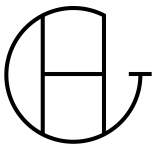


HoGent

Faculteit Bedrijf en Organisatie

Een vergelijkende studie tussen Data Vault 2.0 en het dimensioneel model bij het modelleren van een datawarehouse.

Jorik Spiesschaert

Scriptie voorgedragen tot het bekomen van de graad van
professionele bachelor in de toegepaste informatica

Promotor:
Stijn Lievens
Co-promotor:
Jochen Stroobants

Instelling: DHL Pharma-Logistics

Academiejaar: 2018-2019

Tweede examenperiode

Faculteit Bedrijf en Organisatie

Een vergelijkende studie tussen Data Vault 2.0 en het dimensioneel model bij het modelleren van een datawarehouse.

Jorik Spiesschaert

Scriptie voorgedragen tot het bekomen van de graad van
professionele bachelor in de toegepaste informatica

Promotor:
Stijn Lievens
Co-promotor:
Jochen Stroobants

Instelling: DHL Pharma-Logistics

Academiejaar: 2018-2019

Tweede examenperiode

Woord vooraf

Samenvatting

Inhoudsopgave

Lijst van figuren

Lijst van tabellen

1. Inleiding

De inleiding moet de lezer net genoeg informatie verschaffen om het onderwerp te begrijpen en in te zien waarom de onderzoeksvraag de moeite waard is om te onderzoeken. In de inleiding ga je literatuurverwijzingen beperken, zodat de tekst vlot leesbaar blijft. Je kan de inleiding verder onderverdelen in secties als dit de tekst verduidelijkt. Zaken die aan bod kunnen komen in de inleiding (Pollefliet, 2011):

- context, achtergrond
- afbakenen van het onderwerp
- verantwoording van het onderwerp, methodologie
- probleemstelling
- onderzoeksdoelstelling
- onderzoeksvraag
- ...

1.1 Probleemstelling

Uit je probleemstelling moet duidelijk zijn dat je onderzoek een meerwaarde heeft voor een concrete doelgroep. De doelgroep moet goed gedefinieerd en afgelijnd zijn. Doelgroepen als "bedrijven," "KMO's," systeembeheerders, enz. zijn nog te vaag. Als je een lijstje kan maken van de personen/organisaties die een meerwaarde zullen vinden in deze bachelorproef (dit is eigenlijk je steekproefkader), dan is dat een indicatie dat de doelgroep goed gedefinieerd is. Dit kan een enkel bedrijf zijn of zelfs één persoon (je co-promotor/opdrachtgever).

1.2 Onderzoeksvraag

Wees zo concreet mogelijk bij het formuleren van je onderzoeksvraag. Een onderzoeksvraag is trouwens iets waar nog niemand op dit moment een antwoord heeft (voor zover je kan nagaan). Het opzoeken van bestaande informatie (bv. “welke tools bestaan er voor deze toepassing?”) is dus geen onderzoeksvraag. Je kan de onderzoeksvraag verder specifiëren in deelvragen. Bv. als je onderzoek gaat over performantiemetingen, dan

1.3 Onderzoeksdoelstelling

Wat is het beoogde resultaat van je bachelorproef? Wat zijn de criteria voor succes? Beschrijf die zo concreet mogelijk.

1.4 Opzet van deze bachelorproef

De rest van deze bachelorproef is als volgt opgebouwd:

In Hoofdstuk 2 wordt een overzicht gegeven van de stand van zaken binnen het onderzoeksdomein, op basis van een literatuurstudie.

In Hoofdstuk 3 wordt de methodologie toegelicht en worden de gebruikte onderzoekstechnieken besproken om een antwoord te kunnen formuleren op de onderzoeksvragen.

In Hoofdstuk 4, tenslotte, wordt de conclusie gegeven en een antwoord geformuleerd op de onderzoeksvragen. Daarbij wordt ook een aanzet gegeven voor toekomstig onderzoek binnen dit domein.

2. Stand van zaken

Dit hoofdstuk bevat een literatuurstudie omtrent data warehousing. Na het lezen van dit hoofdstuk zullen begrippen zoals dimensioneel modelleren, data vault 2.0 en data warehousing jou niet meer onbekend zijn en waarom er nood is aan data warehousing. Ook zullen beide modelleertechnieken dieper bekeken worden.

2.1 Inleiding data warehousing

> Torture the data, and it will confess to anything.
>
> *Ronald Coase*
> *Winnaar Nobelprijs in Economie (1991)*

Veel moderne, digitale bedrijven genereren tegenwoordig enorme volumes data. Deze data kan afkomstig zijn uit verschillende bronnen: CRM-systeem, flat-files (vb. rekenbladen), Twitter-feeds, ... Bestuursleden gebruiken data om beslissingen te nemen die de onderneming toelaat om te (blijven) groeien of om bepaalde problemen op te sporen. Stel dat een onderneming meer kosten maakt dan opbrengsten. Op basis van alle gegevens die het bedrijf bezit, kan hieruit dan een analyse gemaakt worden. Zijn er overbodige kosten? Worden onze producten/diensten aan een te lage prijs verkocht? Dit zijn maar enkele vragen die kunnen opgelost worden wanneer het bestuur de correcte rapporteringen ontvangt.

2.1.1 Soorten data

Er kan een onderscheid gemaakt worden tussen verschillende soorten data. Voornamelijk kunnen we informatie opdelen in 2 categorieën: gestructureerde en ongestructureerde data. Volgens een artikel van Justin Langseth (2005), bestaat 95% van de globale informatie uit ongestructureerde data.

Gestructureerde data

Data afkomstig uit een relationele databank (RDBMS) is meestal gestructureerd. Deze data is meestal ingedeeld in categorieën, denk bijvoorbeeld maar aan postcode, naam, klantennummer, ... Hieruit volgt dat deze data heel gemakkelijk te doorzoeken is.

Ongestructureerde data

Deze informatie kan niet gemakkelijk worden opgeslagen in databanken. Denk maar aan rekenbladen, emails, tweets, muziek, ... Data afkomstig uit IoT-apparaten zijn meestal ook ongestructureerd. Deze data bevat vaak ook heel nuttige informatie die organisaties graag willen benuttigen. Denk bijvoorbeeld maar aan tweets: hoe gelukkig zijn klanten over een bepaald product? Hoeveel mails worden er maandelijks ontvangen met klachten?

