UNIVERZITET U NIŠU ELEKTRONSKI FAKULTET



**Interna struktura i organizacija skladišta podataka MongoDB**

Seminarski rad iz predmeta Sistemi za upravljanje bazama podataka

|  |  |
| --- | --- |
| Kandidat: | Mentor: |
| Katarina Filipović, br. ind. 1439 | Doc. dr Aleksandar Stanimirović |

**Sadržaj**

**1. UVOD .............................................................................................................. 2**

**2. MongoDB baza podata....................................................................................3**

2.1. Karakteristike MongoDB baze podaka…….........................................................................3

2.2. Prednosti MongoDB baze podaka……................................................................................4

2.3. Nedostaci MongoDB baze podaka……...............................................................................5

**3. Struktura baze podataka .............................................................................. 6**

3.1 Kolekcije ………………..................................................................................................... 6

3.1.1 Ograničena kolekcija.......................................................................................................... 7

3.1.2 TTL kolekcije……………………….......,…………………………….................................. 8

3.1.3 Sistemske kolekcije................................................................................................................. 8

3.1.4 Normalne kolekcije.................................................................................................................8

3.1.5 Geoprostorne kolekcije............................................................................................................8

3.2 Dokument……………………………………..................................................................... 9

3.2.1 Struktura dokumenta...............................................................................................................10

3.2.2 Ograničenje veličine dokumenta.............................................................................................14

3.2.3 GridFS……………….............................................................................................................14

3.2.3.1 GridFS kolekcije……...........................................................................................................14

3.2.3.2 GridFS indeksi......................................................................................................................17

3.2.3.3 GridFS Sharding...................................................................................................................18

3.2.3.4 Ograničenja GridFS..............................................................................................................18

3.2.3.5 Primena GridFS……............................................................................................................18

**4. Mehanizmi za skladištenje.............................................................................20**

4.1 WiredTiger mehanizam (Default)…………….................................................................... 20

4.1.1 Konkurencija na nivou doku…................................................................................................20

4.1.2 Snimci i kontrolne tačke….......................................................................................................20

4.1.3 Kompresija ………………......................................................................................................21

4.2 In-Memory mehanizam ………...…………….................................................................... 21

**5. Zaključak………………….............................................................................22**

**6. Literatura………………….............................................................................23**

1. **Uvod**

MongoDB baza podataka je NoSQL baza podataka koja ima fleksibilnu šemu. Fleksibilne šeme stvaraju dodatni izazov za dizajnere aplikacija. Kod baza podataka kod kojih je fiksna šema, svi podaci u bazi podataka moraju biti u skladu sa trenutnom verzijom šeme. Međutim, u MongoDB-u, zbirka može uključivati dokumente koji su drugačije strukturirani.

Osnovna struktura MongoDB baze podataka je sama baza podataka u koju se skladište kolekcije, a kolekcija sadrži više dokumenata u kojima se nalaze odredjeni podaci.

Za smeštanje podataka u bazu podataka iskorišćen je BSON dokument. MongoDB skladišti podatke u BSON formatu i interno i preko mreže, ali to ne znači da ne može da se zamisli MongoDB kao JSON bazu podataka. Sve što možete da se predstavi u JSON-u može biti izvorno uskladišteno u MongoDB-u i isto tako lako preuzeto u JSON-u. Maksimana veličina BSON dokumenta je ograničena . Maksimalna veličina dokumenta pomaže da se osigura da jedan dokument ne može da koristi preveliku količinu RAM-a ili, tokom prenosa, preveliku količinu propusnog opsega. Ukoliko je potrebno smestitit dokument koji premašuje maksimalnu veličinu koja može da se smesti, potrebno je da se iskoristi GridFS.

GridFS mehanizam deli velike dokumete na delove i zatim smešta te delove. MongoDB ima odličan način za rukovanje datotekama koje su iznad navedenog ograničenja dokumenta. GridFS, iako nije savršen i dalje je odličan za slučajeve upotrebe kada je potrebno da se datoteke i metapodaci održe doslednim. Tipičan slučaj upotrebe GridFS-a može biti kada se koriste geografski distribuirani skupovi replika. MongoDB može da automatski distribuira datoteke i njihove metapodatke na brojne mongod instance i objekte.

Osnovne karakteristike i ono što je karakteristično za MongoDB bazu podataka biće opisano u drugom poglavlju. Osim osnovnih karakteristika u drugom poglavlju biće opisane prednosti i mane ove baze podataka. U trećem poglavlju biće prikazana struktura MongoDB baze podataka. Biće opisana kolekcija kao i dokument koji predstavljaju delove baze podataka. Detaljnije će biti opisana struktura dokumenta. Takodje biće opisan BSON tip dokumenta koji se koristi za smeštanje podataka u bazu podataka. U trećem poglavlju se pominje i GridFS sistem koji se koristi za smeštanje većih dokumenata. Biće objašnjeno na koji način se smeštaju dokumenti pomoću ovog sistemi, kolekcije koje se kreiraju i u koje se smeštaju delovi kao i indeksi. Takodje biće pomenuto i kad se primenjuje ovaj sistem i njegova oragničenja.

1. **MongoDB baza podataka**

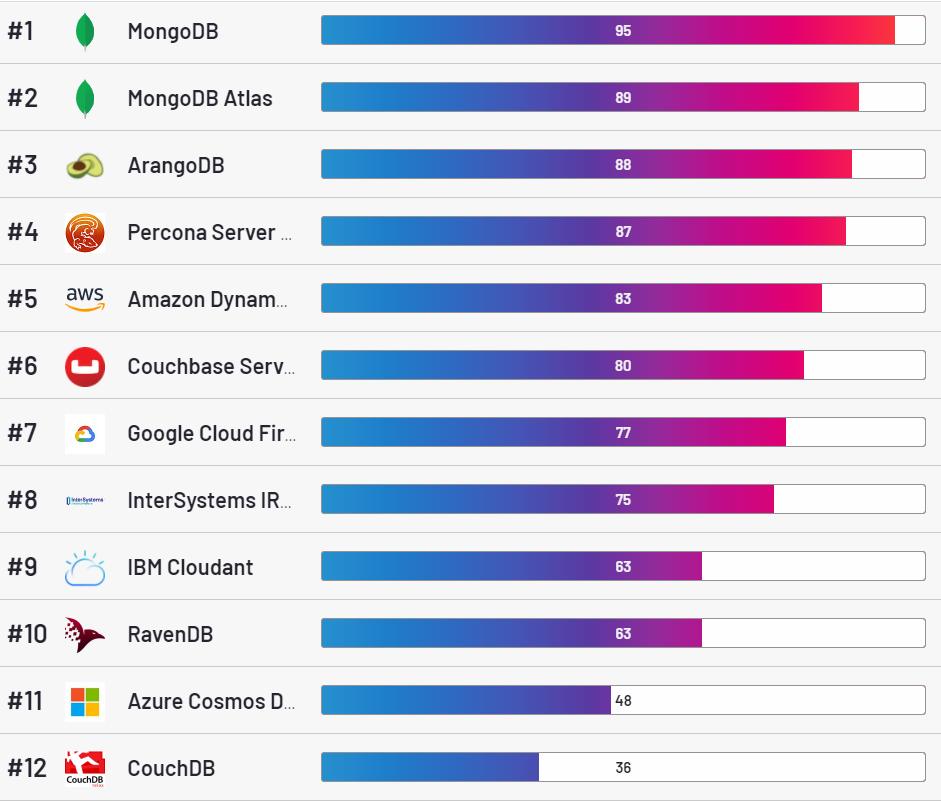
Dvajt Meriman i Eliot Horovic su kreirali MongoDB. Oni su se susreli sa problemima u razvoju i skalabilnosti sa tradicionalnim pristupima relacionim bazama podataka dok su pravili web aplikacije u DoubleClick-u, kompaniji za onlajn oglašavanje koja je sada u vlasništvu Google Inc. Ime baze podataka je izvedeno iz reč ogromna da predstavlja ideju podrške velikim količinama podataka. [1]

