1. **Inleiding**

Naast relationele databanken, die gestructureerde gegevens opslaan, bestaan er nog andere type gegevens:

* RFID-scans
  + Radio Frequency IDentification
    - Een methode om van een afstand informatie op te slaan en te lezen van zogenaamde RFID tags
      * Bv. Magazijnbeheer van pakketten goederen
* Sensordata
  + Data afkomstig van apparaten die voorzien zijn van 1 of meerder sensoren
    - Meetapparatuur voor:
      * Lucht
      * Water
      * Gas
      * …
* Clickstreamdata
  + Onthoudt waar je geklikt hebt met additionele informatie zoals
    - Webbrowser
    - Router
    - Proxy
    - …

Deze gegevens opslaan in een databank voor zowel gestructureerde als ongestructureerde gegevens.

1. **Terminologie**

Een transactie in een database

* Nieuw gegeven toevoegen, bestaand gegeven wijzigen of verwijderen, acties/taken die gelijktijdig uitgevoerd moeten worden = DML statements
* Verloop:
  + Begin de transactie
  + Voer de statements uit
  + Geen errors: committen en eindigen transactie
  + Errors: roll back en eindigen transactie

Database/gegevensbank/databank

* Een digitaal opgeslagen archief, ingericht met het oog op een flexibele dataopslag, raadpleging en gebruik. Ze spelen een belangrijke rol voor het archiveren en actueel bijhouden van gegevens voor elk bedrijf, overheid en financiële instelling.

Het woord database wordt voor verschillende begrippen gebruikt:

* De opgeslagen gegevens als zodanig
* De wijze waarop de gegevens zijn opgeslagen, zie datamodel;
* De software waarmee databases kunnen worden aangemaakt en benaderd, zie databasemanagementsystemen (DBMS)

Datawarehouse

* Gegevensverzameling die in een zodanige vorm gebracht is dat terugkerende en ad-hoc vragen in relatief korte tijd beantwoord kunnen worden, zonder dat de bronsystemen zelf daardoor overmatig belast worden.
  + Verschilt van een standaard database omdat de betreffende gegevens afkomstig zijn van de bronsystemen en op een geautomatiseerde manier worden onttrokken ervan.
  + Data wordt omgezet naar informatie
    - Wordt gedaan door ETL-programmeur
    - ETL (Extract, Transform, Load)
      * Extract: De gegevens worden uit het transactiesysteem gehaald
      * Transform: De omzetting van ‘data’ naar ‘informatie’
      * De actie waarbij de informatie worden doorgeschoven naar het informatiesysteem voor opslag.
  + Gegevens kunnen in een datawarehouse niet worden ingevoerd op aangepast door de gebruiker zelf

Datamining

* Op basis van relevante gegevens zoeken naar (statistische) verbanden in gegevensverzamelingen met als doel profielen op te stellen voor wetenschappelijk, journalistiek of commercieel gebruik.
  + Op basis hiervan kunnen dan onderbouwde beslissingen genomen worden
    - Is er een verband tussen de leeftijd van de klant en het type shampoo?
    - …
* CRISP-DM : Cross Industry Standard Process for Data Mining
  + Stap 1: Business understanding
    - Criteria opstellen van wat men wilt bereiken, zodat men later zinnige en onzinnige resultaten van elkaar kan scheiden
  + Stap 2: Data Understanding
    - Kennis van informatie vergaren waaruit o.a. de betrouwbaarheid en de strekking van de gegevensinformatie bepaald kan worden
  + Stap 3: Data Preparation
    - De kwaliteit en de volledigheid van de data controleren en zo nodig aan passen of verbeteren
  + Stap 4: Modeling
    - Informatie wordt onderzocht
  + Stap 5: Evaluation
    - De controle van de resultaten van de modelingsstap tegen de informatie en nagaan of de resultaten de gewenste antwoorden zijn volgens de criteria van stap 1
  + Stap 6: deployment
    - Het toepassen van de uitkomsten van de datamining binnen de zakelijke omgeving

Er zijn andere processen maar er is altijd een voorbereidings- modelings- en verwerkings/evaluatie stap voor de gegevens.

Meestal duurt het even voor er zinnige resultaten verschijnen, parameters worden elke keer bijgesteld.

1. **Big Data**
   1. **Big Data bestaat al jaren**

Hoofdzaak: evolutie in data en op een andere manier omgaan met data.