2.1.2 Wat is een data warehouse?

De definitie van een data warehouse luidt als volgt: *ëen subject-georiënteerde, geïntegreerde, tijd-variante, niet-vluchtige collectie van gegevens dat in eerste instantie gebruikt wordt bij organisaties om beslissingen te nemen"* (Vassiliadis, 2000).

Subject-georiënteerd

Dit begrip slaat op het feit dat een data warehouse met de reden gebouwd is om data te analyseren, niet om transacties op toe te passen. Dit wordt uitgebreid besproken in subsectie 2.1.5.

Geïntegreerd

Dit betekent dat de data warehouse een "centrale"databank is die gegevens bevat vanuit verschillende bronsystemen (bijvoorbeeld gegevens uit het klantenbestand en gegevens uit het verkoopsysteem). Deze data kan effectief ingeladen worden, maar ook opgeslagen worden in virtuele tabellen.

Tijd-variant

Alle data van het verleden, moet terug te vinden zijn in de data warehouse. Dit betekent dat data uit het verleden (bijvoorbeeld een vorig adres van een klant) moet beschikbaar

zijn, ook al is deze in het transactioneel systeem aangepast.

Niet-vluchtig

De data die in het systeem zit, moet onveranderlijk zijn, ook al zijn deze fout. Om de foutieve data toch aan te passen, zal er een nieuwe rij moeten toegevoegd worden die de juiste data bevat, die een hogere versie bevat dan de vorige rij.

Conclusie

We kunnen dus uit de definitie van een data warehouse afleiden dat het een grote databron is die alle (gestructureerde en ongestructureerde (indien mogelijk)) gegevens bevat die een organisatie bezit vanaf het moment waarbij de data warehouse geïmplementeerd is tot het heden. Op deze databron worden dan analyses gemaakt.

2.1.3 Waarom is er nood aan een data warehouse?

Een organisatie heeft tegenwoordig heel wat data ter beschikking. Vaak is deze data gefragmenteerd over verschillende systemen. Wanneer men een analyse wil opmaken op basis van de verspreide data, zal dat niet evident zijn.

Om deze reden wordt een data warehouse ontworpen. Hierin worden gegevens, verspreidt over meerdere bronnen, in één bron verzameld. Zo kunnen rapporteringen makkelijk en flexibel opgebouwd worden.

Een andere reden voor het opbouwen van een data warehouse is dat je de historiek van alle data kan bijhouden. Wanneer er bijvoorbeeld gegevens aangepast zijn in het transactionele systeem, dan zijn de oude gegevens vaak moeilijk te achterhalen (omdat deze dat vaak overschreven wordt). Door verschillende versies bij te houden van entiteiten, kan je oudere data makkelijk opzoeken.

Wanneer men rapporteringen wil opvragen aan het transactionele systeem, vergt dit ook extra belasting van de server. Dit komt doordat het datamodel die opgebouwd werd niet geoptimaliseerd is om zware SELECT-queries af te handelen. Dit zou niet alleen de server meer belasten, bovendien zal dit ook zorgen voor een tragere rapportering.

2.1.4 Wat is het doel van een data warehouse?

Het belangrijkste doel dat een data warehouse heeft is om een **correcte** rapportering op te leveren. Beslissingen in organisaties worden genomen op basis van rapporten.

Data kwaliteit

Zoals eerder aangekaart, is het belangrijk dat rapporten de juiste gegevens bevat. Hieruit volgt dat data kwaliteit een heel belangrijk aspect is. Vaak zijn er verschillende oorzaken waarom de data kwaliteit niet voldoet:

- Inconsistente data tussen verschillende systemen
- Incorrecte gegevens
- Onvoldoende validatie bij het invoeren van gegevens
- Onjuiste gegevensbewerkingen
- ...

(Markus Helfert, 2002)

Gegevens die in een data warehouse geladen worden, ondergaan een proces (zie paragraaf 2.1.6). In dit proces wordt de data gemanipuleerd zodat de data kwaliteit verhoogd wordt.

Performantie

We kunnen een onderscheid maken tussen 3 verschillende soorten beslissingen: operationele (dagelijks), tactische (jaarlijks) en strategische (lange termijn) beslissingen. Wanneer we operationele rapporten nodig hebben, verwachten we dan ook dat deze onmiddelijk kunnen opgeleverd worden. Het datamodel van een data warehouse wordt geoptimaliseerd voor het ophalen van data in plaats van het te kunnen stockeren. De data wordt 's nachts ingeladen zodat werknemers geen performantieproblemen hieromtrend ondervinden.

De toekomst Door de komst van in-memory databanken merken we op dat een deel van de (operationele) rapportering opnieuw verhuist naar de transactionele databanken. Dit heeft enerzijds te maken met de snelheid van de databanken en anderzijds met het feit dat queries om operationele rapporten op te vragen gebruikelijk niet zo belastend zijn. Zo kan er gewerkt worden met **live data** (doordat deze niet 's nachts moet ingeladen worden in de data warehouse). De voorwaarde hiervoor is dat alle benodigde data beschikbaar is binnen dat geïntegreerd systeem. Een voorbeeld van een in-memory databank is HANA, een technologie ontwikkelt door SAP.

Automatisering

Doordat alle rapporteringsnoden geautomatiseerd kunnen worden, heeft dit natuurlijk als voordeel dat personeelsleden deze niet meer manueel hoeven te maken/berekenen. Zo kunnen ze hun tijd spenderen aan andere prioriteiten. Deze data kan dan worden voorgesteld in een overzichtelijke omgeving (zie sectie 2.4).

2.1.5 Wat is OLTP en wat zijn de verschillen met OLAP?

On-line transactional processing (OLTP) systemen zijn voornamelijk klantgericht. Het datamodel is opgebouwd rond het efficiënt verwerken van transacties. On-line analytical processing (OLAP) systemen zijn martkgericht. De data in een OLAP-systemen worden gebruikt om analyses op uit te voeren (Satyanarayana, 2010).

Inhoudelijk

Bij OLAP systemen worden meta data opgeslagen bij de entiteiten. Voorbeelden hiervan zijn: tijdstip van inladen, van welke bron de data komt, ... Het grote voordeel hierbij is dat wanneer een fout gebeurt, er gemakkelijker kan achterhaalt worden waar het fout liep. Ook wordt de historische data ook bewaard, in tegenstelling tot OLTP. Bij OLTP wordt de te wijzigen data overschreven. Het gevolg hiervan is dat de volume data bij OLAP doorgaans groter zal zijn.

