Felix Filipi

2301877590

Pengerjaan menggunakan MySQL untuk nomer 1, 2, 3, 4, dan 6,10. Karena menurut dokumentasi MySQL untuk soal nomor 5 tidak bisa dikerjakan, karena MySQL tidak support soal tersebut.

Untuk nomor 5:  
A stored function or trigger cannot modify a table that is already being used (for reading or writing) by the statement that invoked the function or trigger.

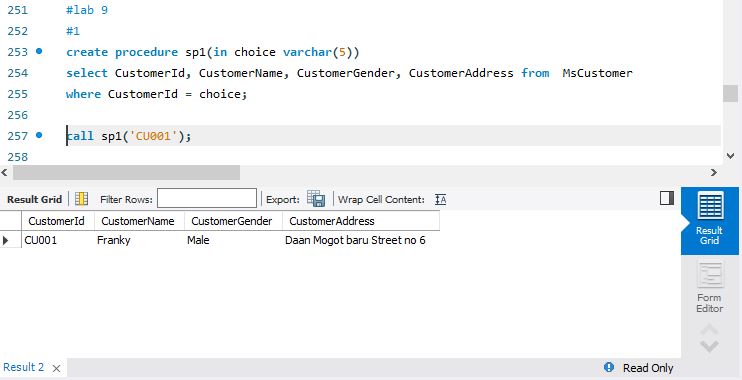
#1

create procedure sp1(in choice varchar(5))

select CustomerId, CustomerName, CustomerGender, CustomerAddress from MsCustomer

where CustomerId = choice;

call sp1('CU001');



#2

Delimiter $$

Create procedure sp2(in input varchar(50))

Begin

if (select (length(CustomerName) % 2 = 1) from MsCustomer where input=CustomerName) then

select 'Character Length of Mentor Name is an Odd Number';

else select MsCustomer.CustomerId, MsCustomer.CustomerName, MsCustomer.CustomerGender,

HeaderSalonServices.TransactionId, HeaderSalonServices.TransactionDate

from MsCustomer,HeaderSalonServices where MsCustomer.CustomerName like

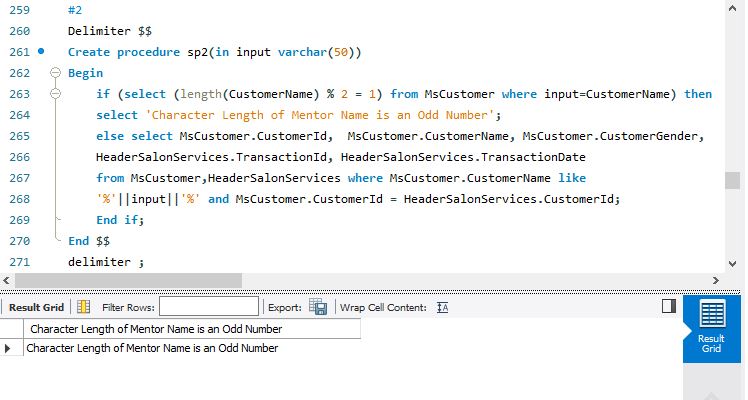
'%'||input||'%' and MsCustomer.CustomerId = HeaderSalonServices.CustomerId;

End if;

End $$

delimiter ;

call sp2('Elysia Chen');



#3

Delimiter $$

Create procedure sp3(in InputId char(5),InputName varchar(50),

InputGender varchar(5), InputPhone varchar(13))

Begin

if exists(select \* from MsStaff Where InputId = StaffId)

then update MsStaff

set StaffName = InputName, StaffGender = InputGender, StaffPhone = InputPhone

where InputId = StaffId;

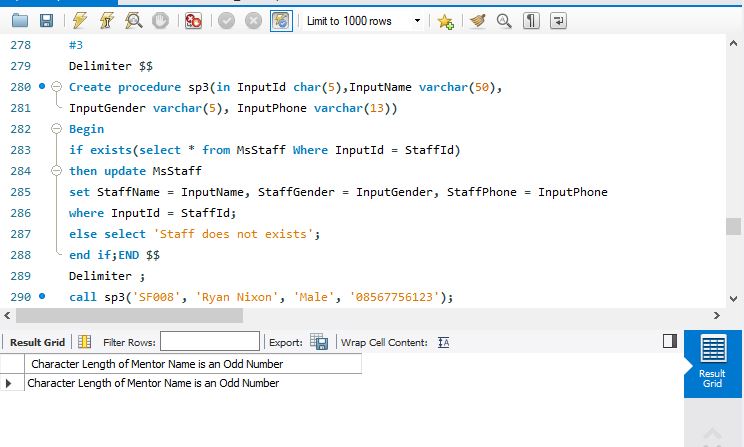
else select 'Staff does not exists';

end if;

END $$

Delimiter ;

call sp3('SF008', 'Ryan Nixon', 'Male', '08567756123');



