DATA WAREHOUSING FOR SMALL SCALE ENTERPRISE

BY

EBUNOLUWA, OLOWO ADEMOLA

17010301042

DEPARTMENT OF COMPUTER SCIENCE AND MATHEMATICS, COLLEGE OF BASIC AND APPLIED SCIENCES, MOUNTAIN TOP UNIVERSITY, MFM PRAYER CITY

September, 2021

DECLARATION

I hereby declare that this project has been written by me and is a record of my own research work. It has not been presented in any previous application for higher degree of this or any other University. All citation s and sources of information are clearly acknowledged by means of reference.

•••••

OLOWO, EBUNOLUWA

.....

Date

CERTIFICATION

This project titled, **DATA WAREHOUSING FOR SMALL SCALE ENTERPRISE**, prepared and submitted by **OLOWO EBUNOLUWA ADEMOLA** in partial fulfillment of the requirements for the degree of **BACHELOR OF SCIENCE** (**Computer Science**), is hereby accepted.

Dr F.A. Kasali

Supervisor

Dr M.O Adewole

Coordinator of Department

DEDICATION

I dedicate this project work to God almighty.

ACKNOWLEDEMENT

I want to show my appreciation to my parents and also to every member of staff that helped me to work on this project research. Special appreciation to my supervisor Dr. F.A Kasali and the head of department of computer science and mathematics department.

ABSTRACT

This project work contains five chapters and references. The chapter one includes introduction to the data warehousing and the basic concepts to why and how data warehouses can be useful to small scale businesses, the chapter two contains literature review of the project summing up all the nitty-gritties of the data warehousing solution in view ranging from the history to facts tables and variety of dimension tables, the chapter three contains the methodology to design and develop the data warehouse as well as the necessary tools used in the development process and all of the requirements for the solution to be implemented by a user, the chapter four contains implementation details of the data warehouse highlighting the four major modules of this data warehousing solution: data source module, data integration module, data warehouse and business intelligence which may be known as reporting, the chapter five contains the summary being the result of my research and conclusion listing the recommendations and round-up.

TABLE OF CONTENTS

DECLARATION	i
CERTIFICATION	ii
DEDICATION	iii
ACKNOWLEDEMENT	iv
ABSTRACT	v
TABLE OF CONTENTS	vi
LIST OF FIGURES	viii
LIST OF TABLES	x
CHAPTER ONE: INTRODUCTION	1
1.1 Background to The Study	1
1.2 Statement of The Problem	
1.3 Aim and Objectives	
1.4 Scope and Limitations of Study	2
1.5 Significance of Study	······2 2
1.6 Definition of Terms	
CHAPTER TWO: LITERATURE REVIEW	
2.0 LITERATURE REVIEW	
2.1.1 Definition of data warehouse	
2.1.2 History of data warehouse	5
2.2 Impact of Data Warehouses	5
2.2 Objectives of Droposed Data Warehouse	
2.5 Objectives of Hoposed Data Watchouse	
2.4 Inmon vs Kimoan Data warenouse:	
2.5 Data Warehousing and Data Warehouse	
2.6 Fact and Dimension Tables	
2.6.1 Fact tables	
2.6.2 Dimension tables	
2.7 Schema Types in Data Warehouse Design	
2.7.1 The star schema	
2.7.2 The snowflake schema	
2.7.3 The constellation schema	
2.8 Data warehouse types	12
2.9 Information Systems	13
2.10 Review of Related Works	
CHAPTER THREE: METHODOLOGY	
3.1 Requirement Gathering	
3.2 Architecture of The Designed Data Warehouse System	16
3.2.1 System of records (SOR)	
3.2.2 System of integration (SOI)	
2.2.2 System of analysis (SOA)	
5.2.5 System of analysis (SOA)	
3.2.4 Business intelligence (BI)	
3.3 INSTALLATION OF SOFTWARE	

CHAPTER FOUR: IMPLEMENTATION OF DESGNED SYSTEM					
4.1 Implementation Environment					
4.3 Description of Developed System for Data Warehousing					
4.3.1 Analysis of system of records					
4.3.2 Implementation of system of integration					
4.2.3.1 Creation of tables in database:	23				
4.2.3.2 Data population from SSIS:	34				
4.3.3 Implementation of business intelligence					
CHAPTER FIVE: SUMMARY AND CONCLUSION	55				
5.1 Summary	55				
5.1 Limitations					
5.2 Recommendation					
REFERENCES					
APPENDIX					

LIST OF FIGURES

Figure 4.1: sales_datasample CSV data source (Example of System of Record)	. 22
Figure 4.2: Creation of Staging Table	. 24
Figure 4.4 Creation of Date Dimension Table	. 27
Figure 4.6: Creation of Product Dimension Table	. 29
Figure 4.7: Creation of Contact Dimension Table	. 30
Figure 4.8 Display of Dimension Tables	. 31
Figure 4.9: Creation of Fact Table and Relationship	. 32
Figure 4.11: Data Flow Task to Populate Staging Table	. 35
Figure 4.12 Module-to-module Sequence in Data Flow	. 36
Figure 4.13: Flat File Configuration and Column Structure	. 37
Figure 4.14: Data Conversion for Column Datatypes	. 38
Figure 4.15: Column Mapping From Data Source to Destination	. 39
Figure 4.16: Successful Execution of Data Flow Task From CSV to SQL Staging Table	. 40
Figure 4.17: Selection of Populated Staging Table	. 41
Figure 4.18: Various Data Flow Tasks for Populating Dimension Tables.	. 42
Figure 4.19: SQL Command Accessing for Product Dimension Table	. 43
Figure 4.20: Column Mappings for Columns From Staging Tables into Product Dimension	
Table	. 44
Figure 4.21: Successful Execution of Data Flow Tasks From Staging Table to Product	
Dimension Table	. 45
Figure 4.22: Successful Execution of Data Flow Tasks From Staging Table to Contact	
Dimension Table	. 46
Figure 4.23: Successful Execution of Data Flow Tasks From Staging Table to	
StateDimension Table	. 47
Figure 4.24: Successful Execution of Data Flow Tasks From Staging Table to Date	
Dimension Table	. 48
Figure 4.25: Selection of Data From Dimension Table.	. 49
Figure 4.26: SQL Command to Populate Data From Staging Table into Fact Table	. 50
Figure 4.27: Mapping Columns From Staging Table to Fact Table.	. 51
Figure 4.28: Successful Execution of Data Flow Tasks From Staging Table to Facts Table	. 52
Figure 4.29: Data visualisation of data warehouse data with Microsoft Power BI	. 54

LIST OF TABLES

TABLE 2.1: INMON VS KIMBALL DWH ARCHITECTURE	8
Table 2.1 Slowly Changing Dimensions (Type 1)- Before Change.	. 10
Table 2.3: Slowly Changing Dimensions (Type 1)- After Change.	. 10

CHAPTER ONE INTRODUCTION

1.1 Background to The Study

Small scale enterprises contribute highly to the economic buoyancy of any country in the world. A lot of focus is now lately placed on small scale enterprises following the success of SMEs in the economic growth of Asian countries Ojo, (2003), Small scale enterprises are concerned majorly with the satisfaction of customers in their environment but they are often growth-oriented. Although, as stated in Masurel and Montfort, (2006), not every business is growth-oriented and for certain enterprises, growth is voluntary. Data management is a common problem for small scale businesses that may face a lack of skilled staff. If data management is done properly the growth of a business is partially assured. According to a business intelligence website, bi-survey.com, 18% of companies are using 20 or more data sources for decision-making, businesses use data for decision making and that is a great deal.

Data Warehouse (DWH) is a Decision Support System (DSS) that integrates data from multiple heterogeneous sources and integrates them under a unified schema into a single site (Inmon, 2002). The modern Data Warehouse depends on the problem and it has no reference architecture or standard terminology except if there are some product-oriented architectures from the vendor i.e., the customer (Santoso & Yulia, 2017). This project examines how product-oriented modern warehouses can be implemented for small scale enterprises to ensure cost optimization and improved data management. This data warehouse would combine data from operational and transactional systems (ERP, CRM, finance apps, IoT devices, online and mobile systems) to allow small business owners to access data from a variety of sources and present it in a structured format, ready for analysis and further exploitation.

1.2 Statement of The Problem

Small scale enterprise owners are faced with a lot of record-keeping challenges which is an aftermath of the traditional record-keeping method (Pen and Paper). Errors could be easily overturned without many difficulties if record keeping is enabled digitally.

1.3 Aim and Objectives

This project aims to design and implement a data warehouse for small scale business enterprises and the specific objectives of this study are:

- i. Identify a data collection model suitable for a large percentage of small-scale businesses.
- Design a data warehouse system to suit business processes of small-scale businesses generically.
- iii. Implementation of data warehouse system.

1.4 Scope and Limitations of Study

This work will research different techniques, various database applications that can be used for developing a successful data integration system. This project will be focused on the design of data warehouse systems for small scale businesses. This study is restricted to only Lagos State, Nigeria.

1.5 Significance of Study

This research work is done to reduce the impact of poor management and other grossing problems affecting their growth as to the benefit of immediate users. The benefits of data warehousing include its ability to enhance business intelligence, improvement of data quality and consistency, increased time efficiency and enhanced performance. On the other hand, Researchers will benefit from more anecdotes as regard documentations that guide to build and manage a data warehouse.

1.6 Definition of Terms

DSS- Decision support systems: A decision support system (DSS) is an information system that aids in the making of corporate or organizational decisions.

DWH- Data Warehouse: This is a repository of heterogeneous sources accumulated under a unified schema at a single site to facilitate decision making.

SSE- Small Scale Enterprise: This is a term used to describe a small, privately owned business with a small number of employees and low sales volume. This is the typical outcome of a successful small business's slow and steady expansion.

MSE- Medium Scale Enterprise: This is the typical outcome of a successful small business's slow and steady expansion.

CHAPTER TWO

LITERATURE REVIEW

This chapter sums up all the nitty-gritties of the data warehousing solution in view ranging from the history to facts tables and variety of dimension tables.

2.1 Definition of data warehouse

A data warehouse is a tool that compiles data from multiple sources so that it can be juxtaposed and analyzed for business intelligence. A data warehouse must be subjectoriented, time-variant, integrated and non-volatile (Almeida, 2017)

Subject oriented: Data in warehouses is structured around various topics like transactions, clients, and goods. This is in contrast to transactional structures, which arrange data according to business processes like order entry, inventory management, and accounts receivable. Although service-oriented architectures SOA, will modularize the services and simultaneously provide the possibility of integrating them to execute a business processes (Rojas, Arias, & Renteria, 2020)

Integrated: Data is obtained from several sources and grouped around topics. Customer data, for example, maybe derived from internal (and external) databases and combined around a customer identifier to provide a holistic view of the customer.

Time variant. A warehouse maintains historical data (i.e., it includes time as a variable). Unlike transactional systems, where only recent data, such as for the last day, week, or month, are maintained, a warehouse may store years of data. Historical data is needed to detect deviations, trends, and long-term relationships.

Nonvolatile. A warehouse is nonvolatile – users cannot replace or update the data. Non-volatility ensures that all users are working with the same data. The warehouse is updated, but through automated controlled load processes rather than by users. As stated in (Watson, 2002) Because of the first characteristic (Subject Oriented), Warehousing can be designed to suit different scales of businesses of which this project will be focusing on a small scale.

2.1.1 History of data warehouse

Data warehouses provide availability of data that may originate from different sources. Although, the idea of the data warehouse is not novel it dates as far back as the 1980s when Paul Murphy and Barry Devlin worked on something they called "business data warehouse". The aim of data warehouses has always been to ensure better decision support systems.

The history of the data warehouse is protracted. In the 1960s General Mills at Dartmouth College started joint research to develop the terms dimensions and facts. As time went on, in the 1970s A.C. Nielsen and IRI developed multi-dimensional data marts for retail sales and Bill Inmon started to define the term "data warehouse". In 1988, Barry Devlin and Paul Murphy published an article called "business data warehouse". In 1990, Ralph Kimball introduced Red Brick Warehouse software specifically for data warehouse, in 1991 Bill Inmon, introduced Prism Warehouse Manager, a software for developing a data warehouse. Ever since data warehouse development has been a subtle competitive market.

2.2 Impact of Data Warehouses

A lot of businesses benefit from data warehouses in numerous ways which include improvement of decision making, growth management and proper organization of historical data. Asides from the aforementioned benefits data warehouses seem to be behind the chain of events that leads to data analysis and data security in most enterprises.

According to (Golfarelli & Rizzi, 2009), Data warehousing has proven to be very useful in the following areas:

5

Trade: Data warehouses help to organize sales and claim analysis, customer relationship and shipment & inventory control.

