Napredni modeli i baze podataka – ispit 06.02.2017.

U zadacima 1-3 pretpostavlja se korištenje PostgreSQL SUBP-a i baze podataka sa slike 1.

Baza podataka je namijenjena pohrani podataka o nominacijama i osvojenim filmskim nagradama Oscar.

film

filmId	***nazivFilm***	***...***	***opisFilm***	***allTSV***
100	Captain Fantastic	…	In the forests of the Pacific Northwest, a father devoted to raising his six kids...	'captain':1A 'devot':12B 'fantast':2A 'father':11B 'forest':5B 'kid':17B 'northwest':9B 'pacif':8B 'rais':14B
101	La La land	…	A jazz pianist falls for an aspiring actress in Los Angeles…	'actress':11B 'angel':14B 'aspir':10B 'fall':7B 'jazz':5B **'la':1A,2A** 'land':3A 'los':13B 'pianist':6B"
102	Hacksaw ridge	…	American Army Medic Desmond T. Doss, who served during the Battle of Okinawa...	'american':3B 'armi':4B 'battl':13B 'desmond':6B 'doss':8B 'hachsaw':1A 'medic':5B 'okinawa':15B
…	…	…		

kategorija

katId	***nazivKat***	***...***
100	Directing	…
101	Actress in a leading role	…
102	Actor in a leading role	…
…	…	…

osoba

osobaId	***prezime***	***ime***	***...***
100	Gosling	Ryan	
101	Mortensen	Viggo	
…		…	

nominacija

filmId	***katId***	***osobaId***	***godina***	***...***
100	102	101	2017	
101	101	150	2017	
102	102	100	2017	
…	…	…	…	…

slika 1

1. **(4 boda)** Korištenjem funkcionalnosti PostgreSQL SUBP napišite SQL naredbu kojom će se ispisati godina i naziv filma i
   - ukupan broj nominacija filma (***brNomFilm***),
   - ukupan broj svih nominacija u toj godini (***brNomGod)***
   - rang filma u godini obzirom na broj nominacija (***rangFilmGod***)

godina	***nazivFilm***	***brNomFilm***	***brNomGod***	***rangFilmGod***
…	…	…	…	…
2017	La La Land	12	118	1
2017	Arrival	7	118	2
2017	Hacksaw Ridge	5	118	3
…	…	…	…	…

2. **(4 boda)** Da bi podržao efikasnu pretragu filmova temeljem naziva i opisa filma administrator sustava je predvidio u tablici ***film*** atribut ***allTSV.*** U nastavku je primjer INSERT naredbe za tu tablicu.

```
INSERT INTO film
VALUES (100, 'Captain Fantastic',
        'In the forests of the Pacific Northwest, a father devoted to raising his six kids...',
        setweight(to_tsvector('english', 'Captain Fantastic'), 'A') ||
        setweight(to_tsvector('english','In the forests of the Pacific Northwest, a father devoted to
                                        raising his six kids...'), 'B'));
```

   Pri kojoj vrsti pretrage teksta ovaj atribut može pridonijeti efikasnosti? Objasnite zbog čega. Također, objasnite koje je značenje i uloga slova A i B i brojeva koji se uz ta slova nalaze u sadržaju atributa ***allTSV***. Objasnite na koji način utječu na pretragu.

3. **(5 bodova)** Na temelju tablice ***nominacija*** napišite M/R upit koji će za svaku godinu vratiti identifikator filma s najviše nominacija (ako ih ima više, vratiti bilo kojeg).
   Pretpostavite istu arhitekturu (ponašanje) M/R-a kao kod MongoDB sustava.

4. **(4 boda)** Što je konzistentno raspršenje? Po čemu se razlikuje od „običnog“? Gdje nalazi primjenu u NoSQL sustavima, navedite primjer.

5. **(3 boda)** Što je svojstvo trajnosti (eng. *durability*)?
Komentirajte svojstvo trajnosti u relacijskim i NoSQL sustavima.
Što je i čemu služi oslabljivanje trajnosti (eng. relaxing durability)? Navedite primjer.

