

Ispit iz Baza podataka

12. srpnja 2019. (50 bodova)

Baza podataka **gimNatjecanja** sadrži podatke o gimnastičkim natjecanjima. Na natjecanjima (**nat**) gimnastičarke i gimnastičari (**gim**) se natječu u različitim programima (**prog**) (npr. *prednatjecanje*, *ekipno finale*, *finale višeboja* i *finale po disciplinama*). Za gimnastičare se evidentira sudjelovanje u disciplinama programa natjecanja (**gimNat**) i konačna ocjena. Na osnovi konačnih ocjena (**gimNat.ocjena**) iz programa koji pripadaju fazi prednatjecanja, gimnastičari se plasiraju u neki od programa finala. Ako konačna ocjena nije postavljena, gimnastičari su prijavljeni, ali još nisu nastupali i dobili ocjene. Pojedine discipline opisaneč su u tablici **disciplina**. Zbog jednostavnosti pretpostaviti da gimnastičari nikad ne mijenjaju klub.

Zadaci 1. – 5. se odnose na bazu podataka **gimNatjecanja** prikazanu na **slici 1** i podrazumijeva se korištenje SUBP-a PostgreSQL. Na slici **nisu prikazane** sve n-torke sadržane u relacijama. Svi tekstni podaci su tipa VARCHAR.

gim						prog		nat		
sifGim	ime	prezime	datRod	spol	sifKlub	sifProg	nazProg	sifNat	datOdr	nazNat
57	Iva	Ivančić	25.12.2003	Ž	310	1	prednatjecanje	3009	24.12.2018	3. Žabica kup
62	Marko	Puh	18.03.2000	M	330	2	ekipno finale	3022	04.05.2019	IX. ATON kup
78	Tjaša	Kranjc	30.06.2009	Ž	340	3	finale višeboja	3047	07.06.2019	Državno prvenstvo
...	4	finale po disciplini
...

gimNat					disciplina			klub		
sifNat	sifDis	sifProg	sifGim	ocjena	sifDis	nazDis	tip	sifKlub	nazKlub	oznDr
3009	1	1	78	8.8	1	parter	O	310	GK Trešnjevka	HR
3022	2	3	62	7.2	2	preča	M	330	GK Split	HR
3047	3	2	57	NULL	3	greda	Ž	340	GC Ljubljana	SI
...

Slika 1.

U zadacima 1. – 3. napisati **po jednu SQL naredbu** kojom će se obaviti sljedeće:

- (5 bodova)** Za svaki klub čiji su gimnastičari ili gimnastičarke ikada sudjelovali na nekom natjecanju ispišite naziv kluba te naziv discipline i najboljeg gimnastičara ili gimnastičarku u klubu u toj disciplini. Najbolji gimnastičar/ka određuje se na temelju najbolje prosječne ocjene u pojedinoj disciplini sa svih natjecanja neovisno o programu natjecanja u pojedinoj disciplini.
Podatke o gimnastičaru/gimnastičarki (ili više njih, ako imaju istu ocjenu) potrebno je ispisati u obliku prvo slovo imena, prezime i godina rođenja („i.prezime-godRod“), npr. „I.Ivančić-2003“. Discipline u kojima se nijedan gimnastičar nije natjecao nije potrebno prikazivati u ispisu.
- (4 boda)** Ispišite oznaku države, broj različitih natjecanja i broj različitih gimnastičara koji su se natjecali na natjecanjima održanim u zadnjih godinu dana u programu „*ekipno finale*“. Pojedina država uključuje sve klubove koji su u toj državi, odnosno imaju istu oznaku države. Za države čiji klubovi u navedenom periodu nemaju gimnastičare koji su se natjecali u tom programu, za broj natjecanja i gimnastičara potrebno je ispisati 0. Natjecanja koja ne sadrže navedeni program ne treba uzimati u obzir.
- (5 bodova)** Napišite SQL naredbu kojom će se u relaciju **gimNat**, na temelju ocjena završenog prednatjecanja za natjecanje „Državno prvenstvo“ koje se održava u trenutnoj godini, unijeti plasiranje gimnastičara za program naziva „*finale po disciplini*“. U finale po disciplinama za svaku disciplinu plasira se 8 najboljih u toj disciplini iz programa s nazivom „*prednatjecanje*“.
- (6 bodova)** Pretpostavite da su u bazi podataka **gimNatjecanja** kreirane sve relacije te definirani svi primarni i strani ključevi. Napišite SQL naredbe kojima će se:
 - osigurati ispravnost atributa **gimNat.ocjena** i **disciplina.tip**. Vrijednost atributa **ocjena** je ispravna, ako ocjena nije postavljena (NULL) ili je u intervalu [0, 10] (uključivo). Discipline po tipu mogu biti u ženskoj, muškoj ili u muškoj i ženskoj konkurenciji (npr. parter). Atribut **tip** je ispravan, ako je tip postavljen na jednu od sljedećih vrijednosti: 'O', 'Ž' i 'M',
 - kreirati minimalan broj potrebnih objekata kojim će se pri unosu n-torke u relaciju **gimNat** osigurati ispravne vrijednosti atributa **sifDis** i **sifGim**, te ispravan unos zapisa za programe natjecanja koji nisu prednatjecanja (naziv programa nije „prednatjecanje“).
Vrijednosti atributa **sifDis** i **sifGim** su ispravne, ako je disciplina predviđena (dozvoljena) za spol kojem gimnastičar/ka pripada. Pri narušavanju ovog ograničenja korisniku javiti sljedeću grešku:
„Pogreška: neispravan unos discipline za gimnastičara/gimnastičarku!“
Unos zapisa za programe natjecanja koji nisu prednatjecanja je dozvoljen, ako za gimnastičara postoji ocjena u istoj disciplini i natjecanju iz prednatjecanja (programa s nazivom „prednatjecanje“). Pri narušavanju ograničenja korisniku javiti sljedeću grešku: „Pogreška: ne postoji rezultat iz prednatjecanja za gimnastičara/gimnastičarku!“

5. (2 boda) Administrator baze podataka je, nakon kreiranja baze i relacija u shemi *public*, obavio sljedeće SQL naredbe:

```
REVOKE CONNECT ON DATABASE gimNatjecanja FROM PUBLIC;
```

```
REVOKE ALL ON SCHEMA public FROM PUBLIC;
```

- Napišite SQL naredbe kojima će administrator baze podataka postojećim korisnicima *josip*, *ana* i *petar* dodijeliti ovlasti za spajanje na bazu podataka i pristup objektima u shemi *public*, te dodatno korisniku *josip* pregled podataka tablica ***klub***, ***prog*** i ***nat*** uz mogućnost dodjeljivanja te iste dozvole ostalim korisnicima.
- Neka je korisnik *josip* dodijelio ovlast pregleda podataka tablica ***klub***, ***prog*** i ***nat*** korisnicima *ana* i *petar*. Napišite **jednu** SQL naredbu kojom će administrator ukinuti dozvole pregleda podataka tablica ***klub***, ***prog*** i ***nat*** korisnicima *josip*, *ana* i *petar*.

6. (3 boda) Na relacijskoj shemi $R = \{A, B, C, D, E, F\}$ vrijede sljedeće funkcijske zavisnosti:

$$F = \{AE \rightarrow BDF, A \rightarrow BCF, C \rightarrow D\}$$

- Ispitajte vrijedi li funkcijska zavisnost $AE \rightarrow BCDF$. Za svaki korak dokaza navedite naziv pravila ili aksioma koji se koristi.
- Da li je funkcijska zavisnost $AE \rightarrow BCDF$ potpuna ili nepotpuna? Obrazložite odgovor.

7. (6 bodova) Zaposlenici dnevno evidentiraju sate rada na projektima. Relacijska shema RAD sastoji se od sljedećih atributa:

sifProjekt, *nazProjekt*, *potrebnoSati*, *mbrZap*, *imeZap*, *prezZap*, *sifRadMjestoZap*, *nazRadMjestoZap*, *potrebnoSatiZap*, *datumRadZap*, *brSatiRadZap*.

Atribut ***potrebnoSati*** je ukupan broj sati rada potrebnih za završetak projekta, dok je atribut ***potrebnoSatiZap*** ukupan broj sati koje zaposlenik treba u konačnici odraditi na projektu. Atribut ***brSatiRadZap*** označava broj sati koje je zaposlenik odradio određenog dana (***datumRadZap***) na određenom projektu.