In de jaren 50:

* Basisanalyses, gegevens in tabellen steken en dan manueel berekenen

Nu:

* Spreadsheets en databasetoepassingen

Doel:

* Met analyses van data beslissingen nemen voor de toekomst
  + BI Oplossingen (Business Intelligence)
  1. **Waarom is Big Data de laatste jaren een hype geworden?**

Redenen voor de hype:

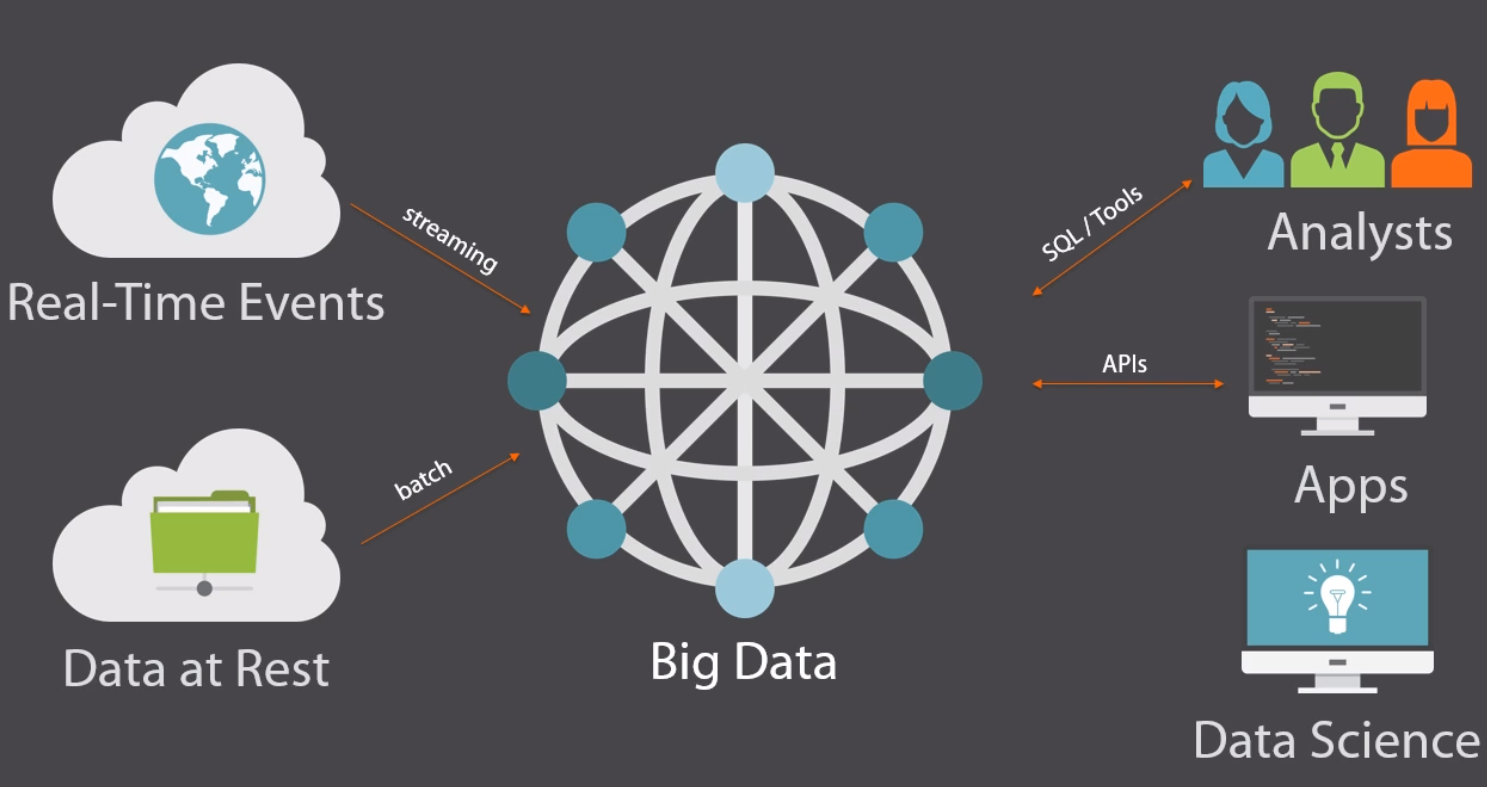
* Hedendaagse hardware-mogelijkheden
* Goedkope servers
* Goedkopere en ruimere opslag
* Mogelijkheden van opensource software
* Beschikbaarheid van massa’s gegevens

