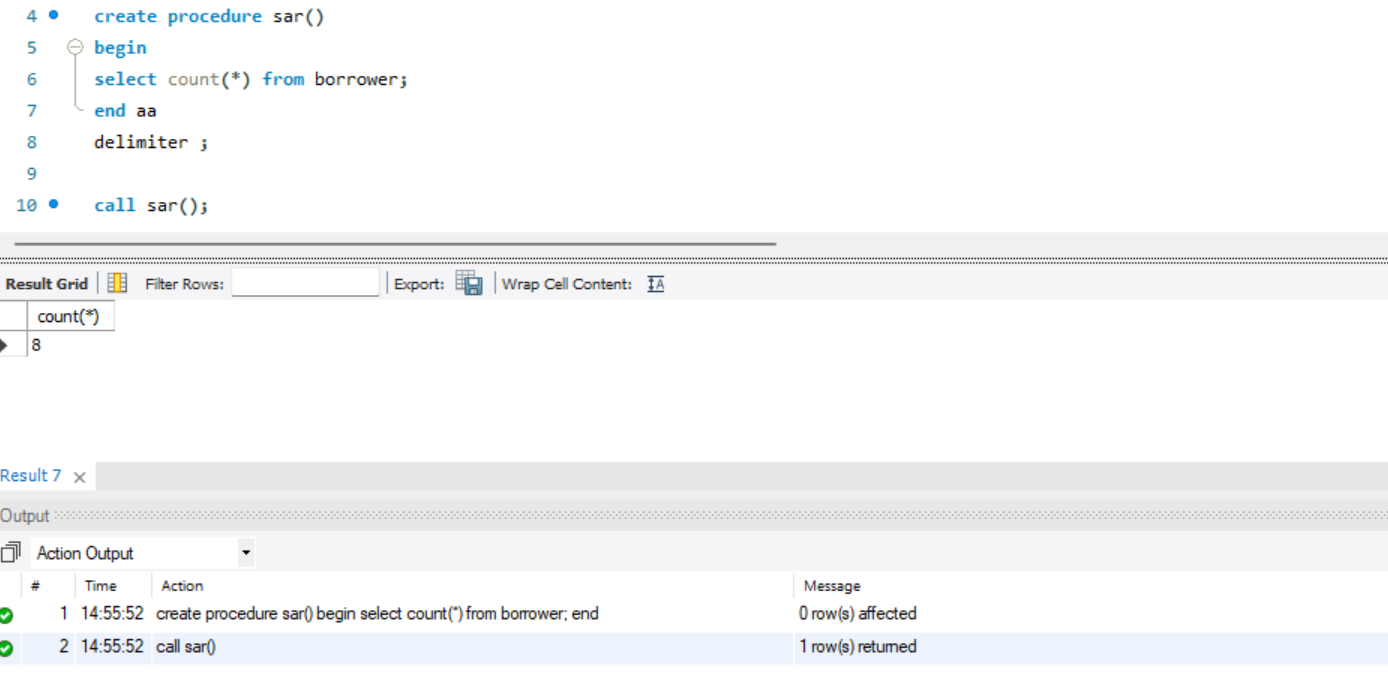
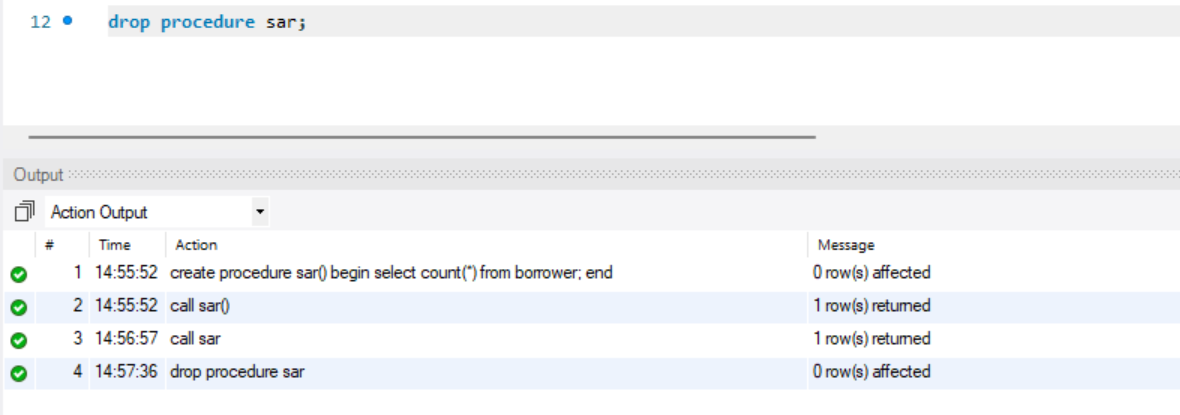
S20230010225\_Lab7

DBMS LAB 7 Queries and solutions

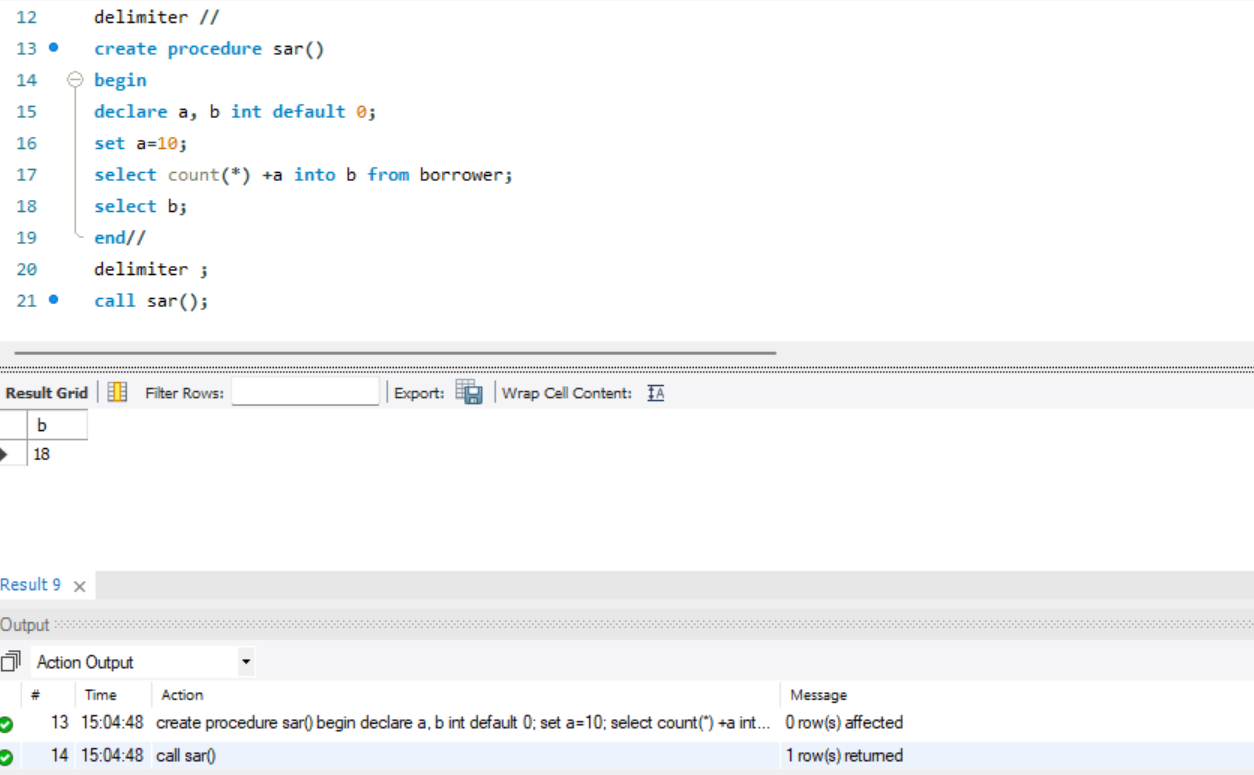
### Example queries:



