Sum

create database programs;

use programs;

DELIMITER //

CREATE PROCEDURE addnum()

BEGIN

DECLARE a INT;

DECLARE b INT;

DECLARE C INT;

SET a=10;

SET b=20;

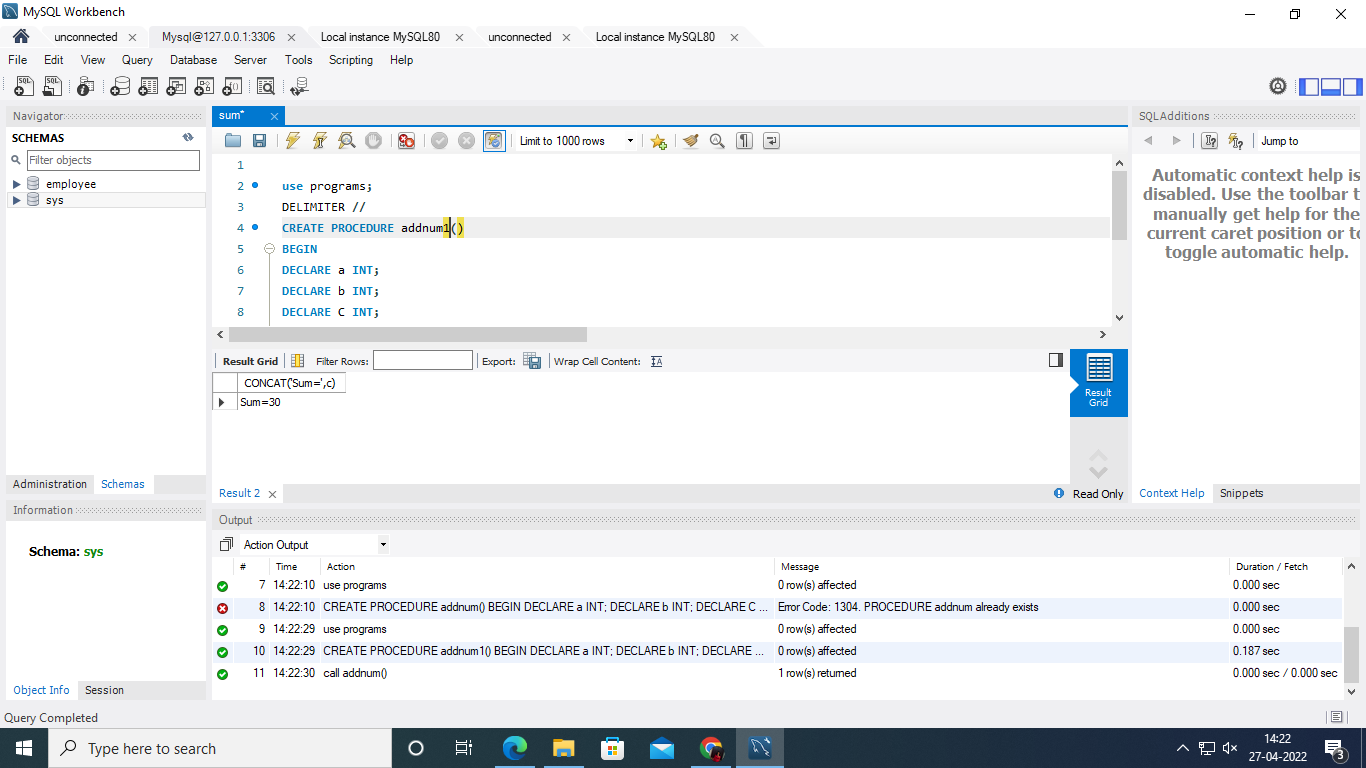
SET c=a+b;

SELECT CONCAT('Sum=',c);

END//

DELIMITER ;

call addnum();



Largest

use programs;

DELIMITER //

drop procedure if exists largestofthree//

CREATE PROCEDURE largestofthree()

BEGIN

DECLARE a,b,c INT;

SET a=10;

SET b=20;

SET c=30;

if(a>=b) then

if(a>=c) then

select concat(a,' is largest nymber');

end if;

else

if(b>=c) then

select concat(b,' is the largest');

Else

select concat(c,' is the largest');

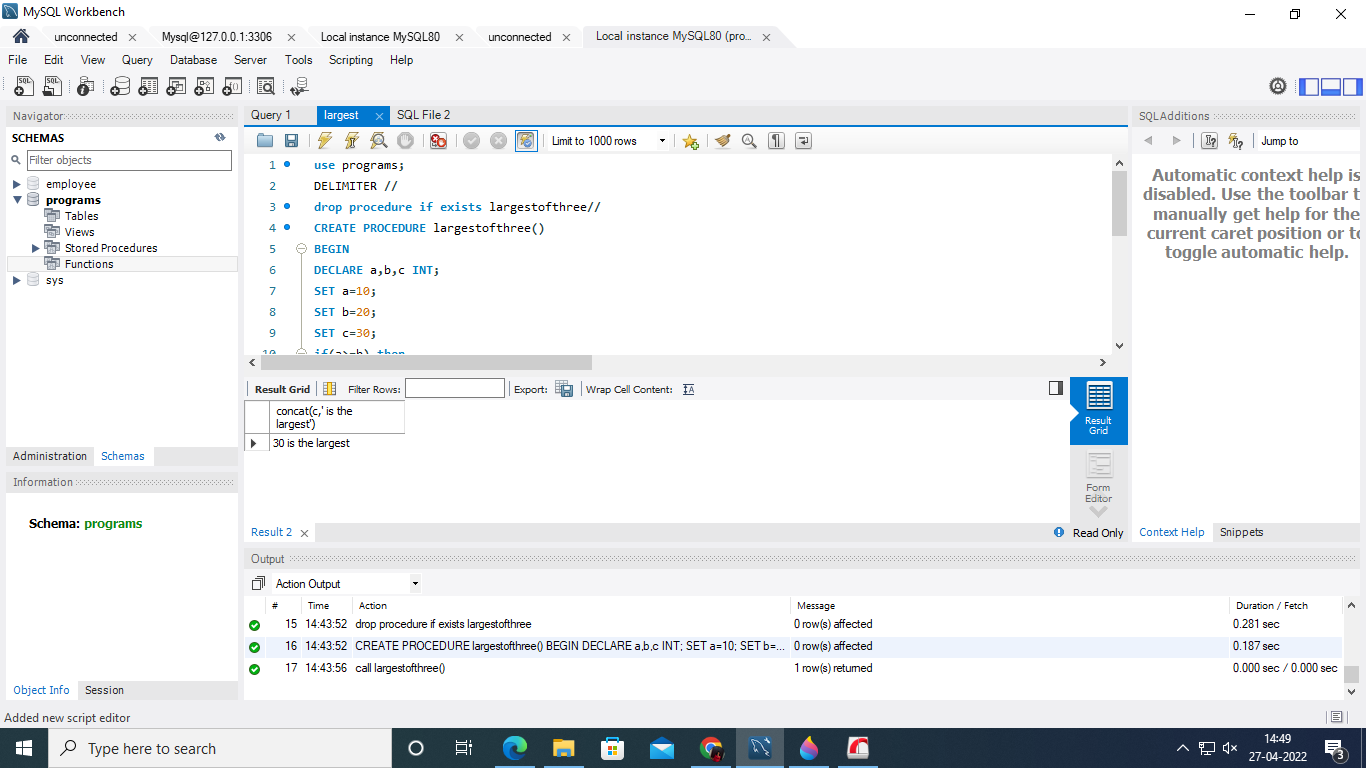
end if;

end if;

