|  |
| --- |
| Onderzoeksverslag MongoDB |
| Versienummer: 1.0 |

|  |
| --- |
| Sem Hones, Lucas van Steveninck  Studentnummers: 659955, 675978  Naam docent: Meron Brouwer  Klas: OOSE-A  Course/semestergegevens: OOSE  Datum: 10-20-2022 |

Inhoud

[Samenvatting 4](#_Toc117169600)

[1. Introductie 5](#_Toc117169601)

[1.1. Vraagstelling 5](#_Toc117169602)

[2. Methoden en materialen 5](#_Toc117169603)

[2.1. Wat is het verschil tussen de query’s van MongoDB en die van een relationele database? 5](#_Toc117169604)

[2.2. Hoe werken relaties in MongoDB in vergelijking met relationele databases? 5](#_Toc117169605)

[2.3. Zijn er veiligheidsrisico's van een MongoDB database en zo ja wat zijn die? 6](#_Toc117169606)

[2.4. Zijn er verschillen in de manier waarop data opgeslagen wordt in een MongoDB database en de manier waarop dit gebeurt in een MySQL database? 6](#_Toc117169607)

[2.5. Hoe ziet MongoDB eruit in een proef applicatie? 6](#_Toc117169608)

[2.6. Hoe ziet MongoDB eruit in de spotitube applicatie? 6](#_Toc117169609)

[3. Resultaten 7](#_Toc117169610)

[3.1. Wat is het verschil tussen de query’s van MongoDB en die van een relationele database? 7](#_Toc117169611)

[Select 7](#_Toc117169612)

[Insert 7](#_Toc117169613)

[Update 8](#_Toc117169614)

[Delete 9](#_Toc117169615)

[3.2. Hoe werken relaties in MongoDB in vergelijking met relationele databases? 9](#_Toc117169616)

[3.3. Zijn er veiligheidsrisico's van een MongoDB database en zo ja wat zijn die? 9](#_Toc117169617)

[3.4. Zijn er verschillen in de manier waarop data opgeslagen wordt in een MongoDB database en de manier waarop dit gebeurt in een MySQL database? 10](#_Toc117169618)

[3.5. Hoe ziet MongoDB eruit in een proef applicatie? 10](#_Toc117169619)

[Haal alle records op 10](#_Toc117169620)

[Voeg een record toe 11](#_Toc117169621)

[Verwijder een record 12](#_Toc117169622)

[3.6. Hoe ziet MongoDB eruit in de spotitube applicatie? 12](#_Toc117169623)

[4. Discussie 15](#_Toc117169624)

[4.1. Wat is het verschil tussen de query’s van MongoDB en die van een relationele database? 15](#_Toc117169625)

[4.2. Hoe werken relaties in MongoDB in vergelijking met relationele databases? 15](#_Toc117169626)

[4.3. Zijn er veiligheidsrisico's van een MongoDB database en zo ja wat zijn die? 15](#_Toc117169627)

[4.4. Zijn er verschillen in de manier waarop data opgeslagen wordt in een MongoDB database en de manier waarop dit gebeurt in een MySQL database? 15](#_Toc117169628)

[4.5. Hoe ziet MongoDB eruit in een proef applicatie? 15](#_Toc117169629)

[4.6. Hoe ziet MongoDB eruit in een spotitube applicatie? 15](#_Toc117169630)

[5. Conclusie 16](#_Toc117169631)

[Bibliografie 17](#_Toc117169632)

[18](#_Toc117169633)

# Samenvatting

Binnen dit onderzoek staat de volgende hoofdvraag centraal: “*Wat zijn de verschillen van een MongoDB database vergeleken met een relationele database?***”**. De hoofdvraag wordt in dit onderzoek beantwoord door middel van de volgende deelvragen:

1. Wat is het verschil tussen de query’s van MongoDB en die van een relationele database?
2. Hoe werken relaties in MongoDB in vergelijking met relationele databases?
3. Zijn er veiligheidsrisico's van een MongoDB database en zo ja wat zijn die?
4. Hoe ziet MongoDB eruit in een proef applicatie?
5. Hoe ziet MongoDB eruit in de spotitube applicatie?

