Performantievergelijking van database-systemen
Casus onderzoeksproces

Stan Fieuws[1], Ewout Maertens[2], Chiel Sinnaeve[3], Maxim Eeckhout[4]

Samenvatting
Databasemanagementsystemen worden vandaag de dag het vaakst gebruikt om data te organiseren en up te daten. Hierdoor is het belangrijk om onderzoek uit te voeren om het juiste databasemanagementsysteem te kiezen. In dit artikel wordt onderzocht of MariaDB of MS SQL Server sneller is. Hiervoor werden dezelfde query's op dezelfde data uitgevoerd in de verschillende databasemanagementsystemen. Voor dit document werd er eerst een literatuurstudie gedaan op artikels gevonden op het internet. Dit wordt gevolgd door informatie over het uitgevoerde experiment. Uit het gevoerde onderzoek kan besloten worden dat MariaDB over het algemeen sneller werkt dan SQL Server. Dit onderzoek kan opgevolgd worden met een extra onderzoek dat query 7 en 8 opnieuw onderzoekt en uitvoert.

Sleutelwoorden
Database-beheer. Relationele databases — performantie

Contact: [1] stan.fieuws.y2178@student.hogent.be; [2] ewout.maertens.y1560@student.hogent.be; [3] chiel.sinnaeve.y1690@student.hogent.be; [4] maxim.eeckhout.y2249@student.hogent.be

Inhoudsopgave

1. Inleiding

Het doel van dit artikel is een onderzoek uit te voeren naar de performantie van verschillende relationele databasemanagementsystemen, dit onderzoek is nodig om in een bepaalde situatie het juiste databasemanagementsysteem te kiezen uit verschillende opties. De eerste stap van het onderzoek is een literatuurstudie, gevolgd door eigen experimenten, verder zullen de resultaten van de experimenten ook besproken worden. De literatuurstudie leverde in totaal zes artikels op. Deze zes artikels worden besproken in dit stuk, er wordt gekeken naar welke er al dan niet goed waren en hoe de artikels zijn in vergelijking met Bassil (2012). Verder zal er ook kritisch gekeken worden naar de resultaten van de artikels en de rapportering van de auteur over zijn resultaten. Er wordt in dit artikel steeds gesproken over databasemanagementsystemen maar hiermee worden echter relationele databasemanagementsystemen bedoelt.

1.1 Artikel 1

Het eerste artikel van Naik (2011) vergelijkt SQL met Oracle. Het artikel is in vergelijking met Bassil (2012) een stuk uitgebreider. Dit omdat alle databasemanagementsystemen eerst grondig worden uitgelegd. Zo wordt er een achtergrond, info over de query's, voordelen en nadelen gegeven van beide databasemanagementsystemen. Verder zijn de query's behoorlijk simpel in dit artikel in tegenstelling tot Bassil (2012) waar de laatste query's vrij complex worden. Ook wordt er in dit artikel meer aandacht besteed aan de andere commando's zoals update, delete door meerdere query's te vormen met deze commando's. Dit artikel geeft een degelijk beeld over de algemene performantie van beide databasemanagementsystemen en bewijst dit door grafieken uit te zetten per statement dat men getest heeft. Verder bekijkt men ook hoeveel geheugen de twee databasemanagementsystemen verbruiken. De auteur van dit artikel heeft de resultaten goed gerapporteerd. Steeds zijn de gebruikte statements en resultaten van de experimenten makkelijk terug te vinden. De auteur heeft ook steeds de resultaten duidelijk verwerkt in grafieken zodat de lezer makkelijk beide databasemanagementsystemen kan vergelijken. De experimenten zijn reproduceerbaar doordat de gebruikte statements gegeven zijn.

1.2 Artikel 2

Dit artikel van Reda (2013) is deels gebaseerd op Bassil (2012) en is dus minder waardevol omdat er geen eigen onderzoek gedaan is, ook heeft de schrijver een zeer uitgesproken mening over welke databasemanagementsysteem zijn/haar voorkeur heeft. Het doel van het artikel is ook niet het vergelijken van de performantie van databasemanagementsystemen maar wel aantonen waarom Oracle niet de geschikte keuze is voor meer dan 95% van de bedrijven in de Arabische wereld. Doordat de auteur zich baseerde op onderzoek van iemand anders kunnen we moeilijk zeggen dat dit artikel zelf een goed algemeen beeld geeft over performantie.