END//

DELIMITER ;

call largestofthree();



Sum n to m

use programs;

DELIMITER //

drop procedure if exists sumofmton//

CREATE PROCEDURE sumofmton(in n int,in m int)

BEGIN

DECLARE i,final INT;

SET i=n;

select concat(n,m);

SEt final=0;

label:while(i<=m)do

set final=final+i;

set i=i+1;

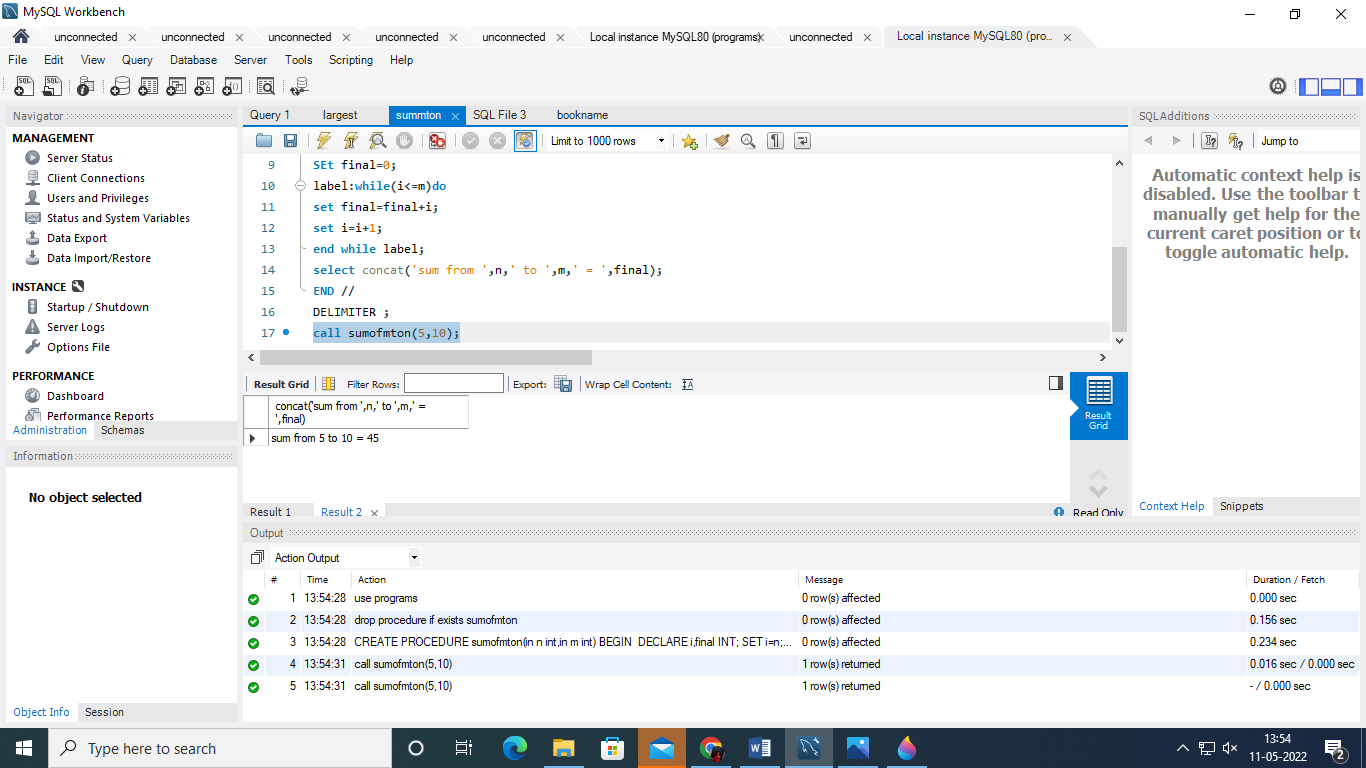
end while label;

select concat('sum from ',n,' to ',m,' = ',final);

END //

DELIMITER ;

call sumofmton(5,10);



Factorial

use programs;

DELIMITER //

drop procedure if exists factorial//

CREATE PROCEDURE factorial(in n int)

BEGIN

DECLARE num,fact INT;

SET num=n;

SET fact=1;

label:while(num>0)do

set fact=num\*fact;

set num=num-1;

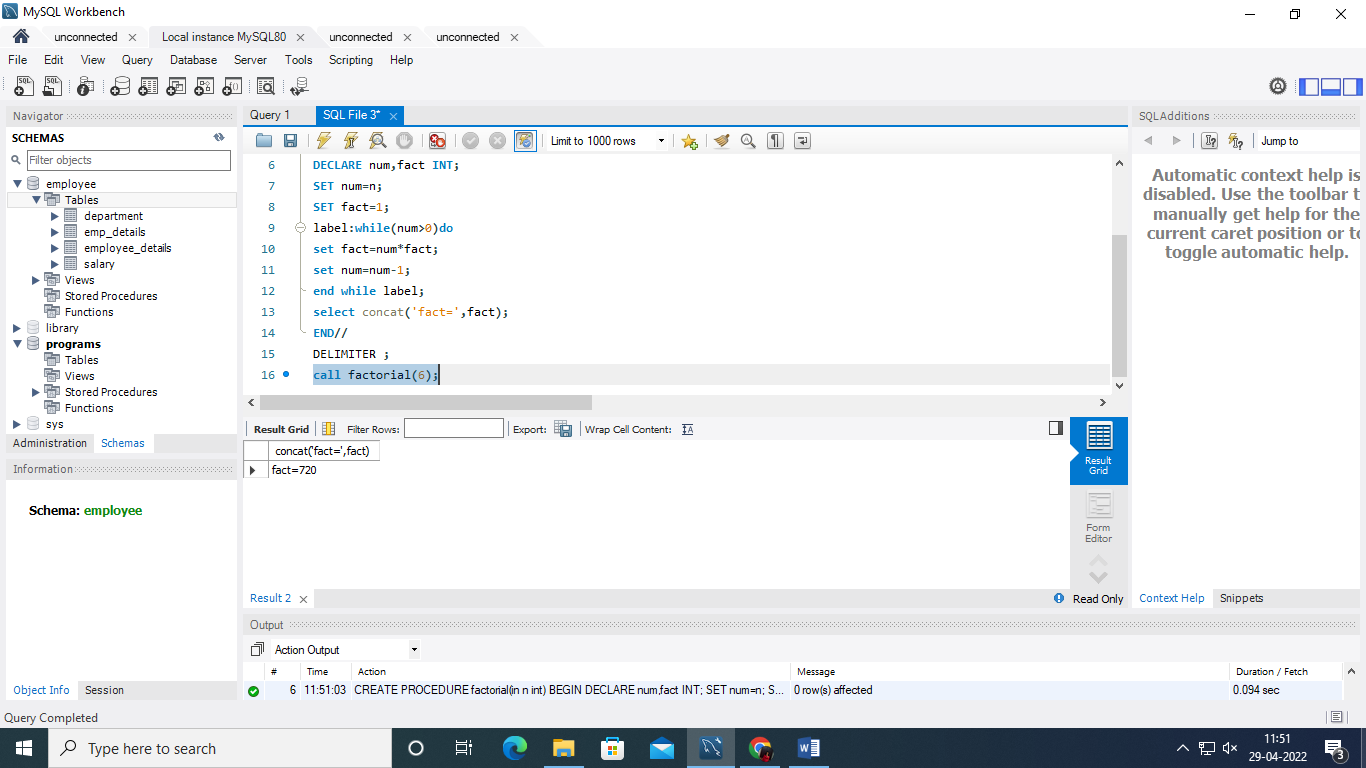
end while label;

select concat('fact=',fact);

