**Project Kramse**

A picture containing boat, sky, ship, water

Description automatically generated

Diren Öztürk, 2158837

Inhoud

[1. Opstart 3](#_Toc98083256)

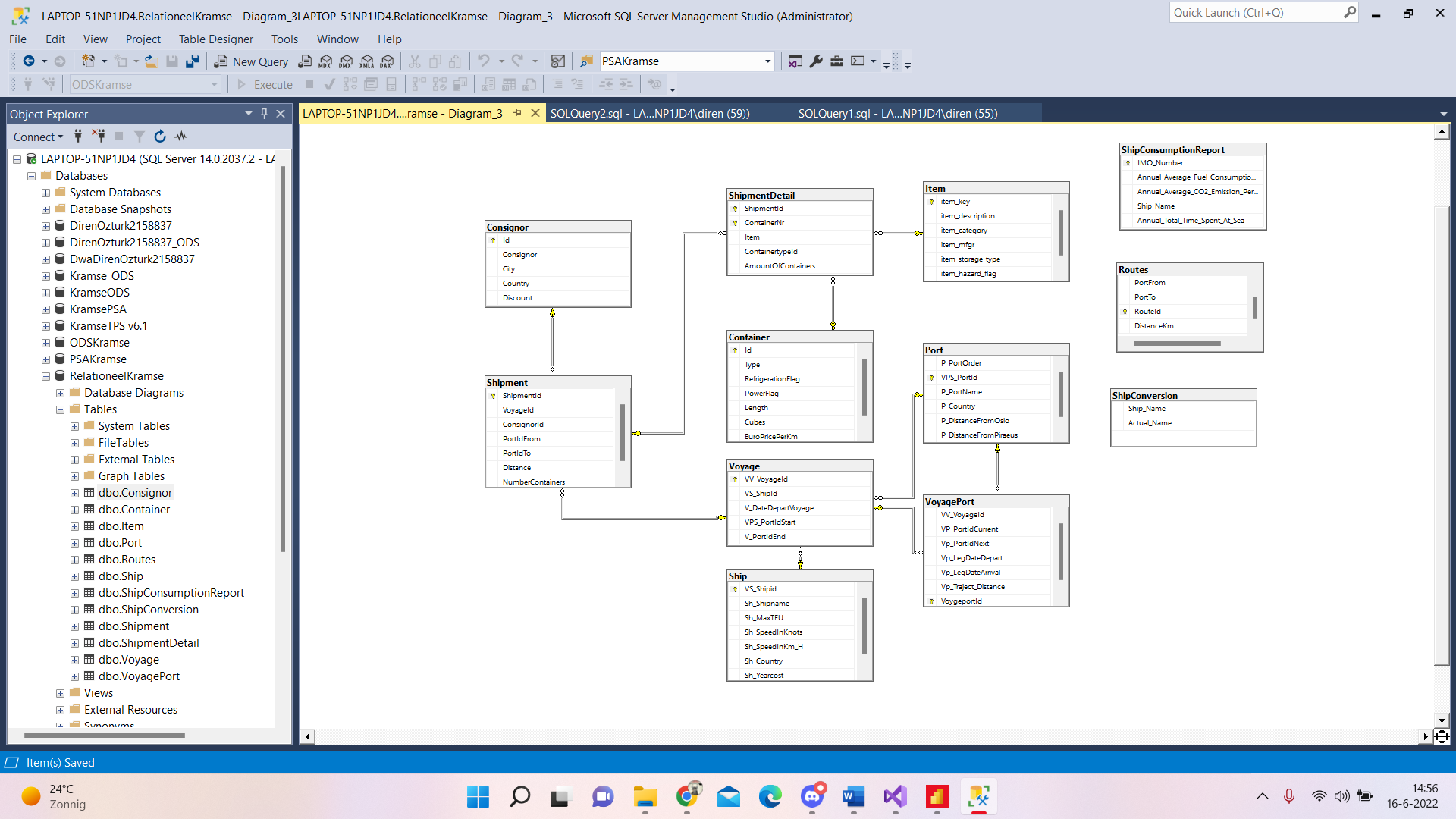
[1.1 SQL Databases 3](#_Toc98083257)

[1.2 Connecties en Control Flow 4](#_Toc98083258)

