Big Data - Korte samenvatting theorie

# Terminologie

**Database (gegevensbank of databank)** = Een digitaal opgeslagen archief met het oog op flexibeledataopslag, raadpleging en gebruik.

**Datawarehouse** = Gegevensverzameling die terugkerende- en ad-hoc vragen in korte tijd kan beantwoorden zonder dat bronsystemen daardoor overmatig belast worden.

**ETL-programmatuur** = Omzetten van data naar informatie. ETL staat voor Extract, Transform en Load:  
- **Extract:**Gegevens worden uit het transactiesysteem gehaald  
- **Transform:**Omzetting van data/gegevens naar informatie  
- **Load:**Informatie wordt opgeslagen in informatiesysteem

**Datamining =**  Op basis van relevante gegevens zoeken naar (statistische) verbanden in gegevensverzamelingen. Kan gebruikt worden voor wetenschap, journalistiek of commercieel gebruik.

# Stappenplan Datamining

**Stap 1: Business Understanding**

Kennis van toepassing vastleggen. Onderscheid maken tussen zinnige en onzinnig resultaten.

**Stap 2: Data Understanding**

Kennis van de informatie vergaren. Betrouwbaarheid en strekking van de gegevens bepalen.

**Stap 3: Data Preparation**

Kwaliteit en volledigheid van de informatie controleren. Aanpassen/verbeteren waar nodig.

**Stap 4: Modeling**

Word gezien als echte datamining, de informatie word in deze stap onderzocht.

**Stap 5: Evaluation**

Controle van de resultaten van Stap 4: Modeling. Nagaan of de resultaten de gewenste antwoorden zijn volgens de criteria van Stap 1: Business Understanding.

**Stap 6: Deployment**

Het toepassen van de uitkomsten van de datamining binnen de zakelijke omgeving.

# Big Data

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

**Big Data hype:**

**-** Hedendaagse hardwaremogelijkheden, waaronder goedkopere servers  
- Goedkopere en ruimere opslag  
- Mogelijkheden van opensource software  
- Beschikbaarheid van massa’s gegevens

**Toepassingen Big Data:**

**-** Voor marketing, zeker bij webbased bedrijven **-** Voor politieonderzoek en ‐opsporing, o.a. bij fraude en cybercrime **-** Voor analyse bij grote datalekken, bijv. WikiLeaks, LuxLeaks, Panama Papers **-** Voor onderzoek in de gezondheidssector, zoals naar ziektes en erfelijkheid **-** Voor de industrie, bijv. om de technologie in auto’s te verbeteren naar veiligheid toe

**Betekenissen Big Data:**

**-** Honderden terabytes  
- Een “klassieke” databank kan de complexiteit niet meer aan. Een alternatief is nodig voor de

niet‐relationele gegevens  
- De 3 V’s van Big Data:  
 - **Volume**: Grote hoeveelheid gegevens, teveel om te verwerken  
 - **Velocity**: Hoge snelheid van datatoevoer en verwerkingssnelheid  
 - **Variety**: Structuur van gegevens varieert en is niet vast zoals bij relationele databank  
- 2 bijkomende V’s uit bepaalde lectuur:  
 - **Veracity**: Betrouwbaarheid van gegevens  
 - **Value**: Info gegenereerd uit de Big Data moet waardevol zijn

**CAP-stelling:**

**- Consistency:**

Mate waarin het DB-systeem aan alle betrokken servers de meest recente gegevens laat zien.  
=> Optimaal als na het uitvoeren van een operatie/transactie alle gebruikers dezelfde gegevens zien op de betrokken servers. (Operatie/transactie = toevoegen, verwijderen of wijzigen van data => acties/taken die gelijktijdig uitgevoerd moeten worden)

**- Availability:**

Mate waarin het DB-systeem beschikbaar is.  
=> Optimaal wanneer het systeem altijd beschikbaar/online is.

**- Partition Tolerance:**

Mate waarin het DB-systeem blijft werken bij uitval van één of meerdere servers.  
=> Computersysteem moet blijven functioneren wanneer er mogelijk één of meerdere servers uit zouden vallen.

CAP-stelling zegt dat als het gaat om een gedistribueerd systeem met meerdere servers, er een keuze gemaakt moet worden tussen consistency en availability => 1 van beide garanties kan dan niet vervuld worden.