6.

a) **(4 boda)** Korištenjem OWL i Turtle sintakse napišite naredbe za definiranje ontologije kojom će biti modeliran segment iz svijeta književnosti. Ontologijom je potrebno definirati sve konstrukte (TBox) potrebne za opisivanje sljedećeg:
   - pisac ***<http://dbpedia.org/resource/George_R._R._Martin>***
     - se zove „George Raymond Martin“.
     - rođen je u ***<http://dbpedia.org/resource/New_Jersey>*** čiji je naziv „New Jersey“.
     - njegovo djelo ***<http://dbpedia.org/resource/A_Song_of_Ice_and_Fire>*** ima naslov "A Song of Ice and Fire" na engleskom, a "Une chanson de glace et de feu" na francuskom jeziku

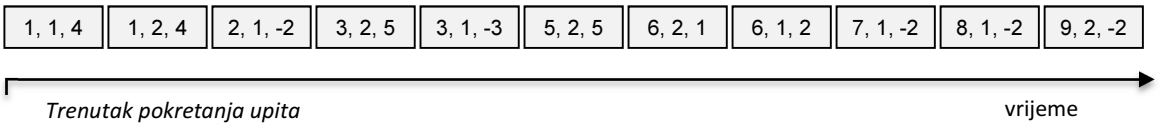
   Pri uvođenju klasa u vlastitu ontologiju možete preuzeti neku od sljedećih klasa:
   Person, Writer, PopulatedPlace, AdministrativeRegion iz <http://dbpedia.org/ontology/>
   Place, AdministrativeArea iz <http://www.schema.org/>
   Book, Publication iz <http://dbpedia.org/class/yago>
   Za predloženu ontologiju napišite Turtle izraze (ABox) kojima ćete opisati gornje izjave vezane uz pisca, njegovo mjesto rođenja i djelo koje je napisao.

b) **(2 boda)** Koristeći predloženu ontologiju napišite SPARQL upit kojim ćete ispisati ukupan broj književnih djela (knjiga) koje je napisao pisac ***<http://dbpedia.org/resource/George_R._R._Martin>***. Možete pretpostaviti da je ***http://example.org/*** SPARQL endpoint pomoću kojeg možete pristupiti vašim podacima.

7.

a) **(1 bod)** Ukratko pojasnite ulogu vremenskih prozora kod kontinuiranih upita u sustavima za upravljanje tokovima podataka. Koja je razlika između *logičkih* i *fizičkih* vremenskih prozora?

b) **(3 boda)** Teleskop Egzo-ZPR lansiran je u Zemljinu orbitu s ciljem otkrivanja novih planeta sličnih Zemlji. U tu svrhu, teleskop vrši mjerenja svjetlosnog intenziteta s dvije zvijezde, a za svaku koristi po jedan senzor. Podaci s teleskopa pristižu na ulaz SUTP-a u obliku jednog toka podataka u kojem je svaka n-torka opisana sljedećom shemom: `<vrijeme_primitka, id_senzora, intenzitet>`. Pretpostavimo da je vrijeme primitka prikazano brojem sekundi od trenutka pokretanja kontinuiranog upita, ID senzora poprima vrijednost 1 ili 2, a intenzitet je cjelobrojna vrijednost u intervalu <-10,10>.
U SUTP-u je pokrenut kontinuirani upit koji u stvarnom vremenu analizira pristigle podatke i na izlaz šalje stanje intenziteta za svaki od senzora u obliku toka podataka s istom shemom kao kod ulaznog toka. Od trenutka pokretanja upita u SUTP su pristigle sljedeće n-torke:



Počevši od početnog stanja (oba senzora imaju vrijednost intenziteta 0) izračunajte i prikažite izlazni tok podataka, ako je:
a) Kontinuirani upit baziran na vremenskom prozoru koji promatra n-torke pristigle u zadnje 4 sekunde, a izlazni tok modelira prema *Turnstile modelu* toka podataka
b) Kontinuirani upit baziran na vremenskom prozoru koji promatra zadnje 3 pristigle n-torke, a izlazni tok modelira prema *Reset modelu* toka podataka

Vremensku oznaku izlaznog toka modelirajte proizvoljno (npr. 1 za prvi par n-torki, 2 za drugi par n-torki itd.)

