

Sveučilište u Zagrebu
Fakultet organizacije i informatike Varaždin

Strukturiranje, dizajn i implementacija relacijske baze podataka RFC-a sa WWW sučeljem

Seminarski rad iz kolegija informacijsko inženjerstvo

Dobrica Pavlinušić, student prve godine poslijediplomskog studija
informacijskih sustava, smjer multimedija

U Varaždinu, 5. lipnja 1998.

Sadržaj

1. UVOD.....	1
2. RFC DOKUMENTI	1
2.1 Što su RFC dokumenti.....	1
2.2 Struktura RFC dokumenata	2
2.3 Pretraživanje RFC dokumenata.....	2
3. ODLUKA O BAZI.....	3
3.1 Da li upotrijebiti relacijsku bazu?.....	3
3.2 Strukturiranje baze podataka	3
3.3 Odabir relacijske baze podataka (RDBMS-a).....	5
3.3.1 miniSQL.....	5
3.3.2 MySQL.....	6
4. IMPLEMENTACIJA.....	6
4.1 Perl i DBI.....	6
4.2 Početno punjenje baze	6
4.3 Izrada sučelja za korisnika.....	7
4.3.1 Oblikovanje vizualnog izgleda sučelja	7
4.3.2 Kreiranje SQL upita.....	8
5. REZULTATI	9
6. SMJERNICE ZA BUDUĆI RAZVOJ.....	11
6.1 Moguće optimizacije.....	11
6.2 Nadopune i moguća proširenja	11
7. ZAKLJUČAK	12
8. LITERATURA I REFERENCE	12

1. Uvod

U ovom ću radu predstaviti izradu pretraživača RFC¹ dokumenata sa sučeljem koje koristi poznate Internet tehnologije kao što su to HTTP² i HTML³ za pristup korisnicima. Na taj način omogućava pristup kako korisnicima sa lokalne mreže, tako i korisnicima Interneta do RFC dokumenata koji predstavljaju izvor znanja o samom Internetu i TCP/IP protokolima koji ga sačinjavaju.

Objasniti ću motivaciju za izradu takvog pretraživača, praktične probleme i rješenja na koja sam naišao izrađujući ga, kao i smjernice za budućnost.

2. RFC dokumenti

Na samom početku potrebno je reći što su to RFC dokumenti. Razvoj Internet-a je u mnogim svojim dijelovima bio veoma neorganiziran i stihijski, pa su tako i njegovi standardi definirani dokumentima koji se zovu Request for comments.

U nastavku ću pokušati objasniti čemu oni služe, kako su strukturirani i što je motivacija za izradu pretraživača dokumenata.

2.1 Što su RFC dokumenti

RFC dokumenti⁴ su radne bilješke istraživačke grupe DARPA⁵-e. Teme obrađene u njima uključuju specifikacije protokola, definicije sučelja, opise programa, pravila ponašanja, izvještaje o problemima, algoritme, obavijesti i humor. Teme mogu biti obrađene u obliku formalnih specifikacija ili jednostavno "ludih" ideja.

RFC dokumenti pokrivaju široko područje interesa. Glavne teme su Internet i TCP/IP protokoli. Ali bilo koje teme povezane sa računarskim komunikacijama mogu biti prihvaćene ako tako odluče urednici RFC-a⁶.

Svatko može predložiti dokument za RFC. Jedan od glavnih predlagača RFC-a je *Internet Engineering Task Force (IETF)*. IETF razvija i upotpunjava svoje radne bilješke

¹ RFC - Request for Comments; dokumenti o kojima će u ovom radu biti riječi

² HTTP - Hyper Text Transfer Protocol definira način razmjene podataka između klijenta i poslužitelja koji se koristi kod realizacije World Wide Web-a (WWW)

³ HTML - Hyper Text Markup Language je jezik koji opisuje izgled stranice na WWW-u

⁴ preuzeto iz RFC 980: Protocol Document Order Information

⁵ DARPA - agencija američke vlade za napredna istraživanja (Defence Advanced Research Project Agency)

(*Internet Drafts I-Ds*) dok ne postanu dovoljno dobre za javno objavljivanje što određuje *Internet Engineering Steering Group (IESG)*, koji ih nakon potvrđivanja šalje urednicima RFC-a.

Većina dokumenata koje predlažu nezavisne strane se također upućuju na procjenu IESG-u da bi se odredio njihov odnos sa poslovima koje već obavlja IETF.

