Degree of Master of Philosophy

Investigation and Development of Fuzzy Logic Based Analytics for Data Warehousing

PARANA PALLIYA GURUGE DINESH ASANKA

158062B

Supervisor(s):

Dr. Amal Shehan Perera

2021-02-19

Department of Computer Science and Engineering

Faculty of Engineering

UNIVERSITY OF MORATUWA - SRI LANKA

Declaration

I declare that this is my own work and this thesis does not incorporate without acknowledgement any material previously submitted for a Degree in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Further, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: ………………………… Date:

PPG Dinesh Asanka

The above candidate has carried out research for the MPhil Dissertation under my supervision.

Signature of the Supervisor:……………….. Date:

Dr. Amal Shehan Perera

Acknowledgments

First, I express my sincere gratitude to my supervisor Dr. Amal Shehan Perera for guiding me throughout the project. Despite his busy schedule, he has given his fullest corporation whenever I wanted his assistance. I would also like to extend my appreciation to the panel members Dr. Dilum Bandara and Dr. Lochandaka Ranathunga for their valuable advice given to me throughout this project lasting more than four years. In addition, I am thankful to Prof. Sanath Jayasena for the valuable ideas given at the initial research stage. These ideas helped me to a great extent.

My loving parents and my wife always encouraged me towards higher studies. Especially, my wife and my two children made many sacrifices. I admit words cannot describe the sacrifices made. Anyhow, I would be failing in my duty if I leave them out of the honors.

This research is on data warehouse and I thank the three companies that contributed with their experience in data warehousing. The first data warehouse where I got the necessary experience was Ocean Software. There, I had an opportunity to work in the UK as well and got the most experience. At Pearson, I was able to implement some of the concepts that I learned importantly, administration and the configuration aspects of the data warehouse. At Virtusa, I was able to work on end-to-end aspects of data warehouse starting from the requirement gathering to deployment including the non-functional requirements of data warehouse. I deeply appreciate the kindness shown and thank all the co-workers and the management of these organizations.

Jagath Robotics (Pvt) Ltd provided me the data set which was used in this research as a proof of concept. Data gathering is a challenging task in any research. Collecting data from a reputed source helped me to proceed with the research.

I was awarded the Microsoft Most Valuable Professional by Microsoft. This allows me to gain experience with the Microsoft BI platform. The experience helped me a lot for this research. Hence, I thank Microsoft for awarding me the MVP award for 11 consecutive years.

I was fortunate enough for opportunities to lecture on data warehouse at different universities and institutions in Sri Lanka like the Sri Lanka Institute of Information Technology (SLIIT), Institute of Information Technology (IIT), Kothalawala Defence University (KDU), University of Ruhuna, University of Kelaniya, and University of UVA Wellassa. These institutions

provided me with the opportunity to lecture and these lectures helped enable me with the necessary knowledge in the area of the data warehouse. I also had the opportunity to supervise many data warehousing projects at MSc and BSc levels. This was a great opportunity. I was able to present data warehouse related sessions at places such as SQL Saturday, Colombo Big Data User Group, Sri Lanka Association of Artificial Intelligence (SLAAI), SQL Server Sri Lanka User Group (SSSLUG), Sri Lanka Azure Road Show, Colombo Dev Day, Malaysia TechInsights which allowed me to gain and present knowledge with data warehouse. I thank the organizers of these user groups for inviting me to these events so that I was able to gain knowledge in the data warehousing.

I pay my gratitude to all the teachers who taught me starting from my first school to B.Sc, MBA and MSc degrees. They gave me a sound base for many technical concepts and research methodologies.

Abstract

Data warehouse is a widely used technology that provides the employees who take strategic decisions within an enterprise with access to any level of required data. Historically, data warehouses were built on crisp values with a key assumption that one attribute value falls into one nominal value. Fuzzy Logic can be built into the data warehouse by treating the dimension value as weightages of different labels. However, in most of the attempts to implement a fuzzy data warehouse, they were limited and non-comprehensive in the implementation when considering end to end aspects of the data warehouse.

Using fuzzy techniques, it is possible to represent fuzzy conceptual information in the original domain, that would lead to better analysis. In this research, different types of fuzzy membership functions are defined using different techniques and data warehouse facts and dimensions are designed accordingly. There can be multiple fuzzy functions for one dimension as well as for one fact table depending on the business domain. Apart from defining fuzzy membership function using data-driven methods, there are other approaches of defining fuzzy membership functions such as a derived method where multiple fuzzy memberships are combined to define several fuzzy membership functions. In the literature reviewed, concepts like ETL and OLAP cube were found to be discussed in a limited manner. Non-function techniques are also identified and addressed in the means of validation, configuration, performance, security, scalability in order to make better usability of the fuzzy data warehouse.

The scope of this research revolves around end-to-end features of fuzzy data warehousing starting from data extraction and transformation to data warehouse modeling. Implementing a fuzzy data warehouse, helps to enable users with better analyses. To verify whether the proposed fuzzy data warehouse can be applied, a feasibility study is carried out for the domains in which fuzzy data warehousing can be implemented. Concepts related to the outcome from this research are verified with the use of a Sri Lankan plantation data set for four years. The results show that concepts introduced by this research can be implemented in realistic scenarios.

Key Words: Data Warehouse, Fuzzy Logic, Fuzzy Membership Function, ETL, OLAP

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Abbreviations

OLAP	Online Analytical Processing	NFR	Non-Functional Requirements
OLTP	Online Transactional Processing	ANN	Artificial Neural Network
ETL	Extract-Transform-Load	HMF	Horizontal Membership Function
BI	Business Intelligence	MF	Membership Function
MIS	Management of Information Systems	DBMS	Database Management System
CRM	Customer Relationship Management	SCD	Slowly Changing Dimension
HR	Human Resource	CDC	Change Data Capture
ERP	Enterprise Resource Planning	MDX	Multidimensional Expressions
KPI	Key Performance Indicators	MOLAP	Multidimensional Online Analytical Processing
FK	Foreign Key	ROLAP	Relational Online Analytical Processing
PK	Primary Key	HOLAP	Hybrid Online Analytical Processing

SVM	Support Vector Machine	MDDB	Multi-Dimensional Database
FD	Fuzzy Dimension	FC	Fuzzy Cube
AI	Artificial Intelligence	BI	Business Intelligence
ML	Machine Learning	RFI	Recency – Frequency - Intensity
RFM	Recency – Frequency - Monetary	SSIS	SQL Server Integration Services
SSAS	SQL Server Analysis Service	FIS	Fuzzy Inference System
DOM	Degree of Membership	IQR	Inter-Quartile Range
FIS	Fuzzy Inference System		

Chapter 1
INTRODUCTION

1.1 Prolegomena

Data warehouse is a comprehensive technology enabling employees who take strategic decisions within an enterprise with access to any level of essential data for insight into data in the organization. It is a framework that permits the strategic management access to all organizational data towards strategic decision making for a competitive advantage over competitors. Typically, a data warehouse is physically designed to store data in a central place. All the data in the organization in multiple data sources is integrated and stored along with the historical data. However, with the formation of a data warehouse, data can be utilized for different types of analytics: subjective, diagnostic, predictive and prescriptive.

In data warehouses, analysis is mostly carried out by dimension attributes. In most cases, attributes are a set of crisp values assigned. For example, most often, when the customer age group is defined, it will be assigned to one specific age group. Yet, there can be customers or employees of different ages or any other entity in that age group. In the data warehouse analysis, customers in one age group are treated as one set of customers with the same age group though there can be differences in the same age group if you consider the boundary values. To overcome such uncertainty or veracity in terms of big data challenges, fuzzy data warehouses can be proposed. By using fuzzy techniques, this information can be retained so that better analysis is possible. Further, dimensions let the users to define the measures in facts with human notions which are more detailed in analyzing. Apart from dimensions, design modifications for fact tables are also identified with respect to the fuzzy data warehouse.

Currently, a few attempts have been introduced with fuzzy techniques for fact and dimensional modeling. However, most of these techniques are limited to data warehouse design and are not extended to other technologies of a data warehouse. The aim of this study is to extend the fuzzy techniques to other key areas of the data warehouse such as slicing, dicing, roll-up, drill-down, aggregates, hierarchies, ETLs, OLAP Cubes, etc, critical to successful operation of a data warehouse.

In addition to functional aspects there are non-functional requirements for data warehouses that will define the implementation success of Fuzzy logic bases analytics for data warehousing. This research, considers: security, performance, validation, configuration and scalability as part of non-functional requirements as those are identified as the key elements in success of of implementation of fuzzy based data warehouse. The security aspect of the data warehouse is a major concern. Traditional transaction systems handled security system-wise. However, the data warehouse integrates data with multiple data sources making data warehouse security an important phenomenon. It may be necessary to use different configurations with a fuzzy data warehouse so that it enhances the usability of the proposed fuzzy warehouse technique. Since users need to get their requests performed satisfactorily within a given period of time, it is worthwhile to consider the performance of the fuzzy data warehouse.

With this research, different fuzzy techniques such as membership function, linguistic fuzzy functions, de-fuzzification and other fuzzy set operations are also discussed to enable fuzzy data warehouse design. Design, alone is insufficient. Proof of practical implementation related to the concept of the fuzzy data warehouse is essential. Therefore, an actual data set from the plantation domain is selected and used for the purpose.

1.2 Background and Motivations

Organisations the world over are fiercely competing against one another for adavatage to retain, promote and attract market share. In such a scenario, the data warehouse has become an absolute necessity for survival. However, over the years, it is noticed that there have been many improvements such as open data-sources to generate near real-time analytics, real-time data warehouses, etc. by using concepts such as data lakes, data factory, etc. Yet, uncertainty aspects prevail with every technology. Further, other domains such as the economy, politics, social, philosophical, etc. exist though not directly related to data warehouses due to complexity. For the big data properties, discussions are taking place for 7Vs and one of them is Veracity. Before the Veracity was added to the “V” family of big data, it was anticipated in the scientific and research community that received data is clean and accurate. This assumption was followed very much in the traditional and legacy data warehouses [42] in the industry as it is much easier to implement and less costly. The data warehouse is mostly used as a strategic tool as against auditing, making it less intense to place much effort into the veracity. Nevertheless, in the modern world, the role played by accuracy is huge and hence, uncertainty cannot be ignored.

Fuzzy Logic is one of the major technologies in use to confront uncertainty specifically in areas such as: Artificial Intelligence (AI), Business Intelligence (BI) and Machine Learning (ML). There have been discussions and research on fuzzy databases into detail. However, data warehouses aspect with respect to fuzzy is discussed minimally. Although there are a couple of research papers that discussed these fuzzy implementations with respect to data warehouses, the discussions were limited to fact tables and dimension tables design only. The implementations are not data-driven implementations but mostly heuristic decisions. According to the literature review carried out, it is evident that the other aspects of data warehouses such as OLAP Analysis [98], ETL [99], KPI, security [100], performance [101] are not discussed with necessary detail. As was mentioned, when other relative areas of data warehousing are ignored, the implementation of a data warehouse becomes challenging. This research is mainly aimed at data-driven design concepts for fuzzy data warehousing and other related areas of data warehousing needed to support fuzzy data warehouse.

1.3 Philosophical Aspect

In computer science, uncertainty with data means data which has deviated from the correct or intended values. In the times of the Big Data era, uncertainty or veracity is a defining characteristic of data which needs to be addressed scientifically rather than with an ad-hoc approach.

Discussions about uncertainty or vagueness initiated with a renowned research by the philosopher Max Black. He specified that in many instances the possible states cannot be clearly defined [99]. He presented an example which considered the attribute that a person's age is young. Since the term "young" has diverse explanations to different personalities as well as to different domains, it cannot be exclusively used to define the age at which an individual is young as against the age at which an individual is not considered to be young. Thus, the proposition is vaguely defined. Traditional logic does not valid under these situations as a different method of explanation must be established. Even if we consider Max Black's example on Age, in the data warehousing, Age is a typical attribute used for analysis. Therefore, the Max Black argument is valid for data warehousing.

The article "Vagueness an exercise in logical analysis" by Max Black, initial cited observations identified by the philosopher Plato on vagueness in geometry. Further, it has elaborated on the findings of Bertrand Russell (1923) who highlighted that "all traditional logic habitually assumes that precise symbols are being employed."

Albert Einstein said in 1921 at a lecture to Persian Academy indicated that all the crisp values that are defined in many domains are far from the truth [100]. With respect to data warehousing, when the crisp values are used, in the term of philosophical aspects it is an approximation. When an approximation is carried out, results are approximated which is not the truth values. Therefore, uncertainty aspects should be built into the system for the result to be more aligned with the correct output. As nature is built with uncertainty, it is critical to include uncertainty nature in the systems. Therefore, in data warehouse, including the fuzzy aspects makes the analysis closer to reality. This is essential with a data warehouse.

1.4 Problem in Brief

Currently, data warehouses use crisp sets and uncertainty is ignored or when making some assumptions uncertainties are neutralized. However, when uncertainty is ignored by any means, the final results are far from accurate or real. Therefore, this research is targeted to incorporate fuzzy techniques to include uncertainty in data warehouses so that organizations can implement fuzzy data warehouse to leverage more benefits than what is being used now. Further, in most instances, data warehouse is considered only from the design aspects. Research done by G. Garani, A. Chernov, I. Savvas and M. Butakova [96], Ruilian Hou [97], N. El Moukhi, I. El Azami and A. Mouloudi [98] indicated that data warehouse should be discussed on only the design but also data extraction and reporting etc. Further, these research has stressed that the data warehouse has to be stated that the data warehouse is linked with many other technologies such as ETL, KPI and OLAP Cubes, which, apparently means, to deliver the correct result, the entire data warehouse has to be extended to fuzzy techniques.

1.5 Aims and Objectives

Research objectives are divided into two areas: the main objectives and the sub-objectives. The primary objective of the research is to define the approach to implement fuzzy logic based analytics of data warehouse. Since there are many data warehouse sub-systems such as ETL, KPIs, and OLAP Cubes for data warehouse, the adaptation of fuzzy aspects of those sub-systems needs to be discussed as part of this research.

The following objectives are identified as outcomes of this research.

- Review current work on data warehousing, fuzzy data warehousing and fuzzy databases.

  Databases, Fuzzy and Data warehousing are identified as main domain areas of this research. Hence it was decided to study and review the current research work on data warehousing, fuzzy data warehousing and fuzzy databases.
- Conduct feasibility study to identify the domains and areas where the fuzzy data warehouse can be implemented. This objective will confirm that real-world implementation can utilize the outcome of this research.
- Define Fuzzy Membership Functions for different scenarios.

  Fuzzy membership function will play a key role in the fuzzy definition and shows no difference when it comes to fuzzy data warehousing as well. Since there are different data sets as well as memberships, function may change with time, it is important to define fuzzy membership function by using data-driven techniques. There is also a situation where fuzzy membership function is defined with a combination of multiple fuzzy membership functions. Most of the organizations have defined range with sense rather than a data-driven approach. Therefore, subjective definition of fuzzy membership function needs to be incorporated to this research.

- Implement Linguistic Analysis of Data warehousing.
  When the fuzzy set theory is implemented there is a possibility of implementing linguistic analysis on data warehouses.
- Design methodology for dimensions and fact tables to support fuzzy attributes. Different types of dimensions and fact tables are identified in Appendix A and Appendix B. Apart from defining a fuzzy element for dimensions and facts tables, defining linguistic fuzzy function is also part of this research.
- Design other relative features of the data warehouse to support fuzzy modeling as listed below:
  - Extract-Transform-Load (ETL)
  - Dimensional Hierarchies
  - Online Analytical Processing (OLAP) Cubes
  - Slicing, Dicing, Roll-up, Drill Down operations
- Define non-functional requirements such as security, scalability, performance, auditing, and configuration aspects for the fuzzy data warehouse. Non-functional requirements are covered as part of the process to improve the usability of the fuzzy data warehouse.
- Provide Proof of concepts for fuzzy data warehouse implementation using a real-world data set.
- Provide evaluation techniques for fuzzy logic based analytics.

It is a very important factor in this research to verify the proposed concepts by using a live and real data set. In this research, Sri Lankan plantation domain data for ten years was chosen as a sample data set. This data set will verify that the outcome of this research can be utilized for the implementation of fuzzy data warehouse concepts in the real world. Research papers were published upon completion of each objective.

1.6 Areas of Concern

In this research, different areas are identified to incorporate fuzzy implementation for the data warehouse. In a typical data warehouse, there are multiple technologies are used in varies areas such as data extraction, data transformation, data warehouse design, OLAP Cubes, and reporting in a typical architecture for the data warehouse [34]. Typically, data warehouse extracts data from diverse types of data sources which can be either relational data sources or non-relational data sources. After extracting data, data will be transformed into a suitable data format which can be used in the data warehouse. Transformed data will be stored to the data warehouse which is typically an OLTP database. In most cases, users do not access the OLTP database directly, mainly due to performance. Currently, OLAP cubes connected to the database are used to process the cubes. This contains reporting areas such as Data Mining, Reports, Dashboards, Pivot Tables, and Query, etc which will be utilized by the end-users for the analysis purposes. Apart from those areas, there are areas such as Monitoring, Administration which are an integral part of a data warehouse. A research paper by Arushi Kohil and Akshay Raina in 2014 [45] listed out the necessary activities for designing and deploying a data warehouse which is used as the areas of the research.

Since multiple sections are involved, in case of a change in one area, subsequent changes must be adapted to related areas. For example, design modification to the data warehouse in the areas of fact measures and dimension attributes will impact analysis, OLAP and data mining, etc. This is least covered in many of the researches, hence there exists a gap in the research area with respect to the fuzzy data warehouse. This research is an attempt to cover end to end aspects of the data warehouse.

1.7 Summary

This chapter has discussed the motivation for the research. The thought behind this research is to mainly integrate benefits of fuzzy techniques to data warehousing so that it can do better analytics as against a legacy crisp data warehouse. Since most of the other research covers the design aspect of fuzzy data warehousing and other aspects are neglected, this research captures all the aspects of the data warehouse such as key performance indicators, dimensions, fact tables, OLAP Cubes and Its operations, etc.

Special attention is provided for Extract-Transform-Load as there are changes to ETL operations and configurations due to the changes in design in the data warehouse. Linguistic analysis, an important factor in the fuzzy data warehouse, is also considered for better analyses. A traditional and legacy data warehouse takes the path of approximation where the introduction of approximation could lead to loss of information. In such a scenario, the fuzzy data warehouse is philosophically found to be more aligned towards the real world. For further progress with the research it is essential to analyse the present state of research with respect to Data Warehouse, Fuzzy Logic, ETL and Fuzzy database modelling.

This thesis is organized into ten chapters where the second chapter is focusing on the literature review of fuzzy data warehousing research and techniques. The research methodology is discussed in the Chapter 3 while the Chapter 4 is dedicated to a feasibility study of the proposed concept of fuzzy logic based data warehousing. Further, Chapter 5 and Chapter 6 are discussed the design aspects of Fuzzy Membership functions and Fact and Dimension Tables Design respectively. The Chapter 7 is focused to Extract-Transform, Load aspects of data warehousing and Fuzzy OLAP cubes are discussed at the Chapter 8. Implementation challenges such as non-functional requirements are discussed in the Chapter 9 and finally, Chapter 10 is dedicated to the Conclusion of the research.

In the next chapter, Literature Review is about current research state and is discussed in detail in the identified areas of fuzzy data warehousing.

Chapter 2

FUZZY DATA WAREHOUSE – STATE-OF-THE-ART

2.1 Introduction

As indicated in the Chapter 1, this research is about an investigation into and the development of a fuzzy data warehouse using fuzzy logic technology. The data warehouse and fuzzy technologies are used. Data Warehouse is not a trending technology. Therefore, it is significant to understand that the data warehouse is a comprehensive technology in use with research or in industry today. Hence, multiple business domains need to be identified so that the data warehouse is useful to many domains and many organizations. To identify that the data warehouse is important technology even in today's competitive world, the literature review reveals latest information regarding data warehouse implementations in various domains. Since this research is on end-to-end aspects of the data warehouse, and since the fuzzy implementation can be impacted on those areas such as Data warehouse Design, Extract-Transform-Tool (ETL), OLAP Cube, the literature review covers these aspects as well.

To identify the research gap, it is significant to find out the existing state of the research. Hence, the literature review is about fuzzy data warehouse implementation at various areas of the data warehouse with this research considering all aspects of data warehouses. The main part of this research is to design a fuzzy data warehouse. Since data warehouse typically adopts relational design structures of a database such as tables, attributes, views, primary keys, etc., many lessons can be learned from the fuzzy database design aspects as well. Considering this aspect, the literature review on the fuzzy databases serves as a means to determine what design aspects can be adopted to fuzzy data warehouse design.

This research is to implement fuzzy technologies to the data warehouse. As such, it is vital to understand the basis of fuzzy knowledge with its implementations. Fuzzy logic solutions are also evaluated in order to find out the relevant implementations using fuzzy logic. To achieve the purpose, the literature review deals with fuzzy logic implementation. Concepts of De-fuzzification and fuzzy rules are also identified during the literature review process. Since OLAP cube is one of the best ways to analyze data warehouse details, it needs to be revealed whether there are any other implementations done on the fuzzy OLAP cube. Therefore, the literature review is extended to the fuzzy OLAP cube as well.

Apart from the technical analysis of technologies for this research, an analysis should determine whether the fuzzy data warehouse concepts can be applied in the industry. Therefore, a feasibility study, to verify how fuzzy data warehouse can be employed in varies domains of implementation, is also carried out. This enables users to identify in which places fuzzy data warehouse can be implemented and at what level.

2.2 Data Warehouse

A data warehouse is typically used as an analytical platform in many organizations [39]. The data warehouse system has differed from the standard OLTP systems mainly from the mandate it has to cater for. In the data warehouse design, special attention is given to develop a multi-dimensional data model. A multi-dimensional data model is usually structured around a central transaction. The main components of a multi-dimensional model are facts and dimensions entities. The fact table comprises of the measurement of multiple business processes which is what business is interested in. These measures provide the additive or semi-additive numerical values that act as an independent variable which are analyzed through dimensional attributes. The facts are typically numerical measures such as sales volume, salary, count, etc. These measures are divided into three types such as additive measures, semi-additive measures, and non-additive measures [1]. Data warehouse design and architecture is discussed in detail in [2], in which the design of data warehousing is discussed with respect to fact tables and dimensions tables. In addition, in [4], the possible implementation for data warehouse was discussed. Appendix A: Fact Tables and Appendix B: Dimension Tables have discussed types of fact and dimension tables in detail which are used to design data warehouse for different purposes.

Data is most likely the most valuable asset for any organization to operate successfully and compete with its competitors. Data that originates from a many heterogeneous systems should be collected and stored in a framework so that the historical data is accessible. This data should to be prepared, cleaned and enhanced for future reference, diagnostic, data analytics, and future predictions. The data warehouse structure is revealing a significant feature of the data warehousing [47] as structural design will help to model data in the organization. Nowadays, most of the organizations depends heavily on the data stored in their data warehouse. Ralph Kimball [18] has proposed departmental data marts to suit for the departmental operations and later to integrate to build the enterprise data warehouse.

In today's highly fiercely competitive market environment, it has now become a very much necessary to create a sustainable inexpensive benefit in contradiction of its contestants by creating a data system with which the current operations can be easily tracked and predicted for future strategies towards competitive advantage. Therefore, one of the developments in the market at the moment is the increasing concentration in the progress of massive enterprise scale data warehouses. The data warehouse is a subject-oriented, time-dependent, integrated and non-volatile collection of data [38]. Further, the data warehouse is a system that extracts, cleans, transforms and loads the heterogenous source data into a data warehouse or data mart. In addition, data warehouse provisions to take data-driven efficient decisions by querying and analyzing. Data warehousing environment contains the extraction of data from a relational database, NO-SQL databases besides from non-relational data sources such as audios, videos, emails, Transformation, Loading also known as ETL techniques, Online Analytical Processing (OLAP) tools and client analysis applications [31]. Figure 2-1 shows the details of reference architecture of modern data warehouse.

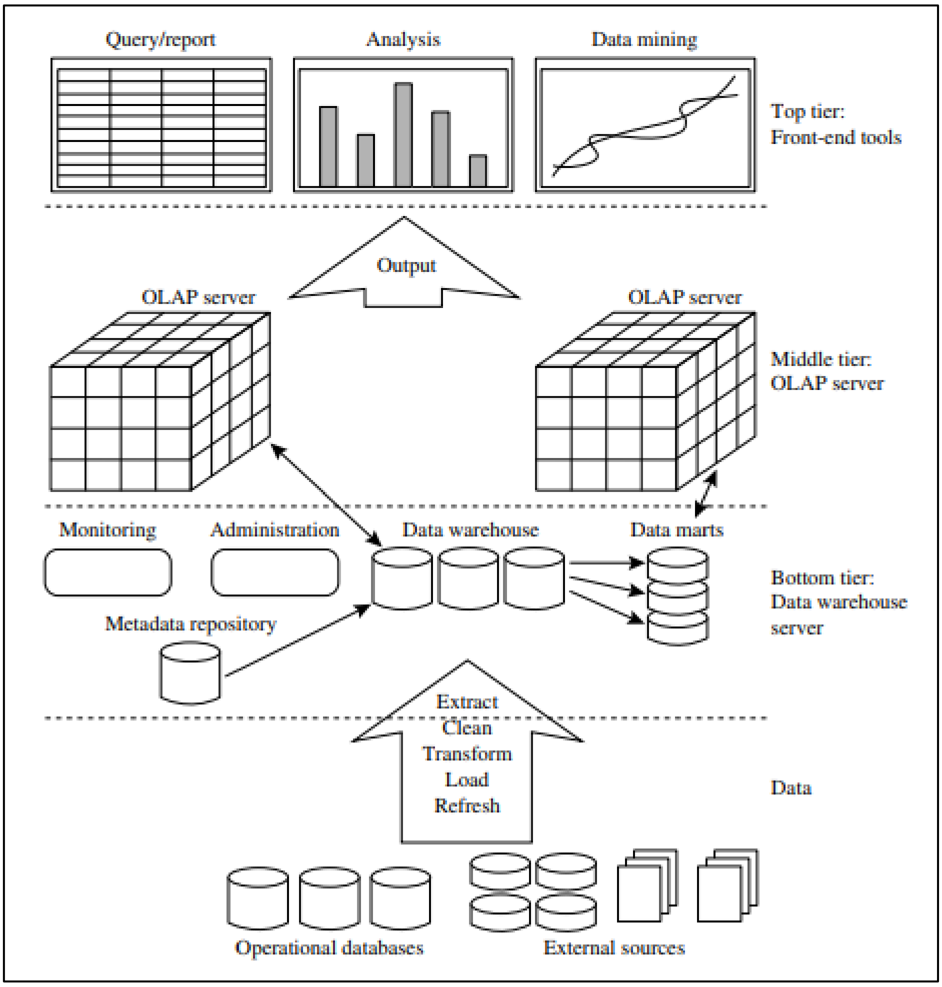


Figure 2-1: Data Warehouse Reference Architecture

Source: [Han J., Kamber M., 2012]

As shown in the Figure 2-1, data warehouse deals with many areas such as data extraction to reporting including data warehouse design and OLAP cubes.

In some context, Business Intelligence is referred to the data warehouse whereas business Intelligence is a combination of ETL, data warehouse engine, OLAP tools, data mining applications and decision taking model [57]. The data warehouse is considered as the major componets of Business Intelligence system that stores aggregated and historical data where it is loaded with many operational and transactional data sources such as Management Information System, Customer Relation Management System, and Enterprise Resource Planning System [83].

The architecture of the data warehouse includes not only data modeling nut also the security, metadata management, the extent of query requirement. Data warehouse architecture further utilized comprehensive technology. Metadata is data that can be stored either as an unstructured or in a semi-structured form that is beneficial in data warehouses. Data warehouses mainly have two types of design patterns namely star schema and snowflake schema which are discussed in detail in Appendix Z.

The data warehouse has three different physical environments such as Operational Environment, Data Warehouse Environment and Business Intelligent Environment [62] as presented in Figure 2-2.

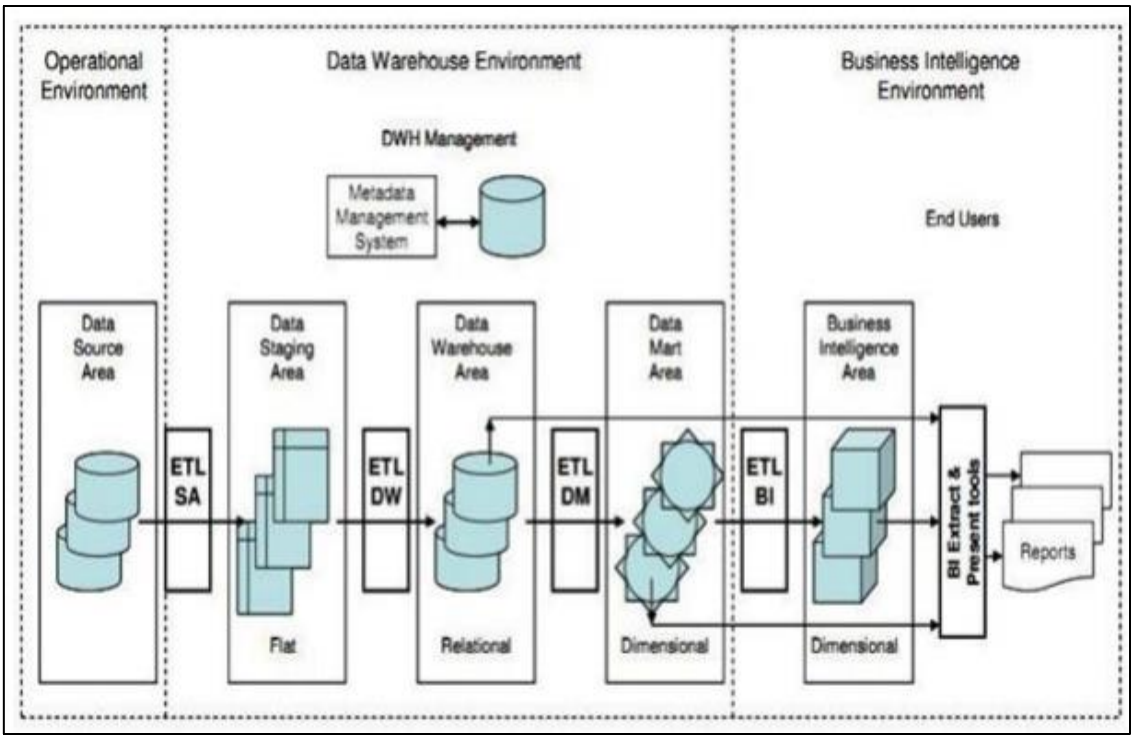


Figure 2-2: Different Data Warehouse Environments

[Source: Samuel A., Pandey A.K., Sharma V.K.]

In the above-said data warehouse environment, there are mini ETLs between each stage. In the case of implementing a fuzzy data warehouse, each stage needs to be considered differently by analyzing the properties of each area.

Operation environment stores the transactions of the business. It is located external to the data warehouse, as the data warehouse has no control upon the data and the format of the data. Operational Environment or Operational Data Store (ODS) is typically used to generate reports which are not possible with the available sources. Since ODS are more transactional reports, there is no requirement to extend fuzzy concepts to the ODS sub-system. Mostly, ODS is mandated to generate cross system reports as there is no other system to generate these reports. The data in the above said ODS systems are stored in many heterogeneous formats such as flat files, hierarchical, document, and relational databases [62] among other formats.

Data-ETL methods are designed to extract data from multiple and heterogeneous data sources. Further, ETLs are used to clean the data, carry out required data transformations, and load the transformed data to data warehouse, and then load the data to data marts in data warehouse. Apart from above tasks, the ETL is used to produce and preserve a central metadata repository and provision the enterprise data warehouse [73].

Data marts can be viewed as an extension to the data warehouse or to mini data warehouses. The data comes from multiple data sources and is integrated before entering the data warehouse system. The data marts comprise subject-specific information to support the necessities of the end-users in separate logical or physical business units. Data marts can be provided with responses at higher rate to end-user requests as most of the data warehouse queries are directed to pre-calculated, aggregated data [73]. If de-normalized structures are not used in the data marts, there will be a performance impact to the end user queries which are accessing large volume of data.

The above analysis convinces that the data warehouse is implemented in many sectors to leverage benefits to analyze data and implement data-driven strategic decisions using available data. Recent research papers on data warehouse are referred to verify that the data warehouse is being used in many sectors even today and also to find to which scale data warehouse is being used.

Agriculture domain is measured as a multipart combine process of air, rainwater, climate, soil, plant, animal, micro-organism, that are undependable and irregular in circulation, covering varies ranges in the domain and interacting at different scales [10]. Use of data warehouse in agriculture is to provide information for several purposes: the efficient consumption of manure [16], reducing the usage of herbicide [19], plant safety [40],variety-specific information [24], management of environment risks in agriculture [9], integrated nutrition management [54], rainforest management [60], yield infection control [38], agricultural processes and its extensions [68], farm modernization [69], seasonal weather forecast [34], integrated contamination regulation [46], labor necessities and arrangement of land usages[50], profit intensification and risk reduction [52] etc. They have been designed and applied successfully mainly to improve commercial revenues, for shifting farming practices or minimizing environmental threats. These are the main benefits from data warehouse among many others. In another paper [21], a system was proposed which collects numerous commodity data from open data sources of the Indian government, filter it by MapReduce steps as table functions and for generating current and yearly developments that can help agriculturalists to sell their harvests at finest values, and going further, delivers detailed investigation for several commodities and marketplaces.

Customer Relationship Management (CRM) system is a concept that needs composite information technology solutions exceptional those of Enterprise Resource Planning (ERP) systems. This exploitation of data warehouse that will assist the practice and administration of information is viewed as the key providers to marketing accomplishments. Data warehouse plays an important role in CRM like in other systems. Data warehouse will enable the association of customer information and the formation of customer knowledge that is critical for preserving client interactions in organization [20]. CRM is discussed in detail in Chapter 4 with regards to possible implementation of fuzzy data warehouses with respect to CRM.

Advanced banking is being considered as a cost reduction and appropriate path by the client and banks when these parties contact with each other to receive services.

It saves time and cost for both parties and there are no geographical and time restrictions. Customers of the bank can have their business transactions globally through information and communication technology at different times of the day [17]. In another research conducted for banks in India [30], it is observed that international banks use data warehousing to measure various aspects. Some of those aspects are performance, risk management, profitability investigation and decision support. Implementation of business intelligence ability is a vital step for banks to tactically use information technology for their operations. Majority of the banks in India have implemented fundamental banking system solutions for multi-channel service distribution to cater to wide variety needs of the customer. To have an incorporated view of their data, data warehouse is best option for the bank so that they can introduced business intelligence through information technology specifically by using the data warehouse. In another research, [58] conducted in India, it has been identified data warehousing and data mining as an opportunity for the banking domain in India to fulfil customer requirement. In another research to help anti-money laundering systems [29], it has been identified that data warehousing is one of the key technologies that the bank uses for marketing management, knowledge management, field service management, sales service automation, and personalization. Apart from banking, there are other financial services where data warehouse technologies are extended to exploit its benefits in the banking sector. In Turkey, [25] the business intelligence and data warehouse concepts were proposed to reduce the number of non-compliant tax payers and ease the complexities on inspection units. The use of the computerized system and data warehousing have resulted in a substantial reduction in the number of non-compliant tax payers.

In recent days, most of the healthcare organizations have built data warehouses that are same to those in the non-healthcare setting but there are differences to other areas mainly due to its domain properties. In healthcare institutes, these data warehouses gather data from multiple areas. Some of those areas are patient care, population-based databases, financial, claims, and administrative transactional systems. Later, collected data is organized to support information retrieval, business intelligence, research and decision making, and future predictions. One of the important aspects in healthcare is Information management that is one of the reasons to use data warehouse in healthcare. Information Management in resource providers, management, and clinical operations are critical to the success of healthcare. Those data warehouses have developed into the central information management platform for decision support. Healthcare systems haves to concentrate more on privacy and security of the health information when compared to non-healthcare system. Further, healthcare has to compliance with federal and government guidelines and protocols, and organizational procedures [27]. The research paper [33] objects to view the effectiveness of using OLAP with a healthcare data warehouse to help the medical professionals to effectively perform their judgments. The healthcare data warehouse was built based on health records from Basra Health Department in Iraq. Three OLAP systems were designed and implemented using Microsoft BI family. Microsoft Excel 2013 Pivot tables, Microsoft SQL Server Integration Service (SSIS) and Microsoft SQL Server Reporting Service (SSRS) are the techniques used for this research. Specifically, this research has utilized SQL Server 2012 version. In this research paper, in the DimAge dimension, AgeRange is used. AgeRange attribute can be considered for the fuzzy data warehouse. Definition of fuzzy membership function will be discussed in detail at the Chapter 5. However, they have not utilized the SQL Server Analysis Service (SSAS) 2012 that is a key member in the Microsoft BI family.

In a research paper [70], a detailed study was carried to implement data warehouse in the healthcare domain. In this research paper, detailed discussion was done on the execution of an OLAP tool on the Comprehensive Assessment for Tracking Communal Health data warehouse used by knowledge workers at a regional health preparation agency in USA. By using this OLAP tool, they were able to accomplish an advice-giving role to their customers. Further they were able to increase the branding in the community they provide services. This research enhances a different aspect to previous research in data warehousing by concentrating on the decision support competencies of OLAP tools.

Another research paper [39] has presented a clinical decision support system based on OLAP with data mining to identify whether a patient can be diagnosed with diabetes with the probabilities of high, low, or medium. It is obvious from the result that the proposed system overcomes the physical plan design and execution requirement in the data warehousing atmosphere.

Agriculture, CRM and Finance, etc. seem to be popular sectors to implement data warehouses. However, in some research [56] data warehousing techniques are used to increase service quality in prison services and operations in Nigeria.

This section expresses that data warehouse is not a legacy technology and many sectors are still leveraging its features. This means that fuzzy data warehouse implementation can be utilized more for these sectors and will be enhanced in the current data warehouse analysis into further analysis. By implementing fuzzy concepts into data warehouse, it can be said that more and precious analyses can be carried out.

2.3 Extract Transform and Load

Data Extraction, Transform and Load also state as ETL are a critical part of a data warehouse. As data warehouse integrates data between multiple heterogeneous data sources, ETL has the mandate of extracting from multiple source systems. After extraction from the Transform phase extracted data is fitted to the multi-dimensional data model. ETL is a challenging task in the data warehouse process as it has to deal with heterogeneous data sources which are mostly unknown area for the data warehouse developers and transformation is also a rigorous process. As described by Ralph Kimball, there are 34 sub-systems to be considered for ETL listed in Appendix X for more detail. ETL is more complicated as data should be retrieved not only from OLTP but from sources such as NO-SQL databases, text files, websites, flat files, spreadsheets, images, videos, chats, web sites etc. [32]. With respect to fuzzy logic, there are scenarios to implement fuzzy lookup and fuzzy groups [87] as data is received from multiple sources and there are excessive chances that standard lookup might not be sufficient. In the case of ETLs, there are cases where both fuzzy lookups and fuzzy grouping needs to be included for better transformations.

Research [22] has introduced Fuzzy ETL into the data warehouse as presented in Figure 2-3.

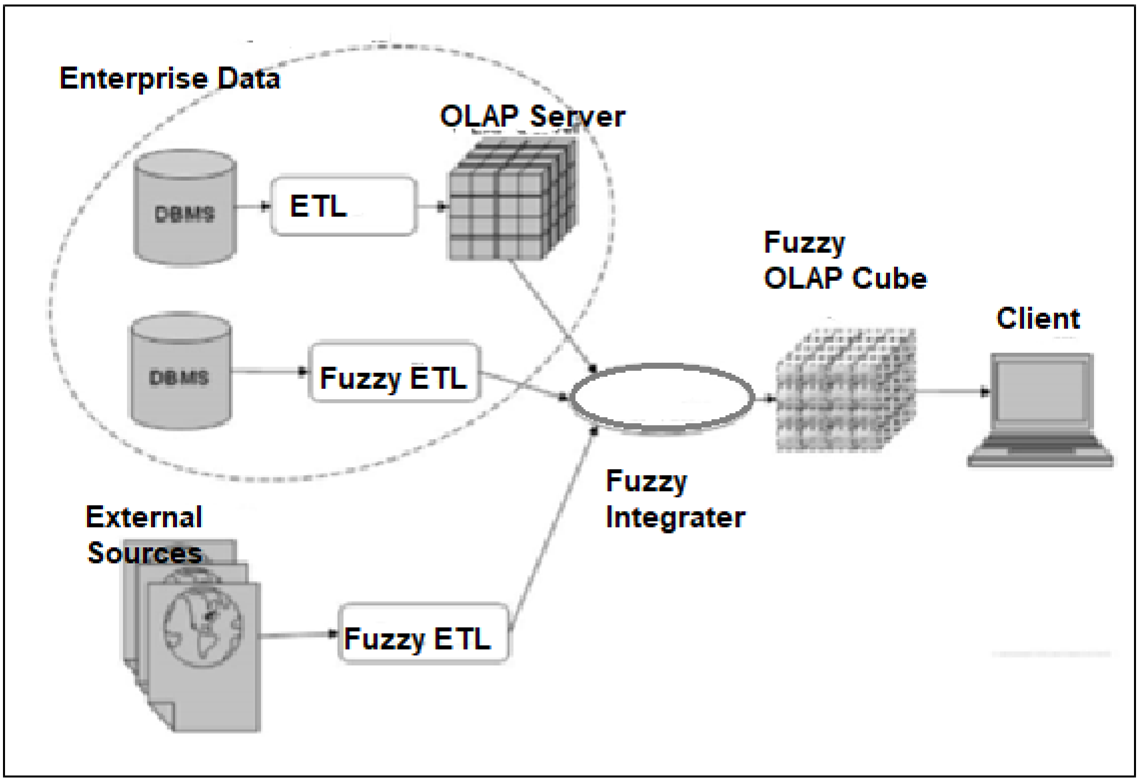


Figure 2-3: Implementation of Fuzzy ETL

[Source: Broek, P.V.D., Noppen, J., 2004]

In the above implementation of ETL, there are two layers for ETL. There is a standard ETL implementation as well as fuzzy ETL layers. The additional layer is introduced to integrate data from standard ETL and Fuzzy ETL using a fuzzy integrator to support fuzzy OLAP. In this method, it was not clear, how to distinguish fuzzy ETL and normal standard ETL. Also, in the above implementation, there are a few Fuzzy ETs. Out of the two ETLs, one ETL is to extract data from a DBMS and the other ETL is for the external sources. It is not clear, what the approach would be if there are multiple DBMS and only option would be introduced - a staging environment and then implement fuzzy ETL to extract data from the staging environment

2.4 Fuzzy Database Modelling

Typically, a data warehouse is built on top of relational database technologies. Therefore, in order to implement fuzzy techniques to the fuzzy data warehouse, it is logical to analyze fuzzy techniques of relational databases. However, before moving into a fuzzy data warehouse, it is essential to analyse research and implementation for fuzzy database modeling as fuzzy database concepts can be utilized to implement a fuzzy data warehouse. Though the fuzzy data warehouse appears to be relatively new to the research, the literature review indicates that fuzzy databases are not new. There are few attempts made to model relational or transactional data with fuzzy concepts.

