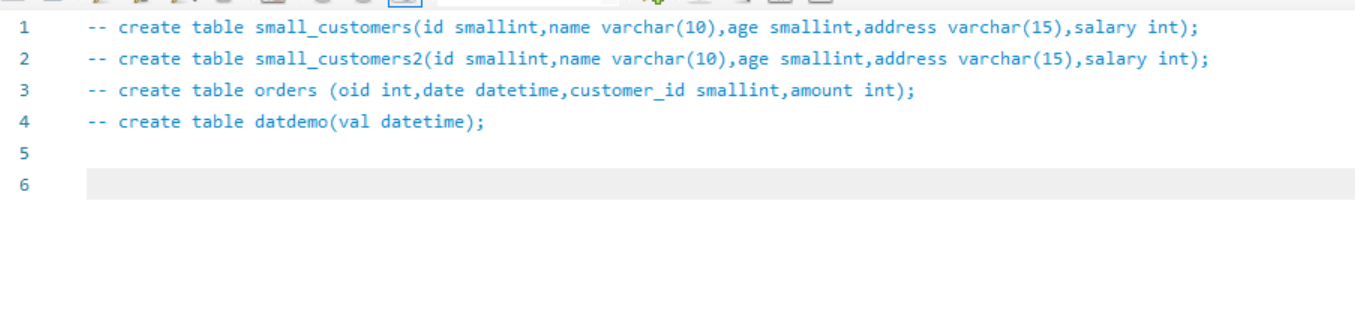
S20230010225 DBMS

LAB 5 SCREENSHOTS

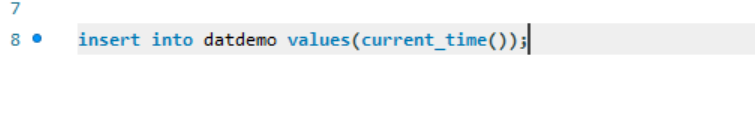
Create tables for practice problems



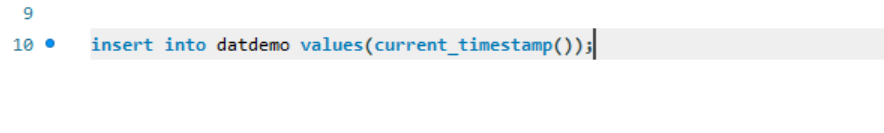
PG1 Q1



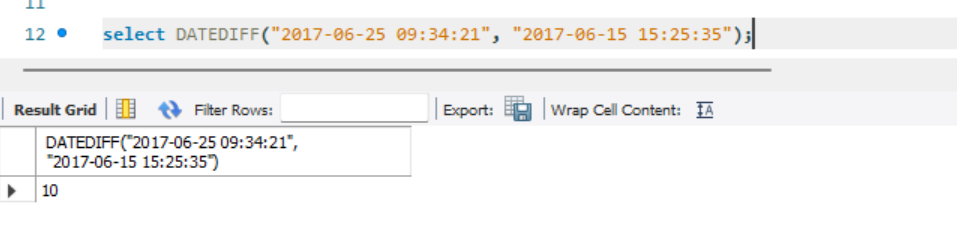
PG1 Q2



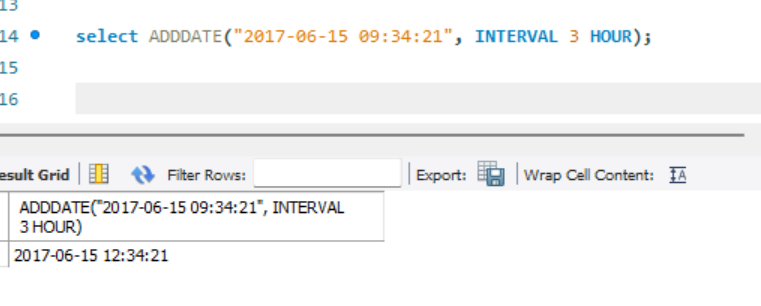
PG1 Q3

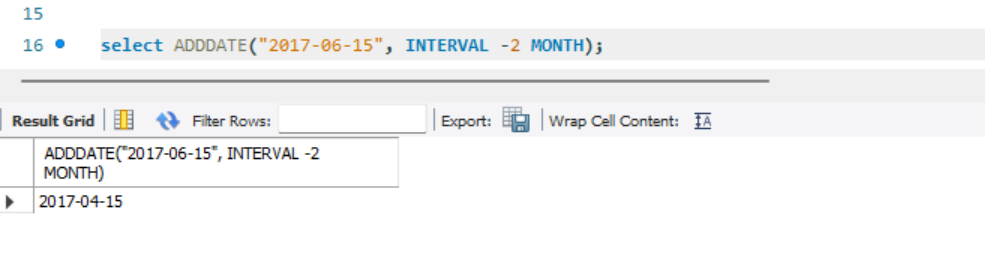