Normalizirajte relaciju postepeno na 1NF, 2NF i 3NF, ako na jednom projektu radi više zaposlenika, a jedan zaposlenik može raditi istovremeno na više projekata. Jedan zaposlenik radi na jednom radnom mjestu, ali može postojati više zaposlenika s istom šifrom radnog mjesta.

8. (3 boda) Navedite barem tri karakteristična problema istodobnog pristupa i opišite ih.

9. (4 boda)

- U kontekstu fizičke organizacije podataka u bazi, što znači da je stablo "balansirano"?
- Zadano je B⁺ stablo reda $n = 100$ koje sadrži ukupno 6000 n-torki. Koliko je u **najgorem** slučaju potrebno U/I operacija da bi se dohvatio blok s podacima? Napišite detaljan postupak izračuna.

10. (8 bodova) U bazi podataka evidentiraju se podaci o sudskim postupcima. Osoba može sudjelovati u nekom postupku u ulozi stranke ili odvjetnika. Za svaku osobu evidentira se OIB, ime, prezime i datum rođenja. Za odvjetnika se dodatno bilježi datum upisa u odvjetničku komoru, a za stranku (redundantno) podatak o broju aktivnih postupaka.

Za sudski postupak se evidentira jedinstveni broj postupka, datum do kojeg mora biti pokrenut, datum razrješenja, stranka koja pokreće postupak, stranka protiv koje je postupak pokrenut te sud na kojem se postupak vodi. Jedan postupak se vodi protiv samo jedne stranke. Svaki postupak se vodi na istom sudu od početka do kraja.

Za sud se evidentira šifra i naziv suda, vrsta suda i mjesto sjedišta (poštanska oznaka i naziv mjesta). Postoji više vrsta sudova (npr. 1- općinski, 2- županijski, 3 - vrhovni...). Sudovi mogu imati nadređeni sud - npr. Županijski sud u Dubrovniku je nadređen Općinskom sudu u Dubrovniku.

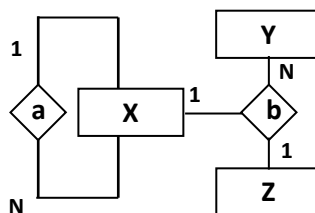
Za stranke se evidentira odvjetnik koji ih zastupa u nekom sudskom postupku. Jednu stranku mogu zastupati različiti odvjetnici na različitim postupcima, ali samo jedan te isti u jednom postupku od početka do kraja.

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**.

11. (4 bodova) Za zadani ER model napišite ekvivalentni relacijski model u obliku SQL naredbi za kreiranje relacija s ugrađenim opisima primarnih, alternativnih ključeva i općih pravila integriteta. Tipove podataka u naredbama nije potrebno navoditi.

	atributi	ključevi
entitet X	$x_1 x_2$	$K = \{x_1\}$
entitet Y	$y_1 y_2$	$K = \{y_1, y_2\}$
entitet Z	$z_1 z_2 z_3$	$K = \{z_1\}$



RJEŠENJA:

1. (5 bodova)

```
SELECT klub.nazKlub AS klub, disciplina.nazDis AS disciplina,
       SUBSTRING(gim.ime FROM 1 FOR 1) || '.' || gim.prezime || '-' || EXTRACT(YEAR FROM
datRod) AS najboljiGim
FROM klub
      JOIN gim ON gim.sifKlub = klub.sifKlub
      JOIN gimNat ON gimNat.sifGim = gim.sifGim
      JOIN disciplina
            ON disciplina.sifDis = gimNat.sifDis
GROUP BY klub.sifKlub, klub.nazKlub, disciplina.sifDis, disciplina.nazDis, gim.sifGim,
gim.prezime, gim.datRod
HAVING AVG(ocjena) >= ALL (SELECT AVG(ocjena)
                           FROM gimNat
                           JOIN gim
                                ON gim.sifGim = gimNat.sifGim
                           WHERE gim.sifKlub = klub.sifKlub
                                AND gimNat.sifDis = disciplina.sifDis
                           GROUP BY gim.sifGim);
```