END//

DELIMITER ;

call factorial(6);



Odd or even

use programs;

DELIMITER //

drop procedure if exists oddoreven//

CREATE PROCEDURE oddoreven(in n int)

BEGIN

DECLARE num INT;

if mod(n,2)=0 then

select concat(n,' is even number');

else

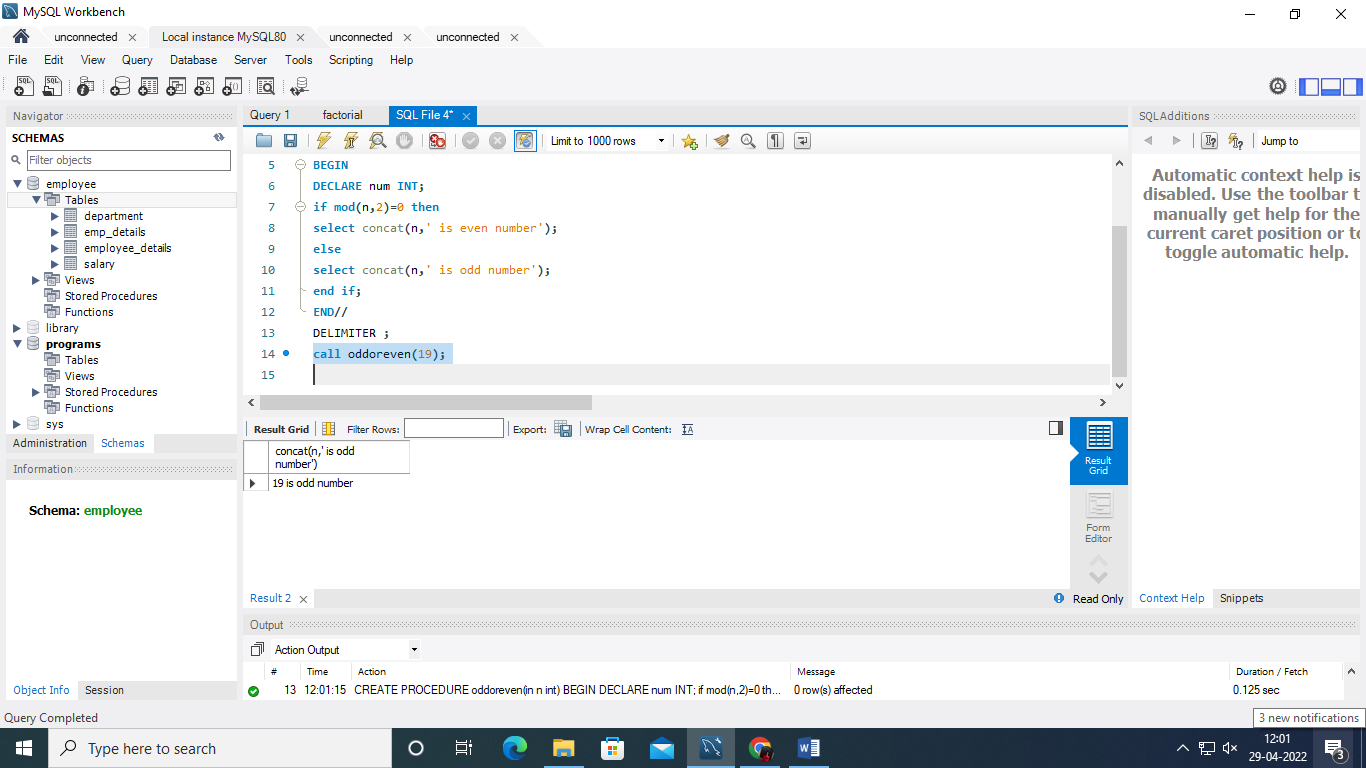
select concat(n,' is odd number');

end if;

END//

DELIMITER ;

call oddoreven(19);



create database library;

use library;

create table Bookdetails(bookno int primary key,bookname varchar(30),bookauthor varchar(30));

create table issuedetails(issueid int primary key,bookno int,issuedate date,returnstatus char(5));

create table returndetails(returnid int primary key,issueid int,bookno int,returndate date);

insert into bookdetails(bookno,bookname,bookauthor)values(111,"Introduction to c","E.Balaguruswmi");

insert into bookdetails(bookno,bookname,bookauthor)values(123,"Java programming","Surbhi Kakar");

insert into bookdetails(bookno,bookname,bookauthor)values(122,"OOPS with c++","Sourav Sahay");

insert into bookdetails(bookno,bookname,bookauthor)values(134,"ADS","Alfred");

insert into bookdetails(bookno,bookname,bookauthor)values(124,"Operating system","D.m Dhamdheer");

DELIMITER //

drop procedure if exists orderofbooktitle//

CREATE PROCEDURE orderofbooktitle()

BEGIN

select \* from bookdetails order by bookname asc;

END //

DELIMITER ;

call orderofbooktitle();

DELIMITER //

drop procedure if exists issuebook//

CREATE PROCEDURE issuebook()

BEGIN

insert into issuedetails values(6,4,curdate(),'no');

select \* from issuedetails;

END //

DELIMITER ;

call issuebook();

DELIMITER //

drop procedure if exists bookreturn//

CREATE PROCEDURE bookreturn()

