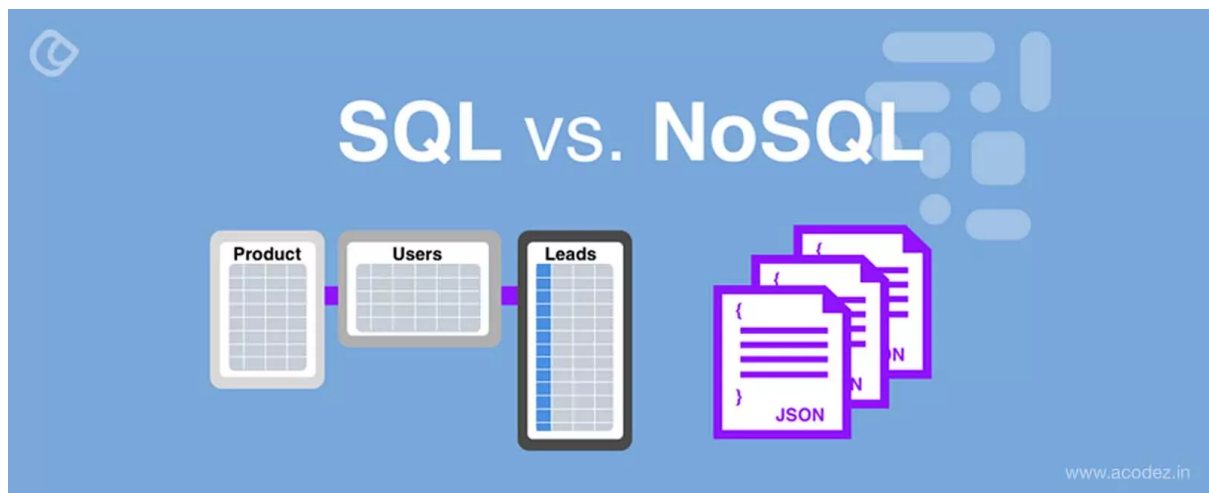


OOSE - DEA

Onderzoeksopdracht Enterprise Applications

# Van SQL naar NoSQL Database



<b>Naam</b>	El Kaddouri, I.
<b>Studentnummer</b>	658025
<b>Datum</b>	02-04-2021
<b>Course</b>	Develop Enterprise Application

# Inhoudsopgave

<b>1 Inleiding</b>	<b>3</b>
Leeswijzer	4
<b>2 Onderzoek</b>	<b>5</b>
2.1 Wat is een NoSQL database?	5
2.2 Wat zijn de verschillen en overeenkomsten tussen een SQL en NoSQL?	5
2.3 Hoe werkt MongoDB?	7
2.3.1 Gegevensmodel	7
2.3.2 MongoDB database	7
2.3.4 Een Applicatie verbinden met MongoDB	8
2.3.5 Het ophalen en aanpassen van data	8
2.4 Hoe implementeer je MongoDB in Spotitube?	8
2.4.1 Omzetten database structuur	9
2.4.2 Opzetten database	9
2.4.3 MongoDB Driver installeren	9
2.4.4 Spotitube aanpassen	9
<b>3 Resultaten</b>	<b>12</b>
<b>4 Conclusie</b>	<b>15</b>
<b>5 Literatuurlijst</b>	<b>17</b>
<b>6 Bijlagen</b>	<b>19</b>

# 1 Inleiding

Het bedrijf achter Spotitube heeft de opdracht gegeven om Spotitube gebruik te laten maken van een niet relationele database (NoSQL), specifiek: MongoDB. In dit onderzoeksrapport wordt er onderzoek gedaan naar hoe de huidige relationele database, MySQL, vervangen kan worden door een NoSQL database in de Spotitube applicatie, de voor en nadelen van deze migratie zullen uitvoerig worden besproken.

Middels dit onderzoeksrapport zal de opdrachtgever inzicht krijgen van de resultaten en gemaakte keuzes. Het onderzoeksrapport is opgesteld door de developer van de applicatie Spotitube.

