

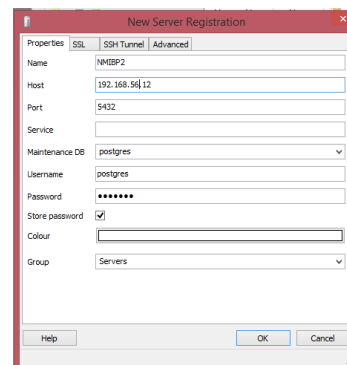
Zadaci za vježbu

(uz predavanje 2 – Napredni SQL)

Pokrenite virtualno računalo prema uputama objavljenima na stranici predmeta. Zadatke koji slijede možete obaviti na PostgreSQL sustavu instaliranom na preuzetom virtualnom računalu.

Za obavljanje SQL upita nad PostgreSQL bazom podataka možete koristiti neki od SQL editora prilagođenih za rad s PostgreSQL-om npr. PgAdmin čiju instalaciju možete preuzeti sa <http://www.pgadmin.org/download/>.

U *pgAdmin-u* definirajte postavke (kao na slici desno) za spajanje na PostgreSQL server instaliran na virtualnom računalu pomoću opcija: *File – Add Server*



Spojite se na PostgreSQL SUBP. Kreirajte novu bazu podataka (desni klik mišem na Databases – New Database; Name: *poželji*, Owner: *postgres*). Osvježite sadržaj podstabla Databases (desni klik mišem na Databases - Refresh), odaberite bazu s kojom želite raditi i otvorite prozor za izvođenje upita (Tools – Query Tool).

Podesite format prikaza datumskog tipa podataka na *dd.mm.yyyy* izvođenjem sljedeće naredbe:

```
SET DateStyle = 'German, DMY';
```

Sa stranica predmeta (Vježbe\napredniSQLStruct.sql) preuzmite SQL naredbe za kreiranje relacija (i punjenje relacija odgovarajućim sadržajem) potrebnih za ovu vježbu.

Prozori, particije, okviri i funkcije za rad s podacima u prozorima

Zadatak 1.

Za predmet i ispitni rok odrediti kumulativan broj ispita i kumulativan prosjek ocjena. Kumulativan broj ispita, odnosno prosjek ocjena, uključuje sve ispite s tog i prethodnih ispitnih rokova iz istog predmeta. (vidi slide „Motivacijski primjer 1“ iz prezentacije drugog predavanja tema Funkcije za rad s podacima u prozoru)

ispit

sifPred	sifStud	datumRok	ocjena
1	1	01.06.2013	1
1	3	01.06.2013	5
1	1	01.07.2013	4
...

Zadatak 2.

Odrediti ukupan broj bodova i rang studenta za svaki predmet. (vidi slide „Motivacijski primjer 2“ iz prezentacije drugog predavanja tema Funkcije za rad s podacima u prozoru).

studProvjera

sifPred	sifStud	kratProvjera	Bod
1	1	1DZ	5.00
1	2	1DZ	10.00
1	3	1DZ	5.00
...

Zadatak 3.

Odrediti ukupan broj bodova i rang studenta za svaki predmet. Dodatno, odrediti prosječan broj bodova koje studenti osvoje na predmetu.

Za zadatke 4-6 se koriste tablice *orgJed*, *djelatnik*, *djelatnikIsplata* i *projekt* čija je shema i sadržaj definirana naredbama u datoteci *napredniSQLStruct.sql*. Shema i dio sadržaja relacija prikazana je na sljedećoj slici.

orgJed					djellsplata						
sifOrgJed		nazOrgJed				sifDjel		datumIsplata		iznos	
1		Maloprodaja Radnička				1		01.01.2013		4700.00	
2		Maloprodaja Vukovarska				1		01.02.2013		4700.00	
3		Maloprodaja Ilica				1		01.03.2013		4700.00	
...		
djelatnik					projekt						
sifDjel	imeDjel	prezDjel	sifOrgJed	sifNadrDjel	sifProjekt	sifNadlDjel	datumPoc	datumKraj			
1	Ana	Par	1		234	1	01.3.2013	20.7.2013			
2	Šime	Knez	1	1	764	2	05.4.2013	15.8.2013			
3	Petra	Pilić	1	1							
...					