Toegankelijkheid

Wanneer men data wil verkrijgen/wijzigen in een OLTP systeem, moet er rekening gehouden met een aantal aspecten. Een transactie in een OLTP omgeving moet voldoen aan enkele eisen:

- **Atomic:** Wanneer een transactie afgebroken is, mag er niets gewijzigd zijn in de databank.
- **Consistent:** Als een deel van de transactie faalt, zullen alle doorgevoerde wijzigingen in die transactie ongedaan gemaakt worden en zal de databank terugkeren naar een consistente staat.
- **Isolated:** Transacties worden geïsoleerd, transacties mogen in geen enkel geval elkaar beïnvloeden.
- **Durable:** Wanneer een transactie is doorgevoerd, kan deze niet meer ongedaan gemaakt worden.

Bij een OLAP-systeem worden geen transacties doorgevoerd, enkel leesopdrachten. Dat vermindert de complexiteit en verhoogt de snelheid van de queries (Satyanarayana, 2010).

2.1.6 Wat zijn de benodigdheden voor een data warehouse?

Voor er kan begonnen worden met het opbouwen van een data warehouse, zijn er enkele benodigdheden. Zo zal er een keuze moeten gemaakt worden voor een bepaalde methodologie en een architectuur. Ook zal er fysieke opslagplaats nodig zijn. Hiervoor kan gebruik gemaakt worden van Cloud oplossingen of een on-premise server. Maar in dit hoofdstuk bespreken we welke software-aspecten er nodig zijn bij het opbouwen van de data warehouse.

RDBMS

Voor het opmaken en beheren van de data warehouse zal er een RDBMS moeten uitgekozen worden. Hiervoor zijn heel wat mogelijkheden beschikbaar op de markt. Bijvoorbeeld:

- Oracle DB
- Microsoft SQL Server
- IBM DB2
- Microsoft Office Access
- ...

Klassieke databanken slaan hun gegevens op op harde schijven en/of SSD's. Maar sinds kort zien we de komst van een nieuwe technologie, genaamd een in-memory databank. Hierbij worden de gegevens initieël opgeslagen in het RAM-geheugen. Dit zorgt voor een veel snellere lees- en schrijftijd. Het nadeel van deze nieuwe technologie is het prijskaartje.

Data integratie software

Het data inladen is een proces die een verantwoordelijkheid is voor de data integratie software, al is dat niet zijn enige verantwoordelijkheid. Hij is verantwoordelijk voor het gehele ETL-proces:

- Extraction: Ophalen van de data vanuit de bron
- Load: Transformaties en manipulaties uitvoeren op die data
- Transformation: De data wegschrijven naar de nieuwe bron

2.2 Dimensioneel modelleren via Kimball

Wie een consultant Business Intelligence is, heeft ongetwijfeld al gehoord van dimensioneel modelleren. Over de jaren heen is het als het ware een standaard geworden wanneer men een data warehouse wilt ontwerpen. Dimensioneel modelleren heeft als voordeel dat het model niet complex is, dus gemakkelijk te begrijpen, zelf voor niet IT-opgeleide personen. Ook is er vaak een snel resultaat beschikbaar. In deze sectie zal het dimensioneel modelleren aan de hand van Kimball dieper bekeken worden.

2.2.1 Architectuur

Bij het dimensioneel modelleren wordt het proces opgedeeld in 2 soorten lagen:de data warehouselaag en de data marts. Op basis van de data warheouselaag worden de data marts opgebouwd. Rapporteringsomgevingen connecteren met de data marts om hun data op te halen.

Data warehouselaag

In deze laag wordt het ETL-proces toegepast. Eerst en vooral wordt de data vanuit (een) bron(nen) in de warehouse geladen. Daarna wordt de data bewerkt en gemanipuleerd. Bijvoorbeeld records met bepaalde lege records weglaten. De data kwaliteit (zoals eerder aangekaart) is zeer belangrijk in een data warehouse. Het doel is om de toegekomen data consistent te maken en ervoor te zorgen dat de integriteit bewaarborgd blijft (Ralph Kimball, 2013)

Data marts

In deze laag worden OLAP-cubes of relationele ster schema's gemaakt op basis van de data warehouselaag. Deze laag kan eigenlijk als de presentatielaag beschouwd worden. Deze laag moet gedetailleerde data bevatten. Een data mart moet gebaseerd zijn rond een business unit (Ralph Kimball, 2013).

2.2.2 Componenten

Een sterschema of OLAP-cube bestaat uit dimensies en facts. Wat deze precies zijn, wordt hieronder uitgelegd.

Dimention tabel

Voor elke dimensie wordt een primaire sleutel aangemaakt (of afkomstig uit het systeem als business sleutel). Deze sleutel wordt gebruikt in een fact table als vreemde sleutel, zodat er een relatie kan worden gelegd tussen beide entiteiten.

Naast de primaire sleutel wordt in deze tabel ook de beschrijvende data bewaart voor een bepaalde rij. Deze data kan gebruikt worden om in de rapporteringsomgeving de verschillende assen te kiezen. Een typische dimensie is bijvoorbeeld 'naam' of 'Woonplaats'.

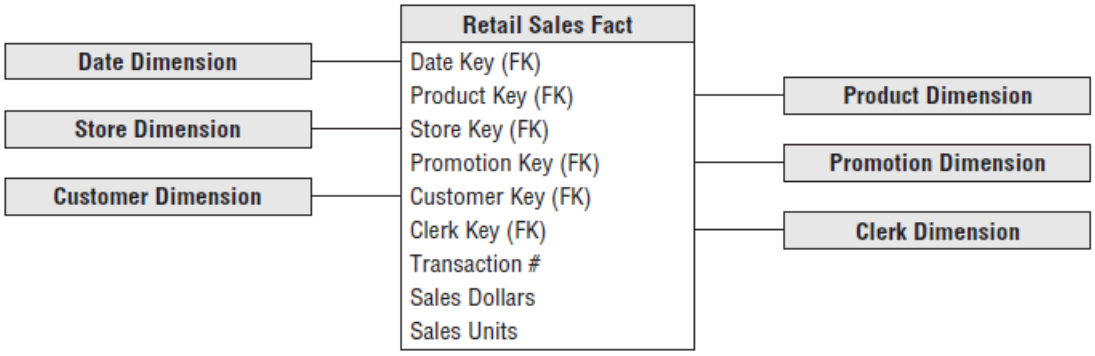
Fact tabel

In een fact table worden alle meetbare cijfers bijgehouden. Meetbare cijfers betekent dat bewerkingen moeten kunnen mogelijk zijn op die data. Een bankrekeningnummer is een getal, maar hier kunnen geen bewerkingen met uitgevoerd worden (bijvoorbeeld gemiddelde bankrekeningnummer geven). Invoice amount is wel een goed voorbeeld. Hierop kunnen enkele bewerkingen worden uitgevoerd, bijvoorbeeld: gemiddelde, minimum, maximum, totaal, ... Deze gegevens worden in vaktermen als **measures** aangeduidt.