A Research paper [14] has come up with a relational algebra operation consisting of the same four parts as regular relational algebra operation. Further, there is a part introduced to define the minimum similarity thresholds. Equation 2-1 shows relational algebra equation for fuzzy database.

PROJECT (REL 0: APTITUDE, AGE) WITH THRES (THRES (APTITUDE) ≥ 0.70, THRES (AGE) ≥ 0.85)

Equation (2-1)

The relation REL 0 has any quantity of domains however one of the variables is named as APTITUDE and the next variable is named as AGE. The relation created by the PROJECT operation covers only domain APTITUDE and AGE. That is, the new relation is created by removing all domains from REL 0 excluding these two domains. The final form of the relation is attained by amalgamating rows through the respective domain values until no additional rows can be combined without disturbing either the minimum threshold for APTITUDE of 0.70 or the minimum threshold for AGE of 0.85. The minimum threshold limitations will be then referred to as the level values.

To prove its theory, a research paper has come up with an application with regards to the baseball team selection. In the case of baseball, there are skilled places for each player. Some players are superior in numerous positions and whom are called all-rounders. When swapping, these players with the other players, fuzzy query features are employed in which data is stored in a database in relational schema.

A research paper [7] that is focused on Inadequate Information in Fuzzy Databases, has gone into a step where it has defined data types for fuzzy databases as normally defined for traditional databases as presented in Figure 2-4.

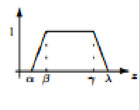
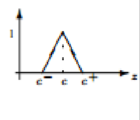
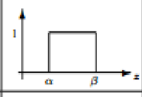
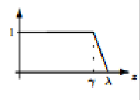
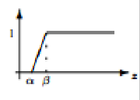
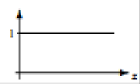
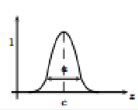
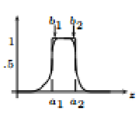
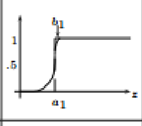
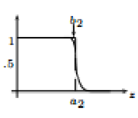
Data type A	Model	Representation	Parameters	$\mu_A(z)$
Fuzzy range label e.g. *age*■ more or less between 20 and 30	I.1		$\alpha, \beta, \gamma, \lambda$	$\mu_A(z) = \begin{cases} 1, & \text{if } \beta \le z \le \gamma; \\ \frac{\lambda - z}{\lambda - \gamma}, & \text{if } \gamma < z < \lambda; \\ \frac{z-\alpha}{\beta-\alpha}, & \text{if } \alpha < z < \beta; \\ 0, & \text{Otherwise.} \end{cases}$
Approximate value e.g. *age*■about 35	I.2		c, c^-, c^+	$\mu_A(z) = \begin{cases} 1, & \text{if } z = c; \\ \frac{c^+ - z}{c^+ - c}, & c < z < c^+; \\ \frac{z - c^-}{c - c^-}, & c^- < z < c; \\ 0, & \text{Otherwise.} \end{cases}$
Interval e.g. *age* $\in [25, 35]$	I.3		α, β	$\mu_A(z) = \begin{cases} 1, & \text{if } \alpha \le z \le \beta; \\ 0, & \text{Otherwise.} \end{cases}$
Less than value e.g. age■ less than 35	I.4		γ, λ	$\mu_A(z) = \begin{cases} 1, & \text{if } z \le \gamma; \\ 0, & \text{if } z < \lambda; \\ \frac{\lambda - z}{\lambda - \gamma}, & \text{if } \gamma \le z \le \lambda. \end{cases}$
More than value e.g. age■ more than 35	I.5		α, β	$\mu_A(z) = \begin{cases} 1, & \text{if } z \ge \beta; \\ 0, & \text{if } z \le \alpha; \\ \frac{z-\alpha}{\beta-\alpha}, & \text{if } \alpha < z < \beta. \end{cases}$
Unknown	I.6			$\mu_A(z) = 1 : z \ge 0$
Undefined	I.7			$\mu_A(z) = 0 ; z \ge 0$
Real number e.g. *age*■30	I.8		c	$\mu_A(z) = \begin{cases} 1, & \text{if } z = c; \\ 0, & \text{Otherwise.} \end{cases}$
Linguistic label e.g. *age*■young	II.1		a, c	$\mu_A(z) = \frac{1}{(1+(a(z-c)^2}; \ z \ge 0$
Linguistic label e.g. *age*■young	II.2		a_1, a_2, b_1, b_2	$\mu_A(z) = \begin{cases} \frac{1}{1+\frac{z-a_1-b_1}{b_1}}, & \text{if } z < a_1 + b_1, \\ 1, & \text{if } a_1 + b_1 \le z \le a_2 - b_2; \\ \frac{1}{1+\frac{z-a_2+b_2}{b_2}}, & \text{if } z > a_2 - b_2. \end{cases}$
Linguistic label *age*■very old	II.3		a_1, b_1	$\mu_A(z) = \begin{cases} \frac{1}{1+\frac{z-a_1-b_1}{b_1}}, & \text{if } z < a_1 + b_1; \\ 1, & \text{if } a_1 + b_1 \le z; \end{cases}$
Linguistic label e.g. *age*■very young	II.4		a_2, b_2	$\mu_A(z) = \begin{cases} 1, & \text{if } z \le a_2 - b_2; \\ \frac{1}{1+\frac{z-a_2+b_2}{b_2}}, & \text{if } z > a_2 - b_2. \end{cases}$

Figure 2-4: Fuzzy Data Types

[Source: Bahdi A., Chakhar S., Naiija Y., 2005]

Typically, in databases, there are data types like string, integer, date, image, text, etc. However, this research has introduced several data types for fuzzy databases which can be utilized for fuzzy data warehouses. Unlike standard database data types, fuzzy database data types have different parameters and usage of these data types is not straight forward like in the fuzzy databases. Out of these above data types, for this research, there are some important fuzzy data types that need to be further considered.

In addition to this research, another research [51], have adopted four levels for implementing imperfect information in databases. Those levels are database system level, database level, metadata level and model base level. A description of these levels follows.

- Database System Level: This level consists of extended data manipulation languages devoted to handling different fuzzy operations that the database system should support.
- Database Level: This level was considered for the way the imperfect information is internally saved. This concerns both attributes values and extended definition of different fuzzy relations or classes.
- Metadata Level: This level is concerned about the intended definition of fuzzy relations or classes. In some of the research, this level is called meta-knowledge.
- Model base Level: This level groups the definition of all function used to calculate membership degrees and the functions associated with diverse data types which are used to produce their probable distribution [13].

In the case of the fuzzy data warehouse, modeling can be done at Metadata level which will be more effective than implementing at another level. Database level implementation would be difficult since it is difficult to find the relevant fuzzy data types, instead these fuzzy data types can be implemented at the Meta data level.

Literature review on fuzzy database modeling has indicated that there are few techniques which can be utilized in the area of fuzzy data warehouse designing as well. An important thing in fuzzy database modeling is that the introduction of fuzzy data types as data types for databases. Another important outcome is that the level of applying the fuzzy technique to the databases. Fuzzy Logic can be applied to database level metadata level as well as to the model level and it is identified that for fuzzy data warehouse Metadata level is selected.

2.5 Fuzzy Data Warehouse Modelling

There are a few attempts made to implement fuzzy techniques into the data warehouse. As discoursed in the section 2.4, fuzzy relational databases are used as the base for the fuzzy data warehouse modeling research.

A research paper by Sapir [63] discussed on fuzzy data warehouse design met has suggested an improvement to the standard Ralph Kimball four-step methods for dimensional modeling. In this altered comprehensive process, research has suggested a step to recognise a grain of the crisp and fuzzy data. In the dimension selection process, fuzzy dimension identified as an addition of a crisp dimension. However, philosophically crisp is an reduction of fuzzy modeling as many of the real-world cases fall into fuzzy dimensions which are clearly different from what this research paper claims.

In the customer dimension, age is selected as a fuzzy dimension attribute. Age attribute can be represented as a fuzzy function as indicated in Figure 2-5.

SOURCE: TABLE- CUSTOMERS, FIELD- AGE
DESTINATION: FUZZY DIMENSION- DIMCUSTOMERAGE
CRISP DIMENSION - DIMCUSTOMERS
FUZZY AGE LABELS:

LABEL NAME	ALPHA	BETA	GAMMA	DELTA
YOUNG	15	20	33	45
MID AGE	30	39	45	60
OLD	50	60	INFINITY	NULL

INTERVALS: YES {0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1}

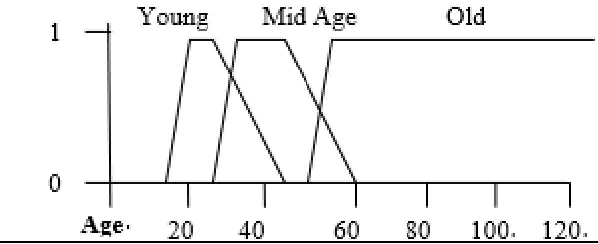


Figure 2-5: The Fuzzy Customer Age Dimension Report

In this example, Trapezoidal membership function was used with alpha, beta, gamma and delta values. It is not clear, on what basis Trapezoidal function was selected and the reason for those parameter values.

In this research, defined parameters are stored in a configuration table and can be modified which will make easy the configurations. However, it is not clear whether the dimensions are considered as type 2 slowly changing dimensions or whether they are simple type 1 overwriting dimensions.

Fuzzy hierarchies are presented in this research in order to perform further analysis into data. Technically, fuzzy hierarchy contains parent and child levels where they have a many-to-many relationship. Dimensional hierarchies are very common in the traditional data warehouse. Not only data warehouses, even in database systems, it is constantly recommended to remove the many-to-many relationship by introducing a bridge or intermediate table. In traditional data warehouses, product categories and subcategories fall into a natural dimensional hierarchy. In fuzzy data warehouse modeling, one product can be part of numerous categories with different degrees. For example, the product Helmet can be allocated to different membership degrees to two or more product categories as shown in the Equation 2-2.

$$\frac{0.2}{\text{Component}} + \frac{0.4}{\text{Accessories}} + \frac{0.4}{\text{Wear}}$$

Equation (2-2)

This degree of these categories can be modified depending on the diverse seasons and promotion events. When it is changed, the historical aspect needs to be stored as type 2 slowly changing dimensions as specified in Appendix B:Dimension Tables which are not considered in this research.

Fuzzy measures are also considered in this research. Initially, fuzzy and crisp facts will be incorporated to the fact table. Typically, during the determination of the facts, the designer must recognize what the business users want to analyse. For example, crisp facts are the sales quantity, sales amount, etc. There can be computed measure columns such as profits (profits = sales price – cost price) and it will be calculated and stored in the fact table to improve the performance of the OLAP queries which is a standard practice in data warehouse design [32]. Fuzzy measures are defined on the base of crisp measures when the designer desires to represent measures that cannot be interpreted to a crisp number. For example, if the transaction cost is considered, fuzzy measures called sale profitability into low_Profits, normal_Profits, and high_Profits can be defined. This measure can be computed from the ratio between the sales price and cost price of the transaction. If the ratio is high, then it will have a higher membership degree to the high_Profits label. Alternative option would be to use fuzzy dimensions attributes as measures. As an example, product popularity fuzzy measure can be added. Each measure in the fact table has a product popularity values to the fact table line.

Figure 2-6 shows the schema diagram of the fuzzy data warehouse proposed by the research for the AdventureworksDW database which, the sample OLAP database provided for the Microsoft SQL Server 2012 RDBMS database engine.

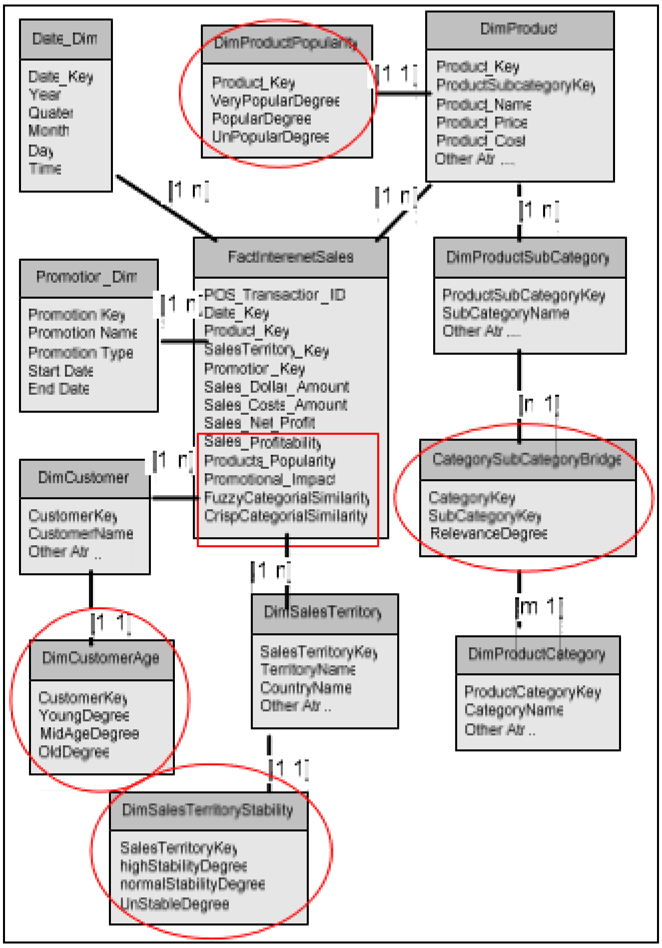


Figure 2-6: The Fuzzy Data Warehouse Schema Comprising Fuzzy Hierarchies

Highlighted dimensions are the dimensions where fuzzy techniques were introduced. In this research, Microsoft SQL Server 2012 sample database *AdventureWorksDW* is used. However, the AdventureWorksDW database has some simple implementation and this database has data which is neatly done. This scenario is far from the real world, therefore, this database cannot be considered as real-world implementation.

There are problematic areas identified in this research. In classical data warehousing, aggregation functions such as minimum, maximum, sum, count, average, variance are being used frequently. However, in the case of fuzzy data warehousing, it has been identified as a problematic area. Further, it has been identified that the pre-processing of the fuzzy measures is time-consuming. Not only the pre-processing but also the query time may be increased as well.

Apart from the above research, many other approaches to the integration of fuzzy concepts in the data warehouse have been proposed in few other research. Another research [22] introduces fuzzy dimension, in which a dimension attribute is composed of a fuzzy concept. This research has proposed a two-layer model in which the core objective is to hide the difficulty of the model and provide a more understandable result to the end-user. To accomplish this, the research has proposed to use a fuzzy summary operator that provides more instinctive results but retaining as much information as possible. The architectural diagram of the suggested model is presented in Figure 2-7.

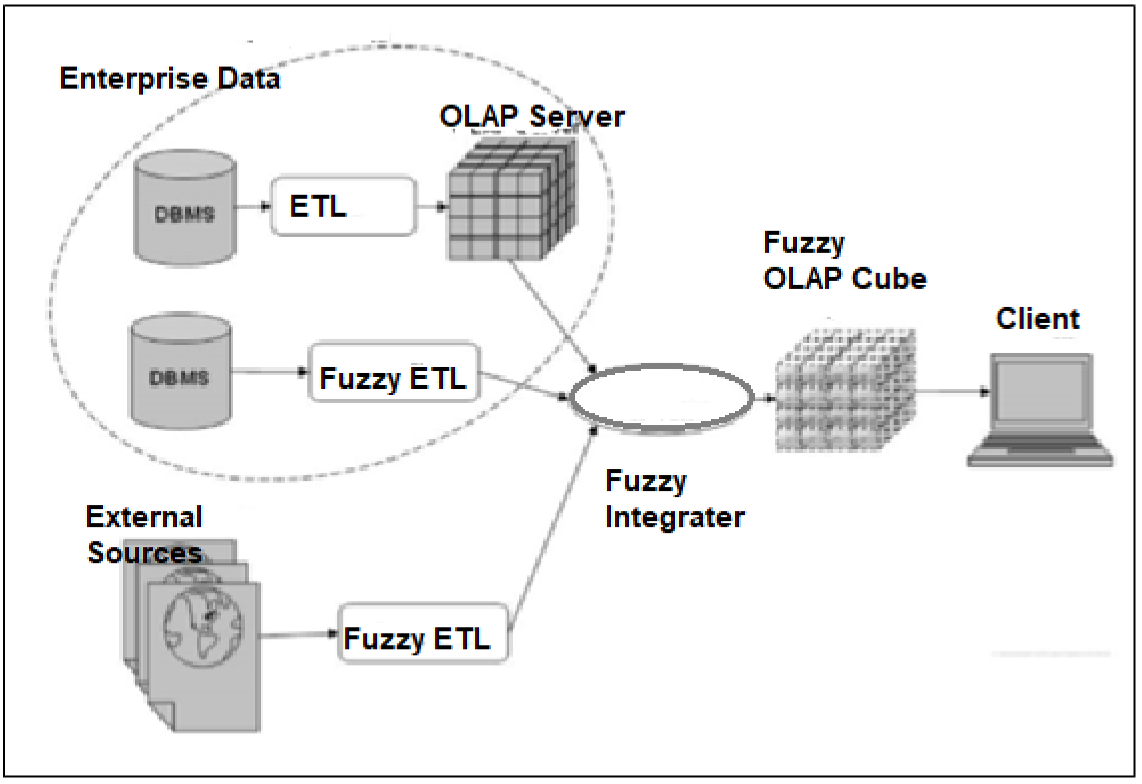


Figure 2-7: Proposed Architecture for Fuzzy Multidimensional model for OLAP

[Broek, P.V.D., Noppen, J., 2004]

In this proposed model, database schemata and cube structure will not include fuzzy techniques which were included in as future work. However, there is a fuzzy integrator and fuzzy ETL, which is not discussed in detail in this research paper. This research has an option of disabling the fuzzy layer which is a great option as it would enable configurability of the system. Since fuzzy data warehouse is a new component, end users may not require and it would be a great option to disable the fuzzy component when necessary. However, there is a performance degrade at fuzzy integrator which is a concerning factor of this proposed method.

A different research paper focused on fuzzy data warehouse model [28], introduced strategies for modeling the fuzzy meta-tables and a meta-model for the fuzzy data warehouse are discussed in this research paper.

A research paper on OLAP over Undefined and imprecise data [15] is more focused about demonstrating data ambiguity, such as imprecision and uncertainty. In this research, uncertain measures are stored as a range of possible values together with the probability of each possible value. Those values have representedvalue for an uncertain measure as a probability distribution function over values from an associated base domain. This research has generalized the OLAP model to represent data ambiguity. This has been addressed by both imprecise dimension values and uncertain measure values. This research has introduced a criterion that guides the choice of semantics for aggregation queries over ambiguous data. It has been proposed an algorithm for calculating aggregation queries including MAXIMUM, MINIMUM, VARIANCE, DEVIATION, AVERAGE, COUNT, DISTINCT_COUNT and SUM for ordinary measures and LinOp for uncertain measures along with complexity analysis. One of the major aspect of this research is that it has done a performance comparison with the introduction of uncertain data representation.

Another research [53] has recognized that in most cases fuzzy inference is defined in an ad-hoc method entirely based on an expert knowledge. When domains consist of rational knowledge in Business Intelligence, expert's decisions are also reliable. Therefore, it has been recommended that the use of fuzzy inference for modeling common sense should be stretched in Business Intelligence for better decision making.

A Research focused on Ambiguity in Decision Making [65] has stressed the use of fuzzy techniques for decision making. The research paper has discussed a reason to investigate the data using fuzzy techniques. The main emphasis of the research is its prominence on designing the warehouse. In this research, relational database format was selected as the data storage while it supports fuzzy queries.

The literature review on fuzzy data warehouse indicates that schema design is the major consideration factor for all the research. In the literature reviewed, it was identified that most of the cases, fuzzy membership function was subjectively defined and not as a data-driven method. This emphasises the need of data driven method to define a fuzzy membership function. Further, most of the research carried out research on the well- defined data set, hence, it is difficult to assess how fuzzy data warehouse can be employed in the real world scenarios.

2.6 Fuzzy Logic

This research is targeted to identify opportunities to extend the usage of fuzzy logic in data warehousing. Though there is not much research being done on data warehouse and fuzzy logic together, there is limited research carried out on the implementation of fuzzy logic in different domains. Current research has spread across numerous domains from control, pattern recognition, and knowledge-based systems to computer vision and artificial intelligence. Fuzzy logic emerged as a new way of representing uncertainty in last few decades. Fuzziness can be used to explain certain types of uncertainty related with linguistic information or intuitive information. Examples of vague information are that the data quality is “good” or that the transparency of an optical element is “acceptable.” In addition, in terms of semantics, even the term vague and fuzzy cannot be usually considered synonyms as described Zadeh [37]. According to Zadeh, every vague proposition is fuzzy.

Discussions about vagueness initiated with a well-known work by the Alex Black (1937) well-defined a vague proposition as a proposition where the possible statuses defined with regard to inclusion. As an example, consider the age of a person as young. As the term “young” has different interpretations to different individuals and for different situations, it is difficult to decisively determine the age at which an individual is not considered to be “young”. Thus, the proposition is vaguely defined. Traditional logic does not hold under these conditions; therefore, fuzzy logic is defined as a different method of interpretation.

Fuzzy logic is benefited from the neural network in the form of more profound learning abilities. Most categories employed in describing real-world objects do not process well-defined crisp limitations. For example, notations like high temperature, medium pressure, small neighborhood, etc., in which the adjective identifies the sources of fuzziness. From the inauguration of the theory, fuzzy sets have been defined as a collection of objects with the membership values between zero and one. zero means complete exclusion whereas one means complete inclusion of membership. The membership value expresses the degree to which each object is compatible with the properties of features distinctive to the collection. Fuzzy sets are formally defined as follows:

A fuzzy set is characterized by a membership function mapping the elements of a domain, space or universe of discourse X to the unit interval [0,1] which is A: X -> [0,1]. A fuzzy set is a generalization of the concept of set whom membership function takes on only two values {0, 1}. Fuzzy membership function defines how this function belongs to one set. Typical fuzzy membership functions are described in detail in the Appendix D: Types of Fuzzy Membership Functions. In this, six types of fuzzy membership functions are described out of which triangular and trapezoidal fuzzy membership functions are more popular among the users. There are a few methods to select a required membership function which is given in detail in Appendix F: Fuzzy Membership Function Determination.

Fuzzy logic is used for ranking in sports as well. In the research paper titled, Fuzzy logic and its application in football team standings [81], fuzzy set theory and fuzzy clustering are used to analyze football team standings. Based on some certain rules, the proposed four parameters to calculate fuzzy similar matrix, obtained equivalence matrix.

Orthodox techniques for system analysis are becoming unsuited for dealing with humanistic systems. This is mainly due to the fact that these systems are influenced by human judgment, perception and emotions. Definition of linguistics variable will help to introduce human perception into the data warehouses. Definitions of the linguistic variable for fuzzy is discussed in detail in Appendix G: Computing with Linguistic Variables.

In a research paper [5], type – II fuzzy set was used to overcome uncertainty and developed a fuzzy decision support system of manures usage in the field of agriculture. The type-II fuzzy system takes cropping time and soil nutrition in the form of spatial surfaces as input and it has used a type-II fuzzy membership function and indicates fuzzy rules on it in the fuzzy inference engine. In the proposed system, type-II fuzzy membership functions were introduced for Nitrogen, Phosphorus, Potassium and cropping time in the domain of agriculture. In this research, the method of discovery of membership function was not disclosed and it is assumed that membership functions were derived using some industry parameters.

In a research paper concentrated on Fuzzy Iterative Learning for Traffic Flow Systems [76], the fuzzy technique was suggested for traffic flow systems of a single lane freeway with arbitrary bounded off-ramp traffic volumes. In another research paper titled, A Fuzzy Goal Programming Model for Bakery Production [11] has addressed uncertainty with bakery production by means of fuzzy systems. The fuzzy goal programming technique is used to maximize the daily sales profits, minimize overtime and maximize the utility of machines during the daily production of a small and medium enterprise in producing different bakery products. In this research, the fuzzy membership function was defined as a function of the goal. However, as in many other research, the definition of the fuzzy membership function is not discussed in detail and in most of the scenarios, definition of the fuzzy membership functions are very primitive.

In a research paper [26] to help medical doctors to refine knowledge for the stratification of unstable angina risk the respect to vagueness in information this paper develops an intelligent system combining genetic algorithm and fuzzy association rule mining. In this research, they have utilized fuzzy membership functions of both semi-trapezoidal shape and triangular shape as those functions are in general the most suitable shapes and the extensively employed in fuzzy systems. In this research, twenty-seven variables were identified out of which nine variables are fuzzy sets variables such as Age, Systolic Blood Pressure, etc and eighteen variables are crisp sets variables such as Sex, Smoke, Atrial premature beat, etc.

In yet another research paper, [2] fuzzy membership function was built to the classification of fruit Harumanis which is a different version of mango. In this research, the fuzzy classification system was introduced to classify the Harumanis fruit quality based on fruit outline features. Couple of input features of fruit quality attributes, including size and shape, were acquired from different Harumanis images. Triangular fuzzy membership function was introduced in this research as it appears to be the simplest fuzzy membership function. Also, it is mentioned that the triangular fuzzy membership function has fewer overlap data.

In a further research paper focused in Fuzzy Clustering Technique with Big Data for Disease Diagnosis, [64] a fuzzy technique is integrated to handle vagueness in medical big data. A squeezed fuzzy model is produced using subtractive clustering. As shown in Figure 2-8 below, Fuzzy Inference System (FIS) was used to disease diagnosis with five inputs and two outputs.

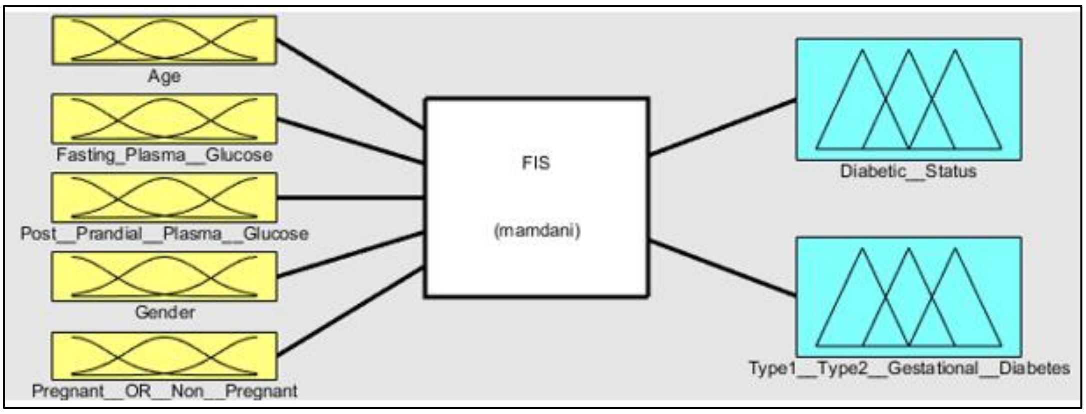


Figure 2-8: Fuzzy Inference System for Disease Diagnosis

In the defined FIS, two fuzzy membership functions were used. Considered inputs were Age, Fasting Plasma Glucose, Post Prandial Plasma Glucose, Gender, Pregnant or Non-Pregnant. As in most research, fuzzy membership functions were not derived in the data-driven method in this research but simple technique was used. Though it was not specified, it is understandable that Gender and Pregnant or non-pregnant is not fuzzy functions but crisp functions. This means that, in the derived fuzzy inference system, both fuzzy functions and standard functions are used in combination.

A further research paper [82], identifies several types of fuzzy membership functions defined. In this research paper, several types of fuzzy membership functions were assessed in the fuzzy control of an induction motor drive. The common fuzzy membership functions considered were triangular, trapezoidal, gaussian, bell, sigmoidal and polynomial types. In this research, the triangular function was used as a base fuzzy membership function to derive other types of fuzzy membership functions.

However, in a research paper by Karina Tomaszewska [71], the application of horizontal fuzzy membership function is discussed in detail. This research paper emphasises that horizontal fuzzy membership function as considerably facilitating fuzzy calculations to deliver multi-dimensional problem solutions. Pairwise comparison is another method introduced by Saaty (1974) to determine fuzzy membership function. The ranking ordering method is a simplified method of pairwise comparison method introduced by Saaty. In the Ranking ordering is to assess preferences by an individual, a committee, a poll and other opinion approaches are employed to assign fuzzy membership values to a fuzzy variable. Preference is determined by pairwise comparisons. In another research [12], a new method is introduced to determine rankings from Pairwise comparisons. In this research linear programming is used to identify the fuzzy membership function.

In most research, triangular and trapezoidal fuzzy membership functions are used and in a research paper on Developing membership functions has mentioned that triangular membership functions are more suitable in imprecise conditions such as in agricultural data [79].

Most of the research has used naïve methods to calculate fuzzy membership function and it is imminent that research has more focus towards the outcome of the fuzzy logic rather than the fuzzy membership function. This has led to place less focus on the method of determining the fuzzy membership function. Though there are a couple of researches done with semi-data-driven method to derive fuzzy membership function, it is evident that scientific data driven method is needed to determine the fuzzy membership function.

2.7 De-Fuzzification and Fuzzy Rules

Defining fuzzy rules and de-fuzzification is part of fuzzy logic functionalities. Different fuzzy variables can be combined to build the fuzzy rules so that output fuzzy attributes can be defined. Depending on the de-fuzzification method, the output value of the state can be defined.

In the research paper on Linguistic Variables [61] the triangular fuzzy membership functions were defined arbitrarily. To combine the multiple fuzzy variables, max-min direct inference approach was used. In this research paper, the use of a de-fuzzification method that captures the joint effect of the linguistic variables. The aim of this paper is to study the implications of using two alternatives de-fuzzification methods that are largest of maximum and center of the area and to highlight numerous interpretation and modeling challenges related with each de-fuzzification method.

In this research paper, finding the product attractiveness was done by using two fuzzy variables Perceived Customer Service and Perceived Delivery Timeliness. Fuzzy rules are defined as Figure 2-8 by using Mamdani’s Max-Min model.

Defining fuzzy rules and de-fuzzification are part of fuzzy logic functionalities. Different fuzzy variables can be combined to build the fuzzy rules so that output fuzzy attributes can be defined. Depending on the de-fuzzification method, the output value of the state can be defined.

Taking into consideration what has been done so far, this research paper relates to finding product attractiveness using two fuzzy variables Perceived Customer Service and Perceived Delivery Timeliness. Fuzzy rules are defined as Figure 2-9 by using Mamdani's Max-Min model.

	Perceived Customer Service	Perceived Delivery Timeliness	Product Attractiveness
1	Low	Low	Low
2	Low	Medium	Low
3	Low	High	Medium
4	Medium	Low	Medium
5	Medium	Medium	Medium
6	Medium	High	High
7	High	Low	Medium
8	High	Medium	High
9	High	High	High

Figure 2-9: Fuzzy Rules

After the fuzzifications were done, de-fuzzification was done using the Largest of Maximum method and the center of Area method as shown in Figure 2-10.

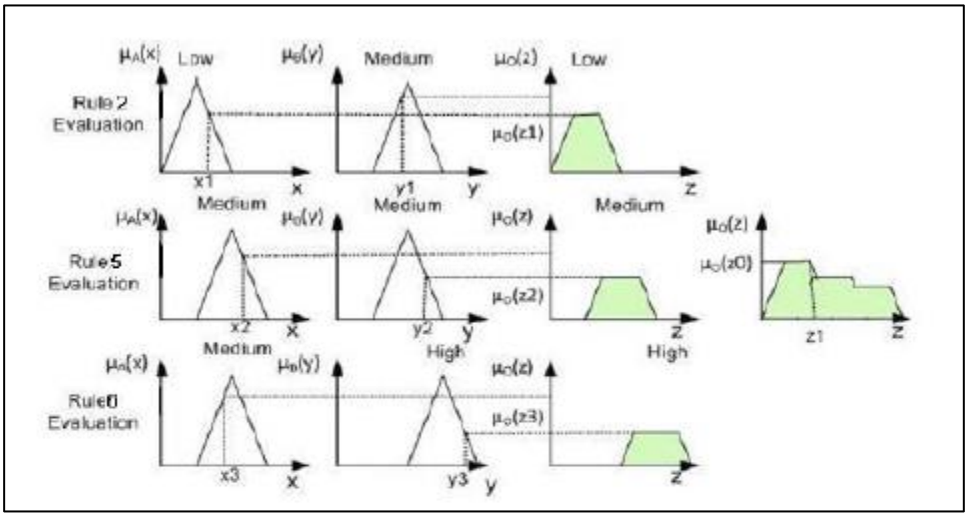


Figure 2-10: Largest of Maximum De-fuzzification of the Fuzzy Set

In another research paper by V.N. Huynh [37] a model was suggested for the parametric demonstration of linguistic hedges or linguistic modifiers in Zadeh's fuzzy logic. In the suggested model truth value for every linguistic truth value is recognized depending on the primary terms.

In another research paper [85] three-layered data warehouse semantic model comprising of quantitative summarization, qualitative summarization and quantifier summarization is suggested. A rule-based technique was introduced for different scenarios in another research paper [86]. This research paper has suggested an improved rule-based method with reduced rules in linguistic analysis. Nevertheless, this research has not identified the process of physical implementation of these models with respect to data warehouses.

2.8 OLAP Cube

An Online Analytical Processing (OLAP) cube is a multi-dimensional database that is enhanced for a data warehouse as well as for online OLAP applications mainly for faster analysis of data as data warehouse deals with the large volume of data. If the standard relational databases are used for data analysis in data warehouse, there will be huge performance issues, since databases are mainly support for OLTP transactions.

Therefore, in today's competitive environment, organizations prefer to use OLAP cubes in order to provide on time information to the end users.

An OLAP cube is a technique of storing data in a multi-dimensional form, commonly for reporting commitments. In OLAP cubes, measures are categorized by dimensions. OLAP cubes are often pre-summarized across dimensions to extremely improve query time over relational databases which is called Multi-dimensional OLAP (MOLAP). In the case of MOLAP, summarized data is pre-calculated and stored in the OLAP cube. Relational OLAP (ROLAP) is the opposite of the MOLAP which will use the data from the sources. In between MOLAP and ROLAP, there exists another cube type called Hybrid OLAP which has the properties of both MOLAP and ROLAP.

Table 2-1 has a detailed list of comparison of properties of MOLAP and ROLAP storage mechanisms of OLAP.

Table 2-1: Comparison of Properties for MOLAP and ROLAP

	ROLAP	**MOLAP**
Full Form	ROLAP stands for Relational Online Analytical Processing.	MOLAP stands for Multidimensional Online Analytical Processing.
Storage & Fetched	Data is stored and fetched from the main data warehouse.	Data is stored and fetched from the Proprietary database Multidimensional Databases.
Data Form	Data is stored in the form of relational tables.	Data is Stored in the large multidimensional array made of data cubes.
Data volumes	Large data volumes.	Limited summaries data is kept in Multidimensional Databases.
Technology	Uses Complex SQL queries to fetch data from the main warehouse.	MOLAP engine created a pre-calculated and pre-fabricated data cubes for multidimensional data views.

In most of the implementation, tools have the options of different configuration of different level of MOLAP and ROLAP. In SQL Server Analysis Service 2012, there is another storage mode called Hybrid OLAP (HOLAP) which is the combination of both MOLAP and HOLAP. The query language is employed to interact and perform tasks with OLAP cubes is multi-dimensional expressions (MDX) queries. The MDX queries have been adopted by many OLAP vendors of multi-dimensional databases to support their own OLAP databases.

As shown in Figure 2-11, Kimbal's data warehousing toolkit is described where, a star schema is mapped to an OLAP cube.

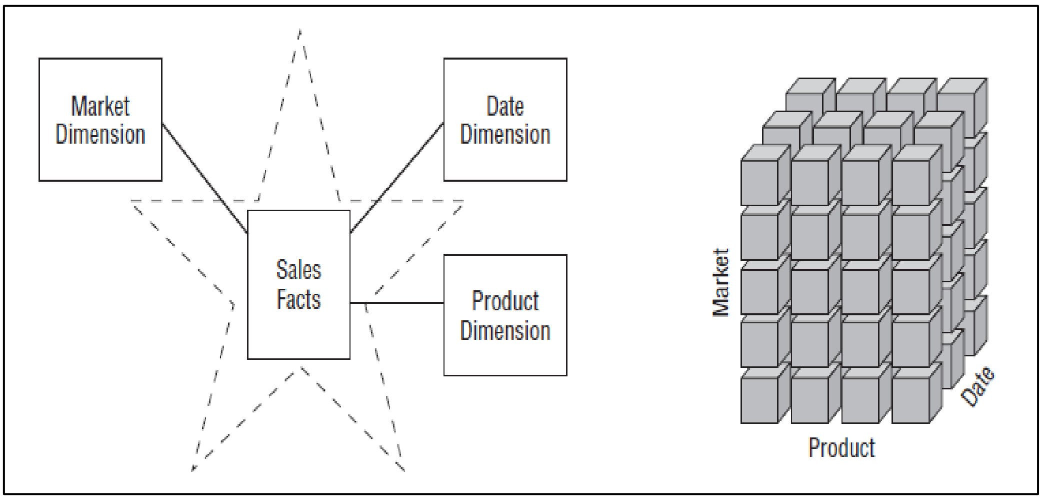


Figure 2-11: Mapping of Star Schema to OLAP Cube

[Source: Hand D., Mannila H., Smyth P.]

Typically, a star schema is preferred over snow-flex schema in data warehouse design due to many reasons as discussed in Appendix Z. One of the major reason to adopt star schema is the simple implementation. As shown in Figure 2-10, the OLAP cube design is mapped to Star Schema. In the case of snowflake schema, referenced relationships are used mapped to OLAP cube.

In the same book, Figure 2-12 explains how slicing and dicing can be done using an OLAP cube.

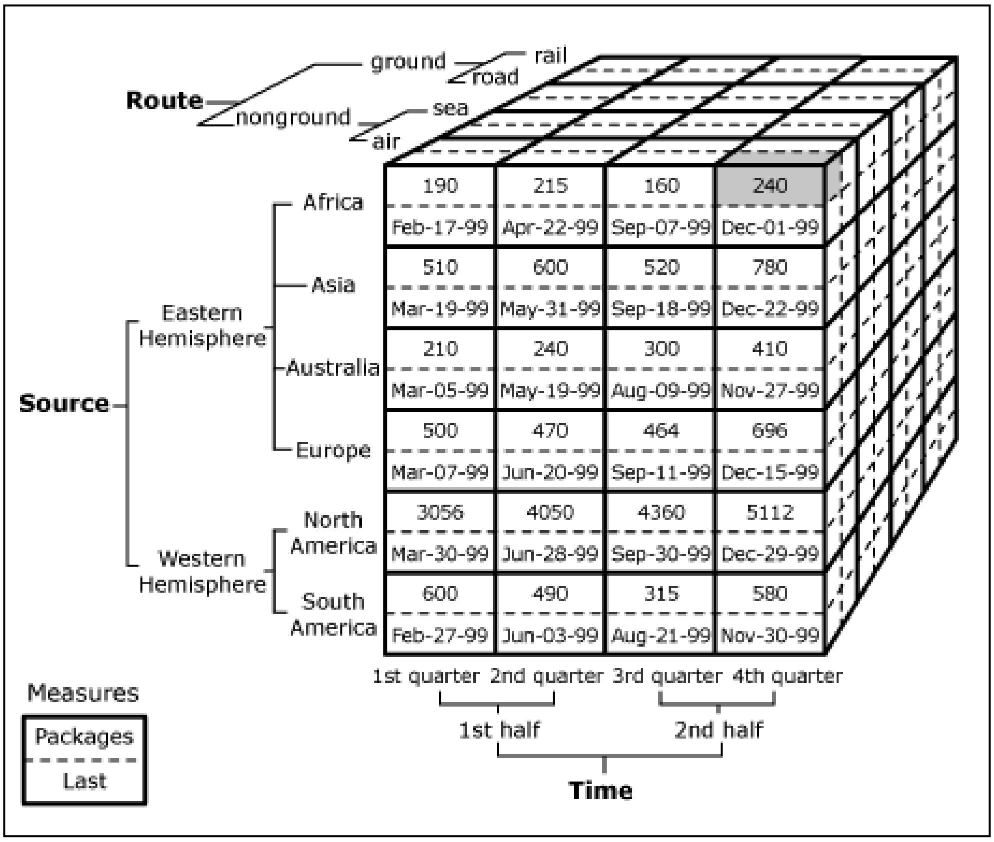


Figure 2-12: Slicing and Dicing of OLAP Cube

[Source: Kimball, R., Ross M., 2007]

Dicing and slicing are two major operations in OLAP cubes. Since OLAP cubes are large in volume, it is essential to perform dicing and slicing so that end-users can narrow down the usage of OLAP cubes. In Figure 2-11, dicing means a sub OLAP cube of the existing main cube. Dicing is similar to slicing. In slicing, filtering is done to focus on a specific attribute while Dicing is more a zoom feature that selects a subset over all the dimensions but for specific values of the dimension.

A research paper [72] has defined of basic OLAP operations Slice, Dice, Rollup and Drill Down [72]. Though slicing and dicing are the most common, there are other operations as shown in Figure 2-12.