# Relationeel Model

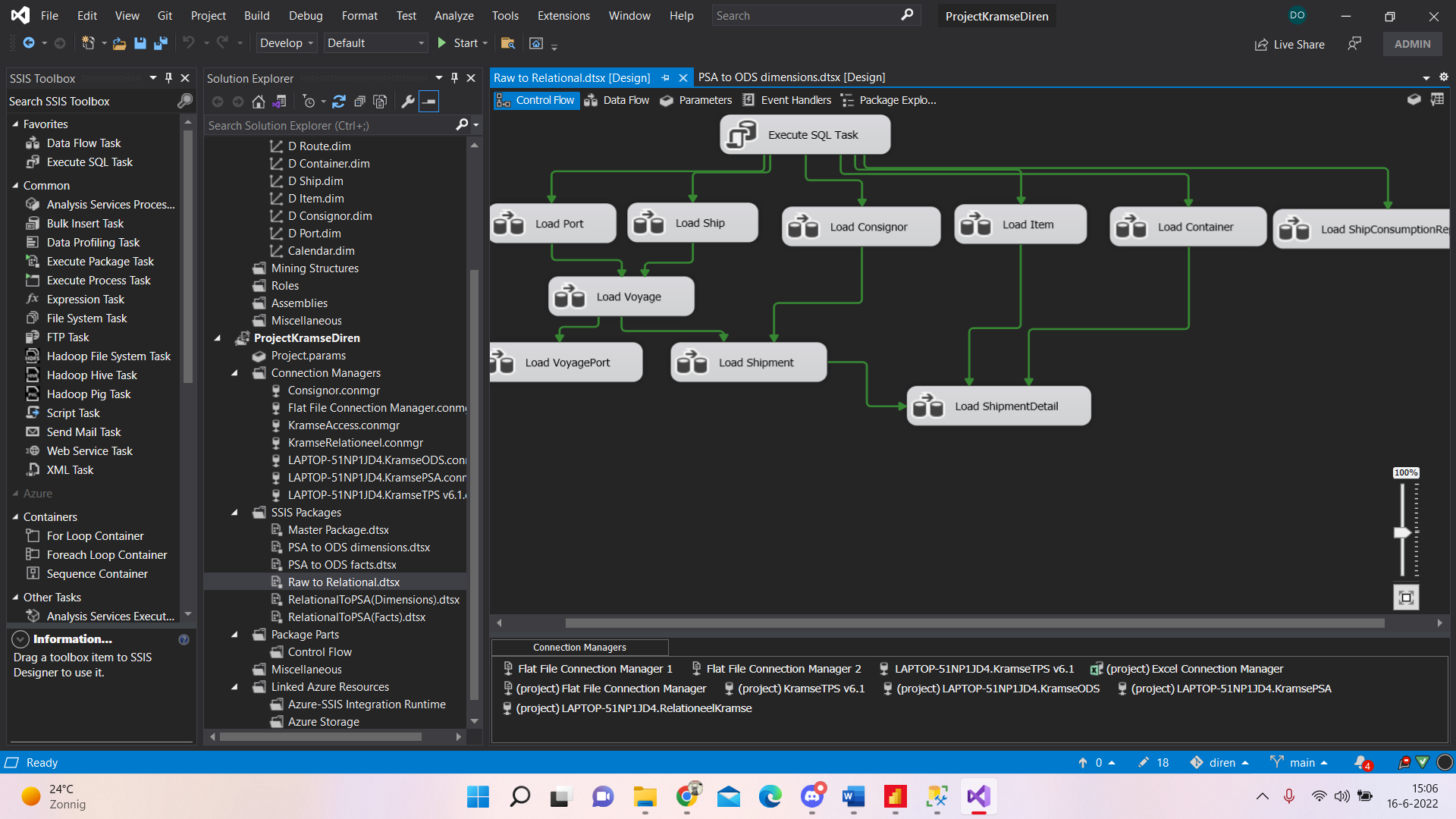
## 1.1 SQL Database Relationeel

Ik ben begonnen met het bestuderen van alle bronnen die ons gegeven werden. Consignor, Container, KramseTPS, en de Publication of Information. Deze heb ik allemaal onder een ruwe SQL database geïmporteerd. Vanuit de Access Database (KramseTPS), heb ik gekeken naar de relaties die er gelegd waren. Deze had ik overgenomen vanuit de ruwe database, en was daarbij begonnen relaties te leggen, zoals aangegeven in de Access database. Daarbij keek ik ook naar de drie aparte tabellen van Consignor, Container, en Publication of Information. De Publication of Information heb ik besloten tijdelijk achterwege te leggen, omdat het nog niet relevant was voor de beschikbare data. Dit staat in de ShipConsumptionReport, maar heeft dus geen relaties. Ik ben verder gegaan door connecties te leggen voor Consignor en Container. De ID van Consignor heb ik verbonden met de ConsignorId van Shipment. De ID van Container heb ik verbonden met de ContainertypeId van ShipmentDetail. Hiervoor heb ik bij Container de Length, Cubes en EuroPricePerKm verandert naar een Decimal(18,2), en de ID naar een integer. Ook heb ik een Tabel aangemaakt voor Routes wat ik ga gebruiken om alle beschikbare routes in op te slaan. Bijvoorbeeld Bilbao-Barcelona. Als laatst heb ik een shipConversion om de namen van schepen consistent te maken. Xin Being wordt dus Xin Beijing en Mærsk Stepnica wordt Maersk Stepnica.