Craftmanship: Data warehouses manage production cost control along with demand and supply chain.

Transportation industry: Vehicle scheduling and management.

Health care services: Patient admission and discharge analysis, as well as bookkeeping in accounts departments, are all part of the health care services.

Financial services: Risk analysis and fraud detection in financial services

Telecommunication services: Data warehouses track client profiles and do call flow analysis.

2.3 Objectives of Proposed Data Warehouse

The proposed enterprise data warehouse aims to integrate data from multiple heterogeneous sources of data from businesses, this might contain information from a POS machine, data from billing software, and unprocessed data from excel sheets for businesses.

However, for the effective functioning of the enterprise data warehouse, the data warehouse must have the following objectives:

Outline of Data Collection Model using MS Data Collector: Specification of the data collection model is necessary to help guide a continuous data control flow. The Data Collector is a SQL Server component that gathers several types of data. Data gathering can be done on an ongoing basis or according to a user-defined schedule. The data collector stores the collected data in a relational database known as the management data warehouse. (Microsoft, 2021)

Integration of Data from Different Sources with SSIS: SSIS is a component of the Microsoft SQL server that is used to perform data integration and transformation tasks. This will create a control flow diagram to outline the data integration process. (Microsoft, 2021)

Creation of Fact and Dimension Tables: Fact and dimension tables are created for the relational data. A couple of Structured query language SQL queries will be used to secure this. (Edward M. Leonard, 2011)

Training of Data Warehouse end-users: Most end-users have to familiarize themselves with the data warehouse.

Continuous Review of the Data Warehouse System: Training of the data warehouse users and administrators is very cogent because the data warehouse artefact is created in a way that it interacts with users through complex queries. Therefore, the DWH has to be updated so that more queries can be executed on it.

2.4 Inmon vs Kimball Data Warehouse:

Bill Inmon and Ralph Kimball have both contributed to data warehouse methods and evaluation of both data warehouse types is necessary. Bill Inmon's approach to the data warehouse is called the enterprise data warehouse and Ralph Kimball's approach to the data warehouse is called the dimensional data warehouse.

TABLE 2.1: INMON VS KIMBALL DWH ARCHITECTURE

Architecture	Kimball	Inmon
Model	Used dimensional model e.g., star	Used entity-relation ER model for data
	schema, snowflake schema to organize	warehouse and dimensional model for data
	data in the data warehouse. The	marts
	dimensional model is used for all of	
	the data	
Data	In Kimball's Architecture separation of	Used data marts as separation from the EDW
Separation	the dimensional data warehouse from	
	data mart is not needed	
Analytic	In Kimball's Architecture analytic	In Inmon's architecture, analytic systems can
Systems	systems access data directly	only access data via data marts

2.5 Data Warehousing and Data Warehouse.

Data warehousing includes a mechanism for end-users to gain access to data collected and stored which may include analysis and a data warehouse is simply datasets stored in form of fact tables, aggregated fact tables and lookup fact tables and an accumulation of data that facilitate the process of decision making of the management. (Qureshi & Jamal, 2012)

2.6 Fact and Dimension Tables

Every data warehouse has fact and dimension tables that describe the data warehouse.

2.6.1 Fact tables

A fact table usually has two types of columns: the foreign key which would be the primary key of the dimension tables and the measures that contain numeric values most of the time. A fact table typically contains data in an aggregated or detailed level. A fact-value can be additive, semi-additive or nonadditive. For example, in a fact table with sales items the profit can be said to be additive, the percentage gain or loss is semi-additive and the product name is nonadditive.

2.6.2 Dimension tables

The dimension table is usually structured with one or more hierarchies that classify data. In cases where the dimension has no hierarchies and levels, it is called a flat dimension or a list. The primary key of dimension tables is part of the foreign keys of fact tables. Dimension tables are smaller in size than the fact tables. There are different types of dimensions but the most determinant type is the slowly changing dimension.

Slowly Changing Dimensions (SCD): Dimensions that vary extremely slowly over time rather than according to the regular timetable.

Below are some popular ways of SCDs:

Type 0: This approach is passive. There are no changes to records that have been added.

Type 1: Without recording previous data, the old record is overwritten with fresh values.

 Table 2.1 Slowly Changing Dimensions (Type 1)- Before Change.

Cust_ID	Cust_Name	City	State
1234	John James	Parkot	Myannar

Table 2.3: Slowly Changing Dimensions (Type 1)- After Change.

Cust_ID	Cust_Name	City	State
1234	John James	Parkot	Delhi

Type 2: This method uses a version number or the start date of a column to track historical

data. The downside is that the table expands in height. Also, more space is required.

		Customer			
Cust_ID	Cust_Name	City	State	Country	
1234	ABC	Albany	NY	US	
		Customer_History			
Cust_ID	Cust_Name	City	State	Country	CreateDate
1233	ABC	Rochester	NY	US	22 nd Dec 2015
1234	ABC	Albany	NY	US	7th July 2013

Figure 2.1 Slowly Changing Dimensions (Type 2). (Nikhilesh, 2015)

Type 3: This approach is also used to track historical data by adding a new value to the column. It retains a historical limit.

Cust_ID	Cust_Name	Current_City	State	Country	EffectiveDate	Old_City
1234	ABC	Albany	NY	US	22 nd Dec 2015	Rochester

Figure 2.2 Slowly Changing Dimensions (Type 3). (Nikhilesh, 2015)

Type 4: This technique utilizes a separate table to store the previous data and holds the current data. Type 4: And they will both point to the same substitute key.

		Customer			
Cust_ID	Cust_Name	City	State	Country	
1234	ABC	Albany	NY	US	
		Customer_History			
Cust_ID	Cust_Name	City	State	Country	CreateDate
1233	ABC	Rochester	NY	US	22 nd Dec 2015
1234	ABC	Albany	NY	US	7 th July 2013

Figure 2.3 Slowly Changing Dimensions (Type 4). (Nikhilesh, 2015)

2.7 Schema Types in Data Warehouse Design

There are basically, three schema types used in data warehouse design. Star schema, Snowflake schema and constellation schema. But before that, the principle of fact and dimension tables must be clarified.

2.7.1 The star schema

In a star schema, all dimension tables are connected directly with a fact table. Hence the Dimension tables are highly denormalized in a star schema. In a star schema, only one join is required to get data from fact and dimension tables. The main advantage of star schema is that it saves space and it processes queries faster because of fewer joins.

2.7.2 The snowflake schema

Some dimension tables in a snowflake schema are not directly related to the fact table; instead, they are connected to another dimension table, which is connected to either a fact or another dimension table. Snowflake schema means that there would be more joins. The major tradeoffs of the snowflake schema are that the DWH will take slightly more space and query processing might be slower because of the many joins.

2.7.3 The constellation schema

It is feasible to build a factual constellation scheme for any star scheme (for example by splitting the original star schema into more star schemas, each of them describes facts on another level of dimension hierarchies). There are numerous fact tables in the constellation architecture which share many dimensional tables.

The major weakness of the scheme is a more difficult design as numerous versions must be examined and selected for certain types of aggregation. In addition, tables of dimension remain big.

2.8 Data warehouse types.

Data marts: collect the data from the data warehouse and send it to the OLAP layer. Now data is ready to be used by end-users as an accessible tool. A data mart is a subcategory of the data warehouse. It is specially designed for a particular line of the bus.

Operational Data Store: The OPD, which is also called the ODS, is nothing more than but data store when reporting needs are not supported by the Data warehouse or by OLTP systems. Data storage in ODS is updated in real-time. For regular operations such as keeping employee records, it is therefore often favoured.

Enterprise Data Warehouse (EDW): is a centralized warehouse. It delivers decision support services across the organization. It gives the organization and representation of data a unified approach. It also offers the ability to categorize and grant access to data according to the subject.

2.9 Information Systems.

Information systems are formal, socio-technical, organizational information systems (IS) aimed at collecting, processing, storing and distributing information. From a socio-technical point of view, information systems consist of four components: task, persons, structure, and technology. Information systems may be described as the integration of data components that offer information, store and process data, contribute to knowledge and digital products that aid decision-making. Information systems can also include data collecting. Information systems are classified into several categories, including transaction processing systems, decision support systems, knowledge management systems, learning management systems rely on information technologies, which have traditionally been designed to enable humans to perform tasks for which the human brain is not well suited, such as handling large amounts of data, performing complex calculations, and controlling multiple processes at once.

2.10 Review of Related Works.

(Santoso & Yulia, 2017), Data Warehouse with Big Data Technology for Higher Education was a research work made to tackle the lack of efficiency of big data technology to handle the load and analytic process of data. With the use-case of a university system where vast amounts of data are collected and is growing significantly. Hadoop was proposed as a big data analytics tool for data staging and ingestion. The research came to a conclusion outlining future directions relating to the development and implementation of data warehouse technology on Big Data.

In 2011 Edward Leonard of Marquette university presented a thesis called 'design and implementation of an enterprise data warehouse' for higher institutions in which he implemented an enterprise data warehouse that showed how sets of different databases go through the process of Extract, transform and load (ETL) and loaded into a star schema. The limitation of this system is that it does not offer a chance to share reports against an existing system and the DWH was not structured in a way that other institutions can use.

(Qureshi & Jamal, 2012), Taxonomy based Data Marts depicted an approach that added a new layer for categorization of queries using data warehouse technology and quickly responds to multidimensional analytic queries. The algorithm in this journal article proved to be more accurate and time-saving which was possible because of the query classification technique.

(Almeida, 2017), Concepts and Fundaments of Data Warehousing and OLAP, discussed the fundamental concepts of data warehouses and explores the concepts associated with data

warehousing and analytical information analysis using OLAP. The fundamental concepts of data warehouses and explores the concepts associated with data warehousing and analytical information analysis using OLAP.

In 2016, Hughes, R. made research for Agile Data Warehousing for the Small Enterprise. The work contained the building of different types of data warehouses using different agile methods. It was difficult to find limitations in this work because of the versatile approaches employed. Hughes concluded in his work that: The combination of XP and SCRUM could be used in the development of a data warehouse; RUP could be used in place of XP and SCRUM.

In a research work performed by Abhisek Gupta and Arun Sahayadhas (2020) a few punchy attributes were explained including Slowly Changing Dimension types, Schemas, ETL and

14

ELT architectures and MD5 (Multi-Dimensional) function. The survey established that there are still many open future aspects of research in the same area.

CHAPTER THREE METHODOLOGY

This chapter explains the methodology for the design and development of a data warehouse, the tools essential to build a user-developed solution and all needs

3.1 Requirement Gathering

The data warehouse is subject oriented meaning it must be modelled after the business process. Matter of fact this entire process is called IDENTIFICATION OF BUSINESS PROCESS in data warehousing. Identification of business process was done by an interview with organization stakeholders and by first-hand observation of the data integrated from the existing data sources. The data warehouse should provide a generic structure for SSEs and MSEs by applying the business process in sales information gathering.

3.2 Architecture of The Designed Data Warehouse System

3.2.1 System of records (SOR)

The system of record is an information storage and retrieval mechanism in a system with numerous sources of the same element, the authoritative source for a certain data element. In the case of this proposed system, it most likely contains spreadsheets, data from point-of-sale machines, and unprocessed data. The System of Integration receives each data source choice.

3.2.2 System of integration (SOI)

The system of integration consolidates the datasets from different sources into a unified schema in other words the imperative part of data warehousing is done in the system of integration.

3.2.3 System of analysis (SOA)

The availability of analysis options is done by the system of analytics. The whole process is conducted by the OLAP Server with the help of the data marts so the necessary queries can be executed on the data warehouse.



Figure 3.1: Data Warehousing System Architecture

3.2.4 Business intelligence (BI)

Business intelligence is the preferred front-end tool for decision support. It directly interacts with the OLAP server in the System of Analysis.

3.3 INSTALLATION OF SOFTWARE

Microsoft SQL Server 2019 was used for writing queries to design schemas and create joins, Microsoft Excel 2019 was used to create random values for the data warehouse which were later exported to the SQL database, Microsoft Visual Studio Community 2019 hosted the SSIS package where integration projects were performed, Microsoft Power BI 2019 was used for reporting data that was on the data warehouse and Microsoft SQL Server Integration Services Package was used to integrate the data warehouse. The development of this work was carried out on a DELL LATITUDE E6440 with the following hardware specification;

- 500GB Hard Disk
- 8 Gigabytes RAM (Random Access Memory)
- 2.6 Gigahertz Processor Speed (Intel Core i5)
- 64-bit Operating System (Windows 10)

The Installation process was done step by step.