De fact table bevat niet alleen measures, maar ook de vreemde sleutels van de dimensies waarmee het verbonden is. Zo kan je bruikbare informatie toevoegen aan je measure in de rapporteringsomgeving.

Ster schema

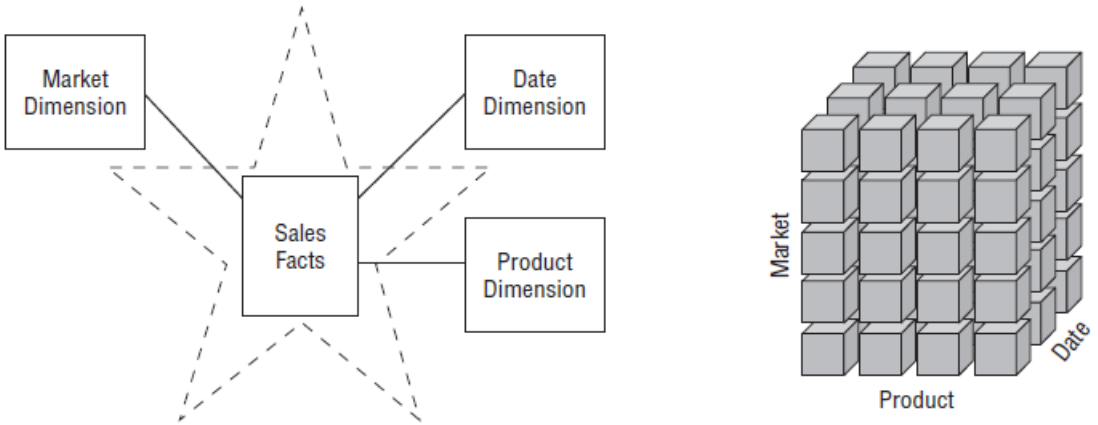
In een ster schema worden dimensies en fact tabellen verbonden door enerzijds de primaire sleutels in de dimensies, en anderzijds bij de vreemde sleutels in de fact tabel. Wanneer meerdere tabellen verbonden worden met elkaar zien we een centraal punt in het model, dat de fact tabel is (zie figuur 2.1).



Figuur 2.1: Ster sschema voorgesteld door Ralph Kimball (2013).

Verschil tussen een sterschema en een OLAP-cube

Het verschil tussen beide zit niet in het ontwerp, maar puur in het 'fysieke' gedeelte. OLAP cubes zijn geoptimaliseerd voor een drill down of een frill up te doen in de gegevensset. Drill down betekent dat de gegevens op een dieper detailniveau zullen bekeken worden, bijvoorbeeld vertrekkend uit een productcategorie niveau, ga je naar een productniveau. OLAP cubes zorgen ervoor dat er meer anlytische functies beschikbaar zijn in vergelijking met SQL. Maar als nadeel heeft de OLAP cubes dat het niet zo performant is als een ster schema (zie figuur 2.2). (Ralph Kimball, 2013).



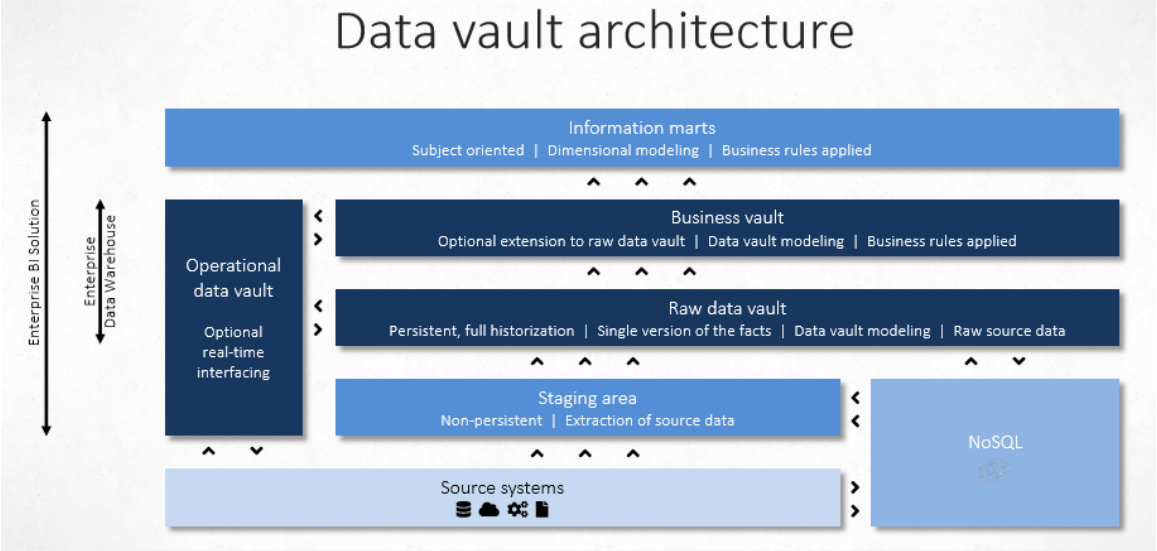
Figuur 2.2: Sterschema (links) en OLAP cube (rechts) voorgesteld door Ralph Kimball (2013).

2.3 Modelleren via Data Vault 2.0

Data Vault 2.0 is een modelleertechniek die ontworpen is door Daniel Linstedt. Het model zorgt ervoor dat dimensies gemakkelijk uitgebreidt kunnen worden en dat databronnen toevoegen vlot moet gaan. Linstedt is van mening dat business requirements vaak veranderen, dus moet het model waarin de data warehouse ontworpen is ook flexibel zijn. (Daniel Linstedt, 2016)

2.3.1 Architectuur

Data vault maakt gebruik van een 3-lagen model. Dit heeft als voordeel dat processen duidelijk kunnen onderscheiden worden per laag en blijft alles overzichtelijk. Deze architectuur ondersteunt om data op halen via een batch, maar ook om live data op te halen. NoSQL kan ook gebruikt worden om de data warehouse te ontwerpen. Wanneer de data live opgehaald wordt, valt de staging area weg en wordt de data onmiddelijk in de raw data vault geladen. (Daniel Linstedt, 2016)



Figuur 2.3: Data Vault architectuur voorgesteld door Stroobants (2018).