Aan het eind van dit onderzoek zal duidelijk zijn hoe MongoDB geïmplementeerd moet worden in Spotitube en wat dit allemaal met zich mee brengt. Om hierachter te komen zullen er een aantal onderzoeksvragen beantwoord moeten worden:

De hoofdvraag voor dit onderzoek luidt als volgt:

- **Hoe implementeer ik het gebruik MongoDB in Spotitube.**

Om een correct antwoord op de hoofdvraag te kunnen geven is het noodzakelijk om eerst de bijbehorende deelvragen te beantwoorden. Om elk deelvraag te kunnen beantwoorden zal er gebruik worden gemaakt van verschillende ICT research methoden<sup>1</sup>.

## 1. Wat is een NoSQL Database?

Deze vraag zal behandeld worden in Hoofdstuk 2.1. Deze vraag wordt deels beantwoord door het gebruik van de onderzoeksmethode *literature study*, om erachter te komen wat een NoSQL/MongoDB eigenlijk is.

## 2. Wat zijn de verschillen en overeenkomsten tussen een SQL en NoSQL?

Het antwoord op deze vraag wordt behandeld in hoofdstuk 2.2. Gelukkig is er veel bekend over NoSQL databases. Hierdoor hoeft er zelf geen lab onderzoek gedaan te worden. Bij het beantwoorden van deze vraag is het onderzoeksmethode *literature study* gebruikt om te kijken wat het verschil precies is.

## 3. Hoe werkt MongoDB?

Deze vraag wordt behandeld in Hoofdstuk 2.3. Om achter de werking van MongoDB te komen zal er weer gebruik worden gemaakt van *Literature Study*. Tevens zal er gebruik worden gemaakt van een werkplaats onderzoeksmethode *Prototyping*.

## 4. Hoe implementeer je MongoDB in Java?

Om erachter te komen hoe MongoDB te gebruiken is in Java en wat ervoor nodig is om dit te realiseren zal er gebruik worden gemaakt van de onderzoeksmethode *Literature Study* en *Prototyping*.

---

<sup>1</sup> <https://ictresearchmethods.nl/Methods>

## Leeswijzer

In het eerste hoofdstuk van dit onderzoeksrapport wordt de doelstelling van het onderzoek en de daarbij behorende vragen besproken. Ook worden de gebruikte onderzoeksmethoden per deelvraag genoemd. In hoofdstuk 2 van dit verslag zijn alle onderzoeksvragen uitvoerig behandelt en besproken waarna in hoofdstuk 3 de resultaten van het onderzoek wordt behandelt. De conclusie van het onderzoek wordt in hoofdstuk 5 besproken door middel van kort samengevatte antwoorden op elke deelvraag. De laatste twee hoofdstukken 5 en 6 zijn respectievelijk een literatuurlijst van de gebruikte bronnen en lijst van bijlagen.

## 2 Onderzoek

### 2.1 Wat is een NoSQL database?

De term “NoSQL” doet het type database niet ten goede, het suggereert dat het gaat om een database waarbij SQL niet als taal wordt gebruikt. Dit is echter niet het geval, de term “NoRelational” zou een meer passend zijn geweest. Zoals het laatstgenoemde term al zegt is een NoSQL database een database waarbij, standaard, geen rekening wordt gehouden met relaties tussen data, zoals bij een SQL database wel het geval is. De data in een NoSQL database hoeft daarom bij voorhand niet te worden voorzien van een structuur zoals bij een SQL database waarbij tabellen worden gebruikt.

Doordat er geen rekening wordt gehouden met datastructuren krijgt een NoSQL database de mogelijkheid om extreem grote hoeveelheden data op te slaan. Tevens heeft een NoSQL database de mogelijkheid om volatile data (snel veranderende data) te verwerken. Deze twee eigenschappen maken een NoSQL database erg schaalbaar. Meer voor en nadelen wordt besproken in de volgende paragraaf.

### 2.2 Wat zijn de verschillen en overeenkomsten tussen een SQL en NoSQL?

Er zijn talloze voor- en nadelen voor en nadelen van een NoSQL database. Echter is dat niet belangrijk in het bepalen van de juiste database. Belangrijker is het analyseren van het verschil tussen de twee typen databases om vervolgens daaruit een keuze te maken die past bij de applicatie.

