

BAZY DANYCH

LISTA 8

WIOLETTA ŁUPKOWSKA, 244831

PON 15:15

1

a)

```
1 • use klubydb;
2   -- 1
3   -- a
4 • CREATE function bramkiWMiesiacu
5   (miesiac int) -- wejscowe
6   returns int
7   deterministic
8   return ((select sum(gole_gosp) from mecze where month(data)=miesiac)+
9   (select sum(gole_gosc) from mecze where month(data)=miesiac)); -- wyjscowe
10 • drop function if exists bramkiWMiesiacu;
11 • select bramkiWMiesiacu(1);
12   -- -----
13   -- b
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	bramkiWMiesiacu(1)			
▶	114			

```

13 -- b
14 delimiter $$
15 • CREATE function zarobekSedziego
16 (nazwisko VARCHAR(255))
17 returns int
18 deterministic
19 -- select id from sedziowie where substring(2,nazwisko)=nazwisko;
20 return (1150*(select count(mecz_ID) from mecze where sedzia_id=
21 (select id from sedziowie where substring(2,nazwisko)=nazwisko))+
22 (RAND()*(42000-38500)+38500)); -- wyjsciowe
23 end $$
24 delimiter ;
25
26 • drop function if exists zarobekSedziego;
27 • select zarobekSedziego("Dean");
28 -----

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
zarobekSedziego("Dean")			
▶ 39039			

b)

c)

```

29 -- c
30 DELIMITER $$
31 • create procedure zarobek
32 (gosp varchar(255), gosc varchar(255), kwotaZakladu int, kurs int, wynik varchar(6), out z int)
33 BEGIN
34 DECLARE wynikTrue varchar(6) ;
35 set @gospID= (select id from kluby where nazwa=gosp);
36 set @goscID= (select id from kluby where nazwa=gosc);
37 set wynikTrue= concat((select gole_gosc from mecze where gosc_id=@goscID and gospodarz_id=@gospid),
38 " : ", (select gole_gosp from mecze where gosc_id=@goscID and gospodarz_id=@gospID));
39 if wynik=wynikTrue THEN
40 set z= kwotaZakladu*kurs-kwotaZakladu;
41 ELSEIF wynik!= wynikTrue THEN
42 set z= 0;
43 end if;
44 END $$
45 DELIMITER ;
46 • drop procedure if exists zarobek;
47 • call zarobek ("Everton", "Chelsea", 20, 2, "2:1" ,@z);
48 • select @z;
49

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
zarobekSedziego("Dean")			
▶ 39039			

d)

```
-- 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));

66 if wynikGosp=wynikGosc THEN
67 set k= "remis";
68 ELSEIF wynikGosp>wynikGosc THEN
69 set k= "gosp";
70 ELSEIF wynikGosc<wynikGosc THEN
71 set k= "gosc";
72 end if;
73 END $$
74 DELIMITER ;
75 • drop procedure if exists kartki;
76 • call kartki ("Swansea", "Chelsea", 20, 2 , @komunikat);
77 • select @komunikat;
78
79 -----
80 • create database czasooismo;
```

Result Grid

@komunikat
gosp

Zad.2

```

80 • create database czasopismo;
81 • use czasopismo;
82
83 • create table prenumeratorzy(
84     idImieNazwisko varchar(100) primary key,
85     adresDost varchar(100),
86     dataZlecPrenum date,
87     typPrenum enum ("elektroniczna", "papierowa"));
88
89 • drop table prenumeratorzy;
90
91 • create table archiwum(
92     idImieNazwisko varchar(100) primary key,
93     adresDost varchar(100),
94     dataZlecPrenum date,
95     typPrenum enum ("elektroniczna", "papierowa"),
96     zakonczenieZlec date);
97
98 • drop table archiwum;
99
100 • create table zlecenia_wplaty(
101     id int primary key,
102     naleznosc int,
103     dataZlec date,
104     FOREIGN KEY (dataZlec) REFERENCES prenumeratorzy(dataZlec));
105
106 • drop table zlecenia_wplaty;
107
108 • create table zarejestrowane_wplaty(
109     id int primary key,
110     dataWplaty date);
111
112 • drop table zarejestrowane_wplaty;

```

Zad.2

a)

