

## Međuispit iz Baza podataka (30 bodova)

27. travnja 2020.

**Zadaci 1. – 5. odnose se na bazu podataka prikazanu na slici 1.**

U bazi podataka se evidentiraju podatci o borbama/dvoboju timova igrača (**igrac**, **tim**, **igracTim**) u nekoj od tematskih igara (**igra**). Igre se, zbog različitih zahtjeva za opremom, igraju u jednoj od arena (**arena**). Svaka igra može biti sadržana u više paketa (**paket**), a ovisno o predviđenom trajanju igre (**paket.trajanje**, tip INTERVAL), definirane su cijene paketa po igraču (**paket.cijena**). Osnivač tima (**tim.idOsnivac**) je uvijek član tima (**igracTim**) kojeg je osnovao, a igrač može biti i član i osnivač više timova. Relacija **borba** sadrži podatke o odigranim, planiranim i borbama koje su u tijeku: početak rezerviranog termina za borbu, vrijeme završetka borbe (NULL ako borba nije završila ili kraj nije poznat); igra i predviđeno trajanje (identificiraju paket), te timovi koji sudjeluju i pobjednik ako je poznat. Pobjednik je, naravno, jedan od timova koji su sudjelovali u dvoboju ili nepoznat (NULL) ako borba nije završila ili je iz nekog razloga neevidentiran.

U bazi podataka postoje mehanizmi koji osiguravaju konzistentnost podataka u tablici **borba** (igrač ne igra u isto vrijeme u dva tima, istovremeno u istoj areni se ne odvija više borbi i slično) i o tome ne treba brinuti.

Primarni ključevi relacija su podcrtani.

**Slika 1** je ilustrativan prikaz mogućih n-torki u bazi podataka. Prikazane n-torke ne moraju odgovarati u potpunosti podatcima u bazi.

igrac

<u>idIgrac</u>	ime	prezime	datRodjenja
1	Christian	Reeves	30.03.1977
2	Lleyton	Pugh	28.06.1998
3	Tahmina	Preece	14.12.1965
...	...	...	...

tim

<u>idTim</u>	nazivTim	datVrOsnutak	idOsnivac
1	Titanmaya	3.12.2018 15:15:34	1
2	James Names	8.2.2018 12:23:00	3
3	Vileman	1.4.2017 16:00:00	3
4	A Dozendusti	2.4.2017 10:02:03	2
...	...	...	...

igracTim

<u>idIgrac</u>	<u>idTim</u>
1	1
2	1
3	2
4	2
...	...

arena

<u>idArena</u>	nazivArena
1	Primrose View Arena
2	Lakeside Arena
3	Tony's Arena
...	...

igra

<u>idIgra</u>	nazivIgra	idArena
1	Domination	1
2	Deathmatch	1
3	Battle Royale	2
4	Chernobyl Stalker	3
5	Zombie Biohazard	2
...	...	...

paket

<u>idIgra</u>	trajanje	cijena
1	00:30:00	100
1	01:00:00	150
1	02:00:00	250
2	01:00:00	170
2	02:00:00	230
3	02:00:00	200
4	01:00:00	160
5	01:30:00	180
...	...	...

borba

<u>idBorba</u>	pocetakTermin	zavrsetakBorba	idIgra	trajanje	idTim1	idTim2	idTimPobjeda
1	12.04.2019 14:00:00	12.04.2019 15:50:00	3	02:00:00	1	2	1
4	05.03.2020 18:00:00	05.03.2020 18:25:00	1	00:30:00	1	3	3
5	10.04.2020 14:00:00	10.04.2020 16:20:00	3	02:30:00	3	4	3
6	05.04.2020 12:00:00	05.04.2020 12:30:00	1	00:30:00	1	3	3
7	07.04.2020 12:00:00	07.04.2020 12:45:00	4	01:00:00	1	3	1
16	27.04.2020 17:00:00	NULL	3	02:00:00	2	4	NULL
...	...	...	...	...	...	...	...

Slika 1

## 1. (3 boda)

a)

Velikim slovima ispišite nazive svih timova koji su pobijedili u borbama. U obzir uzmite samo borbe u kojima su odigrani paketi koji sadrže igru čiji se naziv sastoji od dva ili više znakovnih nizova odvojenih razmakom, a koja se igra u areni pod nazivom 'Snowy Arena'. Dodatno ispišite i naziv te igre, predviđeno trajanje igre u odigranom paketu i tekst koji ukazuje na duljinu predviđenog trajanja igre u tom paketu: 'kratko' – predviđeno trajanje igre u tom paketu je kraće od 1 sat, 'srednje' - predviđeno trajanje igre u tom paketu je u intervalu [1 sat, 2 sata] ili 'dugo' – predviđeno trajanje igre u tom paketu je dulje od 2 sata.

U ispisu eliminirajte duplikate.

```
SELECT DISTINCT UPPER(tim.nazivTim), igra.nazivigra, borba.trajanje,
CASE
    WHEN (borba.trajanje) < '1 HOUR'::INTERVAL THEN 'kratko'
    WHEN (borba.trajanje) BETWEEN '1 HOUR'::INTERVAL AND '2 HOUR'::
    INTERVAL THEN 'srednje'
    ELSE 'dugo'
END
FROM borba NATURAL JOIN igra
        NATURAL JOIN arena
        JOIN tim ON borba.idtimpobjeda = tim.idtim
WHERE arena.nazivarena = 'Snowy Arena' and igra.nazivIgra LIKE '_% _%'
```

b)

Malim slovima ispišite nazive svih timova koji su izgubili u borbama. U obzir uzmite samo borbe u kojima su odigrani paketi koji sadrže igru čiji naziv počinje slovom 'C' i sastoji se od dva ili više znakovnih nizova odvojenih razmakom. Dodatno ispišite i naziv te igre, njezino predviđeno trajanje u odigranom paketu i tekst koji ukazuje na duljinu predviđenog trajanja igre u tom paketu: 'kratko' – predviđeno trajanje igre u tom paketu je kraće od 1 sat, 'srednje' - predviđeno trajanje igre u tom paketu je u intervalu [1 sat, 2 sata] ili 'dugo' – predviđeno trajanje igre u tom paketu je dulje od 2 sata.

U ispisu eliminirajte duplikate

```
SELECT DISTINCT LOWER(tim.nazivTim), igra.nazivigra, borba.trajanje,
CASE
    WHEN (borba.trajanje) < '1 HOUR'::INTERVAL THEN 'kratko'
    WHEN (borba.trajanje) BETWEEN '1 HOUR'::INTERVAL AND '2
    HOUR'::INTERVAL THEN 'srednje'
    ELSE 'dugo'
END
FROM borba NATURAL JOIN igra
        JOIN tim ON (borba.idTim1 = tim.idTim or borba.idTim2 =
        tim.idTim ) and borba.idtimpobjeda != tim.idtim
WHERE igra.nazivIgra LIKE 'C% _%'
```

c)

Velikim slovima ispišite nazive svih timova koji su izgubili u borbama. U obzir uzmite samo borbe u kojima su odigrani paketi koji sadrže igru čiji naziv sadrži znakovni niz 'str' bez obzira na mala i velika slova.

Dodatno ispišite i naziv te igre, njezino predviđeno trajanje u odigranom paketu, cijenu tog paketa i tekst koji ukazuje na skupoću tog paketa: 'jeftin' – cijena paketa je manja od 160 kn, 'srednje skup' – cijena paketa je u intervalu [160 kn, 200 kn>, te 'skup' – cijena paketa je 200 kn ili više.