	Relationele Database	Niet-relationale databases
Toepassingen	<ul style="list-style-type: none"><li>• Het verwerken van logische gerelateerde data waarvan de structuur vooraf is opgesteld</li><li>• De structuur moet synchroon lopen met de applicaties die de database gebruiken.</li><li>• Applicaties waarbij er verschillende samengevoegde data moet worden opgehaald.</li></ul>	<ul style="list-style-type: none"><li>• Het verwerken van grote hoeveelheden niet gerelateerde data.</li><li>• Verwerking van volatile (snel veranderende) data.</li><li>• Applicaties met een extreem grote userbase. (wereldwijde gebruikers).</li><li>• Applicaties waarbij beschikbaarheid en prestatie belangrijker zijn dan de consistentie van data.</li></ul>
Voorbeeld Applicaties	<ul style="list-style-type: none"><li>• Fintech, boekhoud- en banksystemen</li><li>• Betaalsystemen (Adyen, Mollie, etc.)</li></ul>	<ul style="list-style-type: none"><li>• Verwerken van Big data</li><li>• Analyses in realtime</li><li>• Applicaties met grote hoeveelheid gebruikers</li></ul>

	<ul style="list-style-type: none"> <li>• Voorraadbeheersystemen</li> </ul>	<ul style="list-style-type: none"> <li>• Applicaties met inconsistente data</li> </ul>
<b>Schaalbaarheid</b>	<ul style="list-style-type: none"> <li>• Data verspreiden over verschillende servers (sharding).</li> </ul>	<ul style="list-style-type: none"> <li>• Het verhogen van het server vermogen.</li> </ul>
<b>Gegevensmodel</b>	<ul style="list-style-type: none"> <li>• tabellen, key/value pairs</li> <li>• Alle data wordt opgeslagen aan de hand van het datatype</li> </ul>	<ul style="list-style-type: none"> <li>• 3 verschillende gegevensmodellen: key/value pairs, (JSON) documenten en grafiek- en kolommendatabase.</li> <li>• Data wordt opgeslagen afhankelijk van bovenstaand databasetypen.</li> </ul>
<b>Prestatie</b>	<ul style="list-style-type: none"> <li>• De prestaties zijn grotendeels afhankelijk van de optimalisatie van zoekquery's, indexen en tabelstructuren.</li> <li>• Het storagestelsel speelt ook een rol in de structuur</li> </ul>	<ul style="list-style-type: none"> <li>• Hardwarecluster afhankelijk</li> <li>• Netwerksnelheid</li> </ul>
<b>APIs</b>	<ul style="list-style-type: none"> <li>• Database mutaties worden verricht dmv een gestructureerde query taal: SQL</li> </ul>	<ul style="list-style-type: none"> <li>• Databases mutaties worden verricht dmv een API. Dit maakt het erg makkelijk voor een developer om data op te slaan/aan te roepen.</li> </ul>

*Tabel 1: Verschillen tussen een SQL en NoSQL database gecategoriseerd per onderwerp.*

## 2.3 Hoe werkt MongoDB?

MongoDB is een implementatie van een NoSQL database die wordt gebruikt door grote tech bedrijven zoals Facebook, Google, EA Sports, Adobe, Uber, Cisco, etc. Dat deze bedrijven gebruik maken van MongoDB heeft alleen maar voordelen: er is meer dan voldoende documentatie op het internet beschikbaar om de werking ervan te leren.

Voordat er verder wordt gelezen is het noodzakelijk om de terminologie te bespreken. In MongoDB zijn er geen sprake van tabellen maar collecties. Verder wordt, wat bij een sql database een record/row wordt genoemd, in MongoDB een Document genoemd.

### 2.3.1 Gegevensmodel

Het gegevensmodel dat MongoDB gebruikt zijn documenten met key, value pairs. Deze documenten zijn vergelijkbaar met JSON objecten. het type van het document lijkt dan ook sterk op het vorig genoemde object: BSON.

