



Alternativni pristupi u izgradnji sistema baza podataka

*NoSQL, ugrađene i temporalne
baze podataka*

Sadržaj

- NoSQL baze podataka
- Ugrađene baze podataka
- Temporalne baze podataka

NoSQL baze podataka

- **Termin *NoSQL***

- nastao 2009. godine

- u opštem smislu, objedinjuje sve baze podataka i skladišta podataka koje ne slede primarne principe relacionih baza podataka

- *No SQL* ili *Not Only SQL*

- ne predstavlja jedan proizvod ili tehnologiju

- već klasu proizvoda i kolekciju različitih, ponekad povezanih koncepata o načinima skladištenja i manipulacije podacima

NoSQL baze podataka

- **Motivacija**

- pojava Interneta prouzrokuje

- eksploziju količine podataka koju je potrebno skladištiti

- problem **velikih količina podataka**

- » degradacija performansi RSUBP

- » količina podataka ima trend stalnog rasta

- » veličina pojedinačnih podataka takođe raste

- potreba za visokim nivoom **skalabilnosti**

- » zahteva se unapređenje skalabilnosti

- potreba za visokim nivoom **raspoloživosti**

- eksploziju broja istovremenih zahteva nad BP

- milioni istovremenih zahteva

- » odgovor mora biti u razumnom vremenskom roku

NoSQL baze podataka

- **Motivacija**

- pojam **veliki podaci** predstavlja kolekciju digitalnih informacija
 - *eng. big data*
 - čiji obim prevazilazi sposobnost prosečne percepcije ljudi pa i kapacitete raznih softverskih alata
 - za prihvatanje, obradu i upravljanje podacima
 - nije tačno određena donja granica veličine, definisane pojmom veliki podaci
 - trenutno - svaki skup podataka veći od nekoliko terabajta
- dužina transakcija postaje problem
 - nepredvidiva dužina transakcije
 - ACID osobine transakcija više ne odgovaraju zahtevima, vezanim za obradu velikih količina podataka

NoSQL baze podataka

- **ACID osobine**

- skraćenica od *eng. Atomicity, Consistency, Isolation, Durability*
- skup osobina BP
 - garantuju pravilno izvršenje transakcija
- definisane od strane Jim Gray-a
 - krajem 1970-ih godina
 - prilikom definisanja pouzdanog transakcionog sistema

NoSQL baze podataka

- **ACID osobine**

- **atomičnost**

- u slučaju uspešnog izvršenja transakcije
 - sve operacije obuhvaćene transakcijom će biti uspešno izvršene
 - baza podataka ostaje u konzistentnom stanju
 - u slučaju neuspešnog izvršenja transakcije
 - efekti izvršenja operacija obuhvaćenih transakcijom neće biti sačuvani u bazi podataka
 - baza podataka takođe ostaje u konzistentnom stanju
 - » onome, koje je prethodilo datoj transakciji

NoSQL baze podataka

- **ACID osobine**

- **konzistentnost**

- ukoliko su se podaci pre izvršenja transakcije nalazili u konzistentnom stanju, nalaziće se i po okončanju transakcije

- **izolacija**

- operacije tekuće transakcije ne utiču na druge transakcije
 - koje se istovremeno izvršavaju

- **trajnost**

- promene koje nastaju nakon završetka transakcije ostaju trajno sačuvane
 - čak i u slučaju pada čitavog sistema

NoSQL baze podataka

- **CAP teorema**

- sistem koji **skladišti deljene podatke** ne može obezbediti istovremeno zadovoljenje sledećih uslova
 - **konzistentnost**
 - *eng. Consistency*
 - **raspoloživost**
 - *eng. Availability*
 - **tolerancija razdvojenosti**
 - *eng. Partition tolerance*
- primenljiva na sisteme zasnovane na distribuiranoj arhitekturi

NoSQL baze podataka

- **CAP teorema**

- **konzistentnost**

- svako čitanje iz baze podataka kao rezultat ima najnoviju verziju podataka

- **raspoloživost**

- odziv sistema u garantovanim vremenskim okvirima
 - baza podataka će uvek biti dostupna
 - nezavisno kada je postavljen upit
 - postiže se
 - velikim brojem fizičkih servera
 - replikacijom podataka

NoSQL baze podataka

- **CAP teorema**

- **tolerancija razdvojenosti**

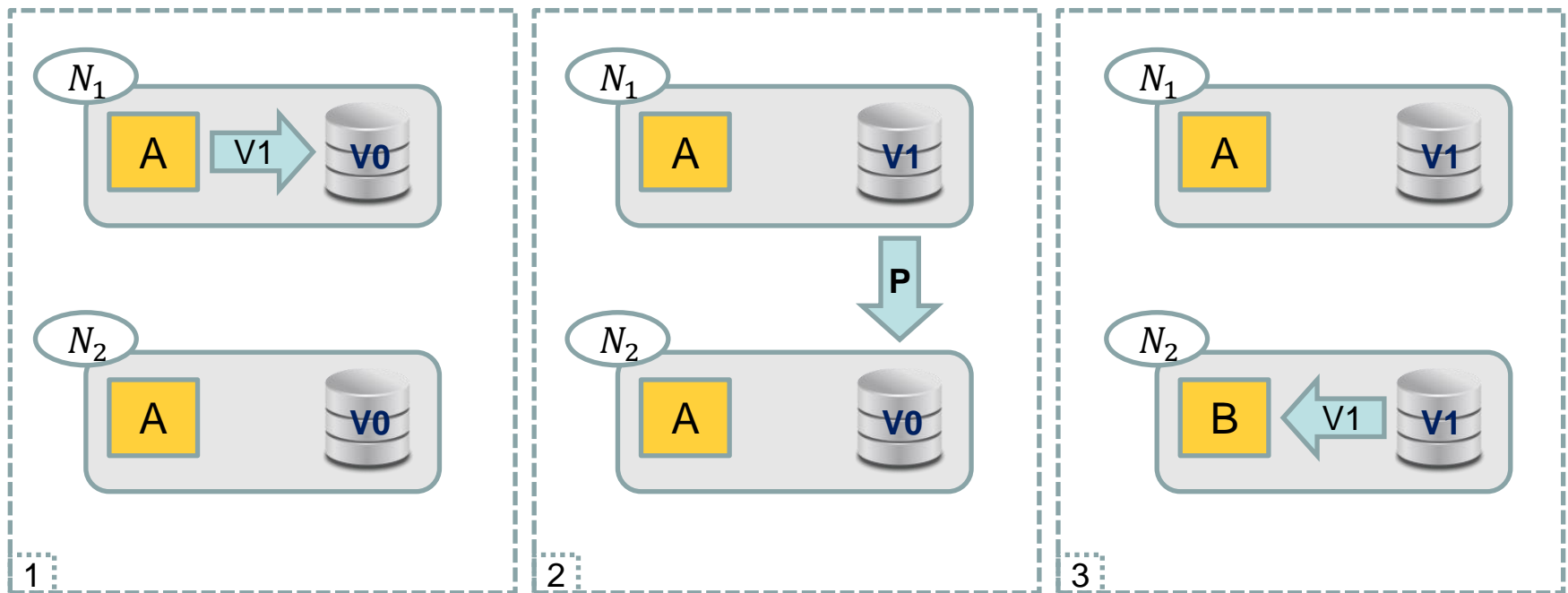
- nijedan skup otkaza, osim potpunog otkazivanja, ne sme da proizvede neispravan odziv sistema baze podataka
 - sistem mora da prihvata delimične otkaze i nastavlja ispravan rad

- **razdvojenost**

- stanje komunikacione mreže kod koje su delovi sistema podeljeni u dve ili više particija
 - » između kojih ne postoji komunikacija
 - rešenje:
 - » replikacija
 - » odloženo (asinhrono) pisanje u trenutno nedostupne delove baze podataka

NoSQL baze podataka

- CAP teorema - dokaz



NoSQL baze podataka

- **CAP teorema – dokaz**
 - šta se događa ako poruka **P** ne stigne na **odredište?**
 - dolazi do razdvojenosti delova sistema
 - tri moguća događaja
 - **transakcija A se poništava**
 - sistem **ne prihvata razdvojenost** svojih delova
 - **transakcija A se uspešno nastavlja**
 - sistem **nije konzistentan**
 - **transakcija A čeka na uspešno slanje poruke P**
 - sistem **nije raspoloživ**
 - » nepoznato vreme odaziva
 - u svakom od događaja **jedan od uslova nije ispunjen**

NoSQL baze podataka

- **CAP teorema – kompromisi**
 - odbacivanje prihvatanja razdvojenosti
 - obezbeđuju se konzistentnost i raspoloživost
 - alternative
 - kompletan sistem na jednom računaru
 - » rad bez razdvojenosti
 - višestruko umrežavanje
 - » redundantne veze
 - veoma skupe
 - ređe se pravi ovakav kompromis

