BAZY DANYCH

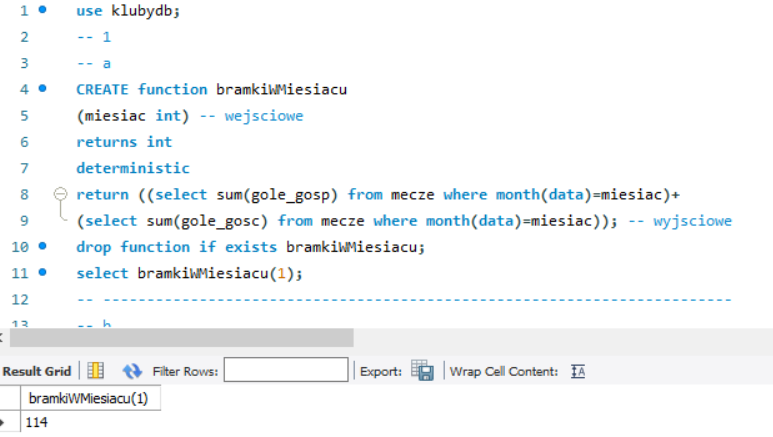
LISTA 8

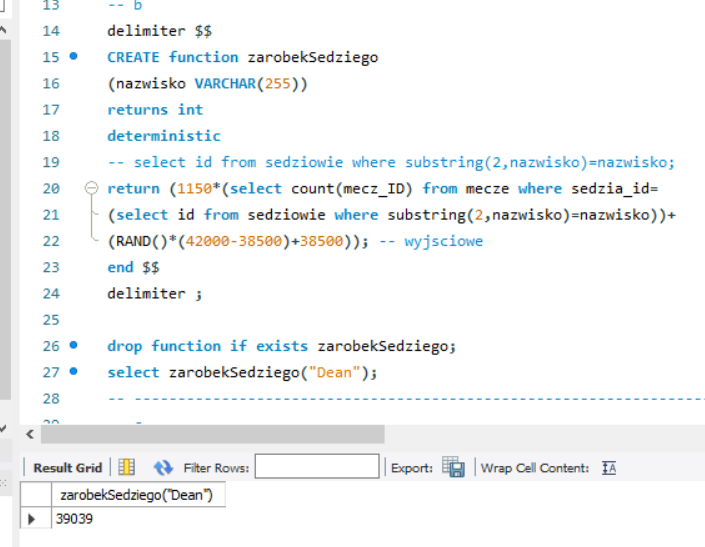
WIOLETTA ŁUPKOWSKA, 244831

PON 15:15