De vragen worden volgens verschillende onderzoeksmethoden onderzocht. Zo wordt er gebruikgemaakt van de onderzoeksmethoden literatuurstudie en prototyping.

# Introductie

## Vraagstelling

Om de hoofdvraag goed te kunnen beantwoorden zijn de volgende deelvragen opgesteld:

1. Wat is het verschil tussen de query’s van MongoDB en die van een relationele database?
2. Hoe werken relaties in MongoDB in vergelijking met relationele databases?
3. Zijn er veiligheidsrisico's van een MongoDB database en zo ja wat zijn die?
4. Hoe ziet MongoDB eruit in een proef applicatie?
5. Hoe ziet MongoDB eruit in de spotitube applicatie?

# Methoden en materialen

Bij dit onderzoek beantwoorden we de deelvragen door middel van de onderzoeksmethoden zoals beschreven in het ICT Research Methods Pack, omdat de Methods Pack een totaaloverzicht van de beschikbare onderzoeksstrategieën weergeeft. Ook is dit ons aangeraden en onderwezen vanuit de Hogeschool Arnhem Nijmegen, waarvoor wij dit onderzoek uitvoeren. In dit onderzoek is er gebruik gemaakt van de volgende onderzoeksmethodes om alle deelvragen te beantwoorden:

* Bibliotheek – available product analysis
* Bibliotheek – literatur study
* Labonderzoek – component test
* Werkplaatsonderzoek - prototyping

## Wat is het verschil tussen de query’s van MongoDB en die van een relationele database?

Wij hebben ervoor gekozen om deze deelvraag te beantwoorden door middel van available product analysis. Dit houdt in dat we gaan kijken naar bestaande oplossingen. In de context van deze vraag betekent dit dus dat we gaan kijken naar het verschil tussen MongoDB query’s en query’s zoals deze eruit zien bij een relationele database. Wij hebben voor deze methode aangezien er veel voorbeelden te vinden zijn van query’s voor zowel MongoDB als een relationele database. Door deze te vergelijken en de verschillen op te merken verwachten wij antwoord op deze deelvraag te krijgen.

## Hoe werken relaties in MongoDB in vergelijking met relationele databases?

Deze deelvraag gaan wij beantwoorden met behulp van de literature study methode. Wij gaan algemene informatie zoeken over het verschil tussen relaties in MongoDB en die in een relationele database. Vervolgens kiezen we een aantal interessante bronnen om goed door te lezen en vatten we samen wat we tijdens het lezen ontdekt hebben. Deze methode lijkt ons effectief omdat er veel informatie te vinden over de werking van relaties in MongoDB. Door onze ontdekkingen samen te vatten beantwoorden wij uiteindelijk deze deelvraag.

## Zijn er veiligheidsrisico's van een MongoDB database en zo ja wat zijn die?

Bij het beantwoorden van deze deelvraag gaan wij gebruikmaken van de literature study methode. We gaan naar bronnen zoeken die informatie bevatten over de veiligheidsrisico’s die komen met het gebruik van een MongoDB database. Vervolgens beantwoorden we deze vraag door samen te vatten wat we ontdekken uit de gekozen bronnen. Wij hebben ervoor gekozen om deze methode te gebruiken aangezien er hoogstwaarschijnlijk al veel bekend is over de veiligheidsrisico’s van een MongoDB database. Veiligheid is namelijk erg belangrijk bij een database.

## Zijn er verschillen in de manier waarop data opgeslagen wordt in een MongoDB database en de manier waarop dit gebeurt in een MySQL database?