NoSQL baze podataka

- **CAP teorema – kompromisi**
 - **odbacivanje raspoloživosti**
 - obezbeđuju se konzistentnost i tolerancija razdvojenosti
 - u slučaju razdvojenosti ne garantuje se vreme odziva
 - problem se redukuje uspostavljanjem što niže sprege između čvorova
 - koristi se ređe
 - sistem koji nije raspoloživ nije ni upotrebljiv

NoSQL baze podataka

- **CAP teorema – kompromisi**
 - **odbacivanje konzistentnosti**
 - obezbeđuju se raspoloživost i tolerancija razdvojenosti
 - ne garantuje čitanje poslednje verzije podataka u slučaju razdvojenosti
 - u suprotnosti sa ACID skupom osobina
 - definiše se BASE skup osobina
 - » eng. ***B**asically **A**vailable, **S**oft state, **E**ventually consistent*

NoSQL baze podataka

- **BASE osobine**

- **suštinski raspoloživ**

- većina podataka je dostupna veći deo vremena

- **nekonzistentno stanje**

- baza podataka ne mora biti konzistentna u svakom trenutku

- **konvergentna konzistencija**

- dodatkom novog čvora u sistem sadržaj se može replicirati na njega
 - ne postoji garancija da će u svakom trenutku svi čvorovi sadržati identične kopije podataka
 - teži se vremenskoj tački u kojoj će svi čvorovi sadržati konzistentne podatke

NoSQL baze podataka

- ***MapReduce* algoritam**
 - omogućava upravljanje velikim količinama podataka
 - posredstvom distribuirane mreže čvorova
 - deli veliki zadatak na manje delove
 - pogodne za paralelno procesiranje
 - *MapReduce* okruženje
 - upravlja izvršenjem MR algoritma
 - obezbeđuje spregu sa korisnikom

NoSQL baze podataka

- **MapReduce algoritam**
 - **map korak**
 - **glavni čvor**
 - preuzima ulazne podatke
 - deli ulazne podatke na manje delove
 - distribuira manje delove u radne čvorove
 - **radni čvorovi**
 - procesiraju dodeljene zadatke
 - vraćaju odgovor glavnom čvoru
 - **reduce korak**
 - **glavni čvor**
 - preuzima rezultate radnih čvorova
 - kombinuje rezultate sa ciljem dobijanja traženog rezultata

NoSQL baze podataka

- **NoSql sistemi**

- nastali usled potreba za
 - skladištenjem velike količine podataka
 - pristupom od strane velikog broja korisnika
- realizacija baza podataka se oslanja na distribuiranu arhitekturu
 - CAP teorema
 - sa BASE skupom osobina
 - » uvodi pojam konvergentne konzistentnosti kao kompromisno rešenje
 - *MapReduce* algoritam
 - efikasna obrada podataka

NoSQL baze podataka

- **NoSql sistemi – osobine i prednosti**
 - ne poseduju formalno specificiranu šemu baze podataka
 - rukovanje fleksibilnim strukturama podataka
 - nestrukturirani i polustrukturirani podaci
 - ne oslanjaju se na relacioni model podataka
 - ne podržavaju operacije spajanja
 - unapređene performanse u radu sa velikom količinom podataka
 - horizontalna skalabilnost i elastičnost
 - povećanje kapaciteta i performansi u radnom režimu sistema
 - dodavanjem čvorova u distribuiranu mrežu

NoSQL baze podataka

- **NoSql sistemi – osobine i prednosti**
 - arhitektura lokalizovanih resursa
 - *eng. shared nothing architecture*
 - svaki server se oslanja na lokalnu masovnu memoriju
 - particionisanje baze podataka
 - ukupan broj zapisa je distribuiran u particije
 - dodeljene različitim serverima
 - particije se repliciraju
 - asinhrona replikacija
 - podaci nisu replicirani onog momenta kada su zapisani

NoSQL baze podataka

- **NoSql sistemi – osobine i prednosti**
 - izbegavanje objektno-relacionog mapiranja
 - podaci se čuvaju u posebnim strukturama podataka
 - obično jednostavnim
 - bliskim objektno orijentisanim programskim jezicima
 - često predstavlja sistem za upravljanje datotekama
 - tolerancija na otkaze
 - daje prednost performansama
 - na štetu konzistentnosti
 - pristup putem API-ja ili nekog drugog interfejsa
 - ne oslanja se na SQL
 - dominantno otvorenog koda

NoSQL baze podataka

- **NoSql sistemi – nedostaci**
 - nepostojanje standarda
 - nepostojanje standardnog upitnog jezika
 - UnQL (*eng. Unstructured Query Language*)
 - predlog upitnog jezika
 - zasnovan na JSON-u
 - dominantno otvorenog koda
 - nedostatak dokumentacije i korisničke podrške

NoSQL baze podataka

- **Podela NoSQL sistema**
 - skladišta podataka tipa ključ-vrednost
 - kolonski orijentisano skladište podataka
 - baze podataka orijentisane ka dokumentima
 - baze podataka orijentisane ka grafovima

NoSQL baze podataka

- **Skladišta podataka tipa ključ-vrednost**
 - *eng. key – value stores*
 - svaki zapis se sastoji od skupa parova ključ – vrednost
 - proizvoljan broj parova
 - svojstva sa nedostajućim vrednostima se izostavljaju
 - **ključ** predstavlja naziv svojstva
 - jedinstveni identifikator koji ukazuje na vrednost
 - definiše se prilikom formiranja zapisa
 - **vrednost** sadrži podatak ili pokazivač na lokaciju koja sadrži podatak
 - u opštem slučaju vrednosti predstavljaju nizove karaktera
 - zapisi se čuvaju u okviru tabela

NoSQL baze podataka

- **Skladišta podataka tipa ključ-vrednost**
 - zasnovane na principima struktura podataka u višim programskim jezicima
 - kolekcije
 - rečnici
 - rasute tabele
 - mogu da služe kao mehanizmi za keširanje

NoSQL baze podataka

- Skladišta podataka tipa ključ-vrednost

Tabela: Kupac

Zapis

ID: 1
Ime: Petar
Prezime: Petrovic
Adresa: Bulevar Oslobođenja 12
Grad: Novi Sad
Drzava: Srbija
ZIP: 21000
Poslednja narudzba: 252

Zapis

ID: 2
Ime: Marko
Prezime: Markovic
Adresa: Gavrila Principa 42
Grad: Banja Luka
ZIP: 21000
Poslednja narudzba: 262

Tabela: Narudzba

Zapis

ID: 252
Iznos: 30000 Din
Stavka 1: 56432
Stavka 2: 98726

Zapis

ID: 262
Iznos: 42000 Din
Stavka 1: 86413

NoSQL baze podataka

- Skladišta podataka tipa ključ-vrednost**

Ključ-vrednost	Riak	Tokyo Cabinet/Tyrant	Voldemort
Napisana u	<i>Erlang</i>	<i>C</i>	<i>Java</i>
Platforme	<i>Linux</i>	<i>Linux</i>	<i>Bilo koja Java</i>
Protokol	<i>HTTP/REST</i>	<i>HTTP/memcached</i>	<i>HTTP/Thrift</i>
Način čuvanja podataka	<i>plug-in BP</i>	<i>RAM ili disk</i>	<i>RAM ili BerkeleyDB</i>
Replikacija	<i>asinhrona</i>	<i>asinhrona</i>	<i>asinhrona</i>
Particionisanje	<i>da</i>	<i>ne</i>	<i>da</i>
Indeksi	<i>da</i>	<i>ne</i>	<i>ne</i>
MapReduce	<i>da</i>	<i>ne</i>	<i>ne</i>
Transakcije	<i>ne</i>	<i>lokalne</i>	<i>ne</i>

NoSQL baze podataka

- **Kolonski orijentisano skladište podataka**
 - *eng. wide column stores*
 - ne zhteva kompletnu šemu baze podataka
 - samo unapred definisane familije kolona
 - skup kolona koji predstavlja logičku celinu
 - kolone unutar familije ne moraju biti unapred definisane
 - visoke performanse prilikom vršenja upita
 - pogodne za OLAP
 - sporiji upis podataka
 - nisu pogodne za OLTP sisteme

NoSQL baze podataka

- **Kolonski orijentisano skladište podataka**
 - trodimenzionalna struktura
 - identifikator reda
 - familija kolona i identifikator kolone
 - vremenska oznaka
 - revizija podataka
 - red u skladištu je analogan dokumentu
 - grupisani u kolekcije
 - primarni ključ zapisa služi za pristup redu
 - familije kolona igraju ulogu sekundarnog ključa

