

Završni ispit iz Baza podataka

21. lipnja 2013.

Zadaci 1-7 odnose se na bazu podataka **legoDB** u kojoj se pohranjuju podaci o kompletima kockica i elementima (tj. kockicama) koje ti kompleti sadrže. Kako je moguća prodaja i kompleta i pojedinačnih elemenata, svi kompleti i elementi navedeni su u tablici **artikl**, a pojedini artikl može biti vrste K ili E (komplet ili element). Jedinstveni identifikator svakog artikla je šifra artikla, a osim vrste artikla, za artikl se navodi naziv i trenutna cijena. Kompleti su dodatno opisani u relaciji **komplet**, a elementi u relaciji **element**. Atributi **proizvOd** i **proizvDo** predstavljaju datume početka, odnosno završetka proizvodnje kompleta. U relaciji **sadrzaj** naveden je sadržaj kompleta (atribut **sifArtiklK**), tj. koje elemente (atribut **sifArtiklE**) u kojoj boji i količini (atributi **boja** i **brElem**) taj komplet sadrži.

Na slici **nisu** prikazane sve n-torke sadržane u relacijama.

artikl				komplet			sadrzaj			
sifArtikl	nazArtikl	vrArtikl	cijena	sifArtikl	proizvOd	proizvDo	sifArtiklK	sifArtiklE	boja	brElem
1	Off-Road Power	K	219.99	1	01.03.2010	30.05.2013	1	3	black	12
2	T-Rex	K	59.99	2	01.06.2012	NULL	1	4	blue	7
3	Brick 1X2	E	0.80	6	01.01.2004	01.01.2008	1	4	red	4
4	Plate 2X3	E	0.60				1	4	black	4
5	Brick 2X2	E	0.80				2	3	red	2
6	Creator Box	K	99.49				2	5	red	8
7	Wheel Technic 30x 14	E	5.49							

element	
sifArtikl	vrElement
3	brick
4	plate
5	brick
7	wheel

- Napisati SQL naredbe za kreiranje relacija u bazi **legoDB**, kojima će biti definirana sva integritetska ograničenja navedena u uvodnom opisu modela (koja je moguće kreirati CREATE TABLE naredbom), kao i sljedeća ograničenja:
 - atribut **sifArtiklK** u relaciji **sadrzaj** može poprimiti vrijednost šifre artikla navedene u tablici **komplet**, a atribut **sifArtiklE** vrijednost šifre artikla navedene u tablici **element**;
 - u tablici **sadrzaj** smije postojati samo jedan zapis kojim se evidentira da neki komplet sadrži neki element u određenoj boji (npr. smije postojati samo jedan zapis o tome da komplet 1 sadrži element 3 u crnoj boji);
 - NULL vrijednost smije poprimiti samo atribut **proizvDo** (za komplete koji se još uvijek proizvode);
 - ne može postojati više kompleta istog naziva niti više elemenata istog naziva.Smisleno odabrati tipove atributa. **(4 boda)**
- Napisati SQL naredbe kojima će biti spriječena promjena vrijednosti atributa **vrArtikl** u tablici **artikl**, ako za artikl postoji odgovarajući zapis u tablici **komplet** ili **element**, ovisno o vrsti artikla. Prilikom pokušaja obavljanja takve izmjene, dojaviti jednu od poruka: "Artikl *sifra* - komplet je već opisan!" ili "Artikl *sifra* - element je već opisan!". Napomena: *sifra* u tekstu poruke mora odgovarati stvarnoj vrijednosti šifre artikla. **(6 bodova)**
- Ispisati naziv i cijenu elemenata čiji naziv sadrži znakovni niz '2x2' ili '1x2', nisu vrste *brick*, a ima ih barem 6 komada u crnoj (*black*) boji u barem jednom kompletu. Bliže početku liste ispisati skuplje elemente, a one iste cijene poredati po nazivu elementa. Podatke o elementu ispisati samo jednom, čak i ako se traženi element pojavljuje u više kompleta **Zadatak riješiti jednim SQL upitom, bez korištenja podupita.** **(3 boda)**
- Za elemente koji se ne pojavljuju niti u jednom kompletu čija je proizvodnja započela od početka prošle godine, ispisati naziv elementa, te ukupan broj kompleta u kojima se element pojavljuje (bez obzira na boju elementa i vrijeme proizvodnje kompleta). Rezultat mora sadržavati i elemente koji se nikada nisu pojavili niti u jednom kompletu - za takve je elemente za ukupan broj kompleta potrebno ispisati vrijednost 0. **(4 boda)**
- Napisati (jednostavan) SQL upit koji predstavlja primjer za **agregaciju** (operaciju relacijske algebre) te napisati rezultat obavljanja napisanog upita. U ovom zadatku pretpostaviti da su na slici prikazani svi podaci pohranjeni u bazi podataka. **(2 boda)**

