DALHOUSIE UNIVERSITY

Faculty of Computer Science

CSCI 5408
Data Management, Warehousing, and Analytics

Case study report

Building data warehouse and analytics platform

2020-03-03

Submitted by: Team-19

Sagar Moghe – B00838037
Sarabjeet Singh – B00847541
Monil Panchal – B00838558

Table of content

ABSTRACT

The following synopsis is a case study on building a data warehouse and analytics platform for sample sales data. The goal is to collect, transform, and present the data in the form of visualization to the end-user. The report aims to give a summary of the collected data; the process involved in building a data warehouse; and using a business intelligence tool to project various trends in the data, thereby enabling the user to take business-oriented decisions.

1. INTRODUCTION

1.1 PURPOSE

The primary purpose of this case study is to collect a sample sales data Kaggle [1], perform the required transformation, and load the data into IBM Cognos [2], wherein the users can infer various trends in the data and take desired business-oriented decisions.

1.2 SCOPE

The overall scope of this case study is to gather and analyze the data from the given source, perform the required operations to clean and remodel the data, follow the steps like creating schemas for building a data warehouse, and to provide a platform to the end-user for visualizing the data in the desired manner.

1.3 BACKGROUND

A data warehouse is a vast data storage in any enterprise. The data is collected from various homogenous and heterogeneous sources and transformed into the desired format for storing and retrieving. The data is then loaded typically into a target database or a system to be consumed by end-users [3]. The Extraction-Transformation-Loading (ETL) process is one of the techniques used in building the data warehouse. In modern systems, the amount of data collected is humongous, and it is usually free-flowing, structured, semi-structured, or unstructured data.

A data warehouse collects this data and stores them in a particular, enterprise-specific format. Data from this massive store is extensively used for getting critical insights for business intelligence. The on-demand data visualization has been in enormous demand from the last couple of years [4]. Many platforms or tools have emerged over a period of time to give key insights to end-users, allowing them to draw the inferences and trends without knowing the complexity of the data store. Having an efficient BI tool is crucial to discover hidden trends and patterns in these data sets. Visualization of data plays a central role in business intelligence. It may help the end-users to gain insight over the current market situation and evaluate customer's needs, thereby allowing them to increase their business.

2. DATA EXTRACTION, ANALYSIS, AND TRANSFORMATION

2.1 DATA EXTRACTION

The data used in building the data warehouse for this case study is a sample sales record collected from Kaggle [1]. The data is downloaded in CSV format and is highly structured and denormalized. It contains order information about the sales of automobiles, ships, aircrafts and locomotive to various customers span across number of countries and regions over a period of two and a half years.

Following are some statistics about the dataset:

- It contains 2823 records spanned across 25 columns.
- It contains attributes related to the orders, products, sales, locations, and customer information.

2.2 DATA ANALYSIS AND TRANSFORMATION

After carefully analysing the dataset, we can conclude that the data can be segregated in the following entities.

- Products
- Orders
- Sales
- Customers
- Time
- Region

As the given dataset is already has no missing data, there's very less cleaning required. The dataset is segregated using MS Excel into the following entities containing specific attributes.

1. **Products**
   This entity set contains the following attributes related to products from the given dataset.

Table:1- Information about Products table | Source: Author

Attribute name	Data type	Description
ProductCode	Text	Unique product number
ProductLine	Text	Name of the product line
MSRP	Number	Suggest retail price from the manufacture
ORDERDATE	Text	Order date

2. Orders

This entity set contains the following attributes related to orders from the given dataset.

Table:2- Information about Orders table | Source: Author

Attribute name	Data type	Description
OrderLine	Number	Order line number
OrderNumber	Number	Unique order number
QuantityOrdered	Number	Number of items per order
PriceEach	Number	Price per each item in the order
Sales	Number	Total price for the order
DealSize	Text	Size scale of the order

3. Customers

This entity set contains the following attributes related to customers from the given dataset.

Table:3- Information about Customers table | Source: Author

Attribute name	Data type	Description
CustomerId	Number	Unique customer number
CustomerName	Text	Name of the customer
ContactFirstName	Text	Customer's contact person's first name
ContactLastName	Text	Customer's contact person's last name
Phone	Number	Phone number of the customer
AddressLine1	Text	Address line 1 of the customer
AddressLine2	Texts	Address line 2 of the customer
City	Text	City where the customer belongs to
State	Text	State where the customer belongs to
PostalCode	Text	Postalcode
Region	Text	Region of the country

4. **Time**
   This entity set contains the following attributes related to order time from the given dataset.

