



Onderzoeksrapport DEA

Patrick Roelofs

Studentnummer: 584025
Docent: Michel Portier
Klas: ITA-OOSE-A-f
Datum: 31/3/2021

Inhoudsopgave

Inhoudsopgave	1
Inleiding	2
Hoofd en deelvragen	2
Onderzoeksmethode	3
Library research method	3
Lab research method	3
Workshop research method	3
Onderzoek	4
Wat is MongoDB?	4
Wat is het verschil tussen NoSQL (Mongodb) en SQL (Mysql)?	5
Hoe kan MongoDB gebruikt worden in een Java Applicatie?	7
Wat moet er verandert worden aan de huidige implementatie van Spotitube om MongoDB werkend te krijgen in de Spotitube applicatie?	9
Conclusie	10
Samenvatting	11
Bronnen	12

Inleiding

Dit onderzoeksrapport is geschreven om uit te zoeken hoe een bestaande Java applicatie met MySQL database verbinding omgewisseld kan worden naar een MongoDB NoSQL database. De applicatie, Spotitube is een back-end Java applicatie dat voor de clientside een REST API neerzet en verbinding maakt met een MySQL database om gegevens van Spotitube gebruikers op te slaan.

Het verslag zal laten zien hoe je een MongoDB verbinding maakt met Java en of dit makkelijk, mogelijk en handig is. Uit ervaring is gebleken dat de MongoDB datastructuur persoonlijker zit dan een SQL structuur, ik zou dus persoonlijker liever met MongoDB werken dan MySQL. Dit document zal duidelijk maken of het binnen Java net zo fijn is.

Hoofd en deelvragen

In dit verslag wordt als hoofdvraag onderzocht “hoe Spotitube uitgebreid kan worden met een NoSQL MongoDB Database verbinding”. De hoofdvraag is opgedeeld in enkele deelvragen:

- Wat is MongoDB?
- Wat is het verschil tussen NoSQL en SQL?
- Hoe kan MongoDB gebruikt worden in een Java Applicatie?
- Wat moet er verandert worden aan de huidige implementatie van Spotitube om MongoDB werkend te krijgen in de Spotitube applicatie?

Ieder van deze deelvragen zal individueel behandeld worden met de onderzoeksmethodes van de ICT research methods. (HAN, z.d.)

Onderzoeksmethode

De ICT Research methods (HAN, z.d.) zijn opgedeeld in 6 hoofd-onderzoeksmethodes ieder van deze onderzoeksmethodes heeft een aantal onderzoeken die erop kunnen uitgevoerd worden. Op basis van de gemaakte deelvragen zijn deze onderverdeeld in de volgende onderzoeksmethodes en onderzoeken.

Library research method

Deelvragen

- Wat is MongoDB?
- Wat is het verschil tussen NoSQL en SQL?

Literature Study: Hier is onderzoek gedaan naar bestaande literatuur over MongoDB, er is op het internet gezocht naar bronnen over MongoDB en deze zijn bestudeerd om informatie te achterhalen dat voor dit onderzoek nuttig is. Daarnaast is er onderzocht naar de verschillen tussen MongoDB (een NoSQL database) en MySQL (een SQL database) deze informatie zal gebruikt worden om de volgende research methode toe te passen.

Lab research method

Deelvraag

- Wat is het verschil tussen NoSQL en SQL?

Data analytics: Met de onderzochte informatie is MongoDB en MySQL vergeleken door de opgenomen data te analyseren. Het resultaat van dit onderzoek geeft beeld wat het verschil is tussen NoSQL en SQL en daarbij kan de lezer ook een mening vormen welke hij persoonlijk beter vindt.

Workshop research method

Deelvraag

- Hoe kan MongoDB gebruikt worden in een Java Applicatie?
- Wat moet er verandert worden aan de huidige implementatie van Spotitube om MongoDB werkend te krijgen in de Spotitube applicatie?