CHAPTER FOUR

IMPLEMENTATION OF DESIGNED SYSTEM

This chapter shows the implementation of the designed system. For implementation, multiple data sources were reviewed.

4.1 Implementation Environment

The implementation for the data warehousing was done using five software: Microsoft SQL Server 2019, Microsoft Excel 2019, Microsoft Visual Studio Community 2019, Microsoft Power BI 2019 and Microsoft SQL Server Integration Services Package.

Microsoft SQL Server 2019 is a Microsoft-designed relational database management system. As a database server, it is a software product that mainly has the role of storing and retrieving data on the same or other computer via a network, as required by other software programs.

Microsoft Excel 2019 is a spreadsheet developed by Microsoft. It includes computing, graphic design tools, pivot tables and a Visual Basic for applications macro programming language. For those systems, in particular since version 5 in 1993, it has been an extremely often used spreadsheet, and Lotus 1-2-3 has been supplanted as an industry standard. Excel is part of the software package of Microsoft Office.

Microsoft Visual Studio 2019 is a Microsoft-based built-in development environment. It is used for the development of computer software, websites, web applications, and mobile applications. Visual Studio is using the Windows API, Windows Forms, Windows Presentations Foundation, Windows Store and Microsoft Silverlight software development platforms. It is capable of producing both native and managed code.

Microsoft Power BI 2019 is a Microsoft business analytics service. It seeks to offer a simple interface to allow interactive visualizations and business intelligence capabilities to construct their reports and dashboards for end-users

Microsoft SQL Server Integration Service is a component of the Microsoft SQL Server database program that may be utilized to carry out the various data migration activities is the SQL Server Integration Services. SSIS is a data integration and workflow platform. It has a data storage tool used to extract, process and load data. The program may be used also to automate SQL Server database maintenance and changes to multidimensional cube data.

4.3 Description of Developed System for Data Warehousing

The system for the data warehousing developed for small scale businesses is built after the architecture in FIGURE 3.1 Taking the prospective systems one after the other the implementation will be explained in the following subsections

4.3.1 Analysis of system of records

The implementation of the system of records was done with the Microsoft excel 2019 spreadsheet software where a certain comma separated file (csv) sales_datasample.csv which was sourced from Kaggle.io was chosen because the dataset has high similarity with small scale business operations and the dataset was analyzed. Some certain formulas were used to

generate the values

e.g.,

=RANDBETWEEN(\$A1, 50)

=RAND(()*(33-45)*15)

Other formulas were just used to manipulate the data sets and to make them structured e.g.,

```
=VLOOKUP(A13,SalesR2,2,FALSE)
=YEAR(G8)=$C$2
=SUM(M3:M503)
=IF(AND([@[Total
Sales]]>B6,[@Account]>=B7),[@[Total
```

The sales operations were analyzed to find out the measures so as to implement fact and dimension tables in the data mart. The CSV file is a data source that will provide the records and fields in the system of integration.

E	5 ×∂	· @ 6	Q 🗄	å+ ° I	i ÷						sales_dat	asample.c	sv - Excel							SwiFt	er Inc. 🛕 🖪	-	đ X
Fil	e Hom	e Insert	Page Lay	vout F	ormulas	Data Review	View	Developer	Help	Inquire	Q Tell	me what y	/ou want to d	0									A Share
Pas v	Circle Copy	v at Painter	Calibri B I <u>U</u>	! • :::	• <u>11 •</u> • <u> &</u> •		■ ≫ = €	~ c ^b Wrap ∋≣ ⊞ Merg	e Text je & Center	Ge • \$	neral • % *	▼ €0_00 00 €0] Conditio Formattir	nal Formata	as Cell Styles ~	€ ∎ Insert ×	Delete Form	Till → Cle	toSum ×	Sort & Fin Filter ~ Sele	AutoFormat		
	Clipboard	RI.		Font		Y	A	lignment		12	Number		2	Styles			Cells		Editi	ng	New Group		~
A1	Ŧ	i X	$\sqrt{f_x}$	ORD	ERNUMBE	R																	۷
	A	B	С	D	E	F	G	Н	1	J	К	L	М	N	0	Р	Q	R	S	T	UV	W	X 🛛
1 0	ORDERNU Q	UANTITY P	RICEEACIOR		SALES	ORDERDATE	STATUS	QTR_ID N	IONTH_IN	EAR_ID	PRODUCT	MSRP	PRODUCT	CUSTOME	PHONE	CITY	STATE	POSTALCO	COUNTRY	(TERRITO	R'CONTACTNAME		
2	10107	30	95.7	2	2871	2/24/2003 0:00	Shipped	1	2	2003	Motorcycl	9	95 S10_1678	Land of To	2.13E+09	9 NYC	NY	10022	USA	NA	Yu Kwai		
3	10121	34	81.35	5	2765.9	5/7/2003 0:00	Shipped	2	5	2003	Motorcycl	9	95 S10_1678	Reims Col	26.47.155	8 Reims		51100	France	EMEA	Henriot Paul		
4	10134	41	94.74	2	3884.34	7/1/2003 0:00	Shipped	3	7	2003	Motorcycl	9	95 S10_1678	Lyon Souv	+33 1 46 6	i: Paris		75508	France	EMEA	Da Cunha Daniel		
5	10145	45	83.26	6	3746.7	8/25/2003 0:00	Shipped	3	8	2003	Motorcycl	9	95 S10_1678	Toys4Grov	6.27E+09	Pasaden	a CA	90003	USA	NA	Young Julie		
6	10159	49	100	14	5205.27	10/10/2003 0:00	Shipped	4	10	2003	Motorcycl	9	95 S10_1678	Corporate	6.51E+09) San Fran	ci CA		USA	NA	Brown Julie		
7	10168	36	96.66	1	3479.76	10/28/2003 0:00	Shipped	4	10	2003	Motorcycl	9	95 S10_1678	Technics S	6.51E+09) Burlingar	m CA	94217	USA	NA	Hirano Juri		
8	10180	29	86.13	9	2497.77	11/11/2003 0:00	Shipped	4	11	2003	Motorcycl	9	95 S10_1678	Daedalus	20.16.155	8Lille		59000	France	EMEA	Rance Martine		
9	10188	48	100	1	5512.32	11/18/2003 0:00	Shipped	4	11	2003	Motorcycl	9	95 S10_1678	Herkku Gi	+47 2267	3 Bergen		N 5804	Norway	EMEA	Oeztan Veysel		
10	10201	22	98.57	2	2168.54	12/1/2003 0:00	Shipped	4	12	2003	Motorcycl	9	95 S10_1678	Mini Whe	6.51E+09) San Fran	ci CA		USA	NA	Murphy Julie		
11	10211	41	100	14	4708.44	1/15/2004 0:00	Shipped	1	1	2004	Motorcycl	9	95 S10_1678	Auto Cana	(1) 47.55.	6 Paris		75016	France	EMEA	Perrier Dominiq	ue	
12	10223	37	100	1	3965.66	2/20/2004 0:00	Shipped	1	2	2004	Motorcycl	9	95 S10_1678	Australian	03 9520 4	5 Melbour	n Victoria	3004	Australia	APAC	Ferguson Peter		
13	10237	23	100	7	2333.12	4/5/2004 0:00	Shipped	2	4	2004	Motorcycl	9	95 S10_1678	Vitachrom	2.13E+09	9 NYC	NY	10022	USA	NA	Frick Michael		
14	10251	28	100	2	3188.64	5/18/2004 0:00	Shipped	2	5	2004	Motorcycl	9	95 S10_1678	Tekni Coll	2.02E+09) Newark	NJ	94019	USA	NA	Brown William		
15	10263	34	100	2	3676.76	6/28/2004 0:00	Shipped	2	6	2004	Motorcycl	9	95 S10_1678	Gift Depot	2.04E+09) Bridgewa	at CT	97562	USA	NA	King Julie		
16	10275	45	92.83	1	4177.35	7/23/2004 0:00	Shipped	3	7	2004	Motorcycl	9	95 S10_1678	La Rochell	40.67.855	8 Nantes		44000	France	EMEA	Labrune Janine		
17	10285	36	100	6	4099.68	8/27/2004 0:00	Shipped	3	8	2004	Motorcycl	9	95 S10_1678	Marta's Re	6.18E+09) Cambrid	giMA	51247	USA	NA	Hernandez Mart	a	
18	10299	23	100	9	2597.39	9/30/2004 0:00	Shipped	3	9	2004	Motorcycl	9	95 S10_1678	Toys of Fir	90-224 85	8 Helsinki		21240	Finland	EMEA	Karttunen Matti		
19	10309	41	100	5	4394.38	10/15/2004 0:00	Shipped	4	10	2004	Motorcycl	9	95 S10_1678	Baane Mir	07-98 955	Stavern		4110	Norway	EMEA	Bergulfsen Jonas	5	
20	10318	46	94.74	1	4358.04	11/2/2004 0:00	Shipped	4	11	2004	Motorcycl	9	95 S10_1678	Diecast Cl	2.16E+09	Allentow	/r PA	70267	USA	NA	Yu Kyung		
21	10329	42	100	1	4396.14	11/15/2004 0:00	Shipped	4	11	2004	Motorcycl	9	95 S10_1678	Land of To	2.13E+09) NYC	NY	10022	USA	NA	Yu Kwai		
22	10341	41	100	9	7737.93	11/24/2004 0:00	Shipped	4	11	2004	Motorcycl	9	95 S10_1678	Salzburg C	6562-955	5 Salzburg		5020	Austria	EMEA	Pipps Georg		
23	10361	20	72.55	13	1451	12/17/2004 0:00	Shipped	4	12	2004	Motorcycl	9	95 S10_1678	Souvenier	+61 2 949	5 Chatswo	o NSW	2067	Australia	APAC	Huxley Adrian		
24	10375	21	34.91	12	733.11	2/3/2005 0:00	Shipped	1	2	2005	Motorcycl	9	95 S10_1678	La Rochell	40.67.855	8 Nantes		44000	France	EMEA	Labrune Janine		
25	10388	42	76.36	4	3207.12	3/3/2005 0:00	Shipped	1	3	2005	Motorcycl	9	95 S10_1678	FunGiftIde	5.09E+09	New Bed	If MA	50553	USA	NA	Benitez Violeta		
26	10403	24	100	7	2434.56	4/8/2005 0:00	Shipped	2	4	2005	Motorcycl	9	95 S10_1678	UK Collect	(171) 555	- Liverpoo	I	WX1 6LT	UK	EMEA	Devon Elizabeth		
27	10417	66	100	2	7516.08	5/13/2005 0:00	Disputed	2	5	2005	Motorcycl	9	95 S10_1678	Euro Shop	(91) 555 9	Madrid		28034	Spain	EMEA	Freyre Diego		
28	10103	26	100	11	5404.62	1/29/2003 0:00	Shipped	1	1	2003	Classic Ca	21	L4 S10_1949	Baane Mir	07-98 955	Stavern		4110	Norway	EMEA	Bergulfsen Jonas	5	
29	10112	29	100	1	7209.11	3/24/2003 0:00	Shipped	1	3	2003	Classic Ca	21	4 S10_1949	Volvo Mo	0921-123	SLule		S-958 22	Sweden	EMEA	Berglund Christi	na	
	}	sales_da	tasample			- ! !			-						4								

Figure 4.1: sales_datasample CSV data source (Example of System of Record)

4.3.2 Implementation of system of integration

The integration of the data warehouse was done with Microsoft SQL Server Management studios. A couple of SQL queries were used to create joins and relationships. In this module, the dimension and fact tables were created to match the aforementioned sales_datasample.csv records and fields a staging table to hold initials from the csv file was also created. The idea of a staging table is to hold data verbatim from the csv flat file. In the integration the data will be passed on from the csv file into the SQL server via SQL server integration service package.