Zadatak 4.

Za svaku isplatu zaposleniku izračunati ukupan iznos isplaćen zajedno sa 2 isplate koje su joj prethodile. Podatke ispisati u sljedećem obliku:

prezDjel	imeDjel	datumIsplata	iznos	uklZnos
Par	Ana	01.01.2013	4700.00	4700.00
Par	Ana	01.02.2013	4700.00	9400.00
Par	Ana	01.03.2013	4700.00	14100.00
Par	Ana	01.04.2013	4700.00	14100.00
Par	Ana	01.05.2013	4700.00	14100.00
...

Zadatak 5.

Za svaku organizacijsku jedinicu ispisati podatke o najbolje i najslabije plaćenom zaposleniku u sljedećem obliku:

nazOrgJed	prezDjel	imeDjel	ukZarada	komentar
Maloprodaja Ilica	Milić	Petar	26100.00	najslabije plaćen
Maloprodaja Ilica	Burić	Marija	96900.00	najbolje plaćen
...

Zadatak 6.

Za svakog zaposlenika odrediti koliko puta mu se promijenila plaća. Podatke ispisati u sljedećem obliku:

prezDjel	imeDjel	brPromjena
Milić	Petar	0
Knez	Šime	3
Jelić	Marko	0
Par	Ana	2
...

Zadatak 7.

Za svakog zaposlenika odrediti ukupan isplaćen iznos te postotak od iznosa najbolje plaćenog zaposlenika u organizacijskoj jedinici, te u cijeloj tvrtki. Podatke ispisati u sljedećem obliku:

nazOrgJed	prezDjel	imeDjel	ukZarada	posOdMaksOJ	posOdMaksT
Maloprodaja Radnička	Par	Ana	43000.00	100.00	44.37
Maloprodaja Radnička	Knez	Šime	40850.00	95.00	42.15
Maloprodaja Radnička	Pili	Petra	32400.00	75.34	33.43
Maloprodaja Vukovarska	Kralj	Anita	78900.00	100.00	81.42
Maloprodaja Vukovarska	Jelić	Marko	67500.00	85.55	69.65
...

Rekurzivni upiti

Zadatak 8.

Pronađi sve osobe s kojima je osoba **ana123@hotmail.com** povezana direktno ili preko prijatelja. U obzir uzeti prijateljstva u kojima je **ana123@hotmail.com** (odnosno njeni prijatelji, prijatelji prijatelja i td.) inicijator (foaf.osoba1 = 'ana123@hotmail.com') ali i ona u kojima je netko drugi inicirao prijateljstvo (foaf.osoba2 = 'ana123@hotmail.com' i td.).

foaf

osoba1	osoba2
ana123@hotmail.com	zecG@gmail.com
zecG@gmail.com	iva.malic@fer.hr
klaraB@gmail.com	ana123@hotmail.com
zecG@gmail.com	jezV@hotmail.com
iva.malic@fer.hr	jezV@hotmail.com
klaraB@gmail.com	mia95@yahoo.com
jon.doe@info.hr	klaraB@gmail.com
cat56@hotmail.com	jon.doe@info.hr
...	...

Zadatak 9.

Za svaki mjesec u kojem je projekt sa šifrom 746 bio aktivan, odredi ukupan iznos primanja koja su isplaćena zaposlenima na projektu za vrijeme trajanja projekta.

Pivotiranje

Zadatak 10.

Za svaku godinu ispisati po mjesecima ukupno isplaćene iznose u sljedećem obliku:

godina	siječanj	veljača	ožujak	travanj	svibanj	lipanj	srpanj	kolovoz	...
2013	47840.00	47840.00	48070.00	48840.00	48890.00	48890.00
...

