DATAWAREHOUSE

Groep: E

KLAS: 3

Groep E

|  |  |
| --- | --- |
| Studentnummer | Naam + voornamen |
| 19005830 | Akash Datadin |
| 19053460 | Roderick Heiligers |
| 19029217 | Richal Rambaran |
| 16076877 | Jimmy van Barneveld |
|  |  |

Inhoudsopgave

[Inhoudsopgave 2](#_Toc73737852)

[1. Inleiding 3](#_Toc73737853)

[2. Planning 4](#_Toc73737854)

[3. Managementvragen Great Outdoor 6](#_Toc73737855)

[3.1 Management vragen 6](#_Toc73737856)

[3.2 Bronnenmodel 7](#_Toc73737857)

[3.3 Gegevensmodel 8](#_Toc73737858)

[3.3 Query Analyse 10](#_Toc73737859)

[4. Datawarehouse ontwerp 17](#_Toc73737860)

[4.1 Dimensie matrix 17](#_Toc73737861)

[4.2 Feitmodel 20](#_Toc73737862)

[4.3 Dimensie & dimensie hiërarchieën 24](#_Toc73737863)

[4.4 Conformer dimension 27](#_Toc73737864)

[4.5 Sterschema 28](#_Toc73737865)

[5. Datawarehouse 29](#_Toc73737866)

[5.1 Create scripts Data warehouse 29](#_Toc73737867)

[5.2 Create scripts Staging 32](#_Toc73737868)

[5.3 ETL mapping 34](#_Toc73737869)

[5.4 ETL dataflows 39](#_Toc73737870)

[5.5 Test/waarborginggegevenskwaliteit 48](#_Toc73737871)

[5.6 Aanpassingen in ontwerp en datawarehouse als gevolg van beveiligingsmaatregelen 53](#_Toc73737872)

[6. Overzicht van de Dashboards 55](#_Toc73737873)

[7. Reflectie & bewijslast 59](#_Toc73737874)

[8. Literatuurlijst 62](#_Toc73737875)

# Inleiding

Dit is het portfolio document van DWH. In dit document wordt de benodigde informatie en de wekelijkse opdrachten beschreven.

Het bevat een strikte planning met daarin beschreven in welke week, welke activiteiten behandeld zullen worden. Ook wordt er uitgebreid per activiteit beschreven wie welke taak op zich zal nemen, wat de voortgang is van de huidige week, en een stukje eigen reflectie over wat beter kan en wat goed is gegaan.

Verder worden de managementvragen besproken en worden de dimensiematrix, feitmodel, eventuele dimensie-hiërarchieën en het uiteindelijke starschema verwerkt in de vorm van tabellen.

Ook worden er scripts gemaakt en in dit document geplaats, en worden er zaken besproken zoals de ETL, testen, waarborgingsgegevenskwaliteiten, en aanpassingen in ontwerpen datawarehouse als gevolg van beveiligingsmaatregelen.

Tot slot zal er een overzicht van het dashboard gemaakt worden, en worden de reflectie en de bewijslast besproken.

# Planning

Neem hier de planning per week op welke jullie opgesteld hebben met de taakverdeling per groepslid. Op welk manier zullen jullie de opdrachten uitvoeren? Waarom hebben jullie voor deze methode gekozen?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Week** | **Planning** | **Groepslid** | **Taakverdeling** | **Werkwijze** |
| 1 | Management vragen opstellen, bronnenmodel maken. | Akash | Inleiding maken en manager vragen opstellen. | Het brainstormen van mogelijke manager vragen. Database tools installeren. |
| Richal | Bronnenmodel maken. | Datawarehouse ontwerp stap 1: een bronnenmodel maken. Het voorbeeld gebruikt uit het pdf-bestand gebruikt om die van ons te maken. Database tools installeren. |
| Roderick | Helpen manager vragen opstellen. | Helpen met het brainstormen van manager vragen. Database tools installeren. |
|  |  |  |
| 2 | Feitmodel en dimensies opstellen. | Akash | Beginselen van feiten creëren | Feiten uit de manager vragen en database tabellen halen. |
| Richal | Helpen dimensies en meetwaardes vast te stellen. | Helpen met dimensies opstellen meetwaardes uit de tabellen halen en zinvolle berekeningen noteren. |
| Roderick | Gegevensmodel genereren en dimensies identificeren om feiten en dimensies te identificeren. | Datawarehouse ontwerp stap 2: gegevensmodel creëren. MS SQL-server functies gebruiken om gegevensmodel te krijgen. Manier vinden om dit ook op MySQL te doen. |
|  |  |  |
| 3 | Geconformeerde dimensies en sterschema. | Akash | Feedback op manager vragen verwerken. | De ontvangen feedback van de docent op Teams verwerken zodat de manager vragen correct gesteld zijn. |
| Richal | Helpen verbeteren van manager vragen. | Hulpverlenen aan feedback verwerking. |
| Roderick | Dimensie hiërarchieën maken. | Dimensies verder uitwerken, geconformeerde dimensies identificeren en hiërarchieën identificeren. |
| Jimmy | Introduceren aan groep. Op de hoogte brengen van werk gedaan in week 1 en 2. | Werk dat in vorige weken verricht is, uitleggen. Hulpverlenen waar nodig. |
| 4 | ETL. | Akash | Mapping schema maken | Beide databasen bekijken en een mapping schema maken |
| Richal | Sterschema diagram en ERD maken fysiek ontwerp | Maak het sterschema diagram en fysiek ontwerp. Houdt rekening mee met feit en dimensietabellen, hiërarchieën, surrogaatsleutels en primary/foreign keys. |
| Roderick | ETL uitwerken | De databases van A en C en GO doorgaan om te kijken waar de data vandaan moet komen |
| Jimmy | SSIS installeren en uitvogelen hoe het werkt. | Links op BlackBoard doorlezen en daarna praktisch toepassen, beginnen van maken ETL scripts. |
| 5 | Staging ETL vervorderen | Akash | Beveiligingsmaatregelen opstellen | Opschrijven welke maatregelen er toegepast moet worden, queries opschrijven en testen |
| Richal | SSIS installeren, sterschema ERD verbeteren. | Zoek uit hoe SSIS geïnstalleerd moet worden en hoe het werkt. |
| Roderick | Test en waarborging | De databases doorlopen binnen de code om te verifiëren of de data in de gemergde database overeen komt met de data binnen de originele databases |
| Jimmy | De anderen helpen met hun taken indien nodig, het alvast doornemen van de op BlackBoard genoemde pagina’s over Power BI zodat ik kan beginnen als we de databasewarehouse af hebben. | Links BlackBoard en overige mogelijk handige informatie doornemen. |
| 6 | Afronding | Akash | ETL installeren en dataflows | Alle benodigdheden installeren in Visual Studio en de diagram maken |
| Richal | ETL handleidingen opzoeken, dataflows maken | Maak de dataflows en voer ze uit om de staging en datawarehouse te vullen met data van de databronnen. |
| Roderick | Controleren en aanvullen/test en waarborging | Verder met databases doorlopen en daarnaast verdere uitleg geven bij eventueel matig onderbouwde stukjes |
| Jimmy | Maken van rapporten en dashboards, de management vragen beantwoorden, data controleren, beveiligingsmaatregelen toepassen. | Het eerst maken van een schets van vraag 1 (in tekst), daarna het daadwerkelijk uitvoeren in Power BI, als dit gelukt is, de overige drie vragen uitwerken, en een dashboard maken, en de rest uitwerken. |

**Extra informatie**

Jimmy van Barneveld is aan het begin van week 3 aan onze groep toegewezen en komt dus niet voor in week 1 en 2.

# Managementvragen Great Outdoor

Neem hieronder de managementvragen op die jullie als groep hebben uitgewerkt. Geef hier de interpretatie van deze managementvragen. Neem ook de gemaakte analyses op de bronsystemen op in relatie met de managementvragen. Doe dit middels SQL-query’s. Neem deze op als embedded object.

## 3.1 Management vragen

Vraag 1: Hoeveel omzet wordt er behaald, en hoeveel producten worden er verkocht per maand, per jaar, per producttype, per product, per land van de klant en per klant?

Vraag 2: Hoeveel retourneringen zijn er per product, per product type, per reden, per klant, welke verkocht zijn door een verkoper op een bepaald moment?

Vraag 3: Hoeveel facturen worden er gemaakt per maand, per werknemer, per klant?

Vraag 4: Wat is het meest bestelde product per maand, per werknemer, per korting van de klant?

Nader feedback is vraag 3 veranderd.

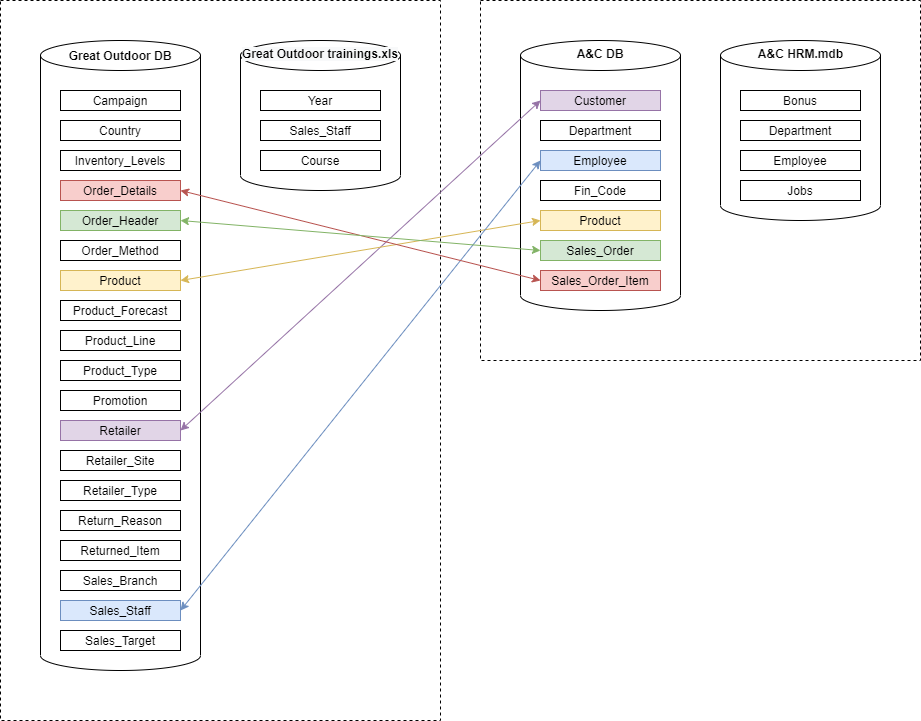
**Verwerkte feedback**

*Vraag 3:* *wat bedoel je precies met het gemiddelde salaris? Hoe bereken je dat? Of bedoel je meer: geef inzicht in de verdiende salarissen van de werknemers per land, per stad?*

*Vraag 4: Als je op zoek bent naar de grootste order qua aantal besteld, dan moet je sowieso het aantal bestellingen bijhouden per klant. Dus ik zou het iets algemener formuleren: wat is het aantal bestelde producten van een klant bij een bepaalde korting...enz. Wat bedoel je met per type (=type klant?)*

## 3.2 Bronnenmodel

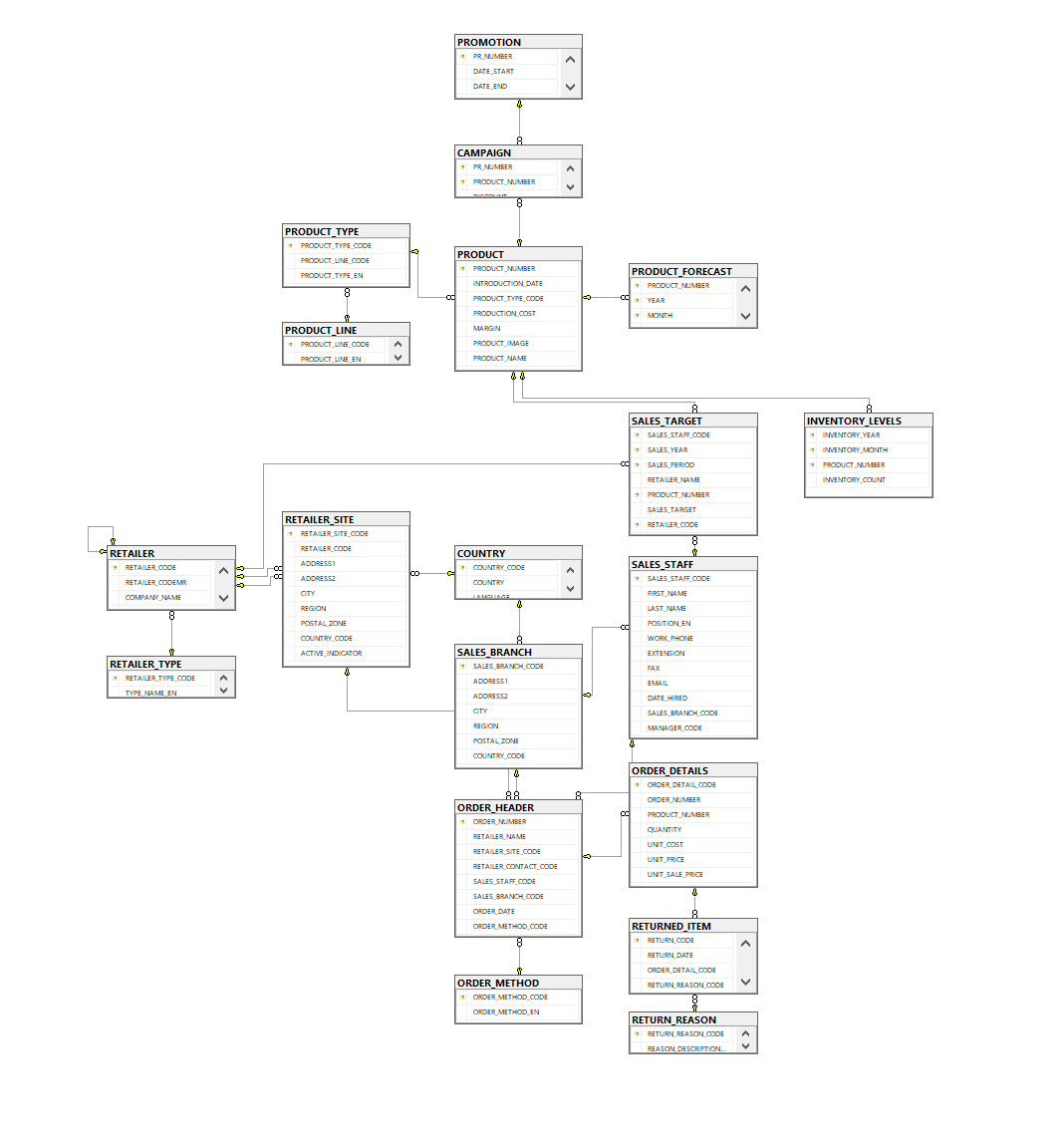
****

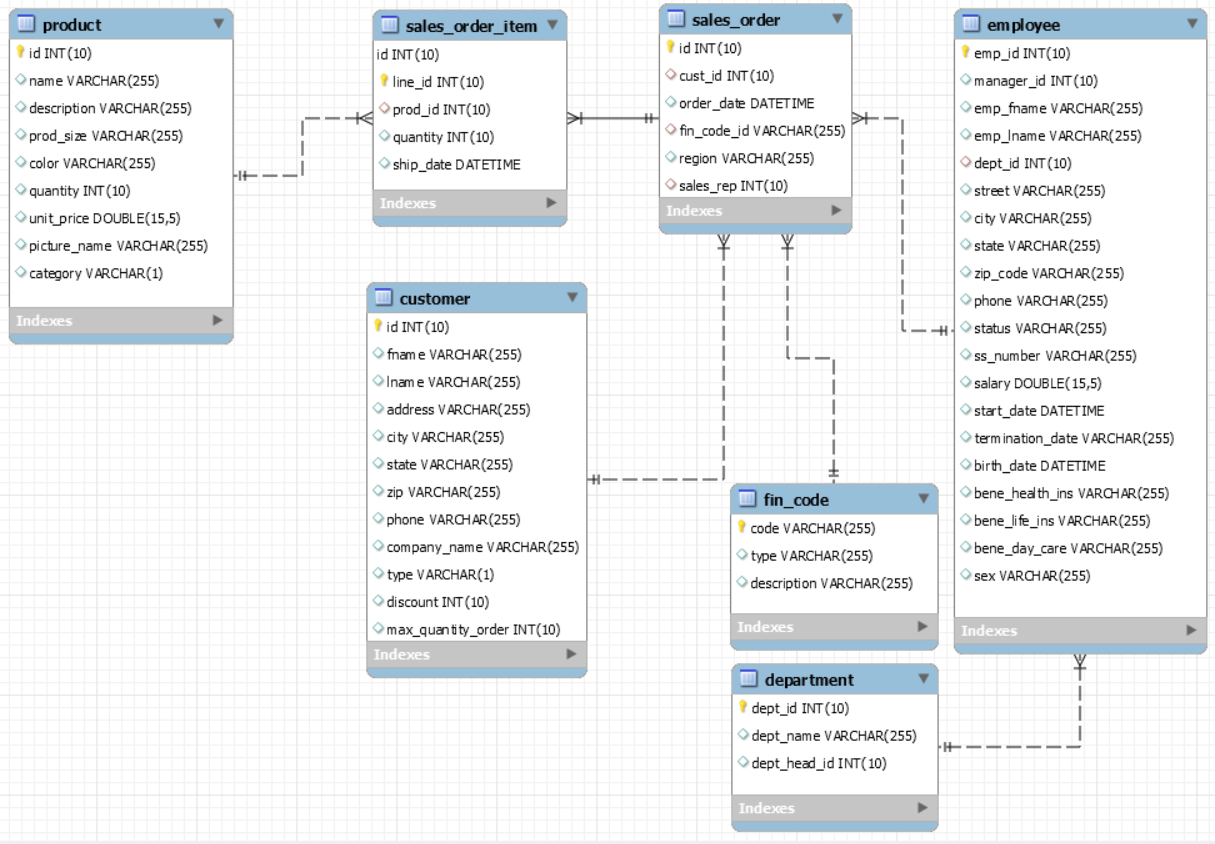


**Extra informatie**

Vergeleken met het standaard bronnenmodel zijn er enigszins bewuste afwijkingen. Overeenkomende databasetabellen zijn met kleuren en pijlen aangegeven. Hiervoor werd gekozen vanwege betere overzichtelijkheid.