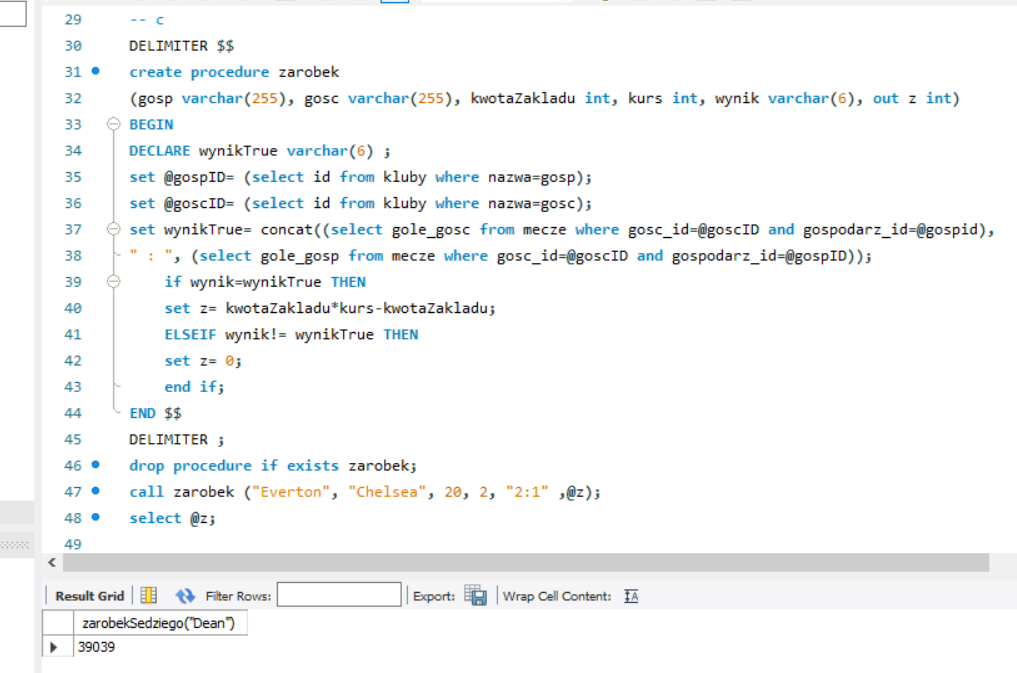
1

a)

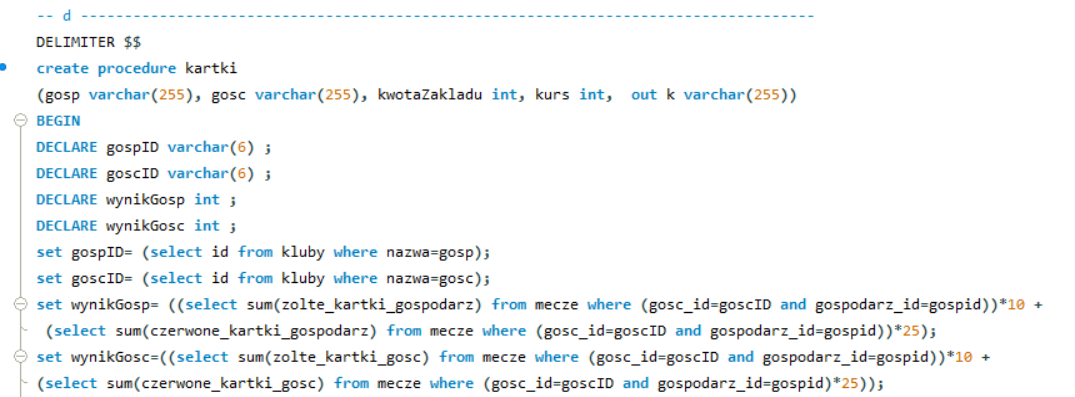


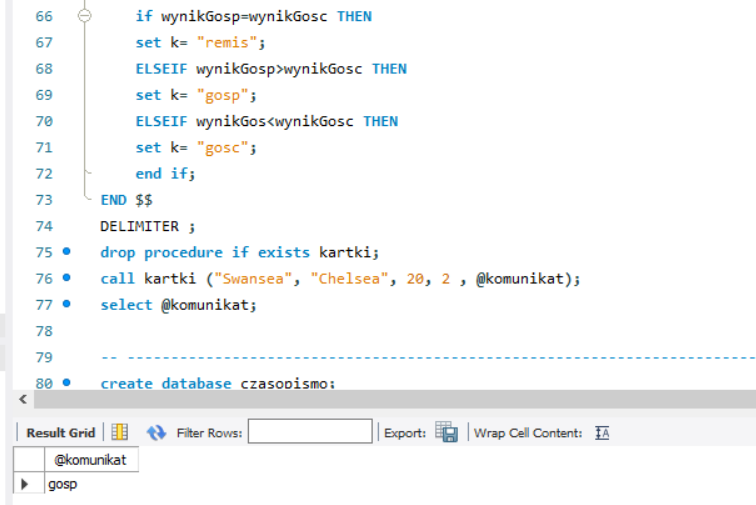
b) 

c)

