

Migratieplan

Groep 3c

Teo Curcic

Joshua van Ee

Cas Ros

Stan Boeter

# Kwaliteit van de gegevens

Als we kijken naar de verschillende databronnen van de twee organisaties dan zien we dat er veel gegevens erin verwerkt staan. Het is dus ook onmogelijk om te voorkomen dat er geen fouten in zitten. Om een succesvolle migratie uitvoeren, proberen we eerst alle fouten uit de oude databronnen te halen, zodat die niet terug te zien zijn in de nieuwe database van Outdoor Paradise.

Door de databronnen grondig te onderzoeken, zijn er verschillende fouten geconstateerd:

* Inconsistente naamgeving: De verschillende attributen in de databasen zijn zo geformuleerd dat ze niet logisch benoemd zijn en is het niet duidelijk wat de attributen betekenen.
* Veel dubbele waardes: In de databasen zitten zoveel waardes verwerkt dat er ook veel dubbele waarden in zitten. Voor het onderhoud van de databasen is dit zeer onoverzichtelijk.
* Foute gebruik van opslagtypen: De meeste attributen zijn als een varchar() opgeslagen, terwijl het alleen getallen zijn of een karakter is. Dat wil je niet in je database hebben omdat je dan onjuist gebruik maakt van de verschillende datatypen.
* Geen relatie tussen verschillende tabellen: Er zijn tabellen die los van elkaar staan en met geen enkele andere tabel een relatie hebben. Het is niet de bedoeling dat er tabellen los van elkaar komen te staan.
* Te uitgebreide tabellen: Een tabel met te veel attributen en te veel waardes is niet overzichtelijk om te onderhouden. De tabellen moeten zo compact mogelijk gemaakt worden en op een logische manier verdeelt zijn.
* Verouderde data: Er is veel data dat verouderd is en niet meer gebruikt gaat worden door de organisatie. Je wilt zoveel mogelijk up-to-date data in je databasen hebben zodat de database optimaal te gebruiken is.

Door alle fouten te onderzoeken, gaan we ervoor zorgen dat er consistente naamgevingen gebruikt gaat worden, dat alle dubbele waarden uit de database komt en dat alle tabellen juist samengevoegd wordt, zodat ze allemaal een relatie krijgen.

# ETL Tools

## SQL Server Intergration Services

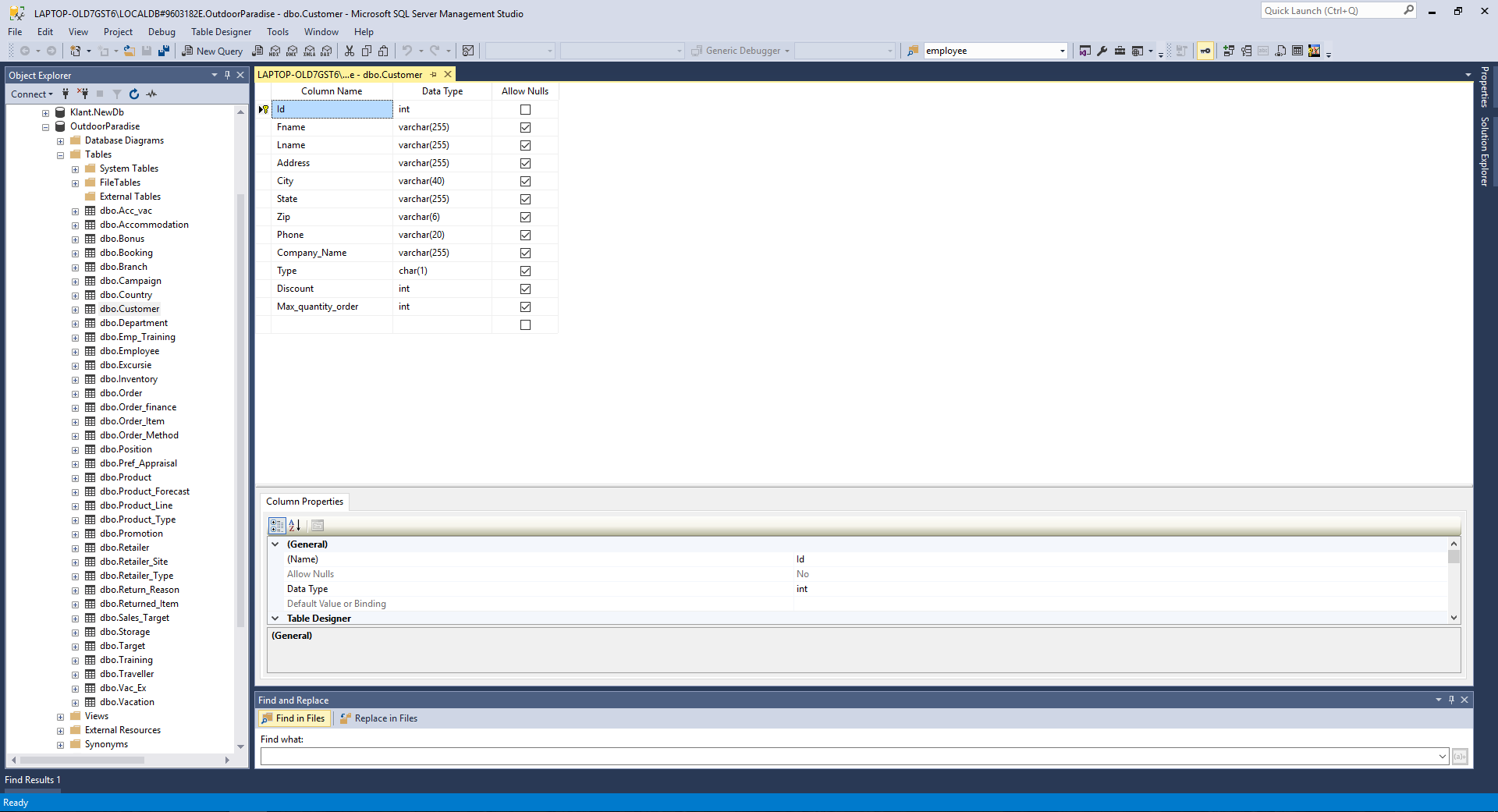
Homepagina: <https://docs.microsoft.com/en-us/sql/integration-services/sql-server-integration-services>