 => Kiezen van prioriteiten

**Relationele databanken:**

Consistency en Availability (CA) het belangrijkste => Hier word hoofdzakelijk met operationele gegevens gewerkt die constant en consistent bereikbaar moeten zijn. (MySQL)

**NoSQL databanken:**

- **CP**: - Sommige data niet direct bereikbaar  
 - Data is wel consistent  
 - Data mag gespreid zijn over meerdere servers

-**AP**: - Systeem is beschikbaar maar niet altijd volledig   
=> Nuttig bij analyse van Big Data waar consistency minder belangrijk is en data verdeeld is over verschillende servers

**Keerzijde van Big Data:**

Privacy => We willen niet dat iedereen die dat wil, weet wat we doen, waar we zijn geweest, etc…

# Database principes

**Transactie (gebruikt bij ACID):**

Een transactie bestaat één of meerdere data-manipulerende statements en queries die gegevens lezen/schrijven in een database. Consistency en Integrity van data is cruciaal.

- **Patroon van een transactie**: - Begin de transactie  
 - Voer een set van data manipulaties en queries uit  
 - Commit en eindig de transactie als er geen errors optreden  
 - Rollback en eindig de transactie als er errors optreden

**ACID:**

- **Atomic:**

Elke transactie slaagt volledig of slaagt niet. Als 1 statement in de set van data manipulaties/queries een fout bevat => Hele transactie ondergaat een rollback

**- Consistent (NIET hetzelfde als Consistency bij CAP):**

Gegevens mogen niet tegenstrijdig zijn. Op elk moment moet aan alle constraints/voorwaarden voldaan zijn. Als een bepaalde statement uit een transactie gegevens tegenstrijdig maakt   
=> Transactie wordt automatisch afgebroken.

**- Isolated:**

Elke transactie wordt los van een andere transactie uitgevoerd. Transacties die parallel/gelijktijdig worden uitgevoerd kunnen elkaars tussenresultaten van de statements niet zien.

**- Durable:**

Als de transactie is voltooid en er een commit heeft plaatsgevonden => Permanent/Onomkeerbaar

**BASE:**

- **Basic Availability:**

Het systeem garandeert availability zoals voorzien in de CAP-stelling. Er mogen tijdelijk onvolkomenheden zitten of er kan tijdelijk onbereikbare data zijn.

**- Soft State:**

Het systeem kan in de tijd wijzigen zonder input van data. => Ontwikkelaar moet dit opvangen

**- Eventual Consistency:**

Na een tijd zal de databank eventueel (=uiteindelijk, niet misschien) consistent worden.

# Distributed datastores

**Distributed computersyteem:** Bestaat uit verschillende computers en/of software componenten die via een netwerk communiceren.

**Distributed datastores:** Datastores die inzetbaar zijn in een distributed (computer)systeem.  
 => Geen RDBMS in aanmerking dus => NoSQL-datastores gebruiken

**Voordelen van distributed (computer)systeem:**

- 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 kracht en snelheid

- Open system: elke client(user) met voldoende rechten kan beschikken over elke service

- Performance: hogere performantie is mogelijk doordat de taken verdeeld kunnen worden

over verschillende servers/nodes

**Nadelen van distributed (computer)systeem:**

**-** Minder software support  
- Netwerkproblemen  
- Beveiligingsproblemen

# NoSQL

**NoSQL DBMS =** Niet-relationeel databasemanagement systeem ontwikkeld voor distributed data stores die big data moeten bevatten  
=> Vereist geen vaste structuren en vermijd JOIN-operaties

**RDBMS VS NoSQL:**

|  |  |
| --- | --- |
| **RDBMS** | **NoSQL** |
| Gestructureerde data | Not Only SQL – ook ongestructureerde data |
| SQL – structured query language | Geen query language |
| Data en relaties worden in aparte tabellen  opgeslagen | Geen vooraf gedefinieerde structuur |
| DML – data manipulation language  DDL – data definition language | Soms onvoorspelbare data |
| Altijd consistent | Eventual consistent maar wel hoge  performantie |
| ACID‐transacties | BASE‐transacties |

**Voor- en nadelen NoSQL:**

|  |  |
| --- | --- |
| **Voordelen NoSQL** | **Nadelen NoSQL** |
| Hoge scalability | Geen standaard |
| Distributed computing | Beperkte query mogelijkheden |
| Lagere kost | Eventual consistentie is moeilijk  programmeerbaar |
| Flexibiliteit in structuur van data |  |
| Geen gecompliceerde relaties/joins |  |