Zadatak 11.

Za organizacijske jedinice Maloprodaja Radnička, Maloprodaja Vukovarska, Maloprodaja Ilica ispisati ukupno isplaćene iznose u sljedećem obliku:

	Maloprodaja Radnička	Maloprodaja Vukovarska	Maloprodaja Ilica
2013	177420.00	116250.00	146400.00
...

RJEŠENJA:

Prozori, particije, okviri i funkcije za rad s podacima u prozorima

Zadatak 1.

```
SET DateStyle = 'German, DMY';
```

```
CREATE TABLE ispit (  
    sifPred INT NOT NULL,  
    sifStud INT NOT NULL,  
    datumRok DATE NOT NULL,  
    ocjena SMALLINT NOT NULL);
```

```
INSERT INTO ispit VALUES (1, 1, '01.06.2013', 1);  
INSERT INTO ispit VALUES (1, 3, '01.06.2013', 5);  
INSERT INTO ispit VALUES (1, 1, '01.07.2013', 4);  
INSERT INTO ispit VALUES (1, 2, '01.07.2013', 5);  
INSERT INTO ispit VALUES (1, 4, '01.07.2013', 3);  
INSERT INTO ispit VALUES (1, 5, '01.07.2013', 2);  
INSERT INTO ispit VALUES (2, 1, '01.07.2013', 3);  
INSERT INTO ispit VALUES (2, 2, '01.07.2013', 2);
```

```
SELECT DISTINCT sifPred, datumRok  
    , COUNT(*) OVER (PARTITION BY sifPred ORDER BY sifPred, datumRok)  
    kumBrIspit  
    , AVG(ocjena) OVER (PARTITION BY sifPred ORDER BY sifPred, datumRok)  
    kumPrOcj  
    FROM ispit  
ORDER BY sifPred, datumRok
```

Zadatak 2.

```
CREATE TABLE studProvjera (  
    sifPred int NOT NULL,  
    sifStud int NOT NULL,  
    kratProvjera CHAR(5) NOT NULL,  
    bod numeric (5,2) NOT NULL);--
```

```
INSERT INTO studprovjera VALUES (1, 1, '1DZ', 5.00);  
INSERT INTO studprovjera VALUES (1, 2, '1DZ', 10.00);  
INSERT INTO studprovjera VALUES (1, 3, '1DZ', 5.00);  
INSERT INTO studprovjera VALUES (1, 1, 'MI', 10.00);  
INSERT INTO studprovjera VALUES (1, 2, 'MI', 5.00);  
INSERT INTO studprovjera VALUES (1, 3, 'MI', 20.00);  
INSERT INTO studprovjera VALUES (2, 1, 'MI', 20.00);  
INSERT INTO studprovjera VALUES (2, 2, 'MI', 25.00);  
INSERT INTO studprovjera VALUES (2, 3, 'MI', 30.00);
```

```
SELECT sifPred  
    , sifStud  
    , SUM(bod) ukBod  
    , rank () OVER (PARTITION BY sifPred ORDER BY SUM(bod) DESC) AS rang  
    FROM studProvjera  
GROUP BY sifPred, sifStud  
ORDER BY sifPred, rang
```

Zadatak 3.

CREATE TABLE i INSERT naredbe su jednake kao u rješenju prethodnog zadatka.

```
SELECT sifPred
      , sifStud
      , SUM(bod) ukBod
      , rank () OVER (partition by sifPred ORDER BY SUM(bod) DESC) AS rang
      , AVG(SUM(bod)) OVER (partition by sifPred) AS prosUkBod
FROM studProvjera
GROUP BY sifPred, sifStud
ORDER BY sifPred, rang
```

Zadatak 4.

```
select prezDjel, imeDjel, djelIsplata.datumIsplata, iznos
      , SUM(iznos) OVER (PARTITION BY djelIsplata.sifDjel
                        ORDER BY datumIsplata
                        ROWS 3 PRECEDING) ukIznos
from djelIsplata, djelatnik
where djelIsplata.sifDjel = djelatnik.sifDjel
```

Zadatak 5.