## 1.2 Raw To Relational

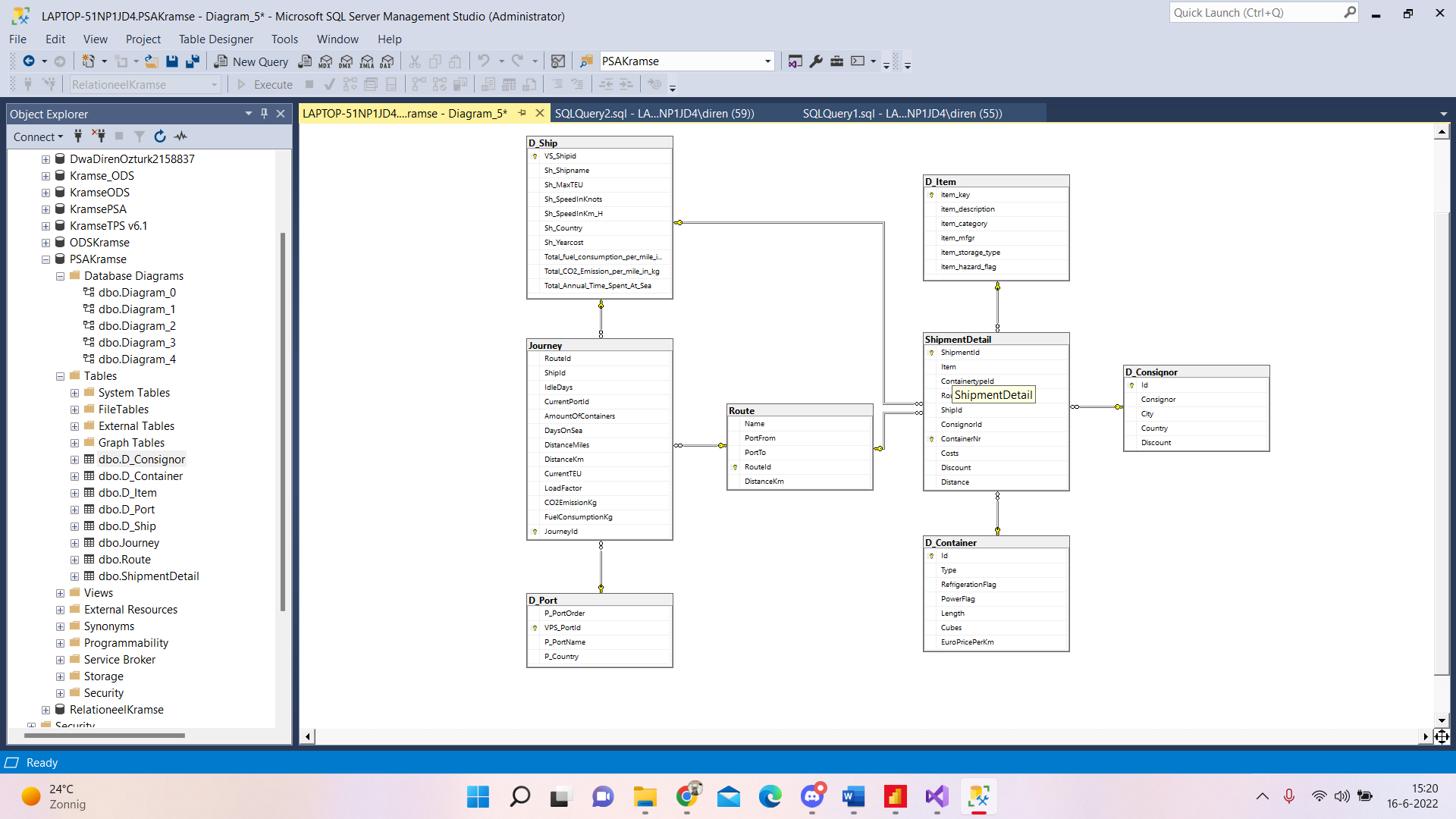
Nadat de relaties gelegd waren, heb ik via Visual Studio connecties aangelegd met de data bestanden en de nieuwe database. Voor Container, wat een txt bestand was, heb ik een connectie gelegd door middel van een Flat File connectie. Voor Consignor heb ik een Excel connectie gebruikt, en voor de Access Database een ADO.NET connectie. Voor ShipConsumptionReport heb ik de Publication of Information sheet gebruikt. Nadat de connecties gemaakt waren heb ik een SSIS Package aangemaakt. Mijn Control Flow begint met een SQL Script om alle waarden te verwijderen. Nadat de SQL Script is uitgevoerd, worden de parent tables geladen via de bestanden. Ik begin met het laden van Port, Ship, Consignor, Item, Container en ShipConsumptionReport. Als dat geladen is wordt Voyage geladen. Hierna worden VoyagePort en Shipment geladen, en als laatst ShipmentDetail.