**2.1 Karakteristike MongoDB baze podataka**

Neke od karakteristika MongoDB baze podataka:

* MongoDB je *dokumentno orijentisana* NoSQL baza podataka koja je dizajnirana za skladištenje velikog obima podataka i takođe omogućava da rad sa podacima bude efikasan . Umesto da koristi tabele i redove kao u tradicionalnim relacionim bazama podataka, MongoDB koristi kolekcije i dokumente. Dokumenti se sastoje od parova ključ/vrednost koji su osnovna jedinica podataka u MongoDB-u. I svaki dokument sadrži svoj jedinstveni ID objekta. Kolekcije sadrže skupove dokumenata i funkcije koje su ekvivalentne tabelama relacionih baza podataka.[1]
* Kao i druge NoSQL baze podataka, MongoDB *ne zahteva unapred definisanu šemu* . To je odlična karakteristika koju pruža MongoDB. Baza podataka bez šema znači da jedna kolekcija može da sadrži različite tipove dokumenata u sebi. Ili drugim rečima, u MongoDB bazi podataka, jedna kolekcija može da sadrži više dokumenata i ti dokumenti se mogu sastojati od različitog broja polja, sadržaja i veličine. Nije neophodno da jedan dokument bude sličan drugom dokumentu kao u relacionim bazama podataka. Zbog ove sjajne funkcije, korisnicima se nudi fleksibilnost da kreiraju bilo koji broj polja u dokumentu, što olakšava skaliranje MongoDB baza podataka u poređenju sa relacionim bazama podataka.[2]
* *Indeksiranje*: U bazi podataka MongoDB, svako polje u dokumentima je indeksirano primarnim i sekundarnim indeksima, što olakšava i oduzima manje vremena za dobijanje ili pretragu podataka iz skupa podataka. Ako podaci nisu indeksirani, onda baza podataka pretražuje svaki dokument sa navedenim upitom što oduzima mnogo vremena i nije tako efikasno.[2]
* *Skalabilnost*: MongoDB obezbeđuje horizontalnu skalabilnost uz pomoć sharding(deljenje). Deljenje znači distribuciju podataka na više servera, ovde se velika količina podataka deli na delove podataka pomoću ključa deljenja, a ovi delovi podataka su ravnomerno raspoređeni na delove koji se nalaze na mnogim fizičkim serverima. Takođe će dodati nove mašine u pokrenutu bazu podataka.
* *Replikacija*: MongoDB obezbeđuje visoku dostupnost i redundantnost uz pomoć replikacije, kreira više kopija podataka i šalje te kopije na drugi server, tako da ako jedan server ne uspe, onda se podaci preuzimaju sa drugog servera.
* *Agregacija*: Omogućava izvođenje operacija na grupisanim podacima i dobijanje jednog ili izračunatog rezultata. Slično je klauzuli SQL GROUPBY. Obezbeđuje tri različite agregacije, tj. cevovod agregacije, funkciju smanjenja mape i metode agregacije za jednu svrhu
* *Visoke performanse:* Performanse MongoDB-a su veoma visoke i postojanost podataka u poređenju sa drugom bazom podataka zbog njenih karakteristika kao što su skalabilnost, indeksiranje, replikacija itd.
* Takođe pruža zvaničnu podršku za drajvere za sve popularne jezike kao što su C, C++, C# i .Net, Go, Java, Node.js, Perl, PHP, Pithon, Motor, Rubi, Scala, Svift, Mongoid. Dakle, mogu da se kreiraju aplikacije koristeći bilo koji od ovih jezika. Danas postoji toliko mnogo kompanija koje koriste MongoDB kao što su Facebook, Nokia, eBai, Adobe, Google, itd. za skladištenje velike količine podataka.[2]

Na slici 1. su prikazani najpopularniji primeri dokument baze podataka na osnovu korišćenja i ocene korisnika. Medju njima na prvom mestu se nalazi MongoDB baza podataka.



Slika 1. Prikaz najpopularnijih primera dokument baze podataka [3]

**2.2 Prednosti MongoDB baze podataka**

Neke od prednosti baze podataka su:

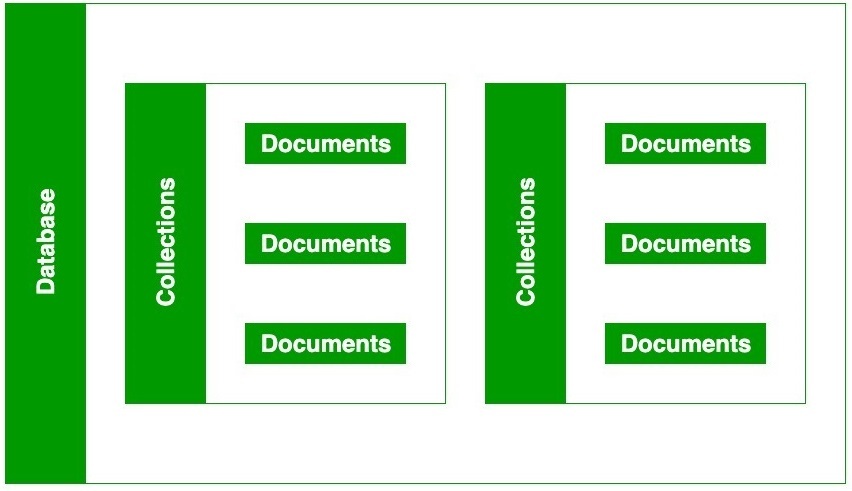
* Jedna od prednosti korišćenja dokumenata je ta što se ovi objekti mapiraju u izvorne tipove podataka u brojnim programskim jezicima. Takođe, ugrađeni dokumenti smanjuje potrebu za spajanjem baze podataka, što može smanjiti troškove.[1]
* To je NoSQL baza podataka bez šeme. Ne mora da se dizajnira šema baze podataka kada se radi sa MongoDB.
* Ne podržava operaciju pridruživanja.
* Pruža veliku fleksibilnost poljima u dokumentima.
* Sadrži heterogene podatke.
* Pruža visoke performanse, dostupnost, skalabilnost.
* To je dokumentno orijentisana baza podataka i podaci se čuvaju u BSON dokumentima.
* Takođe podržava prelaz višestrukih dokumenata ACID
* Ne zahteva nikakvu SQL injekciju.
* Lako se integriše sa Big Data Hadoop-om [2]

**2.3 Nedostaci MongoDB baze podataka**

* Koristi veliku memoriju za skladištenje podataka.
* Nije dozvoljeno da se skladišti više od 16 MB podataka u dokumentima.
* Ugnežđenje podataka u BSON-u je takođe ograničeno, nije dozvoljeno da se ugnežde podaci više od 100 nivoa.