NoSQL baze podataka

- **Kolonski orijentisano skladište podataka**

Tabela: Kupac

Red

ID: 1
CF/SC: Ime
C: Ime: Petar
C: Prezime: Petrovic
CF/SC: Adresa
C: Adresa: Bul. Osl. 12
C: Grad: Novi Sad
C: Drzava: Srbija
C: ZIP: 21000
CF/SC: Narudzbe
C: Poslednja narudzba: 252

Red

ID: 2
CF/SC: Ime
C: Ime: Marko
C: Prezime: Markovic
CF/SC: Adresa
C: Adresa: Gavrila Principa 42
C: Grad: Banja Luka
C: ZIP: 21000
CF/SC: Narudzbe
C: Poslednja narudzba: 262

Tabela: Narudzba

Red

ID: 252
CF/SC: Cena
C: Iznos: 30000 Din
CF/SC: Stavke
C: Stavka 1: 56432
C: Stavka 2: 98726

Red

ID: 262
CF/SC: Cena
C: Iznos: 42000 Din
CF/SC: Stavke
C: Stavka 1: 86413

NoSQL baze podataka

- Kolonski orijentisano skladište podataka**

Ključ-vrednost	Cassandra	Hbase	SimpleDB
Napisana u	<i>Java</i>	<i>Java</i>	<i>Erlang</i>
Platforme	<i>Linux, Windows</i>	<i>Cross - platform</i>	<i>EC2, POSIX</i>
Protokol	<i>TCP/Thrift</i>	<i>HTTP/REST ili TCP/Thrift</i>	<i>HTTP/REST</i>
Način čuvanja podataka	<i>disk</i>	<i>Hadoop File System</i>	<i>S3</i>
Replikacija	<i>asinhrona</i>	<i>asinhrona</i>	<i>asinhrona</i>
Particionisanje	<i>da</i>	<i>da</i>	<i>da</i>
Indeksi	<i>da</i>	<i>ne</i>	<i>da</i>
MapReduce	<i>da</i>	<i>da</i>	<i>ne</i>
Transakcije	<i>lokalne</i>	<i>lokalne</i>	<i>ne</i>

NoSQL baze podataka

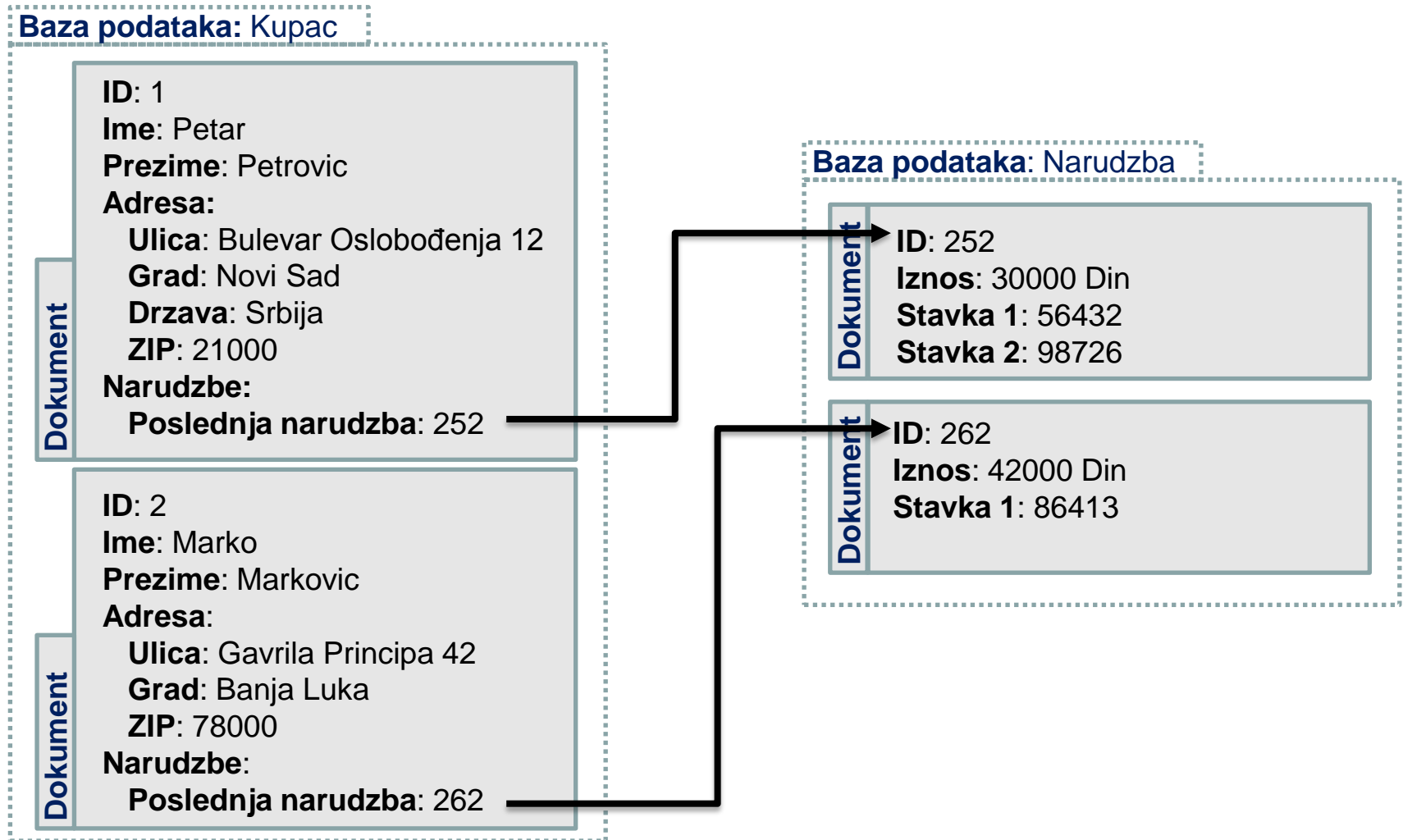
- **Baze podataka orijentisane ka dokumentima**
 - *eng. document-oriented databases*
 - dokument
 - ekvivalentan redu u relacionim bazama podataka
 - u bazi podataka postoji niz samostalnih dokumenata
 - rezultat upita su podaci sadržani u jednom dokumentu
 - ne u povezanim tabelama
 - ne postoji potreba za šemom baze podataka
 - dokumenti u opštem slučaju ne moraju da imaju istu strukturu
 - dokumenti mogu da budu **međusobno povezani** putem URI-ja

NoSQL baze podataka

- **Baze podataka orijentisane ka dokumentima**
 - osobine
 - objekti mogu biti snimljeni kao dokumenti
 - serijalizacija objekta u dokument
 - jednostavna realokacija dokumenata
 - dokumenti mogu biti složeni
 - dokumenti su nezavisni
 - zapisi o jednom entitetu se nalaze u jednom dokumentu
 - jednostavni formati dokumenata
 - JSON ili XML
 - nepostojanje fiksne šeme baze podataka
 - ugrađeno praćenje verzije dokumenata

NoSQL baze podataka

- Baze podataka orijentisane ka dokumentima



NoSQL baze podataka

- Baze podataka orijentisane ka dokumentima

Ključ-vrednost	CouchDB	MongoDB	RavenDB
Napisana u	<i>Erlang</i>	<i>C++</i>	<i>C#</i>
Platforme	<i>Linux</i>	<i>Linux, Mac, Windows</i>	<i>Windows</i>
Protokol	<i>HTTP/REST</i>	<i>sopstveni TCP/IP</i>	<i>HTTP/REST</i>
Način čuvanja podataka	<i>disk</i>	<i>disk</i>	<i>memorija i disk</i>
Replikacija	<i>Peer-based</i>	<i>Master-Slave</i>	<i>plug-in</i>
Particionisanje	<i>da</i>	<i>da</i>	<i>da</i>
Indeksi	<i>da</i>	<i>da</i>	<i>da</i>
MapReduce	<i>da</i>	<i>da</i>	<i>da</i>
Transakcije	<i>ne</i>	<i>ne</i>	<i>da</i>