## 3.3 Gegevensmodel

**Great Outdoor**

**A&C**

**Toelichting bronnen**

In het bronnenmodel wordt er met kleuren en pijlen duidelijk gemaakt wat de soortgelijke tabellen zijn van de Great Outdoor en A&C databases. Verder wordt er in het gegevensmodel een overzicht gegeven wat de relaties zijn tussen tabellen in de databases. Daarbij laat het ook voor elke tabel de attributen zien met datatype.

## 3.3 Query Analyse

Uit het bronnen- en gegevensmodel kunnen de relevante tabellen afgeleid worden en d.m.v. queries geanalyseerd worden per managementvraag.

1. **Hoeveel omzet wordt er behaald, en hoeveel producten worden er verkocht per maand, per jaar, per producttype, per product, per land van de klant en per klant?**

****

SELECT OD.ORDER\_DETAIL\_CODE,

OH.RETAILER\_NAME,

OD.ORDER\_NUMBER,

CONVERT(DATE, OH.ORDER\_DATE) AS ORDER\_DATE,

P.PRODUCT\_NUMBER,

P.PRODUCT\_NAME,

PT.PRODUCT\_TYPE\_EN,

OD.QUANTITY,

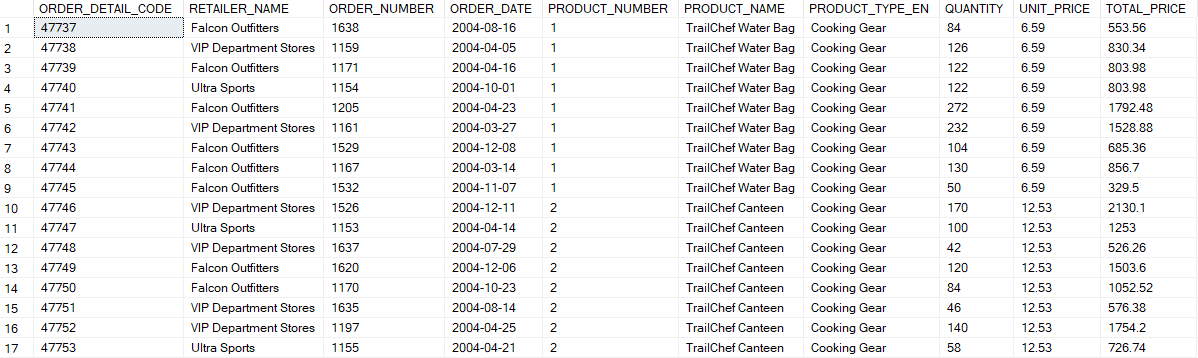
OD.UNIT\_PRICE,

(OD.QUANTITY \* OD.UNIT\_PRICE) AS TOTAL\_PRICE

FROM ORDER\_DETAILS OD

LEFT JOIN PRODUCT P ON OD.PRODUCT\_NUMBER = P.PRODUCT\_NUMBER

LEFT JOIN ORDER\_HEADER OH ON OD.ORDER\_NUMBER = OH.ORDER\_NUMBER

 LEFT JOIN PRODUCT\_TYPE PT ON P.PRODUCT\_TYPE\_CODE = PT.PRODUCT\_TYPE\_CODE

(afbeelding toont slechts een gedeelte)

In de Great Outdoor database zijn er significant meer tabellen dan in de A&C database. Veel join-statements worden gebruikt om data aan elkaar te linken. Met 43.063 aankopen is GO de grootste speler qua omzet in de uiteindelijke datawarehouse. Daarbij zijn ook hoeveelheid en prijs een stuk hoger die in rol spelen in deze management vraag.



SELECT soi.id AS order\_id,

soi.line\_id, c.company\_name,

CAST(so.order\_date AS DATE) AS order\_date,

p.id AS product\_id,

p.name,

p.category,

soi.quantity,

ROUND(p.unit\_price, 2) AS unit\_price,

ROUND((soi.quantity \* unit\_price), 2) AS total\_price

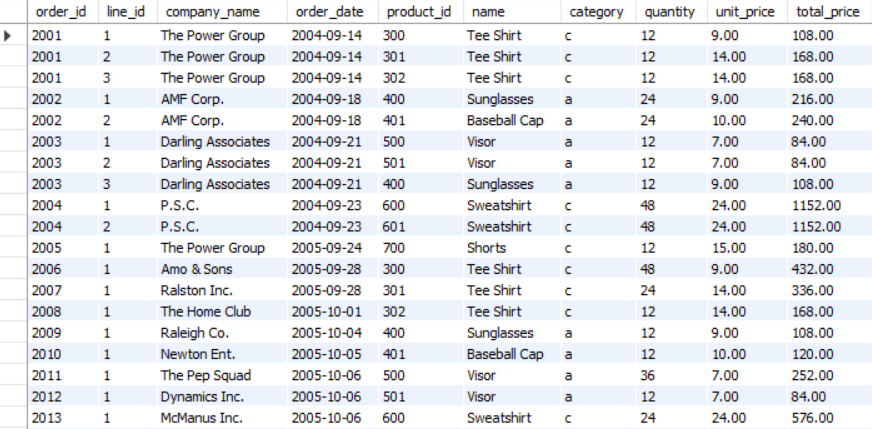
FROM sales\_order\_item soi

LEFT JOIN sales\_order so ON soi.id = so.id

LEFT JOIN customer c ON so.cust\_id = c.id

LEFT JOIN product p ON soi.prod\_id = p.id

ORDER BY order\_id, line\_id ASC



(afbeelding toont slechts een gedeelte)

A&C bestaat uit een stuk minder records vergeleken met Great Outdoor. A&C heeft totaal 1.000 aankopen met lagere prijzen en hoeveelheden, aangezien het bedrijf voornamelijk kleding en accessoires verkoopt. Dit zorgt er ook voor dat er slechts 2 waardes te zien zijn in categorie.

1. **Hoeveel retourneringen zijn er per product, per product type, per reden, per klant, welke verkocht zijn door een verkoper op een bepaald moment?**

****

SELECT RI.RETURN\_CODE,

CONVERT(DATE, RI.RETURN\_DATE) AS RETURN\_DATE,

OH.RETAILER\_NAME,

OH.ORDER\_NUMBER,

RI.ORDER\_DETAIL\_CODE,

SS.SALES\_STAFF\_CODE,

SS.FIRST\_NAME,

SS.LAST\_NAME,

OD.PRODUCT\_NUMBER,

P.PRODUCT\_NAME,

PT.PRODUCT\_TYPE\_EN,

RR.REASON\_DESCRIPTION\_EN,

RI.RETURN\_QUANTITY

FROM RETURNED\_ITEM RI

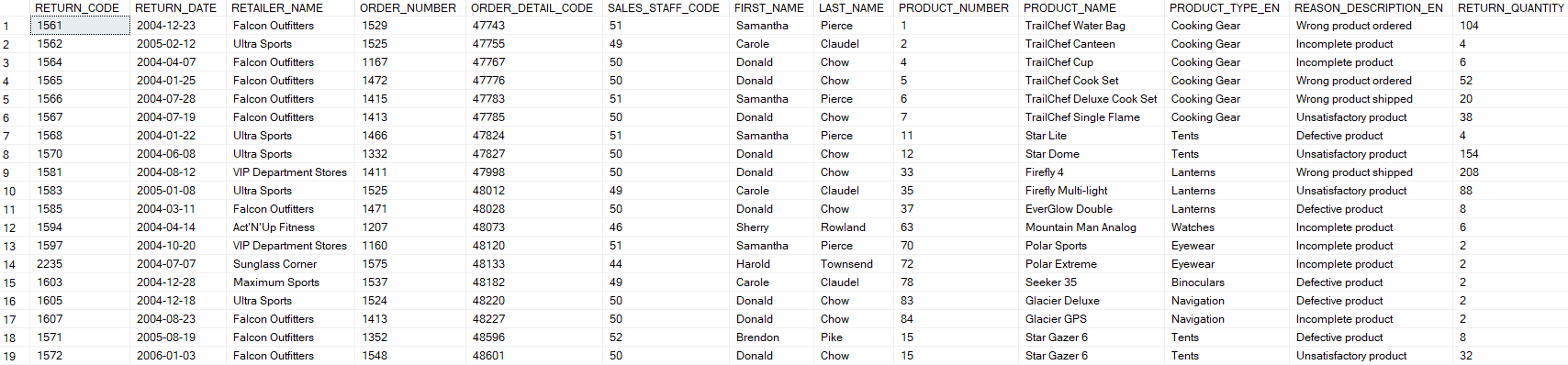
LEFT JOIN RETURN\_REASON RR ON RI.RETURN\_REASON\_CODE = RR.RETURN\_REASON\_CODE

LEFT JOIN ORDER\_DETAILS OD ON RI.ORDER\_DETAIL\_CODE = OD.ORDER\_DETAIL\_CODE

LEFT JOIN PRODUCT P ON OD.PRODUCT\_NUMBER = P.PRODUCT\_NUMBER

LEFT JOIN PRODUCT\_TYPE PT ON P.PRODUCT\_TYPE\_CODE = PT.PRODUCT\_TYPE\_CODE

LEFT JOIN ORDER\_HEADER OH ON OD.ORDER\_NUMBER = OH.ORDER\_NUMBER

 LEFT JOIN SALES\_STAFF SS ON OH.SALES\_STAFF\_CODE = SS.SALES\_STAFF\_CODE

(afbeelding toont slechts een gedeelte)

Great Outdoor heeft in totaal 706 keer een product terug ontvangen in diverse hoeveelheden. Elke order heeft een verkoper die verantwoordelijk voor het factuur van de aankoop en een klant die het product in bepaalde hoeveelheid heeft teruggestuurd. Daarbij is er ook bijgehouden op welke datum het retour is gemaakt.

A&C als bedrijf doet zelf niet aan het retouren van producten en heeft dus geen bruikbare gegevens voor deze vraag. A&C zal dus ook niet voorkomen in de datawarehouse om deze rede.

1. **Hoeveel facturen worden er gemaakt per maand, per werknemer, per klant?**



SELECT OH.ORDER\_NUMBER,

CONVERT(DATE, OH.ORDER\_DATE) AS ORDER\_DATE,

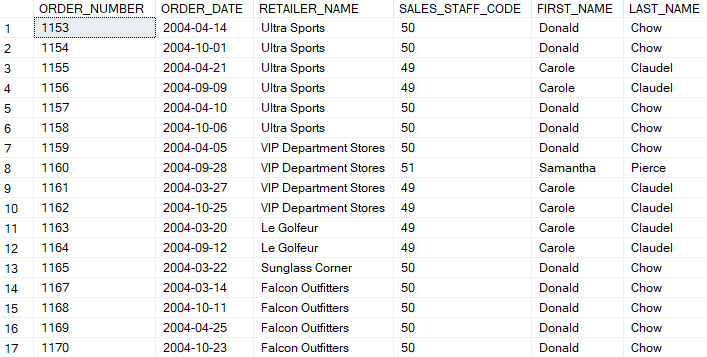
OH.RETAILER\_NAME,

SS.FIRST\_NAME,

SS.LAST\_NAME

FROM ORDER\_HEADER OH LEFT JOIN SALES\_STAFF SS ON OH.SALES\_STAFF\_CODE = SS.SALES\_STAFF\_CODE

ORDER BY OH.ORDER\_NUMBER



(afbeelding toont slechts een gedeelte)

De gegevens voor deze vraag zijn redelijk simpel om op te vragen. Great Outdoor heeft in totaal 5.360 facturen gemaakt voor meerdere klanten door meerdere werknemers. De aankopen die in deze facturen zitten maken niet uit voor deze vraag.



SELECT so.id,

CAST(so.order\_date AS DATE) AS order\_date,

c.company\_name,

e.emp\_id,

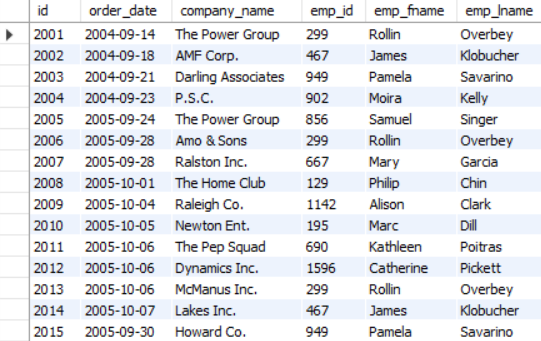
e.emp\_fname,

e.emp\_lname

FROM sales\_order so

LEFT JOIN customer c ON so.cust\_id = c.id

LEFT JOIN employee e ON so.sales\_rep = e.emp\_id



(afbeelding toont slechts een gedeelte)

A&C heeft in totaal 650 facturen en zijn net zo gemakkelijk op te halen als in Great Outdoor met relatief weinig join-statements. De gegevens zijn zeer vergelijkbaar met die van Great Outdoor voor deze management vraag.

1. **Wat is het meest bestelde product per maand, per werknemer, per korting van de klant?**



SELECT OD.ORDER\_DETAIL\_CODE,

OH.RETAILER\_NAME,

OD.ORDER\_NUMBER,

CONVERT(DATE, OH.ORDER\_DATE) AS ORDER\_DATE,

SS.SALES\_STAFF\_CODE,

SS.FIRST\_NAME,

SS.LAST\_NAME,

P.PRODUCT\_NUMBER,

P.PRODUCT\_NAME,

PT.PRODUCT\_TYPE\_EN,

OD.QUANTITY

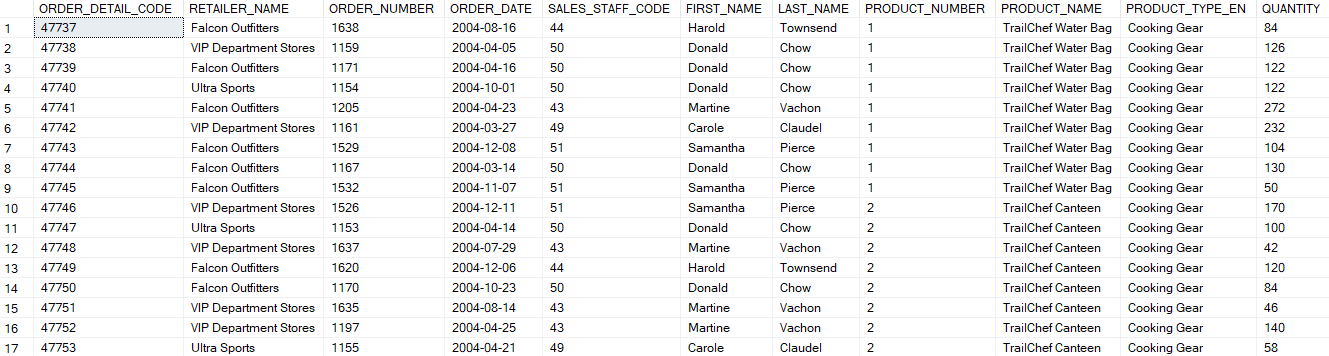
FROM ORDER\_DETAILS OD

LEFT JOIN PRODUCT P ON OD.PRODUCT\_NUMBER = P.PRODUCT\_NUMBER

LEFT JOIN ORDER\_HEADER OH ON OD.ORDER\_NUMBER = OH.ORDER\_NUMBER

LEFT JOIN PRODUCT\_TYPE PT ON P.PRODUCT\_TYPE\_CODE = PT.PRODUCT\_TYPE\_CODE

LEFT JOIN SALES\_STAFF SS ON OH.SALES\_STAFF\_CODE = SS.SALES\_STAFF\_CODE

ORDER BY OD.ORDER\_DETAIL\_CODE

(afbeelding toont slechts een gedeelte)

Voor deze vraag zijn vergelijkbare gegevens nodig als bij vraag 1, alleen draait het nu niet om omzet maar om het maximale aantal producten verkocht. Een belangrijk punt is dat Great Outdoor geen korting geeft aan haar klanten, in tegenstelling tot A&C. In het datawarehouse zou hier dus NULL of 0 terecht komen als waarde.