U ispisu eliminirajte duplikate.

```
SELECT DISTINCT UPPER(tim.nazivTim), igra.nazivigra, paket.trajanje,
paket.cijena,
CASE
    WHEN (paket.cijena) < 160 THEN 'jeftin'
    WHEN (paket.cijena) >= 160 AND (paket.cijena) < 200 THEN
        'srednje skup'
    ELSE 'skup'
END
FROM borba NATURAL JOIN igra
    NATURAL JOIN paket
    JOIN tim ON (borba.idTim1 = tim.idTim or borba.idTim2 =
        tim.idTim ) and borba.idtimpobjeda != tim.idtim
WHERE lower(igra.nazivIgra) LIKE '%str%'
```

d)

Malim slovima ispišite nazive svih timova čiji naziv započinje slovom 'T', a koji su pobijedili u borbama.

U obzir uzmite samo borbe u kojima su odigrani paketi koji sadrže igru čiji naziv sadrži samo jednu riječ.

Dodatno ispišite i naziv te igre, njezino predviđeno trajanje u odigranom paketu, cijenu tog paketa i tekst koji ukazuje na skupoću tog paketa: 'jeftin' – cijena paketa je manja od 160 kn, 'srednje skup' – cijena paketa je u intervalu [160 kn, 200 kn>, te 'skup' – cijena paketa je 200 kn ili više.

U ispisu eliminirajte duplikate.

```
SELECT DISTINCT LOWER(tim.nazivTim), igra.nazivigra, paket.trajanje,
paket.cijena,
CASE
    WHEN (paket.cijena) < 160 THEN 'jeftin'
    WHEN (paket.cijena) >= 160 AND (paket.cijena) < 200 THEN
        'srednje skup'
    ELSE 'skup'
END
FROM borba NATURAL JOIN igra
    NATURAL JOIN paket
    JOIN tim ON borba.idtimpobjeda = tim.idtim
WHERE igra.nazivIgra NOT LIKE '_% %' AND tim.nazivTim LIKE 'T%'
```

## 2. (2 boda)

a)

Za sve **odigrane** pakete s početkom borbe (*pocetakTermin*) u intervalu:

[trenutak izvođenja upita - (5 mjeseci i 8h), trenutak izvođenja upita - 10h] , ispišite naziv igre koja je sadržana u paketu, predviđeno trajanje igre u tom paketu, koliko je puta taj paket odigran u zadanom intervalu te cijenu paketa po igraču.

Rezultate poredajte uzlazno po nazivu igre i predviđenom trajanju.

```
SELECT igra.nazivIgra, paket.trajanje, COUNT(*) AS brOdigranPaket, paket.cijena
FROM paket NATURAL JOIN igra
      NATURAL JOIN borba
WHERE borba.pocetakTermin BETWEEN (CURRENT_TIMESTAMP - '5 MONTHS 8 HOURS' ::
INTERVAL) AND (CURRENT_TIMESTAMP - '10 HOURS'::INTERVAL)
      and zavrsetakBorba IS NOT NULL
GROUP by igra.nazivIgra, paket.Trajanje, paket.cijena
ORDER BY nazivIgra, paket.trajanje
```

b)

Za sve **odigrane** pakete s početkom borbe (*pocetakTermin*) u intervalu:

[trenutak izvođenja upita - (7 mjeseci i 8h), trenutak izvođenja upita - 10h] , ispišite naziv igre koja je sadržana u paketu, predviđeno trajanje igre u tom paketu, koliko je puta taj paket odigran u zadanom intervalu te cijenu paketa po igraču.

Rezultate poredajte uzlazno po nazivu igre i predviđenom trajanju.

```
SELECT igra.nazivIgra, paket.trajanje, COUNT(*) AS brOdigranPaket, paket.cijena
FROM paket NATURAL JOIN igra
      NATURAL JOIN borba
WHERE borba.pocetakTermin BETWEEN (CURRENT_TIMESTAMP - '7 MONTHS 8 HOURS' ::
INTERVAL) AND (CURRENT_TIMESTAMP - '10 HOURS'::INTERVAL)
      and zavrsetakBorba IS NOT NULL
GROUP by igra.nazivIgra, paket.Trajanje, paket.cijena
ORDER BY nazivIgra, paket.trajanje
```

c)

Za sve **odigrane** borbe čiji je početak (*pocetakTermin*) bio u intervalu:

[trenutak izvođenja upita - (1 tjedan i 8h), trenutak izvođenja upita - 10h] , ispišite naziv igre koja se igrala, koliko je puta ta igra odigrana u tom intervalu (neovisno o paketu tj. trajanju i cijeni) te naziv arene u kojoj se igrala.

Rezultate poredajte uzlazno po nazivu igre.

```
SELECT igra.nazivIgra, COUNT(*) AS brojOdigranaIgra, arena.nazivarena
FROM borba NATURAL JOIN igra
      NATURAL JOIN arena
WHERE borba.pocetakTermin BETWEEN (CURRENT_TIMESTAMP - '1 WEEK 8 HOURS' ::
INTERVAL) AND (CURRENT_TIMESTAMP - '10 HOURS'::INTERVAL)
      and zavrsetakBorba IS NOT NULL
GROUP by igra.nazivIgra, arena.nazivarena
ORDER BY nazivIgra
```

d)

Za sve **odigrane** borbe čiji je početak (*pocetakTermin*) bio u intervalu:

[trenutak izvođenja upita - (5 tjedana i 8h), trenutak izvođenja upita - 10h] , ispišite naziv igre koja se igrala, koliko je puta ta igra odigrana u tom intervalu (neovisno o paketu tj. trajanju i cijeni) te naziv arene u kojoj se igrala.

Rezultate poredajte uzlazno po nazivu igre.

```
SELECT igra.nazivIgra, COUNT(*) AS brojOdigranaIgra, arena.nazivarena
FROM borba NATURAL JOIN igra
NATURAL JOIN arena
WHERE borba.pocetakTermin BETWEEN (CURRENT_TIMESTAMP - '5 WEEK 8 HOURS' ::
INTERVAL) AND (CURRENT_TIMESTAMP - '10 HOURS'::INTERVAL)
      and zavrsetakBorba IS NOT NULL
GROUP by igra.nazivIgra, arena.nazivarena
ORDER BY nazivIgra
```

### 3. (4 boda)

- a) Za sve parove timova koji su se međusobno borili u travnju, bilo koje godine, ispišite njihov naziv i sveukupni omjer pobjeda. U omjer pobjeda ulaze sve borbe koje su ti timovi **ikad** igrali, neovisno o paketu i mjesecu. **Napomena:** u relaciji **borba** je u **idTim1** uvijek upisana vrijednost manjeg identifikatora u odnosu na vrijednost upisanu u **idTim2**. Borbe završavaju isti dan kada su i započele.

Npr. ako bismo pretpostavili da relacija **borba**, sa slike koju ste dobili u opisu modela baze podataka, prikazuje sve n-torke, rezultat upita bio bi:

<i><b>nazivTim1</b></i>	<i><b>nazivTim2</b></i>	<i><b>omjerPobjeda</b></i>
Vileman	A Dozendusti	1:0
Titanmaya	James Names	1:0
Titanmaya	Vileman	1:2

```
SELECT DISTINCT t1.NazivTim as nazivTim1, t2.nazivTim as nazivTim2,
(SELECT COUNT(*) FROM borba AS borba2
WHERE borba2.idTim1 = borba.idTim1 AND borba2.idTim2 =
borba.idTim2 AND borba2.idTimPobjeda = borba.idTim1) || ':' ||
(SELECT COUNT(*)FROM borba AS borba2
WHERE borba2.idTim1 = borba.idTim1 AND borba2.idTim2 =
borba.idTim2 AND borba2.idTimPobjeda = borba.idTim2) AS
omjerPobjeda
FROM borba JOIN tim t1 ON borba.idTim1 = t1.idTim
JOIN tim t2 ON borba.idTim2 = t2.idTim
WHERE EXTRACT (MONTH FROM pocetaktermin) = 4
```

- b) Ispišite nazive timova, naziv igre i omjer pobjeda za sve parove timova koji su se međusobno borili u toj igri (neovisno o trajanju igre). U obzir uzmite samo timove koji su osnovani u 2018. godini.