#4

Delimiter $$

Create Trigger trig1

after update on MsCustomer for each row

begin

CREATE TEMPORARY TABLE temp(

customerid char(5),

customername varchar(50),

customergender varchar(50),

customerphone varchar(50),

customeraddress varchar(50)

);

insert into temp values (old.customerid,old.customername,old.customergender,old.customerphone

,old.customeraddress),(new.customerid,new.customername,new.customergender,new.customerphone

,new.customeraddress);

end $$

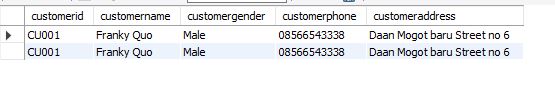
Delimiter ;

Code :



Querry : Update MsCustomer SET CustomerName = 'Franky Quo' WHERE CustomerId = 'CU001';

Result:



Untuk nomor 5 menggunakan PostgreSQL sebagai alternatif MySQL

#5 (PostgreSQL)

create function temp2()

returns trigger as $$

begin

Delete from mscustomer where customerid in

(select customerid from mscustomer limit 1);

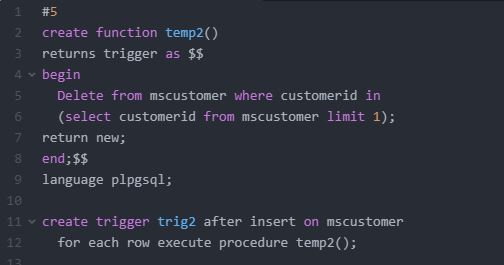
return new;

end;$$

language plpgsql;

create trigger trig2 after insert on mscustomer

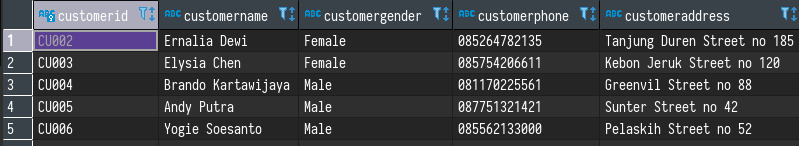
for each row execute procedure temp2();



Querry:

Insert into MsCustomer values (‘CU006’,’Yogie Soesanto’,’Male’,’085562133000’,’Pelaskih Street no 52’);

Result:



#6

delimiter $$

create trigger trig3

before delete on MsCustomer for each row

begin

create temporary table temp(

customerid char(5),

customername varchar(50),

customergender varchar(50),

customerphone varchar(50),

customeraddress varchar(50)

);

insert into mscustomer\_removed values

(old.customerid, old.customername, old.customergender, old.customerphone,

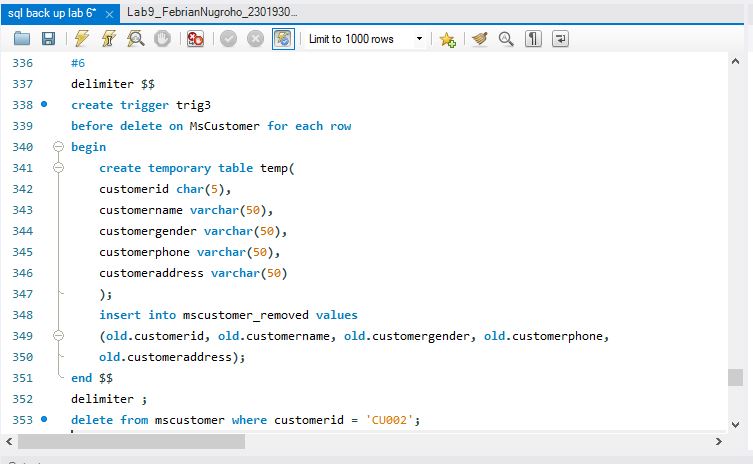
old.customeraddress);

end $$

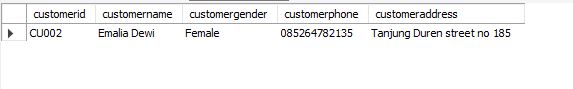
delimiter ;

delete from mscustomer where customerid = 'CU002';

Code:



Querry: delete from mscustomer where customerid = 'CU002';

Result: 

#7 (PostgreSQL)

create or replace function cur1() returns text as $$

declare staffname msstaff%rowtype;

declare tulisan text default 'The length from Staffname';

declare ganjil\_genap record;

declare curs\_staffname cursor for

select msstaff.staffname from msstaff;

begin open curs\_staffname;

loop fetch curs\_staffname into ganjil\_genap;

exit when not found;

if length(ganjil\_genap.staffname) % 2 = 0 then

raise notice '% % is odd number',tulisan,ganjil\_genap.staffname;