1.3 Artikel 3

Het volgende artikel bekijkt SQLite, MySQL en PostgreSQL. Per database worden de mogelijke datatypes gegeven, de voordelen, de nadelen en wanneer de database best wel of niet gebruikt wordt. Zo wordt een overzicht gegeven van welke database bij een bepaald project past. Dit artikel geeft een vergelijking tussen databases die voornamelijk dicht bij elkaar liggen. MySQL, SQLite en PostgreSQL liggen dichter bij elkaar dan de databasemanagementsystemen in Bassil (2012). Dit artikel is dus meer voor iemand die al beslist heeft om met SQL te werken. In dit artikel zijn er echter geen waardes en dus geen bewijzen, daardoor is Bassil (2012) veel beter. Het artikel presenteert ook geen goed beeld van de algemene performantie. Hier worden geen waardes getoond, noch een vergelijking van performantie met de verschillende databases. Het is volledig gebaseerd op de mening van de auteur, tot die met waardes kan bewijzen dat wat hij zegt effectief correct is. Doordat er geen duidelijke vermelding is van tijden en andere performantie waardes, zoals bijvoorbeeld CPU-gebruik, kunnen we concluderen dat de rapportering van de auteur niet goed is.

1.4 Artikel 4

Het artikel van Khawar, Kamran, S.A.K., Muhammad en Syed Asim (2017) vergelijkt dezelfde 5 databases als in Bassil (2012). Maar deze worden echter oppervlakkiger vergeleken. Er wordt nog dieper ingegaan op SQL Server 2016, Oracle 12c en MySQL Server, wat dus een verschil is. Ook worden er open-source databases vergeleken. Er zijn dus veel gelijkenissen met Bassil (2012), op vlak van testen tonen ze beide goed aan wat men kan verwachten van elke database. In het artikel wordt een heel goed beeld gegeven van performantie. Er wordt in een tabel informatie weergegeven en er zijn duidelijke figuren met de uitslagen van de testen die zijn gedaan. Er wordt nog dieper ingegaan op SQL Server 2016, Oracle 12c en MySQL Server 5.7. Van de testen op deze databasemanagementsystemen zijn er ook heel duidelijke figuren en de manier waarop de testen uitgevoerd zijn. De database is niet aanwezig, dus de waarden kunnen niet echt gecontroleerd worden. De auteurs hebben dit heel goed opgesteld. Ze geven eerst duidelijke informatie over al de mogelijke databases. Daarna geven ze een duidelijke vergelijking ertussen. Om af te ronden gaan ze nog eens dieper in op drie van die databases. Het is makkelijk om te volgen wat ze doen, hoe ze het doen en waarom ze het doen. Alles is dan ook nog eens duidelijk te volgen met figuren. Om af te sluiten hebben ze ook een goede conclusie gemaakt, waar alles nog eens mooi instaat.

1.5 Artikel 5

Het vijfde artikel van Abubakar (2014) vergelijkt populaire open source relationele databasemanagementsystemen. Deze zijn PostgreSQL, MariaDB, MySQL en SQLLite De vergelijking gebeurt aan de hand van de uitvoeringstijd van query's op 1000, 20.000, 40.000 en 100.000 en 1KB records. Dit artikel houd echter geen rekening met het gebruik van de CPU en het geheugengebruik van de query's, wat in Bassil (2012) wel gebeurd. Wat ook niet terug komt in het artikel van Abubakar (2014) zijn de gebruikte query's of de gebruikte database waardoor men niet kan verifiëren of de gestelde resultaten kloppen. De gebruikte diagrammen zijn onduidelijk. De plotlijnen lopen bij bepaalde punten dicht bij elkaar. Er zijn ook geen precieze waardes terug te vinden op de diagrammen. De auteur van het artikel sluit wel af met een duidelijke conclusie van wat hij uit zijn experiment geleerd heeft.

