

**Implementing Data Warehouse On-premises**

Abiodun Bamidele Obisesan

Degree Programme in Business Information Technology

Bachelor's Thesis  
Business Information Tech  
2021



<b>Author(s)</b> Abiodun Bamidele Obisesan	
<b>Degree programme</b> Business Information Technology	
<b>Thesis title</b> Implementing Data Warehouse On-premises	<b>Number of pages</b> <b>8 + 50</b>
<p>Data whether historical or live is now more crucial to the success of a business more than ever before. Organizations are leveraging on the ability to query the stored data to make sound business decision.</p> <p>This work has examined the implementation of Datawarehouse from its historical viewpoint, importance, and actual implementation. Specific attention was paid to data warehouse evolution overtime. Associated tools like SSIS, SSAS and SSRS were thoroughly examined.</p> <p>Furthermore, this work replicated the production of data warehouse on-premises. All possible scenarios in a real project was carefully considered. For example, the issue of business rules and steps to data warehouse implementation were duly considered in chapter two and three.</p> <p>The actual implementation was carried out in Chapter four. The objectives of the work were to understand how to implement a data warehouse from an enterprise standpoint. Emphasis was placed on extract load and transform where scalability and performance were important factors. Chapter five of this work discussed the result while appropriate recommendations were made.</p>	
<b>Keywords</b> Data, Data Warehouse, Reporting, SSIS, ETL	

## **Abbreviations**

- ETL Extract, Transform and Load
- DW Data Warehouse
- MDBs Multidimensional database
- AWS Amazon Web Services
- SQL Structured Query Language
- ERP Enterprise Resource Planning
- NAV Microsoft NAV – an ERP system
- DSS Decision Support Systems
- EDW Enterprise Data Warehouse
- KPI Key Performance Indicator

## List of Figures

(i)	Figure 3.1	Conceptual Modelling	Page 18
(ii)	Figure 3.2	Logical Modelling	Page 18
(iii)	Figure 3.3	Physical Modelling	Page 19
(iv)	Figure 3.4	Data Warehouse Architecture	Page 20
(v)	Figure 3.5	SQL Server Integration Services	Page 21
(vi)	Figure 3.6	Package Definition	Page 23
(vii)	Figure 3.7	Connection Manager	Page 23
(viii)	Figure 3.8	Task Definition	Page 24
(ix)	Figure 3.9	ETL Definition	Page 24
(x)	Figure 3.10	SSAS Architecture	Page 26
(xi)	Figure 3.11	SSRS Architecture	Page 27
(xii)	Figure 4.1	Query Result	Page 31
(xiii)	Figure 4.2	Final Architecture	Page 32
(xiv)	Figure 4.3	SSIS Set-Up	Page 33
(xv)	Figure 4.4	Implementation Connection Manager	Page 34
(xvi)	Figure 4.5	Completed Packages	Page 35
(xvii)	Figure 4.6	Fact Table	Page 36
(xviii)	Figure 5.1	Execute SQL	Page 37
(xix)	Figure 5.2	Dimension Table	Page 39
(xx)	Figure 5.3	Final Outlook	Page 39
(xxi)	Figure 5.4	Update Test	Page 41

## List of Tables

- (i) Types of Reports in SSRS Page 28

## Table of contents

1 Introduction.....	8
1.1 The Research Objectives and Questions.....	9
1.2 Research Method and Data Collection .....	10
1.3 Thesis Outline .....	10
2 Data Warehouse Concept .....	11
2.1 Why Data Warehouse?.....	11
3 Data Warehouse Evolution.....	13
3.1 Business Intelligence Tools vs Data Warehouse.....	14
3.2 Application Database vs Data Warehouse.....	15
3.3 Data Warehouse Processes.....	16
3.4 SQL Server Integration Services .....	20
3.4.1 Features of SSIS.....	21
3.4.2 Advantages of SSIS.....	22
3.4.3 ETL Set-Up Within SSIS.....	22
3.4.4 Package Definition.....	23
3.4.5 Connection Manager.....	23
3.4.6 Other Components.....	23
3.5 Microsoft Analysis Services (SSAS) .....	24
3.6 SQL Server Reporting Services .....	24
4 Data Warehouse Implementation.....	29
4.1 Resources.....	29
4.2 Process of Implementation.....	30
4.3 Defining Business Process.....	30
4.4 Data Source Exploration.....	30
4.5 Data Destination and Architecture.....	33
4.6 ETL Processes.....	33
4.6.1 SSIS Set-up.....	33
4.6.1 Loading Dimension Table.....	34
4.6.1 Loading Fact Table.....	35
5 Discussion of Result.....	37
5.1 Test.....	40
5.2 Challenges.....	41
5.3 Recommendation.....	42
5.4 Conclusion.....	42
References .....	46

Appendix 1. DimCustomer Script.....	46
Appendix 2. DimPurchase Script .....	46
Appendix 3. DimItem Script.....	47
Appendix 4. DimSales Script.....	49
Appendix 5. DimDate Script.....	51
Appendix 5. DimFactTable Script.....	53
Appendix 5. DimFactTable Script.....	57

## 1 Introduction

The background to the present study is to research the way to place value on the organization data. Organizations are established to unravel customers need and successively make profits. The primary way to solve customer needs is to offer products and services that meet their needs. Traditionally, organizations attempt to reach customers through newspapers, media whether television or radio and word of mouth. Recently social media has added flavour to the means whereby organization can get across to the end-users.

However, there is another means organization could leverage to succeed in customers whether potential or current. meaning is concealed in data. Data is not just a bit of data. It is a unit of information which will provide useful details for a crucial and actionable decision-making process. Data are like nuclei that would propel a corporation to life. In recent times, data has become and is playing an important role within the lifetime of organization. In essence data provides means to know customers, their needs, and the way to present products and services being offered to them. Data also provide means by which organization can understand their own process and the way to enhance on them. Through appropriate use of data, a corporation could leverage on the data at its disposal to make new products and services that meet customers' needs. Data are a part of organization assets.

Despite the important qualities and characteristics of data mentioned above, the way data are stored and accessed in most organization does not represent the value it represents. The major problem is that each department in the organization tends to store data in different formats and access same for immediate needs. The underlying problem here is that different personnel within the same organization have different data sources. Hence, those data become problematic and does not serve a higher purpose when an organization want to have an overview.

In this context, this work seeks to research and implement the way to place value on the organization data through the acceptable implementation of a standard repository that might be beneficial to the general organization benefits.



## **1.1 The Research Objective/Questions**

This work aims at demonstrating how to implement an enterprise data warehouse. Thus, the objectives are to implement a data warehouse that is usable for data analysis and business intelligence purposes. The weight of this work will be placed on extract, transform and load (ETL) job. ETL jobs are core component of a data warehousing project. This work will attempt to answer the following questions:

- a. how ETL jobs impacts on performance during data transfer
- b. how scalability can be implemented with SSIS (Microsoft Integration Services) seamlessly
- c. how to move data into the data warehouse and synchronize them and same objective for learning purposes

In this work, the core objective is to identify how to implement a data warehouse in an optimum way. Performance and scalability are two critical issues that would be considered while designing the ETL job in the SSIS package. Subject to data availability, the emphasis would be placed on how to synchronize data from different sources (if such a need arises during implementation) together given the fact metadata could be different. This would be to find an optimum way to implement data warehouse without incurring additional costs.

In conclusion, this work seeks to implement an enterprise data warehouse that can support business decision on premise. Specifically, this implementation will be carried-out within Haaga-Helia University of Applied Sciences premises by using Nav2015:7230/BIG4TF023\_Rehn\_2022 as the data source.

The scope of this work has been limited to a business intelligence course where different business processes e.g., sales or marketing data would form the basis of primary data source. The total combination of the data would be synchronized and move into data warehouse.

## **1.2 Research Method and Data Collection**

This thesis work is hybrid. This implies that the work includes elements of research and element of creating a useable product.

The approach to data collection of this work is to source data from articles, scientific journals, and various publication to understand what has been done. In essence, the methodology used for data collection in this work is a derived data. Hence, data source came mainly from:

- a. Articles
- b. Scientific journal

The major limitation to data collection was the inability to get this work commissioned by an organization. If this thesis work had used live data coming from the organization, it would have been possible to speak with different stakeholders (business users, project sponsors, end-users, and IT professionals) and extract useful information that could have been central to the implementation of this work.

### **1.3 Thesis Outline**

This work has been arranged in such a way to make it easy and simple for the reader to read, digest and understand the work. This thesis work is hybrid, it consists of research where past works are examined. The other part of this work is to create a product based on knowledge gained from the research part.

In view of the above, the work is divided into two broad categories namely the research part and product. Chapter one of this work opens the reader of this work to its doors. Chapter one gave the background to the work and what would be achieved.

Chapters two and three were the theoretical part of this thesis. In those two-chapter, author was able to look back in history of data warehouse. These two parts form the research part of the work. Chapter four was the implementation area of the work. In this chapter how the data warehouse was implemented was adequately documented and explained.

Finally, chapter five deals with the recommendations and conclusions.

## **2 Data Warehouse Concept**

In computing, a data warehouse (DW or DWH), is additionally referred to as an enterprise data warehouse (EDW). It is a system used for reporting and data analysis. It is considered a core component of business intelligence (Canan Eren Atay & Gerogia Garani, 2019, Pg 196-206). In a more contextual form, DW may be a central repository of integrated data from one or more different sources.

Data warehousing is the process of managing a data warehouse and its components, called data marts (David Loshin, 2003). The data warehouse management processes also include ongoing support needs of the refresh data cycle, database maintenance, and continual refinements to the underlying data model. DW can be used to store current and historical data in one unified platform. Once the data warehouse is fully functional, it can be used for support in reporting analysis, business intelligence or for data scientist work (Dedić, Nedim et. al 2016). Data warehousing can also be viewed as a tool to enhance and support business intelligence processes.

A data warehouse is a subject-oriented, integrated, non-volatile, and time-variant collection of data in support of management's decisions. The DW contains granular corporate data. Data in the data warehouse can be used for many different purposes, including sitting and waiting for future requirements which are unknown today (Douglas K. Barry & David Dick. 2013).