BEGIN

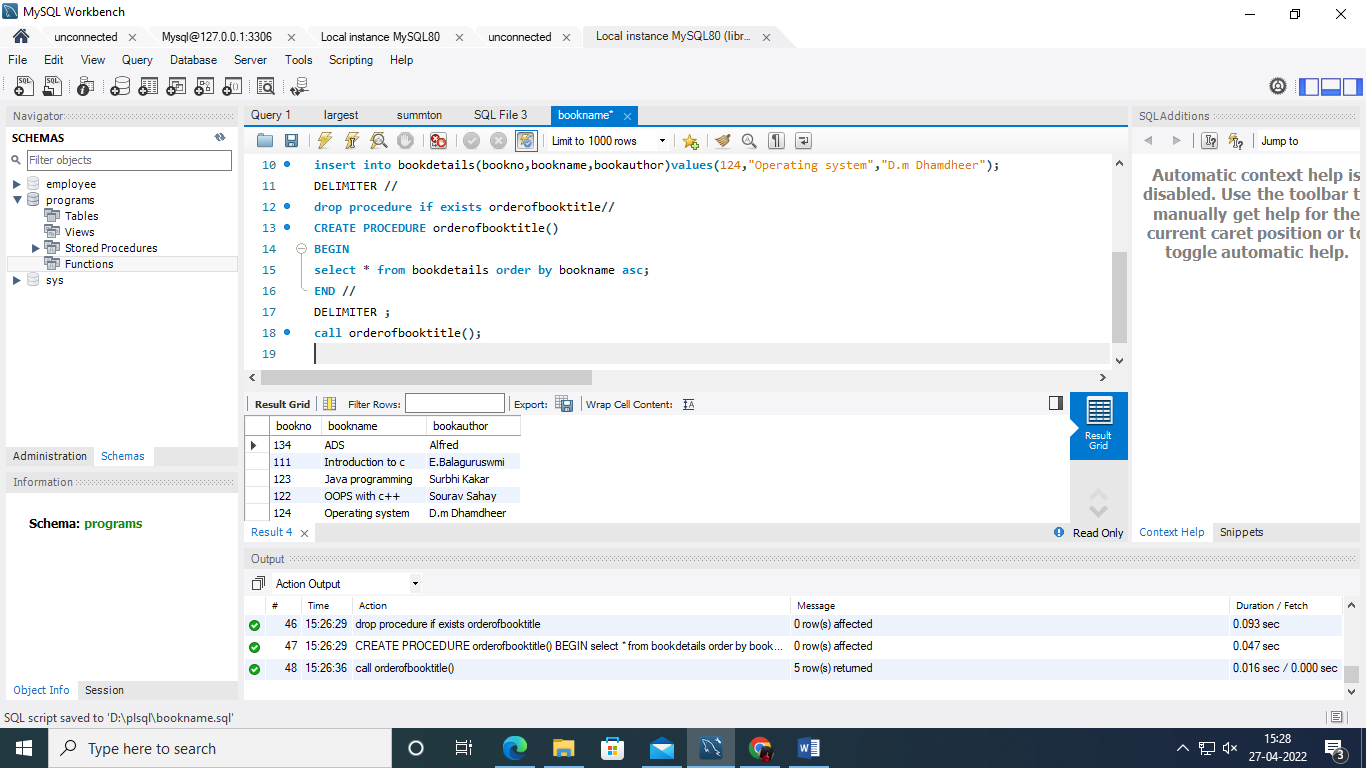
insert into returndetails values(3,4,4,curdate());

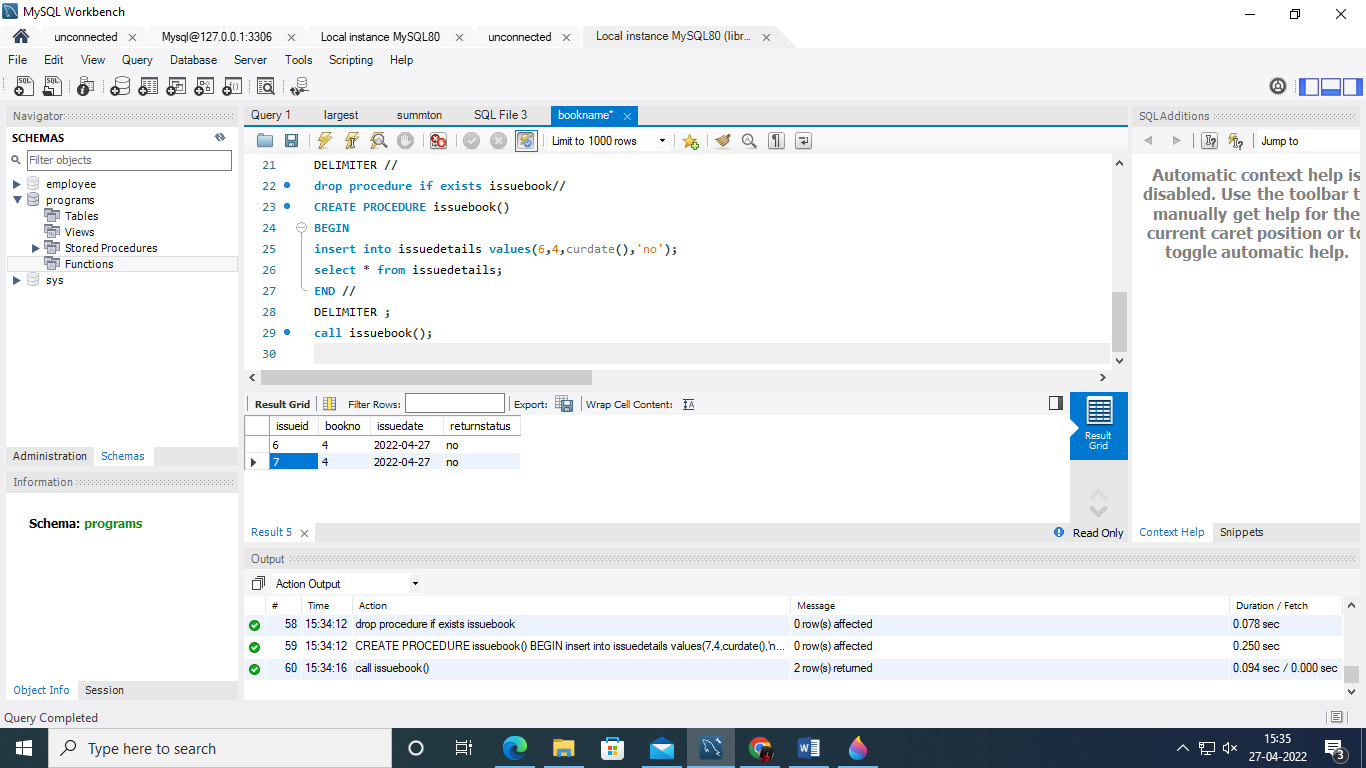
update issuedetails set returnstatus='yes' where issueid=7;