Waarom gekozen om te onderzoeken?

De tool is van Microsoft, en omdat wij al met visual studio werken vinden wij dit erg makkelijk.

Impressie:

SSIS is enorm goed gedocumenteerd. Microsoft zelf heeft een erg gedetailleerde documentatie pagina, de tool heeft zo veel gebruikers dat het internet vol staat met vragen en antwoorden. De tool is erg geavanceerd en heeft een redelijk bekend uiterlijk door Visual Studio. Ook wordt er op de website genoemd dat deze tool ondersteuning heeft voor het updaten van data warehouses, wat wij later nodig zullen hebben. Verder is er geen account nodig voor het gebruik van SSIS, wat ook een pluspunt is. SSIS wordt ook als de snelste commercieel beschikbare ETL tool gezien (Bron: <https://technet.microsoft.com/en-us/library/dd537533(v=sql.100).aspx>)



Conclusie:

SSIS is onze eerste keuze voor ETL tool die wij gaan gebruiken voor het project. Door de grote hoeveelheid documentatie, en het professionele uiterlijk lijkt het ons een fijn programma om mee te werken.

## CloverETL

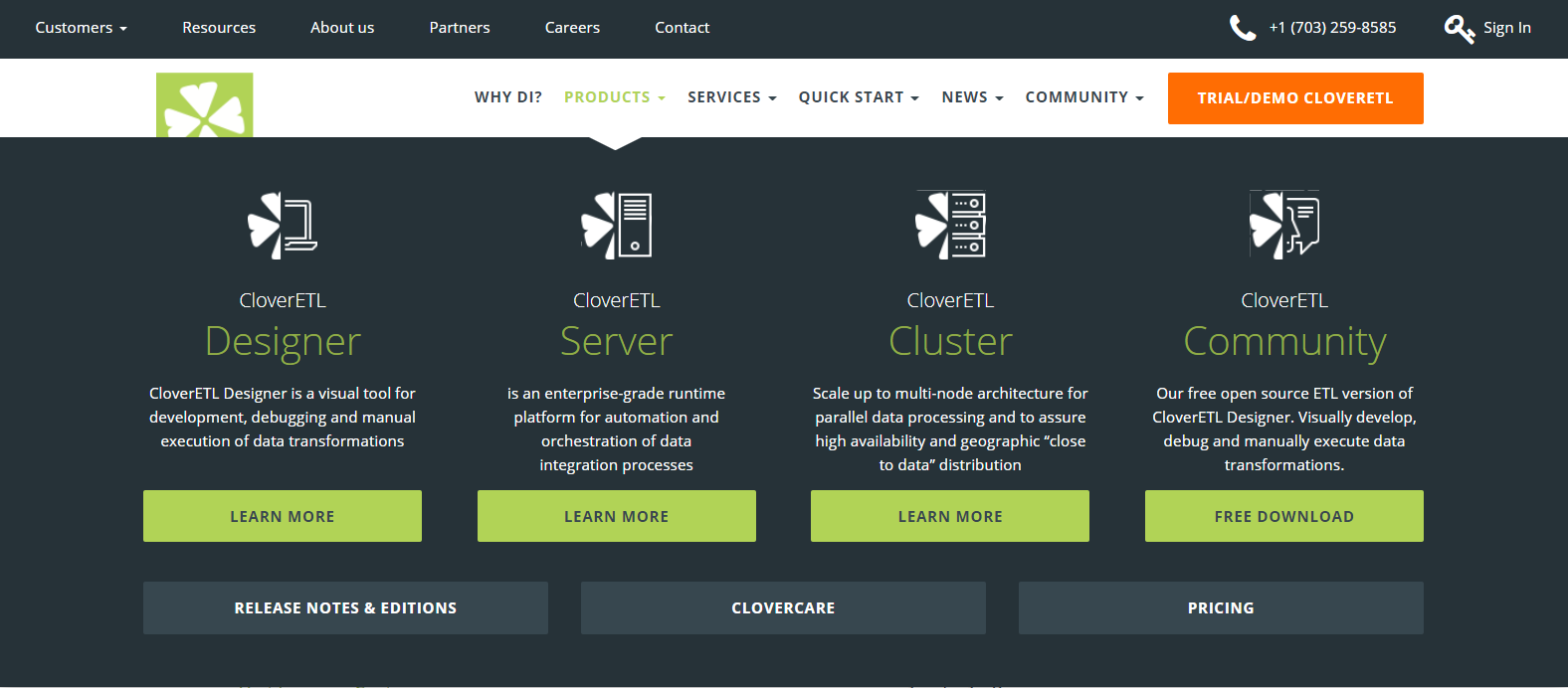
Homepagina: <https://www.cloveretl.com/>

Waarom gekozen om te onderzoeken?