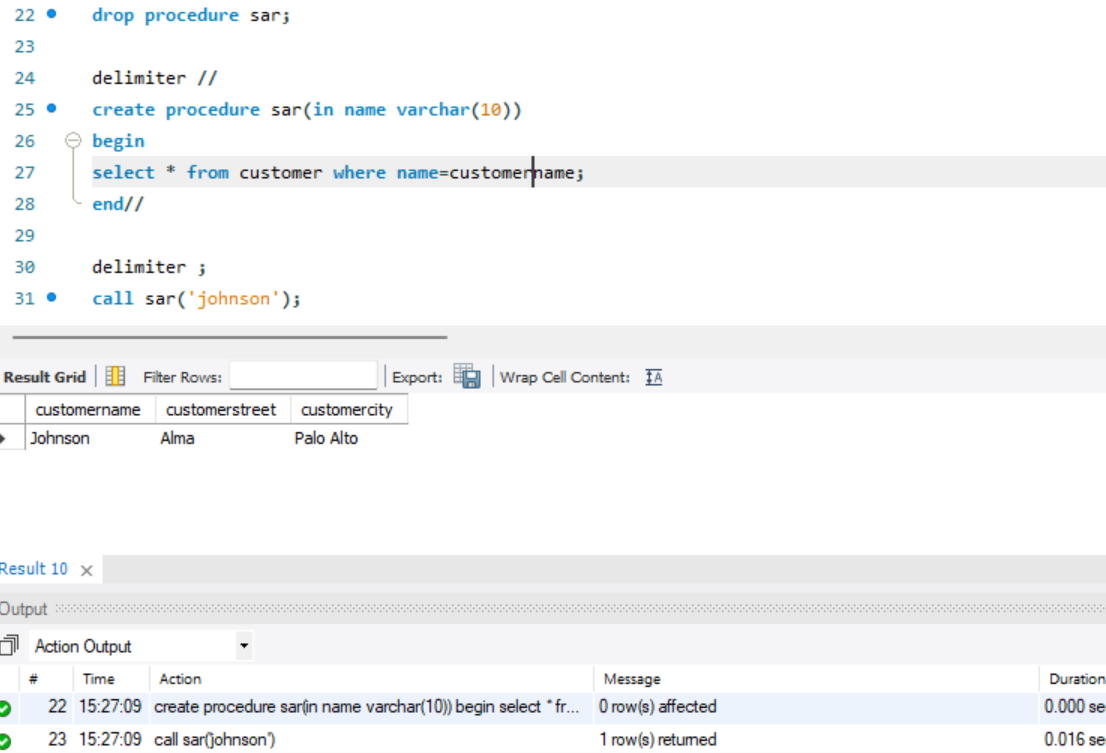
Drop procedure:



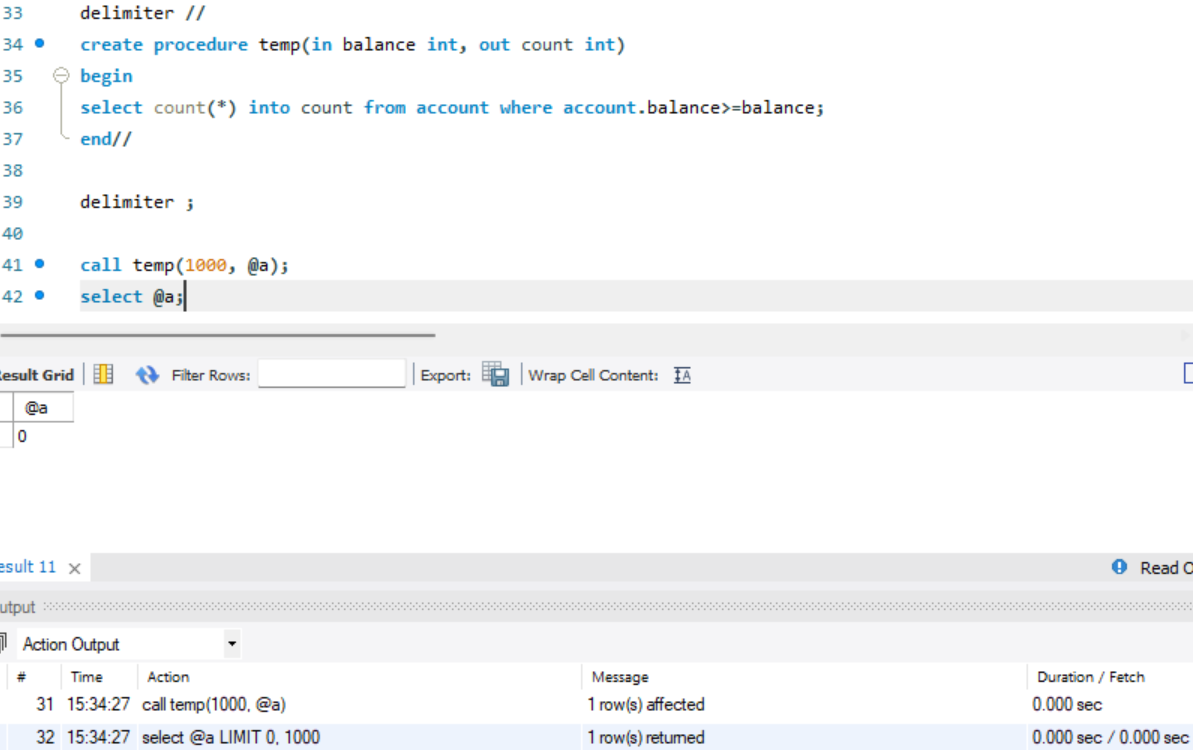
Variables:



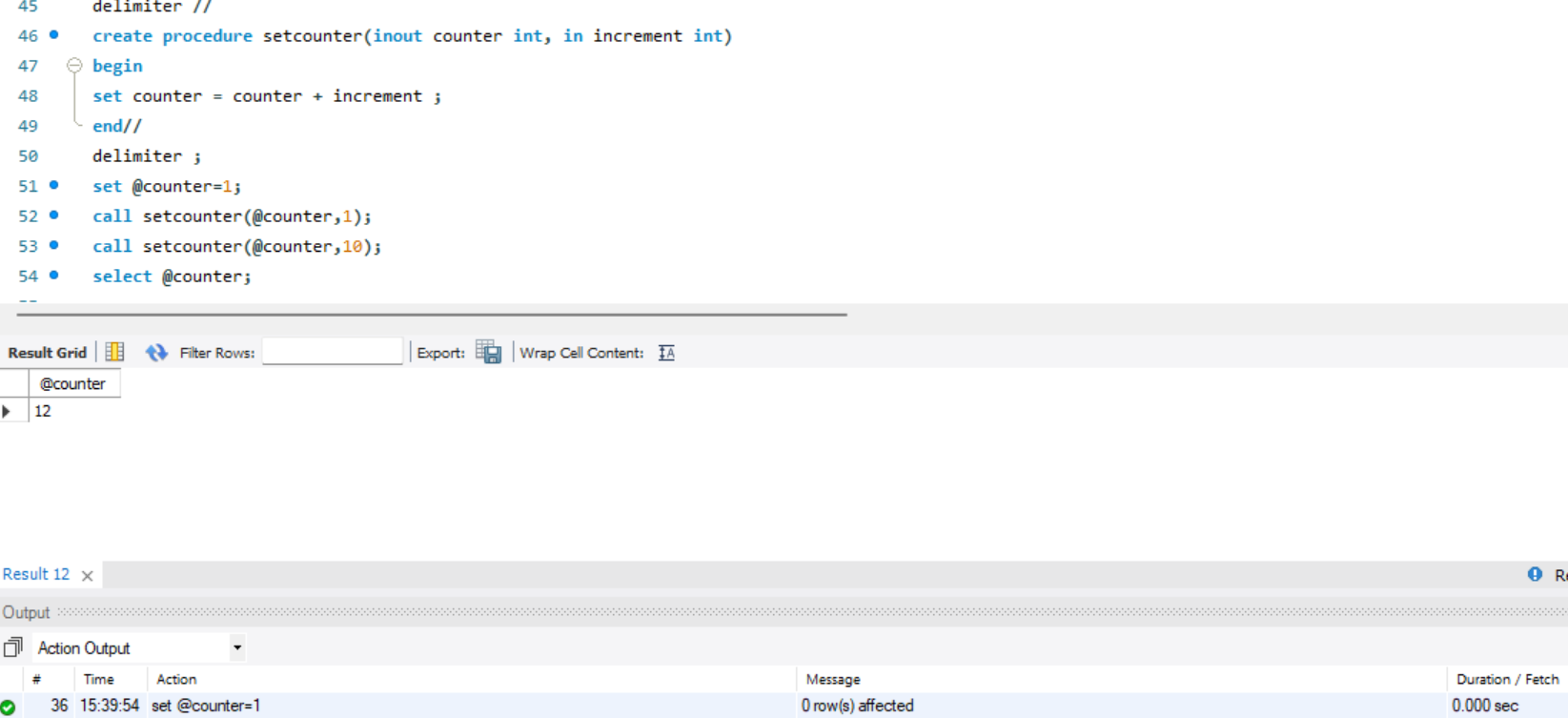
IN Parameter



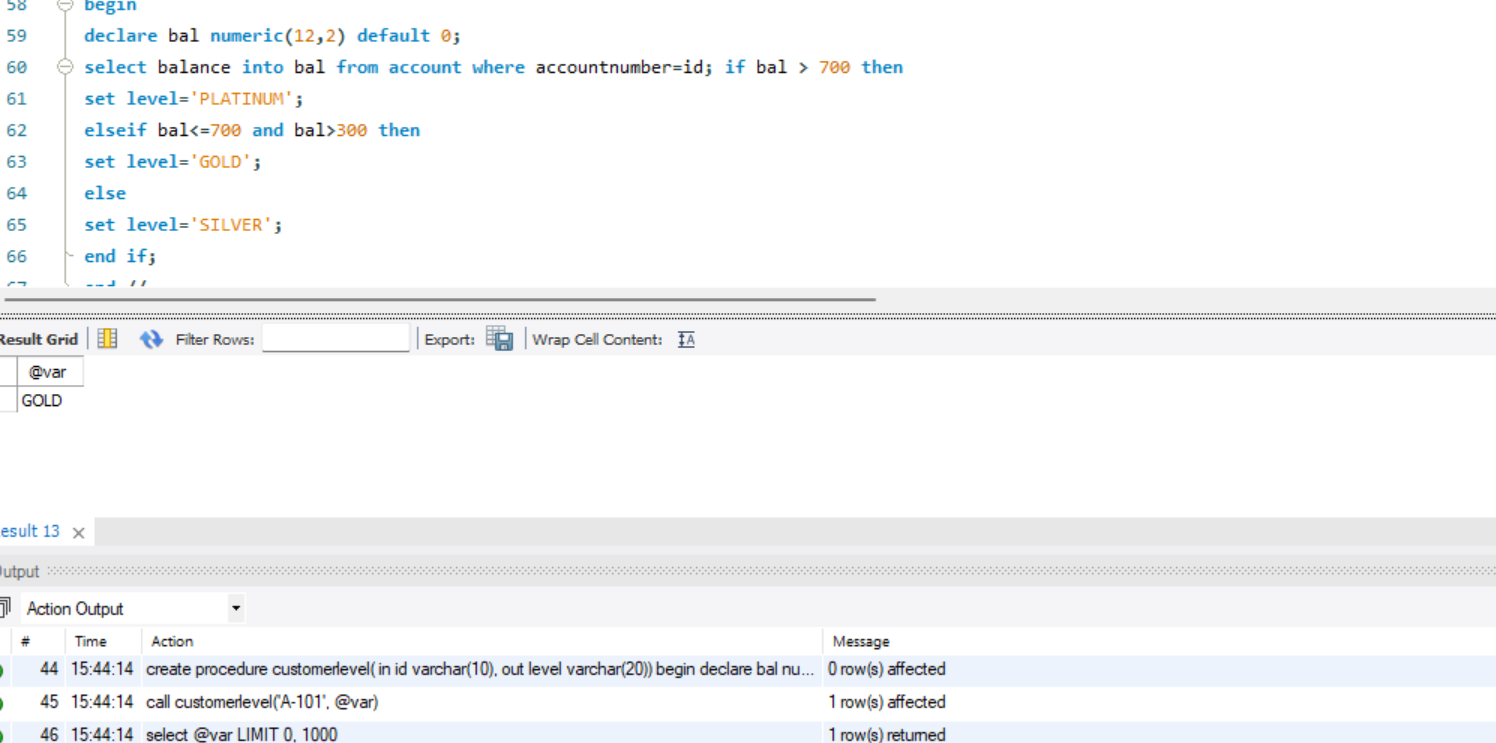
OUT parameter



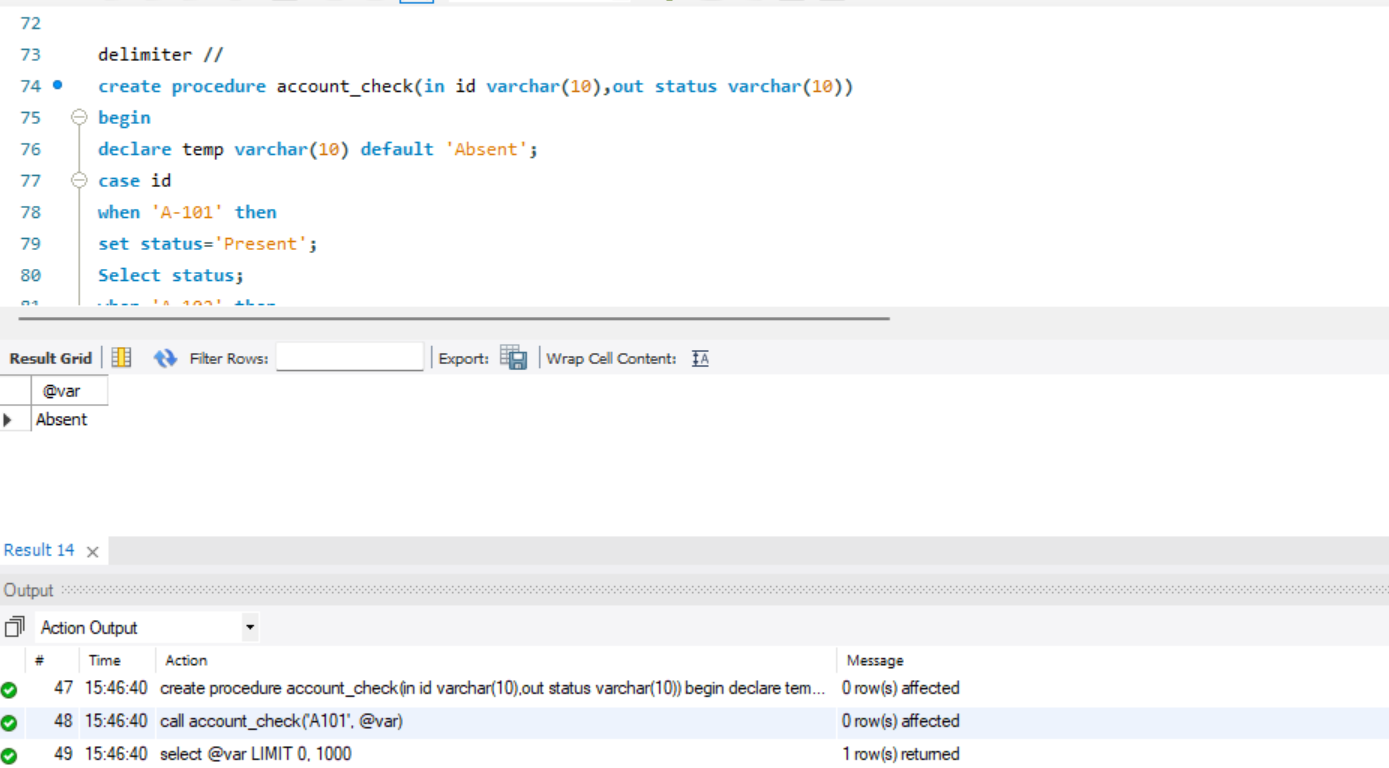