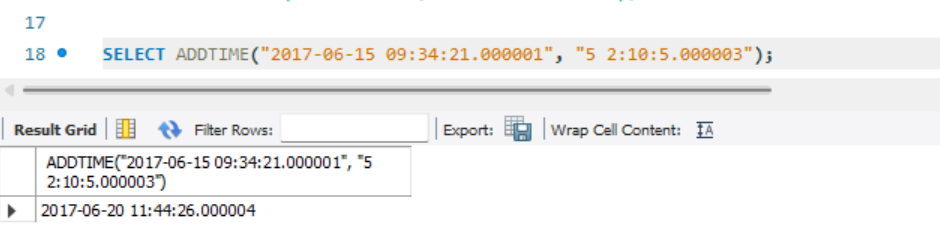
PG1 Q4



PG1 Q5

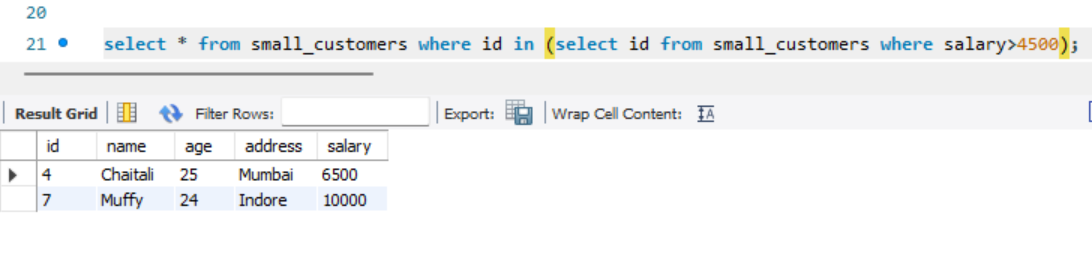


PG 1 Q6

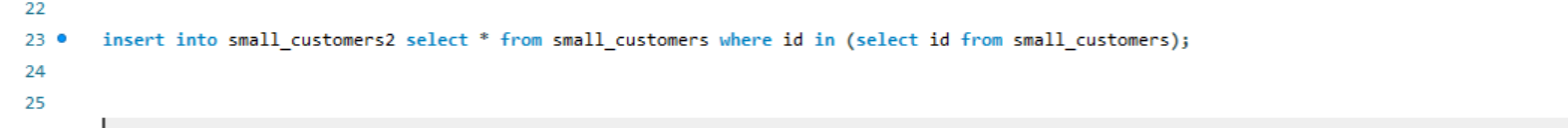


All questions on page 1 are finished

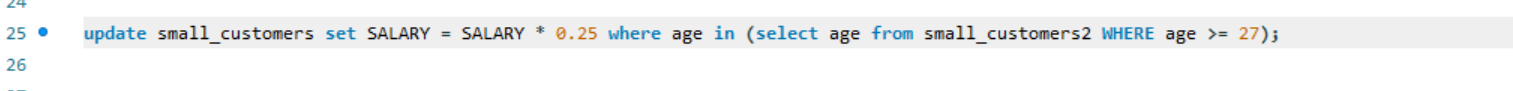
PG3 Q1



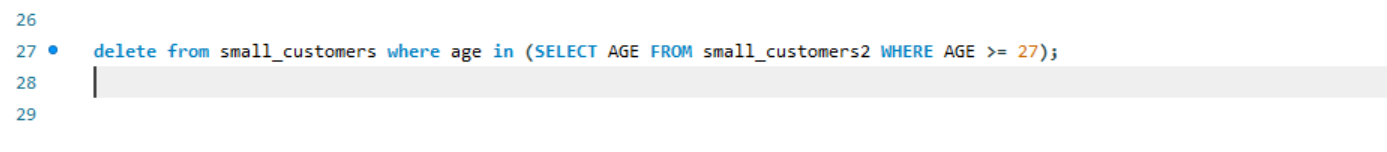
Pg3 q2



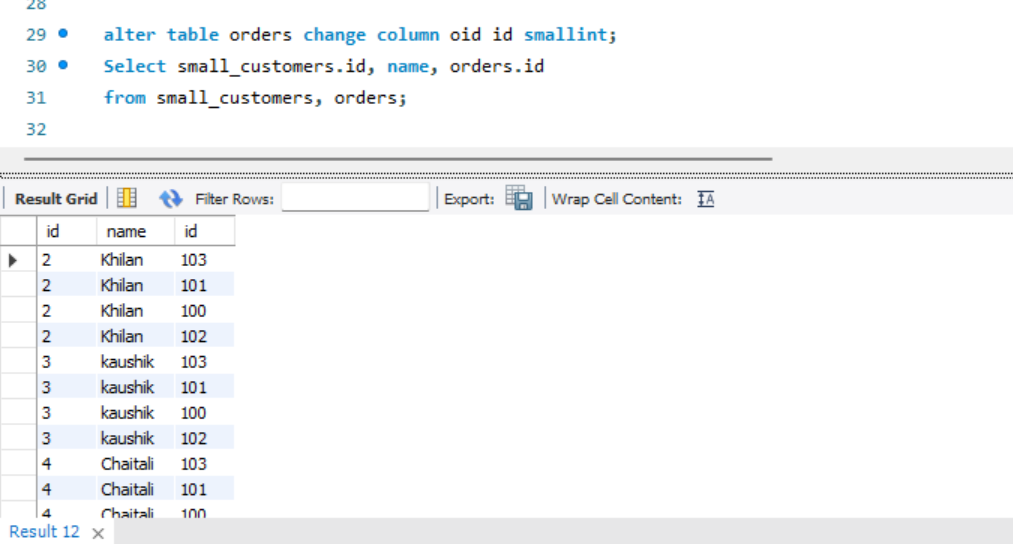
PG3 q3