Staging area

In deze laag wordt alle data ingeladen (of een virtuele tabel gebruikt) van een bepaalde bron via een batch. Hier wordt alle data onbewerkt ingeladen. Deze data bevat dan ook nog geen historische data. De tabel worden dus gedupliceerd van de bron(nen). Deze laag is niet persistent.

Raw data vault

De data wordt overgeladen van de staging area naar de raw data vault via data integratie software. Deze laag is persistent, logisch ook wanneer we verschillende versies en de

historiek behouden van onze entiteiten. Vanaf deze laag beginnen we te modelleren in Data Vault. De data wordt in deze laag gemanipuleerd (ofwel getransformeerd (ETL)). Ook wordt er metadata toegevoegd aan de records zodat er audits kunnen plaatsvinden.

Business vault

Dit is een optionele laag. Deze laag wordt enkel en alleen toegevoegd wanneer er 'Business regels' moet toegepast worden in het model. De business vault wordt in principe niet opgeslagen in een aparte laag, maar vaak wordt deze opgeslagen als een uitbreiding van de raw data vault. Een voorbeeld van een 'Business rule' is dat je bijvoorbeeld geen producten wil promoten wanneer er maar minder dan 10 in voorraad zijn.

Information marts

Vertrekkend uit de business vault (of raw data vault wanneer deze niet aanwezig is) zullen er information marts moeten aangemaakt worden, ook wel bekend als data marts . Daniel Linstedt (2016) spreek liever over 'information' mart omdat het doel van een enterprise data warehouse duidelijk is: informatie aanbieden. De information marts bestaan uit sterschema's. Hierop connecteren de eindgebruikers om hun informatie te verschaffen.

2.3.2 Componenten

Een data vault model bestaat uit 3 soorten componenten: hubs, links en sattelieten. Elk component heeft zijn functie en zijn doel.

Hubs

Daniel Linstedt (2016) beschrijft hubs als pilaren voor het Data Vault model. Een hub bestaat altijd uit minmaal 4 attributen:

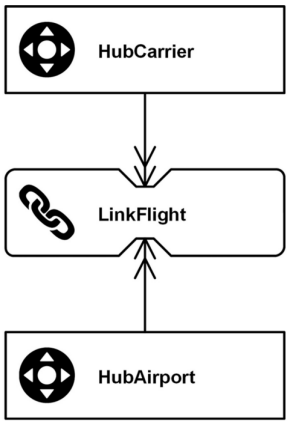
- **Hashkey (PK):** Als primaire sleutel van de entiteit wordt een gehashte identifier van de entiteit gebruikt.
- **LoadDate:** Datum/tijdstip wanneer de record is ingeladen
- **Record source:** Van welke databron is de record afkomstig?
- **Business key(s):** De business key(s) van de bijhorende entiteit

De business key is een unieke sleutel, die vaak een betekenis heeft naar de business. Voorbeelden zijn: ISBN-nummers, Klantnummer, Chassisnummer, ...

Hubs zijn vooral heel handig wanneer er meerdere databronnen zullen zijn, zo kan je meerdere bronnen aan een hub hangen. In een hubs zit nooit andere data buiten een hash key, metadata en business keys.

Links

Wanneer we 2 hub-entiteiten willen verbinden, zullen we ze niet rechtstreeks met elkaar verbinden. Twee hub-entiteiten worden namelijk verbonden door middel van een link. Andere verantwoordelijkheden voor een link zijn hierarchieën, redefinities of business termen. De bedoeling is om een zo laag mogelijke granulariteit te creeëren. Links zorgen ervoor dat het data vault model heel flexibel wordt en makkelijk uitbreidbaar is.



Figuur 2.4: Link entiteit die 2 hub entiteiten met elkaar verbindt. (Daniel Linstedt, 2016).

Ook bij links moeten er een aantal attributen aanwezig zijn:

- **Hashkey (PK):** Als primaire sleutel van de entiteit worden alle business keys gehasht naar 1 sleutel.
- **LoadDate:** Datum/tijdstip wanneer de record is ingeladen
- **Record source:** Van welke databron is de record afkomstig?
- **Business key(s):** Alle business key(s) van de 2 gelinkte hub-entiteiten

Sattelieten

In een satteliet worden alle gegevens gestockeerd dat een business object, relatie of transactie beschrijft. In de entiteit zelf is het belangrijk dat de historiek wordt bijgehouden. (Daniel Linstedt, 2016)

Bij een satteliet vinden we minstens volgende attributen terug:

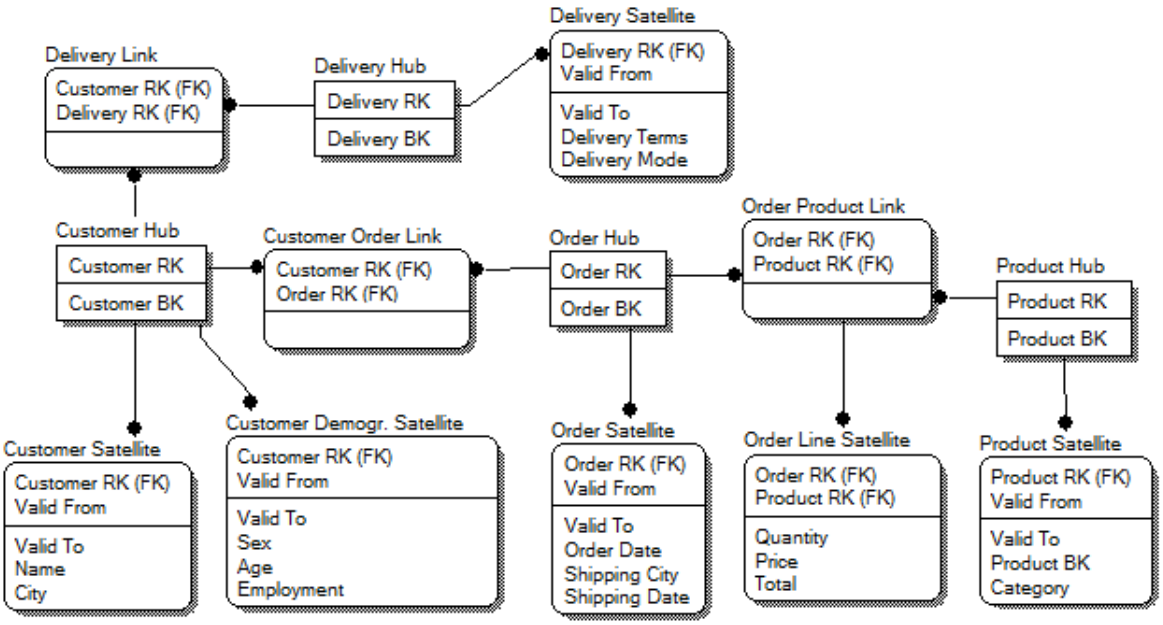
- **Parent Hashkey (PK):** Als primaire sleutel van de entiteit worden alle business keys gehasht naar 1 sleutel.
- **LoadDate:** Datum/tijdstip wanneer de record is ingeladen
- **Record source:** Van welke databron is de record afkomstig?
- **End load date:** Hierin wordt het moment geladen wanneer de entiteit niet meer gebruikt wordt (belangrijk voor het bewaren van de historiek).

