Data Warehouse Project

Introduction

The purpose of Business Intelligence is to allow the business to collect, analyze and use data in support of the various decision making and managing departments. The data is transformed into Information so that it can be used for tactical and strategic decisions.

In our assignment we focus on the Northwind database, a fictitious company that imports and exports food. The database contains data related to employees, customers, suppliers, distributors, products information and inventory, sales, purchases, and invoices of the company.

Our goal is to collect data from the Database, analyze it, import it into a Data Mart and finally transform it into information useful for the management.

Generally the transformation of data from the source database to the data mart can be done in different ways. The main approaches are either to import the data directly from one to another by using a series of SQL commands, or by extracting the data from the database using an ETL process. The first approach seems to be faster, however the second is more realistic. In the real world, generally the companies do not give full access to their own data warehouse. In most of the cases they prefer to give partial access and limited in time, only for the purpose of the collection of the data necessary for the analysis. For this reason we opted for the second way, using Microsoft Visual Studio for the staging and loading phases.

RDBMS, relational database management system

When choosing the tools, we had to choose which relational database management system (RDBMS) to use. A study provided during the course, called "Magic Quadrant for Data Management Solutions for Analytics", Published: 20 February 2017, also available online, provides a set of strengths and cautions for 21 players on the market.

These players are divided in 4 groups: Leaders, Challengers, Niche and Visionaris. The challengers include big companies like Google and Hewlett Packard Enterprise, the niche, also a big group, include for example Huawei. The group of leaders is composed by Microsoft, Teradata, Oracle, Amazon, Ibm and SAP.

Moreover, an online research, done through various websites, among them the www.educba.com, tells us that the most used database systems are Oracle and SQL and various sources analyse the difference between these two main players.

According to some of this sources, Microsoft SQL is faster when it comes to writing the queries and to run and it is easier than Oracle.

On the other hand, regarding security and data corruption, the sources analyzed suggest that Oracle is far better than the competitor. Oracle is more suitable for really large applications whereas Microsoft: SQL is easier to work with.

Considered that we are are using the database for learning purposes and that our priority is the easiness of use and processing speed, we decided to use Microsoft SQL for its simplicity and for being the suitable for small application or for studies.

Balanced Scorecard

The strategy execution tool is the Balanced Scorecard BSC, that helps companies to clarify strategy, monitor progress and manage action plans. It focuses on 4 KPIs: financial, customer, internal business process and learning and growth perspective.

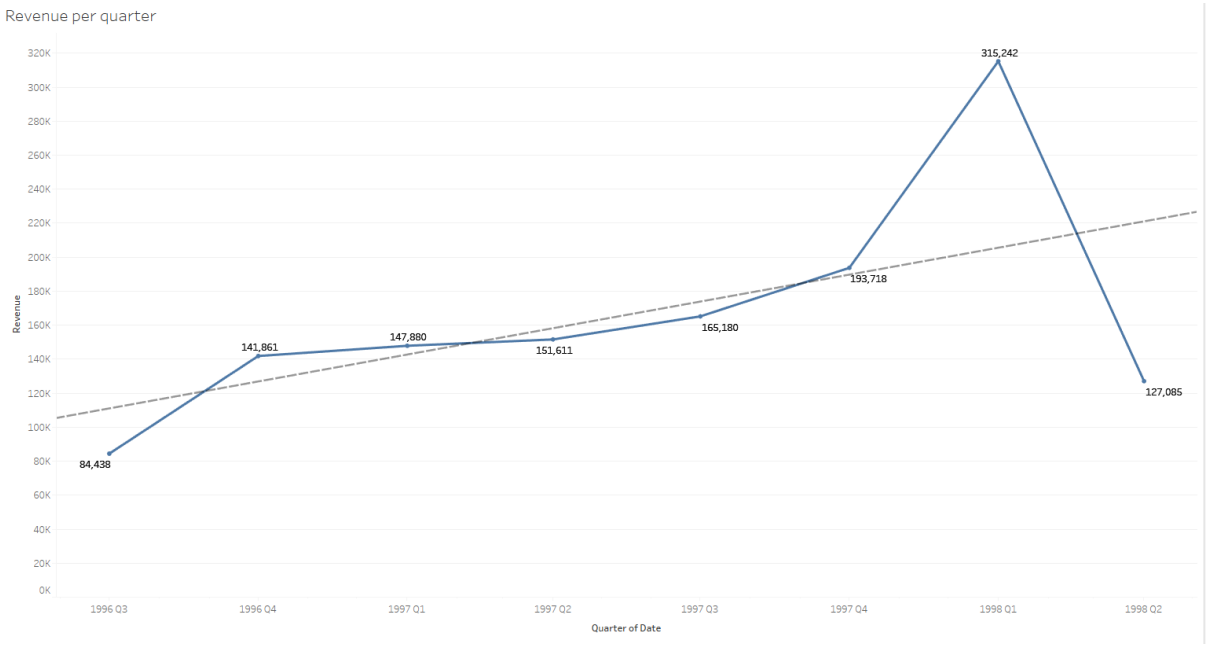
- For our financial KPI we focussed on the general growth of the company, analysing and visualizing the data divided per quarter and per region. The benefit of this is that the financial department that uses this information to verify the general financial health of the company will be able to do so in greater detail and will be able to use this information to increase future performance in both these regards .
- For the customer KPI we focussed on the Customer Retention Rate. It gives an idea on successful the company is in retaining the existing customers. For a company this rate is a sign not only of “how well we are doing”, but it also gives signs that other factors, internal and external, are changing. For example it might tell the company that analysis on the market is required, to see if new competitors or new products entered the market or if the prices are not in line with trends specific to different countries.
- Internal business KPI: this refers to the ratio of orders that are shipped on time. This analysis is directed to the department or the management that deals with logistics. It is important for a company to notice if there is a certain frequency or seasonality for the delays so that an action plan can be implemented and delays can be reduced.
- Revenue per Employee KPI - This information useful in monitoring the performances of the employees, identifying on a regular basis if there is any need for training, as well as target-setting and implementation of bonuses.

- **FINANCIAL PERSPECTIVE KPI`s**

KPI Revenue per quarter

As outlined in the Bernard Marr`s book: “Key Performance Indicators - the 75+ measures every manager needs to know” Revenue (also referred to as turnover or sales) is simply the income that a company receives from its normal business activities, usually the sale of goods and or services.

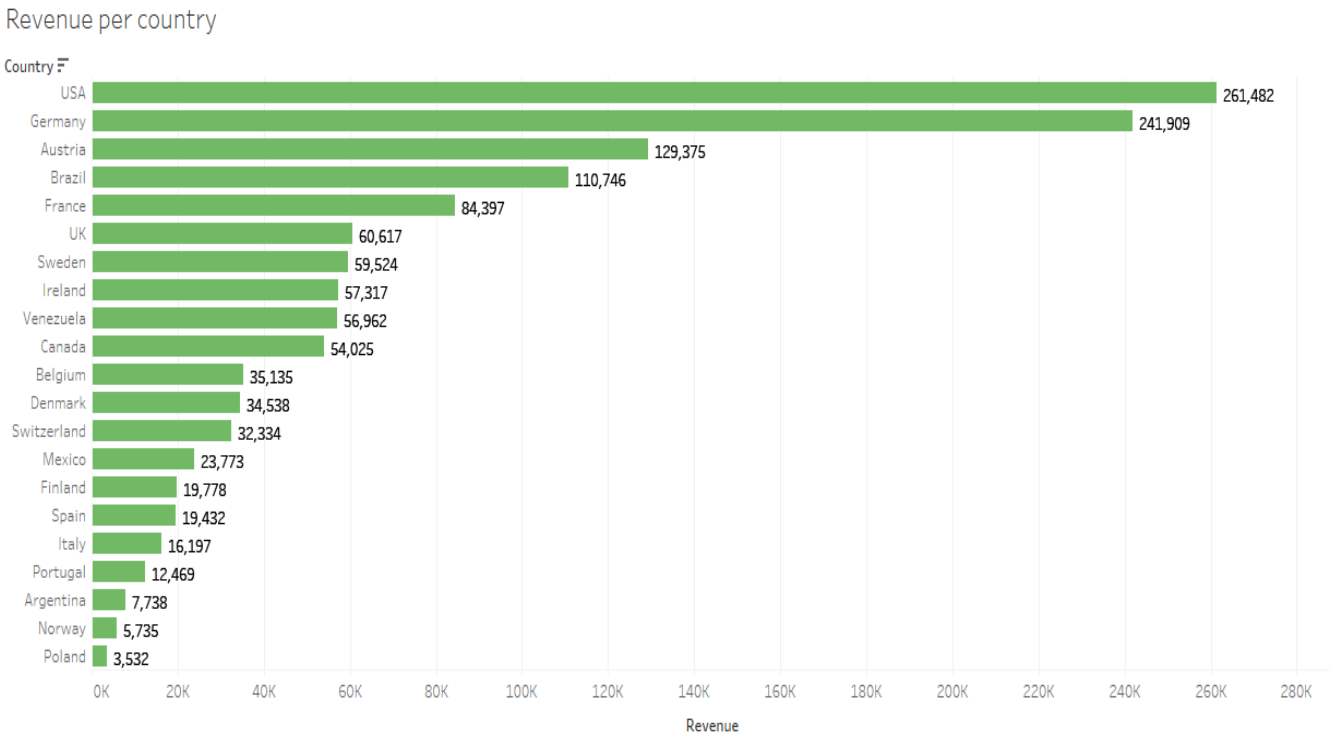
Total Revenue = Price of goods or services x Quantity sold



The analysis is done in tableau and this graphic shows the evolution of the quarterly revenue. As we can see, the revenue over the 8 quarters analized is not a linear one.