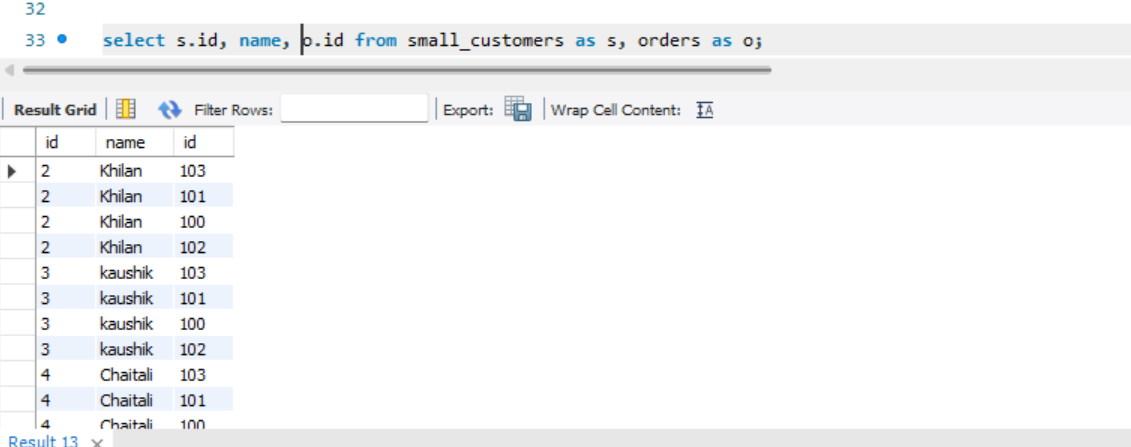
PG3 Q4



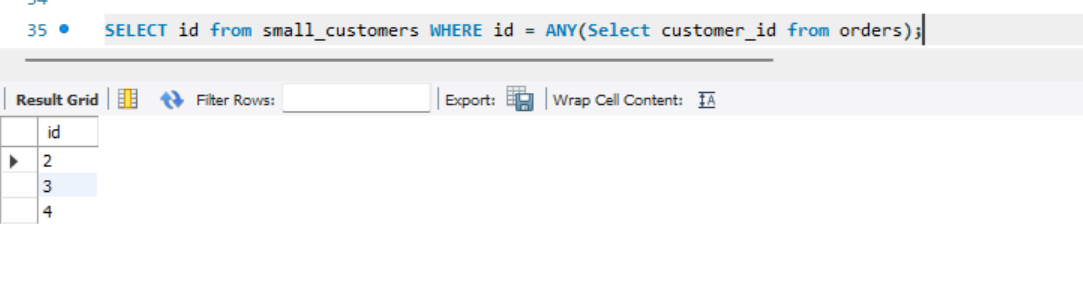
PG4 Q1



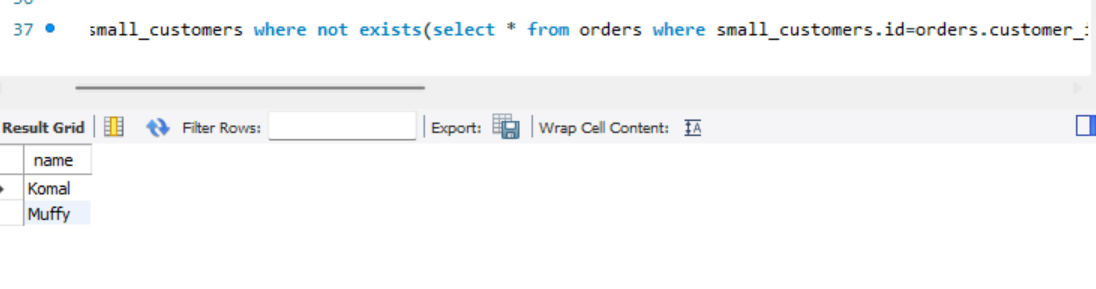
PG4 Q2



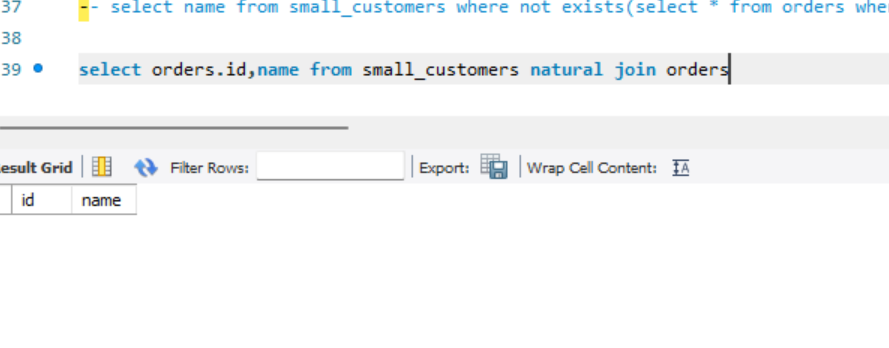
PG4 Q3

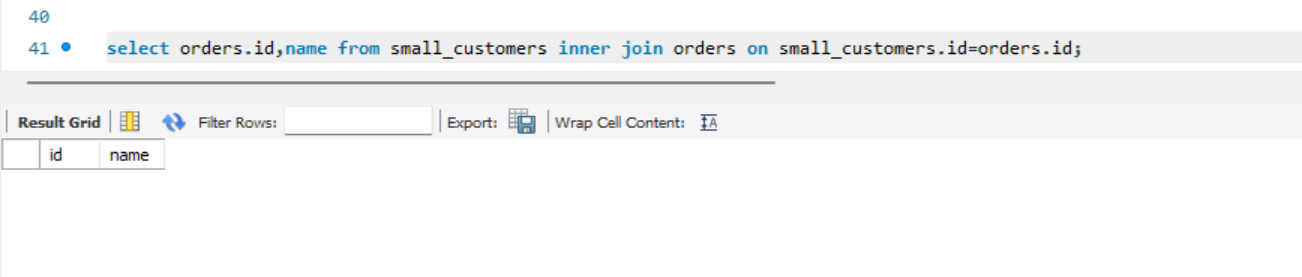


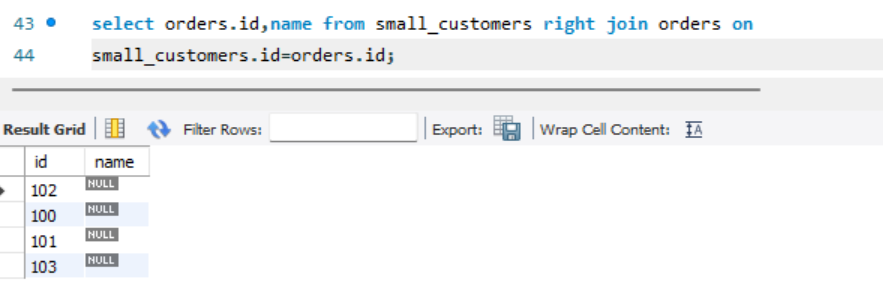
Test for empty relations:

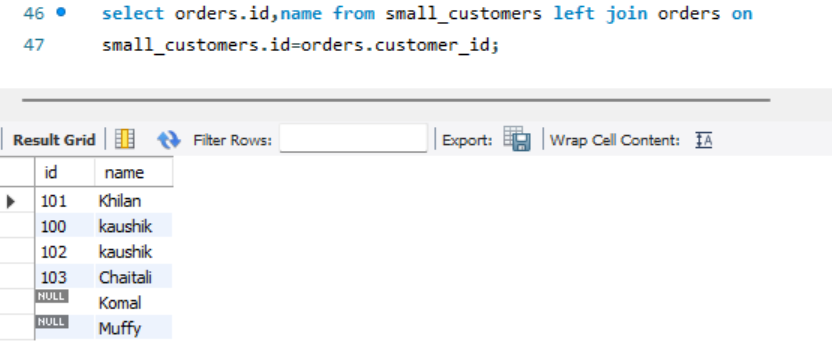


Joins:



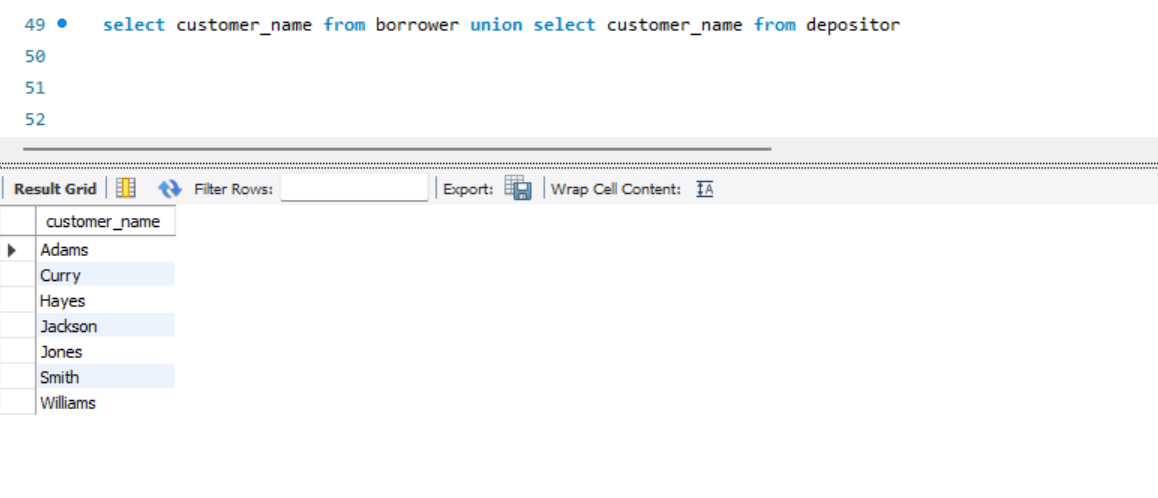




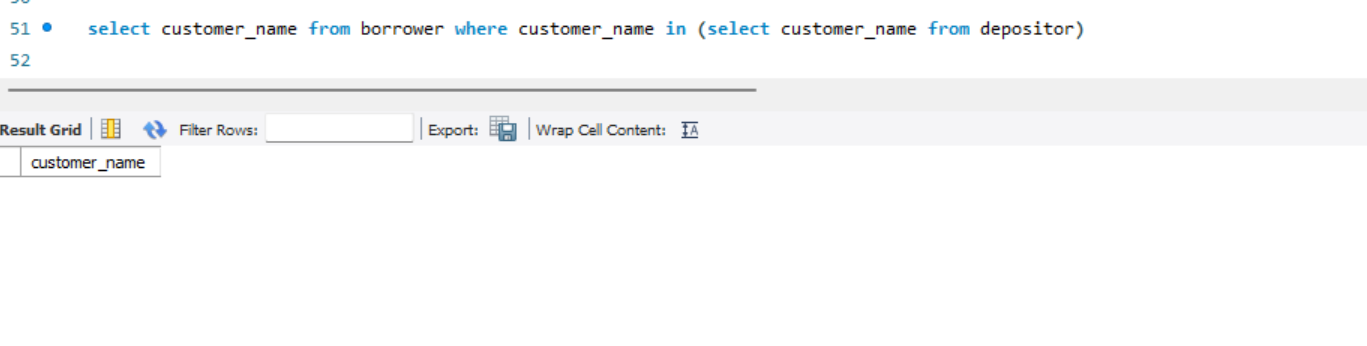


# Questions:

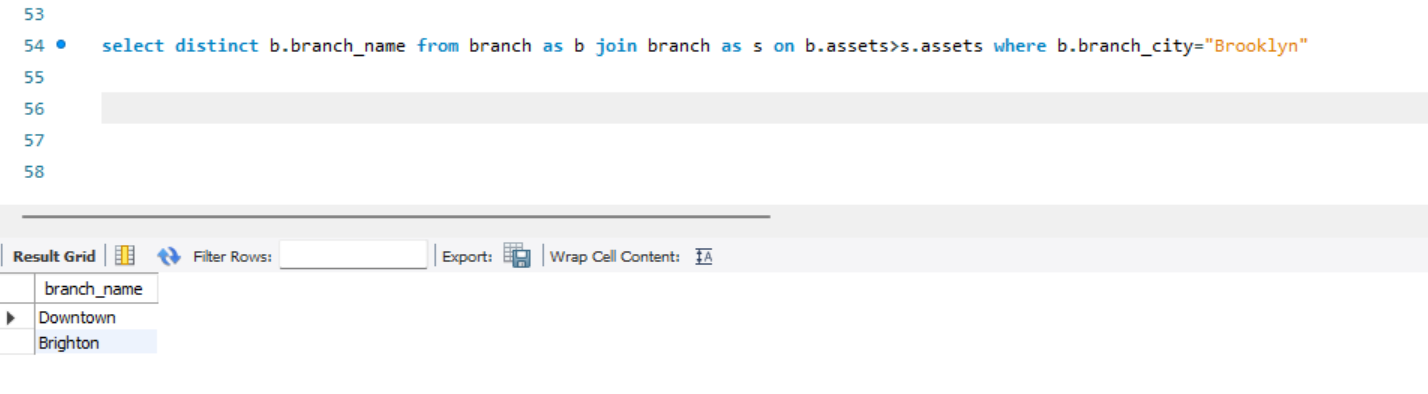
Q1



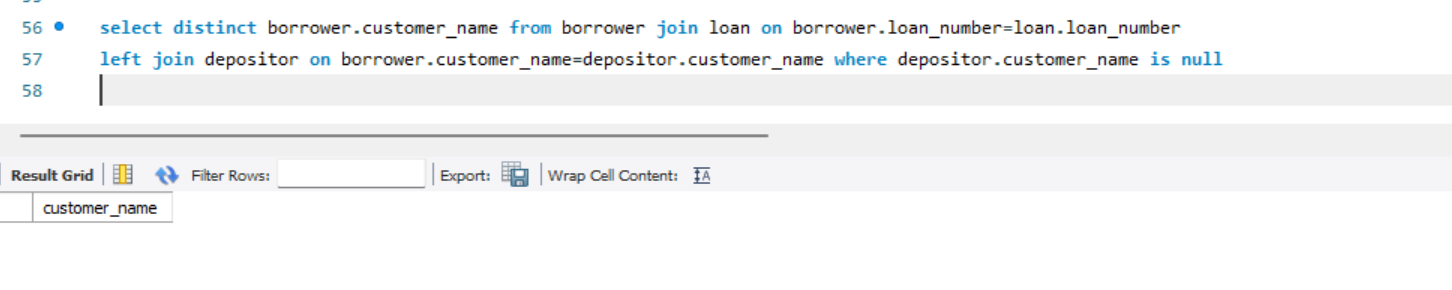
Q2



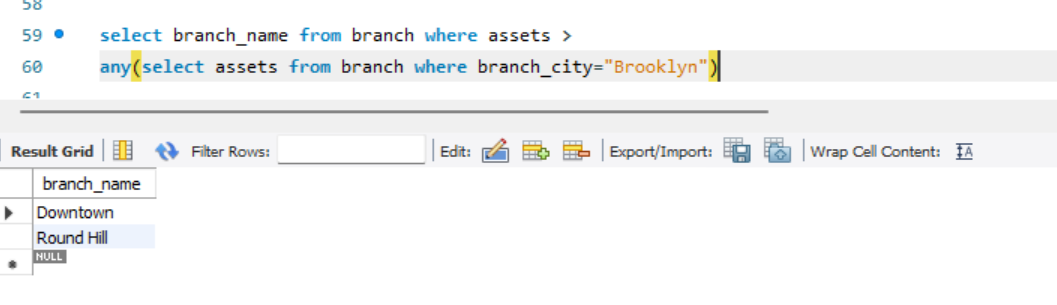
Q3



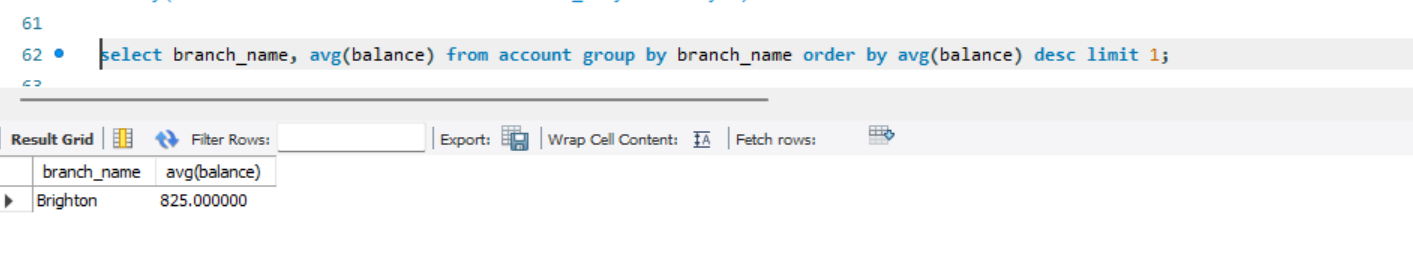
Q4



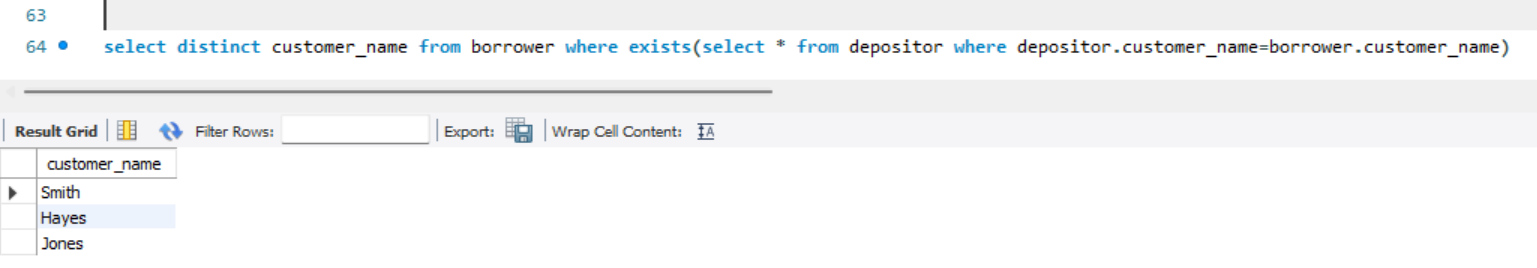
Q5



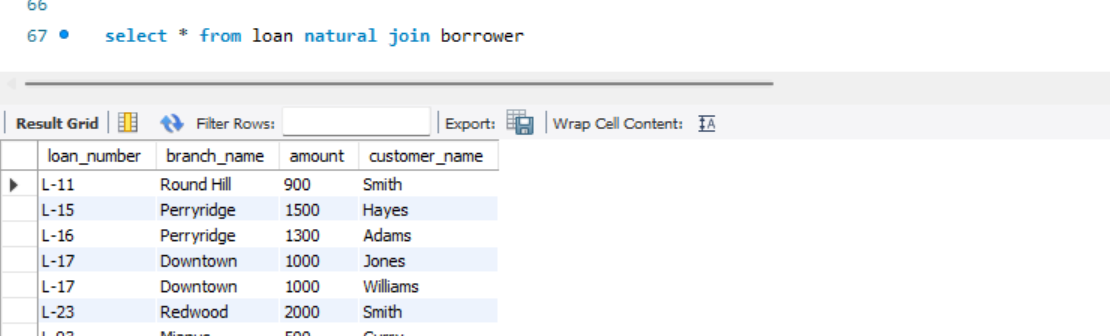
Q6



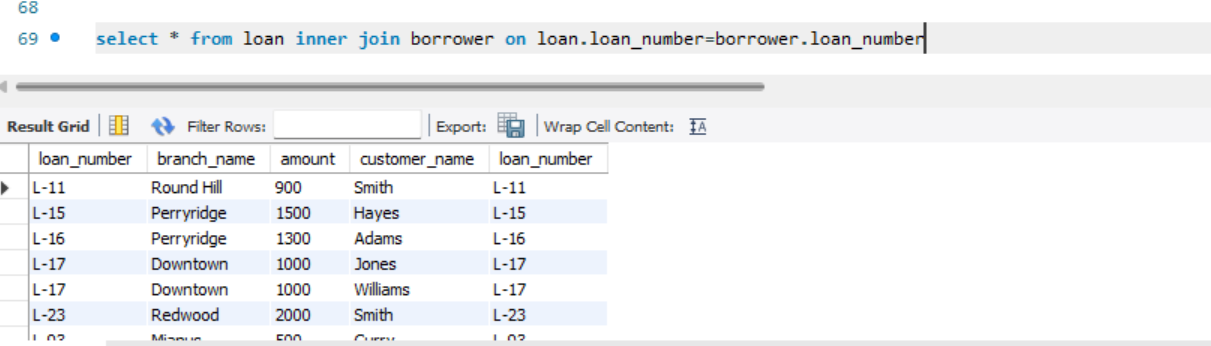
Q7



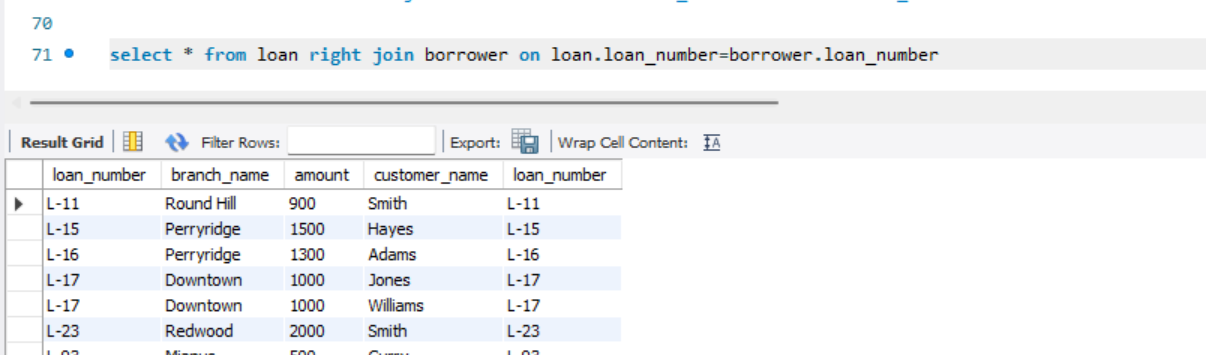
Q8



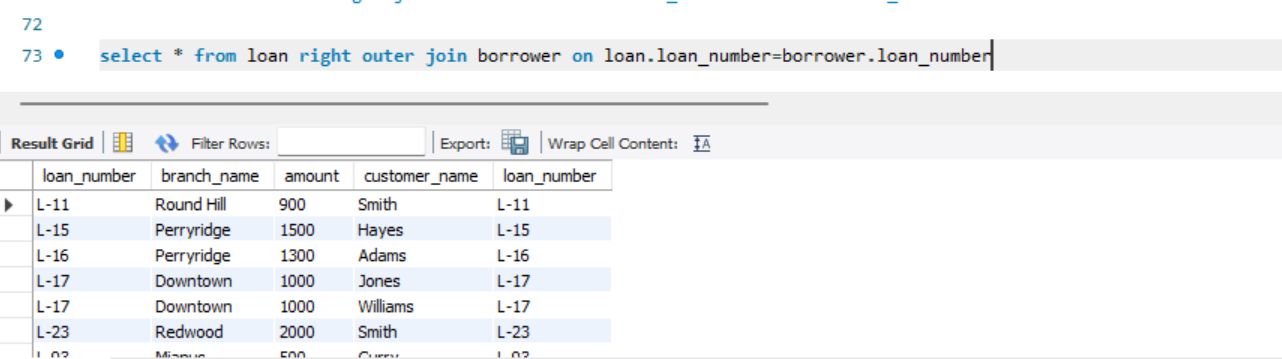
Q9



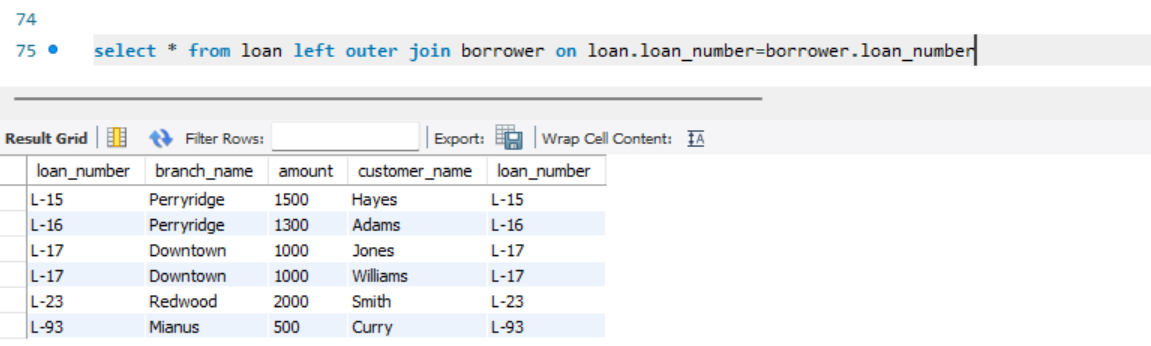
Q10



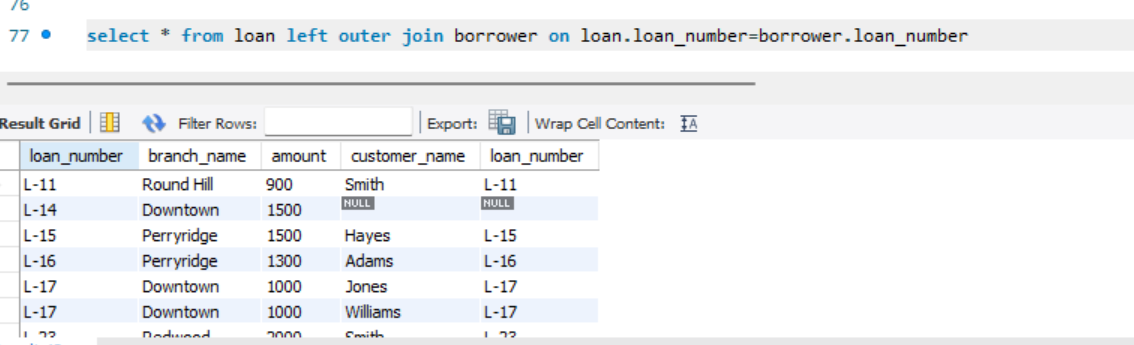
Q11



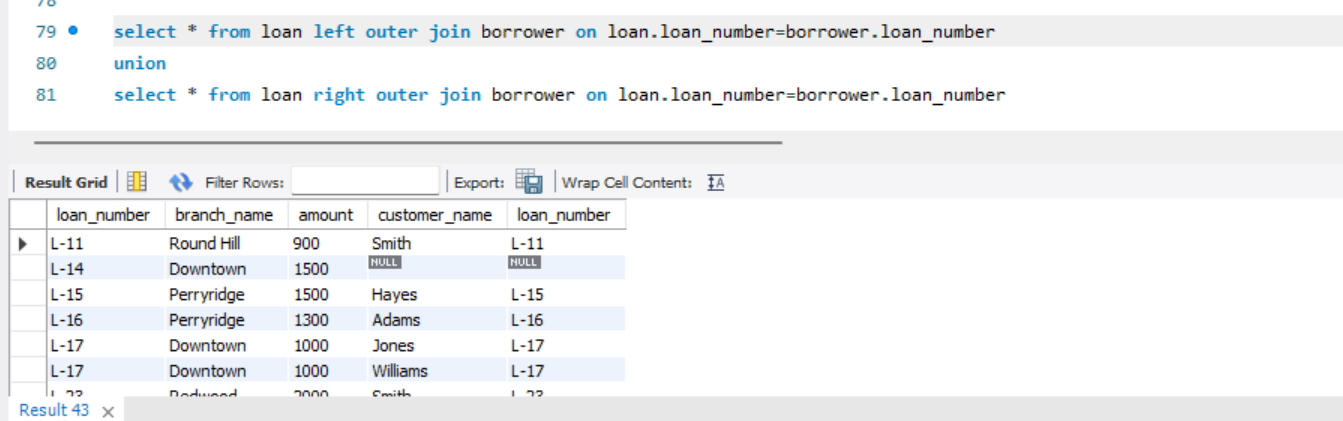
Q12



Q13



Q14



SQL QUERY: Please uncomment each query before running:

-- create table small\_customers(id smallint,name varchar(10),age smallint,address varchar(15),salary int);