Za ispis komentara 'najbolje plaćen' odnosno 'najslabije plaćen' treba nam uvjetna naredba CASE. Ovo je sintaksa CASE naredbe u PostgreSQL:

```
CASE WHEN condition THEN result
      [WHEN ...]
      [ELSE result]
END
```

Rang zaposlenika u organizacijskoj jedinici pri čemu je najviše rangiran zaposlenik s najvećim primanjima odredi se pomoću izraza:

```
rank () OVER (PARTITION BY nazOrgJed
              ORDER BY SUM(iznos) DESC) najVise
```

Sličnim izrazom se odredi rang zaposlenika pri čemu je najviše rangiran zaposlenik s najmanjim primanjima:

```
rank () OVER (PARTITION BY nazOrgJed
              ORDER BY SUM(iznos)) najManje
```

Primijetiti da su pomoću PARTITION BY nazOrgJed definirane particije tako da se u istoj particiji nalaze n-torke koje pripadaju istoj organizacijskoj jedinici.

Pomoću sljedećeg upita se dobiju podaci o ukupnom iznosu zarade te rangovima zaposlenika unutar svojih organizacijskih jedinica:

```
SELECT nazOrgJed, djelatnik.sifDjel
      , prezDjel, imeDjel, SUM(iznos) ukZarada
      , rank () OVER (PARTITION BY nazOrgJed
                      ORDER BY SUM(iznos) DESC) maksZ
      , rank () OVER (PARTITION BY nazOrgJed
                      ORDER BY SUM(iznos)) minZ
FROM djelIsplata, djelatnik, orgjed
WHERE djelIsplata.sifDjel = djelatnik.sifDjel
AND djelatnik.sifOrgJed = orgjed.sifOrgJed
GROUP BY djelatnik.sifOrgJed, orgjed.nazOrgjed
      , djelatnik.sifDjel, prezDjel, imeDjel
```

Ovo je njegov rezultat:

nazOrgJed	sifDjel	prezDjel	imeDjel	ukZarada	maksZ	minZ
Maloprodaja Ilica	8	Milić	Petar	26100.00	3	1
Maloprodaja Ilica	7	Rendić	Slavica	54420.00	2	2
Maloprodaja Ilica	6	Burić	Marija	96900.00	1	3
Maloprodaja Radnička	3	Pilić	Petra	32400.00	3	1
Maloprodaja Radnička	1	Par	Ana	42900.00	2	2
Maloprodaja Radnička	2	Knez	Šime	43620.00	1	3
Maloprodaja Vukovarska	5	Jelić	Marko	67500.00	2	1
Maloprodaja Vukovarska	4	Kralj	Anita	78900.00	1	2

Gornji upit je imenovan sa `rangIsplata` i ugniježđen u vanjski upit:

```
SELECT rangIsplata.*
, CASE
    WHEN rangIsplata.najVise = 1 AND
        rangIsplata.najManje > 1 THEN 'najbolje plaćen'
    WHEN rangIsplata.najVise > 1 AND
        rangIsplata.najManje = 1 THEN 'najslabije plaćen'
END
FROM (
    SELECT nazOrgJed, djelatnik.sifDjel
    , prezDjel, imeDjel, SUM(iznos) ukZarada
    , rank () OVER (PARTITION BY nazOrgJed
        ORDER BY SUM(iznos) DESC) najVise
    , rank () OVER (PARTITION BY nazOrgJed
        ORDER BY SUM(iznos)) najManje
    FROM djelIsplata, djelatnik, orgjed
    WHERE djelIsplata.sifDjel = djelatnik.sifDjel
    AND djelatnik.sifOrgJed = orgjed.sifOrgJed
    GROUP BY djelatnik.sifOrgJed, orgjed.nazOrgjed
    , djelatnik.sifDjel, prezDjel, imeDjel
    ) rangIsplata
WHERE rangIsplata.najVise = 1 OR rangIsplata.najManje = 1
```

