|  |  |
| --- | --- |
|  | |
| Working With Data–Assessment 2  TU060 : Data Warehouse Modelling / Data Analysis / Machine Learning using SQL | |
|  | |
| Ciaran Finnegan  Student No : D21124026  5/1/2022 |  |
|  |  |

Table of Contents

1 Project Overview 3

1.1 High Level Description 3

1.2 Environment Assumptions 3

1.3 Project Execution Instructions 4

2 Part 1: Business Drivers 5

2.1 Background 5

2.2 Subject Area for Analysis 5

2.3 The Balanced Scorecard for Northwind Traders 6

2.4 Vision and Goals for the Data Warehouse 10

2.5 Key Stakeholders 10

3 Part 2: Data Modelling 11

3.1 Data Warehouse Schema 11

3.2 Reasons for Design 12

4 Part 3: Implement Tables and ETL Procedure 17

4.1 Implementation using Microsoft SQL Server 17

4.2 Why use this Microsoft solution (SSIS)? 31

4.3 Comparison of SSIS with Azure Data Factory V2 32

5 Part 4 : Reporting and Analysis 34

5.1 Dashboard Design 34

5.2 Dashboard Analysis 44

6 Appendices 51

6.1 Appendix 1 – SQL Scripts to build the Data Warehouse 51

6.2 Appendix 2 – SQL Scripts to Populate Data Warehouse Dimensions 52

6.3 Appendix 3 – SQL Scripts To Populate Data Warehouse Fact Table 53

6.4 Appendix 4 – SQL Scripts To Create/Populate KPI Tables 53

7 References 54

7.1 Data Warehouse Design 54

7.2 SSIS versus ADF 54

# Project Overview

## High Level Description

This document covers the design, implementation and observations on all parts of the December 2019 CA for the Data Warehousing and Business Intelligence module.

## Environment Assumptions

The IDEs used for development were a combination of Microsoft Visual Studio Community 2019 and Microsoft SQL Server Management Studio.

The database was Microsoft SQL Server 2014, which was installed and ran locally on a laptop (the same laptop on which the VS 2019 and SQL script development took place).

The server name for the database is ISVGD0LT0000292, which can be seen in a number of the screenshots in Section 4.1 of this document.

Most ETL development was done through an SSIS project, which is also described in detail in Section 4.1 of this document.

Reporting dashboards were developed and presented in Tableau 2019. A number of the worksheets build up graphs directly from the tables in the Northwind Traders Data Warehouse, which is described in Section 3 of this document.

Other Tableau worksheets build tabular reports based on data generated by specific KPI SQL scripts.

The entire reporting process, with outputs, is described in Section 5 of this document.

To summarise, the key software components used in development of this project were;

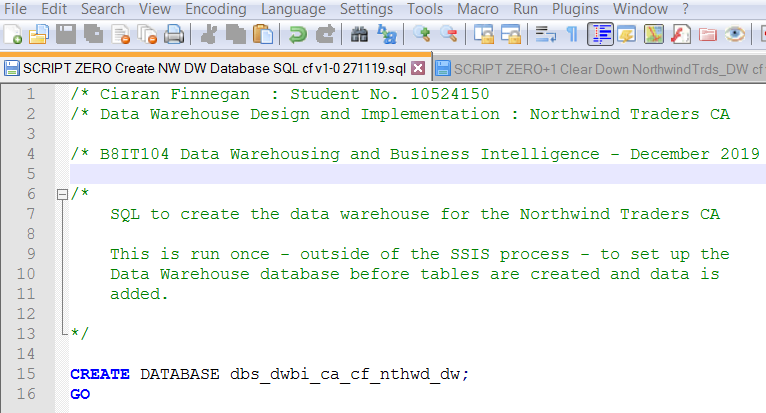
* Microsoft SQL Server 2014
* Microsoft SQL Server Management Studio
* Notepad++ v7.5.8 (for SQL script development)
* Visual Studio 2019
* SQL Server Integration Services Projects extension for Visual Studio
* Tableau 2019.3

## Project Execution Instructions

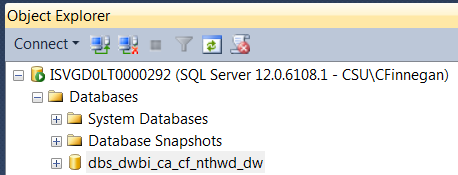
The project data warhouse solution is executed from the ***DW\_BI\_CA\_10524150.sln*** file within Visual Studio, and dashboard reports are then presented through the ***DW\_BI\_CA\_10524150\_Dashboards.twb*** Tableau file.

The steps, along with the key pre-requisites, to generate my Northwind Traders Data warehouse and associated dashboards are as follows;

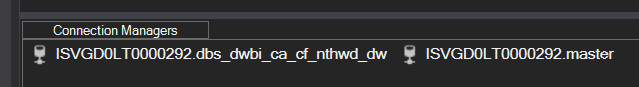
1. Use scripts provided on Moodle to set up the ‘Northwind’ database on SQL Server.
2. Use a standalone script to create my database for the Northwind Data Warehouse. This is a small SQL script executed outside of the Visual Studio project to create the database. There are no table creation or data population routines in this script, but the Control Flow/Data flow blocks in the SSIS Visual Studio will refer to this database name. I named my Data Warehouse ‘***dbs\_dwbi\_ca\_cf\_nthwd\_dw***’. The script looks like this;



1. Check in Microsoft SQL Server Management Studio that the Data Warehouse has been created:



1. Open Visual Studio 2019 and the ***DW\_BI\_CA\_10524150.sln*** project file. My connection manager is set to point to the Data Warehouse I just created (this would need to be changed if the project were to run on a different machine).



1. Execute the SSIS packages in order. This is explained in detail in Section 4.1 of this document.
2. Open the ***DW\_BI\_CA\_10524150\_Dashbaords.twb*** Tableau file. There are two dashboards for this project in this workbook with multiple worksheets making up the reports in tiles on those dashboards.

# Part 1: Business Drivers

## Background

The Northwind Traders database has been packaged with various Microsoft databases in recent years.

For the purpose of this project the Northwind database was installed on a local instance of SQL Server 2014.

Northwind Traders is primarily a wholesale food supplier. It sells a variety of products, bought from various other suppliers, to a worldwide customer base.

The structure of the Northwind Traders tables and content is reasonably intuitive. A review of the entity-relationship model (not shown in this document) provides a general insight into the set-up of the company data.

## Subject Area for Analysis

In this assignment I have chosen to focus on an analysis of **Sales performance**.

Determining which type of Sales activity generates the most revenue is an obviously essential exercise for any business. This type of analysis would include not just the optimum choice of products to offer customers, but also how effective the salesforce within the company is performing.

Certain types of products may perform noticeably better in terms of sales, and Northwind Traders may need to consider discontinuing less profitable product lines to favour investment in more popular products.

Seller performance can vary over time and it is important to know who is performing well, along with who may be under-achieving. Successful sellers can be rewarded to ensure they stay working in the company, and support can be provided to those sellers who may be in need of assistance in finding new customers in their regions.

## The Balanced Scorecard for Northwind Traders

When a company like Northwind Traders is look to make decisions that will benefit the business, and in the future, it is wise to take a SMART approach.

**S** – Start with Strategy – Have a one page plan. What are the questions that Northwind Traders should be asking?

**M** – Measure metrics and data – define the KPIs using both quantitative and qualitative data.

**A** – Analyse the data – determine meaningful insights.

**R** – Report Results - promote understanding, increase understanding.

**T** – Transform the business – define what actions need to take place.

A Balanced Scorecard is a performance measurement and management methodology to assist companies to clarify strategy, monitor progress, and define action plans. I am going to apply this approach to determine what data I should capture and ultimately report upon for Northwind Traders.

Typically the Balanced Scorecard looks at the following four perspectives;

* **Financial perspectives**. What are the financial objectives?
* **Customer perspectives**. What are the objectives for the customer base and the market in general?
* **Internal business perspectives**. What are the internal business process objectives?
* **Learning and growth perspectives**. What are the objectives when considering employees, culture, and Information Technology?

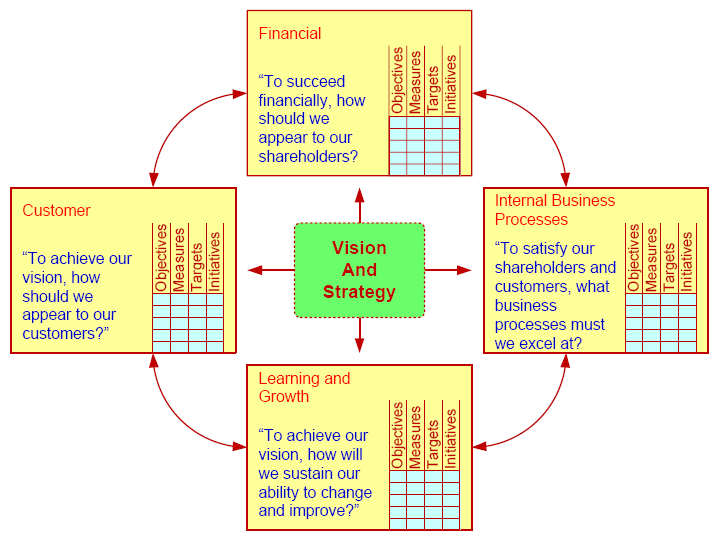
A Balanced Scorecard (BSC) consists of the following components;

* Strategic Map
* KPIs
* Action Plan

### Strategic Map

A Strategy Map is a visual one page representation of a company’s strategy. The strategy map places the four BSC perspectives in relation to one another, and shows how the strategic objectives of each perspective support each other.

For Northwind Trading, a Strategy Map could look like;



***This is a generic image taken directly from class notes but I include it here for context and to frame the following sections in the document.***

### KPIs

To start with definitions;

A **Key Performance Question** (KPQ) is a management question that captures what a manager needs to know in order to better understand the performance of the company or organisation.

A **Key Performance Indicator** (KPI) is a measure to provide managers with the most important performance information required to enable them, or their stakeholders, understand the performance of the business.

For Northwind Traders, I have identified the following KPIs to assist management in understanding Sales performance;

**Customer KPI**

Net Sales By Country: Measures into which geographical client base is Northwind Traders selling the most product.

**Financial Performance KPIs**

Total Half Yearly Net Sales: Broken down by half years from 29916 H2 to 1997 H2 (inclusive), this is a measurement of actual sales.

1997 H1 Growth (v 1996 H2): A measure of the percentage improvement in Sales growth as compared to the second half year period in 1996.

1997 H1 Growth (v 1996 H2): A measure of the percentage improvement in Sales growth as compared to the second half year period in 1996.

**Employee KPIs**

Top Five Sellers (By Net Sales Value): 1997 metric on who were the best sellers, calculated on Net Sales made.

YTD Net Sales (1998): Measure of which seller is performing best year to date in 1998 and with which product categories, based on Net Sales made.

Employee Sales - How does employee compare to average Net Sales Amount achieved by the salesforce in a given year: This is a measure per seller, per year as to how their performance rates when compared against their peers. The formula for this KPI calculates the % variance of the sellers Net Sales amount for a given year from the average for the salesforce for that year.

### Action Plan

This project will detail the steps to create the Data Warehouse and then follow through on the Business Intelligence tools, techniques, and applications to analyse the data stored in this Data Warehouse.

In a real world scenario, Senior Management, and/or other stakeholders, would determine an action plan, based on dashboards and KPI reports, to improve Sales performance.

I do not elaborate in this project on what this action plan might look like for Northwind Traders but there is a body of information here to assist in a decision making process around areas such as;

* Should Northwind Traders discontinue the purchase of grain and cereal from suppliers from 1999 onwards? What communications and other plans need to follow on from that decision?
* Should we set a plan in place to focus certain seller activity on more profitable product lines, such as beverages, from 1999 onwards?

## Vision and Goals for the Data Warehouse

The Data Warehouse created in this project, from the Northwind Traders operational database, is going to be used to generate Tableau based dashboards containing KPI reports.

These reports will provide an analysis of company sales performance that could be used to assist in the decision making process by Senior Management.

## Key Stakeholders

The KPI reports produced in this project are Sales performance data based on quarterly, half-yearly, and yearly timeframes.

This information is thus not expected to be updated daily and is more strategic in value.

The key stakeholder for the type of reporting produced in this project would therefore be;

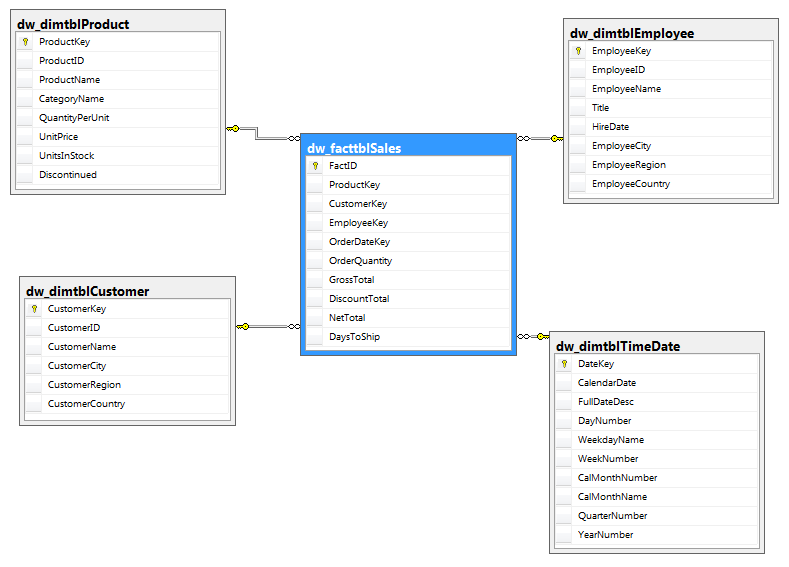
* **Senior Management.** Sales performance data that could change product lines or seller rewards is information that can only really be actioned by those within Northwind Traders who control major resourcing and policy decisions.
* **Employees.** Those in the salesforce itself in Northwind Traders. The dashboards provide a means for sellers to assess their performance when judged against the performance of their peers. It also allows for sellers to see which product lines are working best for them. The information should assist in either providing incentives or information for their own personal business decisions.
* **Company owners.** Is the Northwind Traders company profitable? Are there KPIs that show the business is being well run? If reports show that sales are consistently growing over time then this allows the owners of the company to have confidence in Senior Management. If sales were falling then the owners of Northwind Trading may look to take corrective action and seek new management.
* **Investors.** Is Northwind Trading a company that appears to have a promising future? Is it a candidate for external investment? The dashboard I generate for 1997 Sales performance contains the type of reports that would feature in a year-end report to market investors.