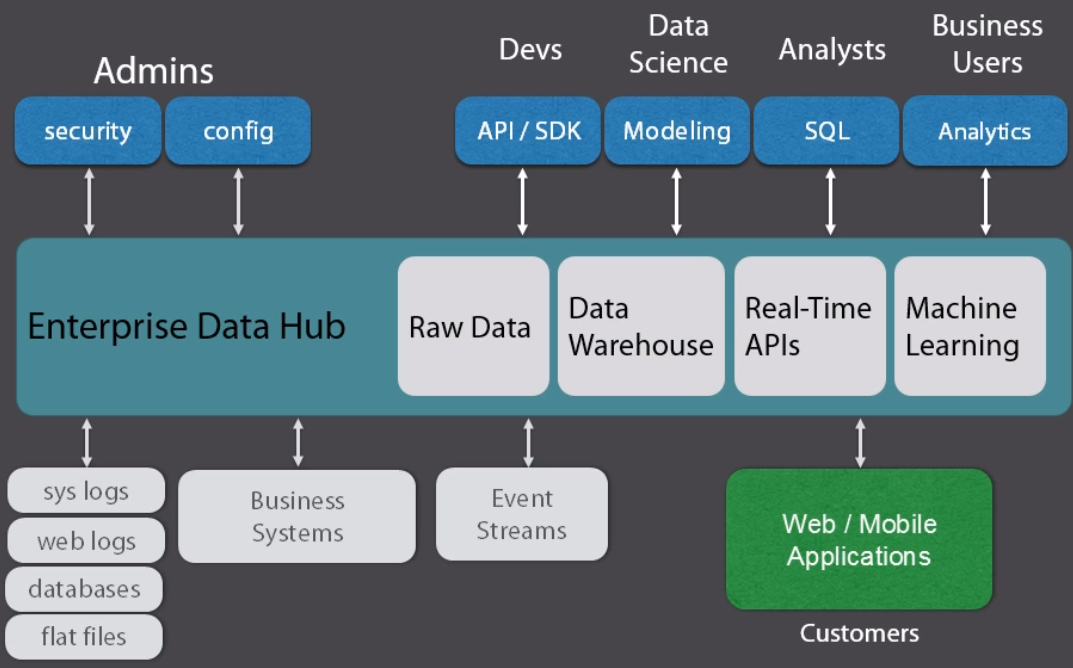
Toepassingen van Big Data:

* Marketing
  + Web-based bedrijven (Facebook, Twitter, Instagram, …)
* Politie- en opsporingsonderzoek
  + Fraude
  + Cybercrime
* Analyse bij grote datalekken
  + WikiLeaks
* Onderzoek in de gezondheidssector
  + Onderzoek naar ziektes en erfelijkheid
* Industrie
  + Verbeteren van technologie
  1. **Wanneer spreken we van Big Data**

Er zijn verschillende situaties wanneer we spreken over big data:

* Honderden terabytes
* Er is nood aan het opdelen van de data in kleinere eenheden
* Een ‘Klassieke’ databank kan de complexiteit niet meer aan. Een alternatief is nodig voor de niet-rationele gegevens
* De 5 V’s die aanleiding kunnen geven tot het spreken over Big Data
  + Volume:
    - Een grote hoeveelheid gegevens, teveel om te verwerken op een traditionele manier
  + Velocity
    - Een hoge snelheid van datatoevoer en een hoge verwerkingssnelheid
  + Variety
    - De structuur van de gegevens varieert en is niet vast zoals bij een relationele database
  + Veracity
    - Betrouwbaarheid van gegevens
  + Value
    - De info die gegenereerd wordt vanuit de big data moet waardevol zijn
  1. **Hoe werkt Big Data**
     1. **Architectuur**





* + 1. **Distributed Systems met verschillende nodes**

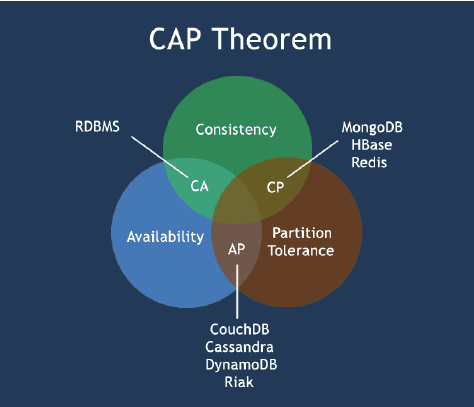


1 grote node = 1 point of entry.