**Napomena:** u relaciji **borbe** je u **idTim1** uvijek upisana vrijednost manjeg identifikatora u odnosu na vrijednost upisanu u **idTim2**.

Npr. ako bismo pretpostavili da relacija **borba**, sa slike koju ste dobili u opisu modela baze podataka, prikazuje sve n-torke, rezultat upita bio bi:

<i><b>nazivTim1</b></i>	<i><b>nazivTim2</b></i>	<i><b>nazivIgra</b></i>	<i><b>omjerPobjeda</b></i>
Vileman	A Dozendusti	Battle Royale	1:0
Titanmaya	James Names	Deathmatch	1:0
Titanmaya	Vileman	Domination	0:2
Titanmaya	Vileman	Chernobyl Stalker	1:0

```
SELECT DISTINCT t1.NazivTim as nazivTim1, t2.nazivTim as nazivTim2,
igra.nazivIgra,
  (SELECT COUNT(*) FROM borba AS borba2
   WHERE borba2.idTim1 = borba.idTim1 AND borba2.idTim2 =
   borba.idTim2
   AND borba2.idTimPobjeda = borba.idTim1 AND borba2.idIgra =
   borba.idIgra) || ':' ||
  (SELECT COUNT(*)FROM borba AS borba2
   WHERE borba2.idTim1 = borba.idTim1 AND borba2.idTim2 =
   borba.idTim2 AND borba2.idTimPobjeda = borba.idTim2 AND
   borba2.idIgra = borba.idIgra) AS omjerPobjeda
FROM borba JOIN tim t1 ON borba.idTim1 = t1.idTim
  JOIN tim t2 ON borba.idTim2 = t2.idTim
  JOIN igra on borba.idIgra = igra.idIgra
WHERE EXTRACT (YEAR FROM t1.datvrosnutak) = 2018 AND EXTRACT (YEAR FROM
t2.datvrosnutak) = 2018
```

c)

Za svaki tim osnovan u lipnju ili srpnju (neovisno o godini), a koji je barem jednom sudjelovao u nekoj borbi, ispišite njegov identifikator, naziv te broj njegovih pobjeda (**brPobjeda**) i poraza (**brPoraza**) bez obzira na vrstu igre i njezino trajanje.

```
SELECT distinct tim.idtim, tim.nazivTim,
  (SELECT COUNT(*) FROM borba AS borba2
   WHERE borba2.idTimpobjeda = tim.idTim) AS brPobjeda,
  (SELECT COUNT(*) FROM borba AS borba2
   WHERE borba2.idTimpobjeda != tim.idTim and (borba2.idTim1
   = tim.idTim or borba2.idTim2 = tim.idTim)) AS
  brPoraza
FROM tim join borba ON tim.idtim = borba.idTim1 OR tim.idTim =
borba.idTim2
WHERE EXTRACT (MONTH FROM tim.datvrosnutak) IN (6, 7)
```

d)

Za svaki tim osnovan u svibnju (neovisno o godini) i za svaku igru u kojoj je taj tim sudjelovao ispišite njegov identifikator, njegov naziv, naziv igre te broj njegovih pobjeda (**brPobjeda**) i poraza (**brPoraza**) u toj igri, neovisno o paketu i trajanju.

```
SELECT distinct tim.idtim, tim.nazivTim, igra.nazivIgra,
    (SELECT COUNT(*) FROM borba AS borba2
     WHERE borba2.idTimpobjeda = tim.idTim and borba2.idIgra =
       borba.idIgra) AS brPobjeda,
    (SELECT COUNT(*) FROM borba AS borba2
     WHERE borba2.idTimpobjeda != tim.idTim and (borba2.idTim1
       = tim.idTim or borba2.idTim2 = tim.idTim) and borba2.idIgra =
       borba.idIgra) AS brPoraza
FROM tim JOIN borba ON tim.idtim = borba.idTim1 OR tim.idTim =
  borba.idTim2
JOIN igra on borba.idIgra = igra.idIgra
WHERE EXTRACT (MONTH FROM tim.datvrosnutak) = 5
```

#### 4. (4 boda)

a)

Za sve pakete s predviđenim trajanjem duljim od sat i pol, ispišite trajanje paketa, nazive arena u kojima je predviđeno održavanje igara u tom trajanju i prosječno neiskorišteno vrijeme tih paketa **odigranih** (pocetakTermin) u tim arenama u periodu od 15. srpnja 2019. u 8 sati do 15. rujna 2019. u 20 sati (obje granice uključene).

Pod neiskorištenim vremenom smatra se vrijeme koje je (trajanjem paketa) bilo predviđeno za borbu, ali je borba završila prije kraja predviđenog vremena. Niti jedna borba ne traje dulje od vremena predviđenog trajanjem paketa.

Za sve odigrane pakete s predviđenim trajanjem duljim od sat i pol i arene u kojima se održavaju igre iz tog paketa, za koje u tom periodu ne postoje borbe (iz tog paketa), za prosječno neiskorišteno vrijeme ispišite vrijednost **NULL**.

**Napomena:** argument agregatne funkcije može biti i izraz npr. **'SELECT MIN(a+b/c-d) FROM ...'**

```
SELECT paket.trajanje, nazivArena, AVG(borba.pocetakTermin+
  borba.trajanje - borba.zavrsetakBorba) as prosjek
FROM paket
NATURAL JOIN igra
NATURAL JOIN arena
LEFT JOIN borba
  ON paket.idIgra = borba.idIgra
AND paket.trajanje = borba.trajanje
AND borba.pocetakTermin BETWEEN '15.07.2019 08:00:00' AND '15.09.2019
  20:00:00'
  AND borba.zavrsetakBorba IS NOT NULL
WHERE paket.trajanje > '1 HOUR 30 MINUTES'::INTERVAL
GROUP BY paket.trajanje, nazivArena
```

b)

Za sve pakete s predviđenim trajanjem duljim od jedan sat, ispišite trajanje paketa, nazive arena u kojima je predviđeno održavanje igara u tom trajanju i prosječno neiskorišteno vrijeme tih paketa **odigranih** (pocetakTermin) u tim arenama u periodu od 16. rujna 2019. u 10 sati i 30 minuta do 23. prosinca 2019. u 18 sati (obje granice uključene).

Pod neiskorištenim vremenom smatra se vrijeme koje je (trajanjem paketa) bilo predviđeno za borbu, ali je borba završila prije kraja predviđenog vremena. Niti jedna borba ne traje dulje od vremena predviđenog trajanjem paketa.

Za sve odigrane pakete s predviđenim trajanjem duljim od jedan sat i arene u kojima se održavaju igre iz tog paketa, za koje u tom periodu ne postoje borbe (iz tog paketa), za prosječno neiskorišteno vrijeme ispišite vrijednost **NULL**.

