

Pitanja i odgovori po prezentacijama

1. Pretraživanje teksta u RSUBP

1. Zašto su modeli podataka bitni?

Predstavljaju temelj na kojem se gradi programska potpora ili informacijski sustav

2. Što je model podataka?

Formalni sustav koji koristimo kod modeliranja baza podataka

3. Nabroji koncepte modela podataka (3)

Osnovni objekti, operacije i integritetska ograničenja

4. Zbog čega je pretraživanje teksta izazov? Navedi dva primjera.

Tekst pisan prirodnim jezikom je često nejasan i dvosmislen te podliježe pravopisnim i gramatičkim pravilima jezika.

5. Što je FTS? Objasni komponente

FTS- Full Text Search - Traženje dokumenata koji zadovoljavaju postavljeni **uvjet** i rangiranje u skladu sa **sličnošću** dokumenta s uvjetom.

Uvjet je obično niz riječi, a **sličnost** brojčana vrijednost koja je u najjednostavnijoj implementaciji povezana s frekvencijom pojavljivanja riječi iz uvjeta u pretraživanom dokumentu

6. Zašto pretraživati tekst u relacijskim bazama podataka?

Zbog velike količine tekstualnih podataka pohranjenih u relacijskim bazama podataka.

7. Nabroji nedostatke standardnog RSUBP u pretraživanju teksta.

Ne postoje efikasni indeksi pa pri procesiranju svakog upita treba "obraditi" svaki dokument => sporo

8. Nabroji i objasni 3 pristupa pri pretraživanju teksta.

- **Točno podudaranje**
 - uvjet i tekst se u cijelosti podudaraju, identični su
 - uvjet je sadržan u dijelu teksta (djelomino podudaranje)
 - =, LIKE, NOT LIKE, SIMILAR TO, regularni izrazi
- **Podudaranje temeljem morfologije, sintakse i semantike jezika**
 - Koriste se tzv. gramatiki algoritmi koji pronalaze podudarnost izmeu uvjeta i teksta temeljem normaliziranog oblika riječi (korigen, leksem), temeljem sinonima i itd.
 - TSVector, TSQuery tipovi podataka, @@ operator
- **Približno podudaranje (fuzzy)**
 - Algoritmi temeljni na „udaljenosti“ izmeu znakovnih nizova - Levenshtein funkcija, Hamming
 - Q-Gram algoritmi - % operator, similarity funkcija
 - Fonetsko podudaranje - Soundex, Metaphone funkcije

9. Nabroji što se točno podrazumjeva pod obradom teksta za pretraživanje?

- parsiranje dokumenta i rastavljanje na tokene (riječi, brojevi, tagovi...)
- uklanjanje riječi koje nemaju semantičko značenje u tekstu(stop riječi)
- identificiranje sinonima

10. Što omogućuje dobar rječnik?

- definiranje stop riječi
- definiranje sinonima
- stvaranje veza između fraza i pojedinih riječi

11. Što je parser u PostgreSQL-u? Rječnik? Navedi tipove rječnika

Parser - razdvaja izvorni tekst na tokene i utvrđuje tip tokena (ne modificira izvorni tekst), ts_debug funkcija

Rječnik - koristi se za uklanjanje stop riječi, normalizaciju teksta...

tipovi : Simple Dictionary (uklanja stop riječi i velika slova svodi na mala), Synonym Dictionary, Thesaurus Dictionary (prepoznaje fraze), Snowball Dictionary (algoritamski svodi riječi na korijenski oblik i uklanja stop riječi)

12. Što su stop riječi? Navedi primjer

Riječi koje se vrlo često pojavljuju, gotovo u svakom dokumentu - ignoriraju se pri pretrazi teksta izuzev pri traženju fraza. (npr. za engleski : I, me, a, the...)

13. Kojim redoslijedom se pretražuju rječnici pri normalizaciji tokena?

Redoslijedom kojim su navedeni. Ako neki rečnik prepozna token, rječnici nakon njega se ne konzultiraju.

14. Što je TSVector? Provodi li normalizaciju?

Podatkovni tip koji predstavlja dokument i optimiran je za fts pa obuhvatnije pretražuje tekst od LIKE, SIMILAR... Ne provodi normalizaciju

15. Što je TSQuery? Provodi li normalizaciju?

Podatkovni tip za predstavljanje upita s podrškom za Booleove operatore (AND i OR). funkcije to_tsquery i plainto_tsquery provode normalizaciju.

16. Vrste indeksa za FTS? Karakteristike?

GIN (generalized inverted index) - atribut mora biti tip TSVector (brža pretraga ali dulja izgradnja, sporiji UPDATE i zauzima više mjesta od GIST-a

- CREATE INDEX idxName ON tableName USING gin(attrName)

GIST (generalized search tree) - atribut mora biti tipa TSVector ili TSQuery

- CREATE INDEX idxName ON tableName USING gist(attrName)

17. Što je fuzzy text search? Nabroji i objasni osnovne algoritme (3)

Tehnika pronalaženja dokumenta/niza znakova koji se približno podudara s traženim uzorkom. Kvaliteta podudaranja se mjeri ovisno o vrsti primijenjenog algoritma

1. Algoritmi temeljeni na udaljenosti znakovnih nizova

- Udaljenost uređivanja (edit distance) znakovnih nizova s_1 i s_2 je minimalan broj operacija potrebnih da se jedan niz transformira u drugi. Moguće operacije su: izmjena, umetanje i brisanje znaka.
- 2. Q-Gram algoritmi
 - Tekst/dokument je (multi) skup q-grama
 - Ako je riječ A slična riječi B, one vjerojatno sadrže barem jedan podudaran (zajednički) podniz duljine Q
- 3. Soundex, Metaphone algoritam
 - traže riječi koje se slično izgovaraju

18. Kakav utjecaj imaju stop riječi na rangiranje rezultata?

za jednak upit, rang istog dokumenta je različit ovisno o tome jesu li stop riječi uklonjene ili ne. (smanjuju rang ako su prisutne)

19. Koja je razlika između ts_rank i ts_rank_cd?

ts_rank - rangira rezultate temeljem frekvencije leksema koji se podudaraju u upitu i dokumentu

ts_rank_cd - računa rang na način : dokument duljine 1000 riječi u kojem se tražena riječ pojavljuje 10 puta nije jednako relevantan kao dokument od 100 riječi u kojem se također pojavljuje 10 puta

2. Napredni SQL

1. Što je pivotiranje?

Uobičajena tehnika za predstavljanje podataka u sustavima poslovne inteligencije (BI alati). Standardnim SQL-om se podaci prikazuju u recima, a pivotiranjem se ti isti podatci predočavaju u stupcima.

Nije definirano SQL standardom!

2. Koja su ograničenja korištenja funkcije crosstab?

Funkcija crosstab kao rezultat uvijek vraća nekakav skup n-torki.

crosstab (text sql) - proizvodi pivot tablicu koja sadrži imena redaka i N imena stupaca pri čemu je N određen upitom i predstavlja tip i broj vrijednosti.

SQL naredba mora vratiti trojke: (prva vrijednost u retku, kategorija stupca, vrijednost ćelije)

Imena stupaca moraju biti eksplicitno definirana u FROM dijelu pozivajuće SQL naredbe!

Oblik SQL upita s crosstab funkcijom

```
SELECT * FROM crosstab('...') AS ct (prvaVrijUREdu text, kat1 text, kat2 ext,...);
```

Rezultat SQL naredbe mora biti sortiran prema prvaVrijUREdu, a zatim prema kategoriji.

crosstab (text srcSql, text categorySql) - proizvodi pivot tablice čiji su stupci specificirani drugim upitom (argumentom).

3. Usporedi GROUP BY i funkcije za rad s prozorima u PostgreSQL-u

GROUP BY atr_1, \dots, atr_n	Funkcije za rad s prozorima
Jedna n-torka na izlazu za svaku grupu	Jedna n-torka na izlazu za svaku u prozor ulaznu n-torku
Vrijednosti agregatnih funkcija se računaju za cijelu grupu	Vrijednosti agregatnih funkcija se računaju za prozore (particije/okvire)
Samo jedan način grupiranja u jednoj SELECT naredbi	Agregati u istoj SELECT naredbi mogu biti izračunati temeljem n-torki sadržanih u različitim prozorima (particijama/okvirima)

4. Što je prozor (window)? Koje su podcjeline prozora?

Prozor je tranzijentni skup n-torki pomoću kojeg se definiraju particija i okvir.

Particija se definira pomoću PARTITION BY dijela u OVER() i može sadržavati okvire te je nepomična. Slično kao GROUP BY, omogućuje podjelu n-torki relacije pri čemu svaka n-torka pripada jednoj particiji. Izostavimo li PARTITION BY, cijela relacija će biti jedna particija.

Okvir se definira pomoću frame i ORDER BY dijelova u OVER() i pomiče se unutar particije. Svaka n-torka pripada okviru, a okvir pripada particiji. Bez frame i ORDER BY dijela cijela particija je jedan okvir.

5. Nabroji vrste okvira, navedi 4 primjera ROW okvira

Postoje dvije vrste okvira: ROWS okvir i RANGE okvir.