The quarters with the highest revenue are 97Q1 (147k) and 98Q1 (316K) ,which spikes very abruptly above the average. As both of the best quarters are in Q1 we can suspect a seasonality in the business. Though there are high differences between the revenues, the trend line shows a gradual increase in our revenue over time

In terms of the total revenue per country, USA with a total over 261k, followed by Germany 241k and Austria 129k are the countries that register the biggest revenue for this dataset. At the other extreme are Argentina, Norway and Poland with revenues under 8k

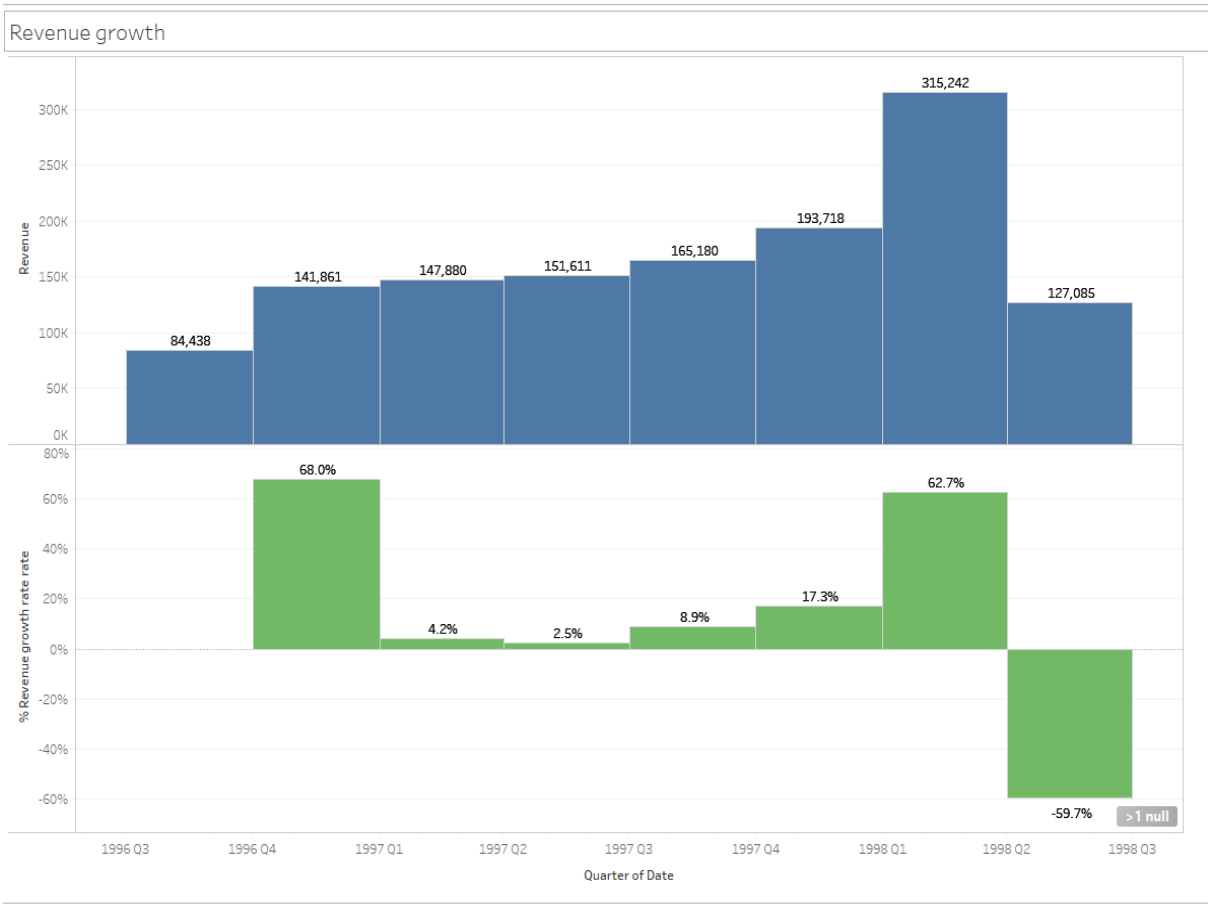


KPI Revenue Growth Rate

Revenue Growth Rate is an indicator of how well a company is able to grow its sales revenue over a given time period. While the revenue is an actual number, the revenue growth rates simply compares the current sales figures (total revenue) with a previous period (typically quarter to quarter or year to year).

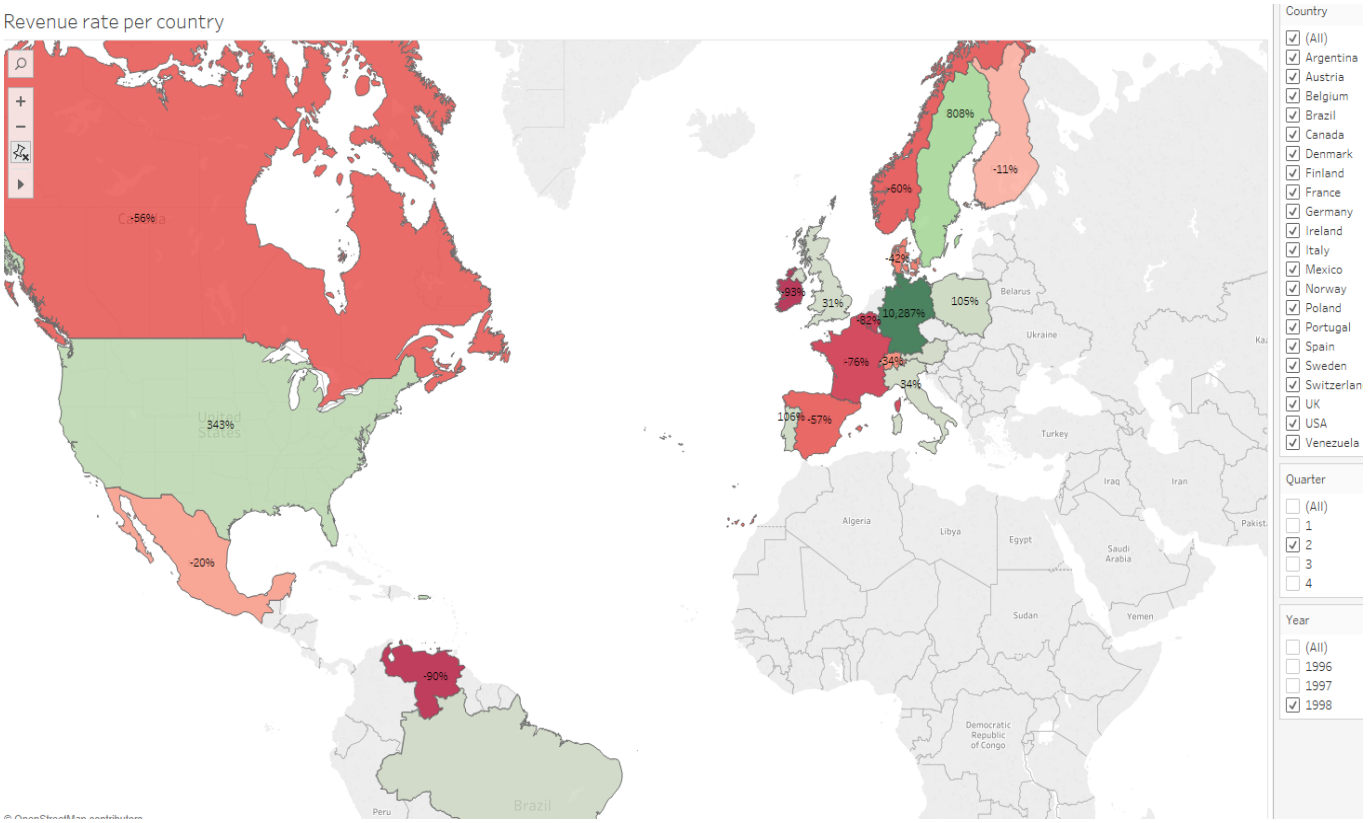
Revenue Growth Rate = Revenue this period (quarter) / revenue previous period (quarter)

And if we multiply it by 100 we get the percentage revenue growth rate



This split histogram shows the total quarterly revenue registered on top and the percentual Revenue Growth Rate on the bottom. The biggest revenue growth rate is registered 96Q4 (68%) and in 98Q1 (62.7%). The big spike in 98Q1 is followed by the biggest decrease in revenue growth rate to -59.7% in 98Q2

The next graph shows the evolution of the revenue growth rate per countries.