**Napomena:** argument agregatne funkcije može biti i izraz npr. **'SELECT MIN(a+b/c-d) FROM ...'**

```
SELECT paket.trajanje, nazivArena, AVG(borba.pocetakTermin+
borba.trajanje - borba.zavrsetakBorba) as prosjek
FROM paket
    NATURAL JOIN igra
    NATURAL JOIN arena
    LEFT JOIN borba
        ON igra.idIgra = borba.idIgra
        AND borba.pocetakTermin BETWEEN '16.09.2019 10:30:00' AND
'23.12.2019 18:00:00'
        AND paket.trajanje = borba.trajanje
        AND borba.zavrsetakBorba IS NOT NULL
WHERE paket.trajanje > '1 HOUR'::INTERVAL
GROUP BY paket.trajanje, nazivArena
```

c)

Za sve pakete s predviđenim trajanjem duljim od sat i pol, ispišite trajanje paketa, naziv igre sadržane u tom paketu i prosječno neiskorišteno vrijeme tih paketa **odigranih** (pocetakTermin) u periodu od 10. listopada 2019. u 9 sati do 10. siječnja 2020. u 16 sati i 30 minuta (obje granice uključene).

Pod neiskorištenim vremenom smatra se vrijeme koje je (trajanjem paketa) bilo predviđeno za borbu, ali je borba završila prije kraja predviđenog vremena. Niti jedna borba ne traje dulje od vremena predviđenog trajanjem paketa. Za sve pakete s predviđenim trajanjem duljim od sat i pol koji nisu odigrani u tom periodu, i za igre iz tih paketa, za prosječno neiskorišteno vrijeme ispišite vrijednost **NULL**.

**Napomena:** argument agregatne funkcije može biti i izraz npr. **'SELECT MIN(a+b/c-d) FROM ...'**

```
SELECT paket.trajanje, igra.nazivIgra, AVG(borba.pocetakTermin+
borba.trajanje - borba.zavrsetakBorba) as prosjek
FROM paket
    NATURAL JOIN igra
    LEFT JOIN borba
        ON paket.idIgra = borba.idIgra
        AND paket.trajanje = borba.trajanje
        AND borba.pocetakTermin BETWEEN '10.10.2019 09:00:00' AND
'10.1.2020 16:30:00'
        AND borba.zavrsetakBorba IS NOT NULL
WHERE paket.trajanje > '1 HOUR 30 MINUTES'::INTERVAL
GROUP BY paket.trajanje, igra.nazivIgra
```



- d) Za sve pakete s predviđenim trajanjem duljim od jedan sat, ispišite trajanje paketa, naziv igre sadržane u tom paketu i prosječno neiskorišteno vrijeme tih paketa **odigranih** (pocetakTermin) u periodu od 15. siječnja 2020. u 12 sati i 15 minuta do 15. ožujka 2020. u 22 sata (obje granice uključene).

Pod neiskorištenim vremenom smatra se vrijeme koje je (trajanjem paketa) bilo predviđeno za borbu, ali je borba završila prije kraja predviđenog vremena. Niti jedna borba ne traje dulje od vremena predviđenog trajanjem paketa. Za sve pakete s predviđenim trajanjem duljim od jedan sat koji nisu odigrani u tom periodu, i za igre iz tih paketa, za prosječno neiskorišteno vrijeme ispišite vrijednost **NULL**.

**Napomena:** argument agregatne funkcije može biti i izraz npr. **'SELECT MIN(a+b/c-d) FROM ...'**

```
SELECT paket.trajanje, igra.nazivIgra, AVG(borba.pocetakTermin+
borba.trajanje - borba.zavrsetakBorba) as prosjek
FROM paket
    NATURAL JOIN igra
    LEFT JOIN borba
        ON igra.idIgra = borba.idIgra
        AND borba.pocetakTermin BETWEEN '15.01.2020 12:15:00' AND
        '15.03.2020 22:00:00'
        AND paket.trajanje = borba.trajanje
        AND borba.zavrsetakBorba IS NOT NULL
WHERE paket.trajanje > '1 HOUR'::INTERVAL
GROUP BY paket.trajanje, igra.nazivIgra
```

## 5. (4 boda)

- a) Kroz najviše 3 naredbe kreirajte tim „Golden players“ čiji je identifikator 124. Za osnivača postavite igrača s identifikatorom 172, a za trenutak kreiranja tima postavite sadašnji trenutak. Novokreiranom timu, uz osnivača, dodajte još 4 igrača koja su do sada potrošila najmanje na borbe. U obzir nemojte uzeti igrače koji nisu ništa potrošili.

```
INSERT INTO tim VALUES (124, 'Golden players', CURRENT_TIMESTAMP, 172);

INSERT INTO igracTim VALUES (172, 124);

INSERT INTO igracTim (idIgrac, idTim)
    SELECT igracTim.idIgrac, 124
    FROM igracTim
        JOIN borba ON igracTim.idTim = borba.idTim1 OR igracTim.idTim =
        borba.idTim2
        JOIN paket ON paket.idIgra = borba.idIgra AND paket.trajanje =
        borba.trajanje
    WHERE idIgrac <> 172
    GROUP BY igracTim.idIgrac
    ORDER BY SUM(paket.cijena) ASC
    LIMIT 4
```

- b) Kroz najviše 3 naredbe kreirajte tim „Loyal players“ čiji je identifikator 125.  
Za osnivača postavite igrača s identifikatorom 82, a za trenutak kreiranja tima sadašnji trenutak.  
Novokreiranom timu, uz osnivača, dodajte još 4 igrača koja su do sada potrošila najviše na borbe.

```
INSERT INTO tim VALUES (125, 'Loyal players', CURRENT_TIMESTAMP, 82);

INSERT INTO igracTim VALUES (82, 125);

INSERT INTO igracTim (idIgrac, idTim)
  SELECT igracTim.idIgrac, 125
  FROM igracTim
JOIN borba ON igracTim.idTim = borba.idTim1 OR igracTim.idTim =
borba.idTim2
  JOIN paket ON paket.idIgra = borba.idIgra AND paket.trajanje =
borba.trajanje
  WHERE idIgrac <> 82
  GROUP BY igracTim.idIgrac
ORDER BY SUM(paket.cijena) DESC
LIMIT 4
```

- c) Kroz najviše 3 naredbe kreirajte tim „Young players“ čiji je identifikator 126.  
Za osnivača postavite igrača s identifikatorom 27, a za trenutak kreiranja tima sadašnji trenutak.  
Novokreiranom timu, uz osnivača, dodajte još 4 najmlađa igrača koja su potrošila manje od 15000 kn.

```
INSERT INTO tim VALUES (126, 'Young players', CURRENT_TIMESTAMP, 27);

INSERT INTO igracTim VALUES (27, 126);

INSERT INTO igracTim (idIgrac, idTim)
  SELECT igrac.idIgrac, 126
  FROM igracTim
  JOIN borba ON igracTim.idTim = borba.idTim1 OR igracTim.idTim =
borba.idTim2
  JOIN paket ON paket.idIgra = borba.idIgra AND paket.trajanje =
borba.trajanje
  JOIN igrac on igracTim.idigrac = igrac.idigrac
WHERE igrac.idIgrac <> 27
GROUP BY igrac.idIgrac
HAVING (SUM(paket.cijena )) < 15000
ORDER BY igrac.datrodjenja DESC
LIMIT 4
```

- d) Kroz najviše 3 naredbe kreirajte tim „Best years“ čiji je identifikator 127.  
Za osnivača postavite igrača s identifikatorom 256, a za trenutak kreiranja tima sadašnji trenutak.  
Novokreiranom timu, uz osnivača, dodajte još 4 najstarija igrača koja su potrošila više od 20000 kn.

```
INSERT INTO tim VALUES (127, 'Best years', CURRENT_TIMESTAMP, 256);

INSERT INTO igracTim VALUES (256, 127);

INSERT INTO igracTim (idIgrac, idTim)
  SELECT igrac.idIgrac, 127
  FROM igracTim
  JOIN borba ON igracTim.idTim = borba.idTim1 OR igracTim.idTim =
borba.idTim2
  JOIN paket ON paket.idIgra = borba.idIgra AND paket.trajanje =
```

```

borba.trajanje
      JOIN igrac on igracTim.idigrac = igrac.idigrac
WHERE igrac.idIgrac <> 256
GROUP BY igrac.idIgrac
HAVING (SUM(paket.cijena )) > 20000
ORDER BY igrac.datRodjenja ASC
LIMIT 4