# Part 2: Data Modelling

## Data Warehouse Schema

The Data Warehouse for this project will be implemented with a Star Schema design.

This involves one central Fact table surrounded by a number of Dimension tables.



The operation database from which this Data Warehouse is build (the ‘Northwind’ database) has a normalised relational structure.

To optimise queries on the Data Warehouse the tables are effectively ‘de-normalised’.

The Dimension tables contain descriptive information. The Fact table contains keys to all the dimension table Primary keys, and all the measurable attributes required to meet the reporting purpose of this Data Warehouse.

## Reasons for Design

The Data Warehouse in this project is built following the design principles as described in Kimball’s four step process.

1. Identify the Business Process. Do not re-model the Business Department / Area.
2. Identify the Grain.
3. Choose the Dimensions.
4. Choose the Facts

These steps will be applied to the creation of a new Data Warehouse for this project, but this process could also be applied to the enhancement of an existing Data Warehouse to include a new business process reporting objective.

The objectives in creating the star schema model for the Data Warehouse are;

* Be simple.
* Be easy to use.
* The ETL process loading into these tables should be as simple as possible.
* Queries should perform well with Tableau, or other tools.

### Identify the Business Process

This is the first step in designing the Data Warehouse.

A ‘Business Process’ can be defined as a natural operational activity performed in the organisation, in this case Northwind Traders, that is supported by some form of data collection.

The following should be considered when identifying the process on which we wish to focus;

1. **Look at the business process not the business department.** This allows for data to be collated and reported on in a more consistent manner across the organisation. It helps in avoiding duplication of data, which might occur if we replicate the structure of business units in the Data Warehouse. In this project we are looking at the Sales process by measuring Sales performance and we are not looking to build a Northwind Sales Department report. I will focus on the Sales business process to provide a singular vision of Sales performance in Northwind Traders, which actually somewhat spans across business departments.
2. **Assess impact and risk in reporting on the chosen business process**. Impact is generating reports that the business actually want on a regular basis. I am assuming that Northwind Traders are dependent on quarterly, half yearly and yearly reports on Sales performance to make decisions on optimising its sales force and product range. The risk with reporting on some business processes for Northwind Traders is that there are some data gaps in the database and this could complicate the reporting in certain areas. For example, there are NULL values in the ‘Region’ column in the Employees table in the Northwind Traders operational database. This could be patched but it is additional effort and complexity. (Arguably it would not be very difficult to work around but we can avoid the risk by choosing a business process that is not reliant on that column).
3. **In a real world scenario, the business users would provide guidance on my impactful business process for them.** Business users can also help decipher complex business processes. The Northwind Traders business model is relatively simple to understand, so it is my assumption the Sales performance is an important business area for the company.

From the above points, it is important to emphasise that the Data Warehouse I am creating is intended to service reporting needs for the entire Northwind Traders organisation.

### Identify the Grain

This is the second step in designing the Data Warehouse.

This is the most important phase of the design process. Redesigning a Data Warehouse at a later date to increase the level of granularity could be an expensive and time consuming process.

The resultant Fact table will be at the centre of our star schema. This table contains all of the measurable facts about the captured business process. We will use the Fact table to extract information on how successful, or not, the Sales performance is, and has been, in Northwind Traders.

I have followed three particular guidelines in my project to identify the correct level of granularity when considering the design of this Data Warehouse.

1. **What is represented by one ‘Fact’ row?** What level of granularity is captured?
2. **Choose the most atomic level of information.** The data cannot be meaningfully subdivided any further. It also allows for easy and effective aggregations.
3. **Allow scope for future reporting requirements**. It is hard to predict future user requirements so the granularity is important to allow further, possibly ad-hoc, reporting requirements.

In my Fact table the focus is on measuring Sales performance. Thus in my dimensional model one Fact row represents *one product sold to a customer by one seller on one given day*.

This is a lower grain than what a seller sold in a day, as the seller may make multiple sales on one day.

My Tableau dashboards, as shown in Section 5 of this document, look at Sales performance at a quarterly, half yearly, and yearly basis. However, it is still good practice to store information in the Data Warehouse at a daily level to future proof for other Sales performance reporting requirements.

### Choose the Dimensions

This is the third step in designing the Data Warehouse.

Guidelines for this process can be summarised as follows;

1. Who, what, where, when?
2. Best attributes are descriptive.
3. De-normalizes design focuses on high performance reads.
4. Use smallest data types possible.

To capture the attributes of the Sales performance process in Northwind Traders, I need to have the information on **who** (Seller) sold which **what** product to **whom** (Customer) and **when**. Fields like phone numbers, and postal codes are not used in any of my KPIs and hence are not included in any of the dimension tables.

This question dictated the choice of the dimension tables I selected for my Data Warehouse schema;

* Product (*dw\_dimtblProduct*)
* Customer (*dw\_dimtblCustomer*)
* Employee (*dw\_dimtblCustomer*)

As an example, the choice of supplier is not relevant for reporting on the Sales performance process, so I did not include a Supplier dimension table.

The TimeDate dimension table (*dw\_dimtbltimeDate*) is built to provide additional date granularity and a conversion of the date into an integer format to improve reporting performance.

A new ‘surrogate key’ has been created for each of the dimension tables. It is a simple numeric value that I have set to increment in the SQL scripts used in the CREATE TABLE routines.

The surrogate key is necessary to uniquely identify each row in the dimension table and to avoid any confusion with the source Primary Keys from the Northwind operational database (‘Northwind’). This is particularly useful if the key structure in the Northwind Traders operational database changed in the future. Such changes will not then have a knock on impact on the Data Warehouse and reporting applications should still be valid.

The surrogate keys of each dimension table are usually simple integer values and are also added to the Fact table. This is done to minimise the number of joins needed to fetch data, which improves the response time of queries (as does the use of simple integer key values).

### Choose the Facts

This is the fourth step in designing the Data Warehouse.

The Fact table exists at the centre of the star schema, as can be seen in Section 3.1.

Defining the measures for the Fact table should follow guidelines such as these;

1. **How does the business measure success?** For Sales performance we are looking at the Net Sales totals and what Products generate the most, who is the best employee at selling based on sales, and where are the highest sales being made (based on the Net Sales amount)?
2. **The best measures are fully additive**. It should be possible to roll up the measures and easily perform aggregations. In my Tableau workbook I generate reports that display Net Amount values totalled by geography and individual sellers (as an example).
3. **Data access tools, such as Tableau, are suitable for non-additive measures.** Year To Date averages are calculated in one of my dashboards, but would not be a meaningful unit of data in the Fact table.

The Net Sales amount is a relatively simple metric to report on, as it is stored in the Fact table.

I have also included the GrossTotal amount, OrderQuantity, DiscountTotal, and DaysToShip in the Fact table. These measures are ultimately not used in the Tableau dashboards in Section 5 but are included them to allow for possible future reports.

The Net Sales amounts stored at the level I have chosen in the Fact table allows for the KPIs in Sales performance to be easily reported in Tableau (although with supplementary SQL in some cases).

# Part 3: Implement Tables and ETL Procedure

## Implementation using Microsoft SQL Server

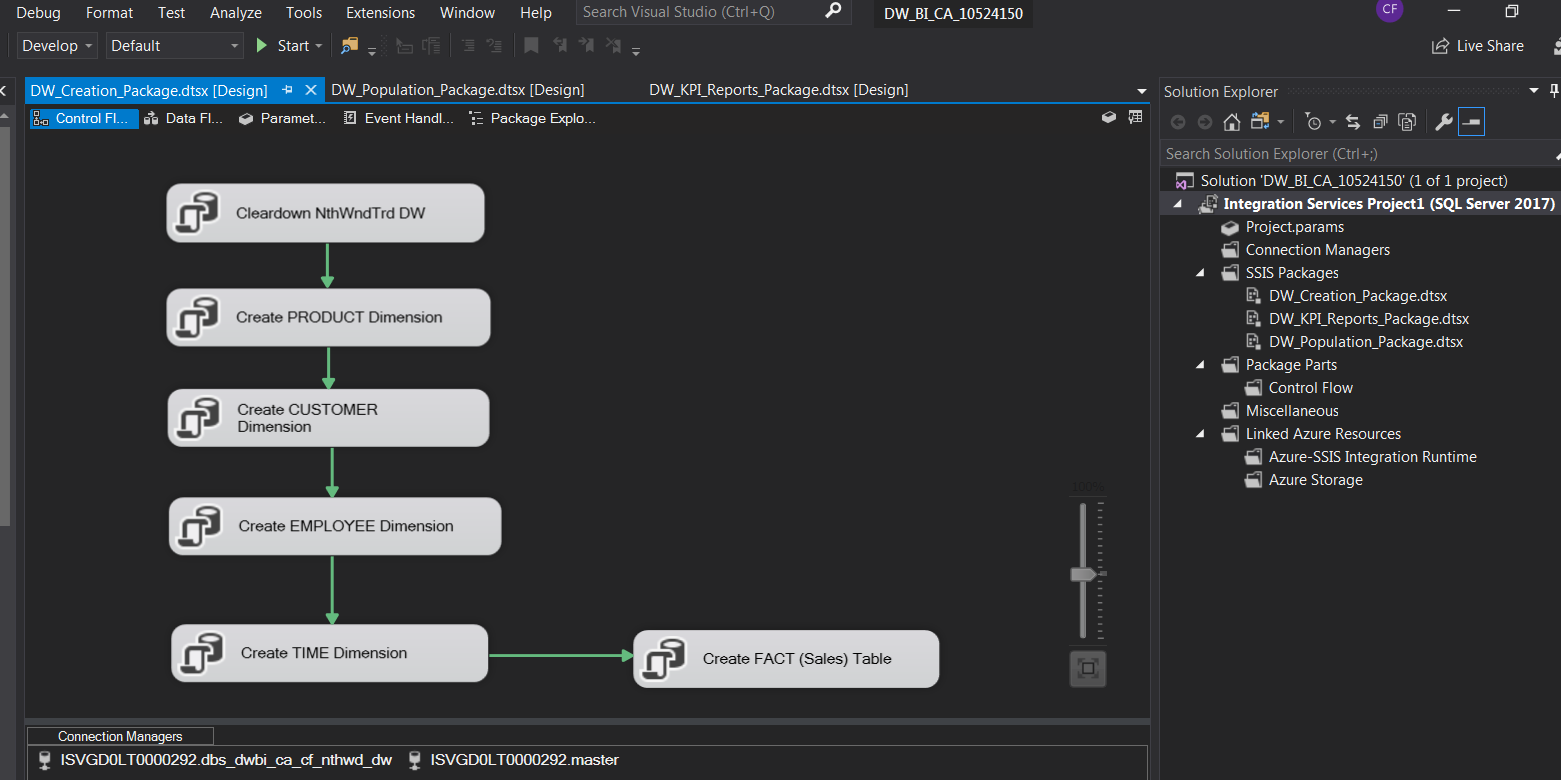
The Northwind Traders Data Warehouse for this project was generated through a Microsoft SQL Server Integration Services project (SSIS).

My SSIS project was built under the Visual Studio 2019 solution file ***DW\_BI\_CA\_10524150.sln***.

The SSIS project contains three separate Packages;

* ***DW\_Creation\_Package.dtsx***. The Control Flow tab contains the ‘Execute SQL Task’ blocks that create the Data Warehouse tables.
* ***DW\_Population\_Package.dtsx***. The Control Flow tab contains the ‘Execute SQL Task’ and Data Flow blocks that populate the Data Warehouse Dimension and Fact tables.
* ***DW\_KPI\_Reports\_Package.dtsx***. The Control Flow tab contains the ‘Execute SQL Task’ blocks that populate the supplementary Data Warehouse tables used for KPI reporting.

This screen shot shows a cross section of the Visual Studio 2019 IDE with the packages created, and focuses on the initial Control Flow for creating the Data Warehouse tables.



A significant amount of the development and testing of the SQL used in the SSIS ‘Execute SQL Task’ blocks was done through Microsoft SQL Server Management Studio.

This SQL, and the SQL scripts on which the table creation Data Flow blocks were based, are included in the Appendix Section 6.1, 0, and 0.

### Data Warehouse Creation Using SSIS

Sections 1.2 and 1.3 explain the necessary prerequisites required and steps to execute the actual project.

This section will describe the development and execution of the SSIS steps to create the actual Data Warehouse tables I choose to set up for this project.

Section 2.2 elaborates on the chosen area for analysis in my project, which is the Sales process, and more specifically Sale performance.

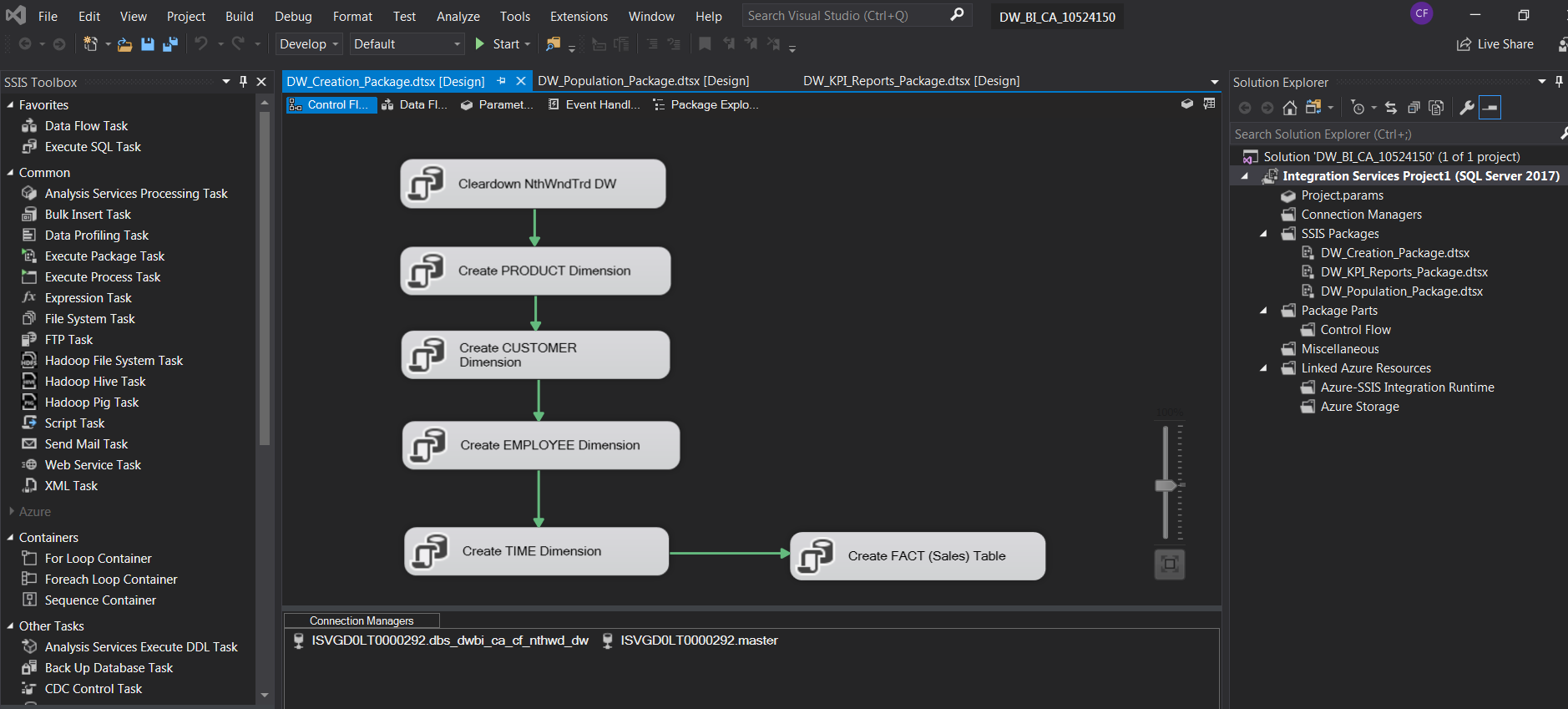
Section 3.1 explains the layout of the Data Warehouse schema that I intend to implement. The ***DW\_Creation\_Package.dtsx*** package sets up the initial phase of the ETL process by creating the Data Warehouse tables. Population of the Date Warehouse tables is described in Section 4.1.2.

