Productverantwoording project Kramse

Geschreven door: Xin Wang, 2154458

[**Context 3**](#_y2lln69ndx9x)

[Doelen 3](#_3m9i9alq9efu)

[Scope: 3](#_c4hrnxv5cee1)

[Stakeholders 4](#_jlvkrudmodpj)

[**Informatiebehoeften 4**](#_uzqd2kvq2p2s)

[Aannames 4](#_u1c4hi52f9xa)

[**ETL-Proces 7**](#_oimdydhe1jjz)

[Informatiebronnen 7](#_j0bwmhczgtf1)

[Verschillende databronnen naar 1 database 7](#_mzi5hr9kx2fp)

[Raw data naar Relationeel model 8](#_t1c82ds3mvtc)

[Consistentie Problemen oplossen. 10](#_vmqzvzs6m0ej)

[ETL Proces van RDBS -> PSA -> ODS 11](#_5s72mkslacat)

[Ontwerp staging area/PSA. 12](#_4xlfidisygso)

[Type schema 12](#_twn4mneseshp)

[Berekeningen en aggregaties tussen RDBS en PSA. 13](#_tzq3lauzdpl1)

[Dimensietabellen 15](#_nqnwb2crbskt)

[Slowy-changing dimensions 15](#_odae1noneb4j)

[Feitentabellen. 16](#_tip06s5hwnan)

[**Dashboard 19**](#_prh5vxawat8c)

[Vormgeving 19](#_sd860h8njt28)

[Conclusies dashboard 20](#_k7l5pikwu6py)

[Mogelijke beslissingen 20](#_40nd5qwwzso0)

[**Bijlagen: 21**](#_n7iiwf6pn43m)

# Context

Het bedrijf Kramse heeft als missie om de supply chain te verbinden en te vereenvoudigen over de hele wereld. Men wil dat bereiken als een geïntegreerde organisatie, met meerdere merken onder hun beheer.

Ook wil men hun doelen behalen door een top-line growth, een groei in hun brutowinst en omzet.

Naast de bedrijfsmissie, streeft de organisatie ernaar om hun Sustainable Development doelen te behalen. Die de verenigde naties heeft opgezet in 2015. In dit geval wil men hun emissies omlaag halen.

## Doelen

Om hun bedrijfsstrategie en SDG’s te behalen, heeft Kramse een aantal doelen opgesteld.

Voor het verlagen van de CO2 uitstoot, wil men ervoor zorgen dat het brandstofverbruik met 10 procent daalt, per vracht ton per zeemijl. Dit is niet mogelijk met de beschikbare data, omdat het maar over 1 jaar gaat en er dus geen vergelijkingen gemaakt kunnen worden. Plus er ontbreken verschillende data die nodig zijn om dit doel te behalen. Er kan dus alleen een trend weergeven worden.

Een ander doel is dat de organisatie ervoor wil zorgen dat de utilisatiegraad verbeterd door de wachttijden in de havens te reduceren naar 2 dagen per wacht. Dat is anders dan de opdracht, omdat volgens bronnen het laden en uitladen van de containers 2 dagen duurt, 5 procent reductie is te laag en dus is 2 dagen wachttijd een beter doel. Daarnaast leidt een 5 procent reductie in halve dagen, wat niet tot nuttige resultaten leiden.

Tenslotte streeft de organisatie ernaar dat de beladingsgraad per schip per vaarroute geoptimaliseerd wordt naar 90 procent.

## Scope:

* Dit rapport zal gerelateerd zijn aan de brandstofverbruik per schip per trip, in verband met de Co2 emissies
* In dit rapport zal de wachttijd per haven en per schip weergeven worden.
* Omdat er binnen de dataset alleen gegevens bezit over het jaar 2016, kan er niet vergelijkingen gemaakt worden over de brandstof reductie ten opzichte van voorgaande jaren.
* Dit rapport heeft betrekking tot de utilisatiegraad van ieder schip per trip, om te controleren of ieder schip aan zijn beoogde utilisatiegraad doel voldoet.
* Het rapport kan geen kosten over meerdere jaren weergeven, omdat de gegevens in de database van 1 jaar zijn.

## Stakeholders

De partijen die er baat bij hebben om deze doelen zullen mensen die handelen op tactisch niveau. Ze kunnen daarom geclassificeerd worden als Explorer, omdat de organisatie naast hun vaste informatiebehoeften zoals wachttijd en brandstofverbruik ook erachter wil komen waarom deze data tot stand is gekomen. Daarnaast wil men weten of het ook aan de wetgeving voldoet voor co2 emissies.

## Informatiebehoeften

Aan de hand van de bedrijfsdoelen, kunnen een aantal informatiebehoeften in kaart brengen.

Om hun informatiebehoefte van de **utilisatiegraad** te bevredigen, zal deze resultaten in context gebracht worden in schip, haven en scheepvaart contexten. Hiervoor zal een KPI gemaakt worden om deze utilisatiegraad te weergeven. De minimale eis is 90%.

Het bedrijf wil er ook zorgen dat de **wachttijden** maximaal 2 dagen in de havens zitten. Deze gegevens zullen in de context van Haven en schip gezet worden.