4.2.3.1 Creation of tables in database:

A staging table was created with a series of complex queries, afterwards, the dimension tables were created with respect to the hierarchy of the schema. For instance, in a snowflake schema that has two or more dimension tables that are related to each other the dimension table that is farthest from the fact table will be created first. The hierarchy of table creation ensures consistency of the data warehouse. The creation and display of the table is outlined in FIGUREs 4.2 & 4.3

🚶 SOURCE DATA STAGING.sql - SWIFTER\SQLEXPR	Quick Launch (Ctrl+Q) 👂 🗕 🗗 🗙	
File Edit View Query Project Tools V	Vindow Help	
🖁 🗸 🖸 🚼 🕶 🎦 - 🎦 💾 🕌 💭 New Qu	ery 🛢 🖓 🖓 🕼 🖉 🙏 🗗 🗇 📍 🦿 - 🕅 🕞 📕 PERSONS_VS_GENDER 🛛 🗕 💭 🖊 🏛 🖸 📲	
🕂 💘 master 🔹 🕨 Exec	ute ■ ✔ 양 등 등 양 양 등 등 등 등 등 등 등 등 등 등 등 등 등 등	
Object Explorer	SUCKCE DATA STAGULEK(BE_manni (ST))* # X	
Connect 🕈 🏋 🖺 Y 🖸 🚸	2 60	4
⊟ SWIFTER\SQLEXPRESS (SQL Server 15.0.208) ∧		
🖯 📕 Databases	4 SET ANSI NULLS ON	
System Databases	5 60	
Database snapsnots Adventure/WorksDW2010	6	
	7 SET QUOTED_IDENTIFIER ON	
	8 60	
	9	
	10 SET ANSI_PADDING ON	
	11 60	
R PYDWH	12 🗉 CREATE TABLE [FYDWH].[Source_Data_Staging](
🕀 🛑 Database Diagrams	13 [Source_StagingID] [int] IDENTITY(1,1) NOT NULL,	
Tables	14 [ORDERNUMBER] [int] NULL,	
🗄 📕 Views	15 [QUANTITYORDERED] [int] NULL,	
🗉 📒 External Resources	16 [PRICEEACH] [float] NULL,	
🗉 📒 Synonyms	17 [SALES] [float] NULL,	
🗄 🛑 Programmability	18 [ORDERLINENUMBER] [int] NULL,	
🕀 📕 Service Broker	19 [ORDERDATE] [date] NULL,	
🕀 🛑 Storage	20 [STATUS] [varchar](50) NULL,	
🗄 📕 Security	21 [OTR ID] [int] WULL,	
🗄 🛢 FYPP	22 [MONTH ID] [int] NULL,	
🕀 🗑 HR	23 [YEAR ID] [int] NULL,	
🗉 🗑 IF_Statement	24 [PRODUCTLINE] [varchar](50) NULL,	
🗑 🗑 INEC	25 [MSRP] [int] NULL,	
🗄 📄 MTUTESTS	26 [PRODUCT CODE] [varchar](50) NULL,	
🗄 📄 NIIT	27 [CUSTOMERNAME] [varchar](50) NULL,	
🗄 📄 OYO STATE REVENUE	28 [PHONE] [varchar] (50) NULL,	
🗄 📄 PERSONALEXPENSES	29 [CITY] [varchar] (50) NULL,	
PERSONS_VS_GENDER	30 [STATE] [varchar] (50) NULL.	
🗄 🗑 PPPRA	31 [POSTALCODE] [INT] NULL.	
B SALESDWH	32 [COUNTRY] [varchar] (50) NULL.	
	33 [TERRITORY] [varchar] (50) NULL.	
	34 [CONTACTNAME] [Varchar] (50) NULL.	
	35 PRIMARY KEY CLUSTERED	
		Ŧ
< >>	SWIFIER/SQLEXPRESS (15.0 KIM) SWIFIEK\ea_manni (51) master 00:00:00 0 rows
🗇 Ready	Ln1 Cal1 Ch1 INS	
) 26°C Matthy clear A R T div 1204 AM

Figure 4.2: Creation of Staging Table

🧏 SQLQuery1.sql - SWIFTER\SQLEXPRESS.FYDWH	H (SWIFTER) ea_manni (58)) - Microsoft SQL Server Management Studio	Quick Launch (Ctrl+Q) 🛛 📍 🗕 🗗 🗙
File Edit View Query Project Tools V	Window Help	
🖁 🗣 🛛 🕆 🛅 🕈 🔛 💾 🕌 New Qu	buery 鼻 品 品 品 品 よ 日 合 [ヴ - ヴ - 「 罰 - ♬ PERSONS <u>VS GENDER - </u> 罰 / 音 回	
🕴 🕴 🕎 FYDWH 🔹 🕨 Exect	kute ■ ✔ \$2 🗊 🗐 🗑 🎟 🕼 🎟 🕼 🗏 🦉 🖅 🍬 -	
Object Explorer 🔹 🖣 🗙	< SQLQuery1.sql - STER\ea_manni (58)) * × SOURCE DATA STAGIER\ea_manni (51))*	•
Connecty # X# II V C A	1 /***** Script for SelectTopNRows command from SSMS ******/	÷
	– 2 ⊟SELECT TOP (1000) [Source_StagingID]	A
SWIFTER(SQLEXPRESS (SQL Server 15.0.208)	3 ,[ORDERNUMBER]	
🚽 📕 Databases	4 , [QUAITITYORDERED]	
🖞 🧧 Jystem Dalabases	5 ,[PRICEEACH]	
Database Shapshots	6 ,[SALES]	
	7 ,[ORDERLINENUMBER]	
CONCURRENT TRANSACTION	8 ,[ORDERDATE]	
	9 ,[STATUS]	
	10 ,[QTR_ID]	
H DWH	11 ,[MONTH_ID]	
R P FYDWH	12 , [YEAR_ID]	
🗄 📒 Database Diagrams	13 ,[PRODUCTLINE]	
🗉 🧧 Tables	14 , [MSRP]	
🗄 📕 System Tables	15 ,[PRODUCT_CODE]	
🗄 📕 FileTables	16 ,[CUSTOMERNAME]	
🗄 📕 External Tables	17 ,[PHOIE]	
🗄 📕 Graph Tables	18 , [CIIY]	٧
FYDWH.CONTACT	110% • (•
🗄 🏢 FYDWH.DATE	🖩 Results 📲 Kessages	
🗄 🏢 FYDWH.DIMSTATE	Urge Stating ORDERNUMBER QUANTITYORDERED PRICEEACH SALES ORDERLINENUMBER ORDERDATE STATUS OTR ID MONTH ID YEAR ID PRODUC	TLINE MSRP PRODUCT CODE CUSTOMERNAME
∃ FYDWH.FACTSALES		
FYDWH.PRODUCT		
🗄 🏢 FYDWH.Source_Data_Staging		
🕀 📕 Views		
External Resources		
E Synonyms		
🗄 📕 Programmability		
E Service Broker		
H Storage		
IF Statement		
H MTUTESTS		
		}
,	Q Query executed successfully. SWIFTER/SQLEXPRESS (15.0 RTM)	SWIFTER\ea_manni (58) FYDWH 00:00:00 0 rows
🗇 Ready	Ln1 Col1 Ch1 INS	
1 2 0 0 1 6		Mostly clear 🛆 🛱 🗔 (Ju) 12:07 AM
		110011 (1001 - 2 - 4 + 1) 8/2/2021 - 4

Figure 4.3: Viewing Staging Table
The records are absent from the staging table because the Source_Data_Staging table is yet to be populated.

A dimension table is created for date, product, address and contact details distinctively. The date table has the following columns: ORDERDATE, MONTH_ID, QTR_ID and YEAR_ID. The product table has the following columns: PRODUCTLINE, MSRP, PRODUCT_CODE. The state dimension table contains the following columns: CITY, STATE, POSTALCODE. The contact dimension contains CUSTOMERNAME, PHONE and CONTACTNAME columns. The creation of the tables were displayed in Figure 4.4, Figure 4.5, Figure 4.6, Figure 4.7.

A fact table is then created containing relationships with the other tables. The reason for the hierarchy of creation becomes clearer at this point because if you try to create a relationship with a table that does not exist yet a lot of error will follow the execution simply because of the current inexistence of those dimension tables. The fact table will have the primary keys of the dimension tables as the foreign key of the fact table. Therefore, a clean and clear relationship.

Image: SWIFTER SQLEXPRESS.FYDWH (SWIFT File Edit View Query Project Tools W Image: Original Stress Image: Original Stress Image: Origin	Riea_manni (69)) - Microsoft SQL Server Management Studio ndow Help y 물 없 없 없 상 단 습 ?? - ? - 정 · , 를 PERSONS_VS_GENDER · , 회 분 홈 전 · , e = J 양 명 미 명 양 입 명 명 기 등 역 2 후 한 10 -	Quick Launch (Ctrl+Q)	<i>۹</i> _ ۵ ×
Object Explorer • ↓ × Connect • ♥ *♥ ■ ● ● ▲ Databases COMBINED SELECTED_FOODS CONCURRENT_TRANSACTION DAILY_EXPENSES DOCTORSUSPATIENTS DWH DATABASE DOCTORSUSPATIENTS DWH FORWH Concurrent_TRANSACTION POWH Tables Views External Resources Synonyms Forger Storage Security Forger HR If Statement NIT OYO STATE REVENUE PRESONS_VS_GENDER PPRA SALESDWH SQLVIEWSs TEMONE 	DATE-ogl-SWIFEL-TERteg-manni(50) * X [MISTATE-sql-SWL_TERteg-manni(50)] PRODUCT.sql-SWL_TERteg-manni(50)] CONTACT.sql-SWL_TERteg-manni(50) DATE-ogl-SWIFEL-TERteg-manni(50)] * X [MISTATE-sql-SWL_TERteg-manni(50)] CONTACT.sql-SWL_TERteg-manni(50)] DATE-ogl-SWIFEL-TERteg-manni(50)] * X [MISTATE-sql-SWL_TERteg-manni(50)] CONTACT.sql-SWL_TERteg-manni(50)] A [/****** Object: Table [Crime].[DimYear] Script Date: 2/3/2015 11:03:48 AM ******/ S SET ANSI_MULLS ON 6 G0 7 8 SET QUOTED_IDENTIFIER ON 9 G0 10 SET ANSI_PADDING ON 12 G0 13 EREARE TABLE [PYOMH].[DATE](15 [ORDERDATE] [DATE](DIT NULL, 16 [OTAL [DIT] NULL, 17 [PONTH_ID] [Int] NULL, 18 [VEAR_ID] [Int] NULL, 19 PRIMARY KEY CLUSTERED 20 (21 [ORDERDATE] ASC 22 [NITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROM_LOCKS = ON, ALLOW_PAGE_LOCKS 23] ON [PRIMARY] 24 [25 G0 26] 27 SET ANSI_PADDING OFF 28 G0	I)) = ON) ON [PRIMARY]	• » • • • •
TicketDB	110% - (© Connected (1/1) SWIETER/SYN EVDDESS (15.0 PT.A)	SWIFTER\ea manni (60) EVDW	H 00:00:00 0 rows
	20 connector (11.1)	over ruv(ca_manni (os) _ rYDW	000000 01005
🗋 Ready	Ln1 Col1 Ch1 INS		12-12-414
🚺 🗖 9 0 9 🖿	🌻 🛓 🛄 🤽 刘 🗓 🔍	C Mostly clear \land 🖗 🔟 🛛	\$0) 8/2/2021 ↓

Figure 4.4 Creation of Date Dimension Table.