```

## 6. (3 boda)

a)

Objasnite riječima što će biti rezultat sljedećeg izraza relacijske algebre:

$$\rho_{\text{REZULTAT(idTim, broj)}} \left( \text{idTim} \text{G}_{\text{COUNT(idIgrac)}} \left( \left( \text{igracTim} \bowtie \text{igrac} \right) \setminus \right. \right. \\ \left. \left. \sigma_{(\text{CURRENT\_DATE} - \text{datRodjenja})/365 \geq 18} \left( \text{igracTim} \bowtie \text{igrac} \right) \right) \right)$$

**Napomena:** pretpostavite da 1 godina ima 365 dana.

→ Nastat će nova relacija **rezultat** koja sadrži identifikatore timova (idTim) i broj maloljetnih članova u tim timovima (broj) za timove koji imaju bar jednog maloljetnog igrača.

b)

Objasnite riječima što će biti rezultat sljedećeg izraza relacijske algebre:

$$\rho_{\text{REZULTAT(idTim, broj)}} \left( \text{idTim, (idIgrac)} \left( \left( \text{igracTim} \bowtie \text{igrac} \right) \setminus \right. \right. \\ \left. \left. \sigma_{(\text{CURRENT\_DATE} - \text{datRodjenja})/365 < 18} \left( \text{igracTim} \bowtie \text{igrac} \right) \right) \right)$$

**Napomena:** pretpostavite da 1 godina ima 365 dana.

→ Nastat će nova relacija **rezultat** koja sadrži identifikatore (idTim), nazive timova (nazTim) i broj punoljetnih članova u tim timovima (broj) za timove koji imaju bar jednog punoljetnog igrača

c)

Objasnite riječima što će biti rezultat sljedećeg izraza relacijske algebre:

$$\left( \pi_{\text{idOsnivac}, \text{idTim}} \left( \sigma_{\text{idIgrac} = \text{idOsnivac}} (\text{igrac} \bowtie \text{tim}) \right) \right) \setminus \left( \pi_{\text{idOsnivac}, \text{idTim}} \left( \sigma_{\text{tim.idTim} < \text{idTimPobjeda}} (\text{tim} \bowtie \text{borba}) \right) \right)$$

$\text{idTim} = \text{idTim1} \vee$   
 $\text{idTim} = \text{idTim2}$

→ Nastat će nova relacija koja sadrži identifikatore timova i njihovih osnivača, koji nikada nisu izgubili ili timova koji još dosada nisu izgubili, ali trenutno igraju igru za koju još ne znamo ishod.

d)

Objasnite riječima što će biti rezultat sljedećeg izraza relacijske algebre:

$$\left( \pi_{\text{idOsnivac}, \text{idTim}} \left( \sigma_{\text{idIgrac} = \text{idOsnivac}} (\text{igrac} \bowtie \text{tim}) \right) \right) \setminus \left( \pi_{\text{idOsnivac}, \text{idTim}} \left( \sigma_{\text{tim.idTim} = \text{idTimPobjeda}} (\text{tim} \bowtie \text{borba}) \right) \right)$$

$\text{idTim} = \text{idTim1} \vee$   
 $\text{idTim} = \text{idTim2}$

→ Nastat će nova relacija koja sadrži identifikatore timova, koji nikada nisu pobijedili ili timova koji još dosada nisu pobijedili, ali trenutno igraju igru za koju još ne znamo ishod. Uz to se ispisuju i identifikatori osnivača tih timova.

## 7. (2 boda)

a)

Objasnite na primjerima Armstrongove aksiome koristeći funkcijske zavisnosti koje vrijede na relacijama **igra** i **borba**.

PRAVILO REFLEKSIVNOSTI:

**idIgra, trajanje -> idIgra**

PRAVILO UVEĆANJA:

Ako vrijedi: **idBorba -> trajanje**

Onda vrijedi: **idBorba, pocetakTermin -> trajanje**

PRAVILO TRANZITIVNOST

Ako vrijedi: **idBorba -> idIgra**

I vrijedi: **idIgra -> nazivIgra**

Onda vrijedi i: **idBorba -> nazivIgra**

b)

Objasnite na primjerima Armstrongove aksiome koristeći funkcijske zavisnosti koje vrijede na relacijama **igra** i **arena**.

PRAVILO REFLEKSIVNOSTI :

**idIgra -> idIgra**

PRAVILO UVEĆANJA :

Ako vrijedi: **idIgra -> idArena**

Onda vrijedi: **idIgra, nazivIgra -> idArena**

PRAVILO TRANZITIVNOST

Ako vrijedi: **idIgra -> idArena**

I vrijedi: **idArena -> nazivArena**

Onda vrijedi i: **idIgra -> nazivArena**

c)

Objasnite na primjerima pravilo unije i dekompozicije te aksiom o tranzitivnosti koristeći funkcijske zavisnosti koje vrijede na relacijama **igra** i **arena**.

PRAVILO UNIJE :

Ako vrijedi: **idIgra** -> **nazivIgra**

I vrijedi **idIgra** -> **idArena**

Onda vrijedi i: **idIgra** -> **nazivIgra, idArena**

PRAVILO DEKOMPOZICIJE :

Ako vrijedi: **idIgra** -> **nazivIgra, idArena**

Onda vrijedi i: **idIgra** -> **idArena**

PRAVILO TRANZITIVNOST :

Ako vrijedi: **idIgra** -> **idArena**

I vrijedi: **idArena** -> **nazivArena**

Onda vrijedi i: **idIgra** -> **nazivArena**

d)

Objasnite na primjerima pravilo unije, dekompozicije i pseudotranzitivnosti koristeći funkcijske zavisnosti koje vrijede na relacijama **igra** i **borba**.

Napomena:  $K_{igra} = idIgra$ ;  $K1_{borba} = idBorba$ ;  $K2_{borba} = idIgra$ , **pocetakTermin**

PRAVILO UNIJE:

Ako vrijedi: **idBorba** -> **pocetakTermin**

I vrijedi **idBorba** -> **zavrsetakBorba**

Onda vrijedi i: **idBorba** -> **pocetakTermin, zavrsetakBorba**

PRAVILO DEKOMPOZICIJE:

Ako vrijedi: **idBorba** -> **pocetakTermin, zavrsetakBorba**

Onda vrijedi i: **idBorba** -> **pocetakTermin**

PRAVILO PSEUDOTRANZITIVNOSTI:

Ako vrijedi: **idIgra** -> **idArena**

I vrijedi: **idArena, pocetakTermin** -> **trajanje**

Onda vrijedi: **pocetakTermin, idigra** -> **trajanje**

## 8. (2 boda)

a)

Napisati SQL naredbe koje će kreirati najmanji mogući broj indeksa za učinkovito obavljanje (pomoću B<sup>+</sup> stabla) svih dolje navedenih upita. Indeksima dodijelite imena i1, i2, ..., in.

a)

- 1) SELECT \* FROM borba WHERE pocetakTermin :: DATE BETWEEN '01.1.2020' AND '10.04.2020' ;
- 2) SELECT \* FROM borba WHERE trajanje < '02:00:00' ;
- 3) SELECT \* FROM borba WHERE idTim2 = 1 and idTim1 = 3 ;
- 4) SELECT \* FROM borba ORDER BY pocetakTermin ASC, trajanje DESC;
- 5) SELECT \* FROM borba ORDER BY idTim1, idTim2;
- 6) SELECT \* FROM borba ORDER BY idBorba, trajanje;
- 7) SELECT \* FROM borba WHERE idTim2 = 3;
- 8) SELECT \* FROM borba WHERE idTim1 = 3;

→ CREATE INDEX i1 ON borba (trajanje)	2
CREATE INDEX i2 ON borba (idTim1, idTim2)	3 i 5 i 8
CREATE INDEX i3 ON borba (pocetakTermin, trajanje DESC)	1 i 4
CREATE INDEX i4 ON borba (idBorba, trajanje)	6
CREATE INDEX i5 ON borba (idTim2)	7

b)

Što će se dogoditi s B-stablama prilikom unosa nove n-torke u relaciju borba? Ukoliko dolazi do promjene nekog B-stabla, precizno navedite ime pripadnog indexa.