# PSA

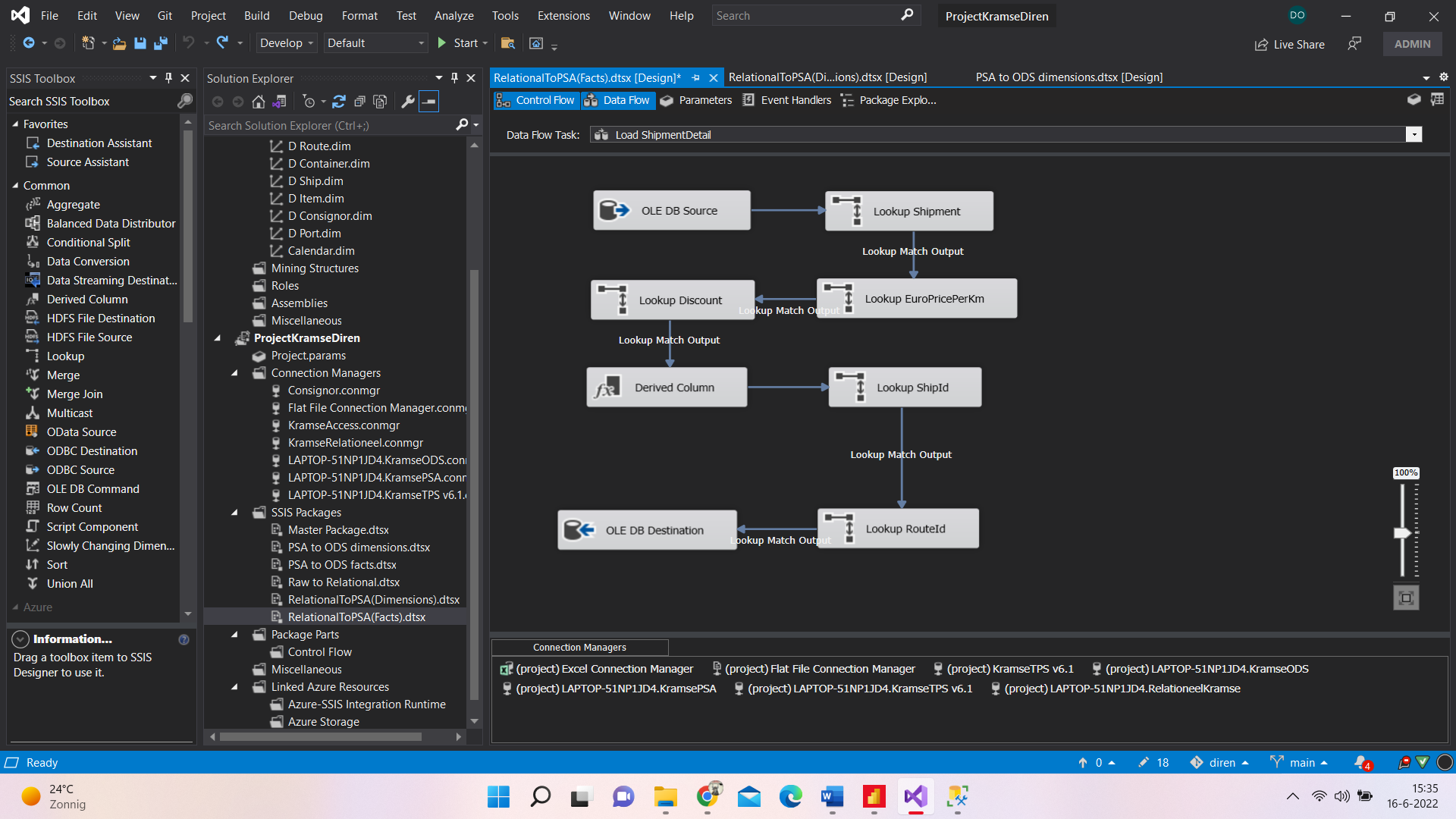
## 2.1 SQL Database PSA

In mijn PSA heb ik ervoor gekozen om twee Feiten te maken. Dit zijn ShipmentDetail en Journey. Journey is alles wat bij een reis hoort. Dit heb ik samengevoegd met de values uit ShipConsumptionReport. En is daarbij gelinkt aan Ship, Port en Route. Ship en Route werken beiden tussen de feiten in, omdat ze context geven over beide feiten. Zo kun je bijvoorbeeld afstand per route opvragen maar ook de co2 emissie per route. Om ShipmentDetail heb ik de dimensies gelaten als ze waren. Item Consignor en Container hadden geen veranderingen nodig. Ik heb wel de discount van consignor toegevoegd aan de ShipmentDetail.