SELECT soi.id AS order\_id,

soi.line\_id,

c.company\_name,

c.discount,

CAST(so.order\_date AS DATE) AS order\_date,

e.emp\_id,

e.emp\_fname,

e.emp\_lname,

p.id AS product\_id,

p.name,

p.category,

soi.quantity

FROM sales\_order\_item soi

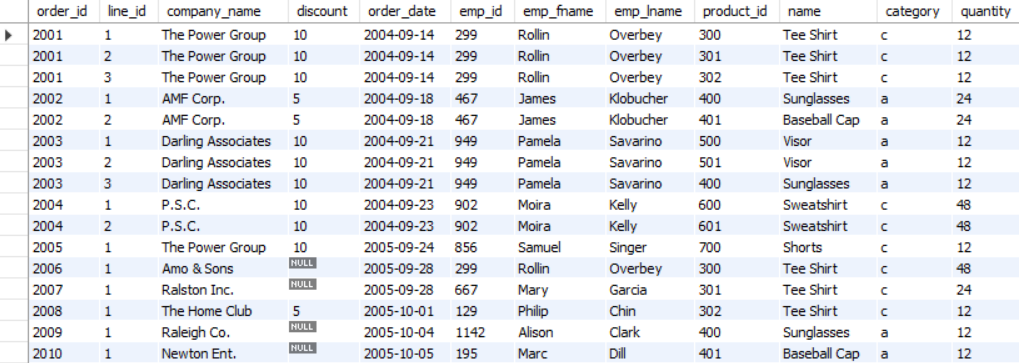
LEFT JOIN sales\_order so ON soi.id = so.id

LEFT JOIN customer c ON so.cust\_id = c.id

LEFT JOIN product p ON soi.prod\_id = p.id

LEFT JOIN employee e ON so.sales\_rep = e.emp\_id

ORDER BY order\_id, line\_id ASC



(afbeelding toont slechts een gedeelte)

A&C houdt per klant bij hoeveel korting er wordt gegeven, in tegenstelling tot Great Outdoor. Nogmaals gebruikt het voor een groot gedeelte dezelfde gegevens als bij vraag 1, alleen gaat het puur om de hoeveelheid verkocht.

# Datawarehouse ontwerp

In dit hoofdstuk neem je de dimensiematrix, feitmodel, eventuele dimensie-hiërarchieën en het uiteindelijke starschema op. Neem de constraints ook mee in het ontwerp. Schenk ook aandacht aan SCD’s.

Neem alle ontwerpen als een embedded object op. Welke keuzes hebben jullie moeten maken en waarom? Breng dit ook in relatie met de managementvragen. Voor elk onderdeel moet een toelichting zijn opgenomen en moet daarnaast ook een duidelijke opbouw in verschillende onderdelen te herkennen zijn. Wees dus consequent in de uitwerkingen.

## 4.1 Dimensie matrix

Neem hieronder de dimensiematrix op. Dit houdt in: Alle feiten en de nodige dimensies per managementvragen/feit.

Om de dimensies en meetwaardes te weten van de tabel hebben wij eerst gekeken naar de gegevens die wij tot onze beschikking hadden, en de gegevens die wij nodig hebben. De 4 management vragen die wij hiervoor hebben gebruikt zijn:

* Vraag 1: Hoeveel omzet wordt er behaald, en hoeveel producten worden er verkocht per maand, per jaar, per producttype, per product, per land van de klant en per klant?
* Vraag 2: Hoeveel retourneringen zijn er per product, per product type, per reden, per klant, welke verkocht zijn door een verkoper op een bepaald moment?
* Vraag 3: Hoeveel facturen worden er gemaakt per maand, per werknemer, per klant?
* Vraag 4: Wat is het meest bestelde product per maand, per werknemer, per korting van de klant?

Vanuit deze vragen hebben wij 2 tabellen uitgewerkt en hieronder neergezet.

In de 1e tabel is er een overzicht gemaakt van de feiten, dimensies, gegevens en de bronnen. Is een lijst aan gegevens bij elke dimensie en bij elke dimensie staat 1 a 2 bronnen waar de gegevens vandaan gehaald worden.

In de 2e tabel is er een klein overzicht van welke feiten en dimensies bij elkaar horen. Alle gebruikte dimensies staan hierin en er wordt hier per feit aangevinkt welke dimensies bij dit feit horen.

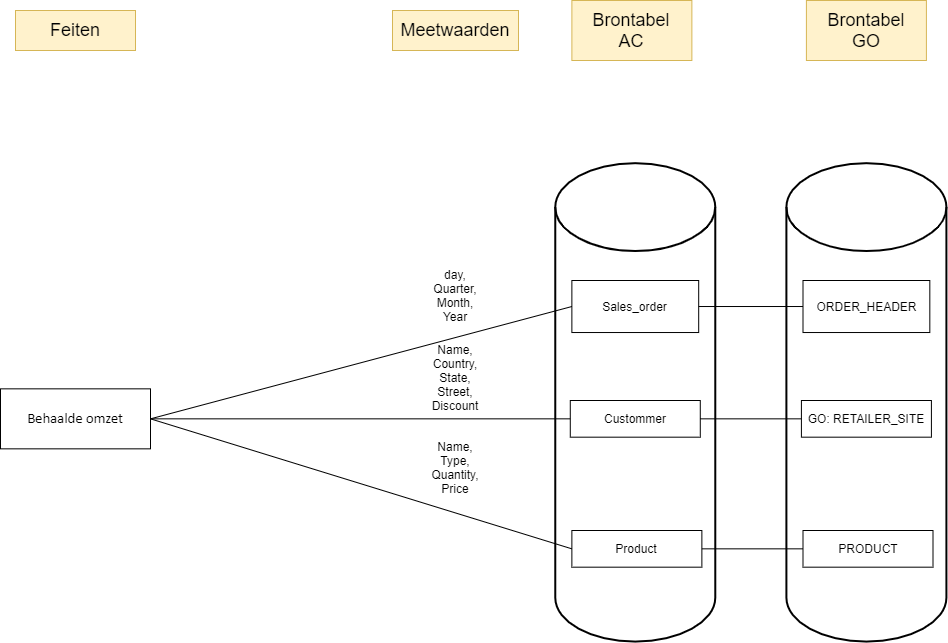
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Feit | Dimensie | Gegevens | Bron | |
| Behaalde omzet | Time | day  quarter  month  year | GO: ORDER\_HEADER  AC: sales\_order |
| Customer | name  country  state  address  discount | GO: RETAILER\_SITE |
| AC: customer |
| Product | name  type  quantity  price | GO: PRODUCT |
| AC: product |
| Geretourneerd product | Time | day  quarter  month  year | GO: ORDER\_HEADER  AC: sales\_order |
| Customer | name  country  state  address  discount | GO: RETAILER\_SITE |
| AC: customer |
| Product | name  type  quantity  price | GO: PRODUCT |
| AC: product |
| Gemaakte facturen | Time | day  quarter  month  year | GO: PRODUCT  AC: sales\_order |
| Customer | name  country  state  address  discount | GO: RETAILER\_SITE |
| AC: customer |
| Employee | name  department  sex | GO: SALES STAFF |
| AC: Employee |
| Meest verkochte product | Time | day  quarter  month  year | GO: ORDER\_HEADER  AC: sales\_order |
| Customer | name  country  state  address  discount | GO: RETAILER\_SITE |
| AC: customer |
| Product | name  type  quantity  price | GO: PRODUCT |
| AC: product |
| Employee | name  department  sex | GO: SALES STAFF |
| AC: Employee |
| Sales | quantity | GO: ORDER\_HEADER |
| AC: Sales\_order |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Meetwaarde/Dimensie* | **Time** | **Customer** | **Product** | **Employee** | **Sales** |
| **Aantal producten**  **berekenen** | x | x | x |  |  |
| **Retour van producten** | x | x | x |  |  |
| **Facturen van producten** | x | x |  | x |  |
| **Meeste profijt per product** | x | x | x | x | x |

## 4.2 Feitmodel

Neem hieronder het feitenmodel op. Ook hier geldt dat alle gemaakte keuzes onderbouwd dienen te worden. Doe dit voor elke feit.

1. **Hoeveel omzet wordt er behaald, en hoeveel producten worden er verkocht per maand, per jaar, per producttype, per product, per land van de klant en per klant?**



|  |  |  |
| --- | --- | --- |
| **Feit** | **Meetwaarde** | **Zinvolle berekening** |
| Customer buys product | * Quantity * Price * Discount | Sum (quantity)  Sum (quantity \* price)  Sum (quantity \* discount) |
| Company calculates their turnover | * Price * Quantity | Sum (quantity \* price)  Sum (quantity) |

**Uitleg:**

Om de omzet te kunnen berekenen hebben wij 2 feiten nodig:

* Een klant/klanten die een product/producten koopt
* Het bedrijf dat berekend hoeveel klant/klanten er gewinkeld hebben bij het bedrijf

Bij deze 2 feiten zijn er meerdere dingen die gemeten moeten worden, de meetwaarden die we nodig hebben per feit zijn:

Een klant/klanten die een product/producten verkoopt

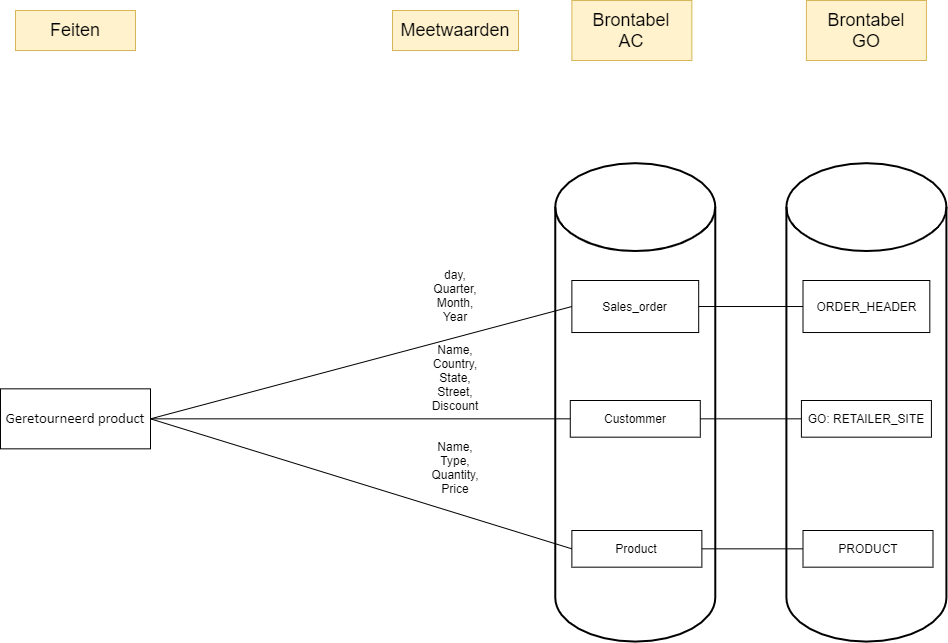
* Hoeveelheid
* Prijs
* Korting

Het bedrijf dat berekend hoeveel klant/klanten er gewinkeld hebben bij het bedrijf

* Prijs
* Hoeveelheid

Op basis hiervan hebben wij het feitmodel gemaakt van vraag 1

1. **Hoeveel retourneringen zijn er per product, per product type, per reden, per klant, welke verkocht zijn door een verkoper op een bepaald moment?**



|  |  |  |
| --- | --- | --- |
| **Feit** | **Meetwaarde** | **Zinvolle berekening** |
| Customer returns the product | * Quantity | Sum (quantity) |

**Uitleg**

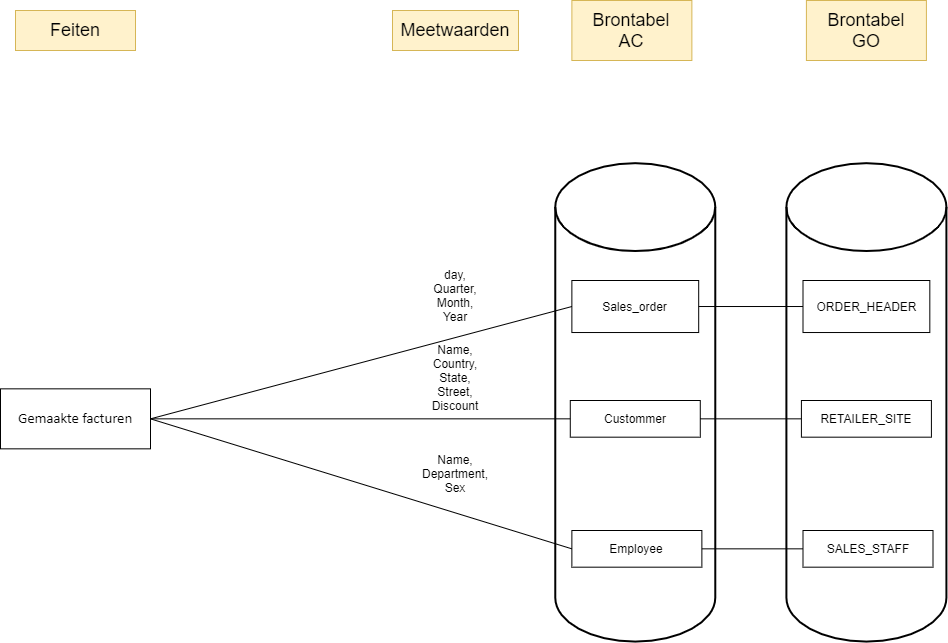
Om de hoeveelheid aan retourneringen bij te houden is er 1 belangrijk feit wat bijgehouden moet worden:

* Hoeveel bezoekers retourneren hoeveel producten?

