

1. (5 bodova) Objasnite što su prozori (particije i okviri) u kontekstu SQL upitnog jezika. Pri tom objasnite u kojem su odnosu particije i okviri te osnovne razlike među njima.
Navedite primjer SQL upita kojim ćete demonstrirati upotrebu particija i okvira. Vodite računa o tome da se problem kojeg upit rješava ne bi trebao moći riješiti bez primjene prozora i funkcija za rad s prozorima (npr. pomoću GROUP BY).
Skicirajte shemu i sadržaj relacije (ili više njih) koje se koriste u upitu te rezultat upita.

2. (6 bodova) U zadatku se pretpostavlja korištenje SQL standarda.

U bazi podataka pohranjuju se podaci o zaposlenicima i projektima u nekom poduzeću. Prikazanim naredbama kreirani su tipovi **projektT** i **osobaT**. Također su kreirane i napunjene tipizirane tablice **projektOR** i **osobaOR** temeljene na tim tipovima. Imena stupaca koji sadrže jedinstvene identifikatore objekata u tipiziranim tablicama su **OIDprojekt** odnosno **OIDosoba**. U atributu **osobaT.projekti** evidentirani su projekti na kojima je osoba radila.

```
CREATE TYPE projektT AS (  
  sifProjekt INTEGER,  
  nazProjekt CHAR(50))  
INSTANTIABLE NOT FINAL  
REF IS SYSTEM GENERATED;
```

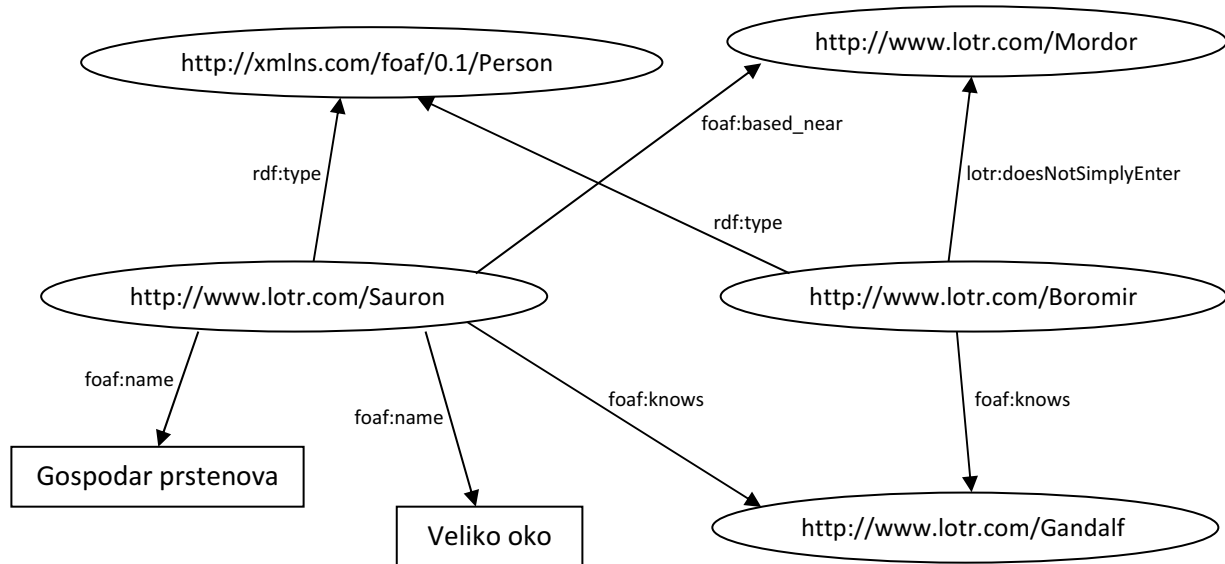
```
CREATE TYPE osobaT AS (  
  sifOsoba    INTEGER,  
  ime         CHAR(50),  
  prezime     CHAR(75),  
  projekti    REF(projektT) MULTISSET)  
INSTANTIABLE NOT FINAL  
REF IS SYSTEM GENERATED;
```