### **2.1 Why Data Warehouse?**

The need for a data warehouse arises due to the ever-increasing number of databases within an enterprise or organization. Multiple databases arose because each department within an organization tends to keep their data for immediate needs. However, when it comes to a bigger picture of using the data for the entire organization, departmental data may not satisfy the bigger requirement. Having scattered data within departments could place an extra burden on extracting data from different databases which could be laborious and time-consuming. The implication for this could be delayed decision, wasted opportunities, inability to spot where an organization could make money or where concentrated effort could be targeted. Data warehouses provide means to solve the problem of multiple databases, data analysis and business intelligence. In this regard data utilization and consumption can come from a single repository which then places value on both the data and the storage. Through the implementation of a data warehouse there would be a coordinated effort between the existing systems and implementing an enterprise data

warehouse. The existing data system must be coordinated to reflect the current needs and how to address them. In this regard, the need to have a single location for data storage is crucial.

The types of data maintained by a DW are used in the management of reports, various business queries, decision support systems, management information systems, and data mining applications by linking the various data that are distributed throughout an organization. All these benefits could be achieved by using an enterprise data warehouse.

### 3 Data Warehouse Evolution

Inmon described the origins of data warehousing and decision support systems (DSS) processing as far back as the very early days of computers and information systems (Inmon, 2013). The pre-advent implementation of the data warehouse left many users with frustrating experiences because the IT department acted as an intermediary between the data user and the data itself (Douglas K. Barry & David Dick, 2013.).

The perspective of data warehousing projects changed in the 1990s. In the same period, there was an information explosion from the organization's point of view. In terms of business activities, the organization and business enterprise became more complex and global centric. The business competition became more intense. The major source of information that drove businesses was majorly operational data from a computer system. Operational data alone, however, could not provide adequate direction for business needs. The real data needed for the business decision were different kinds of information that could be readily leveraged on to make strategic decisions. Such kinds of data would include information from sales, marketing, or human resources (Douglas & David 2013.).

The growth of data information in the 1990s stimulated the interest to analyse those data as they grew. The ability to analyse data within organization and offer correct business decisions became a critical selling point for most organizations (Inmon et al.,2008). An increasing number of organizations are realizing that the vast amounts of collected data can and must be used to guide their business decisions (IBM, Redbooks). Typically, the management of the organization wants to answer complex business questions based on the collected data. This trajectory significantly alters the perspective on the data warehousing project and how it has evolved. For example, online applications processing was the brain-behind between technology such ATM (automated teller processing), bank teller processing, claim processing, and retail point of sale among many others (Inmon et al.,2008).

The evolution of data warehouse helped the organizations to overcome the shortcomings observed in operational data and other areas such as the bring-your device strategy. The evolution thus has caused the implementation of the data warehouse to be improved. The approach to data warehouse has been changed as well. It is possible to have a data warehouse on premises or cloud. Cloud based data warehouse platforms include Redshift (AWS native) or

Databricks or Polybase (Azure native). Other cloud-based vendor data warehouse platforms include Snowflake and Panolpy.

The implementation of a data warehouse project can either be a traditional approach or a modern approach. A traditional approach is an approach where ETL processes are needed to be defined and implemented before moving the data into the final data warehouse repository. The major challenge here is scalability and maintenance. Notwithstanding, most organizations still work with the traditional approach. The advent of a modern approach to data warehouse project does not seem to have altered or taken away the traditional approach off the market. It is important to have skills in the data pipeline and ETL processes. In modern approach to data warehouse, emphasis is not on the ETL process or as much as the way the PANOLPY vendor defined it.

### **3.1 Business Intelligence Tools vs Data warehouse**

Data warehouse implementation has been revolutionized over time. Data warehouse serves as a fabric foundation where data analytics can be performed. However, it is also possible to conduct data analysis or business intelligence activity without having a data warehouse as a foundation. This has been made possible thanks to the evolution of business intelligence tools like PowerBI, Qlikview, Tableau, Cognos among several others.

According to Data Warehousing Institute (Data Warehouse Institute Faculty Newsletter, 2002), business intelligence has been defined as the processes, technologies, and tools needed to turn data into information, information into knowledge, and knowledge into plans that drive profitable business action. Business intelligence (BI) encompasses data warehousing, business analytic tools, and content/knowledge management. From this definition, it is apparent that data warehousing is the foundation of business intelligence. It precedes other tools that can be used for business intelligence. The value of BI could be realized in the context of profitable business action. This implies that if the knowledge that can be used for profitable decision is properly used, the beauty of business intelligence becomes crystal clear. Conversely, if knowledge that should be used for profitable business action is ignored, the practice of business intelligence would be of little value. When BI is properly utilized in a business decision, it can lead to increased profitability, decreased cost, and improved customer relationship management. Also, business intelligence exposed areas where risk within business needed to decrease. For example, business

intelligence activities reveal areas where marketing activities needed to be campaigned because data has been intensely analysed through the BI tools.

The challenges however with conducting data analysis or BI without a data warehouse is about the singleness of data or a single source of truth. The essence of business intelligence is to ensure that consistency and a single data source when conducting data analysis. This will ensure that decisions are not based on an incorrect data source or political response. BI tools like Power BI or Qlikview however do not have the capability to hold different data together in a denormalized form. This is where the separation between business intelligence tools and data warehouse becomes obvious. Hence authenticity and reliability of business intelligence and data analysis conducted based on data warehouse is superior to using tools Power BI only. Business intelligence tools are built on the foundation of a data warehouse.

### **3.2 Application Database vs Data warehouse**

In principle, the data warehouse is also a database. In web application or other data store, a database serves as a repository where customers' information or other important information central to transactions, organization or otherwise are stored. A database captures real-time online processing information or data. It is imperative to emphasize that this study is limited to a relational database. There are other databases that are non-relational. Example in this category would be Cognos, MongoDB or Redis.

Data warehouse on the other hand captures the historical data that comes from data store of an organization. The tiny line between application database and data warehouse comes from their purposes and their use. In the application database, the data captured during the transaction are captured in real-time. The essence of capturing data in real time could be for business growth, customer security and tracking of changes that are taking place within the business life itself. In fact, it is also possible to query the application database for data analysis. Querying the application data for data analysis however comes at a greater price. This is the tiny yet important difference between application database and data warehouse. When SQL queries are run against application databases, databases are not left in their original state because of fragmentation that occurs when queries are run. The greatest change that happens to the database is degradation

over time. The response time of database to respond to queries degrades over time. Besides, writing a bad query could also be problematic for the application database.

One of the ways to solve challenges attributed to querying application databases is to develop a data warehouse. Besides the ability of a data warehouse to solve the application database degradation, it could also be used to meet business needs through business intelligence application and data analysis.

### 3.3 Data Warehouse Processes

Data warehouse implementation follows sets of principles and rules (as defined by the author of this work). This work will make use of six D's principles that represent appropriate steps for data warehouse processes and implementation. They are:

#### a. Defining Business Case

Data warehouse is intended to be a massive, comprehensive receptacle that is crucial to the operation of a business. It should contain all the organizations' data regardless of its source or its destination. It is housed in a single repository. Implementation of data warehouse project is not just to satisfy some fancy needs. Data warehouse project must be backed by a valid business need or case. Snowflake in its blog itemized five priorities needed when implementing a data warehouse. This includes:

- The need to access and act on data in real time
- The search for a holistic view of the customer
- Understanding data lineage to ensure regulatory compliance
- Enabling non-technical people to query data quickly and cheaply
- The need to bring data together into a single location (Snowflake's blog).

#### b. Defining Core Business Processes

Business processes are the operational activities performed by the organization. Such activities could include taking orders, the processes of insurance claims, student registration or taking a snapshot of database account every week or month. For a clearer view, business processes in an organization could be sales processes, marketing processes, finance processes among several others. In data warehouse development, business processes or events can be used to capture performance that translates into facts in a fact table. Business process results are the pragmatic way to generate fact tables



during data warehouse development processes. It is imperative to choose or define an appropriate business process during the development of the data warehouse. The choice of appropriate business processes defines a specific target and allows the grain, dimensions, and facts to be declared. Each business process corresponds to a row in the enterprise data warehouse bus matrix.

c. Defining the Data Source

In principle, appropriate core business processes determine the data source for the data warehouse project. However, it goes much deeper. Efforts must be made to identify what kind of data would go into the data warehouse repository. It has been said that source identification is critical to the success of a data warehouse project. The essence of this phase is to ensure that ideal data are moved into data the warehouse. Inability to define appropriate data source or narrow the scope of data moved into the data warehouse could result in expensive operation in the future and can also impact the performance of the data warehouse. It is important to consider the following questions when determining the sources and costs of data for the Data Warehouse implementation:

- Where does the data come from?
- What processes are used to obtain the data?
- What does it cost to obtain the data?
- What does it cost to store the data?
- What does it cost to maintain the data (W.H. Inmon, Derek Strauus, Genia Neushloss,2008.)?

The ability to maintain balance between data identification and being bogged by detail is crucial. Excessive details of the data source are not important in this phase. What is of great significance is that the right data are identified, and appropriate data are moved as defined.

d. Defining Model for the Data Warehouse:

Three important concepts that must be fully understood when going into data warehouse implementation. These are conceptual data modelling, logical model, and physical model. These three concepts represent how data are properly modelled before going into actual implementation.

1. Conceptual Modelling: Conceptual model represents the highest-level relationship between the different entities. When conducting conceptual modelling, no specific attributes are specified, and no primary keys are defined.

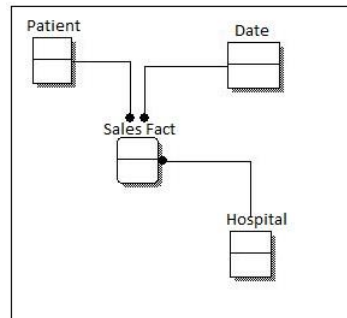


Figure 3.1: Conceptual Modelling - <https://www.guru99.com>

2. The logical data model describes more details compared to the conceptual data model. In the logical model, more information is available. Although it does not describe how the model is implemented. The main feature in this phase is that attribute keys and primary keys are specified. Foreign keys are also specified. Instead of normalization, denormalization occurs at this level.