CloverETL staat hoog beoordeeld als ETL tool. De software komt vaak voor in vergelijkingen tussen ETL tools en komt er zeker niet slecht uit.

Impressie:

De eerste impressie van de website is redelijk goed. De site ziet er voor het grootste deel professioneel uit. Alleen van de genoemde “leading companies” komen ons er maar twee bekend voor. Ook de footer ziet er niet goed uit, en de kleine lettertjes helemaal onder aan de pagina geven nog meer slechte impressies. De FAQ is enorm uitgebreid en ze hebben zelfs een erg gedetailleerd gebruikers forum. Ook biedt CloverETL een quickstart guide bestaande uit meerdere filmpjes. Het pakket zelf is enorm duur, 5000 euro per license, maar biedt wel een demo.



Conclusie:

De tool ziet er geweldig uit, maar wij kiezen toch liever voor SSIS omdat deze van microsoft is. Ook is de tool erg duur, al biedt het wel een demo, maar SSIS is gratis. Ook willen wij liever niet met de limitaties van de demo werken.

## Apatar

Homepagina: <http://apatar.com/>

Waarom gekozen om te onderzoeken?

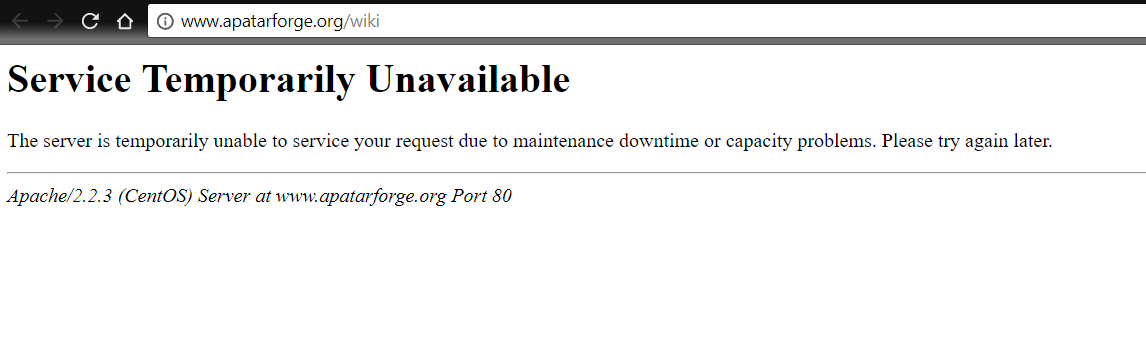
Deze ETL tool stond in de lijst van bekende en goede tools.