ROWS okviri -> poredak n-torki ne mora biti definiran

vrste:

- 1) ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW (sve n-torke od početka particije do tekuće n-torke)
- 2) ROWS BETWEEN 1 PRECEDING AND 1 PRECEDING (prethodna n-torka)
- 3) ROWS BETWEEN 1 FOLLOWING AND 1 FOLLOWING (sljedeća n-torka)
- 4) ROWS BETWEEN 1 PRECEDING AND 1 FOLLOWING (prethodna, trenutna i sljedeća n-torka)

RANGE okviri -> poredak n-torki u particiji je važan

6. Zašto se definiran i imenovan prozor može koristiti samo u SELECT i ORDER BY?

Funkcije za rad s prozorima mogu koristiti izračunate vrijednosti agregatnih funkcija (COUNT, AVG, SUM,...) nakon procesiranja FROM, WHERE, GROUP BY i HAVING dijelova SELECT naredbe

Posljedica: mogućnost obavljanja ugniježđenih agregacija: AVG(SUM(bod)), MIN(SUM(bod))

7. Koje funkcije za rad s prozorima predviđa SQL standard?

- 1) funkcija za rangiranje
- 2) funkcija za distribuciju
- 3) funkcija za određivanje n-torke temeljem pozicije (row number)
- 4) agregatna funkcija na razini prozora

8. Koja je ideja CTE? Kakve veze imaju s rekurzivnim upitima?

CTE - Common Table Expressions - Ideja:

- 1) Odrediti sadržaj relacija R1, R2, ... , Rn i pohraniti ga u privremene relacije
- 2) Procesirati upit koji uključuje R1, R2, ... , Rn i druge relacije
- 3) Ukloniti R1, R2, ... , Rn
omogućuje implementaciju rekurzivnih upita na sljedeći način

```
WITH RECURSIVE
  R AS (upit
        UNION
        rekurzivni upit)
<upit koji uključuje R (i druge relacije)>
```

9. Može li PostgreSQL razriješiti beskonačnu petlju?

može, prepozna beskonačnu rekurziju i uspije je razriješiti

3. OOBP i ORBP

1. Zašto objektno orijentirane baze podataka?

Relacijske baze podataka nisu prikladne za aplikacije koje koriste složene tipove podataka ili nove tipove podataka za velike nestrukturirane objekte.

2. Usporedi OO model i relacijski model.

OO Model - Razred, Objekt, Varijable razreda, Metode

Relacijski model - Relacijska shema, entitet (n-torka), atributi, procedure

Kod prebacivanja iz relacijskog u OO model pojavljuje se tzv. objektno relacijska neusklađenost.

3. Nabroji osnovna načela OO SUBP.

koristi koncepte OO sustava (razrede, složene objekte, hijerarhije klasa, ućahurivanje...) te koncepte SUBP (perzistencija podataka, fizička organizacija, paralelni rad)

4. Što je OID? Koje su mu karakteristike?

jedinstven, nepromjenjiv identifikator objekta generiran od OO sustava. Neovisan je o vrijednostima atributa objekta, nevidljiv je korisniku i koristi se za referenciranje objekta.

5. Što je ODMG standard i od čega se sastoji?

sastoji se od: objektnog modela, Jezika za specificiranje objekata, objektnog upitnog jezika i veze na programske jezike.

6. Što je OQL i počemu se razlikuje od SQL-a?

on je upitni jezik napravljen po uzoru na SQL. Razlika je što podržava referenciranje na objekte unutar tablica i objekti mogu ugnježdjavati druge objekte, ne podržava ključne riječi iz SQL te podržava matematičke izračune unutar OQL izraza.

7. Navedi prednosti i mane OOSUBP.

prednosti:

- bolje i brže upravljaju sa složenim objektima i vezama
- podržavaju hijerarhiju, klase i nasljeđivanje
- jednostavan podatkovni model - objekti u bazi i objekti u aplikaciji su jednaki
- identifikacija objekata skrivena od korisnika

mane:

- nema logičke neovisnosti podataka
- nedostatak dogovorenih standarda (postojeći nije u potpunosti implementiran)
- ovisnost o jednom programskom jeziku
- nedostatak Ad-Hoc upita
- nije interoperabilan s alatima i mogućnostima koje se koriste u SQLu

8. Nacrtaj DBMS Matrix

ad hoc upiti	relacijska baza	objektno relacijska baza
nema ad hoc upita	datoteka	objektna baza
	jednostavni podaci	složeni podaci

9. Koja objektno relacijska proširenja uključuje SQL standard?

unaprijed definirani tipovi (atomarni tip), izgrađeni tipovi (referenca, kolekcije, row), korisnički definirani tipovi (distinct tip, strukturirani tip)

10. Nabroji i opiši podjelu tipova podataka kako ih je opisao SQL standard i kako ih implementira PostgreSQL.

a. izgrađeni tipovi podataka

- atomarni - REF
- kompozitni
 - kolekcije:
 - ARRAY - 1D polje s max brojem elemenata
 - MULTISSET - neuređena kolekcija koja dozvoljava duplikate
 - SET - neuređena kolekcija koja ne dozvoljava duplikate
 - LIST - uređena kolekcija koja dozvoljava duplikate
 - ROW - sekvenca od jednog ili više elemenata definiran parom (ime_ele, tip_pod)

ARRAY - najbitnije funkcije : UNNEST(ime_polja) i COLLECT(ime_polja) za prebacivanje između ARRAY i pojedinačnih n-torki

ROW - može se koristiti za definiranje složenih atributa relacije

```
CREATE TABLE osoba (  
  sifOsoba INTEGER,  
  ime VARCHAR(25),  
  prezime VARCHAR(25),  
  adresa ROW ( ulica VARCHAR(50),  
               mjesto ROW (postBr INTEGER,  
                          nazMjesto VARCHAR(40))  
            )  
);
```

b. korisnički definirani tipovi podataka

- distinct - nema nasljeđivanja tipova, za usporedbu treba raditi CAST
- strukturirani - podržano nasljeđivanje tipova

11. SQL razlikuje 3 tipa pohranjenih rutina. Nabroji ih
Funkcija, metoda i procedura

12. Kada je moguće postići korisnički definiran CAST?
Kod složenog tipa podataka (kod DOMAIN nije moguće).

13. Što SQL standard propisuje o nasljeđivanju? Kako to PostgreSQL implementira?
nasljeđivanje tipova - moguće samo za strukturirane tipove, hijerarhija tipova
nasljeđivanje tablica - moguće samo za tipizirane tablice, odgovara E-R pojmu
specijalizacije/generalizacije, omogućava više tipova podataka dozvoljavajući istovremeno postojanje entiteta u više od jedne tablice

PostgreSQL - moguće nasljeđivanje samo temeljnih tablica (ne nasljeđuju se indeksi, UNIQUE, PRIMARY i FOREIGN KEY ograničenja te okidači), dozvoljeno višestruko nasljeđivanje

14. Prednosti i mane ORDBMS?
prednosti:

- zadržane sve mogućnosti relacijskih baza podataka
- mogućnost ponovnog korištenja i dijeljenja funkcionalnosti

mane:

- složenost
- nezadovoljstvo pobornika relacijskog modela i OO modela

4. Vremenske baze

1. Kako se može modelirati vrijeme?

- konačno/ beskonačno
- diskretno/kontinuirano
- apsolutno/relativno (31.listopada 2015. 9:15 / dva tjedna)

2. koja je razlika između stanja i događaja?

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 - opisuju činjenice vezane uz neki objekt u bazi podataka koje su se dogodile u određenom trenutku (chrononu) i nemaju trajanje.

3. koje su mane upravljanja vremenom kroz aplikaciju?

Semantika vremena, operacije te ograničenja integriteta moraju biti ugrađeni izravno u aplikaciju. Složeni upiti podložni greškama i nedjelotvorno izvođenje upita.

4. koja su dva moguća pristupa implementaciji vremenskih koncepata?
ugrađivanje koncepata u : 1. korisničku aplikaciju ili 2. SUBP.

5. Navedi i opiši osnovne vremenske tipove podataka. Kako su podržani u PostgreSQLu?
Osnovni tipovi podataka:

- **instant** - određeni chronon na vremenskoj liniji (12.11.2015. 19:15) - **timestamp** u PostgreSQL
- **interval** - neusidreni interval na vremenskoj liniji, ima samo trajanje (2 sata) - interval u PostgreSQL
- **period** - usidreni interval na vremenskoj liniji (05.10.2015. - 29.01.2016.) - tsrange u PostgreSQL
- **periods** - skup disjunktih usidrenih intervala

6. Nabroji druge operatore nad vremenskim podacima. (5)

zbiranje i razlika perioda, izdvajanje donje ili gornje granice perioda, pripadnost vremenskog trenutka periodu i odnosi između vremenskih trenutaka.

7. Nabroji i opiši vremenske dimenzije.

postoje dvije dimenzije vremena u kontekstu temporalnih baza podataka:

- **Vrijeme valjanosti** - vrijeme u stvarnom svijetu kada se neki događaj dogodio u kojem je neka činjenica važeća, nezavisno od trenutka kada je ta informacija zapisana u bazu podataka
- **Transakcijsko vrijeme** - vrijeme kada je određena promjena zabilježena u bazi podataka ili vremenski interval tijekom kojeg se baza podataka nalazi u određenom stanju

8. Nabroji i objasni temporalne relacije.

s obzirom na sposobnost upravljanja vremenom valjanosti i transakcijskim vremenom, razlikujemo 4 vrste temporalnih relacija:

1. **trenutačne relacije** - opisuju jedan trenutak u stvarnom vremenu (najčešće sadašnjost), staro stanje baze se zaboravlja nakon INSERT/UPDATE/DELETE
2. **relacije vremena valjanosti** - pohranjuju povijest podataka u stvarnom svijetu, sadrže period valjanosti kao metapodatak definiran početkom i krajem
3. **relacije transakcijskog vremena** - pohranjuju izmjene u bazi podataka, moguće je reproducirati stanje baze iz bilo kojeg trenutka u prošlosti
4. **bitemporalne relacije** - pohranjuju stanje baze podataka duž obje vremenske osi, podržavaju svojstva i relacije vremena valjanosti i relacije transakcijskog vremena

9. Prednosti relacije vremena valjanosti.

- većina poslovnih podataka je podložna promjenama
- omogućavaju pojednostavljenje programskog koda
- omogućavaju poboljšane performanse
- transparentne su u odnosu na nasljeđene aplikacije

10. Što o proširivosti vremenskih relacija kaže SQL standard, a što PostgreSQL?

SQL standard proširuje postojeću sintaksu opcijama za transakcijski period SYSTEM_TIME dok PostgreSQL ne podržava niti jednu od njih.

5. Geoprostorne baze podataka

1. Formalna definicija GIS-a?

Informacijski sustav za upravljanje, analizu, vizualiziranje i distribuiranje informacija o objektima i pojavama, čiji referentni sustav je definiran na površini Zemlje.