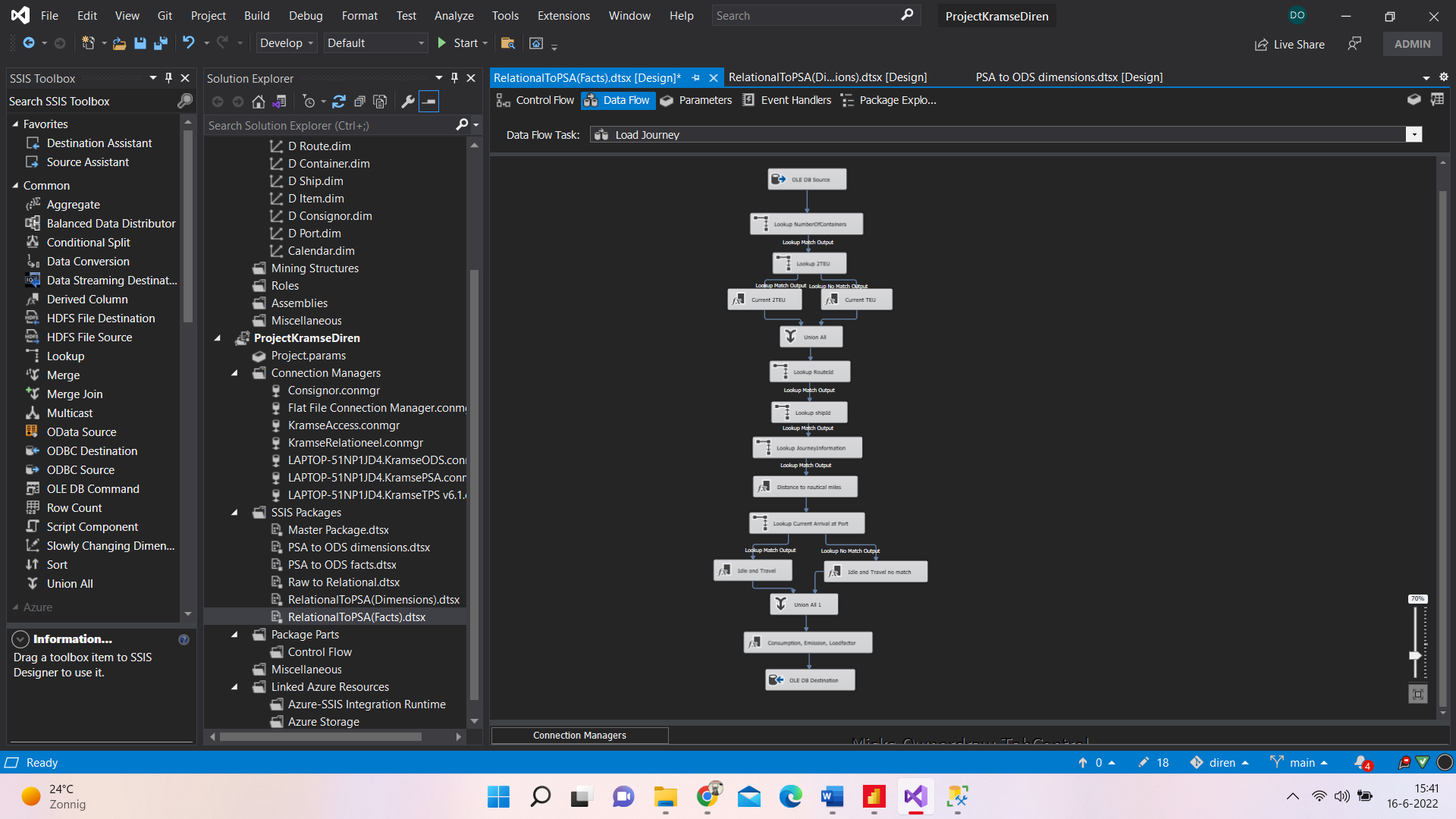
## Relational To PSA

### ShipmentDetail



ShipmentDetail van de relationele database is mijn source. Hierop doe ik een lookup om Shipment informatie uit te halen. Ook haal ik waardes uit de Container en Consignor. Dit zijn de EuroPricePerKm en Discount. Dit komt omdat het waardes zijn die een context nodig hebben, dus in een feit thuis horen. Hierna worden de ShipId en RouteId opgehaald om ze uiteindelijk in mijn feit te zetten.