This package is a relatively straightforward sequence of ‘Execute SQL Task’ blocks.

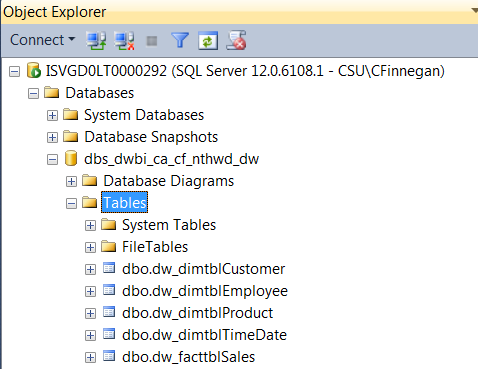
1. The first block is labelled ***‘Cleardown NthWndTrd DW’*** and clears down any tables (including data) that may already exist in the Data Warehouse. This is not an essential step but I found it was easier to include this action step because it would allow me to easily run the package multiple times during development, without having to delete tables or data externally in Microsoft SQL Server Management Studio.
2. The second block creates the Product Dimension table.
3. The third block creates the Customer Dimension table.
4. The fourth block creates the Employee Dimension table.
5. The fifth block creates the TimeDate Dimension table.
6. The sixth, and last, block creates the Fact table for Sales data.

The sequence of the blocks is relatively arbitrary, with the exception of the Fact table. It must be run last or the Package execution will fail because of referential integrity issues when trying to populate the rows in the Fact table.

A screen shot of the package within the Visual Studio IDE is given here:



Once the package is run you can see that the following tables are created in the Data Warehouse:



The Date Warehouse table names follow the convention format of ***dw\_dimtbl<Dimension>***, hence the following four Dimension tables have been created;

* dw\_dimtblCustomer
* dw\_dimtblEmployee
* dw\_dimtblProduct
* dw\_dimtblTimeDate

The following Fact able is also created;

* dw\_facttblSales

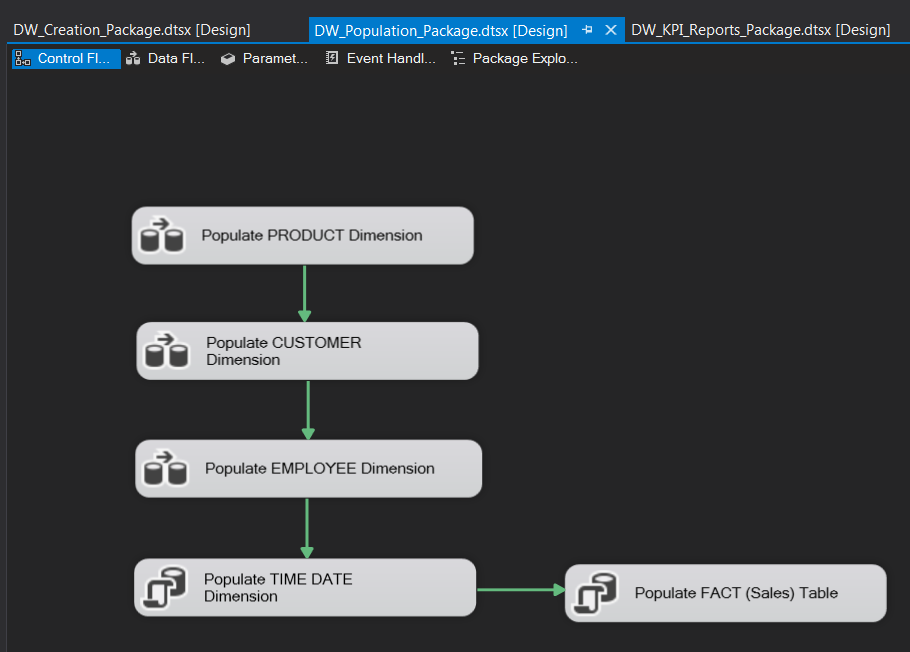
Section 6.1 of this document (Appendices) provides the SQL scripts used to prepare and validate the Control Flow blocks in this Package.

### Data Warehouse Population Using SSIS

The ***DW\_Population\_Package.dtsx*** SSIS package executes an ETL procedure to extract data from the operation database, which in this project is the ‘Northwind’ database set up as part of the perquisite steps.

The data is then transformed by Data Flow blocks in this package and then loaded into the Data Warehouse tables.

The overall Control Flow for the Package is show here;



**PRODUCT / CUSTOMER / EMPLOYEE**

The Package reads the operational data in the Northwind database for the source dimensions of PRODUCT, CUSTOMER, and EMPLOYEE.

The Control Flow in the Package extracts data in sequence.

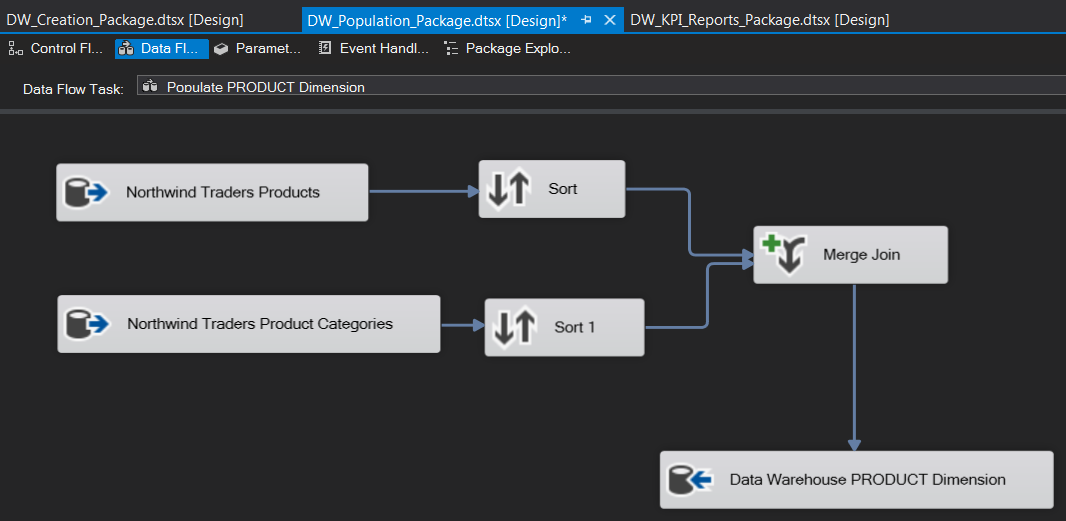
Each associated Data Flow uses a ‘Source Assistant’ tool to read the Northwind database table.

Information is sorted and merged as required, again by using an SSIS tool.

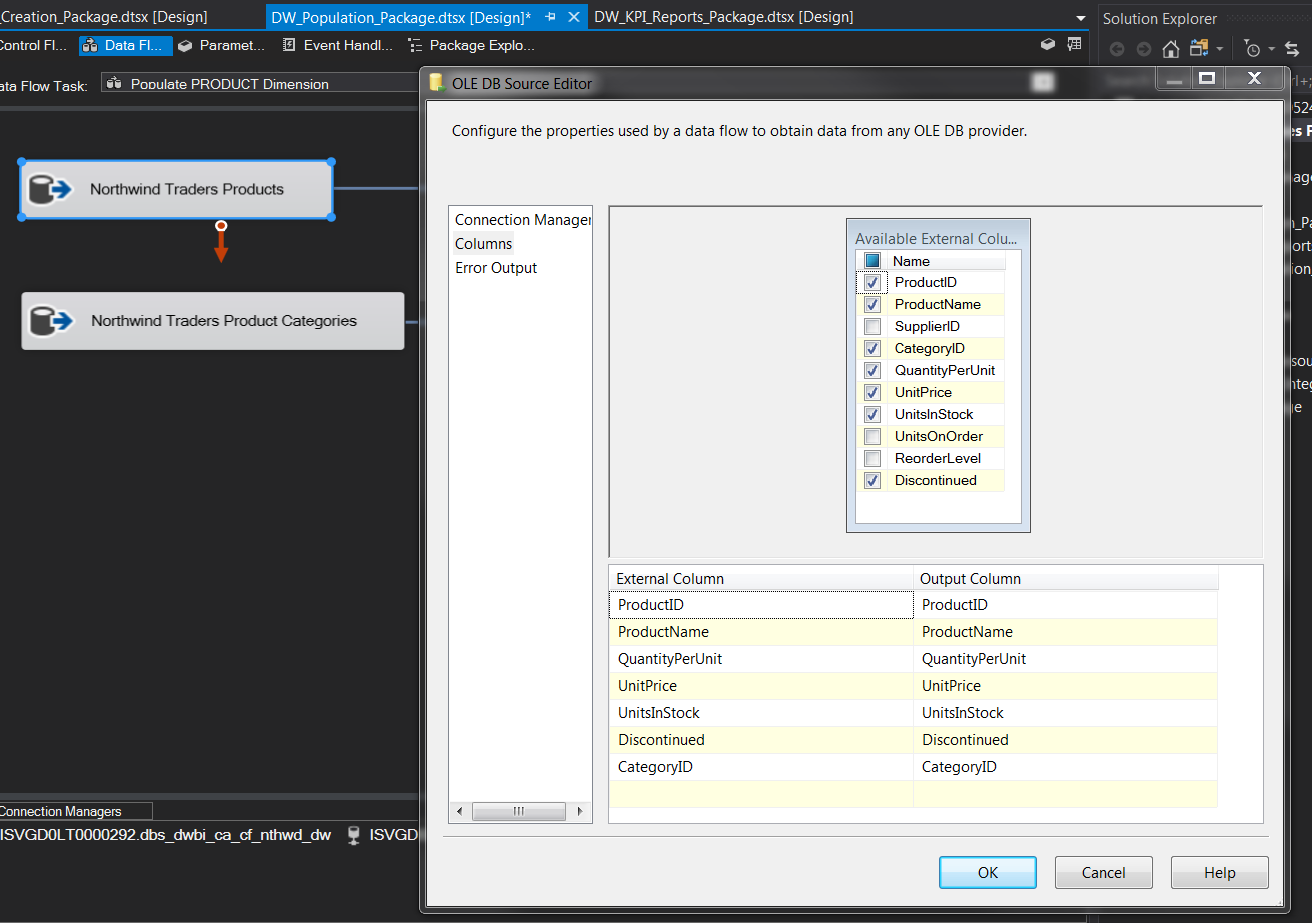
A ‘Destination Assistant’ then loads the dimension information into the required table in the Data Warehouse database (dbs\_dwbi\_ca\_cf\_nthwd\_dw).

PRODUCT

The Data Flow Task in the Package for the Product is shown here;



If we look at the details behind the ‘Northwind Traders Products’ Destination Assistance tool we see the following extraction logic:



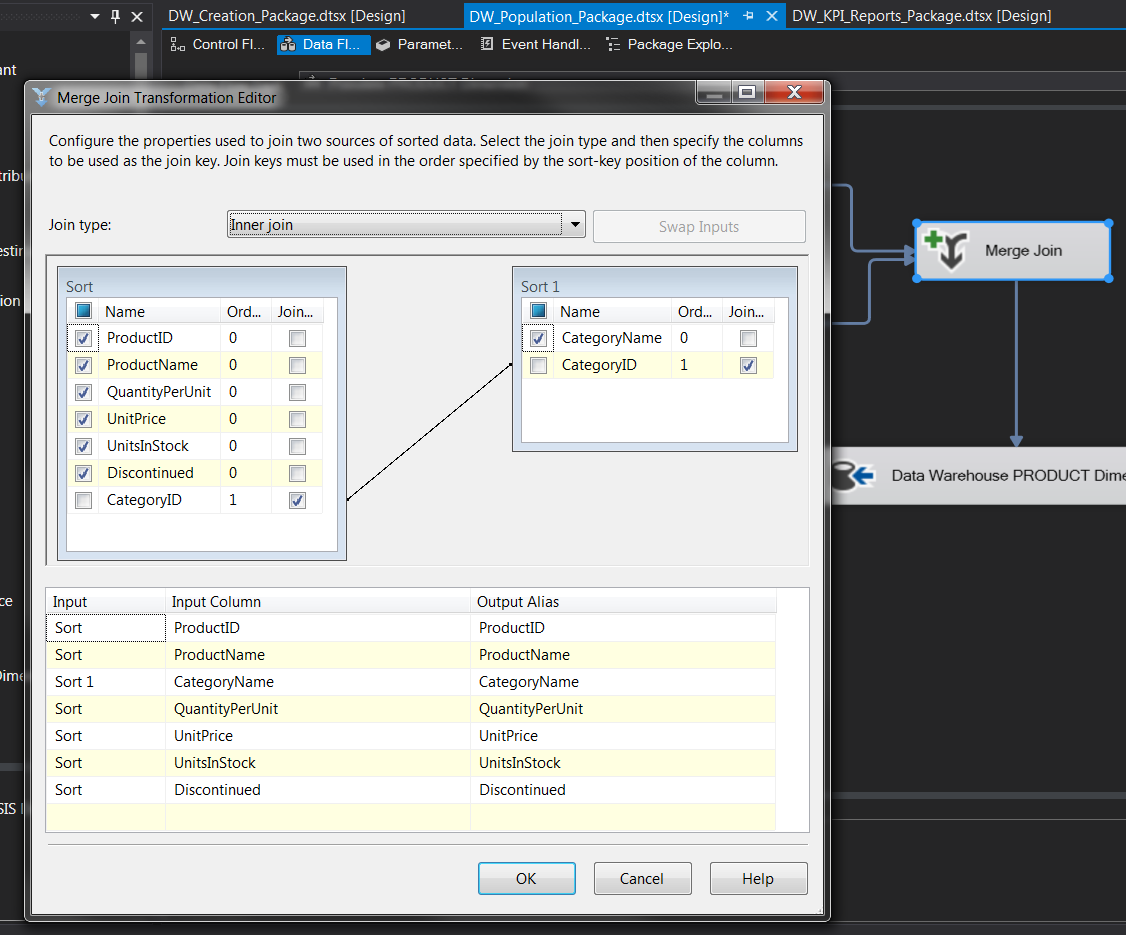
The check boxes show the information that I wish to extract from the source Northwind database table for Product.