2. Marble & Peuquet te DJ Maguire imaju svoje definicije GIS-a. Koje su?

Marble & Peuquet: četiri podsustava:

1. Sustava za unos podataka skuplja i/ili procesira geoprostorne podatke prikupljene iz postojećih mapa, senzora, itd.
2. Sustav za pohranjivanje i dohvat podataka organizira i pohranjuje podatke u obliku koji omogućuje brz dohvat za buduće analize, kao i brza i točna ažuriranja podataka u geoprostornoj bazi podataka
3. Sustav za rukovanje i analizu podataka koji obavlja razne zadaće kao npr. Promjenu formata podataka koristeći korisnički definirana agregacijska pravila ili procjena parametara različitih geoprostornih simulacija i sl.
4. Sustav za izvještavanje koji može prikazati cijelu ili dio baze podataka, ako i upravljati podacima i ostvariti izlaz bilo u tabličnom ili kartografskom obliku.

DJ Maguire:

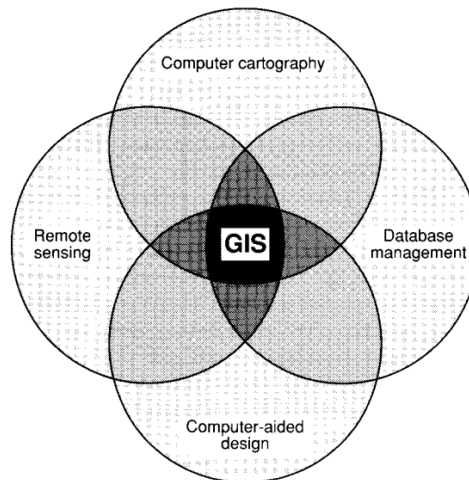


Fig. 1.1 The relationship between GIS, computer-aided design, computer cartography, database management and remote sensing information systems.

3. Na koja pitanja odgovaraju GIS i povezane tehnologije?

Što se nalazi na nekoj lokaciji? Gdje je nešto? Što se promijenilo od...? Koji geoprostorni obrasci postoje? Koji su odnosi između dva ili više skupa podataka koji se odnose na istu lokaciju? Koje geografske varijacije postoje s obzirom na prostor? Što ako...? "What if?"

4. Koji su problemi s modeliranjem geoprostornih objekata u relacijskom modelu?

Nepostojanje relevantnih geometrijskih operacija/funkcija. Rekonstrukcija objekata rezultira kompleksnim upitima i skupim operacijama spajanja - loše performanse sustava. Povećanje geometrijske kompleksnosti objekata neminovno vodi povećanju kompleksnosti modela

5. Opiši SUGBP

Implementiran proširenjem objektno-relacijskog SUBP

Posjeduje skup geoprостornih apstraktnih tipova podataka (GeoATP) kao i upitni jezik koji podržava te tipove podataka i operacije nad njima

Prostorno indeksiranje, djelotvorni algoritmi za operacije nad geoprостornim tipovima podataka

Specifična pravila za optimiranje upita

6. Definiraj topologiju

- geoprостorni odnosi između susjednih objekata. Topologija govori gdje su objekti s obzirom na jedan drugoga, te kako se odnose. Ti odnosi mogu biti jednostavni (npr. udaljenost), ali uključuju i složenije pojmove kao što su susjednost i povezanost.

7. Vrste podataka za geoprостorne baze?

1. Geometrijski (točka, linija, poligon) - Iz njih se izvode složeniji oblici:

- mreže linija (npr. ceste), mreže poligona (npr. mreža županija), plohe, prostorna tijela

2. Grafički (boja, šrafura, simbol, vrsta linije, ...)

3. Opisni oblik - dodatni opisni negeometrijski podatci, npr. nazivi, kućni brojevi, itd.

8. Nabroji i objasni temeljna obilježja geoprостornih objekata.

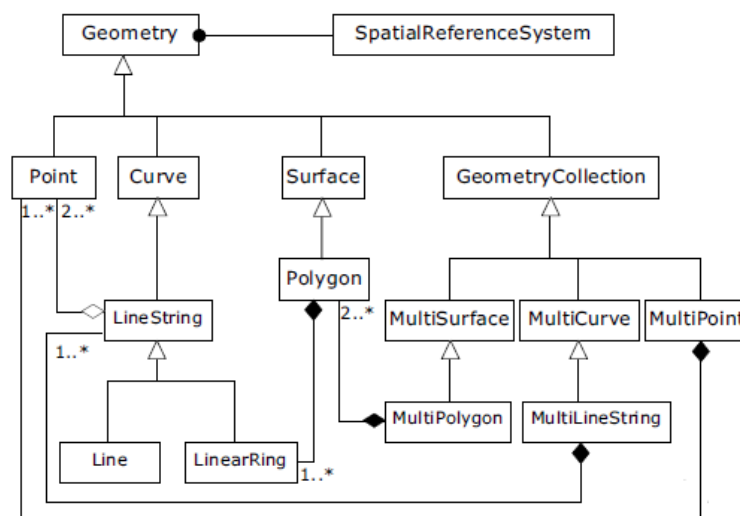
Geometrijska

- Metrička
 - Oblik (apstrakcija geometrijske strukture: točka, linija, poligon)
 - Položaj (u odnosu na referentni koordinatni sustav)
 - Veličina (0D, 1D, 2D ili 3D)
- Topološka
 - Relacije među objektima (susjedstvo, povezanost i sl.)

Tematska - Atributi čije su domene jednostavni tipovi podataka (integer, char ...)

9. Što je OGC standard? Kako izgleda UML objektni model geom. tipova podataka?

OGC - Open Geospatial Consortium, standard u kojem su definirani pojmovi iz geoprостornih tipova podataka



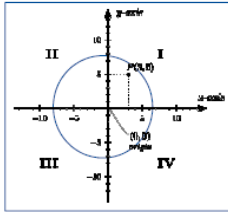

10. Koje tipove podataka raspoznaje postGIS?

Geometry, Geography, Raster i Vector

11. Usporedba rastera i vektora?

Raster	Vector
lako razumljiv	intuitivno
brzo procesiranje	rezolucija
oblik podataka	topologija
dodatne funkcije za obradu	pohrana
izgled	geometrija je složena
preciznost	spor odziv
veličina	sporiji razvoj (inovacije)

12. Usporedba geography i geometry tipa podataka?

Geometry	Geography
	

Geography

- koristi geografske koordinate umjesto Kartezijevog sustava
- koordinate su uvijek prikazane u WGS 84 lon/lat (SRID * 4326) sustavu
- funkcije za mjerenje uvijek rade u metrima (ST_Distance, ST_DWithin, ST_Length, ST_Area)
- jednostavna mjerenja i odnosi na relativno velikom području

Geometry

- puno bogatiji skup funkcija, provjere odnosa su u pravilu brže, bolja podrška u postojećim alatima

13. Kako izgleda topology tip u PostGIS-u?

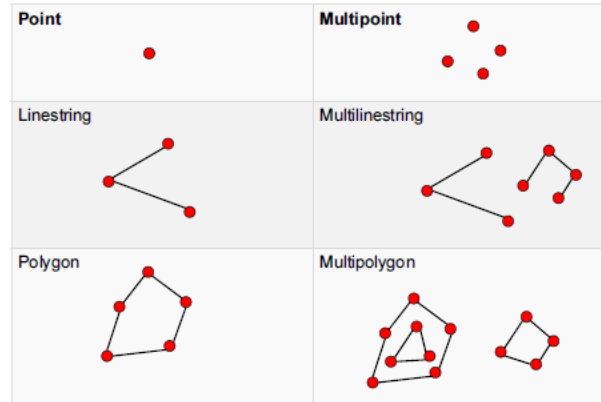
- Integritet podataka (npr. jedinstvene granice između parcela)
- Smanjenje prostora potrebnog za pohraniti podatke (granica se pohranjuje samo jednom)
- Eksplicitni prostorni odnosi (za svaki edge se zna lijevi i desni face, za svaki node se zna kojem face pripada, itd.)

Npr. Dodiruju li se parcela A i B?

Da -> imaju zajednički edge!

14. Nabroji PostGIS (pod)tipove podataka.

Point, Multipoint, Linestring, Multilinestring, Polygon, Multipolygon



15. Kako dijelimo GeoATP operacije?

- 1) Geometrijske operacije
 - a) Skupovne (unija, presjek)
 - b) Aritmetičke (duljina krivulje, površina poligona)
 - c) Druge (buffer, convex hull)
- 2) Topološke relacije (touches, within, disjointed ...)
- 3) Operacije nad grafovima (traženje najkraćeg puta)

16. Objasni Model 9 presjeka

Binarna topološka relacija **R** između dva prostorna objekta **A** i **B** opisuje se usporedbom unutrašnjosti (A^0), granice (∂A) i vanjštine (A^-) dvaju objekata.

Tih 6 komponenata moguće je kombinirati tako da oblikuju 9 temeljnih vrijednosti (presjeka) za opis topoloških relacija. Svaki presjek može poprimiti vrijednosti 0 i $\neg 0$.

Uređeni skup 9 presjeka može se prikazati matricom:

$$R(A,B) = \begin{pmatrix} A^0 \cap B^0 & A^0 \cap \partial B & A^0 \cap B^- \\ \partial A \cap B^0 & \partial A \cap \partial B & \partial A \cap B^- \\ A^- \cap B^0 & A^- \cap \partial B & A^- \cap B^- \end{pmatrix}$$

17. Raspiši topološke relacije između 2 poligona i između poligona i linije.

Topološke relacije između 2 poligona:

Figure 1 illustrates the four basic types of relationships between two sets A and B , each represented by a Venn diagram and a corresponding matrix.

Top Row:

- Diagram 1:** A and B are disjoint. A is dark grey, B is light grey.
- Diagram 2:** A contains B . A is dark grey, B is white.
- Diagram 3:** B contains A . A is white, B is dark grey.
- Diagram 4:** A and B are identical. A and B are white.

Bottom Row:

- Diagram 5:** A and B are disjoint. A is light grey, B is dark grey.
- Diagram 6:** A contains B . A is white, B is dark grey.
- Diagram 7:** B contains A . A is dark grey, B is white.
- Diagram 8:** A and B are identical. A and B are dark grey.

Each diagram is accompanied by a matrix with rows A^+ , ∂A , A^- and columns B^+ , ∂B , B^- .

