PTS Vaja 1

1. Opis domene in sistemskih zahtev

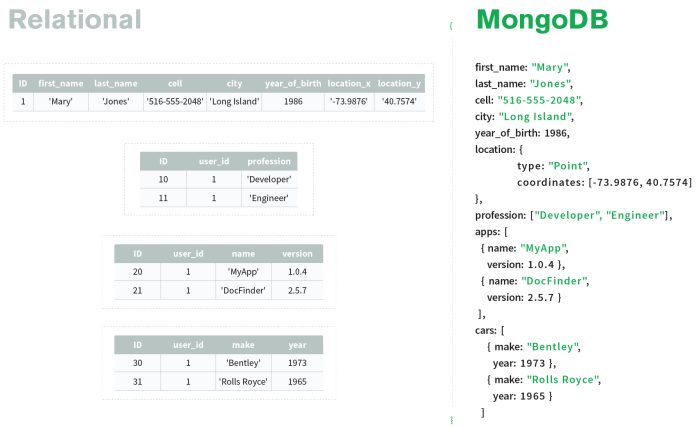
Lastnik verige restavracij želi izboljšati obdelavo naročil, ki prihajajo v posamezne restavracije, da bi olajšal delo svojim delavcem in hitrost obdelave naročil gostov. Glede na to, da se v sistemu beležijo podatki o naročilih oddanih preko spleta in osebno v restavraciji, se je odločil, da potrebuje porazdeljen sistem, ki bo v realnem času shranjeval sprejeta naročila, jih obdelal in beležil statistiko za natančno odločanje upraviteljev. V sistem bodo prihajali podatki iz različnih virov, in sicer iz relacijske podatkovne baze (podatki o ponudbi), Kafka teme (podatki o naročilih preko spleta) in API-ja (podatki o naročilih, ki jih vnesejo natakarji v restavraciji). Vse prispele podatke bomo shranili v sistemu z uporabo tehnologij in orodij dostopnih v ekosistemu Hadoop.

Za vajo 1 bomo uporabili naslednje tehnologije in orodja:

- MongoDB Atlas
- MongoDB Compass
- Postman
- Spark Java (opcijsko)

2. MongoDB

MongoDB je NoSQL dokumentna podatkovna shramba s podatkovnim modelom, ki temelji na dokumentih BSON in opcijski shemi, kot prikazano na sliki 1. Dokumenti so organizirani v zbirke (angl. collections), ki imajo podobno vlogo kot tabele v relacijskih bazah. Ena podatkovna baza lahko vsebuje eno ali več zbirk. Pri vnosu novega ali spremembi obstoječega dokumenta, vsak dokument pridobi samodejno generirani UUID kot unikatni ključ.



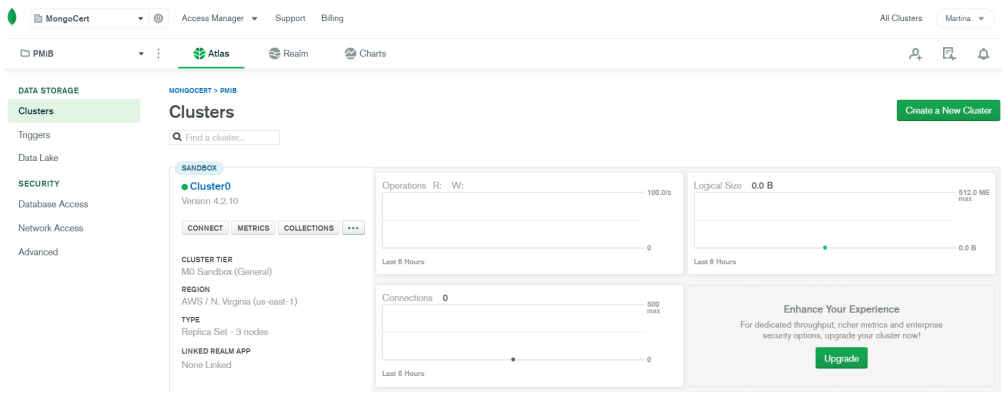
Slika 1: MongoDB podatkovni model.

Za dostop do MongoDB baze in izvedbo povpraševanj lahko uporabimo okolje *mongo shell* ali vizualno orodje *MongoDB Compass*.