Een satteliet hoort bij een hub of een link. Een hub en een satteliet vormen een bepaald business object.

Data Vault schema

Wanneer we zowel hubs, links als sattelieten samengieten in één schema, bekomen we een data vault. Wel zijn er nog enkele belangrijke opmerkingen:

- Hubs mogen nooit rechtstreeks verbonden met elkaar, dit moet altijd gebeuren via een link (anders verliest het model zijn flexibiliteit).
- Sattelieten kunnen zowel met hubs als links verbonden worden.
- Hubs/links kunnen meerdere sattelieten hebben: deze staan meestal voor verschillende databronnen.
- Een satteliet kan maar verbonden worden met 1 hub of link.



Figuur 2.5: Een voorbeeld van een data vault model (Bukhantsov.org)

Dit model heeft veel flexibiliteit te bieden. Enerzijds kan er gemakkelijk nieuwe hubs toevoegen door een nieuwe link te leggen. Ook kan er heel gemakkelijk een nieuwe gegevensbron toevoegen, dit wordt gedaan door een nieuwe satteliet toe te voegen aan een bestaande hub.

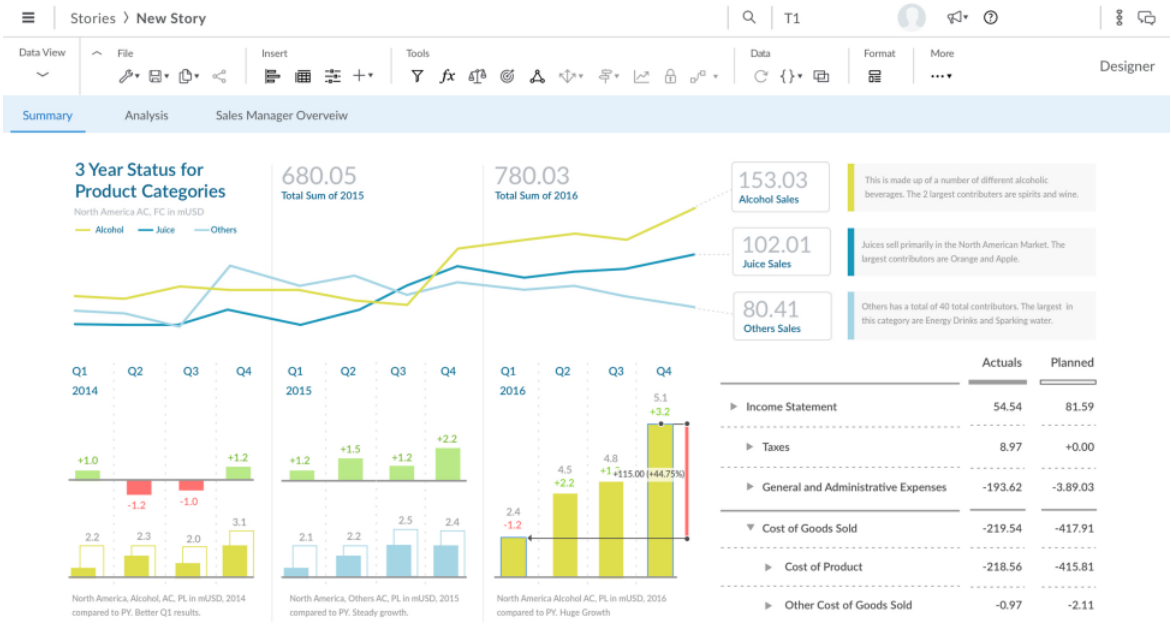
2.4 Rapporteringsomgevingen

Wanneer alle data in de data warehouse ingeladen en getransformeerd is, moet het mogelijk zijn om visuele en interactieve rapporten op te stellen. Op basis van deze rapporten, kan het beleid beslissingen gaan nemen. Ook kunnen dashboards opgesteld worden voor werknemers die geen beleid voeren. De bedoeling in deze sectie is om een schets te geven over welke rapporteringsomgevingen op de markt beschikbaar zijn.

2.4.1 SAP Analytics Cloud

Bij SAP Analytics Cloud kan er gebruik gemaakt worden van realtime analytics. Hiervoor is live data nodig en deze wordt opgehaald in het transactionele systeem. Indien gewenst, kan er toch nog steeds gewerkt worden met een batch die 's nachts geladen wordt.

Analytics cloud biedt heel veel mogelijkheden om interactieve dashboards te ontwerpen. Waar SAP Analytics Cloud zich onderscheidt met de andere spelers, is dat er in deze omgeving planning kan worden toegepast. Er kunnen budgetten opgesteld worden voor de komende jaren (bijvoorbeeld IT-kosten), en deze kunnen dan vergeleken worden met de actuele kosten op dat momenten. Wanneer een bedrijf deze planning-feature wil implementeren, moeten ze hiervoor een pak meer geld op tafel leggen.