Als laatste wil de organisatie voor iedere trip de **kosten** hebben. Ze zullen in de context van schip en haven gezet worden. De bedoeling hiervan is om de kosten te verlagen.

Voor **brandstofverbruik in kilo’s**, zal er per gemaakte zeemijlen per vracht ton het brandstofverbruik gemeten worden.

## Aannames

Sommige gegevens zoals het gewicht van de inhoud van de containers zijn niet aanwezig, zal er een aantal aannames genomen moeten worden.

Sommige gegevens waren berekend per kilo of per mijl. Dus heb ik ze omgerekend naar mijl. 1 mijl is 1,8 km.

Er zal vanuit gegaan moeten worden dat 1 TEU container 10 ton is. Dus een 2 TEU container is 20 ton. Dus 10000 en 20000 kilogram.

Verder waren sommige bedragen in dollars en euro’s, dus zijn alle bedragen in de datawarehouse in euro’s. Sinds de gegevens over 2016 waren, heb ik de gemiddelde wisselkoers van de Euro/Dollar van 2016 gepakt. Dat was in dit geval 1,1086.

Tussen voyageport wordt de begin en einddatum als datum eenheid geïndeindigt.

De grootste eenheid is een dag, dus je weet niet hoelaat een schip is vertrokken en aangekomen.

Daarom zal ik een aanname maken dat de schepen op topsnelheid varen. Ook zal de vaarsnelheid op dagniveau weergeven worden.

De brandstof voor schepen is van het type Heavy Fuel Oil. 1 liter is 0,96 kg.

Sinds een containerschip tussen 20 en 30 bemanningsleden bevat, pak ik het gemiddelde daarvan en ga ervan uit dat er op een schip 25 bemanningsleden bevatten.

Volgens een containerschip kapitein zijn de brandstof kosten gemiddeld 309 en 322 US Dollar per kiloton.

Voor deze scenario pak ik 322 US Dollar per ton om de brandstofkosten te berekenen.

Aanmeerkosten zullen we de haven van Rotterdam gebruiken om ermee te schatten.

Dat zal 3,36 euro per meter schip zijn.

De bunkerprijs is wereldwijdt gemiddeld 394,41 dollar per ton. Dat is omgerekend naar euro’s 355,77 euro.

Gemiddelde salaris voor een schiplid is

|  |
| --- |
| $43,842 |

Omgerekend is dat 39547,18 euro per jaar volgens de wisselkoers van de euro in 2016.

De laad en lossen kosten containers is 550 euro per uur.

Het laden en lossen van een container duurt 2 dagen. 48 Uur.

Zie het hoofdstuk Literatuur om de bronnen te bekijken waarop de verdere aannames gebasseerd zijn.

# 

## 

# ETL-Proces

## Informatiebronnen

Om aan deze informatiebehoeften te voorzien, maken we gebruik van de Kramse TPS MS Access Database. Ook maken we gebruik van een excel-bestand met de Consignor-info, met de land en korting in percentage. Daarnaast zal een tekst bestand geraadpleegd worden waarin de gegevens van containers bevatten. Voor het de data van de gebruikte schepen, zal deze data uit een Excel bestand komen.

Al deze databronnen zullen in een datawarehouse opgeslagen worden, waarin het als single-point of truth fungeert. Op basis van deze data warehouse, zullen beslissingen gemaakt kunnen worden.

## Verschillende databronnen naar 1 database

Om de verschillende databronnen in 1 database te zetten, nam ik alle gegevens van alle databronnen in hun eigen tabellen.

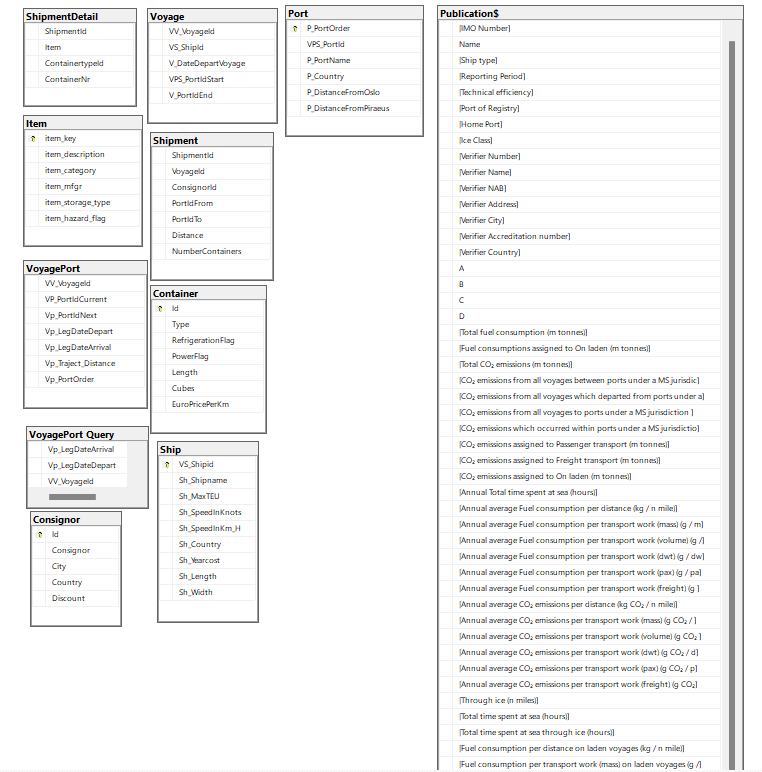
Voor de tekstbestand Container, excel-bestand en MRV excel-bestand heb ik de gegevens in aparte tabellen gezet. Voor de access database, nam ik alle tabellen uit die database over. Hiervan heb ik nog geen relaties gemaakt.

