# Inhoudsopgave

[**Inhoudsopgave 1**](#_6w4sytotfok)

[**H1 - Behoeftebepaling & Data-inventarisatie 2**](#_33pl749ydk0g)

[P1 - Formulering stageopdracht 2](#_ij1iwqs2c9md)

[Bedrijf en Probleemdomein 2](#_zhljhcuvojdk)

[Aanleiding en Probleemstelling 3](#_3vnneg9subqy)

[Doelstelling 3](#_fgqsctz4aog8)

[Concrete Werkzaamheden 4](#_xtowfdv0tbgc)

[P2 - Bedrijfsmodellering 5](#_iuj2d1f4g54p)

[P3 - Brongegevens analyse 6](#_ejp8y8ljyl6h)

[Diagram 6](#_nxo2z95ifnx9)

[Toelichting 7](#_i20y5vhg5375)

[**H2 - Introductie Python 10**](#_edkrbtomiu0o)

[**H3 - Algoritmes en Datastructuren 10**](#_s0yrk75hmr4m)

[**H4 - Data Warehouse-ontwerp 11**](#_uufhf0mxrqai)

[P1 - Dimensie-ontwerp 11](#_nuppr9q1fmgf)

[Feiten 11](#_y5r1mio3qm2l)

[Dimensietabellen 12](#_uenpfq44me0)

[P2 - Feit-tabellen en dimensie-associaties 16](#_4kkbmooknjcc)

[Verborgen dimensies 16](#_iztd1wegg43d)

[Feit-tabellen 16](#_bsuf4d1lfky3)

[Sterdiagram 17](#_i050x2t4rcvb)

[P3 - ETL-ontwerp en ontwikkeling 18](#_p49r17i7v9i6)

[**H5 - Data Warehouse-optimalisatie 19**](#_tee422xwr3e)

[P1 - Voortbouwing H4.P3 19](#_jomjdth65lpm)

[P2 - Converteren naar Slowly Changing Dimensions 19](#_bpi9ujec45rz)

[P3- Conversie naar Executable 20](#_4w1ofanjo27m)

# H1 - Behoeftebepaling & Data-inventarisatie

## P1 - Formulering stageopdracht

* Bedrijf & Probleemdomein: 200-250 woorden.
* Aanleiding & Probleemstelling: 200-250 woorden.
* Doelstelling: 200-250 woorden.
* Concrete werkzaamheden: 200-250 woorden.
* Bedrijf : Great Outdoors

### Bedrijf en Probleemdomein

De stage zal gedaan worden bij Great Outdoors. Great Outdoors is een groothandel voor wandel- en kampeerartikelen die als doel heeft hun klanten producten van hoge kwaliteit te leveren. Zij doen dit door handel te doen met verschillende leveranciers, waar zij de desbetreffende producten van verkrijgen. Het bedrijf doet handel op de wereldmarkt en heeft talloze tevreden klanten op elke continent. In 1980 in Amsterdam opgericht als een klein bedrijfje, maar door de jaren heen uitgegroeid tot één van de grootste bedrijven in de branche. Het bedrijf blijft groeien en de besluitvorming rondom strategie wordt gedaan op het hoofdkantoor. Door de groei blijft de schaal van de bedrijfsvoering ook vergroten.  
  
De stage vindt zich plaats op de hoofdkantoor in Amsterdam. Zoals eerder vermeld worden de strategische besluiten daar gemaakt. Om deze besluiten te maken wilt het bedrijf gebruik maken van datagericht besluitvorming. Voor dit traject ligt onze focus op het datagericht werken. Uit onderzoek van bedrijven die op deze manier werken blijkt dat strategische besluiten met als basis datagedreven werken zorgt voor concrete plannen met minder onzekerheden rondom het bedrijfsvoering. De werknemers vanuit het bedrijf hebben aangegeven dat zij in de toekomst volwaardig ook gebruik willen maken van allerlei andere datagerichte werk-technieken.

### Aanleiding en Probleemstelling

*[Beschrijving van het probleem en al geprobeerde oplossingen ]*

In recente tijden heeft Great Outdoors steeds meer moeite met inzage krijgen in haar prestaties, mede dankzij haar explosieve groei. Hoewel het bedrijf voldoende informatie verzamelt tijdens de dagelijkse gang van zaken, heeft het geen manier om deze informatie beschikbaar te stellen om te helpen in het maken van beslissingen.

Hierdoor Great Outdoors mist momenteel inzage in belangrijke functies betreffende het afhandelen van orders, doelstellingen, trainingen, klanttevredenheid, voorraadniveaus en het voorspellen van toekomstige orderaantallen.