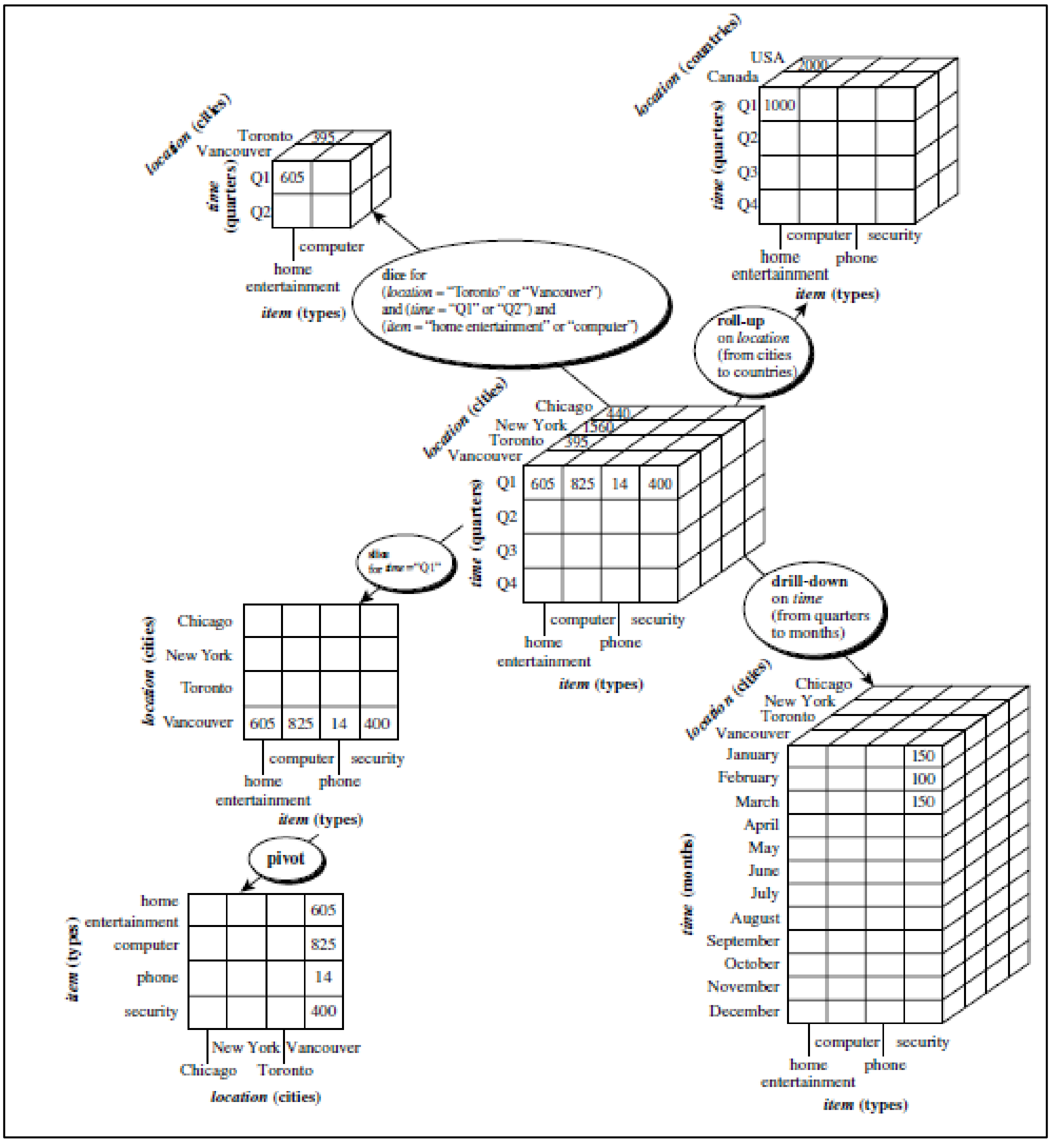


Figure 2-13: OLAP Operations in Multidimensional Cubes

[Source: Hand D., Mannila H., Smyth P.]

As shown in Figure 2-13, there are other OLAP cube operations drill-down, roll-up and pivot operations. When fuzzy techniques are implemented into the data warehouse, it was identified how these operations are viable for the fuzzy data warehouse as well, were identified.

OLAP cubes require the computation of various aggregates from large databases. In the case of data analyze, many aggregates will be needed frequently. Therefore, to improve the performance of these queries, it makes sense to store some of them. Mode of storage for OLAP cube is Lattice Structure which is presented in Figure 2-14.

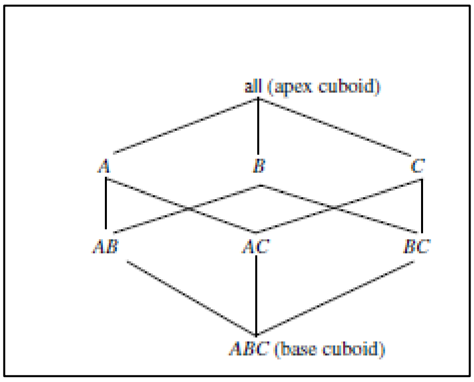


Figure 2-14: Lattice Structure of Cuboids

Full cube computation algorithms are important for many reasons such as accuracy and performance. OLAP cubes used to store data in a secondary storage in order to improve the data reading perofrmance and accessed when required for better manageability. Instead, algorithms can be used to compute smaller cubes, consisting of a subset of the given set of dimensions, or a smaller range of possible values for some of the dimensions. In these cases, the smaller cube is a full cube for the given subset of dimensions and/or dimension values. A thorough understanding of full OLAP cube computation methods will help to develop efficient methods for computing partial cubes. Hence, it is important to explore scalable methods for computing all the cuboids making up a data cube, which is called as full materialization.

The partial materialization of data cubes offers a fascinating trade-off between storage space and response time for OLAP cube. Instead of computing the full cube, it can be used to compute only a subset of the data cube's cuboids, or sub-cubes consisting of subsets of cells from the several cuboids.

OLAP cubes are heavily used in the industry to facilitate data analytics in data warehouse. Microsoft has released two tools, SQL Server Analysis Service Multi-dimensional version and tabular version where Oracle has released Hyperion cubes to provide their users to extend analytics capabilities to the data warehouses.

2.9 Fuzzy OLAP Cube

OLAP cube is used by many end-users of the data warehouse to analyze data. An OLAP cube is a multi-dimensional database that is optimized for data warehouse and OLAP applications. An OLAP cube is a method of storing data in a multi-dimensional form, mostly for reporting and analysis purposes. In OLAP cubes, measures are categorized by dimensions. Technical details of computations and structure of OLAP cube is discussed in Chapter 9. In this section, the research done on the fuzziness aspects of the OLAP cube is discussed. This research is to investigate and development of technology for the fuzzy data warehouse. Since an OLAP cube is part of the data warehouse, this section, current research areas of fuzzy aspects of OLAP cube is discussed.

In a research paper, titled Fuzzy OLAP Cube for Qualitative Analysis [59] has introduced a fuzzy OLAP cube supports qualitative study for the data warehouse. In this research paper, the main four OLAP operations namely: slicing, dicing, roll-up and drill-down are discussed on a fuzzy OLAP cube. This research has used Microsoft SQL Server Analysis Service 2000 as a tool to demonstrate fuzzy OLAP cube which is somewhat legacy tools with respect to OLAP tool since there are seven versions were released after the Microsoft SQL Server 2000. Though this research has provided methods for main OLAP operations, it has not discussed how measures are calculated for different aggregation functions like Average, Summation, Count, Distinct Count, Minimum, Maximum, etc.

Another research [41] has extended the above research and offered an approach based on multi-attribute summarization and fuzzy technology to provide semantics to the data present in the data warehouse. A fuzzy cube has been designed for foreign exchange transactions to perform the four common OLAP operations namely slicing, dicing, roll-up and drill-down in order to prove the present work. A major limitation of this and the previous research is that the discussion is limited for only one fuzzy attribute though, its theory discusses extending this to multi fuzzy dimensions.

2.10 Summary

This literature review was divided into a few areas for a better literature review. Since the research is mainly concerned about data warehouse, research papers in different domains which were published in the last few years were reviewed. The presence of several research papers related to the data warehouse shows that data warehouse is a topic still being carried out through data warehouse and is not an emerging technology.

A handful of researches have been done on Fuzzy databases to utilize fuzzy features into the databases. One of the promising researches has identified fuzzy data types such as Fuzzy Range, Approximate Value, and Linguistic label as relevant data types of interest to this research. Next important literature review area is fuzzy data warehouse implementation which is in the same research. However, it was noted that there are current research which are far from complete and they highlight the need for this research. Mainly, derivation of the fuzzy membership function is done in an ad-hoc manner as the data-driven method would be much effective. Also, fuzzy implementation can be done at various stages which can be database level, schema level or query level. Data warehouse technology plays a major role in this research. Different data warehouse theories such as Kimball and Inmon learning thoughts are identified in this section. Since ETL is the technique and is used to integrate data from source to data warehouse, different ETL techniques and concepts are identified. Fuzzy Logic is the supportive technology used for this research. Different Fuzzy Logic techniques like fuzzy logic membership function and fuzzy logic data types were identified as related areas to this research.

OLAP cubes are the major components in the data warehouse which is used to analyze data and slice, dice, roll-up and drill-down are four main operations in OLAP cubes used by the end-users to facilitate data analyzing.

A fuzzy OLAP cube is discussed in this chapter as related to four major OLAP operations such as slicing, dicing, roll-up and drill-down. In these tworesearches, there are two major limitations: Calculation of measures values for different aggregation and multiple fuzzy dimension implementations are not discussed.

The next chapter defines the research methodology of this research and are explained in detail. The next chapter will discuss about the sub-tasks of the research and discusses the data set that is used to validate the research findings.

Chapter 3
RESEARCH METHODOLOGY

3.1 Introduction

With the initial discussion regarding the objectives of this research, this chapter attempts to discuss in detail the development and design of fuzzy data warehouse. Towards this attempt, it uses the literature review and Feasibility Study, Sub Tasks, Technologies required for the research and required resources are discussed. A spider diagram is used to identify the research coverage so that all required elements are covered. The input-process-output technique is used as the research methodology. This research deals with a real world data set to verify the findings of this research [103].

Since this research, main objective is to introduce fuzzy logic based analytics for data warehouse. This indicates that it is the extension of existing features of data warehouse to support fuzzy logic analytics. L. Sapir research paper on, A methodology for the design of a fuzzy data warehouse [102] has used the constructive design methodology. Therefore, this research is following the constructive design methodology in extending the fuzzy logic into data warehouse.

3.2 Constructive Design Research

In this research, constructive design research methodology is used. In the constructive design research, new knowledge is developed through constructing actual design artifacts such as systems, products and spaces. In this method, it addresses limitations of earlier approaches and this will open up the imaginary visions.

3.3 Input-Process-Output

The detailed study area and the related areas of data warehouses are identified by spider diagram as shown in Figure 3-1.

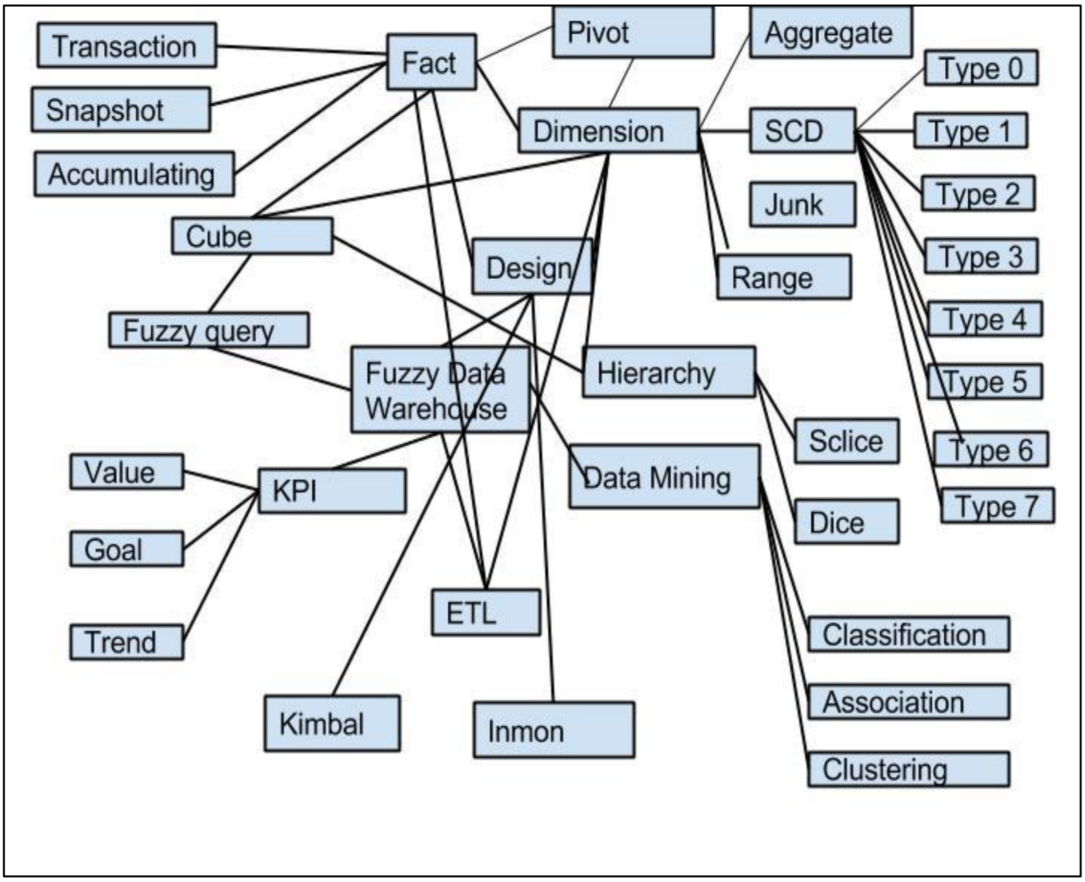


Figure 3-1: Spider Diagram for Fuzzy Data Warehouse

In the spider diagram presented in Figure 3-1, though all the entire area is covered, the data mining area is excluded from this research, for more focus on data warehousing. In this research, fact and dimension designing with respect to the fuzzy data warehouse is discussed in detail. Different types of Fact and Dimensions are discussed in Appendix A: and Appendix B: for further details. There are two design strategies used in the industry. Kimball [44], [45] and Inmon [39] are the major designing patterns which are used in the industry. Aggregation, Hierarchy, Slicing, Dicing, Pivot operations [91] are also included in addition to a detailed discussion regarding OLAP cube designing.

The input-process-output method was used as the methodology for this research. As shown in Figure 3-2 different types were also identified in the Input-Process-Output framework.

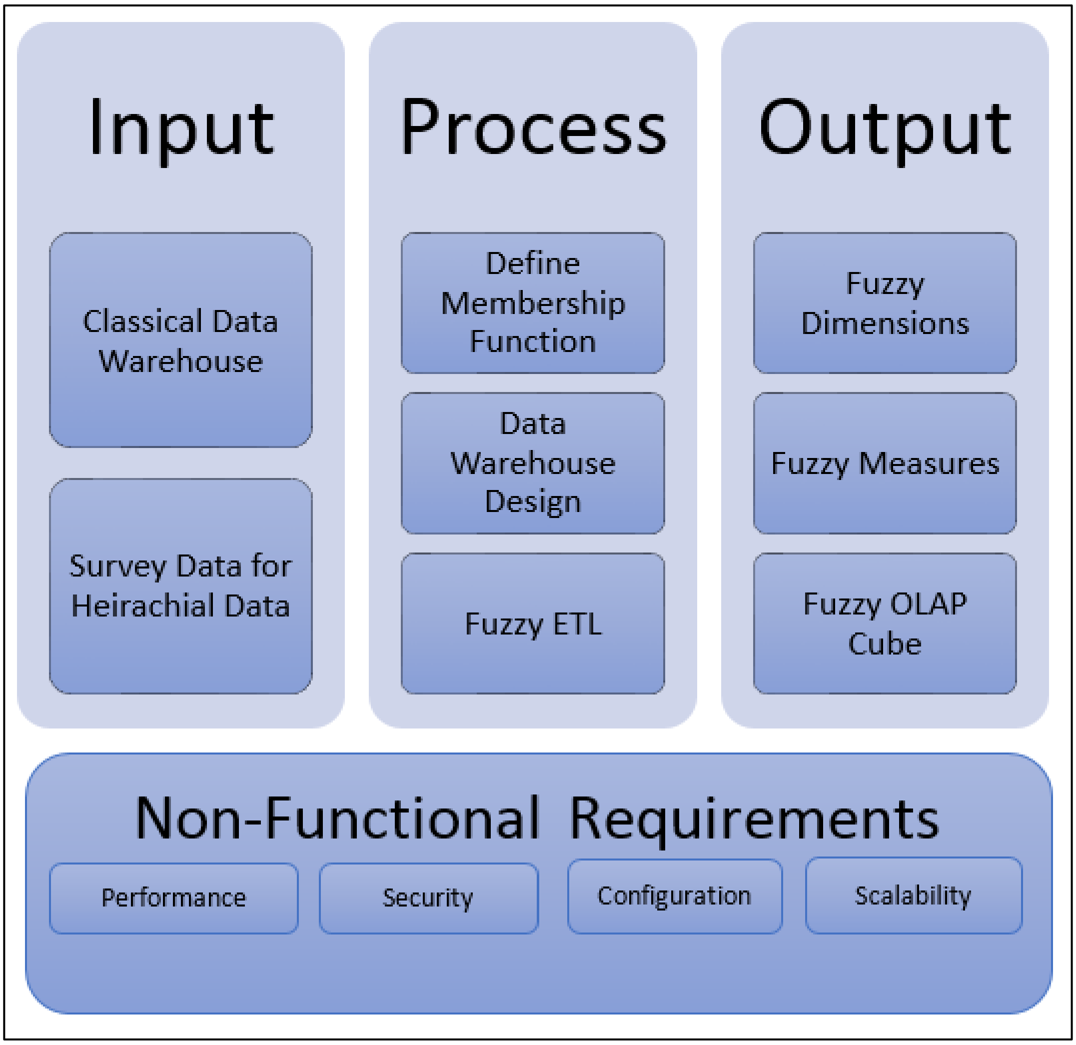


Figure 3-2: Input-Process-Output for Fuzzy Data Warehouse

In this research, classical data warehouse and survey data are used as input for the research. In the process part of this research, the important area in this research, fuzzy membership function design and data warehouse design are discussed in detail since they are the key factors in this research. ETL is also made a key component in this research as it needs to support fuzzy data warehouse structures. Fuzzy dimension and fuzzy measures are the key output of this research. Fuzzy OLAP cube is also discussed in this research.

Non-Functional Requirements are also a key aspect of this research, as research is not feasible if non-functional requirements are not considered and not achieved.

Non-functional requirements should be treated across all the boundaries of input-process-output. Security, performance, configuration and validations are considered with respect to non-functional requirements.

3.4 Sub-Tasks

To achieve the research objective, the entire research is divided into multiple subtasks. These sub taskes can be called as milestones in this research. Sub tasks for the research objectives are presented in Table 3-1.

Table 3-1: Sub Tasks for Research Objectives

Sub Task	**Comments**
Review research paper on fuzzy data warehouse	This identifies the current research areas and areas of implementation.
Feasibility Study	The feasibility study is accomplished to verify the viability of implementing a fuzzy data warehouse.
Define Fuzzy Membership function	The methodology is defined as the derivation of the fuzzy membership function.
Design methodology for dimensions and fact tables	This covers security, performance and enhancement, the configurable ability for new design method.
ETL for fuzzy dimensions	This supports type 2 and type 3 slowly changing dimensions as those are the most used dimensions in the industry today.
Data Cubes	Design a Data Cube to support fuzzy data warehousing.
Slice / Dice / Roll-Up / Drill-Down	A new technique for hierarchies and its slicing and dicing capabilities.
KPI, Query Tool	Related technology to demonstrate the capabilities of fuzzy data warehousing.

Each step was verified using an appropriate sufficient real-world data set as a proof of concept as discussed before. When the major milestones were achieved, the research papers were submitted and the submitted research papers are listed in the Appendix AA and their citations are listed in the Appendix BB. All the related sub-components related to the classical data warehouse is are identified and displayed as a spider diagram in Appendix K:

3.5 Data Set

It is essential to prove that established fuzzy data warehousing concepts can be implemented in the real world. As for proof of concepts, the above tasks must be implemented with real-world data. However, as identified in the literature review section, most of the research is confined to sample data which are well defined and cleaned or research is limited to the concepts [63],[103]. This is far away from the real world scenarios, the real world requires users to engage with heterogeneous and uncleansed data models. However, to find out whether fuzzy data warehouse can be implemented in the real world or the current business organizations, it was decided that a real-world data set would be used. It was decided to use Sri Lankan plantation attendance system. Since, this is a legacy system data is in Dbase3+/Clipper format and data was extracted to Microsoft SQL Server 2016 database using SQL Server Integration Services of Microsoft SQL Server.

Four years (2012-2015) of data is selected for the research and used. Though the data is outdated as the data set is more than four years old, this research does not need latest data. This research is to implement fuzzy logic based analytics for the data warehouse. Since this research does not focusing on extracting data pattern, latest data set is not need. In this research, data set should be more diverse.

For the selected four years, 2.6 million records of transactions records were selected. Figure 3-3 demonstrates the distribution of transactions for each year.

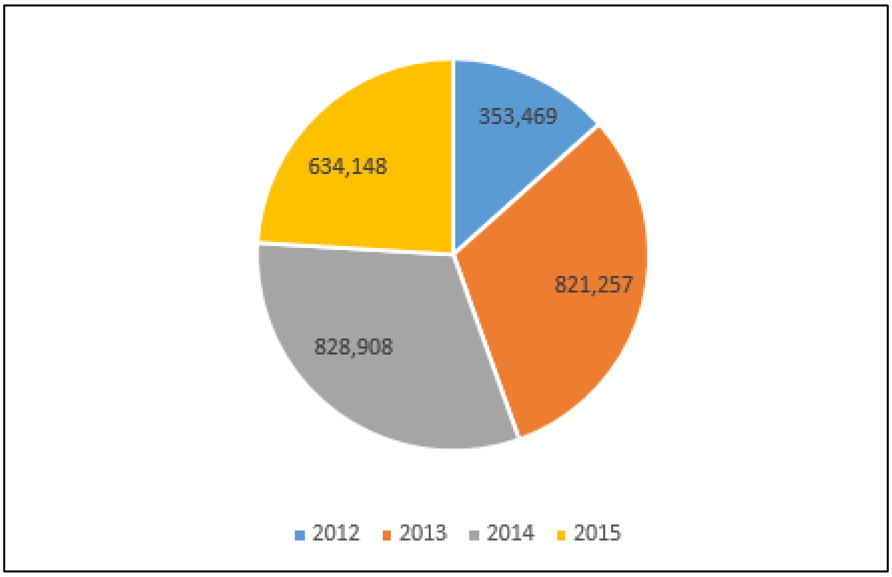


Figure 3-3: Distribution of Transactions Year Wise

In the above transaction distribution, 2013 and 2014 years show a similar number of records while 2012 and 2015 show a smaller number of records due to many reasons such as labour union actions and weather impacts.

In a plantation, employees are managed by divisions to micro manage unskilled, uneducated employees. Typically, there are five to ten divisions to improve the management of the plantation. The selected plantation organization has five divisions and transactions are distributed as shown in Figure 3-4.

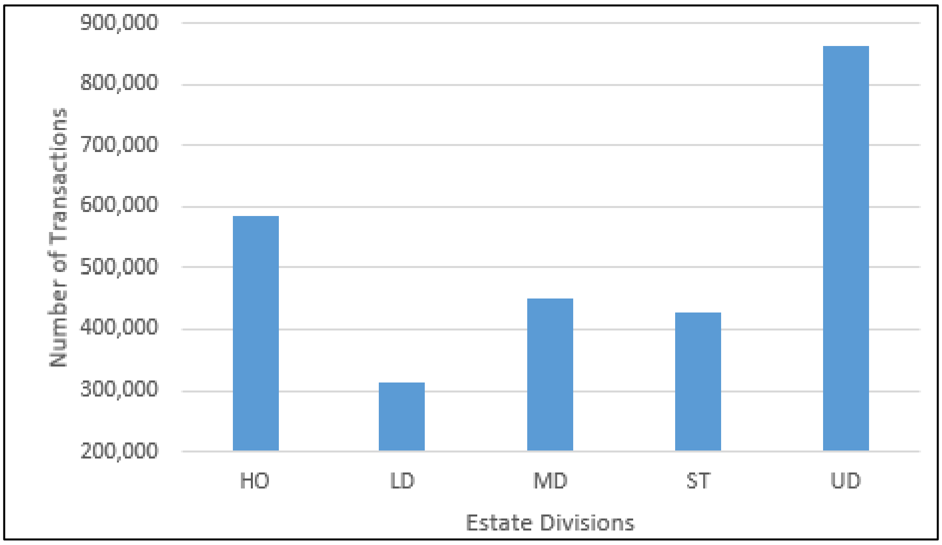


Figure 3-4: Distribution of Transactions Division Wise

HO, LD, MD, ST, and UD are divisions of the selected plantation. As shown in Figure 3-4, UD division has the highest number of transactions. Typically, transactions are not equally distributed mainly due to the fact that some divisions have a higher profitability than others. In addition, these divisions have different acreages and different numbers of bushes. Depending on the nature of the division such as soil type etc, the numbers of employees are different from division to division.

Employees on the plantation are assigned to different types of work. For example, in the case of tea, plucking is the main activity for the plantation. Figure 3-5 shows the distribution of transactions for different types of workers.

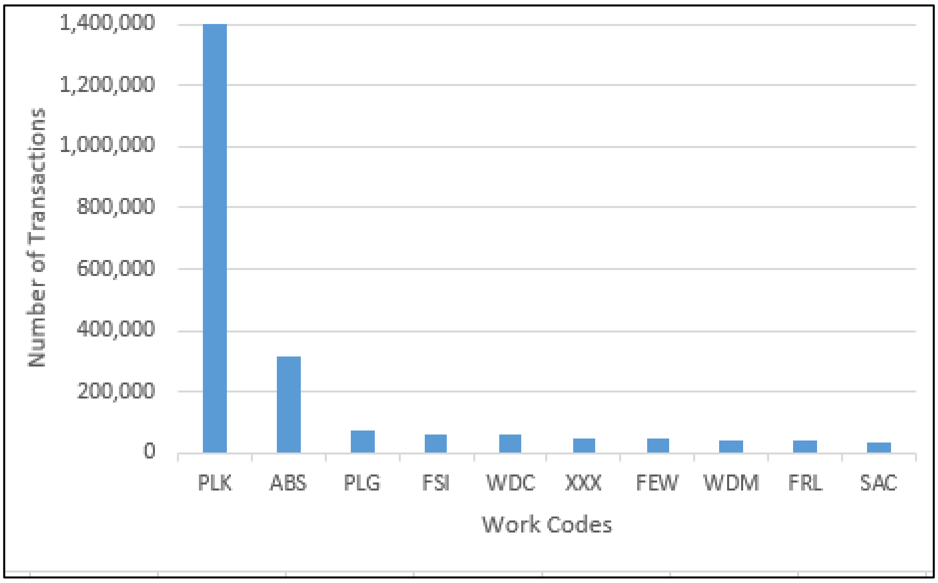


Figure 3-5: Distribution of Transactions Work Code Wise

It is very clear from Figure 3-5 that the main activity of the plantation is plucking. Accordingly, more than 50% of transactions fall into this category of activities. Other work categories are fertilising, fertilizing, tipping and pruning etc.

The employee is an important dimension in the selected data set. The following Figure 4-6 shows the gender distribution of employees in the selected plantation.

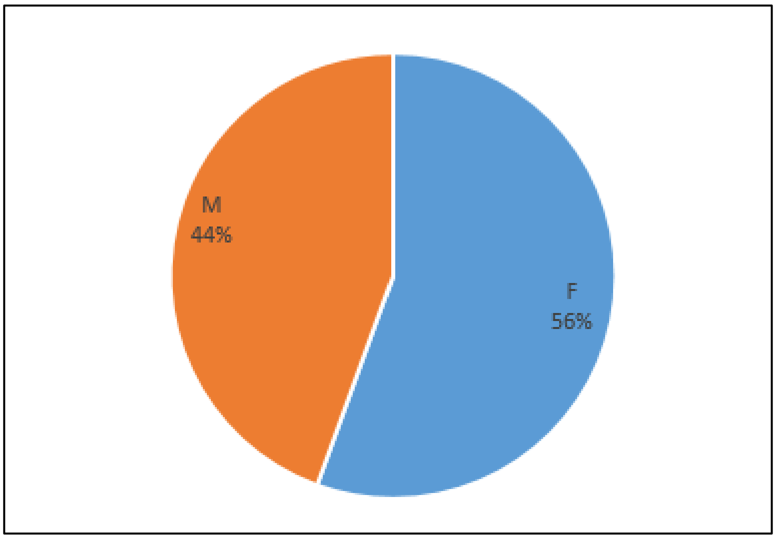


Figure 3-6: Employee Distribution for Gender

Typically, female employees are higher in the plantation section especially connected to the tea crop, a fact confirmed with Figure 3-6.

Since data warehouse works with a large volume of data, it is essential to find out the performance impact of data warehouse after fuzzy techniques are introduced. To verify data warehouse performance which is considered to be a non-functional requirement as identified in the objectives, the data set should be substantial in volume with different dimensions as shown in the above figures. Therefore, the data set was selected for four years.

3.6 Resources Required

For this research, no high scale computers were required. Research objectives were achieved with commodity servers. As data warehouse required storage engine relational database management system was needed. Microsoft SQL Server 2016 Developer edition [88] was used as the storage engine for this research. Microsoft SQL Server is one of the most common relational database management systems as stated in Appendix K. Also, Microsoft SQL Server has rich capabilities of business intelligence and data warehouse capabilities. It is important to note that Microsoft SQL Server 2016 developer edition is free hence no funds were required for the storage tool. Apart from storage, in SQL Server 2016, SQL Server Integration Services (SSIS), and SQL Server Analysis Services (SSAS) are used as ETL tool and OLAP cube respectively.

The data set is required for the proof of concept and is taken from a Sri Lankan plantation organization, in operation since 1980. For the analysis purposes, R [89] and Matlab [90] tools were used. As listed in Appendix J: R is one of the common tools used in the industry and in an academic arena mainly for statistical analysis.

3.7 Summary

Input-Process-Output technique is used as the methodology for this research. Traditional data warehouse is the input whereas output is the fuzzy data warehouse. To implement fuzzy data warehouse design strategy for fuzzy data warehouse was identified. As has been highlighted in many places it is essential to define data driven mechanism to implement fuzzy membership function, a feature lacking in many other relevant researches. It was also identified that the diverse types of fuzzy membership functions such as derived and linguistic can be implemented to achieve better results. Non-Functional Requirements are also identified as Performance, Security, Configuration and Validation. These aspects are discussed in all the areas of Input-Process-Output.

Subs tasks are identified in the research methodology by means of spider chart and to evaluate the feasibility of fuzzy data warehouse implementations, plantation attendance data sets with about four years' data was selected. A feasibility study was done in various business domains to evaluate whether the fuzzy data warehouse is viable.

Fuzzy Logic and data warehouse are the identified techniques for the research and are further evaluated during the research. ETL and OLAP are other technologies used in this research.

Fuzzy data warehouse is an extension to the existing data warehouse, implemented in the many organizations. Though the theoretical frame work can be built to define fuzzy data warehouse, it is essential to identify whether this is possible to implement in the real-world. Therefore, the next chapter is dedicated to find out possible implementation real-world scenarios and how to exploit the fuzzy data warehouse in current classical data warehouse implementation.

Chapter 4

FEASIBILITY STUDY FOR FUZZY DATA WAREHOURE IMPLEMENTATION

4.1 Introduction

Fuzzy techniques can be implemented into the data warehouse. Yet, the feasibility of implementation needs to be identified so that fuzzy data warehouse can be implemented in the real world. The feasibility study helps understand, how fuzzy techniques can be extended to the data warehouse and possible domains. Since there are many sub-areas of the data warehouse, it needs to understand to what extent fuzzy data warehouse can be utilized. For example, it can be a fact table or dimension or at the ETL or an OLAP Cube to facilitate fuzzy data warehouse.

This chapter discusses domains in which fuzzy data warehouse can be implemented and this chapter discusses to what extent it can be utilized using second and third editions of Kimbal's books on data warehouse toolkit. These books discuss possible implementations in different domains.

4.2 Data Warehouse Domains

As discussed in the Chapter 2, Data warehouse is utilized to descriptive analytics, diagnostic analytics as well as predictive analytics. It is essential to identify the in what domains that Fuzzy logic base analytics for data warehousing can be implemented.

The Data Warehouse Toolkit [44], [45] is one of the main sources for several research articles. This toolkit has discussed several real-world cases in detail. Since the second and third editions of Kimbal's books have focused on different business domains, these books are used as references for possible implementations of data warehouses. In these books, Retail Sales, Inventory Management, Purchasing, Order Management, Customer Relationship Management, Financial Accounting, Human Resources Management, Financial Services, Telecommunication, Transportation, Education, Health Care, Insurance sectors are discussed in detail. Perusing through the cases described in these books, the feasibility of fuzzy data warehouse implementation is analysed.

4.3 Retail Sales

Retail Sales are one of the most common business scenarios for many technologies and are not limited to data warehouses. Retail sales and wholesale sales has shown a high penetration in the data warehouse when compared to the other domains of implementations in the data warehouse.

The data warehouse is used to enhance business value not only for Retail Sales but also for many other data warehouses. In most data warehouse business processes date dimension is a supplement towards better analysis. Date dimension is used as a role-playing dimension where there is a lot of date in fact tables. Appendix C: describes the implementation of date dimension

A survey on a small scale grocery house with Date Dimension of a Retail Sales, Selling Season column is set to the name of the retaining season. Seasons can be one of, Thanksgiving, Easter, Valentine's Day, Christmas, Wesak, Awurudu etc. Since the season has a long date range, sales will be different, when it is closer to the season date. In case of holidays/weekend, sales will be in a different order. Normally, seasons start in advance. For example, though the Christmas day falls on December 25 of every year, retail sales will start around end of November. Then sales will increase gradually towards Christmas day and weightage will be different if it is a weekend. Hence, the seasonal factor will be different from time to time. The factor should be derived from using fuzzy techniques.

In the promotion dimension of the retail domain, promotion impact is measured as well. Proposed design from Kimball is to measure the impact. However, there are different ways of calculating the promotion impact as listed in Appendix E. Due to the different types of calculation, promotion impact can be a fuzzy dimension attribute.

Experience is a key attribute in clerk dimension where it will be used to find sales depending on the experience of the clerk. Experience is stored as a number of years. However, the experience can be categorized as high, medium and low which can be treated as a fuzzy attribute. Also, linguistic fuzzy attributes can be defined to perform more analytics like “very old”, “not young AND old”, etc.

In the product dimension, products are typically categorized to subcategories, categories and groups. These groups are more likely to be a crisp group. However, in reality, one product called fuzzy categorization.

With respect to measures in Retail Sales, there are measures like Amount, Cost, Discount, Tax, etc. In a typical analysis, these measures are analyzed in ranges and these ranges are labeled as High, Medium and Low. Since ranges are one of the best cases for fuzzy analysis, fuzzy can be utilized in retail sales domain to assert ranges in retail sales.

This shows that in retail sales there are different options and places to implement fuzzy data warehouse techniques such as fuzzy categorization, linguistic analytics and fuzzy ranges.

4.4 Inventory

Every inventory transaction identifies the date, product, warehouse, vendor, transaction type, etc., the dimensions in the data warehouse implementation. In most cases, the transaction impacts a single amount representing the inventory quality. Star schema for the inventory is shown in Figure 4-1 and the Warehouse Inventory Transaction Fact table is maintaining a granularity of one row per inventory transaction.

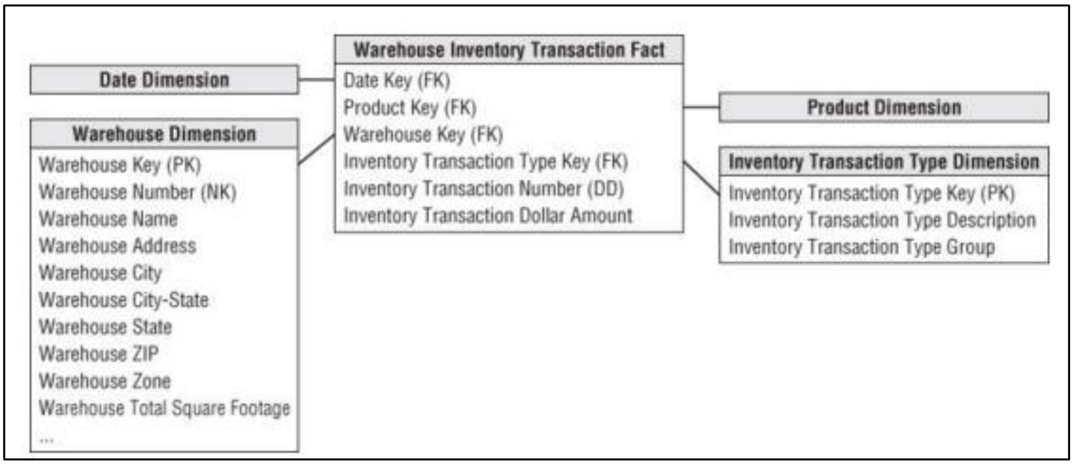


Figure 4-1: Star Schema Design for Inventory

[Source: Kimball, R., Ross M., 2007]

In the model shown in Figure 4-1, Warehouse Total Square Footage can be treated as fuzzy attributes so that the store warehouse size can be analyzed as the size of the warehouse is an important attribute when deciding the storage of the goods.

In Figure 4-2, Data Model for Inventory Receipt Accumulated Fact Table is shown along with the relevant dimension tables.

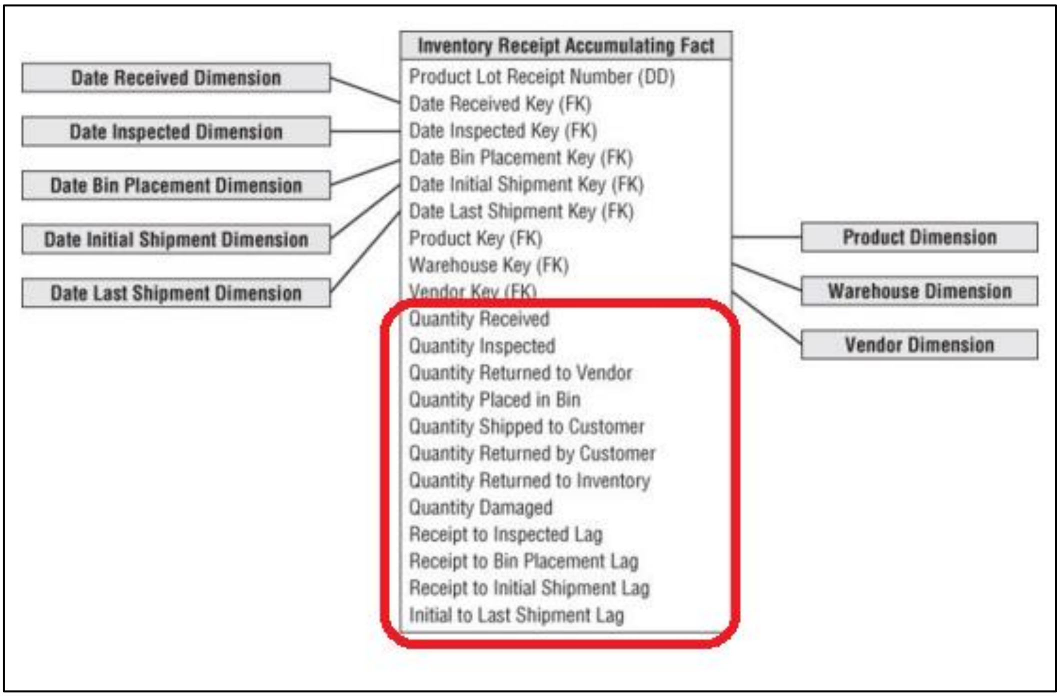


Figure 4-2: Data Model for Inventory Receipt Accumulating Fact Table

[Source: Kimball, R., Ross M., 2007]

In this fact table, there are inventory related measure columns that are highlighted in Figure 4-2. They need to be analyzed for comparison. When these measures are analyzed, they will be set to a crisp set. For example, lag measures are important to analyze delays between stages. Typically, these lags are measured via crisp values such as high, medium and low. These can be converted to fuzzy attributes so that precious analysis can be done.

Apart from measure columns, in the product dimension, there are some products which fall into multiple product categories. For example, some products fall under Appliance and Apparel. This means that there is an uncertainty which needs to be handled by means of fuzzy techniques. In this scenario, products are categorized into multiple product categories with different weightages. Fuzzy techniques need to be designed to calculate different weightages. These weightages may be different, depending on the season and business which will make the calculation of weightages are complex.

4.5 Order Management

Order management is a common business scenario for data warehousing as almost every business needs order management. In the case of order management, data warehouse, customer dimension contains an important attribute called Customer Credit Rating which will define how good customer's credit rating is. This is again a crisp value set which can be configured as a fuzzy attribute so that credit rating can be measured to higher accuracy. Similar to the promotion dimension in the retail sales warehouse, there is a dimension named Deal in order management. Therefore, like the promotion dimension, the deal dimension can be converted to fuzzy dimension.

Customer Satisfaction is an important matrix in Order Management. Business users are often concerned in customer satisfaction metrics such as on time delivery, damage free delivery and complete delivery etc. These are added to the fact table for each line-item level as satisfaction metrics. These new fact columns are populated with additive ones and zeros. By using these three fact measures, customer satisfaction junk dimensions were created. All three of these parameters can be fuzzy attributes and by implementing rules and De-fuzzification techniques customer Satisfaction attribute can be derived.

