**Završni ispit iz Naprednih modela i baza podataka**

27.01.2014.

---

1. **(5 bodova)** Objasnite pojam približnog pretraživanja teksta (Fuzzy Text Search) te navedite barem tri vrste algoritama koje se koriste pri približnom pretraživanju teksta.

2. **(5 bodova)** Objasnite razliku između stanja i događaja u kontekstu vremenskih baza podataka. Koristeći samo SQL (bez dodatne podrške za upravljanje vremenom), napišite primjer naredbe za stvaranje relacije stanja i relacije događaja.

3. **(6 bodova)** Objasniti distribucijske modele u NoSQL sustavima i navesti kako utječu na čitanje odnosno pisanje podataka. Navedite jedan primjer.

4. **(10 bodova)** Ulazni skup podataka je (pojednostavljena) log datoteka nekog web servera. Primjer sadržaja, uz objašnjenje značenja atributa u zapisu, je prikazan na slici:

| IP adresa | Vrijeme | Zahtjev | Response Code | Bytes | Client |
|---|---|---|---|---|---|
| 161.53.200.56 | [09/Dec/2013:11:38:15 +0100] | "GET /dummy/reports.css HTTP/1.1" 304 -"Internet Explorer 6.0" | | | |
| 161.53.150.29 | [09/Dec/2013:11:59:58 +0100] | "HEAD /robots.txt HTTP/1.0" 404 -"-" | | | |
| 161.53.100.37 | [09/Dec/2013:14:39:20 +0100] | "GET /fer/foo HTTP/1.1" 401 1373 "Mozilla Firefox v25" | | | |
| 161.53.100.37 | [09/Dec/2013:14:39:24 +0100] | "GET /favicon.ico HTTP/1.1" 200 7274"Mozilla Firefox v25" | | | |
| 161.53.100.37 | [09/Dec/2013:14:40:14 +0100] | "GET /fer/ HTTP/1.1" 200 2773"Mozilla Firefox v25" | | | |
| 161.53.100.37 | [09/Dec/2013:14:40:14 +0100] | "GET /icons/blank.gif HTTP/1.1" 200 148 "Mozilla Firefox v25" | | | |
| 161.53.100.37 | [09/Dec/2013:14:40:14 +0100] | "GET /icons/back.gif HTTP/1.1" 200 216 "Mozilla Firefox v25" | | | |
| 161.53.100.37 | [09/Dec/2013:14:40:14 +0100] | "GET /icons/compressed.gif HTTP/1.1" 200 1038 "Mozilla Firefox v25" | | | |
| 161.53.100.19 | [09/Dec/2013:14:41:30 +0100] | "GET /fer/ HTTP/1.1" 401 1367"Chrome v31.0.1650.63" | | | |
| ... | | | | | |

Potrebno je napisati MapReduce algoritam kojim će se dobiti ukupan broj zahtjeva i isporučenih bajtova za uspješne zahtjeve (HTTP Response Code je 200) po različitim klijentima (Internet preglednicima)u 2014. godini.
Objasnite (nacrtajte za npr. tri čvora) kako se obavlja vaš MR algoritam.

*Kod „implementacije" možete koristiti bilo koji programski jezik ili pseudo kod. Pritom, ako koristite pseudo kod, ne smijete prelaziti granice mogućnosti modernih programskih jezika, odnosno smijete koristiti različite strukture podataka (npr. hashmap) i funkcije (npr. sort, splitWords), ali ne smijete koristiti izmišljene napredne funkcije primjenjive upravo na ovaj problem (npr. zbrojiSveBajtoveZahtjevasKodom200()).*

5. **(6 bodova)** Na temelju zadanih podataka u N3/Turtle formatu, nacrtajte RDF graf:

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

<http://www.peanuts.com/charlie.brown>
        a                       foaf:Person ;
        foaf:firstName "Charlie" ;
        foaf:knows              <http://www.peanuts.com/linus.van.pelt> ;
        foaf:knows              <http://www.peanuts.com/snoopy> .
<http://www.peanuts.com/linus.van.pelt>
        rdf:type                foaf:Person ;
        foaf:knows              <http://www.peanuts.com/charlie.brown> ;
        foaf:knows              <http://www.peanuts.com/snoopy> ;
        pean:owns               "blanket" .
<http://www.peanuts.com/snoopy>
        foaf:name               "Snoopy" ;
        pean:owns               "doghouse" ;
        pean:likesToSleepOn     "blanket" ;
        pean:likesToSleepOn     "doghouse" .
```

6. **(4 boda)** Nadopunite sljedeće SPARQL upite 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/>
```

   (a) **(2 boda)** Koliko regija u Njemačkoj ima više od 5 milijuna stanovnika?

```
SELECT _____________________________
WHERE {
   ?regija a dbpedia-owl:Region .
   ?regija dbpedia-owl:country <http://dbpedia.org/resource/Germany> .
   ?regija dbpprop:population ?brojSt

   _____________________________
 }
```

   (b) **(2 boda)** Da li u Turskoj postoji naseljeno mjesto čiji engleski naziv je "Batman"?

```
ASK _____________________________
WHERE {
   ?mjesto a dbpedia-owl:PopulatedPlace .
   ?mjesto dbpedia-owl:country <http://dbpedia.org/resource/Turkey> .

   ?mjesto foaf:name _____________________________
 }
```

7. **(4 boda)** Koje su odlike činjeničnih i dimenzijskih tablica.
   Navedite primjer zvjezdastog spoja.

Rješenja:
**1.** Tehnika pronalaženja znakovnog niza koji se približno podudara s traženim uzorkom
Kvaliteta podudaranja se mjeri brojem operacija koje je potrebno obaviti nad znakovnim nizom da bi se u potpunosti podudario s traženim uzorkom
Vrste algoritama:

- algoritmi temeljeni na udaljenosti između znakovnih nizova
  - Hamming,
  - Levenshtein – PostgreSQL: postoji f-ja Levenshtein
- Q-Gram algoritmi
  - slični znakovni nizovi imaju mnogo zajedničkih Q-Gram-ova (podskupovi znakovnog niza duljine Q) - PostgreSQL: podržano, tekst rastavlja na trigrame similarity(text, text), operator %
- algoritmi koji traže riječi koje se slično izgovaraju
  - Soundex, Metaphone algoritam – PostgreSQL postoje i soundex i metaphone funkcije

2

- **Stanja** opisuju činjenice vezane uz neki objekt u bazi podataka koje su istinite u nekom vremenskom intervalu ili periodu. Te se činjenice ne smatraju točnima izvan pridruženog perioda.
- **Događaji** opsuju činjenice vezane uz neki objekt u bazi podataka koje su se dogodile u određenom trenutku (*chrononu*) i nemaju trajanje.

Relacija stanja:

```
CREATE TABLE zaposlenik (
 id         INTEGER,
 ime      CHAR(20),
 prezime CHAR(20),
 placa    NUMBER,
 vrijediOd  TIMESTAMP
 vrijediDo  TIMESTAMP
);
```

Relacija događaja

```
CREATE TABLE ispit (
 sifStudent        INTEGER,
 sifNastavnik     INTEGER,
 sifPredmet       INTEGER,
 ocjena,
 datum             DATE
);
```

3. Pogledati predavanja, to je teoretsko pitanje, slajdovi 7-10 u „NoSQL 2/3“

4.