Dit is een probleem omdat een systeem altijd moet werken, maakt niet uit wat er gebeurt. Daarom de load verdelen over meerdere kleinere systemen. Zodat als er een node uitvalt het systeem nog altijd werkt.

* + 1. **CAP-stelling**

Heel belangrijk als je gebruik maakt van Distributed File Systems. De CAP stelling zegt dat je altijd alleen maar kan voldoen aan 2 van de 3 belangrijke elementen van een database



* + 1. **CAP – 3 belangrijke elementen van een database**
* Consistency
  + De mate waarin het DB-systeem aan alle betrokken servernodes de meest recente gegevens laat zien
  + Is optimaal als na het uitvoeren van een operatie/transactie alle gebruikers dezelfde gegevens zien op alle betrokken servers/nodes
* Availability
  + Is de mate waarin het DB-systeem beschikbaar is
  + Is optimaal wanneer het systeem altijd beschikbaar(online) is en de gebruiker altijd een respons krijgt
* Partition tolerance
  + Is de mate waarin het DB-systeem blijft werken bij uitval van 1 of meerdere server(s)/node(s)
  + Blijft het systeem doorwerken als er individuele onderdelen/verbindingen uitvallen?

Bij relationele databanken

* CA is het belangrijkste (Consistency -Availability)
  + Er wordt hoofdzakelijk gewerkt met operationele gegevens die constant consistent en bereikbaar moeten zijn
* MySQL, MariaDB, Oracle SQL, PostgreSQL, Microsoft SQL Server

Bij NoSQL Databanken:

* CP (Consistency – Partition tolerance)
  + Sommige data is niet direct bereikbaar, maar de data is wel consistent en de data mag verspreid zijn over meerdere servers
* AP (Availability – Partition Tolerance)
  + Het systeem is beschikbaar maar niet altijd volledig.
  + Vooral nuttig bij de analyse van Big Data
    - Consistency speelt een mindere rol
    - Data is verdeeld over verschillende nodes

Belangrijke keerzijde aan Big Data:

* Privacy
  + Willen we dat iedereen al onze gegevens kan weten?

1. **Database principes**

Er zijn 2 belangrijke database principes

* 1. **ACID**

ACID = een transactie in een database moet atomic, consistent, isolated en durable zijn

* Atomic
  + Elke transactie slaagt volledig of niet. Indien er een error optreedt is er een rollback voor de hele transactie
* Consistent
  + Niet hetzelfde als consistency bij CAP
  + Gegevens mogen niet tegenstrijdig worden. Er moet op elk moment aan alle constraints voldaan zijn.
  + Als een bepaald statement uit een transactie de gegevens tegenstrijdig maakt, dan wordt deze transactie afgebroken (atomic)
* Isolated
  + Elke transactie wordt los van elkaar uitgevoerd
    - Dus transacties die tegelijkertijd worden uitgevoerd kunnen elkaars tussenresultaten niet zien
* Durable
  + Als de transactie voltooid is, dan is dit permanent
  1. **BASE**

ACID principe is niet houdbaar voor alle hedendaagse toepassingen (NoSQL databanken)

* BASE-systeem in NoSQL databanken

Consistent zijn van de databank wordt voor een deel opgegeven

* Basic availability
  + Systeem garandeert availability zelfs bij het uitvallen van meerdere nodes/servers.
* Soft state
  + Het systeem kan in de tijd wijzigen, zelfs zonder input van data.
  + Een basis concept van het BASE-Systeem is dat data consistency een probleem is voor de developer en niet van de database
* Eventual consistency
  + uiteindelijk zal de databank consistent worden.
  + Er is geen garantie wanneer dit zal gebeuren, gewoon ‘ergens’ in de toekomst.

1. **Distributed datastores**

Een distributed computersysteem bestaat uit verschillende computers en softwarecomponenten die via een netwerk communiceren. Het systeem kan bestaan uit:

* Verschillende mainframes
* Workstations
* PC’s
* ..