1. **Struktura baze podataka**

MongoDB arhitektura je zasnovana na nekoliko komponenti kao što su baze podataka, kolekcije, dokumenti. Baza podataka sadrži kolekciju, a kolekcija sadrži dokumente i dokumenti sadrže podatke, oni su međusobno povezani. Hijararhija pomenutih elemenata baze podataka prikazana je na slici 2. [4]

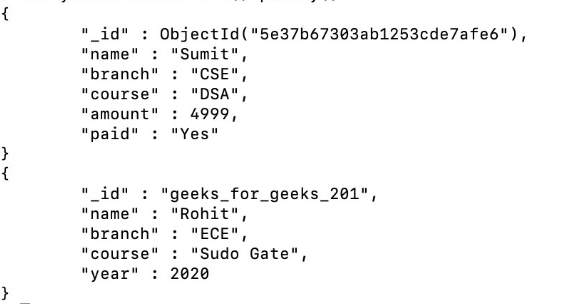
****

Slika 2 Struktura baze podataka [2]

Na MongoDB serveru se može kreirati više baza podataka. [4]

* 1. **Kolekcija**

Kolekcije su isto kao tabele u relacionim bazama podataka, one takođe čuvaju podatke, ali u obliku dokumenata. U jednoj bazi podataka je dozvoljeno skladištenje više kolekcija. MongoDB baza podataka je baza podataka bez šema. Dakle, u kolekciji nije neophodno da je šema jednog dokumenta slična drugom dokumentu. Ili drugim rečima, jedna kolekcija sadrži različite tipove dokumenata kao što je prikazano na slici 3. u nastavku gde kolekcija sadrži dve različite vrste dokumenata. [4] Postojanje kolekcije u bazi podataka nije potrebno pre umetanja dokumenta u MongoDB. Iz programskog jezika, kada se doda prvi dokument u kolekciju i obezbeđena kolekcija ne postoji u toj bazi podataka, MongoDB kreira novu kolekciju. Dakle, može se reći, za umetanje dokumenta, kolekcija ne mora biti prisutna u MongoDB.



Slika 3. Kolekcija koja sadrži dokumente različite strukture [4]

U MongoDB postoje različite vrste kolekcija:

* *Normalna kolekcija* za održavanje normalne kolekcija
* *Ograničena kolekcija za podatke poput reda*
* *TTL kolekcije*
* *Sistemska kolekcija*, ovo je unapred definisana kolekcija
* *GridFS kolekcija* za skladištenje velikih datoteka
* *Indeksirana kolekcija*
* *Geoprostorna kolekcija* za geografske i dugačke ili 2D podatke ili sferne podatke
* *Kolekcija za pretraživanje* celog teksta za jednostavnu pretragu teksta/stringova [5]

**3.1.1 Ograničena kolekcija**

Ograničene kolekcije su kolekcije fiksne veličine, definisan je broj dokumenata koji može da se smesti.Ograničene kolekcije funkcionišu na način sličan kružnim baferima: kada kolekcija ispuni svoj dodeljeni prostor, ona pravi prostor za nove dokumente tako što prepisuje najstarije dokumente. Ograničene kolekcije garantuju očuvanje redosleda umetanja. Kao rezultat, upitima nije potreban indeks za vraćanje dokumenata po redosledu umetanja. Bez ovih dodatnih troškova indeksiranja, ograničene kolekcije mogu podržati veći protok umetanja. [6] Određene operacije nisu dozvoljene na ograničenim kolekcijama. Dokumenti se ne mogu ukloniti ili izbrisati, a ažuriranja koja bi izazvala povećanje veličine dokumenata nisu dozvoljena. Sprečavanjem ove dve operacije, garantuje se da se dokumenti u ograničenoj kolekciji čuvaju po redosledu umetanja. [7]

**3.1.2 TTL kolekcije**

TTL kolekcije omogućavaju čuvanje podataka u MongoDB-u i automatski se uklanjanjaju podaci nakon određenog broja sekundi ili u određeno vreme. Istek podataka je koristan za neke klase informacija, uključujući mašinski generisane podatke o događajima, evidencije i informacije o sesiji koje treba da traju samo ograničeni vremenski period. Posebno svojstvo TTL indeksa podržava implementaciju TTL kolekcija. TTL funkcija se oslanja na pozadinsku nit mongod koja čita vrednosti otkucane datumom u indeksu i uklanja dokumente koji su istekli iz kolekcije. [8]

**3.1.3 Sistemske kolekcije**

MongoDB skladišti sistemske informacije u kolekcijama koje koriste <database>.system.\* imenski prostor , koji MongoDB rezerviše za internu upotrebu. Ne treba kreirati kolekcije koje počinju sa system. [9]

**3.1.4 Normalne kolekcije**

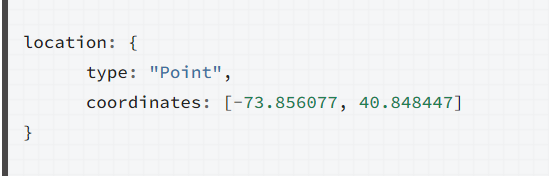
Normalne kolekcije u MongoDB-u se kreiraju dinamički i automatski povećavaju veličinu kako bi se uklopile u dodatne podatke. [7]

**3.1.5 Geoprostorne kolekcije**

U MongoDB-u mogu da se skladište geoprostorni podaaci kao GeoJSON objekte ili kao stare koordinatne parove. Da bi se izračunala geometrija nad sferom nalik Zemlji, podaci o lokaciji se čuvaju kao GeoJSON objekte. Da bi se naveli GeoJSON podaci, koristi se ugrađeni dokument sa:

* poljem pod imenom *type* koje specificira tip GeoJSON objekta i
* polje sa imenom *coordinates* koje navodi koordinate objekta. [10]

Primer GeoJSON objekta dat je na slici 4.