6. Pretpostavka je da je u bazi podataka kreiran samo jedan indeks, naredbom:

```
CREATE INDEX i1 ON sadrzaj (sifArtiklE, boja)
```

Izvodi se sljedeći upit:

```
SELECT * FROM artikl, sadrzaj, element
WHERE artikl.sifArtikl = sadrzaj.sifArtiklE
      AND artikl.sifArtikl = element.sifArtikl
      AND cijena < 30.00
      AND boja = 'red'
```

Optimizator raspolaže sljedećim informacijama:

```
N(artikl) = 12 000
N(sadrzaj) = 300 000
N(element) = 9 000
V(cijena, artikl) = 8000
V(boja, sadrzaj) = 30
```

Nacrtati stablo upita nakon provedene heurističke optimizacije. Redoslijed spajanja relacija određen je redoslijedom kojim su relacije navede u FROM dijelu SELECT naredbe. Navesti sve izraze prema kojima je obavljena procjena broja n-torki u međurezultatima. U planu naznačiti očekivani broj zapisa za međurezultate, te korištene metode pristupa. Dovoljno je nacrtati samo konačno stablo upita. **(5,5 bodova)**

7. Korisnik **k1** je vlasnik baze **legoDB** i svih objekata u njoj. Pristup bazi, bez mogućnosti kreiranja novih objekata, omogućio je samo korisnicima **k2**, **k3** i **k4**, a niti jedan korisnik nema nikakvu dozvolu nad objektima u bazi. Obavljanje prikazanih naredbi pokreće se navedenim redoslijedom, pri čemu će neke naredbe biti obavljene uspješno, a neke će rezultirati pogreškom. Korisnik pokreće naredbu bez obzira je li prethodna naredba uspješno obavljena. Za svaku naredbu navesti hoće li biti uspješno obavljena, te obavezno obrazložiti razlog uspjeha/neuspjeha njezina obavljanja.

```
1) k1: GRANT SELECT ON artikl TO k2 WITH GRANT OPTION;
2) k2: GRANT SELECT ON artikl TO k3;
3) k3: GRANT SELECT ON artikl TO k4;
4) k1: REVOKE SELECT ON artikl FROM k2 RESTRICT;
```

Koji sve korisnici imaju SELECT dozvolu nad relacijom **artikl** i je li ona prenosiva, nakon pokretanja prikazanih naredbi? **(2,5 boda)**

8. U bazi podataka se evidentiraju podaci o kandidatima, političkim strankama i mjestima, potrebni za provođenje izbora za gradonačelnika. O kandidatima za gradonačelnika se bilježi OIB, ime i prezime, datum rođenja te prebivalište. Prate se promjene prebivališta kandidata. Za prebivalište se evidentira ulica i kućni broj prebivanja kao i datum početka prebivanja na konkretnoj adresi. Osoba u jednom trenutku može prebivati samo na jednoj adresi. Za ulicu se bilježi šifra (identifikator) i naziv te mjesto u kojem se ulica nalazi, a za mjesto poštanski broj (identifikator) i naziv.

Za političke stranke se bilježi šifra (identifikator) i naziv stranke te godina osnivanja.

Za svako mjesto za koje se bira gradonačelnik, evidentiraju se kandidati političkih stranaka za gradonačelnika. Jedna politička stranka u jednom mjestu podržava samo jednog kandidata. Pojednog kandidata u mjestu može podržavati više političkih stranaka.