De computers communiceren met elkaar en delen de systeembronnen om een gemeenschappelijk doel te bereiken. Data kan gespreid zijn over verschillende nodes en taken worden verdeeld over de verschillende nodes. (Dit kan een RDBMs niet doen)

Distributed datastores zijn datastores die inzetbaar zijn in een dergelijk distributed systeem.

Daarvoor komen geen RDBMS in aanmerking en hebben we dus NoSQL‐datastores nodig.

Voordelen:

* Reliability: als één of meer nodes crashen dan kunnen de overige blijven verder werken
* Scalability: uitbreiden is altijd mogelijk
* Sharing resources: vele applicaties maken gebruik van dezelfde gegevens en systemen
* Flexibility: eenvoudig om nieuwe services te installeren, te implementeren en te debuggen
* Speed: meer power en snelheid
* Open system: elke client (gebruiker) kan beschikken over elke service indien voldoende rechten
* Performance: hogere performantie is mogelijk doordat de taken verdeeld kunnen worden over verschillende nodes

Nadelen:

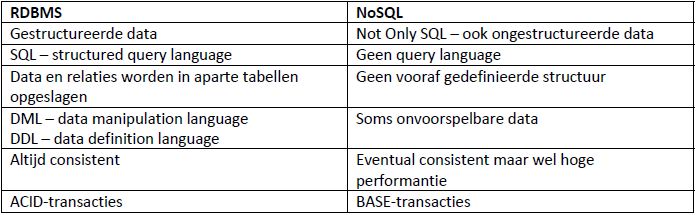
* Minder software support
* Netwerkproblemen
* Beveiligingsproblemen

1. **NoSQL**

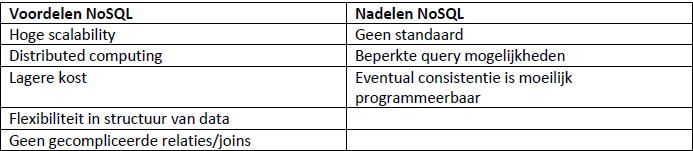
NoSQL DBMS = niet-relationeel databasemanagement systeem

Ontwikkeld voor distributed data stores die big data moeten bevatten.

* Vereist geen vaste structuren
* Vermijdt join-operaties
  1. **RDBMS vs NoSQL**

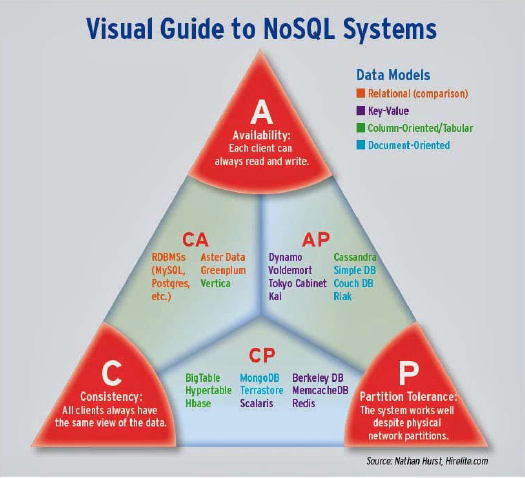


* 1. Pro en contra NoSQL

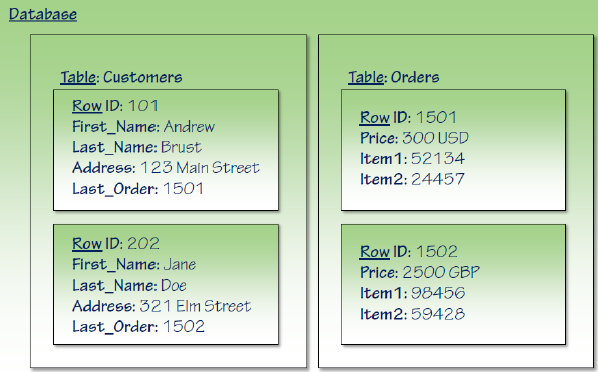


* 1. **NoSQL database systemen**

Belangrijkste databasesystemen op de markt ingedeeld volgens de principes van de CAP-stelling (weet niet of je dit moet kennen of niet)



* + 1. **Key-value stores**



Door gebruik te maken van key value is het mogelijk om grote hoeveelheden informatie parallel te verwerken?