Wij hebben de literature study methode gebruikt om deze vraag te beantwoorden. Aangezien MongoDB een NoSQL database is en MySQL een relationele database is zitten er waarschijnlijk verschillen in de manier waarop data opgeslagen wordt. Hier verwachten we veel informatie over te kunnen vinden dus vandaar de keuze voor de literature study methode.

## Hoe ziet MongoDB eruit in een proef applicatie?

Bij deze deelvraag wordt er gekeken naar hoe MongoDB gebruikt kan worden in een proef applicatie. Door dit eerst in een geïsoleerde omgeving te doen is er meer ruimte voor experimenteren over het gekozen onderwerp. Hiervoor wordt dan ook het labonderzoek - component test. Deze onderzoeksmethode wil dat er in een subsysteem of component in isolatie wordt getest om zo de correctheid te verzekeren voordat het geïntegreerd wordt.

## Hoe ziet MongoDB eruit in de spotitube applicatie?

Bij deze deelvraag wordt er gekeken naar hoe MongoDB in de Spotitube applicatie geïmplementeerd kan worden. Dit gaan wij doen aan de hand van de onderzoeksmethode werkplaatsonderzoek – prototyping. De reden hiervoor is omdat wij graag willen weten of het mogelijk is om MongoDB te gebruiken in de plaats van een relationele database voor de Spotitube applicatie.

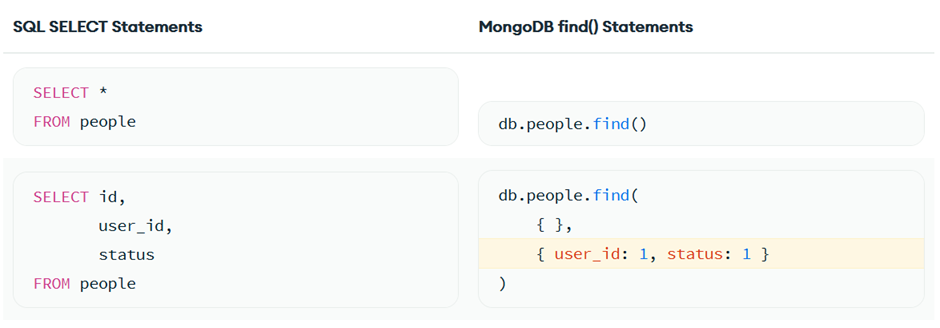
# Resultaten

## Wat is het verschil tussen de query’s van MongoDB en die van een relationele database?

Er zijn verschillen tussen de syntax van MongoDB en die van een relationele database. Een logisch verschil is dat de syntax er anders uit ziet. De programmeertaal voor het schrijven van query’s in MSQL is SQL. De programmeertaal voor het schrijven van query’s in MongoDB is Javascript. Maar naast het verschil in syntax zijn er nog een aantal andere verschillen. Dit zijn voor een applicatie als Spotitube ook gelijk de belangrijkste verschillen. Deze verschillen zitten vooral in de manier waarop data opgeslagen wordt.

### Select

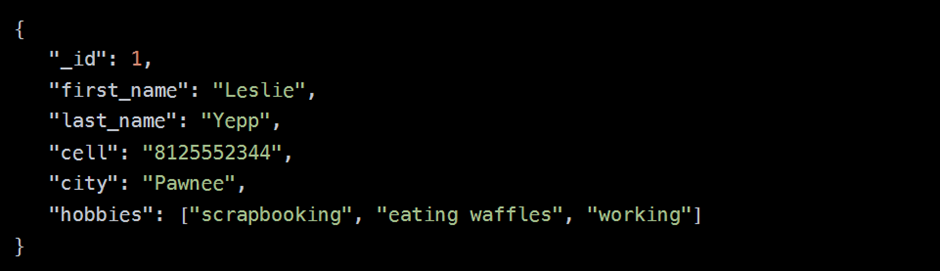
In figuur 1 zie je het verschil tussen de syntax van een select query zoals deze er bij een MySQL database uitziet en hoe dezelfde query er bij MongoDB uit zou zien. In principe zijn er geen grote verschillen tussen de werking van select statements en find() statements.



