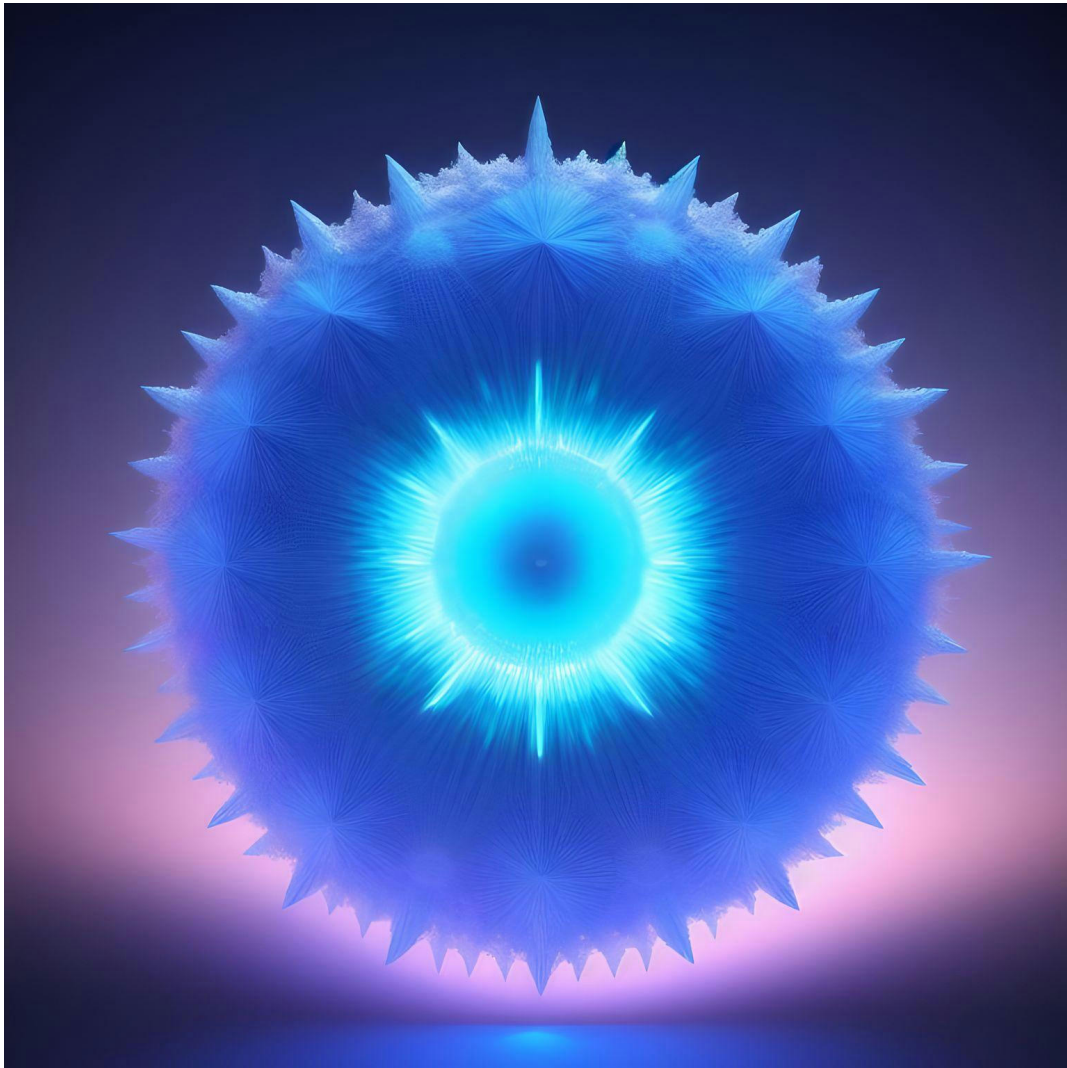


**Competentie: Adviseren**  
Adviesrapport en aanbevelingen



**Student:** Levi Leuwol  
**Studentnummer:** 405538  
**Datum:** 2023-06-30  
**Onderwerp:** Competentie: Analyseren  
**Opleiding:** HBO-ICT  
**Studiejaar:** 3

# Inhoudsopgave

1 Samenvatting .....	3
2 Inleiding .....	4
2.1 Context .....	4
2.2 Onderzoek .....	4
2.3 Randvoorwaarden .....	4
2.4 Overzicht van de structuur .....	5
3 Onderzoek .....	6
3.1 Onderzoeksopzet .....	6
3.2 Resultaten .....	6
3.3 Conclusies .....	6
4 Alternatieven .....	7
4.1 Alternatief 1: Optimalisatie van de huidige setup .....	7
4.2 Alternatief 2: Gebruik van een andere database .....	7
4.3 Alternatief 3: Wijziging van node architectuur .....	7
4.4 Benodigdheden .....	7
4.5 Voor- en nadelen .....	7
5 Conclusie .....	9
5.1 Advies .....	9
5.2 Onderbouwing .....	9
5.3 Actieplan .....	9
6 Bronnen .....	10

# 1 Samenvatting

Dit rapport onderzoekt de mogelijkheden voor het ontwikkelen van een datavisualisatietool in de voedingsindustrie, waarbij Neo4j als onderliggende database wordt gebruikt. De kern van het onderzoek richt zich op het vergelijken van de prestaties tussen het gebruik van een Object Graph Mapper (OGM) en de standaard Neo4j-driver bij het uitvoeren van specifieke query's om alle nodes te verkrijgen.