NoSQL baze podataka

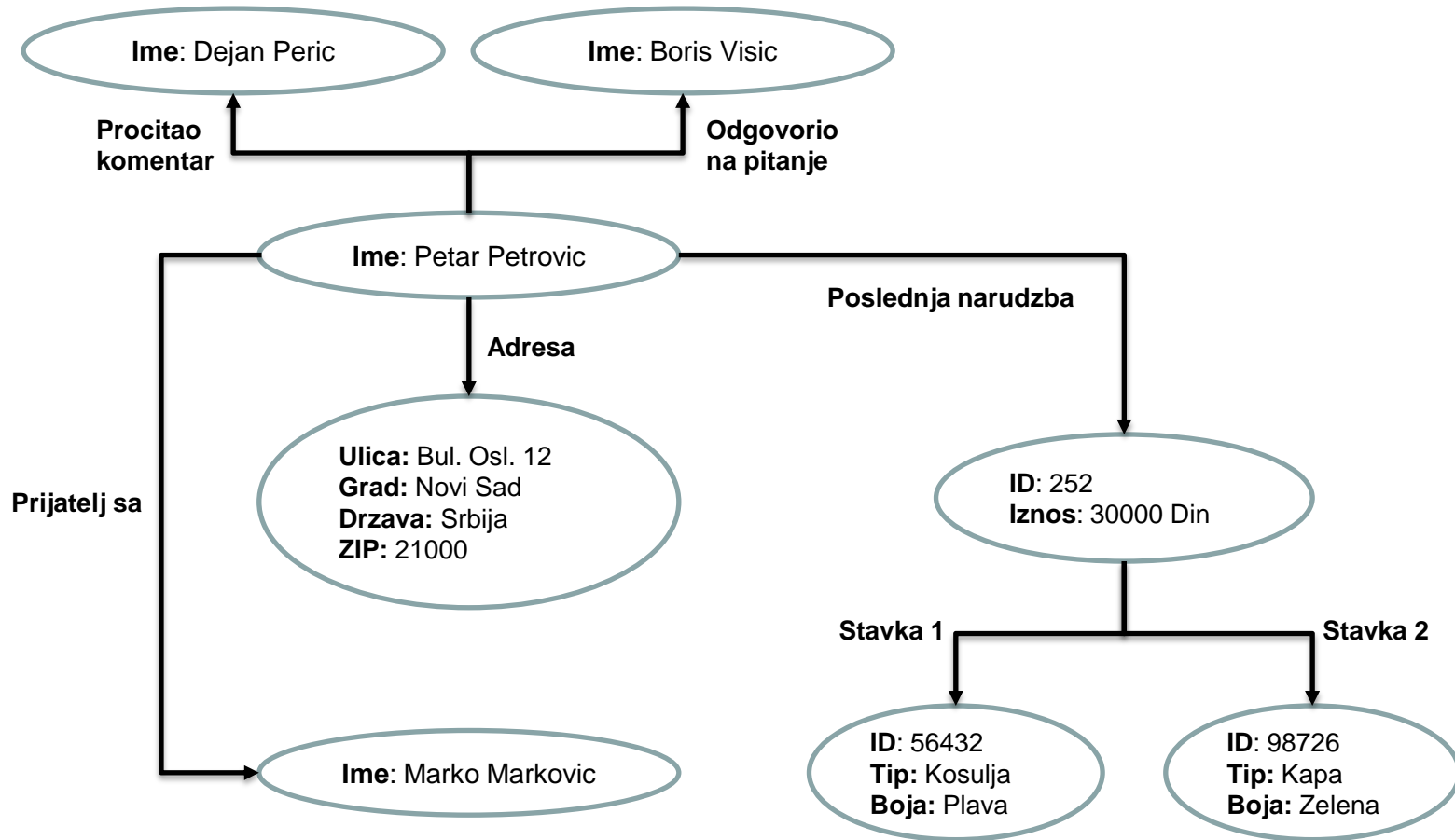
- **Baze podataka orijentisane ka grafovima**
 - *eng. graph databases*
 - zasnovane na teoriji grafova
 - odnosi između kolekcija objekata
 - grafovi se sastoje od
 - **čvorova**
 - koji predstavljaju entitete
 - **grana**
 - koje povezuju čvorove
 - **svojstava**
 - predstavljaju attribute
 - reprezentuju i čuvaju informacije

NoSQL baze podataka

- **Baze podataka orijentisane ka grafovima**
 - nastale sa popularizacijom društvenih mreža
 - veliki broj korisnika koji poseduju veze sa drugim korisnicima, statusima, komentarima itd.
 - pogodne za podatke kojima rukuje semantički web

NoSQL baze podataka

- Baze podataka orijentisane ka grafovima



NoSQL baze podataka

- Baze podataka orijentisane ka grafovima

Ključ-vrednost	Neo4J	InfoGrid	Bigdata
Napisana u	<i>Java</i>	<i>Java</i>	<i>Java</i>
Platforme	<i>bilo koja Java</i>	<i>bilo koja Java</i>	<i>bilo koja Java</i>
Protokol	<i>HTTP</i>	<i>HTTP</i>	<i>HTTP</i>
Način čuvanja podataka	<i>disk</i>	<i>disk, memorija i nekoliko BP</i>	<i>disk</i>
Replikacija	<i>ne</i>	<i>ne</i>	<i>da</i>
Particionisanje	<i>ne</i>	<i>ne</i>	<i>da</i>
Indeksi	<i>ne</i>	<i>ne</i>	<i>da</i>

NoSQL baze podataka

- Uporedne karakteristike i performanse NoSQL sistema i RSUBP-a

	performanse	skalabilnost	fleksibilnost	kompleksnost	big data
key-value	<i>visoke</i>	<i>visoka</i>	<i>visoka</i>	<i>nema</i>	<i>da</i>
wide columns	<i>visoke</i>	<i>visoka</i>	<i>srednja</i>	<i>mala</i>	<i>da</i>
document	<i>visoke</i>	<i>promenljiva</i>	<i>visoka</i>	<i>mala</i>	<i>ne</i>
graph	<i>promenljive</i>	<i>promenljiva</i>	<i>visoka</i>	<i>visoka</i>	<i>ne</i>
RSUBP	<i>promenljive</i>	<i>promenljiva</i>	<i>visoka</i>	<i>srednja</i>	<i>ne</i>

Sadržaj

- NoSQL baze podataka
- Ugrađene baze podataka
- Temporalne baze podataka

Ugrađene baze podataka

- **Motivacija**

- razvoj mobilnih i specijalizovanih uređaja
 - ograničene mogućnosti hardvera
 - specifične softverske platforme
 - zahtev za organizacijom podataka
 - količina podataka ima trend stalnog rasta
- tradicionalni sistemi baza podataka
 - nisu pogodni za upotrebu ovim uređajima
 - nedovoljno jak hardver
 - nisu podržani od strane softverske platforme

Ugrađene baze podataka

- **Ugrađene baze podataka**
 - *eng. embedded database*
 - softverska biblioteka
 - povezana sa klijentskom aplikacijom
 - koriste isti adresni prostor
 - aplikacija postaje jedinstvena programska celina
 - rad sa malim brojem korisnika

Ugrađene baze podataka

- **Osnovne karakteristike i zahtevi**
 - **minimizacija memorijskih zahteva**
 - sistemi imaju skromne memorijske resurse
 - dva aspekta
 - memorijski otisak
 - » *eng. memory footprint*
 - » memorija koju baza zauzima bez podataka
 - prekoračenje podacima
 - » *eng. data overhead*
 - » nesvrshodna potrošnja resursa ili vremena potrebnog za pribavljanje traženog podatka

Ugrađene baze podataka

- **Osnovne karakteristike i zahtevi**
 - **redukovanje alokacije resursa**
 - ugrađena BP mora odgovoriti na ograničenja postavljena od strane tehnologije ugrađenog sistema u kojem egzistira
 - obezbediti integritet i kontinuiran rad
 - prilagoditi resurse trenutnim ograničenjima
 - » predefinisani limiti
 - » trenutno dostupni resursi
 - ručno konfigurisanje upravljanja resursima je neprihvatljivo

Ugrađene baze podataka

- **Osnovne karakteristike i zahtevi**
 - **brzina izvršavanja i predvidivost performansi**
 - varijacije u frekvenciji pristupa i dostupnosti resura
 - ugrađena BP mora biti u mogućnosti da se prilagodi svakoj situaciji
 - sprvode se temeljni test slučajevi
 - ugrađuju se mehanizmi za brz oporavak
 - brzina izvršavanja je od ključnog značaja
 - veliki broj ugrađenih sistema obrađuje podatke u realnom vremenu
 - u cilju postizanja što boljih performansi, ugrađena BP tipično mora koristiti sve raspoložive resurse namenskog hardvera

Ugrađene baze podataka

- **Osnovne karakteristike i zahtevi**
 - **visoka pouzdanost i raspoloživost**
 - ne postoji administrator kao kod tradicionalnih SBP
 - ugrađena BP sama inicira pojedine operacije
 - » indeksiranje, pravljenje rezervne kopije, podešavanje parametara sistema
 - » iniciranje operacija može biti delegirano aplikaciji
 - potrebna je brza reakcija na greške
 - procedura oporavka podataka mora biti vrlo brzo pokrenuta