Bij dit feit is er 1 meetwaarde wat gemeten moet worden

* Hoeveelheid

1. **Hoeveel facturen worden er gemaakt per maand, per werknemer, per klant?**



|  |  |  |
| --- | --- | --- |
| **Feit** | **Meetwaarde** | **Zinvolle berekening** |
| Company calculates amount of invoices | * Quantity | * Sum (quantity) |

**Uitleg**

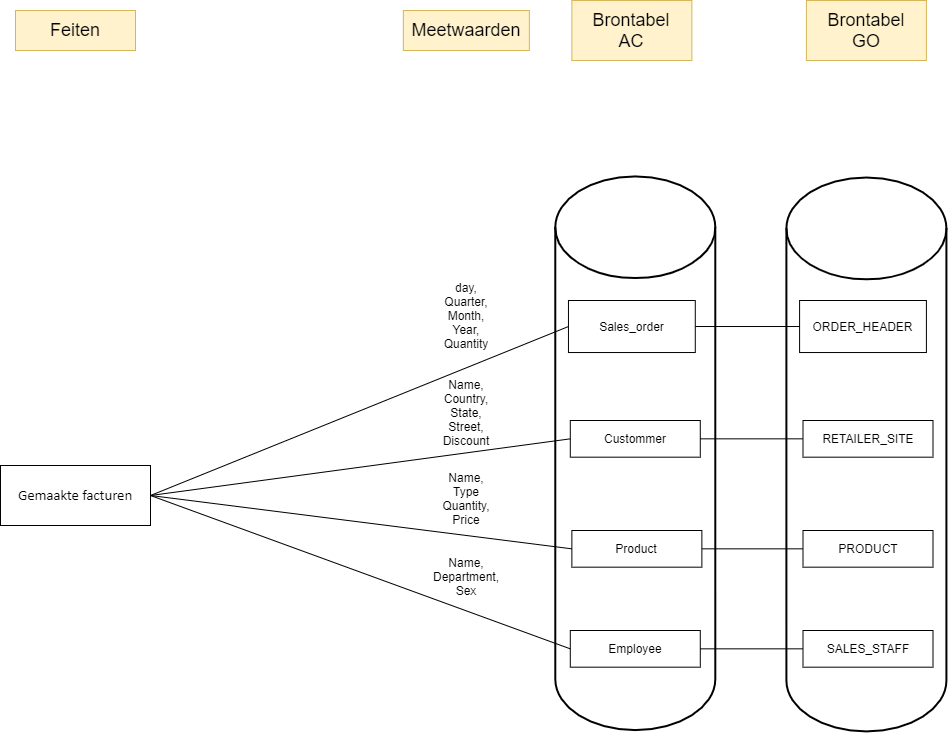
Om de hoeveelheid facturen bij te houden is er 1 feit belangrijk:

* Hoeveel facturen heeft het bedrijf?

Bij dit feit is er 1 meetwaardedie gemeten moet worden:

* Hoeveelheid

1. **Wat is het meest bestelde product per maand, per werknemer, per korting van de klant?**



|  |  |  |
| --- | --- | --- |
| **Feit** | **Meetwaarde** | **Zinvolle berekening** |
| Company calculates which product is sold the most | * Quantity | Max(quantity) |

**Uitleg**

Om het meest bestelde product bij te houden is er 1 feit belangrijk

* Hoeveel producten zijn er besteld van alle producten?

Bij dit feit is er 1 meetwaarde die gemeten moet worden:

* Hoeveelheid

## 4.3 Dimensie & dimensie hiërarchieën

Neem hieronder de opbouw van de dimensies (dus de nodige attributen) op met waar mogelijk ook de hiërarchieën met onderbouwingen. Denk ook aan surrogaatsleutels.

**Hiërarchieën (Van groot naar klein)**

|  |
| --- |
| **Customer** |
| Country  Name |
| State  Name |
| Address  Name |

|  |
| --- |
| **Time** |
| Year  nr |
| Quarter  nr |
| Month  nr |
| Day  nr |

De hiërarchieën gelden voor elke vraag en zijn op volgorde van groot naar klein. Hieronder wordt de logica achter de hiërarchieën uitgelegd:

Customer: een adres bevind zich in een staat, en een staat bevind zich in een land. Alle 3 de attributen zijn herkenbaar in de database door middel van hun naam

Time: een dag zit in een maand, 3 maanden zitten in een kwartaal en 4 kwartalen zitten in een jaar. Alle 4 de attributen zijn herkenbaar in de database door middel van het nummer dat gegeven is aan deze attributen.

1. **Hoeveel omzet wordt er behaald, en hoeveel producten worden er verkocht per maand, per jaar, per producttype, per product, per land van de klant en per klant?**

|  |  |
| --- | --- |
| **Feit** | **Mogelijke dimensiegegevens** |
| Customer buys product | * Time: day of week, month, year * Customer: name, country, state, address, discount * Product: name, type, quantity, price |
| Company calculates their turnover | * Time: day of week, month, year * Customer: name, country, state, address, discount * Product: name, type, quantity, price |

**Uitleg**

Om de omzet te kunnen berekenen hebben wij 2 feiten nodig:

* Een klant/klanten die een product/producten koopt
* Het bedrijf dat berekend hoeveel klant/klanten er gewinkeld hebben bij het bedrijf

Per feit zijn deze dimensies aan gegevens nodig om de vraag uit te kunnen werken:

* De tijd (want per jaar)
* De klant (want per klant en per land)
* Het product (want per product)

1. **Hoeveel retourneringen zijn er per product, per product type, per reden, per klant, welke verkocht zijn door een verkoper op een bepaald moment?**

|  |  |
| --- | --- |
| **Feit** | **Mogelijke dimensiegegevens** |
| Customer sends the product back | * Time: day of week, month, year * Product: name, type, quantity, price * Customer: name, country, state, address, discount |

**Uitleg**

Om de hoeveelheid aan retourneringen bij te houden is er 1 belangrijk feit wat bijgehouden moet worden:

* Hoeveel bezoekers retourneren hoeveel producten?

Voor dit feit zijn deze dimensies aan gegevens nodig om de vraag uit te kunnen werken:

* De tijd (want voor een bepaald moment)
* Het product (want per product, per producttype)
* De klant (want per klant, per reden)

1. **Hoeveel facturen worden er gemaakt per maand, per werknemer, per klant?**

|  |  |
| --- | --- |
| **Feit** | **Mogelijke dimensiegegevens** |
| Company calculates the amount of invoices | * Time: day of week, month, year * Employee: name, department, sex * Customer: name, country, state, address, discount |

**Uitleg**

Om de hoeveelheid facturen bij te houden is er 1 feit belangrijk:

* Hoeveel facturen heeft het bedrijf?

Voor dit feit zijn deze dimensies aan gegevens nodig om de vraag uit te kunnen werken

* De tijd (want per maand)
* De werknemer (want per werknemer)
* De klant (want per klant)

1. **Wat is het meest bestelde product per maand, per werknemer, per korting van de klant?**

|  |  |
| --- | --- |
| **Feit** | **Mogelijke dimensiegegevens** |
| Company calculates which product is sold the most | * Time: day of week, month, year * Product: name, type, quantity, price * Sales: quantity * Employee: name, department, sex * Customer: name, country, state, address, discount |

**Uitleg**

Om het meest bestelde product bij te houden is er 1 feit belangrijk

* Hoeveel producten zijn er besteld van alle producten?

Voor dit feit zijn deze dimensies aan gegevens nodig om de vraag uit te kunnen werken

* De tijd (want per maand)
* Het product (vanwege het meest bestelde product dat bijgehouden moet worden)
* De verkoop (wamt hoeveelheid verkocht)
* De werknemer (want per werknemer)
* De klant (want per korting van de klant)

## 4.4 Conformer dimension

Doorloop de dimensiematrix en kijk naar de dimensies uit paragraaf 4.3 en herleid de conformed dimensies. Vermeld hieronder de conformed dimensions met de attributen.

In totaal maken wij gebruik van 4 dimensies waarvan ze allemaal conformed zijn. Iedere dimensie wordt minstens 2x gebruikt door een feit.

* Time
* Customer
* Product
* Employee

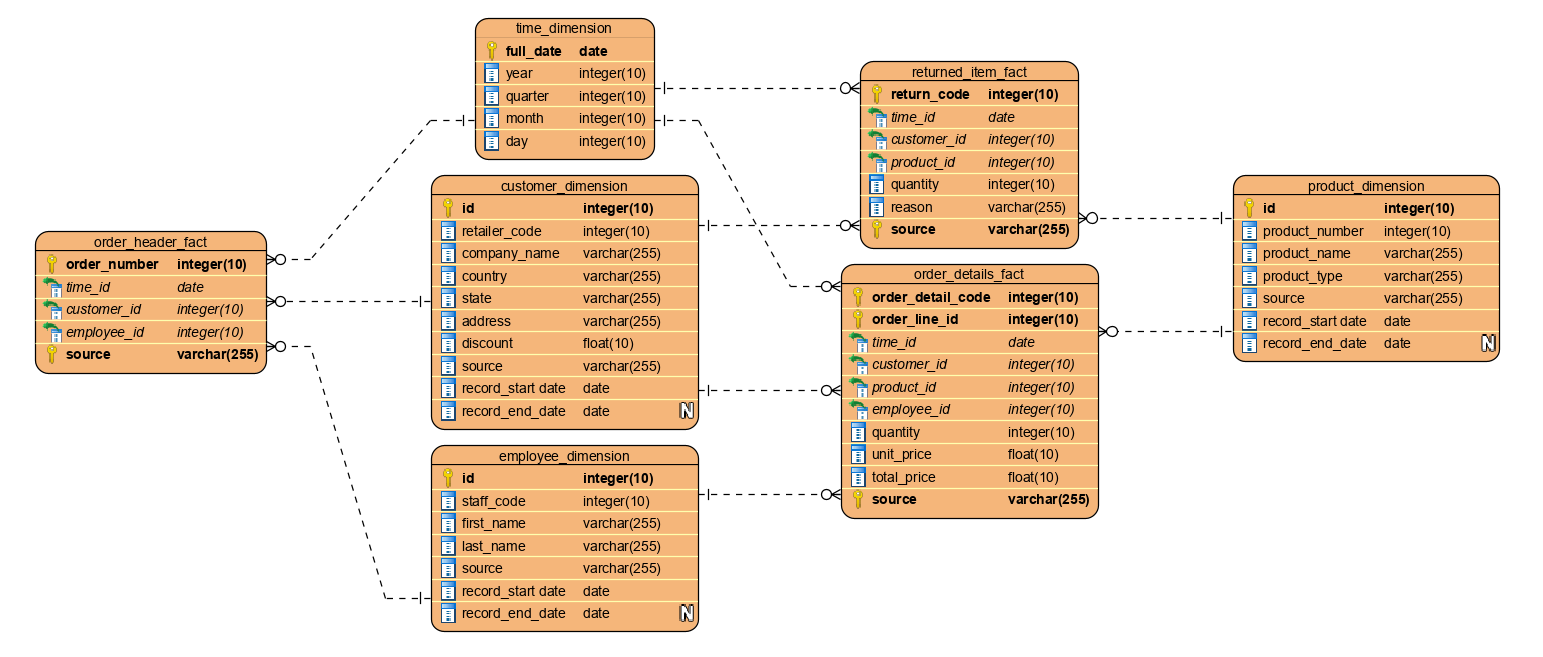
De dimensies bevatten de volgende mogelijke bruikbare attributen

* Time: day month, quarter, year
* Customer: discount, country, state, address
* Product: name, price, introduction date, type
* Employee: name, sex, department

## 4.5 Sterschema

Neem vervolgens het sterschema op. Het schema moet in lijn zijn met de ontwerpen van de vorige paragrafen. Gebruik de CROW’s foot notatie inclusief alle sleutels (zowel PK’s als FK’s). Geef ook een korte toelichting op het sterschema.





**Toelichting sterschema**

Het sterschema bestaat uit 4 dimensietabellen en 3 feittabellen. De dimensietabellen hebben allemaal een surrogaatsleutel genaamd ‘id’ die de primary key is. Ook bevatten ze de oude primary key uit hun OLTP-omgeving, oftewel de business key. Daarnaast bevatten de dimensietabellen hun meetwaarden zoals prijs en adres. De feittabellen bevatten de foreign keys naar de dimensietabellen en eventuele meetwaarden die bij de feit hoort. I.p.v. dat alle foreign keys de primary key voor de fiettabellen vormen is ervoor gekozen de primary key uit de OLTP-omgeving te gebruiken samen met het source attribuut. Met het source attribuut kan nog bijgehouden worden uit welke OLTP-omgeving het record kwam. Deze 2 attributen zullen samen altijd uniek zijn.

Dimensietabellen bevatten ook 2 opvallende records namelijk record\_start\_date en record\_end\_date. Deze 2 attributen bestaan vanwege de toepassing van SCD type 2. Mocht er een record attribuut veranderen in de OLTP-omgeving, dan wordt er een nieuw record aangemaakt in de OLAP-omgeving en aan het oude record wordt er een einddatum gegeven. Zo kan er historie van gegevens worden bijgehouden.

# Datawarehouse

## 5.1 Create scripts Data warehouse



-- create all dimension tables

CREATE TABLE time\_dimension (

full\_date date PRIMARY KEY,

year int NOT NULL,

quarter int NOT NULL,

month int NOT NULL,

day int NOT NULL

);

CREATE TABLE customer\_dimension (

id int IDENTITY(1, 1) PRIMARY KEY,

retailer\_code int NOT NULL,

company\_name varchar(255) NOT NULL,

country varchar(255) NOT NULL,

state varchar(255) NOT NULL,

address varchar(255) NOT NULL,

discount float NOT NULL,

record\_start\_date date NOT NULL,

record\_end\_date date NULL,

source varchar(255) NOT NULL,

);

CREATE TABLE employee\_dimension (

id int IDENTITY(1, 1) PRIMARY KEY,

staff\_code int NOT NULL,

first\_name varchar(255) NOT NULL,

last\_name varchar(255) NOT NULL,

record\_start\_date date NOT NULL,

record\_end\_date date NULL,

source varchar(255) NOT NULL,

);

CREATE TABLE product\_dimension (

id int IDENTITY(1, 1) PRIMARY KEY,

product\_number int NOT NULL,

product\_name varchar(255) NOT NULL,

product\_type varchar(255) NOT NULL,

record\_start\_date date NOT NULL,

record\_end\_date date NULL,

source varchar(255) NOT NULL,

);

-- create all fact tables

CREATE TABLE order\_header\_fact (

order\_number int,

time\_id date NOT NULL,

customer\_id int NOT NULL,

employee\_id int NOT NULL,

source varchar(255) NOT NULL,

PRIMARY KEY (order\_number, source),

FOREIGN KEY (time\_id) REFERENCES time\_dimension(full\_date),

FOREIGN KEY (customer\_id) REFERENCES customer\_dimension(id),

FOREIGN KEY (employee\_id) REFERENCES employee\_dimension(id)

);

CREATE TABLE order\_details\_fact (

order\_detail\_code int,

order\_line\_id int NOT NULL,

time\_id date NOT NULL,

customer\_id int NOT NULL,

product\_id int NOT NULL,

employee\_id int NOT NULL,

quantity int NOT NULL,

unit\_price float NOT NULL,

total\_price float NOT NULL,

source varchar(255) NOT NULL,

PRIMARY KEY (order\_detail\_code, order\_line\_id, source),

FOREIGN KEY (time\_id) REFERENCES time\_dimension(full\_date),

FOREIGN KEY (customer\_id) REFERENCES customer\_dimension(id),

FOREIGN KEY (product\_id) REFERENCES product\_dimension(id),

FOREIGN KEY (employee\_id) REFERENCES employee\_dimension(id)

);

CREATE TABLE returned\_item\_fact (

return\_code int,

time\_id date NOT NULL,

customer\_id int NOT NULL,

product\_id int NOT NULL,

quantity int NOT NULL,

reason varchar(255) NOT NULL,

source varchar(255) NOT NULL,

PRIMARY KEY (return\_code, source),

FOREIGN KEY (time\_id) REFERENCES time\_dimension(full\_date),

FOREIGN KEY (customer\_id) REFERENCES customer\_dimension(id),

FOREIGN KEY (product\_id) REFERENCES product\_dimension(id)

);



-- create procedure to fill time\_dimension

CREATE PROCEDURE fill\_time\_dimension (@startDate date, @endDate date)

AS

BEGIN

DECLARE @loopDate date

SET @loopDate = @startDate

WHILE @loopDate <= @endDate

BEGIN

INSERT INTO time\_dimension (

full\_date,

year,

quarter,

month,

day

)

VALUES (

@loopDate,

YEAR(@loopDate),

DATEPART(quarter, @loopDate),

MONTH(@loopDate),

DAY(@loopDate)

)

SET @loopDate = DATEADD(day, 1, @loopDate)

END

END



-- refresh table to see results

EXECUTE fill\_time\_dimension '2004-01-01', '2007-12-31'

## 5.2 Create scripts Staging



-- Staging dimension tables

create table customer\_GO (

retailer\_code int PRIMARY KEY,

company\_name varchar(255) NOT NULL,

country varchar(255) NOT NULL,

region varchar(255) NOT NULL,

address varchar(255) NOT NULL,

discount float NOT NULL,

record\_start\_date date NOT NULL,

record\_end\_date date NULL,

source varchar(255) NOT NULL

);

create table customer\_AC (

id int PRIMARY KEY,

company\_name varchar(255) NOT NULL,

country varchar(255) NOT NULL,

state varchar(255) NOT NULL,

address varchar(255) NOT NULL,

discount float NOT NULL,

record\_start\_date date NOT NULL,

record\_end\_date date NULL,

source varchar(255) NOT NULL

);

create table employee\_GO (

sales\_staff\_code int PRIMARY KEY,

first\_name varchar(255) NOT NULL,

last\_name varchar(255) NOT NULL,

record\_start\_date date NOT NULL,

record\_end\_date date NULL,

source varchar(255) NOT NULL

);

create table employee\_AC (

emp\_id int PRIMARY KEY,

emp\_fname varchar(255) NOT NULL,

emp\_lname varchar(255) NOT NULL,

record\_start\_date date NOT NULL,

record\_end\_date date NULL,

source varchar(255) NOT NULL

);