Another tool carries out the same type of operation on the source Northwind database table for Categories (Product Categories).

The data is then sorted and merged. The merge is required because the PRODUCT dimension table in the warehouse contains information from both the Product and Categories table in the source Northwind database tables.

Effectively we are denormalizing the data from the Northwind operational database into a dimension table in the Data Warehouse to improve the performance of queries involving Product and Categories.

The ‘Merge’ tool in this part of my SSIS Package is shown in this screen shot;



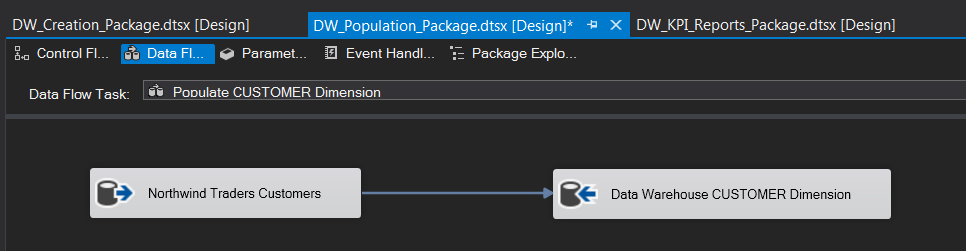
The operations in the previous screen shots effectively replicate the type of actions that SQL scripts, using JOINS and INSERTS, would perform to extract, transform, and load the data.

SQL scripts were written for this project to guide the SSIS Package development process. These are included in Section 0 of this document (the Appendices).

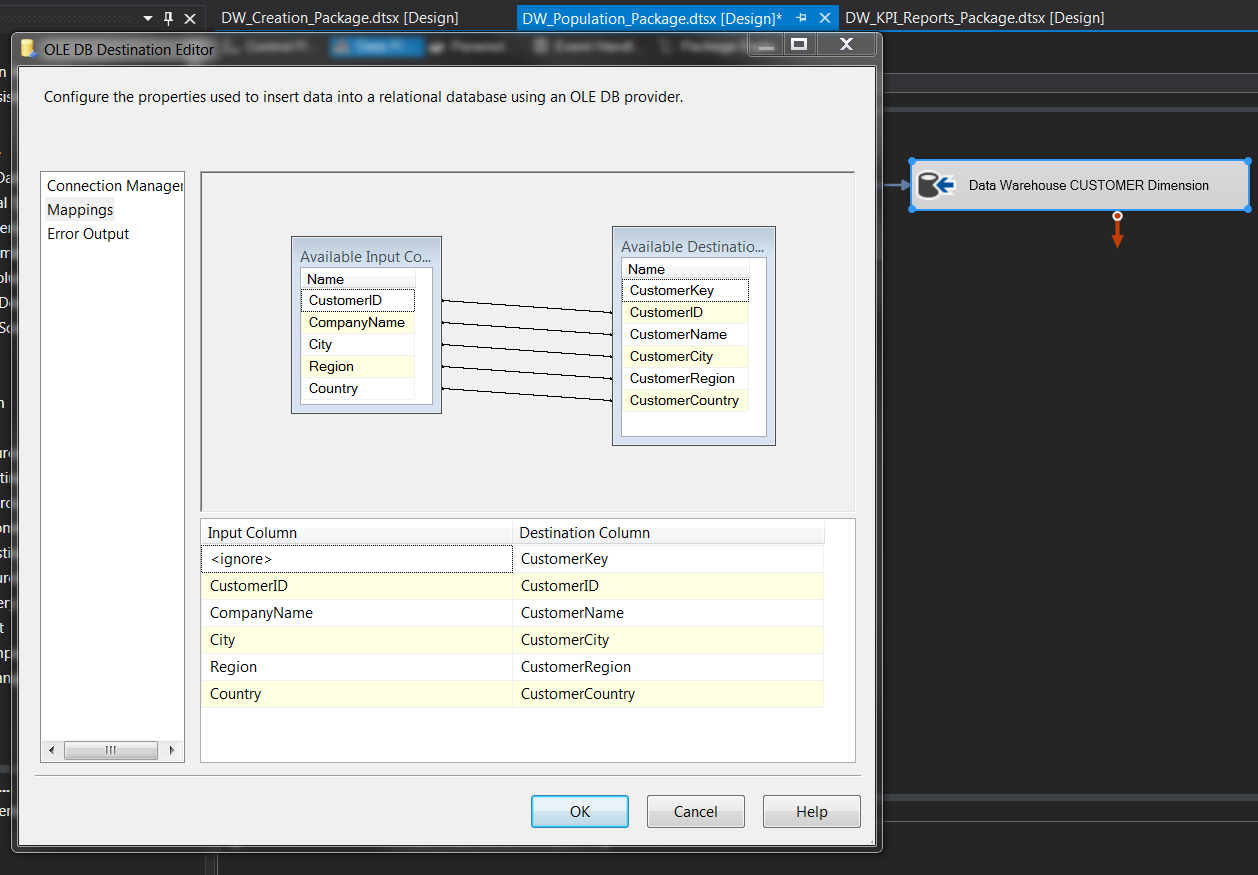
CUSTOMER

This is a more straightforward extraction process. There is no need merge another table with the source Customer table in the Northwind database.

The Data Flow is shown here:



Looking only at the Destination Assistant tool we can see what data is being used to populate the Data Warehouse dimension table for Customers.

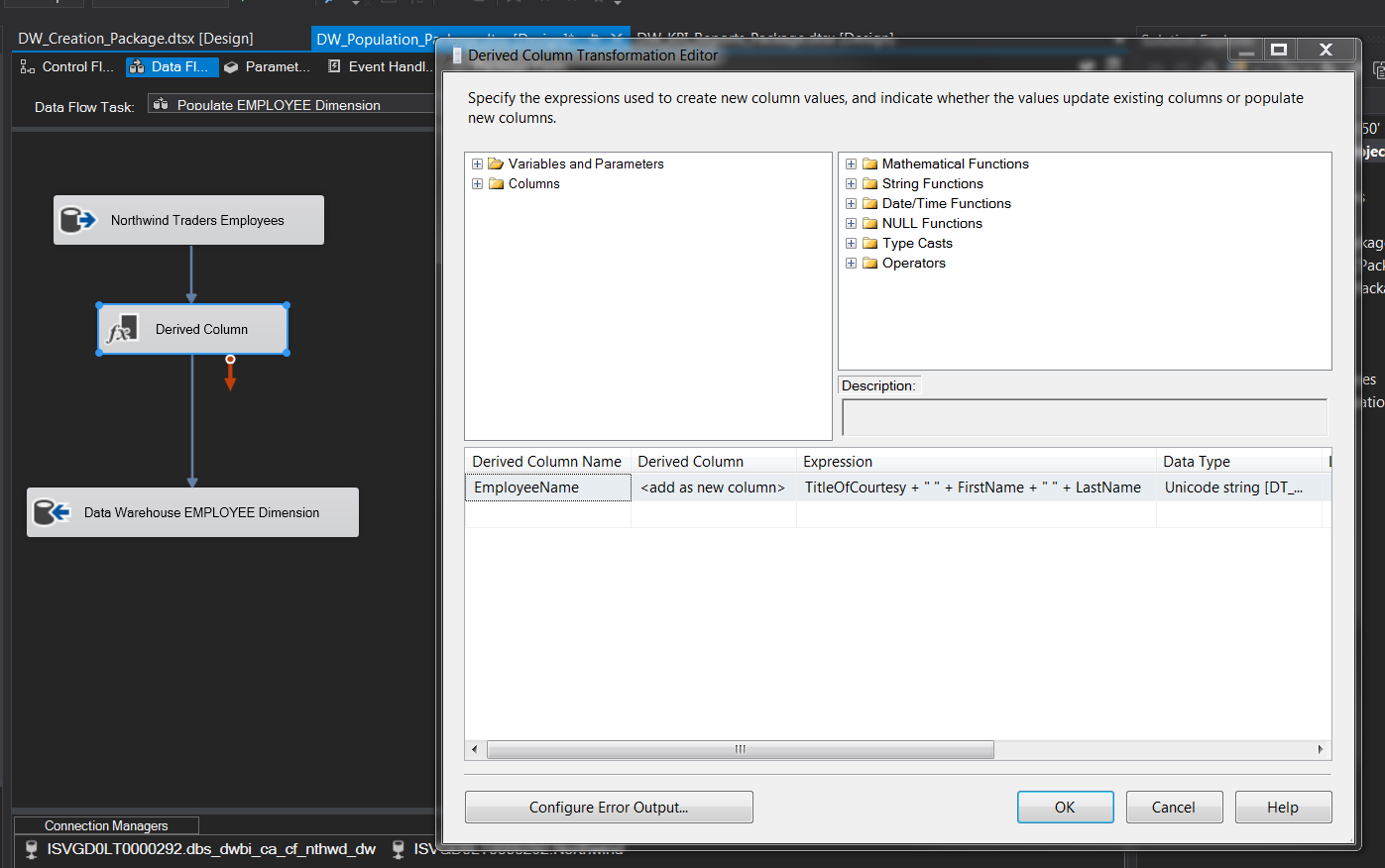


EMPLOYEE

This is also a relatively simple extraction from one table in the operational database to one dimension table in the Data Warehouse.

There is some minor data manipulation we wish to perform on the Employee name so that the full salutation, first name, and last name are stored in a single column. This is done to simply the report generation process.

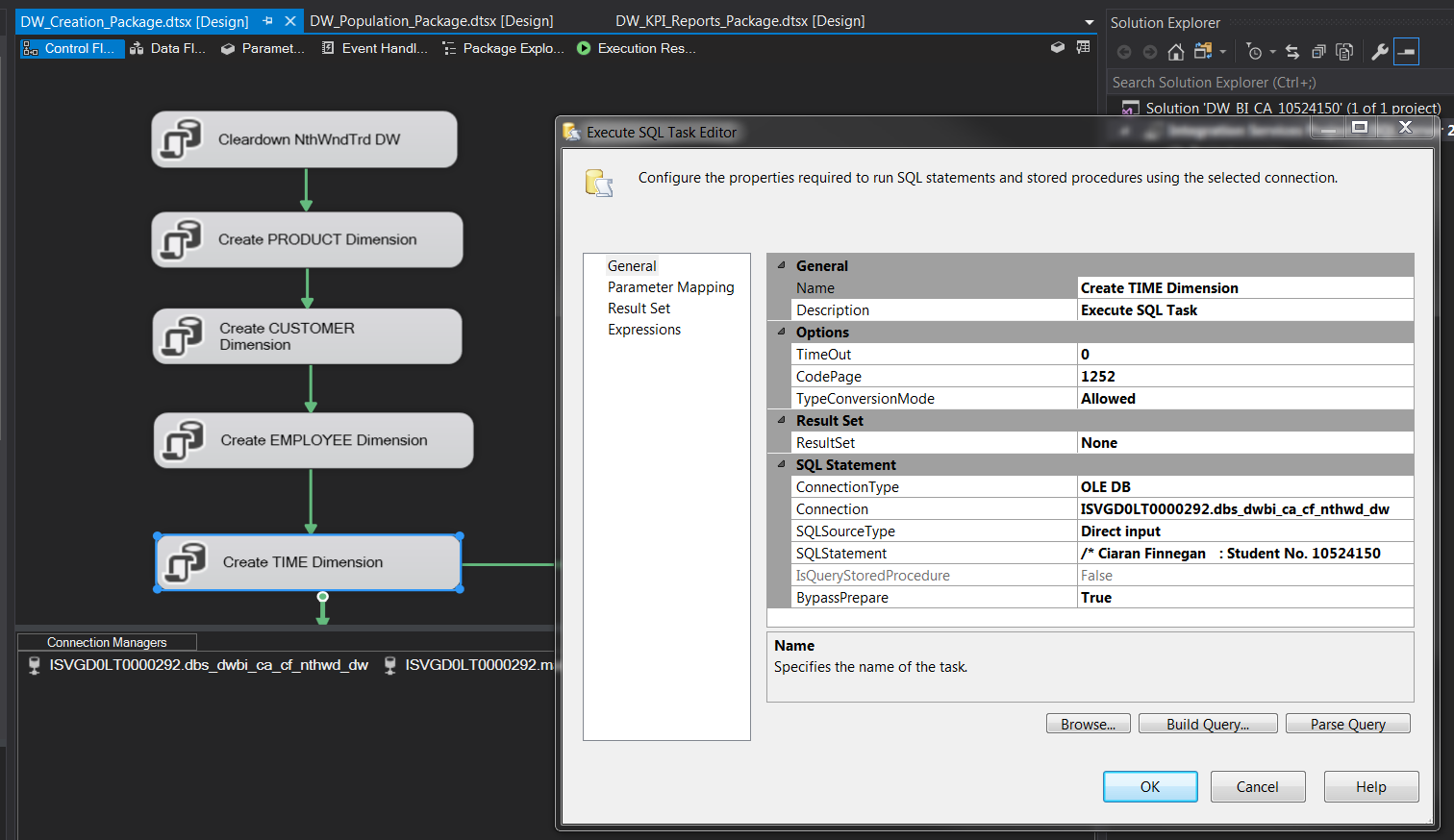
A ‘Derived Column’ tool is used to concatenate the data from the source Northwind table columns into a single column in the Employee dimension table in the Data Warehouse.



TIME / DATE

The Control Flow in this Package uses an ‘Execute SQL Task’ tool to run SQL commands to build up the TimeDate dimension table in the Data Warehouse.

This screen shot shows the ‘Execute SQL Task Editor’ pop up. The SQL will be run against the Data Warehouse database.



