Performantie verschillen bij het gebruikt van subqueries, aggregate functions en indexes

Paper Research Methods: onderzoeksvoorstel 2021-2022

Tuur De Schepper[1], Xeo Gillis[2]

Samenvatting
Deze paper is opgesteld rond de vraag: 'Wat is de meest performante manier om een query te schrijven'. Hierbij wordt er rekening gehouden met indexes, subqueries, joins en aggregate functies. Om effectief een antwoord te vinden zal er rekening gehouden worden met verschillende types databanken. Het hoofddoel is om duidelijk te stellen dat één juist antwoord niet bestaat op de vraag en dat het antwoord varieërt van databank tot databank.

Sleutelwoorden
Databanken en big data; MySQL; performantie

[1] tuur.deschepper@student.hogent.be
[2] xeo.gillis@student.hogent.be

Inhoudsopgave

1. Inleiding

In dit onderzoek zal getracht worden om het verschil in performantie van verschillende queries in MySQL aan te tonen en om zo de snelste schrijfwijze te achterhalen. De keuze voor dit onderzoek werd bepaald op basis van de steeds groeiende data en de nood aan snelle toegang tot deze data.

Er zal getest worden op verschillende types queries, zoals een query met één of meerdere join-statements. Ook zullen subqueries, ingebouwde functies en indexes onder de loop genomen worden. Bij het uitvoeren van deze queries zal er enkel rekening gehouden worden met de performantie, en dus de snelheid van uitvoeren, niet met het geheugengebruik.

Bij het testen van subqueries en aggregate functions zal er gevarieerd worden in de positie van de eerdergenoemde subquery of function. Zo zal bijvoorbeeld de subquery niet enkel voorkomen in de WHERE clause, maar zal die ook in de SELECT en FROM clause gebruikt worden. Analoog zullen aggregate functions voorkomen in de SELECT, WHERE, HAVING en ORDER BY clause.

Finaal wordt er ook getest op de invloed van indexes op queries. Hierbij kan gesteld worden dat de SELECT en WHERE clause een invloed zullen hebben op de performantie, en daar zal dus de focus op liggen.

Aan de hand van de resultaten zal een advies opgesteld worden om efficiëntere queries te schrijven. De te beantwoorden vragen in dit onderzoek zijn:

- Wat is de invloed van subqueries op queries?
- Wat is de invloed van aggregate functies op applicaties?
- Wat is de invloed van indexes op queries?

2. Overzicht literatuur

De drie onderzoeksvragen vereisen logischerwijze drie alinea's met daarin een kort overzicht per vraag. De eerste vraag gaat over het idee dat subqueries de performantie van een MySQL statement zouden kunnen vertragen. Dit vertragen wordt vergeleken met een 'normale' statement zonder subqueries. Volgens Fritchey (2016) is het zo dat een subquery niet per definitie vertragend werkt en tegelijk ook niet per definitie versnellend werkt. Het slagen of falen van een subquery ten opzichte van een 'gewone' query geeft niets te maken met subqueries en alles met de kennis en ervaring van de persoon die de subquery schrijft. Echter wordt er vanuit een ander standpunt gezegd dat joins altijd moeten gekozen worden boven subqueries omdat joins sneller zouden werken. Een subquery moet enkel gebruikt worden als het zonder onmogelijk is om de data te verzamelen (Gravelle, 2021).

Naast subqueries bestaan er ook aggregate functies die kunnen worden gebruikt om data te verzamelen en zo door te sturen. De meest voor de hand liggende functies zijn COUNT(), SUM(), AVG(), MIN() en MAX(). Het voordeel van deze functies is dat ze respetievelijk het aantal, de som, het gemiddelde, het minimum of het maximum van een geselecteerde kolom teruggeven als resultaat. Indien u dit moet doen via de applicatie in plaats van in de queries zelf veroorzaakt dit een slechtere performantie (Zanini, 2022). In het werk van Bassil (2012) worder verschillende queries vergeleken op performantie en geheugengebruik. In query vijf en zes wordt er vergeleken wat het verschil in performantie is wanneer deze aggregate funties wel of niet worden gebruikt in deze queries.

Naast de queries zijn er natuurlijk nog indexes. MySQL zal waar kan de meest performante indexes gebruiken om de query zo snel mogelijk uit te voeren. In dit opzicht is het natuurlijk belangrijk dat er goede indexes worden gekozen.