Matrix 1 (Top Left):

$$\begin{matrix} A^+ & \begin{pmatrix} \emptyset & \emptyset & -\emptyset \end{pmatrix} \\ \partial A & \begin{pmatrix} \emptyset & \emptyset & -\emptyset \end{pmatrix} \\ A^- & \begin{pmatrix} -\emptyset & -\emptyset & -\emptyset \end{pmatrix} \end{matrix}$$

Matrix 2 (Top Middle-Left):

$$\begin{matrix} A^+ & \begin{pmatrix} -\emptyset & -\emptyset & -\emptyset \end{pmatrix} \\ \partial A & \begin{pmatrix} \emptyset & \emptyset & -\emptyset \end{pmatrix} \\ A^- & \begin{pmatrix} \emptyset & \emptyset & -\emptyset \end{pmatrix} \end{matrix}$$

Matrix 3 (Top Middle-Right):

$$\begin{matrix} A^+ & \begin{pmatrix} -\emptyset & \emptyset & \emptyset \end{pmatrix} \\ \partial A & \begin{pmatrix} -\emptyset & \emptyset & \emptyset \end{pmatrix} \\ A^- & \begin{pmatrix} -\emptyset & -\emptyset & -\emptyset \end{pmatrix} \end{matrix}$$

Matrix 4 (Top Right):

$$\begin{matrix} A^+ & \begin{pmatrix} -\emptyset & \emptyset & \emptyset \end{pmatrix} \\ \partial A & \begin{pmatrix} \emptyset & -\emptyset & \emptyset \end{pmatrix} \\ A^- & \begin{pmatrix} \emptyset & \emptyset & -\emptyset \end{pmatrix} \end{matrix}$$

Matrix 5 (Bottom Left):

$$\begin{matrix} A^+ & \begin{pmatrix} \emptyset & \emptyset & -\emptyset \end{pmatrix} \\ \partial A & \begin{pmatrix} \emptyset & -\emptyset & -\emptyset \end{pmatrix} \\ A^- & \begin{pmatrix} -\emptyset & -\emptyset & -\emptyset \end{pmatrix} \end{matrix}$$

Matrix 6 (Bottom Middle-Left):

$$\begin{matrix} A^+ & \begin{pmatrix} -\emptyset & -\emptyset & -\emptyset \end{pmatrix} \\ \partial A & \begin{pmatrix} \emptyset & -\emptyset & -\emptyset \end{pmatrix} \\ A^- & \begin{pmatrix} \emptyset & \emptyset & -\emptyset \end{pmatrix} \end{matrix}$$

Matrix 7 (Bottom Middle-Right):

$$\begin{matrix} A^+ & \begin{pmatrix} -\emptyset & \emptyset & \emptyset \end{pmatrix} \\ \partial A & \begin{pmatrix} -\emptyset & -\emptyset & \emptyset \end{pmatrix} \\ A^- & \begin{pmatrix} -\emptyset & -\emptyset & -\emptyset \end{pmatrix} \end{matrix}$$

Matrix 8 (Bottom Right):

$$\begin{matrix} A^+ & \begin{pmatrix} -\emptyset & -\emptyset & -\emptyset \end{pmatrix} \\ \partial A & \begin{pmatrix} -\emptyset & -\emptyset & -\emptyset \end{pmatrix} \\ A^- & \begin{pmatrix} -\emptyset & -\emptyset & -\emptyset \end{pmatrix} \end{matrix}$$

Između poligona i linije:

Figure 1 illustrates the decomposition of the boundary operator ∂ on a genus-3 surface. The figure is organized into a 3x5 grid. Each row represents a different decomposition of the boundary into two components, A and B . The diagrams show the surface with A and B as submanifolds. The matrices show the boundary operator ∂ acting on the chain complex $A^\circ \rightarrow \partial A \rightarrow A^-$.

Row 1: A is a disk, B is a disk with two handles.

Diagram 1: A is a disk, B is a disk with two handles.

Matrix 1:

$$\begin{matrix} A^\circ & \begin{pmatrix} B^+ & \partial B & B^- \\ \partial A & \begin{pmatrix} \emptyset & \emptyset & -\emptyset \\ \emptyset & \emptyset & -\emptyset \\ -\emptyset & -\emptyset & -\emptyset \end{pmatrix} \\ A^- \end{pmatrix} \end{matrix}$$

Row 2: A is a disk, B is a disk with one handle.

Diagram 2: A is a disk, B is a disk with one handle.

Matrix 2:

$$\begin{matrix} A^\circ & \begin{pmatrix} B^+ & \partial B & B^- \\ \partial A & \begin{pmatrix} -\emptyset & -\emptyset & -\emptyset \\ \emptyset & \emptyset & -\emptyset \\ -\emptyset & -\emptyset & -\emptyset \end{pmatrix} \\ A^- \end{pmatrix} \end{matrix}$$

Row 3: A is a disk, B is a disk with no handles.

Diagram 3: A is a disk, B is a disk with no handles.

Matrix 3:

$$\begin{matrix} A^\circ & \begin{pmatrix} B^+ & \partial B & B^- \\ \partial A & \begin{pmatrix} \emptyset & \emptyset & -\emptyset \\ \emptyset & -\emptyset & -\emptyset \\ -\emptyset & \emptyset & -\emptyset \end{pmatrix} \\ A^- \end{pmatrix} \end{matrix}$$

18. Raspiši dimenzijski prošireni model 9 presjeka.

- Dimenzijski prošireni model 9 presjeka (DE-9IM)
 - Pored operatora unutrašnjosti ($^{\circ}$), granice (∂) i vanjšine ($^-$), uvodi se i operator dimenzije

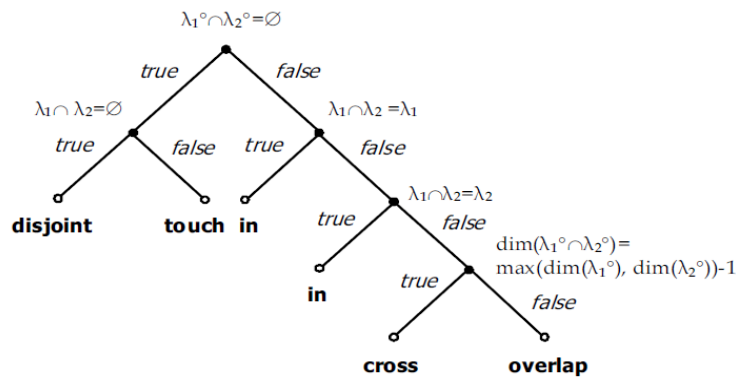
$$\dim(S) = \begin{cases} - & \text{ako je } S = \emptyset \\ 0 & \text{ako } S \text{ sadrži barem točku, ali ne i liniju i površinu} \\ 1 & \text{ako } S \text{ sadrži barem liniju, ali ne površinu} \\ 2 & \text{ako } S \text{ sadrži barem površinu} \end{cases}$$

*S – opći skup točaka

- Svaki element matrice proširuje se dimenzijom
 - Novu matricu moguće je zapisati ovako:

$$DE9I = \begin{pmatrix} \dim(\partial\lambda_1 \cap \partial\lambda_2) & \dim(\partial\lambda_1 \cap \lambda_2^{\circ}) & \dim(\partial\lambda_1 \cap \lambda_2^-) \\ \dim(\lambda_1^{\circ} \cap \partial\lambda_2) & \dim(\lambda_1^{\circ} \cap \lambda_2^{\circ}) & \dim(\lambda_1^{\circ} \cap \lambda_2^-) \\ \dim(\lambda_1^- \cap \partial\lambda_2) & \dim(\lambda_1^- \cap \lambda_2^{\circ}) & \dim(\lambda_1^- \cap \lambda_2^-) \end{pmatrix}$$

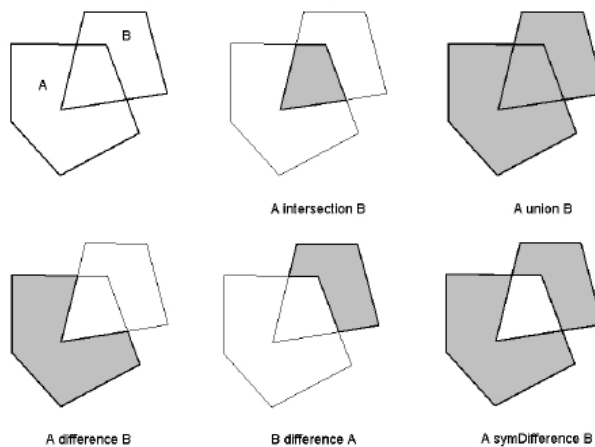
19. Nacrtaj stablo odlučivanja za topološke relacije. Koja su im svojstva?



20. Nabroji skupovne geometrijske operacije.

unija, presjek

- Skupovne geometrijske operacije



- $A \cup B = \{x \in \mathbb{R}^2 \mid x \in A \vee x \in B\}$
- $A \cap B = \{x \in \mathbb{R}^2 \mid x \in A \wedge x \in B\}$
- $A - B = \{x \in \mathbb{R}^2 \mid x \in A \wedge x \notin B\}$
- $A \oplus B = \{x \in \mathbb{R}^2 \mid (x \in A \wedge x \notin B) \cup (x \in B \wedge x \notin A)\}$

=Symmetric difference

21. Što je konveksna ljuska?

Konveksna ljuska skupa točaka S jest najmanji konveksni skup točaka (poligon) za koji je svaka točka skupa S ili na granici ili u unutrašnjosti tog poligona

Konveksni skup točaka (poligon) - linija povučena između bilo koje dvije točke skupa u potpunosti se nalazi u tom skupu

22. Koju geometrijsku operaciju implementira Voronijev dijagram?

Za svaku točku ravnine odrediti koja joj je od N točaka iz skupa S najbliža.

$$\text{voronoi}(t_k) \Leftrightarrow \{x \in \mathbb{R}^2 \mid d(t_k, x) < d(t_i, x), \forall i \neq k\}$$

23. Kakva se struktura koristi za indeksiranje prostornih podataka?

Za indeksiranje prostornih podataka koristi se **R stablo** koje je slično B stablima. Indeksira se položaj objekta u bazi (koordinate). Dijeli prostor na minimalne ograničavajuće pravokutnike koje zovemo MBR (minimal bounding rectangle/box).