Prototyping: Als eerste is er onderzocht hoe MongoDB toegepast kan worden in Java applicaties, wanneer dit duidelijk is zal de huidige Spotitube applicatie uitgebreid worden met een kleine prototype waarin MongoDB verwerkt is in de applicatie.

Onderzoek

Wat is MongoDB?

Er zijn veel databases zoals MySQL, misschien wel de bekendste. Deze databases staan bekend als relationele databases. Een relationeel database bestaat uit tabellen en ze zijn aan elkaar gerelateerd met behulp van Primary keys en Foreign keys. Relationele databases zijn niet goed in het opslaan van “Big Data”. (Verskil tussen NoSQL en MongoDB, z.d.)

MongoDB is een NoSQL database. MongoDB is de meest populaire NoSQL database met over 15 miljoen downloads (MongoDB, z.d.).

NoSQL databases kunnen wel met Big Data omgaan en zijn ook makkelijk te beheren, gegevens worden “geclusterd” in verschillende machines waardoor de kosten verlaagd kunnen worden. In MongoDB worden documenten opgeslagen in de vorm van JSON-achtige bestanden. Een web-developer werkt meestal met JSON bestanden om data te versturen naar pagina's, servers en databases dus wanneer je MongoDB gebruikt hoeft je deze data niet te converteren naar andere data. (Seguin, z.d.)

Veel development teams werken tegenwoordig op Agile gebaseerde software ontwikkeling daarin is mongoDB ook heel krachtig omdat het kan schalen en veranderen in grote hoeveelheden data (Meer snelheid en effectiviteit door Agile werken. Maar wat is Agile precies?, 2020). Daarnaast geeft het makkelijk leesbare JSON bestanden terug dat voor een groot development team veel tijd kan besparen in uitleg en documentatie. Ook zijn de code voor MongoDB queries simpel gemaakt zodat ze door iedereen gebruikt kan worden, hieronder is een van deze queries afgebeeld.

```
{
  "_id": "5cf0029caff5056591b0ce7d",
  "firstname": "Jane",
  "lastname": "Wu",
  "address": {
    "street": "1 Circle Rd",
    "city": "Los Angeles",
    "state": "CA",
    "zip": "90404"
  },
  "hobbies": ["surfing", "coding"]
}
```

```
db.users.find({ "address.zip" : "90404" })
```

Wat is het verschil tussen NoSQL (Mongodb) en SQL (Mysql)?

In dit hoofdstuk zal worden besproken wat de verschillen zijn tussen NoSQL en SQL, hier zal gekeken worden naar snelheid, veiligheid, en gebruiksvriendelijkheid dit zal door statistieken en stukken code verbeeld worden.

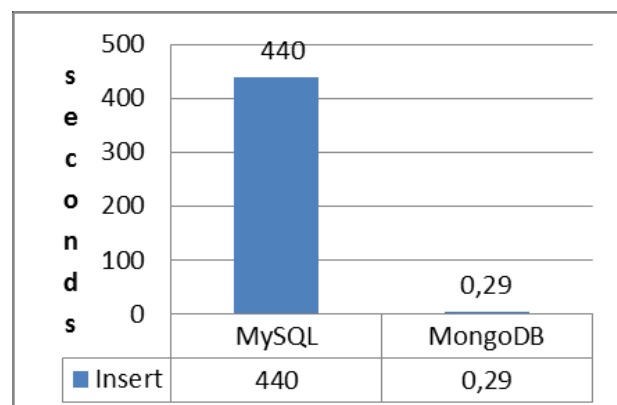
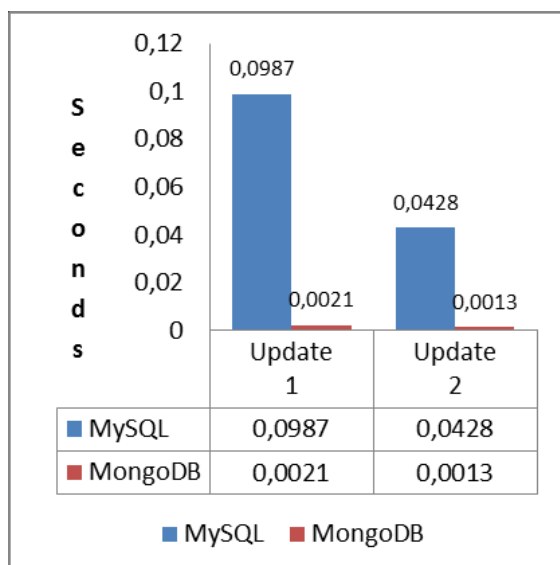
Hieronder (links) is een SQL Query afgebeeld, rechts is een MongoDB Query afgebeeld.

```
DELETE FROM account WHERE mail = "patrick@email.com";
```

```
db.account.remove({  
  "email": "patrick@email.com"  
});
```