Impressie:

Toen wij de documentatie van deze ETL tool wouden doornemen, laadde de pagina niet. Dit is al een slechte eerste impressie. Verder bevat de FAQ alleen maar vragen die in het voordeel vallen voor Apatar. Het komt niet over alsof de vragen in deze FAQ ook echt gevraagd worden door gebruikers, onder door de vraag “Can Apatar server run in a multi-threaded fashion optimizing performance levels?”.

Ook de web demo op de site van Apatar werkt niet. Dit werkt zeker niet in het voordeel voor Apatar.

Voor het gebruik van de tool is een account nodig. Tegen de tijd dat de download klaar was, hadden wij al vele slechte impressies van Apatar. Omdat we liever geen account wouden aanmaken, maar alleen de software even wouden doorkijken, heeft dit de knoop doorgehakt voor het besluit deze software niet te gebruiken. Normaal is een account geen groot probleem, maar omdat de site al zo slecht was denken wij niet dat deze software het waard is.



Conclusie:

Dit wordt niet de ETL tool die wij gaan gebruiken voor ons project. De documentatie werkt niet, de FAQ geeft geen goede impressies, en zelfs de web demo werkt niet. Verder is er een account nodig voor gebruik en ook dit vinden wij niet erg fijn.

# Mapping

## Order



## Training (cursus)



## Retailer\_Site



## Retailer



## Department & Branch



## Customer



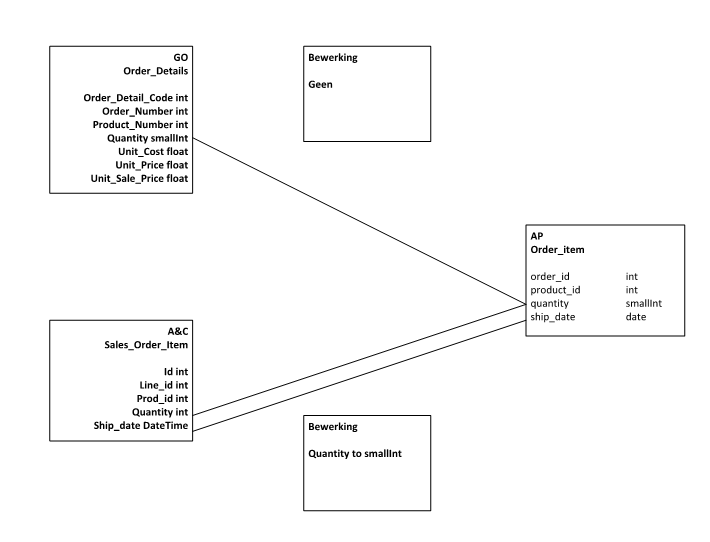
## Bonus



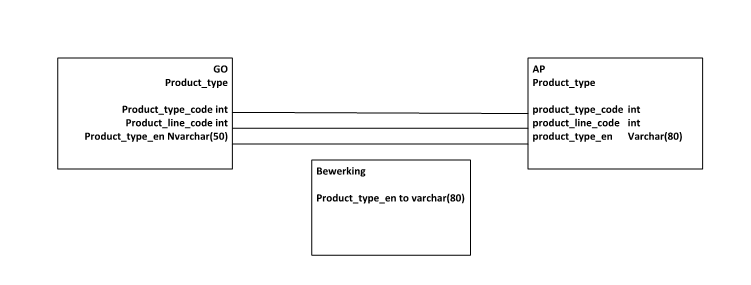
## Product



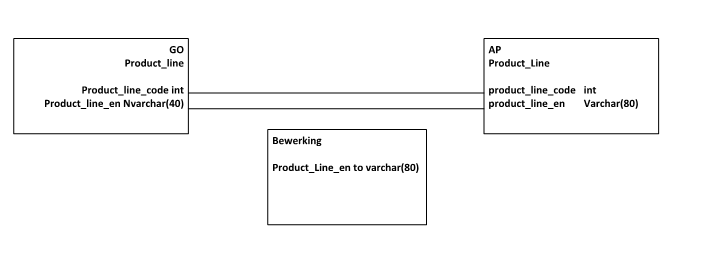
## Order Item



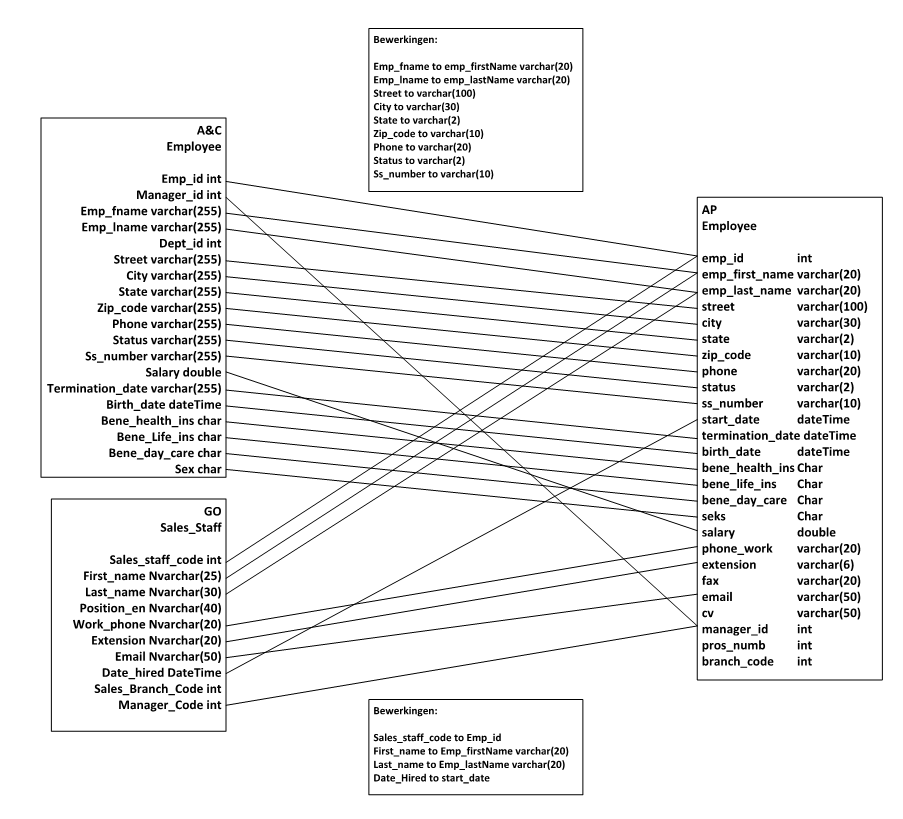
## Product Type



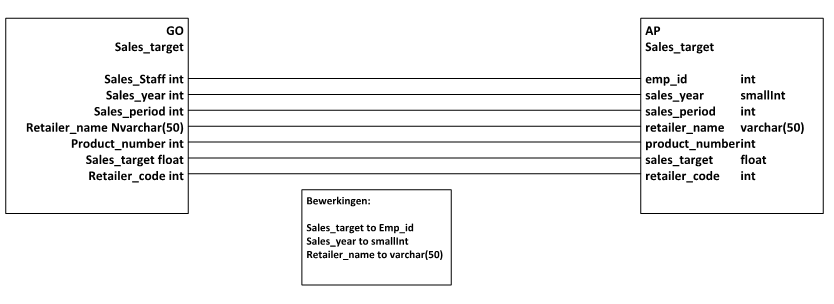
## Product Line



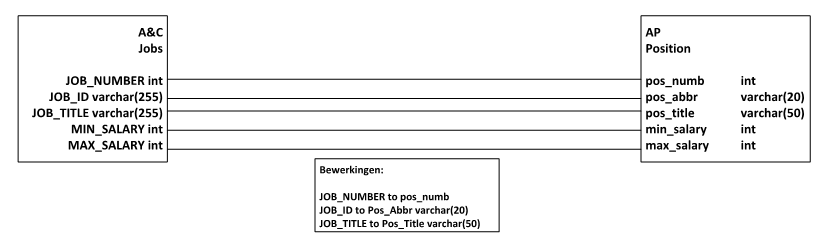
## Employee



## Sales Target



## Position



|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Target Table | Target Column | Data-type | Source Table | Source Column | Data-type | Expression | Default Value if Null | Data Issues/Quality/Comments |
| Order\_Item | Order\_id | Int | Order | Order\_id | Int |  |  | PK |
| Product\_id | Int | Product | Product\_id | Int |  |  | FK |
| Quantity | smallInt | Order\_Details / Sales\_Order\_Item | Quantity / Quantity | smallInt / int |  |  |  |
| Ship\_date | date | Sales\_Order\_Item | Ship\_date | DateTime |  |  |  |
| Product\_Type | Product\_type\_code | Int | Product\_Type | Product\_type\_code | Int |  |  | PK |
| Product\_line\_code | Int | Product\_Type | Product\_line\_code | Int |  |  | FK |
| Product\_type\_en | Varchar(80) | Product\_Type | Product\_type\_en | Nvarchar(50) |  |  |  |
| Product\_line | Product\_line\_code | Int | Product\_line | Product\_line\_code | Int |  |  | PK |