2. (4 boda)

```
SELECT oznDr, COUNT(DISTINCT gimNat.sifNat) AS brNat,
       COUNT(DISTINCT gim.sifGim) AS brGim
FROM gimNat
      JOIN gim ON gim.sifGim = gimNat.sifGim
      JOIN prog ON prog.sifProg = gimNat.sifProg AND
            prog.nazProg = 'ekipno finale'
      JOIN nat ON nat.sifNat = gimNat.sifNat
            AND nat.datOdrz >= CURRENT_TIMESTAMP - '1 year'::INTERVAL
      RIGHT JOIN klub
            ON klub.sifKlub = gim.sifKlub
GROUP BY oznDr;
```

3. (5 bodova)

```
INSERT INTO gimNat(sifNat, sifDis, sifProg, sifGim)
SELECT gimNat.sifNat, sifDis,
       (SELECT sifProg FROM prog WHERE nazProg = 'finale po disciplini'),
       sifGim
FROM gimNat
      JOIN nat
            ON nat.sifNat = gimNat.sifNat
      JOIN prog
            ON prog.sifProg = gimNat.sifProg
WHERE nat.nazNat = 'Državno prvenstvo' AND
      EXTRACT(YEAR FROM datOdrz) = EXTRACT(YEAR FROM CURRENT_TIMESTAMP) AND --CURRENT_DATE
      prog.nazProg = 'prednatjecanje'
ORDER BY ocjena DESC
LIMIT 8;
```

4. (6 bodova)

a) (1 bod)

```
ALTER TABLE gimNat ADD CONSTRAINT provOcjena CHECK (ocjena IS NULL OR ocjena BETWEEN 0
AND 10);

ALTER TABLE disciplina ADD CONSTRAINT provTipDiscipline CHECK (tip IN ('O', 'M', 'Ž'));
```

b) (5 bodova)

```
CREATE OR REPLACE FUNCTION provInsGimNat()
  RETURNS TRIGGER AS $$
DECLARE
  p_spolGim CHAR;
  p_tipDis CHAR;
BEGIN
  SELECT spol INTO p_spolGim
  FROM gim
  WHERE gim.sifGim = NEW.sifGim;

  SELECT tip INTO p_tipDis
  FROM disciplina
  WHERE disciplina.sifDis = NEW.sifDis;

  IF p_tipDis <> 'O' AND p_spolGim <> p_tipDis THEN
    RAISE EXCEPTION 'Pogreška: neispravan unos discipline za gimnastičara!';
    RETURN NULL;
  END IF;

  IF (SELECT nazProg FROM prog WHERE prog.sifprog = NEW.sifProg) <> 'prednatjecanje'
  AND NOT EXISTS
    (
      SELECT *
      FROM gimNat, prog
      WHERE sifNat = NEW.sifNat AND sifDis = NEW.sifDis
        AND prog.sifprog = gimNat.sifProg
        AND sifGim = NEW.sifGim
        AND ocjena IS NOT NULL
        AND prog.nazProg = 'prednatjecanje'
    ) THEN
    RAISE EXCEPTION 'Pogreška: ne postoji rezultat iz prednatjecanja za
gimnastičara!';
    RETURN NULL;
  END IF;

  RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER UnosGimNat
  BEFORE INSERT ON gimNat
  FOR EACH ROW
  EXECUTE PROCEDURE provInsGimNat();
```

5. (2 boda)**a)**

```
GRANT CONNECT ON DATABASE gimNatjecanja to ana, josip, petar;
GRANT USAGE ON SCHEMA public TO ana, josip, petar;
GRANT SELECT ON klub, prog, nat TO josip WITH GRANT OPTION;
```

b) REVOKE SELECT ON klub, prog, nat FROM josip CASCADE;

6. (3 boda)

a) (2 boda)

$AE \rightarrow AE$

refleksivnost

$AE \rightarrow AE \wedge AE \rightarrow BDF \Rightarrow AE \rightarrow ABDEF$

akumulacija

$AE \rightarrow ABDEF \wedge A \rightarrow BCF \Rightarrow AE \rightarrow ABCDEF$

akumulacija

$AE \rightarrow ABCDEF \Rightarrow AE \rightarrow BCDF$

dekompozicija

b) (1 bod)