→ Informacije o n-torki će se unijeti u sva B-stabla (indeksi od i1 do i5) izgrađena nad tablicom borba

Što će se dogoditi s B-stablama ako promijenimo vrijednost atributa *pocetakTermin* u n-torci s *idBorba* = 3. Ukoliko dolazi do promjene nekog B-stabla, precizno navedite ime pripadnog indexa.

→ izmjena vrijednosti atributa *pocetakTermin* jedne n-torke rezultirat će brisanjem i dodavanjem informacija o tom zapisu u B-stablu koje odgovara indeksu i3

**b)**

Napisati SQL naredbe koje će kreirati najmanji mogući broj indeksa za učinkovito obavljanje (pomoću B<sup>+</sup> stabla) svih dolje navedenih upita. Indeksima dodijelite imena i1, i2, ..., in.

a)

- 1) SELECT \* FROM borba WHERE zavrsetakBorba :: DATE BETWEEN  
    '10.1.2020' AND '27.04.2020' ;
- 2) SELECT \* FROM borba WHERE trajanje < '01:30:00' ;
- 3) SELECT \* FROM borba WHERE idTim1 = 1 OR idTim2 = 3 ;
- 4) SELECT \* FROM borba ORDER BY trajanje DESC, zavrsetakBorba DESC;
- 5) SELECT \* FROM borba ORDER BY idTim2, idTim1;
- 6) SELECT \* FROM borba ORDER BY idBorba, trajanje;
- 7) SELECT \* FROM borba WHERE idTim2 = 3;
- 8) SELECT \* FROM borba WHERE idTim1 = 3;

CREATE INDEX i1 ON borba (zavrsetakBorba)	1
CREATE INDEX i2 ON borba (idTim2, idTim1)	3 i 5 i 7
CREATE INDEX i3 ON borba ( trajanje DESC, zavrsetakBorba DESC)	2 i 4
CREATE INDEX i4 ON borba (idBorba, trajanje)	6
CREATE INDEX i5 ON borba (idTim1)	8

b)

Što će se dogoditi s B-stablama prilikom unosa nove n-torke u relaciju borba? Ukoliko dolazi do promjene nekog B-stabla, precizno navedite ime pripadnog indexa.

➔ Informacije o n-torki će se unijeti u sva B-stabla (indeksi od i1 do i5) izgrađena nad tablicom borba

Što će se dogoditi s B-stablama ako promijenimo vrijednost atributa *zavrsetakBorba* u n-torci s *idBorba* = 5. Ukoliko dolazi do promjene nekog B-stabla, precizno navedite ime pripadnog indexa.

➔ izmjena vrijednosti atributa *zavrsetakBorba* jedne n-torke rezultirat će brisanjem i dodavanjem informacija o tom zapisu u B-stablu koje odgovara indeksu i1 i i3.



c)

Napisati SQL naredbe koje će kreirati najmanji mogući broj indeksa za učinkovito obavljanje (pomoću B<sup>+</sup> stabla) svih dolje navedenih upita. Indeksima dodijelite imena i1, i2, ..., in.

a)

- 1) SELECT \* FROM borba ORDER BY pocetakTermin ASC, trajanje DESC;
- 2) SELECT \* FROM borba WHERE idTim2 = 10;
- 3) SELECT \* FROM borba ORDER BY idTim1, idTim2;
- 4) SELECT \* FROM borba WHERE pocetakTermin :: DATE BETWEEN '01.03.2020' AND '01.04.2020' ;
- 5) SELECT \* FROM borba WHERE trajanje < '01:00:00' ;
- 6) SELECT \* FROM borba WHERE idTim2 = 1 and idTim1 = 3 ;
- 7) SELECT \* FROM borba ORDER BY idIgra, trajanje;
- 8) SELECT \* FROM borba WHERE idTim1 = 17;

➔ CREATE INDEX i1 ON borba (pocetakTermin , trajanje DESC)	1 i 4
CREATE INDEX i2 ON borba (idTim2)	2
CREATE INDEX i3 ON borba (idTim1, idTim2)	3 i 6 i 8
CREATE INDEX i4 ON borba (trajanje)	5
CREATE INDEX i5 ON borba (idIgra, trajanje)	7

b)

Što će se dogoditi s B-stablama prilikom unosa nove n-torke u relaciju borba? Ukoliko dolazi do promjene nekog B-stabla, precizno navedite ime pripadnog indexa.

➔ Informacije o n-torki će se unijeti u sva B-stabla (indeksi od i1 do i5) izgrađena nad tablicom borba

Što će se dogoditi s B-stablama ako promijenimo vrijednost atributa *pocetakTermin* u n-torci s *idBorba* = 13. Ukoliko dolazi do promjene nekog B-stabla, precizno navedite ime pripadnog indexa.

➔ izmjena vrijednosti atributa *pocetakTermin* jedne n-torke rezultirat će brisanjem i dodavanjem informacija o tom zapisu u B-stablu koje odgovara indeksu i1

**d)**

Napisati SQL naredbe koje će kreirati najmanji mogući broj indeksa za učinkovito obavljanje (pomoću B<sup>+</sup> stabla) svih dolje navedenih upita. Indeksima dodijelite imena i1, i2, ..., in.

a)

- 1) `SELECT * FROM borba WHERE zavrsetakBorba :: DATE BETWEEN '27.1.2020' AND '27.04.2020' ;`
- 2) `SELECT * FROM borba WHERE trajanje < '01:30:00' ;`
- 3) `SELECT * FROM borba WHERE idTim1 = 5 and idTimPobjeda = 5;`
- 4) `SELECT * FROM borba ORDER BY trajanje DESC, zavrsetakBorba DESC;`
- 5) `SELECT * FROM borba ORDER BY idTimPobjeda, idTim1;`
- 6) `SELECT * FROM borba WHERE idTimPobjeda = 3;`
- 7) `SELECT * FROM borba ORDER BY idIgra, idTimPobjeda;`
- 8) `SELECT * FROM borba WHERE idTim1 = 3;`

<code>CREATE INDEX i1 ON borba (zavrsetakBorba)</code>	1
<code>CREATE INDEX i2 ON borba (idTimPobjeda, idTim1)</code>	3 i 5 i 6
<code>CREATE INDEX i3 ON borba ( trajanje DESC, zavrsetakBorba DESC)</code>	2 i 4
<code>CREATE INDEX i4 ON borba (idIgra, idTimPobjeda)</code>	7
<code>CREATE INDEX i5 ON borba (idTim1)</code>	8

b)

Što će se dogoditi s B-stablama prilikom unosa nove n-torke u relaciju borba? Ukoliko dolazi do promjene nekog B-stabla, precizno navedite ime pripadnog indexa.

➔ Informacije o n-torki će se unijeti u sva B-stabla (indeksi od i1 do i5) izgrađena nad tablicom borba

Što će se dogoditi s B-stablama ako promijenimo vrijednost atributa *zavrsetakBorba* u n-torci s *idBorba* = 15. Ukoliko dolazi do promjene nekog B-stabla, precizno navedite ime pripadnog indexa.