-- create table small\_customers2(id smallint,name varchar(10),age smallint,address varchar(15),salary int);

-- create table orders (oid int,date datetime,customer\_id smallint,amount int);

-- create table datdemo(val datetime);

-- insert into datdemo values(current\_date());

-- insert into datdemo values(current\_time());

-- insert into datdemo values(current\_timestamp());

-- select DATEDIFF("2017-06-25 09:34:21", "2017-06-15 15:25:35");

-- select ADDDATE("2017-06-15 09:34:21", INTERVAL 3 HOUR);

-- select ADDDATE("2017-06-15", INTERVAL -2 MONTH);

-- SELECT ADDTIME("2017-06-15 09:34:21.000001", "5 2:10:5.000003");

-- select \* from small\_customers where id in (select id from small\_customers where salary>4500);

-- insert into small\_customers2 select \* from small\_customers where id in (select id from small\_customers);

-- update small\_customers set SALARY = SALARY \* 0.25 where age in (select age from small\_customers2 WHERE age >= 27);

-- delete from small\_customers where age in (SELECT AGE FROM small\_customers2 WHERE AGE >= 27);

-- alter table orders change column oid id smallint;

-- Select small\_customers.id, name, orders.id

-- from small\_customers, orders;

-- select s.id, name, o.id from small\_customers as s, orders as o;

-- SELECT id from small\_customers WHERE id = ANY(Select customer\_id from orders);

-- select name from small\_customers where not exists(select \* from orders where small\_customers.id=orders.customer\_id)

-- select orders.id,name from small\_customers natural join orders

-- select orders.id,name from small\_customers inner join orders on small\_customers.id=orders.id;

-- select orders.id,name from small\_customers right join orders on