| Product\_line\_en | Nvarchar(40) | Product\_line | Product\_line\_en | Varchar(80) |  |  |  |
| Employee | Emp\_id | Int | Sales\_Staff / Employee | Sales\_staff\_code / Emp\_id | Int / int |  |  | PK |
| Emp\_FirstName | Varchar(20) | Sales\_Staff / Employee | First\_Name / Emp\_fname | Nvarchar(25) / varchar(255) |  |  |  |
| Emp\_LastName | Varchar(20) | Sales\_Staff / Employee | Last\_name / Emp\_lname | Nvarchar(30) / varchar(255) |  |  |  |
| Street | Varchar(100) | Employee | Street | Varchar(255) |  |  |  |
| City | Varchar(30) | Employee | City | Varchar(255) |  |  |  |
| State | Varchar(2) | Employee | State | Varchar(255) |  |  |  |
| Zip\_Code | Varchar(10) | Employee | Zip\_Code | Varchar(255) |  |  |  |
| Phone | Varchar(20) | Employee | Phone | Varchar(255) |  |  |  |
| Status | Varchar(2) | Employee | Status | Varchar(255) |  |  |  |
| Employee | Ss\_Number | Varchar(10) | Employee | Ss\_number | Varchar(255) |  |  |  |
| Start\_Date | dateTime | Sales\_Staff | Date\_Hired | DateTime |  |  |  |
| Termination\_date | dateTime | Employee | Termination\_date | Varchar(255) |  |  |  |
| Birth\_date | dateTime | Employee | Birth\_date | dateTime |  |  |  |
| Bene\_Health\_ins | Char | Employee | Bene\_Health\_ins | Char |  |  |  |
| Bene\_Life\_Ins | Char | Employee | Bene\_Life\_Ins | Char |  |  |  |
| Bene\_day\_Care | Char | Employee | Bene\_day\_Care | Char |  |  |  |
| Sex | Char | Employee | Sex | Char |  |  |  |
| Salary | Double | Employee | Salary | Double |  |  |  |
| Phone\_work | Varchar(20) | Sales\_Staff | Work\_phone | NVarchar(20) |  |  |  |
| Extension | Varchar(6) | Sales\_Staff | Extension | NVarchar(20) |  |  |  |
| Fax | Varchar(20) |  |  |  |  |  |  |
| Email | Varchar(50) | Sales\_Staff | Email | NVarchar(50) |  |  |  |
| Cv | Varchar(50) |  |  |  |  |  |  |
| Manager\_id | Int | Sales\_Staff / Employee | Manager\_id / Manager\_code | Int / int |  |  | FK |
| Pos\_numb | Int |  |  |  |  |  | FK |
| Branch\_Code | Int |  |  |  |  |  | FK |
| Sales\_Target | Emp\_id | Int | Sales\_Target | Sales\_Staff | Int |  |  | PK |
| Sales\_year | smallInt | Sales\_Target | Sales\_Year | Int |  |  |  |
| Sales\_period | Int | Sales\_Target | Sales\_Period | Int |  |  |  |
| Retailer\_name | Varchar(50) | Sales\_Target | Retailer\_Name | NVarchar(50) |  |  | FK |
| Product\_number | Int | Sales\_Target | Product | Int |  |  |  |
| Sales\_target | Float | Sales\_Target | Sales\_Target | Float |  |  |  |
| Retailer\_Code | int | Sales\_Target | Retailer\_Code | Int |  |  |  |
| Postion | Pos\_Numb | Int | JOB\_NUMBER | Jobs | Int |  |  | PK |