Bij NoSQL database systemen zijn er geen foreign keys die de tabellen verbinden.

Eigenschappen van key-value stores:

* Meest gebruikte basistype
* Kan vele TB aan gegevens aan
* Laat ongestructureerde gegevens toe
* Makkelijk uitbreidbaar
* Gegevens worden opgeslagen als een (hash) table waarin elke key uniek is en de value vanalles kan zijn.
* Een key-value pair kan bestaan uit een naam gecombineerd met een waarde.
* Beperking: je kan enkel zoeken via de key
  + - 1. **Map-reduce techniek en het verwerken van Big Data met key-values**

Map reduce = techniek om een grote hoeveelheid ongestructureerde informatie te verdelen over een groot aantal parallel werkende computers om vervolgens de resultaten te bundelen.

Dit maakt het mogelijk om in korte tijd veel informatie te bewerken. (pijlers van big data)

* Map
  + Verdelen van de informatie over een groot aantal compuers.
  + Er bestaan 2 technieken (afhankelijk van de vervolgstap, het onderzoeken van gegevens):
    - Methode 1
      * Voor het doorzoeken van de data wordt slecht één algoritme gebruikt.
      * Informatie wordt gelijkelijk verdeeld over de beschikbare computers
    - Methode 2
      * Voor het doorzoeken van de data worden verschillende algoritmen gebruikt
      * Computers krijgen eerst een algoritme en vervolgens de informatie
      * Informatie wordt zo verdeeld dat er per algoritme en computer steeds evenveel informatie wordt toegekend.
    - In beide situaties is de verwachting dat elke computer ongeveer even lang nodig heeft om de gegevens te verwerken.
* Reduce
  + Het ophalen van de resultaten van de verschillende computers en het combineren ervan.

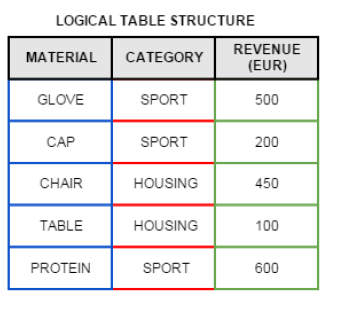
Om problemen in de werking te voorkomen:

Als 1 van de computers vastlopen dan wordt het proces opnieuw gestart op een andere computer of wordt besloten om de eventuele resultaten niet te gebruiken.

Door de key value-methode en de map reduce-techniek te combineren kan snel en eenvoudig Big Data verwerkt en gestructureerd worden.

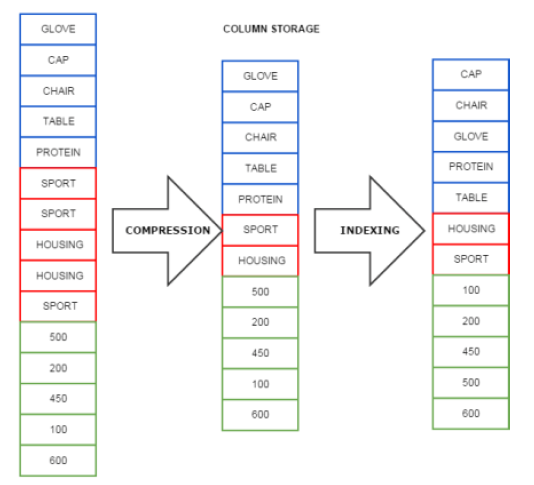
Voorbeeld: het voorkomen van verschillende woorden in een tekst.

* Map
  + Elke computer krijgt een tekst toegewezen
  + Een woord wordt een key genoemd en de value het aantal keren dat het voorkomt
  + Elke computer maakt zijn eigen tabel met keys en values
  + Elke computer is klaar
* Reduce
  + De computers sturen hun tabellen naar één centrale computer die alles samenvoegt tot één tabel.
    1. **Column-oriented stores**