Rezultat:

nazOrgJed	prezDjel	imeDjel	ukZarada	komentar
Maloprodaja Ilica	Milić	Petar	26100.00	najslabije plaćen
Maloprodaja Ilica	Burić	Marija	96900.00	najbolje plaćen
Maloprodaja Radnička	Pilić	Petra	32400.00	najslabije plaćen
Maloprodaja Radnička	Knez	Šime	43620.00	najbolje plaćen
Maloprodaja Vukovarska	Jelić	Marko	67500.00	najslabije plaćen
Maloprodaja Vukovarska	Kralj	Anita	78900.00	najbolje plaćen

Što je potrebno promijeniti u gornjem upitu pa da se dobiju podatci o 2 najbolje odnosno 2 najlošije plaćena zaposlenika u svakoj organizacijskoj jedinici?

Zadatak 6.

Potrebno je stvoriti particiju za svakog djelatnika (`PARTITION BY djelIsplata.sifDjel`), a zapise unutar particije poredati prema datumu isplate (`ORDER BY datumIsplata`). Isplaćeni iznos za svaku n-torku uspoređujemo s prethodno isplaćenim iznosom `lag(iznos)` – po defaultu offset je 1 pa lag dohvaća n-torku koja tekućoj neposredno prethodi. Svaku promjenu isplaćenog iznosa označit ćemo s 1 a nepromijenjen iznos vrijednošću 0.

```

SELECT djelatnik.sifDjel, prezDjel, imeDjel
      , iznos, datumIsplata
      , CASE
            WHEN lag(iznos) OVER (PARTITION BY djelIsplata.sifDjel
                                ORDER BY datumIsplata) != iznos THEN 1
            ELSE 0
          END jePromjena
FROM djelIsplata, djelatnik
WHERE djelIsplata.sifDjel = djelatnik.sifDjel

```

Dio rezultata gornjeg upita:

sifDjel	prezDjel	imeDjel	iznos	datumIsplata	jePromjena
1	Par	Ana	4700.00	01.01.2013	0
1	Par	Ana	4700.00	01.02.2013	0
1	Par	Ana	4700.00	01.03.2013	0
1	Par	Ana	4700.00	01.04.2013	0
1	Par	Ana	4750.00	01.05.2013	1
1	Par	Ana	4750.00	01.06.2013	0
1	Par	Ana	4900.00	01.07.2013	1
1	Par	Ana	4900.00	01.08.2013	0
1	Par	Ana	4900.00	01.09.2013	0
...

Gornji upit je imenovan sa `promjenaPlaca` i ugniježđen u vanjski upit. Broj promjena plaće zaposlenika određen je sumiranjem stupca `jePromjena`.

```

SELECT prezDjel, imeDjel, SUM(jePromjena) brPromjena
FROM
  (SELECT djelatnik.sifDjel, prezDjel, imeDjel
    , CASE
          WHEN lag(iznos) OVER (PARTITION BY djelIsplata.sifDjel
                              ORDER BY datumIsplata) != iznos THEN 1
          ELSE 0
        END jePromjena
    FROM djelIsplata, djelatnik
    WHERE djelIsplata.sifDjel = djelatnik.sifDjel) AS promjenaPlaca
GROUP BY promjenaPlaca.sifDjel, promjenaPlaca.prezDjel,
promjenaPlaca.imeDjel