Figuur 1: Vergelijking van select statements uit MySQL en find() statements uit MongoDB.

### Insert

Insert statements werken in MongoDB niet helemaal hetzelfde zoals deze in MySQL werken. Dit komt door de manier waarop data in MongoDB opgeslagen wordt. MongoDB slaat rijen zoals we die in MySQL kennen op als JSON-objecten. Dit maakt het mogelijk om meerdere waarden in één ‘veld’ op te slaan. Dit wordt bijvoorbeeld met ‘hobbies’ gedaan in de onderstaande afbeelding.



Figuur 2: Voorbeeld van een JSON-object zoals deze in MongoDB worden opgeslagen.

Verder is de syntax nog steeds redelijk vanzelfsprekend behalve dat het in MongoDB mogelijk is om meerdere JSON-objecten te inserten in één statement (zie figuur 4).



Figuur 3: Vergelijking van insert statements uit MySQL en insertOne() statements uit MongoDB.



Figuur 4: Voorbeeld van een insertMany() statement.

### Update

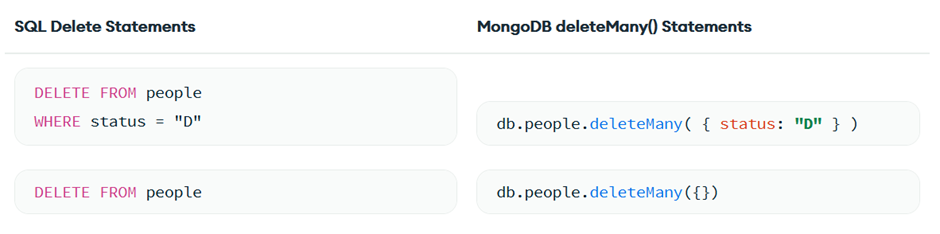
Net als bij de select statements werken update statements niet heel erg anders. De grootste verschillen in de werking van statements hebben bijna allemaal te maken met de verschillen in de manier waarop data opgeslagen wordt. Figuur 5 is een vergelijking van de verschillen in de syntax bij update statements.



Figuur 5: Vergelijking van update statements uit MySQL en updateMany() statements uit MongoDB.

### Delete

Bij de delete statements zijn er ook geen bijzonderheden die niet eerder, of later (bijvoorbeeld bij de resultaten van een andere deelvraag) in dit onderzoek bovenwater komen. Figuur 6 is een vergelijking van de verschillen in de syntax bij delete statements.



Figuur 6: Vergelijking van delete statements uit MySQL en deleteMany() statements uit MongoDB.

## Hoe werken relaties in MongoDB in vergelijking met relationele databases?

MongoDB is een non-relationele database. Een relationele database is een database die met relaties werkt. MongoDB werkt dus zonder relaties. Om dit te kunnen doen moeten er natuurlijk oplossingen bedacht zijn. Non-relationele hebben allemaal verschillende manieren om informatie op te slaan.

het mogelijk om lijsten binnen één JSON-object bij te houden, omdat MongoDB informatie als JSON-objecten opslaat. In de Spotitube applicatie zou dit bijvoorbeeld betekenen dat een playlist zelf al zijn tracks bij kan houden in plaats van dat er een playlist tabel, een track tabel en een soort tussentabel nodig is puur om tracks aan playlists te koppelen. Om deze reden zijn query's die data ophalen bij non-relationele databases vaak ook een stuk sneller aangezien een query die maar in één tabel hoeft te kijken sneller uitgevoerd kan worden dan een query die in 3 tabellen moet kijken.