Ugrađene baze podataka

- **Osnovne karakteristike i zahtevi**
 - **interoperabilnost, prenosivost i podrška različitih operativnih sistema**
 - ugrađeni sistemi poseduju namenske operativne sisteme
 - ugrađena BP mora da podržava takav operativni sistem
 - interoperabilnost sa drugim sistemima BP
 - prenosivost na druge hardverske platforme

Ugrađene baze podataka

- **Osnovne karakteristike i zahtevi**
 - **upotreba fleš memorije**
 - primarni medijum za skladištenje podataka u mobilnim uređajima i uređajima specijalizovane namene
 - trajna memorija, razumnog kapaciteta po prihvatljivoj ceni
 - nema mehaničkih delova
 - » koji prouzrokuju kašnjenja
 - manja potrošnja energije od hard diskova

Ugrađene baze podataka

- **Kriterijumi izbora odgovarajućeg sistema ugrađene BP**
 - izbor platforme
 - podržani operativni sistem
 - izvorni kôd baze podataka
 - zauzeće resursa
 - ocena performansi
 - konkurentnost i skalabilnost
 - zahtevani servisi

Ugrađene baze podataka

- **Tipovi ugrađenih baza podataka**
 - BP integrisane sa aplikacijom na klasičnim računarskim platformama
 - BP integrisane u mobilne uređaje i uređaje specijalizovane namene

Ugrađene baze podataka

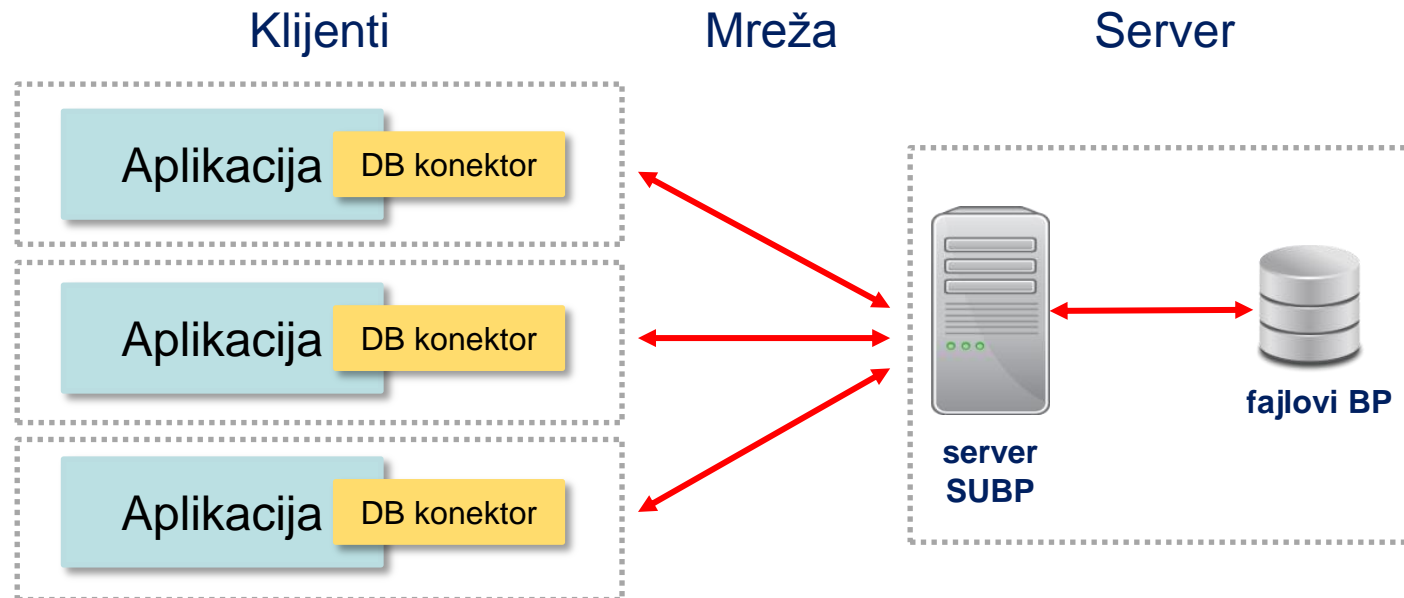
- **BP integrisane sa aplikacijom na klasičnim računarskim platformama**
 - tradicionalni sistemi
 - K/S arhitekture
 - aplikacija preko servera komunicira sa BP
 - BP integrisane sa aplikacijom
 - BP je ugrađena u aplikaciju
 - kao softverska komponenta
 - visoke performanse
 - smanjena kompleksnost komunikacije
 - mali broj korisnika BP

Ugrađene baze podataka

- **BP integrisane sa aplikacijom na klasičnim računarskim platformama**
 - implementacija BP integrisane sa aplikacijom
 - referenciranjem softverske biblioteke
 - koja sadrži implementaciju baze podataka
 - jednostavna ponovna iskoristivost implementacije BP
 - proširenje izvornog koda aplikacije
 - kodom koji implementira BP
 - eliminiše potrebu za postojanjem eksterne softverske biblioteke
 - » pojednostavljena distribucija i instalacija

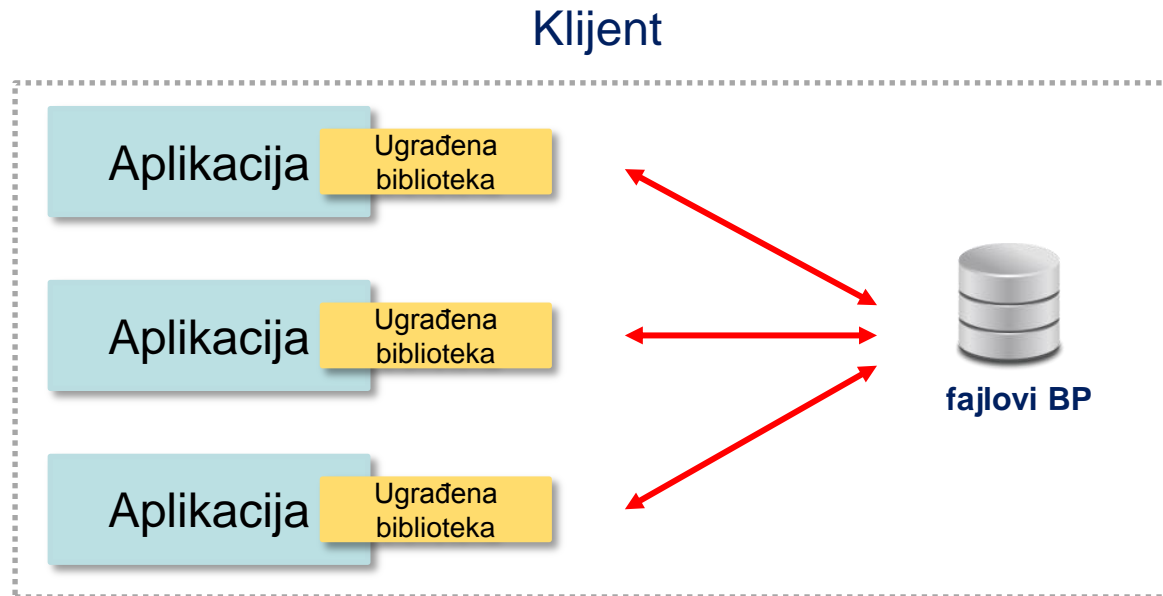
Ugrađene baze podataka

- Tradicionalna arhitektura sistema BP



Ugrađene baze podataka

- Arhitektura sistema BP integrisane sa aplikacijom



Ugrađene baze podataka

- **BP integrisane u mobilne uređaje i uređaje specijalizovane namene**
 - uređaji striktnih hardverskih ograničenja
 - aplikacije namenjene za rešavanje specifičnih problema
 - potreba za ugrađenim bazama podataka
 - prednosti
 - redukovanje troškova razvoja
 - poboljšanje kvaliteta dizajna ugrađenih sistema
 - lakše održavanje i povećana pouzdanost

Ugrađene baze podataka

- **BP integrisane u mobilne uređaje i uređaje specijalizovane namene**
 - koriste se u
 - mobilnom računarstvu
 - inteligentnim uređajima i ugrađenim sistemima
 - smart karticama

Ugrađene baze podataka

- **Pregled postojećih ugrađenih BP**
 - *Berkley DB*
 - najpopularnija NoSQL ugrađena BP
 - softverska biblioteka
 - visoke performanse sa podacima tipa ključ-vrednost
 - napisana u C-u
 - poseduje API-je za većinu modernih programskih jezika
 - podržana većina modernih OS-a
 - visoka konkurentnost i skalabilnost
 - hiljade simultanih upravljačkih niti
 - veličina BP do 256 terabajta
 - memorijski otisak 700 KB – 1.6 MB