```

Zadatak 7.

```

SELECT nazOrgJed
      , prezDjel, imeDjel, SUM(iznos) ukZarada
      , 100* SUM(iznos)
        /nth_value (SUM(iznos),1) OVER (PARTITION BY nazOrgJed
                                       ORDER BY SUM(iznos) DESC ) posOdMaksZ
      , 100* SUM(iznos)
        /nth_value (SUM(iznos),1) OVER (ORDER BY SUM(iznos) DESC) posOdMaksT
FROM djelIsplata, djelatnik, orgjed
WHERE djelIsplata.sifDjel = djelatnik.sifDjel
AND djelatnik.sifOrgJed = orgjed.sifOrgJed
GROUP BY djelatnik.sifOrgJed, orgjed.nazOrgjed
      , djelatnik.sifDjel, prezDjel, imeDjel
ORDER BY djelatnik.sifOrgJed, orgjed.nazOrgjed, SUM(iznos) DESC

```


Rekurzivni upiti

Zadatak 8.

```
create table foaf (
    osoba1 char(50) not null,
    osoba2 char(50) not null);

INSERT INTO foaf VALUES ('anal23@hotmail.com', 'zecG@gmail.com');
INSERT INTO foaf VALUES ('zecG@gmail.com', 'iva.malic@fer.hr');
INSERT INTO foaf VALUES ('klaraB@gmail.com', 'anal23@hotmail.com');
INSERT INTO foaf VALUES ('zecG@gmail.com', 'jezV@hotmail.com');
INSERT INTO foaf VALUES ('iva.malic@fer.hr', 'jezV@hotmail.com');
INSERT INTO foaf VALUES ('klaraB@gmail.com', 'mia95@yahoo.com');
INSERT INTO foaf VALUES ('jon.doe@info.hr', 'klaraB@gmail.com');
INSERT INTO foaf VALUES ('cat56@hotmail.com', 'jon.doe@info.hr');

WITH RECURSIVE prijatelji (prijatelj1, prijatelj2) AS
    ( (SELECT osoba1 as prijatelj1, osoba2 as prijatelj2 FROM foaf)
      UNION
      (SELECT prijatelji.prijatelj1, foaf.osoba2 as prijatelj2
       FROM prijatelji, foaf
       WHERE prijatelji.prijatelj2 = foaf.osoba1)
    )
SELECT * FROM prijatelji
WHERE prijatelj1 = 'anal23@hotmail.com'
   OR prijatelj2 = 'anal23@hotmail.com'
```

Zadatak 9.

Pomoću CTE ćemo u privremenu relaciju `djelatnici` pohraniti podatke o djelatnicima i nadređenim djelatnicima rekurzivno.

```
WITH RECURSIVE djelatnici (sifDjel, sifNadrDjel) AS
    ( (SELECT sifDjel, sifNadrDjel FROM djelatnik)
      UNION
      (SELECT djelatnici.sifDjel, djelatnik.sifNadrDjel
       FROM djelatnici, djelatnik
       WHERE djelatnici.sifNadrDjel = djelatnik.sifDjel)
    )
SELECT * FROM djelatnici
WHERE sifDjel = 2 OR sifNadrDjel = 2
```

Izvođenjem gornjeg upita se dobije sljedeći rezultat:

sifDjel	sifNadrDjel
2	1
4	2
5	2
2	
6	2
7	2
8	2

Djelatnik nadležan za projekt (`projekt.sifNadrDjel`) sa šifrom 746 treba biti jednak nadređenom djelatniku (`djelatnici.sifNadrDjel`) u n-torkama prikazanima u prethodnoj tablici.

Budući da za djelatnika sa šifrom 2 ne postoji n-torka sa `sifNadrDjel = 2`, a želimo i njegov dohodak uračunati u ukupna primanja zaposlenih na projektu, riješit ćemo ovaj detalj sljedećim uvjetom:

```
(projekt.sifNadlDjel = djelatnici.sifnadrDjel OR
 projekt.sifNadlDjel = djelatnici.sifDjel AND
 djelatnici.sifNadrDjel IS NOT NULL)
```

