Wail Abou (1762891)

Inhoud

[Aanleiding 2](#_Toc69942721)

[Kwestie 2](#_Toc69942722)

[Voorgestelde oplossing(en) 3](#_Toc69942723)

[Argumenten 3](#_Toc69942724)

[Conclusie 3](#_Toc69942725)

[Bijlagen 4](#_Toc69942726)

[Queries 6](#_Toc69942727)

[Nawoord 7](#_Toc69942728)

# Aanleiding

Zoals in de opdrachtbeschrijving al genoemd wordt zal het bedrijf zich uitbreiden en daarmee meer data verwerken. Hierbij zijn al 2 van de 5 v’s benoemt (Volume and Velocity), maar ik vind het ook belangrijk om Value toe te lichten. Naar mate het bedrijf groeit zal de data ook groeien, hoe meer data betekent vaak hoe meer het waard is en hoe meer het kost om te onderhouden (beveiliging, opslag).

Data wordt het nieuwe olie genoemd met een reden want goeie analyses uitvoeren op data kan je inzicht geven en op z’n beurt veel geld genereren. Daarom is het ook belangrijk dat je de juiste opslag methode gekozen wordt zodat Volume en Velocity eisen nageleefd worden en daarmee ook de Value bewaart wordt.

# Kwestie

De huidige oplossing kan goed opschalen met veel gebruikers tot een bepaalde aantal, dan zal het bedrijf er voor moeten kiezen om de server verticaal of horizontaal te schalen. Dus om een snellere server voor de database te zetten of meerdere servers die de workload verdelen.

De grenzen zullen vooral liggen bij de business vragen, naar mate de vragen complexer worden zullen de query’s die bedacht en uitgevoerd moeten worden ook complexer worden. Stel je wilt 2 producten vergelijken, dan zal je 2 keer de product tabel moeten inladen verhoog je dit naar 3 producten dan laad je 3 keer de product table in. Dit geldt voor n aantal waarbij n het aantal producten is, dit schaalt dus lineair.

Dit zal voor veel vragenstukken het geval zijn, en dat is nou net wat we niet willen hebben want de truc met big data is inzicht creëren en als dat niet met een redelijke snelheid kan lijkt mij deze oplossing niet geschikt.

Ook schaalt de query executie tijd lineair mee, om de volledige set duurt het 10x zo lang omdat die ongeveer 10x zo groot is.

# Voorgestelde oplossing(en)

Mijn voorgestelde oplossing is om naar een andere database type over te gaan zoals MongoDB. Deze slaat als JSON byte formaat op waardoor dit snel af te lezen is omdat er geen structuur/schema aan vast zit zoals bij MySQL. Bovendien bevat het functies zoals map en reduce die dingen verder kunnen laten versnellen, de manier waarop je query’s schrijft is ook een stuk dichter de logica van een programmeur dus dat kan ook potentieel de ontwikkeling snelheid versnellen.