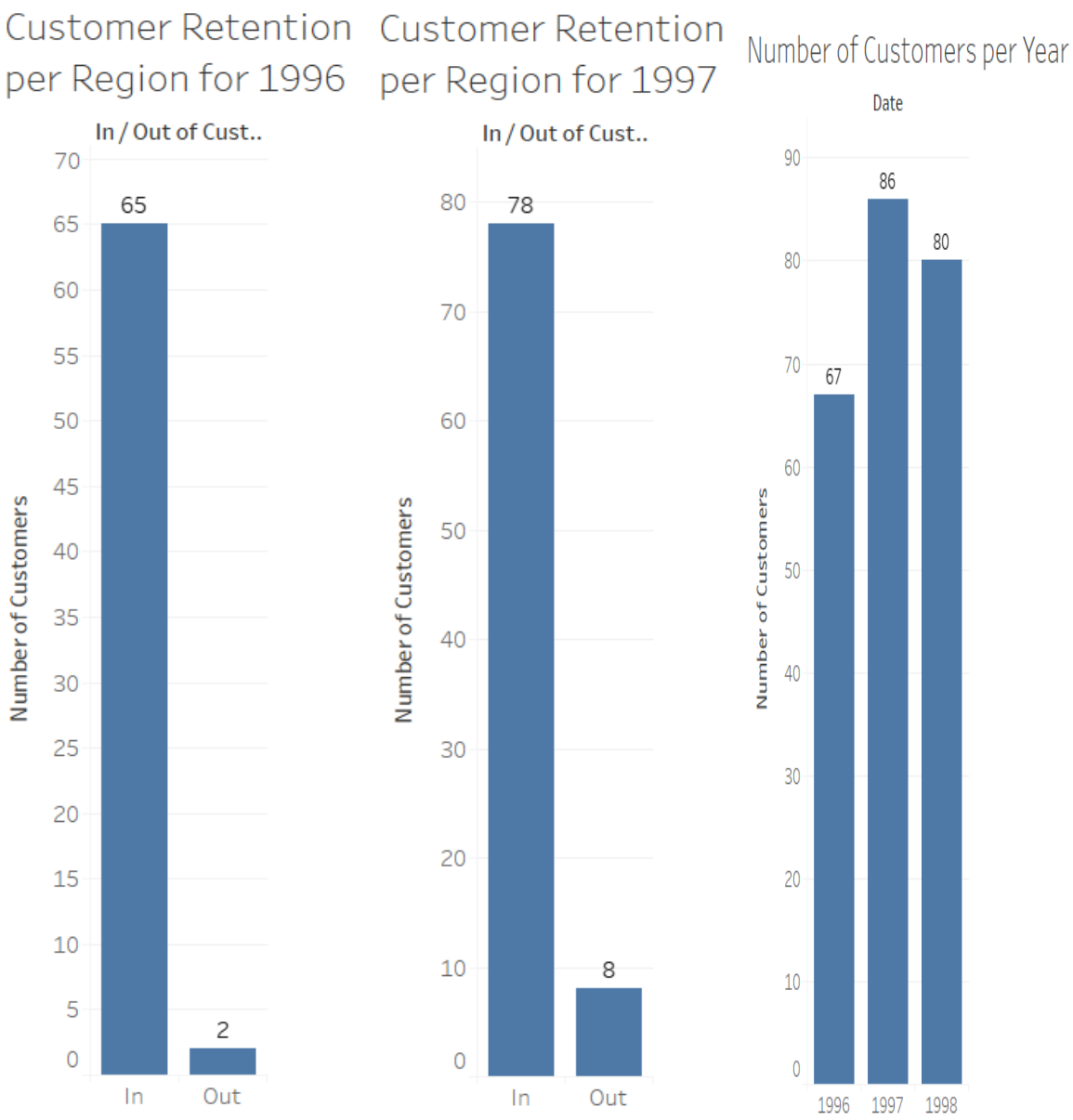
By filtering the tableau generated map, we can see the evolution of revenue growth for each specific quarter and country. This graph highlights where the biggest increases/decreases of the revenue growth occur geographically.

For instance Germany has an increased revenue in 98Q2 of 10,287% which means that the revenue increased over 100 times since the previous quarter. There is also a 808% percent increase in revenue rate for Sweden from the previous quarter. The regions with a negative revenue growth rate are in this case Venezuela witn -90% and Ireland with -93%.

Using the animation feature in tableau, I created a dynamic map that shows this evolution over the 8 quarters.

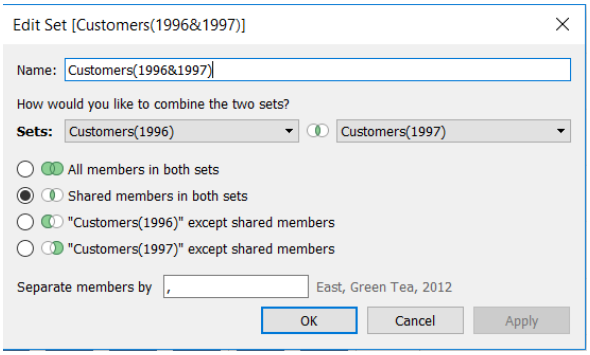
Customer Retention Rate - As outlined by Bernard Marr in his top 25 KPIs, How many of your customers are going to come back for more? And how loyal are they to your brand, organization or service? Due to the data at our disposal we will answer the first question posed, How many customers are going to come back for more? This KPI will be calculated by the number of customers

at the end of a period without counting the number of new customer acquired.



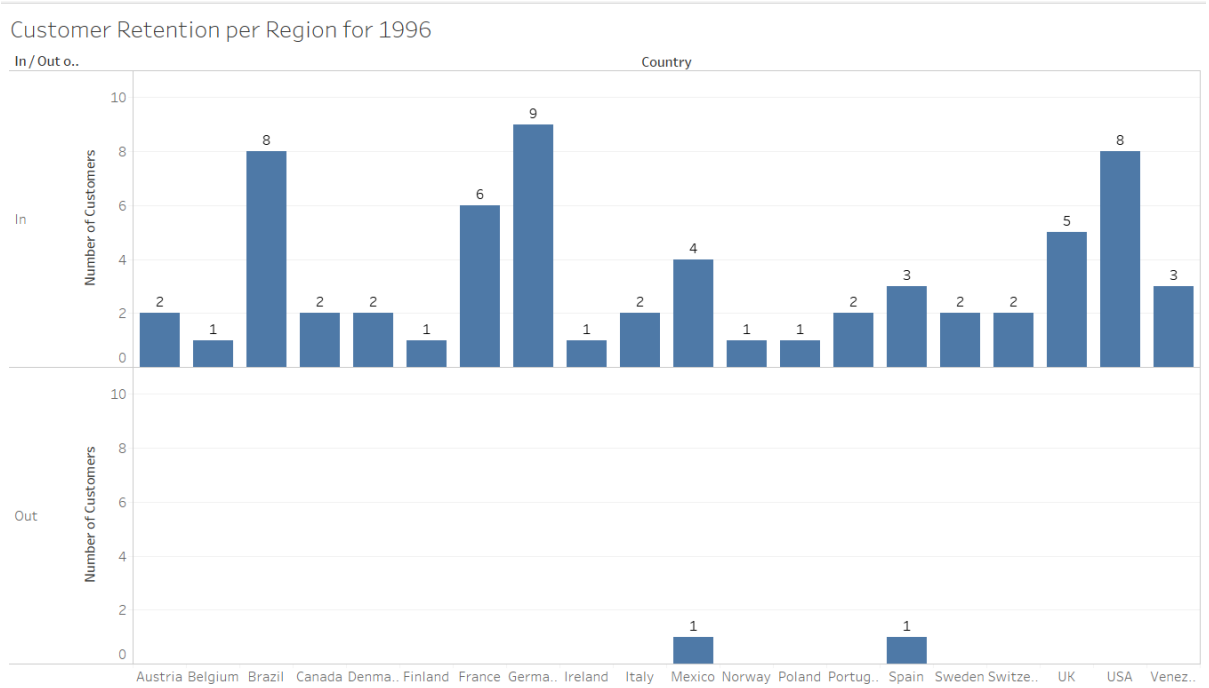
The graphs above show customer retention from 1996 to the following year, 1997 to the following year and the amount of unique customers for 1996, 1997 and 1998. From the above information we can calculate that the percentage customers retained from 1996 is equal to 97% and for 1997 it is 90%.

This information was calculated using the 'set' property in Tableau as shown below. Much like the way a join works in SQL we were able to join the customers based on order date from 1996 and 1997 and then again from 1997 and 1998.