Svaki čvor može sadržavati više elemenata. Element unutarnjeg čvora sadrži pokazivače na svoju djecu te MBR unutar kojeg se njegova djeca nalaze. Element lista sadrži identifikator objekta te MBR unutar kojeg se taj objekt nalazi.

24. Nabroji OGC web service

- **WMS**
 - Web Mapping
 - za iscrtavanje vektorskih i rasterskih podataka kao slike (JPEG, PNG, ...)
- **WFS**
 - Web Feature Service
 - vraća vektorske podatke u nekom XML formatu (GML, KML) ili JSON formatu (GeoJSON)
- **WFS-T**
 - uređivanje vektorskih podataka u transakcijskom stilu
- **Web Tiling Services, Web Coverage Services ...**

6. NoSQL

1. Što radi SUBP?

Skriva od korisnika detalje fizičke pohrane podataka, omogućuje definiciju i rukovanje s podacima, obavlja optimiranje upita i funkciju zaštite podataka (integritet podataka, pristup podacima i osigurava potporu za upravljanjem transakcijama)

2. Što je transakcija? Tko upravlja transakcijama? Implicitne granice?

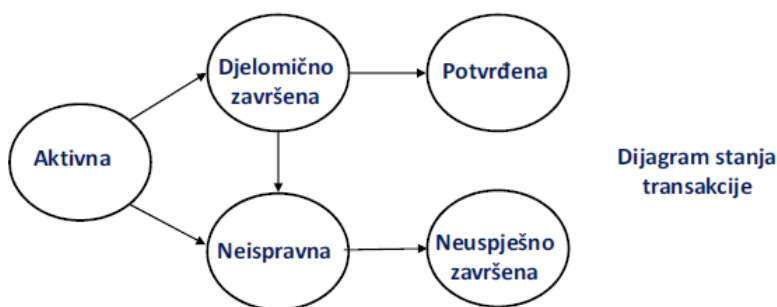
Transakcija je jedinica rada nad bazom podataka.

Upravljač transakcijama (TP monitor) - dio sustava koji brine o obavljanju transakcija i osigurava zadovoljavanje svih poznatih pravila integriteta.

Ako granice nisu eksplicitno definirane naredbama BEGIN/COMMIT/ROLLBACK tada se granice određuju implicitno - svaka SQL naredba se smatra transakcijom za sebe (naročito važno :

UPDATE, DELETE, INSERT u slučajevima kada djeluju nad skupom n-torki)

3. Nacrtaj dijagram stanja transakcija.



4. Koja su svojstva transakcije? (ACID) Opiši svako.

- **Atomicity** - nedjeljivost transakcije (mora se obaviti u cjelosti ili se uopće ne smije obaviti)
- **Consistency** - konzistentnost (transakcijom baza podataka prelazi iz jednog konzistentnog stanja u drugo)
- **Isolation** - izolacija (kod paralelnog izvršavanja, učinak mora biti jednak ako da su se izvršavale sljedno)
- **Durability** - izdržljivost (ako je transakcija obavila svoj posao, njezin učinak ne smije biti izgubljen ako dođe do kvara sustava)

5. Napiši općenito pravilo koje omogućuje obnovu baze podataka.

Redundancija - svaki podatak se mora moći rekonstruirati iz nekih drugih informacija redundantno pohranjenih negdje drugdje u sustavu.

6. Koji je općeniti postupak koji omogućuje obnovu?

1. periodičko kopiranje sadržaja baze podataka na arhivski medij
2. svaka izmjena u bazi podataka se evidentira u logičkom dnevniku izmjena

7. Nabroji i opiši tipove pogrešaka.

- **pogreške transakcija** (transaction failure) - pogreške koje su posljedica neplaniranog prekida transakcije
- **pogreška računalnog sustava** (system failure) - baza podataka nije fizički uništena
- **kvar medija za pohranu** (media failure) - baza podataka je fizički uništena

8. Nabroji i opiši još 3 mehanizma obnove.

Zrcaljenje podataka (mirroring) - dva su područja (primarno i zrcalno) te se promjene provode istovremeno u oba područja, u slučaju pogreške u jednom od područja se nastavlja raditi na ispravnom području

Arhiviranje tijekom obavljanja transakcija -

inkrementalno arhiviranje - omogućava stvaranje arhiva različitih razina (0. kopija čitave baze, 1. tjedna arhiva, 2. dnevna arhiva)

9. Prednosti i mane relacijskih baza podataka.

prednosti:

- prezistencija
- istodobni pristup, ACID
- integracija
- standardni model podataka, upitni jezik

mane:

- impedance mismatch
- velike količine podataka - skalabilnost
- dostupnost

10. Što je skalabilnost i kakve postoje?

Mogućnost sustava da se nosi s rastućom količinom podataka. Postoji **vertikalna** (dodavanje resursa jednom čvoru) i **horizontalna** (dodavanje čvorova u sustav).

11. Što su distribuirana BP i distribuirani SUBP?

DBP - skup logički povezanih baza podataka razmještenih u različitim čvorovima računalne mreže (LAN, MAN, WAN)

DSUBP - programski sustav koji upravlja distribuiranom bazom podataka na takav način da je distribuiranost sustava transparentna prema korisnicima.

12. Kako se oblikuju distribuirane baze podataka?

podaci se razmještaju u čvorove u kojima se najčešće koriste (minimalizira se mrežni promet)

oblikovanje distribucije = fragmentacija + alokacija

fragmentacija - podjela baze podataka u disjunktni skup fragmenata koji obuhvaćaju sve podatke u bazi podataka uz zadovoljenje pravila da se baza podataka može rekonstruirati iz tih fragmenata bez gubitka informacije

alokacija - shema kojom se opisuje koji je fragment pridružen kojem čvoru distribuiranog sustava

13. Transakcija u DSUBP?

- U svakom čvoru se nalazi zasebni, potpuno funkcionalan SUBP
- Transakcija se više ne može promatrati kao (samo) niz logički povezanih operacija koje se izvršavaju u jednom SUBP-u
- Globalna transakcija je skup subtransakcija koje koordinirano izvršavaju SUBP-ovi u više čvorova i pri tome prevode distribuiranu bazu podataka iz jednog u drugo konzistentno stanje

14. Opiši 2PC. Kakva svojstva ima?

2PC - two-phase commit (protokol dvofaznog potvrđivanja)

- u svakom čvoru nalazi se **menadžer transakcija (TM)** - zadaće jednake onima u centraliziranom sustavu: obnova, izolacija, ... razlika: osim lokalnih transakcija, obavljaju se i subtransakcije koje se izvršavaju na njegovoj lokaciji
- u svakom čvoru nalazi se **koordinator transakcija (TC)** - pokreće globalnu transakciju koja ima izvor na njegovoj lokaciji

1. FAZA - TC šalje svim TM poruku *pripremiPotvrđivanje*. Svaki pojedini TM odgovara *spreman* ili *nespreman* ili ne odgovara

2. FAZA

- ako je sa svih lokacija stigla poruka *spreman* TC odluku upisuje u svoj dnevnik i svim TM šalje poruku *globalnoPotvrdi*
 - u slučaju da je neki od TM odgovorio *nespreman* ili se tijekom zadanog vremena neki od TM nije odazvao TC odluku upisuje u dnevnik i svim TM šalje poruku *globalnoPoništi*
 - TM zapisuju odluku TC-a u svoj dnevnik, prema TC-u šalju potvrdu o prihvaćanju odluke i potvrđuju ili poništavaju subtransakcije
 - kada TC dobije potvrde svih TM, u logički dnevnik upisuje oznaku *krajPotvrđivanja*
- protokol je blokirajući i nije nezavisan obzirom na mogućnost obnove

15. Napiši moguće kvarove u DSUBP.

Osim kvarova koji su karakteristični za centralizirane sustave (npr. pogreške programske podrške, sklopovlja, uništenja diska), u DSUBP-u su moguće dodatne vrste kvarova:

- prestanak rada jednog ili više čvorova
- gubitak veze među čvorovima
- gubitak poruka
- podjela mreže: mreža je podijeljena (particionirana) kad je podijeljena u nekoliko podsustava koji međusobno ne mogu komunicirati. Naročiti problem: čvor Si ne može utvrditi je li se dogodila podjela mreže ili je neki čvor Sj prestao raditi

16. Koji su nedostaci DSUBP u odnosu na centralizirani SUBP?

- Bitno veća složenost sustava
- Povećanje troškova (npr. skuplja programska podrška, potreba angažiranja većeg broja administratora sustava)
- Veći problemi sigurnosti
- Veći troškovi u osiguravanju integriteta podataka
- Nedostatak standarda
- Nedostatak iskustva
- Povećanje složenosti postupka projektiranja baze podataka
- Loša implementacija distribuirane baze podataka može uzrokovati povećanje komunikacijskih troškova, smanjenje dostupnosti podataka i smanjenje performansi

17. Kakva je to replicirana baza podataka? Prednosti i mane?

- fragment je repliciran ako je alociran u dva ili više čvorova
- za jedan logički element x (n-torku, fragment, relaciju) postoji više fizičkih elemenata (kopija, replika), x1, x2, ..., u čvorovima S1, S2, ...

prednosti:

- povećanje dostupnosti
- smanjenje volumena prijenosa podataka
- paralelno obavljanje dijelova istog upita

mane: - problem konzistentnosti kopija istog elementa

18. Objasniti 4 vrste protokola za osiguranje konzistentnosti.

Sinkroni (eager) - sve fizičke operacije koje proizlaze iz logičkih operacija inicijalne transakcije obavljaju se unutar granica inicijalne transakcije, tj. sve kopije se moraju izmijeniti u okviru inicijalne transakcije

Asinkroni (lazy) - operacije inicijalne transakcije obavljaju se isključivo u inicijalnom čvoru i niti na koji način ne ovise o komunikaciji s ostalim čvorovima, inicijalna transakcija može završiti prije nego su obavljene izmjene nad svim kopijama. Izmjene ostalih kopija obavljaju se asinkrono

Jednosmjerni - za svaki logički element x određuje se samo jedan fizički element x_p koji se proglašava primarnom kopijom, te se operacija izmjene koju nad elementom x obavlja inicijalna transakcija mora prvo obaviti nad kopijom x_p

Dvosmjerni - inicijalna transakcija može operaciju izmjene obaviti nad bilo kojom fizičkom kopijom, dostupnost sustava se bitno povećava u odnosu na jednosmjerne sustave

19. Nabroji nedostatke dvosmjernih asinkronih protokola.

- neserijski obavljanje transakcija može dovesti do teško ispravljivog narušavanja konzistentnosti podataka
- problem detekcija konflikata: neki konflikti mogu biti otkriveni tek nakon propagacije izmjena (kad je inicijalna transakcija već potvrđena)
- problem razrješavanja konflikata: može zahtijevati poništavanje potvrđene transakcije -> narušavanje svojstva izdržljivosti transakcije
- automatsko rješavanje konflikata često nije moguće - potrebna je intervencija čovjeka

20. Nabroji modele podataka u NoSQL svijetu i njihove reprezentante.

1. Ključ-vrijednost (Key Value)
2. Dokument (Document)
3. Column family (<> column, columnar)
4. Graf (Graph)

21. Što je agregat?