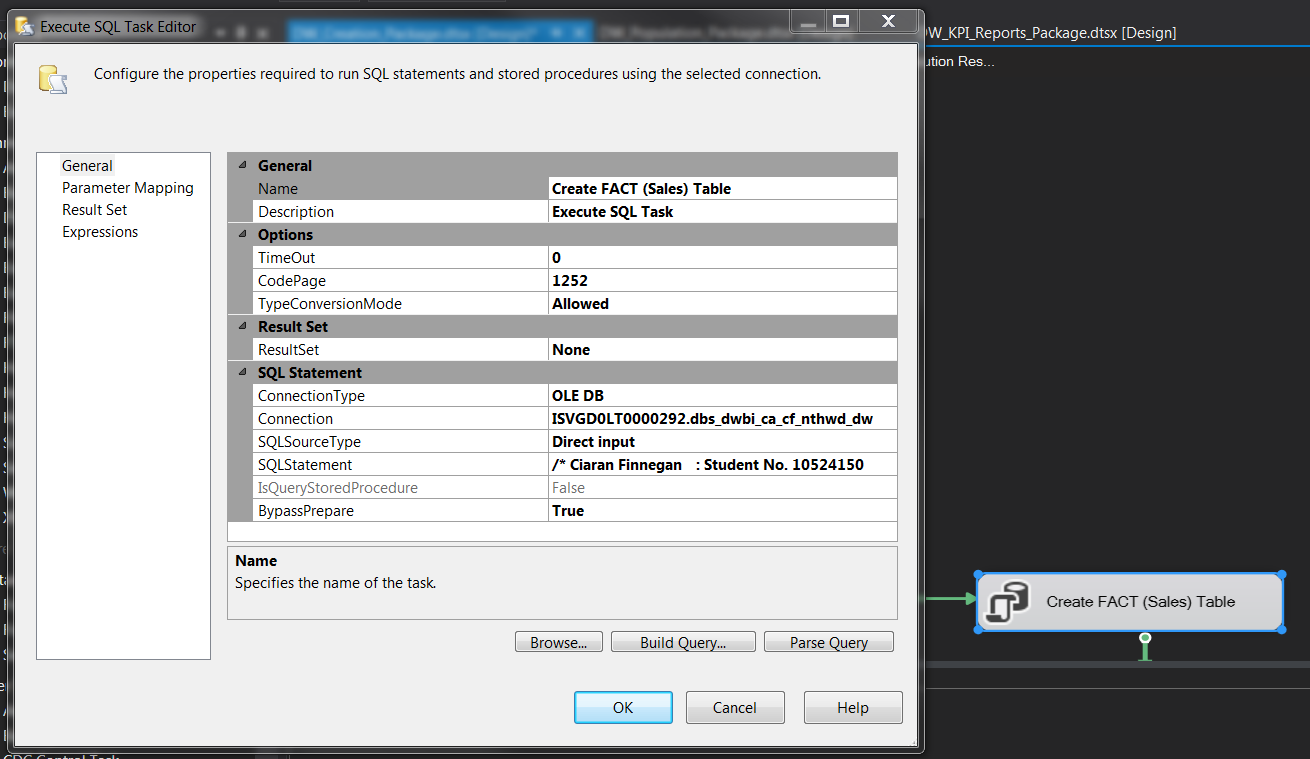
The TimeDate dimension in the Data Warehouse represents the grain of information upon which I wanted to run my reports. This table represents the dates when individual sales were made by sellers (employees).

The date range in the table is determined by an examination of the range of date transactions in the Northwind database.

The SQL used to populate the TimeDate, and other Dimension tables is included in Section 0 of this document.

SALES FACT Table

The population of the FACT table requires another ‘Execute SQL Task’ tool, as shown in this screenshot;



The underlying SQL runs a subquery to join across the Product, Orders, Customers, and Employees tables in the operation (Northwind) database.

The Primary Keys from all these tables are extracted so that they can be added into each row of the fact table.

In addition, the measures for Sales information are added to the Fact table based on the information pulled from the join.

The derived facts, the measures, recorded in my Sales Fact table are;

* OrderQuantity
* Gross Total
* DiscountTotal
* Net Total
* DaysToShip

Strictly speaking, some of the measures such as ‘DaysToShip’ could be removed as they do not feature in any of the dashboards I created for Sales performance. However, I included these measures because it allows for future reporting on Sales if I wanted to relate any of my metrics to how the Products take to arrive at the Customer after being ordered from a Seller.

The SQL used in this tool in the package to populate the Fact table is included in the Appendix (Section 6.2).

### Data Warehouse KPI Reporting Set Up Using SSIS

The two reporting dashboards for this project have been implemented in Tableau.

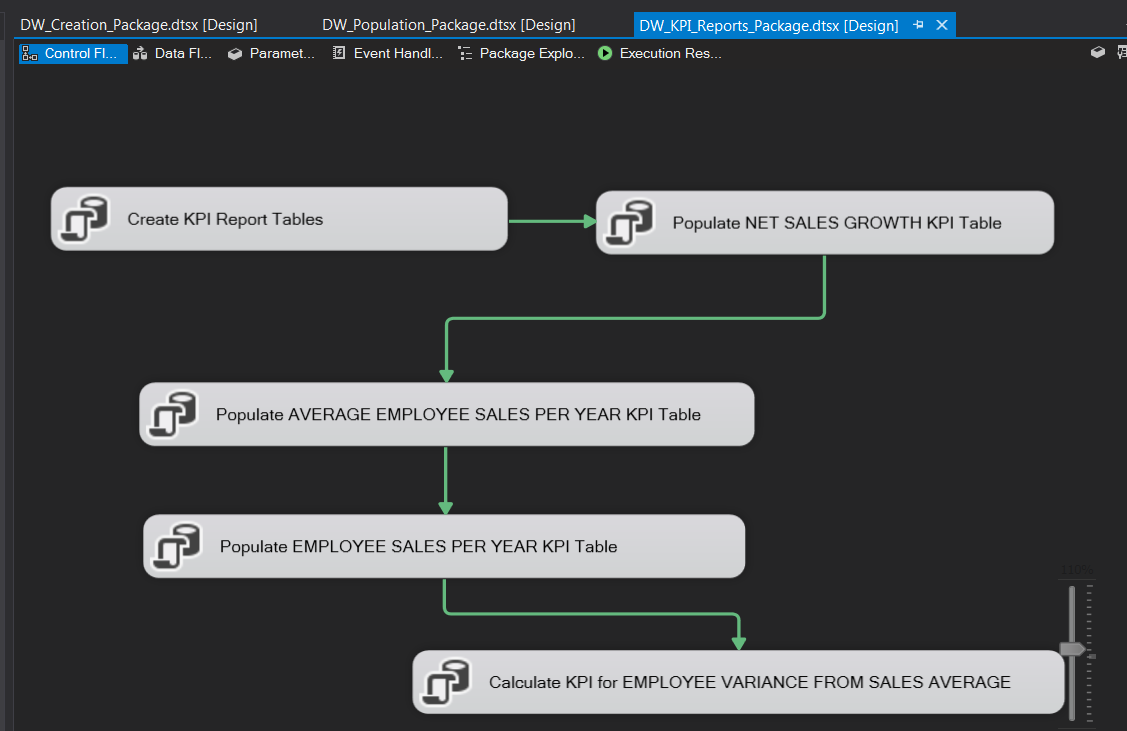
Some of the KPI metrics displayed on these dashboards are generated by reading the Data Warehouse Dimension/Fact tables directly and manipulating the information into graphical forms.

However, other dashboard metrics were more complex and it was considerably easier to run SQL scripts to read data directly from the Data Warehouse and populate dedicated report tables. These tables are then read directly into Tableau.

A comprehensive description of the dashboards created for this project is given in Section 5 of this document.

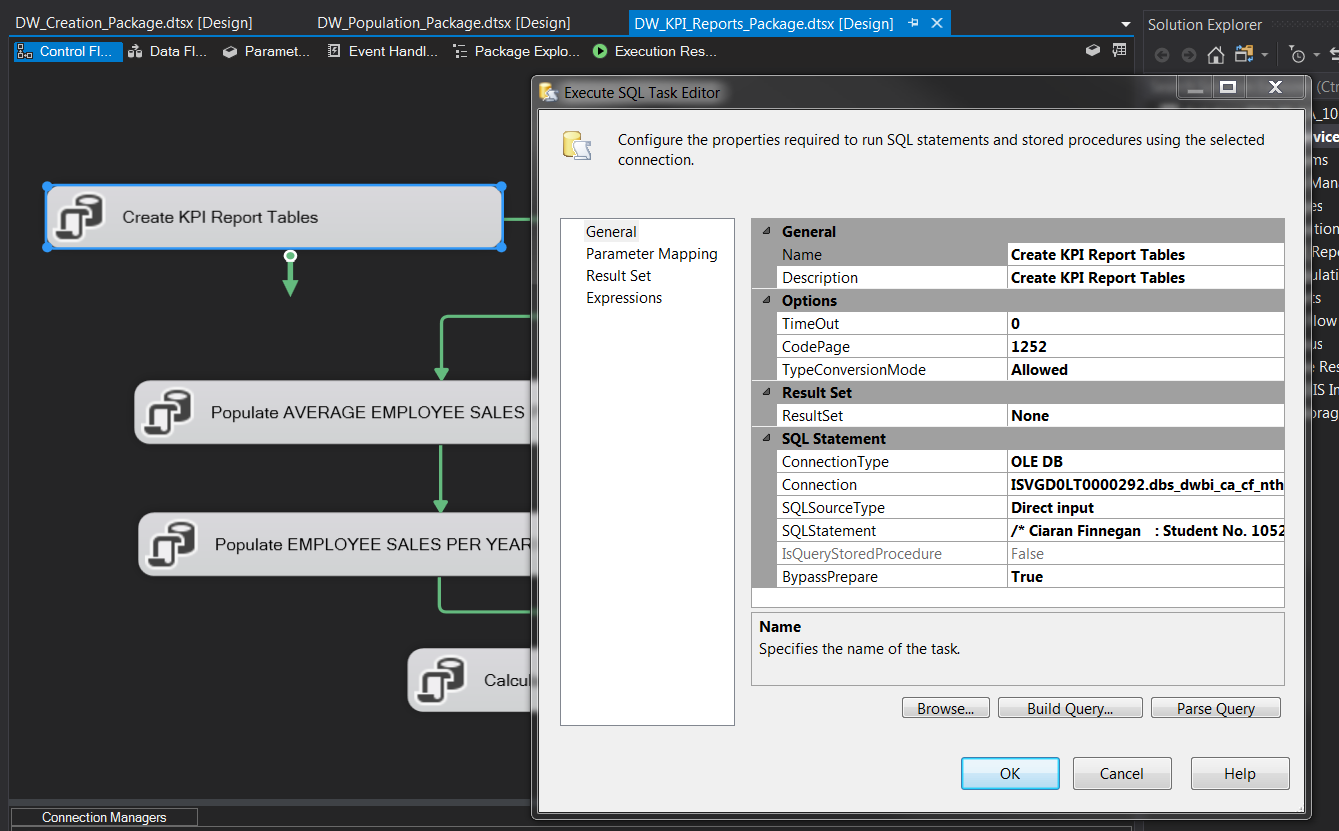
Rather than execute the SQL for these dashboards separately in Microsoft SQL Server Management Studio, I choose to build another Package in my SSIS project and run the report generating logic through a sequence of ‘Execute SQL Task’ blocks.

A screenshot of the Control Flow for this Package is given here;



**Creation of KPI Report Tables**

The first block in this Package creates the KPI tables that will eventually be populated with KPI metric data.

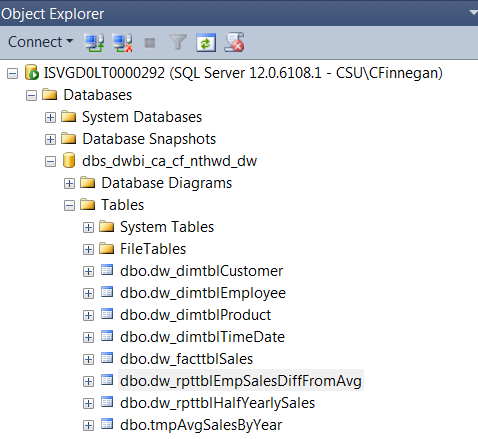


The SQL behind this block is given in the Appendix in Section 6.4.

The tables created for the KPI reports are;

* ***dw\_rpttblHalfYearlySales*** – used to display growth in Net Sales on a half yearly basis.
* ***dw\_rpttblEmpSalesDiffFromAvg*** – used to capture each Sellers performance per year against the average Seller performance.
* ***tmpAvgSalesByYear*** – interim table to store temporary values for the Seller performance table above.

Once this block is executed in the package, the additional KPI tables can be seen in the Data Warehouse;



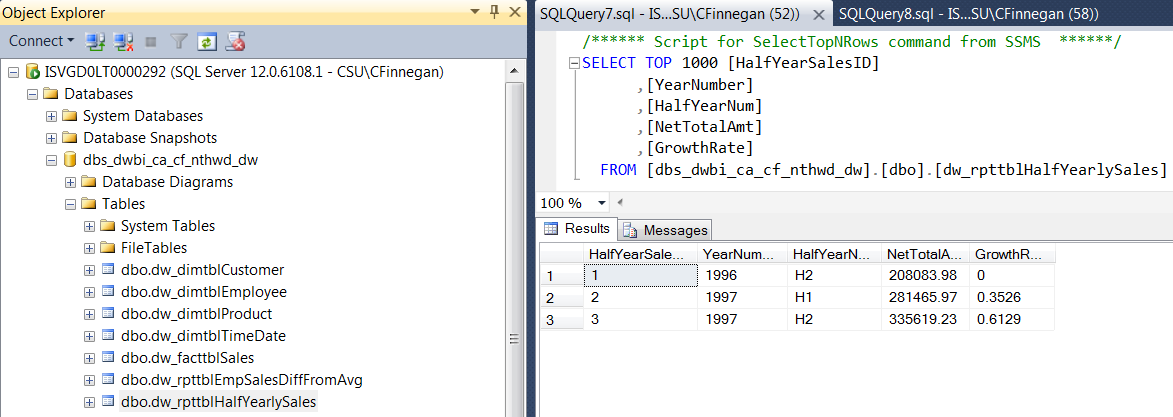
**Population of KPI Report Data**

The SQL to populate is given in the Appendix in Section 6.4.