Nacrtati ER model i opisati entitete i veze. Entitete, osim slabih entiteta, opisati isključivo vlastitim atributima. Sve sheme moraju zadovoljavati 3NF. **(5 bodova)**

9. Objasniti ulogu i opisati faze protokola dvofaznog zaključavanja (2PL). Navesti i ukratko opisati svojstvo transakcije koje taj protokol osigurava. **(4 boda)**

10. Administrator IBM Informix sustava za upravljanje bazama podataka obavlja arhiviranje na sljedeći način:

- razina 0 svakog prvog ponedjeljka u mjesecu (npr. za lipanj 2013: 03. lipanj),
- razina 1 svakih 7 dana (za lipanj 2013: 10. lipanj, 17. lipanj,...),
- razina 2 svakodnevno u 03:00 sata.

Zbog kvara na fizičkom mediju na kojem su pohranjeni podaci iz baza podataka, do kojeg je došlo 12. lipnja 2013 u 11:00 sati administrator mora obaviti obnovu. Sve arhive su ispravne.

- Navesti što i kojim redoslijedom se koristi prilikom obnove.
- Mogu li u postupku obnove biti izgubljeni efekti transakcija koje su potvrđene neposredno prije kvara, te transakcija koje su bile u tijeku u trenutku kvara. Obavezno obrazložiti odgovor. **(4 boda)**

Rješenja:

1. (4 boda)

```
CREATE TABLE artikl (sifArtikl INTEGER PRIMARY KEY
, nazArtikl CHAR(250) NOT NULL
, vrArtikl CHAR(1) NOT NULL CHECK (vrArtikl IN ('K','E'))
, cijena DECIMAL(8,2) NOT NULL
, UNIQUE (nazArtikl, vrArtikl));

CREATE TABLE komplet (sifArtikl INTEGER PRIMARY KEY REFERENCES artikl(sifArtikl)
, proizvoD DATE NOT NULL
, proizvDo DATE);

CREATE TABLE element (sifArtikl INTEGER PRIMARY KEY REFERENCES artikl(sifArtikl)
, vrElement CHAR(20) NOT NULL);

CREATE TABLE sadrzaj (sifArtiklK INTEGER REFERENCES komplet(sifArtikl)
, sifArtiklE INTEGER REFERENCES element(sifArtikl)
, boja CHAR(30)
, brElem SMALLINT NOT NULL
, PRIMARY KEY (sifArtiklK, sifArtiklE, boja));
```

2. (6 bodova)

```
CREATE PROCEDURE provjeriIzmjenu (p_sifArtikl LIKE artikl.sifArtikl,
p_vrArtikl LIKE artikl.vrArtikl)

IF p_vrArtikl = 'K' THEN
  IF EXISTS (SELECT * FROM komplet WHERE sifArtikl = p_sifArtikl) THEN
    RAISE EXCEPTION -746, 0, 'Artikl: ' || p_sifArtikl || ' - komplet je već opisan!';
  END IF
ELSE
  IF EXISTS (SELECT * FROM element WHERE sifArtikl = p_sifArtikl) THEN
    RAISE EXCEPTION -746, 0, 'Artikl: ' || p_sifArtikl || ' - element je već opisan!';
  END IF
END IF
END PROCEDURE;

CREATE TRIGGER artiklUpd
UPDATE OF vrArtikl ON artikl
REFERENCING NEW AS noviArtikl OLD AS stariArtikl
FOR EACH ROW
  WHEN (noviArtikl.vrArtikl <> stariArtikl.vrArtikl)
    (EXECUTE PROCEDURE provjeriIzmjenu (stariArtikl.sifArtikl, stariArtikl.vrArtikl));
```

3. (3 boda)

```
SELECT DISTINCT artikl.nazArtikl, artikl.cijena
FROM artikl JOIN sadrzaj ON sadrzaj.sifArtiklE = artikl.sifArtikl
JOIN element ON artikl.sifArtikl = element.sifArtikl
WHERE brElem >= 6
AND boja = 'black'
AND vrElement <> 'brick'
AND (nazArtikl LIKE '%2x2%' OR nazArtikl LIKE '%1x2%')
ORDER BY artikl.cijena DESC, artikl.nazArtikl
```