Bij het ophalen van de data wordt er gebruik gemaakt van een SSIS om dit proces te automatiseren. Hiermee kan er verbinding gemaakt met de verschillende databronnen en wordt dat in een SQL Server database opgeslagen. Hierin wordt bij iedere process de database verwijdert zodat ik iedere keer een momentopname heb van de data.

In deze fase van het proces, wordt er nog minimale bewerkingen gedaan. Bij het ophalen van de MRV gegevens wordt er een JOIN toegepast tussen de Shipname van beide tabellen, waarmee ik maar enkel Ship Records overhoud die in de ship tabel voorkomen. Verder wordt er nog enkele schipnamen gecorrigeerd. Er zijn geen relaties gelegd tussen tabellen, sommige tabellen zijn nog niet goed genormaliseerd en er zijn inconsistente data aanwezig.

Het doel is om alle data op 1 plek te zetten. Zie figuur 1 voor het resultaat.

Figuur 1. Database voor het ophalen van data uit databronnen, de Raw database.



## Raw data naar Relationeel model

In de volgende fase worden er bewerkingen gedaan, kolommen verwijdert relaties gelegd tussen bepaalde tabellen om tot een genormaliseerde database te komen. Zie figuur 2.

Voor de tabellen die direct uit de access database kwamen, worden de relaties die gelegd waren overgenomen. Vanuit de ShipmentDetails tabel wordt er een foreign key relatie gelegd naar de tabellen Containers en Item met de containerTypeid => Container.Id en Item => Item.ItemKey.

Vanuit Shipment tabel wordt er een foreign key relatie gelegd naar Consignor tabel.

Binnen de Ship tabel wordt er een relatie gelegd met de MRV bestand. Binnen de MRV bestand wordt een ShipId toegevoegd en wordt er een foreign key relatie gelegt naar de Shiptabel.

Daarnaast heb ik van iedere tabel kolommen verwijdert die niet relevant zijn van onze informatiebehoeften of die geen gegevens bevatten.

Van de MRV tabel heb ik alleen de kolommen, IMO-nummer, ShipId, ShipType, ShipName, Total fuel consumption, Total co2 emissions en annual co2 consumption(Hierin wordt co2 uitstoot in kg per mijl opgeslagen, andere naam) en fuel consumption\_kg\_miles(brandstofverbruik in kg per mijl) overgehouden. Omdat de andere kolommen gegevens ontbreken of afwezig zijn.

Ship tabel heb ik de kolom Shipcountry en Sh\_width verwijdert, omdat de breedte van de schip niet relevant was om de kosten, brandstofverbruik en beladingsgraad te berekenen. Ook wordt de Capaciteit van een Schip al vermeld en is deze informatie overbodig. De kolom shipcountry heb ik niet nodig, omdat het alleen aangeeft waar de schepen vandaan komen, maar is niet relevant in deze context omdat de gegevens over transport operaties gaat tussen havens in europa.

Bij de ShipmentDetails heb ik de aantal containers per containerType van een Shipment toegevoegd en heb ik de begin en eindcontainerNr in aparte kolommen gezet.

In de item tabel heb ik de item\_description kolom verwijdert, omdat ik niet hoef te weten wat voor product er precies vervoert wordt.