Slika 4. GeoJSON objekat [10]

* 1. **Dokument**

MongoDB skladišti podatke u BSON formatu interno i preko mreže, ali to ne znači da ne može da se zamisli MongoDB kao JSON bazu podataka. Sve što možete da se predstavi u JSON-u može biti izvorno uskladišteno u MongoDB-u i isto tako lako preuzeto u JSON-u. BSON je jednostavno skraćenica za Binari JSON i pedstavlja binarni prikaz JSON dokumenata. Binarna struktura BSON-a kodira informacije o tipu i dužini, što omogućava da se analizira mnogo brže. Od svoje početne formulacije, BSON je proširen da bi dodao neke opcione tipove podataka koji nisu JSON izvorni, kao što su datumi i binarni podaci, bez kojih bi MongoDB-u nedostajala neka dragocena podrška. Jezici koji podržavaju bilo koju vrstu složene matematike obično imaju cele brojeve različite veličine ili različite nivoe decimalne preciznosti (float, double, decimal128, itd.). Ne samo da je korisno biti u mogućnosti da se predstave te razlike u podacima uskladištenim u MongoDB-u, već takođe omogućava da se poređenja i proračuni dešavaju direktno na podacima na načine koji pojednostavljuju upotrebu koda aplikacije. [11]

Neke od prednosti BSON dokumenta koje ga ujedno i izdvajaju od ostalih su:

* BSON zapisi su obično manji od JSON zapisa, što je najveći razlog za korišćenje
* BSON datoteke se lako skladište i šalju kroz mrežu, što ih čini savršenim za skladištenje i slanje podataka. Održavanje prostornog opterećenja na minimumu važno je za svaki format predstavljanja podataka, posebno kada se koristi preko mreže.
* BSON je napravljen da efikasno skladišti prostor i skenira: u BSON dokumentu, veliki elementi imaju prefiks sa poljem za dužinu. Dokumenti u BSON-u koriste više prostora od JSON-a zbog prefiksa dužine i eksplicitnih indeksa niza, ali zahvaljujući tim prefiksima, u mogućnosti smo da skeniramo i postavljamo upite mnogo brže. Prefiksi olakšavaju poređenje i izračunavanje direktno na podacima, pojednostavljujući potrošnju koda aplikacije.
* Za razliku od JSON-a, u BSON-u možete pronaći tipove podataka kao što su Bindata, Minkey, Maxkey, Binarni podaci, ObjectID, Regular Ekpression, JavaScript, Decimal128 i Date za datum i vreme. Ovi tipovi podataka su ključni kada radite sa specijalnim programima.
* Mnogi savremeni sistemi koriste aritmetiku sa pokretni zarezom zasnovanu na binarnom sistemu da bi predstavili tačne decimalne razlomke preko Float i dvostrukih tipova podataka. Ovi tipovi podataka daju aproksimaciju, ali ne i preciznu vrednost. Korišćenje BSON-a u takvim slučajevima daje vam visoku preciznost. [12]

**3.2.1. Struktura dokumenta**

MongoDB dokumenti su sastavljeni od parova polja i vrednosti i imaju strukturu kao na slici 5



Slika 5. Sintaksa strukture BSON dokumenta [13]

*Vrednost polja* može biti bilo koji od BSON tipova podataka, uključujući druge dokumente, nizove i nizove dokumenata. Svaki BSON tip ima i integer i niz identifikatore kao što je navedeno na slici 6.:



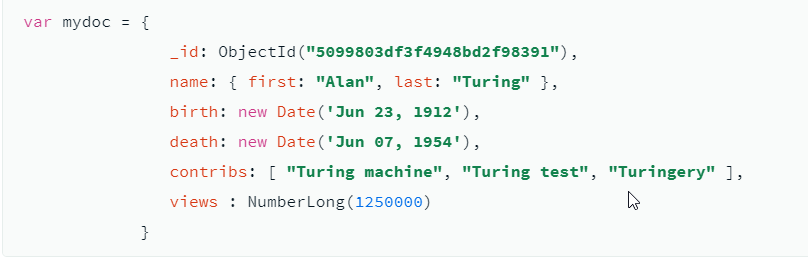
Slika 6. BSON tipovi [14]

*Imena polja* su nizovi. Dokumenti imaju sledeća ograničenja za nazive polja:

* Ime polja \_id je rezervisano za primarni ključ. Njegova vrednost mora biti jedinstvena u kolekciji, nepromenljiva je i može biti bilo kog tipa osim niza. Ako \_id sadrži potpolja, nazivi potpolja ne mogu početi sa ( $).
* Imena polja ne mogu da sadrže null karakter.
* Server dozvoljava skladištenje imena polja koja sadrže tačke ( .) i znakove dolara ( $).[13]

Primer dokumenta je na slici 7. Polja u primeru sa slike 7. imaju sledeće tipove podataka:

* \_*id* sadrži ObjectId .
* *name* sadrži ugrađeni dokument koji sadrži polja firsti last.
* *birth* i *death* čuvaju vrednosti tipa Datum .
* *contribs* sadrži niz stringova .
* *views* sadrži vrednost tipa NumberLong. [13]

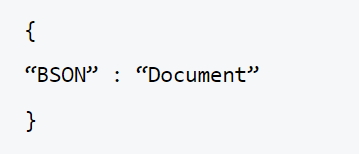


Slika 7. Primer dokumenta [13]

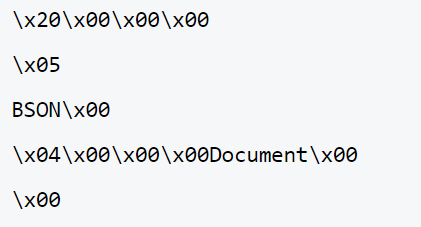
U MongoDB-u, svaki dokument uskladišten u kolekciji zahteva jedinstveno polje \_id koje deluje kao primarni ključ. Ako umetnuti dokument izostavi \_id polje, MongoDB drajver automatski generiše ObjectId za \_id polje. MongoDB klijenti treba da dodaju \_id polje sa jedinstvenim ObjectId-om. ObjectId-ovi su mali, verovatno jedinstveni, brzo se generišu i poređaju. ObjectId vrednosti su dugačke 12 bajtova, a sastoje se od:

* Vremenska oznaka od 4 bajta, koja predstavlja kreiranje ObjectId-a, mereno u sekundama.
* 5-bajtna nasumična vrednost koja se generiše jednom po procesu. Ova nasumična vrednost je jedinstvena za mašinu i proces.
* Brojač sa povećanjem od 3 bajta, inicijalizovan na slučajnu vrednost. [14]

U nastavku biće prikazan primer kako se BSON dokumenet čuva u bazi podataka. Na slici 8. prikazan je primer JSON dokumenta. Na slici 9. Prikazano je kako se JSON dokument konvertuje čuva u bazi kao BSON dokument. [15]



Slike 8. JSON dokument [15]



Slika 9. BSON dokuement [15]