Vrijedi li:

$A \rightarrow BCDF$ ili $E \rightarrow BCDF$

$A \rightarrow BCDF?$

$A \rightarrow A$

refleksivnost

$A \rightarrow A \wedge A \rightarrow BCF \Rightarrow A \rightarrow ABCF$

akumulacija

$A \rightarrow ABCF \wedge C \rightarrow D \Rightarrow A \rightarrow ABCDF$

akumulacija

$A \rightarrow ABCDF \Rightarrow BCDF$

dekompozicija

FZ vrijedi

FZ je nepotpuna jer postoji pravi podskup od AE, A koji funkcijski određuje BCDF.

7. (6 bodova)

1NF: K_RADSATI = sifProjekt, mbrZap, datumRadZap

2NF:

PROJEKT = sifProjekt, nazProjekt, potrebnoSati

K_PROJEKT = sifProjekt

ZAP = mbrZap, imeZap, prezZap, sifRadMjestoZap, nazRadMjestoZap

K_ZAP = mbrZap

PROJEKTZAP = sifProjekt, mbrZap, potrebnoSatiZap

K_PROJEKTZAP = sifProjekt, mbrZap

RADSATI = sifProjekt, mbrZap, datumRadZap, brSatiRadZap

K_RADSATI = sifProjekt, mbrZap, datumRadZap

3NF:

RADMJESTO = sifRadMjestoZap, nazRadMjestoZap

K_RADMJESTO = sifRadMjestoZap

ZAPOSLENIK = mbrZap, imeZap, prezZap, sifRadMjestoZap

K_ZAPOSLENIK = mbrZap

Schema baze podataka sastoji se od relacijskih schema: PROJEKT, PROJEKTZAP, RADSATI, RADMJESTO, ZAPOSLENIK

8. (3 boda)

- Priljavno čitanje - čitanje nepotvrđenih podataka druge transakcije
- Neponovljivo čitanje – ista transakcija obavljanjem istog upita dobije iste n-torke, ali s drugačijim vrijednostima atributa
- Sablasne n-torke - ista transakcija obavljanjem istog upita dobije drugačiji rezultat
- Izgubljena izmjena – kad dvije ili više transakcija u isto vrijeme ažuriraju istu n-torku, ali se samo zadnje ažurirana n-torka pojavljuje kao nova n-torka

9. (4 boda)

a) Stablo je balansirano ukoliko je duljina puta od korijena do lista jednaka za svaki list u stablu.

b) Najgori slučaj -> minimalna popunjenost stabla

Korijen: 2 kazaljke, interni čvor $\lceil \frac{100}{2} \rceil = 50$ kazaljki, list $\lceil \frac{100-1}{2} \rceil = 50$ kazaljki.