Table:4- Information about Time table | Source: Author

Attribute name	Data type	Description
OrderDate	Text	Date of the order
QtrID	Number	Quarter number of the year
MonthId	Number	Month number of the year
YearId	Number	Year

5. **Region**
   This entity set contains the following attributes related to order time from the given dataset.

Table:5- Information about Region table | Source: Author

Attribute name	Data type	Description
country	Text	Country of the order
territory	Text	Territory of the order

3. BUILDING THE DATA WAREHOUSE

3.1 UNDERSTANDING THE PROBLEM

Data Warehousing is a methodology of gathering, transforming, and organizing data from multiple sources, typically related to a business domain, and is generally used to analyze the data for gaining business insights. A data warehouse is like a storage engine, which captures data multiple sources, stores them in the desired integrated format. A data warehouse, in any enterprise, is considered to be the core of business intelligence. The data warehouse build in this case study is based on ETL [3].

3.2 TYPE OF DATA

The data in any data warehouse can be typically in structured, semi-structured, or unstructured format. In this case study, the dataset used is already in structured format (CSV).

3.3 EXTRACTION

The dataset is directly available in CSV format for access from the data source [1]. Since there is only one dataset and is self-sufficient, there is no need to extract any supporting data or metadata.

3.4 TRANSFORMATION

After analysing the dataset, the following steps are performed for the transformation of data into optimized form and generate specific schema for data warehouse.

Step:1 Transformation

The data is filtered and segregated into separate entity specific files (as per the step ***2.2 Data Analysis and transformation***). Following CSV files are generated with this step.

1. Orders.csv
2. ProductCode_ProductLine.csv
3. Customers.csv
4. Time.csv
5. Country_Territory.csv

Step:2 Regrouping the data for 1-1 mapping

The Customers.csv and Orders.csv files are loaded into a Python[6] script to create 1-1 mapping of unique orders and respective customers. CustomerOrder.csv file is programmatically generated with the following structure.

ORDERNUMBER	CUSTOMERNAME	customerID
10107	LandofToysInc	1
10121	ReimsCollectables	2
10134	LyonSouveniers	3
10145	Toys4GrownUpscom	4

Image 1: CustomerOrder.csv file sample | Source: Author

```
dict = {}

file1 = open("CustomerINFO.csv", 'r')
lines = file1.read().split("\n")

for line in lines:

    fields = line.split(",")
    dict[fields[1]] = fields[0]

    # print(fields[3])

print(dict)

file2 = open("interimFileCustomer.csv", 'r')
file3 = open("interimFile3.csv", 'a')

lines = file2.read().split("\n")

for line in lines:
    fields = line.split(",")
    # print(dict[fields[1]])
    if dict.keys().__contains__(fields[1].strip()):
        print(" Entry found !! ")
        line = line + "," + dict[fields[1]] + "\n"
        file3.write(line)
    else:
        line = line + ", ********" + "\n"
        file3.write(line)
```

Image 2: Python script for customer-order | Source: Author

Similarly, orderDate and country has been mapped with unique orders to maintain the foreign key for the central table and 1-1 relationship with respect to individual orders.

Step:3 Using SQL query to calculate measures for central sales table.

The above generated CSV files are loaded in MySQL and tables are created in normalized form. Once we get the normalized table, we created a table with only OrderNumbers and Sales information.
Finally, a ***SalesFact*** table is created, containing the total number of sales for each order is created using aggregation query in MySQL[7].

Following are some statistics derived from the normalized data:

- It contains 307 unique orders among 92 customers.
- The customers are distributed over 20 countries.

```
4
5  create table salesFact (
6      ORDERNUMBER int,
7      SALES float
8  );
9
10
11 select * from salesFact;
12
13 select ORDERNUMBER, SUM(SALES) as totalSUM
14 FROM salesFact
15 GROUP BY ORDERNUMBER;
16
```

Image 3: SQL query to calculate total sales per order | Source: Author

ORDERNUMBER	totalSUM
10100	12133.25
10101	11432.33984375
10102	6864.050048828125
10103	54701.999755859375
10104	44621.96008300781
10105	58871.110107421875
10106	56181.320068359375
10107	25783.759765625
10108	55245.02014160156
10109	27398.820434570312
10110	51017.919860839844
10111	18695.579833984375
10112	9748.999755859375
10113	12398.56005859375
10114	38217.41046142578
10115	24777.409912109375

Image 4: SQL query output | Source: Author

3.5 FACT AND DIMENSION TABLES

After analyzing the data and tables created in the above step, following tables are created.