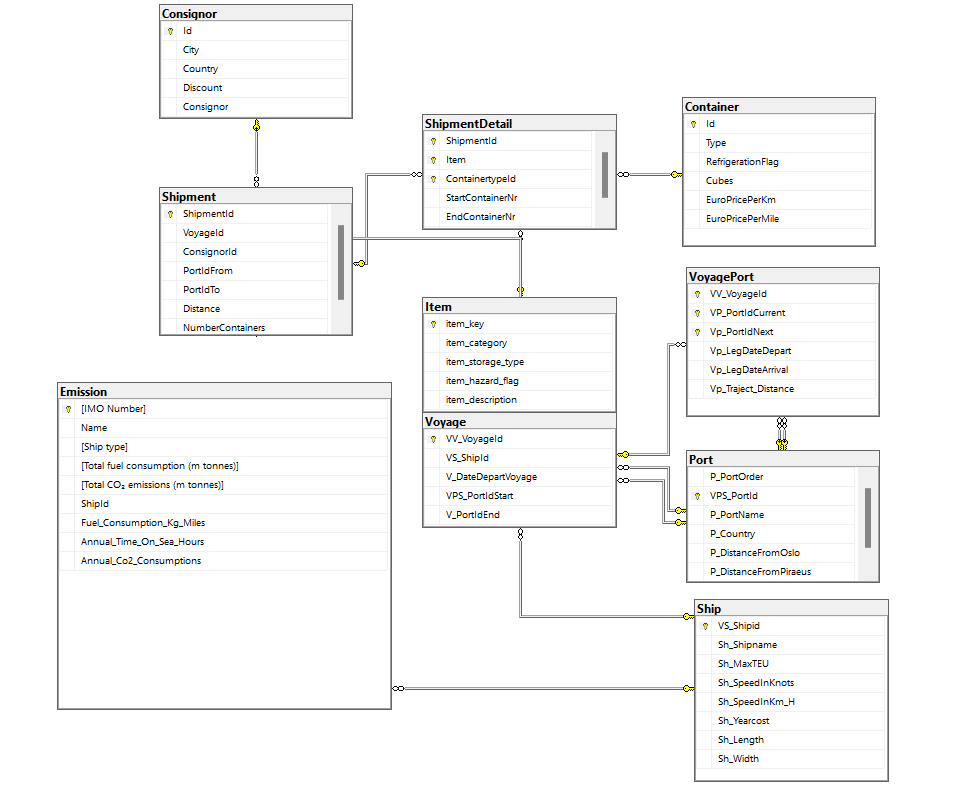
In de container tabel heb ik de Powerflag kolom verwijdert, omdat het de schipment niet beïnvloed. Een aparte kolom voor kosten per zeemijl toegevoegd en daarmee de kosten omgerekend naar zeemijlen. Daarnaast was de tabel nog niet goed genormaliseerd, doordat de containerNr begin en eind nog in 1 kolom zat. Hiervoor heb ik ze gescheiden te houden en daarvan de aantallen berekenen.

De consignor had ik destijds geen kolommen verwijdert.

De uiteindelijke resultaat binnen de RDBS is in figuur 2 weergeven.

Het resultaat voor de databases is hieronder weergeven.

Figuur 2. Genormaliseerde database met alle relaties en bewerkingen.



### Consistentie Problemen oplossen.

Van de raw database naar de relationele database, wordt de data opgeschoont. Hiermee worden inconsistente data consistent gemaakt.

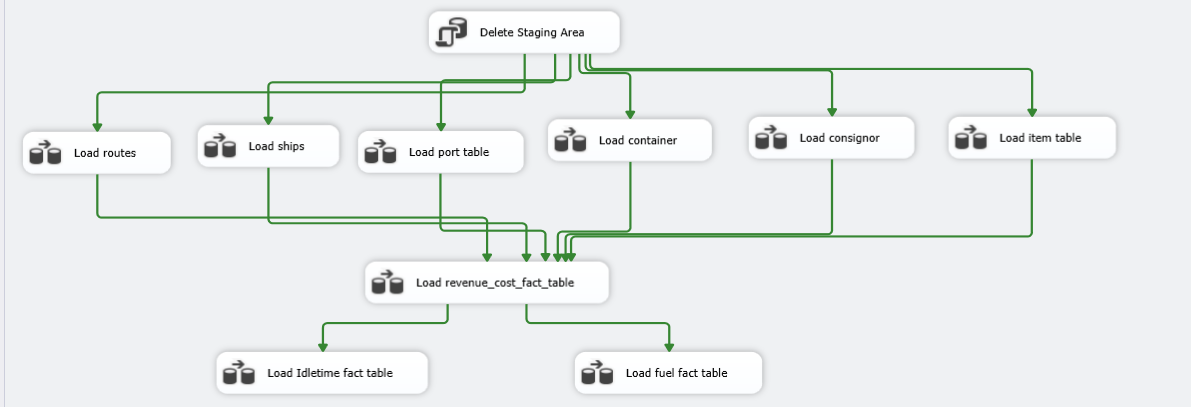
Voor consignor en port tabellen heb ik een aparte database gebruikt om conversies te maken, zodat de landen consistent van elkaar waren en omdat het makkelijker was dan met een landentabel te joinen.

## 

## 

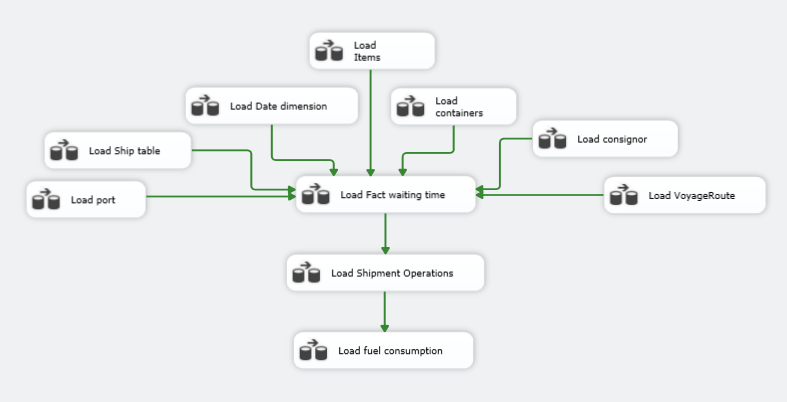
## 

## ETL Proces van RDBS -> PSA -> ODS



In deze fase wordt de data gemigreerd van de relationele database naar de staging database.

