

7. rujna 2018. **(50 bodova)**

Baza podataka **rezervacijaSmjestaj** sadrži podatke o smještaju (**smjestaj**) koji korisnici (**korisnik**) mogu rezervirati (**rezervacija.sifGost**). Domaćin (**smjestaj.sifDomacin**) su također korisnici usluge koji osim što nude mogu rezervirati smještaj. Domaćin za smještaj evidentira cijenu po noćenju (**smjestaj.cijena**) i pogodnosti koje smještaj nudi (**smjestajPogodnost**), odabirom postojećih pogodnosti iz tablice **pogodnost**. Za smještaj se evidentira prosječna ocjena smještaja (**smjestaj.prosOcj**) koja se računa na temelju svih dosadašnjih ocjena koju su smještaju dodijelili gosti (**rezervacija.ocjena**).

Zadaci **1-7** se odnose na bazu podataka **rezervacijaSmjestaj** prikazanu na **slici 1**. U zadacima **1-7** se podrazumijeva korištenje SUBP-a PostgreSQL. Na slici **nisu prikazane** sve n-torke sadržane u relacijama. Relacije mogu sadržavati attribute koji nisu prikazani jer nisu potrebni za rješavanje zadataka na ispitu.

Slika 1.

```

    graph TD
      subgraph "pogodnost"
        sifPog[sifPog]
        nazivPog[nazivPog]
      end
      subgraph "korisnik"
        sifKorisnik[sifKorisnik]
        korisnickolme[korisnickolme]
        datReg[datReg]
      end
      subgraph "smjestaj"
        sifSmjestaj[sifSmjestaj]
        nazivSmjestaj[nazivSmjestaj]
      end
      subgraph "rezervacija"
        sifGost[sifGost]
        sifSmjestaj2[sifSmjestaj]
        datDolazak[datDolazak]
        datOdlazak[datOdlazak]
        ocjena[ocjena]
      end
      sifPog --- sifKorisnik
      sifPog --- sifSmjestaj
      sifGost --- sifSmjestaj2
  
```

The diagram illustrates the database schema for a hotel management system. It consists of four tables:

- pogodnost**: Contains attributes sifPog (primary key) and nazivPog.
- korisnik**: Contains attributes sifKorisnik (primary key), korisnickolme, and datReg.
- smjestaj**: Contains attributes sifSmjestaj (primary key) and nazivSmjestaj.
- rezervacija**: Contains attributes sifGost (primary key), sifSmjestaj (foreign key), datDolazak, datOdlazak, and ocjena.

Relationships are indicated by lines connecting the primary key of one table to the foreign key of another table: sifPog to sifKorisnik, sifPog to sifSmjestaj, and sifGost to sifSmjestaj.

U zadacima 1-2 napisati po jednu SQL naredbu kojom će se obaviti sljedeće:

- (5 bodova)** Korisnik želi rezervirati smještaj koji je raspoloživ od 10.09.2018. do 22.09.2018. te koji ima prosječnu ocjenu veću od 4 i „Wifi“ kao pogodnost. Napisati SQL upit koji će vratiti tražene rezultate o smještaju, gdje ispis sadrži šifru smještaja, prosječnu ocjenu, ukupan broj rezervacija te podatak o raspoloživosti smještaja u navedenom terminu (u obliku „da“, „ne“). Za određivanje raspoloživosti koristiti SQL operator (`start1, end1`) `OVERLAPS` (`start2, end2`), koji se evaluira kao `true`, ako se intervali definirani krajnjim datumima preklapaju, inače kao `false`. Rezultate sortirati uzlazno po prosječnoj ocjeni te zatim silazno po broju rezervacija.
- (5 bodova)** Za potrebe poboljšanja kvalitete usluge potrebno je napisati SQL upit koji će ispisati smještaj (jedan ili više ako ih ima) s najmanjim ukupnim brojem dana rezervacija u odnosu na sve ostale smještaje. Smještaji s nijednom rezervacijom nisu uključeni u ispis. U rezultatu je potrebno prikazati šifru smještaja, korisničko ime domaćina, kalendarsku godinu njegove registracije, ukupan broj dana rezervacija te broj pogodnosti. Za smještaj s nijednom pogodnosti prikazati 0 (nulu).
- (5 bodova)** Bazu podataka je nužno periodički „održavati“. Potrebno je obrisati sve neaktivne korisnike, to su korisnici koji nikad nisu rezervirali ili ponudili smještaj, a registrirali su se prije više od godinu dana. Sve ocjene postavljene u rezervacijama u kojima su korisnici rezervirali sami svoj smještaj potrebno je postaviti na nepoznatu vrijednost (NULL). Prema navedenim zahtjevima napisati odgovarajuće SQL naredbe.
- (7 bodova)** Pretpostavite da su u bazi podataka **rezervacijaSmjestaj** kreirane sve relacije te da su definirani svi primarni i strani ključevi. Napisati SQL naredbe kojima će se:
 - osigurati ispravnost atributa **rezervacija.ocjena**, **rezervacija.datDolazak** i **rezervacija.datOdlazak**. Vrijednost atributa **ocjena** je ispravna, ako ocjena nije postavljena (NULL) ili je u intervalu [1, 5] (uključivo). Atributi **datDolazak** i **datOdlazak** su ispravni, ako je datum odlaska veći od datuma dolaska.
 - kreirati **minimalan** broj objekata koji, pri unosu i izmjeni n-torki relacije **rezervacija**, osiguravaju konzistentnost atributa **smjestaj.prosOcj** i **rezervacija.datDolazak**.
Atribut **prosOcj** izračunava se za smještaj prilikom svakog dodjeljivanja ocjene u rezervaciji. Prosječna ocjena se računa na temelju svih do tada dodijeljenih ocjena.
Spriječiti promjenu i unos zapisa u relaciju **rezervacija** ako korisnik usluge želi rezervirati smještaj više od 3 mjeseca unaprijed. Korisniku dojaviti poruku „Rezervaciju je moguće obaviti najviše 3 mjeseca unaprijed!“.
 - napišite SQL naredbu za kreiranje tablice **rezervacija**
- (6 bodova)** Svi objekti sa slike 1 su kreirani u shemi *public*. Administrator sustava treba napisati niz SQL naredbi kojima će se:
 - kreirati korisnika `dtrotter` s mogućnošću uspostavljanja SQL sjednice uz autentifikaciju pomoću lozinke. Korisniku za inicijalnu lozinku postaviti „plonker“.
 - stvoriti ulogu `gost` koja korisnicima s navedenom ulogom omogućava brisanje i pregledavanje svih podataka vlastitih rezervacija te promjenu vrijednosti atributa `ocjena` u vlastitim rezervacijama.
 - ovlasti iz b) dodijeliti korisniku `dtrotter`.

6. (4 boda) Za sve relacije u bazi *smjestajRezervacija* su definirani primarni ključevi (označeni na slici 1). Obavlja se sljedeći upit, pri čemu optimizator upita raspolaže navedenim statističkim informacijama:

```
SELECT *
FROM smjestaj
    NATURAL JOIN smjestajPogodnost
    NATURAL JOIN rezervacija
WHERE sifPog = 4 AND
    prosOcj > 3.0
```

N(rezervacija) = 2000
V(sifSmjestaj, rezervacija) = 1000
N(smjestajPogodnost) = 10000
V(sifPog, smjestajPogodnost) = 50
V(sifSmjestaj, smjestajPogodnost) = 2500
N(smjestaj) = 3000
V(prosOcj, smjestaj) = 1000

Nacrtati moguća stabla izvođenja upita (za različite moguće redoslijede spajanja relacija) s navedenim procjenama broja n-torki u međurezultatima i načinima pristupa podacima, te obrazložiti koje stablo (redoslijed spajanja) će odabrati optimizator nakon provedene heurističke optimizacije.

7. (5 bodova) Prikazan je odsječak sadržaja relacije *smjestaj* u trenutku započinjanja naredbe {1}. Uz pretpostavku da osim transakcija T1, T2 i T3 niti jedna druga transakcija ne obavlja nikakvu operaciju nad relacijom *smjestaj*, odredite rezultate naredbi {5}, {6}, {8}, {10} i {12} te objasnite što se događa s ključevima i uspješnost izvršavanja pri obavljanju naredbi {4}, {7}, {9}, {11} i {13}.

