

Upute za 1. projekt iz predmeta Napredni modeli i baze podataka

Pretraživanje teksta u relacijskim bazama podataka i napredni SQL akademska godina 2019./2020.

1 Korištenje PostgreSQL-a i testni skup podataka

Potrebno je izraditi jednostavnu tražilicu dokumenata, koji su pohranjeni u relacijskoj bazi podataka, te omogućiti analizu postavljenih uvjeta pretrage uz primjenu znanja koja ste iz ove teme stekli na predmetu. Kao SUBP potrebno je koristiti PostgreSQL.

Prema uputama objavljenima u datoteci **Instalacija PostgreSQL i dvdRent baze podataka - upute.pdf** u repozitoriju **Zimski semestar 2019./2020.** ⇒ **Instalacija PostgreSQL i dvdRent baze- upute** instalirajte PostgreSQL SUBP.

Nakon instalacije, SQL naredbe nad PostgreSQL sustavom možete obavljati pomoću nekog od SQL editora prilagođenih za rad s PostgreSQL-om npr. PgAdmin (koji ste vjerojatno instalirali zajedno s instalacijom SUBP).

Pomoću *pgAdmina* se spojite na PostgreSQL SUBP. Kreirajte novu bazu podataka (desni klik mišem na Databases – Create - Database; Name: *poželji*, Owner: *postgres*). Odaberite bazu s kojom želite raditi i otvorite prozor za izvođenje upita (Tools – Query Tool).

Za podešavanje formata prikaza datumskog tipa podataka na *dd.mm.yyyy* izvedite sljedeću naredbu:

```
SET DateStyle = 'German, DMY';
```

Budući da će vam pri izradi i testiranju aplikacije trebati tekstualni podaci, možete koristiti podatke o filmovima koje smo priredili za potrebe ovog projekta.

Pomoću datoteke *movie.sql* iz repozitorija **Zimski semestar 2019./2020.** ⇒ **Projekti** ⇒ **1. projekt - Pretraživanje teksta i napredni SQL** možete stvoriti tablicu *movie* s 1000 zapisa o filmovima. Naredbe je samo potrebno obaviti pomoću PgAdmina.

Tablica *movie* sadrži 4 tekstualna atributa na engleskom jeziku: naslov filma, kategorije filma, sažetak i opis. Shema tablice nije dovoljna da zadovolji sve zahtjeve projekta. Slobodnu ju proširite.

2 Opis funkcionalnosti aplikacije

Potrebno je izraditi aplikaciju (desktop/web/mobilnu/...) koja će omogućiti:

1. Unos tekstualnog sadržaja u PostgreSQL bazu podataka proizvoljne sheme (minimalne ali prikladne za demonstraciju svih zadataka iz projekta)
2. Pretragu u bazi podataka pohranjenog tekstualnog sadržaja
3. Analizu postavljenih upita u zadanom vremenskom periodu

Izgled aplikacije je proizvoljan kao i izbor programskih tehnologija pomoću kojih ćete ju izraditi.

2.1 Unos tekstualnog sadržaja

Zbog nepostojanja cjelovite podrške za potpunu pretragu teksta za hrvatski jezik koristite tekstove na engleskom jeziku.



Text search in RDB & advanced SQL

Menu

Add

Search

Analysis

Title:

The big sick

Categories:

Family; Comedy

Summary:

Pakistan-born comedian Kumail Nanjiani and grad student Emily Gardner fall in love but struggle as their cultures clash. When Emily contracts a mysterious illness, Kumail

Description:

Kumail (Kumail Nanjiani), in the middle of becoming a budding stand-up comedian, meets Emily (Zoe Kazan). Meanwhile, a sudden illness sets in forcing Emily to be put into a medically-induced coma. Kumail must navigate being a comedian, dealing with tragic illness, and placating his family's desire to let them fix him up with a spouse, while contemplating and figuring out who he really is and what he truly believes.

Add

2.2 Pretraga tekstualnog sadržaja pohranjenog u bazi podataka

Potrebno je omogućiti dohvat dokumenata koji sadrže traženi uzorak (uzorke) u normaliziranom obliku (based on morphology&semantic) pri čemu treba dohvatiti i dokumente koji sadrže bilo koji oblik riječi iz teksta pretrage. Ova vrsta pretrage podrazumijeva normaliziranje riječi, uklanjanje stop riječi i sl.