JIMSTATE.sql - SWIFTER\SQLEXPRESS.FYDWH (S	SWIFTER(ea_manni (63)) - Microsoft SQL Server Management Studio	Quick Launch (Ctrl+Q) 🛛 🖉 🗕 🗗 🗙
- File Edit View Query Project Tools W	Vindow Help	
🖁 🗛 - 🗇 🔯 - 🤄 - 🏜 🖉 New One	🗤 島 魚 魚 魚 其 山 白 り - C - 図 🧃 PERSONS VS GENDER	
	un M 90 P P 90 M, I P 10 P 10 P 23 - 23 - 16 ÷	
Object Explorer 🔹 🖣 🗙	DATE.sql - SWIFTERTER\ea_manni (69)) DIMSTATE.sql - SWITER\ea_manni (63)) 🗢 🗙 PRODUCT.sql - SWITER\ea_manni (56)) CONTACT.sq	I - SWITER\ea_manni (51))
Connect 🛛 🏺 🎽 🔳 🍸 🗳 🚸	1 USE [FYDWH]	÷
🗉 🔒 SWIFTER\SQLEXPRESS (SQL Server 15.0.208) 🗚	2 60	<u>^</u>
🗄 🛑 Databases		
🗄 📕 System Databases		
🗄 📕 Database Snapshots	5 00	
🗄 🗎 AdventureWorksDW2019	7 SET MINTEN THENTTETER ON	
■ COMBINED_SELECTED_FOODS	8 G0	
DAILY_EXPENSES	10 SET ANST PADDING ON	
	11 60	
	12	
Datahase Diagrams	13 ECREATE TABLE [FYDWH].[DIMSTATE](
P [®] dbo.EVDWH	14 [CITY] [varchar] (50) NULL,	
C: dbo.FYDWH.DG	15 [STATE] [varchar] (50) NULL,	
🗉 🛄 Tables	16 [POSTALCODE] [INT] NOT NULL,	
🗑 🗐 Views	17 PRIMARY KEY CLUSTERED	
🕢 📕 External Resources	18 (
🗉 📒 Synonyms	19 [POSTALCODE] ASC	
🗉 📕 Programmability	20)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, A	ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
🕀 📕 Service Broker	21) ON [PRIMARY]	
🕀 📕 Storage	22 [
🕀 📕 Security	23 GO	
₩ ¥YPP	24	
HK Chatananat	25 SET ANSI_PADDING OFF	
	26 GO	
	27	
OYO STATE REVENUE		
PERSONALEXPENSES		
E PERSONS_VS_GENDER		
🗄 🛢 PPPRA		
🗄 🗑 SALESDWH		
🗄 🗑 SQLVIEWSs		
🗄 🗑 TERMONE		Ψ.
🗄 📄 TicketDB 🗸 🗸	110 % •	· · · · · · · · · · · · · · · · · · ·
< >	Connected. (1/1) SWIFTER(SQLEXPRESS (15.0 RTM) SWIFTER\ea_manni (63) FYDWH 00:00:00 0 rows
🗇 Ready	Ln1 Col1 Ch1	INS
🖪 P O 🔃 📑 💽	0 4 1 1 1 1	26°C Mostly clear へ 6 9回 40 8220001 口

Figure 4.5: Creation of State Dimension Table

RODUCT.sql - SWIFTER\SQLEXPRESS.FYDWH (S	WIFTER(ea_manni (56)) - Microsoft SQL Server Management Studio	Quick Launch (Ctrl+Q) 🛛 👂 🗖 🗙
- File Edit View Query Project Tools W	índow Help	
🖁 🗛 🔊 🖹 🛪 📩 🖕 💾 📓 New On		
	····································	
Object Explorer 🔹 🖣 🗙	DATE.sql - SWIFTERTER\ea_manni (69)) DIMSTATE.sql - SWITER\ea_manni (63)) PRODUCT.sql - SWITER\ea_manni (56) 😕 🗙 CONTACT.sql - SWITER\ea_manni (56)	nanni (51)) 🛛 🔍 🔻
Connect 🛛 🏺 🎽 📕 🍸 🗳 🚸	1 USE [FYDMH]	+
🗉 🗟 SWIFTER\SQLEXPRESS (SQL Server 15.0.2081 🗚	2 60	A
🗉 🛑 Databases	5 4 (###### Object, Table [Caime] [DimVern] - Script Date: 3/3/2005 44:03.40 AM #######/	
🗄 📕 System Databases	4 /***** Object: Table [crime].[bimmean] Script Date: 2/5/2015 11:05:48 AM	
🗄 🛑 Database Snapshots		
🕀 🛢 AdventureWorksDW2019	7	
COMBINED_SELECTED_FOODS		
DAILY_EXPENSES	9 du	
B DWH		
e 🗑 FYDWH	12 00	
Database Diagrams	כב א היאספט ובאטוב (באוויה)/	
	15 [roudelline] [varenar](50) Hole,	
	17 [DRODUCT CODE] [varchan]/50/ NOT NULL	
🗑 📕 Vitews		
🛛 🧰 Programmability		
E Service Broker	20 [FNODOLI_CODE] FOC 21 VETH (DAD THDEY - DEE STATTSTTCS HOPECONDITE - DEE TOMORE NID VEV - DEE ALLOW DOW LOCVS - ON ALLOW DAGE L	OCKS - ON) ON [DRIMARY]
₩ 🗍 Storage	21)MITH (RM_INDEX - 01, STRIISICS_HONECONFOLE - 01, INNOL_DUF_KET - 01, RECON_NUM_EDOKS - 00, RECON_FACE_E	OCKS - ON) ON [PRIMART]
🗉 🗐 Security	22) 011 [11111411]	
H 🔒 FYPP	24 60	
HR 🗎 HR	25	
🗄 🗑 IF_Statement		
inec	27 60	
🗄 🗎 MTUTESTS	28	
🗄 🛢 NIT	20	
🗄 📄 OYO STATE REVENUE		
🗄 🗑 PERSONALEXPENSES		
E PPPRA		
E SQLVIEWSS		
	10.0	*
	III 76 T	
()	N Connecteur (1/1)	J KIMJ SWIFIEK(ea_manni (SO) FYDWH UCUUUU O rows
🗇 Ready	Ln1 Col1 Ch1 NS	
E 2 0 0 🖪 👩		26°C Mostly clear ^ 현 현 여 12:43 AM
		8/2/2021

Figure 4.6: Creation of Product Dimension Table

🚶 CONTACT.sql - SWIFTER\SQLEXPRESS.FYDWH (S	WIFTER\ea_manni (51)) - Microsoft SQL Server Management Studio	Quick Launch (Ctrl+Q)	Р _ 8 X
File Edit View Query Project Tools W	indow Help		
🖁 🙃 🔹 📅 📲 🏜 🖉 New Oue	🗤 島 島 島 魚 魚 其 司 台 ヴ - で - 罰 - 🕫 PERSONS VS GENDER 司 🌶 竜 四		
Object Explorer 🔹 🖣 🗙	DATE.sql - SWIFTERTER(ea_manni (69)) DIMSTATE.sql - SWITER(ea_manni (63)) PRODUCT.sql - SWITER(ea_manni (56)) CONTACT.sql - SWITER(ea_manni	ni (51)) 🕫 🗙	≪ ▼
Connect 🔻 🌹 🌹 👅 🍸 🗳 🚸	1 USE [FYDWH]		÷
🗉 🐻 SWIFTER\SQLEXPRESS (SQL Server 15.0.208) 🔺	2 60		
🗉 🛑 Databases			
🗄 📕 System Databases			
🗄 🛑 Database Snapshots	5 00		
🗄 🗑 AdventureWorksDW2019	7 SET GIOTED TDENTTETER ON		
COMBINED_SELECTED_FOODS	8 60		
CONCURRENT TRANSACTION	9		
	10 SET ANSI PADDING ON		
	- 11 G0		
	12		
🕞 🛑 Database Diagrams	13 ⊡CREATE TABLE [FYDWH].[CONTACT](
면물 dbo.FYDWH	14 [CUSTOMERNAME] [varchar](50) NULL,		
ta dbo.FYDWH.DG	15 [PHONE] [varchar] (50) NULL,		
🗄 📕 Tables	16 [CONTACTNAME] [varchar] (50) NOT NULL,		
🗄 📕 Views	17 PRIMARY KEY CLUSTERED		
🛞 🛑 External Resources	18 (
🗄 📕 Synonyms	19 [CONTACTNAME] ASC		
🕀 🛑 Programmability	20)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOC	<pre>KS = ON) ON [PRIMARY]</pre>	
🕀 📕 Service Broker	21) ON [PRIMARY]		
🗄 📕 Storage	22		
	23 60		
	24		
F F Statement	25 SET ANSI_PADDING OFF		
INEC	20 60		
H 🗑 MTUTESTS	27		
🗉 🛢 NIIT			
🗄 📄 OYO STATE REVENUE			
🗄 🛢 PERSONALEXPENSES			
PERSONS_VS_GENDER			
E PPPRA			
SALESUWH SOLVIEWS-			
TicketDB	10% •		
	₩, Connected. (1/1) SWIFTER/SOLEX/PRESS (15.0 RT	M) SWIFTER\ea manni (51) FYDWF	H 00:00:00 0 rows
D Peadu			
			12-//3 AM
🗐 🗾 🧿 O 🤇 📕	🔍 🛓 🛄 🛃 🔍 🔍	/6°C Mostly clear \land 🖗 🐿 ଏ	N) 8/2/2021

Figure 4.7: Creation of Contact Dimension Table

🚶 SQLQuery2.sql - SWIFTER\SQLEXPRESS.FYDWH	SWIFTER\ea_manni (53))* - Microsoft SQL Server Management Studio		Quick Launch (Ctrl+Q)	۹ - ۵ x
File Edit View Query Project Tools W	lindow Help			
😋 = 🗧 📸 = 📩 = 🏠 💾 🚰 💭 New Que	ery 🗿 🖗 🖗 🕼 🖁 🐰 🗇 奇 💙 • 🤆 • 🕅 🔹 🥬 PERSONS_VS_GENDER	• 😡 🌶 🏛 🗵 • 🖕		
FYDWH • Execu	te ■ ✔ 83 目日 87 83 副周囲 日 13 14 14 14 14 14			
Object Explorer 💌 🖡 X	DATE.sql - SWIFTERTER\ea_manni (55)) DIMSTATE.sql - SWITER\ea_manni (64))	PRODUCT.sql - SWITER\ea_manni (65))	SQLQuery2.sql - STER\ea_manni (53))* 🚁 🗙	÷
Connect• ₩ ₩ = ▼ 🖒 🚸	1 ☐SELECT * FROM [FYDWH].[CONTACT]			÷
SWIFTERISOLEXPRESS (SOL Server 15.0.20R)	2 SELECT * FROM [FYDWH].[DATE]			A
Databases	3 SELECT * FROM [FYDWH].[DIMSTATE]			
🗄 📕 System Databases	4 SELECT * FROM [FYDWH].[PRODUCT]			
🗄 📕 Database Snapshots	· · · · · · · · · · · · · · · · · · ·			
🗄 📄 AdventureWorksDW2019				
COMBINED_SELECTED_FOODS				
	110 % •			· · · · ·
B DWH	III Danite pil Haussa			,
🗄 🗎 FYDWH				
🗄 🛑 Database Diagrams				
🖃 📕 Tables				
System lables				
🗄 📕 Fitenal Tables				
🗄 🧧 Ercentor Hores	ORDERDATE OTR ID MONTH ID YEAR ID			
FYDWH.CONTACT				
🗄 🧱 FYDWH.DATE				
FYDWH.DIMSTATE				
⊞ FYDWH.PRODUCT				
H H Views				
External Resources	CITI STATE POSTALOUDE			
🗄 📕 Synonyms				
🗄 🛑 Programmability				
🕀 🛑 Service Broker				
🕀 📕 Storage	PRODUCTUNE MSRP PRODUCT CODE			
E Security				
H A HR				
H 🗎 INEC				
🗄 🗑 MTUTESTS				
UYO SIATE REVENUE				
(SWIFTER(SQLEAPRESS (ISU KIM) SWIFTER(ea_manni (SS) FYDWH	UUUUUUU Ü rows
🗋 Ready		Ln 5 Col 1 Ch 1	NS.	40.11.014
📲 P O 🔍 📕 💽	9 🛓 🔍 📲 🥾 刘		💛 26°C Mostly clear \land 🖗 🖶 🕸) _{8/2/2021}