# Argumenten

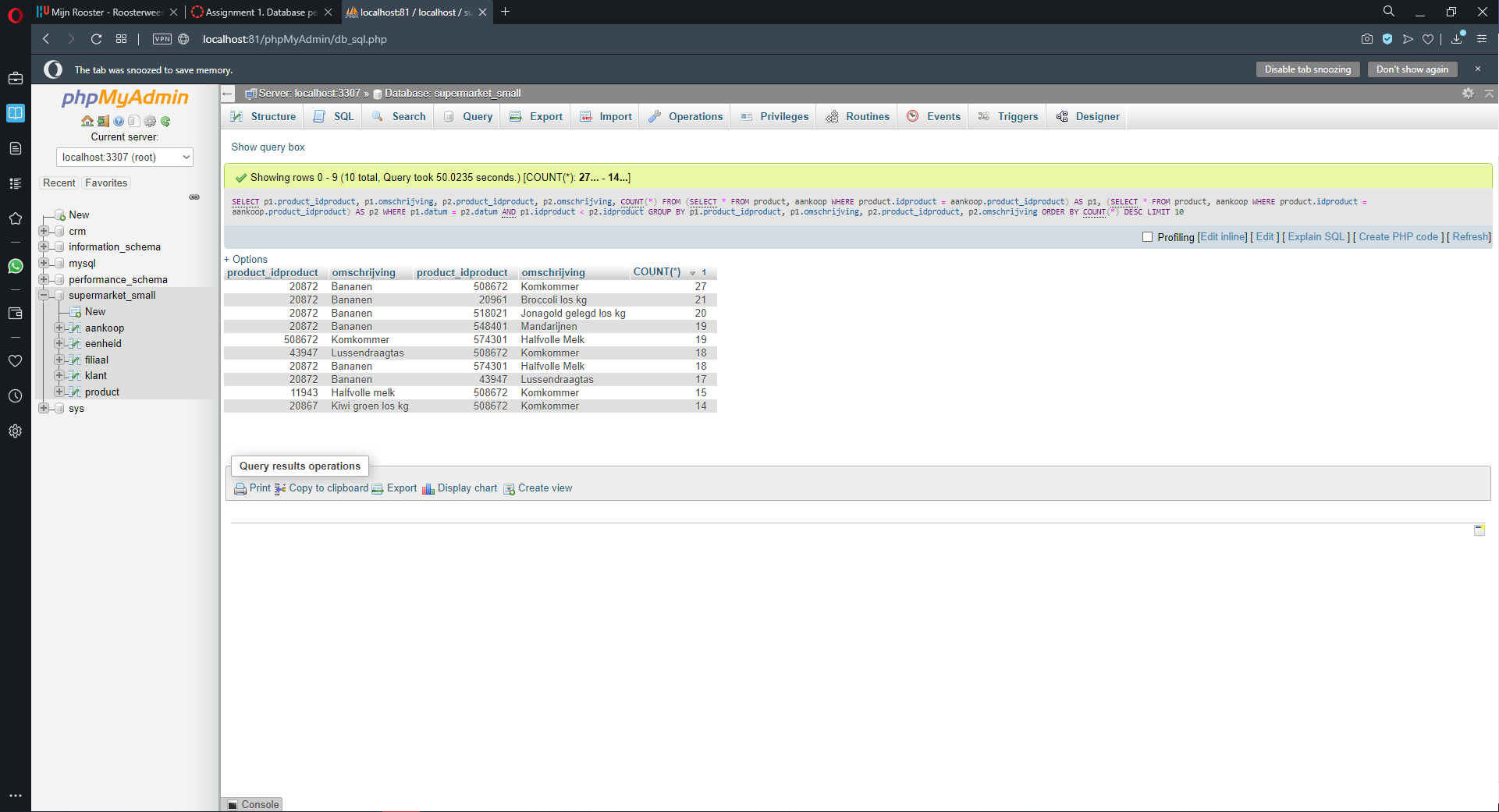
Ik heb deze video gebruikt om JMeter op te zetten: <https://www.youtube.com/watch?v=oy53KAKHpts>. Vervolgens heb ik zelf rondgekeken in het programma en kwam uit op interessante dingen zoals dat je het aantal users kan instellen en dat je zelfs kan instellen of deze users uniek moeten zijn of telkens dezelfde connectie. Hieruit heb ik mijn conclusie kunnen trekken dat de complexiteit van een query een groter probleem is dan het aantal concurrent users.