1.6 Artikel 6

Het laatste artikel van Amlanjyoti, Sherin, Dhondup en Roseline (2015) vergelijkt slechts twee databasemanagementsystemen op een Micorsoft besturingssysteem. De databases zijn de open source MySql en de betalende SQL Server van Microsoft. De testen gebeurden door middel van select, insert, update en delete query's te doen op databanken met 3000 en 5000 rijen. Bij dit experiment wordt de performantie enkel maar gemeten op basis van de uitvoeringstijd. Het artikel komt met een duidelijke conclusie dat SQL Server performanter is dan MySQL. Er zijn echter geen meegeleverde datasets en query's om deze resultaten zelf te controleren.

1.7 Algemeen

Uit deze zes artikels komt het vierde artikel als het meest interessantste naar voor omdat het goed opgesteld is en een duidelijk beeld geeft van performantie. Verder is het tweede artikel niet bruikbaar en het minst interessant voor dit onderzoek doordat de opzet voor het schrijven van dat artikel niet overeen komt met ons onderzoek.

Performantie is niet echt een eenduidig begrip omdat het afhankelijk is van een doelgroep. Een klein bedrijf zal vinden dat een databasemanagementsysteem dat makkelijker te gebruiken is performanter is dan een systeem dat sneller is. In dit onderzoek zullen we echter performantie sterk koppelen aan de uitvoeringstijd dat het nodig heeft. Maar het is belangrijk om te onthouden dat dit zeker niet het enige aspect is van performantie bij databasemanagementsystemen. Andere aspecten kunnen bijvoorbeeld gebruiksvriendelijkheid of de impact op de CPU zijn.

Verder zal in dit artikel het opgezette experiment beschreven worden in Sectie 2, de resultaten van het experiment besproken worden in Sectie 3 en een conclusie gevormd worden in Sectie 4.

2. Methodologie

Om het experiment reproduceerbaar te maken en om er voor te zorgen dat externe factoren zo weinig mogelijk invloed

hadden op de resultaten hebben we gebruik gemaakt van VirtualBox en Vagrant. De code en andere bijhorende bestanden zijn te vinden op Github[1]. De virtuele machine in het eerste deel van het experiment had CentOs 7.4 als besturingssysteem en had een RAM-geheugen van 2GB en één CPU. Via Vagrant werd MariaDB geïnstalleerd op deze virtuele machine. Na de installatie werd er gebruik gemaakt van een script om een honderdtal CSV-bestanden per query uit te schrijven in een dataset. Vervolgens werden de query's 3 en 7 van Bassil (2012) verbetert, ook query 8 had een fout maar deze kon niet verbeterd worden. Na het verbeteren van de query's kregen we de CSV-bestanden voor dit deel van het experiment.

In het tweede deel van het experiment werd gebruik gemaakt van Ubuntu als besturingssysteem ook hier had het besturingssysteem een RAM-geheugen van 2GB en één CPU, hiervoor werd ook een script geschreven dat alle nodige tools van SQL Server installeerde. Het script werd niet geautomatiseerd omdat er een setup doorlopen moest worden. Na de installatie verliep de opzet van het experiment stroef. Het aanmaken van tabellen en deze opvullen in de databank op de virtuele machine gaf veel problemen. Doordat de virtuele machine de gedeelde map met de host niet vond moest alles eerst lokaal op de machine gezet worden, vervolgens is er veel tijd gekropen in het onder de knie krijgen van het bulk insert command. Het gebruik van dit commando gaf ook veel errors waardoor ieder CSV-bestand aangevuld en veranderd moest worden. Zo moesten NULL waardes omgezet worden naar 0 en de datums een ander formaat krijgen doorheen alle data. Verder moesten floats die gebruik maakten van komma's ook aangepast worden naar floats waar punten gebruikt werden. Na al deze aanpassingen werd de database opgebouwd en via het script van het eerste deel van dit experiment kon de data gemakkelijk gegenereerd worden voor SQL Server (enkel het command veranderen om de query uit te voeren). Ook bij dit deel van het experiment konden we geen data generen voor query 8 van Bassil (2012).

3. Experimenten

Het uiteindelijke experiment bestaat uit 10 query's. Elk van deze query's wordt 100 maal uitgevoerd. Dit gebeurd zowel in SQL Server, als in MariaDB. Na de query's uit te laten voeren door het script was er een duidelijk verschil te zien tussen de 2 databanken. De gebruikte code in RStudio voor het analyseren van de data is te vinden op Github.