CREATE TABLE product\_GO (

product\_number int PRIMARY KEY,

product\_name varchar(255) NOT NULL,

product\_type varchar(255) NOT NULL,

record\_start\_date date NOT NULL,

record\_end\_date date NULL,

source varchar(255) NOT NULL

);

CREATE TABLE product\_AC (

id int PRIMARY KEY,

product\_name varchar(255) NOT NULL,

category varchar(255) NOT NULL,

record\_start\_date date NOT NULL,

record\_end\_date date NULL,

source varchar(255) NOT NULL

);

-- Staging fact tables

CREATE TABLE order\_header\_GO (

order\_number int PRIMARY KEY,

time\_id date NOT NULL,

customer\_id int NOT NULL,

employee\_id int NOT NULL,

source varchar(255) NOT NULL

);

CREATE TABLE order\_header\_AC (

id int PRIMARY KEY,

time\_id date NOT NULL,

customer\_id int NOT NULL,

employee\_id int NOT NULL,

source varchar(255) NOT NULL

);

CREATE TABLE order\_details\_GO (

order\_detail\_code int PRIMARY KEY,

time\_id date NOT NULL,

customer\_id int NOT NULL,

product\_id int NOT NULL,

employee\_id int NOT NULL,

quantity int NOT NULL,

unit\_price float NOT NULL,

total\_price float NOT NULL,

source varchar(255) NOT NULL

);

CREATE TABLE order\_details\_AC (

id int NOT NULL,

line\_id int NOT NULL,

time\_id date NOT NULL,

customer\_id int NOT NULL,

product\_id int NOT NULL,

employee\_id int NOT NULL,

quantity int NOT NULL,

unit\_price float NOT NULL,

total\_price float NOT NULL,

source varchar(255) NOT NULL,

PRIMARY KEY(id, line\_id)

);

CREATE TABLE returned\_item\_GO (

return\_code int PRIMARY KEY,

time\_id date NOT NULL,

customer\_id int NOT NULL,

product\_id int NOT NULL,

quantity int NOT NULL,

reason varchar(255) NOT NULL,

source varchar(255) NOT NULL

);

## 5.3 ETL mapping



|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Target Table** | **Target Column** | **Target Data-type** | **Source Table** | **Source Column** | **Source Data-type** | **Expression** | **Default Value if Null** | **Data Issues/Quality/Comments** |
| **Name of target table** | **Column in target table** | **Data type for this target column** | **Name of the Source table** | **Column in source table where you would get this data from** | **Data type for this source column** | **Decodes, aggregates, conversions, if statements, lookup functions……** | **Value to use in target field when source field is null** | **Used to document Not null, value if looked up, upper case, pk, fk, etc…comments, issues** |
| employee\_dimension | id | int |  |  |  | surrogate key, IDENTITY(1, 1) |  | primary key |
| staff\_code | int | GO: SALES\_STAFF | SALES\_STAFF\_CODE | int |  |  | business key |
| AC: employee | emp\_id | int |  |  |
| first\_name | varchar(255) | GO: SALES\_STAFF | FIRST\_NAME | nvarchar(25) | conversion to: varchar(255) |  |  |
| AC: employee | emp\_fname | varchar(255) |  |  |  |
| last\_name | varchar(255) | GO: SALES\_STAFF | LAST\_NAME | nvarchar(30) | conversion to: varchar(255) |  |  |
| AC: employee | emp\_lname | varchar(255) |  |  |  |
| source | varchar(255) |  |  |  | derived value as: "GO" or "AC" |  |  |
| record\_start\_date | date | GO: SALES\_STAFF | DATE\_HIRED | datetime | conversion to: date |  |  |
| AC: employee | start\_date | datetime | conversion to: date |  |  |
| record\_end\_date | date | GO: |  |  |  | NULL |  |
| AC: employee | termination\_date | datetime | conversion to: date |  |  |
| customer\_dimension | id | int |  |  |  | surrogate key, IDENTITY(1, 1) |  | primary key |
| retailer\_code | int | GO: RETAILER | RETAILER\_CODE | int |  |  | business key |
| AC: customer | id | int |  |  |
| company\_name | varchar(255) | GO: RETAILER | COMPANY\_NAME | nvarchar(50) | conversion to: varchar(255) |  |  |
| AC: customer | company\_name | varchar(255) |  |  |  |
| country | varchar(255) | GO: COUNTRY | COUNTRY | nvarchar(50) | conversion to: varchar(255) | "N/A" |  |
| AC: |  |  | derived value as: "United States" |  |  |
| state | varchar(255) | GO: RETAILER\_SITE | REGION | nvarchar(50) | conversion to: varchar(255) | "N/A" |  |
| AC: customer | state | varchar(255) |  |  |  |
| address | varchar(255) | GO: RETAILER\_SITE | ADDRESS1 | nvarchar(50) | conversion to: varchar(255) | "N/A" |  |
| AC: customer | address | varchar(255) |  |  |  |
| discount | float | GO: |  |  | derived value as: 0 |  |  |
| AC: customer | discount | int |  |  |  |
| source | varchar(255) |  |  |  | derived value as: "GO" or "AC" |  |  |
| record\_start\_date | date | GO: |  |  | current date, GETDATE() |  |  |
| AC: |  |  | current date, GETDATE() |  |  |
| record\_end\_date | date | GO: |  |  |  | NULL |  |
| AC: |  |  |  | NULL |  |
| product\_dimension | id | int |  |  |  | surrogate key, IDENTITY(1, 1) |  | primary key |
| product\_number | int | GO: PRODUCT | PRODUCT NUMBER | int |  |  | business key |
| AC: product | id | int |  |  |
| product\_name | varchar(255) | GO: PRODUCT | PRODUCT\_NAME | nvarchar(255) | conversion to: varchar(255) |  |  |
| AC: product | name | varchar(255) |  |  |  |
| product\_type | varchar(255) | GO: PRODUCT\_TYPE | PRODUCT\_TYPE\_EN | nvarchar(50) | conversion to: varchar(255) |  |  |
| AC: product | category | varchar(1) |  |  |  |
| source | varchar(255) |  |  |  | derived value as: "GO" or "AC" |  |  |
| record\_start\_date | date | GO: PRODUCT | INTRODUCTION\_DATE | datetime | conversion to: date |  |  |
| AC: |  |  | current date, GETDATE() |  |  |
| record\_end\_date | date | GO: |  |  | current date, GETDATE() | NULL |  |
| AC: |  |  |  | NULL |  |
| time\_dimension | full\_date | date |  |  |  | value created manually |  | primary key |
| year | int |  |  |  | value created manually |  |  |
| quarter | int |  |  |  | value created manually |  |  |
| month | int |  |  |  | value created manually |  |  |
| day | int |  |  |  | value created manually |  |  |
| order\_header\_fact | order\_number | int | GO: ORDER\_HEADER | ORDER\_NUMBER | int |  |  | primary key part |
| AC: sales\_order | id | int |  |  |
| time\_id | date | GO: ORDER\_HEADER | ORDER\_DATE | datetime | conversion to: date |  | foreign key |
| AC: sales\_order | order\_date | datetime | conversion to: date |  |
| customer\_id | int | GO: RETAILER | RETAILER\_CODE | int |  |  | foreign key |
| AC: sales\_order | cust\_id | int |  |  |
| employee\_id | int | GO: ORDER\_HEADER | SALES\_STAFF\_CODE | int |  |  | foreign key |
| AC: sales\_order | sales\_rep | int |  |  |
| source | varchar(255) |  |  |  | derived value as: "GO" or "AC" |  | primary key part |
| order\_header\_details\_fact | order\_detail\_code | int | GO: ORDER\_DETAILS | order\_detail\_code | int |  |  | primary key part |
| AC: sales\_order\_item | id | int |  |  |
| order\_line\_id | int | GO: |  |  | derived value as: 1 |  | primary key part |
| AC: sales\_order\_item | line\_id | int |  |  |
| time\_id | date | GO: ORDER\_HEADER | ORDER\_DATE | datetime | conversion to: date |  | foreign key |
| AC: sales\_order | order\_date | datetime | conversion to: date |  |
| customer\_id | int | GO: RETAILER | RETAILER\_CODE | int |  |  | foreign key |
| AC: sales\_order | cust\_id | int |  |  |
| product\_id | int | GO: ORDER\_DETAILS | PRODUCT\_NUMBER | int |  |  | foreign key |
| AC: sales\_order\_item | prod\_id | int |  |  |
| employee\_id | int | GO: ORDER\_HEADER | SALES\_STAFF\_CODE | int |  |  | foreign key |
| AC: sales\_order | sales\_rep | int |  |  |
| quantity | int | GO: ORDER\_DETAILS | QUANTITY | int |  |  |  |
| AC: sales\_order | quantity | int |  |  |  |
| unit\_price | float | GO: ORDER\_DETAILS | UNIT\_PRICE | float |  |  |  |
| AC: product | unit\_price | float |  | 0 |  |
| total\_price | float | GO: |  |  | derived value as: QUANTITY \* UNIT PRICE | 0 |  |
| AC: |  |  | derived value as: quantity \* unit\_price | 0 |  |
| source | varchar(255) |  |  |  | derived value as: "GO" or "AC" |  | primary key part |
| returned\_item\_fact | return\_code | int | GO: RETURNED\_ITEM | RETURN\_CODE | int |  |  | primary key part |
| does not exist in AC | | | | | |
| time\_id | date | GO: RETURNED\_ITEM | RETURN\_DATE | datetime | conversion to: date |  | foreign key |
| does not exist in AC | | | | | |
| customer\_id | int | GO: RETAILER | RETAILER\_CODE | int |  |  | foreign key |
| does not exist in AC | | | | | |
| product\_id | int | GO: ORDER\_DETAILS | PRODUCT\_NUMBER | int |  |  | foreign key |
| does not exist in AC | | | | | |
| quantity | int | GO: RETURNED\_ITEM | RETURN\_QUANTITY | int |  |  |  |
| does not exist in AC | | | | | |
| reason | varchar(255) | GO: RETURN\_REASON | RETURN\_DESCRIPTION\_EN | nvarchar(50) | conversion to: varchar(255) |  |  |
| does not exist in AC | | | | | |
| source | varchar(255) |  |  |  | derived value as: "GO" |  | primary key part |

## 5.4 ETL dataflows

Neem ook de realisatie van het ETL-ontwerp hier op en geef uitleg van de dataflow(s) van het ETL. Onderbouw alle gemaakte keuzen. Neem een visualisatie op van elke dataflow als embedded object en geef uitleg hiervan. Maak hierbij gebruik van stored procedures/functies. Neem de code hiervan ook op.

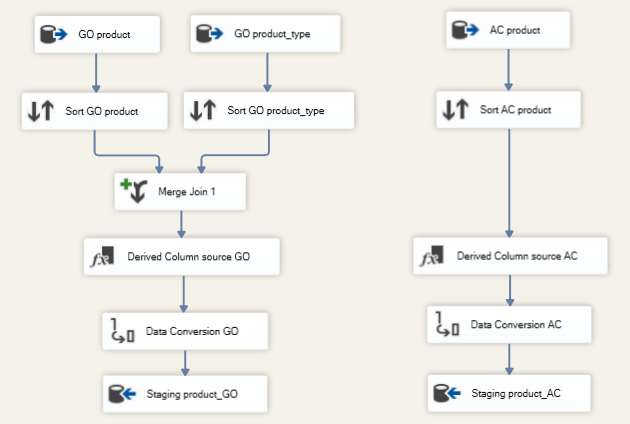
Zijn er gegevens welke om één of ander reden niet mee kunnen? Neem die dan hier ook op met onderbouwing waarom deze gegevens niet meegaan in het ETL-proces.

Voeg het SSIS-project (in ZIP formaat) als embedded object toe.

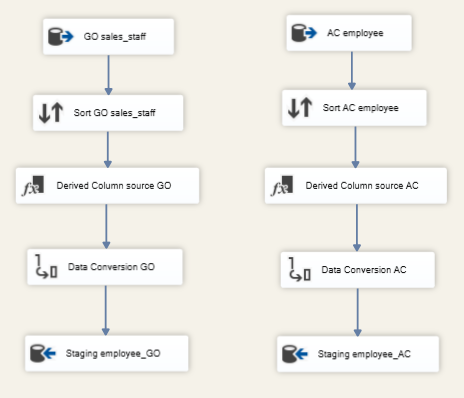




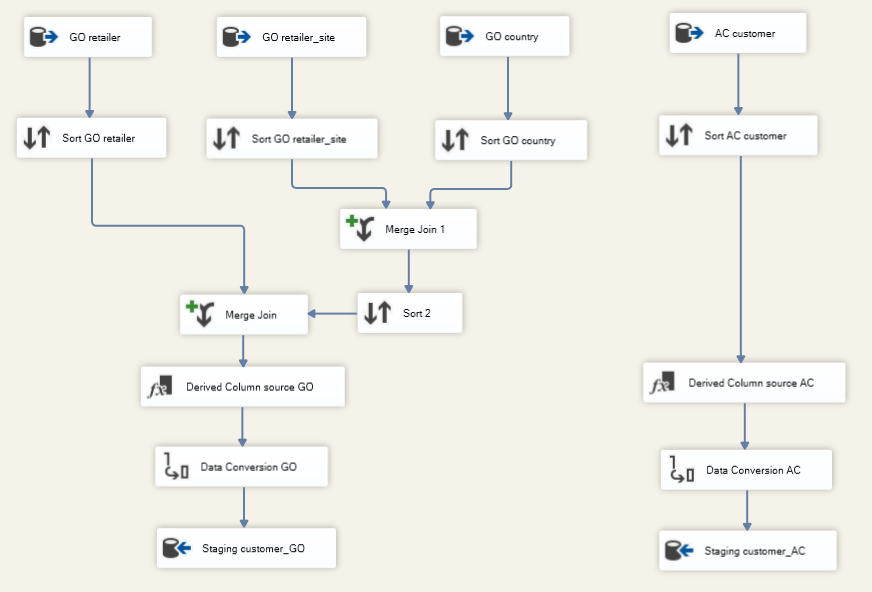
Package die alle dataflows van de source naar de staging bevat



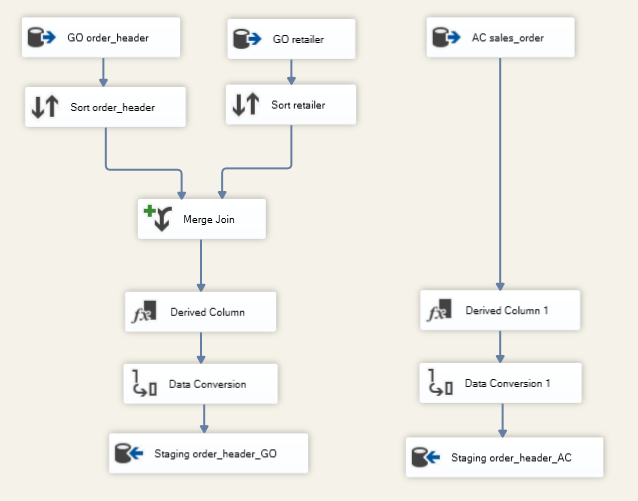
Dataflow van source product naar staging product