In een BSON document kun je een tal van datatypen<sup>2</sup> gebruiken: ints, strings, arrays, etc. Ook is het mogelijk om attributen van andere documenten te integreren, om zo toch een relatie te creëren.

```
var mydoc = {  
  _id: ObjectId("5099803df3f4948bd2f98391"),  
  name: { first: "Alan", last: "Turing" },  
  birth: new Date('Jun 23, 1912'),  
  death: new Date('Jun 07, 1954'),  
  contribs: [ "Turing machine", "Turing test", "Turingery" ],  
  views : NumberLong(1250000)  
}
```

*Figuur 1: Voorbeeld BSON object*

### 2.3.2 MongoDB database

Er zijn twee manieren om MongoDB te gebruiken: een lokale MongoDB installatie<sup>3</sup> of een MongoDB instantie op een Cloud platform naar keuze. In dit onderzoek zal alleen het Cloud platform beperkt worden tot MongoDB zelf: MongoDB Atlas<sup>4</sup>.

Het installeren van een lokale MongoDB instantie is eenvoudig en voor de hand liggend. Er zijn duidelijke installatie instructies<sup>3</sup> en ook is er genoeg informatie op het internet. De community support is erg hoog zo zijn er 12.8k leden in de MongoDB Reddit<sup>5</sup> en StackOverflow<sup>6</sup> heeft maar liefst 140.000 beantwoorde vragen.

---

<sup>2</sup> <https://docs.mongodb.com/manual/reference/bson-types/>

<sup>3</sup> <https://docs.mongodb.com/manual/installation/#std-label-tutorial-installation>

<sup>4</sup> <https://docs.atlas.mongodb.com/getting-started/>

<sup>5</sup> <https://www.reddit.com/r/mongodb/>

<sup>6</sup> <https://stackoverflow.com/questions/tagged/mongodb>

Echter om het de ontwikkelaar toch makkelijker te maken is het nog eenvoudiger om een account te registreren op [atlas.mongodb.com](https://atlas.mongodb.com). Waarna een gebruiker na een aantal kliks al een database heeft opgezeten die klaar is om te gebruiken. In het verdere onderzoek zal hiervan gebruik gemaakt worden, omdat de keuze tussen lokaal of cloud oplossing een onderzoek op zich is.

### 2.3.4 Een Applicatie verbinden met MongoDB

Er zijn een aantal manieren beschikbaar om met MongoDB te verbinden. Zo kan er, net zoals bij een SQL database, gebruik worden gemaakt van de klassieke “Shell”<sup>7</sup> omgeving.

Om een applicatie te verbinden met MongoDB, zijn er drivers<sup>8</sup> beschikbaar, specifiek dient er voor de Spotitube applicatie de Java drivers gebruikt te worden. Deze driver is vergelijkbaar met de reeds gebruikte JDBC driver. De implementatie instructies<sup>9</sup> zijn eenvoudig en duidelijk uitgelegd.

### 2.3.5 Het ophalen en aanpassen van data

In tegenstelling tot SQL databases gebruikt MongoDB standaard geen Query language. Dit betekent dat het niet nodig is om een extra taal aan te leren om deze database te gebruiken. MongoDB beschikt over een groot assortiment aan methoden<sup>10</sup> waarmee het ophalen en muteren van data vrij eenvoudig is voor ontwikkelaars.

Zo zijn er functies zoals `find()`, `insert()`, `remove()`, etc. Alle methoden spreken voor zich en zijn eenvoudig te begrijpen voor elke software ontwikkelaar.

Een voorbeeld van een MongoDB snippet is hieronder weergegeven.

```
MongoCursor<Document> cursor = collection.find().iterator();
try {
    while (cursor.hasNext()) {
        System.out.println(cursor.next().toJson());
    }
} finally {
    cursor.close();
}
```

*Code Snippet 1 : MongoDB alle records uit een collectie te halen.*

---

<sup>7</sup> <https://www.mongodb.com/products/shell>

<sup>8</sup> <https://docs.mongodb.com/drivers/>

<sup>9</sup> <https://docs.mongodb.com/drivers/java/>

<sup>10</sup> <https://docs.mongodb.com/manual/reference/method/js-collection/>



## 2.4 Hoe implementeer je MongoDB in Spotitube?

Om MongoDB in java te implementeren is het handig om alle relevante informatie, i.e. wat voor Spotitube van toepassing is, die beschikbaar zijn in de documentatie van MongoDB op een rijtje te zetten.