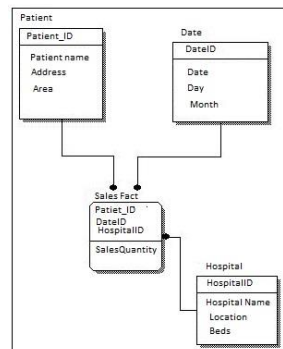


Figure 3.2: Logical Modelling - <https://www.guru99.com>

3. The physical data model describes how the model will be built in the database. In the physical model, all table structures that includes column name, data type, constraints, primary key, foreign key, and their relationships are shown. The detailed attributes in the physical model show tables and columns. Foreign keys are specified to represent

the relationship among different tables. Denormalization of table based on requirement is done here. Denormalization may increase the chances of data redundancy but it will speed up data retrieval mechanism.

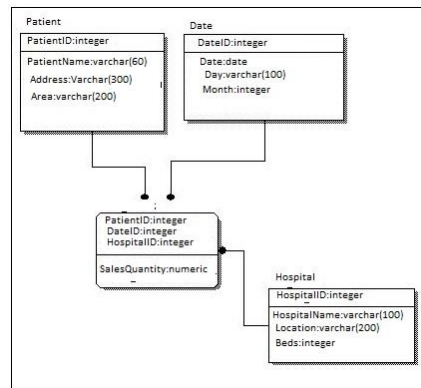


Figure 3.3: Physical Modelling - <https://www.guru99.com>

#### Developing ELT Process:

ETL (Extract, Transform, Load) is a process that takes the required data from the source for analysis. The analysed information is to be transformed into the desired formats that meet the business requirement. The goal of ETL processes is to summarize data, reduce it, and improve its performance for specific business needs. In the ETL process, data are integrated from a variety of different data sources. The ETL process must be treated as an infrastructure and it is. It must be carefully planned and tested to ensure that data are in the correct transformation as planned and they meet specific goals that they were meant for. ETL processes can be automated. In addition, ETL pipeline can be built in batches (using traditional approach and tools like SSIS or Data Stage. AWS/AZURE process e.g., REDSHIFT or DATABRICKS provide enhanced capabilities on cloud infrastructure).

#### e. Defining Actual Implementation:

The final step during the data warehouse project is to schedule moving data from the staging area to data warehouse repository. On the other hand, the staging area might be excluded from the implementation. The process in this phase can be scheduled as on-demand or automated.

The overall data warehouse development process can be seen from the figure below:

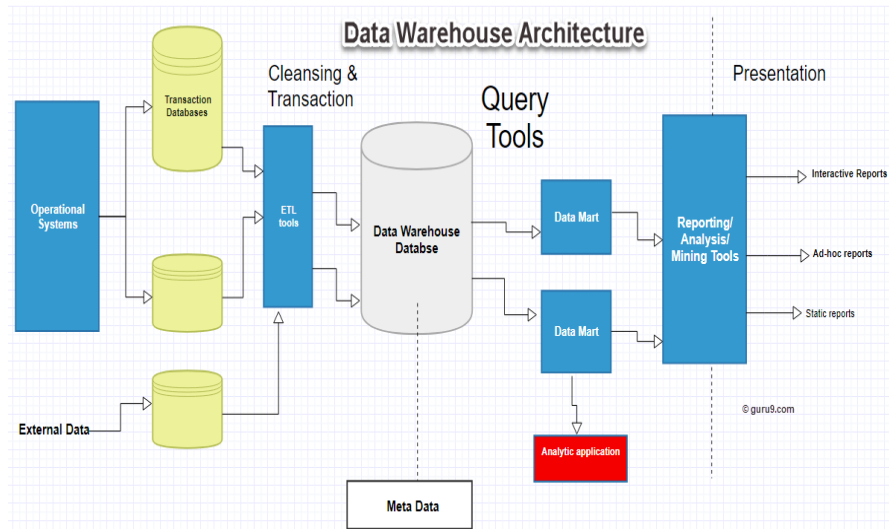


Figure 3.4: Data Warehouse Architecture –  
<https://www.guru99.com/data-warehouse-architecture.html>

### 3.4 SQL Server Integration Services

SQL Server Integration Services is a platform for building enterprise-level data integration and data transformation solutions. Microsoft introduced Integration Services in Microsoft SQL Server 2005 and it is considered the successor to Data Transformation Services (DTS) in SQL Server. The introduction of SQL Server Integration Services has been considered a game changer in data solutions because of the uniqueness of the SSIS interface and ease of use.

SQL Server Integration Services has enabled developers, business analyst and stakeholders to solve complex business problems by copying or downloading files, loading data warehouses, cleansing, and mining data, and managing SQL Server objects and data. The simple architecture of SQL Server Integration Services can be found below in Figure 3.5.

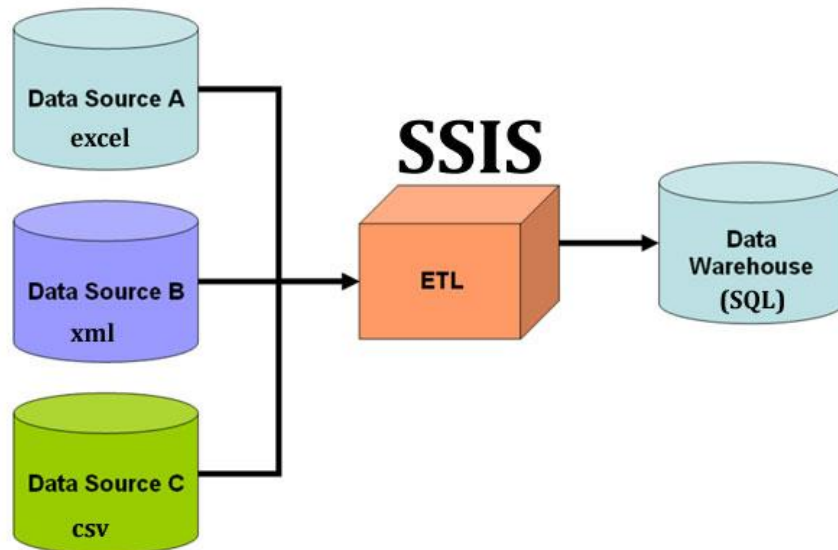


Figure 3.5: SSIS Architecture - <https://www.learnmsbitutorials.net/etl-process-ssis-example.php>

### 3.4.1 Features of SSIS

The most common feature of SSIS is the movement from one source to another. There is plethora of reasons for moving data from one source to another. Moving data across platforms could be due to data migration for business reasons, extracting data for use externally or creating sample test data source for the development environment or loading the data into business intelligence application systems.

Another most powerful feature of SSIS is the ability to read data from different sources. SSIS has capabilities to read source data from ADO Net Source, CDC source, Excel Source and OLE DB Source. Other data sources where SSIS can read data from include raw file source and XML source.

There are many data transformation tools within SSIS. These include aggregate data functions, balance a data distributor, data conversion function and conditional tools. Others include data streaming destination, derived columns, HDFS File Destination, lookup, merge, merge join, multicast among many others. The essence of these tools is to enable developers to perform different tasks within the SSIS setup. Microsoft has from time to time updated these sets of tools to enhance efficiency for relative ease of use of SSIS.

### 3.4.2 Advantages of SSIS

Microsoft SQL Server 2005 Integration Services (SSIS) is the toolset used to help implement data integration process applications among business application system files and databases. Although, SSIS has been described as an ETL tool, it is however more than a simple extract, transform, and load (ETL) process. The economic importance of SSIS can be summarized below:

- a. SSIS enables database administrators and application developers to design, implement, and manage complex, high-performance ETL applications.
- b. SSIS enables developers to select data from one or different sources and perform functions and operations necessary for data integration application. These functions and operations include join, merge, argument, derive, calculate among many others.
- c. SSIS also provides procedures to automate many of the administrative functions for SQL Server databases, tables, On-Line Analytical Processing (OLAP) Cubes, and many other functions for components of SQL Server 2005.
- d. In Data warehouse development, data migration application integration or business intelligence projects approximately 60 percent to 80 percent of all time is expended on the ETL phase. Deployment of SQL Integration Services has proven to reduce time, effort, and cost during the ETL phase. For example, automation of ETL jobs could reduce the repetitive tasks (Ashwani Nanda 2006).

### 3.4.3 ETL Set-up Within SSIS

The ETL works remains one of the most important phases either in data warehousing project or data migration or data distribution. The essence of the ETL job is to standardize data and ensure that the recipient destination received the data as specified within the business rule. This will ensure consistency and clean data are moved to the destination system without loading junks and unnecessary data. The issue of scalability, performance and automation are paramount when designing ETL job.

### 3.4.4 Package Definition

To set-up ETL job in SSIS, a new data package needed to be defined. The package must also be defined appropriately. Please see figure 2.6 below:

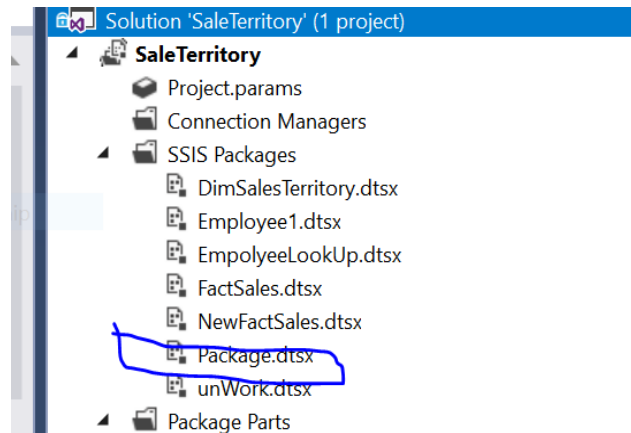


Figure 3.6: Package Definition – Author’s Library

### 3.4.5 Connection Manager

The connection manager in the ETL set-up enables the developer to define both source and destination where they would be taken from and where they would end, respectively.