Smjestaj			
xmin	xmax	sifSmjestaj	cijena
...
20	0	10	1000.00

Transakcija T1 T _{id} =28		Transakcija T2 T _{id} =29		Transakcija T3 T _{id} =30	
{1}	BEGIN TRANSACTION;	{2}	BEGIN TRANSACTION;	{3}	BEGIN TRANSACTION;
{4}	UPDATE smjestaj SET cijena = cijena*1.1 WHERE sifSmjestaj = 10;	{5}	SELECT xmin, xmax, sifSmjestaj, cijena FROM smjestaj WHERE sifSmjestaj = 10;	{6}	SELECT xmin, xmax, sifSmjestaj, cijena FROM smjestaj WHERE sifSmjestaj = 10;
{7}	COMMIT TRANSACTION;	{8}	SELECT xmin, xmax, sifSmjestaj, cijena FROM smjestaj WHERE sifSmjestaj = 10;	{9}	COMMIT TRANSACTION;
{10}	SELECT xmin, xmax, sifSmjestaj, cijena FROM smjestaj WHERE sifSmjestaj = 10;	{11}	UPDATE smjestaj SET cijena = cijena*0.9 WHERE sifSmjestaj = 10;	{12}	SELECT xmin, xmax, sifSmjestaj, cijena FROM smjestaj WHERE sifSmjestaj = 10;
		{13}	ROLLBACK TRANSACTION;		

8. (3 boda)

Objasnite sljedeća integritetska ograničenja:

- entitetski integritet
- integritet ključa

9. (2 boda) Zadana je relacijska shema R = {ABCDEFGH} na kojoj vrijede sljedeće funkcijske zavisnosti:

F → AB, FC → D, EF → GH, B → C.

Ispitati vrijedi li funkcijska zavisnost EF → C, u svakom koraku dokaza napisati korišteno pravilo.

10. (8 bodova)

Modelirajte ER model dijela baze podataka softverskog sustava za potporu građevinskim tvrtkama.

Za svaku tvrtku u sustavu evidentira se njezin jedinstveni registracijski broj, naziv i telefonski broj. Tvrtka evidentira informacije o projektima koje izvodi. Projekt izvodi jedna tvrtka. Za projekt se evidentira njegova šifra, naziv, datum početka i datum ugovaranja te cijena. Projekti se sastoje od različitih faza koje se mogu izvoditi paralelno, za svaku fazu evidentira se redni broj, datum početka faze, njezin naziv koji nije jedinstven (npr. „iskop terena“) i trajanje. Pojedina faza obavlja se ispunjavanjem zadataka, koji su evidentirani šifrom zadatka, opisom i trajanjem. Zadatke u pojedinoj fazi obavljaju zaposlenici tvrtke pri čemu se evidentira broj odrađenih radnih sati tijekom obavljanja zadatka. Zadatke može obavljati više zaposlenika u pojedinoj fazi. Jedan zaposlenik određeni zadatak može obavljati samo u jednoj fazi.

Prilikom ispunjavanja zadataka koriste se materijali te se evidentira podatak o količini utrošenog materijala. Materijal je opisan šifrom, nazivom i mjernom jedinicom.

Za zaposlenike tvrtka evidentira njihov OIB, ime, prezime, elektronička pošta i datum rođenja, te za svaki projekt evidentira jednog zaposlenika koji je voditelj projekta.

Nacrtajte ER model baze podataka. Navedite sheme entiteta i sheme veza (označite ključeve). Svaki entitet opisati **isključivo vlastitim atributima** (osim slabih entiteta, ukoliko postoje u modelu).

Nužno je da sve sheme zadovoljavaju 3NF.