# Conclusie

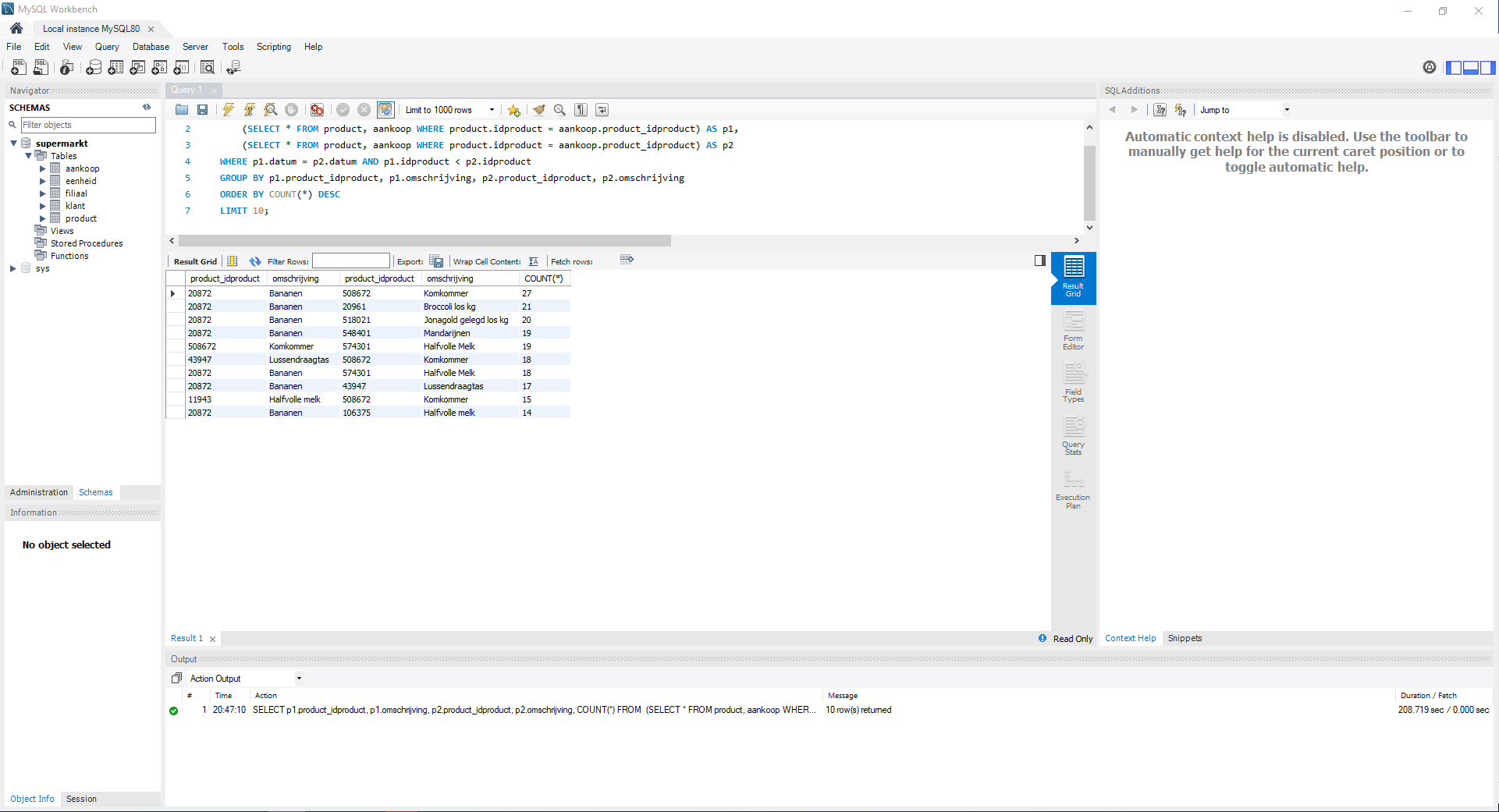
De conclusie waar ik op uit kom is dat het systeem dat er nu staat niet kan blijven. Het is ongeschikt om veel business logic op uit te voeren, dit is niet ideaal want een supermarkt draait hier juist om. Bij een supermarkt wordt er heel veel gebruik gemaakt van business logica om bijvoorbeeld te kijken wat het koopgedrag is vaan een gemiddelde klant en op basis daarvan wordt de winkel/flow aangepast.

