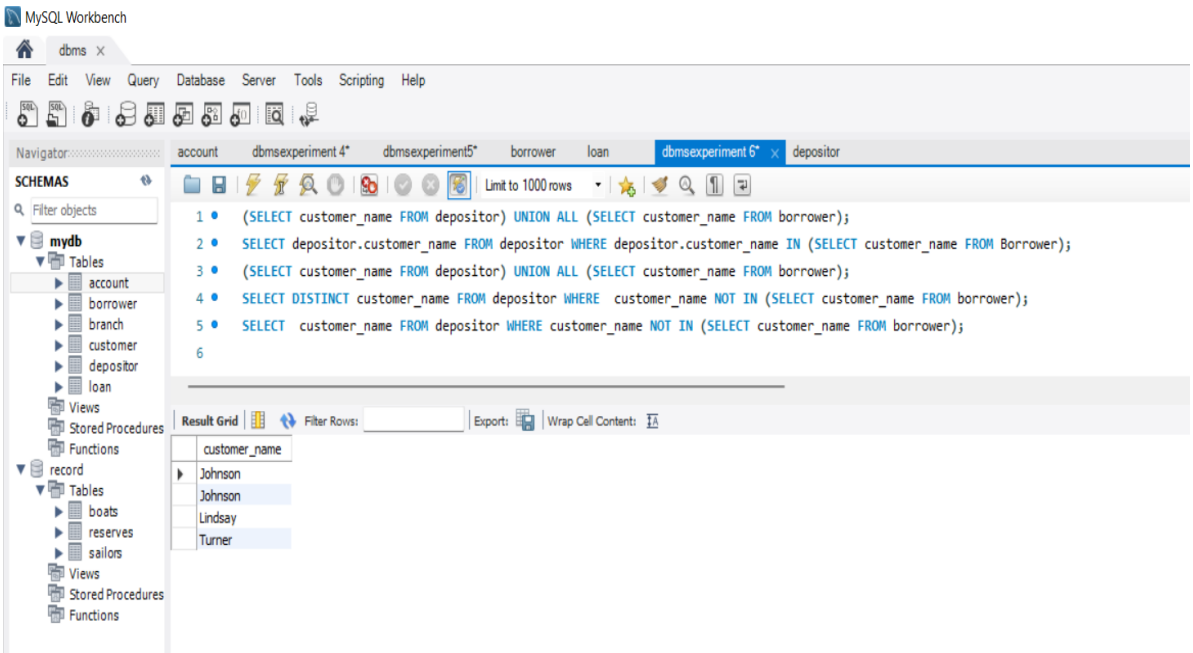
4. To find all customers who have an account but no loan at the bank, without duplicates.

SELECT DISTINCT customer_name FROM depositor WHERE customer_name NOT IN (SELECT customer_name FROM borrower);



5. To find all customers who have an account but no loan at the bank, with duplicates.

SELECT customer_name FROM depositor WHERE customer_name NOT IN (SELECT customer_name FROM borrower);



6. Find the average account balance at the Perryridge branch.

SELECT branch_name, AVG(balance)FROM Account WHERE branch_name = "Perryridge";