INOUT parameter



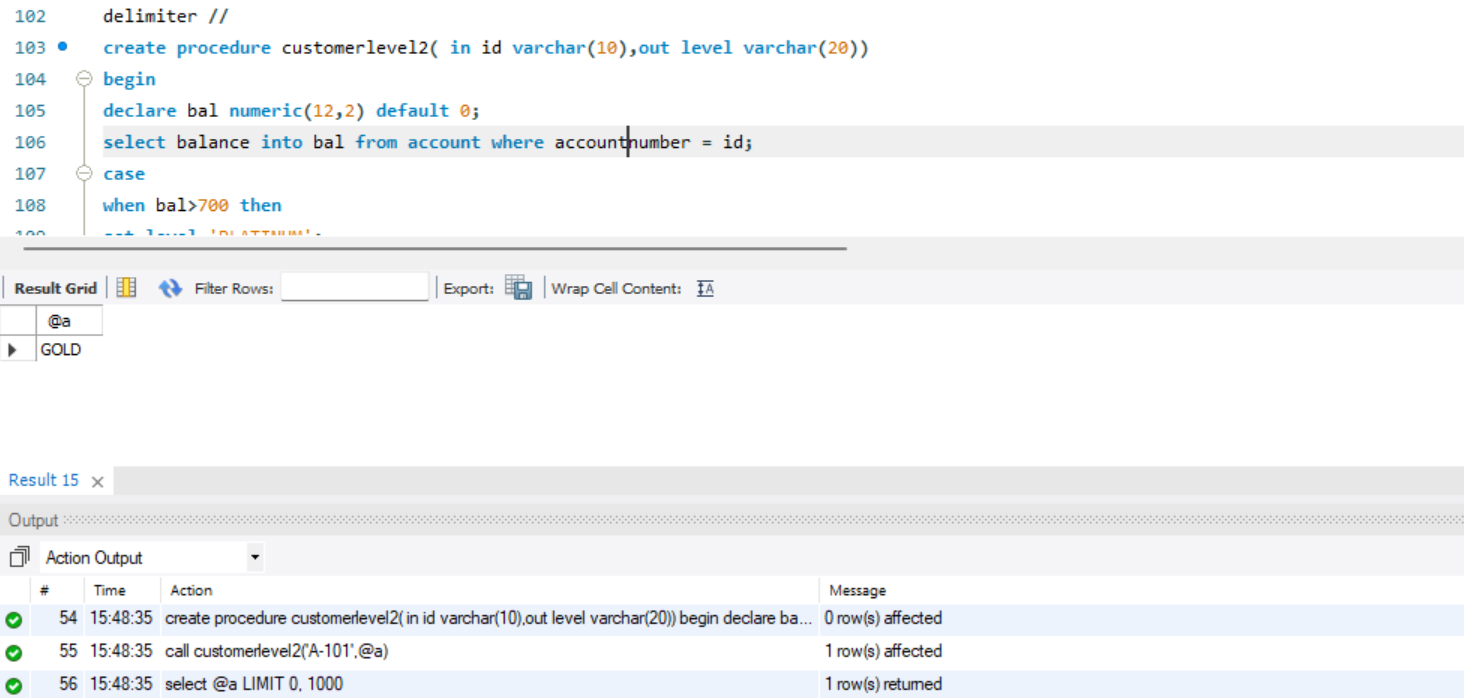
If statements:



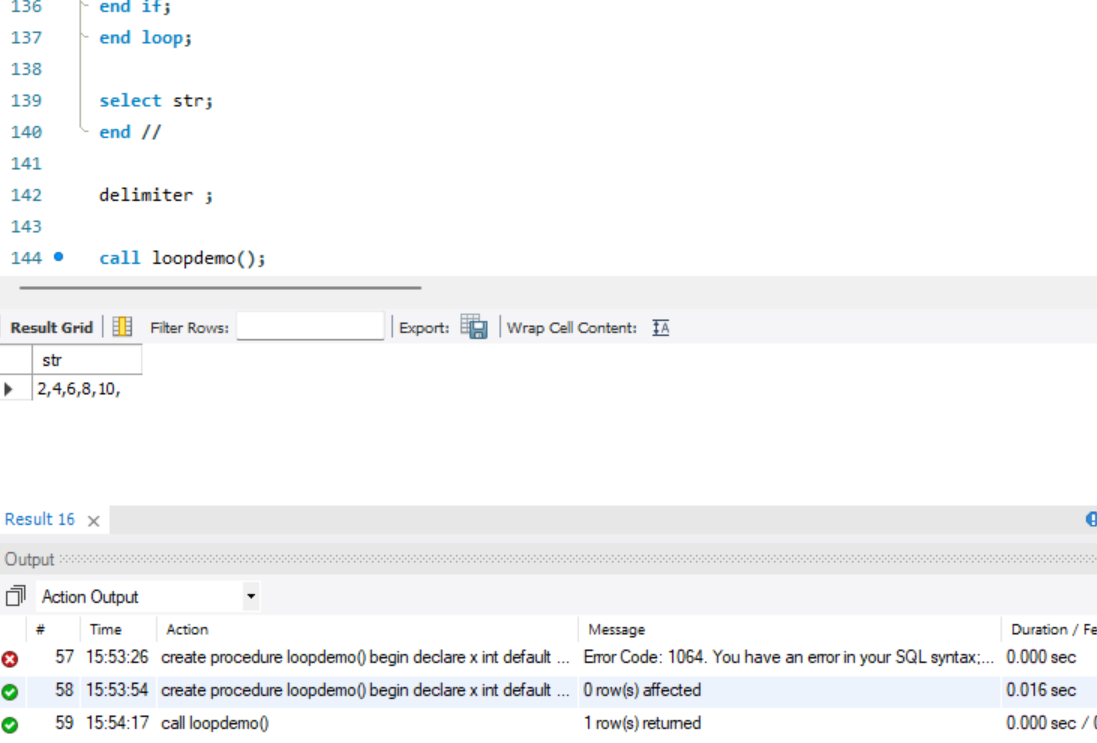
Switch case statements:



IF case statements

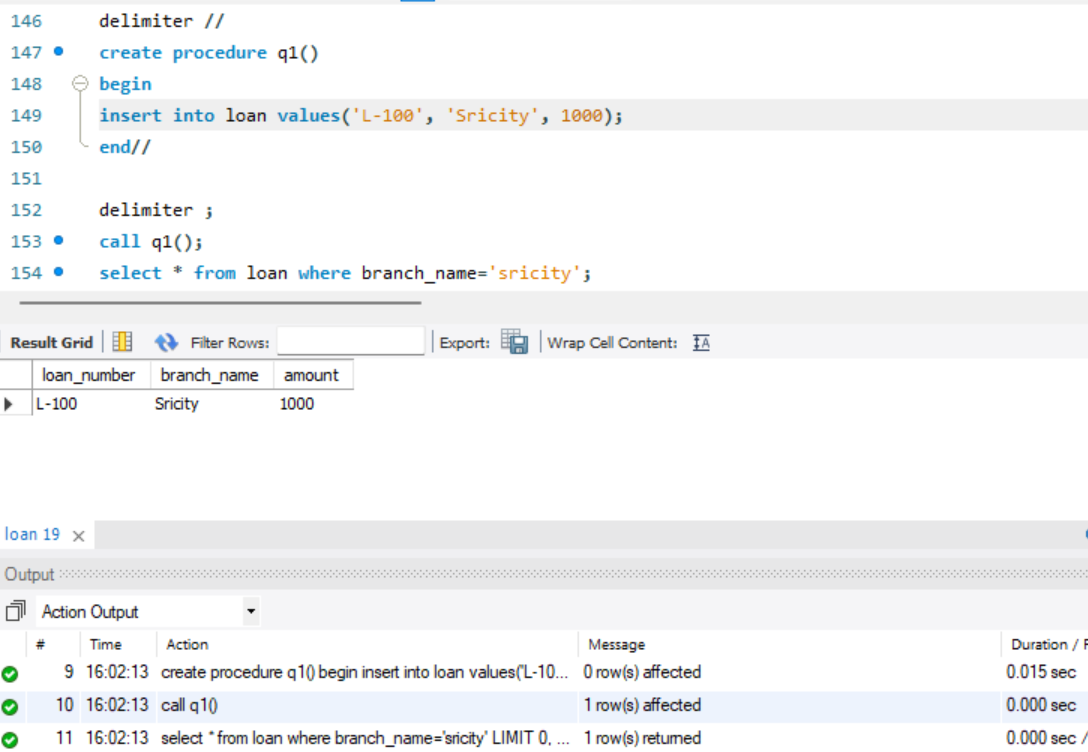


**Loops:**

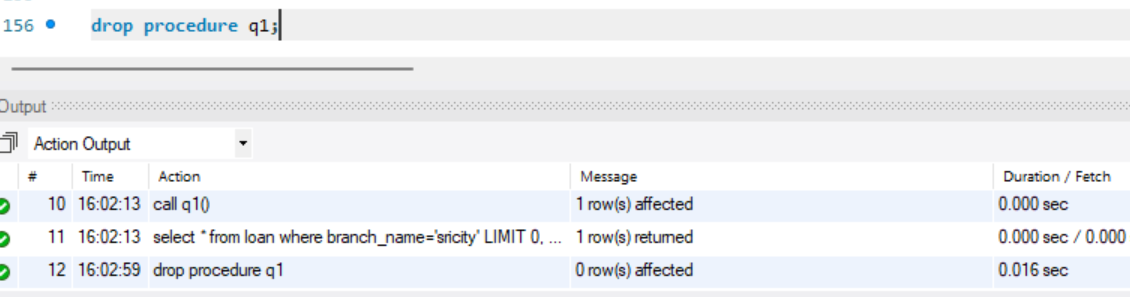
****

**ACTUAL QUESTIONS:**

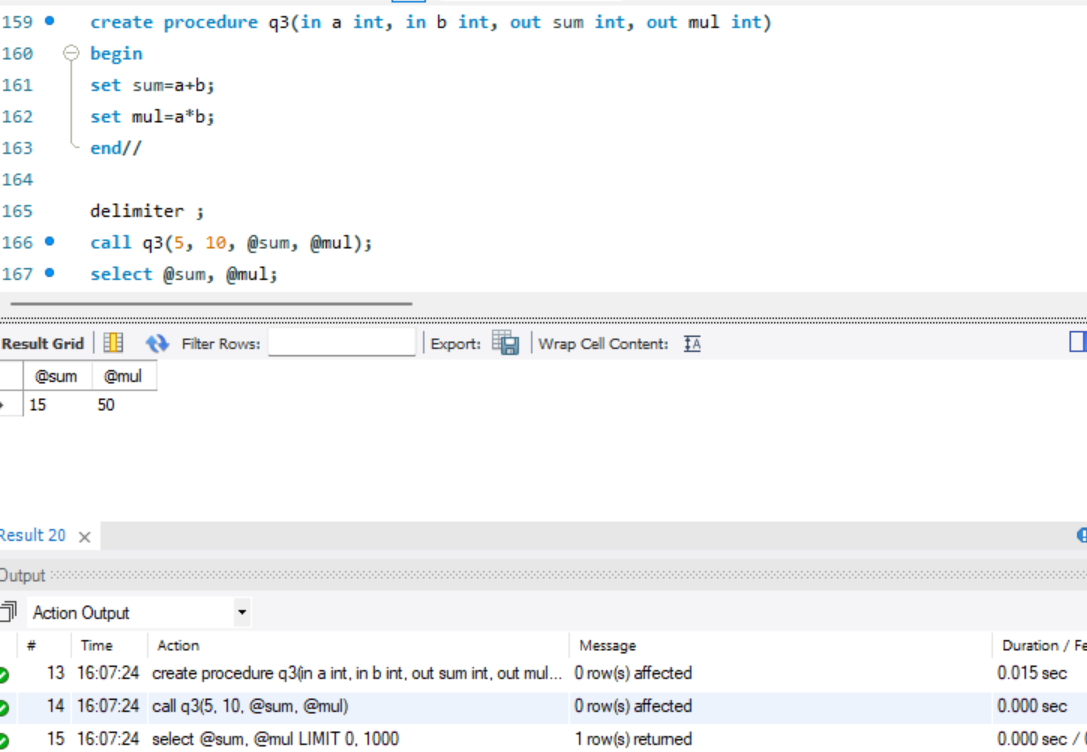
**Q1**

****

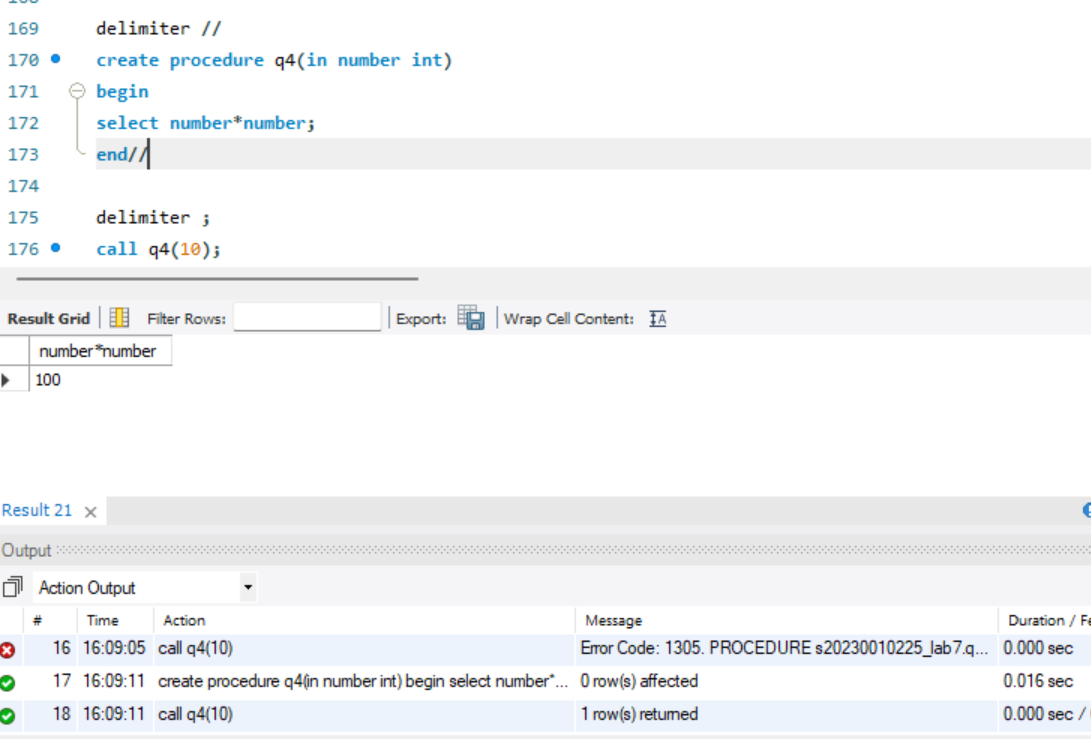
**Q2**

****

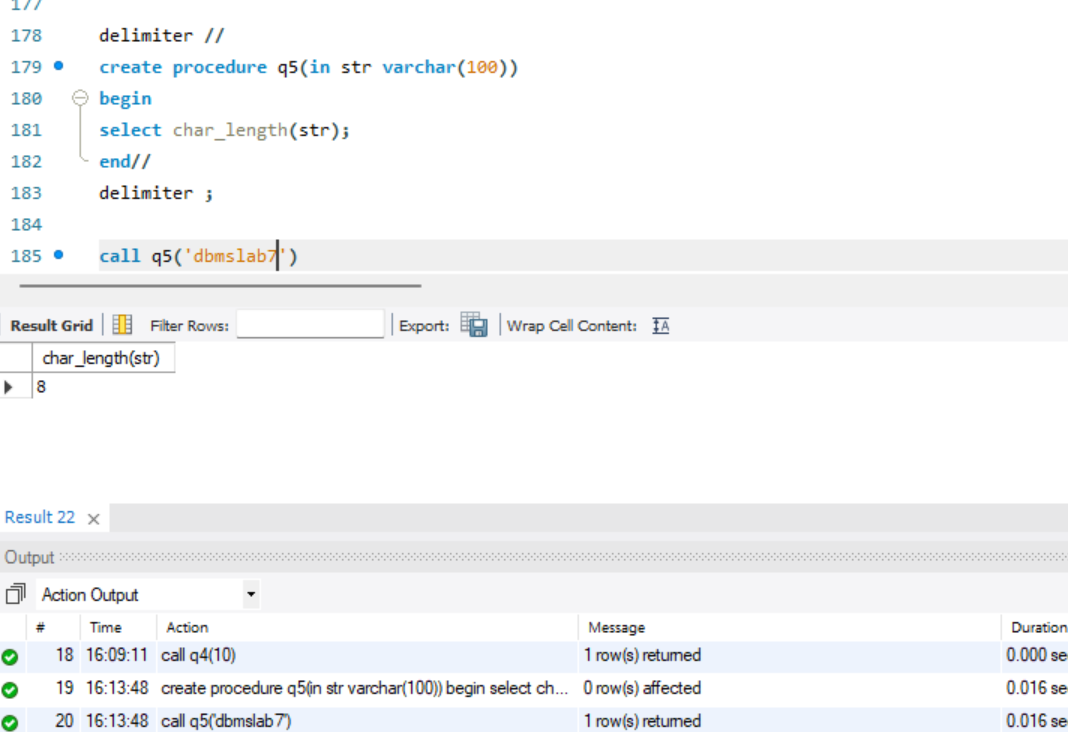
**Q3**

****

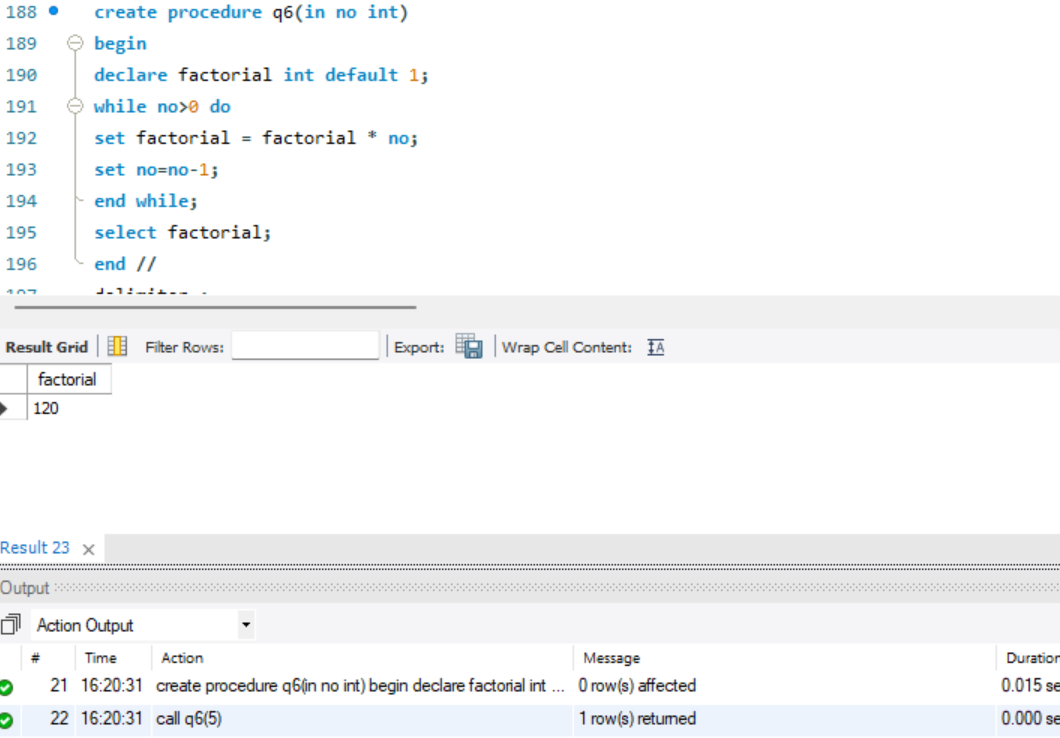
**Q4**

****

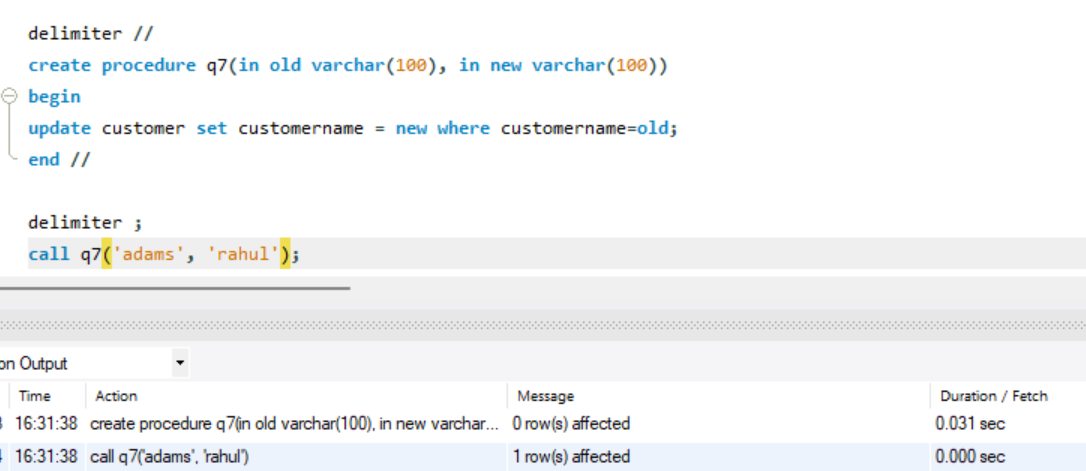
**Q5**

****

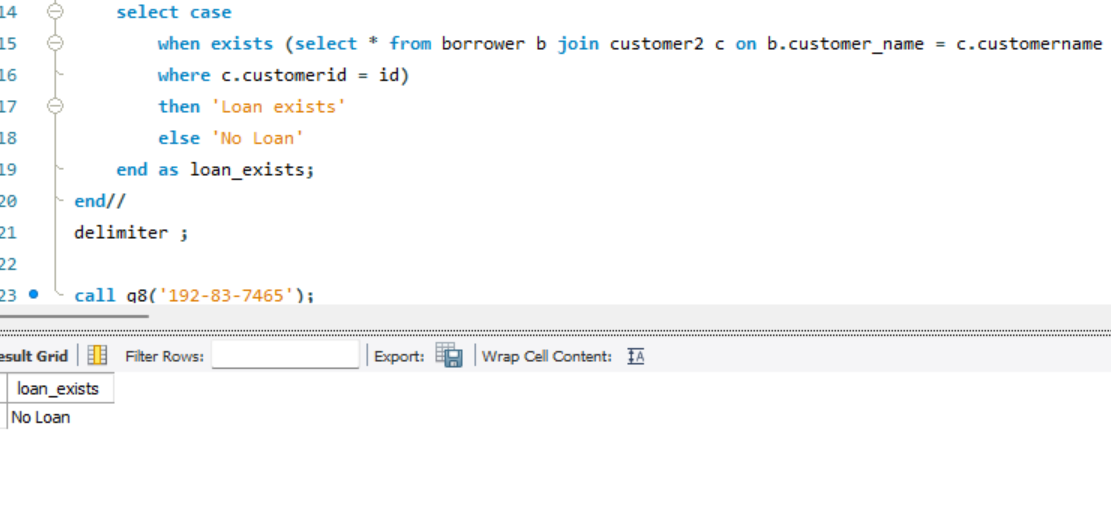
**Q6**

****

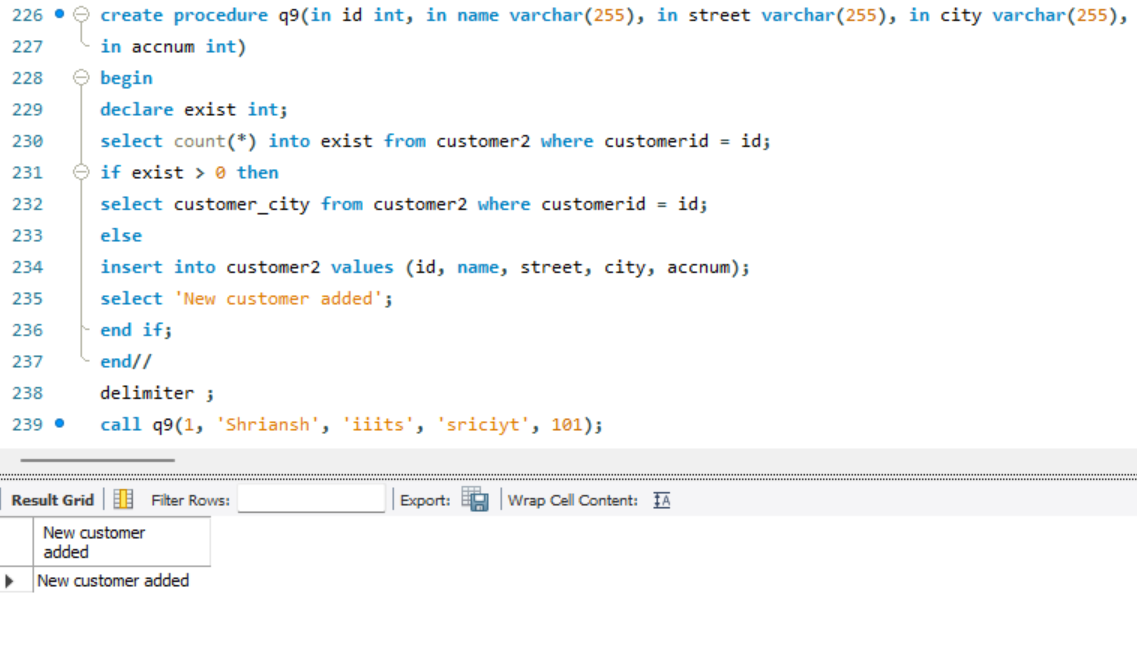
**Q7**

****

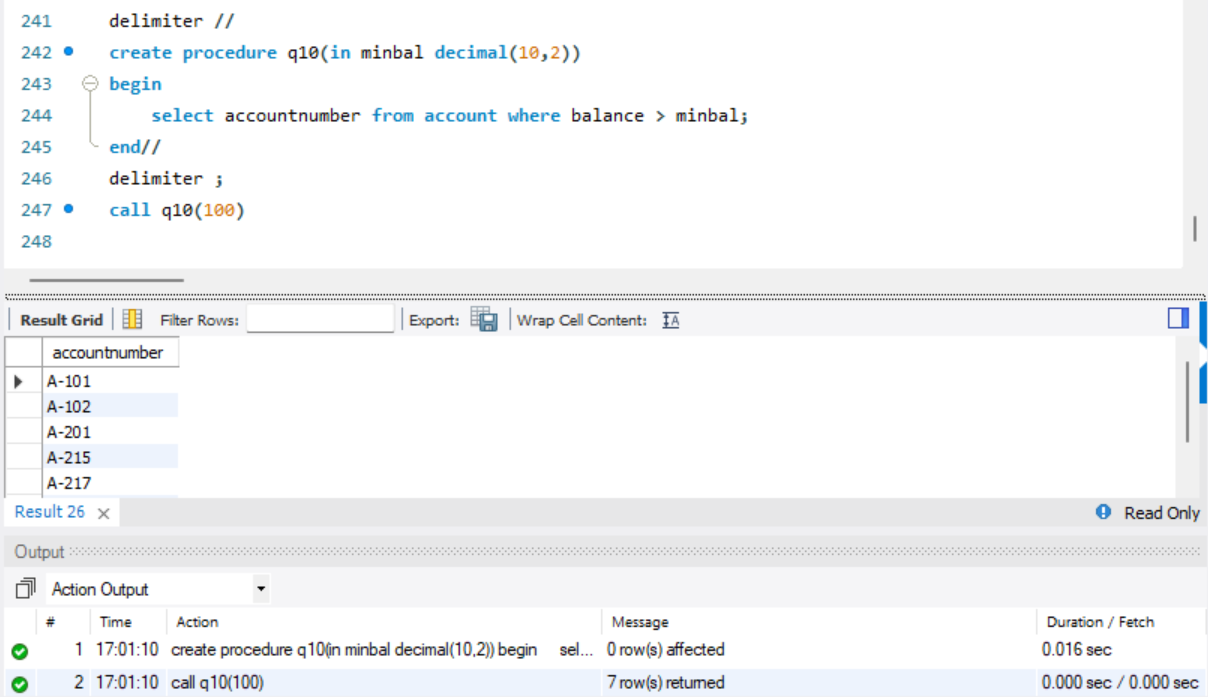
**Q8**

****

**Q9**

****

**Q10**

****

**SQL CODE:**

-- create database S20230010225\_LAB7

-- delimiter aa

-- create procedure sar()

-- begin

-- select count(\*) from borrower;

-- end aa

-- delimiter ;

-- call sar;

-- delimiter //

-- create procedure sar()

-- begin

-- declare a, b int default 0;

-- set a=10;

-- select count(\*) +a into b from borrower;

-- select b;

-- end//

-- delimiter ;

-- call sar();

-- drop procedure sar;

-- delimiter //

-- create procedure sar(in name varchar(10))

-- begin

-- select \* from customer where name=customername;

-- end//

-- delimiter ;

-- call sar('johnson');

-- delimiter //

-- create procedure temp(in balance int, out count int)

-- begin

-- select count(\*) into count from account where account.balance>=balance;

-- end//

-- delimiter ;

-- call temp(1000, @a);

-- select @a;

-- delimiter //