3.1 Query 1

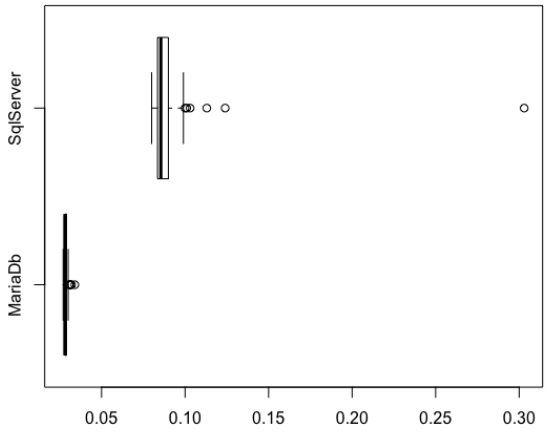
Deze query vraagt alle kolommen op uit een tabel van de database.

```
SELECT * FROM dbo.customer
```

3.2 Query 2

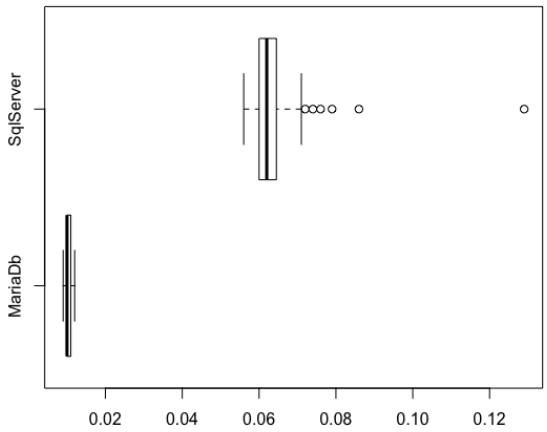
Deze query toont alle kolommen uit een tabel die voldoen aan de gegeven vereisten.

[1] https://github.com/HoGentTIN/ozt-dbperf-g23



Figuur 1. Query 1 - Boxplot van SQL Server en MariaDB

```
SELECT * FROM dbo.invoice
WHERE invoice.in_id > 50 AND
  invoice.in_date > '1-1-2006' AND
  invoice.in_date < '1-1-2007'
  AND invoice.in_description LIKE '%re%'
  AND (invoice.in_total <> 100 OR NOT
  invoice.in_cu_id >= 5 )
  AND (invoice.in_id BETWEEN 1 AND
  10000 OR invoice.in_id > 49+1)
  AND invoice.in_total+33 <> 5
```



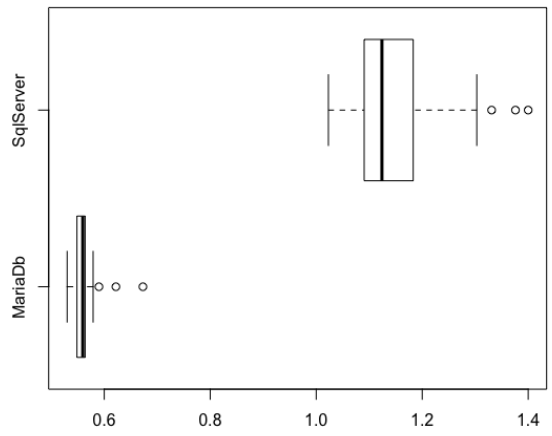
Figuur 2. Query 2 - Boxplot van SQL Server en MariaDB

3.3 Query 3

Deze query zoekt gegevens uit kolommen die voldoen aan de vereisten die verspreid zijn over verschillende tabellen in de database.

```
SELECT customer.cu_id, invoice.in_id,
  invoicedetail.ind_qty,
```

```
  item.it_serialnumber,
  movement.mo_descriptio,
  movement_details.mod_it_id,
  users.us_id,
  users.us_code,
  purchaseorder.po_description,
  supplier.su_name
FROM customer, invoice, invoicedetail,
  item,
  movement, movement_details, users,
  purchaseorder, supplier
WHERE supplier.su_name = "Mike"
  AND customer.cu_id = invoice.in_cu_id
  AND invoicedetail.ind_in_id =
    invoice.in_id
  AND invoicedetail.ind_it_id =
    item.it_id
  AND movement_details.mod_mo_id =
    movement.mo_id
  AND movement.mo_us_id =  users.us_id
  AND purchaseorder.po_us_id =
    users.us_id
  AND purchaseorder.po_us_id =
    users.us_id
  AND purchaseorder.po_su_id =
    supplier.su_id;
```