2.2 Struktura RFC dokumenata

Struktura RFC dokumenata mora biti dovoljno univerzalna da zadovolji potrebu za prikazivanjem ili štampanjem na najrazličitijoj opremi. RFC-i se mogu slati elektronskom poštom, pretraživati nekim od servisa kao što su to Gopher ili Wais ili se distribuirati preko World Wide Web-a (WWW).

Tradicionalno se RFC-i izdaju kao ASCII dokumenti. Sekundarni oblik RFC-a je PostScript dokument⁷ koji omogućava uključivanje dijagrama i crteža u RFC.

2.3 Pretraživanje RFC dokumenata

Mnogo korisnika pretražuje RFC-e korištenjem standardnih alata raspoloživih pod Unix operacijskim sustavima kao što je to *grep*. Međutim, RFC dokumenti u Hrvatskoj su smješteni na ftp server *ftp://ftp.carnet.hr/pub/Internet/rfc/* zbog čega je omogućen pristup samo korištenjem ftp protokola, što onemogućava pretraživanje jer su dokumenti smješteni na udaljenom serveru, pa je i postojala potreba za pretraživačem. Sa istim problemom su se susreli i na Fakultetu elektrotehnike i računarstva u Zagrebu, te su prije nekoliko godina razvili svoj vlastiti pretraživač RFC-a. Međutim, prije nekog vremena, pretraživač je prestao posluživati korisnike “zbog prevelikog opterećenja servera”. Iz toga je slijedio zahtjev da novi pretraživač mora omogućiti efikasno pretraživanje bez obzira na porast broja dokumenata. Naime, godišnje se pojavi čak do dvije stotine novih dokumenata. Kako je arhiva RFC dokumenata i pretraživač ukinut 1997. godine, izgleda da njihov server jednostavno nije bio dorastao porastu broja dokumenata ili je nestalo interesa sa održavanjem takvog servisa. Svi ti problemi moraju biti uzeti u obzir pri izradi novoga pretraživača, korištenjem tehnologija koje mogu odoljeti porastu dokumenata, te kreiranjem automatskih procedura obnavljanja sadržaja poslužitelja bez intervencije administratora.

⁶ preuzeto iz RFC 2223: *Instructions to RFC Authors*

⁷ PostScript je jezik za opis stranice

3. Odluka o bazi

3.1 Da li upotrijebiti relacijsku bazu?

Prvi problem sa kojim sam se susreo kod oblikovanja pretraživača bilo je odabir relacijsku bazu podataka ili nekog drugog načina spremanja podataka kao što je to indeksiranje cjelokupnog sadržaja ili pretraživanje po ključnim riječima. Relativno statična struktura podataka koji se pretražuju, brzina odziva RDBMS⁸-a, kao i dobro poznavanje relacijskih baza podataka te načina njihovog funkcioniranja, zajedno sa izazovom izrade sučelja koje će korisniku omogućiti korištenje pretraživača bez poznavanja jezika za postavljanje upita (u ovom slučaju SQL⁹-a) pridonijeli su izboru relacijske tehnologije.

3.2 Strukturiranje baze podataka

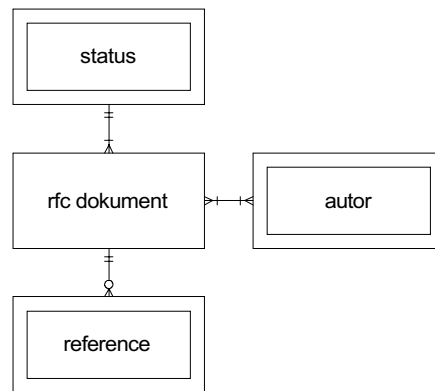
Baza je strukturirana prema datoteci `rfc-index.txt` u kojoj se nalazi popis svih RFC-a sa naslovom, autorima, datumom izdavanja, formatom i referencama na druge RFC-e. Kako je odlučeno da baza sadrži i kratke sadržaje (*abstract*) ukoliko ga RFC posjeduje, dodano je i to polje.

```
####    Title of RFC.  Author 1, Author 2, Author 3.  Issue date.
        (Format: ASCII) (Obsoletes xxx) (Obsoleted by xxx) (Updates
        xxx) (Updated by xxx) (Also FYI ####)
```