Er is geen duidelijkheid over hoeveel omzet en winst per product wordt geboekt, of wat voor invloed het aantal teruggebrachte artikelen(desondanks de redenen daarachter) invloed heeft op de tevredenheid van klanten. Ook is het lastig om accuraat toekomstige orderaantallen te voorspellen omdat het onduidelijk is tot hoeverre de gestelde verkoopdoelen daadwerkelijk gehaald worden.

Doordat een goede representatie van deze informatie ontbreekt, heeft het bedrijf moeite met het voorspellen van positieve en negatieve consequenties van potentiële bedrijfsuitbreidingen op zowel huidige als toekomstige bedrijfsprestaties. Verder leidt de huidige stand van zaken er ook toe dat belangrijke strategische beslissingen op basis van gevoel gemaakt moeten worden in plaats van dat er een besluit op onderbouwde feiten genomen kan worden.

### Doelstelling

*[Doel van de opdracht SMART beschrijven in een generieke manier, zodat verschillende soorten werknemers het doel zouden kunnen behalen]*

Het doel van de opdracht is ervoor zorgen dat er duidelijkheid komt rondom de inzichten die behaald kunnen worden uit de data die we verzamelen. Wij willen dit doen met verschillende statistieken van relevante prestaties van het bedrijf. Het behalen van dit doel zorgt voor betere sturing van de (vooral grootschalige)bedrijfsprocessen.

Een duidelijk overzicht van deze gegevens moet ervoor zorgen dat de verschillende werknemers van het bedrijf duidelijke, accurate informatie beschikbaar hebben om te raadplegen tijdens hun dagelijkse activiteiten om zo de productiviteit te verhogen. Voor het management van het bedrijf moet een inzage in deze gegevens het mogelijk maken om met onderbouwde, feitelijke informatie beslissingen te nemen voor de sturing van het bedrijf.  
  
Met de voltooiing van de opdracht moet het mogelijk worden om de klanttevredenheid te peilen en de relatie tussen dat en het aantal gekochte en geretourneerde producten bloot te stellen. De voorraad in het magazijn moet accuraat af te stemmen zijn op de vraag vanuit klanten. Er moet duidelijkheid komen over of de trainingen voor werknemers daadwerkelijk een positieve invloed hebben op de prestaties van het bedrijf, en over de behaalde omzet en winst per product.  
  
Over het geheel heeft het behalen van deze doelen als gevolg, dus ook een grote bijdrage voor andere bedrijfsdoelen, zoals bijvoorbeeld winst en voorspelbaarheid van de markt. Dit doordat belangrijke strategische beslissingen genomen worden op basis van data in plaats van intuïtie.

### Concrete Werkzaamheden

*[De concrete taken en projectmanagement methodes]*

Voor Great Outdoors is het de bedoeling dat er een dashboard gebouwd wordt dat hun inzage biedt in de verschillende, momenteel ongesorteerde statistieken van het bedrijf.

Dit dashboard moet de medewerkers van het bedrijf verschillende inzichten bieden om hen te ondersteunen in hun dagelijkse werkzaamheden. Deze inzichten moeten hun duidelijkheid bieden door concrete, actuele cijfers en waar toepasselijk bijbehorende grafieken te tonen.   
Verder moet het dashboard gebruikers de mogelijkheid bieden om toekomstige data te voorspellen. Het moet mogelijk zijn om op basis van informatie uit interne systemen van het bedrijf de consequenties van een strategisch besluit op de winstcijfers te voorspellen.