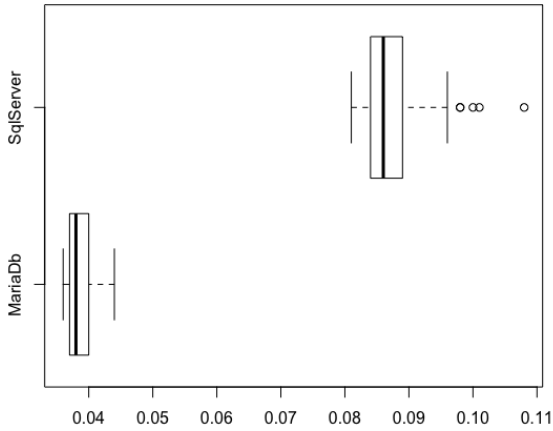


Figuur 3. Query 3 - Boxplot van SQL Server en MariaDB

3.4 Query 4

Bij deze query worden er gegevens uit een aantal kolommen gezocht van een tabel die uiteindelijk gesorteerd worden weergegeven.

```
SELECT customer.cu_id,
  customer.cu_name,
  customer.cu_telephone,
  customer.cu_fax,
  customer.cu_email
FROM customer
ORDER BY customer.cu_id,
  customer.cu_name DESC,
  customer.cu_telephone DESC,
  customer.cu_fax, customer.cu_email
    DESC;
```



Figuur 4. Query 4 - Boxplot van SQL Server en MariaDB

3.5 Query 5

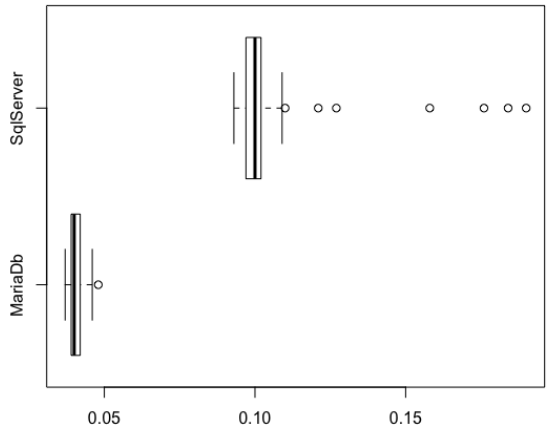
Bij deze query worden de resultaten van verschillende wiskundige berekeningen weergegeven. De gegevens komen dan ook nog eens uit verschillende tabellen van de database.

```
SELECT SUM(invoice.in_total),
  AVG(invoice.in_totalafterdiscount),
  MAX(invoice.in_total),
  COUNT(customer.cu_id),
  SUM(invoicedetail.ind_qty)
FROM customer, invoice, invoicedetail
WHERE customer.cu_id = invoice.in_cu_id
  AND invoice.in_id =
    invoicedetail.ind_in_id
GROUP BY invoice.in_id ;
```

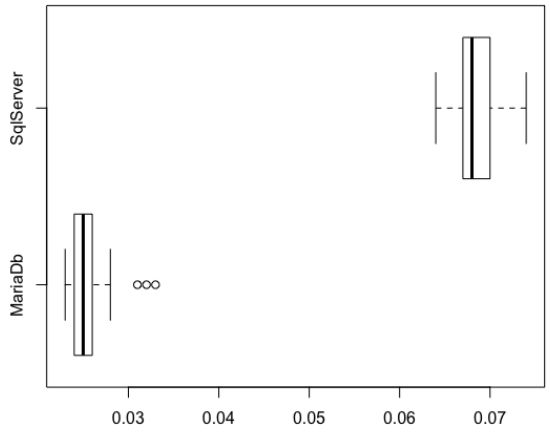
3.6 Query 6

In deze query worden een som en enkele gegevens uit kolommen van verschillende tabellen opgevraagd.

```
SELECT SUM(invoice.in_total)
FROM invoice
JOIN customer ON customer.cu_id =
  invoice.in_cu_id
GROUP BY customer.cu_id
HAVING COUNT(invoice.in_id) > 0
  AND SUM(invoice.in_total) <
  AVG(invoice.in_totalafterdiscount)
ORDER BY SUM(invoice.in_total);
```