### 2.4.1 Omzetten database structuur

Als eerst dient de huidige database structuur omgezet te worden naar een MongoDB database structuur. Echter hebben we eerder geleerd dat een MongoDB database geen voorgedefinieerd structuur nodig heeft. Wél kunnen we de huidige MySQL tabellen omzetten naar documenten. Zo kunnen we de tabellen user, token, track, playlist omzetten naar documenten.

De tabel playlist\_track komt te vervallen aangezien dit een koppeltabel is en niet meer nodig is in een niet relationele database. In plaats van deze koppeltabel kunnen wel het track document integreren in het playlist document en bij het ophalen van de data samenvoegen.

### 2.4.2 Opzetten database

Verder is het nodig om een database op te zetten en te verbinden. Hiervoor maken we een account op MongoDB Atlas<sup>4</sup> en maken we een gratis Cluster aan voor de Spotitube applicatie. Na deze stappen is het alleen noodzakelijk om de connectie string op te slaan, zodat we de MongoDB Java<sup>9</sup> driver kunnen verbinden.

### 2.4.3 MongoDB Driver installeren

Om spotitube te verbinden met de nieuwe MongoDB database dient er een driver te worden geïnstalleerd<sup>11</sup>. Om dit te doen dient er als eerst een Maven dependency te worden toegevoegd aan de `pom.xml`:

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

*Code snippet 2: Maven Dependency van MongoDB driver.*

### 2.4.4 Spotitube aanpassen

In de documentatie van MongoDB zijn er verschillende code snippets<sup>12</sup> beschikbaar voor het toevoegen, lezen, updaten en verwijderen (CRUD) van gegevens. Deze kunnen worden gebruikt voor het aanpassen van de DAO's in de Spotitube applicatie.

Bij de initiële versie van Spotitube is rekening gehouden met het veranderen van de database. Dit betekent dat er voor elke DAO een Interface beschikbaar is die

<sup>11</sup> <https://mongodb.github.io/mongo-java-driver/4.2/driver/getting-started/installation/>

<sup>12</sup> <https://mongodb.github.io/mongo-java-driver/4.2/driver/getting-started/quick-start/>

geïmplementeerd dient te worden met behulp van de MongoDB code snippets. Deze nieuwe MongoDB DAO's kunnen dan gebruikt worden door middel van Dependency injection, verderop in dit hoofdstuk staat hoe dit mogelijk is.

Om een database verbinding te kunnen leggen in elke DAO dient het volgende stukje code<sup>12</sup> te worden gebruikt:

```
ConnectionString connString = new
ConnectionString("mongodb+srv://<USER_HIER>:<PASSWORD_HIER>@cluster0.sbgln.mongodb.net/test?w=majority");
MongoClientSettings settings = MongoClientSettings.builder()
    .applyConnectionString(connString)
    .retryWrites(true)
    .build();

MongoClient mongoClient = MongoClient.create(settings);
MongoDatabase db = mongoClient.getDatabase("databasename")
```

*Code Snippet 3: Voorbeeld MongoDB connectie aanmaken.*

Tot slot moet de annotatie `@Alternative` bovenaan de klasse worden gezet en dient het `beans.xml` aan aangepast te worden door de volgende regel toe te voegen tussen de `beans` tags:

```
<alternatives>
  <class>han.oose.dea.PACKAGE.KLASSENAAM</class>
</alternatives>
```

*Code Snippet 4: Alternative regel toevoegen aan beans.xml*

### 3 Resultaten

De bevindingen die zijn opgedaan in [Hoofdstuk 2](#) zijn (deels) toegepast in de Spotitube applicatie. Specifiek is de `ITrackDAO` geïmplementeerd om te werken met MongoDB.