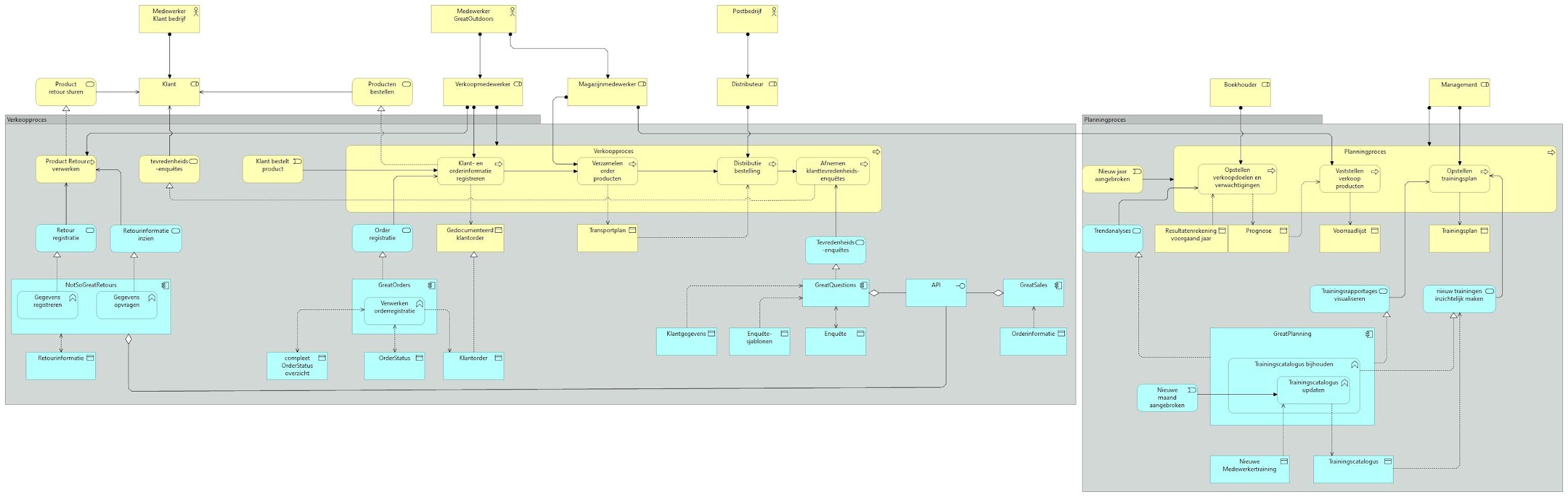
Om deze inzage in de gegevens te realiseren zal er gebruikgemaakt worden van de programmeertaal Python om gegevensanalyse aan te bieden. Voor de implementatie van de data voorspelling zal er gebruikgemaakt worden van AI en Machine Learning. Deze zogenaamde Machine Models worden achteraf van een voorspelling gecontroleerd op correctheid en daarop aangepast om iteratief steeds accuratere voorspellingen mogelijk te maken.

*[projectmanagement methode]*  
De opdracht zal doorlopen worden via een AGILE projectmanagement methode om incrementele voortgang toe te staan. Hiermee moet het mogelijk worden om snel veranderingen door te voeren op basis van feedback vanuit de opdrachtgever.

## 

## P2 - Bedrijfsmodellering

*[ Archimate model][screenshots maken om op te splitsen?]*



*[150-200 woorden aan toelichting]*

Bovenstaande diagrammen zijn delen van het opgestelde Bedrijfsmodel. Hierin staan de processen uitgeschreven zoals op te maken uit de aangeleverde beschrijving van de activiteiten en gebeurtenissen binnen Great Outdoors als bedrijf.   
De twee belangrijkste processen zijn het Verkoopproces (Diagram 1) en het planningsproces (Diagram 2).   
In dit diagram zijn zowel de business- als de applicatielaag opgenomen om één overzichtelijk diagram te kunnen tonen.

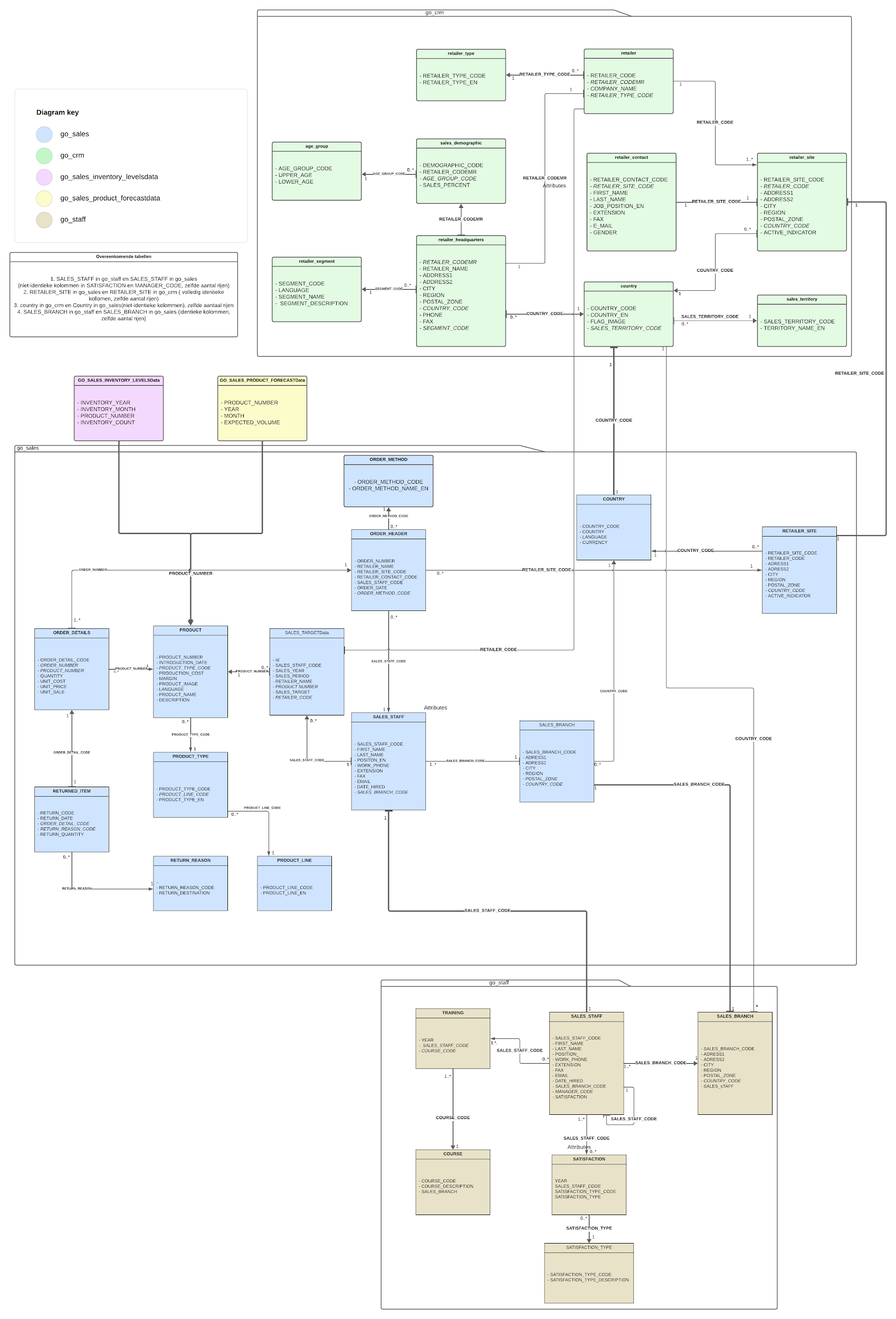
Het verkoopproces wordt gestart als een klant een order plaatst door telefonisch contact op te nemen met een verkoopmedewerker. Deze medewerker is tevens de eindverantwoordelijke voor deze instantie van het proces, wat de reden is dat er een ‘assignment’-relatie bestaat tussen het verkoopproces en de verkoopmedewerker.