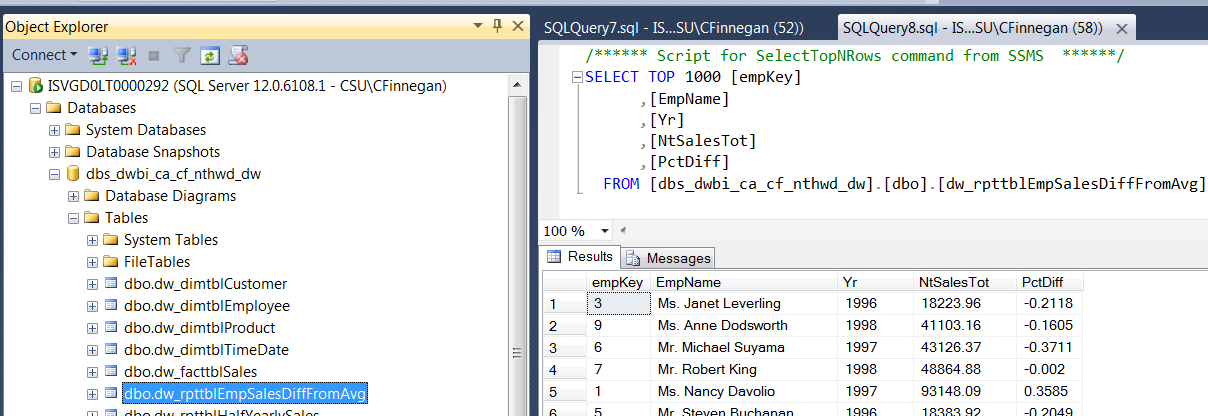
The remaining blocks in the Control Flow of this package populate the KPI report data.

The primary output from these blocks will populate the date with following datasets;

*Half Yearly Sales Figures in 1996 and 1997*



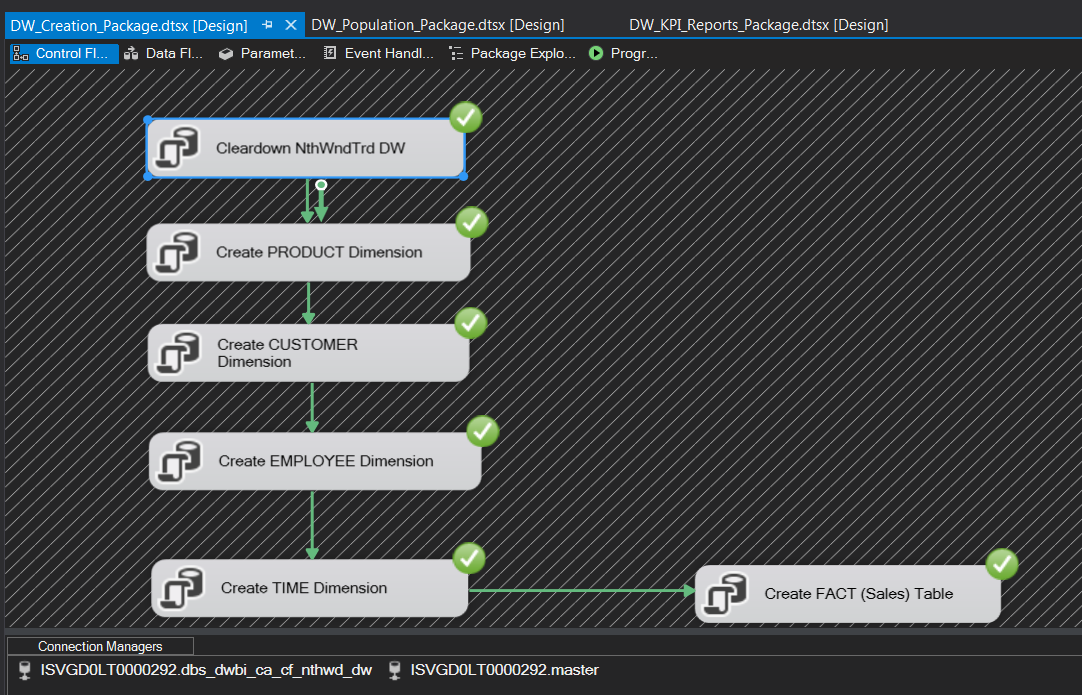
*Employee (Seller) variance from yearly Seller average*



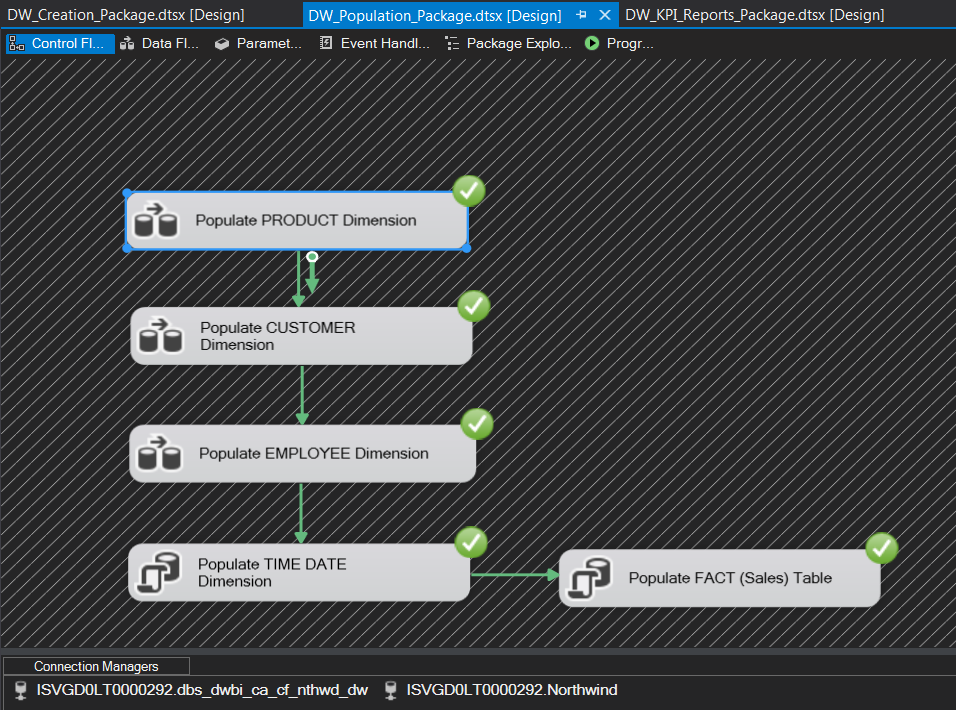
### Executing the SSIS Project Packages

Once completed, my SSIS Packages can be run in sequence to set up and populate the Data Warehouse with the Sales performance data on which I report about later on.

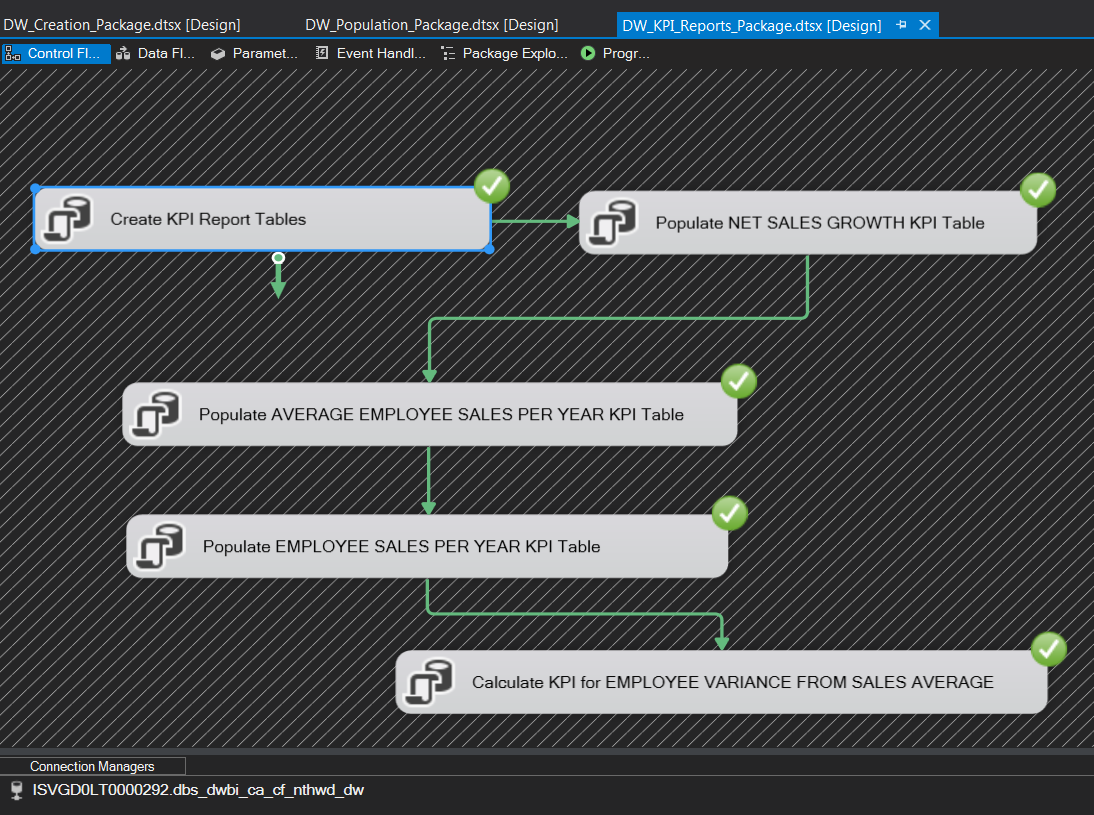
*Creation*



*Population*



*Report Preparation*



## Why use this Microsoft solution (SSIS)?

SSIS stands for *SQL Server Integration Services* and was introduced by Microsoft as a tool with the release of SQL 2005.

This project was my first exposure to working with using SSIS for ETL style work, but I have had industry experience in the past working with Transact- SQL (T-SQL).

Taking both prior career experience and knowledge learned from this project, I consider that SSIS was an excellent choice for this project.

The rationale for my recommendation of SSIS can be summarised in the following points;

* **Debugging is greatly simplified.** The ETL process for this Northwind Traders project was split across three Packages and further separated within Data Flows in the Control Flow for each Package. In Section 4.1.4 there is a visual representation of the execution flow. If one Data Flow failed it would be much easier to isolate and correct. It possible to set up error tracking when using just T-SQL but SSIS provides for sophisticated error handling as part of built in functionality.
* **Workflows are built visually in SSIS and are easier to maintain.** The SSIS designer provides a graphical view of the sequence of work being done. It is therefore easier to understand and adapt.
* **Integration with Visual Studio**. In DBS Visual Studio is the IDE of choice and SSIS is an extension which tightly integrates into this development environment and connectivity to SQL Server was relatively straightforward. It is always easier to manage a development project within and IDE. Although I did not use any source control repositories on this project, it would have been more straightforward to store my SSIS work in a platform like GitHub (as opposed to managing a collection of SQL Scripts).
* **Avoids the need for SQL to build Product, Customer, and Employee dimension tables.** I have used my knowledge of SQL to build a number of SQL Execution Blocks in SSIS but, apart from TimeDate, I was able to use the tools in SSIS to manage extraction and merging of data into the Data Warehouse dimension table. This was an interesting learning experience and a good introduction to the benefits of using an extension like SSIS for ETL jobs.
* **Speed of execution.** It would have been possible to develop the ETL process only using SQL Scripts and run them through Microsoft SQL Server Management Studio but this would have been slower to execute. The process of running each script would have taken more time and lengthened the development/debugging period. It is possible to set up batch files to run the SQL scripts more quickly outside of an application but the graphical output would be more primitive.

## Comparison of SSIS with Azure Data Factory V2

In this section I am not going to make comparisons with a product from another vendor but rather look at a slightly different paradigm that Microsoft offer when it comes to ETL solutions.

Any business that has a need to move data between separate, and possibly disparate sources, is likely to be considering cloud based options, if they are not already doing so.

Azure Data Factory V2 (ADF) became generally available in mid-2018, and is an attractive consideration to any business that uses SSIS, or has a need to use a similar type of tool.

**On Premise vs. Cloud**

SSIS is a tool that has been around for many years and would traditionally be seen as an on premise tool, running within a company network infrastructure. However, recent updates have provided an ability to work with hybrid Azure/On-Prem and pure Azure solutions.

Azure Data Factory (ADF) has no On-Prem only offering. So if a company wants to update to this tool it will possibly require a strategic decision to invest in a broader Azure engagement.

**Data source and Destination Connectivity**

Both ADF and SSIS have the capability to connect to a wide variety of data sources and transform and load that data to different destinations.

ADF has a slightly more complicated approach to data source connection requiring a self-hosted gateway in the business to transfer data to Azure.

SSIS is a more established solution and the *connection manager* process it uses (as seen in this project) is relatively straightforward to set up. There are also a number of third party connection managers that a company can purchase and plug in to provide additional services. At the moment this gives SSIS a slight advantage in terms of flexibility.

**ETL or ELT**

Although this is somewhat of a simplification SSIS can be seen as a tool that performs ETL (Extract Transform and Load) very well, while ADF is better suited to ELT (Extract Load Transform).

ELT can be described as load first, and then call functions in the destination to transform the data. ADF V2 is making significant improvements in this area but SSIS was seen, until very recently, as a stronger tool for in-flight transformation.

**Scale**

If the data volumes with which a company is working with in its ETL processes is relatively small then SSIS may remain a better low cost option.

However, ADF is much more attuned to Big Data high volume challenges and will scale significantly better as the size of the data manipulation challenge grows.

SSIS is easier to implement and set up but an on-Prem solution may require Server tuning and other infrastructure optimisation as data volumes grow.

ADF is more complex to set up and will take more time to get up and running. It is however much better suited to higher data loads and may therefore be a more strategic long term solution.

Cloud bases solutions are inherently more elastic in terms of resource availability. Obtaining additional processing power through ADF for a new or ad-hoc ETL requirement will be generally much more straightforward than adding in-house resources to a local SSIS solution.