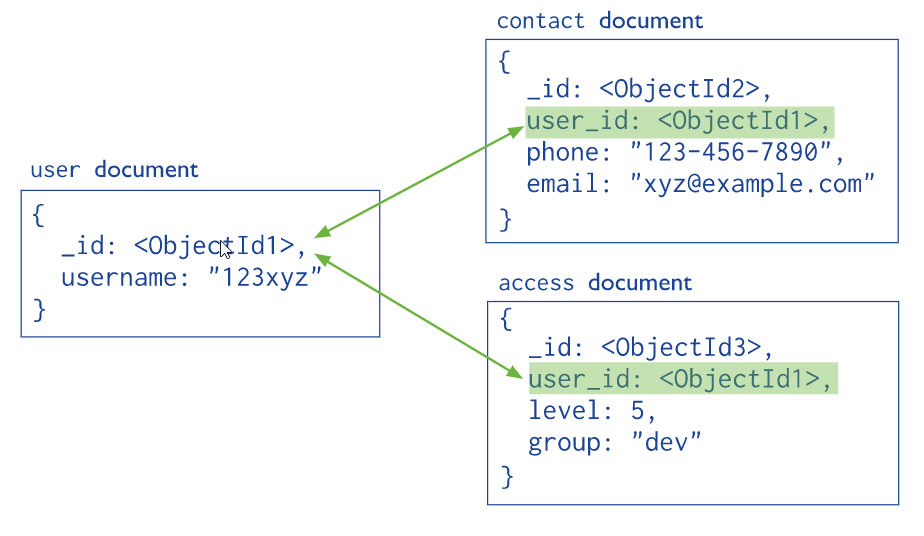
Prvi red označava veličinu dokumenta koja je bila potrebna za skladištenje dokumenta. Drugi red označava tip stringa koji smo koristili za čuvanje dokumenta.Treći red označava ono ime polja koje smo koristili u našim dokumentima. Četvrti red označava vrednost polja za ime polja koje smo koristili u našim dokumentima. Poslednji red označava kraj objekta. Ne možemo da čitamo BSON podatke, oni nisu u formatu čitljivom za ljude, jedine mašine mogu da čitaju BSON tip podataka. Kodiranje formata tipa BSON u MongoDB je binarno. BSON je podržan naprednijim tipom podataka u poređenju sa JSON-om u MongoDB-u. BSON je više bez šema u poređenju sa tipom JSON. [15]

Mogu postojati dva načina za uspostavljanje odnosa između podataka u MongoDB:

* Referentni dokumenti
* Ugradjeni dokumenti

*Referentni dokumenti:* Reference čuvaju odnose između podataka uključujući veze ili reference iz jednog dokumenta u drugi. Aplikacije mogu da razreše ove reference da bi pristupile povezanim podacima. Uopšteno govoreći, ovo su normalizovani modeli podataka. Primer ovog dokumenta prikazan je na slici 10. [15]

*Ugradjeni dokumenti:* obuhvataju odnose između podataka tako što čuvaju povezane podatke u jednoj strukturi dokumenta. MongoDB dokumenti omogućavaju ugrađivanje struktura dokumenta u polje ili niz unutar dokumenta. Ovi denormalizovani modeli podataka omogućavaju aplikacijama da preuzimaju i manipulišu povezanim podacima u jednoj operaciji baze podataka. Primer ovog dokumenta prikazan je na slici 11.[15]



Slika 10. Referentni dokumenti [16]



Slika 11. Ugradjeni dokumenti

**3.2.2 Ograničenje veličine dokumenta**

Maksimalna veličina BSON dokumenta je 16 megabajta. Maksimalna veličina dokumenta pomaže da se osigura da jedan dokument ne može da koristi preveliku količinu RAM-a ili, tokom prenosa, preveliku količinu propusnog opsega. Za skladištenje dokumenata većih od maksimalne veličine, MongoDB obezbeđuje GridFS API. [17]

**3.2.3 GridFS**

*GridFS* je specifikacija za čuvanje i preuzimanje datoteka koje premašuju ograničenje veličine BSON dokumenta od 16 MB. Umesto čuvanja datoteke u jednom dokumentu, GridFS deli datoteku na delove i čuva svaki deo kao poseban dokument. [18]

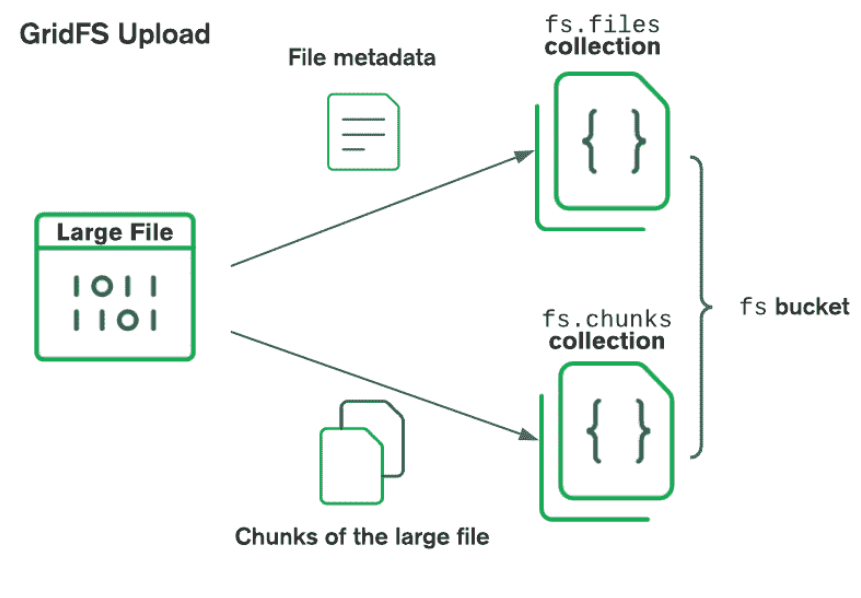
**3.2.3.1 GridFS kolekcije**

Podrazumevano, GridFS koristi podrazumevanu veličinu komada od 255 kB, to jest, GridFS deli datoteku na delove od 255 kB sa izuzetkom poslednjeg dela. Poslednji deo je samo onoliko koliko je potrebno. Slično, datoteke koje nisu veće od veličine komada imaju samo završni komad, koristeći samo onoliko prostora koliko je potrebno plus neke dodatne metapodatke. GridFS koristi dve kolekcije za skladištenje datoteka, jedna kolekcija čuva delove datoteka, a druga čuva metapodatke datoteka:

* *chunks* skladišti binarne delove
* *files* skladišti metapodatke datoteke. [18]

GridFS smešta ove dve kolekcije u zajedničku *korpu* tako što svakoj kolekciji stavlja prefiks sa imenom korpe. Podrazumevani naziv korpe je *fs*  ukoliko se ne navede drugačije, tako da dobijamo *fs.files* i *fs.chunks*  nazive kolekcija*.* [18]