Uit de initiële resultaten blijkt dat er prestatieverschillen zijn tussen de twee benaderingen, maar de dataset was te beperkt om definitieve conclusies te trekken. Daarom wordt aanbevolen om verder onderzoek te doen, met name met een grotere dataset om de resultaten te valideren.

Het rapport biedt drie alternatieven voor het verbeteren van de huidige situatie:

- Het optimaliseren van de huidige Neo4j-setup voor betere prestaties.
- Het overwegen van het gebruik van een andere database die beter past bij de vereisten.
- Een herontwerp van de huidige node-architectuur om efficiënter gebruik te maken van databasebronnen.

De aanbeveling is om te beginnen met het optimaliseren van de huidige Neo4j-setup, aangezien dit zowel kostenefficiënt als tijdbesparend lijkt te zijn. Echter, verder onderzoek met een uitgebreidere dataset is cruciaal voor het nemen van een weloverwogen besluit.

## 2 Inleiding

### 2.1 Context

De huidige digitale transformatie heeft geleid tot een explosieve groei van data binnen verschillende industrieën, waaronder de voedingsindustrie. Databases zijn een onmisbaar component geworden in dit landschap, met Neo4j als één van de meest populaire grafendatabases. De opkomst van grafendatabases heeft geleid tot een groeiende vraag naar datavisualisatietools die de data in deze databases kunnen visualiseren.

In deze context is het project ontstaan om een datavisualisatietool te ontwikkelen voor de voedingsindustrie, gebaseerd op de Neo4j database. De tool moet in staat zijn om de data in de database te visualiseren, en de gebruiker in staat stellen om de data te manipuleren en te analyseren.

De optimalisatie van databasetoegang is van groot belang, vooral als het gaat om applicaties die veel data moeten ophalen en verwerken, zoals een datavisualisatietool. Daarom is het van belang om de verschillende technologieën te onderzoeken die gebruikt kunnen worden om de tool te ontwikkelen, en de architectuur van de tool te optimaliseren om de gewenste functionaliteit te realiseren.

### 2.2 Onderzoek

Bij dit adviesrapport hoort een onderzoeksrapport, waarin de resultaten van het onderzoek worden beschreven. Dit onderzoeksrapport is te vinden in de folder documenten/onderzoeksrapport.pdf.

Het primaire focus van dit onderzoek is het vergelijken van de prestaties tussen een Object Graph Mapper (OGM) en de standaard Neo4j-driver bij het ophalen van alle nodes uit een Neo4j-database. Door de impact van beide methoden op de verwerkingstijden en de variabiliteit van de metingen te analyseren, biedt het onderzoek waardevolle inzichten in het optimaliseren van database-interacties voor de datavisualisatietool.

De onderzoeksvraag die centraal staat in dit onderzoek is: **“Wat is het verschil in prestaties tussen het gebruik van de Object Graph Mapper (OGM) en de standaard Neo4j-driver bij het ophalen van alle nodes uit een Neo4j-database?”**. Om deze vraag te beantwoorden, is een experiment uitgevoerd waarbij de prestaties van een OGM en de standaard Neo4j-driver zijn vergeleken bij het ophalen van alle nodes uit een Neo4j-database. De resultaten van het experiment zijn geanalyseerd om de impact van beide methoden op de verwerkingstijden en de variabiliteit van de metingen te bepalen.

### 2.3 Randvoorwaarden

Randvoorwaarden voor dit project zijn:

- **Webinterface:** De tool moet een webinterface hebben, zodat het toegankelijk is voor gebruikers via een webbrowser. Er dient geen native applicatie ontwikkeld te worden.
- **LCP (Largest Contentful Paint, laadtijd van de grafiek):** Deze moet onder de 2.5 seconden blijven, om een goede gebruikerservaring te garanderen.
- **INP (Input Delay, reactietijd van de pagina):** Deze moet onder de 200ms blijven, om een goede gebruikerservaring te garanderen.