Bij een paar andere stappen in het proces zijn andere rollen verantwoordelijk voor die stappen (magazijnmedewerker, distributeur) en zijn de toepasselijke relaties geschetst.   
Onder de business laag bevindt zich de applicatielaag. Hier staan de verschillende applicaties dat genoemd zijn in de opdrachtomschrijving.

Bij het planningsproces zijn er 3 belangrijke rollen, elk met een ‘assignment’-relatie naar hun relevante taak. De magazijnmedewerker rol heeft een rol in zowel het verkoop- als het planningsproces. Buiten dat is er geen verbinding tussen de twee processen.

## P3 - Brongegevens analyse

### Diagram



### Toelichting

**Overeenkomende tabellen**

1. *SALES\_STAFF in go\_staff en SALES\_STAFF in go\_sales*   
   (niet-identieke kolommen in SATISFACTION en MANAGER\_CODE, zelfde aantal rijen)
2. *RETAILER\_SITE in go\_sales en RETAILER\_SITE in go\_crm*   
   ( volledig identieke kolommen, zelfde aantal rijen)
3. *country in go\_crm en Country in go\_sales*  
   (niet-identieke kolommen, zelfde aantal rijen)
4. *SALES\_BRANCH in go\_staff en SALES\_BRANCH in go\_sales*   
   (identieke kolommen, zelfde aantal rijen)

**Verbindingen tussen tabellen**

go\_staff.SALES\_BRANCH gekoppeld aan go\_crm.country via COUNTRY\_CODE. Dit is gekoppeld met deze versie van Country omdat het de kolom SALES\_TERRITORY\_CODE bevat.

GO\_SALES\_INVENTORY\_LEVELSData gekoppeld aan go\_sales.PRODUCT via PRODUCT\_NUMBER

GO\_SALES\_PRODUCT\_FORECASTData gekoppeld aan go\_sales.PRODUCT via PRODUCT\_NUMBER

**Feiten en Dimensies**   
  
  
De diagram is een model van 3 database bronnen.

Het gedeelte in het blauw is de diagram voor de go\_sales database. In de diagram zijn de feiten

SALES\_TARGETData en Order Details

SALES\_TARGETData is een tabel dat als basis een proces is voor het kwalitatief bijhouden van de data van sales, dus is het een feit. Dit word door de sales staff gebruikt voor verschillende redenen.