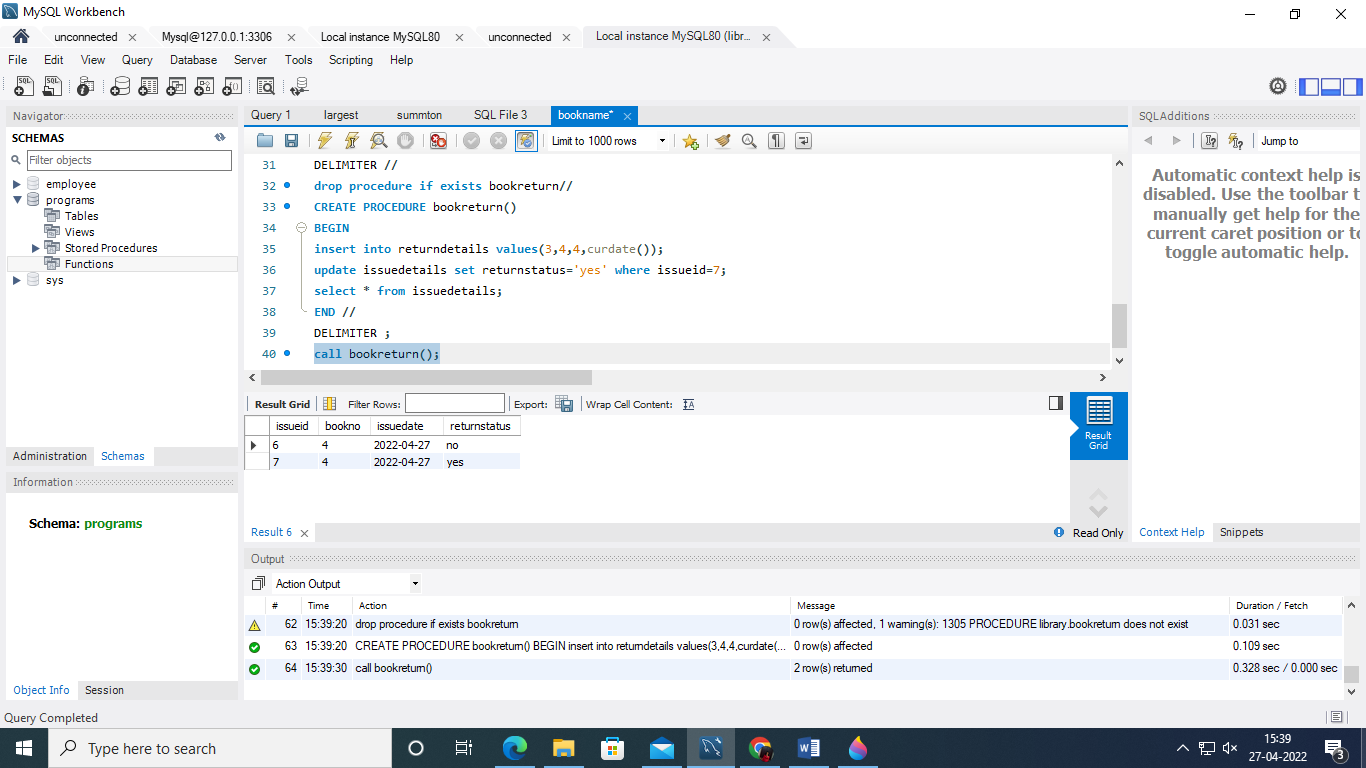
select \* from issuedetails;

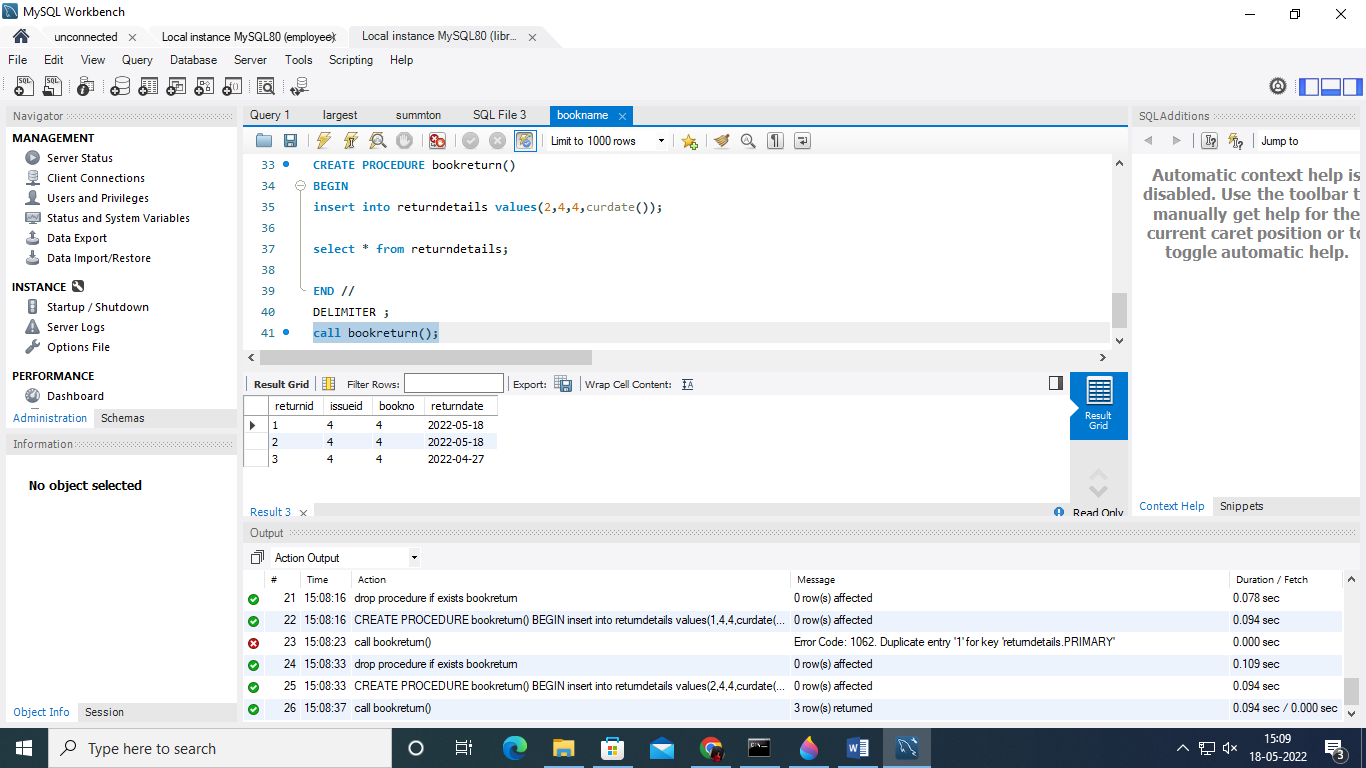
END //

DELIMITER ;

call bookreturn();







Trigger

use library;

create table libHistory(bookno int primary key,bookstatus varchar(20));

select \* from issuedetails;

insert into issuedetails values(5,3,curdate(),'no');

select \* from libHistory;

insert into returndetails values(6,4,4,curdate());

select \* from libHistory;

issue trigger

CREATE DEFINER=`root`@`localhost` TRIGGER `issuedetails\_AFTER\_INSERT` AFTER INSERT ON `issuedetails` FOR EACH ROW BEGIN

declare id int;

declare stat varchar(30);

set stat='issue';

select bookno into id from issuedetails where issueid=new.issueid;

insert into libHistory values(id,stat);

END

Return trigger

CREATE DEFINER=`root`@`localhost` TRIGGER `returndetails\_AFTER\_INSERT` AFTER INSERT ON `returndetails` FOR EACH ROW BEGIN

declare id int;