```
-- a
DELIMITER $$
CREATE TRIGGER prenumerator
AFTER INSERT
on prenumeratorzy for each row
begin
    IF(SELECT new.typPrenum) ="elektroniczna" THEN
        set @cena= 140;
    elseif (SELECT new.typPrenum) ="papierowa" THEN
        set @cena= 190;
    END IF;
    -- set @id =substring(new.idImieNazwisk,1,2) ;
    INSERT INTO zlecenia_wplaty(id,naleznosc,dataZlec)
    VALUES (substring(new.idImieNazwisko,1,2) , @cena, new.dataZlecPrenum);
    -- values( 5, 140, '2017-07-25');
end $$
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)=1;

select * from prenumeratorzy;
```

Prenumeratorzy przed:

	idImieNazwisko	adresDost	dataZlecPrenum	typPrenum
▶	1 Anna Maj	Sienkiewicza 5 Toruń	2017-07-25	papierowa
*	NULL	NULL	NULL	NULL

Zlecenia przed:

	id	naleznosc	dataZlec
*	NULL	NULL	NULL

Prenumeratorzy po:

Result Grid				
Filter Rows:				
	idImieNazwisko	adresDost	dataZlecPrenum	typPrenum
▶	1 Anna Maj	Sienkiewicza 5 Toruń	2017-07-25	papierowa
	2 Ola Czerwiec	Bajki 6 Toruń	2017-07-27	elektroniczna
•	NULL	NULL	NULL	NULL

Zlecenia po:

Result Grid			
Filter Rows:			
	id	naleznosc	dataZlec
▶	2	140	2017-07-27
•	NULL	NULL	NULL

b)

```

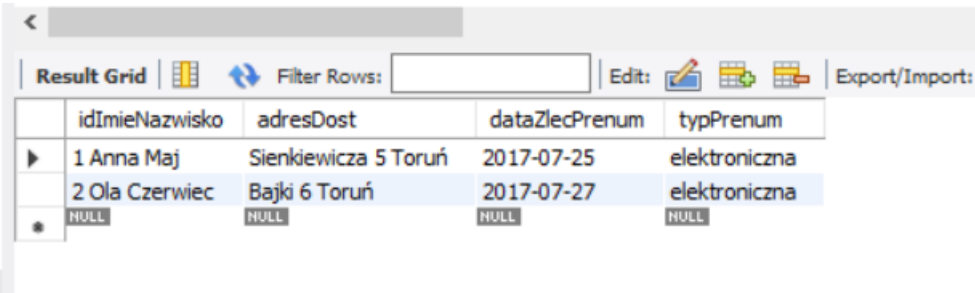
148 delimiter $$
149 • create trigger zmianaPrenumeraty
150 after update
151 on prenumeratorzy for each row
152 begin
153   if new.typPrenum="papierowa" then
154     update zlecenia_wplaty set naleznosc = 190;
155
156   elseif new.typPrenum="elektroniczna" then
157     -- INSERT INTO zlecenia_wplaty(naleznosc)
158     -- values (140);
159     update zlecenia_wplaty set naleznosc = 140;
160   end if;
161 end $$
162 delimiter ;
163
164 • drop trigger if exists zmianaPrenumeraty;
165
166 • update prenumeratorzy set typPrenum = "elektroniczna" where idImieNazwisko = "1 Anna Maj"
167

```

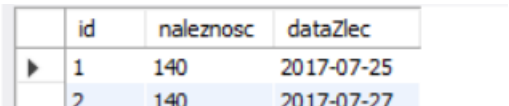
przed:

Result Grid				
Filter Rows:				
	idImieNazwisko	adresDost	dataZlecPrenum	typPrenum
▶	1 Anna Maj	Sienkiewicza 5 Toruń	2017-07-25	papierowa
	2 Ola Czerwiec	Bajki 6 Toruń	2017-07-27	elektroniczna
•	NULL	NULL	NULL	NULL

po:



	idImieNazwisko	adresDost	dataZlecPrenum	typPrenum
▶	1 Anna Maj	Sienkiewicza 5 Toruń	2017-07-25	elektroniczna
	2 Ola Czerwiec	Bajki 6 Toruń	2017-07-27	elektroniczna
*	NULL	NULL	NULL	NULL

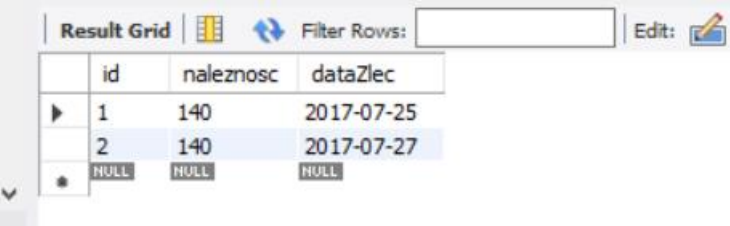


	id	naleznosc	dataZlec
▶	1	140	2017-07-25
	2	140	2017-07-27

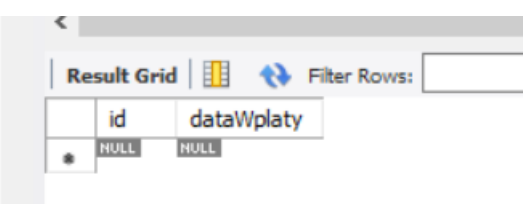
c)