Agregat je: - Složeni zapis koji dozvoljava liste, gniježđenje drugih zapisa

- Skup objekata koji se obrađuju kao jedan zapis (npr. narudžba i stavke narudžbe)

22. Ključ-vrijednost baze podataka. Model, operacije, primjer, prednosti, nedostaci.

Model podataka: (ključ, vrijednost) parovi

Operacije: - Unos(k, v), Dohvat(k), Ažuriranje(k, v), Brisanje(k)

- Neki podražavaju određenu strukturu vrijednosti, attribute vrijednosti
- Neki podržavaju dohvat na temelju raspona ključeva

Neki primjeri: Riak, Dynamo,...

Prednosti: - Jednostavno HTTP sučelje („Riak speaks web“)

- Fault tolerant
- Skalabilnost (lako je dodati novi čvor u klaster, automatska distribucija podataka, „a near-linear performance increase as you add capacity“)

Mane: - Ad-hoc upiti, povezivanje zapisa (ref.int.)

23. Dokument baze podataka: model, operacije, primjer, prednosti, mane

Model podataka: (ključ, dokument)

Dokument: JSON, BSON, XML, YAML, neki drugi polustrukturirani format, binarni podaci

Operacije: - Unos(k, d), Dohvat(k), Ažuriranje(k, d), Brisanje(k)

Primjeri: CouchDB, MongoDB, SimpleDB,...

Prednosti: - dohvat na temelju upita

- dohvat dijela dokumenta
- indeksiranje

Mane: - ograničenja na sadržaj

24. CF baze podataka: model, operacije, primjer, prednosti, mane

Model podatka: column family

- Nije tablica
- Dvorazinska mapa, dvorazinski agregat
- Prvorazinski ključ: ključ retka
- Drugorazinski ključ: ključ stupca
- Svaki stupac je član neke familije stupaca

Primjer : HBase, wiki

Prednosti : Skalabilnost, održavanje verzija, kompresija, memorijski rezidentne tablice, *Fault tolerant*

Mane : - ne može se raditi INSERT, sporija izgradnja indeksa 2-3 puta u odnosu na B-stablo

25. Graf baze podataka: model, operacije, primjer, prednosti, mane

Model podataka: čvorovi, bridovi, svojstva:

- Čvorovi mogu imati svojstva (KV parovi)
- Bridovi imaju oznaku, smjer, početni i odredišni čvor
- Bridovi također mogu imati svojstva

Primjeri: Neo4j, GraphDB, DEX, FlockDB, InfoGrid, OrientDB, Pregel, ...

Upitni jezici : Cypher

Prednosti: - Pogodni za složene, polu-strukturirane, jako povezane podatke

Mane: - Nisu pogodne za distribuciju

26. Prednosti i mane schemaless baza podataka?

prednosti: - Moguće je pohraniti "bilo što"

- Lako je mijenjati "što se sprema", dodati nove podatke
- Lako je raditi s heterogenim podacima

mane: - BP ne poznaje implicitnu shemu - ne može iskoristiti to znanje za efikasniju pohranu i dohvat

- BP ne može provoditi validaciju (integritet)
-

27. Što su materijalizirane virtualne relacije?

Materijalizirane virtualne relacije - upiti izračunati unaprijed i pohranjeni na disku

28. Koje su prednosti i mane distribucije podataka? koje su vrste? Opiši ih.

Prednosti : - Obradivanje veće količine podataka, Veći R/W promet, Veća dostupnost

Mane: - Složenost, novi problemi

Vrste:

- **Fragmentacija (sharding)** - združiti podatke kojima se često pristupa
 - po serverima raspodjeljujemo - geografski, jednoliko, preko domenskih pravila...
 - Poboljšava i čitanje i pisanje
 - Ne poboljšava otpornost sustava na pogreške, čak suprotno (treba održavati više strojeva!)
- **Replikacija (replication)**
 - **Master-Slave** - Korisno za skaliranje kada imamo puno čitanja
 - Read resilience - ako master prestane raditi i dalje se može čitati
 - Brz oporavak - odabir novog mastera (~ hot backup)
 - NEKONZISTENTNOST (što ako master propagacija prestane raditi?)
 - **Peer-to-Peer** - Korisno za skaliranje i za čitanje i za pisanje
 - ravnopravni čvorovi
 - Lako poboljšati performanse - dodati čvor!
 - NEKONZISTENTNOST

29. Navedi neka moguća rješenja za održavanje konzistencije kod pisanja.

Dva pristupa:

- Pesimistični - spriječiti WW konflikt (write locks)
- Optimistični - dopustiti, detektirati, razriješiti (npr. conditional update (prethodni primjer), automatic merge ~ CVS)

Oba pristupa se oslanjaju na konzistentnom poretku akcija - kod jednog servera trivijalno - odabire se jedan ili drugi update

Ali kako u distribuiranoj okolini? - W preko samo jednog čvora ili dopustiti više verzija, pa kasnije razriješiti

30. Vrste konzistencije kod čitanja?

Logička (ne)konzistencija = osigurati se da različiti objekti zajedno imaju smisla

Replikacijska (ne)konzistencija = osigurati da sve replike istog podatka imaju istu vrijednost

31. Opiši i objasni CAP teorem.

U distribuiranim sustavima je moguće ostvariti samo dva od tri navedena svojstva.

Consistency (Cap <> aCid) - Svaki odgovor poslan klijentu je točan

Availability - Svaki zahtjev koji je funkcionirajući server zaprimio mora rezultirati odgovorom (R & W)

Partition tolerance - Rad sustava i u uvjetima kada nastanu izolirane skupine računala

32. Opiši i objasni BASE.

Za mnoge primjene, dostupnost i toleriranje particija su važnije od stroge dosljednosti (npr. (velike) web aplikacije, tražilice, ...)

Basically Available - aplikacija je praktički uvijek dostupna (unatoč povremenim kvarovima)

Soft-state - ne mora uvijek biti konzistentan („mekano stanje”), sustav se stalno mijenja, protočan je

Eventual consistency - biti će, u konačnici, u nekom znanom stanju (izmjene će se, u konačnici, propagirati i svi će ih vidjeti)

33. Objasni kvorum.

- $N = 3$ (N je faktor replikacije \leftrightarrow broj čvorova)
- P2P model
- Za konzistenciju kod pisanja je dovoljna većina: $W > N/2$
- Konzistencija kod čitanja ovisi o W
- Možemo imati konzistenciju kod čitanja i kad nemamo kod pisanja - čitati sve čvorove! -> to ne znači da nećemo imati update conflict, ali ćemo ga detektirati
- Za konzistenciju kod čitanja je dovoljno: $R + W > N$

34. Kako Riak i MongoDB rješavaju problem konzistencije?

Riak : - n_val - faktor replikacije (default = 3), r - broj čitanja nakon kojeg se čitanje smatra uspješnim, w - broj pisanja nakon kojeg se pisanje smatra uspješnim

- Mogu se postaviti kod svakog zahtjeva!

MongoDB: - Opisuje razinu potvrde servera kod obavljanja zadane operacije { w : <value>, j : <boolean>, $wtimeout$: <number> }

w - broj instanci (0, 1, „majority”, <tag set>), j : zapisano u dnevnik, $wtimeout$: vremenski limit

35. Kako se rješava problem s Durability?

Riak:

- Kod pisanja:
- i. Objekt se prvo piše u memoriju (buffer)
 - ii. Potvrđuje se uspješno pisanje
 - iii. Objekt se piše na disk

MongoDB : $w = 0$

36. Za kakve probleme je pogodan Map/Reduce?

Pogodan za određenu vrstu (paralelnih) problema, npr.:

- pretraživanje velike količine teksta
- izgradnja indeksa riječi
- brojanje pristupa web stranicama

37. Objasni osnovnu ideju Map/Reduce. Koja su ograničenja?

map: je funkcija: $map(k1, v1) \rightarrow list(k2, v2)$

- argument: jedna ključ-vrijednost (agregat)
- rezultat: lista odnosno 0 ili više ($k2, v2$) parova
- Map funkcije se obavljaju nezavisno - paralelizam

reduce: je funkcija: $reduce(k2, list(v2)) \rightarrow list(v3)$

- argument: lista vrijednosti za neki ključ
- rezultat: 0 ili više vrijednosti, tipično 0 ili 1 vrijednost

ograničenja: Trade-off: - algoritamska fleksibilnost vs paralelizacija (tj. izračun na klasteru)

- U map funkciji možemo djelovati nad samo jednim agregatom
- U reduce funkciji možemo djelovati nad samo jednim ključem

38. Zadatak M/R Frameworka?

Orkestrira izvođenje:

- Upravlja čvorovima (isti čvorovi se mogu koristiti i za M i R fazu)
- Paralelno pokreće zadatke
- Upravlja razmjenom podataka
- Oporavak od pogreške (što ako neki čvor prestane raditi?)