Het handige aan SQL queries is dat het voor simpele queries “natuurlijke taal” gebruikt, “delete dit, van dit, waar dit” dat is makkelijk leesbaar voor iemand die niet veel weet van de taal en ook makkelijk te debuggen voor simpele typo's maar eenmaal als je bij complexe queries komt worden SQL queries moeilijk leesbaar. Daarentegen schrijf je MongoDB queries met de taal waar je op dat moment in schrijft, dat maakt MongoDB weer makkelijker omdat je toch de taal al begrijpt en je hoeft dan voor een query niet te wisselen van programmeertaal.

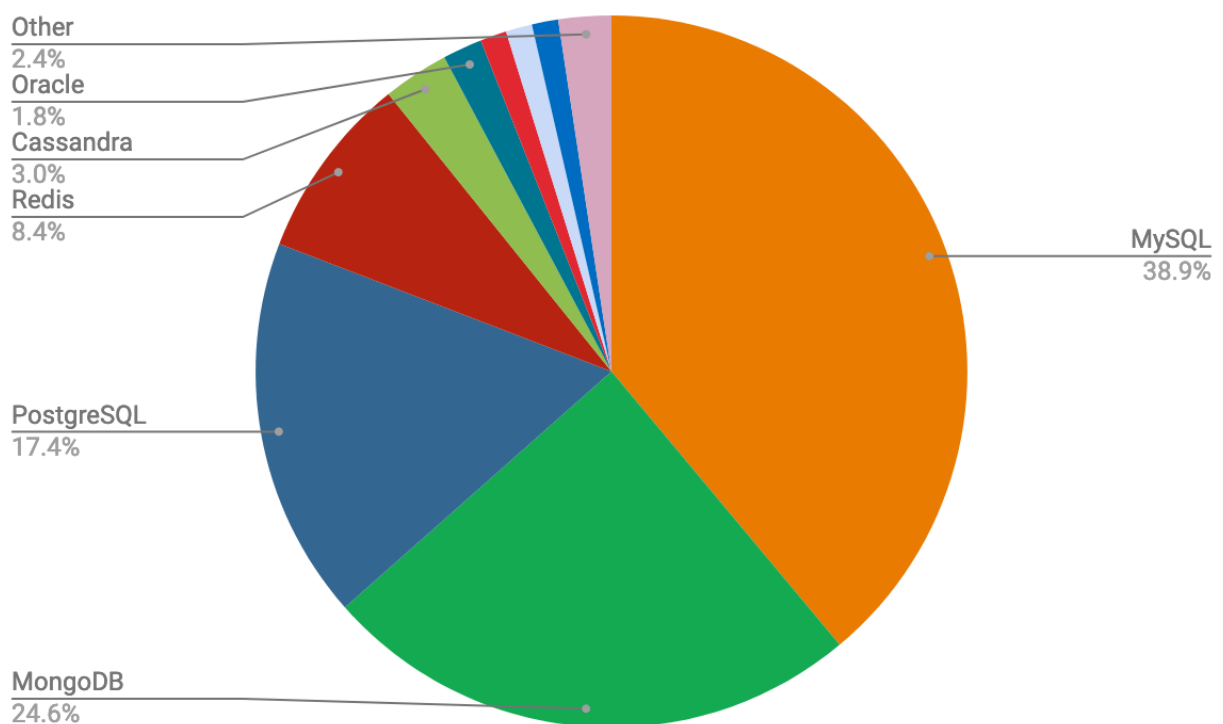
In snelheid is MongoDB vaak sneller aangezien het grote aantallen ongestructureerde data veel sneller kan omzetten dan MySQL, maar het is moeilijk om precies te weten welke sneller is aangezien het verschilt per taak, internetsnelheid en hoe je data sorteert. (Smallcombe, 2020). Wel kan je uit kleine tests achterhalen dat het verschil in snelheid ook erg verschilt per functie die uitgevoerd wordt. Hieronder zijn enkele statistieken afgebeeld waarin te zien is dat het erg verschilt per functie. (Györfödi et al., 2015)