## 2.4 Overzicht van de structuur

1. **Samenvatting:** Dit deel biedt een beknopt overzicht van de belangrijkste punten van het rapport.
2. **Inleiding:** Hier wordt de context en het doel van het rapport uiteengezet, inclusief achtergrondonderzoek en randvoorwaarden.
3. **Onderzoek:** Dit deel beschrijft de onderzoeksopzet, resultaten en conclusies van het onderzoek.
4. **Conclusie:** Hier wordt een advies gegeven over de technologieën die gebruikt kunnen worden voor de ontwikkeling van de tool, en de architectuur van de tool.
5. **Bronnen:** Hier worden alle referenties en bronnen vermeld die zijn gebruikt tijdens het schrijven van het rapport.

## **3 Onderzoek**

### **3.1 Onderzoeksopzet**

Om de gestelde onderzoeksvraag te beantwoorden, is een onderzoeksopzet opgesteld. Het primaire doel was om de prestatieverschillen te meten tussen het gebruik van een Object Graph Mapper (OGM) en de standaard Neo4j-driver bij het ophalen van alle nodes uit een Neo4j-database. De metingen focusten zich op de tijd in milliseconden die nodig was om deze taak te voltooien.

Er werd gebruik gemaakt van een Neo4j-database met 2349 nodes, 7678 relaties, 19998 eigenschappen en 2321 labels. De metingen werden uitgevoerd op een Debian server met een AMD Ryzen 5 3600, 16GB RAM en een 1TB SSD.

### **3.2 Resultaten**

Twee Python-scripts werden gebruikt voor de analyse. Het eerste script maakt grafieken die de tijd weergeven die nodig is om nodes op te halen, met behulp van de bibliotheek Matplotlib. Het tweede script berekent statistische parameters zoals de standaarddeviate en variantie van de metingen, met behulp van de bibliotheek Numpy.

Uit de eerste bevindingen blijkt dat er een merkbaar verschil is in de tijdsduur tussen het gebruik van een OGM en de standaard Neo4j-driver. De exacte implicatie van dit verschil zal verder moeten worden geanalyseerd.

### **3.3 Conclusies**

Ondanks de waargenomen prestatieverschillen is het essentieel op te merken dat dit onderzoek een beperking kent: de omvang van de dataset. Met slechts 2349 nodes is de dataset relatief klein, en is het mogelijk dat de prestatieverschillen tussen een OGM en de standaard Neo4j-driver groter worden bij een grotere dataset.

Ook is het mogelijk dat de prestatieverschillen tussen een OGM en de standaard Neo4j-driver afhankelijk zijn van de complexiteit van de queries, en standaard caching configuraties.

Daarom valt er op basis van dit onderzoek geen definitieve conclusie te trekken over de prestatieverschillen tussen een OGM en de standaard Neo4j-driver. Het is aan te raden om dit onderzoek te herhalen met een grotere dataset, om een definitieve conclusie te kunnen trekken.

## 4 Alternatieven

### 4.1 Alternatief 1: Optimalisatie van de huidige setup

Het eerste alternatief is om de huidige setup te optimaliseren, en uitgebreider onderzoek te doen naar de prestatieverschillen tussen een OGM en de standaard Neo4j-driver. Dit kan gedaan worden door het onderzoek te herhalen met een grotere dataset, en door de complexiteit van de queries te variëren.

Er kan gekeken worden naar het wijzigen van standaardconfiguraties van de database, zoals de caching configuratie. Ook kan er gekeken worden naar het optimaliseren van queries door het gebruik van indexen, dit kan de prestaties van de database mogelijk aanzienlijk verbeteren.

### 4.2 Alternatief 2: Gebruik van een andere database

Een ander alternatief dat overwogen kan worden, is het gebruik van een andere database dan Neo4j. Neo4j is in dit geval gekozen, omdat het een van de meest populaire grafendatabases is. Hierdoor is er veel documentatie en ondersteuning beschikbaar, wat de ontwikkeling van de tool kan versnellen. Ook is Neo4j een open-source database, wat betekent dat er geen licentiekosten aan verbonden zijn. Tenslotte is Neo4j een ACID-compliant database, wat betekent dat het voldoet aan de vier eigenschappen van een transactie: atomiciteit, consistentie, isolatie en duurzaamheid.

Het prestatieverschil met andere databases is niet meegenomen in de beslissing om Neo4j te gebruiken voor dit project. Voor een vollediger beeld zou vervolgonderzoek moeten vergelijken hoe Neo4j presteert ten opzichte van andere database-opties. Tenslotte kunnen andere databases ook andere voordelen bieden, zoals specifieke functionaliteit die niet beschikbaar is in Neo4j.

### 4.3 Alternatief 3: Wijziging van node architectuur

Een ander alternatief dat overwogen kan worden, is het wijzigen van de architectuur van de nodes. In de huidige setup zijn de nodes allemaal van hetzelfde type, en bevatten ze allemaal dezelfde eigenschappen. Het is mogelijk dat het wijzigen van de architectuur van de nodes de prestaties van de database kan verbeteren.