Order details is een tabel dat gegevens verzameld voor verschillende functies zoals het retouren en verzenden van items. een proces dus een feit

De dimensies voor SALES\_TARGETData zijn

PRODUCT --> PRODUCT\_TYPE --> PRODUCT\_LINE

SALES\_STAFF --> SALES\_BRANCH

ORDER\_HEADER --> ORDER\_METHOD

RETAILER\_SITE --> COUNTRY

(--> verwijst naar de foreign key die word gerefereerd uit de desbetreffende tabel)

ORDER\_HEADER word gebruikt om de orders te koppelen met een aantal dimensies. De functie ervan kan bijvoorbeeld voor de sales staff zicht op specifieke orders hebben

Dimensies:  
  
PRODUCT  
PRODUCT\_TYPE  
PRODUCT\_LINE  
SALES\_STAFF  
SALES\_BRANCH  
ORDER\_HEADER  
ORDER\_METHOD  
RETAILER\_SITE  
COUNTRY  
  
Feiten:   
SALES\_TARGETData  
ORDER\_DETAILS  
ORDER\_HEADER

Dimensies bij SALES\_TARGETData:

PRODUCT

PRODUCT\_TYPE

PRODUCT\_LINE

SALES\_STAFF

SALES\_BRANCH

Dimenies bij ORDER\_DETAILS:

PRODUCT

PRODUCT\_TYPE

PRODUCT\_LINE

ORDER\_HEADER

ORDER METHOD

RETAILER\_SITE

COUNTRY

SALES\_STAFF

SALES\_BRANCH

Dimensies BIJ GO\_SALES\_INVENTORY\_LEVELSData:

PRODUCT

PRODUCT TYPE

PRODUCT LINE

DimensiesBIJ GO\_SALES\_INVENTORY\_FORECASTData:  
  
PRODUCT

PRODUCT TYPE

PRODUCT LINE

Database GO\_STAFF:  
  
Dimensies bij TRAINING

COURSE

SALES\_STAFF

SATISFACTION

SATISFACTION\_TYPE

SALES\_BRANCH

# H2 - Introductie Python

Zie de github Repository.

# H3 - Algoritmes en Datastructuren

Zie Github repository.

# 

# H4 - Data Warehouse-ontwerp

## P1 - Dimensie-ontwerp

* PR4-1:
  + Herken de feiten (8 in totaal) en de omliggende dimensies in je eigen gemaakte brongegevensmodel van Great Outdoors (uitwerking week 1, les 3).
  + Maak per dimensie-brontabel een dimensietabel. Voeg waar nodig dimensie-brontabellen samen. Breng de juiste hiërarchieën aan.
  + Modelleer per dimensietabel minimaal één afgeleide dimensiewaarde

### Feiten

1. Salesstaff traint in onderwerp(Go\_staff.training)
   1. Go\_staff.sales\_staff
   2. go\_staff.sales\_branch
   3. go\_staff.course
2. Klant stuurt item retour(Go\_sales.returned\_item)
   1. Go\_sales.order\_details ?
   2. Go\_sales.return\_reason
   3. go\_sales.Product
   4. go\_sales.Product\_Type
   5. go.sales.Product\_Line
3. Salesstaff is tevreden met training(go\_staff.satisfaction)
   1. go\_staff.satisfaction\_type
4. SALES\_TARGETData
   1. go\_sales.Sales\_Staff
   2. go\_crm.Retailer
5. Klant plaatst een order (go\_sales.order\_header)
   1. go\_sales.Order\_Method
   2. go\_sales.Sales\_Staff
   3. go\_sales.Sales\_Branch
   4. go\_sales.Country
6. Klant bestelt een product (go\_sales.order\_details)
   1. go\_sales.Product
   2. go\_sales.Product\_Type
   3. go.sales.Product\_Line
7. SalesStaff constanteert de sales target(go\_sales.SALES\_TARGETdata
8. Aangeven of een retailer site actief is(go\_crm.retailer\_site)

### Dimensietabellen

**Order\_Header**

| Order\_Method | | | | |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
| Order\_Method |  |  |  |  |
| Order\_Method\_Code, Order\_Method\_EN | | | | |

*SALES\_BRANCH\_COUNTRY is afgeleid vanuit go\_sales.country>go\_sales.country\_code>go\_sales.Sales\_branch.country\_code*

| Sales\_Staff | | | | | |
| --- | --- | --- | --- | --- | --- |
| POSITION\_EN | WORK\_PHONE | FAX | *SALES\_BRANCH\_COUNTRY* | |  |
| name | number | number | SALES\_BRANCH\_CODE | |  |
|  |  |  | branch | |  |
| Sales\_Staff | | | | | |

| Sales\_Branch |  |  |  |
| --- | --- | --- | --- |
| COUNTRY\_CODE |  |  |  |
| name |  |  |  |
| REGION |  |  |  |
| name |  |  |  |
| CITY |  |  |  |
| name |  |  |  |
| POSTAL ZONE |  |  |  |
| code |  |  |  |
| Sales\_Branch |  |  |  |
| Sales\_Branch\_Code, Address1, Address2, *POSTAL\_ADDRESS* | | | |