```
169     delimiter $$
170 •   create trigger nowawplata
171     after insert
172     on zarejestrowane_wplaty for each row
173     begin
174         DELETE FROM zlecenia_wplaty WHERE dataZlec=new.dataWplaty;
175     end$$
176     delimiter ;
177
178 •   drop trigger if exists nowawplata;
179
180 •   insert into zarejestrowane_wplaty(id,dataWplaty) values (1,"2017-07-25");
181
```

przed:



	id	naleznosc	dataZlec
▶	1	140	2017-07-25
	2	140	2017-07-27
*	NULL	NULL	NULL



	id	dataWplaty
*	NULL	NULL

Po:

Result Grid		
	id	dataWplyty
▶	1	2017-07-25
✱	NULL	NULL

Result Grid			
	id	naleznosc	dataZlec
▶	2	140	2017-07-27
✱	NULL	NULL	NULL

d)

```

191 -- d
192
193 • create trigger afterDelete
194   after delete
195   on prenumeratory for each row
196   insert into archiwum( idImieNazwisko, adresDost, dataZlecPrenum, typPrenum, zakonczenieZlec)
197   values (prenumeratory.idImieNazwisko, prenumeratory.adresDost, prenumeratory.dataZlecPrenum,prenumeratory.typPrenum, prenumeratory.zakonczenieZlec);
198   delete from zlecenia_wplyty where dataZlec=dataZlecPrenum;
199
200 • drop trigger if exists afterDelete;
201
202 • delete from prenumeratory where idImieNazwisko="1 Anna Maj";
203 • select* from prenumeratory;

```

	idImieNazwisko	adresDost	dataZlecPrenum	typPrenum
▶	2 Ola Czerwiec	Bajki 6 Toruń	2017-07-27	elektroniczna

	id	naleznosc	dataZlec
▶	2	140	2017-07-27
✱	NULL	NULL	NULL

Result Grid					
	idImieNazwisko	adresDost	dataZlecPrenum	typPrenum	zakonczenieZlec
▶	NULL	NULL	NULL	NULL	NULL
✱	NULL	NULL	NULL	NULL	NULL

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=gosclD 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;
```

```
-- trigger
-- 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;
```