MongoDB is wel beter in grote projecten aangezien het geen problemen heeft met “big data” en met Mongo DB's GridFS een functie om bestanden groter dan 16MB op te slaan kan je gemakkelijk video's, documenten, plaatjes en grote hoeveelheden research data opslaan zonder problemen. GridFS zal de bestanden splitsen in kleine onderdelen en ze dan beschikbaar maken voor gebruik wanneer nodig. SQL heeft tegen GridFS eigenlijk geen competitie. (MongoDB, z.d.-a)

Als we kijken naar een project zoals Spotitube zou de keuze tussen MongoDB en MySQL voor de gebruiker niet veel uitmaken de query's zijn vrij simpel en zullen allebei op bijna dezelfde snelheid uitgevoerd worden. De keuze zal zijn welke de developers fijner vinden.

Als we kijken naar welke database is populairder dan is MySQL nog steeds de populairste met MongoDB er kort naast. Dat betekent dat zelfs qua populariteit het niet veel uitmaakt welke er gebruikt wordt. (Györödi et al., 2015)



Hoe kan MongoDB gebruikt worden in een Java Applicatie?

Om erachter te komen hoe MongoDB gebruikt kan worden in een java applicatie is er een kleine test-applicatie gemaakt waarin verschillende api-routes gemaakt worden. De MongoDB applicatie (bijlage I) is een aparte programma dat op de computer geïnstalleerd moet worden voordat deze applicatie zal werken. De volgende beschrijving zal zijn hoe het project is opgebouwd en zal code voorbeelden geven, het project is ook in de bijlage (bijlage I) beschikbaar.

Om MongoDB te importeren in Maven kan deze dependency toegevoegd worden. Deze dependency zorgt ervoor dat MongoDB en Java met elkaar kunnen spreken.

```
<dependency>
  <groupId>org.mongodb</groupId>
  <artifactId>mongodb-driver-sync</artifactId>
  <version>4.2.2</version>
</dependency>
```

De volgende code zal een MongoClient, een verbinding met MongoDB maken. De MongoDB code zal de specifieke database vinden die opgevraagd wordt, in dit geval "mongojavatest". Omdat MongoDB met collecties werkt kan je met het laatste stukje code de specifieke Collectie ophalen die opgevraagd wordt dit kan bijvoorbeeld een Users collectie zijn.

```
MongoClient mongoClient = MongoClient.create();
MongoDatabase mongoDatabase = mongoClient.getDatabase("mongojavatest");
MongoCollection<Document> collection = mongoDatabase.getCollection("test");
```



```

// Find
Document document = collection.find().first();

// Create
Document document = new Document("test", "testdocument");
collection.insertOne(document);

// Delete
collection.deleteOne(eq("test", "testdocument"));

```

Hierboven zijn de 3 belangrijkste stukjes code van een database uitgeschreven, het vinden van een document, het maken van een document en het verwijderen van een document. Dit verschilt eigenlijk niet tot MongoDB voor Javascript of andere talen, dat maakt MongoDB erg krachtig.