Figure 4.8 Display of Dimension Tables

🚶 FACTSALES.sql - SWIFTER\SQLEXPRESS.FYDWH	SWIFTER(ea_manni (52)) - Microsoft SQL Server Management Studio	Quick Launch (Ctrl+Q) 🛛 📍 🗖 🗙
File Edit View Query Project Tools W	indow Help	
🖁 🕞 🔹 🕄 🔹 📩 🖕 📓 New Out	av 周 요 요 요 🗶 司 슈 ヴ - で - 🕅 - 🧋 PERSONG VS GENDER - 🖬 🌶 🏛 🗆 -	
Object Explorer 🔹 🖣 🗙	FACTSALES.sql - STER.ea_manni (52)) 🐐 🗙 CONTACT.sql - SWL_TER.ea_manni (51))	SOURCE DATA STAGIER\ea_manni (53)) 🗯 🗙 👻
Connect 🛛 🛱 🌹 🍸 🕻 🚸	8 GU	†
🗉 🐻 SWIFTER\SQLEXPRESS (SQL Server 15.0.2081 🔺	9 CCIOSESISWITERIDOCUMENTS (POJECTI Y SCRIPTS (PACTOALESSO) - SWITERISOLEA/RESS/TOWH (SWITERIE& manni (32)) 10 SET ANST DANNTIK AN	
🖃 🛑 Databases	10 SCT MIST_MODINE ON	
🕀 🛑 System Databases		
🗄 📒 Database Snapshots		
🗄 🗎 AdventureWorksDW2019	10 [OULNUTTYVORDERED] [int] NULL	
■ COMBINED_SELECTED_FOODS	15 [DRTCERCH] [flost] WH	
■ CONCURRENT_TRANSACTION	15 [FALCERON] [TIVAL] NULL	
	17 [ORDEDIATE] [data] WWW	
DOCTORSVSPATIENTS	17 [UNDERDATE] [USEE] NUEL,	
⊞ ₩ DWH		
e 🗑 FYDWH	19 [PUSIALCODE] [INI] NULL,	
Database Diagrams		
TE dbo.FYDWH		
	22 OV	
🗧 📕 Tables	23 ERLIEK TABLE [FTUMH].[FACISHES] WITH CHELK AUD CUNSTRAINT [FK_DATEID] FUREIGN RET([UNDERDATE])	
H System lables	24 [nereneucos [ritum].[umie]([umuenumie])	
🗄 📕 File lables	25 UU	
🗄 📕 Cranh Tables	20 ALTER TABLE [FYUMH].[FACTSALES] CHECK CUNSTRAINT [FK_DATEID]	
	2/ 60	
	28 20 DULTER THREE [EURIN] [EASTALIES] UTTUALING AND ADVETRITUT [EV. ADVETATED] CONTACTOR VEV/[CONTACTORNE]]	
R EVDWH.DIMSTATE	29 EALTER TABLE [FYUMH].[FACTSALES] WITH CHELR ADD CONSTRAINT [FK_CONTACTID] FOREIGN REY([CONTACTNAME])	
FYDWH.FACTSALES	30 [KEEKENCES [FTOWN].[CUNTACTINAME])	
FYDWH.PRODUCT		
FYDWH.Source Data Staging	32 ALTEK TABLE [FYDWH].[FACTSALES] CHECK CUNSTRALNT [FK_CUNTACTID]	
Tiews	33 GO	
External Resources	34	
General Synonyms	35 HALTEK TABLE [FYDWH].[FACTSALES] WITH CHECK ADD CONSTRAINT [FK_PRODUCTID] FOREIGH REY([PRODUCT_CODE])	
🗄 📒 Programmability	36 [REFERENCES [FYDWH].[PRODUCT]([PRODUCT_CODE])	
🗄 📒 Service Broker		
🕀 🛑 Storage	38 ALTEN TABLE [FYDWH].[FACTSALES] CHECK CONSTRAINT [FK_PRODUCTID]	
🕀 🛑 Security	39 GO	
🗄 🗎 FYPP	40	
🗄 🛢 HR	41 ⊟ALTER TABLE [FYDWH].[FACTSALES] WITH CHECK ADD CONSTRAINT [FK_DIMSTATEID] FOREIGN KEY([PUSTALCODE])	
🗄 🛢 IF_Statement	42 [REFERENCES [FYDWH].[DIMSTATE]([POSTALCODE])	Ψ.
H 🗑 INEC 🗸 🗸	110 % • ()
< > >	₩ Connected. (1/1) SWIFTER/SQLEXPRESS (15.	0 RTM) SWIFTER\ea_manni (52) FYDWH 00:00:00 0 rows
🗖 Ready	- In 38 Coló Chó INS	
		1+55 AM
E 2 0 Q 📕 💽	🎯 🛓 🕮 🔣 💐 🔍 🐨 🤌 2	6°C Mostly clear ^ ⓒ

Figure 4.9: Creation of Fact Table and Relationship.



Figure 4.10: Diagram of Star Schema Implemented From Sales Dataset

4.2.3.2 Data population from SSIS:

Because each source table or file should have its own staging table in a warehouse, the staging table is loaded from the csv data source. The raw data is supplied by operational databases and external sources, as depicted in the diagram. Special logic monitors changes in data sources and generates periodic extracts of new data. Staging tables record the history of data entered from periodic extracts. This is accomplished through the use of a data flow job, which consists of data transitioning from a flat file source through a data conversion module, and finally into the SQL table destination. Following that, the Dimension tables are filled.



Figure 4.11: Data Flow Task to Populate Staging Table



Figure 4.12 Module-to-module Sequence in Data Flow



Figure 4.13: Flat File Configuration and Column Structure

C File Edit View Git Project	t Build Debug Format Ie: ™ + Developi + Default	st Analyze Iools E	xtensions Window H 휜 _국	elp Search (Ctrl+Q)		p Da	ta Warehous	e			S) –	۵ ×
§ SSIS Toolbox ▼ 4 X	Staging.dtsx [Design]* ⊕ X load	Dim and Fact Tables.dtsx [L	Design]						v 12	Solution Explorer		• 4 Х
Search SSIS Toolbox	🖁 Control Flow 🖞 Data Flow	Parameters 37 Event	Handlers - Package Ext	plorer				- V	● 弾	C O 🎧 🖉 'o - 🕃	đ 🖗 🥕 -	
A Favorites	Data Row Task: 👔 Populate D	40 Data Conversion Iranst	formation Editor				-	ЦX	v	Search Solution Explorer (Ct	rl+;)	ρ-
C Destination Assistant		Configure the properties	used to convert the data typ	pe of an input column to a diff	erent data ty	oe. Depending	g on the data	type to which		Solution 'Data Warehouse	use' (1 of 1 project)	
Common		the column is converted,	set the length, precision, so	cale, and code page of the cold	mn.					Project.params		
(i ^{)C} Aggregate										Connection Ma	nagers	
Balanced Data Distribut			1	Available Input Columns						SSIS Packages	d Fart Tables dtsy	
မှု 🖓 Conditional Split				✓ Name	•					E Staging.dtsv	u l'act nuoressoress	
ତି ୍ତ୍ରି Data Conversion										🖌 🕤 Package Parts		
🖣 🗖 Data Streaming Destina			Ì	QUANTITYORDERED						🕤 Control Flow	i i i i i i i i i i i i i i i i i i i	
j, Derived Column				PRICEEACH						Miscellaneous		
HDFS File Destination										Linked Azure Ke	sources	
HDFS File Source										Azure-Stora	ntegration nuntime	
‡ Lookup				V SHLLS								
f⊈ Merge												
Werge Join												
		Input Column	Output Alias	Data Type	Length	Precision	Scale	Code Pag ^				
CDRC Destination		ORDERNUMBER	TRANSFORMED ORDE	four-byte signed integer								
		QUANTITYORDERED	TRANSFORMED QUA	four-byte signed integer								
OLE DB Command		PRICEEACH	TRANSFORMED PRICE	double-precision float [D.,,						Properties		• ¶ X
Row Count		ORDERLINENUMBER	TRANSFORMED ORDE	four-byte signed integer						Data Conversion Data Flow	Component	Ŧ
Script Component		SALES	TRANSFORMED SALES	double-precision float [D.,,					100%			
🖾 Slowly Changing Dime		ORDERDATE	TRANSFORMED ORDE	database date IDT DBDA						ContactInfo	Data Conversion;	Microsoft Cc 🔺
↓† Sort		STATUS	TRANSFORMED STATUS	string (DT_STR)	50			1252 (Al		Description	Data Conversion	
↓ Union All			TRANSFORMED OTR ID	four-bute signed integer				iese y a		D	2	
♦ Azure				Four but a signed integer				v		IdentificationString	Data Conversion	
▲ Other Transforms ▼		MUNIH_D K	IKANSPUKMED MUN	four-byte signed integer				>	-	IsDefaultLocale	True	
v Information									Н	Name	English (United St Data Conversion	iates)
Drag a toolbox item to 5515	Connection Managers	Configure Erro	or Output		OK	Car	ncel	Help		Show Editor, Show Advance	d Editor	
	CSV Connection Manager	lat File Connection Manager	SWIFTER/SQLEXPRES	SS.FYDWH	-	-	-			Name Specifies the name of the co	omponent.	
Criteria												
Output												
🗋 Ready										Ŷ	Add to Source Con	trol 🔺 👍
🖷 P O 🔃 🗍	0 0 🛓 刘	0 🖬 🧕							2	26°C Mostly clear \land 9	€ € 12:0 8/2	02 PM 2/2021 ↓

Figure 4.14: Data Conversion for Column Datatypes



Figure 4.15: Column Mapping From Data Source to Destination.

SSIS Toolbox • 4 >	Staging ofter (Design) 🗢 🗙 load Dim and Fact Tables.ofter (Design)	v ¢ Solutio
Favorites Destination Assistant	Data Row Task: Ropulate Data from CSV to Staging	> Explorer
Common (p ^E Aggregate Conditional Split Gonditional Gonditional Split Gonditional Gonditional Split Gonditional Gonditional Split Gon	Less rows Less rows Less rows Less rows Soll Sever Destination	Sit Changes
Acure Information Drag a toolbox item to SSIS Designer to use it.	Connection Managers © Connection Manager Pat File Connection Manager Pat File Connection Manager SWIFTER/SQLEXPRESS.FYDWH	

Figure 4.16: Successful Execution of Data Flow Task From CSV to SQL Staging Table.

🚶 SQLQuery1.sql - SWIFTER\SQLEXPRESS.FYDWH	(SWIFTE	:R\ea_manni (68))* - I	Microsoft SQL Serve	er Management Studio									Quick Lau	nch (Ctrl+	, Q) , P	- 8 X
File Edit View Query Project Tools V	Nindow	Help														
🖸 🕫 🕈 🕈 🕇 📩 🖕 🍟 🖉 New Qu	iery 🔒	HER DAR MAN	¥ 🗗 â 기 •	(° + 🕅 - ≠ 🎜	PERSONS_VS	GENDER	- 😡 🌶 🕯	• D • -								
🕴 🕴 FYDWH 🔹 🕨 Exect	ute 🔳	√ 80 🖬 🖥			•											
Object Explorer 🔹 🖡 🗙	SQLC	Query1.sql - STER\ea	a_manni (68))* ⊣¤	X												•
Connect 📲 🍟 🗏 🍸 🖒 🚸		1														÷
□ 🗑 SWIFTER\SQLEXPRESS (SQL Server 15.0.208) ∧		2 SELECT *	FROM].[Source_Data_S	taging]											A
Databases																
🗄 📕 System Databases	110.0	4 _ 4														Y
🗉 📕 Database Snapshots	110 %															/
🗄 🗑 AdventureWorksDW2019		Results 📲 Message	#8													
COMBINED_SELECTED_FOODS		Source_StagingID	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	SALES	ORDERLINENUMBER	ORDERDATE	STATUS	QTR_ID	MONTH_ID	YEAR_ID	PRODUCTLINE	MSRP	PRODUCT_CODE	CUSTOMEF A
	1	1	10107	30	95.7	2871	2	2003-02-24	Shipped	1	2	2003	Motorcycles	95	S10_1678	Land of Toj
	2	2	10121	34	81.35	2765.9	5	2003-05-07	Shipped	2	5	2003	Motorcycles	95	S10_1678	Reims Colle
	3	3	10134	41	94.74	3884.34	2	2003-07-01	Shipped	3	7	2003	Motorcycles	95	S10_1678	Lyon Souve
⊞ 🗑 DWH	4	4	10145	45	83.26	3746.7	6	2003-08-25	Shipped	3	8	2003	Motorcycles	95	S10_1678	Toys4Grow
B FYDWH	5	5	10159	49	100	5205.27	14	2003-10-10	Shipped	4	10	2003	Motorcycles	95	S10_1678	Corporate C
Database Diagrams	6	6	10168	36	96.66	3479.76	1	2003-10-28	Shipped	4	10	2003	Motorcycles	95	S10_1678	Technics S
🗧 📕 Tables	7	7	10180	29	86.13	2497.77	9	2003-11-11	Shipped	4	11	2003	Motorcycles	95	S10_1678	Daedalus C
🗄 📕 System Tables	8	8	10188	48	100	5512.32	1	2003-11-18	Shipped	4	11	2003	Motorcycles	95	S10_1678	Herkku Gift
🗄 📕 Filetables	9	9	10201	22	98.57	2168.54	2	2003-12-01	Shipped	4	12	2003	Motorcycles	95	S10_1678	Mini Wheel:
External radies	10	10	10211	41	100	4708.44	14	2004-01-15	Shipped	1	1	2004	Motorcycles	95	S10_1678	Auto Canal
	11	11	10223	37	100	3965.66	1	2004-02-20	Shipped	1	2	2004	Motorcycles	95	S10_1678	Australian C
	12	12	10237	23	100	2333.12	7	2004-04-05	Shipped	2	4	2004	Motorcycles	95	S10_1678	Vtachrome
I III FYDWH.DIMSTATE	13	13	10251	28	100	3188.64	2	2004-05-18	Shipped	2	5	2004	Motorcycles	95	S10_1678	Tekni Cole
	14	14	10263	34	100	3676.76	2	2004-06-28	Shipped	2	6	2004	Motorcycles	95	S10_1678	Gift Depot I
FYDWH.PRODUCT	15	15	10275	45	92.83	4177.35	1	2004-07-23	Shipped	3	7	2004	Motorcycles	95	S10_1678	La Rochelle
🗑 🎹 FYDWH.Source Data Staging	16	16	10285	36	100	4099.68	6	2004-08-27	Shipped	3	8	2004	Motorcycles	95	S10_1678	Marta's Rep
H I Views H	17	17	10299	23	100	2597.39	9	2004-09-30	Shipped	3	9	2004	Motorcycles	95	S10_1678	Toys of Finl
	18	18	10309	41	100	4394.38	5	2004-10-15	Shipped	4	10	2004	Motorcycles	95	S10_1678	Baane Mini
🗄 📕 Synonyms	19	19	10318	46	94.74	4358.04	1	2004-11-02	Shipped	4	11	2004	Motorcycles	95	S10 1678	Diecast Cla
🗄 📒 Programmability	20	20	10329	42	100	4396.14	1	2004-11-15	Shipped	4	11	2004	Motorcycles	95	S10 1678	Land of Toy
🗄 📕 Service Broker	21	21	10341	41	100	7737.93	9	2004-11-24	Shipped	4	11	2004	Motorcycles	95	S10 1678	Salzburg Cc
🗄 📕 Storage	22	22	10361	20	72.55	1451	13	2004-12-17	Shipped	4	12	2004	Motorcycles	95	\$10 1678	Souveniers
🗄 📕 Security	23	23	10375	21	34.91	733.11	12	2005-02-03	Shipped	1	2	2005	Motorcycles	95	\$10 1678	La Rochelle
🗄 🗑 FYPP	24	24	10388	42	76.36	3207.12	4	2005-03-03	Shipped	1	3	2005	Motorcycles	95	\$10 1678	FunGiftIdea
🗄 🗑 HR	25	25	10403	24	100	2434.56	7	2005-04-08	Shinped	2	4	2005	Motorcycles	95	S10 1678	UK Collecta
🗄 🗑 IF_Statement	26	26	10417	66	100	7516.08	2	2005-05-13	Disputed	2	5	2005	Motorcycles	95	S10_1678	Furo Shoop
🗄 📄 INEC	27	27	10103	26	100	5404.62	- 11	2003-01-29	Shinped	-	1	2003	Classic Cars	214	S10 1949	Raane Mini
	1	L,	leree	20	IVV	V1V		2000 0. 20	on ubbaa			2000	Grand Gard	B 15	010_1010	busine tonin V
NIT V																/
< >>	00	luery executed succes	ssfully.							SWIFTE	<pre>k\sqlexpress</pre>	s (15.0 RTM)	SWIFTER\ea_ma	anni (68)	FYDWH 00:00:01	2,823 rows
🗇 Ready							Ln 2	Col 23	Ch 23		INS					
# P O 인 🖡 🞯	0	🛓 関	.	×								26%	C Mostly clear	^ ĝ	(点 100 中) 8/2	20 PM 2/2021 🗘