### 4.4 Benodigdheden

Voor het implementeren van deze alternatieven zijn de volgende benodigdheden nodig:

- **Tijd:** Het uitvoeren van het onderzoek en het implementeren van de alternatieven kost tijd. Het is belangrijk om deze tijd in te plannen, en te zorgen dat er voldoende tijd beschikbaar is om de alternatieven te implementeren.
- **Kennis:** Het implementeren van de alternatieven vereist kennis van de verschillende technologieën die gebruikt worden. Het is belangrijk om te zorgen dat er voldoende kennis beschikbaar is om de alternatieven te implementeren.
- **Geld:** Het implementeren van de alternatieven kan kosten met zich meebrengen, bijvoorbeeld voor het aanschaffen van licenties voor andere databases. Het is belangrijk om deze kosten in kaart te brengen, en te zorgen dat er voldoende budget beschikbaar is om de alternatieven te implementeren.

### 4.5 Voor- en nadelen

De voor- en nadelen van de alternatieven zijn als volgt:

- **Alternatief 1:** Het voordeel van dit alternatief is dat het optimaliseren van de huidige setup minder tijd en geld kost dan het implementeren van een ander alternatief. Het nadeel is dat het optimaliseren van de huidige setup mogelijk niet voldoende is om de gewenste prestaties te behalen.

- **Alternatief 2:** Het voordeel van dit alternatief is dat het gebruik van een andere database mogelijk betere prestaties kan bieden dan Neo4j. Het nadeel is dat het implementeren van een andere database meer tijd en geld kost dan het optimaliseren van de huidige setup.
- **Alternatief 3:** Het voordeel van dit alternatief is dat het wijzigen van de architectuur van de nodes mogelijk betere prestaties kan bieden dan de huidige architectuur. Het nadeel is dat het wijzigen van de architectuur van de nodes meer tijd kost dan het optimaliseren van de huidige setup.



## **5 Conclusie**

### **5.1 Advies**

Mijn eerste advies is om te beginnen met het optimaliseren van de huidige Neo4j-setup, zoals aangegeven in Alternatief 1. Dit lijkt de meest efficiënte en kostenbesparende optie op korte termijn. Daarnaast is het zinvol om verder onderzoek te doen naar de prestatieverschillen tussen een OGM en de standaard Neo4j-driver.

Voor een langetermijnstrategie zou ik echter aanraden om Alternatief 2 niet volledig van tafel te vegen. Als de optimalisaties in Alternatief 1 onvoldoende blijken, zou het zinvol kunnen zijn om de prestaties van Neo4j te vergelijken met andere databases.

### **5.2 Onderbouwing**

Het kiezen voor een optimalisatie van de huidige setup is gebaseerd op de resultaten van het voorlopige onderzoek, die duidelijk een verschil in prestatie tonen tussen de gebruikte methodes. Echter, het onderzoek is niet uitputtend en de dataset is relatief klein, dus het is een voorzichtige eerste stap.

Daarnaast biedt Neo4j verschillende voordelen zoals ACID-compliance en een rijke set aan documentatie, wat de leercurve en implementatietijd kan verkorten. Het overstappen naar een andere database zou deze voordelen kunnen wegnemen, en mogelijk nieuwe problemen introduceren.

### **5.3 Actieplan**

Een eerste actiepunt is het optimaliseren van de huidige Neo4j-configuratie. Dit omvat het finetunen van queries en eventueel aanpassen van database-instellingen. Deze aanpassingen zijn relatief snel door te voeren en vereisen geen drastische wijzigingen in de huidige setup.

Vervolgens, plan voor het uitvoeren van een vervolgonderzoek met een grotere dataset. Hierbij kan ook de complexiteit van de queries worden gevarieerd om een meer genuanceerd beeld te krijgen van de prestatieverschillen. Op basis van deze resultaten kan dan een gefundeerde beslissing gemaakt worden over het al dan niet switchen van database.

Tenslotte is het ook van belang om andere alternatieven te overwegen, als deze zich voordoen. Het is mogelijk dat er andere alternatieven zijn die niet zijn meegenomen in dit adviesrapport, maar die wel een betere oplossing bieden voor het probleem.

## 6 Bronnen

### Bijlagen

Hanzehogeschool Groningen logo	Hanzehogeschool Groningen	<a href="https://freebiesupply.com/logos/hanzehogeschool-groningen-logo/">https://freebiesupply.com/logos/hanzehogeschool-groningen-logo/</a>
Titelpagina figuur	DALL-E-3, OpenAI	bijlagen -> realisatierapport -> OIG.jpg
Symfony Server Source Code	L. J. J. Leuwol	<a href="https://github.com/Shambuwu/neo4j-symfony-app">https://github.com/Shambuwu/neo4j-symfony-app</a>