Sinds ik de staging database als een galaxy model heb ontworpen, dan voer ik bij het inladen van de feitentabel al berekeningen en aggregaties uit. Ik heb dat gedaan in deze fase, omdat ik de rol van deze fase meer fungeert om alle data goed geaggregeerd te hebben en het proces van de staging database naar de ODS alleen maar historie en nieuwe data zal toevoegen. Dus het proces van RDBS naar Staging Database en van Stagingdatabase naar ODS hebben afzonderlijke verantwoordelijkheden.



In deze fase wordt de data van de PSA naar ODS gemigreerd, als de data nog niet voorkomt in de ODS.

Bij het inladen van de feitentabel worden alleen data toegevoegd als de record nog niet voorkomt in de database. Omdat het historische data bijhoud en het niet wenselijk is om records die al gebeurt zijn nog een keer op te slaan. Hiermee voorkomen we duplicatie in de database.

In de dimensietabellen wordt een SDC type 2 toegepast, omdat ik de database niet drastisch hoef te veranderen.

# Ontwerp PSA en ODS.

## Type schema

Voor de staging area en ODS, zal het datawarehouse een Galaxy schema toegepast worden. Een belangrijke reden dat er een Galaxy schema is toegepast is omdat het voor mij makkelijker was om iteratief aan iedere informatiebehoeften te voldoen. Dus voor iedere behoeften werd in 1 of meerdere feitentabel gestopt.

In het schema zullen de dimensietabellen Schip, Port, Datum, Container, Consignor, Item en VoyageRoute voorkomen.

Een schip dimensie is nodig om te kunnen filteren per schip of type schip. Ook kunnen we van ieder schip de Maximale TEU ermee vergelijken met de Beladingsgraad feitentabel. Het is ook nodig om de informatiebehoeften voor wachttijden en beladingsgraad te kunnen aggregeren.

Een port dimensie is nodig om op haven en land te kunnen aggregeren. De port dimensie is namelijk nodig om de informatiebehoeften van wachttijden en Beladingsgraad te bevredigen.

Een datum dimensie is verwerkt in de datawarehouse, omdat er gevraagd wordt om per seizoen de brandstof verbruik te kunnen aggregeren. Verder is het ook handig om de trends van wachttijden te kunnen berekenen.

Een container, consignor en item dimensie tabel is toegevoegd in de database, omdat het van toepassing is bij het aggregeren van kosten. Ze zijn relevant voor de kosten feitentabel, omdat iedere dimensietabel de kosten kunnen beïnvloeden. Discount, inhoud en cost per mile om precies te zijn.

VoyageRoute dimensie tabel is toegevoegd, waarmee er gefilterd en geaggregeerd kan worden op vaarroute en subroutes. Een vaarroute heeft naast een begin en eindbestemming ook nog 1 of meerdere tussenstops tussen beide bestemmingen. Om iedere subroutes te laten aggregeren, heb ik een aparte dimensietabel toegevoegd.

Een item dimensie zal van toepassing zijn voor de shipment operation tabel en is nuttig om te weten welke items de meeste omzet en kosten maken.

Een consignor dimensie tabel kan nuttig zijn om per klant te kunnen filteren.

Er zullen naast dimensietabellen drie feitentabellen gemaakt worden. Een shipment operation feitentabel met de kosten, belading en brandstofverbruik.

De shipment operation tabel is een tabel die kosten brandstofverbruik en belading per consignor, containers, items en vaarroute laten zien. Dit is gemaakt om op zeer gedetailleerd niveau de data te bekijken.

Een andere feitentabel is de brandstofverbruik tabel en laat per vaarroute de brandstofverbruik en emissies zien. Het is minder gedetailleerd dan de shipment operation.

Een wachttijd tabel is bedoeld om de wachttijden in dagen te berekenen per haven en schip. Deze feitentabellen zijn gemaakt om zo aan de informatiebehoeften van Kramse te voldoen. Ieder van deze feitentabellen bevredigt 1 of 2 informatiebehoeften van de opdracht en zullen gebruikt worden om erin te analyseren.

## 

## Berekeningen en aggregaties tussen RDBS en PSA.

**DistanceMiles en Traject Distance**:

DistanceMiles = Distance \* 0.621371192

Vp\_Traject\_Distance = Distance \* 0.621371192

**Wachttijden:**

Binnen de access database, bestond er nog geen wachttijden tabel en had ik alleen een voyageport tabel om de wachttijden te berekenen. Echter beschreef de voyageport alleen de tijd tussen twee havens en niet de tijd die een schip in de haven was. Verder had niet iedere voyage een subroute en daarmee een wachttijd.

Daarom heb ik een join gedaan tussen voyagePortDeparture en voyagePortCurrent en haalde alleen gegevens op die dezelfde voyageId had.

Het resultaat was dat er een tabel was met daarin de datum waarin een schip aankwam en vertrok. Het verschil tussen de twee datums werd berekent en het resultaat was de wachttijd in dagen.

Wachttijd in dagen = Datum vertrek - datum aankomst.

**revenue:**

Om dit bedrag te kunnen berekenen, zal de distance in mijlen gebruikt te worden en dat te vermenigvuldigen met de Price\_Per\_mile.

Revenue = Price\_Per mile \* Distance

**DiscountPrice en Gross profit:**

Iedere consignor heeft een kortingspercentage. Dat wordt toegepast op de total cost per mile.