## Zijn er veiligheidsrisico's van een MongoDB database en zo ja wat zijn die?

Er blijken groten verschillen te zitten tussen NoSQL databases en relationele databases op gebied van veiligheid. Een groot deel van deze problemen die gelden voor de meeste NoSQL databases worden opgelost door MongoDB. Zo ondersteunt MongoDB bijvoorbeeld de ACID-regels. Een ander probleem met NoSQL databases is een gebrek aan encryptie-tools. Aangezien MongoDB erg populair is zijn er wat meer opties dan bij andere NoSQL databases maar in vergelijking met SQL-databases vallen de encryptiemogelijkheden zwaar tegen.

Uiteindelijk weet MongoDB een groot deel van de veiligheidsrisico’s waar de meeste NoSQL databases tegenaan lopen op, maar het probleem ligt hem meer in de toegankelijkheid van deze oplossingen. Aangezien SQL-databases al langer gebruikt worden en bijna overal dominant zijn is er veel meer bekend over het gebruik hiervan. Er zijn oplossingen bedacht op allerlei specifieke problemen. Waar je ook vastloopt, op het internet is er een antwoord te vinden. Bij MongoDB is dit niet het geval en dit zorgt ervoor dat er vaak minder voor beveiliging gedaan wordt. Het is wel mogelijk om MongoDB op een veilige manier te gebruiken maar veel programmeurs weten niet hoe. Voor de Spotitube applicatie is veiligheid erg belangrijk aangezien we werken met accounts en dus wachtwoorden van gebruikers. Als we de MySQL database willen vervangen voor een MongoDB database zal er erg veel tijd besteed moeten worden aan het beveiligen van de nieuwe database.

## Zijn er verschillen in de manier waarop data opgeslagen wordt in een MongoDB database en de manier waarop dit gebeurt in een MySQL database?

MongoDB is een NoSQL database. NoSQL databases slaan data op een andere manier op dan hoe relationele databases zoals MySQL dit doen. Los hiervan verschillen NoSQL databases ook onderling van elkaar. MongoDB slaat informatie op als JSON-objecten. Dit betekent dat er geen tabellen zijn die je kan bekijken. In plaats van tabellen slaat MongoDB deze JSON-objecten op in verschillende ‘collections’.

Er geen gebruik gemaakt van relaties tussen collections aangezien MongoDB een non-relationele database is. In plaats hiervan is het mogelijk om een JSON-object een lijst bij te laten houden. In MySQL kan een rij uit een tabel niet in één veld meerdere waardes bijhouden dus zijn er relaties nodig om dit probleem op te lossen. Een resultaat van de oplossing die MongoDB heeft voor het gebrek aan relaties is dat er vaak dubbele data wordt opgeslagen.

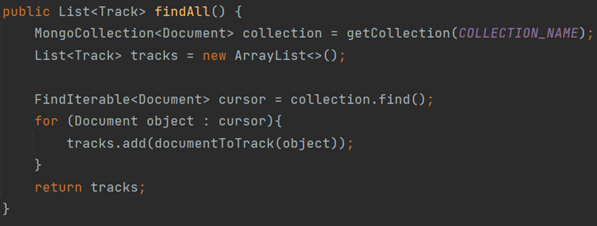
## Hoe ziet MongoDB eruit in een proef applicatie?

Om MongoDB in een proef applicatie aan de praat te krijgen hebben we gebruik gemaakt van MongoDB Community Server versie 6.0.2 (momenteel de huidige versie) en JDK 17.

Hieronder staan code blokken die laten zien hoe data opgehaald en weggeschreven wordt met MongoDB als database. Dit gebeurt allemaal in een applicatie met de minimale benodigdheden om de methodes te testen.