1 razina = točno 1 čvor, **najmanje** 2 kazaljke

2 razine = 2 čvora, $2 \cdot 50 =$ **najmanje** 100 kazaljki

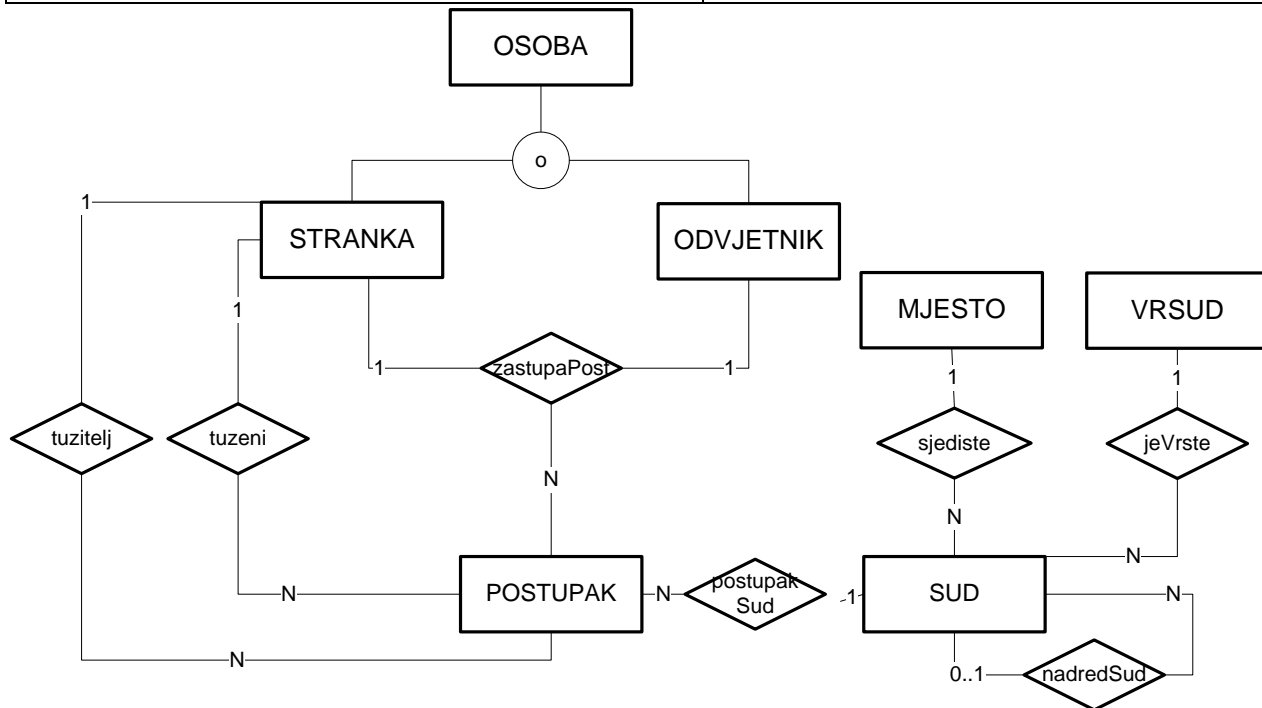
3 razine = najmanje 100 čvorova, $2 \cdot 50 \cdot 50 =$ **najmanje** 5000 kazaljki

4 razine = najmanje 5000 čvorova, $2 \cdot 50 \cdot 50 \cdot 50 =$ 250 000 kazaljki

B+ stablo reda 100 čije kazaljke u listovima pokazuju na 6000 n-torki može imati najviše 3 razine. Za takav (najgori) slučaj potrebne su 4 U/I operacije (3 za dohvat lista i 1 za dohvat bloka s podacima).

10. (8 bodova)

entiteti	veze
OSOBA = <u>OIB</u> , ime, prezime, datumRodjenja	tuzitelj = <u>jedinBrPost</u> , OIBTuzenik
ODVJETNIK = <u>OIB</u> , datUpisKomora	tuzeni = <u>jedinBrPost</u> , OIBTuzeni
STRANKA = <u>OIB</u> , brAkPostupaka	zastupaPost = <u>OIBTuzenik</u> , <u>jedinBrPost</u> , OIBodvjetnik
MJESTO = <u>postBr</u> , nazivMjesto	postupakSudu = <u>jedinBrPost</u> , sifSud
VRSD = <u>sifVrSud</u> , nazivVrSud	sjediste = <u>sifSud</u> , postBr
SUD = <u>sifSud</u> , nazivSud	jeVrste = <u>sifSud</u> , sifVrSud
POSTUPAK = <u>jedinBrPost</u> , datPokretanjaDo, datZavrsetka	nadredSud = <u>sifSud</u> , sifNadSud



11. (4 boda)

```

CREATE TABLE X (
    x1 ...
    , x2 ...
    , x1refl ... NOT NULL
    , PRIMARY KEY (x1)
    , FOREIGN KEY (x1refl) REFERENCES X(x1));
CREATE TABLE Y (
    y1 ...
    , y2 ...
    , PRIMARY KEY (y1, y2));
CREATE TABLE Z (
    z1 ...
    , z2 ...
    , z3 ...
    , PRIMARY KEY (z1));
CREATE TABLE b (
    x1 ...
    , y1 ...
    , y2 ...
    , z1 NOT NULL,
    , PRIMARY KEY (x1, y1, y2)
    , UNIQUE (y1, y2, z1)
    , FOREIGN KEY (x1) REFERENCES X(x1)
    , FOREIGN KEY (y1, y2) REFERENCES Y(y1, y2)
    , FOREIGN KEY (z1) REFERENCES Z(z1))
  
```