| Pos\_Abbr | Varchar(20) | JOB\_ID | Jobs | Varchar(255) |  |  |  |
| Pos\_title | Varchar(50) | JOB\_TITLE | Jobs | Varchar(255) |  |  |  |
| Min\_salary | Int | MIN\_SALARY | Jobs | Int |  |  |  |
| Max\_salary | int | MAX\_SALARY | Jobs | Int |  |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Target Table | Target Column | Data-type | Source Table | Source Column | Data-type | Expression | Default Value if Null | Data Issues/Quality/Comments |
| Order | Order\_id | INT | Order\_Header; | Order\_Number | INT |  | Auto increment | Not null, pk |
| Sales\_order | id | INT |
| Order\_date | DATETIME | Order\_Header | Order\_Date | DATETIME |  | null |  |
| Sales\_order | Order\_date | DATETIME |
| Emp\_id | INT | Order\_Header | Sales\_Staff\_Code | INT |  |  | Fk |
| Sales\_order | Sales\_rep | INT |
| Finance\_code | INT | Sales\_order | Fin\_code\_id | VARCHAR(255) |  |  | Fk |
| Oder\_method | INT | Order\_Header | Oder\_Method\_Code | ÌNT |  |  | Fk |
| Retailer\_Site\_Code | INT | Order\_Header | Retailer\_Site\_Code | INT |  |  | Fk |
| Customer\_id | INT | Sales\_order | Cust\_id | INT |  |  | Fk |
| Region | VARCHAR(50) | Sales\_order | Region | VARCHAR(255) |  |  |  |
| Training | Year | SMALL INT | Go\_training | Year | FLOAT |  |  | Pk |
| Course | VARCHAR(30) | Go\_training | Course | NVARCHAR(255) |  |  | Pk |
| Pos\_Numb | INT | Go\_training | Sales\_staff | NVARCHAR(255) |  |  | Pk, Fk |
| Retailer\_Site | Retailer\_Site\_Code | INT | Retailer\_Site | Retailer\_Site\_Code | INT |  |  | Pk |
| Retailer\_Code | INT | Retailer\_Site | Retailer\_Code | INT |  |  | Fk |
| Address1 | VARCHAR(50) | Retailer\_Site | Address1 | NVARCHAR(50) |  |  |  |
| Address2 | VARCHAR(50) | Retailer\_Site | Address2 | NVARCHAR(50) |  |  |  |
| City | VARCHAR(40) | Retailer\_Site | City | NVARCHAR(40) |  |  |  |
| Region | VARCHAR(50) | Retailer\_Site | Region | NVARCHAR(50) |  |  |  |
| Postal\_Zone | VARCHAR(10) | Retailer\_Site | Postal\_Zone | NVARCHAR(10) |  |  |  |
| Country\_Code | INT | Retailer\_Site | Counrty\_Code | INT |  |  | Fk |
| Active\_Indicator | BOOL | Retailer\_Site | Active\_Indicator | BIT(1) |  |  |  |
| Retailer | Retaielr\_Code | INT | Retailer | Retailer\_Code | INT |  |  | Pk |
| Company\_Name | VARCHAR(255) | Retailer | Company\_Name | NVARCHAR(255) |  |  |  |
| Retailer\_Type\_Code | INT | Retailer | Retailer\_Type\_Code | INT |  |  | Fk |
| Department | Dept\_id | INT | Department | Dept\_id | INT |  |  | Pk |
| Dept\_Name | VARCHAR(50) | Department | Dept\_name | VARCHAR(255) |  |  |  |
| Dept\_Head\_id | INT | Department | Dept\_head\_id | INT |  |  | Fk |