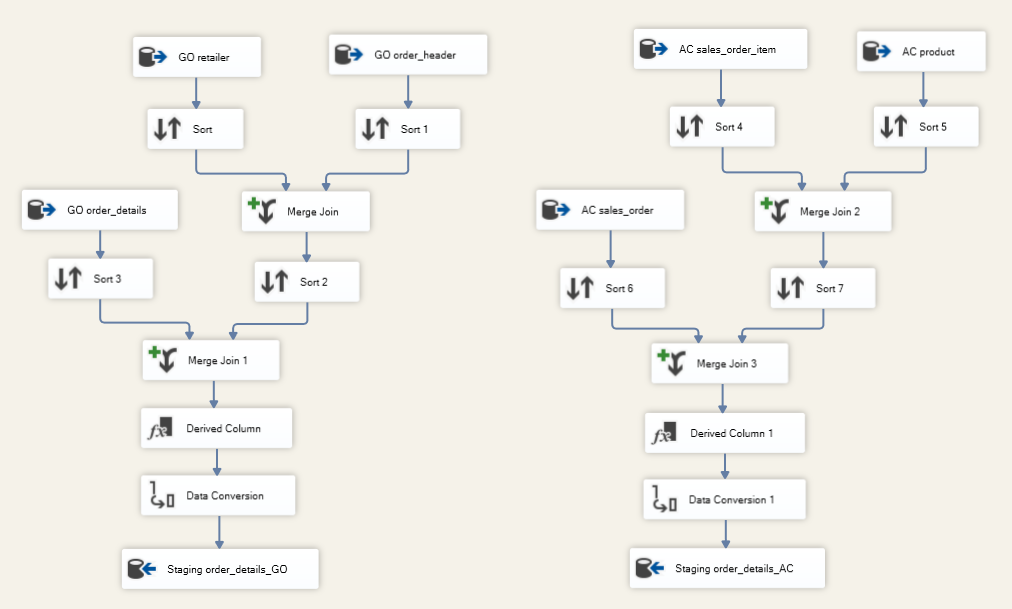
Dataflow van source employee naar staging employee



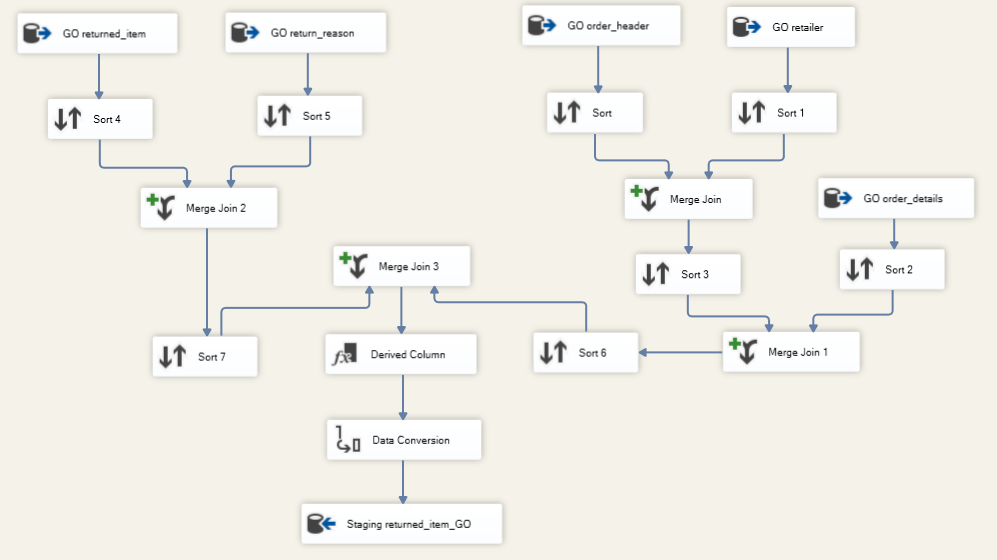
Dataflow van source customer naar staging customer



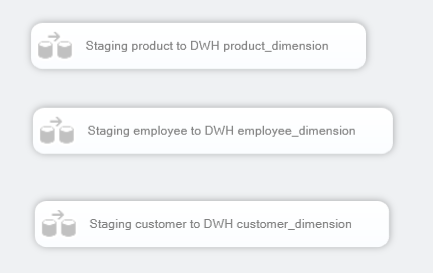
Dataflow van source order\_header naar staging order\_header



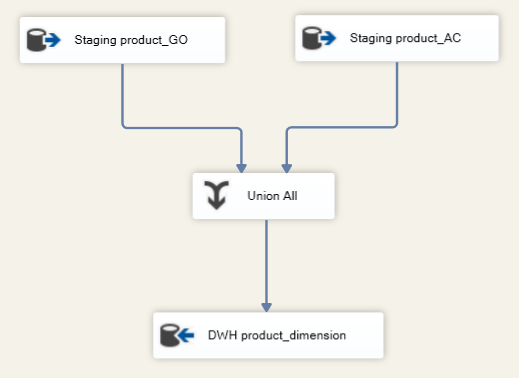
Dataflow van source order\_details naar staging order\_details



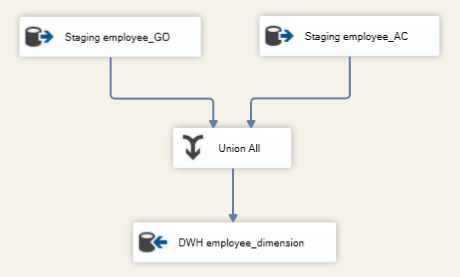
Dataflow van source returned\_item naar staging returned\_item



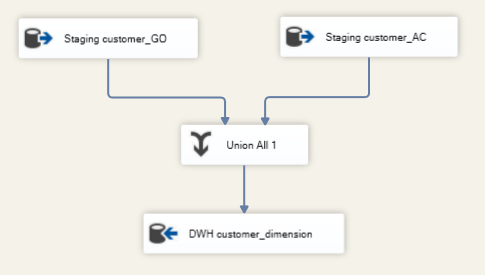
Package die alle dataflows van de staging naar de datawarehouse dimensies bevat



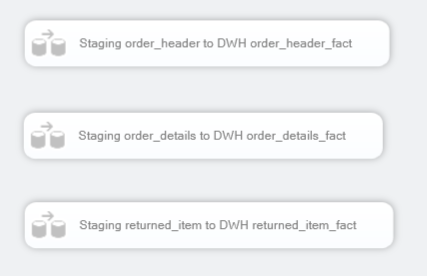
Dataflow van staging product naar datawarehouse product\_dimension



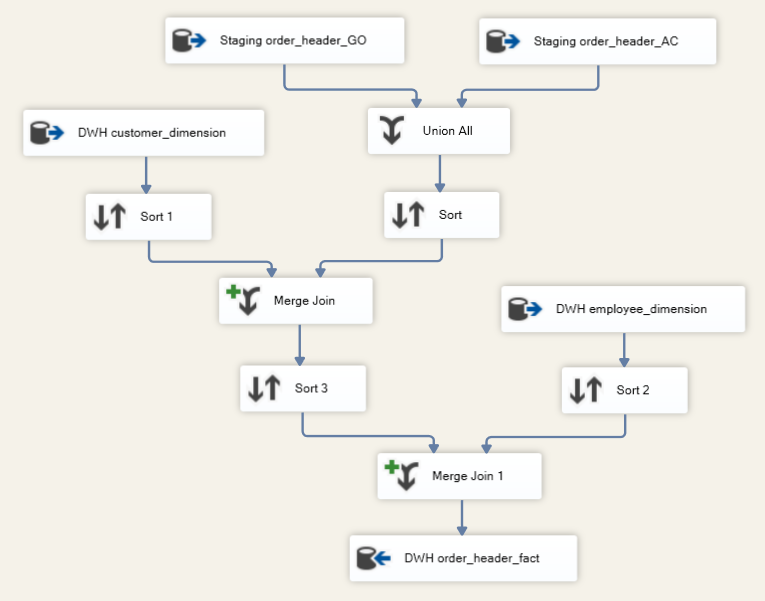
Dataflow van staging employee naar datawarehouse employee\_dimension



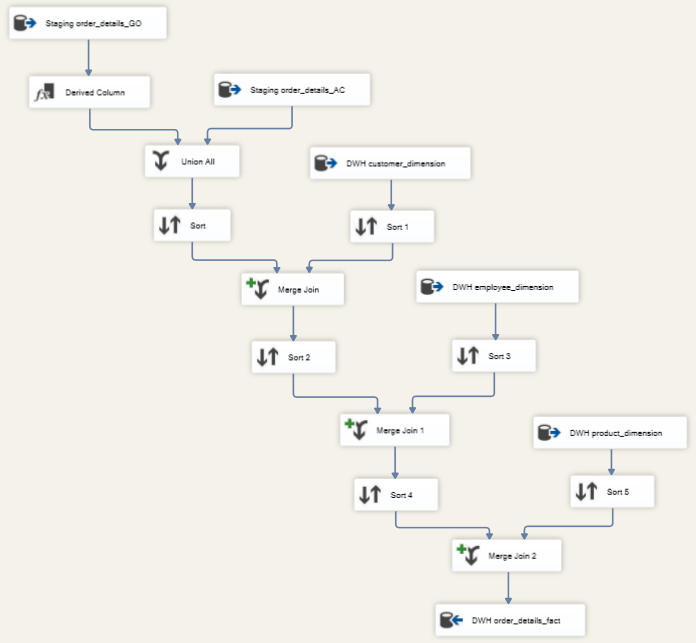
Dataflow van staging customer naar datawarehouse customer\_dimension



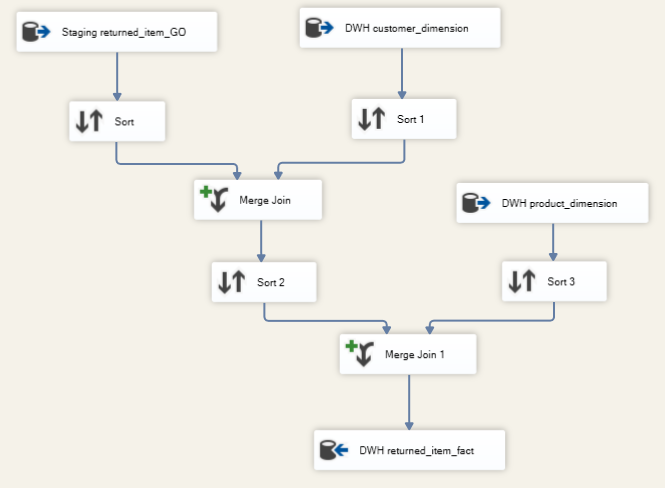
Package die alle dataflows van de staging naar de datawarehouse feiten bevat



Dataflow van staging order\_header naar datawarehouse order\_header\_fact



Dataflow van staging order\_header\_details naar datawarehouse order\_header\_details\_fact



Dataflow van staging returned\_item naar datawarehouse returned\_item\_fact

**Back-up van de datawarehouse, gevuld met gegevens d.m.v. de dataflows**



## 5.5 Test/waarborginggegevenskwaliteit

Neem hieronder de teststrategie op die jullie hebben toegepast voor de ET (bijvoorbeeld met aantallen).

Welke keuzes hebben jullie gemaakt? Neem de testresultaten op.

Waarborginggegevenskwaliteit:

* Let op dubbele records
* Noteer de records
* Neem ze niet mee in eventuele merges in het ETL-proces



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Target Table** | **Target Column** | **Target Data-type** | **Source Table** | **Source Column** | **Source Data-type** | **Duplicate data** | **Explanation** |
| **Name of target table** | **Column in target table** | **Data type for this target column** | **Name of the Source table** | **Column in source table where you would get this data from** | **Data type for this source column** | Is there duplicate data in the table? | Explanation for the duplicate data |
| employee\_dimension | id | int |  |  |  | No |  |
| staff\_code | int | GO: SALES\_STAFF | SALES\_STAFF\_CODE | int | No |  |
| AC: employee | emp\_id | int | No |  |
| first\_name | varchar(255) | GO: SALES\_STAFF | FIRST\_NAME | nvarchar(25) | No |  |
| AC: employee | emp\_fname | varchar(255) | No |  |
| last\_name | varchar(255) | GO: SALES\_STAFF | LAST\_NAME | nvarchar(30) | No |  |
| AC: employee | emp\_lname | varchar(255) | No |  |
| source | varchar(255) |  |  |  | No |  |
| record\_start\_date | date | GO: SALES\_STAFF | DATE\_HIRED | datetime | No |  |
| AC: employee | start\_date | datetime | No |  |
| record\_end\_date | date | GO: |  |  | No |  |
| AC: employee | termination\_date | datetime | No |  |
| customer\_dimension | id | int |  |  |  | No |  |
| retailer\_code | int | GO: RETAILER | RETAILER\_CODE | int | No |  |
| AC: customer | id | int | No |  |
| company\_name | varchar(255) | GO: RETAILER | COMPANY\_NAME | nvarchar(50) | No |  |
| AC: customer | company\_name | varchar(255) | No |  |
| country | varchar(255) | GO: COUNTRY | COUNTRY | nvarchar(50) | No |  |
| AC: |  |  | No |  |
| state | varchar(255) | GO: RETAILER\_SITE | REGION | nvarchar(50) | No |  |
| AC: customer | state | varchar(255) | No |  |
| address | varchar(255) | GO: RETAILER\_SITE | ADDRESS1 | nvarchar(50) | No |  |
| AC: customer | address | varchar(255) | No |  |
| discount | float | GO: |  |  | No |  |
| AC: customer | discount | int | No |  |
| source | varchar(255) |  |  |  | No |  |
| record\_start\_date | date | GO: |  |  | No |  |
| AC: |  |  | No |  |
| record\_end\_date | date | GO: |  |  | No |  |
| AC: |  |  | No |  |
| product\_dimension | id | int |  |  |  | No |  |
| product\_number | int | GO: PRODUCT | PRODUCT NUMBER | int | No |  |
| AC: product | id | int | No |  |
| product\_name | varchar(255) | GO: PRODUCT | PRODUCT\_NAME | nvarchar(255) | No |  |
| AC: product | name | varchar(255) | No |  |
| product\_type | varchar(255) | GO: PRODUCT\_TYPE | PRODUCT\_TYPE\_EN | nvarchar(50) | No |  |
| AC: product | category | varchar(1) | No |  |
| source | varchar(255) |  |  |  | No |  |
| record\_start\_date | date | GO: PRODUCT | INTRODUCTION\_DATE | datetime | No |  |
| AC: |  |  | No |  |
| record\_end\_date | date | GO: |  |  | No |  |
| AC: |  |  | No |  |
| time\_dimension | full\_date | date |  |  |  | No |  |
| year | int |  |  |  | No |  |
| quarter | int |  |  |  | No |  |
| month | int |  |  |  | No |  |
| day | int |  |  |  | No |  |
| order\_header\_fact | order\_number | int | GO: ORDER\_HEADER | ORDER\_NUMBER | int | No |  |
| AC: sales\_order | id | int | No |  |
| time\_id | date | GO: ORDER\_HEADER | ORDER\_DATE | datetime | No |  |
| AC: sales\_order | order\_date | datetime | No |  |
| customer\_id | int | GO: RETAILER | RETAILER\_CODE | int | No |  |
| AC: sales\_order | cust\_id | int | No |  |
| employee\_id | int | GO: ORDER\_HEADER | SALES\_STAFF\_CODE | int | No |  |
| AC: sales\_order | sales\_rep | int | No |  |
| source | varchar(255) |  |  |  | No |  |
| order\_header\_details\_fact | order\_detail\_code | int | GO: ORDER\_DETAILS | order\_detail\_code | int | No |  |
| AC: sales\_order\_item | id | int | No |  |
| order\_line\_id | int | GO: |  |  | No |  |
| AC: sales\_order\_item | line\_id | int | No |  |
| time\_id | date | GO: ORDER\_HEADER | ORDER\_DATE | datetime | No |  |
| AC: sales\_order | order\_date | datetime | No |  |
| customer\_id | int | GO: RETAILER | RETAILER\_CODE | int | No |  |
| AC: sales\_order | cust\_id | int | No |  |
| product\_id | int | GO: ORDER\_DETAILS | PRODUCT\_NUMBER | int | No |  |
| AC: sales\_order\_item | prod\_id | int | No |  |
| employee\_id | int | GO: ORDER\_HEADER | SALES\_STAFF\_CODE | int | No |  |
| AC: sales\_order | sales\_rep | int | No |  |
| quantity | int | GO: ORDER\_DETAILS | QUANTITY | int | No |  |
| AC: sales\_order | quantity | int | No |  |
| unit\_price | float | GO: ORDER\_DETAILS | UNIT\_PRICE | float | No |  |
| AC: product | unit\_price | float | No |  |
| total\_price | float | GO: |  |  | No |  |
| AC: |  |  | No |  |
| source | varchar(255) |  |  |  | No |  |
| returned\_item\_fact | return\_code | int | GO: RETURNED\_ITEM | RETURN\_CODE | int | No |  |
| does not exist in AC |
| time\_id | date | GO: RETURNED\_ITEM | RETURN\_DATE | datetime | No |  |
| does not exist in AC |
| customer\_id | int | GO: RETAILER | RETAILER\_CODE | int | No |  |
| does not exist in AC |
| product\_id | int | GO: ORDER\_DETAILS | PRODUCT\_NUMBER | int | No |  |
| does not exist in AC |
| quantity | int | GO: RETURNED\_ITEM | RETURN\_QUANTITY | int | No |  |
| does not exist in AC |
| reason | varchar(255) | GO: RETURN\_REASON | RETURN\_DESCRIPTION\_EN | nvarchar(50) | No |  |
| does not exist in AC |
| source | varchar(255) |  |  |  | No |  |

**Uitleg waarom er geen duplicate data zal plaatsvinden**

Er is geen duplicate data aanwezig omdat van de 4 beschikbare databronnen er 2 onnodig zijn om onze vragen te beantwoorden.