Hieronder is een simpele Java API GET request geschreven, hierin kan je duidelijk zien dat een document vinden niet moeilijk is. Niet moeilijker dan met MySQL. Aangezien MongoDB met JSON werkt hoeft je dus de document niet nog eens specifiek om te zetten met Domains en DTO's, dat kost veel minder werk en tijd.

```

@GET
@Path("/")
@Produces(MediaType.APPLICATION_JSON)
public Response getDocument() {
    Document document = collection.find().first();

    return Response.status(Response.Status.OK).entity(document).build();
}

```

Wat moet er verandert worden aan de huidige implementatie van Spotitube om MongoDB werkend te krijgen in de Spotitube applicatie?

De huidige implementatie van de DAO-laag in Spotitube maakt het makkelijk om een MongoDB implementatie toe te voegen, er is al gebruik gemaakt van interfaces dus de programmeur kan gemakkelijk een nieuwe implementatie toevoegen.

Om ervoor te zorgen dat de MongoDB DAO ook gezien wordt door Java kan er in de bestand *beans.xml* een alternatieve class toegevoegd worden, daarnaast moet boven de DAO ook een *@Alternative* toegevoegd worden, dit zorgt ervoor dat de alternatieve implementatie gezien wordt door Java.

```
<alternatives>
  <class>com.spotitube.datasource.dao.mongo.UserDAOMongoDB</class>
</alternatives>
```

Hieronder is de implementatie van *verifyUser()* afgebeeld, de code is op eerste gezicht veel schoner dan in MySQL, dat komt omdat er geen queries geschreven hoeven te worden in een andere taal, de *MongoCursor* code zal itereren over de users collectie en zal true teruggeven als er eentje in staat, net als de *UserDAO* van MySQL doet. Dit stukje code geeft dus true terug wanneer de username in de database staat.

```
@Alternative
public class UserDAOMongoDB implements IUserDAO {
    public static final String DATABASE = "spotitube";
    public static final String USERS_COLLECTION = "users";

    MongoClient client = MongoClient.create();
    MongoDBDatabase db = client.getDatabase(DATABASE);
    MongoCollection<Document> users = db.getCollection(USERS_COLLECTION);

    @Override
    public boolean verifyUser(String username, String password) throws InternalServerErrorException {
        try {
            MongoCursor<Document> cursor = users.find(eq("user", username)).iterator();

            return cursor.hasNext();

        } catch (Exception e) {
            throw new InternalServerErrorException(e);
        }
    }
}
```

Conclusie

Het implementeren van MongoDB in Java kan zorgeloos gedaan worden. MongoDB geeft een moderne alternatief op andere databases en omdat MongoDB JSON bestanden terug geeft en binnen krijgt betekent dat je minder code hoeft te schrijven. Wat er binnen een MongoDB query komt komt er op dezelfde manier ook weer uit. Daar komt de kracht van MongoDB echt terug, het niet hoeven te werken met tabellen en complexe queries maar realistische makkelijk bruikbare JSON data.

Bestaande Java applicaties kunnen met enig werk omgezet worden naar MongoDB vooral als er gebruik gemaakt wordt van moderne programmeer stijlen zoals Interfaces en Dependency Injection. Wel zal de programmeur moeten opletten dat data in MongoDB anders wordt gebruikt dan een MySQL database, dit kan er mogelijk voor zorgen dat er onduidelijkheden in de datastructuur van een project komt. En is het misschien slimmer om een keuze te maken tussen een NoSQL en een SQL database en niet allebei tegelijk te draaien.