4. (4 boda)

```
SELECT nazArtikl, (SELECT COUNT (DISTINCT sifArtiklK) FROM sadrzaj
WHERE sifArtiklE = artikl.sifArtikl)
FROM artikl
WHERE vrArtikl = 'E'
AND sifArtikl NOT IN (SELECT DISTINCT sifArtiklE
FROM sadrzaj JOIN komplet
ON sadrzaj.sifArtiklK = komplet.sifArtikl
WHERE YEAR(proizvoD) >= YEAR(TODAY)-1);

ili

SELECT nazArtikl, COUNT (DISTINCT sifArtiklK)
FROM artikl LEFT JOIN sadrzaj ON sifArtiklE = artikl.sifArtikl
WHERE vrArtikl = 'E'
AND sifArtikl NOT IN (SELECT DISTINCT sifArtiklE
FROM sadrzaj JOIN komplet ON sadrzaj.sifArtiklK = komplet.sifArtikl
WHERE YEAR(proizvoD) >= YEAR(TODAY)-1)
GROUP BY nazArtikl;
```

5. (2 boda)

agregacija - primjenom agregatne funkcije dobije se relacija stupnja 1 i kardinalnosti 1, npr.

`SELECT COUNT(*) from artikl`

6. (5,5 bodova)

- procjena broja n-torki u međurezultatima dobivenom selekcijama:

$$\text{sadrzaj}_1 = \sigma_{\text{boja}='red'}(\text{sadrzaj})$$

$$N(\text{sadrzaj}_1) = N(\text{sadrzaj}) / V(\text{boja}, \text{sadrzaj}) = 300\,000 / 30 = 10\,000$$

$$\text{artikl}_1 = \sigma_{\text{cijena} < 30}(\text{artikl})$$

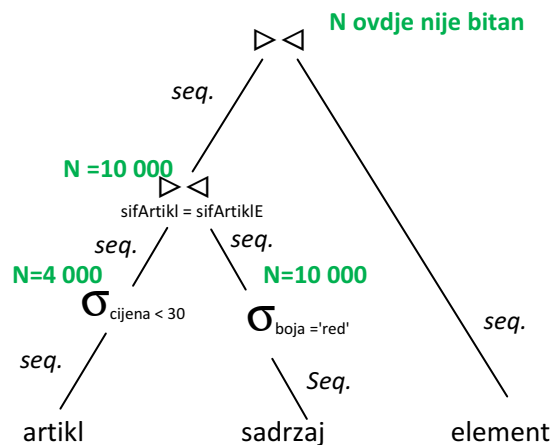
$$N(\text{artikl}_1) = N(\text{artikl}) / 3 = 12\,000 / 3 = 4\,000$$

- procjena broja n-torki u međurezultatu dobivenom spajanjem:

$$N(\text{artikl}_1 \bowtie \text{sadrzaj}_1) = N(\text{sadrzaj}_1) = 10\,000$$

sifArtikl = sifArtiklE

($\text{artikl}_1 \cap \text{sadrzaj}_1$ je ključ relacije artikl)