Discount Price = Total cost(mile) \* (Discount / 100)

Om het bedrag met korting te kunnen berekenen, maken we gebruik van de volgende berekeningen.

CostAfterDiscount = Total cost(mile) - Discount Price

**CostPerTonnage:**

Naast de korting en price per mile, is er ook de jaarlijkse schipkosten. Sinds de jaarlijkse schipkosten een vaste last is, zal de kortingspercentage niet toegepast worden op dit bedrag.

Om de CostPerTonnage te berekenen, zal de volgende berekening uitgevoerd worden:

CostPerTonnage = (**DiscountPrice**  + Sh\_**Shipcost**) / TEU\_Load / 10

**CurrentTEU:**

Om per vaarroute de beladinggraad per schip te kunnen berekenen, zal er eerst berekent moeten worden hoeveel TEU een container is.

1 TEU = 35,01 m3.

2 TEU = 70,02 m3

Sinds de containers in m3 zijn berekent, zal er eerst omgerekent worden naar TEU.

TEU = (Cubes / 35,01)

Vervolgens zal er voor iedere item de totale TEU belasting uitgerekent moeten worden.

Total\_TEU\_Load = Amount \* TEU.

Er wordt vervolgens een JOIN gedaan met de shipment tabel en zal er vervolgens gegroepeert worden op VoyageId, departurePortId en ArrivalPortId.

Het resultaat is de CurrentTEU per vaarroute en per schip.

**Beladingsgraad en beladingsfactor:**

Cargo\_Load\_Percentage = Total\_TEU\_Load / Sh\_Max\_TEU \* 100

**Cargo load factor:**

Cargo\_Load\_Factor = 1 + (TEU\_Load / Sh\_MaxTEU)

**Fuel\_Distance\_Miles\_kG:**

Fuel\_Distance\_Miles\_kG = Fuel\_Consumption\_Kg\_Miles \* DistanceMiles

**SpeedFactor**:

SpeedFactor = avg\_Voyage\_speed\_miles\_hours / Sh\_SpeedInKnots

**Fuel\_Consumption\_TotalVoyage**:

Fuel\_Consumption\_TotalVoyage = SpeedFactor \* Fuel\_Distance\_Miles\_kG

**Fuel\_Consumption\_PerTonnage\_Freight**:

Fuel\_Consumption\_PerTonnage\_Freight = Fuel\_Consumption\_TotalVoyage / (TEU\_Load \* 10) / DistanceMiles

Port\_lay\_Costs:

Port\_lay\_Costs = IdleTime \* 3.36 \* Sh\_Length

## Dimensietabellen

Voor de dimensie tabel Container, Items en Consignor heb ik de tabellen die in de genormaliseerde database voorkomen overgenomen als dimensietabellen. Ze beschrijven wat een container, item en consignor en zag niet de noodzaak om verdere aanpassingen te doen.

Voor dimensietabel datum heb ik de attributen Jaar, maand, dag, Seizoen, Kwartaal toegevoegd, omdat de datums op jaar-maand-dag niveau voorkwamen. Verder werd er gevraagd om op seizoen niveau geaggregeerd worden. Ook kon er misschien patronen herkent worden als er op seizoens niveau gefilterd kan worden. Een seizoen kan het weer beïnvloeden en daarmee ook de scheepvaart.

Tijd was niet aanwezig in de dataset en zag de noodzaak niet om op een gedetailleerde niveau kolommen te maken.

Voor de port dimensietabel heb ik land en stad gebruikt, omdat de andere kolommen niet nuttig waren en ze maar voor 1 context relevant waren. De distanceToOslo en distance Piraeus om precies te zijn, maar zijn niet relevant voor de afstand tussen andere havens.

Voor de Ship dimensietabel heb ik Shipname, Shiptype, brandstof en co2 emissies toegevoegd. Shipname om per schip te aggregeren. Per shiptype om per type schip te analyzeren.

Voor de Voyageroute dimensietabel heb ik de begin- en eind bestemmingshaven en de subroutes tussen de havens toegevoegd. Een voyage kan meerdere tussenstops hebben tussen de begin en eindbestemming. Daarom heb ik een voyageRoute dimensietabel toegevoegd.

### Slowy-changing dimensions

Op de datum dimensietabel na, wordt op alle dimensietabellen historisch bijgehouden. SCD Type 2 wordt hier toegepast.

Het voordeel hiervan is dat de structuur van de data warehouse niet drastisch veranderd hoeft te worden. De feiten tabellen hebben verwijzingen naar de rowIds van iedere dimensietabellen die een dimensie record vertegenwoordigt.

Daarnaast behoudt men historie bij van iedere dimensie record en weet men van oude feiten records welke historische dimensie records het bezit. Nieuwe feiten records krijgen de meest up to date dimensies.

Zo wordt er historie bewaard.

Iedere record in een dimensietabel heeft daarom een start en einddatum om aan te geven wanneer een dimensie gebruikt wordt en wat de huidige dimensie is. Als een dimensie nog niet gewijzigd is, dan is de einddatum NULL. Als een dimensie gewijzigd wordt, dan wordt een record toegevoegd met de gewijzigde gegevens en ontvangt de einddatum van de oude record een datum wanneer het niet gebruikt wordt.