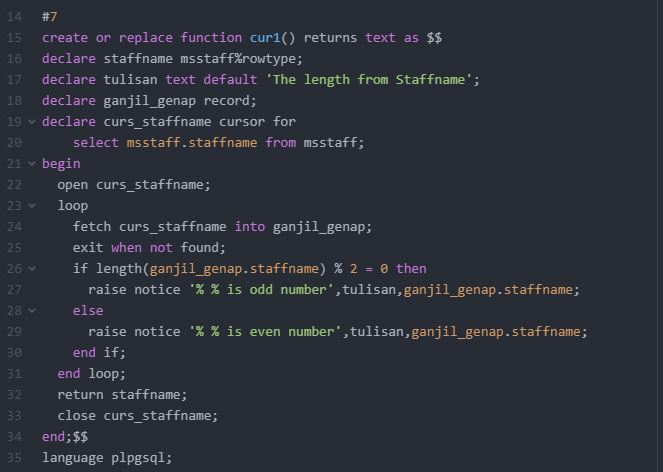
else raise notice '% % is even number',tulisan,ganjil\_genap.staffname;

end if; end loop; return staffname;

close curs\_staffname;

end;$$ language plpgsql;

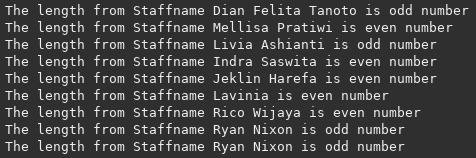
Code:



Querry:

Select cur1();

Result:



#8 (PostgreSQL)

create or replace function sp4(inputan varchar)

returns text as $$

declare staffname msstaff%rowtype;

declare tulisan text default 'Staffname';

declare name\_position record;

declare curs\_msstaff cursor(inputan varchar)

for select \*from msstaff where msstaff.staffname like inputan;

begin

open curs\_msstaff(inputan);

loop

fetch curs\_staffname into name\_position;

exit when not found;

raise notice '% : % : %',tulisan, name\_position.staffname, name\_position.staffposition;

end loop;

return staffname;

close curs\_staffname;

end;$$

language plpgsql;

Code:



Querry:

Select cur2();

Result:



#9 (PostgreSQL)

create or replace function sp5(inputan varchar)

returns text as $$

declare tulisan text default 'Customer Name';

declare customername mscustomer%rowtype;

declare nama\_tanggal record;

declare curs\_customer cursor(inputan varchar)

for select \*from mscustomer, headersalonservices, detailsalonservices

where headersalonservices.transactionid = detailsalonservices.transactionid and

mscustomer.customerid = headersalonservices.customerid and

headersalonservices.customerid like inputan;

begin

open curs\_customer(inputan);

loop

fetch curs\_customer into nama\_tanggal;

exit when not found;

if (right(nama\_tanggal.treatmentid, 1)::int) % 2 = 0 then

raise notice '% : % Transaction Date is %',titles, nama\_tanggal.CustomerName, nama\_tanggal.TransactionDate;

else

raise notice 'User ID is odd number';

end if;

end loop;

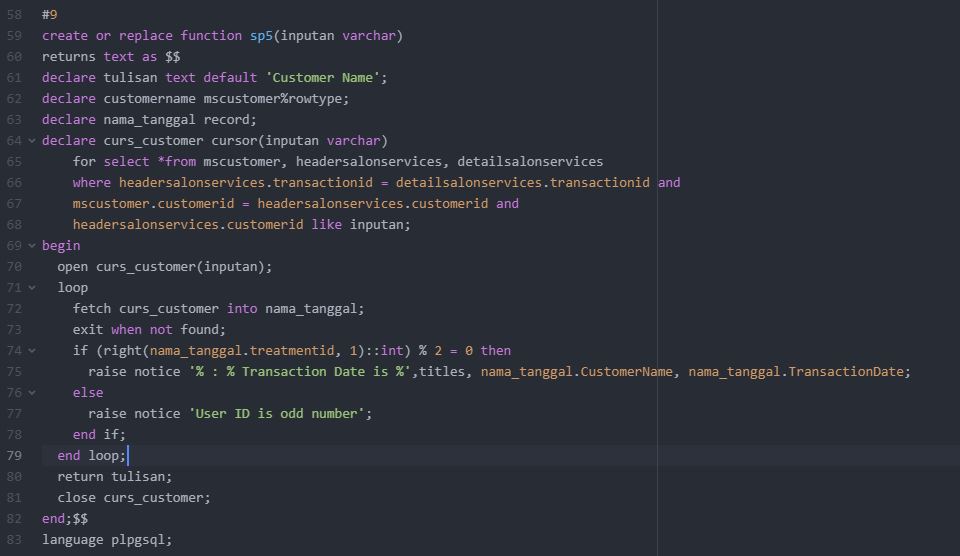
return tulisan;

close curs\_customer;

end;$$

language plpgsql;

Code:



Querry:

Select cur3();

Result:



#10

drop function sp1;

drop function sp2;

drop procedure sp3;

drop function sp4;

drop function sp5;

drop trigger trig1;

drop trigger trig2;

drop trigger trig3;