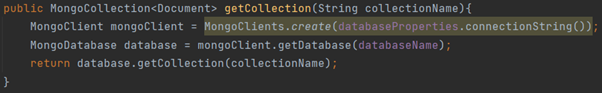
### Haal alle records op

Om alle records op te halen uit een tabel wordt er gebruik gemaakt van de .find() methode die aangeroepen wordt op een MongoCollection>Document> object hoe dat object opgehaald wordt kom ik later op terug. Door de find methode aan te roepen komt er een lijst met FindIterable<Document> terug. Dit is een lijst met alle records die in de database onder de collection met de collectionName. De lijst met records wordt daarna per object vertaald naar een track object (iets verder wordt laten zien hoe dat gebeurt) en toegevoegd aan een lijst met alle tracks. Dit doe ik om de objecten later uit te kunnen lezen.



Code blok 1: findAll.

In Code blok 2 : getCollection wordt er een connectie met de database gemaakt. Daar uit de gewenste database opgehaald en daarna wordt de gewenste collection terug gegeven door de methode getCollection.



Code blok 2: getCollection.

In code blok 3 : documentToTrack wordt het object vertaald naar een track object om dit later uit te kunnen lezen.

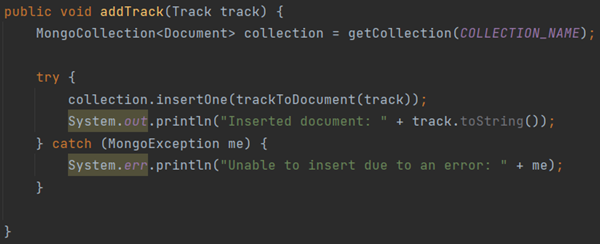
Afbeelding met tekst

Automatisch gegenereerde beschrijving

Code blok 3: documentToTrack.

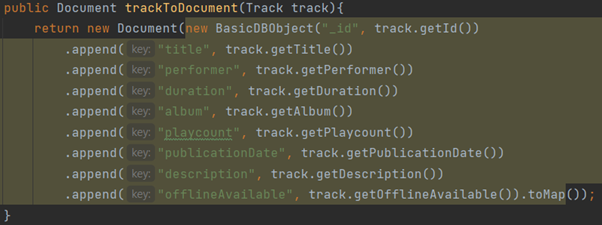
### Voeg een record toe

Om een track toe te voegen aan de collection tracks in de database wordt de functie insertOne met een Track object die vertaald wordt naar een Document object. Als er geen database gevonden kan worden met de desbetreffende collection worden die aangemaakt bij de aanroep.



Code blok 4: addTrack.

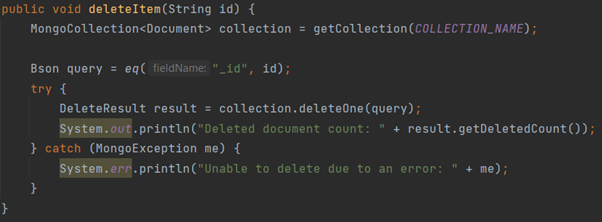
Voordat een Track object naar de database gestuurd kan worden wordt het object vertaald naar een BasicDBObject en die wordt direct door vertaald naar een Document object.



Code blok 5: trackToDocument.

### Verwijder een record

Om een record te verwijderen uit de database wordt de functie deleteOne aangeroepen. Deze functie verwacht een mee gegeven query om te weten wat die moet verwijderen. De deleteOne methode heeft een DeleteResult waarmee gekeken kan worden hoeveel records verwijderd zijn.

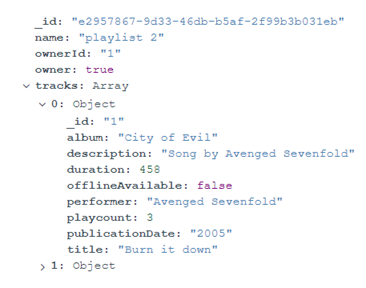


Code blok 6: deleteItem.

## Hoe ziet MongoDB eruit in de spotitube applicatie?

Iets dat ons meteen is opgevallen tijdens het implementeren van MongoDB in de spotitube applicatie is dat alle query's wegvallen omdat er voor de standaard query’s methodes bestaan in de library com.MongoDB.client.MongoCollection.