Dimension tables

The segregated .csv files generated as a result of step ***3.4 Transformation*** are taken as dimension tables.

Table:6- Information about all Dimension tables | Source: Author

Dimension table	Description
Product_dimension	Table containing dimensions related to product attributes
Order_dimension	Table containing dimensions related to order attributes
Customer_dimension	Table containing dimensions related to customer attributes
Region_dimension	Table containing dimensions related to country and territory attributes
Time_dimension	Table containing dimensions related quarter, month and year attributes

Fact table

As the dataset is about sales of products to customers located over various regions; attributes like orderNumber, customerId, country, orderDate and total sales for each order are considered as crucial metrics for the sales process. These attributes mapped using foreign key reference from the dimension tables.

Table:7- Information about all Fact Table | Source: Author

Fact table attribute	Type	Description
OrderNumber	PK, FK	OrderNumber referencing to the **Order_dimension table**
CustomerId	FK	CustomerId referencing to the **Customer_dimension table**
OrderDate	FK	OrderNumber referencing to the **Time_dimension table**
Country	FK	OrderNumber referencing to the **Region_dimension table**
status		Attribute showing the order status
Total sales		Derived attribute showing the total number of sales for each other

3.6 SNOWFLAKE SCHEMA FOR MODELLING

As the tables are normalized, snowflake schema, an extended form of star schema is chosen to model this data warehouse. The main benefit of using snowflake schema is its extensibility. If any related dimension table is added, this model can easily adapt the additional dimension tables. Also, since snowflake supports normalization it will help us to remove any redundancies in data and provide data integrity.