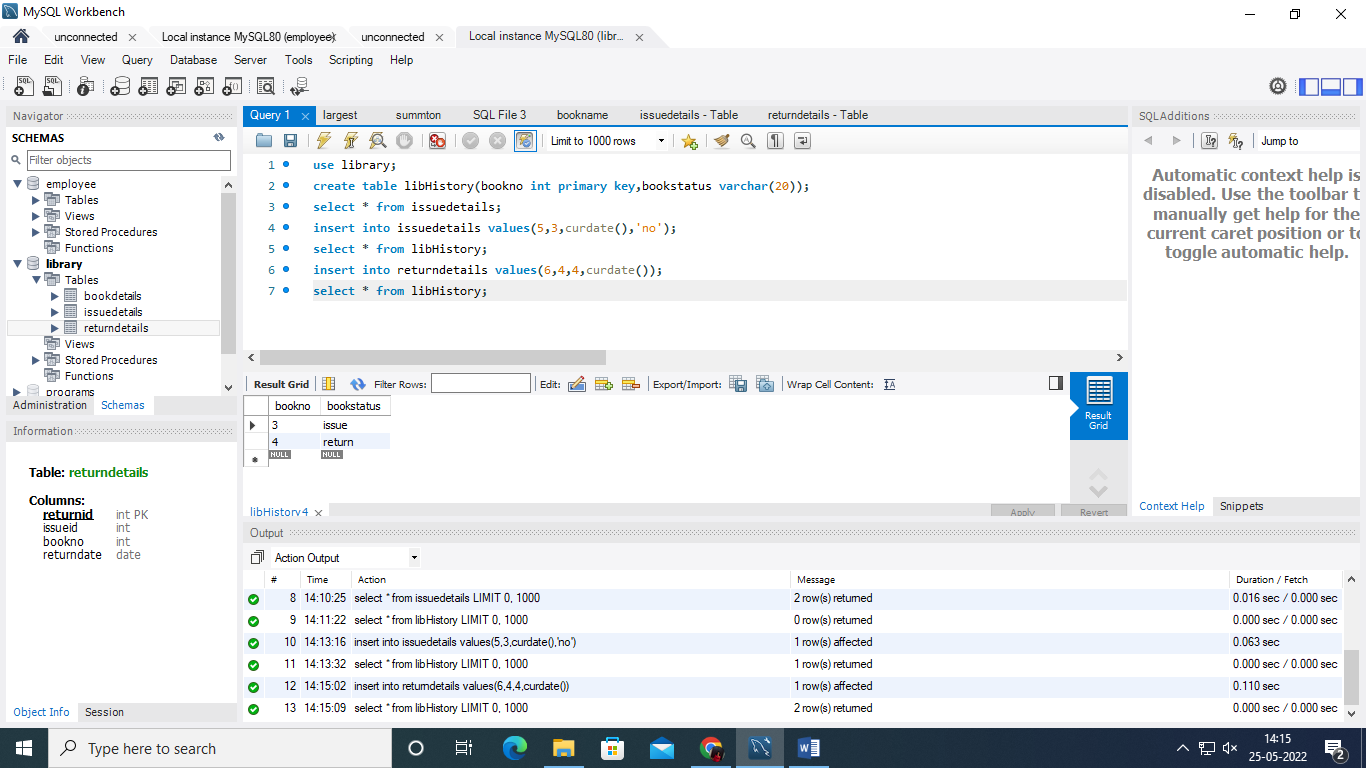
declare stat varchar(30);

set stat='return';

select bookno into id from returndetails where returnid=new.returnid;

insert into libHistory values(id,stat);

END



Funcions

select operation('insert',8);

select \* from bookdetails;

functions code

CREATE DEFINER=`root`@`localhost` FUNCTION `operation`(op varchar(20),id int) RETURNS int

DETERMINISTIC

BEGIN

if (op = 'insert') then

insert into bookdetails values(id,'Book\_5','Author\_7');

elseif (op = 'update') then

update bookdetails set bookname = 'book\_001', bookauthor='Auth\_001'

where bookno = id;

elseif (op = 'delete') then delete from bookdetails where bookno = id;