De volgende databronnen zijn aan ons gegevens.

* GreatOutdoor DB
* GreatOutdoor trainings.xls
* A&C DB
* A&C HRM.mdb

De volgende databronnen worden niet gebruikt:

* GreatOutdoor trainings.xls
* A&C HRM.mdb

Doordat deze 2 databronnen niet gebruikt worden, komt dezelfde data niet meerdere keren voor. Met name A&C HRM.mdb die exact dezelfde data bevat als A&C DN in enkele tabellen. De kans dat duplicate data bestaat in de databases bestaat nog wel, maar nadat wij deze data handmatig gecontroleerd hebben, is het nu zeker dat er geen duplicate data voorkomt in deze bronnen.

**Test code**

Hieronder staat de code die wij gebruikt hebben voor de databases om te controleren hoeveel rijen er zijn per databasetabel en wat er verwacht wordt in de data warehouse.

GO  
SELECT COUNT(ORDER\_DETAIL\_CODE)  
FROM ORDER\_DETAILS

A&C  
SELECT COUNT(sales\_order.id)  
FROM sales\_order

GO resultaat: 43063   
A&C resultaat: 650  
order\_details\_fact verwachting: 43713

GO  
SELECT COUNT(SALES\_STAFF\_CODE)  
FROM SALES\_STAFF

A&C  
SELECT COUNT(emp\_id)  
FROM employee

GO resultaat: 102  
A&C resultaat: 78  
employee\_dimension verwachting: 180

GO

SELECT COUNT(PRODUCT.PRODUCT\_NUMBER)

FROM PRODUCT

A&C

SELECT COUNT(PRODUCT.ID)   
FROM PRODUCT

Go resultaat: 115  
A&C resultaat: 10  
Product\_dimension verwachting: 125

GO

SELECT COUNT(RETAILER.RETAILER\_CODE)

FROM RETAILER

A&C

SELECT COUNT(CUSTOMER.ID)   
FROM CUSTOMER

Go resultaat: 109  
A&C resultaat: 126  
Customer\_dimension verwachting: 235

## 5.6 Aanpassingen in ontwerp en datawarehouse als gevolg van beveiligingsmaatregelen

Denk eraan dat de securitymaatregelen ook leiden tot aanpassingen in het ontwerp en de datawarehouse. Neem dus hieronder ook de aangepaste ontwerpen en scripts op voor de volledigheid.

Het datawarehouse heeft zelf geen ontwerpveranderingen. Ter beveiliging van de gegevens die in de datawarehouse zitten, zijn er scripts gemaakt waarbij er bepaalde rechten worden toegekend of afgedwongen aan de gebruiker.

1. **Hoeveel omzet wordt er behaald en hoeveel producten worden er verkocht in een maand, per jaar, per producttype, per product, per land van de klant, per klant?**

De beveiliging hier is dat er een groep komt genaamd “Marketing” die tot 5 jaar terug kunnen inzien hoeveel producten worden verkocht. Er kunnen alleen de benodigde gegevens worden gelezen zoals de product en de hoeveelheid.

Script:

Create role marketing;

grant SELECT (product\_id, quantity)

on order\_details\_fact

to marketing

with GRANT OPTION;

1. **Hoeveel retourneringen zijn er per product, per product type, per reden, per klant die verkocht is door een verkoper op een bepaald moment?**

De beveiliging dat hier zal worden toegepast is dat er een groep genaamd “Retour” komt met de rechten om de hoeveelheid retourneringen te bekijken met alleen benodigde gegevens zoals per product, per producttype en per klant die verkocht is door een verkoper

Script:

Create role retour;

grant SELECT (quantity, product\_id, customer\_id)

on returned\_item\_fact

to marketing

with GRANT OPTION;

1. **Hoeveel facturen worden er gemaakt per maand, per werknemer, per klant?**

De beveiliging dat hier zal worden toegepast is dat er een groep genaamd “manager” komt met de rechten om de facturen te bekijken die per maand per werknemer per klant wordt gemaakt.

Script:

Create role manager;

grant SELECT (employee\_id, time\_id, customer\_id)

on order\_header\_fact

to marketing

with GRANT OPTION;

1. **Wat is het meest bestelde product per maand, per werknemer, per korting van de klant?**

De beveiliging dat hier zal worden toegepast is dezelfde groep genaamd “Marketing” met de rechten om de gemiddelde korting per klant per staat te bekijken.

Script:

grant SELECT (id, discount, retailer\_code)

on customer\_dimension

to marketing

with GRANT OPTION;

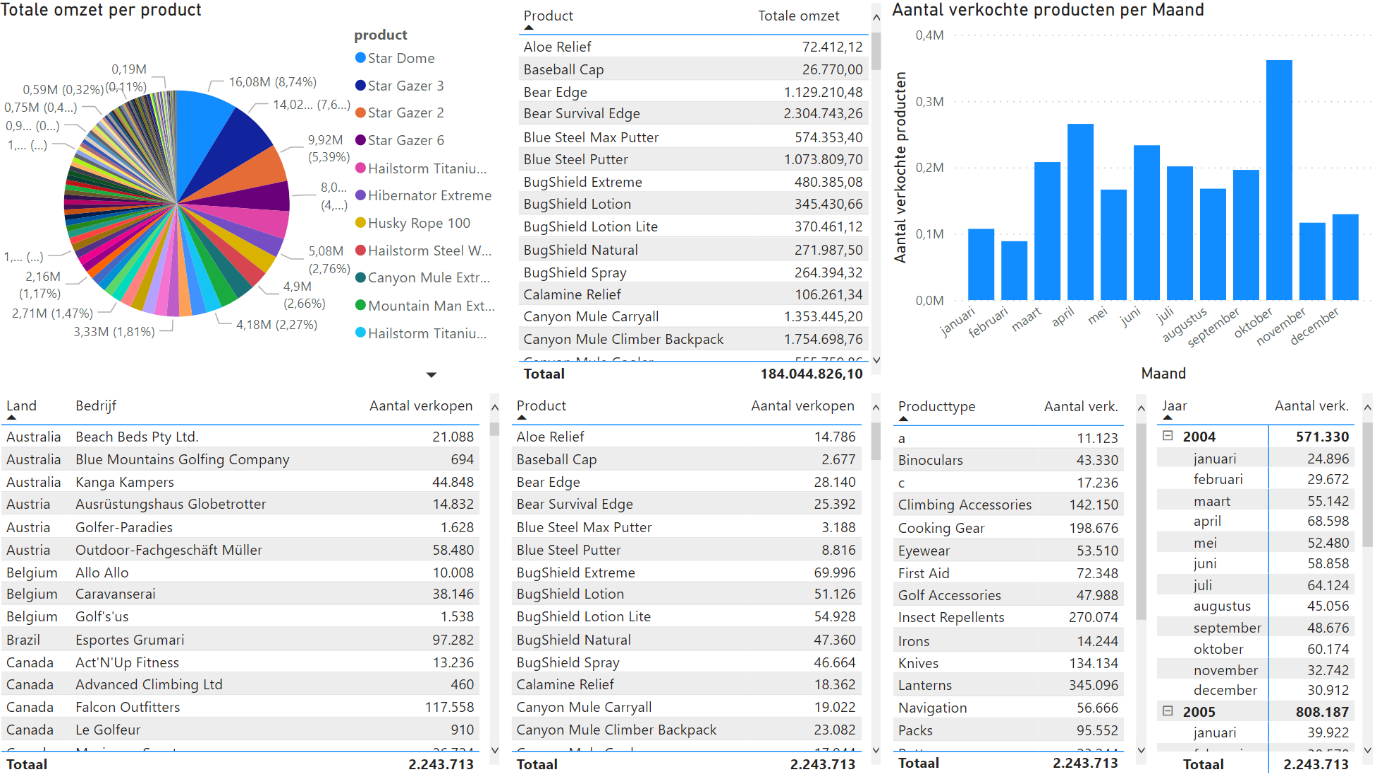
# Overzicht van de Dashboards

Neem hieronder de gemaakte dashboard(s) op die gemaakt zijn middels screenshots en daarnaast ook de file(s) als embedded object. Neem ook de schetsen (ontwerpen) op. Geef daarnaast ook aan hoe de managementvragen beantwoord worden. Ga ook na als de gegevens die in het dashboard getoond worden, ook klopt met de gegevens uit de datawarehouse.

Pas de beveiligingsmaatregelen toe en neem de tests van deze beveiligingsmaatregelen ook op indien jullie gekozen hebben om de security op dashboardniveau uit te voeren. Onderbouw keuzes die jullie hebben moeten maken.



**Vraag 1:** Hoeveel omzet wordt er behaald, en hoeveel producten worden er verkocht per maand, per jaar, per producttype, per product, per land van de klant en per klant?

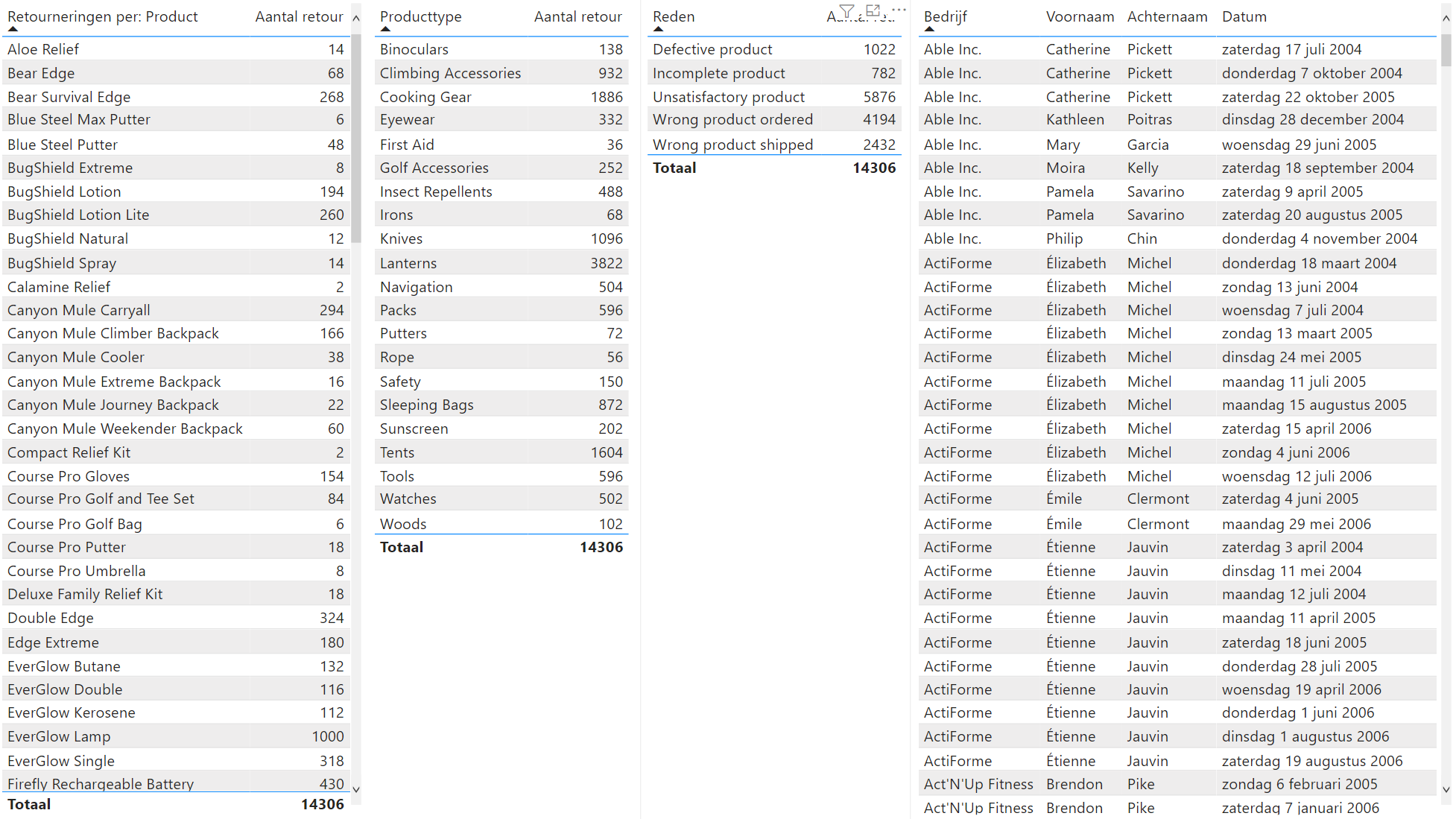


**Omzet**: Als ringdiagram en tabel naast elkaar, zodat er een overzicht is welk product welk percentage uitmaakt van de omzet, en een tabel met de daadwerkelijke omzet in euro’s met een totaalsom.

**Per maand, per jaar:**Een staafdiagram dat het totaal per maand laat zien van alle jaren, en daar onder een matrixtabel, waar de jaren uitgeklapt kunnen worden, waar de verkochte producten per jaar en per maand in getallen gezien kunnen worden, en mits geselecteerd, dan past het staafdiagram zich aan, aan de selecteerde maand/jaar.

**Alle overige tabellen:** (Matrix)tabellen met hun respectievelijke data, afhankelijk van de vraag wat voor data er in staat.

**Vraag 2:** Hoeveel retourneringen zijn er per product, per product type, per reden, per klant, welke verkocht zijn door een verkoper op een bepaald moment?



**Hoeveelheid retourneringen:** De hoeveelheid retourneringen zijn aangegeven bij elke tabel als het gegeven “totaal” onderin de tabellen.

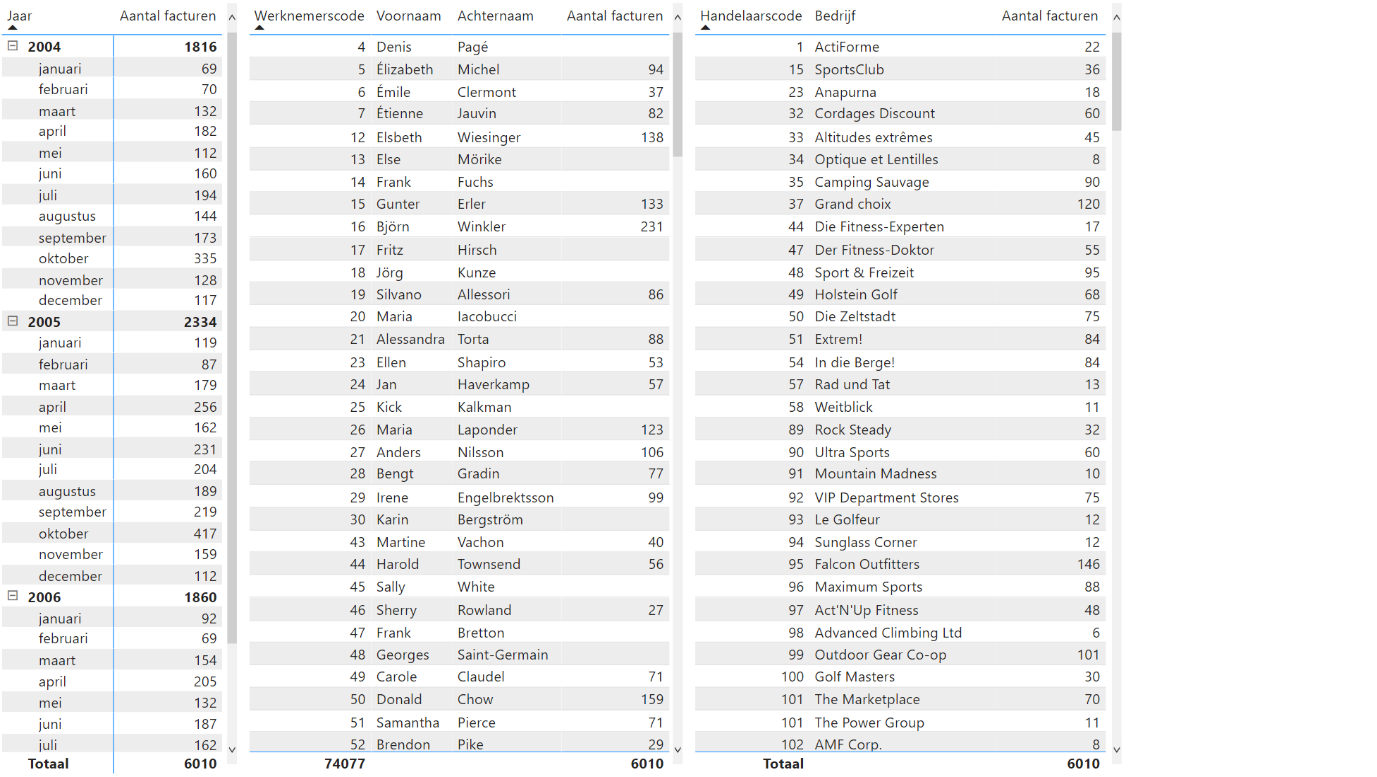
**Per product:** In de 1e tabel van het bestand rapport DWDH herkansing (op pagina 2) staat de hoeveelheid retours per product aangegeven met de naam en de aantal retours die zich plaats hebben gevonden per product.