Figuur 5. Query 5 - Boxplot van SQL Server en MariaDB



Figuur 6. Query 6 - Boxplot van SQL Server en MariaDB

3.7 Query 7

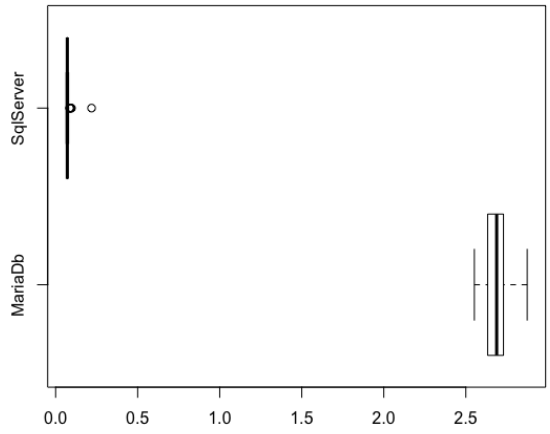
Het doel van deze query is om een kolom weer te geven met gegevens die voldoen aan vereisten verspreid over meerdere tabellen.

```
SELECT customer.cu_name
FROM customer
JOIN users ON users.us_name
  = customer.cu_name
JOIN supplier ON customer.cu_fax
  = supplier.su_fax
WHERE users.us_class like '%P%'
  AND
supplier.su_phone like '%1%'
```

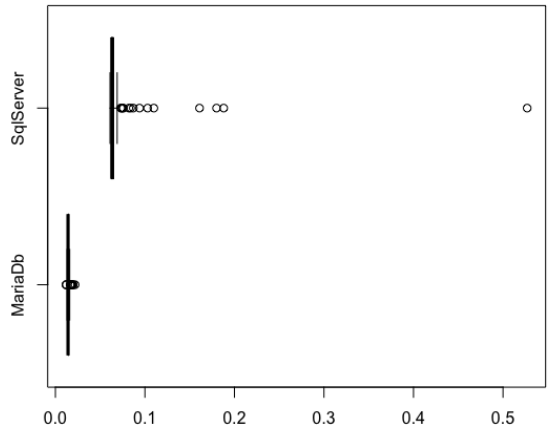
3.8 Query 9

In deze query worden de gegevens die aan de vereisten voldoen aangepast.

```
UPDATE item
SET item.it_price = (item.it_price
  * 0.1),
  item.it_qtity = 10,
  item.it_description = 'TV'
WHERE item.it_id > 10
  AND item.it_expirydate >
    '1-1-2007'
  AND item.it_expirydate <
    '1-1-2008'
```



Figuur 7. Query 7 - Boxplot van SQL Server en MariaDB



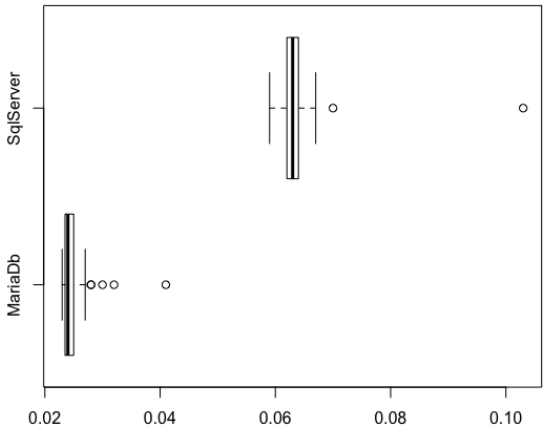
Figuur 8. Query 9 - Boxplot van SQL Server en MariaDB

3.9 Query 10

Bij deze query worden de gegevens die met de opgegven vereisten overeen komen uit een tabel verwijderd.

```
DELETE FROM invoicedetail
WHERE invoicedetail.ind_id
  IN (SELECT in_id FROM invoice
    WHERE invoice.in_description
```

```
    LIKE '%t%' )
AND invoicedetail.ind_id
  IN (SELECT in_id FROM invoice
    WHERE invoice.in_date >
      '1-1-2006' )
AND invoicedetail.ind_id
  IN (SELECT in_id FROM invoice
    WHERE invoice.in_date <
      '1-1-2007' )
```