Tablica 1: format datoteke `rfc-index.txt` koja se koristi pri kreiranju baze

⁸ RDBMS - Relational Database Management System - sustav za upravljanje relacijskom bazom podataka

⁹ SQL - Sequential Query Language - jezik za postavljanje upita i upravljanje podacima RDBMS-a

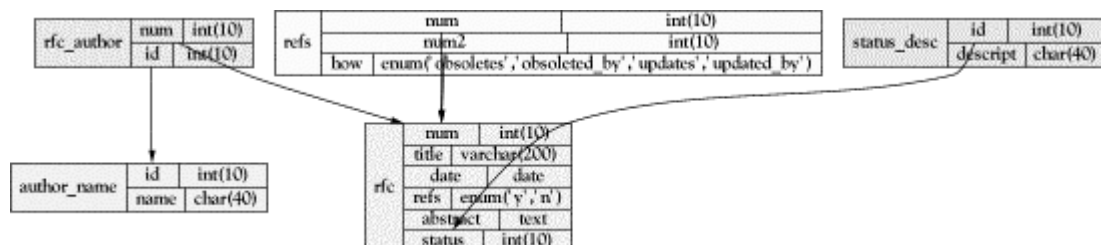


Slika 1: ERA model baze podataka pretraživača RFC-a

rfc dokument	<u>broj</u> , naslov, datum, <u>status</u> , kratak_sadržaj
status	<u>oznaka</u> , opis
autor	<u>oznaka</u> , ime_autora
reference	<u>broj</u> , <u>broj</u> , tip_reference

Tablica 2: atributi tablica u BP

Pri implementaciji gore navedenih tablica i unošenju atributa, trebalo je odrediti tip podataka za svaki od atributa i pretvoriti vezu “više-prema-više” između tablica rfc dokument i autor u dvije veze “jedan-prema-više”. Također je dodan statusni atribut u tablicu rfc dokumenata koji govori da li zadani dokument ima reference na drugi dokument ili nema, čime je moguće uštedjeti jedan upit kod pretraživanja. Konačna struktura tablice, sa stvarnim imenima tablica i atributa, koji će biti i korišteni u daljnjem tekstu, dana je na slici 2.



Slika 2: Stvarna struktura BP

3.3 Odabir relacijske baze podataka (RDBMS-a)

Dakle, nakon što je odabrana relacijska tehnologija kao način rješavanja problema izgradnje pretraživača, ostao je problem izbora konkretne baze podataka. Kako nisu postojala novčana sredstva dodijeljena ovom projektu, bilo je potrebno odlučiti se za jednu od dobrih besplatnih baza podataka.

Iako je konkurencija bila velika (osim ovdje obrađenih miniSQL-a i MySQL-a u obzir su dolazile i druge baze podataka koje su za akademsku i istraživačku primjenu besplatne, kao što su to Postgres ili Leap) bilo je nekoliko glavnih zahtjeva:

- mogućnost jednostavnog programiranja baze podataka u nekom proceduralnom jeziku (dakle, mogućnost primjene proceduralnog jezika uz SQL, po mogućnosti perl-a)
- jednostavno izvođenje sučelja na Web-u (mogućnost korištenja perl-a osigurava ovu točku)
- subjektivno brz odaziv korisničkim upitima (uključuje vrijeme upita i stvaranje izvještaja u obliku HTML stranice za manje od tri sekunde)
- razumni zahtjevi za računarskim resursima: baza podataka bi se sa dva do tri istovremena korisnika trebala moći izvršavati na serveru koji je u klasi i486 na 80 Mhz sa 16 Mb RAM memorije i Linux operacijskim sustavom. To je ujedno bila i ciljna poslužiteljska platforma sa mogućnošću prelaska na jače računalo ukoliko se u budućnosti pokaže potreba za time.

3.3.1 miniSQL

Prva baza podataka sa kojom smo se susreli, a koja je zadovoljavala sve postavljene uvjete bila je upravo miniSQL baza podataka. U njoj je i bila realizirana prva faza implementacije, tj. implementirano početno punjenje baze podataka. Međutim, nakon dodatnih restrukturiranja i optimizacije baze (o čemu detaljnije piše u odjeljku o implementaciji) pokazala se potreba za npr. *enum* tipom podataka što ova baza podataka nije omogućavala. Ova baza nije omogućavala niti definiranje vanjskih ključeva, a njezino relativno zaleđeno stanje razvoja doprinijelo je odluci da se potraži druga baza podataka za konačnu implementaciju. Jednom riječju, ova baza je bila ipak previše “mini” za ovu primjenu.