| Country | | | |
| --- | --- | --- | --- |
| Language |  |  |  |
| name |  |  |  |
| Country | | | |
| Country\_Code, Country, Currency\_Name | | | |

| Retailer\_Site | | | |
| --- | --- | --- | --- |
| COUNTRY\_CODE | Active\_Indicator |  |  |
| name | boolean |  |  |
| REGION |  |  |  |
| name |  |  |  |
| CITY |  |  |  |
| name |  |  |  |
| POSTAL ZONE |  |  |  |
| code |  |  |  |
| Retailer\_Site | | | |
| Retailer\_Site\_Code, Retailer\_Code | | | |

| Retailer | | | |
| --- | --- | --- | --- |
| Retailer\_Type\_Code |  |  |  |
| number |  |  |  |
| Retailer | | | |
| Code, CodeMR, CompanyName | | | |

**Staff\_Training**

*Tabel “Course” ingevoegd*

| Sales\_Staff | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| POSITION\_EN | WORK\_PHONE | FAX | SALES\_BRANCH\_CODE | | DATE\_HIRED | |
| name | number | number | number | | date | |
| Manager\_Code |  |  |  |  |  | |
| number |  |  |  |  |  | |

| Sales\_Staff\_Code, FIRST\_NAME, LAST\_NAME,*FULL\_NAME*, EXTENSION, EMAIL | | | |
| --- | --- | --- | --- |
|  |  |  |  |
| Sales\_Branch |  |  |  |
| COUNTRY\_CODE | REGION | CITY | POSTAL ZONE |
| name | name | name | code |

**Order\_Details**

*Tabellen “Product\_Type” en “Product\_Line” ingevoegd.*

| Product | | | | | |
| --- | --- | --- | --- | --- | --- |
| Introduction\_Date | Product\_Type | Language | **Product\_Type** |  | **Product\_Line** |
| date | number | name | Code, Product\_Line,Product\_Type\_EN | | Code,Product\_line\_EN |
| Product | | | | | |
| Product\_Code, Production\_Cost, Margin, Product\_Image, Product\_Name, Product\_Description | | | | | |

**Returned\_Item**

|  |  |  |
| --- | --- | --- |
| Return\_Reason | | |
|  |  |  |
|  |  |  |
| Return\_Reason | | |

| Product | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| Introduction\_Date | Product\_Type | Language | Product\_Type |  | Product\_Line |  |
| date | number | name | Code, Product\_Line,Product\_Type\_EN | | Code,Product\_line\_EN |  |
| Product | | | | | | |
| Product\_Code, Production\_Cost, Margin, Product\_Image, Product\_Name, Product\_Description | | | | | | |

## P2 - Feit-tabellen en dimensie-associaties

* PR4-2:   
  Bouw voort op je uitwerking van vorig practicum:
  + Herken de feiten (8 in totaal) en zet ze om in feitentabellen. Bedenk hierbij in totaal minstens 5 afgeleide meetwaarden.
  + Herken de verborgen dimensies in de feiten en zet deze om naar extra dimensietabellen.
  + Breng de juiste associaties aan tussen de feit- en dimensietabellen.

### Verborgen dimensies

**Tijd**

| Date |
| --- |
| Year |
| number |
| Month |
| number |
| Date |
| date |

### Feit-tabellen

Geaggregeerd feit : Combinatie Order\_Details en Order\_Header

| Feit | Dimensie |
| --- | --- |
| Order\_Details | Product |
| Product\_Line |
| Product\_Type |
| Retailer\_Site |
| Retailer |
| Retailer\_Contact |
| Retailer\_Type |

| Feit | Dimensie |
| --- | --- |
| Returned\_Item | Product |
| Product\_Type |
| Product\_Line |
| Return\_Reason |

### Sterdiagram

<https://lucid.app/lucidchart/e38012cd-acdf-460d-bb94-8b2d4dbd509f/edit?viewport_loc=149%2C264%2C1109%2C526%2C0_0&invitationId=inv_89fd2f0a-98e3-48a9-ba30-21dbaa262ed6>

Dimensiewaardes koppelen?

Product, Sales\_Staff kwamen veel voor

*Een andere kijk op data - deel 2*. (n.d.). Ensior. <https://www.ensior.com/nl/blogs/blogs/1065-een-andere-kijk-op-data-deel-2>

## P3 - ETL-ontwerp en ontwikkeling

* PR4-3:   
  Bouw voort op je uitwerking van het vorige practicum:
  + Maak een database in SSMS, dit wordt je uiteindelijke Data Warehouse.
  + Maak een ETL-schema van elke tabel in het Sterschema.
  + Implementeer elk gemaakt ETL-schema in Python (met Jupyter Notebook). Gebruik hierbij de technische informatie uit week 2.
  + Zorg er ten slotte voor dat je het Sterschema en elk ETL-schema aan je portfolio hebt toegevoegd. Hier word je in week 6 op beoordeeld.