-- small\_customers.id=orders.id;

-- select orders.id,name from small\_customers left join orders on

-- small\_customers.id=orders.customer\_id;

-- select customer\_name from borrower union select customer\_name from depositor

-- select customer\_name from borrower where customer\_name in (select customer\_name from depositor)

-- select distinct b.branch\_name from branch as b join branch as s on b.assets>s.assets where b.branch\_city="Brooklyn"

-- select distinct borrower.customer\_name from borrower join loan on borrower.loan\_number=loan.loan\_number

-- left join depositor on borrower.customer\_name=depositor.customer\_name where depositor.customer\_name is null

-- select branch\_name from branch where assets >

-- any(select assets from branch where branch\_city="Brooklyn")

-- select branch\_name, avg(balance) from account group by branch\_name order by avg(balance) desc limit 1;

-- select distinct customer\_name from borrower where exists(select \* from depositor where depositor.customer\_name=borrower.customer\_name)

-- select \* from loan natural join borrower

-- select \* from loan inner join borrower on loan.loan\_number=borrower.loan\_number

-- select \* from loan right join borrower on loan.loan\_number=borrower.loan\_number

-- select \* from loan right outer join borrower on loan.loan\_number=borrower.loan\_number

-- select \* from loan left outer join borrower on loan.loan\_number=borrower.loan\_number

-- select \* from loan left outer join borrower on loan.loan\_number=borrower.loan\_number

-- select \* from loan left outer join borrower on loan.loan\_number=borrower.loan\_number

-- union

-- select \* from loan right outer join borrower on loan.loan\_number=borrower.loan\_number