De functie om een track toevoegen aan een playlist werd eerder afgehandeld in de trackDAO. De data afhandeling wordt nu gedaan door playlistDAO. De reden hiervoor is omdat in de huidige database ik ervoor gekozen heb om tracks direct op te slaan als een array in de playlist collection. In collection 1 : playlist is te zien hoe de data er uit ziet.

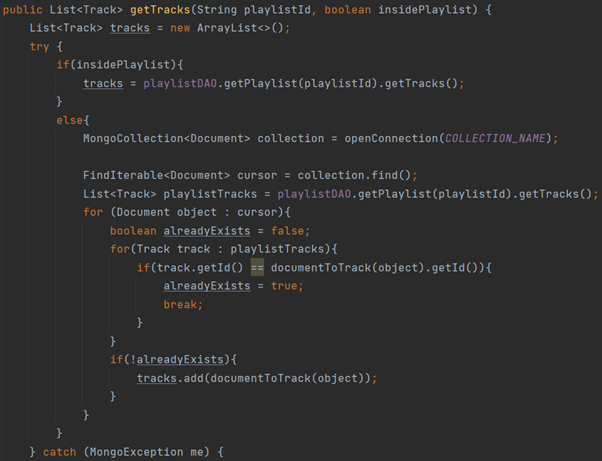


Collection 1 : playlist.

Door de data anders op te slaan als in een relationele database is er code veranderd in de DAO’s om hetzelfde voor elkaar te krijgen. Er wordt door een simpele boolean mee te krijgen een andere query aangeroepen dit is terug te zien in Code blok 7 : oude getTracks. Maar omdat er geen relaties zijn in MongoDB moeten om dezelfde resultaten code geschreven worden. Dit is terug te zien in code blok 8 : nieuwe getTracks.

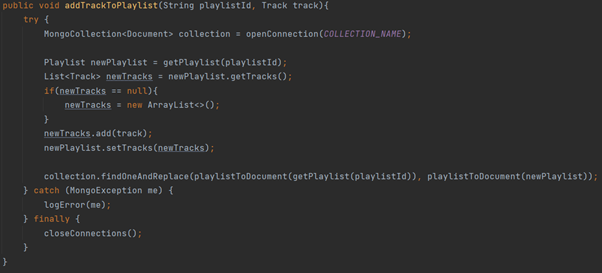


Code blok 7: oude getTracks.



Code blok 8: nieuwe getTracks.

In code blok 9 : addTrackToPlaylist staat code om een track toe te voegen aan een playlist. Dit doe ik door gebruik te maken van findOneAndReplace methode. Hoe wij dit gebruiken is door de playlist op te halen en in een newPlaylist object te zetten. Daarvan halen we de tracks op. Als de playlist geen tracks heeft maakt die een nieuwe lijst met tracks aan. Daarna voegt het de track toe aan de lijst. Wanneer dit gedaan is wordt de findOneAndReplace aangeroepen hier wordt als eerst de parameter de playlist mee gegeven die we willen aanpassen en als tweede parameter de nieuw gemaakte playlist die eigenlijk de oude playlist is + een nieuwe track in de lijst met tracks.



Code blok 9: addTrackToPlaylist.

# Discussie

## Wat is het verschil tussen de query’s van MongoDB en die van een relationele database?

Bij het beantwoorden van deze deelvraag hebben we gebruik gemaakt van de officiële website van MongoDB. Wij vinden dit een betrouwbare bron gezien de informatie van de officiële website komt.

## Hoe werken relaties in MongoDB in vergelijking met relationele databases?

Bij het beantwoorden van deze deelvraag hebben we gebruik gemaakt van de officiële website van MongoDB. Wij vinden dit een betrouwbare bron gezien de informatie van de officiële website komt.