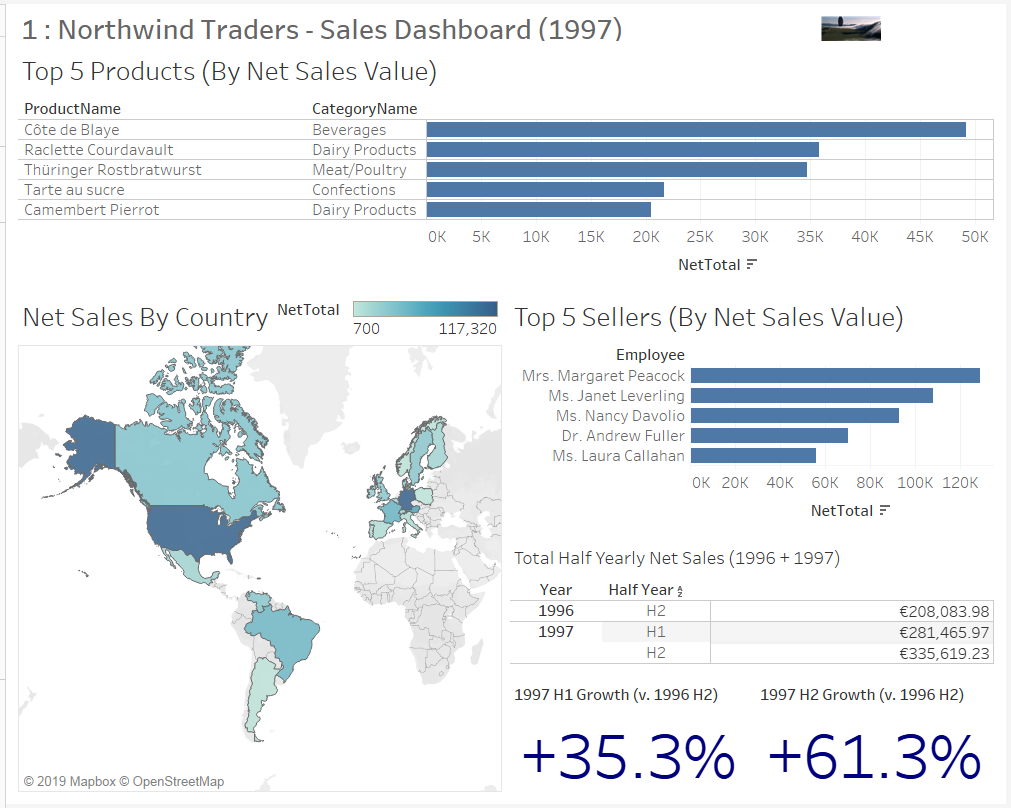
# Part 4 : Reporting and Analysis

## Dashboard Design

The Key Performance Indicators were defined in Section 0 of this document.

Two custom Tableau dashboards were designed to present this information to end users.

**Dashboard 1: Sales Dashboard for 1997**



The first dashboard is an ‘executive summary’ overview of one year (1997).

This dashboard was designed with the following key design principles in mind:

* There are six tiles with metrics, so we have kept to the principle of no more than 7 metrics per dashboard.
* The colour scheme has been kept fairly minimal by just using varying shades of blue. The intension is not to distract the user from the focus of the dashboard, which is the information it is presenting.
* Tableau allows for a dynamic drill down on the Sales Net Total data into a tabular form, which can be exported. This provides the end user with optional analysis that does not clutter the initial presentation of the dashboard.
* The heavy use of graphics in the dashboard is intended to be attractive and possibly prompt questions from the user that can be answered by a drill down into the individual Tableau worksheet, if required.
* Type is upper and lower case, and is clear and easy to read.
* Legends and supplementary text are avoided. The one legend that is used is there to provide upper and lower values for one graph.
* Percentage value tile are used to emphasise the other numerical data in the tile.

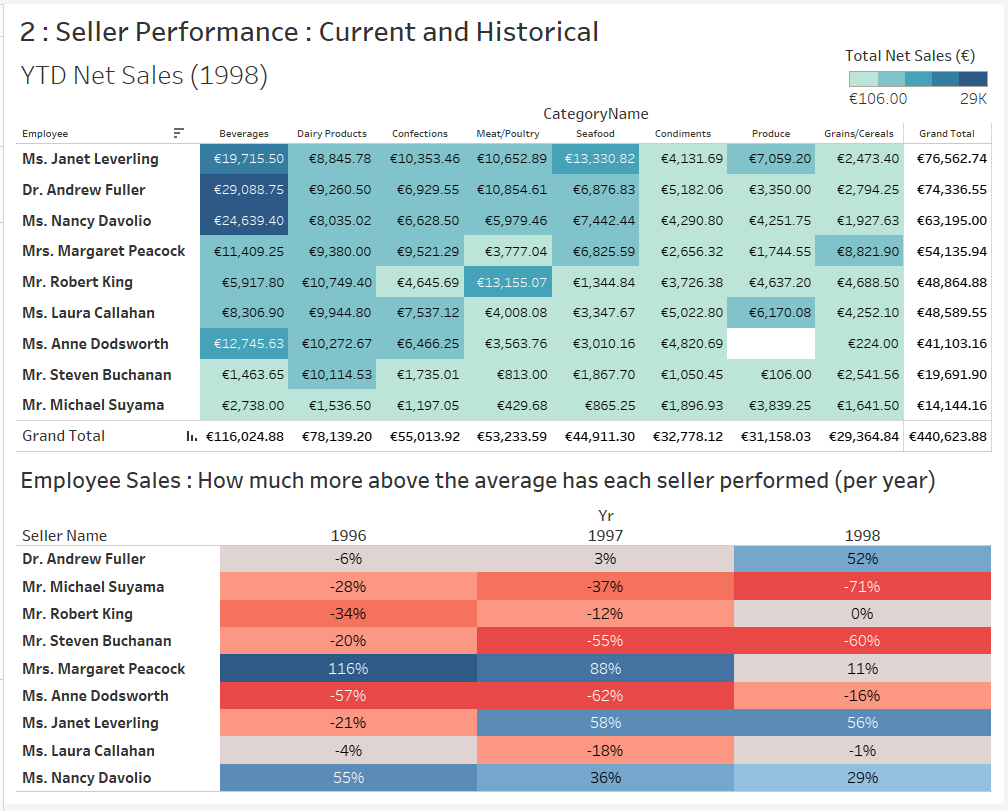
There is an overarching theme to this dashboard;

* **What** products sell the best?
* **Where** are the best markets?
* **Who** are the best sellers?
* **How** are we performing?

The layout of the tiles in this dashboard are intended to follow a top-down, left to right structure to support the sequence of the themes listed above (in that sequence).

The information in the dashboard is very high level but it intended as an ‘at a glance’ overview for Senior Management. The message is clear and focuses on the key successes of 1997.

**Dashboard 2: Seller Performance – 1998 and Historical**



The second dashboard is a more detailed look at 1998 sales performance to date and placing it in context with an historical overview of seller activity.

This dashboard was designed with the following key design principles in mind:

* A table is the best way to represent volumes of numerical data.
* A table will also work well when there is a need to make localised comparisons with a number of dimensions.
* Colour mapping in tables can be used to draw a user’s attention to key patterns in the data.
* Declutter the dashboard as much as possible. Let the numbers carry the message and avoid unnecessary text and legends.

### Dashboard 1: 1997 - Design

The layout of Dashboard 1 consists of 6 separate tiles, representing separate KPIs.

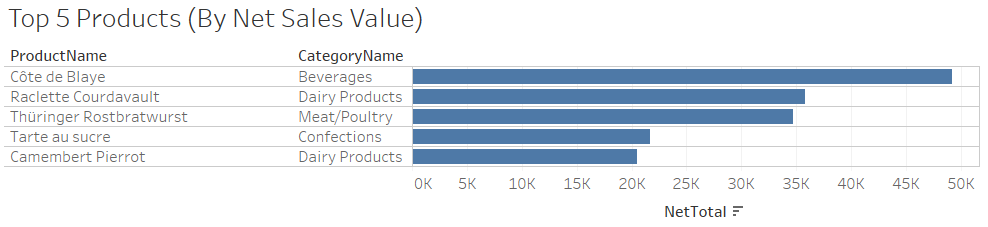
|  |  |
| --- | --- |
| **Dashboard Component** | **KPI** |
| **Tile 1** | Top 5 Product (By Net Value) |
| **Tile 2** | Net Sales By Country |
| **Tile 3** | Top 5 Sellers (By Net Sales Value) |
| **Tile 4** | Total Half Yearly Net Sales (1996 + 1997) |
| **Tile 5** | 1997 H1 Growth (versus 1996 H2 figure) |
| **Tile 6** | 1997 H2 Growth (versus 1996 H2 figure) |

The graphical elements of this dashboard are created by Tableau worksheets reading directly from the Data Warehouse tables.

The table and percentage data is generated by SQL statements invoked as part of the SSIS Data Warehouse generation process as described in Section 4.1 of this document. It was found to be simpler to carry out the more complex analysis Net Sales data in SQL scripts and then read the result set directly into Tableau for presentation.

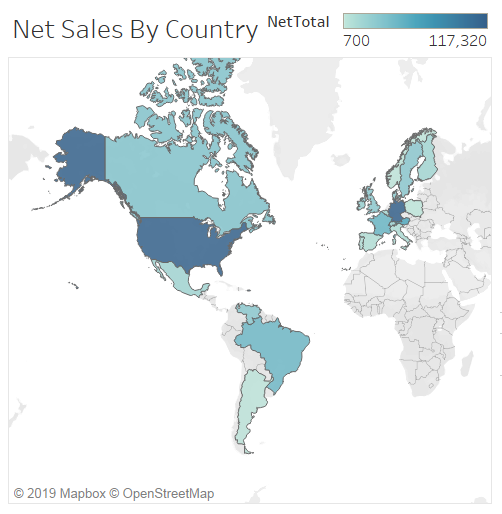
Unless where indicated, the data in this dashboard has used the information in the TIME dimension table to focus only on Net Sales activity in 1997.

**Tile 1: Top 5 Products**



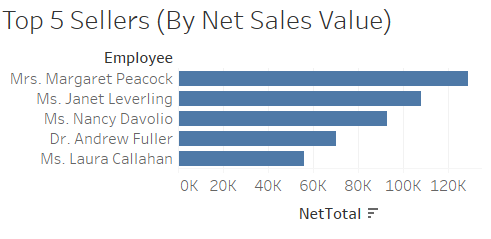
* The ‘Top 5 Products’ tile follows a ‘Golden Rectangle’ dimension as it is approximately 50% wider than tall. That dimension suits the graphic as is allows for an easy comparison of net sales amount per product.

**Tile 2: Net Sales By Country**



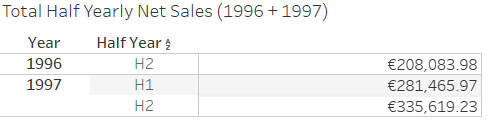
* The ‘Net Sales By Country’ tile uses a graduated scale of blue colours to show the countries into which Northwind Traders sell. The varying shades of blue show the varying quantity of net sales. This is more effective in showing a natural hierarchy of where sales are highest. It is also easier to read for someone with challenges in colour perception.
* The shape of the ‘Net Sales By Country’ tile is a better dimension for showing geographical data. Stretching the tile across the entire length of the dashboard would have flattened the globe. Emphasising the vertical dimension is a better fit for the graphic.

**Tile 3: Top 5 Sellers**



* The ‘Top 5 Sellers’ tile follows the clear principle of ‘don’t make me think’. It is a simple representation of top seller performance.

**Tile 4: Total Half Yearly Net Sales (1997 + 1998)**



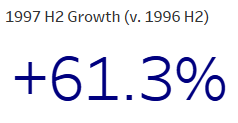
* Using a table in the ‘Total Half Yearly Net Sales (19967 + 1997)’ tile is a more effective way to display this numerical data and it provides the background the percentage tiles below.

**Tile 5: 1997 H1 Growth (v. 1996 H2)**



* Percentage values compliment other graphical and tabular data in this spreadsheet. A ‘+’ has been added to emphasis the performance success. The font in the percentage tile has been increased to draw user attention

**Tile 6: 1997 H2 Growth (v. 1996 H2)**



* The font in this percentage tile has also been increased to draw user attention. It would be possible to add conditional formatting to change the colour from dark blue to red if the metric was a minus value, but that was not deemed necessary for this dashboard.

### Dashboard 2: 1998 and Historical Design

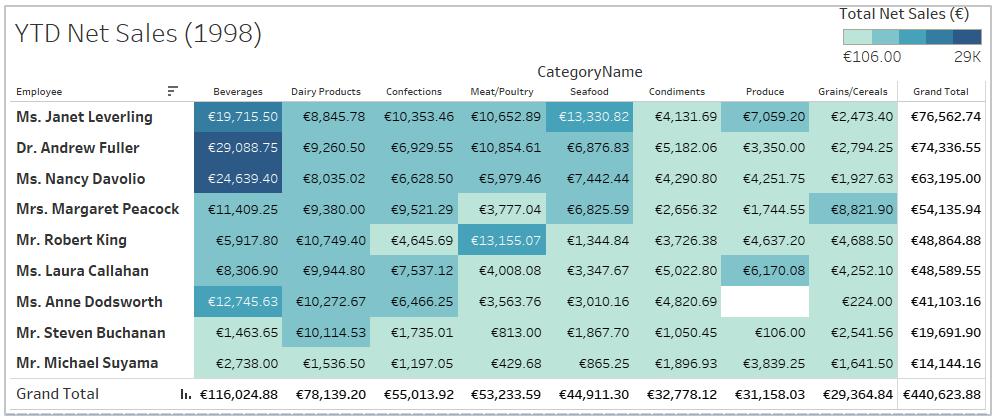
The layout of Dashboard 2 consists of 2 separate tiles, generating two tables of Sales information.

|  |  |
| --- | --- |
| **Dashboard Component** | **KPI** |
| **Tile 1** | Seller Performance : current and Historical |
| **Tile 2** | Employee Sales : Compare Seller to Yearly Average |

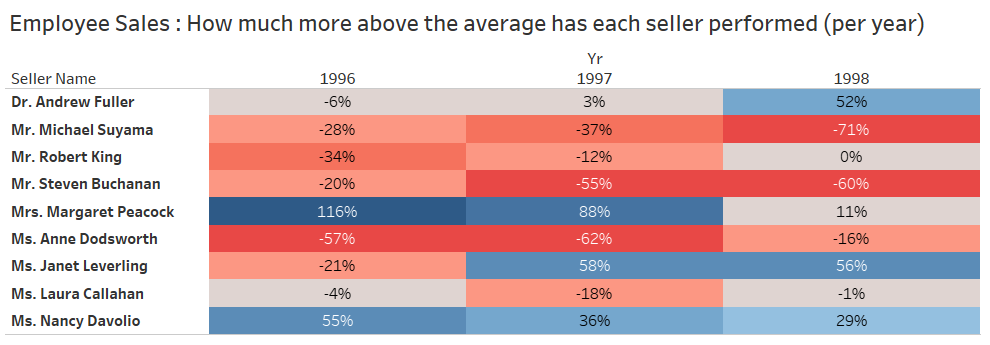
The first table was generated in Tableau by reading the Data Warehouse Dimension and Fact tables directly, and then graphically working the information into a Highlight table.

The second table involved a more complex analysis of the Data Warehouse information. SQL scripts were written to populate additional database tables dedicated to storing the output of the Sales data queries. Tableau then read these datasets directly and formatted the information into another highlight table.