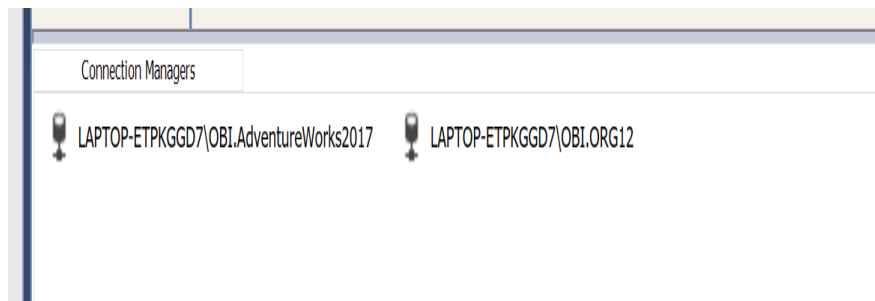


Figure 3.7: Connection Manager – Author’s Library

### 3.4.6 Other Component

SSIS toolbox comprises control flow, data flow, parameters, event handlers and package explore. Each of these components are essential in the development of adequate ETL jobs.

For instance, in the control flow task definition would be defined in from that space. Two possible tasks are possible in the control flow task definition. These are data flow task and execute SQL task. Once the appropriate task has been defined and configured, the flow goes from control flow to data flow. An example of a task defined in control flow can be found below:

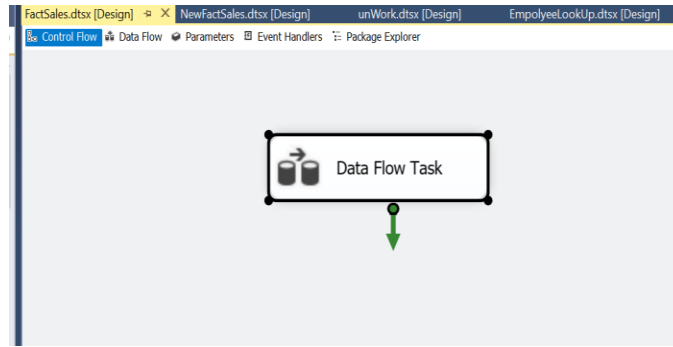


Figure 3.8 Data Flow Task – Author’s Library

In Data Flow, detail steps of ETL are defined. Example of this can be found below:

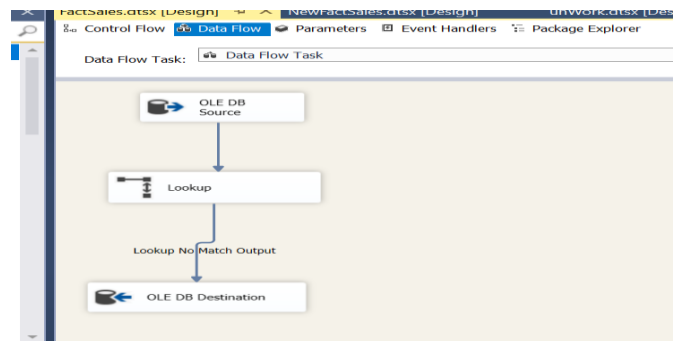


Figure 3.9 ETL Detailed Step – Author’s Library

Finally, the package needs to be saved, run, and deployed.

### 3.5 Microsoft Analysis Services (SSAS)

Analysis Services is an analytical data engine used in decision support and business analytics (Business Intelligence A-Z, 2017). SSAS support reporting tools like Power BI, Excel and SSRS. It has extensive application in data analysis and business intelligence. Data slicing and dicing is possible because SSAS is the engine that runs in the run background especially in online analytic



processing (OLAP). It is part of Microsoft SQL Server and helps perform analysis using various dimensions (guru99.com).

According to Microsoft articles on SSAS (2018), SSAS is a scalable, reliable, and secure enterprise class multidimensional database server. SSAS consist of architecture that can be scale-out and scale-up. It also helps in terms of very large database capabilities. (Microsoft Articles on SSAS, 5<sup>th</sup> February 2018)

Analysis Services is available on different platforms:

- a. SQL Server Analysis Services: This can be installed on-premises or VM server machines. SSAS support both the tabular and multidimensional models. It also supports data mining or power pivot for SharePoint.
- b. Azure Analysis Services: This is created as Azure resources, Azure Analysis Services server resources which support tabular models. (Microsoft Articles on Install SQL Server Analysis Services, 29th January 2020).

The architecture of SSAS consist of:

- a. RDBMS: The data from different sources like Excel, Database, Text, others can be pulled with the help of the ETL tool into the RDBMS.
- b. SSAS: Data from RDBMS is pushed into SSAS cubes by using analysis services projects in SSDT. The cubes will create an analysis database, and once the analysis database is ready, it can be used for many purposes.
- c. Client: Clients can access data using Dashboards, Scorecards, Portals etc.

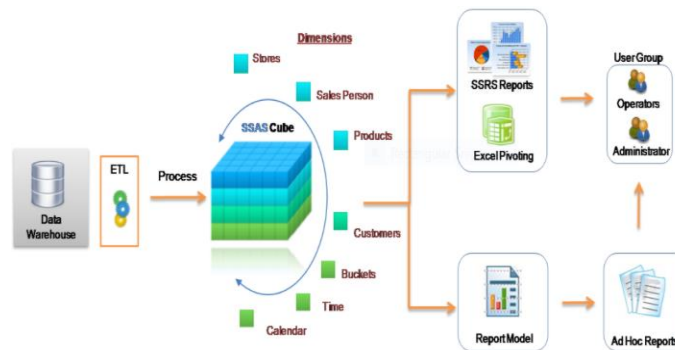


Figure 3.10: SSAS Architecture - <https://www.codeproject.com/Articles/658912/Create-First-OLAP-Cube-in-SQL-Server-Analysis-Serv>

### 3.6 SQL Server Reporting Services

Naresh (2019) defined SQL Server Reporting Services (SSRS) as a reporting software that allows end-users to produce formatted reports with tables in the form of data, graphs, images, and charts. These reports are hosted on a server that can be executed at any time using parameters defined by the users. It is part of the Microsoft SQL Server Services suite (Naresh Beniwal, What is SSRS, 2019?).

SSRS is an enhanced tool and possesses the ability to process report both on tabular and multidimensional data model. SSRS report gives the business user a better and accurate decision-mechanism platform. It also gives the end-user the ability to bypass IT professionals when creating the report. One of the unique features of the SSRS report is that it can be accessed over the internet. This feature allows reports to be exported in different formats. Reports can also be delivered using email. Finally, SSRS provides a host of security features, which helps the administrator to control, who can access which report (Guru99. Article on SQL Server Reporting Services Tutorial for Beginners).

The SSRS service provides an interface into Microsoft Visual Studio. This allows either developers or SQL administrators to connect to SQL databases and use SSRS in their work as business case specifies. It also provides a 'Report Builder' tool for less technical users to format SQL reports of lesser complexity.

Features of SSRS includes:

- a. It allows developers to create ad-hoc reports and save them to the server
- b. It can retrieve data from managed, OLE ODBC, and DB connections
- c. It is possible to create custom controls by using the report processing extension.
- d. It can store reports, and process custom reports
- e. Key Performance Indicator (KPI) data can be displayed using the chart and gauge control feature.

The schematic architecture of SSRS can be found below:

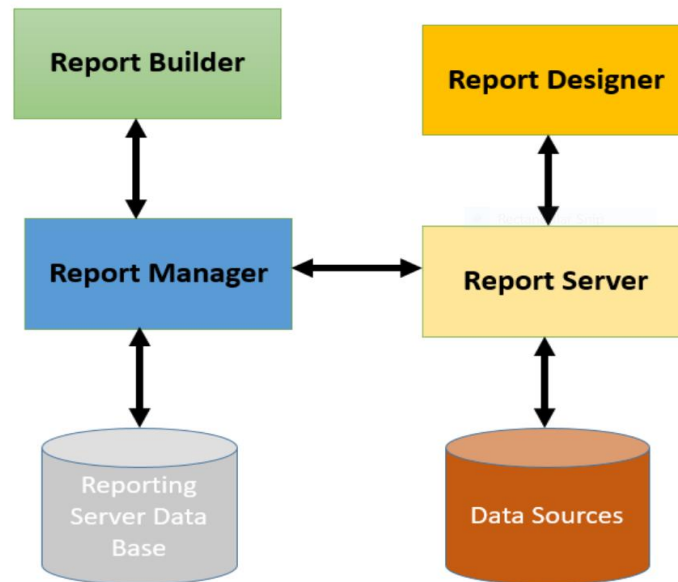


Figure 3.11: SSRS Architecture - <https://www.guru99.com/ssrs>

## Types of Reports in SSRS

<b>Report Type</b>	<b>Explanation</b>
Parameterized reports	This type of report uses input values to complete the report or data processing.
Linked Reports	A linked report offers a point to an existing report. It is derived from an existing report.
Snapshot reports	A snapshot report contains layout information and query results which can be retrieved at a specific point in time.
Cached Reports	The cached report allows you to create a copy of the processed reports. They are used to enhance performance by reducing the number of processing requests and time to retrieve large reports.
Drilldown Reports	Drill down reports help to hide complexity. It allows users to toggle between hidden reports, items to control and how much detailed to be displayed. It retrieves all possible information that can be shown in the report.
Drillthrough Reports	Drillthrough reports are standard reports which can be accessed through a hyperlink on a text box in the original report. It works with the main report and is the target of a drill through an action for a report item like a placeholder text or a chart.
Sub-reports	As the name suggests, sub-reports are reports which displays another report inside the body of the main report.

Table 3.1 - <https://www.guru99.com/ssrs-tutorial.html>

## 4 Data Warehouse Implementation

This chapter deals primarily with the empirical part of this thesis project. The actual implementation part of this work was achieved in this section and detailed report is provided.

It is important to state the goal of the work again so that objectives can be re-emphasized. The objectives are:

- a. How to effectively enhance the performance of ETL processes during the implementation of data warehouse project.
- b. How to enhance the scalability of ETL processes during the data warehouse project.
- c. Expanding the knowledge-based so that author of this work can have a thorough understanding of the implementation of data warehouse project.