Voor eventuele snelheidsproblemen of andere afwegingen in “performance” hoeft een team dat kleine projecten maakt geen zorgen te maken, MongoDB en MySQL zijn beide snel genoeg voor bijna elk klein project. Mocht een team werken met Big Data, videos, plaatjes of andere grote documenten dan zal MongoDB een grote voorstander zijn tegenover MySQL.

Samenvatting

Dit onderzoeksrapport heeft duidelijk gemaakt dat het toevoegen van een MongoDB Database binnen een Java applicatie mogelijk is en ook heel handig kan zijn. MongoDB is een NoSQL database waarin data opgeslagen wordt als BSON (Binary JSON) hiernaast is een van dit soort documenten afgebeeld.

De reden dat JSON data zo handig is is omdat de meeste browser applicaties tussen client side en back end praten met elkaar via JSON data, als je dezelfde binnenkomende data kan opslaan in de database bespaar je veel conversie tijd en vooral ook regels code.

Praten met een MongoDB database is net zoals de JSON data makkelijker gemaakt, waarin je met SQL een nieuwe taal moet leren en schrijven werk je in MongoDB in de programmeertaal van dat moment.

In snelheid is MongoDB voor de kleine queries vaak sneller dan MySQL maar dat verschilt erg tussen hoe de data erin staat en hoe de server ingesteld is, realistisch gezien zou dit voor veel projecten niet uitmaken. Het gaat hier om milliseconden en niet een serieus groot getal. Binnen MongoDB kan je gebruik maken van grote-files met GridFS, hierin kan je plaatjes, video's en andere grote bestanden opslaan.

Je Java project omzetten naar een MongoDB database kan zonder grote moeite gedaan worden, binnen het Spotitube project zijn dit simpelweg de DAO laag overnieuw schrijven aangezien nagedacht is over Interfaces en Dependency Injection.

```
{
  "_id": "5cf0029caff5056591b0ce7d",
  "firstname": "Jane",
  "lastname": "Wu",
  "address": {
    "street": "1 Circle Rd",
    "city": "Los Angeles",
    "state": "CA",
    "zip": "90404"
  },
  "hobbies": ["surfing", "coding"]
}
```

```
db.users.find({ "address.zip" : "90404" })
```

Bronnen

Györödi, C., Györödi, R., Pecherle, G., & Olah, A. (2015, 11 juni). *A Comparative Study: MongoDB vs. MySQL*. ResearchGate.
https://www.researchgate.net/publication/278302676_A_Comparative_Study_MongoDB_vs_MySQL

HAN. (z.d.). *ICT research methods*. ICT Research Methods. Geraadpleegd op 29 maart 2021, van http://ictresearchmethods.nl/Main_Page

Meer snelheid en effectiviteit door Agile werken. Maar wat is Agile precies? (2020, 1 april). Scrum Company. <https://www.scrumcompany.nl/wat-is-agile/>

MongoDB. (z.d.-a). *GridFS — MongoDB Manual*. MongoDB GridFS. Geraadpleegd op 31 maart 2021, van <https://docs.mongodb.com/manual/core/gridfs/>

MongoDB. (z.d.-b). *The most popular database for modern apps*. Geraadpleegd op 29 maart 2021, van <https://www.mongodb.com/>

Seguin, K. (z.d.). *The Little MongoDB Book*. The Little MongoDB Book. Geraadpleegd op 31 maart 2021, van <https://www.openmymind.net/mongodb.pdf>

Smallcombe, M. (2020, 20 oktober). *MongoDB vs. MySQL: Detailed Comparison of Performance and Speed*. Xplenty. <https://www.xplenty.com/blog/mongodb-vs-mysql/>

Verschil tussen NoSQL en MongoDB. (z.d.). Verschil tussen NoSQL en MongoDB. Geraadpleegd op 29 maart 2021, van <https://nl.differencevs.com/6845251-difference-between-nosql-and-mongodb>