-- create procedure setcounter(inout counter int, in increment int)

-- begin

-- set counter = counter + increment ;

-- end//

-- delimiter ;

-- set @counter=1;

-- call setcounter(@counter,1);

-- call setcounter(@counter,10);

-- select @counter;

-- delimiter //

-- create procedure customerlevel( in id varchar(10), out level varchar(20))

-- begin

-- declare bal numeric(12,2) default 0;

-- select balance into bal from account where accountnumber=id; if bal > 700 then

-- set level='PLATINUM';

-- elseif bal<=700 and bal>300 then

-- set level='GOLD';

-- else

-- set level='SILVER';

-- end if;

-- end //

-- delimiter ;

-- call customerlevel('A-101', @var);

-- select @var;

-- delimiter //

-- create procedure account\_check(in id varchar(10),out status varchar(10))

-- begin

-- declare temp varchar(10) default 'Absent';

-- case id

-- when 'A-101' then

-- set status='Present';

-- Select status;

-- when 'A-102' then

-- set status='Present';

-- when 'A-201' then

-- set status='Present';

-- when 'A-215' then

-- set status='Present';

-- when 'A-217' then

-- set status='Present';

-- when 'A-222' then

-- set status='Present';

-- when 'A-305' then

-- set status='Present';

-- else

-- Set status='Absent';

-- end case;

-- end //

-- delimiter ;

-- call account\_check('A101', @var);

-- select @var;

-- delimiter //

-- create procedure customerlevel2( in id varchar(10),out level varchar(20))

-- begin

-- declare bal numeric(12,2) default 0;