Ahmad Yaseen vergelijkt verschillende indexes met dezelfde queries op basis van de 'logical reads', CPU-tijd en de totale uitvoeringstijd. Deze indexes werken niet altijd versnellend zoals Ahmad ook vermeld. Soms kiest MySQL ook voor een ander uitvoeringsplan dat de index niet gebruikt. De queries worden best geschreven zodat MySQL ook daadwerkelijk gebruik maakt van deze indexes (Yaseen, 2018).

3. Methodologie

Om het onderzoek tot een succes te laten leiden zullen er verschillende stappen moeten genomen worden om de juiste resultaten te verzamelen. Eerst en vooral moeten er meerdere verschillende types databanken opgesteld worden met verschillende indexes. Zo zou het ideaal zijn moest het aantal tabellen, rijen en kolommen variëren van databank tot databank om zo per type een accuraat advies te kunnen geven.

Na het maken van de juiste databanken moeten er uiteraard ook queries uitgedacht worden om de vergelijkende studie te starten. Zo is het idealiter dat elk resultaat kan bekomen worden met verschillende queries en dat zo het performantieverschil kan vastgesteld worden. Deze stap wordt bij aggregate functies vervangen door het opstellen van een API waarbij enerzijds enkel queries worden gebruikt en anderzijds queries worden gebruikt om alle data te verzamelen en dat de API zelf verantwoordelijk is voor het groeperen van de data via JavaScript.

Als derde en laatste stap moet het experiment nog uitgevoerd worden. Doordat één test doen zelden of nooit een goed idee is, zal er sprake zijn van een simulatie waarbij de queries honderden tot duizenden keren worden uitgevoerd om zo tot een eerlijke conclusie te komen.

4. Verwachte conclusies

Het verwachtte resultaat is dat er kan vastgesteld worden dat de performantie verschilt van databank tot databank, van query tot query en van index tot index. Zo zal een subquery enkel relevant zijn als het zo een groot aantal joins kan ontwijken. In alle andere gevallen zullen joins een betere performantie veroorzaken. Deze verwachte conclusie komt enerzijds uit ervaring en anderzijds uit het artikel van Gravelle (2021).

Verder wordt er verwacht dat het gebruik van indexes het proces ten allen tijde zal versnellen, maar dat de query wel moet aangepast worden naar deze indexes. Als er te veel verschillende kolommen nodig zijn om de query uit te voeren zal een index de performantie niet kunnen verhogen. Opnieuw kan er dus gesteld worden dat de kennis van de databank van vitaal belang is om de databank te gebruiken.

Als derde en laatste individuele conclusie zal het zo zijn dat aggregate functies altijd sneller zullen werken dan het gebruik van JavaScript in de API, dit kwam ook aan bod in het onderzoek van Zanini (2022).

Algemeen is de conclusie dat kennis het sterkste wapen is dat er kan gebruikt worden bij queries. Als de persoon die de databank gebruikt echt begrijpt hoe de databank in elkaar zit, dan zal die ten allen tijde de meest performante oplossing kunnen vinden door efficënt gebruik te maken van indexes, subqueries, joins en aggregate functies.

Referenties

Bassil, Y. (2012, februari 1). *A Comparative Study on the Performance of the Top DBMS Systems* (onderzoeksrap. Nr. 957). Lebanese Association for Computational Sciences. https://doi.org/https://doi.org/10.48550/arXiv.1205.2889

Fritchey, G. (2016, oktober 24). *A SUB-QUERY DOES NOT HURT PERFORMANCE: The things you read on the internet, for example, dont use a sub-query because that hurts performance.Truly?Where do people get these things?* Verkregen 11 april 2022, van https://www.scarydba.com/2016/10/24/sub-query-not-hurt-performance

Gravelle, R. (2021, februari 18). *Joins versus Subqueries: Which is faster?* Verkregen 11 april 2022, van https://www.navicat.com/en/company/aboutus/blog/1704-joins-versus-subqueries-which-is-faster

Yaseen, A. (2018). Tracing and tuning queries using SQL Server indexes. (9). Verkregen 11 april 2022, van https://www.sqlshack.com/tracing-and-tuning-queries-using-sql-server-indexes/

Zanini, A. (2022). Improving performance with SQL aggregate functions: SQL aggregate functions helped me take the performance of my backend application to the next level and avoid the bottlenecks that were slowing it down. *arctype*. https://arctype.com/blog/sql-aggregate-functions/