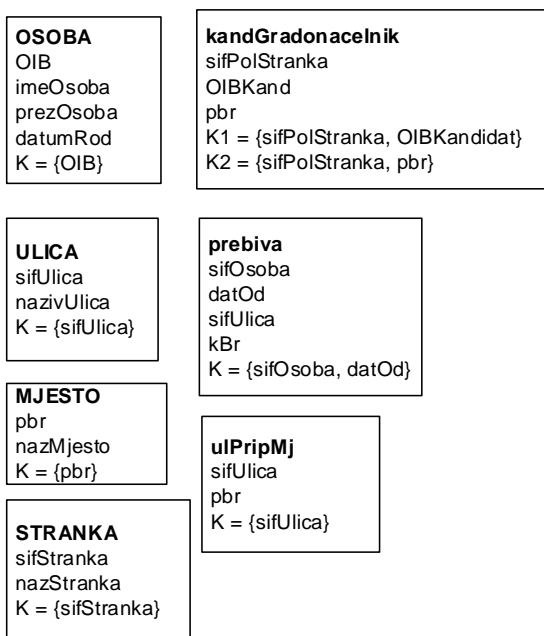
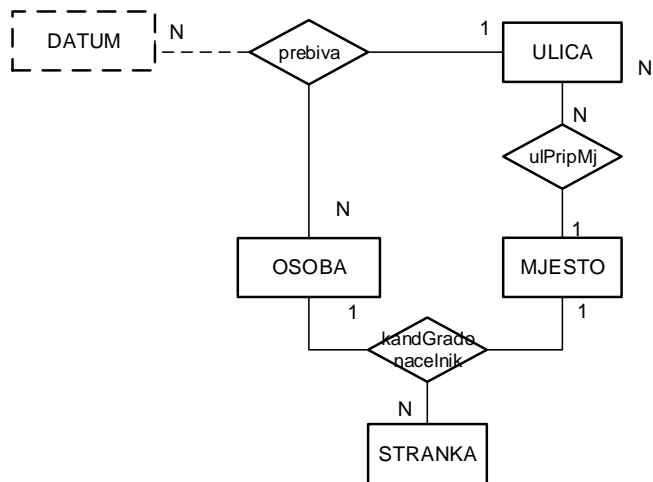


7. (2,5 boda)

- 1) k1-vlasnik baze dodjeljuje prenosivu dozvolu korisniku k2 - ok, jer je vlasnik objekta
- 2) k2-dobio je prenosivu dozvolu, pa daje dozvolu pregleda korisniku k3 -ok;
- 3) k3-nema prenosivu dozvolu pa ne može dati dozvolu pregleda korisniku k4 -pogreška;
- 4) k1 zbog opcije RESTRICT neće moći povući dozvolu s k2, jer k2 dodijelio dozvolu korisniku k3

Na kraju bi dozvolu imali k1, k2 i k3. Od toga k1 i k2 prenosivu.

8. (5 bodova)



9. (4 boda)

Protokol dvofaznog zaključavanja osigurava serijalizabilan redoslijed izvršavanja transakcija.

Sastoji se od 2 faze:

1. prije obavljanja operacije nad objektom (npr. n-torkom iz baze), transakcija mora za taj objekt zatražiti ključ
2. nakon otpuštanja ključa transakcija ne smije više zatražiti nikakav ključ

Transakcije koje poštuju 2PL protokol imaju 2 faze - fazu pribavljanja ključeva (faza rasta - **growing phase**) i fazu otpuštanja ključeva (fazu sužavanja - **shrinking phase**)

2PL protokol osigura svojstvo izolacije transakcije - kada se paralelno obavljaju dvije ili više transakcija, njihov učinak mora biti jednak kao da su se obavljale jedna iza druge

10. (4 boda)

Obnovu je moguće obaviti na sljedeći način:

1. Arhiva razine 0 od 3.lipnja
2. Arhiva razine 1 od 10.lipnja
3. Arhiva razine 2 od 12.lipnja
4. Logički dnevnik započet 12.lipnja nakon izrade arhive razine 2

Efekti transakcija koje su potvrđene prije kvara su očuvani, a efekti transakcija koje su bile u tijeku u trenutku kvara su izgubljeni.

Sve promjene koje se obavljaju nad bazom podataka se prvo bilježe u dnevniku. Sadržaj spremnika dnevnika mora biti zapisan u dnevnik prije nego što završi procedura potvrđivanja transakcije (write-ahead log rule). Dakle, u trenutku potvrđivanja transakcije sigurno postoje sve informacije o transakciji u dnevniku. Stoga se efekti transakcija koje su potvrđene prije kvara mogu ponoviti na temelju dnevnika pa su ti efekti očuvani. Efekti transakcija koje nisu završene prije kvara se poništavaju na temelju zapisa iz dnevnika (čak i u slučaju da su bili u bazu zapisani u trenutku kvara).