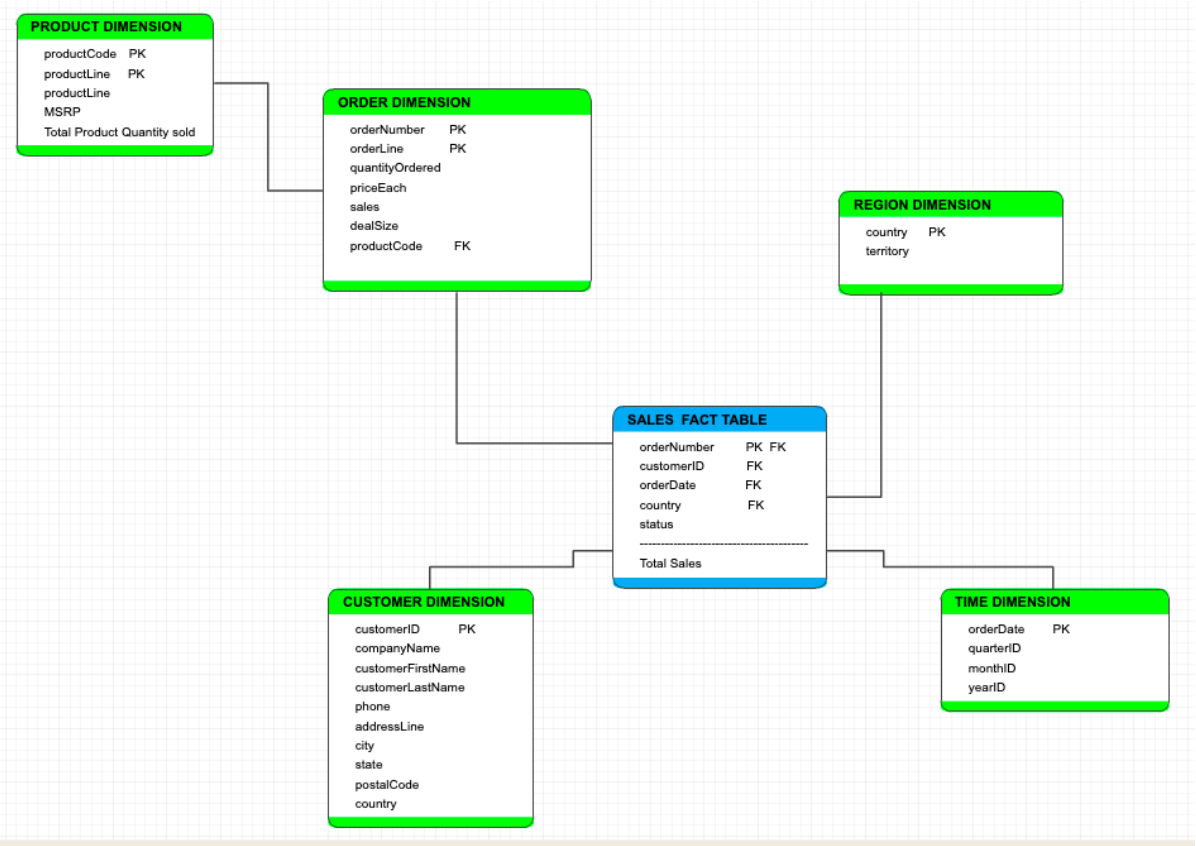

Image 5: Snowflake schema | Source: Author

4. IBM COGNOS TOOL FOR REPORTING AND ANALYTICS

IBM Cognos is a Business intelligence tool to analyze data, create or view reports to help making important business decisions [5].

4.1 COGNOS SCHEMA/DATA MODEL

The CSV files are loaded into IBM Cognos as source and following data model is created.

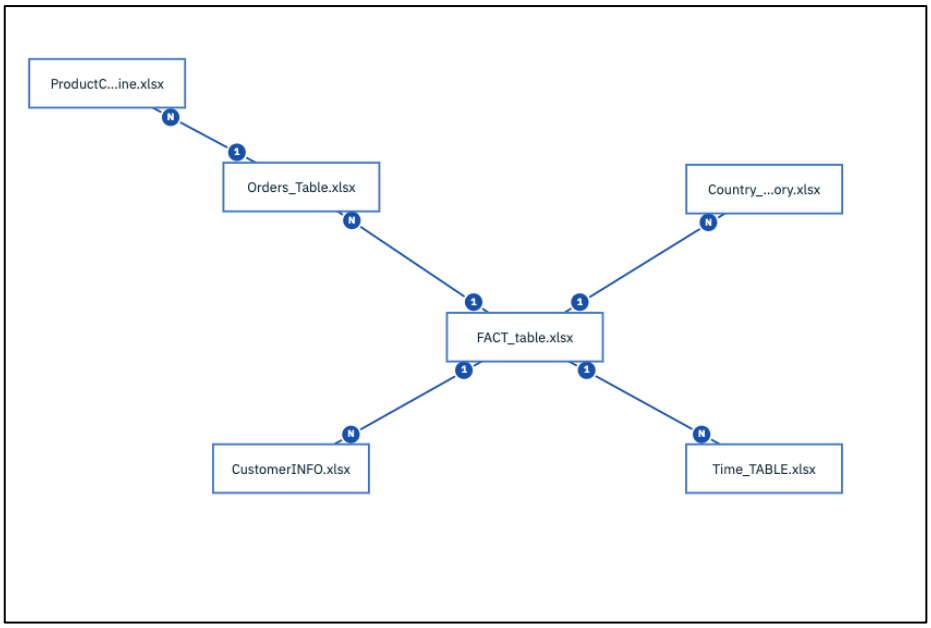

Image 6: Cognos data model | Source: Author

4.2 COGNOS REPORT AND ANALYTICS

To visualize the data trends and perform some analytics, we have created a dashboard in Cognos over the data model we have created in the last section. In the dashboard various different tabs has been made using different visualization items such as bar graph, maps, pie charts etc.
We have tried to analyze every single relationship we can think of from the given dataset and presented the same in the dashboard using multiple tabs.
Some of them are shown below and rest of the screenshots are placed in Appendix.