4.6 Customer Relationship Management

In Customer Relation Management (CRM), customer segmentation plays a huge role, so that a better relationship with a customer can be maintained. Some of the most influential attributes in a customer dimension are segmentation classification or scores. These attributes are generated by business context. Gender, Ethnicity, Age, Income, Status, Referring Source are the important attributes among many other attributes when defining customer segmentation. However, since Age and Income can be treated as fuzzy attributes, customer segmentation will become fuzzy attributes. Like in the Customer Order Management, by defining rules and de-fuzzification techniques, the function of customer segmentation can be done to implement fuzzy techniques.

County Demographics Outrigger dimension can be introduced to identify different county-level attributes as shown in Figure 4-3.

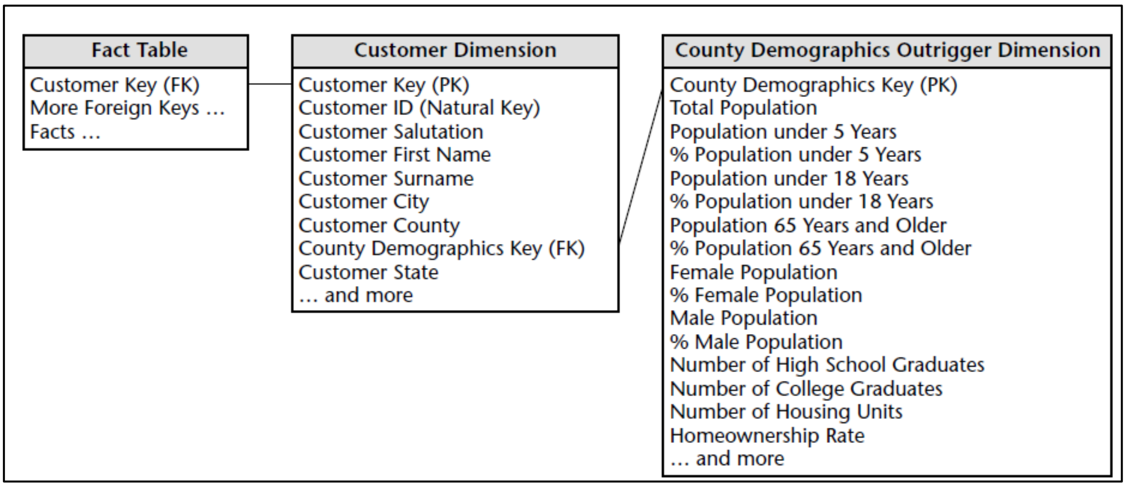


Figure 4-3: County Demographics Outrigger Dimension

[Source: Kimball, R., Ross M., 2007]

In the County Demographics Outrigger dimension table, there are continuous measures which can be converted to bucketized measures and fuzzy analysis can be done on those measures for better analysis.

Table 4-1 shows different ranges for age and income level for analysis. In classical data warehouse analysis, continuous variable classified into ranges is called bucketized in data warehousing so that better analysis can be done.

Table 4-1: Range Dimension for Age and Income

Demographic Key	Age	Gender	Income Level
1	20 – 24	Male	< $ 20,000
2	20 – 24	Male	$20,000 - $ 24,999
3	20 – 24	Male	$25,000 - $ 29,999
18	25 – 29	Male	$20,000 - $ 24,999

Both age and Income level columns are fuzzy columns as they contained data in the form of ranges as shown in the Table 4-1. Customers can be categorized using Age, Gender and Income Level. Customer Categorization too can be a fuzzy variable since Age and Income Level can be treated as fuzzy variables. In this scenario, customer categorization is a derived membership function of Age and Income Level and de-fuzzification techniques can be utilized to find the customer category. This can be represented in a de-fuzzification model as presented in Figure 4-4.

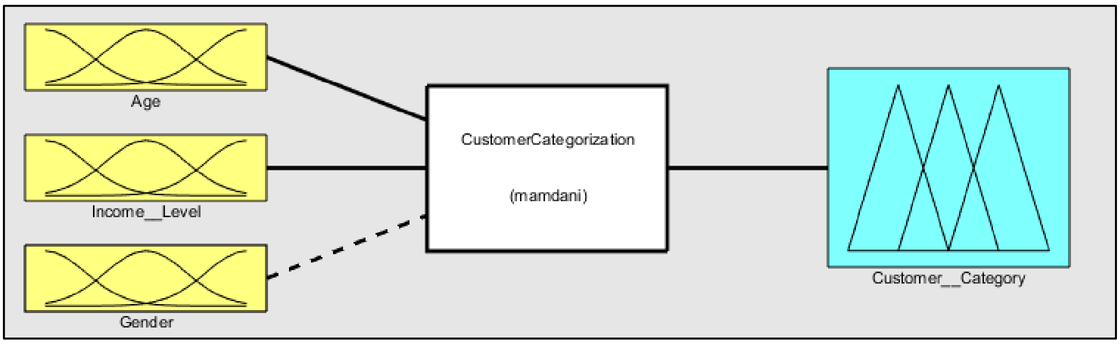


Figure 4-4: Identify Customer Relationship using Mamdani de-fuzzification Method.

In the above model Mamdani de-fuzzification method is used for de-fuzzification whereas the Gender showed in a dotted line to indicate that it is not a fuzzy variable.

In a CRM system, customer behavior is an important key performance indicator to measure in which the design is displayed in Figure 4-5.

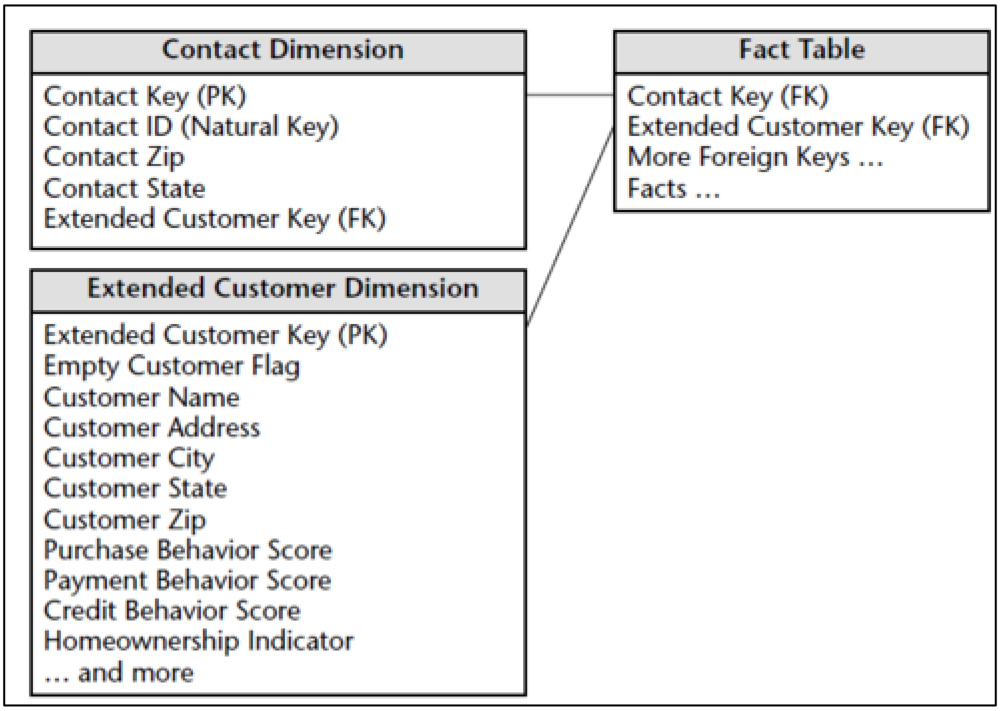


Figure 4-5: Extended Customer Dimension

[Source: Kimball, R., Ross M., 2007]

In this data model, Purchase Behavior Score, Payment Behavior Score, Credit Behavior Score can be considered as KPIs which are candidates for fuzzy attributes. Rather than analyzing these in bucketized or as range values, by introducing fuzzy attributes, better analysis can be done.

4.7 Human Resource Management

Human Resource Management (HRM) is a key area to compete with other organizational systems. In most organizations, the Human Resource Management system is the heart of the organization as HRM caters to the entire organization's business operations.

Skills of employees are stored and analyzed in an HRM system. Since each employee will have a variable and an unpredictable number of skills, the skill dimension is a chief candidate to be multi-valued dimension keywords by their nature, usually, are open-ended. New keywords are created as skills as presented in Figure 4-6.

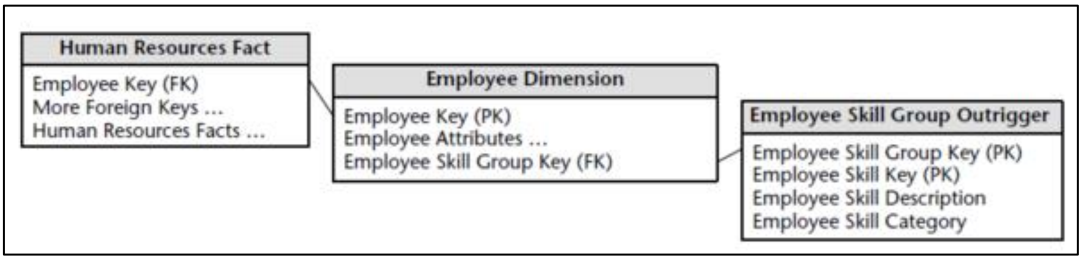


Figure 4-6: Employee Skill Group Dimension

[Source: Kimball, R., Ross M., 2007]

Since an employee can have multiple skills, there is an uncertainty about the skill levels. Also, skills can have weightage for every employee since different employees have different levels of skills. Therefore, skill is an excellent parameter to consider as a fuzzy attribute. There is a skill category which can be another fuzzy attribute since one skill can be a fall into multiple skill categories with different weightages. Apart from skills, there is another important fact table named headcount which is a snapshot fact table in the HRM system which is shown in Figure 4-7.

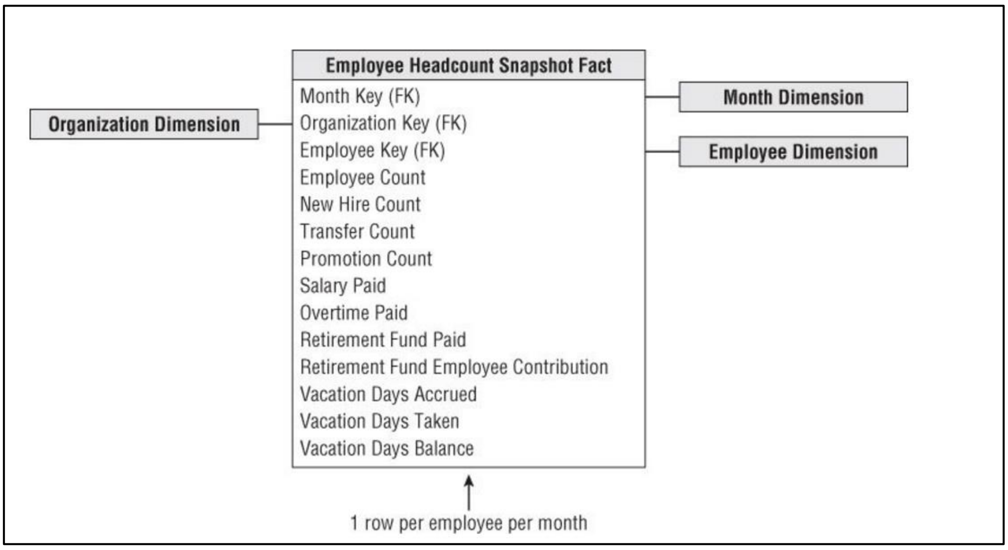


Figure 4-7: Data Model for Employee Headcount Snapshot Fact Table

[Source: Kimball, R., Ross M., 2007]

In the fact table shown in Figure 4-7, most of the measures count for a given month for one employee. When it comes to analysis, typical range values will be used. However, the better option would be changing the ranges to the fuzzy dimensions for better analysis. Also, Skills and Skill group are the better places to implement linguistic analysis where fuzzy techniques can be utilized.

4.8 Financial Services

Financial service is one of the main sectors to benefit from the data warehouse technology over the years as it can provide variety ranges of analysis from the data warehouse. The data model is shown in Figure 4-8, an example for the finance service which maintains customer account balances in a bank.

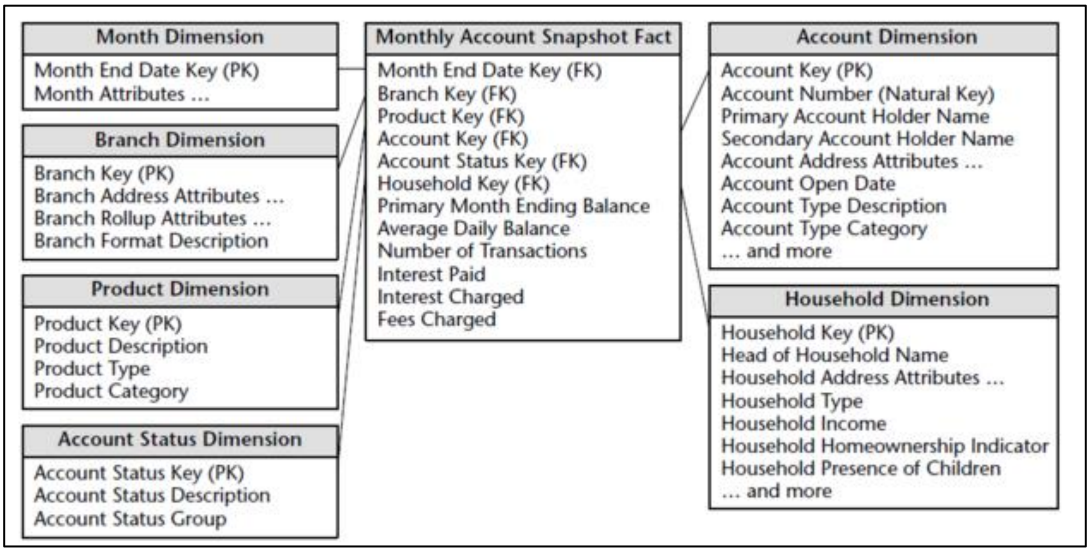


Figure 4-8: Data Model for Monthly Account Snapshot Fact Table

[Source: Kimball, R., Ross M., 2007]

Household Income will be a candidate for the fuzzy attribute as it can be analyzed as low, medium and high which is crisp variable in the classical data warehouse analysis. Month Ending Balance, Interest Paid, Charged, and the number of transaction measures can be converted to range dimensions for better analysis. Therefore, those columns are candidates for fuzzy attributes.

Apart from the bank account balance, there are other important measures in the bank such as the number of deposits, ATM usages, etc. which the bank management will be interested in. Those fact tables are shown in the data model diagram shown in Figure 4-9.

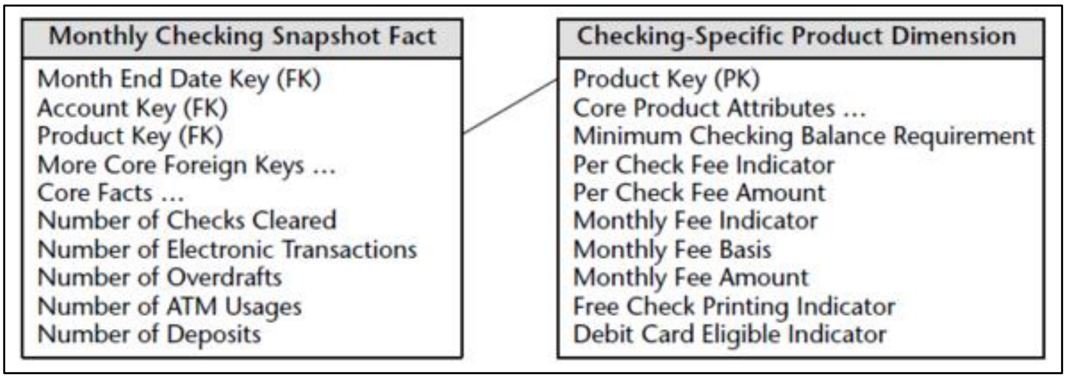


Figure 4-9: Data Model for Monthly Checking Snapshot Fact Table

[Source: Kimball, R., Ross M., 2007]

Those measures are better off when analyzing with ranges and as mentioned in multiple times range values are better candidates for fuzzy attributes. Nature of Financial Service allows defining derived fuzzy membership functions. Also, by using linguistics analytics, better analysis can be done in the fuzzy data warehouse.

4.9 Healthcare

Healthcare is one of the most popular domains where the data warehouse is implemented. Also, it has the potential to grow in the data warehouse implementation as well. In healthcare, there can be multiple diagnoses currently handled by a weighting factor according to Ralph Kimball as presented in Figure 4-10.

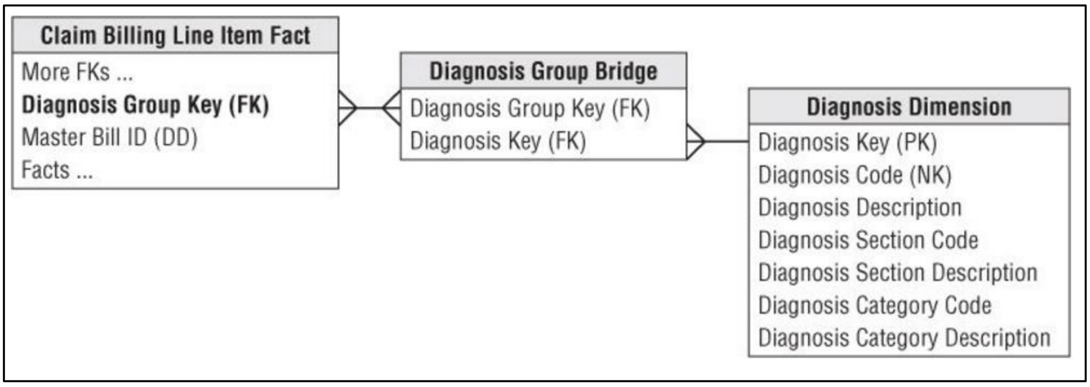


Figure 4-10: Data Model for Diagnosis Fact Table

[Source: Kimball, R., Ross M., 2007]

If a patient is diagnosed with multiple illnesses, the patient is allocated to a diagnosis group with multiple corresponding rows in the interconnected table as presented in Figure 4-11.

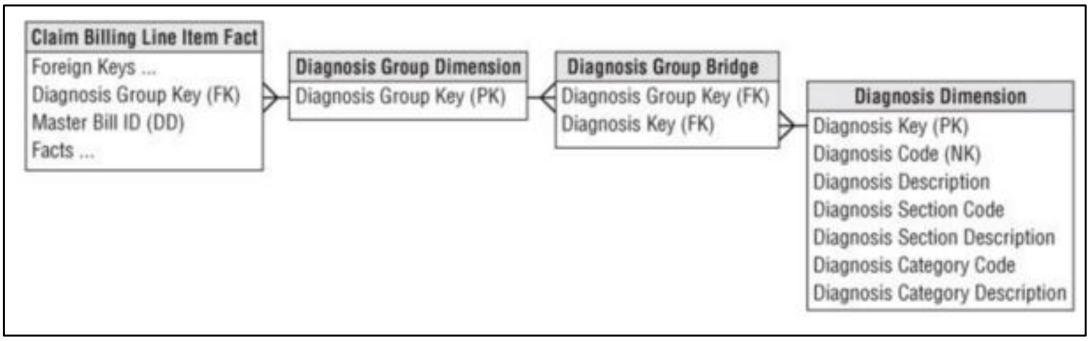


Figure 4-11: Data Model for Diagnosis Group Dimension Table

[Source: Kimball, R., Ross M., 2007]

For example, if a patient is diagnosed with three illnesses, there will be three rows in the bridge table. However, in the real-world scenario, it is difficult to manage this with the crisp values as illness will have different weightages. Better analysis and predictions can be done if it is converted into fuzzy data warehouse techniques. In the case of health care, linguistic analysis can be done specially for diagnosis where (HIGH temperature AND LOW blood pressure) is an example of linguistic analysis. Further, there are few number of classifications are done in the health care. Most of those classifications are crisp classification where one-to-many relationship is maintained from disease category to the disease. However, in the real world it is many-to-many with different weightages which can be better handled by fuzzy techniques. For example, if a patient is diagnosed with three illnesses, there will be three rows in the bridge table. However, in the real-world scenario, it is difficult to manage this with the crisp values as illness will have different weightages. Better analysis and predictions can be done if it is converted into fuzzy data warehouse techniques. In the case of health care, linguistic analysis can be done specially for diagnosis where (HIGH temperature AND LOW blood pressure) is an example of linguistic analysis. Also, there are few number of classifications are done in the health care. Most of those classifications are crisp classification where one-to-many relationship is maintained from disease category to the disease. However, in the real world it is many-to-many with different weightages which can be better handled by fuzzy techniques.

4.10 Summary

This chapter has done a feasibility study in what domains fuzzy techniques can be implemented in the data warehouse. Retail Sales, Inventory Management, Order Management, Customer Relationship Management, Human Resource Management, Financial Services, Health Care case studies from various research papers and main reference Kimball Ralph book were identified. In many cases, range dimensions can be treated as fuzzy data warehouse attributes.

Weighted contributions such as shown in the health care example can be considered as fuzzy implementation. It was noticed that in most of the categorization, one-to-many relationship is maintained mainly due to the fact that current data warehousing techniques are limited to crisp analysis. However, it was identified that if the fuzzy techniques are used, these traditional classifications can be extended to the fuzzy categorization or the fuzzy classification.

Linguistic analysis is another technique which can be implemented by using fuzzy techniques. By introducing linguistics, users will have the option of analyzing data warehouse to next level. Also, by defining the rules, de-fuzzification can be implemented in the data warehouse for better analysis. Also, there are many places where derived fuzzy membership function can be implemented by using multiple membership functions.

As discovered at the literature review, the main draw back in the current fuzzy data warehousing technique is the lack of a data driven method for fuzzy membership functions. Also, there is a requirement for different types of fuzzy membership function such as derived, and linguistic fuzzy membership function. In the next chapter, design methodologies for fuzzy membership functions are described to address the gaps in the current research areas. This chapter is about a feasibility study in what domains fuzzy techniques can be implemented in the data warehouse. Retail Sales, Inventory, Order Management, Customer Relationship Management, Human Resource Management, Financial Services, Health Care cases studies from various research papers and main reference Kimball Ralph book were identified. In many cases, range dimensions were treated as fuzzy data warehouse attributes.

Weighted contributions such as shown in the health care example can be considered as fuzzy implementation. It was noticed that in most of the categorization, one-to-many relationship is maintained mainly due to the fact that current data warehousing techniques are limited to crisp analysis. However, it was identified that if the fuzzy techniques are used, these traditional classifications can be extended to the fuzzy categorization or the fuzzy classification.

Linguistic analysis is another technique which can be implemented by using fuzzy techniques. By introducing linguistics, users will have the option of analyzing data warehouse to next level. Defining the rules, de-fuzzification can be implemented in the data warehouse for better analysis. There are many places where derived fuzzy membership function can be implemented by using multiple membership functions. As discovered at the literature review, the main drawback in the current fuzzy data warehousing technique is the absence of a data driven method for fuzzy membership functions, where there is a requirement for several types of fuzzy membership function such as derived, and linguistic fuzzy membership function.

In the next chapter, the design methodologies for fuzzy membership functions are described to address the gaps in the current research areas.

Chapter 5
FUZZY MEMBERSHIP FUNCTIONS

5.1 Introduction

In Chapter 2 Fuzzy Data Warehouses – State of Art, a detailed analysis of current research, claims the lack of research regarding data-driven methodology to determine fuzzy membership functions, especially in the fuzzy data warehouse in addition to other technical domains. Since fuzzy membership function plays main roles for the success of the fuzzy data warehouse, it is very much essential to derive fuzzy membership in a data drive method opposite to the subjective method, though there is a need for subjective method in some cases.

Most research defined fuzzy membership function as an ad-hoc method. In this research, different methods are used to define fuzzy membership function as fuzzy membership function plays a vital role in the fuzzy data warehouse. There are several types membership functions such as derived fuzzy membership functions which are a combinations of multiple fuzzy membership functions. Equally, there are linguistic fuzzy membership functions which can be utilized to perform linguistic analytics in the fuzzy data warehouse.

5.2 Box Plot Method

Box Plot is typically used to show distributional shapes and to detect unusual observations in a data set. The Box Plot also known as box and whisker diagram, is a regular way of presenting the distribution of data based on the five-number summary: minimum, first quartile, median, third quartile, and maximum. This is also referred to as five number properties of a data set. Diagram of a Box Plot is presented in Figure 5-1 along with the relevant box plot parameters.

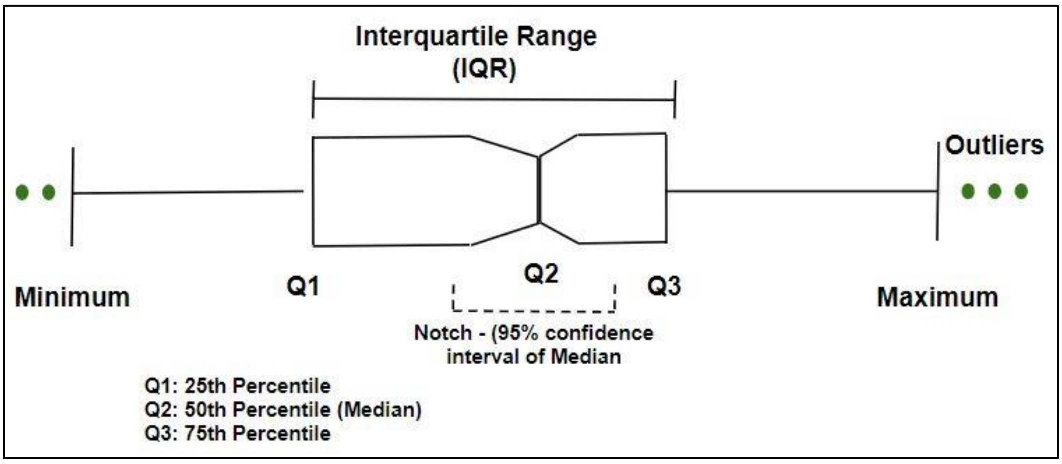


Figure 5-1: Parameters of Box Plot

Source: GeeksforGeeks

As displayed in Figure 6-1, the box shows the interquartile range (IQR). The IQR is the 25th percentile to 75th percentile. Percentile is also identified as Q1 and Q3. The IQR is where the center 50% of data points will fall. The whiskers add 1.5 times the IQR to the 75 percentile which is knows as Q3, and subtract 1.5 times the IQR from the 25 percentile also known as Q1. The whiskers should include 99.3% of the data if from a normal distribution. The line is the median of the data.The Notch displays the confidence interval around the median which is normally based on the median +/- 1.57 x IQR/squre root of no of data points. Outliers are also identified in the box plot as shown in Figure 6-1. According to Graphical Methods for Data Analysis, although not a formal test, if two boxes' notches do not overlap there is strong 95% confidence, their medians differ [55].

Next, box plot is used as the data driven method to define fuzzy membership function and two methods were identified as stated follows.

5.3 Five State Fuzzy Membership Function

In fuzzy membership function, there can be multiple states depending on the area selected. In this research, two fuzzy membership functions, where the main difference is the number of states, were identified. As presented in Figure 6-2 and

Figure 6-3, the first fuzzy membership function has five states and the other fuzzy membership function has three states respectively.

Figure 5-2 displays the five-state fuzzy membership function in using trigonometric and trapezoidal functions [82] where all the states of a box plot is used in the fuzzy membership function.

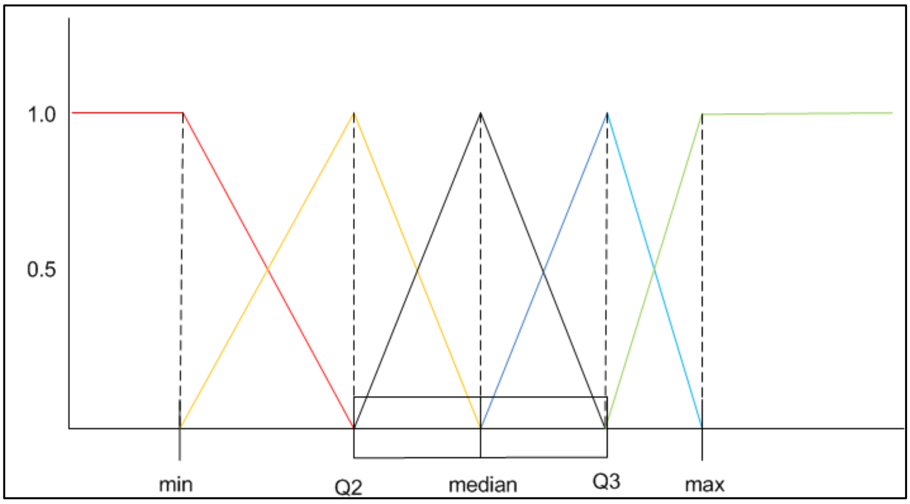


Figure 5-2: Five State Fuzzy Membership Function

State five fuzzy membership function is defined as shown in the Equation 5-1.

m_i – Minimum value

Q1 - First Quartile

m – Median

Q3 – Third Quartile

m_x – Maximum value

$$State1 = \begin{cases} 1 & x \leq m_i \\ \frac{-}{Q_1 - m_i} x + \frac{Q_1}{Q_1 - m_i} & m_i < x \leq Q_1 \\ 0 & x > Q_1 \end{cases}$$

$$State2 = \begin{cases} 0 & x \leq m_i \\ \frac{1}{Q_1 - m_i}x - \frac{m_i}{Q_1 - m_i} & m_i < x \leq Q_1 \\ \frac{-1}{Q_1 - m}x + \frac{m}{Q_1 - m} & Q_1 < x \leq m \\ 0 & x > m \end{cases}$$

$$State3 = \begin{cases} 0 & x \leq Q_1 \\ \frac{1}{m - Q_1}x - \frac{Q_1}{m - Q_1} & Q_1 < x \leq m \\ \frac{-1}{Q_3 - m}x + \frac{Q_3}{Q_3 - m} & m < x \leq Q_3 \\ 0 & x > Q_3 \end{cases}$$

$$State4 = \begin{cases} 0 & x \leq m \\ \frac{1}{Q_3 - m}x - \frac{m}{Q_3 - m} & m < x \leq Q_3 \\ \frac{-1}{m_x - Q_3}x + \frac{m_x}{m_x - Q_3} & Q_3 < x \leq m_x \\ 0 & x > m_x \end{cases}$$

$$State5 = \begin{cases} 0 & x \leq Q_3 \\ \frac{1}{m_x - Q_3}x - \frac{Q_3}{m_x - Q_3} & Q_3 < x \leq m_x \\ 1 & x > m_x \end{cases}$$

Equation (5-1)

The equation 5-1 shows the mapping of five state fuzzy membership function to the box plot five numbers.

During the implementation of this fuzzy membership function, different states can be assigned depending on the domain. For example, states of employee experience of plantation workers can be defined as Low, Low-Medium, Medium, High-Medium and High so that it can be mapped to fuzzy membership states.

5.4 Three State Fuzzy Membership Function

Three state fuzzy membership can be defined similar to the definition of the Five State Fuzzy membership function that was discussed in the previous section. In the three state fuzzy membership function, first quartile and the third quartile are ignored. Figure 5-3, shows the three-state fuzzy membership function that is combine trigonometric and trapezoidal functions where only minimum, median and maximum values are used in the fuzzy membership function.

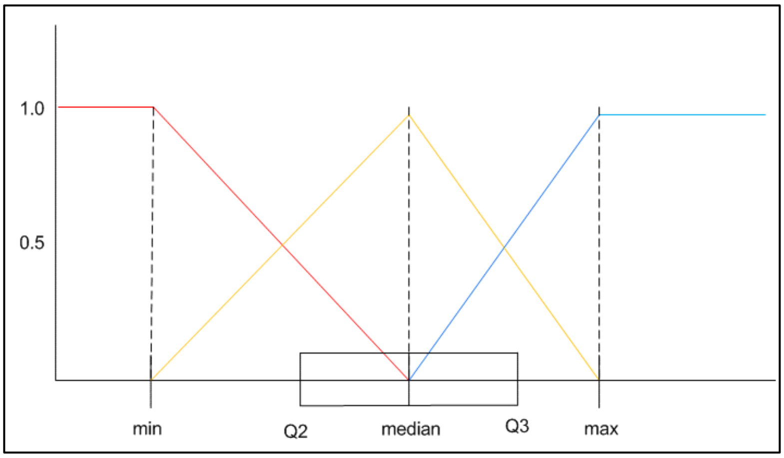


Figure 5-3: Three State Fuzzy Membership Function

Similar to five state membership function, three state membership function was defined as stated in the Equation 5-2.

m_i – Minimum value

m – Median

m_x – Maximum value

$$State1 = \begin{cases} 1 & x \leq m_i \\ \frac{-}{m-m_i}x + \frac{m}{m-m_i} & m_i < x \leq m \\ 0 & x > m \end{cases}$$

$$State2 = \begin{cases} 1 & x \leq m_i \\ \frac{1}{m-m_i}x - \frac{m_i}{m-m_i} & m_i < x \leq m \\ \frac{-1}{m_x-m}x + \frac{m_x}{m_x-m} & m < x \leq m_x \\ 0 & x > m_x \end{cases}$$

$$State3 = \begin{cases} 0 & x \leq m \\ \frac{1}{m_x-m}x - \frac{m}{m_x-m} & m < x \leq m_x \\ 1 & x > m_x \end{cases}$$

Equation (5-2)

In the three state fuzzy membership function derived method, Q1 and Q3 data points of box plots are ignored. During the implementation of this fuzzy membership function, different states can be assigned depending on the domain. For example, states of employee experience of plantation workers can be defined as Low, Medium and High three states.

During the implementation of this fuzzy membership function, different states can be assigned depending on the domain and the date set. In the defined fuzzy membership functions it is essential to identify the properties of those functions [94].

- Cross over point: In both fuzzy sets, for all states cross over point is 0.5 which indicates these fuzzy sets are symmetric.
- Height of the fuzzy is 1 in both fuzzy sets.

5.5 Implementation using Box-Plot Method

Before implementing a fuzzy membership function, it is essential to recognize what the constraints to implementing the proposed method are as this method has some limitations and boundries. Initially, it was decided to partition data to verify whether the data is uniform. Though there are several partition variables to be selected in any data set, the selected variable should not be biased. Most of the time, a year can be chosen as the partition variable as by partitioning by year will not be biased in most of the data domain. As an example, when fuzzy membership function is to be implemented on employees' age of a plantation factory, minimum age of the employee for the year or in other words age at the first month of the year was chosen for the boxplot.

All the boxplots are drawn on the same scale to detect the deviations of each boxplot as presented in Figure 5-4. R script for this plot and other related calculations are listed in the Appendix L: Sample R Script for Model Box Plot.

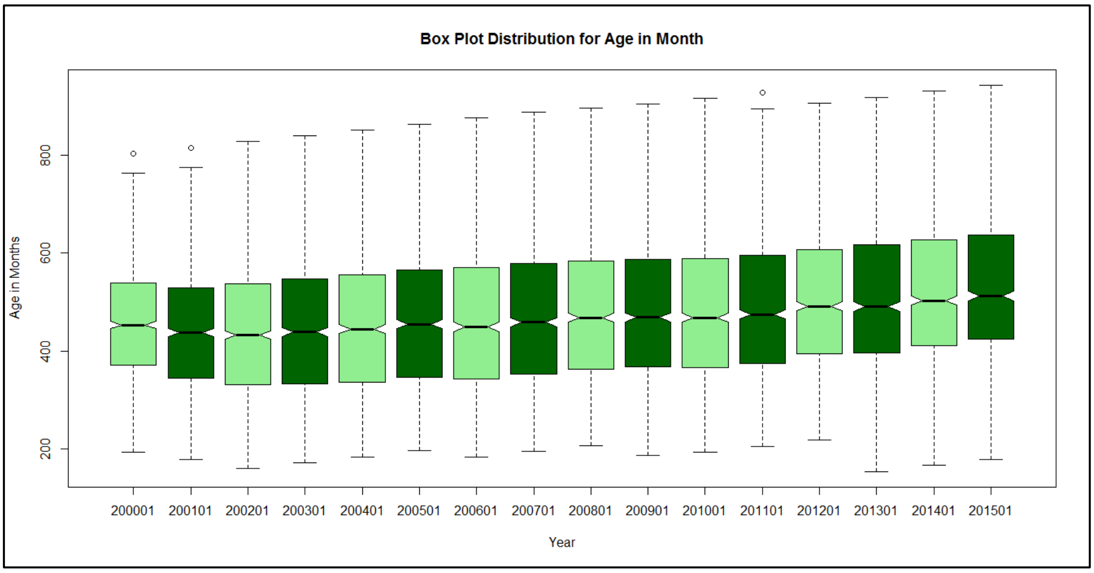


Figure 5-4: Range of Box Plots for Age Year Wise

Means of five values of Boxplot and stand deviation error is calculated to find out whether the data set can be applied to the boxplot mechanism as presented below in Table 5-1.

Table 5-1: Mean, Standard Deviation and Standard Error of the Mean of the Box Plots

	Mean	Standard Deviation	Standard Error of the Mean
m_i	151.40	0.81	0.05
Q_1	220.34	34.40	2.15
M	286.25	28.67	1.79
Q_3	363.00	37.72	2.36
m_x	570.75	78.19	4.89

Next, the calculated notch of the box plot needs to be validated. In the preceding example, notch ranges from 278 to 294 and Range percentage is around five percent. Since all of the Standard Error of the Mean values are less than five and notch range percentage is close to five percent, these data set can be considered to utilise box plot method to define fuzzy membership function.

After it was decided to use box plot methodology, the next step was to find out five numbers for the final box plot. As presented in Table 6-1, mean values for five numbers were used to define the average or the effective box plot for the data set which is presented in Figure 5-5.

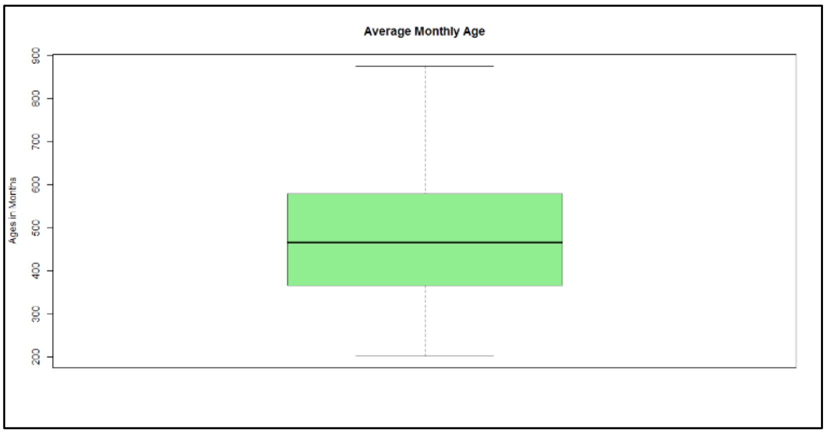


Figure 5-5: Range of Box Plots for Age Year Wise

After the effective Box Plot is defined for the entire data set, the number of states for the fuzzy membership function should be defined. Depending on the number of states, whether it is five or three, and relevant fuzzy membership function is selected. The sample R script is listed in Appendix M. Figure 5-6 displays all the steps to define effective membership function using the proposed box plot method which was described before in detail.

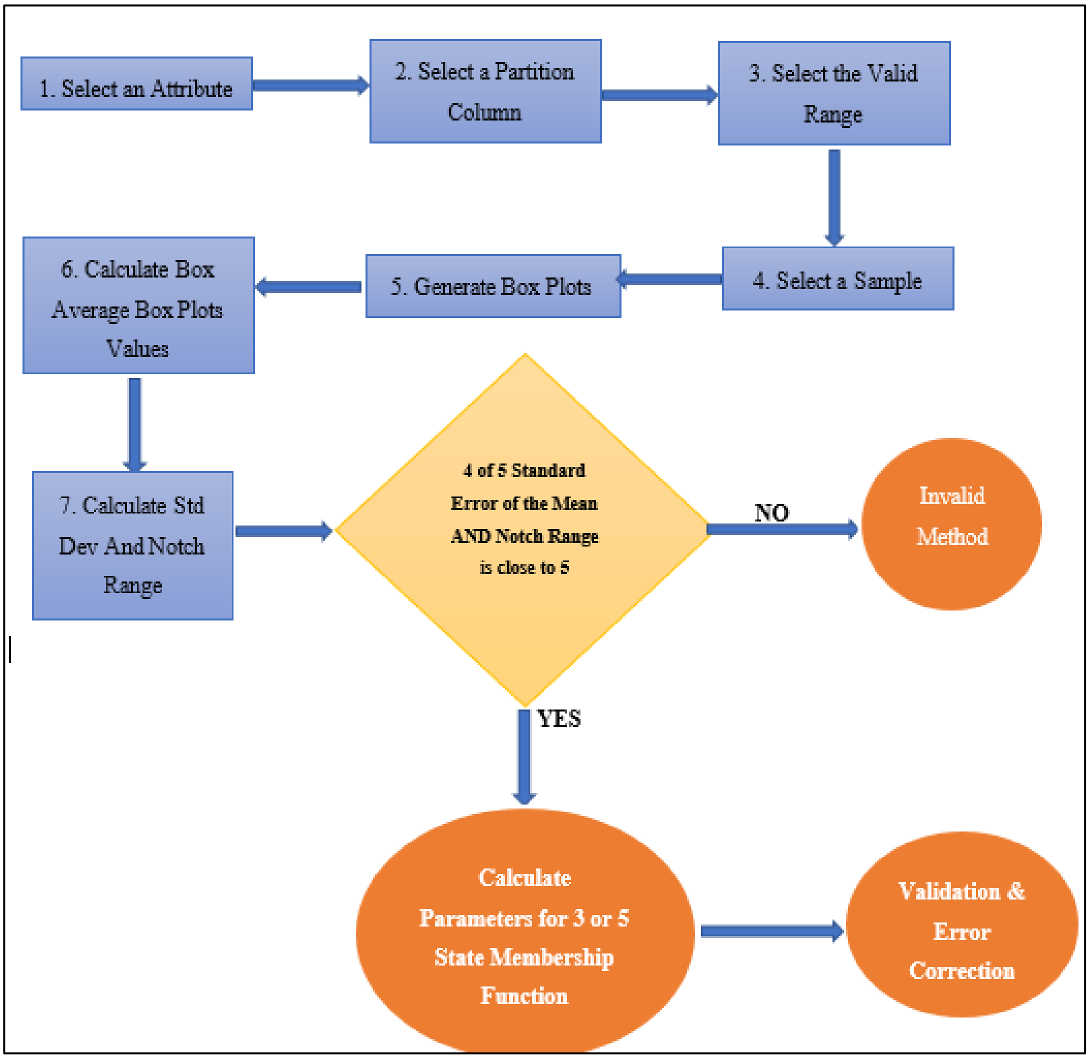


Figure 5-6: Steps for Box Plot Method

Figure 5-6, shows detailed flow for the box plot method to define fuzzy membership function. In this an important decision had to be made whether the four of five standard error of the mean and norch range was close to 5. If the criteria was not met, box plot method could not be used. A different method could be selected to define fuzzy membership function. Validation and error correction areas will be discussed later in the chapter.