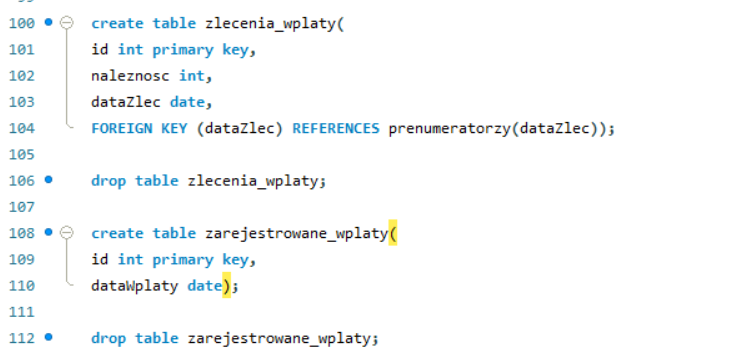
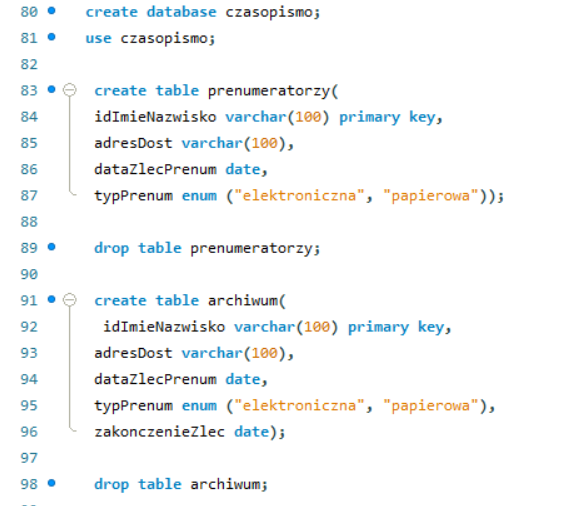


d)



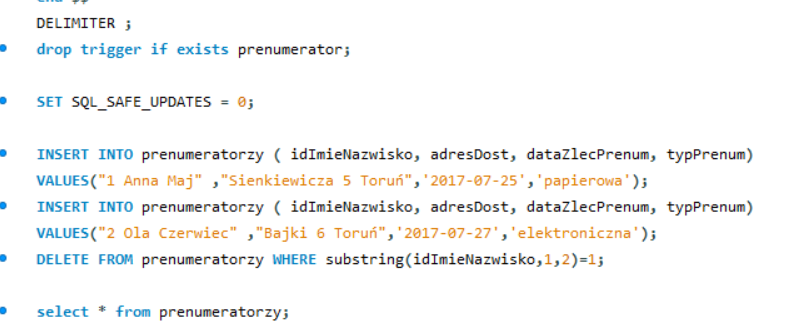
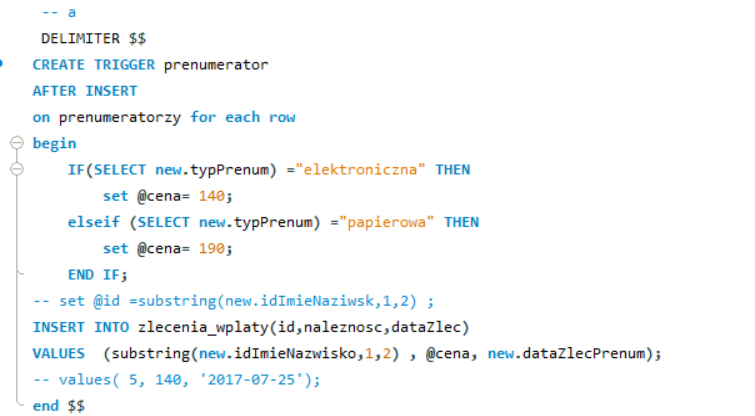


Zad.2

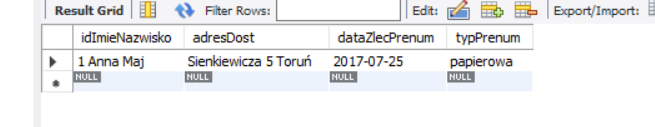


Zad.2

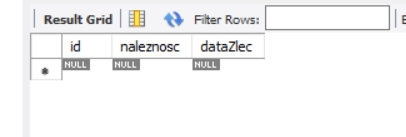
a)



