# MongoDB vs MySQL

## MySQL – basis

MySQL is een managementsysteem voor relationele databases. Ze werd in 1995 ontwikkeld door MySQL AB en later overgenomen door Oracle Corporation. Ze is ontwikkeld met gebruik door internettoepassingen op het oog. MySQL is compatibel met en gratis beschikbaar voor alle gangbare besturingssystemen. Informatie wordt opgeslagen in tabellen met rijen en kolommen. De community edition (MySQL CE) wordt gezien als het meest gebruikte databasesysteem ter wereld.

## MongoDB – basis

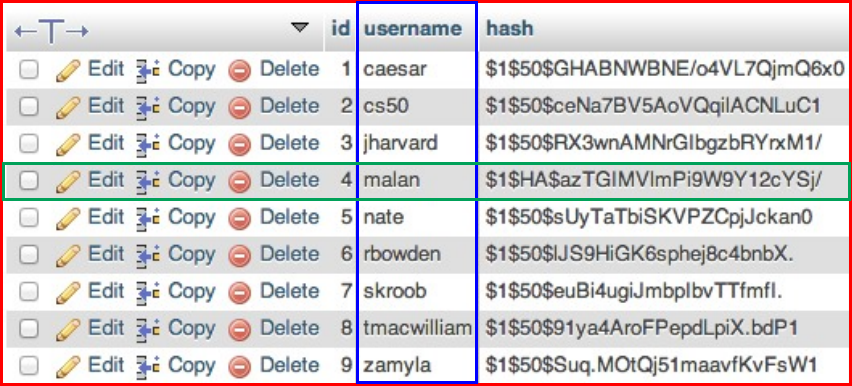
MongoDB is een NoSQL databasesysteem, ontwikkeld door MongoDB, Inc sinds 2007 met hun eerste openbare versie in 2009. Het is volledig open-source beschikbaar op GitHub. MongoDB wordt vooral gebruikt voor big-data en real-time toepassingen. Informatie wordt opgeslagen in collecties met documenten en velden. Ook MongoDB is compatibel met alle gangbare besturingssystemen, maar wordt vooral op UNIX-systemen gebruikt.

## Jargon

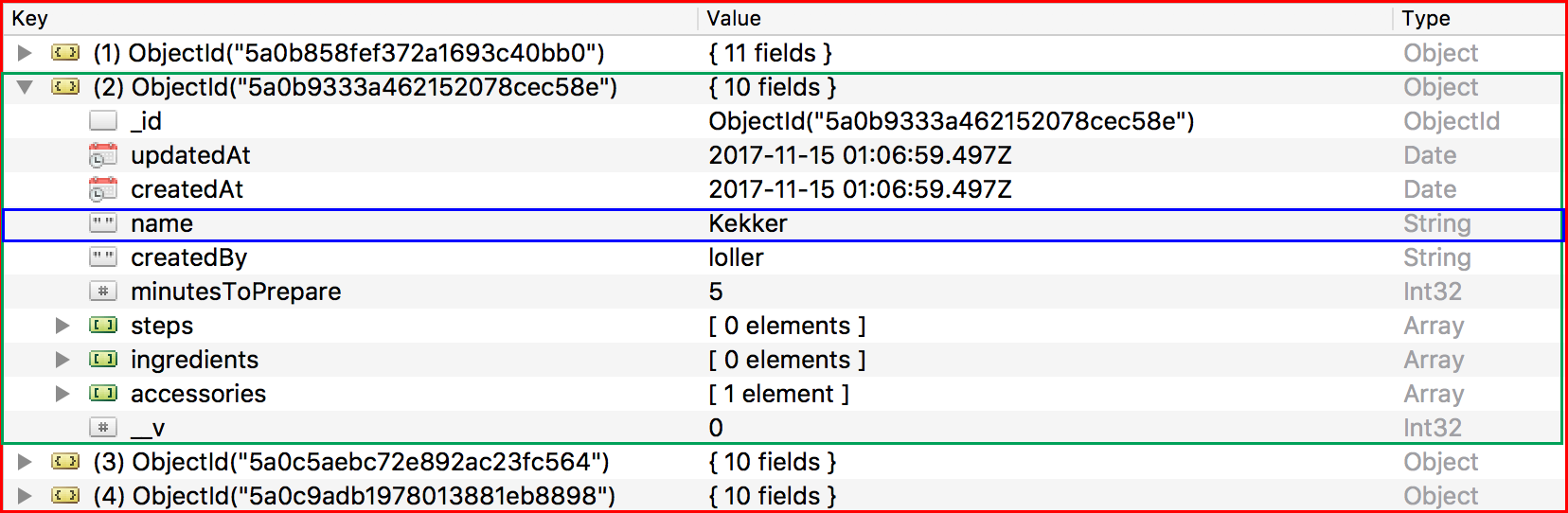
MySQL en MongoDB gebruiken andere terminologie voor gelijkende concepten. De meest voorkomende vindt u hier terug, met nadien een uitleg:

Wat men in MySQL een *Table* of *Tabel* noemt, heet in MongoDB een *Collection* of *Collectie*. Een *Row* of *Rij* wordt een *Document*. Ten laatste krijg de term *Column* of *Kolom* bij MongoDB de noemer *Field* of *Veld*. Hier vindt u nog een korte uitleg over elke term, aan de hand van foto’s:

MySQL



MongoDB



## Schaalbaarheid

MongoDB staat bekend voor zijn gebruik in big-data toepassingen. Dit wilt ook zeggen dat schaalbaarheid redelijk belangrijk is: als de hoeveelheid data de capaciteit van de server dreigt te bereiken, moet hier een oplossing voor zijn. Vaak is de eerste oplossing, zowel bij MySQL als bij MongoDB, vertical scaling. Dit doe je door meer kracht aan de database, of de server waarop die draait, te geven. Zo kan je de server krachtiger maken door er bijvoorbeeld een extra of een betere CPU bij te steken of de hoeveelheid RAM op te waarderen.

Vertical scaling heeft echter een limiet: een server kan maar zo krachtig worden voor hij niet meer verder kan. Wanneer je dit punt bereikt, ga je over op horizontal scaling: meer servers installeren, en de database over deze verschillende servers simultaan laten lopen.

Vertical scaling in MySQL is relatief complex en moet met het oog op het type data gebeuren. Vertical scaling in MongoDB is vrijwel ingebouwd door middel van *sharding*. Bij sharding wordt de database opgesplitst op basis van een index. Als je een database hebt met records met een id van 1-1000, kan deze bijvoorbeeld opgesplitst worden in 2 databases van 1 tot 500, en van 501 tot 1000. Als de applicatie of user document 300 nodig heeft, spreekt hij de eerste database aan enzovoort.

## Querytaal

Bij MySQL worden database opgemaakt, data opgehaald en bewerkt door middel van de taal SQL. MongoDB gebruikt een querytaal die gebaseerd is op JavaScript. Als ik de gegevens van de gebruiker met naam *Adriaan* wil ophalen uit een database, zou ik volgende code gebruiken:

MySQL

SELECT \* FROM users WHERE name = ‘Adriaan’;

MongoDB

db.users.find( { name : ‘Adriaan’ } );

## Benchmarks (vanuit het opzicht van node.js)

Als je databasesystemen vergelijkt, mogen niet enkel de definities, het gebruik en de taal vergeleken worden, maar ook de snelheid en performance. Daarom schrijf ik enkele benchmarking tests die twee systemen in de praktijk tegen elkaar afwegen.

Ik schrijf deze tests in node.js met gebruik van de npm packages mysql voor MySQL en mongoose voor MongoDB. Lees- en schrijfsnelheden kunnen dus door het platform en deze packages beïnvloed worden, maar ik verwacht dat de tests toch nuttige informatie zullen opleveren, vooral voor node.js ontwikkelaars, maar ook voor ontwikkelaars die niet met node.js werken. Node.js is een redelijk goed platform om deze tests op te draaien, omdat ze alle beschikbare resources van de machine probeert te gebruiken, als ze niet gelimiteerd is.

De tests draaien op dezelfde machine, met dezelfde achtergrondprocessen draaiende (Microsoft Word, GitHub Desktop, Visual Studio Code, Slack, Spotify). De twee databases zullen allebei lokaal draaien, om foute voorstellingen in de resultaten door internet-latency te vermijden.

De specificaties van de machine zijn als volgt:

MacBook Pro (13-inch, 2016)  
CPU: 2,9 GHz Intel Core i5  
RAM: 8 GB 2133 MHz LPDDR3

### Test\_1 – Schrijfsnelheid: 1.000.000 documenten/rijen toevoegen

Ik schrijf een script dat nagenoeg dezelfde (randomised) data zowel naar een MongoDB als naar een MySQL database schrijft. Ik maak een Mongoose schema aan voor MongoDB, en creëer een MySQL database die voor dezelfde data past.

### Test\_2 – Leessnelheid:

Hoeveelheid > x

Kleur = x

Sorteren op hoeveelheid

…

Tests:

1. MongoDB vs MySQL 1.000.000 documenten toevoegen
2. MongoDB vs MySQL documenten ophalen uit een grote database
3. MongoDB (met indexering) vs MySQL documenten ophalen uit een grote database