- a) Napisati SQL naredbu kojom će se za svaki projekt čiji naziv sadrži tekst "Croatian Post" ispisati naziv projekta i broj zaposlenika koji su radili samo na tom projektu. Zadatak riješiti bez korištenja podupita.
- b) Tablica **osobaOR** sadrži zapis o zaposleniku sa šifrom 3 (**sifOsoba**), koji do sada nije radio niti na jednom projektu. Napisati SQL naredbu kojom će se evidentirati da je taj zaposlenik radio na projektu sa šifrom 2 (**sifProjekt**).
3. (3 boda) Navedite različite načine na koje možemo definirati pojam vremena i vremenske podatke u sustavima za upravljanje bazama podataka. Što je to **chronon**?
4. (3 boda) Objasnite mehanizam kvoruma kod ostvarivanja konzistencije. Navedite primjere i za čitanje i za pisanje.
5. (5 bodova) Što je *combinable reducer*? Objasnite na primjeru.

6. (4 boda) Napišite sve trojke od kojih se sastoji sljedeći graf:

Napomena: radi jednostavnosti, možete koristiti Notation3 (N-Triples, Turtle) format i sljedeće prefikse:

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix lotr: <http://www.lotr.com/> .
```



7. (4 boda) Nadopunite sljedeći SPARQL upit nad semantičkim izvorom DBpedia:

Pretpostavite upotrebu sljedećih prefiksa

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>
PREFIX dbpprop: <http://dbpedia.org/property/>
```

Dohvatiti engleski("en") naziv i broj stanovnika svih gradova u Hrvatskoj koji u nazivu sadrže slovo "S".
Rezultate sortirati po broju stanovnika od većeg prema manjem.

```
SELECT _____
WHERE {
    ?grad a dbpedia-owl:City .
    ?grad dbpedia-owl:country <http://dbpedia.org/resource/Croatia> .
    ?grad rdfs:label ?naziv .
    ?grad dbpprop:population ?brojSt

    _____
    _____
}
```

Rješenja:

1. **(5 bodova)** Prozori (particije i okviri) su tranzijentni skupovi n-torki koji dijele relaciju ili međurezultate SQL upita na logičke podskupove n-torki. Particija može sadržavati više okvira. Okvir ne može izlaziti izvan granica particije. Particija ne može biti pomična dok okvir može.

2. a) (4 boda)

```
SELECT p.OIDprojekt -> nazProjekt, COUNT(*)
  FROM osobaOR o, UNNEST (o.projekti) AS p(OIDprojekt)
 WHERE p.OIDprojekt -> nazProjekt LIKE '%Croatian Post%'
    AND CARDINALITY (o.projekti) = 1
 GROUP BY p.OIDprojekt, p.OIDprojekt -> nazProjekt;
```

b) (2 boda)

```
UPDATE osobaOR SET projekti = MULTISET(SELECT p.OIDprojekt FROM p.projektor
                                         WHERE p.sifProjekt = 2)
 WHERE sifOsoba = 3;
```

3. (3 boda) Slajd 10 u predavanju o vremenskim bazama.

4. (3 boda) Vidjeti predavanja

5. (5 bodova) Vidjeti predavanja

6. (4 boda)

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix lotr: <http://www.lotr.com/> .
<http://www.lotr.com/Sauron>
  a foaf:Person ;
  foaf:name "Gospodar Prstenova" ;
  foaf:name "Veliko Oko" ;
  foaf:knows <http://www.lotr.com/Gandalf> ;
  foaf:based_near <http://www.lotr.com/Mordor> .
<http://www.lotr.com/Boromir>
  a foaf:Person ;
  foaf:knows <http://www.lotr.com/Gandalf> ;
  lotr:doesNotSimplyEnter <http://www.lotr.com/Mordor> .
```

7. (4 boda)

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>
PREFIX dbpprop: <http://dbpedia.org/property/>
SELECT ?naziv ?brojSt
WHERE {
  ?grad a dbpedia-owl:City .
  ?grad dbpedia-owl:country <http://dbpedia.org/resource/Croatia> .
  ?grad rdfs:label ?naziv .
  ?grad dbpprop:populationTotal ?brojSt
  FILTER(LANG(?naziv) = "en")
  FILTER(REGEX(?naziv,"S"))
}
ORDER BY DESC(?brojSt)
```