Figuur 2.6: Een voorbeeld van een dashboard gemaakt met SAP Analytics Cloud (sap.com)

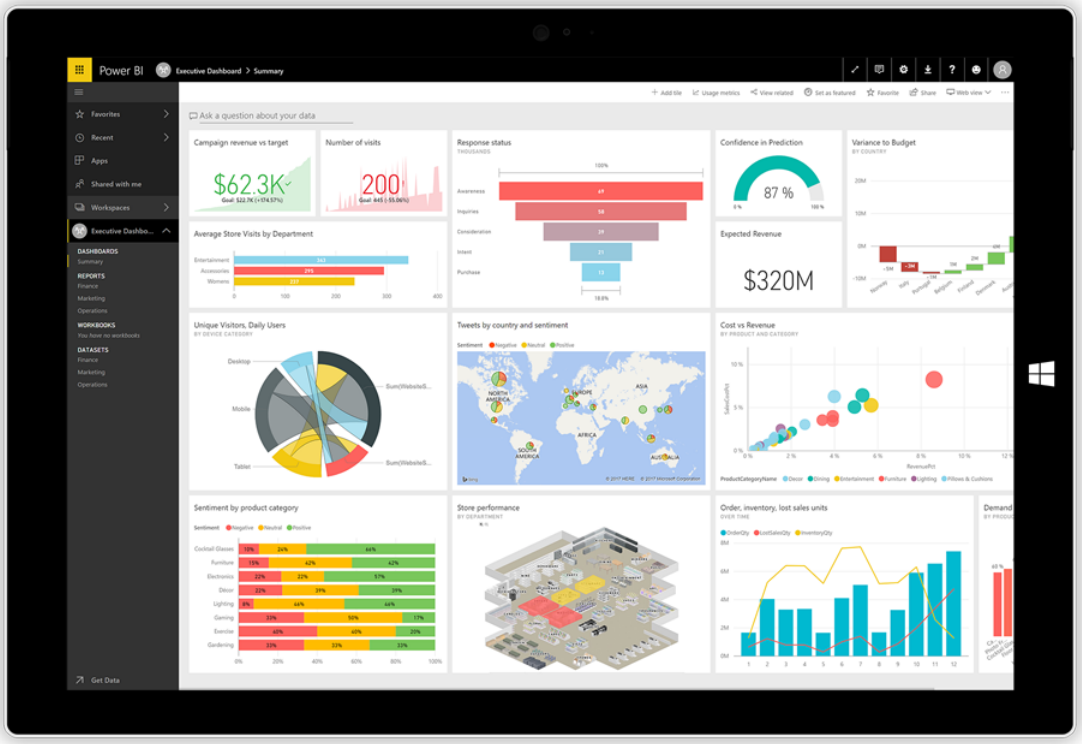
2.4.2 Power BI

Power BI is het programma bij uitstek die gebruikt wordt bij datavisualisatie van ontwikkelaar Microsoft. PowerBI is een uitstekende keuze wanneer bedrijven Office 365 en microsoft Dynamics geïmplementeerd hebben in hun infrastructuur.

In tegenstelling tot SAP Analytics cloud kan je Power BI wel on-premise installeren. Maar indien gewenst, kan er steeds gebruikt gemaakt worden van een cloud-omgeving.

2.4.3 Andere omgevingen

Naast de mogelijkheden die 2 grootmachten in de ERP-markt aanbieden, zijn er nog een aantal andere opties beschikbaar om rapporteringen visueel aantrekkelijk te maken. Enkele



Figuur 2.7: Een voorbeeld van een dashboard gemaakt met Power BI (microsoft.com)

voorbeelden zijn:

- Tableau
- Sisense
- Domo
- IBM Watson Analytics
- ...

3. Methodologie

4. Conclusie

A. Onderzoeksvoorstel

Het onderwerp van deze bachelorproef is gebaseerd op een onderzoeksvoorstel dat vooraf werd beoordeeld door de promotor. Dat voorstel is opgenomen in deze bijlage.

A.1 Introductie

Wanneer het management in het bedrijf een strategische of tactische beslissing wil maken, is deze beslissing gebaseerd op data afkomstig uit verschillende databronnen. Daarom is er bij grote ondernemingen (en tegenwoordig ook bij KMO's) nood aan een rapporteringssysteem. Voor het opstellen en onderhouden van datawarehouses wordt een bepaald budget voorzien. Relaties leggen tussen verschillende data is dan ook een grote uitdaging. Daarom is het dus belangrijk dat het model op de juiste manier ontworpen wordt om kosten te beperken wanneer men de datawarehouse wil onderhouden/uitbreiden. Hiervoor bestaan verschillende modelleertechnieken. In dit onderzoek worden enkel het Kimball dimensioneel modelleren en Data Vault 2.0 vergeleken. We proberen in dit onderzoek de volgende vraag te beantwoorden: **Waar zitten de verschillen bij het modelleren met Data Vault 2.0 en het dimensioneel modelleren?**

Ook zal er een antwoord trachten gevonden te worden op volgende deelvragen:

- Zijn er verschillende manieren van aanpak mogelijk?
- Hoe flexibel/schaalbaar zijn beide systemen?
- Is er een verschil in performantie?
- Hoe verschillen de technieken naar onderhoud toe?

Bij DHL Pharma Logistics gebeurt het berekenen van de KPI's (Key Performance In-

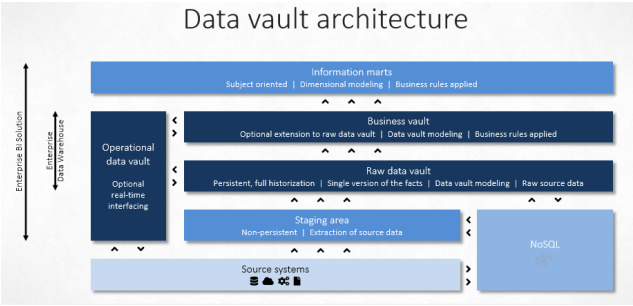
dicators) nog altijd manueel. Zo worden de KPI's berekend via een rekenmachine en handmatig ingevoerd in een Excel-bestand. De informatie die nodig is om verschillende berekeningen te maken is afkomstig uit verschillende databronnen (mainframe, Cronos, Excel-sheets, ..). Dit neemt veel tijd in beslag, dus beslist de firma om een deel van de KPI's te automatiseren. Hiervoor zal een datawarehouse moeten opgezet worden. Deze zal gemodelleerd worden in Data Vault 2.0. Maar is dit wel de beste oplossing? Dit onderzoek zal uitwijzen of Data Vault 2.0 wel degelijk de beste oplossing is.

A.2 Literatuurstudie

A.2.1 Data Vault 2.0

Architectuur

De architectuur van Data vault bestaat voornamelijk uit 3 lagen: De staging area, de raw data vault area en de Business vault. De staging area wordt gebruikt om alle data tijdelijk te stockeren. Daarna wordt de data doorgezonden naar de volgende laag: de raw data vault. Hierbij wordt de architectuur omgevormd naar een data vault. Hierop worden dan data manipulaties gemaakt en wordt de data doorgezonden naar de volgende laag, de business vault. Data marts worden gecreëerd op de business vault (Linstedt, 2015).