**Per producttype:** In de 2e tabel van het bestand rapport DWDH herkansing (op pagina 2) staat de naam van het producttype er, met daarnaast het aantal retours dat dit producttype heeft gehad.

**Per reden: :** In de 3e tabel van het bestand rapport DWDH herkansing (op pagina 2) staan de aangegeven redenen voor de retours met daarnaast het aantal dat de reden voor komt bij de retourneringen.

**Per klant, op een bepaald moment:** **:** In de 3e tabel van het bestand rapport DWDH herkansing (op pagina 2) staat de voor en achternaam van de klant. Er wordt bij deze tabel vermeld bij welk bedrijf deze klacht is ingediend en daarnaast staat de datum van retour erbij.

**Vraag 3:** Hoeveel facturen worden er gemaakt per maand, per werknemer, per klant?



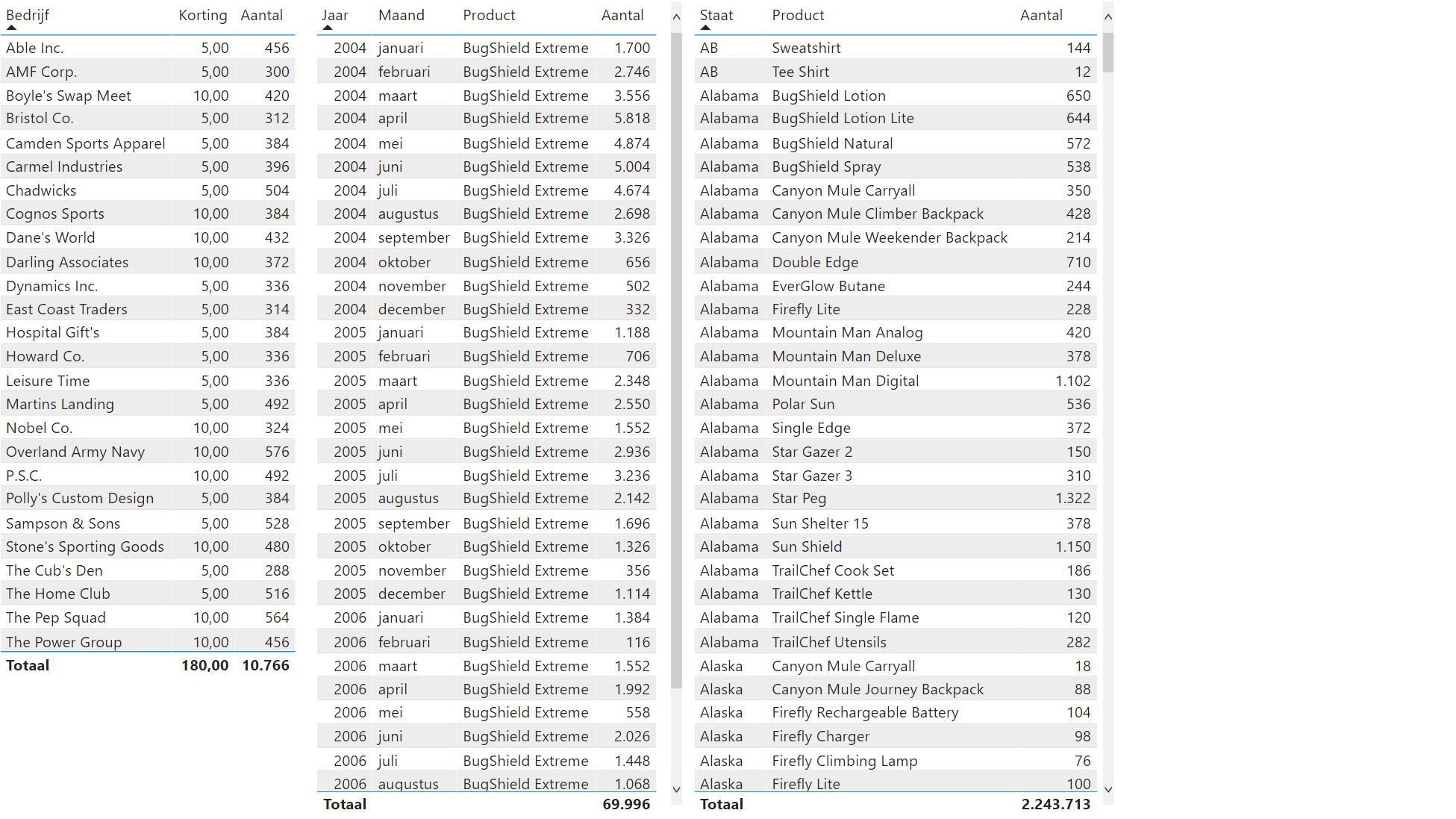
**Hoeveel facturen:** De hoeveelheid facturen zijn aangegeven bij elke tabel als het gegeven “totaal” onderin de tabel.

**Per maand: :** In de 1e tabel van het bestand rapport DWH herkansing (op pagina 3) staat het jaar en de maand aangegeven in de linker kolom, in de rechter kolom staat hoeveel facturen er in deze maand gebruikt zijn.

**Per werknemer:** In de 2e tabel van het bestand rapport DWH herkansing (op pagina 3) staat de code van de werknemer met voor en achternaam. Uiterst rechts van deze tabel staat hoeveel facturen de werknemer heeft behandeld.

**Per klant:** In de 3e tabel van het bestand rapport DWH herkansing (op pagina 3) staat de code van de klant (handelaarscode), de naam van de klant (dit zijn bedrijven), een aan het einde staat het aantal facturen per bedrijf.

**Vraag 4:** Wat is het meest bestelde product per maand, per werknemer, per korting van de klant?



**Meest bestelde product:** Het meest bestelde product is te vinden door het aantal t soorteren in de 1e tabel.

**Per korting van de klant:** In de 1e tabel van het bestand rapport DWDH herkansing (op pagina 4) staan de bedrijven met de hoeveelheid korting die ze verkopen per product.

**Per maand:** In de 2e tabel van het bestand rapport DWDH herkansing (op pagina 4) staat het jaartal en de maand van de bestelde producten aangegeven.

**Per werknemer:** In de 3e tabel van het bestand rapport DWDH herkansing (op pagina 4) staan het bedrijf met de locatie waar het bedrijf zich bevind. Het aantal verkochte producten staat rechts van het bedrijf.

**Belangrijk:** Deze uitwerking is niet volledig correct, er missen onderdelen in dit stuk. Het is niet gelukt de vierde vraag uit te werken, aangezien ik het niet voor elkaar kreeg om met zowel TOPN, MAXX, als andere manieren, het meest bestelde product kon krijgen, zonder dat elk product dezelfde naam had (zie de middelste kolom). Een vergelijkbaar probleem had ik bij de andere twee, dus deze vraag hebben we niet volledig af kunnen ronden.

# Reflectie & bewijslast

Neem per groepslid de reflectie op de leerdoelen per week.

Neem ook per groepslid een logboek op per week en bewijslast van wat gedaan is per groepslid in relatie tot de leerdoelen. Wat heb je geleerd en hoe heb je dit aangepakt?

Hieruit moet het aandeel van elk groepslid duidelijk naar voren komen en moet dus ook aansluiten op de reflectie en de gemaakte planning.

Voer dit elk week uit nadat de werkzaamheden zijn uitgevoerd voor die week.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Week** | **Leerdoel** | **Groepslid** | **Logboek** | **Reflectie** |
| 1 | Verschil tussen OLTP en OLAP, de casus doornemen en een bronnenmodel maken. | Akash | De inleiding is afgemaakt en de manager vragen zijn erin gezet. | Ik heb geleerd wat het doel van een datawarehouse is en hoe de business vragen leidend zijn. |
| Richal | Casus gelezen. Diagram gemaakt om een globaal beeld te krijgen van de gegevens m.b.v. [www.draw.io](http://www.draw.io). en Visual Paradigm | Het opstellen van een bronnenmodel geeft een goed overzicht van de benodigde bronnen voor de datawarehouse. Ik heb geleerd dat dit anders is dan een ontwerp voor een productiedatabase. |
| Roderick | Geholpen met het bedenken en noteren van manager vragen. | Het bedenken van manager vragen heeft me laten zien wat voor informatie een datawarehouse uiteindelijk moet leveren. |
|  |  |  |
| 2 | Leren wat een feitmodel is en wat dimensies zijn en hoe dit later het sterschema zal vormen. | Akash | Feiten opgesteld voor het sterschema | Met behulp van het ontwerp.pdf document heb ik aan de hand van de voorbeelden geleerd hoe een feitenmodel gemaakt wordt. |
| Richal | Geholpen met dimensies van de feiten opstellen en mogelijke meetwaardes en berekeningen | Het sterschema maakt uiteindelijk gebruik van 1 feittabel met meerdere dimensietabellen. Het leren van goede dimensies opstellen zal later in het sterschema ontwerp goed van pas komen. |
| Roderick | Gegevensmodel van Great Outdoor database gegenereerd. A&C moet nog gebeuren, het genereren is hier moeilijker. | Het gegevensmodel van Great Outdoor heeft me geleerd om tabellen die veel FK’s bevatten te identificeren als feit en de tabel waar de FK’s naar leiden als dimensies. |
|  |  |  |
| 3 | Het sterschema creëren en fysiek ontwerp maken | Akash | Geholpen met het sterschema | Ik heb geleerd hoe het sterschema gemaakt moest worden en wat de doel hiervan is geweest. |
| Richal | Management vraag 3 opnieuw gemaakt i.v.m. feedback. Vraag 4 aangepast | In week 1 waren onze 2 verzonnen vragen niet volledig goedt omdat ik niet goed bergeep wat iets een dimensie maakt. Vraag 3 werd |
| Roderick | Dimentie matrix aangepast op basis van feedback, | Geleerd dat de dimentiematrix in 1 tabel moet |
| Jimmy | Stof ingehaald, niks nieuws gemaakt. | - |
| 4 |  | Akash | Mapping schema deels gemaakt | Ik heb geleerd waarom het nodig is om een mapping schema te maken en hoe dit gemaakt moet worden |
| Richal | Sterschema diagram en ERD gemaakt | Een duidelijk beeld gecreëerd van de uiteindelijke datawarehouse tabellen en de attributen. |
| Roderick | Het ETL-dataflow ontwerp maken, het dimentiematrix/dimentie hiërarchieën aanpassen op basis van een aanpassing aan vraag 3 | Het dataflow ontwerp afgerond en de hiërarchieën kunnen aanpassen op basis van vraag 3 |
| Jimmy | Het was mijn taak om met SSIS de ETL scripts te maken, maar vanwege mijn persoonlijke klachten (oververmoeidheid, vermeld tijdens voortgangsoverleg maandag week 5) is dit helaas niet gelukt. | Het is gelukt ETL te installeren in zowel SSMS als in Visual Studio, het is mij helaas alleen niet gelukt hier vervolgens wat mee te doen i.v.m. hetgeen hier links beschreven. |
| 5 |  | Akash | Beveiligingsmaatregelen opgesteld | Ik heb geleerd waarom er beveiligingsmaatregelen zijn en voor wat dit precies goed is |
| Richal | SSIS geïnstalleerd en uitzoeken hoe het programma werkt. | Door 1 persoon van de groep bekend te laten worden met het programma kunnen andere groepsleden om hulp vragen aan deze persoon. Ik heb geleerd hoe SSIS precies te werk gaat met het connecten en verwerken van data uit verschillende plekken. |
| Roderick | Test en waarborging plan opgesteld | Ik heb geleerd waarom je tests uitvoert voor het maken van een nieuwe database |
| Jimmy | Het was mijn taak de anderen te helpen met hun taken indien het nodig was en informatie over Power BI lezen, zodat ik kon beginnen als we de databasewarehouse af hadden. | Ik heb geholpen met de testcode te maken, en heb daarnaast online documenten over Power BI doorgenomen, Power BI geïnstalleerd, en wat test tabellen gemaakt op basis van de GreatOutdoor database, om het uit te proberen, wat prima lukte. |
| 6 |  | Akash | Geholpen met het maken van de ETL en de queries van de security noteren. Verder het document grondig bekeken | Ik heb geleerd hoe je met ETL te werk gaat en hoe je de connecties maakt met de servers. |
| Richal | Dataflows gemaakt en scripts deels geschreven voor de tabellen in de staging database | Op de dag van de deadline was er nog veel werk te doen. De scripts en dataflows moesten naast de datawarehouse ook gemaakt worden voor de staging database. Ik heb geleerd hoe het totale ETL-proces in aparte delen gebroken kan worden om het simpeler te maken. Helaas missen er een paar scripts en dataflows. Ook was het best moeilijk om MySQL te verbinden met SSIS waardoor te dataflows runnen nogal irritant werd. |
| Roderick | Test en waarborging uitwerken, daarnaast uitleg gegeven bij vorige vragen | Ik heb geleerd hoe je tests uitvoert voor het maken van een nieuwe database. Ik heb daarnaast geholpen met de scripts van de dataflow. |
| Jimmy | Maken van rapporten en dashboards. Niet gelukt, omdat het maken van de datawarehouse niet gelukt is. Wel is er een rapport van vraag 1 van de GreatOutdoor database toegevoegd. | Ik heb geleerd hoe je met Power BI rapporten moet opzetten en deze in moet vullen. Het was even uitvogelen hoe tabellen en grafieken gemaakt moeten worden met de onderlinge relaties, maar dat is (vind ik) goed gelukt. Ook al is het alleen gelukt voor vraag 1 en niet met de correcte database, vind ik het eindresultaat van dit rapport, op de lange getallen na, erg overzichtelijk. Het enigste wat mij niet gelukt is, is de naam van het producttype te krijgen in de tabel rechts onderin, ik kreeg het gewoon niet werkend. |

# Literatuurlijst

Neem hier alle bronnen op volgens APA-vermelding

Berry Pieters (2017). Datawarehouse Design, video 01 t/m 06  
<https://www.youtube.com/playlist?list=PLT1UBlxXIqEkvSX5J3Xa_magH71WQqIzu>

TechBrothersIT (2021). How to Load Data From SQL Server to MySQL in SSIS Package Step by Step - SSIS Tutorial 2021 <https://www.youtube.com/watch?v=dpLF5sGSJek>

Hadi Fadlallah (2019). SSIS Derived Columns with Multiple Expressions vs Multiple Transformations  
<https://www.sqlshack.com/ssis-derived-columns-with-multiple-expressions-vs-multiple-transformations/>

Learn SSIS (2017). 23 Data Conversion Transformation in SSIS  
<https://www.youtube.com/watch?v=aQJCky2qfCs>

Anette Allen (2013). SSIS Basics: Using the Merge Join Transformation  
<https://www.red-gate.com/simple-talk/databases/sql-server/bi-sql-server/ssis-basics-using-the-merge-join-transformation/>

Connector/ODBC 8.0.25 for ODBC connection managers  
<https://dev.mysql.com/downloads/connector/odbc/>

Aan de slag met Power BI Desktop (2021) David Isminger  
<https://docs.microsoft.com/nl-nl/power-bi/fundamentals/desktop-getting-started>

Gegevensbronnen in Power BI Desktop (2021) David Isminger  
<https://docs.microsoft.com/nl-nl/power-bi/connect-data/desktop-data-sources>