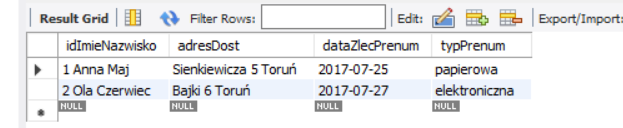
Prenumeratorzy przed:



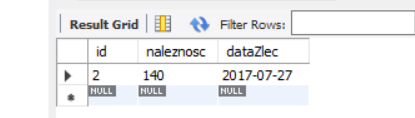
Zlecenia przed:



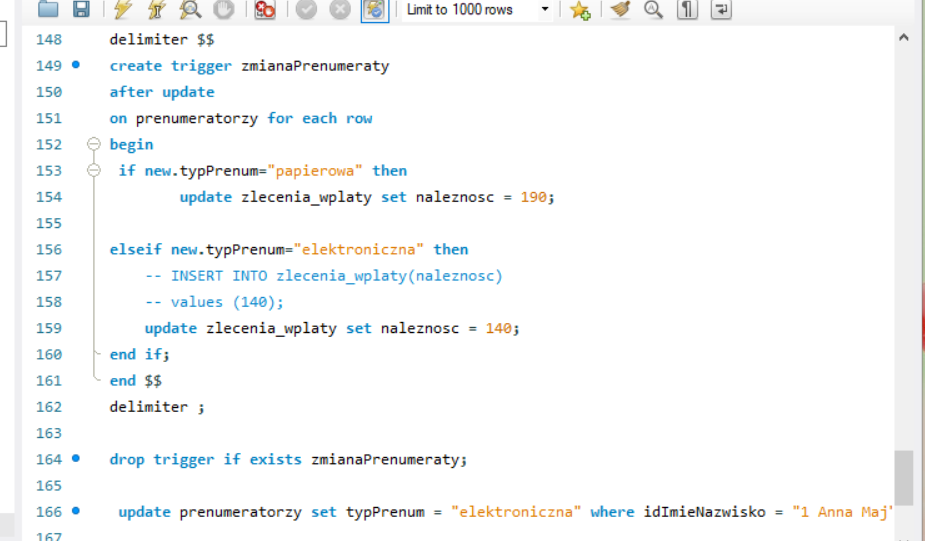
Prenumeratorzy po:



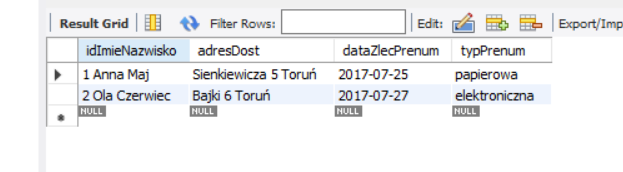
Zlecenia po:



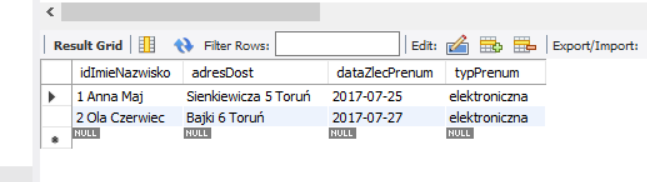
b)

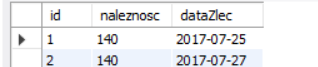


przed:

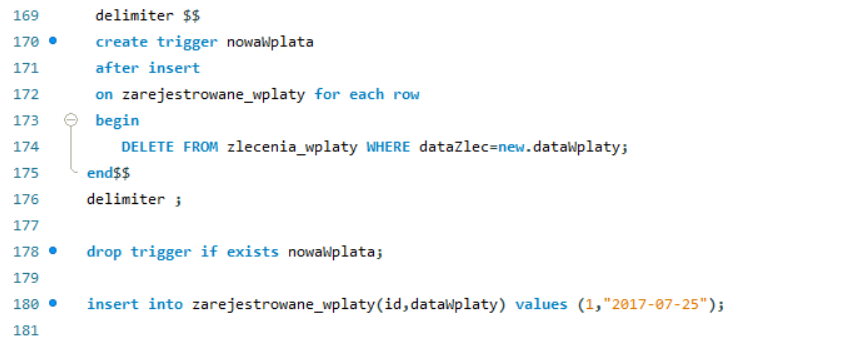


po:

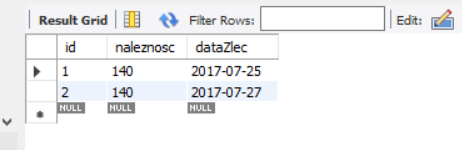


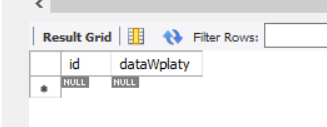


c)

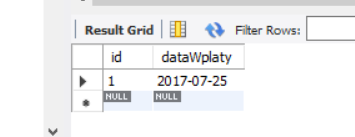


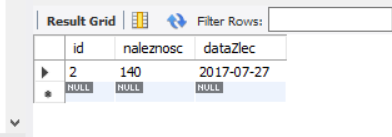
przed:



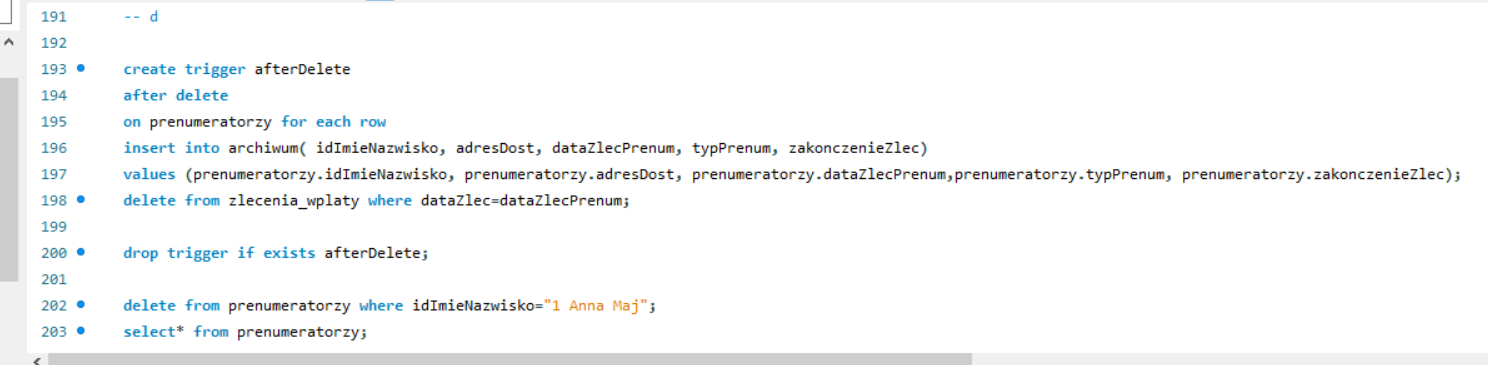


Po:

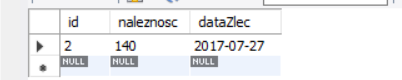


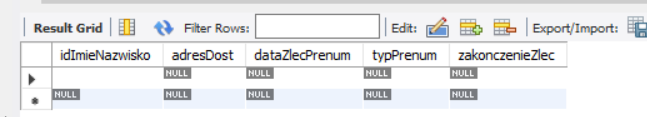


d)









Kod:

use klubydb;