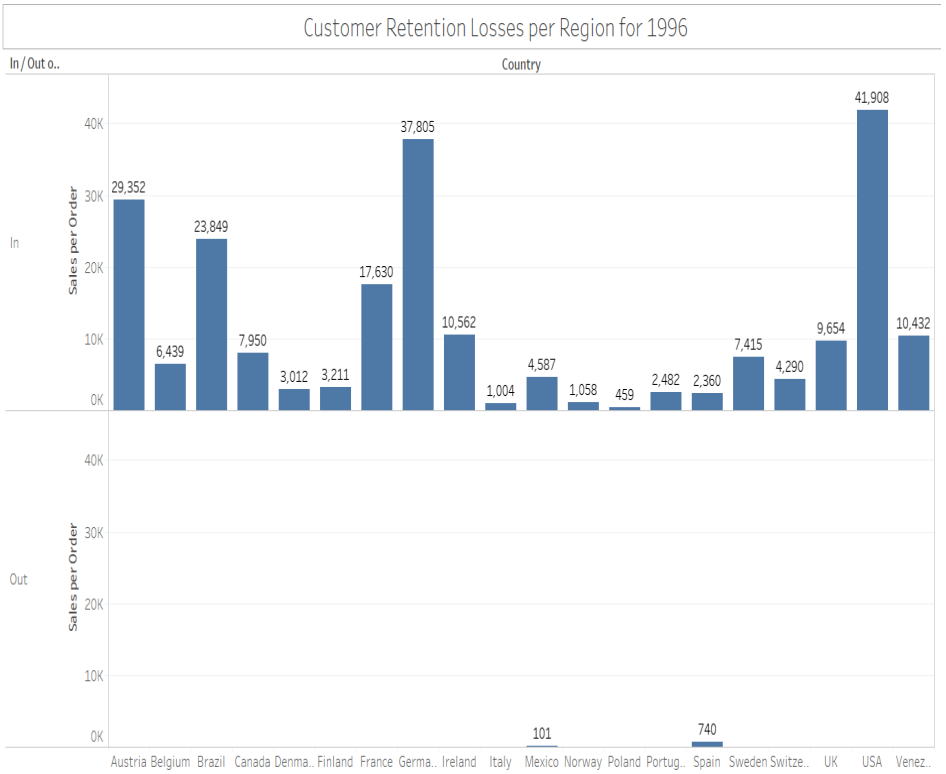


Creating a set in Tableau.

In sticking with our one of our main business drivers which is analysis of performance by region we would ideally drill down on the numbers given above to provide us with more information on how the customer retention rate is measured per region. From our initial graphs of customer retention we can see than from 1996 to 1997 we failed to retain 2 customers out of 67 who purchased from Northwind in 1996. Below gives us the regions where this occurred.

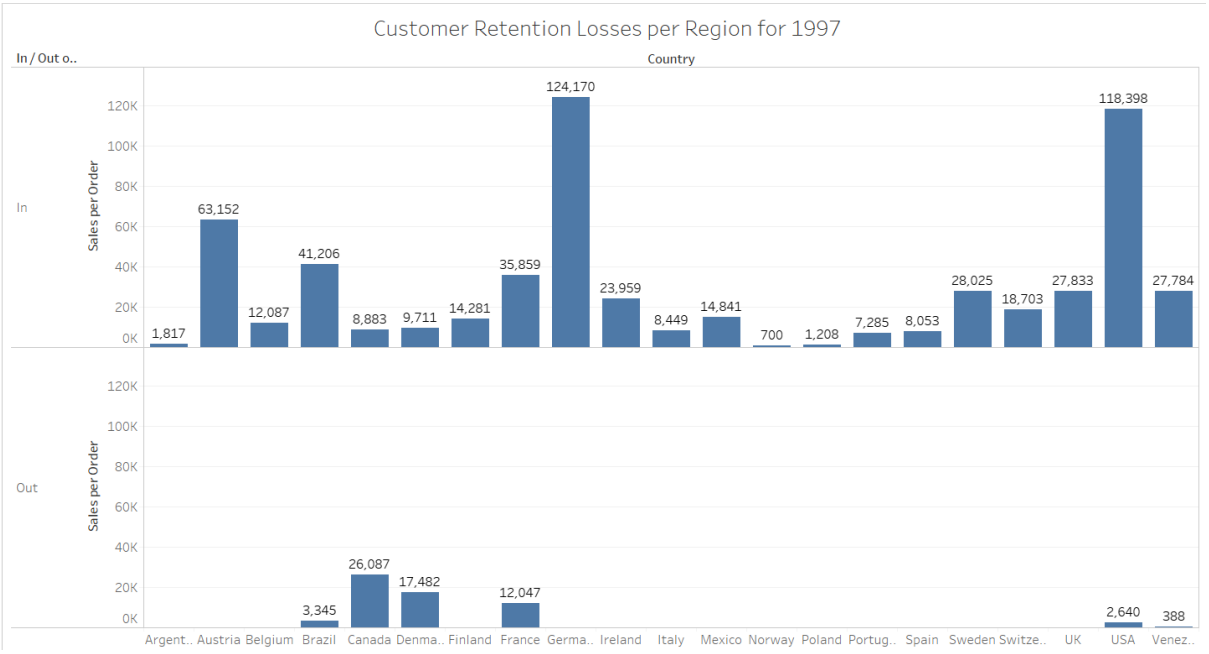


Expanding on this we can also visualize what the overall revenue impact is on the business from this failure to retain the customers shown above. The graph below highlights this by showing the overall sales per region of customers of those we have retained and those we failed to retain.



In total sales as per the table shown the customers we failed to retain represent 0.37% of total sales for 1996.

	In / Out of Customers..	
Country	In	Out
Austria	29,352	
Belgium	6,439	
Brazil	23,849	
Canada	7,950	
Denmark	3,012	
Finland	3,211	
France	17,630	
Germany	37,805	
Ireland	10,562	
Italy	1,004	
Mexico	4,587	101
Norway	1,058	
Poland	459	
Portugal	2,482	
Spain	2,360	740
Sweden	7,415	
Switzerland	4,290	
UK	9,654	
USA	41,908	
Venezuela	10,432	
Grand Total	225,457	841



Using the same method for 1997 our failure to retain customers for that year represented 9.4% of total sales for that year.

	In / Out of Customers..	
Country	In	Out
Argentina	1,817	
Austria	63,152	
Belgium	12,087	
Brazil	41,206	3,345
Canada	8,883	26,087
Denmark	9,711	17,482
Finland	14,281	
France	35,859	12,047
Germany	124,170	
Ireland	23,959	
Italy	8,449	
Mexico	14,841	
Norway	700	
Poland	1,208	
Portugal	7,285	
Spain	8,053	
Sweden	28,025	
Switzerland	18,703	
UK	27,833	
USA	118,398	2,640
Venezuela	27,784	388
Grand Total	596,400	61,989

Further measures that the business might decide to take in light of losing these customers might be to look at other KPIs from those regions to see if there are any obvious indications as to why these

customers were lost. For example, what is the Order Cycle fulfillment time in these regions? How is the employee responsible for this region performing? Can we offer these customers a further discount in order to win their business back?

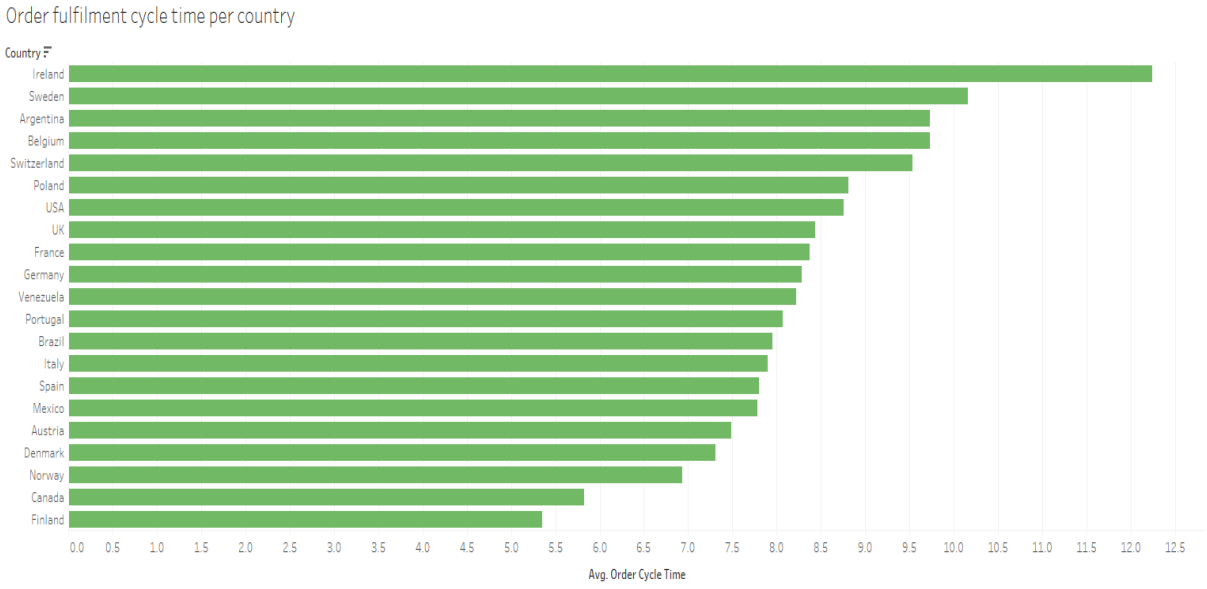
KPI Order fulfillment cycle time

Order Fulfillment Cycle Time measures the time it takes from customer order to the receipt of the product or service by the customers. It therefore provides an insight into the internal efficiency and supply chain effectiveness

The calculation formula for this indicator is **: Order Fulfillment Cycle Time = Source Cycle Time + Make Cycle Time + Delivery Cycle Time**

In NorthWind database example we don`t have a Source or a Make Cycle time so what we have in this case in the Delivery Cycle Time that determines the value of this KPI

When it comes to product categories, we find that there is no significant difference between the order fulfilment cycle time on the different categories.