3.3.2 MySQL

Logičan nasljednik miniSQL baze podataka bila je MySQL baza, koja se još uvijek intenzivno razvija, a kompatibilna je na razini programskog sučelja (API-ja) sa miniSQL bazom. To je omogućavalo da se već razvijeni kod za početno punjenje baze podataka iskoristi bez modifikacija. Na žalost, API kompatibilnost ne znači i kompatibilnost na nivou SQL-a. Dok miniSQL podržava veoma ograničen podskup SQL92 mogućnosti, MySQL je mnogo moćnija baza koja podržava mogućnosti kao što su *not null* i *auto increment* definicija atributa, definiranje primarnih i vanjskih ključeva (doduše bez provjere integriteta), jednom riječju skoro sve mogućnosti SQL92, osim što se neke u trenutnoj verziji ignoriraju, a postoje zbog kompatibilnosti. Sva nestandardna rješenja miniSQL-a (kao što je to npr. **create sequence**) jasno su se pokazala pri prelasku na MySQL.

4. Implementacija

4.1 Perl i DBI

Pri implementaciju baze odlučio sam se za korištenje programskog jezika perl koji je prilagođen radu sa tekstualnim datotekama (što je osnovni oblik RFC datoteka), izradi CGI sučelja programa (koje je potrebno za izradu korisničkog sučelja na WEB-u), te ima podršku za DBI (*database interface*) što je API koji omogućava pristup različitim bazama podataka korištenjem istog programskog koda. Kao što smo to napomenuli kod opisa miniSQL-a i MySQL-a to ne znači (nažalost) korištenje i istih SQL upita, međutim, DBI API je omogućio jednostavan prijelaz sa miniSQL-a na MySQL bez mijenjanja same strukture koda.

4.2 Početno punjenje baze

Početno punjenje baze odvija se korištenjem perl programa koji čita datoteku `rfc-index.txt` i stvara tablice te ih puni podacima. Za svaki RFC dokument pretražuje se i cijeli tekst dokumenta da bi se našao kratak sadržaj (*abstract*) ukoliko ga dokument posjeduje.

Iako je struktura zapisa u datoteci `rfc-index.txt` definirana u samoj datoteci, ubrzo se pokazalo da se sami zapisi u datoteci ne pridržavaju tog formata. Tako se npr. sreću

datumi oblika “Mjesec, 1999” gdje ponekad ima, a ponekad nema zareza, a postoji i drugi oblik datuma koji je “Mmm-dd-1999”. Na svu sreću, moćni *regular expression*-i u perl-u problem datuma su riješili sa samo nekoliko specijalnih slučajeva¹⁰.

Program za punjenje baze već sada podržava punjenje baze samo od nekog određenog broja RFC-a, što je predviđeno za periodično dodavanje novoizdanih RFC-a bazi. Međutim, kako sadašnja struktura baze, radi ispravnog navođenja referenci na druge RFC-e, zahtijeva i modifikaciju postojećih dokumenata, ta se mogućnost još ne koristi¹¹.

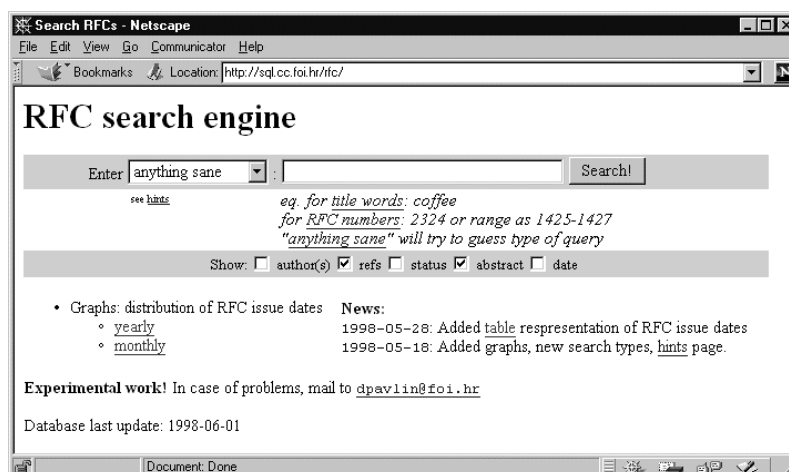
Izrada strukture baze podataka zajedno sa programiranjem početnog punjenja baze (i prelaskom sa miniSQL DBMS-a na MySQL) trajala je 14 dana.