-- select balance into bal from account where accountnumber = id;

-- case

-- when bal>700 then

-- set level='PLATINUM';

-- when bal<=700 and bal>300 then

-- set level='GOLD';

-- else

-- set level='SILVER';

-- end case;

-- end //

-- delimiter ;

-- call customerlevel2('A-101',@a);

-- select @a;

-- delimiter //

-- create procedure loopdemo()

-- begin

-- declare x int default 1;

-- declare str varchar(255) default '';

-- loop\_label: loop

-- if x>10 then

-- leave loop\_label;

-- end if;

-- set x= x+1;

-- if (x mod 2) then

-- iterate loop\_label;

-- else

-- set str=concat(str, x, ',');

-- end if;

-- end loop;

-- select str;

-- end //

-- delimiter ;

-- call loopdemo();

-- delimiter //

-- create procedure q1()

-- begin

-- insert into loan values('L-100', 'Sricity', 1000);

-- end//

-- delimiter ;

-- call q1();

-- select \* from loan where branch\_name='sricity';

-- drop procedure q1;

-- delimiter //

-- create procedure q3(in a int, in b int, out sum int, out mul int)

-- begin

-- set sum=a+b;

-- set mul=a\*b;

-- end//

-- delimiter ;

-- call q3(5, 10, @sum, @mul);

-- select @sum, @mul;

-- delimiter //

-- create procedure q4(in number int)

-- begin

-- select number\*number;

-- end//

-- delimiter ;

-- call q4(10);

-- delimiter //

-- create procedure q5(in str varchar(100))

-- begin

-- select char\_length(str);

-- end//

-- delimiter ;

-- call q5('dbmslab7')

-- delimiter //

-- create procedure q6(in no int)

-- begin