| Branch | Branch\_Code | INT | Sales\_branch | Sales\_branch\_code | INT |  |  | Pk |
| Adress1 | VARCHAR(50) | Sales\_branch | Address1 | NVARCHAR(50) |  |  |  |
| Adress2 | VARCHAR(50) | Sales\_branch | Address2 | NVARCHAR(50) |  |  |  |
| City | VARCHAR(40) | Sales\_branch | City | NVARCHAR(40) |  |  |  |
| Region | VARCHAR(50) | Sales\_branch | Region | NVARCHAR(50) |  |  |  |
| Postal\_Zone | VARCHAR(10) | Sales\_branch | Postal\_zone | NVARCHAR(10) |  |  |  |
| Dept\_id | INT | Department | Dept\_id | INT |  |  | Fk |
| Country\_code | INT | Sales\_branch | Country\_code | INT |  |  | Fk |
| Customer | Id | INT | Customer | Id | INT |  |  | Pk |
| Fname | VARCHAR(255) | Customer | Fname | VARCHAR(255) |  |  |  |
| Lname | VARCHAR(255) | Customer | Lname | VARCHAR(255) |  |  |  |
| Adress | VARCHAR(255) | Customer | Adress | VARCHAR(255) |  |  |  |
| City | VARCHAR(40) | Customer | City | VARCHAR(255) |  |  |  |
| State | VARCHAR(255) | Customer | State | VARCHAR(255) |  |  |  |
| Zip | VARCHAR(6) | Customer | Zip | VARCHAR(255) |  |  |  |
| Phone | VARCHAR(20) | Customer | Phone | VARCHAR(255) |  |  |  |
| Company\_Name | VARCHAR(255) | Customer | Company\_Name | VARCHAR(255) |  |  |  |
| Type | CHAR | Customer | Type | CHAR |  |  |  |
| Discount | INT | Customer | Discount | INT |  |  |  |
| Max\_quantity\_Order | INT | Customer | Max\_quantity\_order | INT |  |  |  |
| Bonus | Bonus\_date | DATE | Bonus | Bonus\_date | DATETIME |  |  | Pk |
| Bonus\_amo | DOUBLE | Bonus | Bomus\_amo | NUM |  |  | Pk |
| Emp\_id | INT | Bonus | Emp\_id | NUM |  |  | Pk, Fk |
| Product | Product\_number | INT | Product (AenC) | id | INT |  |  | Pk |
| Product (GO) | Product\_number | INT |
| Introduction\_date | DATETIME | Product (GO) | Introduction\_date | DATETIME |  |  |  |
| Product\_type | INT | Product (AenC) | category | VARCHAR(1) |  |  | Fk |
| Product (GO) | Product\_type\_code | INT |
| Production\_cost | FLOAT | Product (GO) | Production\_cost | FLOAT |  |  |  |
| Margin | FLOAT | Product (GO) | Margin | FLOAT |  |  |  |
| Product\_image | VARCHAR(255) | Product (AenC) | Picture\_name | VARCHAR(255) |  |  |  |
| Product (GO) | Product\_image | VARCHAR(150) |
| Product\_name | VARCHAR(255 | Product (AenC) | name | VARCHAR(255) |  |  |  |
| Product (GO) | Product\_name | VARCHAR(255) |
| Product\_description | VARCHAR(255) | Product (AenC) | description | VARCHAR(255) |  |  |  |
| Product\_size | ENUM | Product (AenC) | Prod\_size | VARCHAR(255) |  |  |  |
| Product\_color | VARCHAR(255) | Product (AenC) | color | VARCHAR(255) |  |  |  |