4.3 Izrada sučelja za korisnika

Izrada korisničkog sučelja pretraživača se zapravo sastoji od dva prilično nezavisna dijela: samog oblikovanja vizualnog izgleda maske u koju korisnik upisuje podatke i programiranja programske podrške koja će pretraživati bazu podataka.

4.3.1 Oblikovanje vizualnog izgleda sučelja

Kako izgleda korisničko sučelje kod unosa upita možete pogledati na slici 3.



Slika 3: izgled korisničkog sučelja kod unošenja upita pretraživaču

¹⁰ Točnije, sa dva slučaja. Za prvi oblik datuma koristi se regexp oblika “(\\w\\w\\w)\\w*,* *([1|2]\\d\\d\\d)”, dok se za drugi oblik koristi “(\\w\\w\\w)-(\\d\\d)-([1|2]\\d\\d\\d)”. Dodatne informacije o regular expression-ima se mogu dobiti ukoliko se na Unix serverima zada naredba `man perlre`.

¹¹ Iskoristiti će se u trenutku kada baza više neće sadržavati posebne slogove za reference tipa “*obsoleted by*” i “*updated by*” već samo na “*obsoletes*” i “*updates*” što je u planu i o čemu će još biti riječi.

Osim mjesta za unos podataka za pretraživanje možete odrediti i jedan od sljedećih oblika upita:

1. automatska selekcija tipa upita prema unesenim podacima (pretražuje po riječima u naslovu RFC-a ukoliko je unesena riječ ili po broju te rasponu brojeva ukoliko su uneseni brojevi),
2. pretraživanje po broju ili rasponu brojeva,
3. pretraživanje riječi samo u naslovu RFC-a,
4. pretraživanje riječi samo u kratkom sadržaju RFC-a,
5. pretraživanje riječi u naslovu i kratkom sadržaju.

Također možete odrediti koje sve podatke želite prikazati za nađene dokumente. Ukoliko ne promijenite ponuđene vrijednosti, ispisivat će se reference i kratki sadržaji dokumenata, dok autor(i), status i datum izdavanja neće biti prikazani.

4.3.2 Kreiranje SQL upita

Iz prije spomenutog, može se zaključiti da je potreban veliki broj različitih oblika SQL upita da bi se dobili svi podaci koje korisnik traži. Jedan od zahtjeva je bio i brzina, tako da je cilj postavljati što manje upita. Tako programska podrška nakon što korisnik unese podatke sama generira SQL upit u zavisnosti od načina pretraživanja i količine podataka o svakom dokumentu koju je korisnik odabrao.

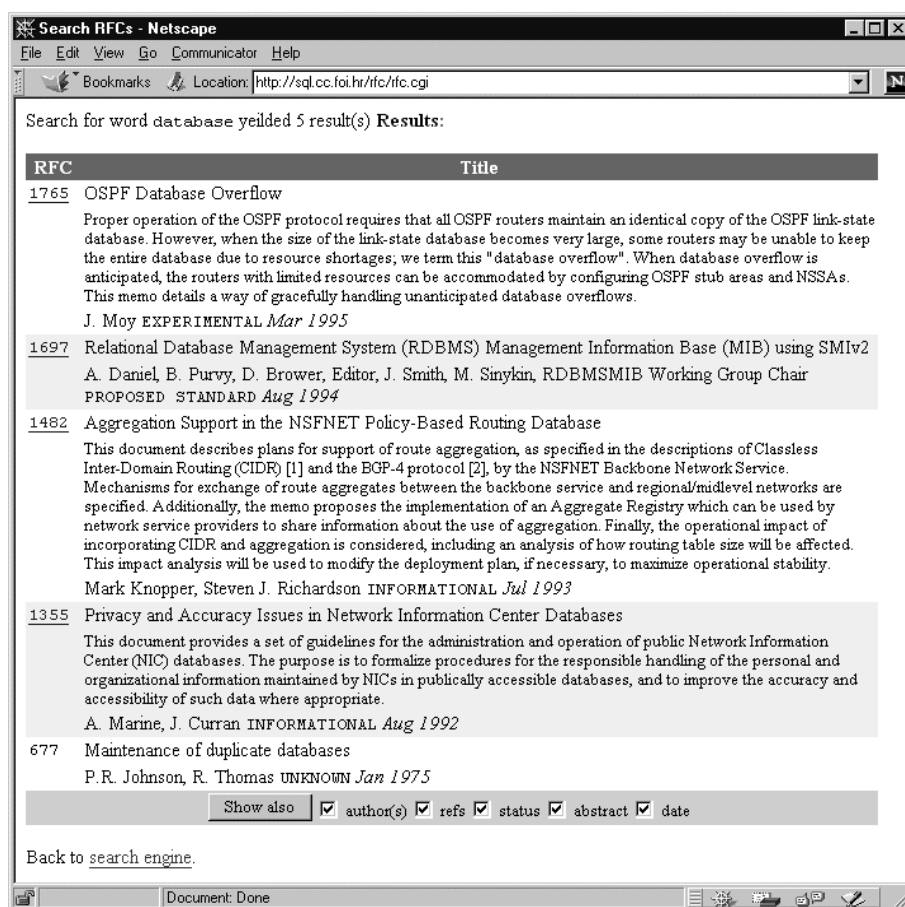
Primjer SQL upita koji ispisuje sve RFC dokumente u kojima se u naslovu prikazuje riječ “coffee” je sljedeći:

```
SELECT DISTINCT rfc.num,rfc.title,rfc.abstract,rfc.refs FROM rfc
WHERE rfc.title like '%coffee%' ORDER BY rfc.num DESC
```