Tabellen aanmaken in Microsoft SQL Server Manager.

Alle velden zijn van het type varchar. Met Pandas zullen numerieke waardes naar integers of decimals vertaald worden.

*How to Delete Rows with Null Values in a Specific Column in Pandas DataFrame | Saturn Cloud Blog. (2023, October 20).* [*https://saturncloud.io/blog/how-to-delete-rows-with-null-values-in-a-specific-column-in-pandas-dataframe/*](https://saturncloud.io/blog/how-to-delete-rows-with-null-values-in-a-specific-column-in-pandas-dataframe/)

*How to Split a Date Column into Separate Day Month Year Columns in Pandas | Saturn Cloud Blog*. (2023, November 2). <https://saturncloud.io/blog/how-to-split-a-date-column-into-separate-day-month-year-columns-in-pandas/>

GfG. (2021, November 16). *Designing the Star schema in data warehousing*. GeeksforGeeks. <https://www.geeksforgeeks.org/designing-the-star-schema-in-data-warehousing/>

# H5 - Data Warehouse-optimalisatie

## P1 - Voortbouwing H4.P3

## P2 - Converteren naar Slowly Changing Dimensions

PR5-2: Maak van elke dimensie in je tot dusver gemaakte Jupyter Notebook, waarin je een datastraat hebt gemaakt die invulling geeft aan het Data Warehouse van Great Outdoors, een Slowly Changing Dimension. Test uitvoerig of correcte relaties met bijbehorende feittabellen behouden blijven.

In deze stap moeten Surrogate Keys geïmplementeerd worden. Surrogate Keys zijn alternatieve primaire sleutels dat gebruikt worden in plaats van de PK's in de brontabel. Een Surrogate Key bestaat uit een nummer, met een tweede kolom voor een timestamp(dit is niet gedeelte van de SK). Op deze wijze kan bij elke wijziging op een rij een duplicaat aangemaakt worden met de gewijzigde gegevens. Dat wilt zeggen dat je de originele rij en de gemodificeerde rij beide in de tabel houdt. Met behulp van de SK en de timestamp kun je zo wijzigingen over een bepaalde periode van tijd bijhouden.

Een SK vervangt de FK. Maar hoe weet de Feittabel dat het de nieuwe, gewijzigde record in de dimensie tabel moet hebben? Misschien is dit te doen door de Foreign Key en de Foreign Surrogate Key te combineren in combinatie met een Trigger of Stored Procedure. Als ~~de Dimension gewijzigd~~ er nieuwe informatie ingevoerd wordt, kan er een Trigger geactiveerd worden dat op basis van de Foreign Key gaat zoeken naar de Foreign Surrogate Key met het meest recente timestamp.

In de brontabellen heb je Primary Keys en Foreign Keys. De verwijzingen naar andere tabellen gebeurt met het gebruik hiervan, en deze waardes hebben een zinvolle betekenis voor de bedrijfsgang.

Surrogate Keys worden echter pas toegevoegd in de Data Warehouse. Hierdoor heb je geen zinvolle waardes om een PK-FK relatie op te baseren. Maar misschien kan dit gedaan worden door in de CREATE TABLE statements gebruik te maken van DEFAULT values of TRIGGERS.

Ik heb besloten om de date dimension niet te gebruiken. Ik heb mijn handen vol aan de surrogate keys, en ik wil niet nog meer tijd kwijt zijn aan het implementeren van de date dimensie. Ik snap wat de meerwaarde is (op basis van de datum opslitsen in eenheiden zoals dag van de week), maar niet hoe je het het best kan implementeren.

Trivedi, J., & Trivedi, J. (n.d.). *Surrogate key in SQL server*. <https://www.c-sharpcorner.com/UploadFile/ff2f08/surrogate-key-in-sql-server/>

Sanchez, E. (2024, February 27). *How to Create a Data Warehouse from SQL Server*. Skyvia Blog. <https://blog.skyvia.com/sql-server-data-warehouse-the-easy-and-practical-guide/>

*Insert into a star-schema*. (n.d.). Stack Overflow. <https://stackoverflow.com/questions/2496610/insert-into-a-star-schema/2499607#2499607>

*How to change datetime format in Pandas | Saturn Cloud Blog*. (2023, October 19). <https://saturncloud.io/blog/how-to-change-datetime-format-in-pandas/>

Roach, J. (2024, January 31). *Mastering Slowly Changing Dimensions (SCD)*. <https://www.datacamp.com/tutorial/mastering-slowly-changing-dimensions-scd>

## P3- Conversie naar Executable

PR5-3: Converteer je tot dusver gemaakte Jupyter Notebook, waarin je een data straat hebt gemaakt die invulling geeft aan het Datawarehouse van Great Outdoors, naar een “executable” Python Script. Volg hiervoor de vier stappen die in het hoorcollege behandeld zijn.