3. MongoDB Atlas

Da se izognemo lokalni inštalaciji MongoDB strežnika bomo za oblikovanje MongoDB baze uporabili MongoDB Atlas. MongoDB Atlas je t.i. rešitev podatkovne baze kot storitve (*Database-as-a-service* (DBaaS)), ki omogoča lažjo konfiguracijo in delo z Mongo bazo v oblaku. Za potrebe vaj bomo uporabili brezplačni plan, ki se imenuje *free tier*, kjer bomo zgradili MongoDB gručo in pripravili bazo za vnos naročil.

Prvi korak je registracija na `https://www.mongodb.com/cloud/atlas`. Ko se prijavimo v storitev, ustvarimo nov projekt in ga poimenujemo *PTS*. Znotraj projekta bomo ustvarili brezplačno gručo na AWS-ju z osnovno konfiguracijo (ni treba spreminjati predlaganih postavk).

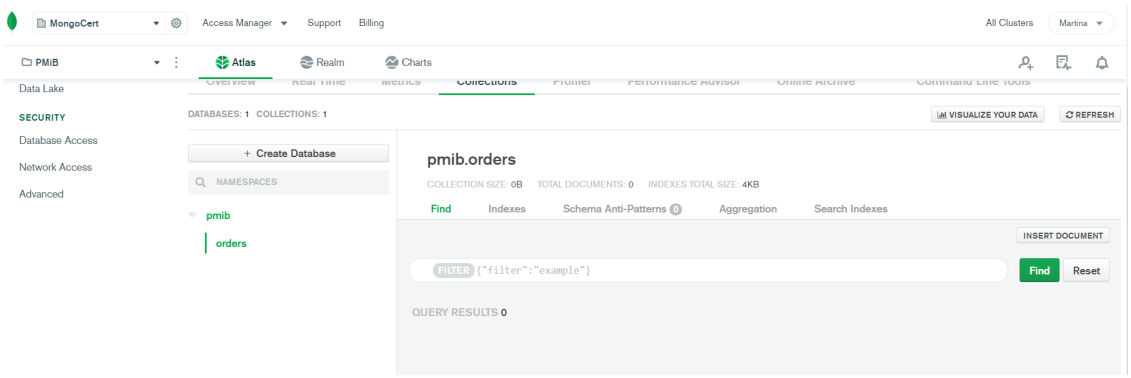


Slika 2: Uspešno ustvarjena gruča v MongoDB Atlasu.

V naslednjih korakih je potrebno omogočiti dostop do ustvarjenega MongoDB strežnika. Najprej je potrebno ustvariti uporabnika podatkovne baze (angl. database user) pod opcijo *Database access*. Ustvarili bomo novega uporabnika *pts-user*, ki bo do strežnika dostopal z vnosom gesla, in ima pravici branja in pisanja v

katero koli bazo. Dodatno je treba tudi pod opcijo *Network access* omogočiti dostop do strežnika z vseh IP naslovov.

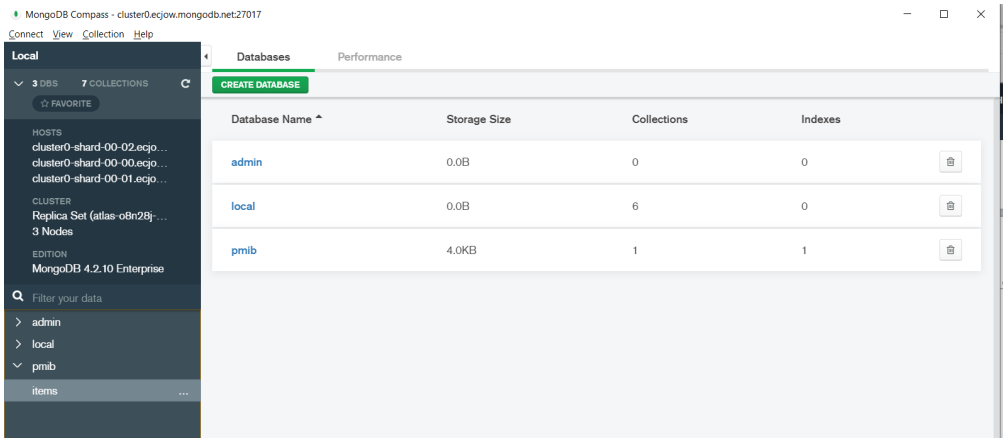
Zdaj bomo ustvarili MongoDB podatkovno bazo z zbirko naročil. Pod opcijo *Clusters* se prikaže seznam gruč. Če izberemo našo gručo, pod opcijo *View data* se prikaže seznam zbirk. Tukaj izberemo opcijo *Add My Own Data*, ustvarimo novo bazo *pts* in znotraj nje zbirko *orders*. Kot rezultat dobimo ustvarjeno zbirko znotraj baze, v katero lahko vnašamo nova naročila. Do zbirke dostopamo preko imenskega prostora (angl. namespace), ki vključuje imeni baze in zbirke združeni s piko, oz. *pts.orders*.