➔ izmjena vrijednosti atributa *zavrsetakBorba* jedne n-torke rezultirat će brisanjem i dodavanjem informacija o tom zapisu u B-stablu koje odgovara indeksu i1 i i3.

9. (2 boda)

a)

Koliko se zapisa može pohraniti u tablicu nad kojom je izgrađeno B+-stablo reda 5 dubine 6 s najmanjom popunjenošću?

**Obrazložite rješenje**, obavezno navodeći izračunati broj kazaljki za **svaku vrstu čvora** stabla.

$$K=2, \check{C}=3, L=2$$

$$2 \times 3 \times 3 \times 3 \times 3 \times 2 = 324$$

b)

Koliko se zapisa može pohraniti u tablicu nad kojom je izgrađeno B+-stablo reda 5 dubine 6 s najvećom popunjenošću?

**Obrazložite rješenje**, obavezno navodeći izračunati broj kazaljki za **svaku vrstu čvora** stabla.

$$K=5, \check{C}=5, L=4$$

$$5 \times 5 \times 5 \times 5 \times 5 \times 4 = 12500$$

c)

Koliko se zapisa može pohraniti u tablicu nad kojom je izgrađeno B+-stablo reda 6 dubine 5 s najmanjom popunjenošću?

**Obrazložite rješenje**, obavezno navodeći izračunati broj kazaljki za **svaku vrstu čvora** stabla.

$$K=2, \check{C}=3, L=3$$

$$2 \times 3 \times 3 \times 3 \times 3 \times 3 = 162$$

d)

Koliko se zapisa može pohraniti u tablicu nad kojom je izgrađeno B+-stablo reda 6 dubine 5 s najvećom popunjenošću?

**Obrazložite rješenje**, obavezno navodeći izračunati broj kazaljki za **svaku vrstu čvora** stabla.

$$K=6, \check{C}=6, L=5$$

$$6 \times 6 \times 6 \times 6 \times 5 = 6480$$

## 10. (4 boda)

a)

Online učionica za maturante nudi pomoć maturantima u pripremi za državnu maturu. Aplikacija u svojoj bazi podataka evidentira podatke o predmetima za koje se nudi priprema, suradnicima koji drže pripreme, njihovoj struci te satnici. Relacijska shema **NASTAVA** se sastoji od sljedećih atributa:

<b>oznPred</b>	– jedinstveni identifikator predmeta
<b>nazPred</b>	– naziv predmeta
<b>oznSurad</b>	– jedinstveni identifikator suradnika
<b>imeSurad</b>	– ime suradnika
<b>prezimeSurad</b>	– prezime suradnika
<b>oznStruke</b>	– jedinstveni identifikator struke suradnika
<b>nazivStruke</b>	– naziv struke suradnika
<b>cijenaSata</b>	– cijena sata predavanja
<b>brojSati</b>	– broj odrađenih sati suradnika na predmetu

Jedan suradnik može držati pripreme iz više predmeta. Svi suradnici na nekom predmetu ne moraju odraditi isti broj sati. Pretpostavite da jedna osoba može biti samo jedne struke. Svi suradnici iste struke imaju jednaku cijenu sata koja je neovisna o predmetu.

Odrediti ključ relacijske sheme **NASTAVA** tako da ona bude u 1NF, a zatim **postupno** normalizirati relacijsku shemu na 2NF i 3NF.

- **Normalizacija na 1NF**

Vrijede sljedeće FZ:

- $\text{oznPred} \rightarrow \text{nazPred}$
- $\text{oznSurad} \rightarrow \text{imeSurad}, \text{prezimeSurad}, \text{oznStruke}, \text{nazivStruke}, \text{cijenaSata}$
- $\text{oznPred}, \text{oznSurad} \rightarrow \text{brojSati}$

**NASTAVA** = { oznPred, nazPred, oznSurad, imeSurad, prezimeSurad, oznStruke, nazivStruke, cijenaSata, brojSati }

$K = \{ \text{oznPred}, \text{oznSurad} \}$

- **Normalizacija na 2NF**

**PREDMET** = oznPred, nazPred

$K = \{ \text{oznPred} \}$

**SURADNIK** = oznSurad, imeSurad, prezimeSurad, oznStruke, nazivStruke, cijenaSata

$K = \{ \text{oznSurad} \}$

**ODRAĐENISATI** = oznPred, oznSurad, brojSati

$K = \{ \text{oznPred}, \text{oznSurad} \}$

- **Normalizacija na 3NF**

**STRUKA** = oznStruke, nazivStruke, cijenaSata

$K = \{ \text{oznStruke} \}$

**SURADNIK2** = oznSurad, imeSurad, prezimeSurad, oznStruke

$K = \{ \text{oznSurad} \}$

- **Konačna shema baze podataka sadrži sljedeće relacijske sheme:**

**PREDMET** = oznPred, nazPred

K = { oznPred }

**STRUKA** = oznStruke, nazivStruke, cijenaSata

K = { oznStruke }

**SURADNIK2** = oznSurad, imeSurad, prezimeSurad, oznStruke

K = { oznSurad }

**ODRAĐENISATI** = oznPred, oznSurad, brojSati

K = { oznPred, oznSurad }

b)

U bazi podataka evidentiraju se podatci o vrtićima u Zagrebu i vanjskim stručnim suradnicima (npr. psiholog, logoped, edukacijski rehabilitator i sl.) koji po potrebi dolaze raditi u određeni vrtić. Relacijska shema **VRTIĆ** se sastoji od sljedećih atributa:

<b>oznVrtic</b>	– jedinstveni identifikator vrtića
<b>nazVrtic</b>	– naziv vrtića
<b>oznSurad</b>	– jedinstveni identifikator suradnika
<b>imeSurad</b>	– ime suradnika
<b>prezimeSurad</b>	– prezime suradnika
<b>oznStruke</b>	– jedinstveni identifikator struke suradnika
<b>nazivStruke</b>	– naziv struke suradnika
<b>cijenaSata</b>	– cijena sata
<b>brojSati</b>	– broj odrađenih sati suradnika u vrtiću

Jedan suradnik može raditi u više vrtića. Svi suradnici u nekom vrtiću ne moraju odraditi isti broj sati. Pretpostavite da jedna osoba može biti samo jedne struke. Svi suradnici iste struke imaju jednaku cijenu sata koja je neovisna o vrtiću u kojem rade.

Odrediti ključ relacijske sheme **VRTIĆ** tako da ona bude u 1NF, a zatim postupno normalizirati relacijsku shemu na 2NF i 3NF.