Figure 4.17: Selection of Populated Staging Table



Figure 4.18: Various Data Flow Tasks for Populating Dimension Tables.

🕅 File Edit View Git Project Build Del	ıg Format Test Analyze Tools Extensior	nns Window Help Search (Ctrl+Q)	Data Warehouse	9 -	0)	X
😋 • 💿 👌 • 🍅 🔛 🖓 • 🖓 • 🖓 •	• Default • > Continue • 🖹 🙆	. ■ 0 → : ? : <i>#</i> .		Ŕ	ADMIN	
Process: [11728] DtsDebugHost.exe 🔹 💽 Lifec	cle Events 🔹 Thread:	- 🔻 🔻 XX Stack Frame:	v .			
SSIS Toolhoy	X Staging.dtsr.[Design] Inad Dim and Fa	art Tables dity [Design]* Jr V			▼ ∅	So
Search SSIS Toolbox	Contro		- 🗆 X		ت	lutio
 ✓ Favorites □ Destination Assistant □ Source Assistant ✓ Common d^{SE} Apgregate 	Data Roy Configure the properties used b Connection Manager mode	by a data flow to obtain data from any OLE DB provider. :fly an OLE DB connection manager, a data source, or a data source view le. If using the SQL command access mode, specify the SQL command a	, and select the data access ither by typing the query or by		v	Explorer Git Chang
Imaging and the set of	Columns using Error Output Data SQL S P N P F F	g Query Builder. DB connection manager: FTER/SQLEX/RESS.FYDWH access mode: command QL command text: SELECT DISTINCT PRODUCTLINE, SPRODUCTLINE, MSRP, AS MSRP, PRODUCT_CODE AS PRODUCT_CODE FROM [FIDWH][Source_Data_Staging]	New New New Parameters Build Query Browse Parse Query		100%	ges
Azure Information Drag a toolbox item to SSIS Designer to use it.	0	Preview	Cancel Help		[1]	
Exception Settings Command Window Immediate Windo	Connel SwiftERISQLEXPRESS FYDWH Output Error List					
🗇 Ready				↑ Add to Source C	ontrol 🔺 🧍	1
📲 P O 💽 🔳 🞯 🧯	🛓 🖉 📲 🤱 🕅			🔲 27℃ Light rain へ Ĝ 候 幅 40 💡	2:54 PM	

Figure 4.19: SQL Command Accessing for Product Dimension Table.

K File Edit View Git Proje	ect Build Debug Format T (* + Developi + Default	fest Analyze Tools E 🔹 🕨 Start + 🔃 🙆	xtensions Window He] _↓	elp Search (Ctrl+)	Q P	Data Warehouse			§ –	d X admin
X II v	Staging.dtsx [Design]	Dim and Fact Tables dtsy (D	esion1* -b X					Solution Evolater		• I X
Castrin SSIS Toolhow	Control Row 🔒 Data Row	A Paramatam II Evant	Handlern 🛄 Dackade Eve	larar 🐧 Evan tian	Posuito					T 1
		OIF DB Destination Edi	natuleis 🖛 rackaue cw tor		nesuls	- П	X		5 ° ° 7 –	0
G A Favorites	Data Row Task: 👔 LOAD PRO					-	v	Search Solution Explorer (Ctrl+;)	ρ.
Le Destination Assistant		Configure the properties	used to insert data into a re	elational database usi	ng an OLE DB provider.			Solution 'Data Warel	10use' (1 of 1 project)	
P La Source Assistant								Data Warehouse Deviert parameters	! 	
Common								Connection N	ls Nanagers	
Aggregate		Connection Manager					_	SSIS Packages	i integers	
Balanced Data Distribut		Mannings						🗄 Ioad Dim i	and Fact Tables.dtsx	
Conditional Split		Fror Output						🖺 Staging.dt	XX	
Data Conversion		Linoi Output						🖌 🕤 Package Parts	i	
🔍 👰 Data Streaming Destina			Avail	able Input Colu		Available Destinati	on	🕤 Control Fl	ow	
fr Derived Column			Nam	1e		Name		Miscellaneou	ŝ	
HDFS File Destination			PRO			PRODUCTLINE		🖌 🕍 Linked Azure	Resources	
HDFS File Source			MSI	RP -		MSRP		Azure-SSC	integration Kuntime	
Lookup			PRO	DUUCI_CODE		PRODUCT_CODE		Azure stor	aye	
[#] ¥ Merge							_			
🗘 Merge Join										
Multicast										
🖏 OData Source			Input Column		Destination Column					
ODBC Destination			PRODUCTLINE		PRODUCTLINE					
ODBC Source			MCRD		MSRD			0		- 1 V
GI OLE DB Command			DRODUCT CODE					Properties		······ • • •
HRow Count			PRODUCT_CODE		PRODUCT_CODE		10036	Staging Table Source Da	ta Flow Component	٣
Script Component							100%			
🖉 Slowly Changing Dime								ContactInfo	OLE DB Source;M	icrosoft Cor; 🔺
↓↑ Sort								Description	OLE DB Source	
¥ Union All								ID	25	
♦ Azure								IdentificationString	Staging Table Sou	rce
Other Transforms								lsDefaultLocale	True	
								LocaleID	English (United St	ates)
V Information								Name	Staging Table So	urce 🔻
Drag a toolbox item to SSIS Decigner to use it	Connection Managers				OK	Cancel He	p	Show Editor, Show Advan	ced Editor	
besigner to use it.		<u>п</u>								
	T SWITTENDQLEAFTESS.FTDM	m						Name		
								Specifies the name of the	component.	
Output										
🗇 Ready									↑ Add to <u>Source Con</u>	trol 🔺 🤚
		1					5	El 27°C Light rain	ດີເຊັນ 11	6PM
							Ľ		E 11/ E № 8/2	12021 🖓

Figure 4.20: Column Mappings for Columns From Staging Tables into Product Dimension Table.



Figure 4.21: Successful Execution of Data Flow Tasks From Staging Table to Product Dimension Table



Figure 4.22: Successful Execution of Data Flow Tasks From Staging Table to Contact Dimension Table



Figure 4.23: Successful Execution of Data Flow Tasks From Staging Table to StateDimension Table



Figure 4.24: Successful Execution of Data Flow Tasks From Staging Table to Date Dimension Table.

SQLQuery3.sql - SWIFTER\SQLEXPRESS.FYDWH	I (SWIFTER\ea, manni (51))* - Microsoft SQL Server Management Studio Quick Launch (Ctrl+Q) 👂	- 8 X
- File Edit View Query Project Tools V	Window Help	
🖁 🖸 🔹 🗇 📅 🕈 📩 🖕 💾 🖉 🚇 New Ou	uery 晶晶晶晶晶 从 品 向 🤊 • 🤄 • 🕅 🐳 🗊 PERSONS VS GENDER 🛛 • 🗊 🌶 🏛 🖸 • 🗉	
H W EVDWH		
Object Explorer 🔻 🖡 🗙	SQLQuery3sql - S., TERea_manni (51)/* * X FACTSALES.sql - not connected DIMSTATE.sql - not connected PRODUCT.sql - not connected	•
Connect 🛛 🌹 🌹 🗏 🍸 🖒 🚸	1 ESELECE * FROM [FYUM]. [CUTIACE]	†
🗉 🐻 SWIFTER\SQLEXPRESS (SQL Server 15.0.208) 🗚	2 SELECT * FRUM [FTUMH].[UHIE]	<u> </u>
🗉 🛑 Databases	3 SEECE * FRUM [FTUMF].[UUFSTATE]	
🗄 🛑 System Databases	4 [SEECE FINAN [FIDIN].[FIDIN].[FIDIN].	
🗄 🛑 Database Snapshots	110 % *)
🗄 🗑 AdventureWorksDW2019	I Results 📲 Messages	
	50 Scandnavian Gift Ideas 0695-34 6555 Larsson Maria	٨
	51 Marselle Mini Autos 91.24.4555 Lebihan Laurence	
	52 Gift Ideas Corp. 2035554407 Lewis Dan	
DOCTORSVSPATIENTS	53 Royal Canadian Collectabl (614) 555-4555 Lincoln Eizabeth	
	54 Mini Auto Werke 7675:3555 Mendel Roland	
🗃 📕 FIDWH	EE U.S. F. C.	
🖞 📕 Database Diagrams	ORDEHUARIE CITRID MONTHID YEARID	^
🛛 🧰 Tables	8 20051243 NULL NULL NULL	
FileTables	84 ZUG5TZ16 NULL NULL NULL	
External Tables	25 ZUG7Z49 NULL NULL NULL	
🕀 📕 Graph Tables	85 ZUA40F42 NULL NULL NULL	v
FYDWH.CONTACT		<u>^</u>
🗄 🏢 FYDWH.DATE	20 Caniv MA 512L7	
🗄 🏢 FYDWH.DIMSTATE	21 Brivin MA 5020	
🗄 🧱 FYDWH.FACTSALES	22 lile 5900	
🗄 🧱 FYDWH.PRODUCT	22 Erold (1000	
🗄 🏢 FYDWH.Source_Data_Staging	24 Nashua NH 62016	
🕀 📕 Views	25 9rah £700	
H 📕 External Resources	26 June 600	
🗄 📒 Synonyms		¥
🕀 📕 Programmability	PRODUCTUNE MSRP PRODUCT_CODE	^
Service Broker	26 Classic Cars 163 S18 2238	
H Storage	27 Vintage Cars 60 \$18,2240	
E Security	28 Trucks and Bu 122 S18, 2319	
	29 Vintage Cars 127 S18_2285	
III III	30 Trucks and Bu 60 S18_2432	
	31 Planes 84 \$18,2801	
	32 Matargoles 60 \$18,2625	
	33 Vintage Cars 168 S18, 2795	٧v
<pre></pre>	Query executed successfully. SWIFTER/sq. tranni (51) FYDWH 00000	02 508 rows
		22.014
📲 P O Q 📕 🧿	💿 🛓 🕮 🚺 🔍 🐵 🛊	2/2021

Figure 4.25: Selection of Data From Dimension Table.