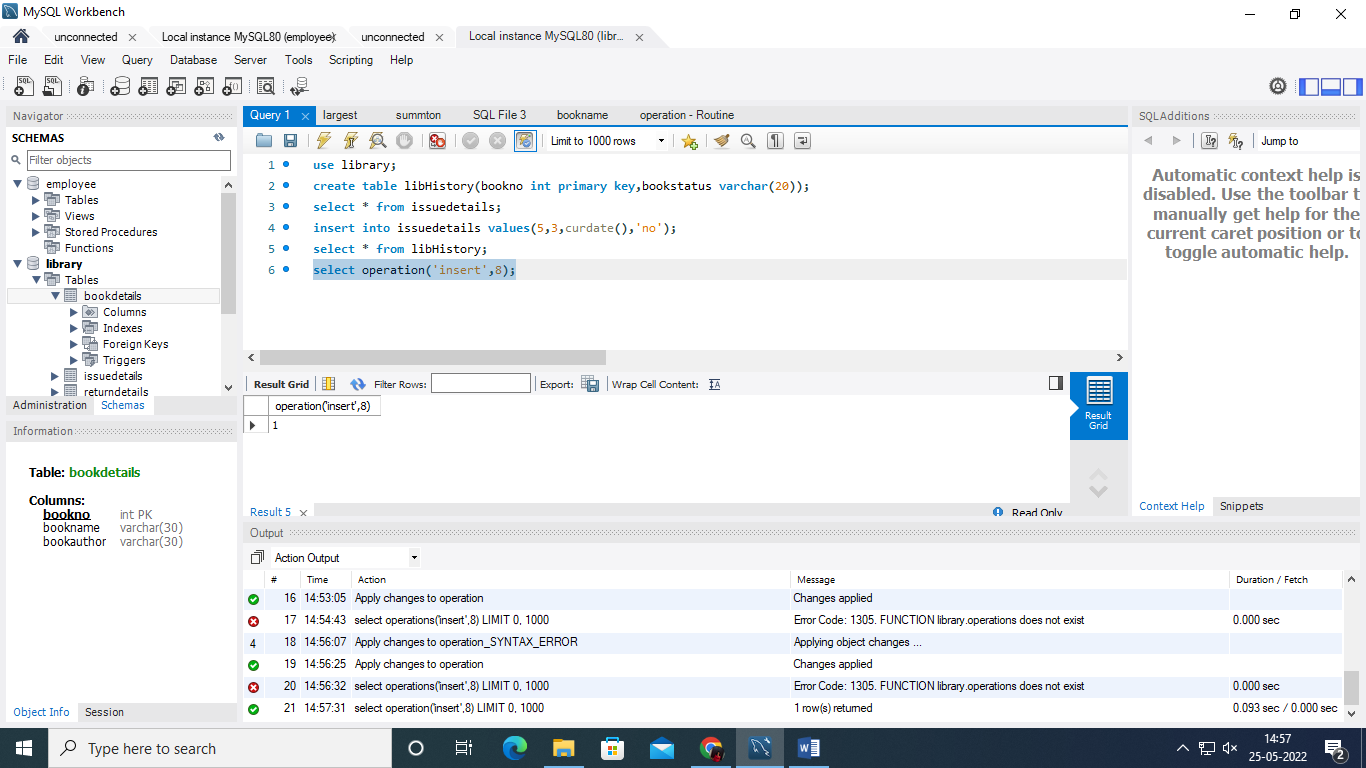
else

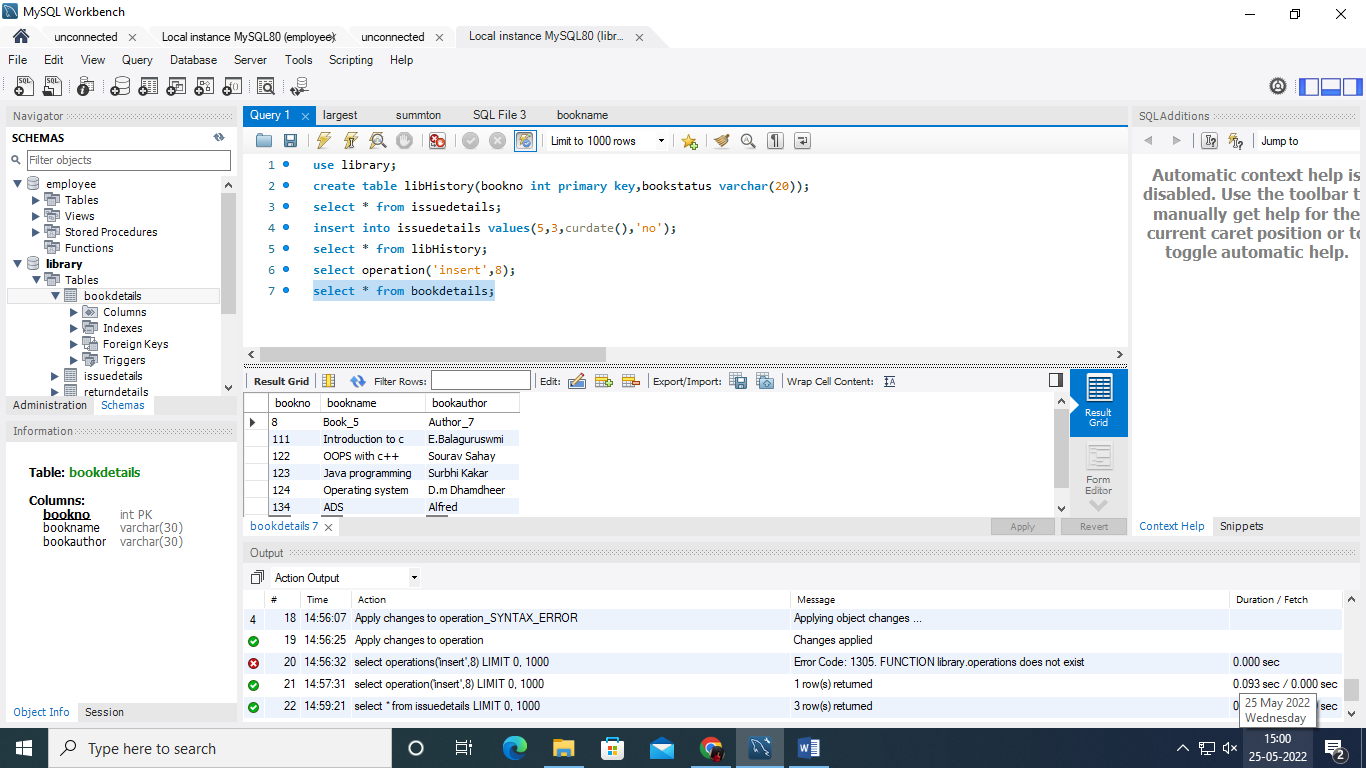
return 'INVALID';

end if;

RETURN 1;

END





Function 2

CREATE DEFINER=`root`@`localhost` FUNCTION `returnUpdate`(id int) RETURNS int

DETERMINISTIC

BEGIN

delete from issuedetails where issueid=id;

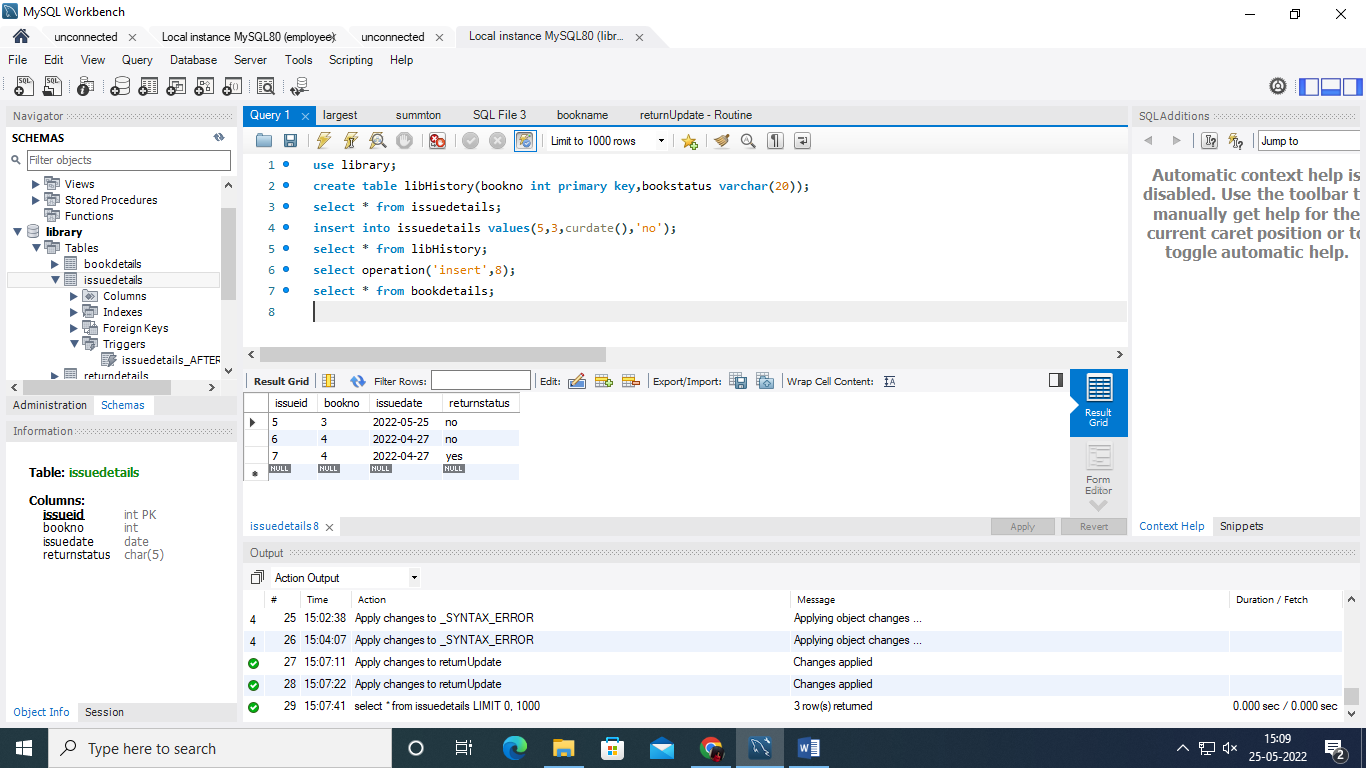
RETURN 1;