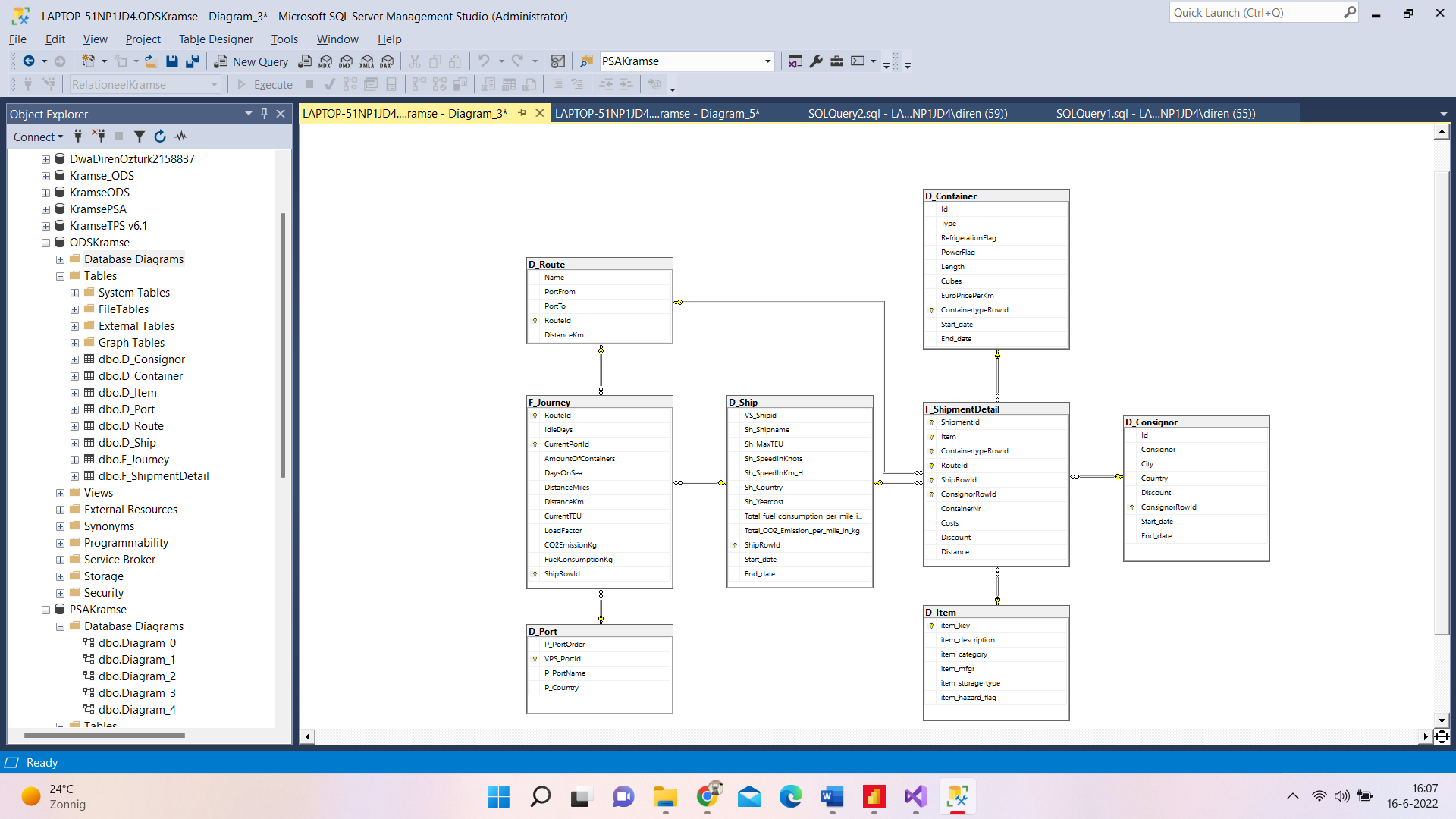
### Journey

Als basis voor mijn Journey heb ik VoyagePort gebruikt. Hierbij heb ik in mijn lookup een query gemaakt om het aantal containers op te vragen, omdat deze gegeven in mijn Journey stond. Ook heb ik een lookup met een query om de TEU op te halen. Na deze samen te voegen met een union all, zoek ik naar routeId en ShipId om in mijn Journey te zetten als een foreign key. De JourneyInfo wordt opgehaald vanuit de Ship tabel en hiervoor heb ik de gegevens MaxTeu, SpeedInKM/H, Total Fuel, Annual time at sea en Total emission opgevraagd. Daarna zet ik de kilometers om in zeemijlen. Om de IdleTime en Travel time doe ik eerst een lookup op een andere voyageport. Mijn CurrentPortId moet daarbij gelijk zijn aan de NextPortId van de andere voyageport. Ook moeten de voyageIds overeenkomen. Dan haal je dan de arrivaltime van de schip op je currentPort op. Dat gebruik ik dan om IdleTime te berekenen bij een match output. Als er geen match is, betekent het dat de reis daar start, dus heb je geen idletime. Dan is de value 0. Als laatst reken ik de consumption en emission uit door eerst het totale te delen door eerst de afstand te berekenen door Annual time at sea te vermenigvuldigen met de snelheid van de schepen. Dan heb je de totale afstand afgelegd op zee. Deze gebruik ik door de total te delen zodat je de emissie en verbruik per mijl hebt. Ook reken ik de beladingsgraad uit door de currentTEU te delen door maxTEU.

# ODS

## 3.1 SQL Database ODS

Ik heb besloten om alleen Consignor, Container en Schip historisch te maken. Dit komt omdat de data van een Item, Port, en Route niet zo snel veranderen. Verder heb ik een unique key constraint gezet op mijn feiten om het zo uniek mogelijk te houden. Voor mijn Journey heb ik daarvoor mijn routeId, CurrentPortId en ShipRowId gebruikt om samen een primary key te vormen. Voor shipmentDetail gebruik ik ShipmentId, Item(Id), ContainerTypeRowId, RouteId, ShipRowId en consignorRowId als primary key, en is er een unique key aanwezig.