U aplikaciji je potrebno:

podržati povezivanje zadanih uzoraka pretrage logičkim operatorom

a) AND

b) OR

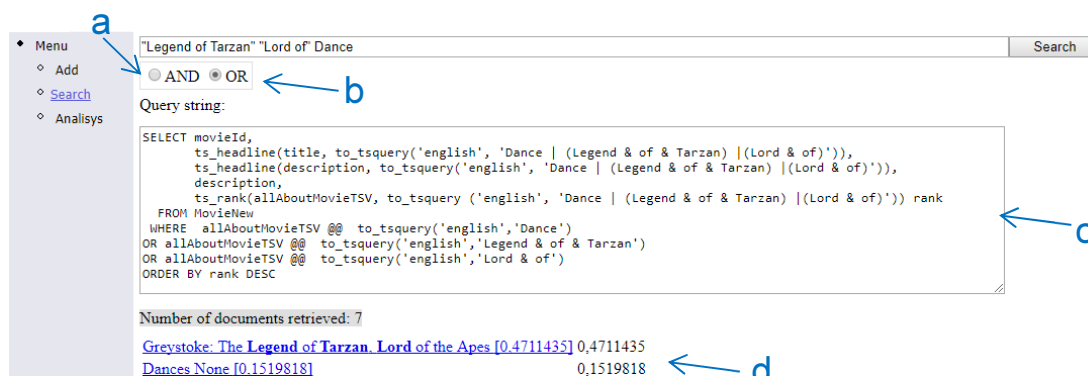
c) prikazati SQL upit kojim su temeljem korisnikovog uvjeta pretrage dohvaćeni relevantni dokumenti

d) prikazati informativne podatke o dohvaćenim dokumentima s podebljanim riječima temeljem kojih je dokument kvalificiran kao rezultat te **rang** dokumenta. Primijetite da su na donjoj slici riječi Legend Tarzan i Lord podebljane. Proučite kako rade funkcije za rangiranje dokumenata implementirane u PostgreSQL-u (preporučujemo sadržaj na <http://shisaa.jp/postset/postgresql-full-text-search-part-3.html>) te u aplikaciji upotrijebite funkciju za rangiranje koju smatrate prikladnom s parametrima koje smatrate prikladnima.

e) podržati traženje fraza (znakovni niz naveden unutar navodnika) i "jednostavnih" riječi kombiniranih logičkim operatorima AND i OR. Donja slika prikazuje jedan od načina kako se može pronaći zapise koji sadrže bilo jednu od fraza "Legend of Tarzan" ili "Lord of" ili riječ Dance

f) pri unosu uvjeta pretrage implementirati tzv. *autocomplete* tj. nuditi korisniku npr. do 5 smislenih dovršetaka uvjeta pretrage. Osmislite vlastiti algoritam kojim ćete to podržati. Pri implementaciji ove funkcionalnosti možete smatrati da korisniku treba nuditi dovršetke obzirom na riječi/nizove riječi upotrijebljene u atributu *movie.summary*. Iako je ovu funkcionalnost moguće implementirati na mnogo načina radi jednostavnosti odlučite se za jedan od sljedeća dva:

- implementacija tzv. *search tree* (vidi <https://www.postgresql.org/docs/9.6/static/ltree.html>).
- korištenje neke od fuzzy text search mogućnosti PostgreSQL-a (vidi npr. <https://www.sitepoint.com/awesome-autocomplete-trigram-search-in-rails-and-postgresql/>)



Menu

- ◊ Add
- ◊ Search
- ◊ Analysis

Query string: "Legend of Tarzan" "Lord of" Dance

Query string: ☐ AND ☒ OR

Query string:

```
SELECT movieId,  
ts_headline(title, to_tsquery('english', 'Dance | (Legend & of & Tarzan) |(Lord & of)'),  
ts_headline(description, to_tsquery('english', 'Dance | (Legend & of & Tarzan) |(Lord & of)'),  
description,  
ts_rank(allAboutMovieTSV, to_tsquery('english', 'Dance | (Legend & of & Tarzan) |(Lord & of)')) rank  
FROM MovieNew  
WHERE allAboutMovieTSV @@ to_tsquery('english', 'Dance')  
OR allAboutMovieTSV @@ to_tsquery('english', 'Legend & of & Tarzan')  
OR allAboutMovieTSV @@ to_tsquery('english', 'Lord & of')  
ORDER BY rank DESC
```

Number of documents retrieved: 7

Greystoke: The Legend of Tarzan, Lord of the Apes [0.4711435] 0,4711435

Dances None [0.1519818] 0,1519818

Modelirajte bazu podataka tako da, koliko god možete, ubrzate pretragu - npr. dodatna pohrana normaliziranog teksta, kreiranje specijalnih indeksa (koliko god i koji god mogu ubrzati pretragu). Za realizaciju pretrage i *autocomplete*-a odaberite operatore odnosno funkcije PostgreSQL-a koje smatrate najprikladnijima. Izbor morate biti u stanju argumentirati.