- **Normalizacija na 1NF**

Vrijede sljedeće FZ:

- oznVrtic → nazVrtic
- oznSurad → imeSurad, prezimeSurad, oznStruke, nazivStruke, cijenaSata
- oznVrtic, oznSurad → brojSati

**VRTIĆ** = { oznVrtic, nazVrtic, oznSurad, imeSurad, prezimeSurad, oznStruke, nazivStruke, cijenaSata, brojSati }

K = { oznVrtic, oznSurad }

- **Normalizacija na 2NF**

**VRTIĆ** = oznVrtic, nazVrtic

K = { oznVrtic }

**SURADNIK** = oznSurad, imeSurad, prezimeSurad, oznStruke, nazivStruke, cijenaSata

K = { oznSurad }

**ODRAĐENISATI** = oznVrtic, oznSurad, brojSati

K = { oznVrtic , oznSurad }

- **Normalizacija na 3NF**

**STRUKA** = oznStruke, nazivStruke, cijenaSata

K = { oznStruke}

**SURADNIK2** = oznSurad, imeSurad, prezimeSurad, oznStruke

K = { oznSurad}

- **Konačna shema baze podataka sadrži sljedeće relacijske sheme:**

**VRTIĆ** = oznVrtic, nazVrtic

K = { oznVrtic }

**STRUKA** = oznStruke, nazivStruke, cijenaSata

K = { oznStruke}

**SURADNIK2** = oznSurad, imeSurad, prezimeSurad, oznStruke

K = { oznSurad}

**ODRAĐENISATI** = oznVrtic, oznSurad, brojSati

K = { oznVrtic , oznSurad }

c)

Baza podataka **APARTMANI** evidentira podatke o izgradnji kompleksa apartmana na području Vira. Bilježe se podatci o osobama koje su sudjelovale u izgradnji pojedinog apartmana, njihovoj struci (električar, dizajner interijera...) te satnici. Relacijska shema **APARTMANI** se sastoji od sljedećih atributa:

<b>oznApartman</b>	– jedinstveni identifikator apartmana
<b>nazApartman</b>	– naziv apartmana
<b>oznRadnik</b>	– jedinstveni identifikator radnika
<b>imeRadnik</b>	– ime radnika
<b>prezimeRadnik</b>	– prezime radnika
<b>oznStruke</b>	– jedinstveni identifikator struke radnika
<b>nazivStruke</b>	– naziv struke radnika
<b>cijenaSata</b>	– cijena sata utrošenog na izgradnji apartmana
<b>brojSati</b>	– broj odrađenih sati radnika na izgradnji apartmana

Jedna osoba može sudjelovati u izgradnji više apartmana u tom kompleksu. Sve osobe koje sudjeluju u izgradnji nekog apartmana ne moraju odraditi isti broj sati u njegovoj gradnji. Pretpostavite da jedna osoba može biti samo jedne struke. Sve osobe iste struke imaju jednaku cijenu sata koja je neovisna o apartmanu kojeg grade. Odrediti ključ relacijske sheme APARTMANI tako da ona bude u 1NF, a zatim **postupno** normalizirati relacijsku shemu na 2NF i 3NF.

- **Normalizacija na 1NF**

Vrijede sljedeće FZ:

- oznApartman → nazApartman
- oznRadnik → imeRadnik, prezimeRadnik, oznStruke, nazivStruke, cijenaSata
- oznApartman, oznRadnik → brojSati

**APARTMANI** = { oznApartman, nazApartman, oznRadnik, imeRadnik, prezimeRadnik, oznStruke, nazivStruke, cijenaSata, brojSati }

K = { oznApartman, oznRadnik }

- **Normalizacija na 2NF**

**APARTMAN** = oznApartman, nazApartman

K = { oznApartman }

**RADNIK** = oznRadnik, imeRadnik, prezimeRadnik, oznStruke, nazivStruke, cijenaSata

K = { oznRadnik }

**ODRAĐENISATI** = oznApartman, oznRadnik, brojSati

K = { oznApartman, oznRadnik }

- **Normalizacija na 3NF**

**STRUKA** = oznStruke, nazivStruke, cijenaSata

K = { oznStruke }

**SURADNIK2** = oznRadnik, imeRadnik, prezimeRadnik, oznStruke

K = { oznRadnik }

- **Konačna shema baze podataka sadrži sljedeće relacijske sheme:**

**APARTMAN** = oznApartman, nazApartman

K = { oznApartman }

**STRUKA** = oznStruke, nazivStruke, cijenaSata

K = { oznStruke }

**SURADNIK2** = oznRadnik, imeRadnik, prezimeRadnik, oznStruke

K = { oznRadnik }

**ODRAĐENISATI** = oznApartman, oznRadnik, brojSati

K = { oznApartman, oznRadnik }

d)

Firma *Vjenčanja d.o.o.* pomaže mladencima u organizaciji vjenčanja u Hrvatskoj. U svojoj bazi podataka **VJENCANJA** firma evidentira podatke o vjenčanjima koje organizira. Bilježe se i podatci o zaposlenicima koji sudjeluju u organizaciji vjenčanja, njihovoj struci (cvječar, fotograf...) te satnici. Relacijska shema **VJENCANJA** se sastoji od sljedećih atributa:

<b>oznVjencanje</b>	– jedinstveni identifikator vjenčanja
<b>datumVjencanja</b>	– datum vjenčanja
<b>oznZaposlenik</b>	– jedinstveni identifikator zaposlenika
<b>imeZaposlenik</b>	– ime zaposlenika
<b>prezimeZaposlenik</b>	– prezime zaposlenika
<b>oznStruke</b>	– jedinstveni identifikator struke zaposlenika
<b>nazivStruke</b>	– naziv struke zaposlenika
<b>cijenaSata</b>	– cijena sata utrošenog na organizaciju
<b>brojSati</b>	– broj odrađenih sati zaposlenika na vjenčanju

Jedan zaposlenik može sudjelovati u organizaciji više vjenčanja. Svi zaposlenici na nakon vjenčanju ne moraju odraditi isti broj sati. Pretpostavite da jedan zaposlenik može biti samo jedne struke. Svi zaposlenici iste struke imaju jednaku cijenu sata koja je neovisna o vjenčanju u čijoj organizaciji sudjeluju.

Odrediti ključ relacijske sheme **VJENCANJA** tako da ona bude u 1NF, a zatim **postupno** normalizirati relacijsku shemu na 2NF i 3NF.

- **Normalizacija na 1NF**

Vrijede sljedeće FZ:

- $\text{oznVjencanja} \rightarrow \text{datumVjencanja}$
- $\text{oznZaposlenik} \rightarrow \text{imeZaposlenik}, \text{prezimeZaposlenik}, \text{oznStruke}, \text{nazivStruke}, \text{cijenaSata}$
- $\text{oznVjencanja}, \text{oznZaposlenik} \rightarrow \text{brojSati}$

**VJENCANJE** = { oznVjencanja, datumVjencanja, oznZaposlenik, imeZaposlenik, prezimeZaposlenik, oznStruke, nazivStruke, cijenaSata, brojSati }

$K = \{ \text{oznVjencanja}, \text{oznZaposlenik} \}$

- **Normalizacija na 2NF**

**VJENCANJE** = oznVjencanja, datumVjencanja

$K = \{ \text{oznVjencanja} \}$

**ZAPOSLENIK** = oznZaposlenik, imeZaposlenik, prezimeZaposlenik, oznStruke, nazivStruke, cijenaSata

$K = \{ \text{oznZaposlenik} \}$

**ODRAĐENISATI** = oznVjencanja, oznZaposlenik, brojSati

$K = \{ \text{oznVjencanja}, \text{oznZaposlenik} \}$

- **Normalizacija na 3NF**

**STRUKA** = oznStruke, nazivStruke, cijenaSata

$K = \{ \text{oznStruke} \}$

**ZAPOSLENIK 2** = oznZaposlenik, imeZaposlenik, prezimeZaposlenik, oznStruke

$K = \{ \text{oznZaposlenik} \}$

- **Konačna shema baze podataka sadrži sljedeće relacijske sheme:**

**VJENCANJE** = oznVjencanja, datumVjencanja

$K = \{ \text{oznVjencanja} \}$

**STRUKA** = oznStruke, nazivStruke, cijenaSata

$K = \{ \text{oznStruke} \}$

**ZAPOSLENIK2** = oznZaposlenik, imeZaposlenik, prezimeZaposlenik, oznStruke

$K = \{ \text{oznZaposlenik} \}$

**ODRAĐENISATI** = oznVjencanja, oznZaposlenik, brojSati

$K = \{ \text{oznVjencanja}, \text{oznZaposlenik} \}$