Slika 3: Ustvarjena zbirka za naročila znotraj MongoDB baze.

Za lažje delo z ustvarjeno bazo lahko uporabimo namizno orodje *MongoDB Compass*. Ko ga zaženemo, je potrebno ustvariti novo povezavo na strežnik. Znakovni niz za povezavo (angl. connection string) na ustvarjeno bazo v MongoDB Atlasu lahko vidimo znotraj samega Atlasa, če izberemo opcijo *Connect* pri ustvarjeni gruči. Izberemo želeni način povezave (MongoDB Compass v našem primeru) in se prikaže znakovni niz, ki ga kopiramo in vnesemo v MongoDB Compass. Opomba: analizirajte ta niz in določite, katere podatke potrebujete za povezavo!

Ko se uspešno ustvari povezava na strežnik, v MongoDB Compassu vidimo seznam baz in zbirk kot prikazano na sliki 4.



Slika 4: Uspešna povezava na MongoDB strežnik v Compassu.

MongoDB Compass lahko uporabimo za izvedbo CRUD operacij oz. izvajanje povpraševanj nad dokumenti v zbirki, vnos novih dokumentov, ter posodabljanje in brisanje dokumentov. Če želimo vnesti novo naročilo

v bazo, lahko znotraj imenskega prostora izberemo opcijo *Add data* > *Insert document*. Ko se odpre okno za vnos lastnosti novega dokumenta, vnesemo vrednosti za posamezne ključe, kjer en par ključ-vrednost predstavlja vrednost posamezne lastnosti dokumenta.

Več informacij glede oblikovanja MongoDB "sheme" za različne kardinalnosti povezav med entitetami je dostopnih na `http://learnmongodbthehardway.com/schema/schemabasics`. V našem primeru naročil imamo povezavo s kardinalnostjo 1:N (v določeni restavraciji se lahko naenkrat naroči več izdelkov in en izdelek se lahko naroči v le eni restavraciji). Poskusite oblikovati podatkovni model za shrambo naročil, nato vnesite vsaj en vnos v bazo preko Compassa. Ko vnesete prvi dokument, v Compass-u pod opcijo *Schema* so vidne lastnosti sheme dokumentov v zbirki oz. ime lastnosti, podatkovni tip in vrednosti.

Implementacija API-ja za vnos podatkov v MongoDB

V tem poglavju bomo simulirali spletno aplikacijo, ki bi jo uporabljal natakar v restavraciji za ročne vnose novih naročil v sistem oz. v MongoDB podatkovno bazo. Za beleženje novih naročil je potrebno zgraditi REST API na lokalnem strežniku (localhost), katerem bomo s pošiljanjem POST klicev posredovali podatke o naročilih. Nato se bodo podatki v prilagojeni obliki shranjevali kot novi dokumenti v ustvarjeno zbirko *items pts* MongoDB podatkovne baze.

API lahko implementirate v katerem koli programskem jeziku (npr. PHP, Java, C# itd.) in ga zaženete na *localhostu*. Pomembno je, da pri vnosu novega naročila posredujete naslednje podatke:

- številka naročila,
- datum in čas naročila,
- ime restavracije, kjer je gost naročil izdelke,
- ime izdelka,
- cena izdelka in
- količina naročenega izdelka.