Als eerst is het `pom.xml` aangepast door het toevoegen van de MongoDB dependency in [Hoofdstuk 2.4.3](#):

#### `pom.xml`

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

*Code Snippet 5: Stukje code van de bijlage. Zie Bijlage I.*

Vervolgens maken we een nieuwe `TrackDAO`: `TrackMongoDBDAO` volgens de instructies die opgesteld zijn in [Hoofdstuk 2.4.4](#).

#### `TrackMongoDBDAO.java`

```
package han.oose.dea.persistence;
package han.oose.dea.persistence;

import com.mongodb.ConnectionString;
import com.mongodb.MongoClientSettings;
import com.mongodb.client.*;
import han.oose.dea.domain.Track;
import han.oose.dea.exceptions.PersistenceException;
import org.bson.Document;

import java.util.ArrayList;
import java.util.List;

import static com.mongodb.client.model.Filters.eq;
import static com.mongodb.client.model.Updates.set;

@Alternative
public class TrackMongoDBDAO implements ITrackDAO {
    private final String database = "spotitube";
    private final String collection = "track";

    @Override
    public List<Track> findAll() throws PersistenceException {
        // Connect to database
        MongoDBDatabase db = connect();

        // Get collection
        MongoCollection<Document> trackCollection = db.getCollection(collection);
```

```

// Get all tracks
List<Track> tracks = new ArrayList<>();

MongoCursor<Document> cursor = trackCollection.find().iterator();
try {
    while (cursor.hasNext()) {
        var currentTrack = cursor.next();
        // It is better to use a Data Mapper here.
        tracks.add(new Track(currentTrack.getInteger("_id"),
            currentTrack.getString("title"),
            currentTrack.getString("performer"),
            currentTrack.getInteger("duration"),
            currentTrack.getString("album"),
            currentTrack.getInteger("playCount"),
            currentTrack.getString("publicationDate"),
            currentTrack.getString("description"),
            currentTrack.getBoolean("offlineAvailable")

        ));
    }
} finally {
    cursor.close();
}

return tracks;
}

@Override
public void setTrackAvailability(int trackId, boolean isOfflineAvailable) throws
PersistenceException {
    // Connect to database
    MongoDBDatabase db = connect();

    // Get collection
    MongoCollection<Document> trackCollection = db.getCollection(collection);

    trackCollection.updateOne(eq("id", trackId), set("offlineAvailable",
isOfflineAvailable));
}

private MongoDBDatabase connect() {
    ConnectionString connString = new
ConnectionString("mongodb+srv://<USER_HIER>:<PASSWORD_HIER>@cluster0.sbgln.mongodb.net/
test?w=majority");
    MongoClientSettings settings = MongoClientSettings.builder()
        .applyConnectionString(connString)
        .retryWrites(true)
        .build();

    MongoClient mongoClient = MongoClients.create(settings);
    return mongoClient.getDatabase(database);
}
}

```

*Code snippet 6: Volledig geïmplementeerde MongoDB TrackDAO. Zie Bijlage I.*

Door de onderzoeksmethode Prototyping toe te passen is duidelijk geworden dat MongoDB beschikt over dezelfde functionaliteiten die het eerder gebruikte MySQL ook heeft. Echter, zoals in het voorbeeld te zien is, is het veel eenvoudiger om met MongoDB te werken dan

met SQL, omdat er geen nieuwe taal geleerd hoeft te worden en alle gebruikte methoden voor de hand liggend en voor iedereen leesbaar zijn.

## 4 Conclusie

MongoDB heeft een aantal voordelen tegenover een SQL database. Er kan gemakkelijk grote hoeveelheden data en volatile data worden opgeslagen. Tevens is MongoDB veel schaalbaarder dan een SQL database. Verder zijn alle functionaliteit van een SQL database beschikbaar in MongoDB. Een nadeel kan zijn dat MongoDB relatief inconsistent is terwijl een SQL database altijd consistent is.