Ugrađene baze podataka

- **Pregled postojećih ugrađenih BP**
 - *Hamster DB*
 - mala NoSql ugrađena BP tipa ključ-vrednost
 - napisana u C/C++-u
 - poseduje API-je za Javu, Python, .Net i Erlang
 - podržana većina modernih OS-a
 - » *Google Android i Apple iOS*
 - visoka konkurentnost i skalabilnost
 - memorijski otisak 600 KB

Ugrađene baze podataka

- **Pregled postojećih ugrađenih BP**
 - *Raptor DB*
 - vrlo mala NoSql ugrađena BP
 - u formi perzistentnog rečnika podataka
 - realizovana kao softverska biblioteka
 - dizajnirana za podatke u JSON formatu
 - prihvata i sve ostale vrste podataka
 - dizajnirana samo da dodaje podatke
 - poseduje istorijske/duplicirane podatke
 - visoka konkurentnost i skalabilnost
 - memorijski otisak **40 KB**

Ugrađene baze podataka

- **Pregled postojećih ugrađenih BP**
 - *SQLite*
 - softverska biblioteka
 - zasnovana na relacionom modelu podataka
 - podržava transakcioni režim i očuvanje ACID svojstava
 - nema potrebu za podešavanjem ili administriranjem
 - smeštena u jedinstvenu datoteku
 - moguć prenos na bilo koju platformu
 - napisana u C-u
 - poseduje API-je za C i C++
 - podržana većina modernih OS-a
 - visoka skalabilnost
 - veličina BP do reda veličine terabajta
 - veliki objekti reda veličine gigabajta
 - memorijski otisak 200 KB – 350 KB

Ugrađene baze podataka

- **Pregled postojećih ugrađenih BP**
 - *PicoDBMS*
 - baza podataka za smart kartice
 - podržava moćan podskup SQL standarda
 - vrši autentifikaciju korisnika
 - dozvoljava pristup isključivo dozvoljenom sadržaju
 - koristi EEPROM
 - kao primarnu memoriju

Sadržaj

- NoSQL baze podataka
- Ugrađene baze podataka
- Temporalne baze podataka

Temporalne baze podataka

- **Temporalne baze podataka**
 - sve baze podataka koje poseduju vreme kao aspekt u organizovanju podataka
 - uvode temporalne koncepte
 - na nivou baze podataka
 - aplikacije koriste ove temporalne koncepte

Temporalne baze podataka

- **Reprezentacija vremena**
 - vreme je uređeni niz trenutaka u granularnosti definisanoj od strane aplikacije
 - **kronon**
 - minimalna granularnost za neku aplikaciju
 - svi događaji u okviru kronona se posmatraju kao istovremeni događaji
 - u realnom sistemu to ne mora da bude slučaj

Temporalne baze podataka

- **Kalendar**

- organizuje vreme u različite vremenske jedinice
 - lakše za rukovanje
 - npr. minut, sat, dan, mesec, itd.
- omogućava merenje vremena od neke početne tačke
 - razlikuje se u zavisnosti od kulture, npr:
 - Gregorijanski kalendar
 - Kineski kalendar
 - Islamski kalendar

Temporalne baze podataka

- **Vremenski tipovi u SQL-u**
 - DATE
 - godina, mesec i dan
 - YYYY-MM-DD
 - TIME
 - sat, minut i sekund
 - HH:MM:SS
 - TIMESTAMP
 - kombinacija TIME i DATE
 - YYYY-MM-DD HH:MM:SSS

Temporalne baze podataka

- **Vremenski tipovi u SQL-u**
 - INTERVAL
 - relativni vremenski period
 - 10 dana
 - 250 minuta
 - PERIOD
 - fiksirani vremenski period
 - fiksirana početno vreme
 - 10 dana od 1. januara 2013. do 10. januara 2013. godine

Temporalne baze podataka

- **Vrste događaja u temporalnim BP**
 - **jedinični događaji (činjenice)**
 - obuhvataju jedinstveni vremenski trenutak
 - u definisanoj granularnosti
 - **događaji (činjenice) koji traju**
 - obuhvataju određen vremenski period
 - definisan početnom i krajnjom tačkom u vremenu
 - » obuhvata i sve trenutke između
 - » u definisanoj granularnosti

Temporalne baze podataka

- **Interpretacija vremena u BP**
 - kako interpretiramo vreme povezano sa podacima u BP
 - podaci predstavljaju događaje ili činjenice
 - **validno vreme**
 - vreme kada se događaj zbio
 - vreme kada je činjenica bila tačna
 - u **realnom svetu**
 - **transakciono vreme**
 - vreme kada je podatak upisan u bazu podataka
 - vreme kada je informacija **validna u sistemu**

Temporalne baze podataka

- **Interpretacija vremena u BP**
 - validno vreme i transakciono vreme nazivaju se **vremenskim dimenzijama**
 - moguće i ostale interpretacije vremena
 - **korisnički definisano vreme**
 - korisnik
 - daje semantiku interpretaciji
 - programira aplikaciju da je podrži

Temporalne baze podataka

- **Interpretacija vremena u BP**
 - pristupi implementiranju temporalnih baza podataka
 - **verzionisanje torki**
 - kod relacionih sistema
 - dodaje se vreme svakoj torci
 - prilikom promene torke kopiraju se i atributi koji nisu promenjeni
 - **verzionisanje atributa**
 - kod sistema koji podržavaju složene objekte
 - » objektno-orijentisane BP
 - » objektno-relacione BP

Temporalne baze podataka

- **Interpretacija vremena u BP**
 - podela baza podataka u odnosu na vremenske dimenzije
 - **baze podataka sa validnim vremenom**
 - sadrže samo validno vreme
 - **baze podataka sa transakcionim vremenom**
 - sadrže samo transakciono vreme
 - **bitemporalne baze podataka**
 - sadrže i validno i transakciono vreme

Temporalne baze podataka

- **Relaciona baza podataka – primer**
 - šeme relacije Radnik i Departman
 - torke predstavljaju **trenutno stanje** entiteta u realnom svetu

Temporalne baze podataka

- **Relaciona baza podataka – primer**

- Radnik

Ime	<u>JMBG</u>	Plt	DepID	RukJMBG
-----	-------------	-----	-------	---------

- Departman

Naziv	<u>DepID</u>	RukJMBG
-------	--------------	---------

Temporalne baze podataka

- **BP sa validnim vremenom**
 - zahtev za praćenjem istorije promena nad nekim entitetom
 - uvode se početno i krajnje vreme validnosti entiteta u realnom svetu
 - obeležja s nazivima: Vpv i Vkv
 - trenutno stanje entiteta u temporalnim BP sa validnim vremenom
 - temporalna konstanta **NOW** se dodeljuje obeležju Vkv
 - označava trenutno vreme
 - » uzimajući u obzir napredovanje vremena

Temporalne baze podataka

- **BP sa validnim vremenom**
 - primarni ključ šeme relacije sa validnim vremenom
 - vreme početka validnosti (Vpv)
 - ostala obeležja koja jedinstveno identifikuju entitet
 - ukoliko su netemporalna obeležja u primarnom ključu podložna promenama
 - umesto njih se uvodi jedno obeležje koje predstavlja surogatni ključ
 - njemu se pridružuje Vpv

Temporalne baze podataka

- **BP sa validnim vremenom - primer**

- Radnik_VV

Ime	<u>JMBG</u>	Plt	DepID	RukJMBG	<u>Vpv</u>	Vkv
-----	-------------	-----	-------	---------	------------	-----

- Departman_VV

Naziv	<u>DepID</u>	RukJMBG	<u>Vpv</u>	Vkv
-------	--------------	---------	------------	-----

Temporalne baze podataka

- **BP sa validnim vremenom**
 - brisanje torke
 - torci koja se briše se upisuje vrednost Vkv obeležja
 - torca se zatvara
 - logičko brisanje
 - dodavanje torke
 - upisivanjem nove torke u relaciju
 - Vkv dobija vrednost NOW

Temporalne baze podataka

- **BP sa validnim vremenom**
 - ažuriranje torci
 - „staroj“ torci se upisuje vrednost Vkv obeležja
 - **zatvorena (istorijska) torca**
 - upisuje se nova torca sa izmenjenim vrednostima
 - Vpv označava vreme izmene entiteta u ralnom svetu
 - Vkv sadrži promenljivu NOW