Rješenja:

1.

```
SELECT godina, film.nazivFilm,
       COUNT(nominacija.filmId),
       SUM(COUNT(nominacija.filmId)) OVER (PARTITION BY godina),
       rank() OVER (PARTITION BY godina ORDER BY  COUNT(nominacija.filmId) DESC)
  FROM nominacija
  JOIN film ON nominacija.filmId = film.filmId
  GROUP BY godina, film.nazivFilm, nominacija.filmID
```

2.

film.allTSV pridonosi efikasnosti pretrage temeljem morfologije, semantike i sintakse jezika.
Pri pretrazi temeljem morfologije, semantike i sintakse jezika upit i tekst nad kojim se upit obavlja se predstavljaju u normaliziranom obliku (uklonjene stop riječi, riječi svedene na normalizirani oblik, prepoznati sinonimi,…) tj. kao lista leksema (u Postgresu. kao TSVector tip podatka). Ideja je prepoznati kao jednake riječi koje imaju jednak normalizirani oblik. Ako se sadržaj nad kojim se obavljaju upiti (***film.nazivFilm, film.sadrzaj***) redundantno, unaprijed pohrani u TSVector obliku, taj će se atribut moći koristiti pri pretrazi i to će ubrzati pretragu. Alternativa je svaki put pri evaluaciji upita svoditi ***film.nazivFilm*** i ***film.sadrzaj*** pomoću to_tsvector funkcije na normalizirani oblik. To je naravno sporije rješenje.

Pri određivanju ranga, relevantnosti „dokumenta“ za dani upit različitim dijelovima „dokumenta“ ima smisla dodijeliti različite težine – npr. veću težinu naslovu, a manju opisu filma. U PostgreSQL-u se na to može utjecati pomoću polja weights float4[] Pretpostavljene vrijednosti: {0.1, 0.2, 0.4, 1.0} = {D, C, B, A}.
Broj uz riječ u ***film.allTSV*** ukazuje na redni broj riječi u originalnom tekstu, a slovo na činjenicu pojavljuje li se riječ u naslovu (A) ili opisu (B). Za film „La La land“ riječ „La“ se pojavljuje u naslovu kao prva i kao druga riječ. Budući da je naslovu filma dana težina 'A' uz riječ „La“ se pojavljuju oznake 1A i 2A.
Funkcija koja računa relevantnost dokumenta će veći rang pridijeliti onim „dokumentima“ kod kojih npr. 5 traženih leksema pronađe u naslovu nego li onim dokumentima kod kojih te iste lekseme pronađe u sadržaju.

3.

Bilo je nužno koristiti dictionary u reduce funkciji, ili ulančati dvije MR faze.

M/R

```
map (k, v) {
   var dict = {};
   dict[v.filmId] = 1;
   emit(v.godina, dict);
}
reduce (k, vlist) {
   var dict = {};
   foreach (v in vlist) {
      foreach(key in v.keys) {
         if (dict[key]) {
            dict[key] += v[key];
         } else {
            dict[key] = v[key];
         }
      }
   }
   return dict;
}
finalize (k, vlist) {
   var maxId = 0;
   var v = vlist.first();
   foreach(key in v.keys) {
         if (maxId == 0 || v[key] > v[maxId]) {
            maxId = key;
         }
      }
   return maxId;
}
```