## Zijn er veiligheidsrisico's van een MongoDB database en zo ja wat zijn die?

Bij het beantwoorden van deze deelvraag hebben we gebruik gemaakt van de officiële website van MongoDB. Ook hebben we een artikel van 'SecureCoding' gebruik. Deze bron vinden wij betrouwbaar aangezien de informatie van MongoDB van de officiële website komt. 'SecureCoding' vinden wij betrouwbaar aangezien ze bekend staan om hun artikelen. Ook is er een datum en een auteur bekend wat het ook betrouwbaar maakt.

## Zijn er verschillen in de manier waarop data opgeslagen wordt in een MongoDB database en de manier waarop dit gebeurt in een MySQL database?

Bij het beantwoorden van deze deelvraag hebben we gebruik gemaakt van een artikel over het verschil tussen NoSQL databases en relationele databases. Dit artikel stond op de officiële website van MongoDB. Wij vinden dit een betrouwbare bron gezien de informatie van de officiële website komt.

## Hoe ziet MongoDB eruit in een proef applicatie?

Bij het beantwoorden van deze deelvraag is er gebruik gemaakt van de informatie die staat op de officiële website van MongoDB. De informatie die op deze website staat is betrouwbaar omdat het door de makers van MongoDB gemaakt is.

## Hoe ziet MongoDB eruit in een spotitube applicatie?

Tijdens deze deelvraag is geen onderzoek op bronnen gedaan omdat alle informatie die wij al nodig hebben bekend is door de rest van de deelvragen.

# Conclusie

De hoofdvraag die wij aan het begin van het onderzoek gesteld hebben was: *“Wat zijn de verschillen van een MongoDB database ten opzichte van een relationele database?”* Na alle deelvragen beantwoord te hebben zouden we de hoofdvraag moeten kunnen beantwoorden. In de resultaten van de deelvragen geven wij een aantal verschillen aan. Zo is MongoDB bijvoorbeeld over het algemeen minder veilig. MongoDB is met het uitvoeren van select query’s sneller. Aangezien Spotitube met tracks op playlists werkt is het erg vanzelfsprekend om met relaties te werken. Het is nou eenmaal logisch dat een playlist een relatie heeft met een track in plaats van dat dezelfde track 2 keer bij een andere playlist opgeslagen staat. Dit samen met het veiligheidsprobleem maakt dat wij vinden dat MongoDB geen goede keuze is voor de Spotitube-applicatie.

# Bibliografie

*db.collection.updateOne()*. (n.d.). Retrieved from MongoDB: https://www.mongodb.com/docs/manual/reference/method/db.collection.updateOne/#mongodb-method-db.collection.updateOne

Dechalert, A. (2021 , April 12). *Is MongoDB Actually Safe to Use?* Retrieved from SecureCoding: https://www.securecoding.com/blog/mongodb-security/

*Delete a Document*. (n.d.). Retrieved from MongoDB: https://www.mongodb.com/docs/drivers/java/sync/current/usage-examples/deleteOne/

*Getting Started with MongoDB and Java: Part I* . (n.d.). Retrieved from MongoDB: https://www.mongodb.com/blog/post/getting-started-with-mongodb-and-java-part-i

*Methods*. (n.d.). Retrieved from ictresearchmethods: https://ictresearchmethods.nl/Methods

*MongoDB*. (n.d.). Retrieved from MongoDB: https://www.mongodb.com

*MongoDB vs MySQL Performance: 7 Critical Differences*. (n.d.). Retrieved from HevoData: https://hevodata.com/learn/mongodb-vs-mysql/#sql-vs-nosql

*NoSQL vs. SQL Databases*. (n.d.). Retrieved from MongoDB: https://www.mongodb.com/nosql-explained/nosql-vs-sql

*What Is a Non-Relational Database?* (n.d.). Retrieved from MongoDB: https://www.mongodb.com/databases/non-relational

# 