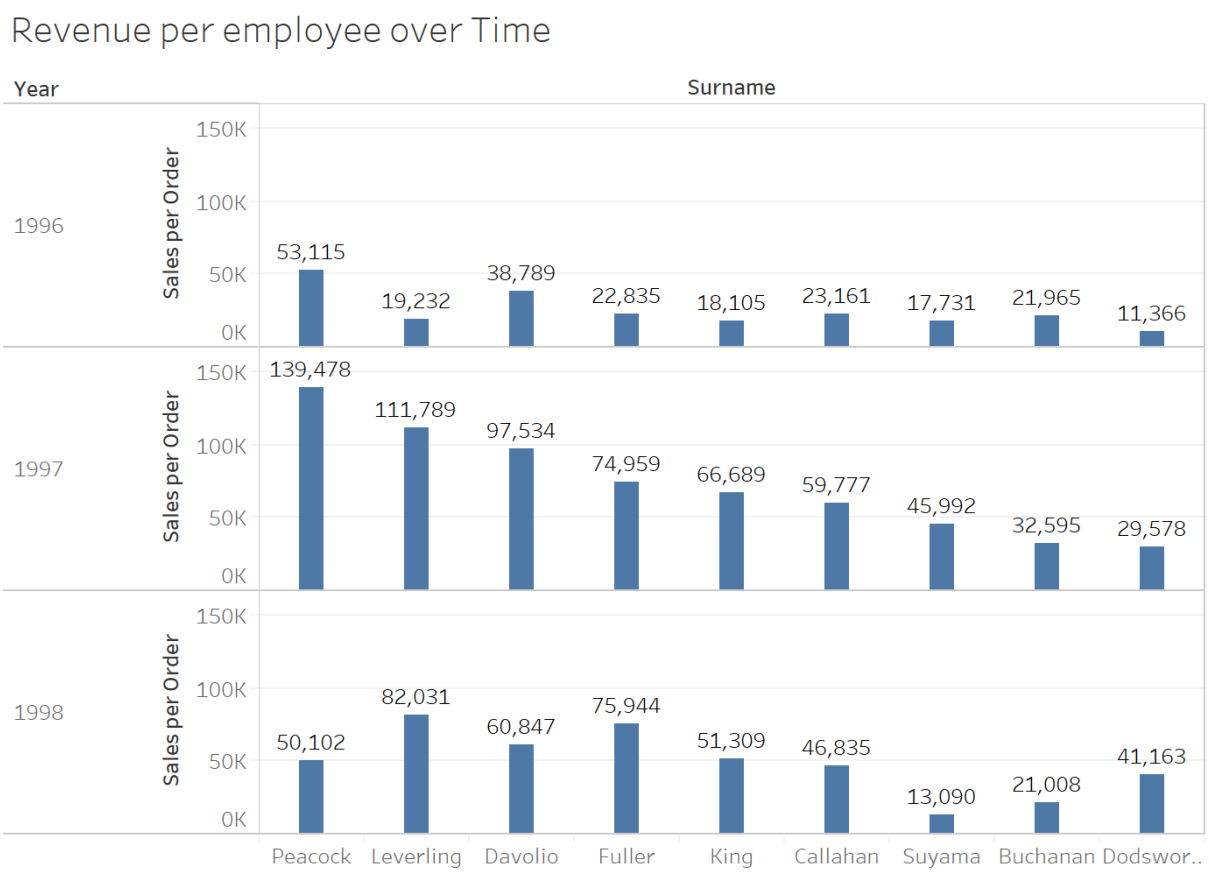


However, when it comes to analyzing this KPI for countries we found that there is a significant difference. The best (shortest) order fulfilment cycle time is in Finland with an average of just under 6.5. At the other extreme is Ireland with the slowest indicator of nearly 12.5

KPI Revenue per Employee:

In the following graph we can see the revenue generated by each of the employees over time. As mentioned above this graph can be useful for the company in gaining a clear picture about the workforce. Clearly here the employee whose surname is Peacock is performing better than the others. Once realizing this, the management might decide to look into his way of work and see if there is any strategy he uses that can be adopted by other employees.

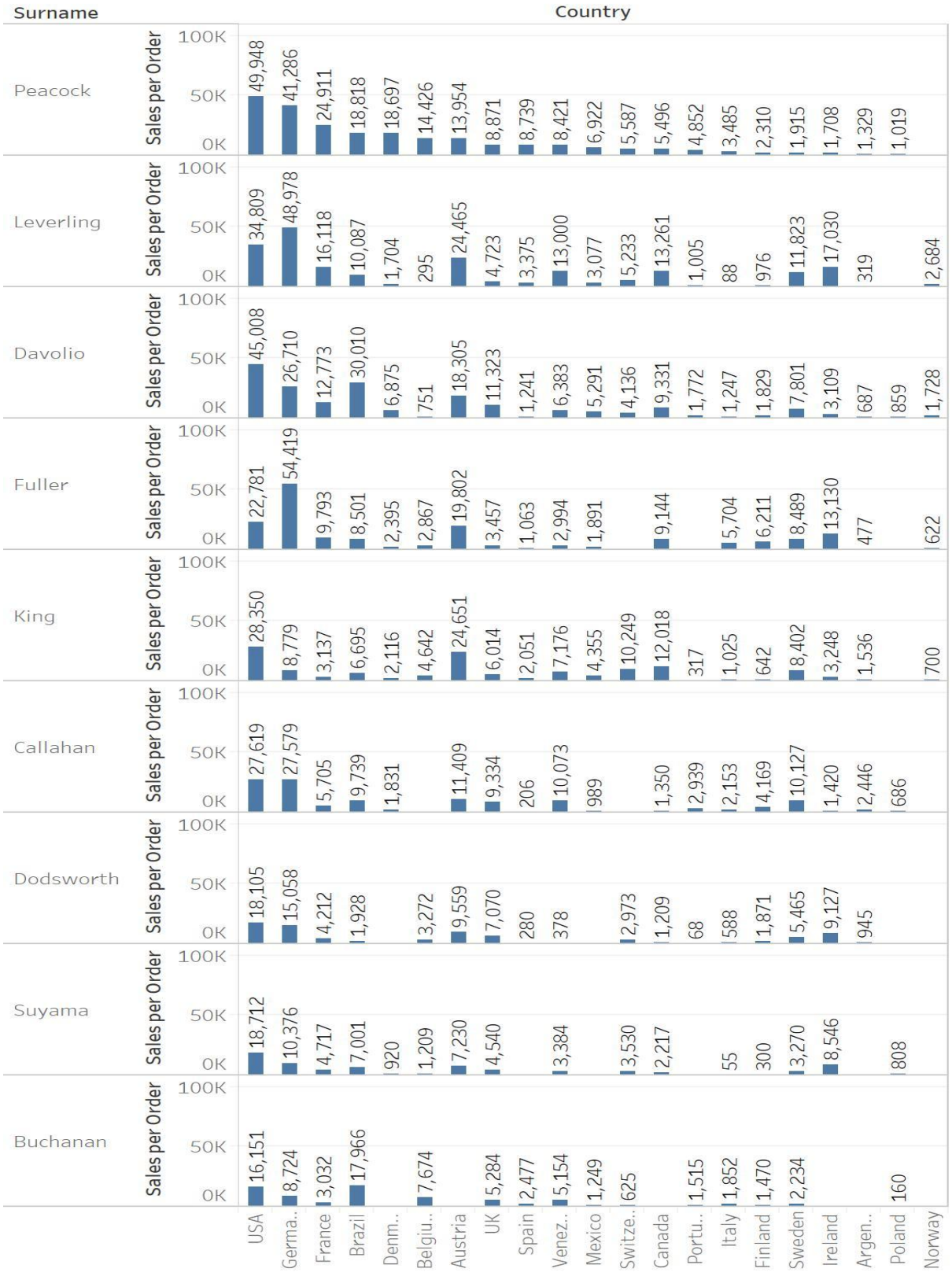
Moreover, this information is useful for the management when it comes to implement of employee bonuses, for example rewarding employees with commission on revenue for the year once certain targets are met, thus motivating the employees who exceed their targets and setting up providing additional training for those whose performances are below par.



Sum of Sales per Order for each Surname broken down by Year. The marks are labeled by sum of Sales per Order.

The following graph identifies the revenue per employee per region. As mentioned in the graph “revenue per country”, the countries who provide the highest revenue are USA, Germany, Austria, Brazil and France. A quick look at the graph shows that the highest revenue per country is generated by Fuller in Germany, the second is Peacock, with nearly 50K revenue generated in the USA followed by Leverling with around 49K also generated in Germany.

Revenue per Employee per Region



Sum of Sales per Order for each Country broken down by Surname. The marks are labeled by sum of Sales per Order.

Due to the large amount of data contained in the previews chart, the visualization is quite difficult and can only be used for the main indicators. For more detail we added one more table that only contains the numbers.