Temporalne baze podataka

- **BP sa validnim vremenom**
 - tipovi ažuriranja
 - **proaktivno ažuriranje**
 - ažuriranje se obavlja **pre** promene u realnom sistemu
 - početno vreme se postavlja na datum u **budućnosti**
 - **retroaktivno ažuriranje**
 - ažuriranje se obavlja **nakon** promene u realnom sistemu
 - početno vreme se postavlja na datum u **prošlosti**
 - **simultano ažuriranje**
 - ažuriranje se obavlja **paralelno** sa promenom u realnom sistemu
 - početno vreme se postavlja na trenutni datum
 - **ne postoji informacija o promeni stanja baze podataka**

Temporalne baze podataka

- **BP sa validnim vremenom – primer**

Ime	JMBG	Plt	DepID	RukJMBG	Vpv	Vkv
Petar	0901251	25000	5	9851244	2002-06-15	2003-05-31
Petar	0901251	30000	5	9851244	2003-06-01	NOW
Marko	3654211	25000	4	9851244	1999-08-20	2001-01-31
Marko	3654211	30000	5	9851244	2001-02-01	2002-03-31
Marko	3654211	40000	5	9851244	2002-04-01	NOW
Dejan	9851244	28000	4	3241545	2001-05-01	2002-08-10
Ivan	3241545	38000	5	NULL	2003-08-01	NOW

Temporalne baze podataka

- BP sa validnim vremenom – primer

Naziv	DepID	RukJMBG	Vpv	Vkv
E1	4	9851244	2001-09-20	NOW
E2	5	9851244	2001-09-20	2002-03-31
E2	5	3241545	2002-04-01	NOW

Temporalne baze podataka

- **BP sa transakcionim vremenom**
 - zahtev za praćenjem promene stanja sistema BP
 - svakoj torci se pridružuje vremenski otisak
 - za početak transakcije (Vpt)
 - za kraj transakcije (Vkt)
 - uobičajeni tip podataka je **TIMESTAMP**
 - **rollback baze podataka**
 - moguća primena operacija logičkog poništavanja (logički *rollback*, tj. *flashback*)
 - u cilju vraćanja stanja određenog dela baze podataka u stanje željenog vremenskog trenutka

Temporalne baze podataka

- **BP sa transakcionim vremenom**
 - trenutno stanje entiteta u temporalnim BP sa transakcionim vremenom
 - temporalna konstanta **UNTIL CHANGED** (UC) se dodeljuje obeležju Vkt
 - označava trenutno transakciono vreme
 - dok torku ne promeni neka druga transakcija
 - primarni ključ šeme relacije sa validnim vremenom
 - vreme početka transakcije (Vpt)
 - ostala obeležja koja jedinstveno identifikuju entitet

Temporalne baze podataka

- **BP sa validnim vremenom - primer**
 - Radnik_TV

Ime	<u>JMBG</u>	Plt	DepID	RukJMBG	<u>Vpt</u>	Vkt
-----	-------------	-----	-------	---------	------------	-----

- Departman_TV

Naziv	<u>DepID</u>	RukJMBG	<u>Vpt</u>	Vkt
-------	--------------	---------	------------	-----

Temporalne baze podataka

- **Bitemporalne baze podataka**
 - zahtev za praćenjem promene stanja sistema BP kao i promene podataka u realnom svetu
 - svaka šema relacije sadrži **obe vremenske dimenzije**
 - primarni ključ bitemporalne šeme relacije
 - vreme početka transakcije (Vpt)
 - vreme početka validnosti (Vpv)
 - ostala obeležja koja jedinstveno identifikuju entitet
 - trenutno stanje entiteta u bitemporalnim BP
 - Vkv ima vrednost NOW
 - Vkt ima vrednost UC

Temporalne baze podataka

- **BP sa validnim vremenom - primer**
 - Radnik_BT

Ime	<u>JMBG</u>	Plt	DepID	RukJMBG	<u>Vpv</u>	Vkv	<u>Vpt</u>	Vkt
-----	-------------	-----	-------	---------	------------	-----	------------	-----

- Departman_BT

Naziv	<u>DepID</u>	RukJMBG	<u>Vpv</u>	Vkv	<u>Vpt</u>	Vkt
-------	--------------	---------	------------	-----	------------	-----

Temporalne baze podataka

- **Bitemporalne baze podataka**
 - modifikacija torke
 - nijedno obeležje se fizički **ne menja** osim V_{kv} i V_{kt}
 - za svaku izmenu se dodaje nova torke
 - nova verzija entiteta
 - postupak modifikacije
 - elementi modifikacije
 - trenutna verzija torke v koja se modifikuje
 - » $v[V_{kv}] = NOW$
 - » $v[V_{kt}] = UC$
 - transakcija T koja modifikuje torku
 - » $TS(T)$ vremenski otisak transakcije T
 - VT trenutak u vremenu kada je entitet promenio stanje u relanom sistemu
 - » VT - trenutak neposredno pre VT

Temporalne baze podataka

- **Bitemporalne baze podataka**

- postupak modifikacije

- koraci modifikacije

1. dodati novu toroku v_2 u relaciju

- » v_2 je kopija torke v

- » $v_2[Vkv] = VT$

- » $v_2[Vpt] = TS(T)$

- » $v_2[Vkt] = UC$

2. dodati novu toroku v_3 u relaciju

- » v_3 je kopija torke v

- » $v_3[Vpv] = VT$

- » $v_3[Vkv] = NOW$

- » modifikuju se vrednosti polja koja se menjaju

- » $v_3[Vpt] = TS(T)$

- » $v_3[Vkt] = UC$

Temporalne baze podataka

- **Bitemporalne baze podataka**

- postupak modifikacije

- koraci modifikacije

3. $v_2[Vkt] = TS(T)$

- toraka v_3 predstavlja trenutnu verziju entiteta u bazi podataka

Temporalne baze podataka

- **Bitemporalne baze podataka**

- brisanje torki

- logičko brisanje torke v
- dodaje se nova torka v_2
 - kopija torke v
 - $v_2[Vpt] = TS(T)$
 - $v_2[Vkv] = VT$
 - $v[Vkt] = TS(T)$

- dodavanje torki

- dodaje se nova torka v_n
 - $v_n[Vpt] = TS(T)$
 - $v_n[Vkt] = UC$
 - $v_n[Vpv] = VT$
 - $v_n[Vkv] = NOW$

Temporalne baze podataka

- BP sa validnim vremenom – primer

Ime	JMBG	Plt	DepID	RukJMBG	Vpv	Vkv	Vpt	Vkt
Petar	0901251	25000	5	9851244	2002-06-15	NOW	2002-06-08, 13:05:58	2003-06-04, 08:56:12
Petar	0901251	30000	5	9851244	2002-06-15	2003-05-31	2003-06-04, 08:56:12	UC
Petar	0901251	30000	5	9851244	2003-06-01	NOW	2003-06-04, 08:56:12	UC
Dejan	9851244	28000	4	3241545	2001-05-01	NOW	2001-04-27, 16:22:05	2002-08-12, 10:11:07
Dejan	9851244	28000	4	3241545	2001-05-01	2002-08-10	2002-08-12, 10:11:07	UC
Ivan	3241545	38000	5	NULL	2003-08-01	NOW	2003-07-28, 09:25:37	UC

Temporalne baze podataka

- BP sa validnim vremenom – primer

Ime	JMBG	Plt	DepID	RukJMBG	Vpv	Vkv	Vpt	Vkt
Marko	3654211	25000	4	9851244	1999-08-20	NOW	1999-08-20, 11:18:23	2001-01-07, 14:33:02
Marko	3654211	30000	5	9851244	1999-08-20	2001-01-31	2001-01-07, 14:33:02	UC
Marko	3654211	40000	5	9851244	2001-02-01	NOW	2001-01-07, 14:33:02	2002-03-28, 09:23:57
Marko	3654211	25000	4	9851244	2001-02-01	2002-03-31	2002-03-28, 09:23:57	UC
Marko	3654211	30000	5	9851244	2002-04-01	NOW	2002-03-28, 09:23:57	UC

Temporalne baze podataka

- BP sa validnim vremenom – primer

Naziv	DepID	RukJMBG	Vpv	Vkv	Vpt	Vkt
E1	4	9851244	2001-09-20	NOW	2001-09-20, 13:14:55	UC
E2	5	9851244	2001-09-20	NOW	2001-09-15, 14:52:12	2002-03-28, 09:23:57
E2	5	3241545	2001-09-20	2002-03-31	2002-03-28, 09:23:57	UC
E2	5	3241545	2002-04-01	NOW	2002-03-28, 09:23:57	UC

Temporalne baze podataka

- **Bitemporalne baze podataka**
 - načini implementacije bitemporalnih relacija
 - **jedna relacija**
 - sve torke pripadaju jednoj relaciji
 - **dve relacije**
 - trenutno aktuelne torke pripadaju jednoj relaciji
 - istorijske torke u drugoj