### 4.1 Resources

The resources utilized for this work are located within the Haaga Helia University of Applied Sciences (UAS) online location. The following resources are central to the implementation of this work which includes:

- a. OnDemand Environment: Ondemand environment is part of the Haaga Helia UAS online environment. This environment contains resources like Microsoft Management Visual Studio, Microsoft SQL Server Integration Services (SSIS), SQL Server, and Power BI among many others.
- b. SQL Server: In this work, SQL Server is extensively used during this work. Both the data sources and eventual data warehouse that would be were housed in Microsoft SQL server. The SQL server that is used in this work can be found within the Ondemand environment.
- c. Microsoft SQL Server Integration Services (SSIS): In this work SSIS tool is used for extract, transform and load (ETL) purposes. It is also used to move the data from the sources table to the target.
- d. Data Source: The data sources used in this work came from NAV2015. The database is BIG4TF023\_Rehn\_2022. It was from this data source that data were moved to the final eventual database target.

## 4.2 Process of Implementation

In chapter two of this work, the under listed process of implementing a data warehouse was established. These are:

- a. Defining Business Case
- b. Defining Core Business Processes
- c. Defining the Data Source
- d. Defining Model for the Data Warehouse
- e. Developing ETL Process
- f. Defining Actual Implementation

In this thesis project work, step one is missing since the business case was not defined for the project work. Otherwise, all other steps were duly followed.

## 4.3 Defining Business Processes

The business processes that are used in work includes:

- a. Sale Process
- b. Purchasing Process

Other derived processes that were connected to either sales or purchasing process are:

- c. Item sub-process
- d. Customer sub-process

In all four business processes are utilized in this work.

## 4.4 Data Source and Exploration

One of the most important job in data warehouse development project is data exploration. Data exploration allows the developers opportunity to understand data types, data length or precision, and how to plan the target database.

In this work, the standard in-built function of Microsoft Management SQL Server was used to perform data exploration on the data source NAV2015/BIG4TF023\_Rehn\_2022. For example, the query below was run to understand the source data in the Item table. The query is as follows:

```
Sp_help '[dbo].[01 Cronus IntW1$Item]'
```

The result of the query has been displayed in the figure below.

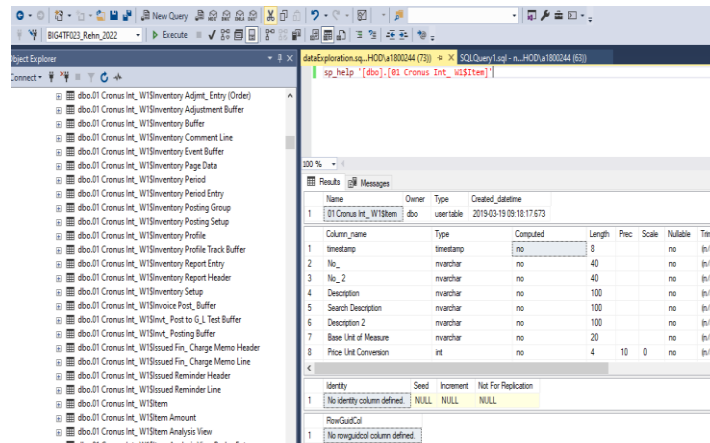


Figure 4.1 Query Result - Source: Author's Library

From the above figure, it was easy to understand the Item table. The query provided the data schema of the Item table such as data types of each column and data precision. These information were crucial and necessary for planning the target table in the target database. Thus, when creating table Item in the destination table, due attention was paid to its properties in the source table. Besides, a clear understanding of table properties was required during data mapping in the ETL processes. If source data and destination data were of different data types, data mapping would fail ultimately, while data loading would not succeed.

#### 4.5 Data Destination and Architecture

From the data sources of this work, NAV/BIG4T023\_Rehnn\_2022m four tables were migrated to the target destination database. They were:

- a. SalesHeader
- b. ItemHeader
- c. Purchaseheader
- d. CustomerHeader

In the destination database called AbitheseisDB, four tables of similar properties to the source table were created. These tables received the data from the source table. The four target tables include:

- e. DimSalesheader
- f. DimItemHeader
- g. DimPurchahseHeader
- h. DimCustomer

The last table in the AbiTheseisDB that did not exist in the source was DimDate. The essence of DimDate was to give time intelligence to data in the data warehouse. Besides, DimDate was a constant table that must exist in every data warehouse development project. The final architecture of this data warehouse can be found below:



Figure 4.2: Final Architecture - Author's Library



## 4.6 ETL Processes

ETL processes represent a crucial phase during the implementation of a data warehouse project. The essence of the ETL process was to collate (in case of data source was from different sources – usually called data lake in cloud platform) and standardize the data before moving them into the destination table. Standardization usually depends on the business rules defined for the data warehouse project and needs of data analytics or data scientists etc.

Many tools that can perform ETL job. Chief among them include Microsoft Integration Services (SSIS), Cognos, Informatica. The task of loading data into data warehouse for this work was achieved by using the SSIS tool. The decision to use SSIS tool is simple because the tool is free and readily available in the Haaga-Helia University of Applied Sciences Ondemand premises. Also, SSIS tool is part of the Haaga-Helia University Microsoft stack.

### 4.6.1 SSIS Set-up

In the SSIS set, four packages were created namely:

- a. SalesHeader
- b. ItemHeader
- c. Purchaseheader
- d. CustomerHeader

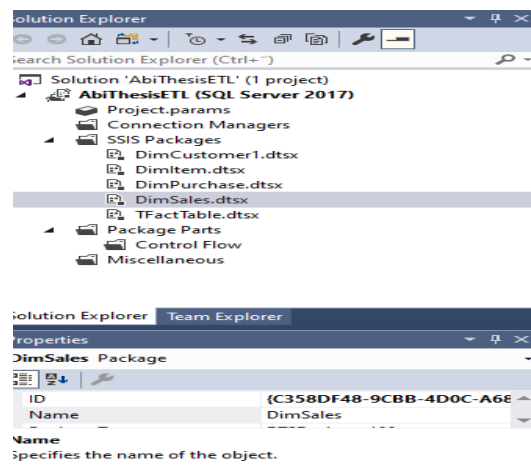


Figure 4.3: SSIS Set-up - Author's Library

To move the data from the source database and to the destination database, two connection managers were created. The first connection manager connected the SSIS tool to the NAV2015 database which was the source database.

The second connection manager is connected to the target or destination database. The connection managers that were created includes:

- a. Source: nav2015.haagahelia.amk.BIG4TF023\_Rehn\_2022
- b. Target: nav-bi.haagahelia.amk.AbiTheseisDB

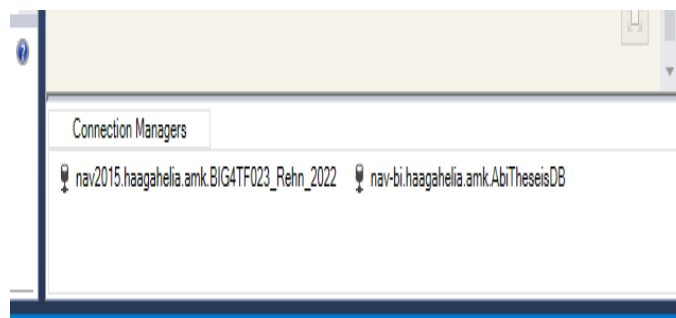


Figure 4.4: Implementation Connection Manager - Author's Library

#### 4.6.2 Loading Dimension Tables

To load the dimension table, two look-up transformation tools were executed. Besides, the DimStaging table was created. The first look-up transformation tool performs the initial loading while the second one performs the update. The update look-up transformation tools initially load the DimStaging table. DimStaging table in turn loads the target database AbiThesis.

Thus, first after completing the SSIS set up as was demonstrated in the previous section, the whole packages were run.

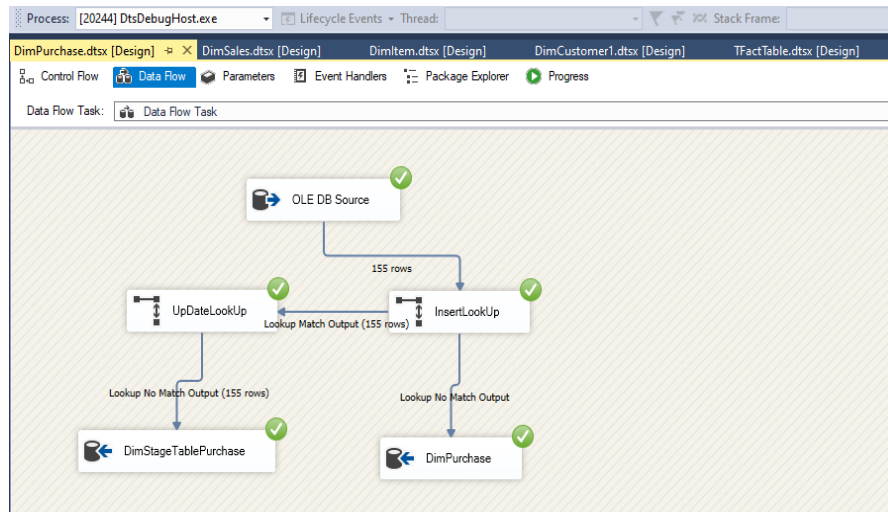


Figure 4.5: Completed Packages - Author's Library

In the Purchase package that was taken as an example, the green button indicated that the package run was successful. It further indicated that data were successful moved from the source database to the intended destination. The intended destination was AbiThesisDB.

### 4.6.3 Loading Fact table

Fact-table consists of facts, metrics, or measurement of a business process. The fact table is usually located at the centre surrounded by dimension tables irrespective of whether star or snowflake schema was implemented.

The order of loading a fact table into a data warehouse usually begins with loading the dimension table. This process had been implemented in the previous section. The first-time fact table is being loaded, the primary keys in the dimension tables had to be dropped and re-enforced after data loading had been completed. To load the fact table for this work, a copy-column transformation tool was utilized. Fact table was loaded once. The architecture for fact table can be found in the Figure 4.6 below. The copy-column tool was used to move data from source table to destination.