Revenue per Employee per Region

					Surname				
Country	Peacock	Davolio	Leverling	King	Callahan	Fuller	Suyama	Dodswo..	Buchan..
USA	49,948	45,008	34,809	28,350	27,619	22,781	18,712	18,105	16,151
Germany	41,286	26,710	48,978	8,779	27,579	54,419	10,376	15,058	8,724
France	24,911	12,773	16,118	3,137	5,705	9,793	4,717	4,212	3,032
Brazil	18,818	30,010	10,087	6,695	9,739	8,501	7,001	1,928	17,966
Denmark	18,697	6,875	1,704	2,116	1,831	2,395	920		
Belgium	14,426	751	295	4,642		2,867	1,209	3,272	7,674
Austria	13,954	18,305	24,465	24,651	11,409	19,802	7,230	9,559	
UK	8,871	11,323	4,723	6,014	9,334	3,457	4,540	7,070	5,284
Spain	8,739	1,241	3,375	2,051	206	1,063		280	2,477
Venezuela	8,421	6,383	13,000	7,176	10,073	2,994	3,384	378	5,154
Mexico	6,922	5,291	3,077	4,355	989	1,891			1,249
Switzerland	5,587	4,136	5,233	10,249			3,530	2,973	625
Canada	5,496	9,331	13,261	12,018	1,350	9,144	2,217	1,209	
Portugal	4,852	1,772	1,005	317	2,939			68	1,515
Italy	3,485	1,247	88	1,025	2,153	5,704	55	588	1,852
Finland	2,310	1,829	976	642	4,169	6,211	300	1,871	1,470
Sweden	1,915	7,801	11,823	8,402	10,127	8,489	3,270	5,465	2,234
Ireland	1,708	3,109	17,030	3,248	1,420	13,130	8,546	9,127	
Argentina	1,329	687	319	1,536	2,446	477		945	
Poland	1,019	859			686		808		160
Norway		1,728	2,684	700		622			

Sum of Sales per Order broken down by Surname vs. Country.

Data Modelling

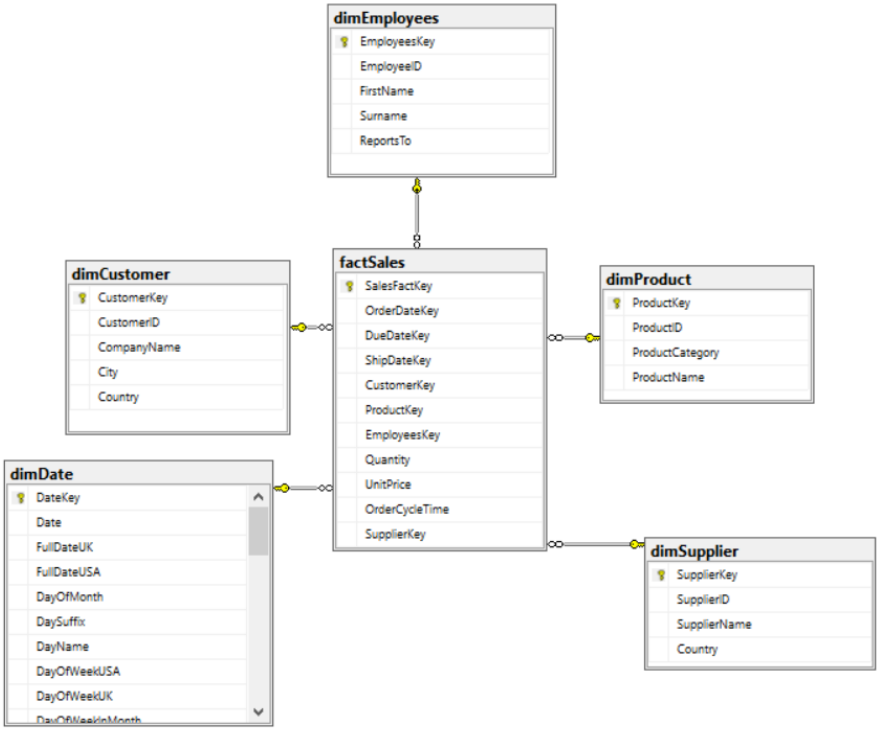
The first step in the modelling of our data was to create a logical data map much like how it would be done in a real world scenario. This was done as per the guidelines given by Kimball and Caserta with the overall aim of providing the functional specifications to build the physical ETL process and to document the relationship between original source fields and final destination fields in our data warehouse. With this in mind I applied the below guidelines creating a logical data map for this project. While other steps given for this process are given by Kimball and Caserta I feel the below are most relevant to our specific requirements.

1. Have a plan - This step involves figuring out the ETL process logically and documenting it. This step was done during classroom time and also by implementing the resources given to us on Moodle by our lecturer. Firstly our star-schema of our data warehouse was designed followed by a logical data map shown below highlighting the relationship between our source data and destination. The logical data map then served as our blueprint to start our ETL process in SSIS and provided us with the commands necessary for staging the data.

Target					Source				Transformation
Table Name	Column Name	Data Type	Table Type	SCD Type	Database Name	Table Name	Column Name	Data Type	
dimEmployee	EmployeesKey	int	Dimension	1				int	Surrogate Key
dimEmployee	EmployeesID	int	Dimension	1	Northwind	Employees	EmployeeID	int	SQL Command 1
dimEmployee	FirstName	varchar(50)	Dimension	1	Northwind	Employees	FirstName	varchar(50)	SQL Command 1
dimEmployee	Surname	varchar(50)	Dimension	1	Northwind	Employees	Surname	varchar(50)	SQL Command 1
dimEmployee	ReportsTo	varchar(50)	Dimension	1	Northwind	Employees	ReportsTo	varchar(50)	SQL Command 1
dimProduct	ProductKey	int	Dimension	1				int	Surrogate Key
dimProduct	ProductID	int	Dimension	1	Northwind	Products	ProductID	int	SQL Command 2
dimProduct	ProductCategory	varchar(50)	Dimension	1	Northwind	Categories	ProductCategory	varchar(50)	SQL Command 2
dimProduct	ProductName	varchar(50)	Dimension	1	Northwind	Products	ProductName	varchar(50)	SQL Command 2
dimSupplier	SupplierKey	int	Dimension	1				int	SQL Command 1
dimSupplier	SupplierID	int	Dimension	1	Northwind	Suppliers	SupplierID	int	SSIS Command 1
dimSupplier	SupplierName	varchar(50)	Dimension	1	Northwind	Suppliers	CompanyName	varchar(50)	SSIS Command 1
dimCustomer	CustomerKey	int	Dimension	1					Surrogate Key
dimCustomer	CustomerID	int	Dimension	1	Northwind	Customers	CustomerID	int	SSIS Command 2
dimCustomer	CompanyName	varchar(50)	Dimension	1	Northwind	Customers	CompanyName	varchar(50)	SSIS Command 2
dimCustomer	City	varchar(50)	Dimension	1	Northwind	Customers	City	varchar(50)	SSIS Command 2
dimCustomer	Country	varchar(50)	Dimension	1	Northwind	Customers	Country	varchar(50)	SSIS Command 2
dimDate	DateKey	varchar(50)	Dimension	1					Surrogate Key
factSales	SalesFactKey	int	Fact	1					Primary Key
factSales	OrderDateKey	int	Fact	1	NorthwindDW	dimDate	DateKey	int	Foreign Key
factSales	DueDateKey	int	Fact	1	NorthwindDW	dimDate	DateKey		Foreign Key
factSales	ShipDateKey	int	Fact	1	NorthwindDW	dimDate	DateKey		Foreign Key
factSales	CustomerKey	int	Fact	1	NorthwindDW	dimCustomer	CustomerKey		Foreign Key
factSales	ProductKey	int	Fact	1	NorthwindDW	dimProduct	ProductKey		Foreign Key
factSales	SupplierKey	int	Fact	1	NorthwindDW	dimSupplier	SupplierKey		Foreign Key
factSales	Quantity	float	Fact	1	Northwind	OrderDetails	Quantity		Measure 1
factSales	Unit Price	float	Fact	1	Northwind	OrderDetails	Unit Price		Measure 2
factSales	OrderCycleTime	int	Fact	1	Northwind	Orders	RequiredDate,ShippedDat	Date	Measure 3

2. Validate calculations and formulas - Measures implemented in our fact table were tested first to ensure they were exactly what was required by the end user. In our cases these were quantity, unit price and order cycle price. Some sample calculations combining these measures were also done, such as Order x Unit Price in MS-SQL so again make sure the output was what was expected.

Schema of Data-Mart

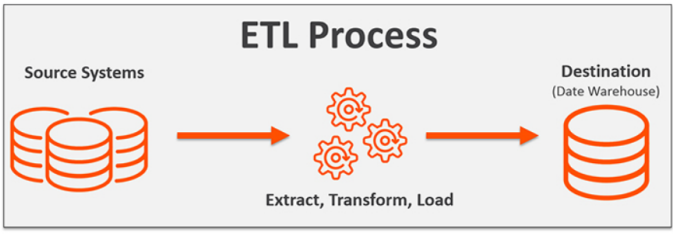


We have chosen a star-schema as its simple structure is the most effective way of organizing the data required for our data mart. Having looked into both the star-schema and the snowflake structures we ultimately chose the star-schema for a couple of reasons. The database design is a lot less complex and only one join is required to create the relationship between our fact and dimension tables.

As with any star-schema we have one dimension table per dimension.Furthermore our dimension tables contain all our attributes while our fact table contains our foreign keys and measures. Referential integrity is also maintained in this schema as every foreign key in our Fact Table is linked to a primary key in one of the dimension tables. As you can see this table is non-normalised and this schema is supported by a wide-range of BI tools. The star schema allows us to simplify analysis and its non-normalisation will also save us time when creating our SQL queries due to the reduced amount of joins as opposed to that of a snowflake schema. Query performance will also be improved due to the lesser number of tables but at the expense of data redundancy due to its non-normalisation.