Rješenja

1. (5 bodova)

```
SELECT S.sifSmjestaj, prosOcj,
(SELECT COUNT(*)
  FROM rezervacija AS R
   WHERE R.sifSmjestaj = S.sifSmjestaj) AS ukRez,
CASE WHEN R.sifSmjestaj IS NULL THEN 'Da' ELSE 'Ne' END AS raspoloziv
FROM smjestaj AS S
  INNER JOIN smjestajPogodnost AS SP
    ON SP.sifSmjestaj = S.sifSmjestaj
  INNER JOIN pogodnost AS P
    ON P.sifPog = SP.sifPog
  LEFT JOIN rezervacija AS R
    ON R.sifSmjestaj = S.sifSmjestaj AND
      ('10.09.2018'::DATE, '22.09.2018'::DATE) overlaps (R.datDolazak, R.datOdlazak)
WHERE prosOcj > 4 AND
      P.nazivPog = 'Wifi'
ORDER BY prosOcj ASC, ukRez DESC

--ili
SELECT S.sifSmjestaj, prosOcj,
COUNT(R.sifSmjestaj) AS ukRez,
CASE WHEN (
  SELECT COUNT(*)
    FROM rezervacija AS R
   WHERE R.sifSmjestaj = S.sifSmjestaj
      AND ('10.09.2018'::DATE, '22.09.2018'::DATE) overlaps (R.datDolazak, R.datOdlazak)
) > 0 THEN 'Ne' ELSE 'Da' END AS raspoloziv
FROM smjestaj AS S
  INNER JOIN smjestajPogodnost AS SP
    ON SP.sifSmjestaj = S.sifSmjestaj
  INNER JOIN pogodnost AS P
    ON P.sifPog = SP.sifPog
  LEFT JOIN rezervacija AS R
    ON R.sifSmjestaj = S.sifSmjestaj
WHERE prosOcj > 4 AND
      P.nazivPog = 'Wifi'
GROUP BY S.sifSmjestaj, prosOcj
ORDER BY prosOcj ASC, ukRez DESC

--ili dva podupita u select listi, jedan za broj rezervacija, drugi za raspolozivost
```

2. (5 bodova)

```
SELECT S.sifSmjestaj, korisnickoIme, EXTRACT(YEAR FROM datReg) AS godReg,
SUM(datOdlazak-datDolazak) AS brojDana,
COUNT(SP.sifPog)
FROM smjestaj AS S
  INNER JOIN rezervacija AS R
    ON S.sifSmjestaj = R.sifSmjestaj
  INNER JOIN korisnik AS K
    ON K.sifKorisnik = S.sifDomacin
  LEFT JOIN smjestajPogodnost AS SP
    ON SP.sifSmjestaj = S.sifSmjestaj
GROUP BY S.sifSmjestaj, korisnickoIme, godReg
HAVING SUM(datOdlazak-datDolazak) <= ALL
  (SELECT SUM(datOdlazak-datDolazak)
   FROM rezervacija
  GROUP BY sifSmjestaj)
```

3. (5 bodova)

```
DELETE FROM korisnik
WHERE CURRENT_TIMESTAMP - datReg > '1 year'::INTERVAL
  AND NOT EXISTS (
    SELECT *
    FROM smjestaj AS S
    WHERE S.sifDomacin = sifKorisnik
  ) AND NOT EXISTS (
    SELECT *
    FROM rezervacija AS R
    WHERE R.sifGost = sifKorisnik
  );

--ili
DELETE FROM korisnik
WHERE CURRENT_TIMESTAMP - datReg > '1 year'::INTERVAL
AND sifKorisnik NOT IN ( SELECT sifGost FROM rezervacija)
AND sifKorisnik NOT IN ( SELECT sifDomacin FROM smjestaj);

UPDATE rezervacija
SET ocjena = NULL
WHERE ocjena IS NOT NULL AND sifGost IN
  (SELECT sifDomacin
   FROM smjestaj AS S
   WHERE S.sifSmjestaj = rezervacija.sifSmjestaj
  );
```

4. (7 bodova)

```
a)
ALTER TABLE rezervacija ADD CONSTRAINT provDatRez CHECK (datOdlazak > datDolazak);
ALTER TABLE rezervacija ADD CONSTRAINT provOcJRez CHECK (ocjena BETWEEN 1 AND 5 OR ocjena IS
  NULL);

b)
CREATE OR REPLACE FUNCTION insUpdRezervacija()
  RETURNS TRIGGER AS $$
DECLARE
  p_prosOcJ DECIMAL(6, 2) DEFAULT NULL;
BEGIN

  SELECT AVG(R.ocjena) INTO p_prosOcJ
  FROM rezervacija
  WHERE sifSmjestaj = NEW.sifSmjestaj
  AND ocjena IS NOT NULL;

  UPDATE smjestaj
  SET prosjOcJ = p_prosOcJ
  WHERE sifSmjestaj = NEW.sifSmjestaj;

  IF CURRENT_DATE - NEW.datDolaska < '3 MONTHS'::INTERVAL
    THEN RAISE EXCEPTION 'Rezervaciju je moguće obaviti najviše 3 mjeseca unaprijed!';
  END IF;

  RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER InsUpdRezervacija
  AFTER INSERT OR UPDATE ON rezervacija
  FOR EACH ROW
  EXECUTE PROCEDURE insUpdRezervacija();

c)
CREATE TABLE rezervacija (
  sifSmjestaj INT REFERENCES smjestaj(sifSmjestaj),
  sifGost INT REFERENCES korisnik(sifKorisnik),
  datDolaska DATE NOT NULL,
  datOdlaska DATE NOT NULL,
  ocjena SMALLINT NULL,
  PRIMARY KEY(sifSmjestaj, sifGost, datDolaska)
);
```