5.6 Evaluation

As the five state method was found to be unsuited to define the fuzzy membership function of the age of the factory employees, three state method was utilized.

Following is the fuzzy membership function for three states as shown in the Equation 5-3. Values are rounded to the fourth decimal point for better configuration and usage.

$$Young = \begin{cases} 1 & x \leq 186 \\ -0.004x + 1.664 & 186 < x \leq 466 \\ 0 & x > 466 \end{cases}$$

$$Middle = \begin{cases} 0 & x \leq 186 \\ 0.004x - 0.664 & 186 < x \leq 466 \\ -0.0024x + 2.14 & 466 < x \leq 875 \\ 0 & x > 875 \end{cases}$$

$$Old = \begin{cases} 0 & x \leq 466 \\ 0.0024x - 1.14 & 466 < x \leq 875 \\ 1 & x > 875 \end{cases}$$

Equation (5-3)

Using the above definition for age, fuzzy membership function was drawn using Matlab and fuzzy logic designer as presented in Figure 5-7.

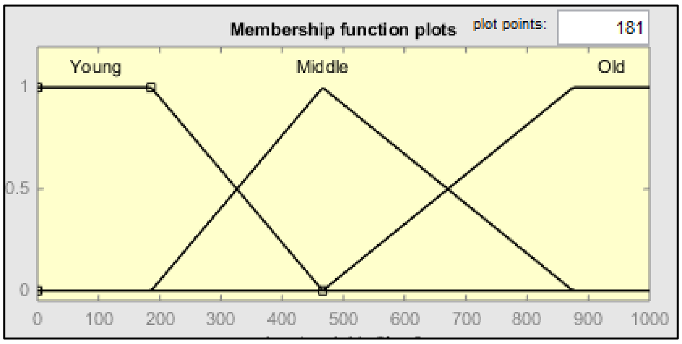


Figure 5-7: Fuzzy Membership Function for Age

In the preceding example, the range dimension was used to analyze range of ages presented in Table 5-2. Range Dimensions are typically used in data warehousing systems and is a crisp data set.

Table 5-2: Ranges of Employee Age Range Dimension

Range Description	Start Range	End Range
Young	1	300
Medium	301	540
Old	541	9999

Compared with Figure 5-8, it can be detected that fuzzy membership function and range dimensions are close to each other.

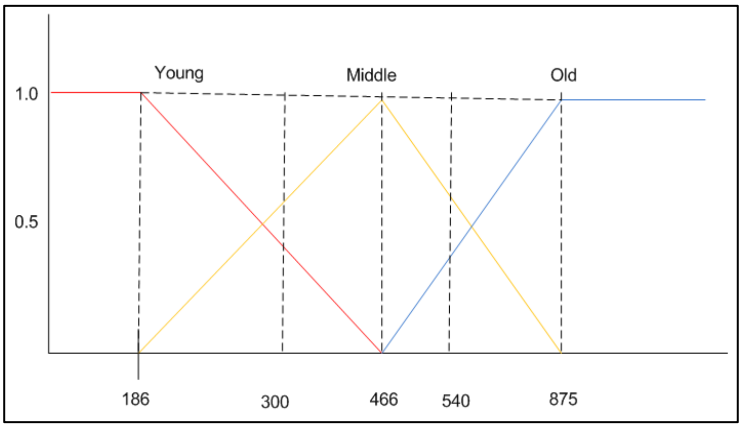


Figure 5-8: Comparison of Fuzzy Membership Function for Age with Crisp Data Sets

A similar evaluation was done for another attribute named experience of employees. The range was 1 month to 60 years. The sample was taken as similar to the Age which is presented in Figure 5-9.

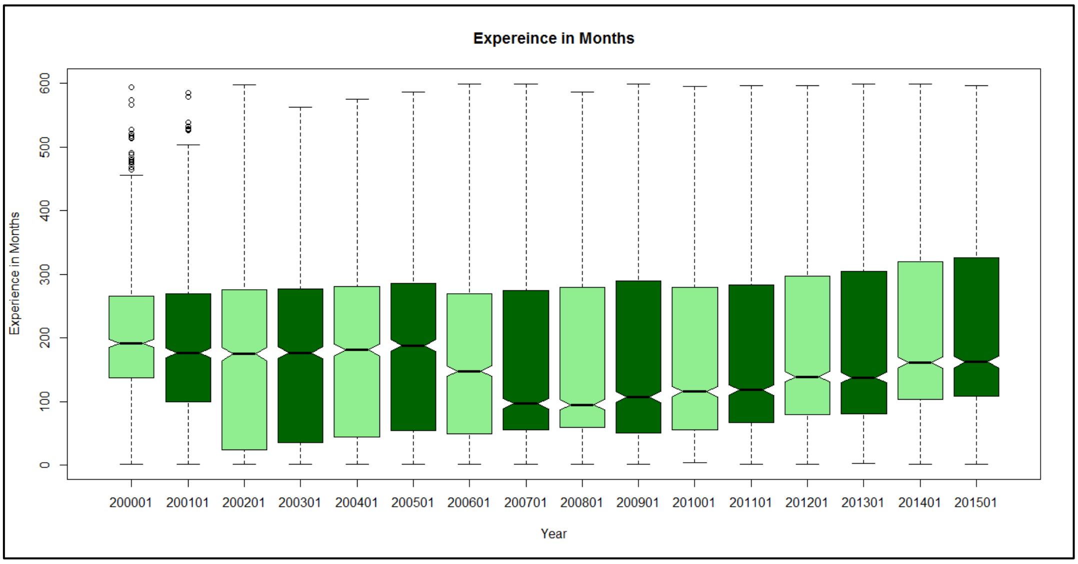


Figure 5-9: Range of Box Plots for Experience

Though the standard deviation values for all the five numbers are less than 60, Range percentage is more than 10. Hence, it can be concluded that this attribute is not suited for Box Plot method. Similarly, the number of years for permanency of factory employees was also evaluated and BoxPlots are drawn as presented in Figure 5-10.

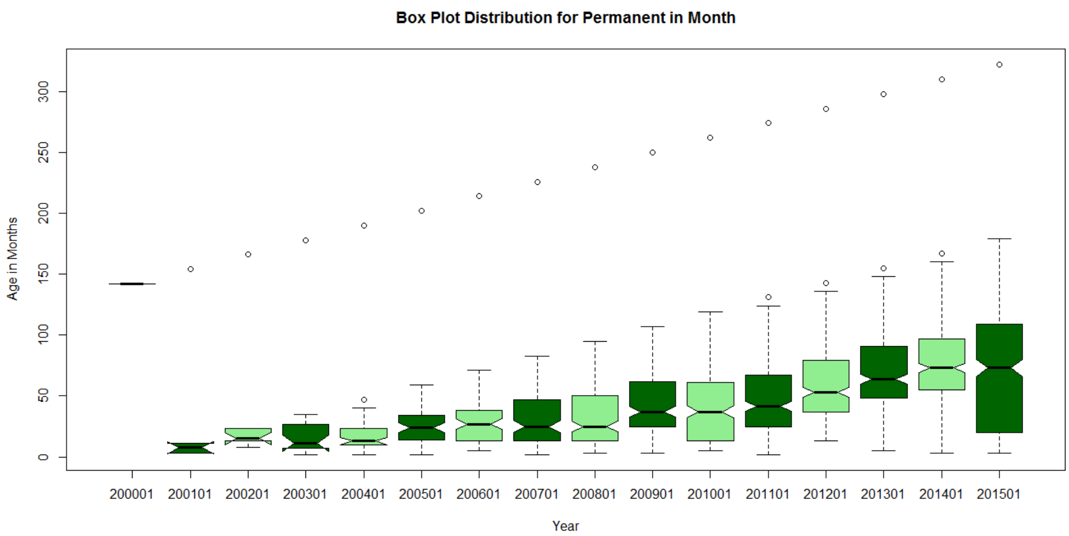


Figure 5-10: Range of Box Plots for Permanent Wise

Though the average box plot falls under defines values, range percentage is 19% which is a high value. Therefore, the number of months of permanency attribute also cannot be considered for the Box Plot mechanism like the experience attribute.

5.7 Validation and Error Correction

After the fuzzy membership function is defined by using the box plot technique, the next step was to define weightages for different data points. For example, ages in the month for employees were defined for range 150 to 1200. A subset of data is displayed in Table 5-3.

Table 5-3: Subset of Data Sample Weightages for Employee Age

Variable	Value	State	Weightage
Age	185	Young	1.000
Age	186	Young	1.000
Age	187	Middle	0.085
Age	187	Young	0.915
Age	188	Middle	0.089
Age	188	Young	0.911
Age	189	Middle	0.093
Age	189	Young	0.907
Age	190	Middle	0.097
Age	190	Young	0.903

As per the fuzzy set theory discussed earlier, for each value sum of weightage should be 1 as Height of the fuzzy set should be 1. However, in Table 5-4, it is shown that some of the areas, fuzzy membership function Height is more than 1. It is evident in the range of 416 and 568.

Table 5-4: Subset of Data Violation of Height of Membership Function for Age

Value	State	Weightage
462	Middle	1.185
463	Middle	1.189
464	Middle	1.193
465	Middle	1.197
466	Middle	1.201

Value	State	Weightage
467	Middle	1.203
468	Middle	1.201
469	Middle	1.199
470	Middle	1.197
471	Middle	1.195
472	Middle	1.193

This mismatch range is displayed in Figure 5-11.

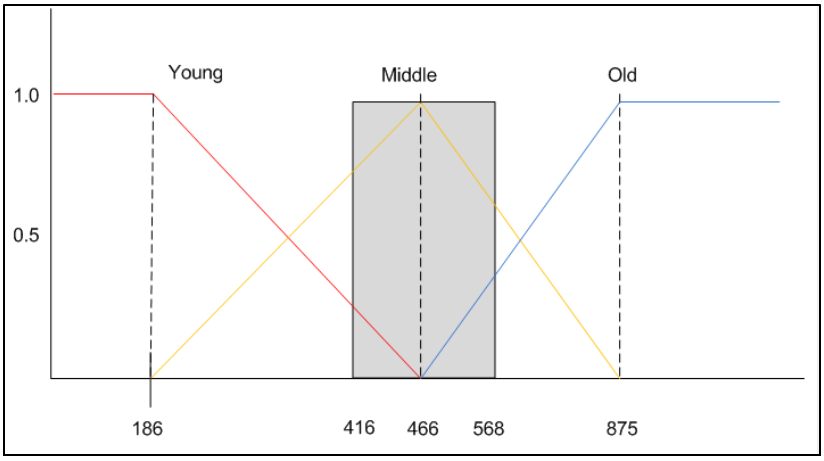


Figure 5-11: Membership Function for Age Before the Error Correction

For some values sum of the weightages or the Height of the fuzzy membership function is more than 1 which is invalid. This error is due to multiple roundups when the fuzzy membership functions is calcuated. The main reason for these invalid values are that there are instances where three functions are active in this range. To fulfil the Height of the fuzzy membership function, the fuzzy membership function is adjusted as presented in Figure 5-12.

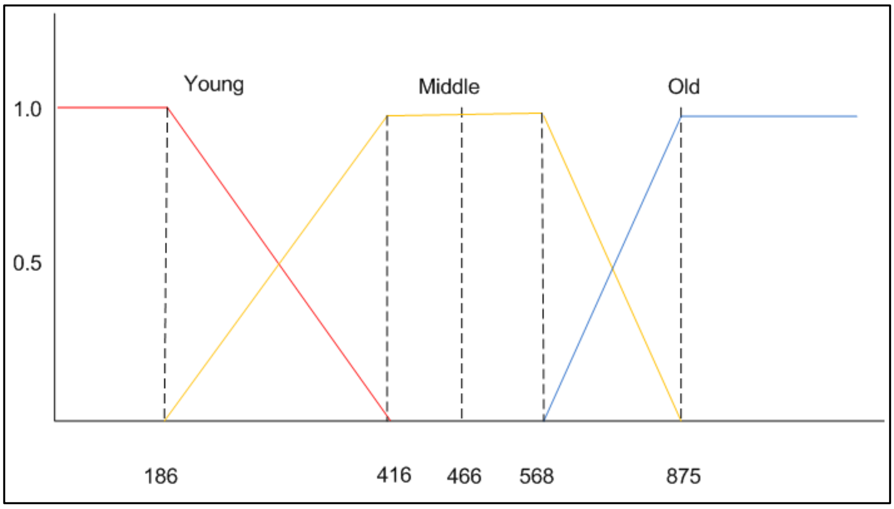


Figure 5-12: Modified Fuzzy Membership Function after Error Correction

After validation steps are executed on the above data set, triangular fuzzy membership function was converted to trapezoidal fuzzy membership function, the summation of weightage for a given value is equal 1 which will follow the basics of fuzzy membership function as the height of the fuzzy membership function is 1.

5.8 Properties of Fuzzy Membership Functions

To compare and contrast fuzzy membership functions, it is important to gather the properties of defined fuzzy membership functions. As discussed in Appendix G: Features of Fuzzy Membership Functions, Core, Support, Boundary, Cross-Over point and Height are the important features. However, in the defined fuzzy membership functions, height is 1 which means, these are normal fuzzy membership functions. Method of calculation is given in Appendix Q using Microsoft T-SQL which is the scripting language for SQL Server 2016. Using the above specified method, the following properties were observed for Age (3 States) and Expense (5 States) fuzzy membership functions as presented in Figure 5-13.

Support	State	Start	End
	Young	151	415
	Middle	187	875
	Old	569	1200
Core	**State**	**Start**	**End**
	Young	151	186
	Middle	416	568
	Old	876	1200
CrossOverPoint	**Function**	**State**	**CrossOverPoint**
	Age	Young	290.75
	Age	Middle	290.75
	Age	Middle	818.5
	Age	Old	818.5
Boundary	**State**	**Boundary**	
	Young	186 - 415	
	Middle	187 - 416 / 568 - 875	
	Old	569 - 876	

Figure 5-13: Properties of Fuzzy Membership Function for Age

These values were calculated and stored in a database table for retrieval later and when the need arose as well as for comparison. It would be much easier to store the pre-calculated values rather than the dynamically calculated values. However, whenever there is a change in the parameters of the fuzzy membership function, these values had to be re-calculated and stored again. Since data warehouse extraction and transformation executes at less frequency such as once per day, this mechanism would not impact the data warehouse operations but it will improve the performance. Following Figure 5-14 shows the properties of the fuzzy membership function for age of employee which consists of five states.

	State	Start	End
Support	Low	1	68
	Low Medium	3	143
	Medium	70	289
	High Medium	153	608
	High	301	720
	State	**Start**	**End**
Core	Low	1	2
	Low Medium	69	69
	Medium	144	152
	High Medium	290	300
	High	609	720
CrossOver	**Function**	**State**	**CrossOverPoint**
	Experience_5	Low	35.53
	Experience_5	Low Medium	35.53
	Experience_5	Low Medium	105
	Experience_5	Medium	105
	Experience_5	Medium	223.86
	Experience_5	High Medium	223.86
	Experience_5	High Medium	467.33
	Experience_5	High	467.33
Boundary	**State**	**Boundary**	
	High Medium	153 - 290 / 300 - 608	
	Low	2- 68	
	Low Medium	3 - 69 / 69 - 143	
	High	301 - 609	
	Medium	70 - 144 / 152 - 289	

Figure 5-14: Properties of membership Function for Experience (5 State)

Though the five-state membership function is rare in the data warehouse as well as in the fuzzy logic, it is essential to define the properties of 5-state fuzzy membership function. These definitions of the state will help during a troubleshooting incident in order to find the state and their cross points and data validation can be done.

5.9 Subjective Fuzzy Membership Function

There can be situations where both data-driven and subjective fuzzy membership functions can be combined to derive the fuzzy membership function. This means that, there can instances where slight changes to the data-driven values. Mostly there can be cases where boundary values are modified as presented in Figure 5-15.

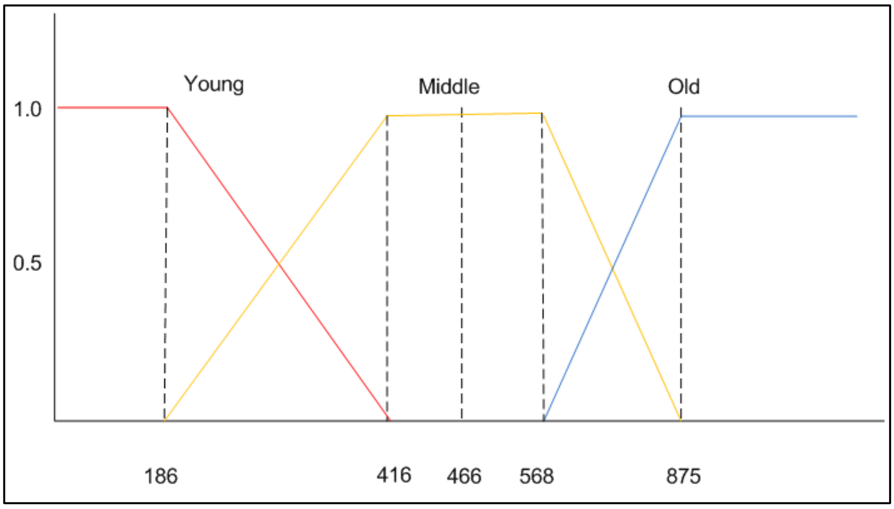


Figure 5-15: Fuzzy Membership Function for Experience

The above fuzzy membership function can be defined as shown in the following Equation 5-4.

Young = (Trapezoidal, {0, 0, 186, 416})

Middle = (Trapezoidal, {186, 416, 568, 875})

Old = (Trapezoidal, {568, 568, 875, 999999})

Equation (5-4)

In the above fuzzy membership function, all the states take the Trapezoidal relationship hence there should be four data points for each data point. There can be situations where fuzzy membership functions take the Trigonometric shape as presented in Figure 5-16.

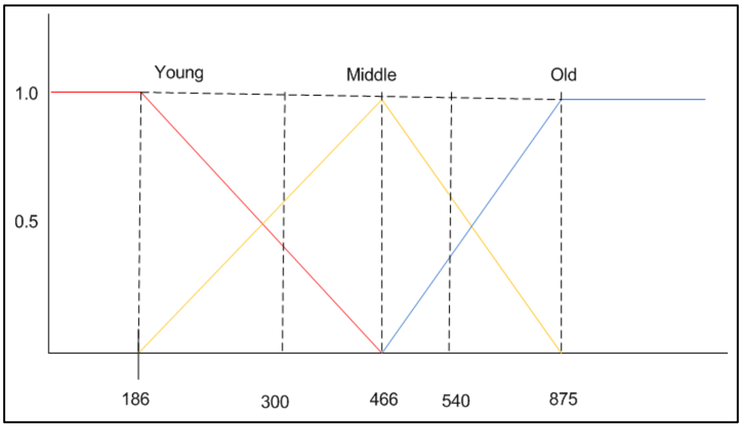


Figure 5-16: Fuzzy Membership Function for Age Comparison with Crisp Data Sets

Above fuzzy membership function can be defined as follows so that it can be matched with the database storage.

Young = (Trapezoidal, {0, 0, 186, 486})

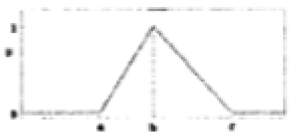
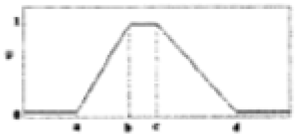
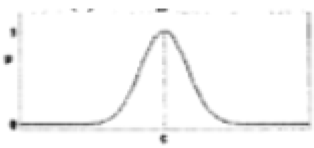
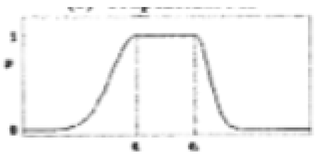
Middle = (Trigonometric, {186, 466, 875})

Old = (Trapezoidal, {466, 466, 875, 999999})

Equation (5-5)

In the Equation 5-5, Trapezoidal function has taken four data points while Trigonometric has taken three data points. This definition of this method is used with MatLab and in most research it is much easier to define the shape of the function and its parameters. Table 5-5 shows examples for each type of fuzzy membership function and its possible parameters.

Table 5-5: Different Fuzzy Membership Function Parameters

Membership Function	**Function Shape**	**Parameters**
Triangular		{a, b, c}
Trapezoidal		{a, b, c, d}
Gaussian		{a, b}
Two Sided Gaussian		{a, b, c,}

Though, there are high number of fuzzy membership functions to be introduced, only selected data types were chosen to support fuzzy data warehouse as they are the most common data types. However, during the data warehouse calculation, using the above configuration will lead to performance issues during the analysis. Therefore, the above configuration will convert to an equation so that the performance of this method is acceptable. Appendix S: has the full script of generating the fuzzy contributions from the subjective method.

This configuration can be stored in the following tabular format as shown in Figure 5-17 so that it can be reused later.

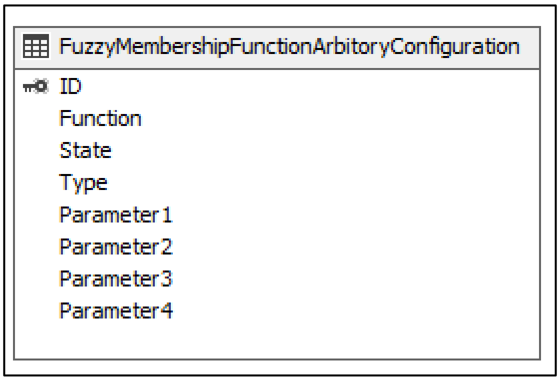


Figure 5-17: Fuzzy Membership Function Arbitory Configuration Table

From the above table, fuzzy membership functions were derived and the fuzzy contribution was calculated later as represented in Figure 6-18.

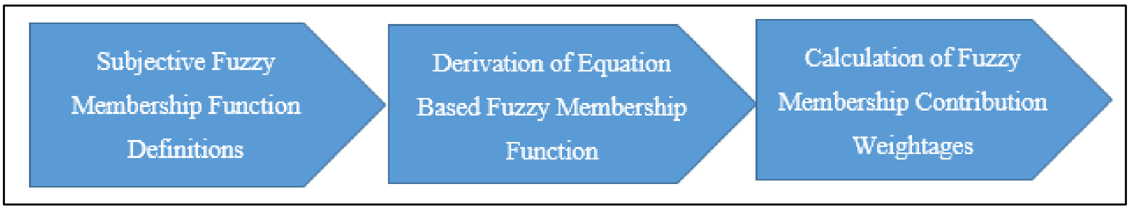


Figure 5-18: Process of Calculating Contribution Weightages for Subjective Fuzzy Membership Function

Figure 5-18, shows the description related to the process of calculating the contribution of weightages. As discussed in the previous chapters, calculating fuzzy membership weightages on the fly will not be helpful as far as performance is considered. Therefore, it was decided to pre-calculate fuzzy membership contribution weightages and use them during the ETL process. To calculate fuzzy membership contribution weightages, Subjective fuzzy membership function was converted to equation-based fuzzy membership function. Later, the subjective fuzzy membership function was converted the equation-based fuzzy membership function weightages were calculated similar to the data-driven method.

5.10 Hybrid Fuzzy Membership Function

There can be situations where both data-driven and subjective fuzzy membership functions can be combined to derive the fuzzy membership function. This means that, there can instances where slight changes to the data-driven values. Mostly there can be cases where boundary values are modified as shown in Figure 5-19.

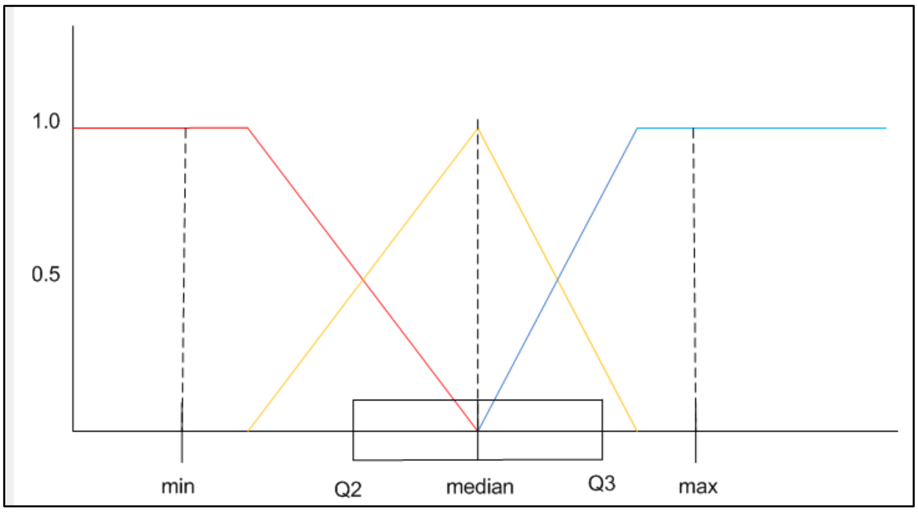


Figure 5-19: Hybrid Fuzzy Membership Function

As shown in Figure 5-19, the minimum value received from the data-driven method is moved ahead. For example, if the minimum value is 28.45 calculated from the data-driven method, it can be modified to 35 considering the environmental conditions. Similarly, the maximum value is moved before the value obtained from the data-driven method. Apart from the minimum and maximum values, other values obtained from the data-driven method were retained. In most cases, values calculated from the data-driven method were rounded-off for better usage which will become a hybrid fuzzy membership function.

5.11 Derived Fuzzy Membership Function

In some cases, fuzzy membership function is function of multiple fuzzy membership functions as shown in the Equation 5-6.

$$f(x) = fMF_1$$

$$g(x) = fMF_2$$

$$h(x) = f\,(f(x), g(x))$$

Equation (5-6)

fMF_1 – Fuzzy Membership Function 1

fMF_2 – Fuzzy Membership Function 2

In the above configuration, f(x) and g(x) are two separate independent membership functions and h(x) derived from f(x) and g(x) as shown in the Figure 5-20.

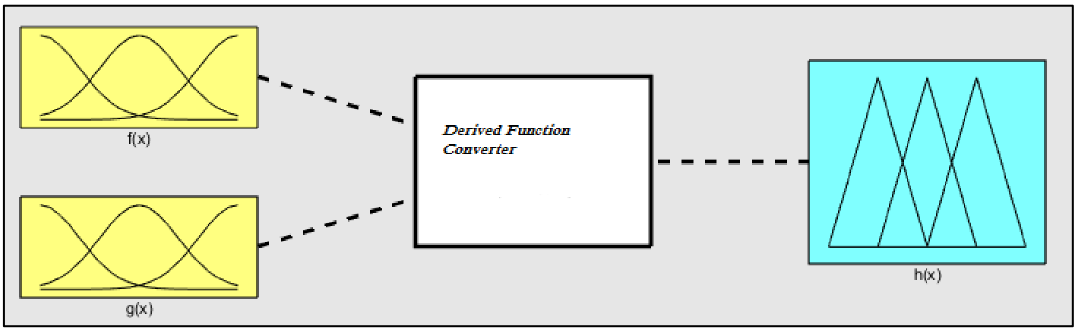


Figure 5-20: h(x) g(x) Fuzzy Membership Function Derived from f(x) and g(x)

h(x) is the fuzzy membership function which is defined by combining f(x) and g(x) functions.

Let us assume f(x), g(x) with defined values of a, b and Figure 5-21 shows the fuzzy membership functions for both f(x) and g(x).

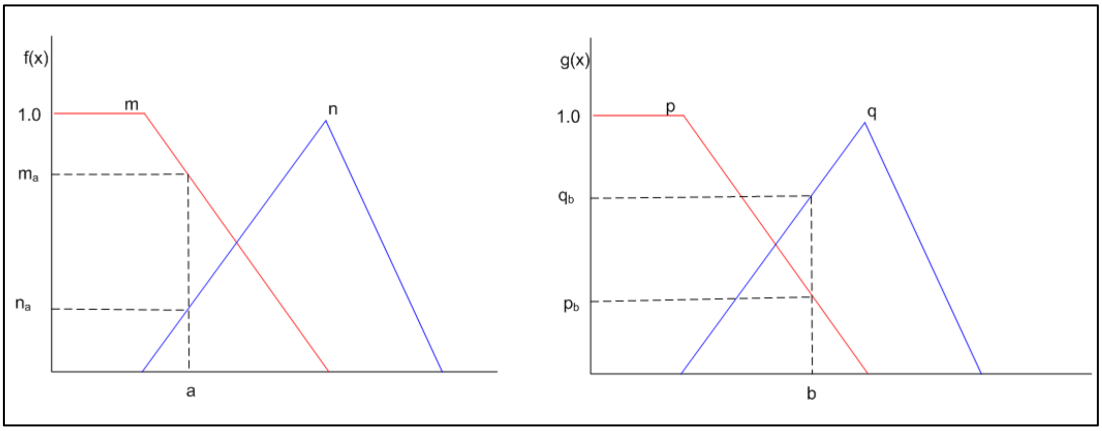


Figure 5-21: f(x) and g(x) Fuzzy Membership Functions

As both are normal fuzzy membership functions, the following properties are identified as shown in the Equation 5-7.

$$m_a + n_a = 1$$

$$p_b + q_b = 1$$

Equation (5-7)

Following is the rule table for derived attribute depending on the two base attributes which are shown in Table 5-6.

Table 5-6: Rules for Derived Attributes

Scenario	Base Attribute 1 & Values	Base Attribute 2 & Value	Derived Attributes & Value
1	M (m_a)	P (p_b)	M ∩ P ($m_a * p_b$)
2	N (n_a)	Q (q_b)	N ∩ Q ($n_a * q_b$)
3	M (m_a)	Q (q_b)	M ∩ Q ($m_a * q_b$))
4	N (n_a)	P (p_b)	N ∩ P ($n_a * p_b$)

It is important to note that some derived attributes status may not exist practically though they are mathematically or theoretically possible. In Table 6-6, either scenario 3 or scenario 4 is possible. There cannot be an instance where both scenarios will occur. It is derived in Appendix U: that the summation of derived attributes is equal to one which means that derived fuzzy membership function also follows the standard properties of a fuzzy membership function.

In the selected plantation attendance data set, similar relationships were found. In the case of the plantation sector, plantation fields are defined sections of the plantation area which are used as sub-areas for easy management purposes. In this plantation field, two important attributes were identified to measure the importance of the field. These attributes are acreage of the field and the number of bushes in the field. In the current crisp analysis, acreage and the numbers of bushes are divided into Low, Medium and High crisp categories. Depending on the combination of acreage and number of trees, importance also changed as shown in Table 5-7.

Table 5-7: Combination of Acreage and Number of Trees

Acreage	Number of Trees	Importance
ANY	NULL	NULL
NULL	ANY	NULL
Low	Low	Low
Low	Medium	Medium
Low	High	Medium
Medium	Low	Medium
Medium	Medium	Medium
Medium	High	High
High	Low	Medium
High	Medium	High
High	High	High

Table 6-7 indicates that if Acreage = Low and Number of Trees = High then the importance of the field will become Medium. There is also an additional state such as ANY and NULL which will be used accordingly. This matrix was saved in a customized table to obtained better performance. If the configurable table is defined for such cases, there will be performance implications. As there are a smaller number of derived fuzzy membership functions, it was decided to implement a customized table considering the performance and configurability. In this scenario, Acreage and

No of Trees of the fields are defined as subjective fuzzy membership functions as follows.

Acreage membership function can be defined as shown in the Equation 5-8.

Low = (Trapezoidal, {0, 0, 1, 6})

Medium = (Trigonometric, {1, 6, 8})

High = (Trapezoidal, {6, 8, 999999, 999999})

Equation (5-8)

The above definition will be converted to the following fuzzy membership function definition as shown in Figure 5-22.

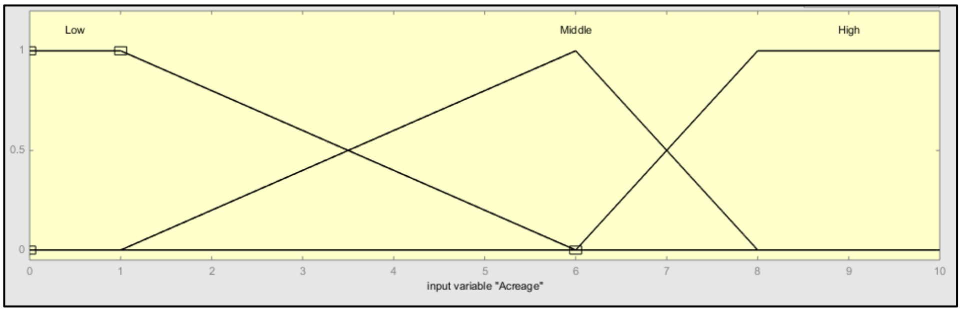


Figure 5-22: Acreage Fuzzy Membership Function

The above representation needs to be converted to a tabular format for implementation. The above graphical representation was derived in tabular format as presented in Table 5-8.

Table 5-8: Membership Function for Acreage in Formula

Function	**State**	**Function Type**	**Range 1**	**Range2**	**Parameter 1**	**Parameter 2**
Acreage	Low	Linear	0.00	1.00	0.0000	1.0000
Acreage	Low	Linear	1.01	6.00	-0.2000	1.2000
Acreage	Medium	Linear	1.00	6.00	0.2000	-0.2000

Function	State	Function Type	Range 1	Range2	Parameter 1	Parameter 2
Acreage	Medium	Linear	6.01	8.00	-0.5000	4.0000
Acreage	High	Linear	6.01	8.00	0.5000	-3.0000
Acreage	High	Linear	8.01	999999.00	0.0000	1.0000

As discussed in Section 6.3, Acreage function is a three-state fuzzy membership function and was stored in the tabular format. Equation 6.10 shows the fuzzy membership function is the number of Trees function.

Number of Trees function is derived as shown in the Equation 5-9.

Low = (Trapezoidal, {0, 0, 500, 1000})

Medium = (Trigonometric, {500, 1000, 25000})

High = (Trapezoidal, {1000, 25000, 999999, 999999})

Equation (5-9)

This definition would be converted to the following definition which was described before and which is shown in Figure 6-23.

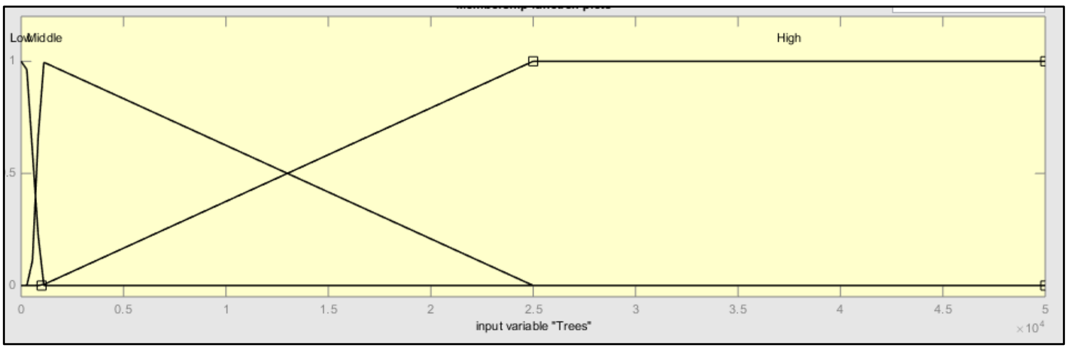


Figure 5-23: Trees Membership Function

Similar to acreage fuzzy membership function, for trees, the fuzzy membership function was converted to a tabular format as presented in Table 5-9.

Table 5-9: Membership Function for Trees in Formula

Function	State	Function Type	Range1	Range2	Parameter1	Parameter2
Trees	Low	Linear	0.00	500.00	0.000000	1.000000
Trees	Low	Linear	500.01	1000.00	-0.002000	2.000000
Trees	Medium	Linear	500.00	1000.00	0.002000	-1.000000
Trees	Medium	Linear	1000.01	25000.00	-0.000042	1.041667
Trees	High	Linear	1000.00	250000.00	0.000004	-0.004016
Trees	High	Linear	250000.01	999999.00	0.000000	1.000000

After defining fuzzy membership functions for base fuzzy membership function parameters as described above, the next was to define fuzzy membership values for derived fuzzy membership function of the value of the plantation field.

Table 5-10: Sample Data Set of Filed

Name of the Field	Area (Hec.)	Trees	Type
01B	3	3,452	Revenue
1A	3.8	16,197	Revenue
1B	4.5	8,466	Revenue
10A	12.5	76,250	Revenue
10B	4.5	27,540	Revenue
11A	1.5	19,566	Revenue
11B	10	62,000	Revenue
11C	12.3	74,113	Revenue
11B1	9.8	38,952	Revenue
11C1	1.5	21,173	Revenue
11D1	3	44,145	Revenue
12	16.8	84,085	Revenue

Value of Field is defined by its size (Acreage) and Number of Trees in the field as presented in Figure 5-24.

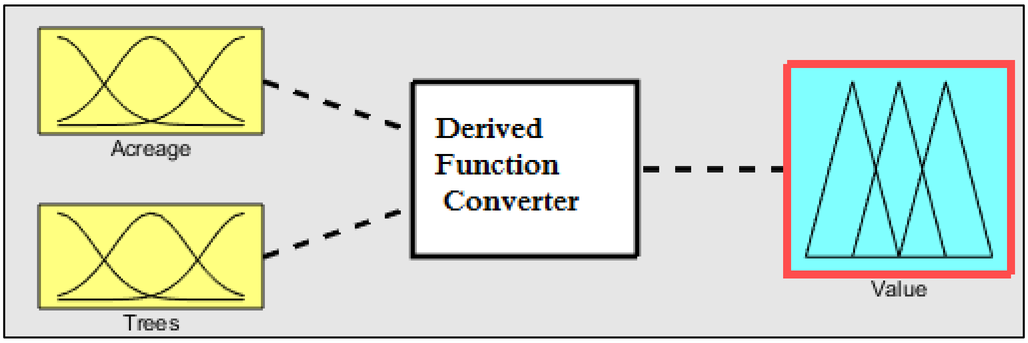


Figure 5-24: Derived Fuzzy Membership Function for Value of Plantation Field

Current data warehousing implementation is done by rules as shown in Table6-10. However, in the case of fuzzy data warehousing, implementing rules are difficult and using it for proper use is more difficult. Therefore, derived fuzzy membership function for Value is derived as shown in Figure 5-24. After fuzzy membership function was defined, fuzzy values were created using the previous method of subjective fuzzy membership function.

Using a script, Appendix V, the importance of the place of work was derived from the Number of Trees and hectare of the field. Details of the calculation are given in Appendix W: As shown in Table 5-11, for each field, fuzzy membership function contributions were defined.

Table 5-11: Sample Data Membership Function for Trees in Formula

Division Code	**PlaceofWork**	**Importance**	**Contribution**
HO	10	High	1.00000
HO	11	High	1.00000
HO	12	High	1.00000
HO	12A	Medium	1.00000
HO	12B	Medium	1.00000
HO	12C	Medium	1.00000
HO	5	High	0.11436
HO	5	Medium	0.88564
HO	5A	High	0.15504
HO	5A	Medium	0.84496
HO	5B	Medium	1.00000
HO	6	High	0.26580

To evaluate this derived fuzzy membership function technique, a state of Importance was compared between states of derived fuzzy membership function and the standard rules techniques. The comparison of a sample set of data is displayed in Table 5-12.

Table 5-12: Comparison of Importance State Between Fuzzy and Rules Techniques

Division Code	**PlaceofWork**	**Importance – Fuzzy**	**Importance -Rules**
HO	10	High	High
HO	11	High	High
HO	12	High	High
HO	12A	Medium	Medium
HO	12B	Medium	Medium
HO	12C	Medium	Medium
HO	5	Medium	High
HO	5A	Medium	Medium
HO	5B	Medium	Medium
HO	6	High	High

Table 5-13, is the confusion matrix for the derived fuzzy membership function for 112 plantation fields for the selected dataset as listed below.

Table 5-13: Confusion Matrix for Importance Attribute of Plantation Field

Classic	Fuzzy Range			
	High	Medium	Low	Grand Total
High	27	3	3	33
Low	2	33	1	36
Medium	2	5	36	43
Grand Total	31	41	40	112

The above table shows that, deriving fuzzy membership function for the importance of plantation field has a high accuracy of 85.71%.

5.12 Linguistic Variables for Fuzzy Membership Function

In the case of classical data warehousing, linguistic variables cannot be used much. Only possible linguistic analytics are negative and conjunctions such as “NOT High” or “High AND Low”. Since variables are crisp sets in classical data warehousing, only these types of analysis can be executed. However, in the case of fuzzy techniques, linguistic variable plays major roles with further analysis.

Linguistics variables are utilized to construct a relationship with the different state. For example, when defining a fuzzy membership function for temperature, there can be a state called “Hot”. Further, there can be two adjacent states such as “Less Hot” and “Very Hot” respectively. These two states can be a function of the Hot state which can be more accurately defined using linguistic fuzzy logic. Less, High, Low, much, more and less are termed as atomic terms. Appendix H: discusses the techniques for calculation of fuzzy linguistic functions.

These atomic terms may be grouped into three main classes.