ETL

ETL is the process of extracting data from our source database and transforming and loading it into our data warehouse. This process was undertaken using Microsoft SQL Server Integrated Services (SSIS) which is an ETL tool. A very general schema for this process is shown below.



One of the major advantages to using a program such as SSIS to perform the ETL process is that it is not heavily dependent on coding and it thus more user-friendly than having to hand-code a process. To quote KImball "The goal of a valuable tool is not to make trivial problems mundane but to make impossible problems possible". ETL also provide other advantages (as listed by Kimball) such as the ability to handle metadata at every step of the process as well as the ability to deliver very good performance even for very large data sets. Of course, if your business has its own IT department with available resources it might decide on using a hand-coded approach over an ETL tool, especially if you are looking for more flexibility in your ETL process and don't wish to be confined by the boundaries of a third party ETL tool as well as their cost.

Microsoft SSIS v other ETL tools

Microsoft SSIS is just one of several ETL tools on the market. Many other large software companies such as IBM and Oracles also sell ETL tools. One of the major advantages of SSIS is that it is not as expensive as some of the other tools on the market such as Informatica Power Centre. As well as that it is tightly integrated with Microsoft Visual Studio and SQL server and it thus very easy to use for anybody familiar with these programmes allowing data transformation from SQL server instances as well as text files . On the other hand, disadvantages of SSIS are that it is only compatible with Windows and while it is cheaper than other ETL software on the market it is arguably not as mature as other packages such as Informatica PowerCentre or IBM Infosphere Datastage.

ETL Staging

As highlighted by Kimball and Caserta the decision to stage your data will depend on your environmental and business requirements but will ultimately be carried out in one form or another. Before loading our data into our data mart staging it offers us a recovery point which means in the case of a transformation failure we won't have to intrude on the source system again. Furthermore in a real world scenario where massive volume of data prevents us from backing up at the database level, staging our data serves as a backup that can be saved, compressed and archived in the event of our data ever being lost.

The ETL staging process for our Northwind Data Mart was carried out in Visual Studio 2017 using SQL Server Integration Systems as a part of SQL Server Data tools. This allowed us store our staging data in simple text files. As pointed out by Kimball advantages to this are that it is faster to write to a flat file as long as one is truncating or inserting and in the case where an ETL process has failed at any point the data has been placed in a flat file the process can always restart by picking up where it left off from the already extracted data.

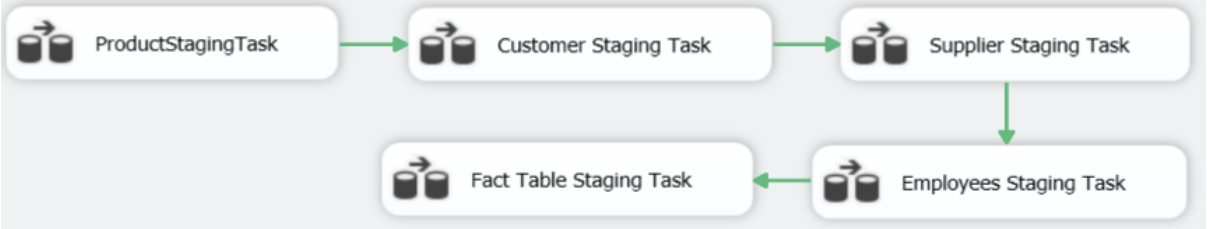
After our Data mart star-schema was created on the basis of our business drivers and after our empty dimensional tables and fact table created we began the staging process. As per the diagram below a control flow and staging task was created in Visual Studio 2017 for each dimension whose data we would stage. Data was either staged by either taking it directly from a table in the Northwind

Source data or by joining it with one or more tables in the source data using an SQL command . Our data dimension table wasn't staged as it was generated using a SQL script and populated with dates.

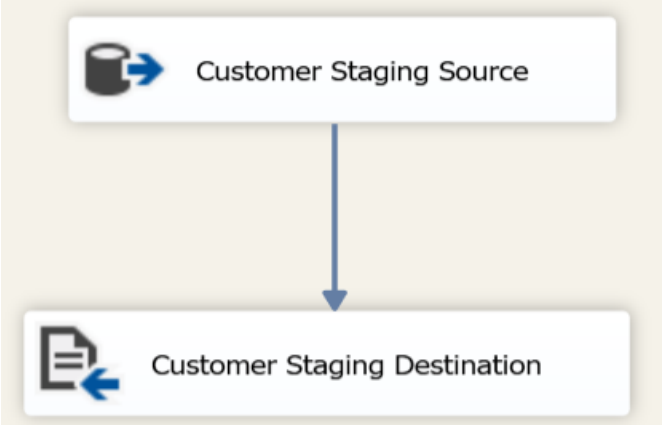
Data Mart Tables	Creation Process
ProductDim Table	SQL command
CustomerDim Table	From NW customers table
SupplierDim Table	From NW Suppliers table
EmployeesDim Table	SQL command
Fact Table	SQL command

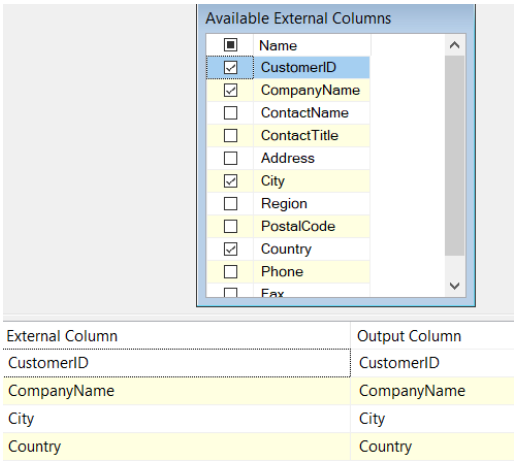
Staging Process Control Flow

The below diagrams shows the control flow used in the staging process.



Within each data flow task an ODBC source was established with our Northwind DB with the staging destination being a flat file. Once our source data was established our desired output columns were selected and mapped in the destination. This was carried out without any major problems for the staging which required SQL commands, I had previously ran them in MSSQL to make sure I achieved the intended output. I did make one small error with the SQL command for the employees dimension table which only came to light during the loading process and which I will discuss below.





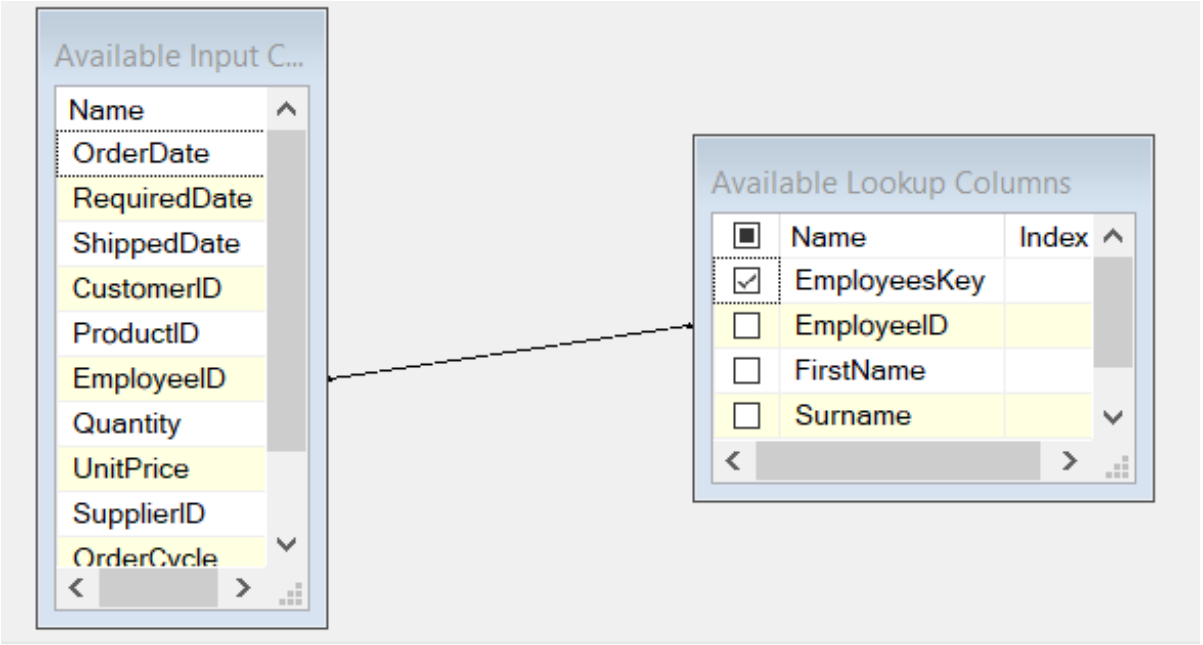
External Column	Output Column
CustomerID	CustomerID
CompanyName	CompanyName
City	City
Country	Country