1. Viewing sales summary: Total sales per time dimension

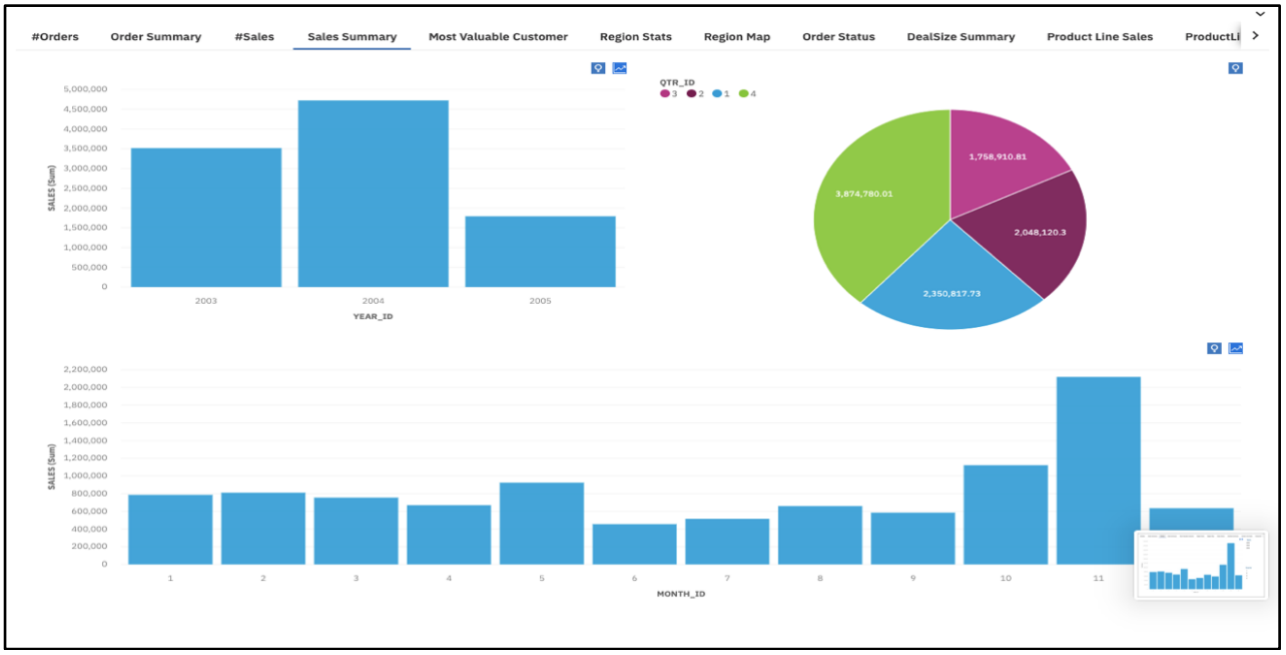

Image 7: Cognos dashboard visualisation for Sales summary | Source: Author

2. Viewing the most valuable customers based on sales for the specific year and quarter.

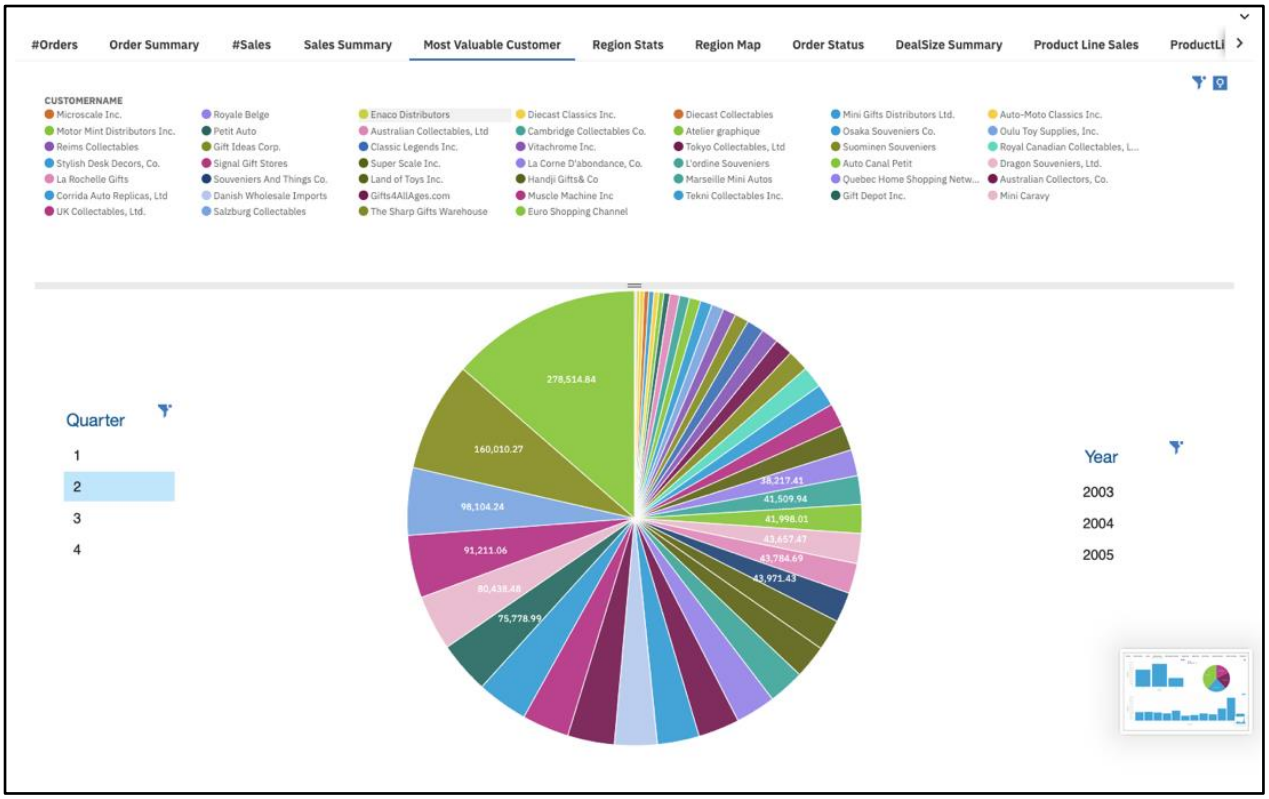

Image 8: Cognos dashboard visualisation for Valuable customers | Source: Author

REFERENCES

[1] "Sample Sales Data", *Kaggle.com*. [Online]. Available: https://www.kaggle.com/kyanyoga/sample-sales-data. [Accessed: 15- Mar- 2020]

[2] "IBM Cognos Analytics", *IBM Cognos Analytics*. [Online]. Available: https://www.ibm.com/ca-en/products/cognos-analytics. [Accessed: 22- Mar- 2020]

[3] Q. Hanlin, J. Xianzhen and Z. Xianrong, "Research on Extract, Transform and Load (ETL) in Land and Resources Star Schema Data Warehouse," *2012 Fifth International Symposium on Computational Intelligence and Design*, Hangzhou, 2012, pp. 120-123.

[4] Zheng, Jack. (2017). Data Visualization for Business Intelligence. 10.4324/9781315471136-6. [Accessed: 26- Mar- 2020]

[5] "IBM Knowledge Center", *Ibm.com*. [Online]. Available: https://www.ibm.com/support/knowledgecenter/en/SSEP7J_10.2.2/com.ibm.swg.ba.cognos.wig_cr.10.2.2.doc/c_gtstd_c8_bi.html. [Accessed: 26- Mar- 2020]

[6] "3.8.2 Documentation", Docs.python.org. [Online]. Available: https://docs.python.org/3/. [Accessed: 20- Mar- 2020].

[7] "MySQL :: MySQL Documentation", Dev.mysql.com. [Online]. Available: https://dev.mysql.com/doc/. [Accessed: 21- Mar- 2020].