In deze stap moet het Jupyter Notebook project omgezet worden naar een Python project.

(Jupyter Notebook is niet specifiek voor Python - Het is gebouwd met HTML, CSS en Javacsript en kan codeblokken bevatten voor verschillende talen.)

Het converteren van Jupyter Notebook naar Python code is handig omdat je dan een executable hebt dat op een server uitvoerbaar is.

Zoals ik het begrijp pakt de datastraat alle gegevens in de brontabellen en voegt het ze toe aan de Data Warehouse. Een logisch gevolg is dan dat als je de straat een 2e keer runt(ook al zitten er nu nieuwe rijen bij) het opnieuw alle gegevens in de brontabellen pakt.

Stel dat ik Rij 1 en Rij 2 heb, met 5 waardes elk. Ik run de datastraat waardoor ze nu in SQL Server staan, en pas de waarde van kolom 1 aan en run de datastraat opnieuw.

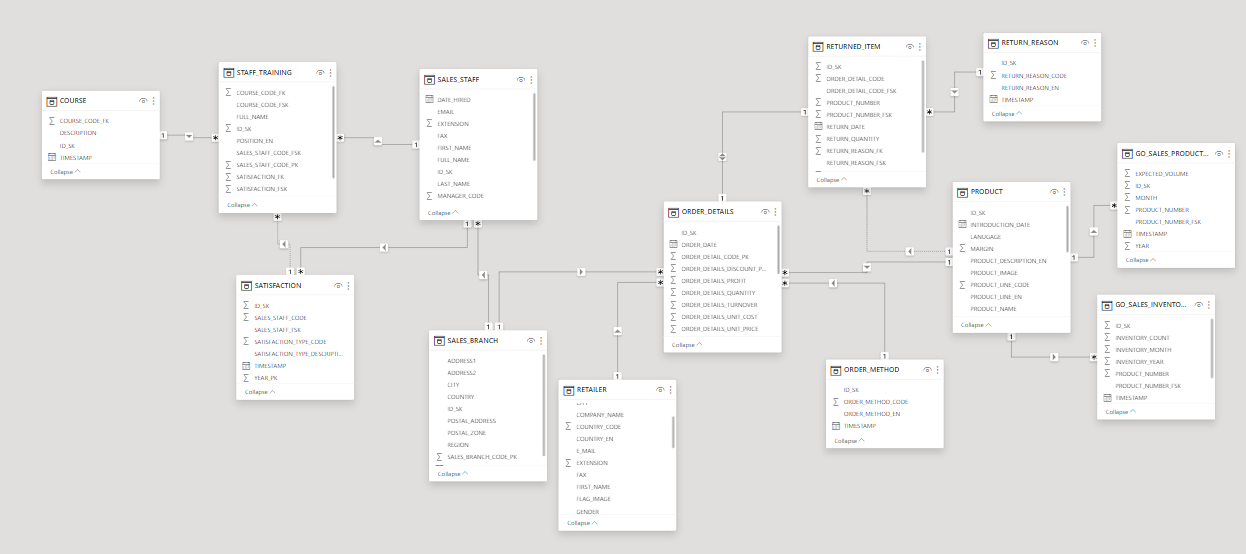
Nu heb ik 4 rijen in de Data Warehouse, in plaats van de 2 originele en de gewijzigde tabel.

Mijn zorg is dat als ik de datastraat meerdere keren run, een wijziging overschreden kan worden. Om het voorbeeld hierboven te gebruiken, ben ik bang dat de wijziging die tijdens run 2 in de datawarehouse is geplaatst ongedaan wordt tijdens run 3.

Maar is mijn denkwijze niet fout hier? Data Warehouses zijn voornamelijk read-only (Een beetje vergelijkbaar met Ledgers zoals in de financiële wereld). Als er iets aan de gegevens aangepast wordt, gebeurt dat niet in de Warehouse, maar in de brongegevens.

# H6 - Data analyse & rapportage

Week 6 gaat over Analyse & Rapportage. Door middel van Power BI (Microsoft Software) moet data inzichtelijk gemaakt worden. Data kan geïmporteerd worden vanuit SQL Server om in Power BI zogenaamde Visuals te maken. Data kan hier op verschillende manieren gecombineerd worden om bepaalde inzchten mogelijk te maken.



<https://www.numerro.io/guides/power-bi-visuals-guide>