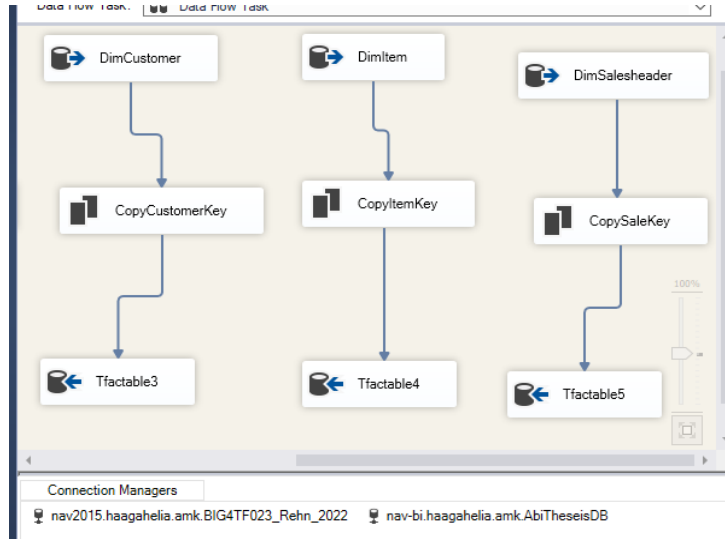


Figure 4.6: Fact Table - Author's Library

## 5 Discussion of Result

The focal point of this work rested on three parameters namely:

- a. Performance of ELT processes
- b. Scalability of ETL processes
- c. how to move data into the data warehouse and synchronize them and same objective for learning purposes

### a. **Performance**

The performance of ETL processes is an important parameter during the design and implementation of the data warehouse. The essence of performance is to reduce the time taken to load from source table to destination.

The performance for this work design was achieved by using execute SQL task when updating or loading new data into the data warehouse. Execute SQL task have the capacity to implement bulk loading. It can also load and move data up to fifteen million per minute at once.

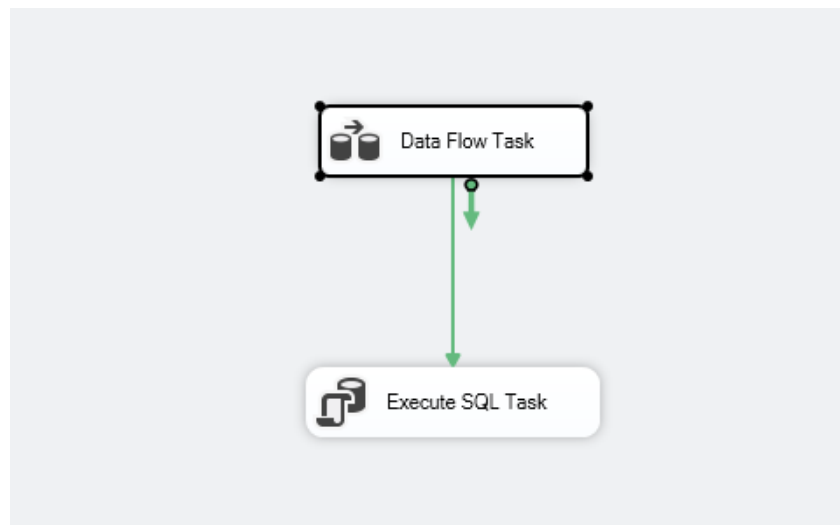


Figure 5.1: Execute SQL Task - Author's Library

Sample script inside Execute SQL Task:

```
Update A  
set  
A.[Pay-to Name] = B.[Pay-to Name],  
A.[Pay-to City] = B.[Pay-to City],  
A.[Buy-from Vendor Name] = B.[Buy-from Vendor Name],  
A.[Currency Factor] = B.[Currency Factor]  
From DimPurchse as A Join DimStageTablePurchse as B  
on A.No_ = B.No_  
  
go  
delete from DimStageTablePurchse  
  
go
```

Thus, a large volume of data can be loaded into the data warehouse by using executing SQL tasks.

#### **b. Scalability**

Scalability described how easy and cost effective it is to expand the overall solution of this design. In scalability, we discussed how easy or difficult it was to include additional resources into the system or how scalable was the entire solution. The scalability designed for this work was achieved by using look-up transformation and execute SQL task tool to move data.

Two separate look-up transformations were implemented. They were insert and update lookup, respectively. The primary job of insert look-up was to load the first time while update look-up performs any update on the data. If there was any late arriving dimension data, both insert and update look-up would ensure that data found its place without the possibility for any error.

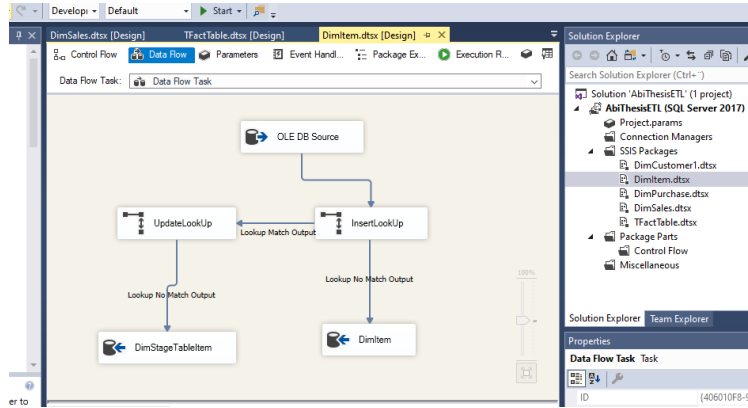


Figure 5.2: Dimension Table - Author's Library

**c. Data Movement and learning curve**

The whole process of this project was to move data from a critical database e.g., Online transaction processing (OLTP) system to online analytical processing (OLAP) system. One of the values here was for the author to gain hands-on experience on the implementation of the data warehouse.

With the context of this work, data was moved from nav2015.haagahelia.amk database. This database represents the source database. The data was moved to navbi.haagahelia.amk.AbiTheseisDB database. The database herein refers to the data warehouse that was implemented. The tool used to move those data was Microsoft SQL Server Integration Services (SSIS). The final architecture of the data warehouse can be found below:

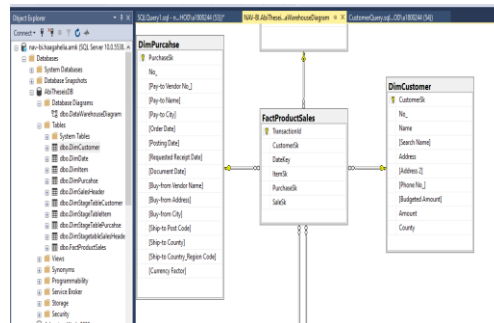


Figure 5.3: Final Outlook - Author's Library

We have four dimensions tables and one fact table by using star schema approach.

## 5.1 Test

The validity of data loaded into the data warehouse was verified by running the SQL queries against the tables. Two tests were run to ascertain whether the product is functional or not. These tests were initial load and update tests, respectively.

### i. Initial Load Test:

The initial load test validates whether there are data in the data warehouse or not. Hence, the following queries were run against the AbiThesisDB to test whether the data was moved successful or not. Please the queries below:

#### a. Query against Customer Table

```
Select
top(4) CustomerSk,
Name,
Address
From [dbo].[DimCustomer]
```

#### Result:

##### **CustomerSK Name Address**

- 1 Spotsmeyer's Furnishings 612 South Sunset Drive
- 2 Progressive Home Furnishings 3000 Roosevelt Blvd.
- 3 New Concepts Furniture 705 West Peachtree Street
- 4 Candoxy Canada Inc. 18 Cumberland Street

#### b. Query against Sales Table

```
Select
top (4) SaleSk,
[Bill-to-Address],
[OrderDate]
From [dbo].[DimSalesHeader]
```

#### Result:

##### **SaleSk Bill-to Address Order Date**

- 1 192 Market Square 2020-04-05 00:00:00.000
- 2 28 Baker Street 2020-05-29 00:00:00.000
- 3 25 Water Way 2020-05-29 00:00:00.000



These two queries validated that new data now exists within the AbiThesis database.

ii. **Update Test:**

The second test was to check if when new data is added to the frontend application of NAV, that such data could be updated to the AbiThesisDB database or not. In conjunction, with the thesis advisor, new data was added to the front-end application of the NAV2015 database. The original script from SSIS was run again and the new data that added could be found in AbiThesisDB. Please see the figure for clarifications.

166	166	LSU-8	Middletone spea...	MIDDLETONE SPEA...	21.00	0	15.00	15.00	
167	167	SPK-100	Spike for LS-100	SPIKE FOR LS-100	21.00	0	15.00	15.00	
168	168	GAMMA	Gamma for task	NULL	3000.00	0	300.00	0.00	
169	169	70070			0.00	0	0.00	0.00	
170	170	70125	BIKE	BIKE	0.00	0	0.00	0.00	
171	171	70126	Paint, sunflower ...	PAINT, SUNFLOWER...	2.70	0	1.40	0.00	VC
172	172	RALFABITEST	Testproduct	TESTPRODUCT	0.00	0	100.00	0.00	

Figure 5.4: Update Test - Author’s Library

The new data that was added from the front-end application was item number 172.

## 5.2 Challenges

In any project, there are challenges whether foreseen or not that impact on the project life cycle. This thesis encountered its own fair share of difficulties and challenges.

One major challenge that occurred this the lifetime of this work was inability to accessed due to implementation of new sign-in system by the Haaga Helia UAS. For example, the HaagaHelia UAS implemented new login system to the Ondemand environment and access in the mid-September 2020. This implementation affected the ability to access resources with the On-Demand environment. The resources situated within the Ondemand environment were central to this thesis work. It was impossible to access the Ondemand environment resource for about six (6) weeks. Access to the Ondemand environment was restored towards the end of October 2020 and the thesis work progressed.

Other challenges encountered along the way was the issue of the subject being so broad. In this regard, it requires sourcing for relevant journals, articles and books that can give more information about the subject at hand.