**Key-Value Stores:**

**Key value methode =** Ontwikkeld om te werken met ongestructureerde informatie   
=> Tekst uit een boek  
- **Key**: woord uit de tekst  
- **Value**: aantal keren dat het woord voorkomt

**Map reduce techniek =** Gebruikt om een grote hoeveelheid ongestructureerde informatie te verdelen over een groot aantal parallel werkende computers en daarna de resultaten weer samen te bundelen.  
= In korte tijd veel informatie verwerken => Pijler Big Data

**Twee stappen (Map & Reduce):**

**- Map:** Het verdelen van informatie over een groot aantal computers

**- Methode 1:** Doorzoeken van informatie wordt slechts door één algoritme gedaan. Bij deze methode wordt de informatie gelijk verdeeld over de beschikbare computers.

**- Methode 2:** Doorzoeken van informatie wordt door verschillende algoritmen gedaan. Computers krijgen eerst een algoritme en daarna de informatie. Op deze manier wordt de informatie verdeeld zodat er per algoritme en computer steeds evenveel informatie wordt toegekend.

**- Reduce:** Het ophalen en combineren van de resultaten van de verschillende computers. Zo word er slechts 1 antwoord gegeven op de totale opdracht.

**- Problemen voorkomen:** Als één van de computers vastloopt of crashed, wordt het proces opnieuw gestart op een andere computer of wordt er besloten deze resultaten niet te gebruiken.

**Eigenschappen Key-Value Stores:**

- Meest gebruikte basistype

- Kan vele TB aan gegevens aan

- Laten ongestructureerde gegevens toe, gemakkelijk uitbreidbaar

- Gegevens worden opgeslagen als een hash table waarin elke key uniek is en de value een

string, JSON‐object, BLOB‐object, … kan zijn

- Een key‐value pair kan bestaan uit een naam gecombineerd met een waarde

- Beperking: je kan enkel zoeken via de key!  
- Bij NoSQL zijn er geen foreign keys die de tabellen verbinden

**Eigenschappen Column-Oriented Stores:**

- Werken met de kolommen en elke kolom wordt apart behandeld en kan deel uitmaken van

een column family (CF)

- Slaan de values van een kolom aaneengesloten op

- Slaan de kolomgegevens op in specifieke files

- Gebruiken keys, maar die verwijzen naar verschillende kolommen

- Laten ook queries toe

- Alle data in een kolomfile zijn van hetzelfde type, waardoor ze gemakkelijk gecompressed

kunnen worden

- Hebben daardoor ook een hoge performance bij gewone en groepsqueries en zijn dan ook

zeer geschikt voor Business Intelligence(BI) en Customer Relationship Management (CRM)

toepassingen

**Eigenschappen Document-Oriented Stores:**

- Bevatten een collection van documenten

- Slaan de data op in documenten waarbij de key toegang geeft tot de waarden

- hebben niet noodzakelijk een vaste structuur, waardoor ze flexibel en gemakkelijk

aanpasbaar zijn;

- Slaan documents op in collections om diverse data te groeperen en deze documents kunnen

verschillende key‐value pairs en zelfs geneste documenten bevatten;

- De documenten zijn JSON objecten;

- Eigenlijk een gespecialiseerde key‐value store met specifieke eigenschappen maar zonder de

beperking van KeyValue nl. je kan nu wel zoeken via andere indexen;

- Vanuit applicaties kan er naar de documenten verwezen worden via URI’s 7;

- Biedt een API of query language zodat er kan gezocht worden op de inhoud van het

document;

**Eigenschappen Graph Stores:**

- Slaan data op in een grafiek;

- Presenteren gegevens op een zeer toegankelijke manier;

- Zijn een verzameling van nodes en edges;

- Elke node vertegenwoordigt een entiteit (student, klant,…) en elke 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;

# NoSQL, Relationeel of beide gebruiken?

In een bedrijf, organisatie zal er meestal gebruik gemaakt worden van verschillende types datastores omdat de gebruikte gegevens zeer uiteenlopend kunnen zijn en dus een aangepast systeem vereisen.



=> Hier word gebruik gemaakt van verschillende oplossingen