END

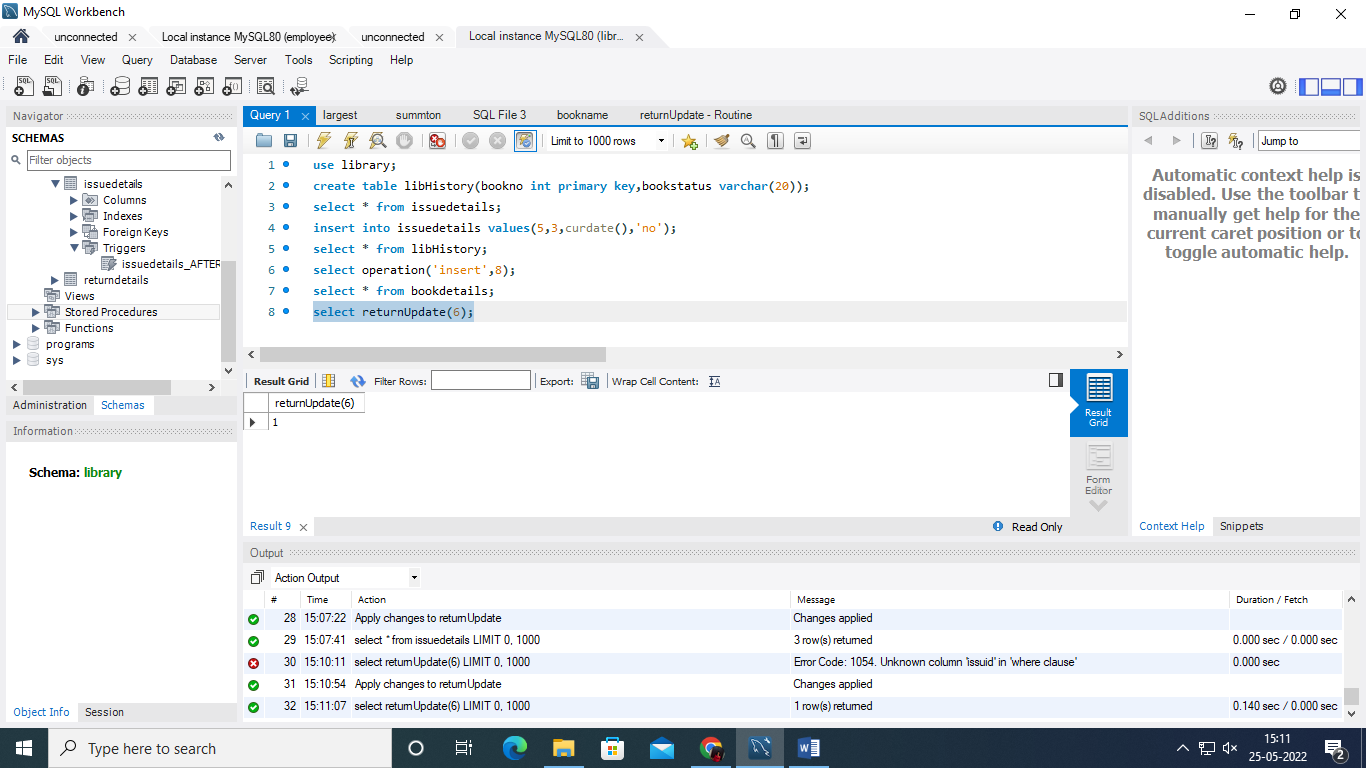
Query

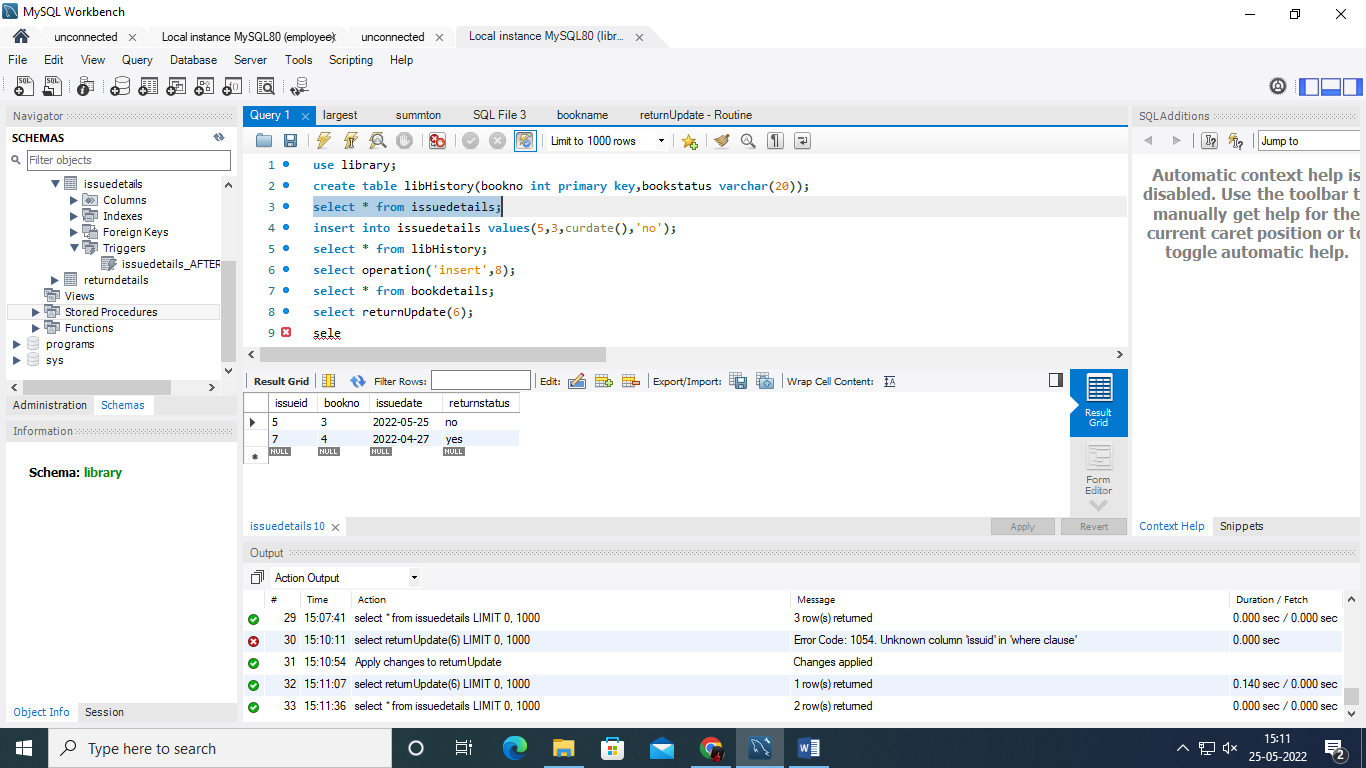
select returnUpdate(6);

select \* from issuedetails;



After





Cursor

delimiter //

drop procedure if exists newcursor;

create procedure newcursor(inout namelist varchar(100))

begin

declare is\_done int default 0;

declare b\_name varchar (20) default " ";

declare libcursor cursor for select bookname from bookdetails;

declare continue handler for not found set is\_done = 1;

open libcursor;

getdata: loop

fetch libcursor into b\_name;

if is\_done =1 then

leave getdata;

end if;

set namelist = concat(b\_name,' , ',namelist);

end loop;

close libcursor;

end//

set @namelist=" ";

call newcursor(@namelist);

select@namelist;