### **5.3 Recommendation**

The incredible pace at which organization data is growing requires a systematic approach in management and governance. Implementing a data warehouse either on-premises or cloud would afford an organization to place value both on the historical and live data. The thesis work was implemented in the data warehouse of the HaagaHelia University of Applied Sciences on-premises using the SSIS tool as means to move data.

This work opens lots of possibilities for future work in data warehouse implementation. For example, the current work did not define the business case for the project. In this context, future implementation could look at how to define the business cases and bring different stakeholders on-board during the project.

Another possibility for future implementation is to look at the complexity of the project. For instance, in the current project, the data modelling technique used was star schema. Future implementation could look at snowflake's schema approach. In such approach, the fact table should be more than one. In addition, rather than importing a whole chunk of tables, customize codes could be developed as a means for data migration.

To improve on the future implementation, it is recommended to provide students with resources to implement the same project on a cloud platform.

### **5.4 Conclusion**

The objective of this work was to build a data warehouse that can serve as a common repository for organization data. The implication of this is that data itself becomes more valuable and can be regarded as an organization asset. Implementing this work from a hands-on perspective showed that actual implementation data warehouse rested on several factors. Such factors include support from project sponsors and the ability to develop a sound process that is re-usable overtime.

In this thesis work, a data warehouse project was implemented. Data were moved from Microsoft NAV to AbiThesisDB. The focal point of the project had the ETL process where performance and scalability were in focus. Also, the synchronization of the whole data movement was given adequate attention. The learning process of the project had very informative and educative.

## References

Ashwani Nanda (2006). Hands-On Microsoft SQL Server™ 2005 Integration Services. Published by McGraw-Hill. Available at: <https://www.oreilly.com>.

Business Intelligence A-Z, (2017). Available at: <https://data-savvy.com/2017/06/04/ssas-01-about-sql-server-analysis-services/> Accessed 10th September 2020

Canan Eren Atay & Georgia Garani, 2019. "Maintaining Dimension's History in Data Warehouses Effectively," International Journal of Data Warehousing and Mining (IJDWM), IGI Global, vol. 15(3), pages 46-62, July.

David Loshin (2003) "Business Intelligence" Published by Morgan Kaufmann. Available at: <https://www.elsevier.com/books/business-intelligence/loshin/978-0-12-385889-4>

Dedić, Nedim, Stanier, Clare, Hammoudi, Slimane; Maciaszek, Leszek; Missikoff, Michele M. Missikoff; Camp, Olivier; Cordeiro, José (eds.). 2016. An Evaluation of the Challenges of Multilingualism in Data Warehouse Development. International Conference on Enterprise Information Systems, 25–28 April 2016, Rome, Italy (PDF). Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016). Available at: [https://eprints.staffs.ac.uk/2770/1/ICEIS\\_2016\\_Volume\\_1.pdf](https://eprints.staffs.ac.uk/2770/1/ICEIS_2016_Volume_1.pdf).

Douglas K. Barry & David Dick. 2013. Web Services, Service-Oriented Architectures, and Cloud Computing. A volume in The Savvy Manager's Guides. Published by Morgan Kaufmann.

Guru99.

Article on SQL Server Reporting Services (SSRS) Tutorial for Beginners Available at: <https://www.guru99.com/ssrs-tutorial.html>. Accessed 5th November 2020.

Guru99. Available at: <https://www.guru99.com/ssas-tutorial.html>. Accessed 30th October 2020.

Info Goal.

Available at: [http://infogoal.com/datawarehousing/data\\_sources.html](http://infogoal.com/datawarehousing/data_sources.html). Accessed 16th September 2020.

IBM, Redbooks. Implementing Tivoli Data Warehouse V 1.2, IBM. 2004. ProQuest Ebook Central, <https://ebookcentral.proquest.com/lib/haaga/detail.action?docID=3306810>. Accessed 7th October 2020.

Microsoft Articles on Install SQL Server Analysis Services. <https://docs.microsoft.com/en-us/analysis-services/instances/install-windows/install-analysis-services?view=asallproducts-allversions>, 29th January 2020. Accessed 1st November 2020.

Microsoft Article on SSAS.

<https://docs.microsoft.com/en-us/analysis-services/instances/high-availability-and-scalability-in-analysis-services?view=asallproducts-allversions>.  
5th February 2018. Accessed 1st November 2020.

Mubin M. Shaikh. 2013. Create and Populate

Date Dimension for Data Warehouse. <https://www.codeproject.com/Articles/647950/Create-and-Populate-Date-Dimension-for-Data-Wareho>. Accessed 10th July 2020.

Naresh Beniwal. 2019. What is SQL Server Reporting Services (SSRS)?

Available at:

<https://www.c-sharpcorner.com/article/what-is-sql-server-reporting-services-ssrs-part-1/>  
Accessed 5th November 2020.

Snowflake Data Warehouse. Available at: <https://www.snowflake.com/blog/5-business-needs-that-fuel-enterprise-data-warehouse-development/>. Accessed 9th July 2020.

The Data Warehouse Institute Faculty Newsletter, Fall. 2002.

Inmon W.H., Derek Strauss, Genia Neushloss. 2008. Chapter 1 - A brief history of data warehousing and first-generation data warehouses, Pages 1-22, ISBN 9780123743190. <https://doi.org/10.1016/B978-0-12-374319-0.00001-4>.

Inmon W.H., Derek Strauus, Genia Neushloss. 2008. Architecture for the Next Generation of Data Warehousing. Published by Morgan Kaufmann.

## Appendices

### Appendix 1. DimCustomer Script

Use AbiTheseisDB

Go

Begin Try

Drop table [dbo].[DimCustomer]

End Try

Begin Catch

/\*No Action\*/

End Catch

Create Table [dbo].[DimCustomer](

CustomerSk Int IDENTITY (1,1) Not Null Primary Key,

No\_ nvarchar (80),

Name nvarchar (400),

[Search Name] nvarchar (400),

Address nvarchar (400),

[Address 2] nvarchar (400),

[Phone No\_] nvarchar (400),

[Budgeted Amount] decimal (17,2),

Amount decimal (17,2),

County nvarchar (240)

)

Select \* From [dbo].[DimCustomer]

Begin Try

Drop table [dbo].[DimStageTableCustomer]

End Try

```
Begin Catch
/*No Action*/
End Catch
```

```
Create Table [dbo].[DimStageTableCustomer](
    CustomerSk Int IDENTITY (1,1) Not Null Primary Key,
    No_ nvarchar (80),
    Name nvarchar (400),
    [Search Name] nvarchar (400),
    Address nvarchar (400),
    [Address 2] nvarchar (400),
    [Phone No_] nvarchar (400),
    [Budgeted Amount] decimal (17,2),
    Amount decimal (17,2),
    County nvarchar (240)
)
Select * From [dbo].[DimStageTableCustomer]
```

## **Appendix 2. DimPurchase Script**

```
Use AbiTheseisDB
Go
Begin Try
    Drop table [dbo].[DimPurchahse]
End Try

Begin Catch
/*No Action*/
End Catch
Create Table [DimPurchahse] (
    PurchaseSk Int IDENTITY (1,1) Not Null Primary Key,
    No_ nvarchar (80),
```

```
[Pay-to Vendor No_] nvarchar(80),
[Pay-to Name] nvarchar(80),
[Pay-to City] nvarchar (80),
[Order Date] datetime,
[Posting Date] datetime,
[Requested Receipt Date] datetime,
[Document Date] Datetime,
[Buy-from Vendor Name] nvarchar (200),
[Buy-from Address] nvarchar(200),
[Buy-from City] nvarchar(200),
[Ship-to Post Code] nvarchar(80),
[Ship-to County] nvarchar(80),
[Ship-to Country_Region Code] nvarchar (80),
[Currency Factor] Decimal (17,2)
)
```

-----

Use AbiTheseisDB

Go

Begin Try

```
Drop table [dbo].[DimStageTablePurchse]
```

End Try

Begin Catch

```
/*No Action*/
```

End Catch

Create Table [dbo].[DimStageTablePurchse](

```
PurchaseSk Int IDENTITY (1,1) Not Null Primary Key,
```

```
No_ nvarchar (80),
```

```
[Pay-to Vendor No_] nvarchar(80),
```

```
[Pay-to Name] nvarchar(80),
```

```
[Pay-to City] nvarchar (80),
```

```
[Order Date] datetime,
```

```
[Posting Date] datetime,
```



```

    [Requested Receipt Date] datetime,
    [Document Date] Datetime,
    [Buy-from Vendor Name] nvarchar (200),
    [Buy-from Address] nvarchar(200),
    [Buy-from City] nvarchar(200),
    [Ship-to Post Code] nvarchar(80),
    [Ship-to County] nvarchar(80),
    [Ship-to Country_Region Code] nvarchar (80),
    [Currency Factor] Decimal (17,2)
)

```

```

Select * FromDimPurchse

```

### **Appendix 3. DimItem Script**

```

Use AbiTheseisDB
Go
Begin Try
    Drop table [dbo].[DimItem]
End Try
Begin Catch
    /*No Action*/
End Catch
Create Table [dbo].[DimItem](

    ItemSk Int IDENTITY (1,1) Not Null Primary Key,
    No_ nvarchar (80),
    Description nvarchar (400),
    Search_Descriptionnvarchar (400),
    [Unit Price] Decimal (17,2),
    [Price_Profit Calculation] int,
    [Unit Cost] Decimal (17,2),
    [Standard Cost] Decimal (17,2),
    [Vendor No_] nvarchar (80),

```

```
[Vendor Item No_] nvarchar (80),  
[Budget Quantity] Decimal (17,2),  
[Budget Amount] Decimal (17,2),  
[Budget Profit] Decimal (17,2),  
[Minimum Order Quantity] Decimal (17,2),  
[Order Multiple] Decimal (17,2)
```

```
)
```