-- declare factorial int default 1;

-- while no>0 do

-- set factorial = factorial \* no;

-- set no=no-1;

-- end while;

-- select factorial;

-- end //

-- delimiter ;

-- call q6(5);

-- delimiter //

-- create procedure q7(in old varchar(100), in new varchar(100))

-- begin

-- update customer set customername = new where customername=old;

-- end //

-- delimiter ;

-- call q7('adams', 'rahul');

-- drop procedure q8;

-- delimiter //

-- create procedure q8(in id varchar(100))

-- begin

-- select case

-- when exists (select \* from borrower b join customer2 c on b.customer\_name = c.customername

-- where c.customerid = id)

-- then 'Loan exists'

-- else 'No Loan'

-- end as loan\_exists;

-- end//

-- delimiter ;

-- call q8('192-83-7465');

-- delimiter //

-- create procedure q9(in id int, in name varchar(255), in street varchar(255), in city varchar(255),

-- in accnum int)

-- begin

-- declare exist int;

-- select count(\*) into exist from customer2 where customerid = id;

-- if exist > 0 then

-- select customer\_city from customer2 where customerid = id;

-- else

-- insert into customer2 values (id, name, street, city, accnum);

-- select 'New customer added';

-- end if;

-- end//

-- delimiter ;

-- call q9(1, 'Shriansh', 'iiits', 'sriciyt', 101);

-- delimiter //

-- create procedure q10(in minbal decimal(10,2))

-- begin

-- select accountnumber from account where balance > minbal;

-- end//

-- delimiter ;

-- call q10(100)