# Aanpak

Wij hebben ten eerste alle eisen aan het nieuwe systeem doorgenomen. Deze hebben we geprioriteerd voor zo ver dat nog nodig was, aangezien het meeste al gedaan was. Hierna hebben we de klassendiagrammen van beide instanties gemaakt. Deze hebben wij naast elkaar gezet om goed te kijken naar de verschillen en overeenkomsten. Hier hebben we één klassendiagram uit gemaakt door de eerdergenoemde overeenkomsten en verschillen te gebruiken. Dit hebben we gedaan met het gebruik van de eisen aan het systeem, maar alleen met de eisen die wij waar konden maken met de bestaande informatie. De eisen die nieuwe klassen in het klassendiagram nodig hebben, zijn er pas bijgekomen nadat we beide oude klassendiagrammen volledig hebben gecombineerd in het nieuwe diagram.

Na het diagram te maken hebben wij gewerkt aan het datadictionary. Dit was niet zo moeilijk, maar vooral veel werk. De verschillende klassen hadden we al, maar hier moesten we de datatypes, of deze nullable zijn, en enige comments nog bij zetten. Om dit te maken hebben we het nieuwe klassendiagram voor de klassen, en de twee oude data dictionaries voor de datatypes gebruikt.

Na het data dictionary was het RIM aan de beurt. Deze hebben we gemaakt door middel van het data dictionary omdat hierin alles staat wat er voor nodig is. Er staan de data types, nullables, primary en foreign keys in. Het RIM samen met de Datadictionary hebben redelijk wat tijd gekost, omdat aanpassingen steeds twee keer moeten gebeuren.

# Testen

We kunnen de migratie op een aantal verschillende manieren testen. Bijvoorbeeld door:

* De eisen aan het nieuwe systeem nog een keer doornemen en testen op het systeem door de eisen in praktijk uit te voeren voor zover dit mogelijk is.
* Test data te gebruiken om alle relaties, constraints en triggers te controleren
* Domeincontrolles: Met domeincontrolles controleren we of de waardes die we migreren wel geldig zijn.
* Bedrijfsregelcontrolles: door te controleren of er geen bedrijfsregels overtreden zijn door de te migreren data, kunnen we bepalen of de kwaliteit van de data goed is.
* Door voor de migratie en na de migratie de som van bepaalde kolommen bij te houden kunnen we controleren of er geen dataverlies/veranderingen plaats gevonden hebben tijdens de migratie.