Kada GridFS kreira novu *korpu,* kreiraju se kolekcije *chunks* i *files,* i dodaje im prefiks korpe. Ukoliko se ne navede drugačije ime, dodaje im se podrazumevani prefiks, kao što smo pomenuli iznad a to je prefiks *fs.* Upravljački program za ove dve kolekcije automatski kreira indeks, kako bi se omogućilo efikasno preuzimanje datoteka i povezanih metapodataka. Korpa se kreira pri prvom upisu ukoliko ne postoji. Da bi skladištili datoteke sa GridFS-om, datoteke se dele na menje delove, a svaki od tih delova predstavlja poseban dokument u *chunks* kolekciji. Kreira se dokument u *files* kolekciji. Sledeća slika 12. prikazuje kako GridFS deli datoteke prilikom otpremanja u korpu. [19]



Slika 12. Postupak deljenja datoteke kada se koristi GridFS [19]

Prilikom preuzimanja datoteka, GridFS preuzima metapodatke iz *files* kolekcije u navedenoj korpi i koristi informacije da rekonstruiše datoteku iz dokumenata u *chunks* kolekciji. Da bi sačuvali i preuzeli datoteke koristeći GridFS , koristi se jedno od sledećeg:

* MongoDB drajver
* Alat mongofiles komandne linije. Uslužni mongofiles program omogućava manipulaciju datotekama uskladištenim u MongoDB u GridFS objektima iz komandne linije. Posebno je koristan jer obezbeđuje interfejs između objekata uskladištenih u sistemu datoteka i GridFS-a. [20] [18]

Na slici 11. prikazan je dokument kolekcije *chunks*:

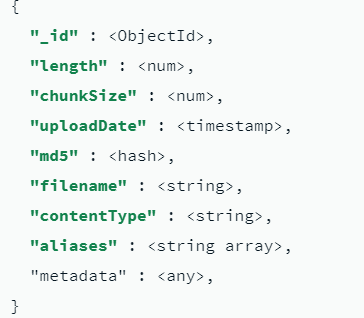


Slika 13. Dokument kolekcije *chunks* [18]

Dokument sadrži sledeća polja:

* *chunks.\_id* : koji predstavlja jedinstveni ObjectId
* *chunks.files\_id* : id roditeljskog dokumenta
* *chunks.n:* redni broj komada. GridFS numeriše sve delove počevši od 0
* *chunks.data:* korisno opterećenje komada kao BSON BinaryData

Svaki dokument u *files* kolekciji predstavlja datoteku u GridFS-u. Na slici 14. prikazan je dokument *files* kolekcije.



Slika 14. Dokument *files* kolekcije [18]

Dokumenti *files* kolekcije sadrže neka ili sva sledeća polja:

* *files.\_id:* Jedinstveni identifikator za ovaj dokument. Tip \_id je podataka koji je izabran za originalni dokument. Podrazumevani tip za MongoDB dokumente je BSON ObjectId .
* *files.length:* veličina dokumenta u bajtovima
* *files.chunkSize:* Veličina svakog komada u bajtovima . GridFS deli dokument na delove veličine chunkSize, osim poslednjeg, koji je samo onoliko koliko je potrebno. Podrazumevana veličina je 255 kilobajta (kB).
* *files.uploadData*: Datum kada je GridFS prvi put uskladištio dokument. Ova vrednost ima Date tip.
* *files.md5:* MD5 algoritam je zabranjen FIPS 140-2. MongoDB drajveri odbijaju podršku za MD5 i ukloniće MD5 generaciju u budućim izdanjima. Aplikacije koje zahtevaju sažetak datoteke treba da ga implementiraju izvan GridFS-a i čuvaju u files.metadata.
* *files.filename:* opciono, čoveku čitljivo ime za datoteku GridFS
* *files.contentType:* Opciono. Važeći MIME tip za datoteku GridFS. Samo za aplikaciju. Koristi files.*metadata se* za čuvanje informacija koje se odnose na MIME tip datoteke GridFS.
* *files.aliases:* Opciono. Niz nizova alijasa. Samo za aplikaciju. Koristi *files.metadatase* za čuvanje informacija koje se odnose na MIME tip datoteke GridFS.
* *files.metadata:* Opciono. Polje metapodataka može biti bilo kog tipa i može da sadrži sve dodatne informacije koje želite da sačuvate. Ako želite da dodate dodatna proizvoljna polja dokumentima u *files* kolekciji, dodajte ih objektu u polju metapodataka. [18]

**3.2.3.2 GridFS indeksi**

Kao što je navedeno u poglavlju iznad, GridFS koristi indekse za svaku od kolekcija *chunks* i *files* radi efikasnosti. Drajveri automatski kreiraju ove indekse radi pogodnosti. Takođe mogu da se kreiraju bilo koji dodatni indeksi po želji da odgovaraju potrebama aplikacije.

Kod *chunks* kolekcija, GridFS koristi jedinstveni , složeni indeks pomoću polja *files\_id* i *. n* . Ovo omogućava efikasno preuzimanje delova, kao što je prikazano na sledećoj slici 15.



Slika 15. Prezimanje delova pomoću *files\_id*  i *.n* [18]

Ako ovaj indeks ne postoji, može se izvršiti sledeća operaciju sa slike 16. da bi se kreirao koristeći mongosh:



Slika 16. Kreiraje indeksa pomoću mongosh [18]

GridFS koristi indeks na *files* kolekciji koristeći polja *filename* i . *uploadDate*. Ovaj indeks omogućava efikasno preuzimanje datoteka, kao što je prikazano na slici 17.:



Slika 17. Preuzimanje datoteke uz pomoću indeksa [18]

Ako ovaj indeks ne postoji, možete izvršiti sledeću operaciju da biste ga kreirali koristeći mongosh kao na slici 18:



Slika 18. Kreiranje indeksa pomoću mongosh [18]

**3.2.3.3 GridFS Sharding**

Da bi se podelila ***chunks*** ***kolekcija***, koristi se složeni { files\_id : 1, n : 1 }ili { files\_id : 1 }kao indeks ključeva šarda. Ukoliko MongoDB drajveri ne pokreću *filemd5* da verifikuju uspešno otpremanje može se koristiti Hashed Sharding za *chunks* kolekciju. Ako je pokretački program MongoDB pokrenut *filemd5*, ne može se koristiti *Hashed Sharding* . [18]

Komanda *filemd5* vraća md5 heš za jednu datoteku sačuvanu pomoću GridFS specifikacije. Klijentske biblioteke koriste ovu komandu da bi proverile da li su datoteke ispravno upisane u MongoDB. Komanda uzima files\_id za dotičnu datoteku i ime korenske kolekcije GridFS-a kao argumente. Na primer:



Slika 19. Komanda filemd5 [21]

MongoDB izračunava filemd5 koristeći sve podatke u objektu datoteke GridFS izvučene uzastopno iz svakog dela u *chunks* kolekciji. [21] *Hashed Sharding* koristi ili heširani indeks jednog polja ili složeni heširani indeks) kao ključ za particionisanje podatak.

***Kolekcija*** ***files*** se često ne dele jer sadrže samo metapodatke i veoma su male. Dostupni ključevi takođe ne pružaju ravnomernu distribuciju . Međutim, ako neko treba da se podeli *kolekciju files*, može se koristiti polje \_id u kombinaciji sa nekim poljima.