MongoDB kan makkelijk worden gebruikt in een Java applicatie doordat er out-of-the-box drivers beschikbaar zijn die eenvoudig te installeren zijn.. Tevens is de documentatie zodanig geschreven dat het voor iedereen leesbaar is. Het is, integendeel tot een SQL database, erg eenvoudig om data op te halen, toe te voegen , te mutieren en te verwijderen. Hiervan zijn voorbeelden beschikbaar in [Hoofdstuk 2.3.5](#). Dit is later ook aangetoond in [Hoofdstuk 3](#).

Als een bestaande Java applicatie preventief rekening heeft gehouden met het veranderen van database is het zeer eenvoudig om te veranderen. Naast het opnieuw implementeren van alle DAO's hoeven alleen nog maar enkele lijnen code te worden toegevoegd om alles werkend te maken.

Al met al is MongoDB een zeer goed alternatief voor de applicatie Spotitube. Het is eenvoudig te implementeren en veel eenvoudiger in gebruik dan een traditionele SQL database.

## 5 Aanbevelingen

Ik adviseer om de opdrachtgever van Spotitube om MongoDB te implementeren, omdat dit vele voordelen met zich mee brengt. Het grootste voorbeeld dat van belang is bij Spotitube is de schaalbaarheid. Het plan is om Spotitube de nummer 1 streamingdienst van de wereld te maken. Dit in het achterhoofd hebbende is het belangrijk om vanaf het begin van de ontwikkeling van spotitube gebruik te maken van schaalbare technologieën. Helaas is SQL daar geen van.



## 5 Literatuurlijst

*BSON Types — MongoDB Manual.* (z.d.). Docs.Google.Com. Geraadpleegd op 2 april 2021, van <https://docs.mongodb.com/manual/reference/bson-types/>

*Collection Methods — MongoDB Manual.* (z.d.). MongoDB Manual. Geraadpleegd op 2 april 2021, van <https://docs.mongodb.com/manual/reference/method/js-collection/>

*Documents — MongoDB Manual.* (z.d.). Docs.Mongodb.Com. Geraadpleegd op 31 maart 2021, van <https://docs.mongodb.com/manual/core/document/>

*Get Started with Atlas — MongoDB Atlas.* (z.d.). Docs.Mongodb.Com. Geraadpleegd op 1 april 2021, van <https://docs.atlas.mongodb.com/getting-started/>

*Install MongoDB — MongoDB Manual.* (z.d.). Docs.Mongodb.Com. Geraadpleegd op 1 april 2021, van <https://docs.mongodb.com/manual/installation/#std-label-tutorial-installation>

*Java MongoDB Driver — MongoDB Drivers.* (z.d.). MongoDB. Geraadpleegd op 1 april 2021, van <https://docs.mongodb.com/drivers/java/>

*MongoDB Documentation.* (z.d.). MongoDB. Geraadpleegd op 1 april 2021, van <https://docs.mongodb.com/drivers/>

*NoSQL Database - What is NoSQL?* (z.d.). Microsoft Azure. Geraadpleegd op 30 maart 2021, van <https://azure.microsoft.com/en-us/overview/nosql-database/>

Plesky, E. (2019, 21 oktober). *NoSQL vs SQL: Examining the Differences and Deciding Which to Choose*. Plesk. <https://www.plesk.com/blog/various/nosql-vs-sql-examining-the-differences-and-deciding-which-to-choose/>

Saive, R. (2020, 26 oktober). *What is MongoDB? How Does MongoDB Work?* Tecmint.Com. <https://www.tecmint.com/what-is-mongodb-how-does-mongodb-work/>

*Shell.* (z.d.). MongoDB. Geraadpleegd op 1 april 2021, van <https://www.mongodb.com/products/shell>

*SQL vs. NoSQL Databases: What's the Difference?* (2020, 18 juni). Ibm.Com. <https://www.ibm.com/cloud/blog/sql-vs-nosql>

Team, M. D. (2021a, maart 2). *Installation.* MongoDB. <https://mongodb.github.io/mongo-java-driver/4.2/driver/getting-started/installation/>

Team, M. D. (2021b, maart 2). *Quick Start*. MongoDB.

<https://mongodb.github.io/mongo-java-driver/4.2/driver/getting-started/quick-start/>

## 6 Bijlagen

1. spotitube.zip