De datum dimensie is niet van het SCD type 2, omdat deze tabel statisch is en er alleen spelfouten gecorrigeerd wordt. Er wordt ook alleen records toegevoegd en gelezen.

Daarom bezit deze tabel een SCD type 1.

## Feitentabellen.

Voor de wachttijden feitentabel heb ik de kolommen Wachttijd in dagen en id’s voor Ship, voyage, departureport, arrivalport, arrivaldate en departuredate toegevoegd. Volgens de opdracht wil men weten hoe lang een schip in een haven wacht voordat het vertrekt. Die tijd wordt in dagen weergeven en is belangrijk om inzicht te krijgen over hun wachttijden. Sinds ze het per schip en haven willen analyseren, heb tussen ship en arrival/departureport een foreign key relatie gelegd tussen de schip en port dimensietabellen. Verder zal de aanmeerkosten per schip weergeven worden.

De feitentabel Shipment Operations bevatten gegevens over de omzet, kosten, brandstofverbruik en beladingsgraad. Deze gegevens heb ik in 1 feitentabel verzameld, omdat al deze feiten gerelateerd zijn. Iedere voyage maakt een schip opbrengsten en kosten. Ook verbruikt het brandstof en vervoert goederen van de ene haven naar de andere.

Onder de omzet en kosten gegevens, laten de kolommen Revenue, Discount, Discount Price, Gross profit, Price per mile, Cost per tonnage de financiële gegevens zien.

Sinds iedere voyage niet even lang is, zal sommige bedragen omgerekend worden naar een enkel eenheid. Voor kosten, zal er per vracht berekend worden hoeveel kosten er gemaakt wordt, waarmee er besloten kan worden of er meer of minder vracht meegevoerd moet worden. Of dat er minder korting gegeven moet worden.

Onder de brandstofverbruik gegevens, worden de kolommen Total fuel consumption kg en Fuel tonnage per mile gebruikt om deze gegevens te ondersteunen. Deze gegevens zijn belangrijk, omdat Kramse zijn emissies wil verlagen en daarvoor heeft het data nodig over zijn brandstofverbruik. Sinds iedere operatie verschillende afstanden en reistijden heeft, zal de brandstofverbruik omgerekend worden zodat er goed vergeleken kan worden met andere records.

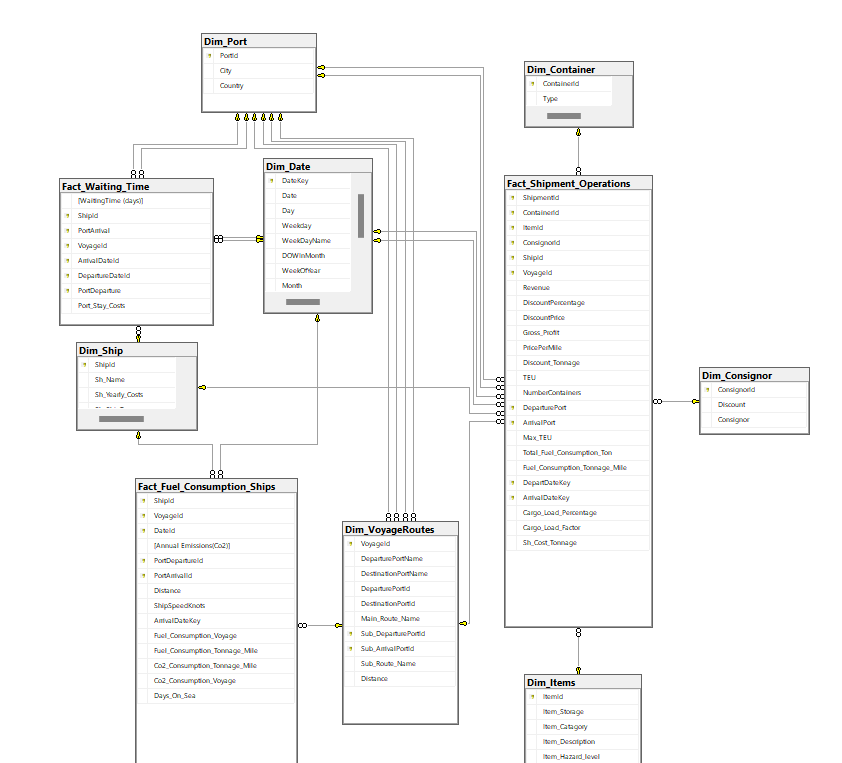
Voor de beladingsgraad wordt de TEU Load, Max TEU, Cargo Load Percentage en Cargo Load Factor gebruikt als kolommen. TEU Load is de daadwerkelijke lading die een schip heeft en de Max TEU is de maximum capaciteit die een schip qua vracht kan meevoeren. De beladingsgraad wordt als percentage weergeven in Cargo\_Load\_Percentage. Het doel van deze gegevens is om te laten hoe gevuld een haven of schip is en of er meer of minder vracht meegeleverd moet worden.

Dit heeft namelijk mogelijk invloed op de brandstofverbruik en kosten/baten.