Figure 4.26: SQL Command to Populate Data From Staging Table into Fact Table.



Figure 4.27: Mapping Columns From Staging Table to Fact Table.

K File Edit View Git Project Build Debug	ormat Test Analyze Tools Extensions Window Hel	p Search (Ctrl+Q) P Data Warehouse	9	- 0 X
6 • 0 8 • 🖕 🖬 📲 7 • C • Developi •	ault) > Continue + 🔍 🙆 ₊ 🕅 ■ O →	1 7 1 22 .		ġ ^r Admin
Process: [13388] DtsDebugHost.exe 🔹 💽 Lifecycle f	nts + Thread: 👘 🕫 XX St	ack Hame:		
SSIS Toolbox 🔹 🖣 🗙	Staging،dtsx [Design] Ioad Dim and Fact Tables.dtsx [Design) + X		* \$
Search SSIS Toolbox $ ho$	🖁 Control Flow 🔒 Data Flow 🤪 Parameters 📱 Event Hand	llers 🔚 Package Explorer 歧 Progress		
Favorites Fourier Assistant Surger Argintent	Data Row Task: 👔 LOAD FACT TABLE			v
Common R ^{IC} Aggregate Aggregate Aggregate Source vasiant Balanced Data Distributor Balanced Data Distributor Data Conversion Data Conversion Data Streaming Destination Derived Column HDFS File Destination HDFS File Destination HDFS File Destination HDFS File Source Lookup Werge Werge OData Source OData Source ODBC Destination ODBC Destination Row Count Script Component Slowly Changing Dimension If Sort		Slaging Table Source		
V Azure				
Orag a toolbox item to SSIS Designer to use it.	Connection Managers			
	Package execution completed. Click here to switch to design mode, or s put. Error List	velect Stop Debugging from the Debug menu,		
	Per		Addie Sauce	ca Control . 🔒
			T Add to Source	
📲 P O Q 🗍 🧿	🖉 🦉 😥 🔍 🖉		🧅 29°C Light rain \land 😳 🌾 🖮 🕸	8/2/2021

Figure 4.28: Successful Execution of Data Flow Tasks From Staging Table to Facts Table.

At this point implementation of the data warehouse is completed but the business intelligence module avails mentioning.

4.3.3 Implementation of business intelligence

The database holds a data warehouse but it is needed to create a reporting systems for the data present in the data warehouse, Visualizations are the best ways to report the data coming from any data source for human digestion. Afterall, the entire purpose of a data warehouse is to enhance better decision support.

After the data is imported measures are created. It is important to understand that measures in power BI are mathematical calculations for some certain facts to be expressed e.g., Imagine a column of the cost price of which we are to find out the total cost of items, a measure would be

```
Cost Price plus Cost Price =
SUM('''Fact Table$'''[Cost Price]) + SUM('''Fact
Table$'''[Cost Price])
```

created, to sum up, all of the cost prices across all the "cost price" columns in the table.



Figure 4.29: Data visualisation of data warehouse data with Microsoft Power BI

CHAPTER FIVE

SUMMARY AND CONCLUSION

This chapter contains the result summary of the project

5.1 Summary

The data warehouse facilitates reporting by implementing OLAP features, however data sources must be accommodated when conversion and extractions are performed. Although, the efficiency of a data warehouse will never be at it's peak because of some certain constraints, the data warehouse can be exploited to give certain results.

This paper presents a system for data warehousing and reporting services that will help small scale businesses thrive. With the application of different data flow tasks we are able to extend ordinary datasets into standard data warehouses.

5.1 Limitations

- i. Time constraints.
- ii. Previous works are scarce.
- iii. Datasets are scarce.

5.2 Recommendation

It is recommended that other researchers should engage implement a datawarehouse with other progamming languages e.g., python, R e.t.c., or solutions such as postgresql database server to find out which of the following approaches yields better performance of a data warehouse.

REFERENCES

Almeida, F. (2017). *Concepts and Fundaments of Data Warehousing and OLAP*. Porto: Instituto Superior Politécnico Gaya.

Arya, K. (2019, June 4). Latest Database Management Trends To Look Out For In 2021.

Retrieved from TechJockey.com: https://www.techjockey.com/blog/latest-databasemanagement-trends-look-2019

Babajide, A. (2012). Effects of Microfinance on Micro and Small Enterprises (MSEs) Growth in Nigeria. *Asian Economic and Financial Review*, *2*, 463-477.

Business Application Research Center. (2017, September 2). Using internal and external data for decision-making. Retrieved from bi-survey.com/: https://bi-survey.com/data-sources

Edward M. Leonard, B. (2011). ABSTRACT DESIGN AND IMPLEMTATION OF AN ENTERPRISE DATA WAREHOUSE. Marquette University.

Golfarelli, & Rizzi. (2009). Data Warehouse Design: Modern Principles and Methodologies.

Gupta, A. (2019). A Complete Reference for Informatica Power Center ETL Tool. International Journal of Trend in Scientific Research and Development, 1063-1070. doi:10.31142/ijtsrd19045

Gupta, A., & Sahayadhas, A. (2020, 7 11). A Comprehensive Survey to Design Efficient Data Warehouse for Betterment of Decision Support Systems for Management and Business Corporates. *International Journal of Management*, 463-471. Retrieved from http://www.iaeme.com/IJM/issues.asp?

Hughes, R. (2016). Agile Data Warehousing for the Small Enterprise. Miami: Elsevier Inc.

Inmon, W. H. (2002). *Building the Data Warehouse*. NEW YORK; CHICHESTER; WEINHEIM; BRISBANE; SINGAPORE; TORONTO: John Wiley & Sons, Inc.

Linstedt, D., & Olschimke, M. (2016). Introduction to Data Warehousing. *Building a Scalable Data Warehouse with Data Vault 2.0*.

Microsoft. (2021, 5 3). *SQL Server documentation*. Retrieved from SQL Server Technical Documentation: https://aka.ms/sqldocs

Mohania, M., Karlapalem, K., & Bellatreche, L. (2002). Some Issues in Design of Data Warehousing Systems.

Nielsen, M. A. (2015). Neural Networks and Deep Learning. Determination Press.

Nikhilesh. (2015, December 31). *Types of Dimensions in Data Warehouse*. Retrieved from helicaltech.com.

Ponniah, P. (2002). Data Warehousing Fundamentals. New York: John Wiley.

Qureshi, A. A., & Jamal, S. M. (2012, December). Taxonomy based Data Marts. International Journal of Computer Applications (0975 – 8887), 60, 6-13.

Rojas, H., Arias, K. A., & Renteria, R. (2020). Service-oriented architecture design for small and medium enterprises with infrastructure and cost optimization. *International Conference on Computer Science and Computational Intelligence*, 488-497.

Santoso, L. W., & Yulia. (2017). Data Warehouse with Big Data Technology for Higher Education. *Information Systems International Conference*, 93-99.

Schutz, C., Neumayr, B., Schrefl, M., & Neubock, T. (2016). Reference Modeling for Data Analysis: The BIRD Approach. *International Journal of Cooperative Information Systems*, 25. doi:10.1142/S0218843016500064.

Sreenivasaiah, P. K., & Kim, D. H. (2010). Current Trends and New Challenges of Databases and Web Applications for. *Frontiers in Physiology* · *December 2010*, 3-7. doi:10.3389/fphys.2010.00147

Watson, H. J. (2002). Recent Developments in Data Warehousing. *Communications of the Association for Information Systems*, 1-25. doi:10.17705/1CAIS.00801

APPENDIX

```
CREATE TABLE [FYDWH].[Source_Data_Staging](
       [Source StagingID] [int] IDENTITY(1,1) NOT NULL,
       [ORDERNUMBER] [int] NULL,
       [QUANTITYORDERED] [int] NULL,
       [PRICEEACH] [float] NULL,
       [SALES] [float] NULL,
       [ORDERLINENUMBER] [int] NULL,
       [ORDERDATE] [date] NULL,
       [STATUS] [varchar](50) NULL,
       [QTR_ID] [int] NULL,
       [MONTH_ID] [int] NULL,
       [YEAR_ID] [int] NULL,
       [PRODUCTLINE] [varchar](50) NULL,
       [MSRP] [int] NULL,
       [PRODUCT_CODE] [varchar](50) NULL,
       [CUSTOMERNAME] [varchar](50) NULL,
       [PHONE] [varchar] (50) NULL,
       [CITY] [varchar] (50) NULL,
[STATE] [varchar] (50) NULL,
       [POSTALCODE] [INT] NULL,
       [COUNTRY] [varchar] (50) NULL,
       [TERRITORY] [varchar] (50) NULL,
       [CONTACTNAME] [varchar] (50) NULL,
PRIMARY KEY CLUSTERED
       [Source StagingID] ASC
)WITH (PAD INDEX = OFF, STATISTICS NORECOMPUTE = OFF, IGNORE DUP KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

Sql Queries to Create Staging Table

```
[PRODUCTLINE] [varchar](50) NULL,
       [MSRP] [int] NULL,
       [PRODUCT_CODE] [varchar](50) NOT NULL,
PRIMARY KEY CLUSTERED
(
       [PRODUCT_CODE] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
                                                                                   Sql
GO
                                                                                 Queri
CREATE TABLE [FYDWH].[DIMSTATE](
       [CITY] [varchar] (50) NULL,
                                                                                es to
       [STATE] [varchar] (50) NULL,
                                                                                 Creat
       [POSTALCODE] [INT] NOT NULL,
PRIMARY KEY CLUSTERED
                                                                                     e
(
                                                                                 Dime
       [POSTALCODE] ASC
                                                                                 nsion
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
                                                                                 Table
) ON [PRIMARY]
                                                                                S
GO
CREATE TABLE [FYDWH].[CONTACT](
       [CUSTOMERNAME] [varchar](50) NULL,
       [PHONE] [varchar] (50) NULL,
       [CONTACTNAME] [varchar] (50) NOT NULL,
PRIMARY KEY CLUSTERED
       [CONTACTNAME] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
CREATE TABLE [FYDWH].[PRODUCT](
       [PRODUCTLINE] [varchar](50) NULL,
       [MSRP] [int] NULL,
       [PRODUCT_CODE] [varchar](50) NOT NULL,
PRIMARY KEY CLUSTERED
([PRODUCT_CODE] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

CREATE TABLE [FYDWH].[PRODUCT](

GO

```
[ORDERNUMBER] [int] NOT NULL,
       [QUANTITYORDERED] [int] NULL,
       [PRICEEACH] [float] NULL,
       [SALES] [float] NULL,
       [ORDERDATE] [date] NULL,
       [PRODUCT_CODE] [varchar](50) NULL,
       [POSTALCODE] [INT] NULL,
       [CONTACTNAME] [varchar] (50) NULL
GO
ALTER TABLE [FYDWH] [FACTSALES] WITH CHECK ADD CONSTRAINT [FK_DATEID] FOREIGN
KEY([ORDERDATE])
REFERENCES [FYDWH].[DATE]([ORDERDATE])
GO
ALTER TABLE [FYDWH]. [FACTSALES] CHECK CONSTRAINT [FK_DATEID]
GO
                                                                                  Sql
ALTER TABLE [FYDWH]. [FACTSALES] WITH CHECK ADD CONSTRAINT [FK_CONTACTID]
FOREIGN KEY([CONTACTNAME])
                                                                                  Qu
REFERENCES [FYDWH].[CONTACT]([CONTACTNAME])
                                                                                  erie
GO
                                                                                 s to
ALTER TABLE [FYDWH]. [FACTSALES] CHECK CONSTRAINT [FK_CONTACTID]
                                                                                  Cre
GO
                                                                                  ate
ALTER TABLE [FYDWH]. [FACTSALES] WITH CHECK ADD CONSTRAINT [FK_PRODUCTID]
                                                                                  Fac
FOREIGN KEY([PRODUCT_CODE])
REFERENCES [FYDWH].[PRODUCT]([PRODUCT_CODE])
GO
                                                                                 Tab
ALTER TABLE [FYDWH]. [FACTSALES] CHECK CONSTRAINT [FK_PRODUCTID]
                                                                                 les:
GO
ALTER TABLE [FYDWH].[FACTSALES] WITH CHECK ADD CONSTRAINT [FK_DIMSTATEID]
FOREIGN KEY([POSTALCODE])
REFERENCES [FYDWH].[DIMSTATE]([POSTALCODE])
GO
ALTER TABLE [FYDWH]. [FACTSALES] CHECK CONSTRAINT [FK_DIMSTATEID]
G0
```

t

G0

CREATE TABLE [FYDWH].[FACTSALES](