Ili:

1. Priprema ulazne podatke za map funkciju, dodjeljuje i dostavlja map čvorovima
2. Pokreće korisničku map funkciju na čvorovima
3. Grupira, (particionira) i raspoređuje (shuffle) izlaz map funkcija i dostavlja reduce čvorovima
4. Pokreće korisničku reduce funkciju na čvorovima
5. Objedinjava sve rezultate reduce funkcije sa svih čvorova

39. Što je combiner? Combinable reducer?

Combiner funkcija sažima sve podatke s istim ključem u jedan zapis.

Combinable reducer je reduce funkcija čiji izlaz odgovara ulazu.

- Zamjenjuje combiner, poziva se odmah na istom map čvoru
- Mogu se pozivati prije nego što je u potpunosti gotova map faza

40. Ulančavanje M/R.

-Izlaz iz jedne M/R faze je ulaz u drugu fazu

-Izlaz iz prve faze se može snimiti (za druge potrebe) kao materijalizirani pogled

41. Svojstva MongoDB?

- Ad-hoc upiti, Indeksiranje
- MS replikacija, Sharding + load balancing
- File storage, Aggregation
- ServerSide Javascript, Language binding
- FLOSS

42. Kako se vrši detekcija konfliktata? Opiši svaki.

Oznake verzija (version stamps) - polje koje se mijenja kod svake promjene odgovarajućih podataka, npr. HTTP ETag

- Opcije za generiranje oznake: Brojač, GUID, Hash, timestamp, kombinacija prethodnih

Poredak i uzročna ovisnost događaja - Označiti sve događaje s vremenskom oznakom (fizički sat)?

- Problem sinkronizacije satova (npr. kvarni sat griješi ~ 1s/11.6 dana)
- Ne možemo odrediti uzročnu ovisnost (eng. causality)

Poredak (happened-before, Lamport 1978.), možemo znati samo na temelju:

(a) redoslijed internih događaja istog procesa

(b) redoslijed slanja i primanja iste poruke

Logički sat - Služe za određivanje redoslijeda događaja, događaju se dodjeljuje broj $C(e)$ (– fizičkom vremenu)

Lamportov sat

Vektorski sat - Opisuje uzročno posljedične veze

- Ideja: svaki proces ima svoj brojač (sat)
- $Vp[p]$ broj događaja koje je ostvario proces p
- $Vp[m]$ broj događaja za koje proces p smatra da ih je ostvario proces m

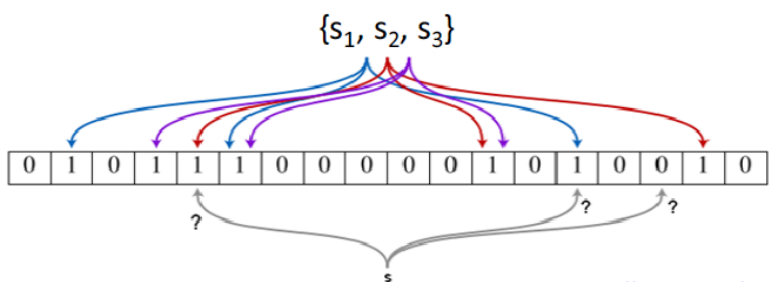
43. Što je Bloomov filter? Kako radi?

Prostorno učinkovita probabilistička podatkovna struktura koja se koristi za ispitivanje članstva elemenata u skupu.

$S = \{s_1, s_2, \dots, s_n\}$, S predstavljamo poljem od m bitova (inicijalno 0)

k nezavisnih funkcija raspršenog (uniformnog) adresiranja: h_1, h_2, \dots, h_k s rasponom $\{0, m-1\}$

Elementi se mogu dodavati u skup (ali se ne mogu izbacivati)



Dodavanje s u skup: - postaviti $h_1(s), h_2(s), \dots, h_k(s)$ bitove na 1

Provjera članstva za element s :

- (vjerojatno) jest član akko $\forall h_i(s)=1, i=1..k$
- inače: (sigurno) nije član

44. Što je Big data?

Oni podaci kojima se ne može upravljati, obrađivati ili analizirati koristeći tradicionalne metode i alate. Problemi uključuju: snimanje, organizaciju, pohranu, pretraživanje, dijeljenje, prijenos, analizu i vizualizaciju.*

3Vs: - Volume = volumen, veličina; Variety = različitost, heterogenost; Velocity = brzina

45. Nabroji prednosti NoSQL-a. (5)

- Elastic scaling - Horizontalno skalabilni
- Big Data - Omogućuje rad s BD-om

- Goodbye DBAs? - Automatski oporavak, podešavanje, distribucija
- Economics - Uglavnom besplatan, FLOSS softver, Temeljeni na commodity računalima
- Flexible Data Models - Schemaless

7. Tokovi podataka

1. Zašto su potrebni sustavi za upravljanje tokovima podataka?

Zbog toga što velike količine sirovih podataka pristižu velikom brzinom na obradu i potrebno ih je analizirati što prije.

2. Što su tokovi podataka i kakvi mogu biti?

Tok podataka je (potencijalno neograničen) slijed n-torki.

Može biti transakcijski tok podataka te tok podataka temeljen na mjerenjima.

3. Nabroji i objasni 3 modela tokova podataka

Postoji ulazni tok a_1, a_2, \dots - pristiže slijedno te opisuje signal $A:[1..N] \rightarrow \mathbb{R}^2$

Elementi a su podaci.

1) Time Series Model (vremenske serije)

svaki element $a(i)$ je jednak $A[i]$ te se pojavljuju u rastućem slijedu varijable i

2) Cash Register Model

elementi $a(i)$ su inkrementi $A[j]$ i tijekom vremena različiti elementi $a(i)$ mogu povećati vrijednost istog elementa $A[j]$

3) TurnStile Model

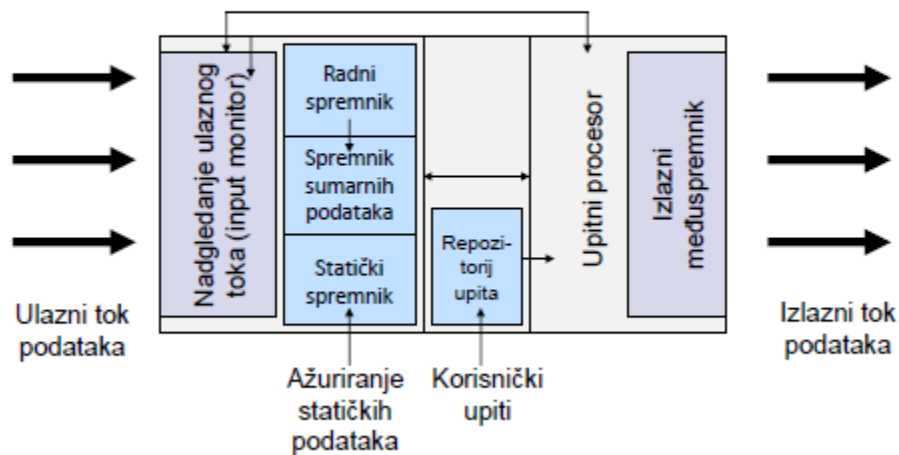
elementi $a(i)$ su ažuriranja elemenata $A[j]$ i tijekom vremena različiti elementi $a(i)$ mogu povećati ili smanjiti vrijednost istog elementa $A[j]$ (koristi se za promatranje potpuno dinamičnih pojmova; najopćenitiji model)

4. Usporedi SUBP i SUTP

SUBP	SUTP
pohranjeni skup relativno statičkih zapisa	podrška za on-line analizu brzo mijenjajućih tokova podataka
dobar za aplikacije koje zahtijevaju perzistentan spremnik podataka i kompleksne upite	tok podataka - stvarnovremenski, uređeni slijed elemenata, prevelik da se pohrani u potpunosti, potencijalno neograničen
	kontinuirani upiti

5. Nacrtaj dijagram općenite arhitekture SUTP

Općenita arhitektura SUTP



6. Koji su aplikacijski zahtjevi na SUTP?

- 1) model podataka i semantika upita (operacije temeljene na vremenu i redoslijedu)
 - selekcija
 - ugniježdjena agregacija
 - multipleksiranje i demultipleksiranje
 - upiti za pronalazak frekventnih elemenata
 - spajanja
 - upiti temeljeni na vremenskim prozorima
- 2) obrada upita (neblokirajući operatori, jednoprolazni algoritmi)
- 3) redukcija podataka (sumarne strukture za aproksimaciju)
- 4) stvarnovremenski odziv
- 5) kontinuirani upiti (varijabilna stanja sustava)
- 6) skalabilnost (raspodijeljeno izvršavanje velikog broja kontinuiranih upita, nadgledanje/obrada više paralelnih tokova podataka)
- 7) dubinska analiza tokova podataka

7. Kako se modelira vrijeme u SUTP?

eksplicitne vremenske oznake:

- zapisane na izvoru podataka; modeliraju događaj iz stvarnog svijeta reprezentiran n-torkom
- vrijeme valjanosti

implicitne vremenske oznake

- definiraju se kao zaseban atribut definiran u okviru SUTP
- vrijeme primitka podatka
- transakcijsko vrijeme
- omogućuju upite temeljene na redoslijedu i vremenskim prozorima

8. Što su interpunkcije?

- u toku podataka označavaju kraj podskupa podataka (razdvajaju pojedine elemente n-torke)
- odblokiravaju blokirajuće operatore
- smanjuju broj stanja za operatore ovisne o stanju

- održavaju redoslijed
- dozvoljavaju narušavanje redoslijeda (razgraničavaju podskupove)

9. Kakvi su zahtjevi na upite SUTP-a? Koje su paradigme?

- kontinuirani (perzistentni) upiti, prilagodljivi upitni operatori i plan izvršavanja
- aproksimativni rezultati upita
- grupno procesiranje podataka
- redukcija podataka (uzorkovanje, sažetci, skice, histogrami, wavelets

paradigme:

- relacijska
- objektna
- proceduralna

10. Obrada upita?

- neblokirajući operatori
- aproksimativni algoritmi
- algoritmi temeljeni na klizajućim prozorima
- on-line dubinska analiza tokova podataka u jednom prolazu

11. Navedi neke primjene SUTP (4)

- senzorske mreže
- analiza mrežnog prometa
- financijski podaci
- on-line aukcije
- analiza transakcijskih dnevnika

12. Navedi 6 osnovnih karakteristika Apache Sparka

- obrada skupnih, interaktivnih i stvarnovremenskih podataka na temelju istog programskog okvira
- ugrađena podrška za integraciju
- razvoj aplikacija na višoj razini apstrakcije
- općenitiji pristup
- lijena evaluacija
- funkcijsko programiranje/jednostavnost korištenja

13. Što je RDD?