**Tile 1: Seller Performance by Category (1998 TYD)**



* This is a blue colour coded Highlight map. The colours are kept within varying shades of blue to avoid difficulties for any user with colour deficiency.
* The colour differences are also in steps, not a gradual spectrum, to highlight the difference in Net Sales amount more clearly.
* Higher values are in a deeper value of blue. The table is also sorted in order of Net Sales by seller (highest at top). This draws the user to the cluster of colours around who is the best performer and with which product category.
* There is a general ‘Golden Rectangle’ dimension to the table to emphasise the spread of date as the user looks at the length of product categories sold by a seller.

**Tile 2: Employee Sales: How much more above average has each seller performed (per year).**

* This data is read directly from a report table in the Northwind Traders Data Warehouse. This is because the logic to generate the information would be complex to do in Tableau but was relatively straightforward to collate using SQL scripts.
* This table is less dense in terms of data than Tile 1 in the dashboard but it does break some of the other rules used that we have applied elsewhere in the two dashboards.
  + There is a mixture of colours. A range from red to blue was chosen because the mid-point of the colour range, which is type of beige colour, represents a **0%** value.
  + The 0% centre point represents the fact that the seller is at the average Net Sales level for that year.
  + The shades of red therefore represent the percentage by which the seller has fallen beneath the Net Sales average for the year. Conversely, the deeper the blue the better the seller has performed against that year’s average.
  + It is not exactly a ‘do not make me think’ table. There is an element of interpretation required by the user when reading this graph but the information is more detailed and sophisticated than elsewhere in the dashboard, hence the slightly increased level of complexity has been allowed.
* The ordering of the seller names is different in this table than the one in Tile 1. Tile 1 will dynamically change throughout 1998 as Net Sales values grow but there is no need to alter the name sequence in this table. It will therefore be consistent when reused in a 1999 dashboard.

## Dashboard Analysis

In Section 5.1of this document the layout of the dashboard was described. In this section we will look in more detail at the actual KPI data extracted from our Northwind Data Warehouse.

### Dashboard 1: 1997 Data Analysis

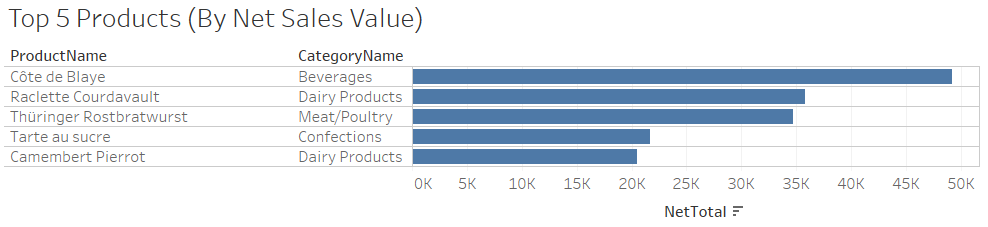
The purpose of this dashboard is to provide an analysis of Net Sales Performance in 1997, with some context to the last half year in 1996.

The KPIs produced can be used by Senior Management in assessing the kind of decision making that should take place in the context of some of the 1998 KPI data in the second dashboard.

**KPI 1: The Top 5 Products**

This metric informs the Sales team about what products are selling well, and into which category they fall.

It could help Senior Management assess where future production efforts can be diverted to maximise profits.



In 1997, Côte de Blaye was the top seller, with nearly 50K in Net Sales. It would be expected that this type of yearly report is generated each year, so that Senior Management could carry out a year by year comparison of the product.

Regardless, the second dashboard shows that ‘Beverages’ is the highest selling product category by some margin, which is the category in which Côte de Blaye belongs.

This may help Senior Management in terms of future investment in the more profitable product categories, such as beverages.

**KPI 2: Net Sales By Country**

Where are we selling the most products? This graphic shows where Net Sales were highest in 1997.

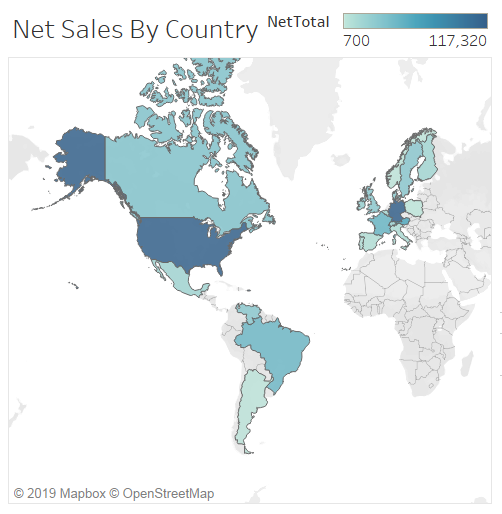


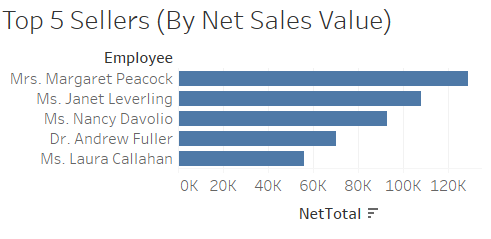
Tableau allows the user to hover over individual countries and look at Net Sales amounts. However, even by visual inspection it is clear that the United States and Germany are the largest markets.

Senior Management can use this data to look at the profile of the sales team in each region. Further analysis might encourage extra sales activities in these profitable regions. The fact that sales business seems reasonably strong in Canada and France might support the case for further Sales team investment in Western and Eastern regions.

**KPI 3: Top 5 Sellers (By Net Sales Value)**

Where sold the most in 1997? This graphic shows which seller sold the most in 1997.

This metric could be used to confirm that these were the employees most rewarded and/or acknowledged at the yearend Sales review in 1997.



The tabular data on Sales performance is particularly useful in context with the second dashboard.

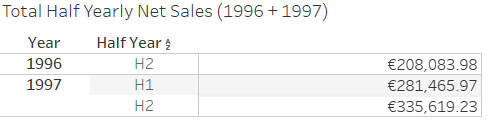
As an example, Mrs Margaret Peacock was the best seller in 1997, making approximately just over 128K in Net Sales that year. However, the second dashboard shows that performance at the end of H1 2018 has dropped somewhat as compared to her peers.

Perhaps Senior Management should reach out to see if Margaret needs additional support?

**KPI 4: Total Half Yearly Net Sales (1996 + 1997)**

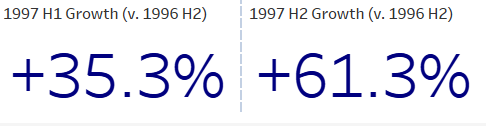
The Data Warehouse for Northwind Traders contains Sales information from H2 1996 through to H2 1998.

Given that we have a full year’s worth of data for 1997, we have a means to see how Net Sales grew that year compared to the data we have on 1996.



This is a simple enough metric for Senior Management. Are making more Net Sales now than we did in the past? Is the business growing?

**KPI 5 and 6: Percentage Growth in Net Sales**



Clearly business is performing well. Net Sales grew by over 35% in the first half of 1997, as compared to the previous six months. The second half of 1997 was even better (over 60% better than the same time in the previous year).

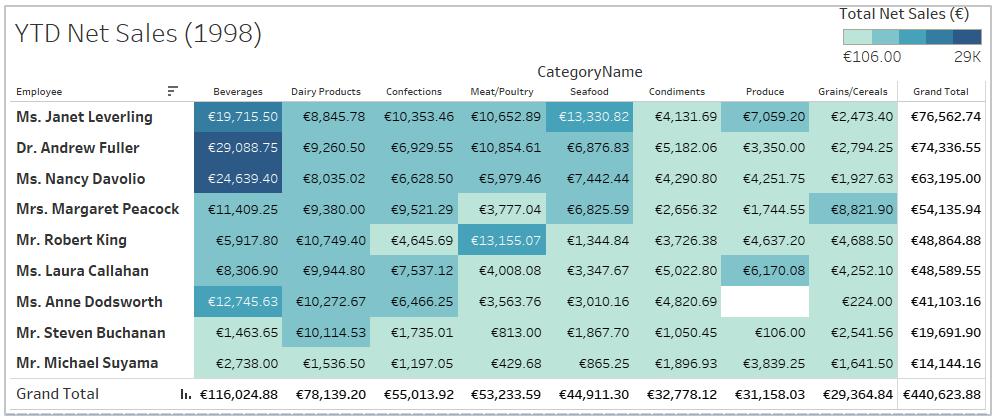
This can be seen as a vindication of Senior Management strategy, at least in terms of Net Sales and sets a strong benchmark for future years.

### Dashboard 2: 1998 and Historical Data Analysis

The purpose of this dashboard is to provide an analysis of Net Sales Performance in 1997, with some context to the last half year in 1996.

**KPI 1: YTD Net Sales (1998)**

This metric informs the Sales team about who is performing best amongst the sellers in the H1 1998 period (Q1 and Q2)



Senior Management can assess if poor performers need support or, alternatively if strong performers should be recognised.

For example, Ms Laura Callahan has dropped out of the top fiver seller list but as it is only by a marginal amount we can probably assume that this is not significant.

It appears that the 1997 top performers are on track to repeat that achievement in 1998. Senior Management would need more contexts but it does appear that Seller performance is consistent so far in 1998 with the metrics seen for 1997.

It also provides a view of what product categories are doing well in 1998 so far.

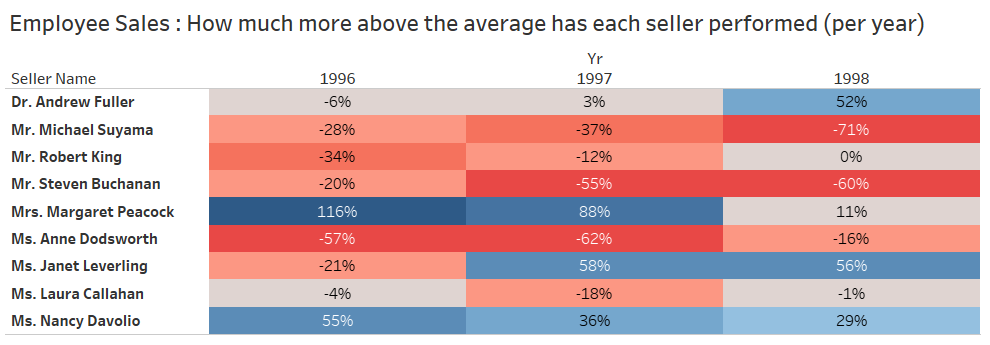
Beverages are a clear top seller, with Condiments, Produce, and Grains/Cereal consistently selling less across most of the sellers with one or two exceptions.

Senior Management may consider this for the remainder of 1998 and the future;

* Should we discontinue Condiments, Produce, and Grains/Cereal and invest further in Beverages?
* Alternatively, should we look at why Ms. Margaret Peacock is able to perform reasonably well with Grains/Cereals, and Ms, Laura Callahan do likewise with Produce? Are there lessons to be learned in their sales approach in their regions?

**KPI 2: Performance of each seller against the yearly Net Sales average (historical)**

This is a more complex set of KPIs and provides a view across the entire available timeline in the Northwind Traders Data Warehouse of how sellers are performing compared to their peer.



The data is generated through SQL scripts following this logic:

1. Calculate the average Net Sales amount for each year. This is the average across the nine employee sellers (not the average Net Sale per transaction).
2. For 1996, 1997, and 1998 we use these three figures (Net Sales average) as a baseline for each seller.
3. For each year and each seller the SQL script compares the total sales for each seller and calculates the percentage difference above or below the average for the year.
4. A table in the Date Warehouse is populated with this Seller/Year/Average Difference data, which is then read directly by Tableau.

This table allows Senior Management to look at the performance profile for each seller from 1996 through to 1998.

For example;

* Dr. Andrew Fuller performed slightly below average in 1996, in terms of his Net Sales.
* He was 6% underneath the yearly average for his peers in 1996 for Net Sales. In 1997 performance improved slightly but it placed him just 3% over the average Net Sales mark for all sellers that year.
* However, in 1998 performance to date is much better. Year to date in 1998, at the end of Q2, he is 52% above the average Net Sales amount set by his peers.
* Is it worth Senior Management understanding more about the reason for Andrew Fullers improvement? Is this something that could be replicated with other sellers?

It is worth noting that this metric is a relative comparison of the sellers against their own performances. It is a metric to see who is performing well in relation to the other sellers. I theory, all sellers could have a bad year in terms of actual Net Sales but still perform well against this KPI.

# Appendices

## Appendix 1 – SQL Scripts to build the Data Warehouse

In the SSIS Package used to create the Data Warehouse tables, the SQL is broken down into individual CREATE TABLE segments.

For simplicity, one single SQL file is included here in this section of the appendix to show the SQL commands used to create the tables.



I wrote these SQL scripts first to validate the schema and then later verify the SSIS data Warehouse creation process.

The SQL files are also included in the .ZIP file as part of the overall CA submission.

Double clicking on the icon above should launch the file into Microsoft SQL Server Management Studio, or whatever application has an association with ***\*.sql*** files on the user system.

## Appendix 2 – SQL Scripts to Populate Data Warehouse Dimensions

The Dimension tables in the Data Warehouse are populated in the SSIS Package by executing the SQL commands in the attached \*.SQL file.

Again, for simplicity, although the SSIS packages the creation of the dimension tables into separate Data Flow actions, I have just included one file in this appendix with the full set of INSERT commands (and other logic).



Double clicking on the icon above should launch the file into Microsoft SQL Server Management Studio, or whatever application has an association with ***\*.sql*** files on the user system.

## Appendix 3 – SQL Scripts To Populate Data Warehouse Fact Table

The Fact table in the Data Warehouse is populated in the SSIS Package by executing the SQL commands in the attached \*.SQL file.



Double clicking on the icon above should launch the file into Microsoft SQL Server Management Studio, or whatever application has an association with ***\*.sql*** files on the user system.

## Appendix 4 – SQL Scripts To Create/Populate KPI Tables

The KPI report tables and metrics data in the Data Warehouse is populated in the SSIS Package by executing the SQL commands in the attached \*.SQL files.





Double clicking on the icon above should launch the file into Microsoft SQL Server Management Studio, or whatever application has an association with ***\*.sql*** files on the user system.

# References

## Data Warehouse Design

In additional to the class Moodle notes I followed the data warehouse design principles that were discussed in these two YouTube training videos;

***Designing Your Data Warehouse from the Ground Up* -** <https://youtu.be/patBYUGwsHE>

***Implementing a Data Warehouse with SQL Server, 01, Design and Implement Dimensions and Fact Tables* -** <https://youtu.be/StoWu2A8Ufs>

## SSIS versus ADF

The article from which I took most of the comparison facts between SQL Server Integration Services (SSIS) and Azure Data Factory V2 (ADF) is located here;

[***https://www.blue-granite.com/blog/the-right-tool-for-the-job-azure-data-factory-v2-vs.-integration-services***](https://www.blue-granite.com/blog/the-right-tool-for-the-job-azure-data-factory-v2-vs.-integration-services)