Temporalne baze podataka

- **Interpretacija vremena u BP**
 - verzionisanje atributa
 - jedan složeni objekat se koristi kako objedinio sve vremenske promene
 - **atribut zavisan od vremena**
 - svaki atribut koji se menja u toku vremena
 - vrednosti su mu verzionisane dodavanjem temporalnih atributa
 - » validno vreme, transakciono vreme ili bitemporalni
 - **atribut nezavisan od vremena**
 - svaki atribut koji se ne menja u toku vremena
 - ne sadrže temporalne attribute

Temporalne baze podataka

- **Interpretacija vremena u BP**
 - verzionisanje atributa
 - atributi se menjaju nezavisno jedni od drugih
 - nema potrebe za kopiranjem celog objekta
 - » već samo atributa koji se menjaju
 - poseban atribut za definisanje validnosti celog objekta
 - **atribut koji opisuje životni vek**
 - » označava periode validnosti objekta kao celine
 - » kako u realnom svetu tako i u sistemu baze podataka
 - » logičko brisanje objekta se obavlja zatvaranjem životnog veka
 - » postavljanje vremena u Vkv i Vtv atribut
 - ograničenje
 - » **svaki vremenski period važenja atributa mora biti podskup životnog veka objekta**

Temporalne baze podataka

- Interpretacija vremena u BP

```
class TEMPORAL_PLATA
```

```
{
```

```
    attribute    Date            Vpv;
```

```
    attribute    Date            Vkv;
```

```
    attribute    float           Plata;
```

```
};
```

```
class TEMPORAL_DEPARTMAN
```

```
{
```

```
    attribute    Date            Vpt;
```

```
    attribute    Date            Vkv;
```

```
    attribute    DEPARTMAN_VT    Dep;
```

```
};
```

Temporalne baze podataka

- Interpretacija vremena u BP

```
class TEMPORAL_RUKOVODILAC
```

```
{
```

```
    attribute    Date           Vpv;
```

```
    attribute    Date           Vkv;
```

```
    attribute    EMPLOYEE_VT    Rukovodilac;
```

```
};
```

```
class TEMPORAL_ZIVOTNI_VEK
```

```
{
```

```
    attribute    Date           Vpv;
```

```
    attribute    Date           Vkv;
```

```
};
```

Temporalne baze podataka

- Interpretacija vremena u BP

```
class RADNIK_VT  
( extent RADNICI )
```

```
{
```

```
    attribute    list< TEMPORAL_ZIVOTNI_VEK > Zivotni_vek;
```

```
    attribute    string                        Ime;
```

```
    attribute    string                        JMBG;
```

```
    attribute    list<TEMPORAL_SALARY>        Plt_istorija;
```

```
    attribute    list<TEMPORAL_DEPT>         Dep_istorija;
```

```
    attribute    list <TEMPORAL_SUPERVISOR>  Ruk_istorija;
```

```
};
```

Temporalne baze podataka

- **TSQL jezik**
 - obuhvata proširenja SQL-a
 - za rad nad bazama podataka sa temporalnim proširenjima
 - tradicionalni uslovi selekcije
 - obuhvataju samo trenutna stanja entiteta
 - temporalni uslovi selekcije
 - obuhvataju sva stanja entiteta
 - trenutna i istorijska stanja
 - uključuje se i vremensko obeležje
 - **čist vremenski uslov**
 - obuhvata samo vremenska obeležja

Temporalne baze podataka

- **TSQL jezik**
 - temporalni uslovi selekcije
 - selektuje torke koje su validne
 - u trenutku u vremenu T
 - u vremenskom periodu $[T_1, T_2]$
 - » skup trenutaka u vremenu između T_1 i T_2
 - » uključujući T_1 i T_2
 - Alenova algebra
 - » obuhvata skup operacija nad vremenskim podacima

Temporalne baze podataka

- **TSQL jezik – operacije**

$[T.Vpv, T.Vkv]$ **INCLUDES** $[T_1, T_2]$

$T_1 \geq T.Vpv$ AND $T_2 \leq T.Vkv$

$[T.Vpv, T.Vkv]$ **INCLUDED_IN** $[T_1, T_2]$

$T_1 \leq T.Vpv$ AND $T_2 \geq T.Vkv$

$[T.Vpv, T.Vkv]$ **OVERLAPS** $[T_1, T_2]$

$T_1 \leq T.Vkv$ AND $T_2 \geq T.Vpv$

$[T.Vpv, T.Vkv]$ **BEFORE** $[T_1, T_2]$

$T_1 \geq T.Vkv$

$[T.Vpv, T.Vkv]$ **AFTER** $[T_1, T_2]$

$T_2 \leq T.Vpv$

$[T.Vpv, T.Vkv]$ **MEETS_BEFORE** $[T_1, T_2]$

$T_1 = T.Vkv + 1$

$[T.Vpv, T.Vkv]$ **MEETS_AFTER** $[T_1, T_2]$

$T_2 + 1 = T.Vpv$

Temporalne baze podataka

- **TSQL jezik – operacije**
 - rezultat operacija nad vremenskim intervalima može biti
 - vremenski trenutak
 - vremenski period
 - temporalni element
 - boolean vrednost

Temporalne baze podataka

- **TSQL jezik – operacije**
 - **temporalni element**
 - skup disjunktih vremenskih perioda
 - za svaka dva perioda $[T_1, T_2]$ i $[T_3, T_4]$ važi
 - $[T_1, T_2] \cap [T_3, T_4] = \emptyset$
 - T_3 nije naredni trenutak u vremenu nakon T_2
 - » u datoj granularnosti
 - T_1 nije naredni trenutak u vremenu nakon T_4
 - » u datoj granularnosti

Temporalne baze podataka

- **TSQL jezik – primer**

- čist vremenski uslov

- *Prikazati sve verzije entiteta radnik koje su bile validne u bilo kom trenutku u 2011 godini.*

SELECT *

FROM Radnik T

WHERE [T.Vpv, T.Vkv] **OVERLAPS** [2011-01-01, 2011-12-31]

Temporalne baze podataka

- **TSQL jezik – primer**

- uslov sa atributima i vremenom

- *Prikazati sve verzije entiteta radnik koje su bile validne u bilo kom trenutku u 2011 godini. Radnici moraju da pripadaju departmanu 5.*

```
SELECT *
```

```
FROM Radnik T
```

```
WHERE [T.Vpv, T.Vkv] OVERLAPS [2011-01-01, 2011-12-31]
```

```
and T.Dep = 5
```

Temporalne baze podataka

- **TSQL jezik**
 - omogućava kreiranje temporalnih relacija
 - opcione AS klauzule CREATE TABLE naredbe
 - **AS VALID STATE <GRANULARITY>**
 - relacija sa validnim vremenom, vreme izraženo kroz periode
 - **AS VALID EVENT <GRANULARITY>**
 - relacija sa validnim vremenom, vreme izraženo kroz trenutke u vremenu
 - **AS TRANSACTION**
 - relacija sa transakcionim vremenom, vreme izraženo kroz periode
 - **AS VALID STATE <GRANULARITY> AND TRANSACTION**
 - bitemporalna relacija, vreme izraženo kroz periode
 - **AS VALID EVENT <GRANULARITY> AND TRANSACTION**
 - bitemporalna relacija, vreme izraženo kroz trenutke u vremenu

Temporalne baze podataka

- **Podaci o vremenskim serijama**
 - *eng. Time Series Data*
 - vremenska serija
 - predefinisana sekvenca trenutaka u vremenu
 - specijalan slučaj **validnih vremenskih podataka**
 - trenuci predefinisani u nekom kalendaru
 - koriste se u finansijskim aplikacijama

Temporalne baze podataka

- **Podaci o vremenskim serijama**
 - SUBP-ovi
 - moraju da omoguće upravljanje serijama podataka
 - operacije nad vremenskim podacima
 - definisanje kalendara
 - kreiranje kalendara na osnovu koga će se definisati vremenska serija

Reference

- Tiwari S, „Professional NoSQL“, John Wiley & Sons, Inc., SAD, 2011
- Todorčić B, „Primena specijalizovanih baza podataka u oblasti upravljanja dokumentima“, Master rad, FTN, 2012.
- Elmasri R, Navathe S B, „Fundamentals of Database Systems“, Šesto izdavanje, Addison-Wesley, SAD, 2011
 - poglavlje 26

Pitanja i komentari



Sadržaj

- NoSQL baze podataka
- Ugrađene baze podataka
- Temporalne baze podataka

Kraj prezentacije

Sistemi baza podataka



Alternativni pristupi u izgradnji sistema baza podataka

*NoSQL, ugrađene i temporalne
baze podataka*