**3.2.3.4 Ograničenja GridFS**

GridFS sistem datoteka ima sledeća ograničenja:

* *Atomsko ažuriranje*: GridFS nema atomsko ažuriranje. Ovo olakšava ručno ažuriranje odabirom potrebne verzije datoteka i održavanjem više verzija datoteka
* *Performanse*: sistem ima tendenciju da bude spor sa sistemom datoteka i web serverom.
* *Radni set*: jedan koristi drugi server kada radi na novom radnom setu. Ovo se radi kako bi se izbeglo ometanje radnog skupa koji radi. [22]

**3.2.3.5 Primena GridFS**

GridFS je koristan ne samo za čuvanje datoteka koje prelaze 16 MB, već i za čuvanje svih datoteka kojima želite da pristupite bez potrebe da učitavate celu datoteku u memoriju. Situacije u kojima se još primenjuje GridFS u MongoDB bazi podataka:

* Ako sistem datoteka ograničava broj datoteka u direktorijumu, može se koristiti GridFS da uskladištiti onoliko datoteka koliko je potrebno.
* Kada želimo da pristupite informacijama iz delova velikih datoteka bez potrebe da učitavate cele datoteke u memoriju, može se koristiti GridFS da bi se pozvali delove datoteka bez čitanja cele datoteke u memoriju.
* Ukoliko želimo da datoteke i metapodaci budu automatski sinhronizovani i raspoređeni na brojnim sistemima i objektima, može se koristiti GridFS. Kada se koriste geografski distribuirane skupove replika , MongoDB može automatski distribuirati datoteke i njihove metapodatke na brojne mongodinstance i objekte. [18]

Ukoliko je potrebno automatski ažurirati sadržaj cele datoteke nije preporučljivo koristiti GridFS. Kao alternativu, možemo da sačuvamo više verzija svake datoteke i navedemo trenutnu verziju datoteke u metapodacima. Možemo ažurirati polje metapodataka koje ukazuje na status „najnoviji“ u atomskom ažuriranju nakon otpremanja nove verzije datoteke, a kasnije ukloniti prethodne verzije ako je potrebno. Štaviše, ako su datoteke manje od ograničenja veličine BSON dokumenta od 16 MB , možemo da čuvanmo datoteke u jednom dokumentu umesto da se koristi GridFS. Možemo da koristimo tip podataka BinData za skladištenje binarnih podataka. [18]

1. **Mehanizmi za skladištenje**

Mehanizam za skladištenje je komponenta baze podataka koja je odgovorna za upravljanje načinom na koji se podaci čuvaju, kako u memoriji tako i na disku. MongoDB podržava više mehanizama za skladištenje, jer različiti mehanizmi rade bolje za određena radna opterećenja. Odabir odgovarajućeg mehanizma za skladištenje za slučaj upotrebe može značajno uticati na performanse vaših aplikacija. Mehanizmi za skladištenje koje MongoDB baza podataka podržava:

* WiredTiger mehanizam (Default)
* In-Memory mehanizam [23]

**4.1 WiredTiger mehanizam (Default)**

WiredTiger je podrazumevani mehanizam za skladištenje koji počinje u MongoDB 3.2. Pogodan je za većinu radnih opterećenja i preporučuje se za nove primene. WiredTiger pruža model istovremenosti na nivou dokumenta, kontrolne tačke i kompresiju, između ostalih funkcija. [23]

**4.1.1 Konkurencija na nivou dokumenta**

WiredTiger koristi kontrolu istovremenosti na nivou dokumenta za operacije pisanja. Kao rezultat toga, više klijenata može istovremeno da menja različite dokumente kolekcije. WiredTiger mehanizam za skladištenje donosi zaključavanje nivoa dokumenta u MongoDB, što znači da upisi više ne blokiraju kolekciju ili bazu podataka [24]. Kada mehanizam za skladištenje otkrije sukobe između dve operacije, jedna će izazvati konflikt pisanja što će dovesti do toga da MongoDB transparentno ponovo pokuša tu operaciju. [23]

**4.1.2 Snimci i kontrolne tačke**

WiredTiger koristi MultiVersion Concurrenci Control (MVCC). Na početku operacije, WiredTiger obezbeđuje snimak podataka u trenutku operacije. Snimak predstavlja konzistentan prikaz podataka u memoriji. [23]

Kada piše na disk, WiredTiger upisuje sve podatke u snimku na disk na konzistentan način u svim datotekama sa podacima. Sada trajni podaci deluju kao kontrolna tačka u datotekama podataka. Kontrolna tačka osigurava da su datoteke podataka konzistentne do i uključujući poslednju kontrolnu tačku; odnosno kontrolni punktovi mogu delovati kao tačke oporavka. [23]

Počevši od verzije 3.6, MongoDB konfiguriše WiredTiger za kreiranje kontrolnih tačaka (tj. zapisivanje snimaka podataka na disk) u intervalima od 60 sekundi. U ranijim verzijama, MongoDB postavlja kontrolne tačke da se javljaju u WiredTiger-u na korisničkim podacima u intervalu od 60 sekundi ili kada se upiše 2 GB podataka dnevnika, šta god se prvo dogodi. [23]

Tokom pisanja nove kontrolne tačke, prethodna kontrolna tačka je i dalje važeća. Kao takav, čak i ako MongoDB prekine ili naiđe na grešku tokom pisanja nove kontrolne tačke, nakon ponovnog pokretanja, MongoDB se može oporaviti od poslednje važeće kontrolne tačke. Nova kontrolna tačka postaje dostupna i trajna kada se WiredTigerova tabela metapodataka atomski ažurira da bi se upućivala na novu kontrolnu tačku. [23]

**4.1.3 Kompresija**

Sa WiredTiger-om, MongoDB podržava kompresiju za sve kolekcije i indekse. Kompresija minimizira upotrebu skladišnog prostora na račun dodatnog CPU-a. Podrazumevano, WiredTiger koristi kompresiju blokova sa bibliotekom brze kompresije za sve kolekcije i kompresiju prefiksa za sve indekse. Za kolekcije su dostupne i sledeće biblioteke kompresije blokova: zlib i zstd (dostupno počevši od MongoDB 4.2)

* 1. **In-Memory mehanizam**

Osim nekih metapodataka i dijagnostičkih podataka, mehanizam za skladištenje u memoriji ne održava nikakve podatke na disku, uključujući konfiguracione podatke, indekse, korisničke akreditive itd. [25]

Mehanizam za skladištenje u memoriji koristi kontrolu istovremenosti na nivou dokumenta za operacije pisanja. Kao rezultat toga, više klijenata može istovremeno da menja različite dokumente kolekcije. [25]

Mehanizam za skladištenje u memoriji je nepostojan i ne upisuje podatke u trajno skladište. Nepostojani podaci obuhvataju podatke aplikacije i sistemske podatke, kao što su korisnici, dozvole, indeksi, konfiguracija skupa replika, konfiguracija podeljenog klastera itd. [25]

1. **Zaključak**

MongoDB je jedna od NoSQL baza podataka bez šeme. Ne mora da se dizajnira šema baze podataka kada se radi sa MongoDB. Pruža veliku fleksibilnost poljima u dokumentima, sadrži heterogene podatke, pruža visoke performanse, dostupnost, skalabilnost.