-----

```
Use AbiTheseisDB
```

```
Go
```

```
Begin Try
```

```
Drop table [dbo].[DimStageTableItem]
```

```
End Try
```

```
Begin Catch
```

```
/*No Action*/
```

```
End Catch
```

```
Create Table [dbo].[DimStageTableItem](
```

```
ItemSk Int IDENTITY (1,1) Not Null Primary Key,
```

```
No_ nvarchar (80),
```

```
Description nvarchar (400),
```

```
Search_Description nvarchar (400),
```

```
[Unit Price] Decimal (17,2),
```

```
[Price_Profit Calculation] int,
```

```
[Unit Cost] Decimal (17,2),
```

```
[Standard Cost] Decimal (17,2),
```

```
[Vendor No_] nvarchar (80),
```

```
[Vendor Item No_] nvarchar (80),
```

```
[Budget Quantity] Decimal (17,2),
```

```
[Budget Amount] Decimal (17,2),
```

```
[Budget Profit] Decimal (17,2),
[Minimum Order Quantity] Decimal (17,2),
[Order Multiple] Decimal (17,2)
)
Select * FromDimItem
```

#### **Appendix 4. DimSales Script**

```
Use AbiTheseisDB
```

```
Go
```

```
Drop table DimSales
```

```
Use AbiTheseisDB
```

```
Go
```

```
Begin Try
```

```
Drop table [dbo].[DimSalesHeader]
```

```
End Try
```

```
Begin Catch
```

```
/*No Action*/
```

```
End Catch
```

```
Create Table [DimSalesHeader](
```

```
    SaleSk Int IDENTITY (1,1) Not Null Primary Key,
```

```
    No_ nvarchar (80),
```

```
    [Document Type] int,
```

```
    [Bill-to Name] nvarchar (100),
```

```
    [Bill-to Address] nvarchar (100),
```

```
    [Bill-to Contact] nvarchar (100),
```

```
    [Ship-to Name] nvarchar (100),
```

```
    [Ship-to Address] nvarchar (100),
```

```
    [Ship-to City] nvarchar (70),
```

```
    [Ship-to Contact] nvarchar (100),
```

```
    [Order Date] Datetime,
```

[Posting Date] Datetime,  
[Shipment Date] Datetime,  
[Requested Delivery Date] Datetime,  
[Promised Delivery Date] Datetime,  
[Currency Factor] Decimal (17,2),  
[Sell-to Customer Name] nvarchar (100),  
[Sell-to Address] nvarchar (100),  
[Sell-to City] nvarchar (100),  
[Due Date] Datetime,  
[Document Date] Datetime,

)

-----

Use AbiTheseisDB

Go

Begin Try

Drop table [dbo].[DimStagetableSalesHeader]

End Try

Begin Catch

/\*No Action\*/

End Catch

Create Table [DimStagetableSalesHeader](

SaleSk Int IDENTITY (1,1) Not Null Primary Key,

No\_ nvarchar (80),

[Document Type] int,

[Bill-to Name] nvarchar (100),

[Bill-to Address] nvarchar (100),

[Bill-to Contact] nvarchar (100),

[Ship-to Name] nvarchar (100),

[Ship-to Address] nvarchar (100),

[Ship-to City] nvarchar (70),

[Ship-to Contact] nvarchar (100),  
[Order Date] Datetime,  
[Posting Date] Datetime,  
[Shipment Date] Datetime,  
[Requested Delivery Date] Datetime,  
[Promised Delivery Date] Datetime,  
[Currency Factor] Decimal (17,2),  
[Sell-to Customer Name] nvarchar (100),  
[Sell-to Address] nvarchar (100),  
[Sell-to City] nvarchar (100),  
[Due Date] Datetime,  
[Document Date] Datetime,

)

Select \* From DimSalesHeader

#### **Appendix 5. DimDates Script**

BEGIN TRY

    DROP TABLE [dbo].[Dim\_Date]

END TRY

BEGIN CATCH

    /\*No Action\*/

END CATCH

CREATE TABLE dbo.Dim\_Date (

    DateKey INT NOT NULL PRIMARY KEY,

    [Date] DATETIME NOT NULL,

    [Day] TINYINT NOT NULL,

    [DaySuffix] CHAR(2) NOT NULL,

    [Weekday] TINYINT NOT NULL,

    [WeekDayName] VARCHAR(10) NOT NULL,

    [WeekDayName\_Short] CHAR(3) NOT NULL,

    [WeekDayName\_FirstLetter] CHAR(1) NOT NULL,

```

[DOWInMonth] TINYINT NOT NULL,
[DayOfYear] SMALLINT NOT NULL,
[WeekOfMonth] TINYINT NOT NULL,
[WeekOfYear] TINYINT NOT NULL,
[Month] TINYINT NOT NULL,
[MonthName] VARCHAR (10) NOT NULL,
[MonthName_Short] CHAR (3) NOT NULL,
[MonthName_FirstLetter] CHAR (1) NOT NULL,
[Quarter] TINYINT NOT NULL,
[QuarterName] VARCHAR (6) NOT NULL,
[Year] INT NOT NULL,
[MMYYYY] CHAR (6) NOT NULL,
[MonthYear] CHAR (7) NOT NULL,
[IsWeekend] BIT NOT NULL
)
SET NOCOUNT ON
TRUNCATE TABLE DIM_Date

```

```

DECLARE @CurrentDate DATE = '01-01-2000'
DECLARE @EndDate DATE = '31-12-2050'

```

```

WHILE @CurrentDate < @EndDate
BEGIN
    INSERT INTO [dbo].[Dim_Date] (
        [DateKey],
        [Date],
        [Day],
        [DaySuffix],
        [Weekday],
        [WeekDayName],
        [WeekDayName_Short],
        [WeekDayName_FirstLetter],
        [DOWInMonth],
        [DayOfYear],

```

```

[WeekOfMonth],
[WeekOfYear],
[Month],
[MonthName],
[MonthName_Short],
[MonthName_FirstLetter],
[Quarter],
[QuarterName],
[Year],
[MMYYYY],
[MonthYear],
[IsWeekend]
)
SELECT DateKey = YEAR(@CurrentDate) * 10000 + MONTH(@CurrentDate) * 100 +
DAY(@CurrentDate),
DATE = @CurrentDate,
Day = DAY(@CurrentDate),
[DaySuffix] = CASE
    WHEN DAY(@CurrentDate) = 1
        OR DAY(@CurrentDate) = 21
        OR DAY(@CurrentDate) = 31
    THEN 'st'
    WHEN DAY(@CurrentDate) = 2
        OR DAY(@CurrentDate) = 22
    THEN 'nd'
    WHEN DAY(@CurrentDate) = 3
        OR DAY(@CurrentDate) = 23
    THEN 'rd'
    ELSE 'th'
END,
WEEKDAY = DATEPART(dw, @CurrentDate),
WeekDayName = DATENAME(dw, @CurrentDate),
WeekDayName_Short = UPPER(LEFT(DATENAME(dw, @CurrentDate), 3)),
WeekDayName_FirstLetter = LEFT(DATENAME(dw, @CurrentDate), 1),

```

```

[DOWInMonth] = DAY(@CurrentDate),
    [DayOfYear] = DATENAME (dy, @CurrentDate),
    [WeekOfMonth] = DATEPART (WEEK, @CurrentDate) - DATEPART (WEEK,
DATEADD(MM, DATEDIFF(MM, 0, @CurrentDate), 0)) + 1,
    [WeekOfYear] = DATEPART (wk, @CurrentDate),
    [Month] = MONTH(@CurrentDate),
    [MonthName] = DATENAME (mm, @CurrentDate),
    [MonthName_Short] = UPPER (LEFT (DATENAME (mm, @CurrentDate), 3)),
    [MonthName_FirstLetter] = LEFT (DATENAME (mm, @CurrentDate), 1),
    [Quarter] = DATEPART(q, @CurrentDate),
    [QuarterName] = CASE
        WHEN DATENAME(qq, @CurrentDate) = 1
            THEN 'First'
        WHEN DATENAME(qq, @CurrentDate) = 2
            THEN 'second'
        WHEN DATENAME(qq, @CurrentDate) = 3
            THEN 'third'
        WHEN DATENAME(qq, @CurrentDate) = 4
            THEN 'fourth'
        END,
    [Year] = YEAR(@CurrentDate),
    [MMYYYY] = RIGHT('0' + CAST(MONTH(@CurrentDate) AS VARCHAR(2)), 2) +
CAST(YEAR(@CurrentDate) AS VARCHAR(4)),
    [MonthYear] = CAST(YEAR(@CurrentDate) AS VARCHAR(4)) +
UPPER(LEFT(DATENAME(mm, @CurrentDate), 3)),
    [IsWeekend] = CASE
        WHEN DATENAME(dw, @CurrentDate) = 'Sunday'
            OR DATENAME(dw, @CurrentDate) = 'Saturday'
            THEN 1
        ELSE 0
        END
-- [IsHoliday] = 0

SET @CurrentDate = DATEADD(DD, 1, @CurrentDate)

```



END

Select \* from Dim\_Date

### **Appendix 6. Fact Table Script**

Use AbiTheseisDB

Go

Begin Try

Drop table [dbo].[FactProductSales]

End Try

Begin Catch

/\*No Action\*/

End Catch

Create Table FactProductSales

(

TransactionId bigint primary key identity,

CustomerSk int ,

DateKey int,

ItemSk int,

PurchaseSk int,

SaleSk int

)

Select \* From FactProductSales

ALTER TABLE FactProductSales ADD CONSTRAINT

FK\_DateKey FOREIGN KEY (DateKey)REFERENCES DimDate(DateKey);

ALTER TABLE FactProductSales ADD CONSTRAINT

FK\_CustomerSk FOREIGN KEY (CustomerSk)REFERENCES DimCustomer(CustomerSk);

```
ALTER TABLE FactProductSales ADD CONSTRAINT  
FK_ItemSk FOREIGN KEY (ItemSk)REFERENCES DimItem(ItemSk);
```

```
ALTER TABLE FactProductSales ADD CONSTRAINT  
FK_PurchaseSk FOREIGN KEY (PurchaseSk)REFERENCES DimPurchahse(PurchaseSk);
```

```
ALTER TABLE FactProductSales ADD CONSTRAINT  
FK_SaleSk FOREIGN KEY (SaleSk)REFERENCES DimSalesHeader (SaleSk);
```