-- 1

-- a

CREATE function bramkiWMiesiacu

(miesiac int) -- wejsciowe

returns int

deterministic

return ((select sum(gole\_gosp) from mecze where month(data)=miesiac)+

(select sum(gole\_gosc) from mecze where month(data)=miesiac)); -- wyjsciowe

drop function if exists bramkiWMiesiacu;

select bramkiWMiesiacu(1);

-- ------------------------------------------------------------------------

-- b

delimiter $$

CREATE function zarobekSedziego

(nazwisko VARCHAR(255))

returns int

deterministic

-- select id from sedziowie where substring(2,nazwisko)=nazwisko;

return (1150\*(select count(mecz\_ID) from mecze where sedzia\_id=

(select id from sedziowie where substring(2,nazwisko)=nazwisko))+

(RAND()\*(42000-38500)+38500)); -- wyjsciowe

end $$

delimiter ;

drop function if exists zarobekSedziego;

select zarobekSedziego("Dean");

-- ----------------------------------------------------------------------

-- c

DELIMITER $$

create procedure zarobek

(gosp varchar(255), gosc varchar(255), kwotaZakladu int, kurs int, wynik varchar(6), out z int)

BEGIN

DECLARE wynikTrue varchar(6) ;

set @gospID= (select id from kluby where nazwa=gosp);

set @goscID= (select id from kluby where nazwa=gosc);

set wynikTrue= concat((select gole\_gosc from mecze where gosc\_id=@goscID and gospodarz\_id=@gospid),

" : ", (select gole\_gosp from mecze where gosc\_id=@goscID and gospodarz\_id=@gospID));

if wynik=wynikTrue THEN

set z= kwotaZakladu\*kurs-kwotaZakladu;

ELSEIF wynik!= wynikTrue THEN

set z= 0;

end if;

END $$

DELIMITER ;

drop procedure if exists zarobek;

call zarobek ("Everton", "Chelsea", 20, 2, "2:1" ,@z);

select @z;

-- d ------------------------------------------------------------------------------------

DELIMITER $$

create procedure kartki

(gosp varchar(255), gosc varchar(255), kwotaZakladu int, kurs int, out k varchar(255))

BEGIN

DECLARE gospID varchar(6) ;

DECLARE goscID varchar(6) ;

DECLARE wynikGosp int ;

DECLARE wynikGosc int ;

set gospID= (select id from kluby where nazwa=gosp);

set goscID= (select id from kluby where nazwa=gosc);

set wynikGosp= ((select sum(zolte\_kartki\_gospodarz) from mecze where (gosc\_id=goscID and gospodarz\_id=gospid))\*10 +

(select sum(czerwone\_kartki\_gospodarz) from mecze where (gosc\_id=goscID and gospodarz\_id=gospid))\*25);

set wynikGosc=((select sum(zolte\_kartki\_gosc) from mecze where (gosc\_id=goscID and gospodarz\_id=gospid))\*10 +

(select sum(czerwone\_kartki\_gosc) from mecze where (gosc\_id=goscID and gospodarz\_id=gospid)\*25));

if wynikGosp=wynikGosc THEN

set k= "remis";

ELSEIF wynikGosp>wynikGosc THEN

set k= "gosp";

ELSEIF wynikGos<wynikGosc THEN

set k= "gosc";

end if;

END $$

DELIMITER ;

drop procedure if exists kartki;

call kartki ("Swansea", "Chelsea", 20, 2 , @komunikat);

select @komunikat;

-- -------------------------------------------------------------------------------------------------

create database czasopismo;

use czasopismo;

create table prenumeratorzy(

idImieNazwisko varchar(100) primary key,

adresDost varchar(100),

dataZlecPrenum date,

typPrenum enum ("elektroniczna", "papierowa"));

drop table prenumeratorzy;

create table archiwum(

idImieNazwisko varchar(100) primary key,

adresDost varchar(100),

dataZlecPrenum date,

typPrenum enum ("elektroniczna", "papierowa"),

zakonczenieZlec date);

drop table archiwum;

create table zlecenia\_wplaty(

id int auto\_increment primary key,

naleznosc int,

dataZlec date);

drop table zlecenia\_wplaty;

create table zarejestrowane\_wplaty(

id int primary key ,

dataWplaty date);

drop table zarejestrowane\_wplaty;