4. vidi predavanja
5. vidi predavanja

6.
a)
```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix dbr: <http://dbpedia.org/resource/> .
@prefix dbo: <http://dbpedia.org/ontology/> .
@prefix schema: <http://www.schema.org/> .
@prefix yago: <http://dbpedia.org/class/yago/> .
@base        <http://exampple.org/vieontology.owl/> .

<http://exampple.org/literature.owl/> rdf:type owl:Ontology ;
                                      owl:imports dbo: .
#    Classes
dbo:Person rdf:type owl:Class .
dbo:Writer rdf:type owl:Class ;
           rdfs:subClassOf dbo:Person .
dbo:PopulatedPlace rdf:type owl:Class .
yago:Publication rdf:type owl:Class .
yago:Book         rdf:type owl:Class ;
                 rdfs:subClassOf yago:Publication .
#    Object Properties
:wasBornIn  rdf:type    owl:ObjectProperty ;
            rdfs:domain dbo:Writer ;
            rdfs:range  dbo:PopulatedPlace .
:hasWritten rdf:type    owl:ObjectProperty ;
            rdfs:domain dbo:Writer ;
            rdfs:range  schema:Book .
#    Data properties
:writerName rdf:type owl:DatatypeProperty ;
            rdfs:domain dbo:Writer ;
            rdfs:range xsd:string .
:placeName  rdf:type owl:DatatypeProperty ;
            rdfs:domain dbo:PopulatedPlace ;
            rdfs:range xsd:string .
:bookTitle  rdf:type owl:DatatypeProperty ;
            rdfs:domain dbo:Book ;
            rdfs:range xsd:string .
#    Individuals
dbr:New_Jersey             rdf:type dbo:PopulatedPlace;
                           :placeName  "New Jersey"@en.
dbr:A_Song_of_Ice_and_Fire rdf:type dbo:Book;
                           :bookTitle  "A Song of Ice and Fire"@en,
                                       "Une chanson de glace et de feu"@fr.
dbr:George_R._R._Martin    rdf:type dbo:Writer;
                           :writerName "George R. R. Martin";
                           :wasBornIn dbr:New_Jersey ;
                           :hasWritten dbr:A_Song_of_Ice_and_Fire.
```
b)
```
@prefix dbr:    <http://dbpedia.org/resource/>
@prefix dbo:    <http://dbpedia.org/ontology/>
@base           <http://exampple.org/literature.owl/>
SELECT COUNT(?bookURI) AS ?noOfBooks
  FROM <http://exampple.org/>
 WHERE {
       dbr:George_R._R._Martin rdf:type dbo:Writer;
                               :hasWritten ?bookURI.
```

```
}
7.
```

S obzirom na to da je tok podataka neograničeni, potencijalno beskonačan niz n-torki :

Iz perspektive sustava – nije praktično pohraniti cijeli tok podataka
Iz perspektive korisnika – noviji podaci su često zanimljiviji/korisniji

Stoga se koriste vremenski prozori za *ograničenje dosega kontinuiranih upita*.

Logički ili vremenski bazirani prozori *(time-based windows)*

- Definirani vremenskim intervalom
- Npr. promatramo zadnjih 15 minuta podataka

Fizički *(count-based* ili *touple-based)*

- Definirani brojem n-torki koje se promatraju
- Npr. fizički *(count-based)* klizajući prozor koji pohranjuje samo zadnjih 100 n-torki

8.

a) U svakom ciklusu generiranja novih izlaznih n-torki (po svakom isteku vremenskog prozora) promatramo sljedeće ulazne n-torke (plavo i crveno):

1, 1, 4	1, 2, 4	2, 1, -2	3, 2, 5	3, 1, -3	5, 2, -5	6, 2, 1	7, 1, 2	8, 1, -4	9, 1, -2	9, 2, -2

Izlazni tok:

1, 1, -1	1, 2, 9	2, 1, -3	2, 2, 5

b) U svakom ciklusu generiranja novih izlaznih n-torki (po svakom isteku vremenskog prozora) promatramo sljedeće ulazne n-torke (plavo i crveno):

1, 1, 4	1, 2, 4	2, 1, -2	3, 2, 5	3, 1, -3	5, 2, -5	6, 2, 1	7, 1, 2	8, 1, -4	9, 1, -2	9, 2, -2

Izlazni tok:

1, 1, -2	1, 2, 4	2, 1, -3	2, 2, -5	3, 1, -4	3, 2, 1