- Primary terms: labels of *fuzzy* sets in X associated with their corresponding meaning, as *high* and *low* in the preceding example.
- Hedges: act as a linguistic modifier of the primary terms such as *very*, *much*, *more* or *less*.
- Connectives and negation: the connectives *and* and *or* and the negation *not* act as linguistic modifiers.

In this research, the fuzzy membership function is defined for primary terms and hedges. Connectives and negations are de-scoped from the research as these are available in classical data warehouse. Reference for the details of computing with a linguistic variable are described in Appendix H. Calculation of relationship operations in Fuzzy Sets is described in Appendix E. Table 6-3 has the five-state membership function for Experience and Permanency of the selected data set using the Box-plot method described before.

In the following, how computations can be done is revealed with a series of examples using typical hedges.

- ***If T = very L , then T(x) = CON_L(x)*** thus very acts as an intensifier. For example, considering the linguistic variable temperature, we may have *temperature* = *very high* with p = 2.
- If T = plus L then T(x) = Con_L(x) with p = 1.5.
- If T = minus L. then T(x) = Dil_L(x) with r = 0.75.

This modifies plus and minus which can be used to define some hedges whose meaning differs only slightly from others. For instance, *highly* could be defined as *highly* = *plus very* or possibly as *highly* = *minus very very*.

- If T = much L , then T(x) = Fuzz_L (x)
- If T= more or less then T(x) = Dil_L (x) with r = 0.4

This concept can be explained from the previously defined fuzzy membership functions. Table 5-14 shows the five state and three membership function parameters for employee experience.

Table 5-14: Five State and Three Fuzzy Membership Function Parameters for Employee Experience

State Type	State	Range1	Range2	Slope	Intersect
Five State	Low	0	2	0	1
		2	69	-0.015	1.033
		69	720	0	0
	Less Medium	0	2	0	0
		2	69	0.0015	-0.033
		69	149	-0.013	1.865
		149	720	0	0
	Medium	0	69	0	0
		69	149	0.013	-0.087
		149	289	-0.007	2.067
		289	720	0	0
	More Medium	0	149	0	0
		149	289	0.007	-1.067
		289	608	-0.003	1.902
		608	720	0	0
	High	0	289	0	0
		289	608	0.003	-0.090
		608	720	0	1
Three State	Low	0	2	0	1
		2	149	-0.007	1.015
		149	720	0	0
	Medium	0	2	0	0
		2	149	0.007	-0.015
		149	608	-0.002	1.324
		608	720	0	0
	High	0	149	0	0
		149	608	0.002	-0.324
		608	720	0	1

Then the equations are defined for both three-state and five-state functions as shown in Equation 5-10 and Equation 5-11. All the states are following a straight line with different intersect and slope for different ranges.

$$Low = \begin{cases} 1 & x \leq 2 \\ -0.015x + 1.033 & 2 < x \leq 69 \\ 0 & x > 69 \end{cases}$$

$$LowMedium = \begin{cases} 0 & x \leq 2 \\ 0.015x - 0.033 & 2 < x \leq 69 \\ -0.013x + 1.865 & 69 < x \leq 149 \\ 0 & x > 149 \end{cases}$$

$$Medium = \begin{cases} 0 & x \leq 69 \\ 0.013x - 0.865 & 69 < x \leq 149 \\ -0.007x + 2.067 & 149 < x \leq 289 \\ 0 & x > 289 \end{cases}$$

$$HighMedium = \begin{cases} 0 & x \leq 149 \\ 0.007x - 1.067 & 149 < x \leq 289 \\ -0.003x + 1.902 & 289 < x \leq 608 \\ 0 & x > 720 \end{cases}$$

$$High = \begin{cases} 0 & x \leq 289 \\ 0.003x - 0.902 & 289 < x \leq 608 \\ 1 & x > 608 \end{cases}$$

Equation (5-10)

$$Low = \begin{cases} 1 & x \le 2 \\ -0.0070x + 1.0150 & 2 < x \le 149 \\ 0 & x > 149 \end{cases}$$

$$Medium = \begin{cases} 0 & x \le 2 \\ 0.0070x - 0.0150 & 2 < x \le 149 \\ -0.0020x + 1.3240 & 149 < x \le 608 \\ 0 & x > 608 \end{cases}$$

$$High = \begin{cases} 0 & x \le 149 \\ 0.0020x - 0.3240 & 149 < x \le 608 \\ 1 & x > 608 \end{cases}$$

Equation (5-11)

Fuzzy contributions were generated for both states and for experience and plotted in Figure 5-25 and Figure 5-26. In both these figures, graphs at the start of the High state, show a sudden up which is an error.

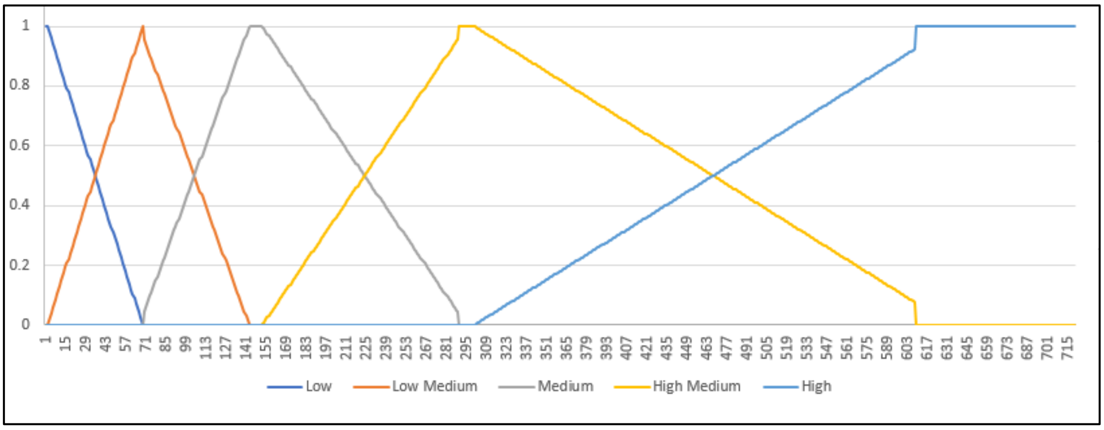


Figure 5-25: Five State Fuzzy Membership Function for Experience

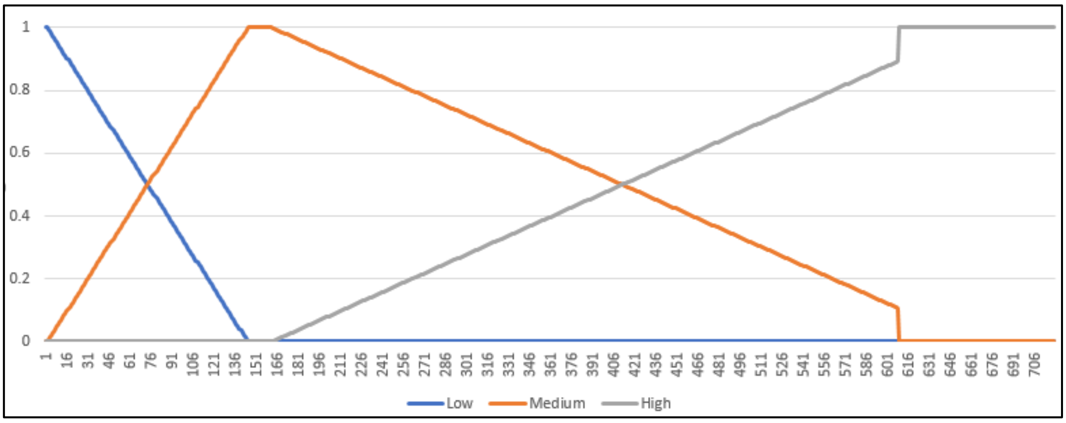


Figure 5-26: Three State Fuzzy Membership Function for Experience

Rather than defining the five-state function, which can be complex, the ideal option would be to derive the Less Medium and High Medium from the much simpler three state function. The linguistic variable option can be utilized.

This is the definition for Less and More Medium as shown in the Equation 5-12.

Let M(x) = Medium,

If T= less then T(x) = Dil_M (x) with r = 0.4

If T = more then T(x) = Con_L (x) with p = 1.6

Equation (5-12)

r and p values are arbitrary values but most common values in fuzzy research areas.

Various linguistic fuzzy states are defined as shown in the following configuration table shown in Figure 5-27.

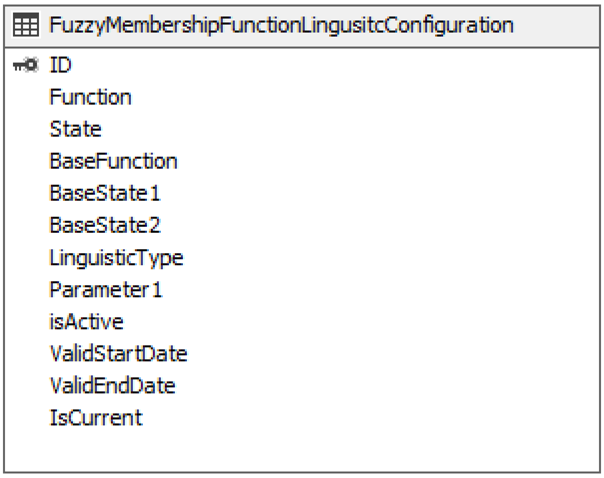


Figure 5-27: Fuzzy Membership Function Linguistic Configuration Table

In the configuration table presented in Figure 5-27, the Function was used to relate to the primary function of the fuzzy function. For example, Age, Experience will be the primary function. The state would define the linguistic state such as Low Medium, Not Low OR Not High, etc. Base Function was to offer the flexibility for several functions to be connected. Two bases states (Base State1 and Base State2) were included so that multiple states could be connected. For Example, Not Low OR Not High state both Low and High states are linked. Linguistics Type and it is parameters were defined in the Linguistic Type and parameter columns. Linguistic Types are standard operations in Fuzzy sets which are detailed in Appendix E: Operations in Fuzzy Sets. Sample data set for Linguistic variable configuration is given in Table 5-15.

Table 5-15: Sample Linguistic Variable Configuration

ID	Function	State	Base Function	Base State1	Base State2	Linguistic Type	Parameter
1	Experience_3	Low Medium	Experience_3	Medium		DIL	0.65
2	Experience_3	High Medium	Experience_3	Medium		CON	1.15
3	Experience_3	Not Low	Experience_3	Low		NOT	
4	Experience_3	Very Low	Experience_3	Low		DIL	0.50
5	Experience_3	Extremely High	Experience_3	High		CON	2.00
6	Experience_3	Low AND Very Low	Experience_3	Low	Low	NOT	
7	Experience_3	Not Low OR Not High	Experience_3	Low	High		
8	Experience_3	Not High	Experience_3	High		NOT	
9	Experience_3	Not Low Medium	Experience_3	Medium		NOT	

To support Type 2 Slowly Changing Dimensions for configuration, ValidStartDate, ValidEndDate and Is Current columns were introduced. Details of different Dimension types are given in Appendix B: Dimension Tables. In the next chapter, details of the implementation of slowly changing dimensions tables are discussed. Calculation process would first calculate dual linguistic terms such as Extremely High, Not Low, etc. Before calculating dual linguistic terms, basic terms calculations should be done. For example, to complete calculations of Extremely High, High values should be calculated beforehand. In case of conjunction linguistic terms such as Not Low OR Not High, Low AND Very Low, basic terms should be calculated beforehand. For example, to calculate Not Low OR Not High, not low, not high values should be calculated beforehand. In case the term is not available, entire linguistic values will not be calculated. Details for this calculation are listed in Appendix O: Script to Generate Fuzzy Contributions for Linguistic variables.

Figure 5-28 shows contributions for different linguistic variables.

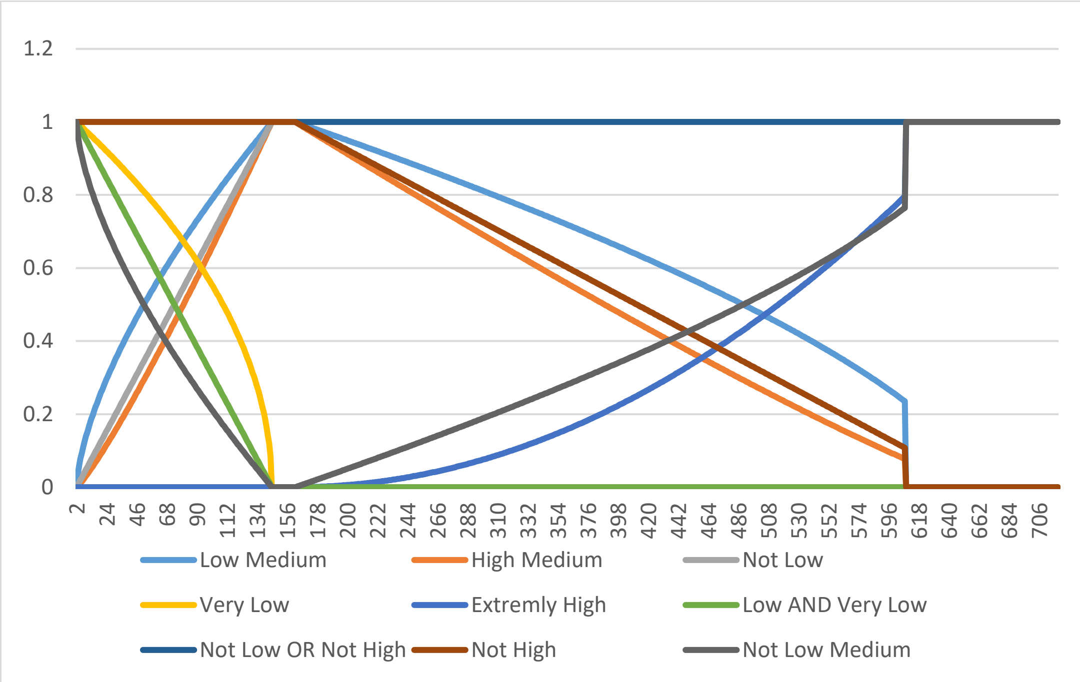


Figure 5-28: Different Linguistic Variables for Employees Experience

By introducing fuzzy linguistic variables, precision analysis on data warehouse can be done with different variables, much closer to the real-world analysis.

5.13 Integration of Fuzzy Membership Functions

As discussed in the previous sections, five different methods were introduced to define fuzzy membership functions. These five types are data-driven fuzzy membership function, subjective fuzzy membership function, hybrid fuzzy membership function, derived fuzzy membership function and linguistics membership functions. Figure 5-29 displays the relationships among the above said fuzzy membership functions to identify the integration of fuzzy membership functions.

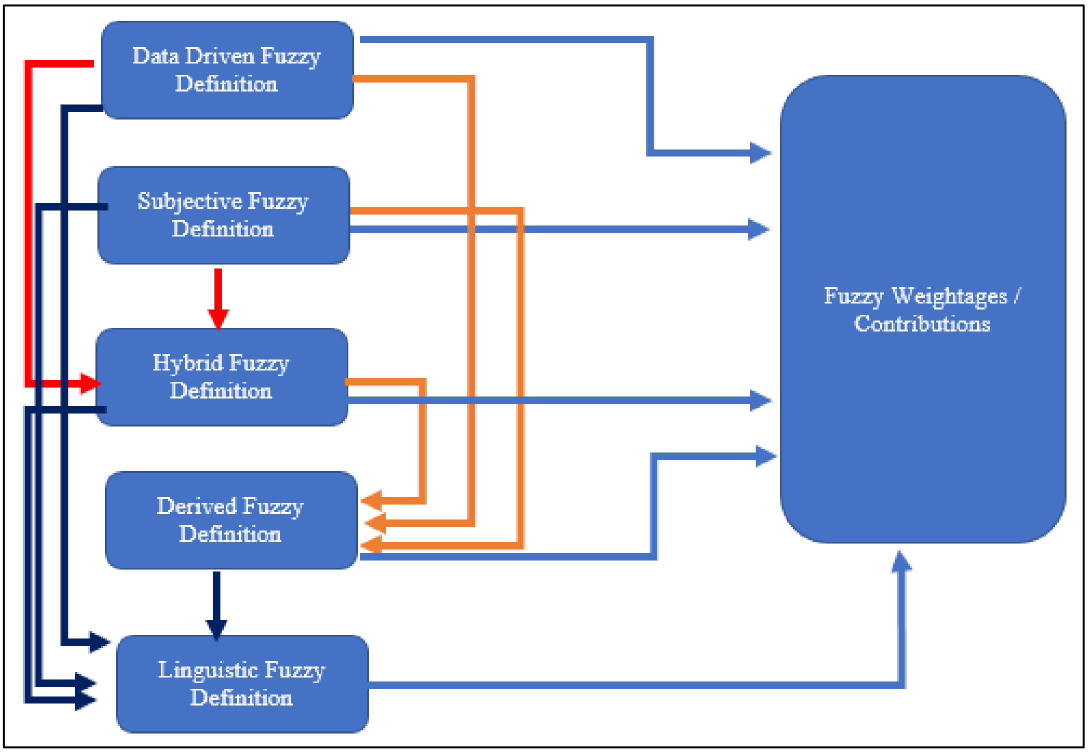


Figure 5-29: Integration of Multiple Fuzzy Membership Functions

As indicated in Figure 5-29, data-driven fuzzy membership functions and subjective fuzzy membership functions are independent fuzzy membership functions. Hybrid fuzzy membership functions are defined by combinations of data-driven and subjective fuzzy membership function. Derived fuzzy membership functions are defined from data-driven, hybrid and subjective fuzzy membership functions. Linguistics fuzzy membership, combines all the other fuzzy membership functions. By using all the five fuzzy membership functions, fuzzy weightages are defined. With physical implementation of these fuzzy membership functions, membership functions are defined in separate physical design. However, fuzzy contributions or weightages are calculated and stored in the same physical table. By storing them in one physical table, the usability of the system is improved.

5.14 Summary

Defining of the fuzzy membership function for fuzzy data warehouse was discussed in detail in this chapter as fuzzy membership function is the key aspect of this research. During the literature survey, it was identified that most of the modern fuzzy data warehouse research do not have detail and data driven definition of fuzzy membership function. It is more surprising that there is a limited research on data-driven fuzzy membership function.

Using the Box plot method, two types of fuzzy membership functions were introduced by mapping the data points of Box Plot data points to the fuzzy membership function. Three states and five states fuzzy membership functions were used. It is evident there are identified limitations in this method as not all the data sets are possible for this method and standard deviation of mean of the partitioned box plots should be considered when selecting this method. Error correction technique is introduced to smoothen the fuzzy membership function. An evaluation method was also proposed for the Box Plot method.

As in most real-world scenarios, the definition of the fuzzy membership function is subjective. Therefore, the fuzzy membership function definition is extended to the definition of the subjective fuzzy membership function. There are instances where some data points of data driven fuzzy membership should be modified to cater to practical reasons. For this purpose, hybrid fuzzy membership function was introduced. In the hybrid fuzzy membership function is the combination of data driven fuzzy membership function and subjective fuzzy membership function.

With data warehousing, there are rule based attributes definitions based on two other different attributes. To facilitate derived fuzzy membership function, the fuzzy membership function definition was extended. In data warehousing, there are derived attributes. Derived attributes are attributes which are defined from two or more base attributes. In the implementation of fuzzy technologies to the data warehouse, the fuzzy membership function for the derived attribute is defined using two base fuzzy membership functions of base attributes. Definition of Linguistic Variables is also identified in this chapter so that more options are available for analysis of data warehouse data.

After fuzzy membership function was defined the next step was to define design methodology for Fact tables and Dimension tables. Fact and Dimension tables are the core objects in data warehouse. Hence, it is essential to discuss design strategies to facilitate fact and dimension tables in data warehousing. The next chapter is dedicated to the discussion on the design methodology for fact and dimension tables in fuzzy data warehousing.

Chapter 6

FACT AND DIMENSION TABLES DESIGN

6.1 Introduction

After defining the fuzzy membership functions in using subjective, data-driven (Box Plot, etc), hybrid, derived and linguistic methods as discussed in detail in Chapter 5, the next step was to define fact and dimension tables for the fuzzy data warehouse.

According to the Ralph Kimball dimension modeling concepts in data warehousing, there are two types of main tables which are Fact and Dimension tables. Fact tables contain measure columns and surrogate keys which are joined to the dimension tables. For example, order amount, order quantity is fact columns. In addition to those measure columns, there are surrogate keys in the fact table, referring to dimension tables. There are three types of fact tables such as Transactional Fact table, Snapshot Fact table, and AccumulatingSnapshot Fact tables. Details of Fact tables are described in detail in Appendix A: Fact Tables.

A dimension is a structure that categorizes facts and measures to enable users to response to business questions. Commonly used dimensions are people, products, place and date. In a data warehouse, dimensions provide structured labeling information to otherwise unordered numeric measures. Details of Dimension tables are described in detail in Appendix B: Dimension Tables. Depending on the relationship between fact tables and dimension tables defined by the design pattern of for data warehouse described in Appendix AA:

In this chapter, design changes that need to be applied to accommodate fuzzy techniques to the data warehouse are elaborated in detail. The chapter also discusses a comparison between standard data warehouse design and fuzzy data warehouse. Further, dimension design techniques for range dimension and dimension categorizations are discussed. A detailed discussion is carried out to adopt historical aspects, a key factor in data warehousing. Different methods to define hierarchical categorization in the fuzzy dimensions are also introduced.

6.2 Standard Implementation of Data Warehouse

Plantation attendance data is used as proof of concepts, described in previous chapters. The main reason to use plantation attendance data is that it is real-world data. The plantation attendance system is used to manage employees in the plantation sector with different dimension such as Place of Work etc. The sample data set for 2012 – 2016 was chosen from a plantation factory in the upper country of Sri Lanka. Properties related to the data set were described in detail in section 4.3.

Figure 6-1 displays a typical implementation of a data warehouse for the above-stated plantation attendance system.

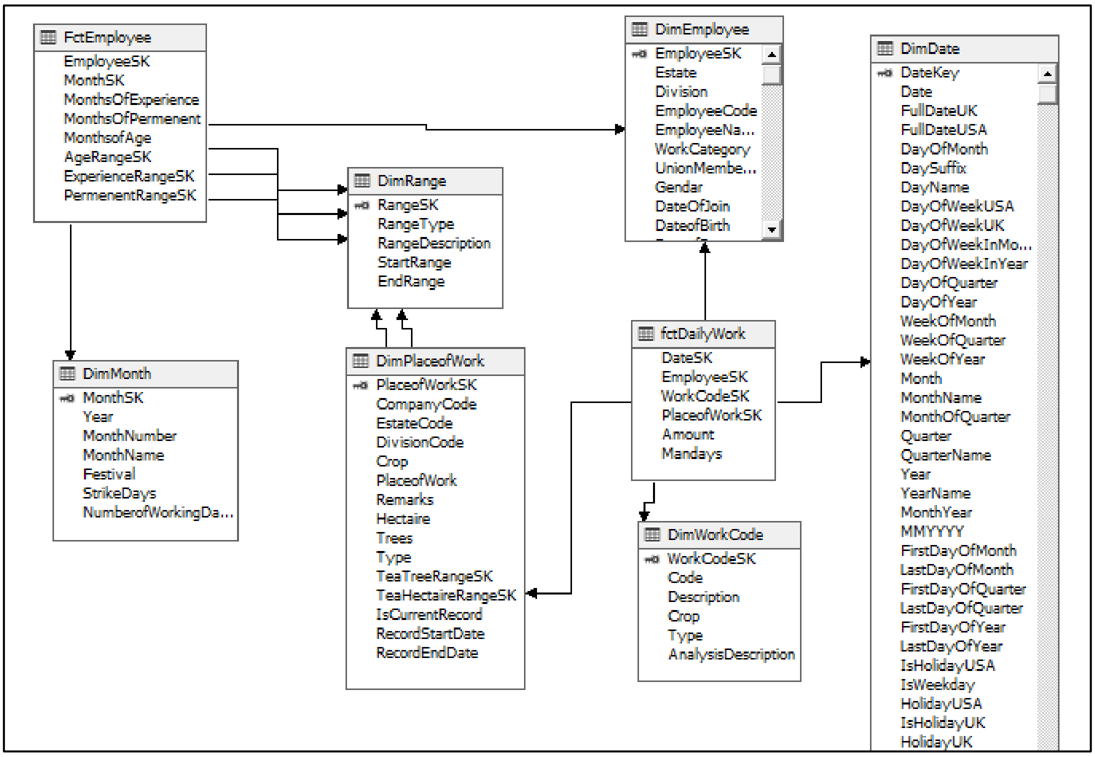


Figure 6-1: Standard Warehouse Design for a Plantation Attendance System

In the defined data warehouse, two transaction fact tables were defined. The fctDailyWork fact table is designed to store the daily work details of employees. Since employees can be employed with different types of work, work code is associated with the fact table. In case of a plantation attendance system, employee analysis is a key factor. To analyze employee’s age, experience, etc., fctEmployee fact table is used.

In this designed model, several dimensions are added. Three main dimensions are DimEmployee, DimWorkCode and DimPlaceofWork. DimEmployee and DimPlaceofWork dimensions are type 2 slowly changing dimensions which means historical aspects of dimension data is maintained. Date and Month dimensions were used to measure the date and month for fctDailyWork and fctEmployee respectively. In this dimension modeling, different granularity was used for date and month. Employees’ age, experience and permanent was measured on a monthly basis for which, month dimension was introduced in addition to the mostly used date dimension. In the data model, employee is measured by fact table, a dimension which is very unique.

DimRange is the dimension used to analyze bucketized data. For example, in the case of plantation data set, to analyze plantation fields, size of fields is bucketized. Low field size is allocated for fields of between hectares 0.01 to 1. Apart from the size of fields, Number of Trees, the age of employees, the experience of employees is the possible implementation for bucketized. In the DimRange dimension, the minimum and maximum values are used. Table 6-1 shows a typical configuration of range dimensions for the classical data warehouse.

Table 6-1: Sample Data for Range Dimension

RangeSK	**RangeType**	**RangeDescription**	**StartRange**	**EndRange**
-1	ALL	N/A	NULL	NULL
1	Age	Young	1.00	300.00
2	Age	Medium	301.00	480.00
3	Age	Old	481.00	9999.00
4	Experience	Low	1.00	60.00
5	Experience	Medium	61.00	120.00
6	Experience	High	121.00	9999.00
7	Permanent	Low	1.00	60.00
8	Permanent	Medium	61.00	120.00
9	Permanent	High	121.00	9999.00
10	Tea_Tree	Low	1.00	12000.00
11	Tea_Tree	Medium	12001.00	40000.00

RangeSK	RangeType	RangeDescription	StartRange	EndRange
12	Tea_Tree	High	40001.00	99999999.00
13	Tea_ Hectare	Low	0.01	1.00
14	Tea_ Hectare	Medium	1.01	3.00
15	Tea_ Hectare	High	3.01	99999.00
16	Tea_ Hectare	No Value	0.00	0.00
17	Tea_Tree	No Value	0.00	0.00

Since this follows a range dimension model, fuzzy implementation can be implemented in DimRange dimension. The DimRange dimension is designed in such a way that different types of ranges can be adopted. As shown in the Table 6-1, Age, Experience, Permanent, Number of Trees and Hectare were configured for ranges. With the above table schema, the ranges of the entire system can be defined,

Figure 6-2 shows how DimRange dimension is linked to other dimensions in the classical data warehouse data modeling.

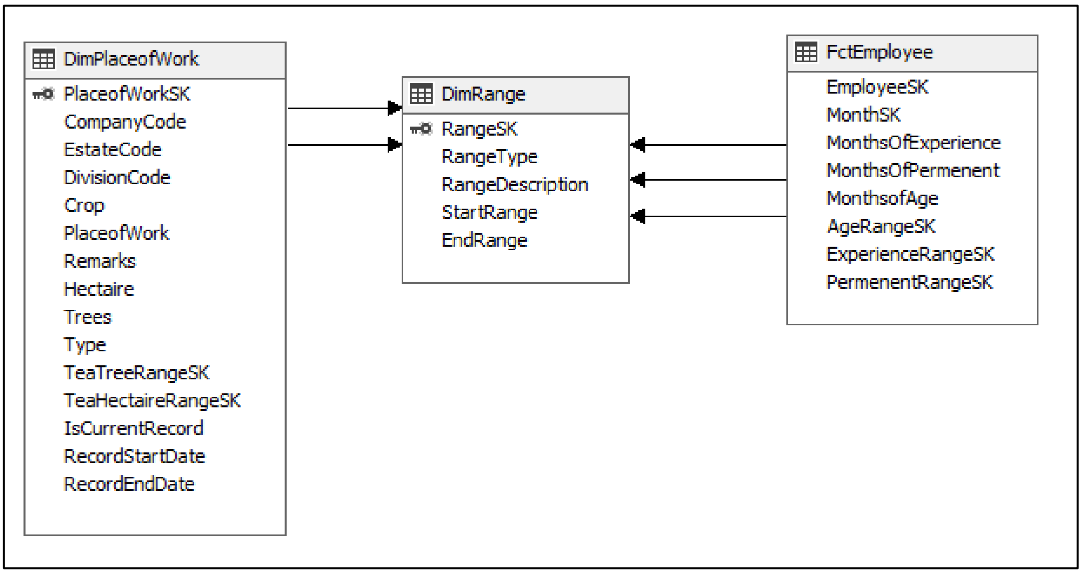


Figure 6-2: DimRange Dimension

Since DimRange is a common dimension table, this table was used to join multiple tables in classical data warehouse design. In the FCTEmployee fact table, AgeRangeSK, ExpereinceRangeSK, PermenentRangeSK are the surrogate keys which were used to join DimRange dimension table. Not only for fact table, but DimRange can also be utilized to implement ranges to Dimension tables as well. For the example in Figure 6-2, DimPlaceofWork dimension contains TeaRange and TeahectairRange columns which are referenced to DimRange like for the fact table.

6.3 Fuzzy Implementation for Range Values

A fuzzy dimension is defined as the dimension in which the attributes involved in analysis are fuzzified. The attribute values in the fuzzy dimension (FD) are mapped to a set of two or multi-valued attributes containing a state (S) and its corresponding membership value as described in Equation 6-1.

F (FD) = domain (FD) 0> <S,[0,1]>

Equation (6-1)

As described in Appendix D: Types of the Membership function, there are six types of fuzzymembership functions. In all six types of fuzzy membership function, at least two parameters are needed. When designing fuzzy dimensions, there can be two aspects which are configuration and dimension values. Like defining a Range Dimension in a standard data warehouse, in case of fuzzy dimensions, fuzzy dimension needs to be defined. The FuzzyMembershipFunctionConfiguration table is used to store the fuzzy membership function configuration details. After considering all the possible fuzzy membership functions, it was identified that all functions can be derived from two ranges and two parameters as shown in Appendix D:

For the fuzzy range configurations Fuzzy Membership Function Configuration was used and to store the fuzzy contributions for each function FuzyContribution table was used and schema structures of these tables are shown in Figure 6-3.

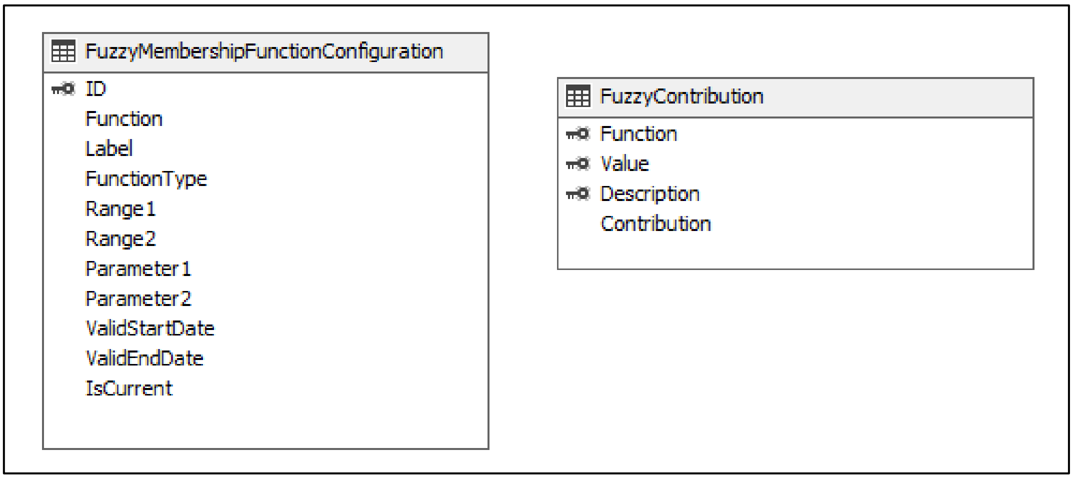


Figure 6-3: Fuzzy Range Dimension Configuration

In this design,FuzzyContribution table was generated by using the configurations of Fuzzy Membership Function Configuration table. In the Fuzzy Membership Function Configuration table, last three columns, ValidStateDate, ValidEndData and IsCurrent columns are used incorporate type 2 slowly dimensions are described in Appendix B. Table 6-2 shows sample set of data set which is stored in the Fuzzy Membership Function Configuration table.

Table 6-2: Fuzzy Membership Function Configuration Table Data

ID	Function	Label	Function Type	Range1	Range2	Parameter1	Parameter2
1	Age	Young	Linear	0.00	186.00	0.0000	1.0000
2	Age	Young	Linear	186.01	466.00	-0.0040	1.6640
3	Age	Young	Linear	466.01	999999.00	0.0000	0.0000
4	Age	Middle	Linear	0.00	186.00	0.0000	0.0000
5	Age	Middle	Linear	186.01	466.00	0.0040	-0.6640
6	Age	Middle	Linear	466.01	875.00	-0.0024	2.1400
7	Age	Middle	Linear	875.01	999999.00	0.0000	0.0000
8	Age	Old	Linear	0.00	466.00	0.0000	0.0000
9	Age	Old	Linear	466.01	875.00	0.0024	-1.1400
10	Age	Old	Linear	875.01	999999.00	0.0000	1.0000

In the sample data set shown in the Table 6-2, all the functions follow a linear profile. Therefore, it requires a range (Range1, Range2), slope (Parameter1), and intersect (Parameter2). In case of other profiles of the fuzzy membership function, different parameters can be stored in the same table of the schema.

Data in the Table 6-2 is represented the following Figure 6-4.

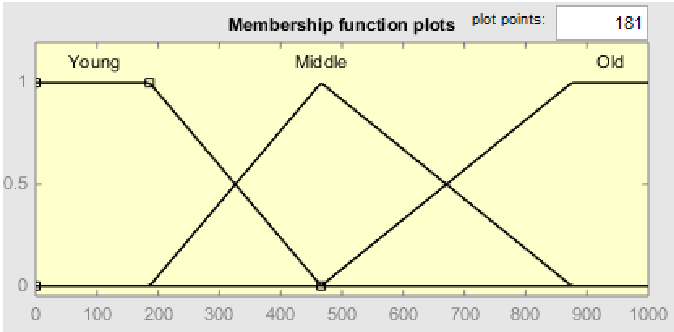


Figure 6-4: Fuzzy Membership Function for Age

As all are linear, it can be represented by intersecting and gradient. The gradient is in the parameter1 and intersect is in the parameter2 of the Fuzzy Membership Function Configuration table. By using the script listed in Appendix N: Fuzzy Contribution table is updated as presented in Table 6-3.

Table 6-3: Fuzzy Contribution Table Sample Data

Function	Value	State	Contribution	Calculation Method
Experience_3	2	Low	1.0000	S
Experience_3	2	Low AND Very Low	1.0000	L
Experience_3	2	Not High	1.0000	L
Experience_3	2	Not Low	0.0000	L
Experience_3	2	Not Low Medium	1.0000	L
Experience_3	2	Not Low OR Not High	1.0000	L
Experience_3	2	Very Low	1.0000	L
Experience_3	3	High Medium	0.0028	L
Experience_3	3	Low	0.9940	S
Experience_3	3	Low AND Very Low	0.9940	L
Experience_3	3	Low Medium	0.0360	L
Experience_3	3	Medium	0.0060	S
Experience_3	3	Not High	1.0000	L
Experience_3	3	Not Low	0.0060	L
Experience_3	3	Not Low Medium	0.9640	L

Function	Value	State	Contribution	Calculation Method
Experience_3	3	Not Low OR Not High	1.0000	L
Experience_3	3	Very Low	0.9970	L
Experience_3	4	High Medium	0.0068	L
Experience_3	4	Low	0.9870	S
Experience_3	4	Low AND Very Low	0.9870	L
Experience_3	4	Low Medium	0.0594	L
Experience_3	4	Medium	0.0130	S
Experience_3	4	Not High	1.0000	L
Experience_3	4	Not Low	0.0130	L
Experience_3	4	Not Low Medium	0.9406	L
Experience_3	4	Not Low OR Not High	1.0000	L
Experience_3	4	Very Low	0.9935	L
Experience_3	5	High Medium	0.0111	L
Experience_3	5	Low	0.9800	S
Experience_3	5	Low AND Very Low	0.9800	L

In table 6-3, both Linguistic (L) and Standard (S) weightages are calculated for each value and each state. By storing the Calculation Method (L or S) in the Fuzzy Configuration Table, it can be easily identified for auditing purposes.

The next stage was to include the fuzzy values to the fact tables. In traditional or classical data warehouses, every fact record will have one record in dimension. However, in the case of fuzzy data warehouses, every fact record will have multiple records in fuzzy states. For example, in case of an employee experience of 314 months, there will be two contributions such as High 0.3040 and Medium 0.6960.

In Figure 6-5, it shows the relationship between the fact table and the fuzzycontribution table.

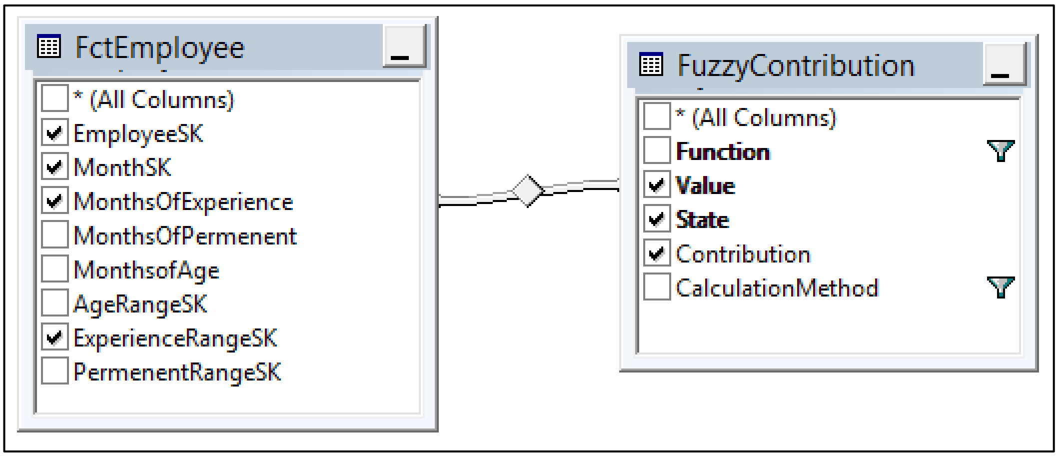


Figure 6-5: Employee Experience in Fuzzy Presentation

As shown in the Figure 6-5, each employee will have multiple records in Fuzzy Contribution. In the sample data set shown in the Table 6-4, classical crisp values are also included in the RangeDescription column so that it can be compared with the fuzzy contributions as shown in the Table 6-4. The script for the following output is available as script1 in Appendix P:

Table 6-4: Primary Fuzzy Contributions with the Comparison of Crisp State

EmployeeSK	Months Of Experience	MonthSK	State	Contribution	Range Description
2	358	200001	High	0.392	High
2	358	200001	Medium	0.608	High
3	183	200001	High	0.042	Medium
3	183	200001	Medium	0.958	Medium
4	72	200001	Low	0.511	Medium
4	72	200001	Medium	0.489	Medium
9	164	200001	High	0.004	Medium
9	164	200001	Medium	0.996	Medium
11	25	200001	Low	0.840	Low
11	25	200001	Medium	0.160	Low
12	58	200001	Low	0.609	Low
12	58	200001	Medium	0.391	Low
13	58	200001	Low	0.609	Low
13	58	200001	Medium	0.391	Low

It is not easy to read this table as single employee is referred in multiple rows. Therefore, pivoting techniques were used to make the data set more readable and for use with standard analytical techniques such as pivot tables, OLAP cubes, etc. as shown in Table 6-5. The relevant script is shown as Script 2 in Appendix P:

Table 6-5: Primary Fuzzy Contributions for Employee Experience

EmployeeSK	MonthsOfExperience	MonthSK	Low	Medium	High	Total
92	181	200403	0.000	0.962	0.038	1
458	185	200407	0.000	0.954	0.046	1
1351	292	200702	0.000	0.740	0.260	1
366	107	200901	0.266	0.734	0.000	1
1088	136	201106	0.063	0.937	0.000	1
245	304	201306	0.000	0.716	0.284	1
464	455	201503	0.000	0.414	0.586	1
2813	44	201512	0.707	0.293	0.000	1
844	339	200102	0.000	0.646	0.354	1
847	162	200211	0.000	1.000	0.000	1
409	317	200508	0.000	0.690	0.310	1
1020	75	200608	0.490	0.510	0.000	1
2292	9	200812	0.952	0.048	0.000	1

Table 6-5, shows the fuzzy contribution for each value. For example, 136 of MonthsofExpereince is mapped to 0.063 Low and 0.937 Medium. Since these are standard fuzzy weightages, the summation of the weightages should be equal to one if not, it does not follow the rules of fuzzy functions.

6.4 Evaluation

Fuzzy contributions need to be evaluated. Since fuzzy contributions have multiple states, it is a challenge to compare and evaluate with the traditional crisp state. It was presumed that tuple which has more than 0.5 contributions is the major contribution. Then the confusion matrix for employee experience was constructed as presented in Table 6-6.

Table 6-6: Confusion Matrix for Fuzzy Range for Experience

Classic	**Fuzzy Range**			
	High	Low	Medium	Grand Total
High	27,099		80,669	107,768
		79,104		79,104
Medium		13,546	113,060	126,606
Grand Total	27,099	92,650	193,729	313,478

Then the error and accuracy were calculated for the experience of employees.