Row storage vs colum storage

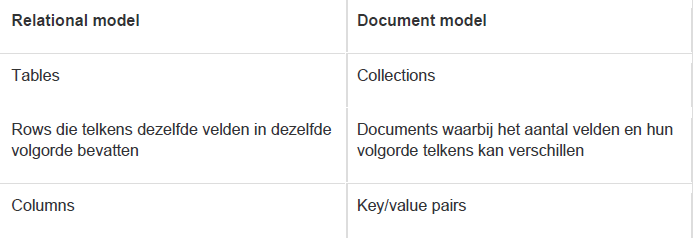


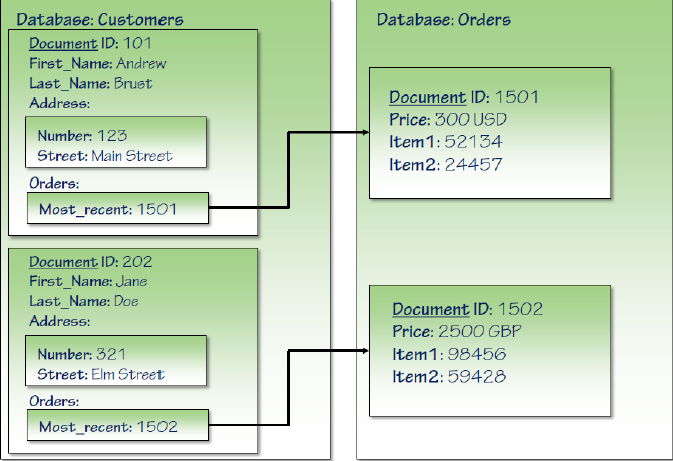




Eigenschappen van column oriented stores:

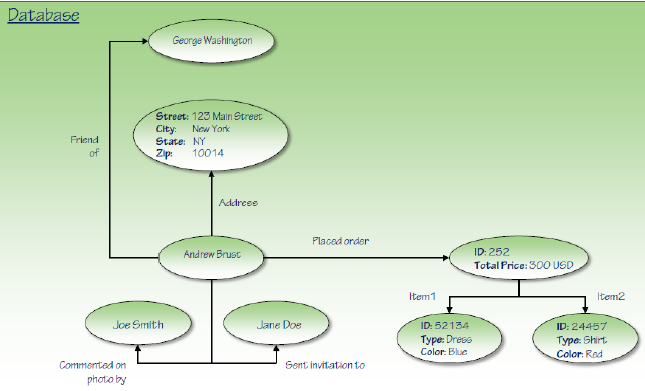
* Waarden worden opgeslagen met een pointer naar alle rijen die die waarde hebben.
* Werken met de kolommen van elke kolom wordt apart behandeld en ka ndeel uitmaken van een Column family
* Slaan de values van een kolom aaneengesloten op
* Slaan de kolomgegevens op in een specifieke file
* Gebruiken keys, maar die verwijzen naar verschillende kolommen
* Laten queries toe
* Alle data in een kolomfile zijn van hetzelfde type waardoor ze makkelijk gecompressed kunnen worden.
  + Hebben een hogeren performance bij gewone en groepsqueries
* Bv. HBase, Cassandra, SimpleDB, SAP HANA
  + 1. **Document-oriented stores**





Eigenschappen van document-oriented stores:

* Bevatten een collection van documenten
  + Diverse data groeperen
  + Kunnen verschillende key/pairs bevatten
  + Kunnen geneste documenten bevatten
  + Documenten zijn JSON objecten
* Slaan de data op in documenten waarbij de key toegang geeft tot de waarden (key/value pair)
* Hebben geen vaste structuur. Zijn flexibel en gemakkelijk aanpaspaar
* Speciale key-value want het heeft niet de beperkingen van de traditionele key-value.
  + Je kan wel zoeken via andere indexen
* Vanuit applicaties kan er naar de documenten verwezen worden via URL’s
* Biedt een API of query language
* Bv. MongoDB, CouchDB
  + 1. **Graph stores**



Eigenschappen van graph stores:

* Slaan data op in een grafiek
* Presenteren gegevens op een zeer toegankelijke manier
* Zijn een verzameling van nodes en edges
  + Node: vertegenwoordigt een entiteit (student, klant, …)
  + Edge: vertegenwoordigt een connectie/relatie tussen 2 nodes
* Elke node en elke edge hebben een unieke identiteit
* Elke node kent zijn aangrenzende nodes
* Gebruikt indexen voor opzoekingen
* Vb. OrientDB, Neo4j, Apache Giraph

1. **NoSQL, Relational or Both?**

In een bedrijf, organisatie zal er meestal gebruik gemaakt worden van verschillende types datastores. (aangepast systeem, combinatie van relationeel / niet-relationeel)