## 3.2 PSA to ODS

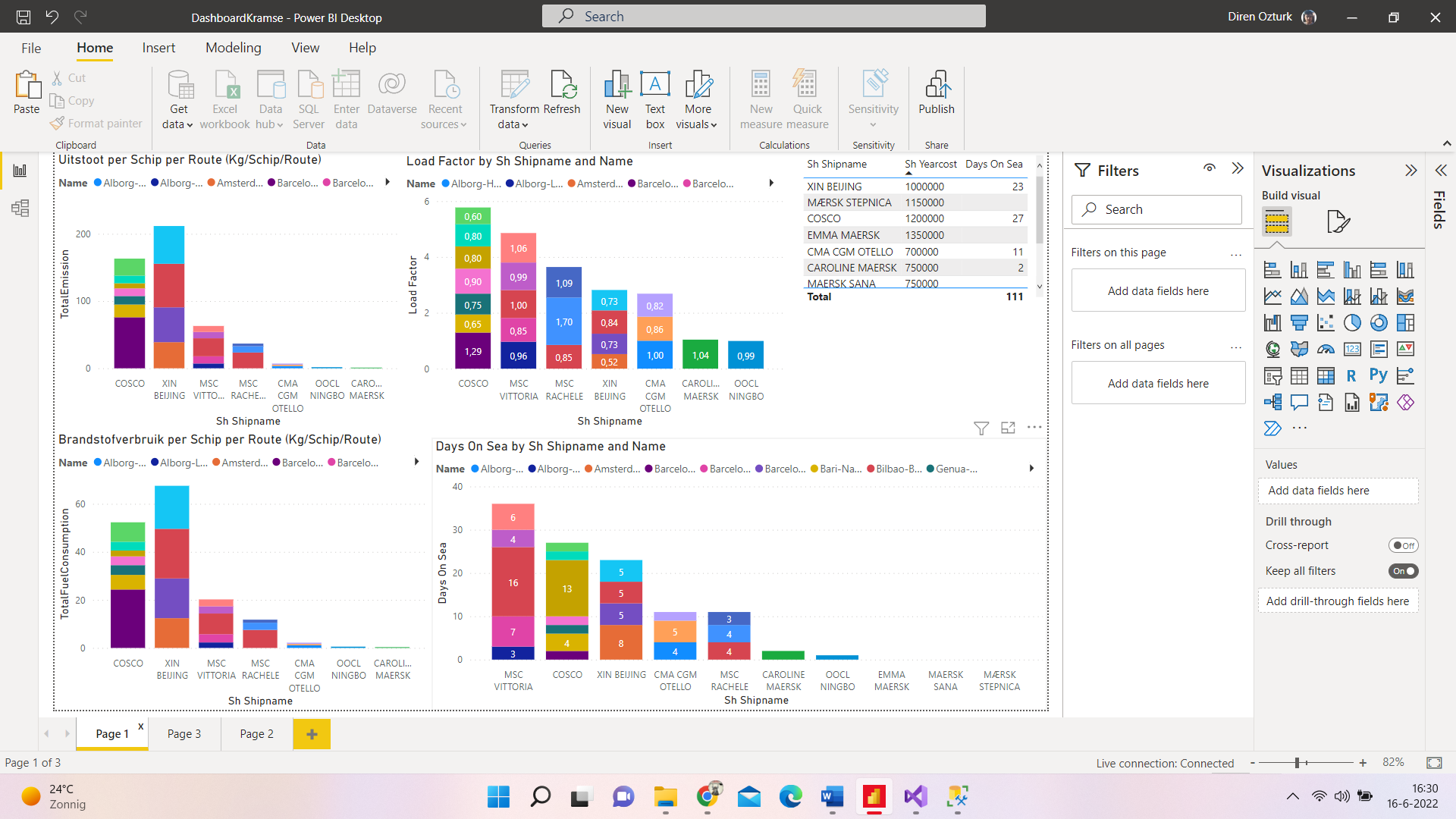
Ik heb dus gekozen om Container, Consignor, en schip historisch te maken. Voor mijn Facts doe ik eerst een lookup op de benodigde Ids. Voor mijn shipmentDetail zijn dit de ShipRowId, ConsignorRowId, en ContainerRowId. Dit naar een lookup dat ervoor zorgt dat al mijn values uniek zijn, en dat alleen de veranderingen worden doorgelaten. Dit doet hij door te kijken naar de ConsignorRowId, ContainerRowId, ShipRowId, ShipmentId, ContainerNr en RouteId.

Voor mijn Journey had ik alleen een lookup nodig om de ShipRowId op te vragen. Daarna is er een lookup gedaan om mijn ODS uniek te houden door te kijken naar mijn RouteId, CurrentPortId en ShipRowId.

## Dashboards

## 4.1 Uitstoot en Brandstofverbruik

Mijn eerste dashboard geeft weer wat de uitstoot en het brandstofverbruik is. Wat hieruit blijkt is dat Cosco en Xin Beijing het hoogste verbruik en emissie hebben. Hierbij heb ik een grafiek geplaatst m weer te geven dat de uitstoot niet aan het aantal dagen op zee ligt. Dit kun je ophalen uit MSC Vittoria die het langst van allemaal op zee was, maar toch een heel stuk lager ligt in uitstoot en brandstofverbruik. Dit betekent dat MSC Vittoria een zuinige