Error = $\frac{13{,}546+80{,}669}{313{,}478}$ = 30.3%

Accuracy Factor is = 69.7 %

Similarly, confusion matrix was built for Age as presented in Table 6-7.

Table 6-7: Confusion Matrix for Fuzzy Range for Age

Classic	**Fuzzy**			
	Middle	Old	Young	Grand Total
Middle	117,324	121,275	4,670	243,269
Old		3,241		3,241
Young			21,375	21,375
Grand Total	117,324	124,516	26,045	267,885

$$\boldsymbol{Error = \frac{121,275 + 4,670}{267,885}}$$

Error = 47.3%

Correction Factor is = 52.7 %

Comparison of Corrections rates of Experience and Age give in the Table 6-8.

Table 6-8: Comparison of Corrections rates of Experience and Age

Domain	Error	Correction Factor
Experience	30.3	69.7
Age	47.3	52.7

In this comparison, fuzzy implementation of experience is more correct than that of Age, compared to the current data warehouse implementation

6.5 Different Types of Dimensions

In data warehousing, the main consideration will be on slowly changing dimensions out of other dimensions which is stated in Appendix B:. However, in the implementation, most common slowly changing dimension implementation is type 1 and type 2 [44]. Therefore, in this research, type 1 and type 2 slowly changing dimensions are considered.

Implementation of Type 1 slowly changing dimensions is trivial. In Type 1, if the dimension record exists it will update if not it will insert a new dimension record. This implementation is shown in Figure 6-6.

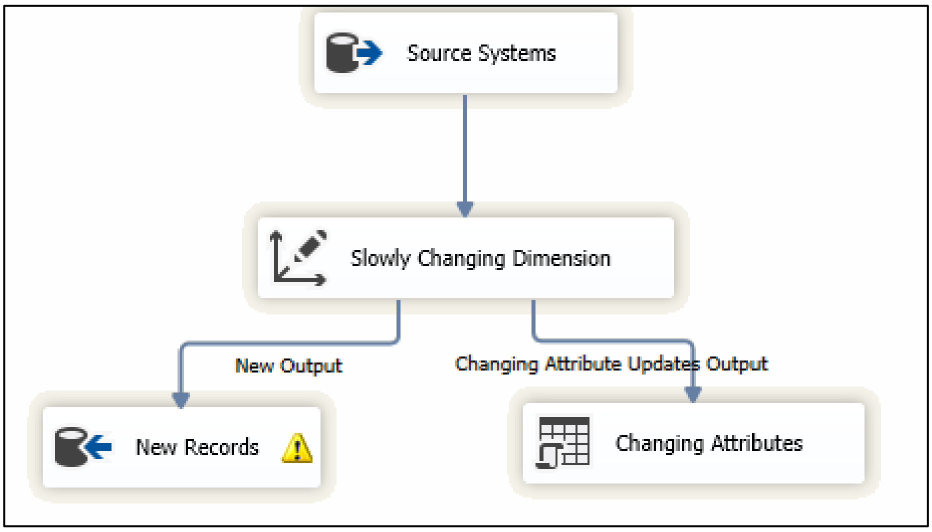


Figure 6-6: Implementation of Type 1 SCD

In the case of fuzzy data warehousing, three will not be any impact as type 1 slowly changes dimensions act similar to normal type 1 dimension. Hence, any design change is unlikely when it comes to the implementation of fuzzy data warehousing.

In the case of type 2 slowly changing dimensions, history aspect needs to be stored. Therefore, implementation of type 2 slowly changing dimensions is complex than that of type 1 slowly changing dimensions. Implementation of type 2 slowly changing dimension is described in Appendix B with an example. Implementation of type 2 SCD is shown in Figure 6-7.

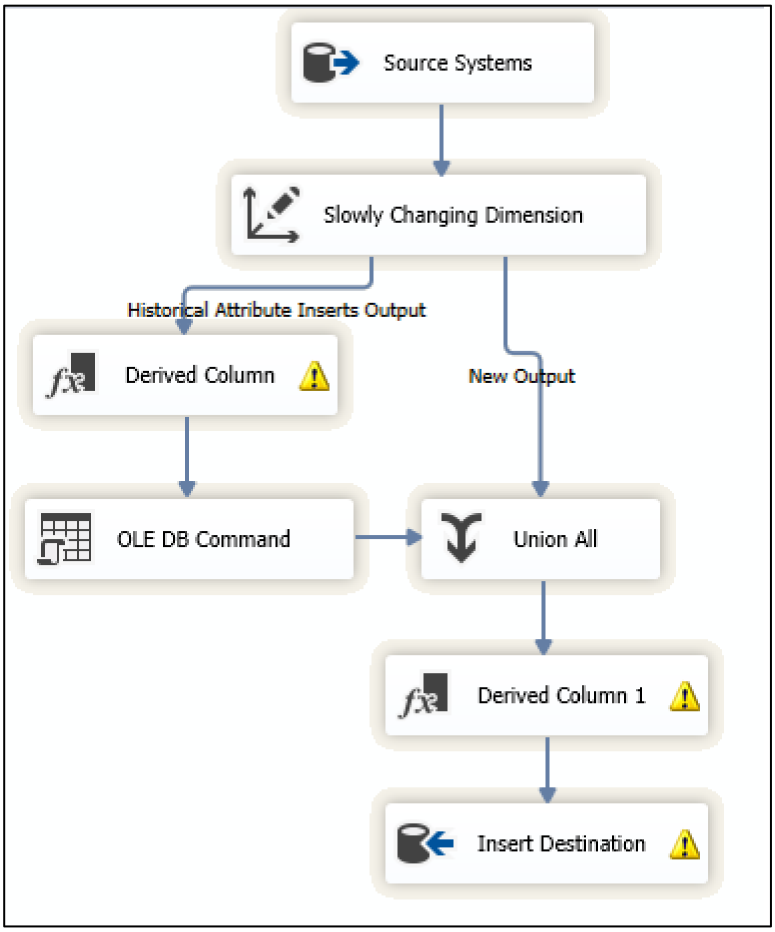


Figure 6-7: Implementation of Type 2 SCD

Comparing Figure 6-6 and Figure 6-7, easily confirms that the implementation of Type 2 SCD is more complicated than Type 1 SCD and this means that Type 2 SCD has a performance impact as well.

6.6 Fuzzy Configuration for Type 2 Slowly Changing Dimension

As type 2 SCD is an unavoidable object in any data warehousing, it is essential to implement type 2 SCD in the fuzzy data warehouse as well. In the proposed fuzzy data warehousing, three configurations tables are introduced.

They are:

- Fuzzy Membership Function Configuration
- Fuzzy Membership Function Arbitory Configuration
- Fuzzy Membership Function Linguistic Configuration

To introduce type 2 slowly changing dimension to these configurations so that historical configurations can be stored, three additional columns are added as shown in Figure 6-8.

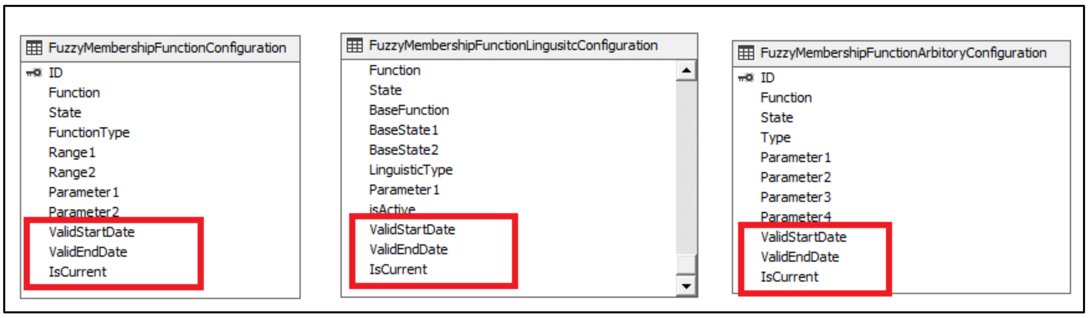


Figure 6-8: Implementation of Type 2 SCD for Fuzzy Configuration Tables

In the above design, three additional columns are introduced namely ValidStartDate, ValidEndDate and IsCurrent. These columns are traditional columns introduced in type 2 slowly changing dimensions. IsCurrent column is to identify the latest records out of the different versions of the record. In most cases, only the isCurrent column is used. However, in this additional ValidStartDate and

ValidEndDate columns are proposed to record the valid time period for the configuration record. In case of need to reload the fact tables, which is a somewhat normal practice in the traditional data warehouses, it is essential to keep the ValidStartDate and ValidEndDate column. In case of a new record, previous current record isCurrent value will be set to 0 to state that it is not the latest anymore and the ValidEndDate will be updated with the current date, which is an operation named, “Date Ending” in the data warehouse. So, the current date value for the current record will be NULL.

The same configuration was introduced to the Fuzzy Contribution table so that historical aspect of fuzzy contribution can be kept as shown in Figure 6-9.

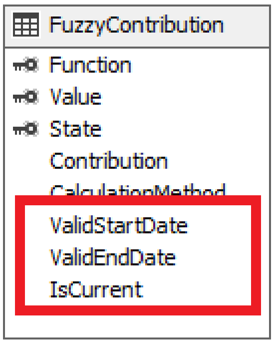


Figure 6-9: Implementation of Type 2 SCD for Fuzzy Contribution Table

Table 6-9 shows sample data set for Fuzzy contribution which contains previous and current contribution for Age value 187.

Table 6-9: Sample Data Set for Fuzzy Contribution

Record #	**Function**	**Value**	**State**	**Contribution**	**StartDate**	**EndDate**	**Is Current**
1	Age	187	Middle	0.085	1900/01/01	2018/12/31	FALSE
2	Age	187	Young	0.915	1900/01/01	2018/12/31	FALSE

3	Age	187	Middle	0.050	2018/12/31	NULL	TRUE
4	Age	187	Young	0.950	2018/12/31	NULL	TRUE

In Table 6-9, Age 187 until January 31st, 2018 has the contribution of Middle 0.085 and Young 0.915. After January 31st of 2108, Middle contribution is 0.05 while Young contribution is 0.095. In the case of current valid records, IsCurrent = TRUE filter should be used while for a relevant date range StartDate and EndDate should be used.

6.7 Hierarchical Categorization Dimensions

Categories are mostly used in data warehousing to perform deep analysis of data in data warehouse. In data warehouse design, categories and subcategories are typically de-normalized to dimensions and are shown in Figure 6-10. This sample table design is extracted from Microsoft Sample database of AdventureworksDW2016 which is freely available with Microsoft SQL Server 2016. A research paper focused on Design strategies of a Fuzzy Data Warehouse [63] has suggested Fuzzy hierarchies. In this research paper, fuzzy data warehouse modeling was introduced to product and product categories. For example, the product Helmet can be assigned to different membership degrees to each product category (e.g., 0.5/Component, 0.4/Accessories and 0.1/Wear). This degree can be altered depending on the session and promotion. However, a major gap in this research paper is that definition of weightages is not data-driven. Instead, it is subjective and was notable in the fuzzy membership functions. With regard to hierarchical dimensions, it should support type 2 Slowly Changing Dimension is discussed in the previous sections and chapters.

In Figure 6-10, table design for normalized structures is used in typical OLTP systems to improve data-write performance as OLTP has “many-writes-many-reads”.

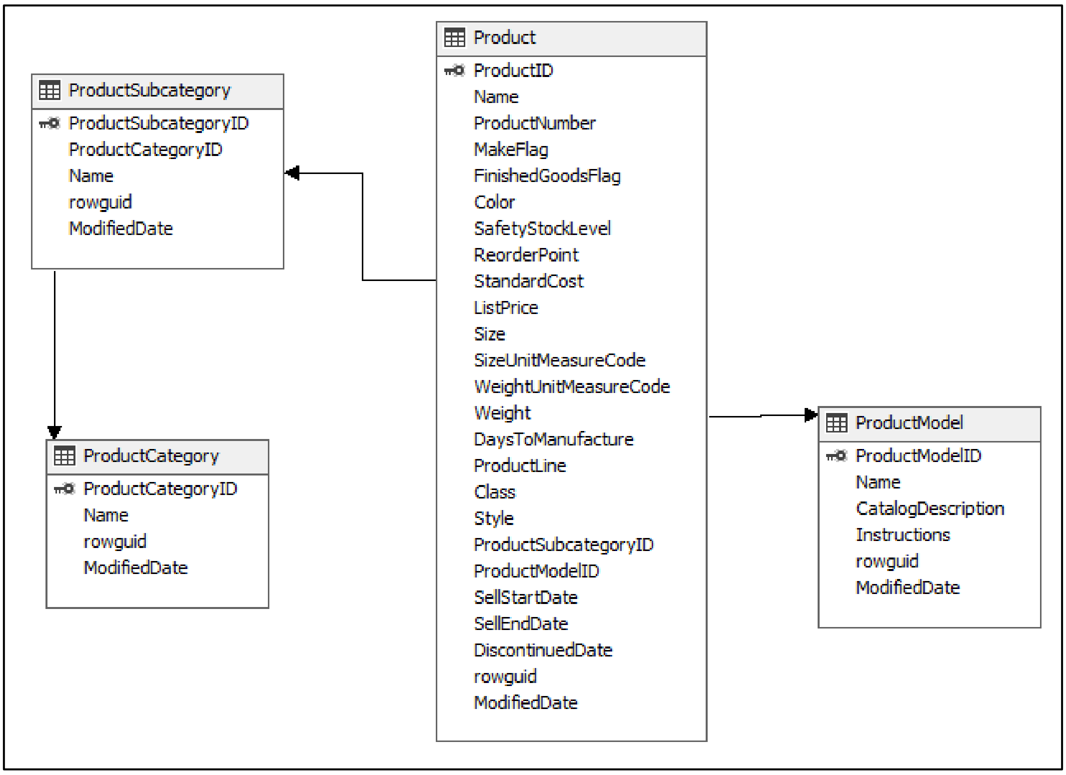


Figure 6-10: OLTP Design of Product and Related Tables

In typical data warehouse design, de-normalized structures are introduced [80] to improve read performance as data warehouses are heavily used for reading. In the above example, product tables and three other related tables, Product Category, Product Sub Category and Product Model are used to analyze product details. Considering the data warehouse, a de-normalized DimProduct table is designed as presented in Figure 6-11.

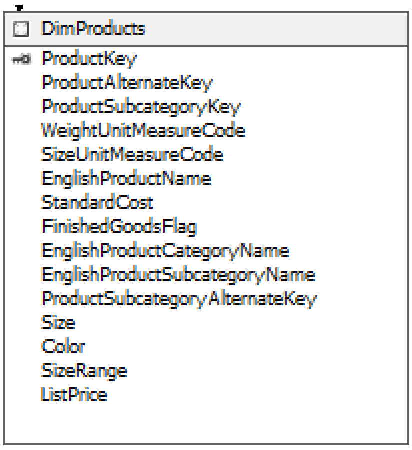


Figure 6-11: De-normalized DimProduct Table

In the above table structure, Product Category and Product Sub category are in a dimension table. Apart from the read performance improvement, there is another distinct advantage with the above design which is the ability to create hierarchies helpful for better analysis for the end-user.

As shown in Figure 6-12, DimProduct dimension table is included in Kimbal's method of star schema dimension modeling. Following data warehouse structures are retrieved from Microsoft SQL Server sample database AdventureWorksDW.

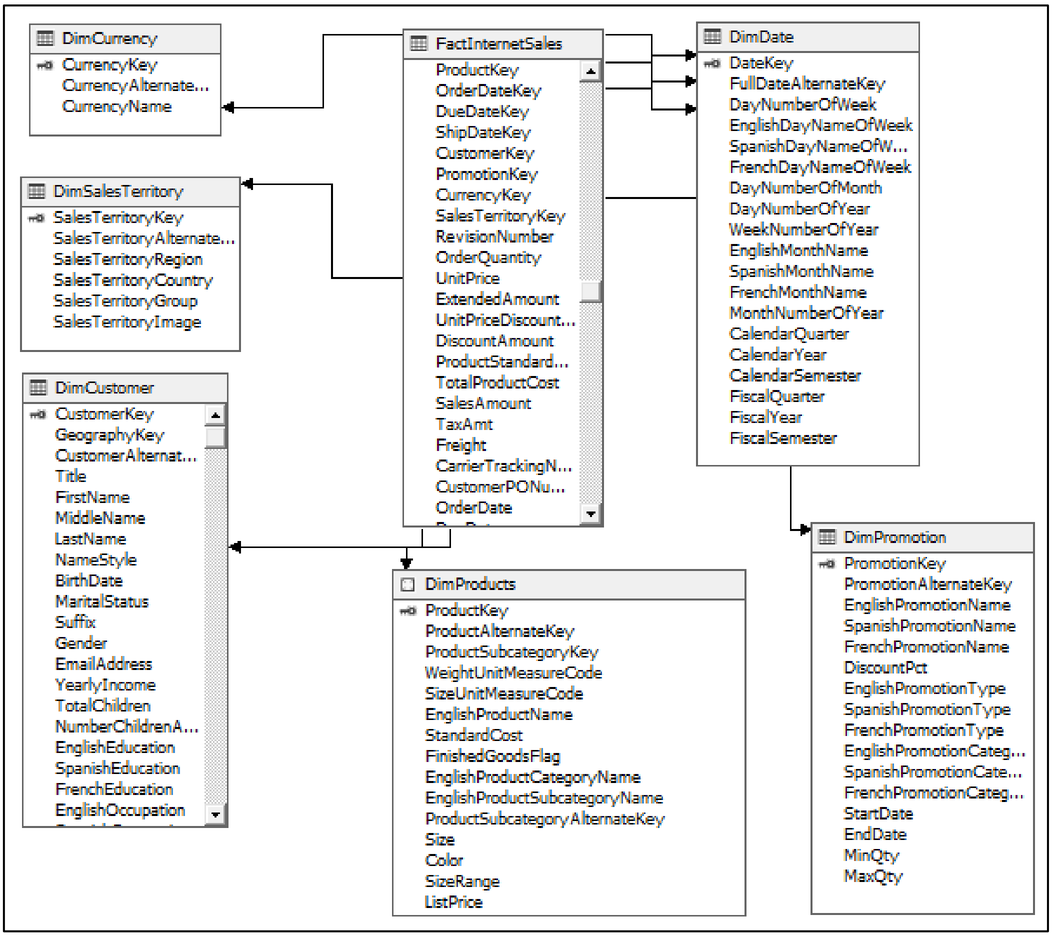


Figure 6-12: Start Schema Which Includes DimProducts

The main advantage of this design is that end-users will have the option of analyzing data in a hierarchical format. PivotTable is the most common technique used by the end-users of data warehouse. Figure 6-13 shows that usage of Microsoft Office Excel PivotTable to analyze data with hierarchical structures.

Row Labels	Count	Sales Amount	Tax Amt
Accessories			
Bike Racks			
Hitch Rack - 4-Bike	328	39,360.00	3,148.80
Bike Stands	**249**	**39,591.00**	**3,167.28**
Bottles and Cages			
Mountain Bottle Cage	2,025	20,229.75	1,618.38
Road Bottle Cage	1,712	15,390.88	1,231.27
Water Bottle - 30 oz.	4,244	21,177.56	1,694.20
Cleaners	**908**	**7,218.60**	**577.49**
Fenders	**2,121**	**46,619.58**	**3,729.57**
Helmets	**6,440**	**225,335.60**	**18,026.85**
Hydration Packs	**733**	**40,307.67**	**3,224.61**
Tires and Tubes	**17,332**	**245,529.32**	**19,642.35**
Bikes			
Mountain Bikes	**4,970**	**9,952,759.56**	**796,220.81**
Road Bikes	**8,068**	**14,520,584.04**	**1,161,646.73**
Touring Bikes	**2,167**	**3,844,801.05**	**307,584.08**
Clothing	**9,101**	**339,772.61**	**27,181.81**
Grand Total	60,398	29,358,677.22	2,348,694.23

Figure 6-13: Hierarchical Analysis for Data Warehouse using Pivot Table

In Figure 6-13, it can be observed that ***Hitch Rack – 4 – Bike*** is sub-categorized under ***Bike Racks*** and categorized under ***Accessories***. With data analysis, the end-user starts with Product Categories and then Product Sub Categories. With this analysis, every product is mapped to one subcategory and every product subcategory is mapped to the product category. When fuzziness is introduced, only one level of the hierarchy would simplify the implementation. This means that one product can be attached to multiple product categories while one product category can be categorized in multiple products. This means many-to-many relationships which are not accepted in the standard database design.

Consider D_i dimension which has n number of Categories (C_n) where n = 1,2,3… n with weightages of N_n where n = 1,2,3,4.. n, that is shown in the Table 6-10.

Table 6-10: Weightages of Dimension and Category

Dimension	Category	Weightages
D_i	C_1	N_1
D_i	C_2	N_2
D_i	C…	N.
D_i	C_n	N_n

In this, $\sum_1^n N_n = 1$

However, Product category cannot be maintained in the one-dimension table because it will create many-to-many relationships with the fact tables. Therefore, dimension and hierarchical category tables should be separated. If separated, table relationships will be as follows. However, it is important to specify as separating attributes of a dimension cannot create hierarchies. This implementation is shown in the Figure 6-14.

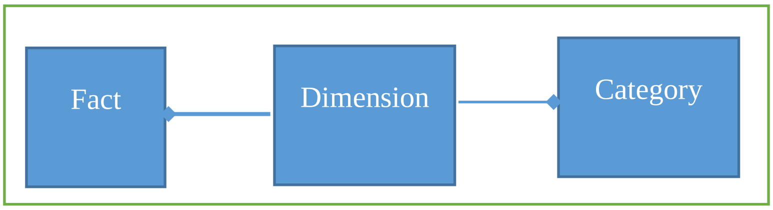


Figure 6-14: Separation of Category Table in Dimensional Modeling to Support Fuzzy Categorization

To implement this some real-world scenario is selected. In the Sri Lankan context, coconut oil is mainly used for cooking purposes. However, there are other purposes for which coconut oil is used. Surveys show that coconut oil is used for cosmetic, religious, medical and lubricant purposes and not for cooking purposes alone.

In this example, assume that contribution of weightages for categories Cooking, Cosmetic and other are 0.6, 0.3 and 0.1 respectively for coconut oil as shown in Equation 6.1.

$$\text{Categorization of Coconut Oil} = \frac{0.6}{\text{Cooking}} + \frac{0.3}{\text{Cosmetic}} + \frac{0.1}{\text{Other}}$$

Equation (6-1)

When this is converted to a dimension modeling, it is important to assume that these weightages can change over time. Therefore, type 2 aspects of the slowly changing dimension should be considered. Figure 6-15 is about selection of only the relevant columns for each dimension.

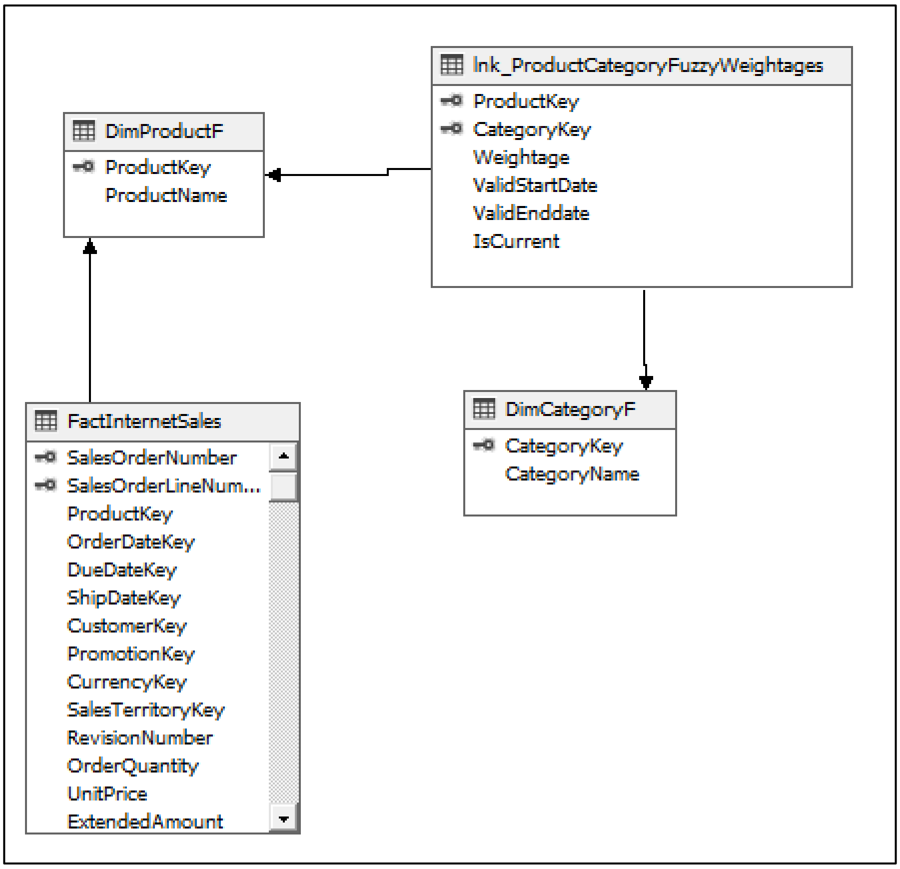


Figure 6-15: Separation of Category Table in Dimensional Modeling to Support Fuzzy Categorization

In Figure 6-15, lnk_ProductCategoryFuzzyWeitages table contains the weightages for different product categories. This link table or the bridge table has a one-to-many relationship with the Product and Category Dimension Tables. The fact table is linked with the product table. *lnk_ProductCategoryFuzzyWeitages* table follows a type 2 SCD which means that the historical aspect of the data is covered in this table as well. Sample queries used for fuzzy categorization are listed in Appendix

X: An important advantage of this fuzzy categorization is that this implementation does not require a fuzzy data warehouse. This can be implemented in the traditional data warehouse with slight changes to the existing data warehouse schema.

6.8 Fuzzy Membership Function for Fuzzy Categorization

In many researches, weightages of fuzzy categorization defined by the subjective method, raise concern regarding the validity of the weightages. With this research, it was decided to introduce data-driven techniques to define fuzzy membership function to calculate contribution weightages for categorization. To build the fuzzy membership function for fuzzy categorization the following four methods were introduced.

1. Subjective Method
2. Horizontal Method
3. Rank Based Horizontal Method
4. Pairwise Comparison Method

6.9 Subjective Method

As in the Fuzzy membership function, a trivial way of defining the fuzzy membership function for fuzzy categorization is a subjective method. In the case of the subjective method, the domain expert will define the weightages for each category.

In this example, assume that contribution of weightages for categories Cooking, Cosmetic and other are 0.6, 0.3 and 0.1 respectively for coconut oil as shown in Equation 6.2.

$$\text{Categorization of Coconut Oil} = \frac{0.6}{\text{Cooking}} + \frac{0.3}{\text{Cosmetic}} + \frac{0.1}{\text{Other}}$$

Equation (6-2)

In this method, the domain expert will change the weightages frequently as weightages tend to change with the time.

6.10 Normal Horizontal Method

As described in Appendix F: horizontal method is one of the main techniques to define fuzzy membership function. To implement a horizontal method a survey was carried out. In this example, coconut oil was used as an implementation as the preliminary study shows that coconut oil falls into multiple product categories such as cooking, medicine, cosmetic, lubricant, and religious in relation to the Sri Lankan context.

130 customers who bought coconut oil were interviewed in a semi-urban area of Sri Lanka for the given month and the results are described in Table 6-11.

Table 6-11: Survey Results for Coconut Oil for Different Categories

Category	Number of Customers	Percentage
Cooking	75	57.69
Cosmetic	35	26.92
Religious	12	9.23
Lubricant	4	3.08
Medicine	4	3.08

By using the horizontal method, the following fuzzy categorization membership function can be defined by rounding up the above percentage to two decimal points as shown in Equation 6-3.

$$\text{Categorization of Coconut Oil} = \frac{0.58}{\text{Cooking}} + \frac{0.27}{\text{Cosmetic}} + \frac{0.09}{\text{Religious}} + \frac{0.03}{\text{Lubricant}} + \frac{0.03}{\text{Medicine}}$$

Equation (6-3)

The major disadvantage of this technique is that there can be many categories with very tiny contributions. Therefore, the baseline number where all the tiny contributions can be summarised to the defined category, can be defined as “Other”.

Then the equation will change as Equation 6.4 combining the lubricant and medicine to the other category.

$$\text{Categorization of Coconut Oil} = \frac{0.58}{\text{Cooking}} + \frac{0.27}{\text{Cosmetic}} + \frac{0.09}{\text{Religious}} + \frac{0.06}{\text{Other}}$$

Equation (6-4)

Some categorizations will change due to the seasonality factor. For example, coconut oil is mostly used as a religious product during the close to "poya days" in Sri Lanka. Poya day is accepted as a holy day in Sri Lanka. Coconut oil is used extensively for religious purposes on "poya days". Therefore, Equation 6-5 will be effective during the close to "poya days" and just after the "poya days".

$$\text{Categorization of Coconut Oil} = \frac{0.51}{\text{Cooking}} + \frac{0.24}{\text{Religious}} + \frac{0.19}{\text{Cosmetic}} + \frac{0.06}{\text{Other}}$$

Equation (6-5)

This means that in case of coconut oil, two categorization fuzzy membership functions are available. However, depending on the seasonal factor, correct membership function should be selected. In this method, inter valued fuzzy sets of categories can be determined. Since this is a survey method, it is difficult to assign a value for contribution. Hence, fuzzy range is defined as described in Appendix F:

In fact, the horizontal method provides an inter-valued fuzzy set, the bounds of which are defined by Equation 6-6, below.

$$A(Xi) - \sqrt{\frac{A(Xi)(1-A(Xi))}{N}}, A(Xi) + \sqrt{\frac{A(Xi)(1-A(Xi))}{N}}$$

Equation (6-6)

Following calculations are done for all four categories as shown in Table 6-12.

Table 6-12: Standard Deviation and Range of Inter-Valued Fuzzy Set

Category	Contribution	Standard Deviation	Min Value of Inter-valued Fuzzy Set	Max Value of Inter-valued Fuzzy Set
Cooking	0.58	0.0433	0.5367	0.6233
Cosmetic	0.27	0.0389	0.2311	0.3089
Religious	0.09	0.0251	0.0649	0.1151
Other	0.06	0.0208	0.0392	0.0808

An inter-valued fuzzy set is shown in Figure 6-16.

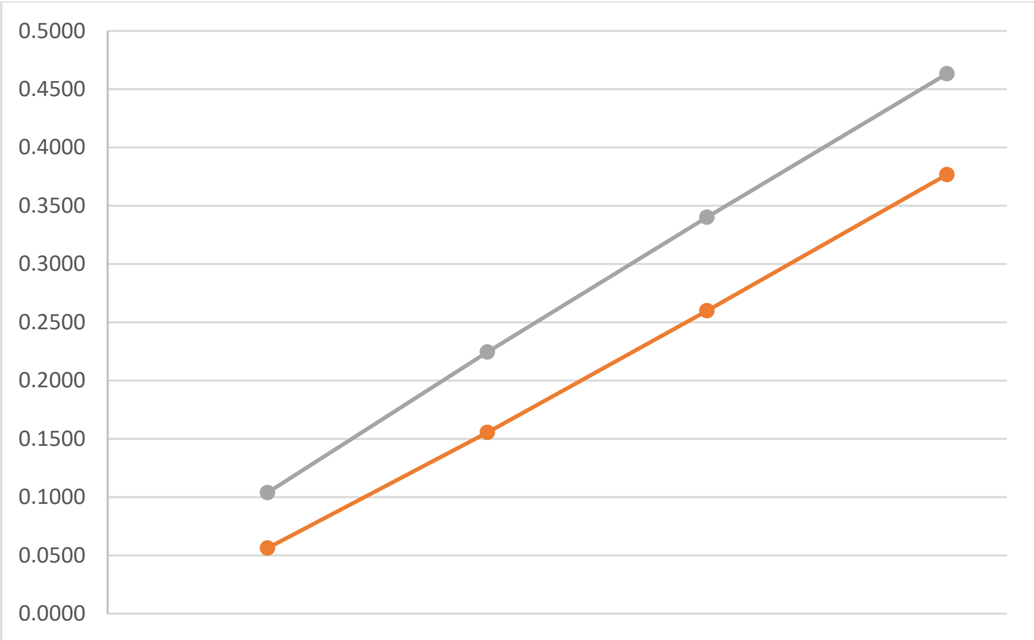


Figure 6-16: Inter Valued Fuzzy Sets for Categorization for Normal Horizontal Method

The major disadvantage of this technique is that this technique is time-consuming. Moreover, this method is implemented with the major assumption that every customer buys the product for one purpose which, in reality, is not the case.

6.11 Ranked Based Horizontal Method

As described in the 6.10 section, the major disadvantage of the previously discussed horizontal method is that it is implemented with the major assumption that every customer buys the product for one purpose, whereas, it is not so. To negate, the ranked based horizontal method is introduced. In this method, the customer has the option of selecting three items in ranking order. If it is ranked one, weightage will be three, if the rank is two, weightage will be two and if it is ranked at third weightage will be one. Customers have to select at least one. Selecting second and third options are optional.

Ranked weightage is calculated using Equation 6-7

$$\text{Ranked Weightage} = (3 * \text{Rank1}) + (2 * \text{Rank2}) + (1 * \text{Rank3}) \qquad \text{Equation (6-7)}$$

Following are the calculated ranked weightages for different product categories as shown in Table 6-13.

Table 6-13: Survey Results for Coconut Oil for Different Categories with Ranking Weightages

Category	**Rank 1**	**Rank 2**	**Rank 3**	**Total**	**Ranked Weightage**	**Contribution**
Cooking	75	35	12	122	307	0.42
Cosmetic	35	48	20	103	221	0.30
Religious	12	30	40	82	136	0.19
Lubricant	4	12	4	20	40	0.05
Medicine	4	3	6	13	24	0.03
Total	130	128	82	340	728	1

For example, the cooking category gets 307 ranked weightage from (3 * 75 + 2 * 35 + 1 * 12). Percentage will be calculated by dividing the 307 by the total Ranked Weightage of 728.

Using a similar technique as described in the normal horizontal method, fuzzy categorization function is built for the ranked horizontal method as shown below. This technique is compared with the previous method for the selected case coconut oil as shown in Figure 6-17.

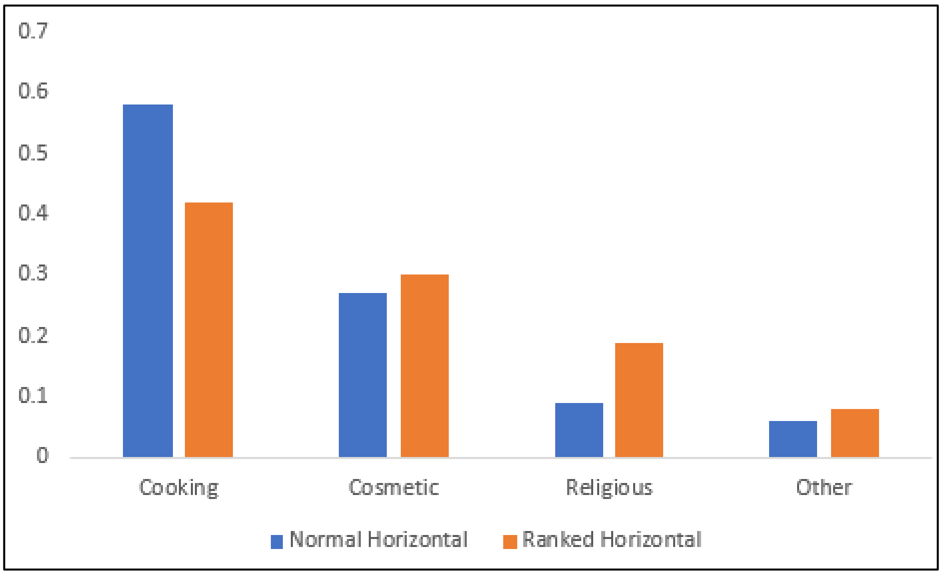


Figure 6-17: Comparison of Normal and Ranked Horizontal Method for Fuzzy Membership Function

Figure 6-17 shows that the introduction of ranked based horizontal method has not changed the contribution numbers drastically.

Following is the fuzzy membership function derived from ranked based horizontal method a shown in Equation 6-8.

$$\text{Categorization of Coconut Oil} = \frac{0.42}{\text{Cooking}} + \frac{0.30}{\text{Cosmetic}} + \frac{0.19}{\text{Religious}} + \frac{0.08}{\text{Other}}$$

Equation (6-8)

Similar to the normal horizontal method, seasonal fuzzy membership function can be introduced in the ranked horizontal method using the same technique. Similar to the Normal Horizontal method, the inter-valued fuzzy set can be defined in the Ranked based horizontal method as shown in Figure 6-18.

Figure 6-18: Inter Valued Fuzzy Sets for Categorization for Ranked Based Horizontal Method

In Figure 6-18, it is observed that ranked based horizontal method has a higher range as against the Normal Horizontal Method.

6.12 Pairwise Comparison Method

Pairwise comparison method as described in Appendix F: is one of the standard techniques for use with building fuzzy membership function. In pairwise comparison method, participants have to choose one category from a given pair. Pairwise comparison is done for the same example as shown in Table 6-14.

Table 6-14: Survey Results for Coconut Oil for Different Categories with Pairwise Comparison

Category	**Cooking**	**Cosmetic**	**Religious**	**Lubricant**	**Medicine**	**Total**	**Percentage**
Cooking		101	102	112	116	431	58%
Cosmetic	60		85	16	12	173	23%
Religious	24	17		60	19	120	16%
Lubricant	2	1	0		1	4	1%
Medicine	3	7	2	5		17	2%
Total						745	

In this survey, users are given the pairs of combinations. {Cooking, Cosmetic}, {Cooking, Religious}, {Cooking, Lubricant}, {Cooking, Medicine}, {Cosmetic, Religious} are the sample set of combinations. In the above example, there will be ten combinations. This implies an obvious disadvantage with a number of categories and the number of combinations is very high. In this example, 60 participants have chosen cosmetic over the category of cooking. As it was done for the previous two methods, very low numbers are combined to other categories and the following Fuzzy membership function as shown in Equation 6-8 was defined.

$$\text{Categorization of Coconut Oil} = \frac{0.58}{\text{Cooking}} + \frac{0.23}{\text{Cosmetic}} + \frac{0.16}{\text{Religious}} + \frac{0.03}{\text{Other}}$$

Equation (6-8)

The main disadvantage of this method is that many data points need to be gathered and it will not be practically easy to capture this much data from customers. Figure 6-19, shows the differences between the different method used to determine the fuzzy membership function.

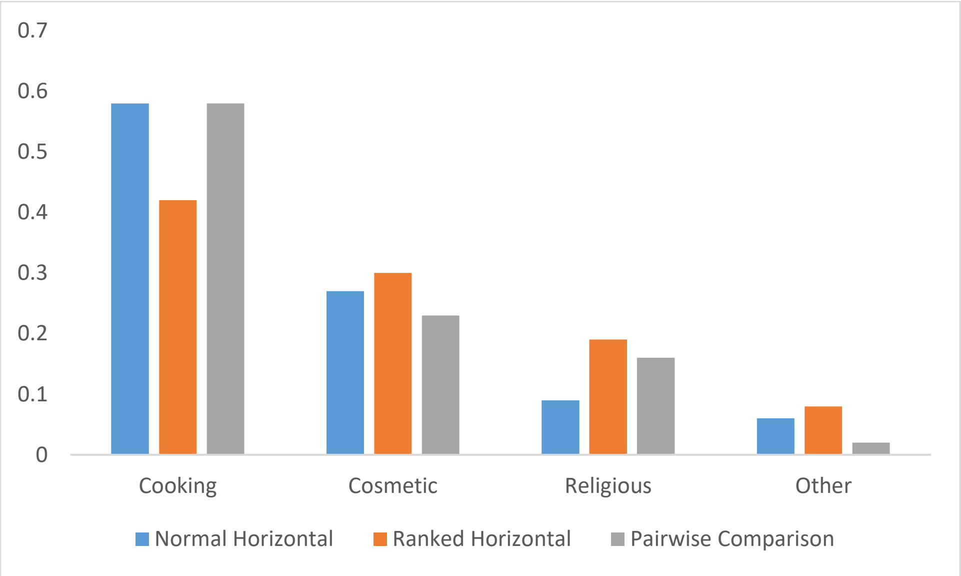


Figure 6-19: Comparison of Normal Horizontal, Ranked Horizontal Method and Pairwise Comparison for Fuzzy Membership Function

Figure 6-19 indicates that Normal Horizontal and Pairwise Comparison have similar values than that of the Ranked Horizontal Method.

6.13 Evaluation of Fuzzy Categorization

This method was evaluated for a supermarket in a semi-urban area in Sri Lanka. Same product, Coconut oil was used as the experimental object. During the comparison, currently, organizations are using only three categories for easy operations which are Cooking, Cosmetic and Other. However, in the data-driven method, there are four categories such as Cooking, Cosmetic, Religious and Other. For the comparison of these two data sets, Religious amount is also taken into the other category. It was found that total sales of Rs. 175,888 done for a given month on coconut oil and actual categorized weightages for Cooking, Cosmetic and Other has 0.6, 0.3 and 0.1 respectively. The same amount was distributed among multiple categories for three methods namely Normal Horizontal, Ranked based and PairWise comparison. All three methods are compared with the current actuals and the standard deviation was calculated to find out the closest option to the current actuals.

Table 6-15 shows a comparison of Normal Horizontal Method vs. the Actuals.

Table 6-15: Comparison of Actual and Normal Horizontal Method

Category	Actual	Normal Horizontal	Difference	Variance
Cooking	105,532.80	102,015.04	(3,517.76)	6,187,317.71
Cosmetic	52,766.40	47,489.76	(5,276.64)	13,921,464.84
Other	17,588.80	26,383.20	8,794.40	38,670,735.68
Total	175,888.00	175,888.00	-	58,779,518.23

The Standard deviation for Normal Horizontal method with the current actuals is 7,667.

Table 6-16 shows a comparison of Ranked based Method vs. the Actuals for the different categories.