APPENDIX

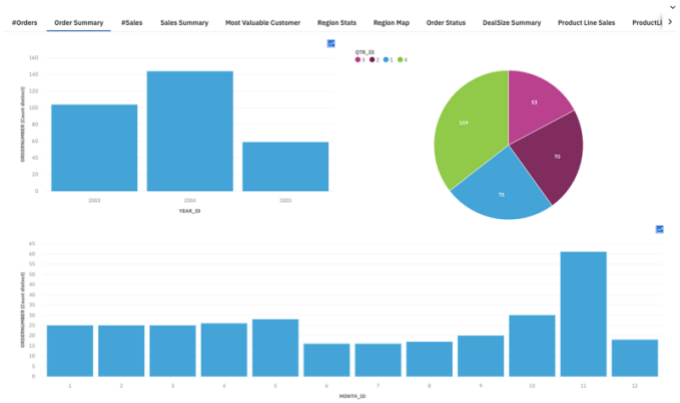

Image 9: Cognos dashboard visualisation for Order Summary | Source: Author

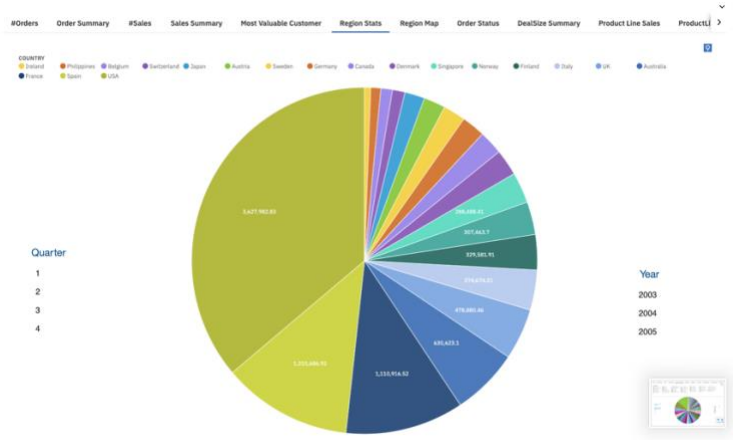

Image 10: Cognos dashboard visualisation for Region Statistics | Source: Author

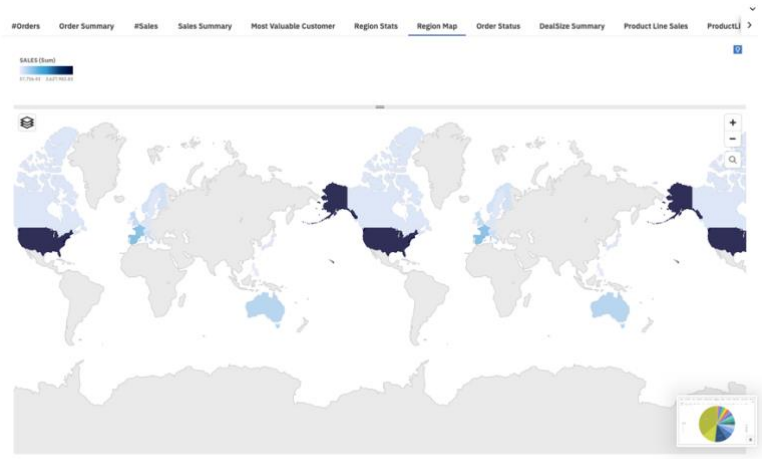

Image 11: Cognos dashboard Map visualisation for Region per Sale | Source: Author

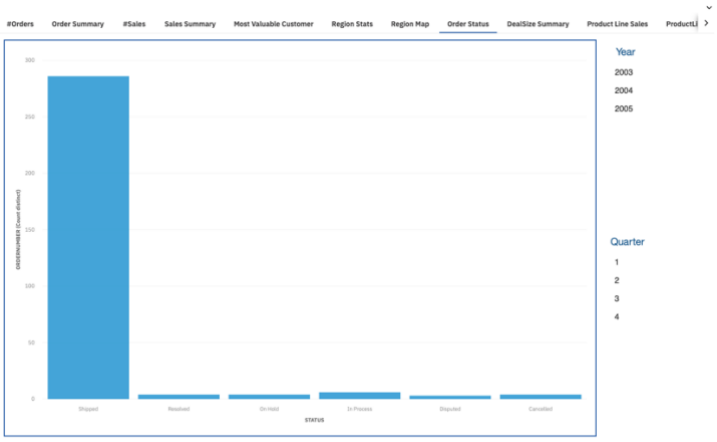

Image 12: Cognos dashboard visualisation for Order status per time dimension | Source: Author

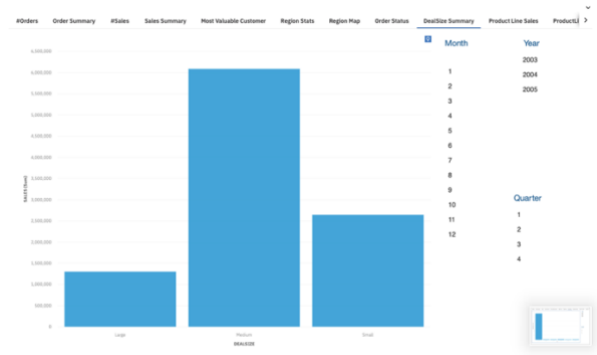

Image 13: Cognos dashboard visualisation for Deal size per visualization | Source: Author

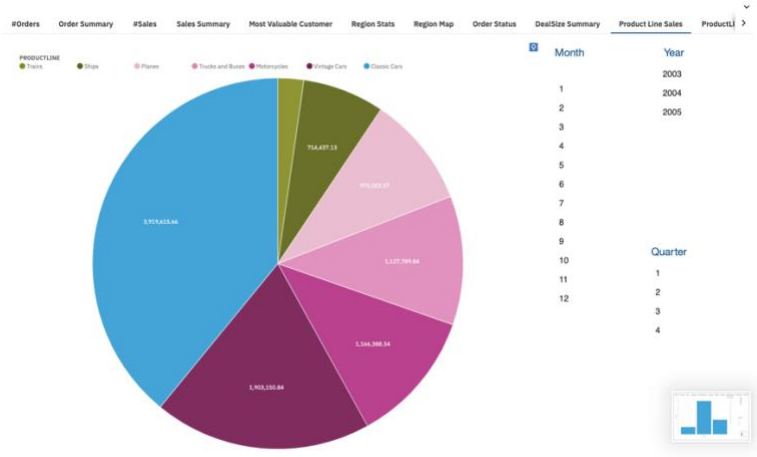

Image 14: Cognos dashboard visualisation for Product line per time dimension | Source: Author