5. (6 bodova)

a)

```
CREATE USER dtrotter WITH PASSWORD 'plonker';
--ili CREATE ROLE dtrotter WITH LOGIN PASSWORD 'plonker';
GRANT CONNECT ON DATABASE rezervacijaSmjestaj TO dtrotter;
```

b)

```
CREATE ROLE gost;
CREATE VIEW mojeRezervacije AS
  SELECT *
  FROM rezervacija AS R
  WHERE R.sifGost IN
    (SELECT sifKorisnik
     FROM korisnik AS K
     WHERE K.korisnickoIme = SESSION_USER) --u ovom slučaju je u redu i CURRENT_USER
  WITH CHECK OPTION;

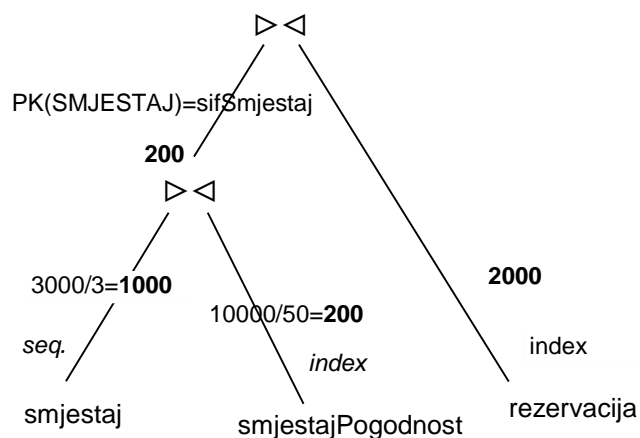
GRANT DELETE, SELECT, UPDATE (ocjena) ON mojeRezervacije TO gost;
```

c) GRANT gost TO dtrotter;

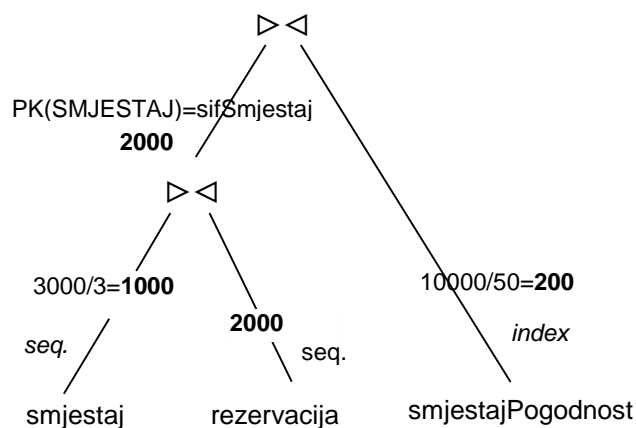
6. (4 boda)

Stablo c)

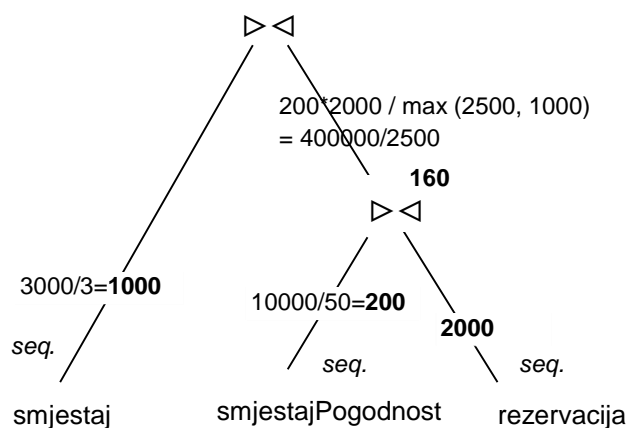
a)



b)



c)



7. (5 bodova)

Objašnjenje naredbi {4}, {7}, {9}, {11} i {13}:

{4}	T1 postavlja ekskluzivni ključ na n-torku u smjestaj sa sifSmjestaj = 10.
{7}	Potvrđuje se transakcija T1.
{9}	Potvrđuje se transakcija T3
{11}	T2 postavlja ekskluzivni ključ na n-torku u smjestaj sa sifSmjestaj = 10.
{13}	T2 se poništava - ROLLBACK

Rezultat naredbi {5}, {6}, {8}, {10} i {12}:

{5} i {6}				{8}, {10}				{12}			
<u>xmin</u>	<u>xmax</u>	<u>sifSmjestaj</u>	<u>Cijena</u>	<u>xmin</u>	<u>xmax</u>	<u>sifSmjestaj</u>	<u>cijena</u>	<u>xmin</u>	<u>xmax</u>	<u>sifSmjestaj</u>	<u>cijena</u>
20	28	10	1000.00	28	0	10	1100.00	28	29	10	1100.00

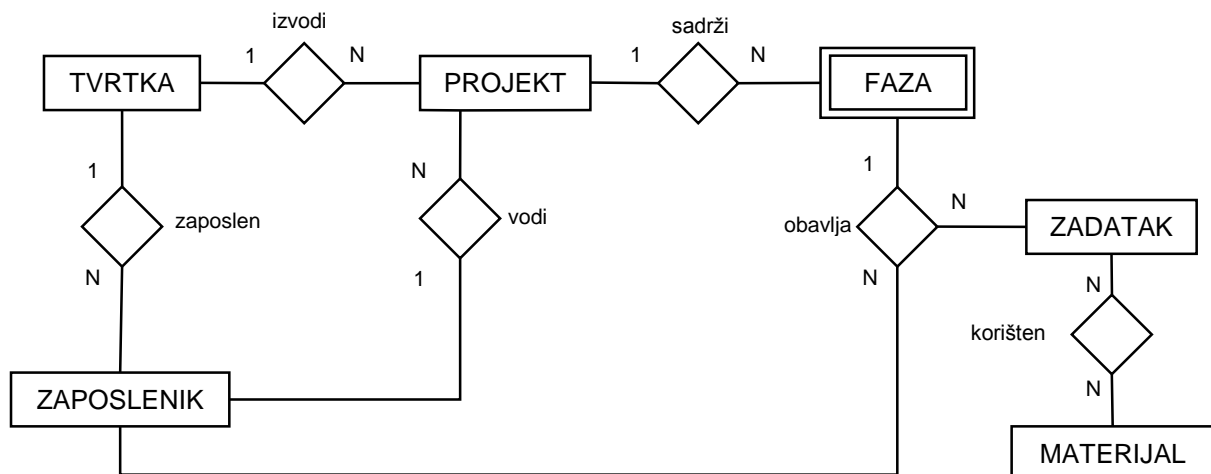
8. (3 boda)

- a) Niti jedan atribut primarnog ključa ne smije poprimiti NULL vrijednost
b) U relaciji ne smiju postojati dvije n-torke s jednakim vrijednostima ključa (vrijedi za sve moguće ključeve)

9. (2 boda)

- EF → EF (refleksivnost)
EF → EF i **EF → GH** ⇒ EF → EFGH (akumulacija)
EF → EFGH i **F → AB** ⇒ EF → ABEFGH (akumulacija)
EF → ABEFGH i **B → C** ⇒ EF → ABCEFGH (akumulacija)
EF → ABCEFGH ⇒ EF → C (dekompozicija)

10. (8 bodova)



PROJEKT <u>sifProjekt</u> nazivP datPoc datUg cijena	TVRTKA <u>regBr</u> nazivT adresa telBroj	ZAPOSLENIK <u>OIB</u> ime prezime eposta datRod	FAZA <u>sifProjekt</u> <u>rbr</u> nazivF datPoc trajanje	ZADATAK <u>sifZadatak</u> opisZ trajanje	MATERIJAL <u>sifMaterijal</u> nazivM mjJedinica
zaposlen <u>OIB</u> regBr	izvodi <u>sifProjekt</u> regBr	vodi <u>sifProj</u> OIB	sadrži <u>sifProjekt</u> <u>rbr</u>	obavlja <u>OIB</u> <u>sifZadatak</u> sifProjekt rbr brSati	korišten <u>sifZadatak</u> <u>sifMaterijal</u> kolicina