Kompletno rješenje:

```
WITH RECURSIVE djelatnici (sifDjel, sifNadrDjel) AS
    ( (SELECT sifDjel, sifNadrDjel FROM djelatnik)
      UNION
      (SELECT djelatnici.sifDjel, djelatnik.sifNadrDjel
       FROM djelatnici, djelatnik
       WHERE djelatnici.sifNadrDjel = djelatnik.sifDjel)
    )
SELECT EXTRACT (MONTH FROM datumIsplata) AS mjesec, SUM(iznos) AS ukIznos
FROM projekt, djelatnici, djelIsplata
WHERE (projekt.sifNadlDjel = djelatnici.sifnadrDjel OR
       projekt.sifNadlDjel = djelatnici.sifDjel AND
       djelatnici.sifNadrDjel IS NOT NULL)
       AND djelIsplata.sifDjel = djelatnici.sifDjel
       AND djelIsplata.datumIsplata BETWEEN projekt.datumPoc AND
projekt.datumKraj
       AND projekt.sifProjekt = 746
GROUP BY mjesec --EXTRACT (MONTH FROM datumIsplata)
ORDER BY mjesec --EXTRACT (MONTH FROM datumIsplata)
```

Pivotiranje

Zadatak 10.

```
CREATE EXTENSION tablefunc;
CREATE TEMP TABLE mjesec
    (rbrMjesec int);
INSERT INTO mjesec VALUES ( 2);
INSERT INTO mjesec VALUES ( 3);
INSERT INTO mjesec VALUES ( 4);
INSERT INTO mjesec VALUES ( 1);
INSERT INTO mjesec VALUES ( 5);
INSERT INTO mjesec VALUES ( 6);
INSERT INTO mjesec VALUES ( 7);
INSERT INTO mjesec VALUES ( 8);
INSERT INTO mjesec VALUES ( 9);
INSERT INTO mjesec VALUES (10);
INSERT INTO mjesec VALUES (11);
INSERT INTO mjesec VALUES (12);
SELECT *
FROM crosstab(
    'SELECT CAST(EXTRACT(YEAR FROM datumIsplata) AS int) AS godina
      , CAST(EXTRACT(MONTH FROM datumIsplata) AS int) AS mjesec
      , CAST(SUM(iznos) AS NUMERIC (10,2))
      FROM djelIsplata
      GROUP BY godina, mjesec
      ORDER BY godina, mjesec'
    , 'SELECT rbrMjesec FROM mjesec ORDER BY rbrMjesec')
AS pivotTable (godina INT, siječanj NUMERIC(10,2), veljača NUMERIC(10,2)
    , ožujak NUMERIC(10,2), travanj NUMERIC(10,2), svibanj NUMERIC(10,2)
    , lipanj NUMERIC(10,2), srpanj NUMERIC(10,2), kolovoz NUMERIC(10,2)
    , rujan NUMERIC(10,2), listopad NUMERIC(10,2), studeni NUMERIC(10,2)
    , prosinac NUMERIC(10,2))
ORDER BY godina
```

Zadatak 11.

```
SELECT *
FROM crosstab(
    'SELECT CAST(EXTRACT(YEAR FROM datumIsplata) AS int) AS godina
        , orgJed.nazOrgJed
        , CAST(SUM(iznos) AS NUMERIC (10,2))
    FROM djelIsplata, djelatnik, orgjed
    WHERE djelIsplata.sifDjel = djelatnik.sifDjel
        AND djelatnik.sifOrgJed = orgJed.sifOrgJed
        AND orgJed.nazOrgJed IN ('Maloprodaja Ilica'
                                , 'Maloprodaja Radnička', 'Maloprodaja Vukovarska')
    GROUP BY godina, orgJed.nazOrgJed
    ORDER BY godina, orgJed.nazOrgJed')
AS pivotTable (godina INT, MaloprodajaIlica NUMERIC(10,2)
               , MaloprodajaRadnička NUMERIC(10,2)
               , MaloprodajaVukovarska NUMERIC(10,2))
ORDER BY godina
```