Resilient Distributed Datasets

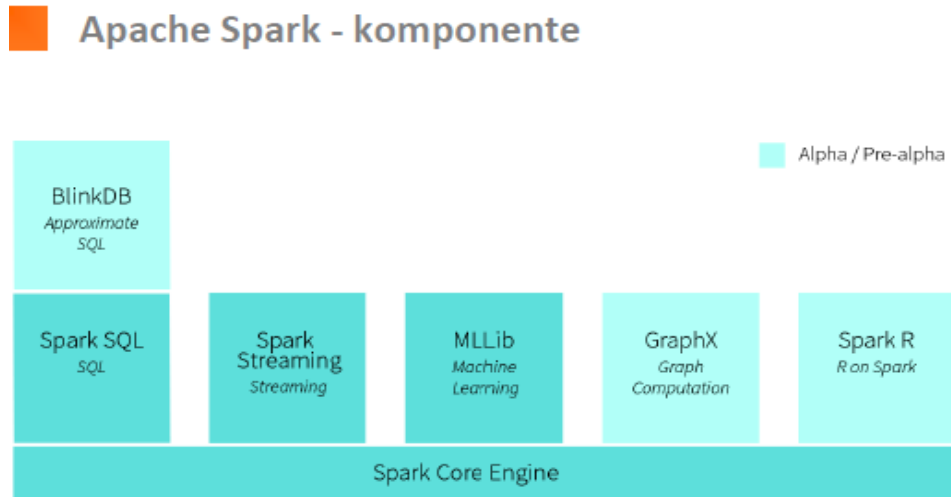
- primarna apstrakcija podataka u Spark-u
- kolekcija elemenata otporna na pogreške, koja se može obrađivati u paraleli
- 2 vrste: paralelizirane kolekcije i Hadoop skupovi podataka
- 2 vrste operacija: transformacije, akcije

14. Kako radi distribuirana obrada na Sparku?

- 1) Master se spaja na cluster manager cluster manager cluster i alocira potrebne resurse
- 2) Pokreće izvršitelje (executors) na čvorovima clustera - procesi izvršavaju obradu, spremaju podatke u priručnu memoriju (caching)
- 3) Master šalje aplikacijski kod izvršiteljima

4) Master šalje zadatke (tasks) izvršiteljima na obradu

15. Nabroji komponente Apache Sparka i objasni uloge svake komponente



8. Semantički web

1. Koja je razlika između WWW-a i Interneta?

Internet je mrežna infrastruktura, a World Wide Web je skup dokumenata pohranjen na poslužiteljima koji su povezani Internetom, te koji su dostupni putem HTTP protokola.

2. Što je semantički Web? Nacrtaj tehnologije semantičkog weba

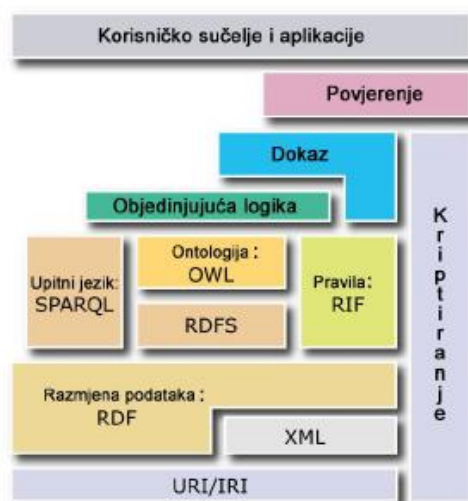
Ideja:

Semantički web je ideja o mogućnosti da podaci na webu budu definirani i povezani na način na koji bi se mogli koristiti osim za sami prikaz i za automatizaciju, integraciju i ponovnu iskoristivost u različitim aplikacijama. Na taj način obećava se mogućnost pronalaženja, sortiranja i klasificiranja informacija, dakle svih onih zadataka koje troše puno vremena provedenog on i off-line.

Danas razlikujemo 2 vizije:

- mreža podataka (otvaranje sadržaja baza podataka za web, objava podataka u nekom od otvorenih formata; potpuni je rješenje)
- mreža oznaka (označavanje podataka na webu; jednostavna semantika)

Semantički web - tehnologije



3. Nabroji tehnologije semantičkog weba

W3C standardi: URI/IRI, XML, RDF, RDFS, OWL, SPARQL, RDFa, RIF

4. Što je URI (u kontekstu semantičkog weba)?

Uniform Resource Identifier. U semantičkom webu svaki pojam (resurs), apstraktan ili opipljiv, ima jedinstveni identifikator koji se zove URI.

5. Što je RDF? Na čemu se temelji?

Resource Description Framework. "Okvir" koji omogućava predstavljanje informacija na Webu, namijenjen je za spremanje podataka o podacima. RDF se temelji na konceptu trojki (subjekt, predikat, objekt) koja predstavlja izjavu pri čemu je subjekt uvijek neki resurs, dok objekt može biti resurs ili podatkovna vrijednost.

6. Što je RDFS? Koji su osnovni elementi?

RDF Schema (stari naziv). RDFS definira značenja, karakteristike i odnose između dozvoljenih termina (omogućuje kreiranje vlastitog RDF rječnika) i omogućava izvođenje zaključaka iz RDFS pravila.

Osnovni elementi:

- Resource
- Class
- Literal
- Property
- domain, range

- subClassOf, subPropertyOf

7. Što je SPARQL?

SPARQL Protocol and RDF Query Language. SPARQL je upitni jezik za RDS, to jest protokol koji definira upotrebu SPARQL upitnog jezika preko HTTP-a koji je napravljen po uzoru na SQL. Uvažava i koristi specifičnosti strukture i načina pohrane podataka u RDF-u, te prijenosa podataka HTTP-om.

8. Od čega se sastoji SPARQL URL?

SPARQL URL se sastoji od tri dijela:

- 1) URL SPARQL endpoint
- 2) Imena grafova nad kojima se postavlja upit
- 3) SPARQL upit

Primjer SPARQL upita:

<http://dbpedia.org/sparql?default-graph-uri=http%3A%2F%2Fdbpedia.org&query=SELECT+distinct+%3Fb+WHERE+%7B%3Fb+a+%3Chttp%3A%2F%2Fumbel.org%2Fumbel%2Fsc%2FArtist%3E+>

9. Što je ontologija? Od čega se sastoji?

Ontologija je model podataka koji predstavlja skup pojmova unutar neke domene i veze između tih pojmova.

Ontologija se sastoji od rječnika pojmova koji opisuju neku domenu (nazivi bitnih koncepata iz domene) te znanja i ograničenja na pojmove iz domene.

10. Što je OWL? Od čega se sastoji? Koji zakoni logike vrijede u OWL-u?

Jezik Semantičkog Web-a stvoren da omogući predstavljanje bogatog i kompleksnog znanja o entitetima (stvarima, pojmovima), grupama entiteta i odnosima među entitetima.

Sastoji se od klasa (i njihove hijerarhije), svojstava koja te klase imaju, ograničenja i odnosa među svojstvima (tip, domena, kardinalnost, jednakost) i instanca klasa

Zakoni:

Vrijedi "Open World" pretpostavka formalne logike koja govori da ako nešto ne znamo, ne znači da je "laž". Dodatno, vrijedi da ne postoji "Unique Name" pretpostavka, odnosno moramo eksplicitno izreći vezu između entiteta.

11. Nabroji načela povezanih podataka

- svi elementi moraju biti identificirani URI-em (nema praznih čvorova)
- svi URI-i moraju se moći razriješiti (eng. dereference) -pronaći HTTP URL koji daje koristan opis elementa identificiranim zadanim URI-em
- moraju se postaviti veze prema drugim URI-ima kako bi se omogućila daljnja laka potraga za podacima

12. Što su vokabulari i zašto su važni?

Vokabulari čine podskup ontologije. Kako bi ostvarili integraciju podataka potrebno je međusobno razumijevanje pojmova, kategorija i veza između pojmova i kategorija.

13. Što je RDFa?

Predstavlja RDF trojke unutar XHTML dokumenata. Ugradnja RDF izjava u XHTML omogućena je korištenjem proširenja XHTML-a -sintakse RDFa. Za izgradnju RDF izjava koriste se atributi XHTML-a, kao i atributi RDFa sintakse.

14. Kako se vrši integracija podataka?

Osnovni koraci integracije podataka:

- 1) Dati značenje podacima -preslikati ih u uvriježeno značenje
- 2) Udružiti preslikane podatke
- 3) Obavljanje upita nad cijelim skupom podataka

15. Koji su problemi kod spajanja relacijske baze i RDF?

- kako odrediti URI, odgovara li on primarnom ključu?
- u bazi podataka iste trojke se mogu javiti više puta
- prilagođavanje tipova podataka odabranoj ontologiji
- šifrirani podaci (npr. spol 'M' i 'Ž')
- ponekad je potrebno spajati više vrijednosti atributa (npr. ime i prezime čine jednu vrijednost u ontologiji)
- NULL vrijednosti

16. Što je D2R?

D2R je alat za objavljivanje sadržaja relacijskih baza podataka na semantičkom webu. Za mapiranje podataka iz relacijske baze u RDF koristi se D2RQ. Upiti koji stižu na D2R Server se prepisuju u SQL upite, koji se izvršavaju na bazi u pozadini, rezultati se formatiraju u RDF i prikazuju korisniku u odabranom formatu

D2R omogućava tri različita “sučelja” za pristupa podacima:

- “Linked data sučelje”
- “SPARQL” sučelje
- klasični HTML pristup