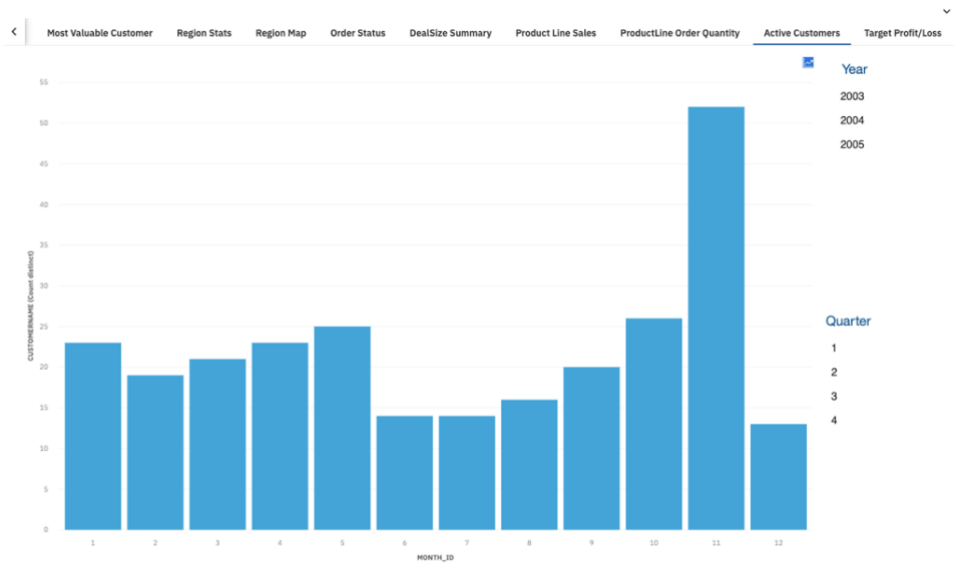

Image 15: Cognos dashboard visualisation for Active Customers| Source: Author

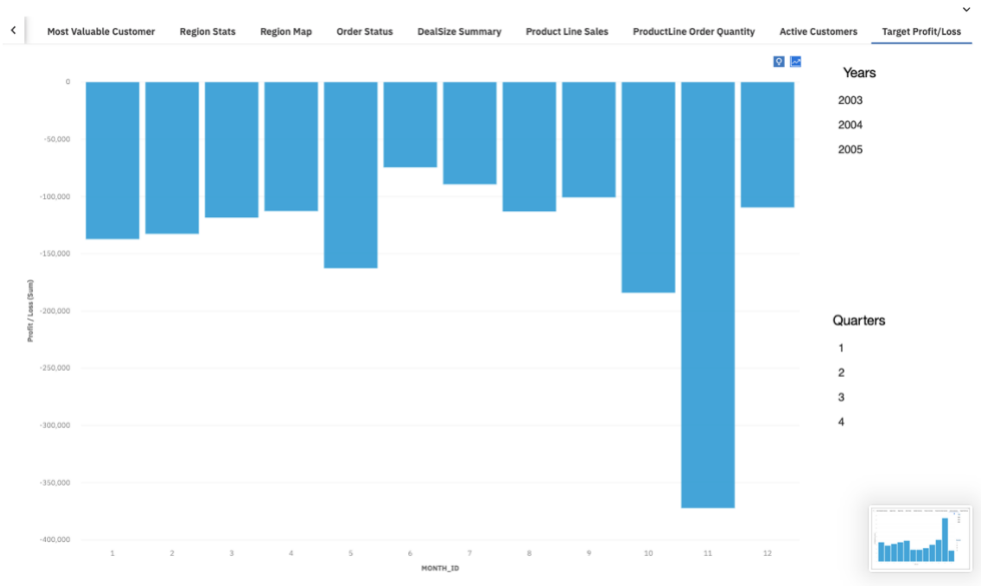

Image 16: Cognos dashboard visualisation for Target Profit/Loss | Source: Author