schip is, ten opzichte van Cosci en Xin Beijing. De beladingsgraad is ook te zien per schip per route. Daarbij kun je ook rekening houden met hoe hoog de schepen waren geladen, en zie je ook dat er een relatie is tussen uitstoot en beladingsgraad. Ook is er een tabel om te laten zien hoelang elk schip op zee vaart, en de kosten die komen kijken bij de schepen. Je merkt hieruit dat er drie schepen zijn die niet varen, maar wel jaarlijkse kosten hebben. Door deze te verkopen zouden ze jaarlijks geld kunnen besparen.



## 4.2 Idle time en Ladingsgraad

Uit mijn tweede dashboard kun je hoelang schepen idle waren bij havens en wat de beladingsgraad was van de schepen bij de havens. Hiervoor heb ik een map gebruikt om de idle time en beladingsgraad per schip weer te geven. Ook heb ik diagrammen waaruit je ook per schip kan zien wat de idle time was per haven Wat je als eerst merkt is dat Bilbao en Barcelona een grote rol spelt bij de idle time. De schepen die daar komen zitten vaak vast aan grote idle times, en moeten soms 5 dagen wachten. Als de idle time omlaag gebracht wilt worden, zou men moeten kijken naar wat er aan de hand is in Spanje. Verder kun je nogmaals zien wat de beladingsgraden zijn van van schepen.