U uvodnom delu ovog rada najpre su prikazane osnovne karakteristike MongoDB baze podataka, kako bi se upoznali malo bolje sa bazom podataka. MongoDB koristi dokument kao format u kom skladišti podatke. Na taj način omogućava da se podaci koji se smeštaju budu bilo koje stukture. Pritom, dokumenti ne moraju da imaju istu strukturu, sto je jedna od prednosti koju nudi ova baza podataka, pa šema ne mora da bude definisana.

Nakon ovog uvodnog dela objašnjena je arhitektura baze podataka, gde su pobrojane osnovne komponente kao što su baza podataka, kolekcija i dokumenti. Detaljnije su opisane sve komponente. Nabrojane su i opisane ukratko vrste kolekcija, kako funkcionišu i koje su njihove karakteristike. Neke od kolekcija koje su pomenute su: normalne kolekcije, ograničene kolekcije, TTL kolekcije, Geoprostorne kolekcije, kolekcije za pretraživanje, sistemske kolekcije, Indeksirane kolekcije.

Opisano je takodje i način na koji MongoDB smešta podatke. MongoDB koristi BSON format za smeštanje podataka u dokumentu. Nabrojane su prednosti zbog kojih se koristi BSON a ne JSON format. Neke od njih su: BSON zapisi su manji od JSON, BSON datoteke se lakše skladište i šalju kroz mrežu, BSON koristi više tipova podataka. Opisana je struktura dokumenta, on se sastoji od niza polja i vrednosti. Nakon toga opisan je način za uspostavljanje odnosa između podataka u MongoDB-u: referentni dokumenti i ugradjeni dokumenti. Kako je veličina dokumenta koja se skladišti ograničena potrebno je da se koristi GridFS sistem kako bi se smestili dokumenti koji su veći od 16 megabajta. Na kraju su opisani mehanizmi koje MongoDB koristi kako bi smestio podatke a to su WiredTiger mehanizam i In-Memory mehanizam.

Zahvaljujući svojoj fleksibilnosti, skalabilnosti i jednostavnosti korišćenja, MongoDB baza podataka postaje sve popularniji izbor baze podataka za programere aplikacija. Ona je dobro prilagođeni različitim aplikacijama i dobro rade sama ili kao deo većih ekosistema sa više baza podataka. Jedna od prednosti je što imati unapred definisanu šemu ili strukturu za svoje podatke nije neophodno sa MongoDB. Mogu da se kreiraju dokumenti bez prethodnog definisanja strukture podataka (npr. polja, vrste njihovih vrednosti) i može lako da se promeni struktura dokumenata dodavanjem novih polja ili brisanjem postojećih.

**6. Literatura**

[1] MongoDB, [https://www.techtarget.com/searchdatamanagement/definition/MongoDB](https://www.techtarget.com/searchdatamanagement/definition/MongoDB%20,%20%20pristupljeno%2014.4) , (pristupljeno 14.aprila 2022.)

[2] What is MongoDB – Working and Features, <https://www.geeksforgeeks.org/what-is-mongodb-working-and-features/> , (pristupljeno 14. aprila 2022.)

[3] The Top 20 Document Databases, <https://www.g2.com/categories/document-databases?tab=highest_rated> , (pristupljeno 16. aprila 2022.)

[4] MongoDB – Database, Collection, and Document, <https://www.geeksforgeeks.org/mongodb-database-collection-and-document/> , (pristupljeno 17. aprila 2022.)

[5] What is collection in MongoDB, <https://www.mysoftkey.com/mongodb/what-is-collection-in-mongodb/> , (pristupljeno 17. aprila 2022.)

[6] Capped collection, <https://www.mongodb.com/docs/manual/core/capped-collections/> , (pristupljeno 17.aprila 2022.)

[7] Special Index and Collection Types, <https://www.oreilly.com/library/view/mongodb-the-definitive/9781449344795/ch06.html> , (pristupljeno 25. aprila 2022.)

[8] Expire Data from Collections by Setting TTL, <https://www.mongodb.com/docs/manual/tutorial/expire-data/> , (pristupljeno 16.aprila 2022)

[9] System collection, <https://www.mongodb.com/docs/manual/reference/system-collections/> , (pristupljeno 16.aprila 2022)

[10] Geospatial Queries, <https://www.mongodb.com/docs/v4.0/geospatial-queries/> , (pristupljeno 25.aprila 2022.)

[11] JSON and BSON, <https://www.mongodb.com/json-and-bson> , (pristupljeno 16.aprila 2022)

[12] Explaining BSON with Examples, <https://www.mongodb.com/basics/bson> , (pristupljeno 16.aprila 2022)

[13] <https://www.mongodb.com/docs/manual/core/document/> , (pristupljeno 16.aprila 2022)

[14] BSON Types, <https://www.mongodb.com/docs/manual/reference/bson-types/> , (pristupljeno 16. aprila 2022)

[15] Definition of MongoDB BSON, <https://www.educba.com/mongodb-bson/> , (pristupljeno 16. aprila 2022)

[16] <https://www.mongodb.com/docs/manual/core/data-modeling-introduction/> , (pristupljeno 17. aprila 2022)

[17] MongoDB Limits and Thresholds, <https://www.mongodb.com/docs/manual/reference/limits/> , (pristupljeno 17.aprila 2022)

[18] GridFS, <https://www.mongodb.com/docs/manual/core/gridfs/> , (pristupljeno 17. aprila 2022)

[19] GridFS, <https://www.mongodb.com/docs/drivers/node/current/fundamentals/gridfs/> , (pristupljeno 17.aprila 2022)

[20] mongofiles, <https://www.mongodb.com/docs/database-tools/mongofiles/?_ga=2.195623340.121587754.1650116756-81978588.1634227231#mongodb-binary-bin.mongofiles> , (pristupljeno 17. aprila 2022)

[21] Filemd5, <https://www.mongodb.com/docs/manual/reference/command/filemd5/#mongodb-dbcommand-dbcmd.filemd5> , (pristupljeno 17. aprila 2022)

[22] Storing Files in MongoDB with GridFS, <https://severalnines.com/database-blog/storing-files-mongodb-gridfs> , (pristupljeno 17. aprila 2022)

[23] Storage Engines, <https://www.mongodb.com/docs/manual/core/storage-engines/> , (pristupljeno 25. aprila 2022)

[24] WiredTiger storage engine, <http://learnmongodbthehardway.com/schema/wiredtiger/> , (pristupljeno 25. aprila 2022)

[25] In-memory storage engine, <https://www.mongodb.com/docs/manual/core/inmemory/>, (pristupljeno 26. aprila 2022)