Ukoliko je polje refs u tablici rfc postavljeno na ‘y’ to označava da je taj RFC u nekoj vezi sa nekim drugim RFC-om, pa se postavlja dodatni upit da bi se te veze i dobile:

```
SELECT DISTINCT num2,how FROM refs WHERE num=2324
ORDER BY num2 DESC
```

Naravno, ukoliko se zahtjeva ispis autora i statusa, izvršavaju se i dodatni SQL upiti. Primjer rezultata jednog takvog upita sa ispisom svih podataka, za riječ “database” u naslovu ili kratkom sadržaju dan je na slici 4.

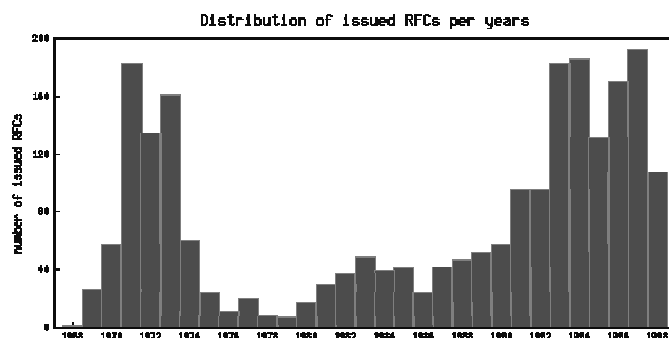


Slika 4: primjer ispisa rezultata pretraživanja

5. Rezultati

Rezultati izrade mogu se sumirati na sljedeći način:

- jednostavnije pretraživanje,
- jednostavno određivanje veza između dokumenata,
- efikasnost pronalaženja informacija,
- mogućnost statističkih analiza (distribucija izdavanja RFC-a po godinama ili mjesecima) grafički i tablično - kao što je prikazano na slikama 5 i 6,
- samostalno obnavljanje sadržaja BP i statistike svakih 5 dana korištenjem cron procesa koji se aktivira svakih 5 dana, skida nove RFC dokumente koji su izdani, dodaje ih u bazu i na kraju obnavlja statistiku.