Figuur 9. Query 10 - Boxplot van SQL Server en MariaDB

3.10 Query 11

Door deze query worden al de gegevens weergegeven uit de tabellen die de aan de juiste vereisten voldoen.

```
SELECT *
FROM category, item
WHERE item.it_ca_id = category.ca_id
```

4. Conclusie

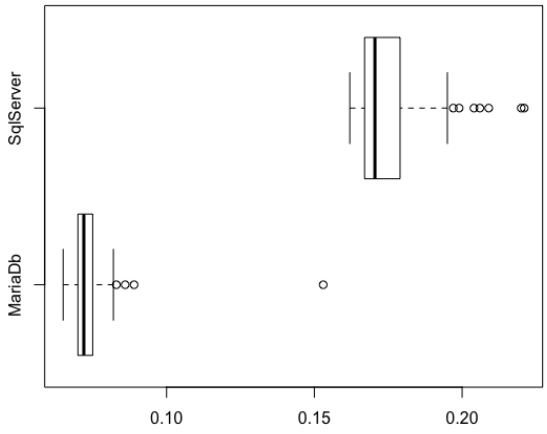
Na alle experimenten uit te voeren en de resultaten met elkaar te vergelijken is er een duidelijk verschil te zien tussen de databasesystemen. MariaDB is in 9 van de 10 query's sneller dan SQL Server.

De tijdsverschillen zijn ook merkwaardig groot. De tijden van SQL Server zijn heel vaak zelfs dubbel zo groot als de tijden van MariaDB.

Uit de figuren van de experimenten kan er ook besloten worden dat MariaDB een constantere uitvoeringstijd heeft. De resultaten zijn vaak veel beter geconcentreerd dan die van SQL Server.

Er is één query waarin SQL Server sneller is. Maar er is een vermoeden dat dit fout is. Dit sluit ook niet aan bij de resultaten die werden verwacht. In Figuur 7 is er duidelijk te zien dat deze resultaten bijna het tegenovergestelde zijn van de rest.

Uit dit onderzoek is dus gebleken dat MariaDB een beter relationeel databasesysteem is dan SQL Server.



Figuur 10. Query 11 - Boxplot van SQL Server en MariaDB

Referenties

Abubakar, Y. (2014). Benchmarking popular open source rdbms: A performance evaluation for it professionals. *IJACT*, *3*, 39. Verkregen van https://www.researchgate.net/publication/316132763_BENCHMARKING_POPULAR_OPEN_SOURCE_RDBMS_A_PERFORMANCE_EVALUATION_FOR_IT_PROFESSIONALS_BENCHMARKING_POPULAR_OPEN_SOURCE_RDBMS_A_PERFORMANCE_EVALUATION_FOR_IT_PROFESSIONALS

Amlanjyoti, S., Sherin, J., Dhondup, D. & Roseline, M. R. (2015). Comparative Performance Analysis of MySQL and SQL Server Relational Database Management Systems in Windows Environment. *IJACT*, *4*. Verkregen van https://www.ijarcce.com/upload/2015/march-15/IJARCCE%2039.pdf

Bassil, Y. (2012). A Comparative Study on the Performance of the Top DBMS Systems. *Journal of Computer Science and Research*, *1*(1), 20–31.

Khawar, I., Kamran, A., S.A.K., B., Muhammad, S. & Syed Asim, A. (2017). *Huge and Real-Time Database Systems: A Comparative Study and Review for SQL Server 2016, Oracle 12c & MySQL 5.7 for Personal Computer* (proefschrift, Department of Computer Science, Federal Urdu University, Karachi, Pakistan). Verkregen van http://www.lifescienceglobal.com/pms/index.php/jbas/article/viewFile/4898/2784

Naik, M. (2011). *Database Management System Performance Analysis and Comparison* (masterscriptie, California State University, Sacramento). Verkregen van http://csus-dspace.calstate.edu/bitstream/handle/10211.9/1316/DBMS_Report.pdf?sequence

Reda, M. A. (2013, juli 14). Microsoft SQL Server VS Oracle Technical Study. Verkregen van https://www.slideshare.net/SoftexSoftware/sql-server-vs-oracle-dbms-comparison-24215631