-- triggery

-- a

DELIMITER $$

CREATE TRIGGER prenumerator

AFTER INSERT

on prenumeratorzy for each row

begin

IF(SELECT new.typPrenum) ="elektroniczna" THEN

set @cena= 140;

elseif (SELECT new.typPrenum) ="papierowa" THEN

set @cena= 190;

END IF;

-- set @id =substring(new.idImieNaziwsk,1,2) ;

INSERT INTO zlecenia\_wplaty(id,naleznosc,dataZlec)

VALUES (substring(new.idImieNazwisko,1,2) , @cena, new.dataZlecPrenum);

-- values( 5, 140, '2017-07-25');

end $$

DELIMITER ;

drop trigger if exists prenumerator;

SET SQL\_SAFE\_UPDATES = 0;

INSERT INTO prenumeratorzy ( idImieNazwisko, adresDost, dataZlecPrenum, typPrenum)

VALUES("1 Anna Maj" ,"Sienkiewicza 5 Toruń",'2017-07-25','papierowa');

INSERT INTO prenumeratorzy ( idImieNazwisko, adresDost, dataZlecPrenum, typPrenum)

VALUES("2 Ola Czerwiec" ,"Bajki 6 Toruń",'2017-07-27','elektroniczna');

DELETE FROM prenumeratorzy WHERE substring(idImieNazwisko,1,2)=2;

select \* from prenumeratorzy;

select \* from zlecenia\_wplaty;

DELETE FROM zlecenia\_wplaty WHERE id=1;

DELETE FROM prenumeratorzy WHERE idImieNazwisko="1 Anna Maj";

DELETE FROM prenumeratorzy WHERE idImieNazwisko="2 Ola Czerwiec";

delimiter $$

create trigger zmianaPrenumeraty

after update

on prenumeratorzy for each row

begin

if new.typPrenum="papierowa" then

update zlecenia\_wplaty set naleznosc = 190;

elseif new.typPrenum="elektroniczna" then

-- INSERT INTO zlecenia\_wplaty(naleznosc)

-- values (140);

update zlecenia\_wplaty set naleznosc = 140;

end if;

end $$

delimiter ;

drop trigger if exists zmianaPrenumeraty;

update prenumeratorzy set typPrenum = "elektroniczna" where idImieNazwisko = "1 Anna Maj" ;

-- ------------------------------------------------

delimiter $$

create trigger nowaWplata

after insert

on zarejestrowane\_wplaty for each row

begin

DELETE FROM zlecenia\_wplaty WHERE dataZlec=new.dataWplaty;

end$$

delimiter ;

drop trigger if exists nowaWplata;

insert into zarejestrowane\_wplaty(id,dataWplaty) values (1,"2017-07-25");

insert into zarejestrowane\_wplaty(id,dataWplaty) values (2,"2017-07-27");

delete from zarejestrowane\_wplaty where id = 1;

select \* from zarejestrowane\_wplaty;

select \* from zlecenia\_wplaty;

-- d

create trigger afterDelete

after delete

on prenumeratorzy for each row

insert into archiwum( idImieNazwisko, adresDost, dataZlecPrenum, typPrenum, zakonczenieZlec)

values (prenumeratorzy.idImieNazwisko, prenumeratorzy.adresDost, prenumeratorzy.dataZlecPrenum,prenumeratorzy.typPrenum, prenumeratorzy.zakonczenieZlec);

delete from zlecenia\_wplaty where dataZlec=dataZlecPrenum;

drop trigger if exists afterDelete;

delete from prenumeratorzy where idImieNazwisko="1 Anna Maj";

select\* from prenumeratorzy;

select \* from zlecenia\_wplaty;

select \* from zarejestrowane\_wplaty;

select \* from archiwum;