Figuur A.1: Data Vault architectuur voorgesteld door Stroobants (2018).

Entiteittype's

Bij data vault wordt er een onderscheid gemaakt tussen 3 verschillende entiteiten: hub, link en satelliet. In een hub wordt een hash sleutel opgeslagen die gebaseerd is op de identifier van die entiteit en metadata (zoals de bron en wanneer de record is ingevoerd). Een link is entiteit die verantwoordelijk is om verschillende hubs met elkaar te verbinden. Hierin worden de hash sleutels van de verbonden entiteiten in opgeslagen. Satellieten kunnen verbonden worden met hubs en links. Deze bevatten de inhoudelijke data van de entiteit.

A.2.2 Dimensioneel modelleren

Architectuur

Bij het dimensioneel modelleren via Kimball is er 1 enkele laag, hierin worden alle operaties uitgevoerd (ETL: Extraction, Transaction en Load). De data wordt ingeladen in een ster-schema. Op deze laag worden dan data marts gebouwd. (Jukic, 2006)

Entiteittype's

Bij deze techniek bestaan er 2 entiteittype's: feit tabellen en dimensionele tabellen. De feit tabellen bevatten alle transactionele data, data waarop je eigenlijk berekeningen kan maken. Dimensionele tabellen bevatten meer informatie over de transactionele data.

A.3 Methodologie

Voor dit onderzoek zullen er twee datawarehouses opgezet worden in een SAP HANA-omgeving. De eerste datawarehouse zal gemodelleerd worden in Data Vault 2.0, de andere in een dimensioneel model. De SAP HANA omgeving is 'on-premise' die draait in een Microsoft Azure omgeving. Het modelleren zal deels gebeuren in Eclipse (die een remote-verbinding maakt met Azure) en deels via een web IDE voor HANA (Xsengine). Wanneer beide datawarehouses operationeel zijn, kan er gestart worden met de vergelijking. De datawarehouses zullen gebaseerd zijn op KPI's die gedefinieerd zijn bij DHL Pharma Logistics.

A.3.1 Performatie

Om de performantie van beide systemen te vergelijken, zullen er een aantal verschillende queries uitgevoerd worden op data marts gebaseerd op deze datawarehouses. Op basis van uitvoeringstijd kunnen we deze dan met elkaar vergelijken. Zo kunnen we te weten komen of er wel degelijk een verschil is tussen beide architecturen in performantie en hoe groot de verschillen zijn.

A.3.2 Audit

Stel dat er op 2 verschillende databronnen klantengegevens opgeslagen wordt, zal er een keuze moeten gemaakt worden. Van welke bron haal ik mijn gegevens? Indien er verschillende problemen optreden met data, willen we graag kunnen onderzoeken waar het probleem zich heeft voorgedaan. Hiervoor voegen we META-data toe aan de data die ons verteld waar en wanneer de data werd opgehaald.

A.3.3 Schaalbaarheid

Hoe wordt er omgegaan met grote hoeveelheden data in beide architecturen? Merken we hier een significant verschil? Zien we de uitvoeringstijden lineair/exponentieel stijgen?

A.3.4 Flexibiliteit

De vereisten voor rapportering verandert vaak bij bedrijven. Soms moeten KPI's worden toegevoegd, soms moeten deze gewijzigd worden. Maar wat als er databronnen in het bedrijfsnetwerk toegevoegd? Hoe gemakkelijk kunnen deze wijzigingen gemaakt worden in beide architecturen? Dit zullen we onderzoeken door een nieuwe KPI toe te voegen aan het systeem.

A.4 Verwachte resultaten

Op basis van het uitgevoerde onderzoek zullen we hiervan een resultaat kunnen opstellen. Ik verwacht dat beide technieken zijn voordelen en nadelen zullen hebben. Zo zal Data Vault 2.0 een modelleertechniek zijn die zeer flexibel is, maar dit zal ten koste gaan van de performantie. Het dimensionele model zal zo performanter zijn, maar weinig flexibiliteit bieden.

A.5 Verwachte conclusies

Aangezien Data Vault 2.0 veel flexibiliteit te bieden heeft, zal dit de beste oplossing zijn wanneer alle data verspreidt staat op verschillende systemen. Bij Data Vault 2.0 is het namelijk mogelijk gemakkelijk nieuwe databronnen toe te voegen in een datawarehouse. Maar wanneer men de de data marts wil ontwerpen, zal men nog steeds moeten gebruik maken van dimensioneel modelleren. Wanneer een bedrijf weinig databronnen heeft en deze weinig veranderen, is dimensioneel modelleren de betere oplossing.

Voor DHL Pharma Logistics zal Data Vault 2.0 dan ook de beste oplossing zijn, aangezien hun data verspreidt staat over enkele systemen. Zo kunnen ze hun KPI's ook nog beter definiëren en makkelijker aanpassen in het systeem.

Bibliografie

Daniel Linstedt, M. O. (2016). *Building a scalable data warehouse with data vault 2.0* (A. Invernizzi, Red.). Todd Green.

Jukic, N. (2006). Modeling strategies and alternatives for dat warehousing projects. *Communications of the ACM*.

Justin Langseth, G. S., Nithi Vivatrat. (2005). SCHEMA AND ETL TOOLS FOR STRUCTURED AND UNSTRUCTURED DATA.

Linstedt. (2015). *Building a Scalable Data Warehouse with Data Vault 2.0*.

Markus Helfert, C. S., Gregor Zellner. (2002). Data Quality Problems and Proactive Data Quality Management in Data-Warehouse-Systems.

Pollefliet, L. (2011). *Schrijven van verslag tot eindwerk: do's en don'ts*. Gent: Academia Press.

Ralph Kimball, M. R. (2013). *The data warehouse toolkit: Third edition*. Wiley.

Satyanarayana, R. (2010). DATA WAREHOUSING, DATA MINING, OLAP AND OLTP TECHNOLOGIES ARE ESSENTIAL ELEMENTS TO SUPPORT DECISION-MAKING PROCESS IN INDUSTRIES. *International Journal on Computer Science and Engineering*.

Stroobants, J. (2018). Modern data warehousing with data vault in SAP HANA.

Vassiliadis, P. (2000). *Data Warehouse Modeling and Quality Issues* (proefschrift, NATIONAL TECHNICAL UNIVERSITY OF ATHENS).