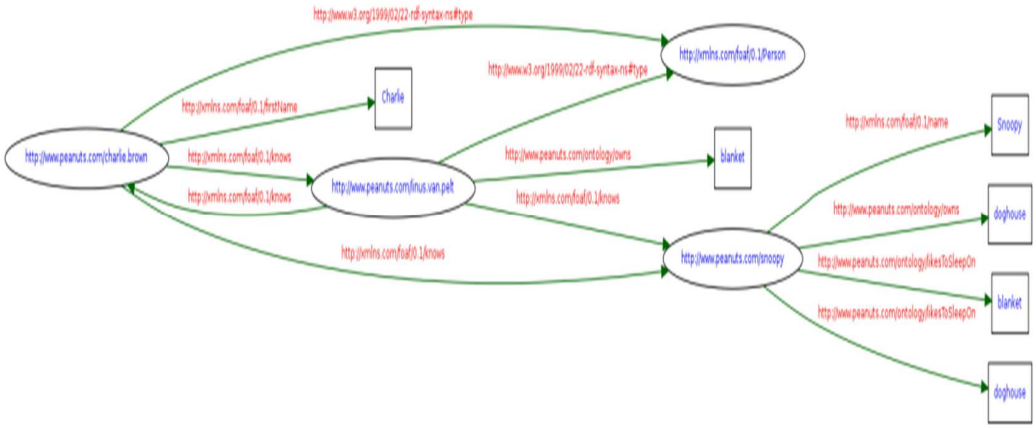
Uvažava se map koji prima jedan redak ili n redaka:

```
map (k, v)
  for line = v.readLine()
     words = line.splitWords();
     if (w[3] == “200” && year(w[1]) == 2014)
        emit (w[5], {cnt: 1, bytes: w[4]})
```

Dobra su i rješenja koja u map funkciji agregiraju (najčešće koristeći dictionary/hashmap structure) podatke na razini preglednika, odnosno emitiraju samo jednu informaciju po pregledniku.

```
reduce (k, v)
  sum = {cnt: 0, bytes: 0}
  foreach req in v
    sum.cnt += req.cnt
    sum.bytes += req.bytes
  return sum
```

5. zadatak




6. zadatak

a)

```
SELECT COUNT(?regija)
WHERE {
   ?regija a dbpedia-owl:Region .
   ?regija dbpedia-owl:country <http://dbpedia.org/resource/Germany> .
   ?regija dbpprop:population  ?brojSt
   FILTER(?brojSt > 5000000)
 }
```

b)

```
ASK
WHERE {
   ?mjesto a dbpedia-owl:PopulatedPlace .
   ?mjesto dbpedia-owl:country <http://dbpedia.org/resource/Turkey> .
   ?mjesto foaf:name "Batman"@en
 }
```

7.

**Činjenična tablica:**

- Odgovara procesu koji se prati u skladištu podataka
- Skladište sadrži **N** činjeničnih tablica
- Sadrži dvije skupine brojčanih atributa:
  - ključevi dimenzijskih tablica
  - mjere (engl. *measures*)
- Često je (gotovo) identična odgovarajućoj tablici u relacijskom izvoru podataka (ali uz prešifriranje ključeva)
- velik broj zapisa ($10^5$, $10^6$, $10^7$, ...)
- jedan redak (n-torka) ne zauzima puno prostora (normalizirana tablica, numerički atributi)

**Dimenzijske tablice:**

- Predstavlja subjekt/objekt koji sudjeluje u procesu koji se prati u skladištu podataka
- Objašnjavaju činjenice pohranjene u činjeničnoj tablici odnosno opisuju kontekst, npr.: koji student ja kada položio predmet kod kojeg nastavnika
- Skladište sadrži N dimenzijskih tablica
- Dimenzijska tablica nije normalizirana
- Često nastaje spajanjem više relacijskih tablica
- Ima jednostavan ključ (cjelobrojni tip):
  - složeni ključevi izvorišnih tablica se obavezno zamjenjuju jednostavnim (generiranim) surogatnim ključem
  - i jednostavni ključ zamijeniti generiranim surogatnim ključem
- mali broj zapisa ($<10^5$, većinom $<10^2$)
- veliko zauzeće prostora po jednom retku (nenormalizirana, npr. 30 atributa, od toga 15-tak znakovni nizovi: 15 * 4 + 15 * 100 = 1560 byte)

**Primjer zvjezdastog spoja:**