De laatste feitentabel is de brandstofverbruik feitentabel. Die is bedoeld om per vaarroute de brandstofverbruik in kilo’s te laten zien. Diezelfde gegevens komen ook voor in de shipment operations tabel, maar bevatten duplicatie gegevens van afstanden en brandstofverbruik. Dat was handig voor het analyseren van kosten per consignor, containers, items en schip. Maar niet voor de afstanden van vaarroutes. Dit kan leiden tot niet representatieve aggregaties. Daarom bestaat deze feitentabel. Verder de dagen op zee weergeven van ieder vaarroute en schip. Maar niet de kosten per vracht ton. Dat zal weergeven worden in de Shipment-operationtabel.

In figuur 3 is een diagram van de PSA weergeven.

Figuur 3. Galaxy schema PSA.



Figuur 4. ODS Kramse diagram

# 

# 

# Dashboard

## Vormgeving

Ik heb gekozen om 8 verschillende dashboards te maken. 4 dashboards om de algemene gegevens voor wachttijden, kosten, brandstofverbruik en beladingsgraad. Hierin zullen grafieken en tabellen weergeven worden over hun onderwerp.

Deze 4 dashboards worden ondersteund met detail-dashboards. De reden dat ik voor deze aanpak heb gekozen is om te laten zien hoe de data tot stand is gekomen. Met detail dashboards kan ik per dimensie laten hoe de data in de vier dashboards in elkaar zit.

Zo kunnen we op gedetailleerd niveau verbanden te zoeken en daarmee betere beslissingen gemaakt kunnen worden.

Voor trends maak ik gebruik van lijn of staafgrafieken. Om erachter te komen welk aandeel een element heeft, dan wordt een treemap of cirkelgrafiek toegepast.

In het geval dat er met plaatsen gewerkt wordt maak ik gebruik van een map-grafiek, om te laten zien op welke plaatsen de meeste dingen heeft van iets.

## Conclusies dashboard

Een aantal conclusies kunnen getrokken worden.

* Grote schepen verbruiken per vracht ton per zeemijl meer brandstof.
* Grote schepen worden onvoldoende beladen.
* De kleinste schip is overbeladen.
* De wachttijden zijn best in de haven van Bilbao
* De kosten stijgen bij korte reizen.
* Sommige containers bevatten goederen die niet in de juiste containers thuishoren
* In de herfst verbruiken schepen meer brandstof.
* De kosten zijn groter op grotere schepen.
* De omzet per schip stijgt als er meer korting wordt gegeven

## Mogelijke beslissingen

Een aantal beslissingen kunnen getrokken worden op basis van onze conclusies.

* Schepen kunnen langzamer varen om brandstofverbruik te verminderen.
* Schepen kunnen hoger beladen worden.
* De reizen kunnen korter gemaakt worden om kosten te besparen.
* Minder reizen maken naar de haven van Bilbao
* De items in de juiste containers gestopt worden.
* Goederen kunnen in wat meer in kleinere schepen geladen worden.
* Minder vaak in de herfst van het jaar goederen vervoeren.
* Meer korting geven.

# Literatuur:

De database scripts zullen meegeleverd worden met de projecten.

Literatuur

[Marine Heavy Fuel Oil (HFO) For Ships - Properties, Challenges and Treatment Methods (marineinsight.com)](https://www.marineinsight.com/tech/marine-heavy-fuel-oil-hfo-for-ships-properties-challenges-and-treatment-methods/)

[Weight units energy (cbs.nl)](https://www.cbs.nl/en-gb/onze-diensten/methods/definitions/weight-units-energy#:~:text=%2D%20Gas%2C%20diesel%2C%20light%20fuel,%3A%201%20litre%20%3D%200.88%20kilogram.)

[Bulk carrier - Wikipedia](https://en.wikipedia.org/wiki/Bulk_carrier#:~:text=The%20crew%20on%20a%20bulk,department%2C%20and%20the%20steward's%20department.)

[(5) How much does it cost to fuel a cargo ship? - Quora](https://www.quora.com/How-much-does-it-cost-to-fuel-a-cargo-ship)

[Tarieven voor boeien, palen en openbaar kadegeld | Port of Rotterdam](https://www.portofrotterdam.com/nl/zeevaart/boeien-en-palen/tarieven-voor-boeien-palen-en-openbare-kadegeld)

<https://www.bunkerindex.com/news/article.php?article_id=18387>

[Job Search - Millions of Jobs Hiring Near You | ZipRecruiter](https://www.ziprecruiter.co.uk/?utm_source=zr-go-redirect)

[Vrachtschepen (docukit.nl)](https://www.docukit.nl/spreekbeurt/vrachtschepen#:~:text=Het%20duurt%20wel%20twee%20dagen,lang%20onderweg%20met%20hun%20lading.)

[Flexport Help Center Article | When Will My Shipment Be Ready to be Picked Up at the Destination Port?](https://www.flexport.com/help/589-pick-up-container-from-destination/)

[Flexport Help Center Article | When Will My Shipment Be Ready to be Picked Up at the Destination Port?](https://www.flexport.com/help/589-pick-up-container-from-destination/#:~:text=Shipments%20will%20not%20be%20ready,than%2010%2C000%20containers%20on%20board.)