Slika 5: grafički prikaz raspodjele broja RFC-a po godinama

RFC analysis - Netscape

File Edit View Go Communicator Help

Bookmarks Location: <http://sql.cc.foi.hr/rfc/table.html>

Distribution of RFC issue dates per year and months

Y/M	1	2	3	4	5	6	7	8	9	10	11	12	sum
1968		1											1
1969	2	1	1	4	3	1	1	4	3	6	1	1	28
1970	5	10	8	8	1	7	5	4	4	5	2	7	66
1971	15	21	7	30	27	12	19	15	23	13	8	10	200
1972	22	22	12	10	17	9	14	12	6	10	9	11	154
1973	30	33	18	13	11	18	14	11	7	6	14	12	187
1974	11	12	11	6	1	3	3	2	1	10	5	7	72
1975	3	2	1	4	3	5	5		1		2	1	27
1976	1	1		2	1	1	2	1	2				11
1977	5	2	2	2	3	1			1	4	5	2	27
1978	1	1	3	1					2			1	9
1979	1		1	1	1		1	1	1				7
1980	3	1			1	2	1	3	4	2		1	18
1981	7			3	2	1	5		11		4		33
1982	6	8	4				5	6	6	2	6	5	48
1983	9	15	1	3	16	1			2	2	4	9	62
1984	2	7		6	1	4	4	2	3	10		5	44
1985	9	13		4	2	2	1	1	5	4	3	8	52
1986	7	5	4	3	2	2	1				3	1	28
1987	8	4	5	2	4	6	2	3	3	5	8	2	52
1988	11	12	2	3	2	4	3	5	1	3	7	5	58
1989	5	12	2	5	1	4	1	9	5	8	4	4	60
1990	6	7	4	6	6	6	5	8	1	8	4	3	64
1991	14	18	7	8	12	6	10	5	6	9	11	11	117

Document: Done

Slika 6: dvodimenzionalni tabelarni prikaz broja izdanih RFC-a po mjesecima i godinama uz naglašavanje vrijednosti bojom

6. Smjernice za budući razvoj

Iako su rezultati ovoga projekta zadovoljavajući, to nikako ne znači da ne postoji mogućnost poboljšanja ovog sustava.

6.1 *Moguće optimizacije*

Pamćenje samo “obsoletes” i “updates” reference u tablici refs, a dobivanje referenci tipa “obsoleted by” i “updated by” iz tih podataka preko drugačijeg upita u bazu. Na taj način bi se izbjeglo mijenjanje podataka za postojeće RFC-e, te pojednostavnio upis novih podataka u bazu podataka jer više ne bi bilo potrebno upisivati starim dokumentima da su zamijenjeni novim jer bi se taj podatak i tako čitao iz podataka o novim dokumentima. To bi također smanjilo veličinu refs tablice, omogućilo točniju bazu podataka (jer ne bi bilo dupliciranja podataka), ali bi zahtijevalo još jedan upit u bazu u slučaju da se koristi prikaz referenci ili posebnu obradu podataka nakon upita u refs bazu. To je međutim, planirano za sljedeću veću reviziju ove baze podataka.

6.2 *Nadopune i moguća proširenja*

Jedno od mogućih proširenja baze je i unos potpunih tekstove RFC-a, a ne samo kratkih sadržaja. Međutim, kod toga treba razmotriti odnos prostora potrebnog za unos potpunih tekstova (o odnosu na trenutno rješenje pozivanja dokumenata korištenjem ftp servera i pojedinačnih datoteka), brzine dohvaćanja dokumenata i mogućnost pretraživanja po cijelim dokumentima. Također, bilo bi potrebno razmotriti ne bi li neki drugi sustav za indeksiranje umjesto pretraživanja korištenjem SQL upita LIKE davao bolje rezultate.

Povezivanje tekstova RFC-a korištenjem HTML linkova na tekstove ili na pretraživač je također jedna od dopuna koja će uskoro biti implementirana.

Zanimljivo je razmotriti i mogućnost stvaranja indeksa i rječnika skraćenica (veoma uporabljivo) koje su veoma česte u RFC dokumentima, a često su im potrebna objašnjenja. Međutim, kako to prelazi okvire ovoga rada, tom temom se ovdje nećemo detaljnije baviti.

Provođenje inkrementalne nadopune baze je nadopuna koja će biti implemenirana čim se promjeni struktura refs tablice, što je već spomenuto.

7. Zaključak

Ovaj rad predstavlja prikaz praktične primjene relacijske tehnologije i Interneta koji pružaju mogućnost izrade sustava za pretraživanje podataka. Prikazani su problemi i rješenja koja su pronađena.

Operativna iskustva sa ovim sustavom su veoma povoljna i poticajna. Po mišljenju autora, ovaj rad može poslužiti i kao smjernica za buduće sustave temeljene na sličnim tehnologijama.

8. Literatura i reference

1. Mario Radovan: Baza podataka - relacijski pristup i SQL, Informator, Zagreb 1993.
2. MiniSQL user guide, Hughes Technologies Pty Ltd, 1997.
3. MySQL reference manual, 1998.
4. MiniSQL home page na adresi <ftp://bond.edu.au/pub/Minerva/msql/>
5. MySQL home site na adresi <http://www.tex.se/>