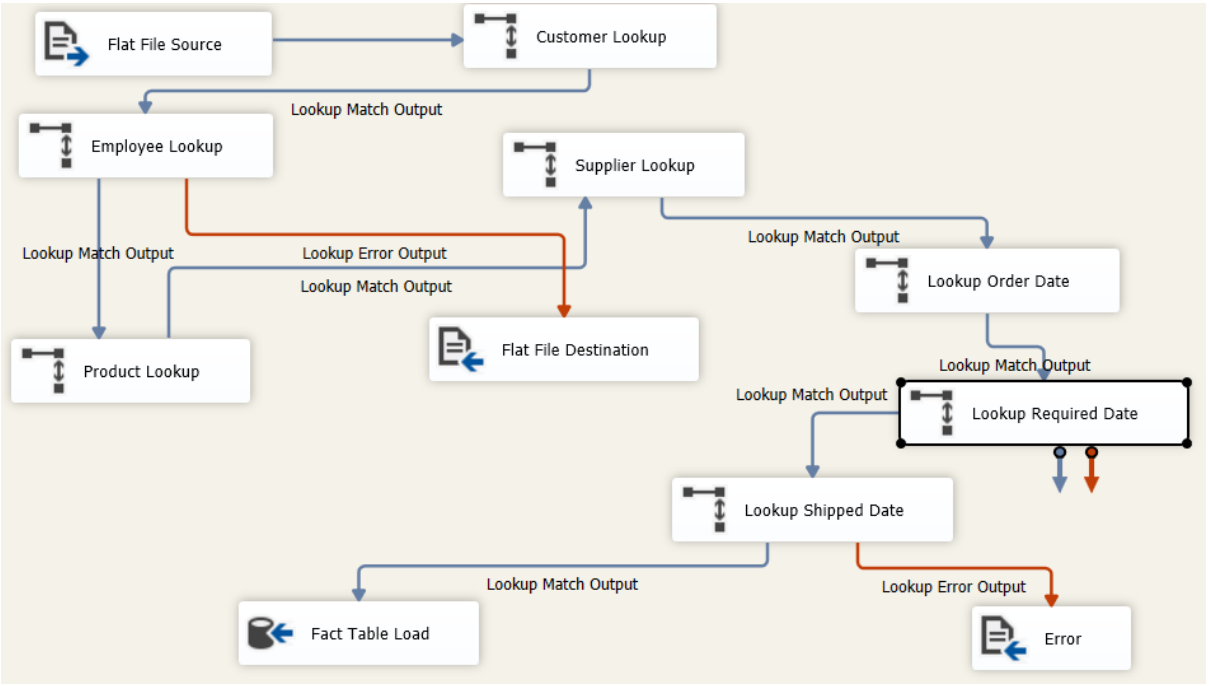
Loading

Once the staging was completed and our flat files were populated we began the staging process. Using these flat files as our source they were loaded into the relevant dimension tables followed finally by the loading of the fact table.

The control flow which can be seen below shows the loading sequence. This process starts with an SQL command which drops all foreign keys in our fact table and then truncates all tables. This was done as to not duplicate data in our data mart which would be caused by loading the data more than once without a process to truncate in place. The reason we might load data more than once is that if an error occurs initially and the process has to be repeated. In our case this is exactly what happened with our EmployeeDim table because our SQL command used to stage the data didn't have the desired output. This was noticed by creating a text file of the error output.

Similarly to the staging process we started by connecting our source (staging flat files) to our destination (dimension and fact tables) in our data mart. We then proceeded to select the columns from the source that we wanted to populate in our dimension table with and then mapped them to the new DB destination. This was done for the product, customer, supplier and employee loading. The sales fact tasks required an extra step that required the use of lookups. These lookups worked as a sort of a join that allowed us to link our foreign keys of our fact table to their primary key in the dimension table. After running the loading process our Dimension and Fact tables became populated with the desired records.





Conclusion

In this assignment we successfully took the source data from the Northwind database, used it to create a data warehouse through use of an ETL tool and analyze a number of KPIs through the use of a BI tool. Furthermore, we got an insight into how to deal with a very realistic problem in a group

setting while under a time-constraint. Since we have mainly only had modules that dealt with database design and programming it was interesting to see how these skills can be combined with BI tools to create an end product that is useful for the measurement of business performances.

Looking back in greater detail we would all agree that getting started on the ETL process was the most challenging part of the assignment. Once our data warehouse was populate the ease-of-use of Tableau and its somewhat likeness to Excel and other visualisation programs as well as being able to visualize an end project made this part the most liked. Another part of the assignment which we found challenging was trying to fit our KPIs with the limited data available to us from Northwind. There were many more KPIs that we would have liked to measure but since they involved measures that were not available to use we were somewhat restricted on what we ended up reporting on.

If we were to do this assignment again I think the one thing we would change would be our management of time. We didn't expect to to spend so much time on the actual ETL process. Since this process had to be done before we could start our reporting we ended up restricting ourselves in the time we had left for this part of the assignment.

Regarding the reporting on the KPQs and KPIs we feel we did a good job in achieving a balanced scorecard as we addressed the four keys areas relating to revenue, customers, employees and products, such as the revenue generated during the various quarters and in the different countries, the revenue generated by each employee divided again in countries and time or also the revenue grow rate and the customer retention.

Learning new tools which are utilised in the real World and the theory behind them is always beneficial. Obviously in the real World we would be dealing with a data warehouse and not a data mart. Naturally we would incur other problems which occur in the real World such as data reliability, data cleaning as well as stakeholder management. Thus, while we realize that this assignment only gives us a hint of what happens in the real world and, after this module, we can all agree that we now have a clear understanding of the importance of a data warehouse in a business and the important process which contributes to it.

Appendix A - ETL Server Commands

SQL command 1 - Used for staging of dimEmployees table.

```
SELECT
Employees.EmployeeID,
Employees.FirstName,
Employees.LastName,
COALESCE(Boss.LastName,'Fuller') as ReportsTo
```

FROM Employees

LEFT JOIN Employees Boss ON Employees.EmployeeID = Boss.EmployeeID

ORDER BY Employees.EmployeeID

SQL command 2 - Used for staging of dimProducts table.

SELECT P.ProductID, P.ProductName, C.CategoryName

FROM products P

INNER JOIN Categories C

ON P.categoryID = C.categoryID

SQL command 3 - Used for staging of dimProducts table.

SELECT O.OrderDate, O.RequiredDate, O.ShippedDate, Cu.CustomerID, P.ProductID, E.EmployeeID, OD.Quantity, OD.UnitPrice,S.SupplierID

FROM Northwind.dbo.Products P

inner join Northwind.dbo.Categories C on P.CategoryID = C.CategoryID

inner join Northwind.dbo.Suppliers S on S.SupplierID = P.SupplierID

inner join Northwind.dbo.[Order Details] OD on P.ProductID = OD.ProductID

inner join Northwind.dbo.Orders O on O.OrderID = OD.OrderID

inner join Northwind.dbo.Customers Cu on Cu.CustomerID = O.CustomerID

inner join Northwind.dbo.Employees E on O.EmployeeID = E.EmployeeID

SQL Command 4 - Used for truncating and dropping of foreign keys before the loading stage ALTER TABLE factSales DROP constraint FK_OrderDate;

ALTER TABLE factSales DROP constraint FK_DueDate;

ALTER TABLE factSales DROP constraint FK_ShipDate;

ALTER TABLE factSales DROP constraint FK_Customer;

ALTER TABLE factSales DROP constraint FK_Product;

ALTER TABLE factSales DROP constraint FK_Employees;

ALTER TABLE factSales DROP constraint FK_Suppliers;

Truncate table factSales;

Truncate table dimCustomer;

Truncate table dimProduct;

Truncate table dimSupplier;

Truncate table dimEmployees;

ALTER TABLE factSales ADD Constraint FK_OrderDate Foreign Key (OrderDateKey) REFERENCES dimDate(DateKey);

ALTER TABLE factSales ADD Constraint FK_DueDate Foreign Key (DueDateKey) REFERENCES dimDate(DateKey);

ALTER TABLE factSales ADD Constraint FK_ShipDate Foreign Key (ShipDateKey) REFERENCES dimDate(DateKey);

ALTER TABLE factSales ADD Constraint FK_Customer Foreign Key (CustomerKey) REFERENCES dimCustomer(CustomerKey);

ALTER TABLE factSales ADD Constraint FK_Product Foreign Key (ProductKey) REFERENCES dimProduct(ProductKey);

ALTER TABLE factSales ADD Constraint FK_Employees Foreign Key (EmployeesKey) REFERENCES dimEmployees(EmployeesKey);

ALTER TABLE factSales ADD Constraint FK_Suppliers Foreign Key (SupplierKey) REFERENCES dimSupplier(SupplierKey)

Sources

1. The Data Warehouse ETL Toolkit Chapters 2 and 3
2. https://elearning.dbs.ie/course/view.php?id=9862 - Moodle Notes provided by lecturer Noel Cosgrave.
3. https://www.youtube.com/watch?v=VYAJf7BgTPc&feature=youtu.be - ETL Staging (AdventureWorks mini mart example)
4. https://www.youtube.com/watch?v=vLLfAv2Xq3U&feature=youtu.be - Mini Data Mart Example (Loading into Dimensions and Facts)
5. https://www.wisdomjobs.com/e-university/data-warehouse-etl-toolkit-tutorial-201/building-the-logical-data-map-8160.html - Building the Logical Data Map
6. https://onlinehelp.tableau.com/current/pro/desktop/en-us/design_and_analyze.htm - Tableau: Build charts and analyze data
7. https://www.educba.com/oracle-vs-mssql/ - Oracle vs MSSQL

8. https://www.trustradius.com/compare-products/oracle-database-vs-sql-server - Oracle Database vs Microsoft SQL Server
9. https://www.bernardmarr.com/default.asp?contentID=773 KPI Library - Bernard Marr