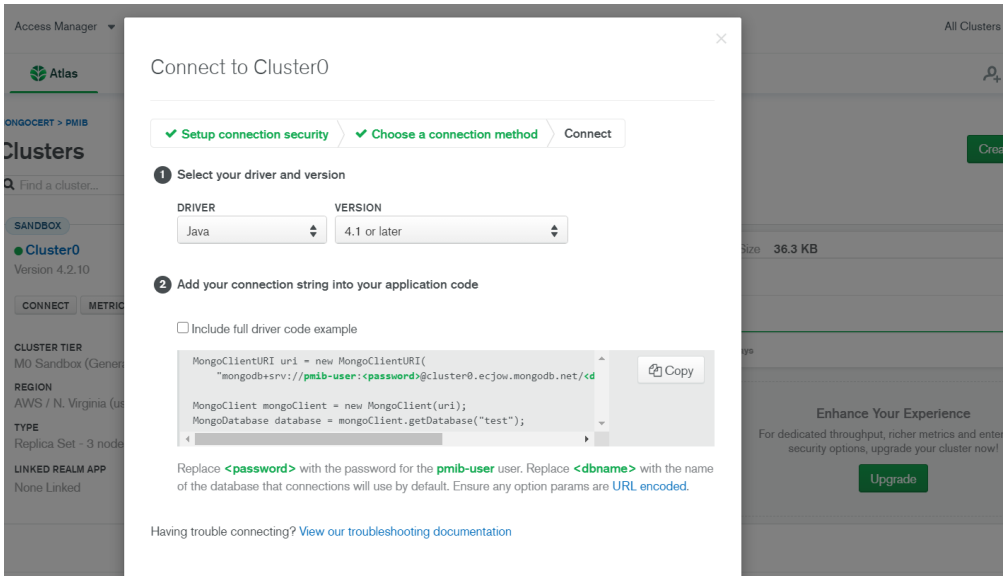
Opomba: Implementacija v izbranem programskem jeziku **NE SME** biti odvisna od vnaprej določene sheme, torej ni dovoljeno implementirati operacije z uporabo orodij, ki omogočajo izvedbo teh operacij na podlagi vnaprej določene sheme kot je npr. Mongoose.

Kot je razvidno iz strukture parametrov za API, osredotočili se bomo na implementacijo bolj preprostega scenarija oz. vsako naročilo vsebuje le en izdelek. Če želite, lahko tudi implementirate bolj kompleksen scenarij, ko se naenkrat lahko naroči več izdelkov.

V izbranem programskem jeziku implementirajte dostopno točko *orders* in najmanj metodi *POST* (za vnos novega naročila v bazo) ter *GET* (za pridobivanje vseh naročil iz baze). Za testiranje API-ja lahko uporabite orodje Postman, kjer lahko izvedete POST klic za vnos novega naročila v naslednji obliki:

```
POST http://localhost:4567/orders?number=16118&date=2020-10-12&restname=El Diablo&
itemname=Mint Sauce 1&itemprice=5.00&quantity=2
```

Glede na izbrani programski jezik za implementacijo, potrebno bo konfigurirati ustrezen gonilnik (angl. driver) za povezavo z ustvarjeno MongoDB bazo na MongoDB Atlasu. Znakovni niz za povezavo lahko ugotovite znotraj samega MongoDB Atlas grafičnega vmesnika in sicer tako da za ustvarjeno gručo izberete opcijo *Connect* > *Connect your application*, ter nato izberete, kateri gonilnik uporabljate (slika 5). Zgenerirano kodo potem lahko uporabite direktno za implementacijo (mogoče bo potrebno spremeniti ime baze in geslo za povezavo).



Slika 5: Primer ustvarjanja kode za povezavo na MongoDB podatkovno bazo za Java gonilnik.

Če nimate izkušenj z implementacijo spletnih API-jev, lahko uporabite Spark Java (`http://sparkjava.com/`) razvojni okvir, ki ne zahteva veliko konfiguracij pri nameščanju. Nasvet: ustvarite zaseben razred, ki bo predstavljal naročilo kot entiteto; za lažjo pretvorbo sprejetih parametrov glede na podatkovni model vaše MongoDB baze pa si lahko tudi ustvarite dodatni razred, ki predstavlja naročilo kot dokument v MongoDB bazi. Primer implementacije podobnega spletnega API-ja s pomočjo Spark okvirja lahko najdete na `https://github.com/eugenp/tutorials/tree/master/spark-java`, za delo z MongoDB bazo pa lahko uporabite primere dostopne na `https://mongodb.github.io/mongo-java-driver/3.4/driver/getting-started/quick-start/` ali na katerem koli spletnem viru.