# Bijlagen

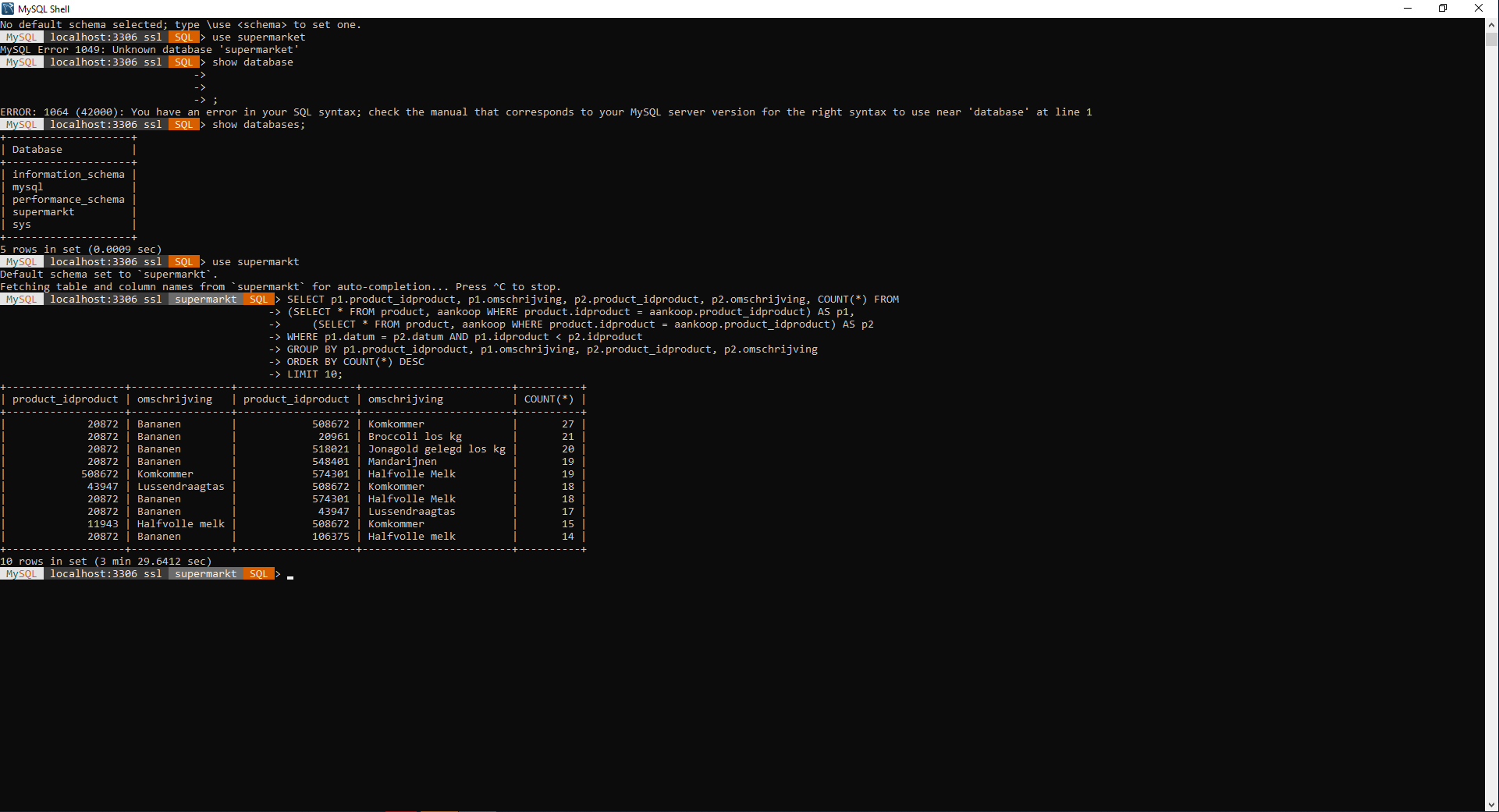
phpMyAdmin:



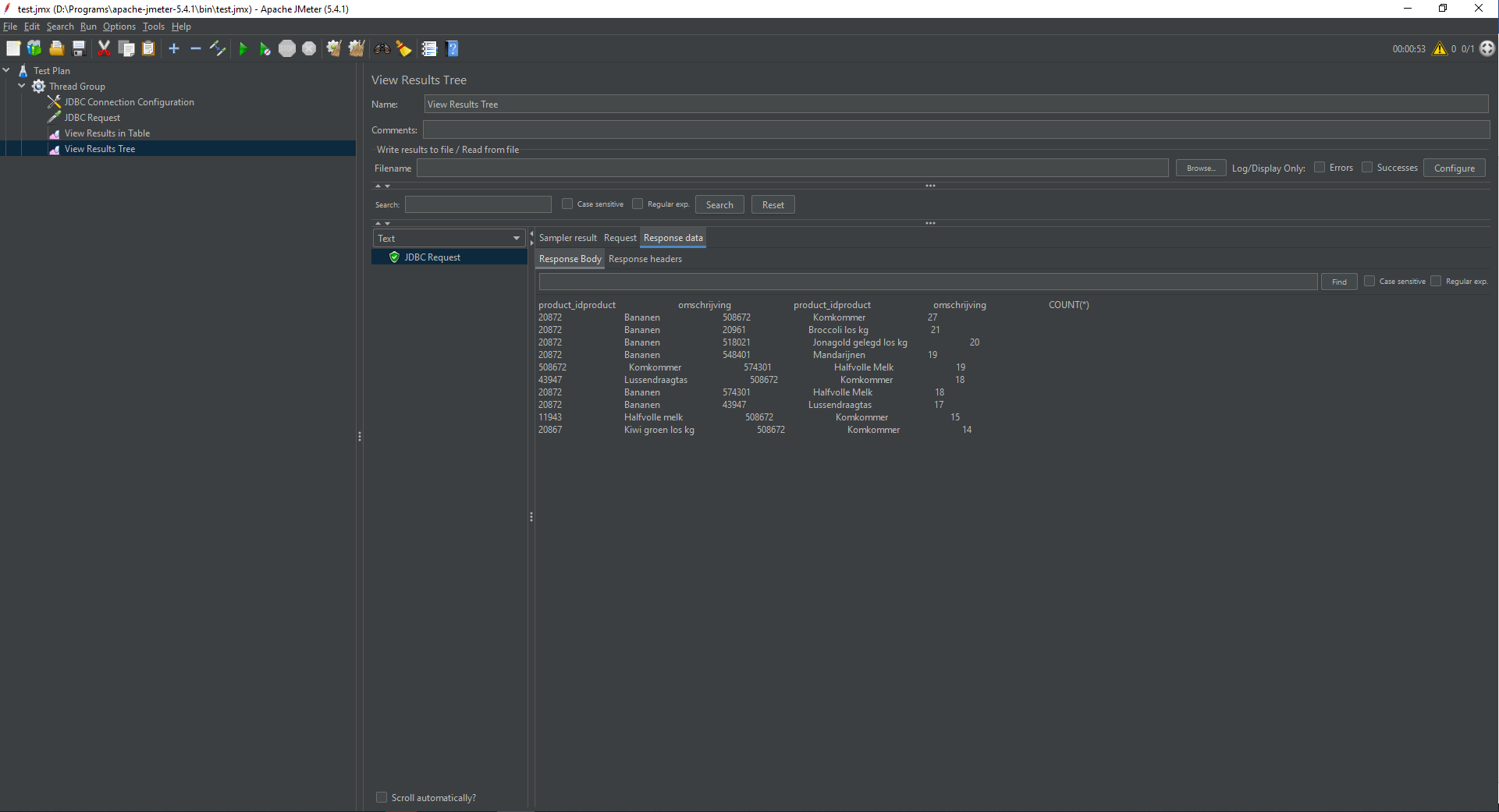
MySQL Workbench:



MySQL shell:



JMeter:



# Queries

Hierbij de query’s die ik gemaakt heb:

Query 1:

Deze was al gegeven als een opzetje, deze voorbeeld query heeft me goed op weg geholpen om de derde query op te lossen.

Query 2:

Deze is geschrapt en heb ik dus niet gemaakt.

Query 3:

SELECT p1.idproduct, p1.omschrijving, p2.idproduct, p2.omschrijving, COUNT(\*) FROM

(SELECT \* FROM product, aankoop WHERE product.idproduct = aankoop.product\_idproduct) AS p1,

(SELECT \* FROM product, aankoop WHERE product.idproduct = aankoop.product\_idproduct) AS p2

WHERE p1.datum = p2.datum AND p1.idproduct < p2.idproduct

GROUP BY p1.product\_idproduct, p2.product\_idproduct

ORDER BY COUNT(\*) DESC

LIMIT 10;

Query 4:

Deze heb ik niet gemaakt omdat mijn MySQL kennis niet gevorderd genoeg hiervoor was en door een andere reden die ik in het nawoord toelicht. Ik heb wel een stappenplan gemaakt over hoe je het moet oplossen maar hoe je het daadwerkelijk moet omzetten tot een query weet ik niet:

* Elke aankoop heeft een klant en een filiaal.
* Bereken het aantal stappen van filialen tussen elke klantenpaar.
* AIs klant a naar dezelfde filiaal als klant b gegaan dan is het aantal stappen 0.
* Als klant a naar de zelfde filiaal als klant b is gegaan en b dezelfde als c dan zit er 1 stap tussen a en b.

# Nawoord

Query 1 was gegeven, 2 hoefde we niet en 3 heb ik gemaakt. Jammer genoeg kwam ik niet op het antwoord van 4, ik zal nader toelichten wat ik geprobeerd heb. Dat het niet gelukt was komt deels doordat ik abnormaal lang moest wachten voor query’s. Bij mij doet mijn gemaakte query van Q3 er ~50 seconden over terwijl precies dezelfde query bij 2 mede klasgenoten ~2 seconden er over duurt met de testset data. Ik heb 4 dingen geprobeerd:

* Ten eerste mijn gebruikelijke manier via MAMP op localhost de MySQL server runnen en met phpMyAdmin de query’s uitvoeren.
* Ten tweede een clean install van MySQL gedaan en ook MySQL Workbench gedownload en met de ingebouwde sqlservice het lokaal runnen
* Ten derde het via MySQL Shell geprobeerd,
* En ten slot het via JMeter getest.

Allen gaven mij dezelfde resultaten. Ik vermoed dat het een hardware probleem is want ik heb in het verleden ook wat MySQL gewerkt en heb in dat proces misschien wat in de system files aangepast.

Ik hoop hiermee genoeg te hebben laten zien hoeveel tijd en moeite ik in deze opdracht heb gestoken, ik kijk uit naar de beoordeling bedankt.