3. Analiza postavljanih upita u zadanom vremenskom periodu

Da bi analiza bila moguća potrebno je bilježiti upite koje korisnici postavljaju. To treba uzeti u obzir pri dizajniranju baze podataka i izradi aplikacije. Procijenite što bi sve o postavljanim upitima bilo dobro bilježiti.

Potrebno je omogućiti korisniku sljedeće:

1. zadavanje vremenskog perioda u kojem treba provesti analizu u obliku:
datumOd – datum do (npr. 10.10.2016 -13.10.2016)
2. Odabir granulacije analize:
 - a. dan ili
 - b. sat

Nakon što je poznat vremenski period za kojeg se radi analiza i granulacija analize, korištenjem mogućnosti SQL-a (pivotiranje) izraditi izvještaj s pregledom broja postavljanja konkretnog upita za zadani period (npr. 10.10.2016 -13.10.2016) po danima ili po satima ovisno o tome što je korisnik odabrao.

Po danima:

	querystring character(200)	d10102016 integer	d11102016 integer	d12102016 integer	d13102016 integer
1	'Dance' & 'Legend of Tarzan' & 'Lord'	4	3	2	
2	'Lord' & 'Dance'	3	2	2	

ili po satima:

	querystring character(200)	s00_01 integer	s01_02 integer	s02_03 integer	s03_04 integer	s04_05 integer	s05_06 integer	s06_07 integer	s07_08 integer	s08_09 integer	s09_10 integer	s10_11 integer	s11_12 integer	s12_13 integer
1	'Dance' & 'Legend of Tarzan' & 'Lord'								1		3	3	1	
2	'Lord' & 'Dance'										3	3	1	

Primijetite da, pored prvog stupca (koji sadrži tekst pretrage), pivot tablica

- sadrži još 24 stupca - za 24 sata u danu kod analize po satima
- sadrži onoliko stupaca koliko je dana u promatranom periodu kod analize po danima.

Kod analize po satima broj u konkretnom stupcu npr. s09_10 predstavlja ukupan broj postavljanja upita (prikazanog u prvom stupcu) između 9 i 10 sati za sve dane iz promatranog perioda.

Pomoć: Da biste mogli koristiti funkcije za Full Text Search i Fuzzy Text Search morate uključiti module **fuzzystrmatch** i **pg_trgm**, odnosno **tablefunc** za korištenje funkcija za pivotiranje. Ako ćete pri implementaciji *autocompletea* koristiti stabla pretrage i *ltree* tip podatka morate uključiti modul **ltree**. Module je potrebno uključiti u svakoj bazi podataka u kojoj ih namjeravate koristiti. „Registriraju“ se izvođenjem sljedećih naredbi:

```
CREATE EXTENSION fuzzystrmatch;    -- (soundex, levenshtein, metaphone)
CREATE EXTENSION pg_trgm;          -- (similarity , show_Trgm,..., %, <->)
CREATE EXTENSION tablefunc;        -- (crosstab)
CREATE EXTENSION ltrees;           -- (ltree)
```

Za prikazivanje sažetih informacija o dohvaćenim dokumentima s podebljanim ključnim riječima možete koristiti funkciju *ts_headline*.

Razmatrat će se i polovična rješenja.

Studente se kod prezentiranja projekta može pitati da:

- objasne segment vlastitog rješenja
- objasne neki koncept iz područje pretrage teksta ili pivotiranja (npr. kako radi operator/funkcija koju su upotrijebili za neku od vrsta pretrage)
- objasne zbog čega su primijenili konkretni operator/funkciju, a ne neki drugi
- pokrenu tj. demonstriraju rješenje (i npr. izvedu upit)
- itd.

Rješenje projekta koje sadrži

- izvorni programski kôd
- readme.txt datoteku koja sadrži shemu korištene baze podataka treba postaviti u vlastiti direktorij pomoću Moodle-a.

NMiBP - 1. projekt

 NMiBP - rješenje 1. projekta - Pretraživanje teksta i napredni SQL

Rok za dostavu rješenja: ponedjeljak 21.10.2019. do 23:59.

Prezentacija projekata: u utorak 22.10.2019. u D259 prema rasporedu kojeg možete vidjeti u vašem osobnom kalendaru