Table 6-16: Comparison of Actual and Ranked Based Method

Category	Actual	Ranked Based	Difference	Variance
Cooking	105,532.80	73872.96	(31659.84)	501,172,734.41
Cosmetic	52,766.40	52766.40	0	-
Other	17,588.80	49248.64	31659.84	501,172,734.41
Total	175,888.00	175888		1,002,345,468.83

The standard deviation for this method with the current actuals is 31,660.

Table 6-17 shows a comparison of Pair-Wise method vs. the Actuals for the different categories.

Table 6-17: Comparison of Actual and Pairwise Comparison Method

	Actual	Pair Wise	Difference	Variance
Cooking	105,532.80	102,015.04	(3,517.76)	6,187,317.71
Cosmetic	52,766.40	40,454.24	(12,312.16)	75,794,641.93
Other	17,588.80	33,418.72	15,829.92	125,293,183.60
	175,888.00	175,888.00	-	207,275,143.24

The standard deviation for this method with the current actuals is 14,397. The standard deviation for all three methods is compared in Figure 6-20.

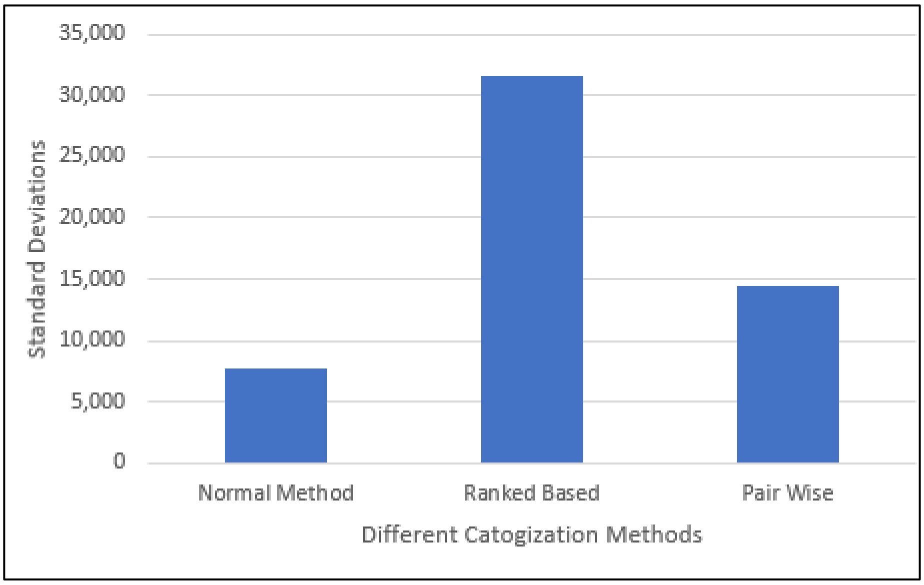


Figure 6-20: Comparison of the Standard Deviations for Normal Horizontal, Ranked Horizontal Method and Pairwise Comparison for Fuzzy Membership Function

As shown in the Figure 6-20, the Normal Method is closer to coconut oil categorization for the given month. Similarly, this method can be used to evaluate the correct method for each product.

6.14 Summary

Fact tables and dimension tables are coherent components of the data warehousing system – legacy or fuzzy model. This chapter, gives a detailed discussion regarding what was done for design methodology for fuzzy fact and dimension tables in the fuzzy data warehouse.

In fact tables, range values are the main possible candidates for fuzzy implementation. Every range is considered to be a crisp set in the standard implementation of data warehouse leaving few limitations in analyses. Introduction of fuzzy set ranges can help analyze with more accuracy. The introduction of linguistic variables, as discussed in the previous chapter, favours more analyses.

Type 1 and Type 2 Slowly Changing Dimensions (SCD) are common implementations in data warehousing, thus in this chapter design concepts for slowly changing dimensions are discussed in detail to adopt historical aspects of data warehouse.

Hierarchical categorization is one of the better ways of analyzing data in data warehousing. Three data driven methods, Normal Horizontal, Ranked Based Horizontal Method and Pairwise comparison method are used to identify the categorization weightages which is the fuzzy membership function eventually. The subjective method is introduced to facilitate practical needs of the business users. This research has introduced the evaluation technique to identify the correct method for each product categorization. Depending on the parameters such as data set and time period a user can select the correct method out of the three methods proposed. An essential aspect is that, to implement the fuzzy categorization, it does not require the data warehouse to be a fuzzy data warehouse Instead this can be implemented in any data warehouse with hierarchical categorizations.

This chapter and the previous chapter conclude the discussion of the modeling fuzzy data warehouse including fuzzy membership functions. The two chapters conclude the design and modeling aspects of fuzzy data warehosue.

The fuzzy data warehosue modeling completes a high percentage of data warehouse implementation when the end to end aspects are considered. The chapters to come are about Extract-Transform-Load part of the fuzzy data warehouse as ETL considers the techniques for bringing in data from hetrougenous data source to the fuzzy data warehouses.

Chapter 7
FUZZY EXTRACT-TRANSFORM-LOAD

7.1 Introduction

ETL (Extract-Transform-Load) technology is used for data to be pulled from heterogeneous data sources including the legacy system environments as well as from the non-relational data sources. ETL technology has the capacity to transform collected data into the data warehouse. The ETL component performs multiple functions, such as,

- Logical translation of data
- Data type and data conversion
- Transform from one DBMS to another DBMS
- Construction of default values when desired
- Summarization of data
- Merging of records
- Sending notifications

The core of ETL is that data enters the ETL jumble as application data and exists as data warehouse data. In the case of extraction of data, there is nothing special in the fuzzy data warehouse to the conventional data warehouses. With respect to Loading to data warehousing, there is again nothing much to discuss as related to fuzzy data warehouses. These facts simply mean that with regard to ETL, only the transformation layer or the design of the fuzzy data warehouse needs discussion. This research, aimed at the investigation and development of technology for the fuzzy data warehouse, requires an evaluation of ETL implementation for a fuzzy data warehouse. Therefore, this chapter discusses in detail the ETL of fuzzy data warehousing.

7.2 Transformation Tasks

As described in Appendix X:, there are 32 subsystems in ETL as indicated by Ralph Kimbal. However, all these subsystems do not require an alteration to implement fuzzy data warehouse as most of these sub-systems are related to extraction and loading layers. The maintenance of sub-systems apparently, does not require any modification to classical data warehouse implementation. Loading tasks are also not considered as fuzzy tasks since they are mainly used to write data to the database and perform other related activities such as notification and auditing. In this sense, transformation tasks of ETL are the tasks which relevant to fuzzy data warehouse implementation. In the sub-systems discussed in Appendix Y: Slowly Changing Dimension, Special Dimension, Fact Tables are the sub-systems which should be considered for fuzzy data warehousing.

7.3 Fuzzy Dimensions

In Chapter 7: Design Methodology for Fact and Dimension Tables in Fuzzy Data Warehouse, designing of fact and dimension tables are discussed in detail with respect to fuzzy data warehouse. For this purpose, fuzzy contributions are pre-calculated for each value and stored in the FuzzyContribution table as presented in the table scheme is in Figure 7-1.

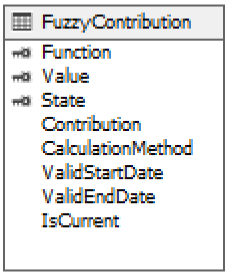


Figure 7-1: Table Schema of Fuzzy Contribution

The initial ETL task is to generate data for the table shown in Figure 7-1 and the next task is to retrieve data from this table for the analysis. The first task is not a very frequent task and fuzzy contributions do not change rapidly. Even if it changes, maximum change frequency will be daily. In this sense, the loading of Fuzzy Contributiontable can be executed on a daily basis. Since this script is listed in Appendix O: executing this will not be a great challenge. However, execution should be done after all the dimension load is completed as it needs a completed dimension load. Type of these dimensions are handled to the way that it is typically handled by the classical dimension which includes new record and updating or end dating the previous records by means of a date column [23], and a status column which as discussed in detail at Appendix B: Dimension Tables. In the case of Type 1 dimension, no additional steps need to be taken for fuzzy dimensions.

ETL configurations need to be considered with fuzzy categorization dimension as well. Figure 7-2 shows the Fuzzy design for hierarchical categorization which was discussed in detail in the previous chapter.

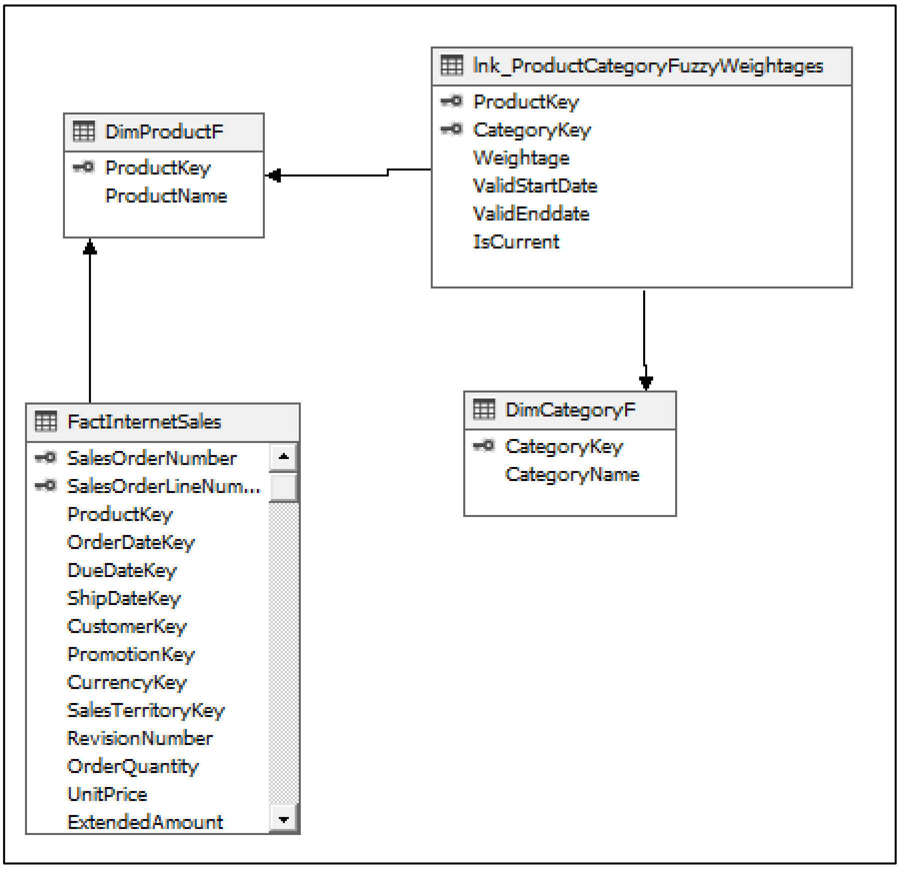


Figure 7-2: Separation of Category Table in Dimensional Modeling to Support Fuzzy Categorization

In the above table schema design shown in Figure 7-2, *lnk_ProductCategoryFuzzyWeightages* is the table which consists of hierarchical weightages. This is the table which should consider the fuzzy ETL. As in the previous case, this table will not be updated on a daily basis as fuzzy categories are even updated at a lesser frequency than the fuzzy dimensions which were discussed before. Like in the previous incident, this table also should be updated after loading the related dimensions. In this example, it would be a category and product dimension. It is important to note that there will be multiple records in the link or bridge table which mean the one-to-many relationship is preserved between the link table and the other related dimension tables.

As discussed in the section 6.8, there are four methods to update a fuzzy hierarchical categorization table. In the case of pre-defined weightages, there is no ETL to be considered. However, in the case of the other three methods, ETL should be done using the survey data. This survey data is kept outside the fuzzy data warehouse as it can be maintained by different resources. Figure 7-3 shows the physical separation of the fuzzy data warehouse and the survey data and data movement will be up to down.

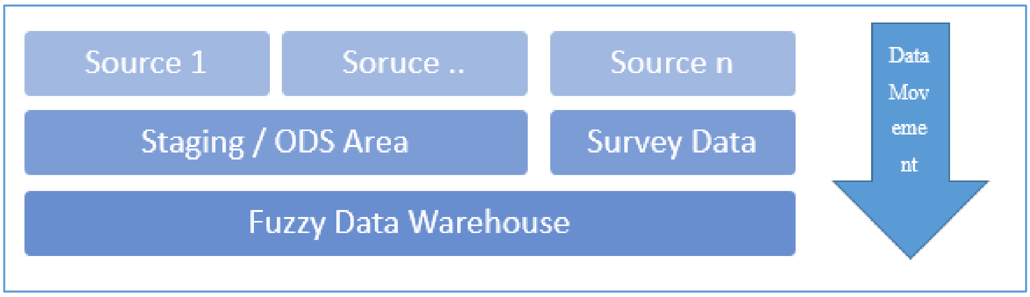


Figure 7-3: Physical Separation of Survey Data and Fuzzy Data Warehouse

The main benefit of this separation is that survey data will be managed separately typically by different teams. It is updated on a different frequency. Therefore, it is recommended to separate survey data physically.

7.4 Fuzzy Fact Measures

Design for Fact measures for the fuzzy data warehouse was discussed in detail in Chapter 7. In summary, the approach is to move this to a range dimension and range dimension is configured for the fuzzy data warehouse. Fuzzy contributions for different attributes are pre-calculated with different weightages to improve the performance. Therefore, there is no direct impact to fact load in the fuzzy data warehouse. As such, ETL is considered to be independent to fact load. The only concern would be the timing of the contribution load. Fuzzy contribution load should be performed before the fact load. There can be cases where the contribution values are not referenced in the fact table and generating the contribution value will be unnecessary. However, electing these specific values will add more load to the ETL and minimize by allocating all possible values.

The next factor is that these range dimension can be type 2 slowly changing dimensions. As discussed before, classical type 2 slowly changing dimension implementation techniques can be used for implementation.

7.5 Fuzzy Transformation Tasks

In the ETL, there are transformation tasks to tie the gap between the different types of data sources and the data warehouse schema. Data type conversions, data merging, data union, data filtering, data splitting, Aggregating, and Lookup are the main transformations. Out of these transformation tasks, the lookup can be considered fuzzified. The classical lookup will check for exact matches. However, due to various data entry and collection issues and other practical situations, there can be cases where the exact match may not be possible. For example, there can be controlled characters and other punctuations in the data source which will not be matched with the dimension. In this situation, fuzzy lookups can be utilized. In classical lookups, there are three parameters: the source column, the target column, and the retrievable column. In the classical lookups, source and target columns should be matched. However, with most practical scenarios, source and target columns appear to mis-match. Hence, fuzzy lookup techniques should be used. In this technique, the output can be multiple rows with similarity values and confidence values and the end-user has to define what needs to be selected.

- Similarity: When Fuzzy lookup is performed, Similarity indicates the overall likeness of the record found.
- Confidence: Confidenec indicates the how close the match record with respect to the non-marched records.

With regrad to this research, in most cases, Employee dimension load will need fuzzy lookups due to differences in the way employees' names are written.

7.6 Summary

In this minor section, ETL aspects of the fuzzydata warehouse are considered. It is very clear that there are no major changes for ETL with respect to the fuzzy data warehouse. However, it is considered that the timing of the contribution load should be done before the fuzzy dimensions and fuzzy facts. In the case of fuzzy categorization, weightages should be done after the dimension load. Besides, handling of type 2 slowly changing dimension shows no difference in the fuzzy data warehouseas compared to the classical data warehouse.

With regard to survey data, it is recommended to physically separate survey data so that it can be handled separately. Fuzzy lookups and Fuzzy grouping are also considered as they will be needed for perfect lookups during the transformation phase of the ETLs. The fuzzy grouping transformation is more similar to the Fuzzy lookup as it uses the identical method to find matches. This process is denoted as de-duplication technique which mainly uses data cleansing in data warehousing.

OLAP cubes are important componets in data warehousing as OLAP cubes are used to analyse data on a large scale as high performance. In the next chapter, fuzzy OLAP cube implementation with respect to the fuzzy data warehouse covers the analysis side of the fuzzy data warehouse.

Chapter 8
FUZZY OLAP CUBE

8.1 Introduction

An OLAP cube is a multi-dimensional database that is improved for data warehouse and OLAP applications for data analytics. An OLAP cube is a technique of storing data in a multidimensional format, usually for reporting and analytical activities. OLAP Cube is used to analyze data warehouse as classical database technology and will not provide an adequate performance benefit for analysis. In OLAP cubes, measures are categorized by dimensions. As described in Chapter 2, OLAP cube is mainly used for slicing and dicing which will enhance the options of analyzing. In this chapter, a detailed discussion is done to accommodate an OLAP cube in the fuzzy data warehouse. The main operations of OLAP cube, are slicing, dicing, roll-up, and drill-down. They are discussed with respect to the fuzzy data warehouse.

8.2 Classical OLAP Cube

Figure 8-1 shows the classical implementation of OLAP cube for the data warehouse in the plantation attendance example, discussed in this research.

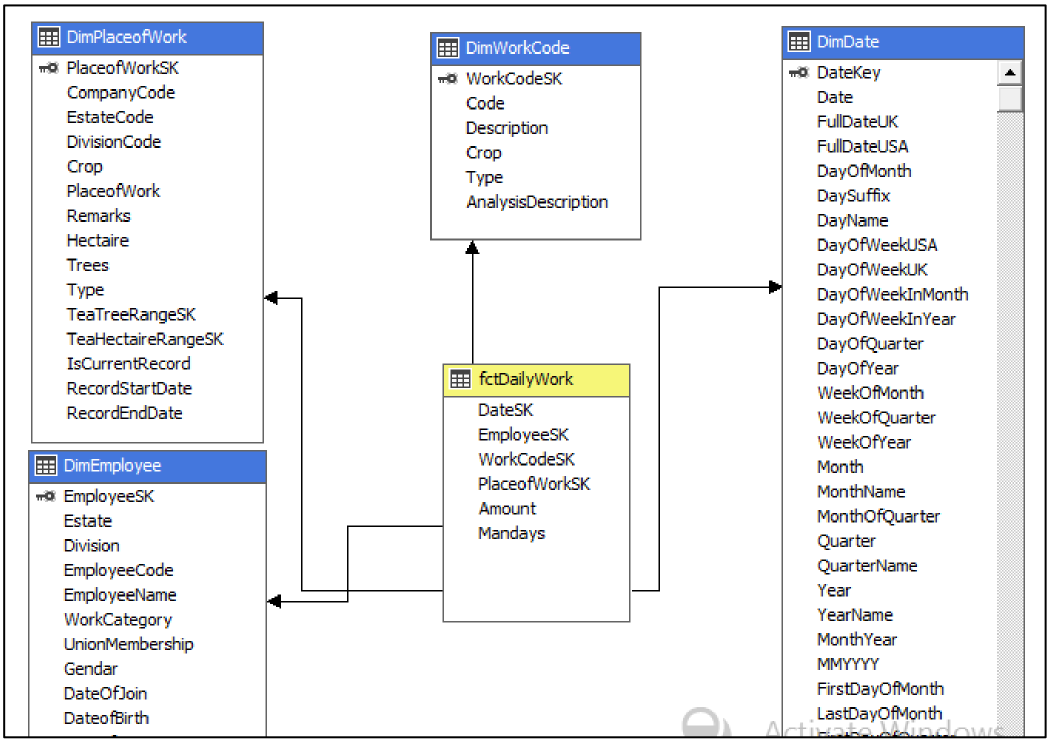


Figure 8-1: Star-Schema for fctDailyWork

In the above OLAP Cube, fctDailywork is the fact table and Amount and the workdays are measure columns. This fact table is linked to DimEmployee, DimDate, DimWorkCode and DimPlaceofWork dimensions. Figure 8-1 shows the star schema diagram in which all dimensions have a direct connection with the fact table.

Following is the cube script for the specified cube discussed in the Figure 8-1.

compute cube DailyWork as

select Date, WorkCode, Employee, PlaceofWork,

sum(Mandays),

from FctDailyWork

cube by Date, WorkCode, Employee, PlaceofWork

In the above cube, Mandays measure is created along with the Date, Workcode, Employee and Place of work as the dimensions. This classical OLAP was built by using Microsoft SQL Server Analysis Service 2016 where the previously referred research papers have used Microsoft SQL Server Analysis Service 2000. Figure 8-2 indicates the OLAP cube design by using Microsoft SQL Server Analysis Service 2016.

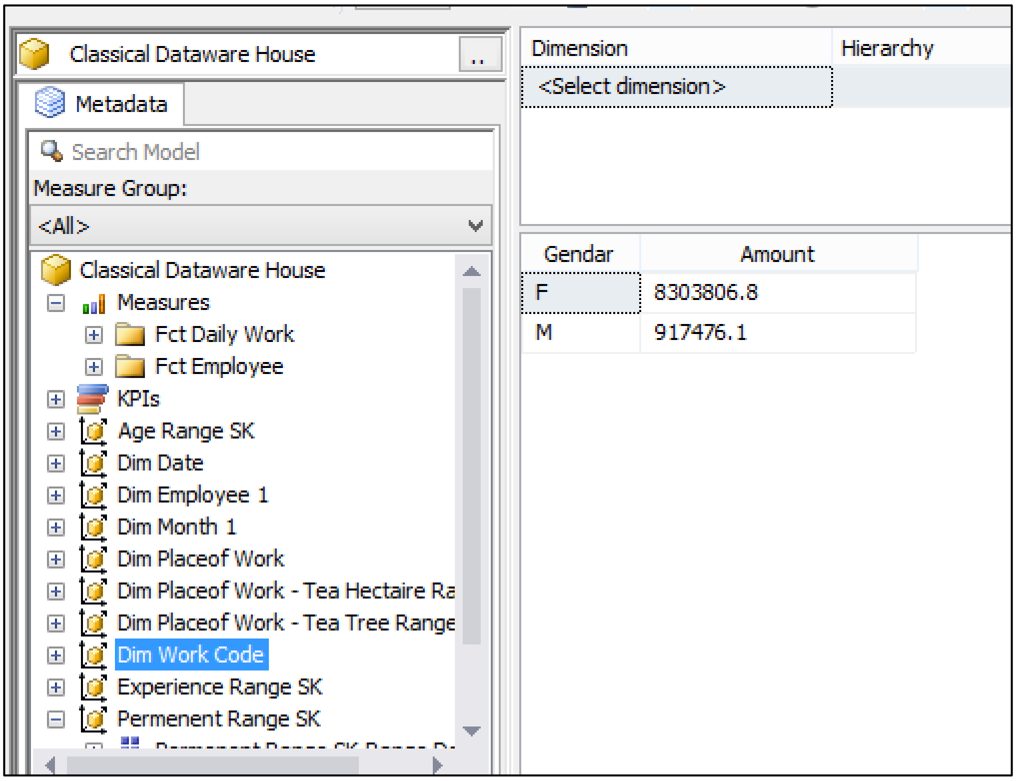


Figure 8-2: Implementation of Classical OLAP Cubes using SQL Server Analysis Service 2016

Figure 8-2 indicates the amount of daily wages by gender. There are measure groups and several dimension tables as shown in the above figure. This was built by the latest Microsoft SQL Server version of 2016 where many research papers have used Microsoft SQL Server 2000 and Microsoft SQL Server 2005, very legacy technical tools. Similarly, An OLAP cube can be built with multiple measures and multiple fact tables using Microsoft SQL Server Analysis Service.

8.3 Basic Fuzzy OLAP Cubes

A fuzzy OLAP cube is a data cube in which the facts and one or more dimensions are fuzzified. A Multi-Dimensional Fuzzy OLAP Cube is defined as a mapping function that is described in the Equation 8-1.

$$\boldsymbol{F\ (FC): domain\ (FD_1)\ \ X\ \ domain\ (FD_2)\ X\ ....\ ..\ ..\ X\ domain\ (FD_n) \dashrightarrow \{<L, [O,l]>\}}$$

Equation (8-1)

Instead of the general fact table, query Script 1 in Appendix Y: is used as a named query. In this OLAP cube, only Employee dimension is considered as fuzzy dimension and age as the fuzzy attributes whereas all the other dimensions and other attributes are classical and standard. In the above mention query, to capture the age of the employee at the start of the month is calculated. Therefore, this query is complex in nature as it has to join several dimension tables and the sample query output for an employee is given in Table 8-1.

Table 8-1: Query Output for Named Query in Fuzzy OLAP Cube

Date SK	Work Code SK	Place Of Work SK	Amount	Mandays	State	fAmount	fMandays
20121211	90	16	20.00	1.00	High	20.0000	1.0000
20130805	90	14	26.00	1.00	High	13.0702	0.5027
20130805	90	14	26.00	1.00	Medium	12.9298	0.4973
20140916	90	10	14.00	1.00	High	14.0000	1.0000
20130905	90	14	20.00	1.00	High	10.0540	0.5027
20130905	90	14	20.00	1.00	Medium	9.9460	0.4973
20120925	90	5	15.00	1.00	High	15.0000	1.0000
20131122	90	6	21.00	1.00	High	7.2387	0.3447
20131122	90	6	21.00	1.00	Medium	13.7613	0.6553
20130916	90	150	20.00	1.00	High	20.0000	1.0000

Due to the difficulty of the query, OLAP cube process will take some time to process. Further, in this OLAP cube, measure values are weightage as displayed in the above query. The next step is to build the query into multiple cubes so that multiple cubes are built. Data will be physically separated depending on the Fuzzy Dimension (FD) attributes. With the previous example, Fuzzy Cubes (FC) were built depending on the state of the fuzzy attribute. As shown in Figure 8-3, the OLAP cube is divided into three OLAP sub-cubes on the No of trees states, Low, Middle, and High.

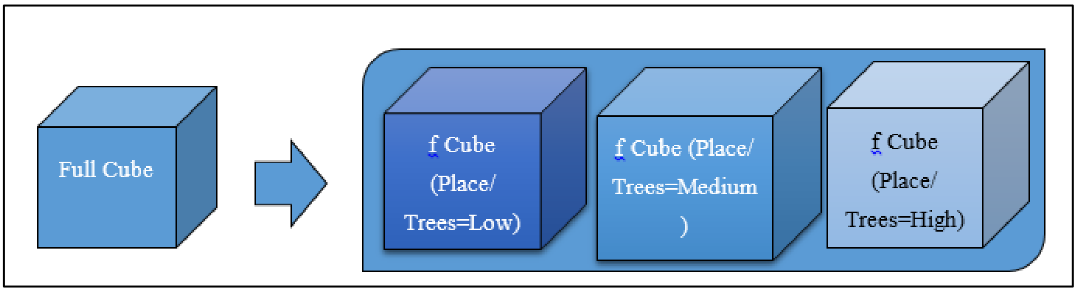


Figure 8-3: Physical Separations of Cubes depending on Fuzzy Dimension

The major drawback of the above method is the maintenance of many cubes with a large number of states in fuzzy attributes. Another limitation is that in the case of the existence of multiple fuzzy dimension attributes in the cube, the above method will not be feasible as the partition unit is not clear.

Another method would be, dividing measures by using fuzzy attributes as shown in the Equation 8.2.

$$\boldsymbol{m = m_1 + m_2 + \ldots + m_n}$$

$$\boldsymbol{m_1 = m * fc_1 / (fc_1 + f_2 + \ldots + f_n)}$$

fc_1 = Fuzzy contribution for attribute 1

Where m is the main measures and $m_1 \ldots m_n$ are the fuzzy attributes wise measure values.

Equation (8-2)

In the case of the above example, Measure Amount is divided for fuzzy attribute Number of trees. The query is listed under Appendix Z: Script 2 and output of that query is displayed in Table 8-2.

Table 8-2: Query Output for Named Query in Fuzzy OLAP Cube for One OLAP Cube and Single Measure

Date SK	**Employee SK**	**Work Code SK**	**Place of Work SK**	**Trees (Low) Amount**	**Trees (Medium) Amount**	**Trees (High) Amount**	**Amount**
20131008	1248	90	11	0.00	0.00	18.00	18.00
20150504	1771	90	58	0.00	1.75	18.25	20.00
20141014	609	90	69	0.00	16.47	7.53	24.00
20150729	643	90	66	0.00	9.03	2.97	12.00
20121116	268	90	6	0.00	24.90	13.10	38.00
20120427	340	90	29	0.00	0.00	35.00	35.00
20150430	336	90	52	0.00	9.44	14.56	24.00
20130423	365	90	52	0.00	7.47	11.53	19.00

The amount is the total measure and it can be observed that three other measures are added to facilitate the fuzzy states of a number of trees. Depending on the fuzzy contribution towards the fuzzy attributes, amount measure is divided to the relevant fuzzy measures.

Figure 8-4 shows the star schema for the above example which will be used to process fuzzy OLAP Cube.

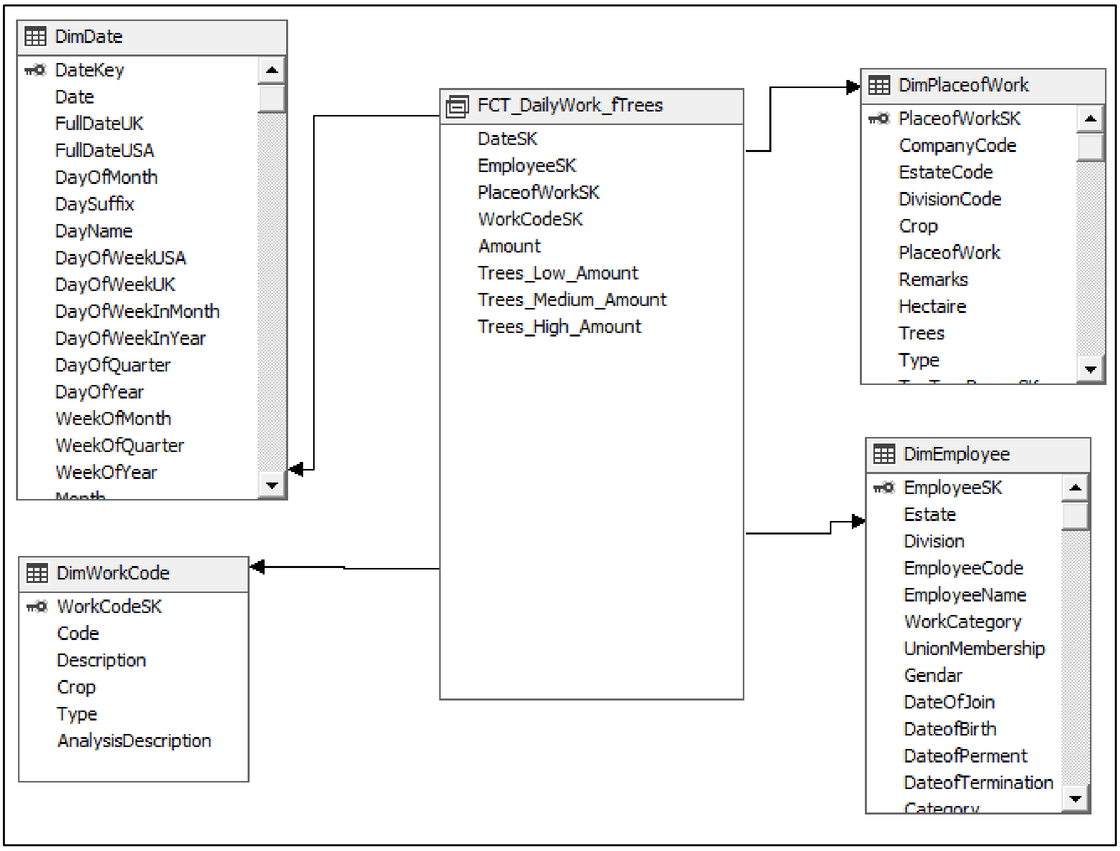


Figure 8-4: Star Schema Fuzzy OLAP Cube for Fuzzy Attributes Number of Trees for Single Measure

In the star schema shown in the Figure 8-4, additional three measures are included as discussed in the previous equation. Other dimensions are linked to the fuzzy included fact table as before. The major drawback of this method is that this model is unable to cater for multiple measures as the pivot is unable to cater for multiple measures.

To facilitate multiple measures, sub-queries were built and joined together. This query is listed in Appendix Z: Script 3: and its output is listed in Table 8-3.

Table 8-3: Query Output for Named Query in Fuzzy OLAP Cube for One OLAP Cube and Multiple Measures

Columns	**Row1**	**Row2**	**Row3**	**Row3**	**Row4**
DateSk	20121210	20140627	20140428	20130401	20150227
EmployeeSK	1926	1638	2091	2092	924
WorkCodeSK	90	90	90	90	90
PlaceofWorkSK	22	34	68	68	24
Trees_Low_ Mandays	0.00	0.00	0.00	0.00	0.00
Trees_Medium_	0.00	0.56	0.00	0.00	0.71

Columns	Row1	Row2	Row3	Row3	Row4
Mandays					
Trees_High_ Mandays	1.00	0.44	1.00	0.50	0.29
Mandays	1.00	1.00	1.00	0.50	1.00
Trees_Low_ Amount	0.00	0.00	0.00	0.00	0.00
Trees_Medium_ Amount	0.00	12.81	0.00	0.00	10.71
Trees_High_ Amount	29.00	10.19	18.00	12.00	4.29
Amount	29.00	23.00	18.00	12.00	15.00

In the above fact table the standard table format is pivoted to provide more readability as this fact table has a large number of measures unlikely to be displayed in standard tabular format. Similar to the previous example, Amount and the man-days are weighted according to the contribution of fuzzy attributes.

Figure 8-5 shows the star-schema for the above model.

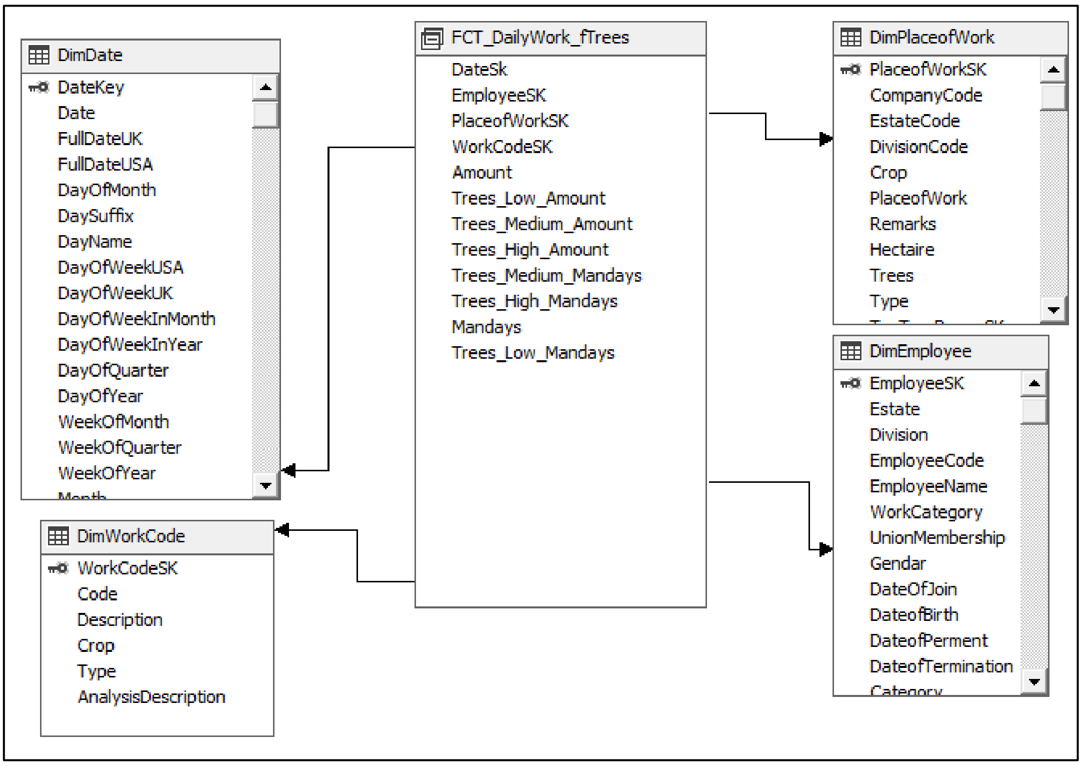


Figure 8-5: Star Schema Fuzzy OLAP Cube for Fuzzy Attributes Number of Trees for Multiple Measures

There are no changes to the other dimensions and only modification is the fact table where additional measures are added. In this model, fuzzy measures are added as a direct measure to the fact table.

The disadvantage of these models is, the inability to scale as fuzzy attributes are increased. For example, in introducing an additional state, the cube has to be modified and reprocessed. Further, a performance impact is likely when the number of fuzzy attributes increases. However, considering the fact that dynamic states will have a performance impact on users, it is recommended to utilize the above technique.

8.4 Slicing

As indicated before, slicing operation in OLAP cube is referred to as decreasing the number of dimensions by taking a projection of facts on a proper subset of dimensions and for nominated values of dimensions that are being released from the OLAP Cube.

Figure 8-6 displays the 1st example for Slicing in Fuzzy Cube which is discussed OLAP Fuzzy cube.

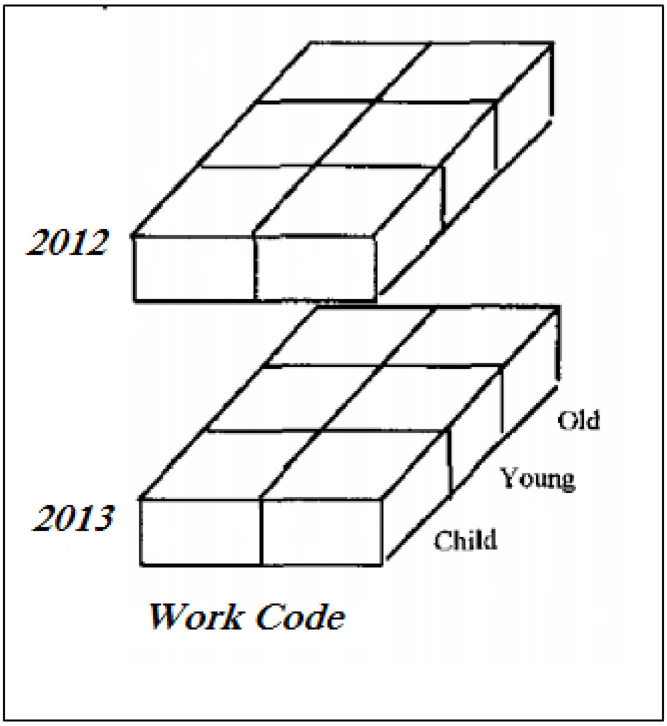


Figure 8-6: Slicing Operations for Fuzzy Cube – Example 1

Figure 8-6 can be denoted by FC^3, WorkCode, Age, Year =2012. In this Fuzzy Cube, slicing is done on Workcode and Year where Age is the fuzzified dimension.

Figure 8-7 displays the 2nd example for Slicing in Fuzzy Cube which is the previously discussed OLAP Fuzzy cube. This is can be denoted by FC^3, WorkCode, Year, Age =”Child”. In this Fuzzy Cube, slicing is done on Workcode and Year where Age is the fuzzified dimension,

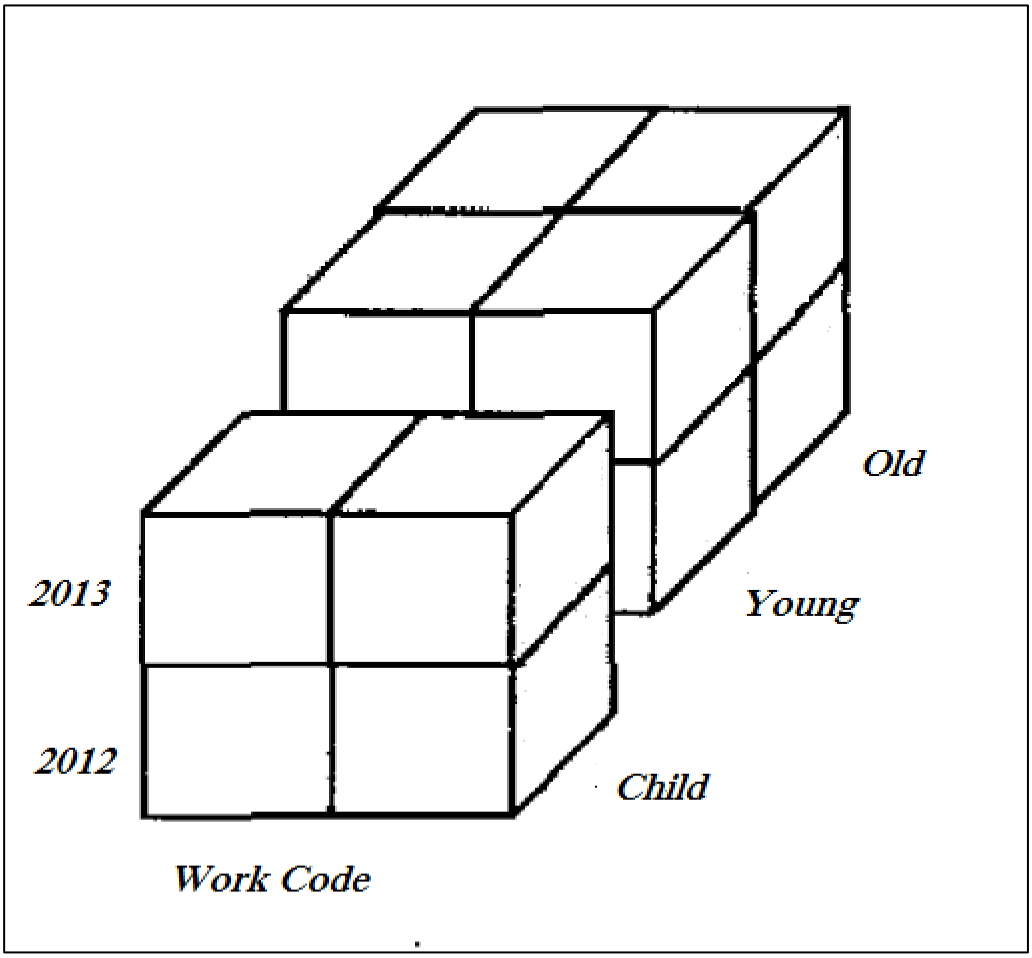


Figure 8-7: Slicing Operations for Fuzzy Cube – Example 2

From the above examples described in Figure 8-6 and Figure 8-7, it can be observed that slicing of Fuzzy OLAP cube can be done on either on fuzzified dimension attribute or on non-fuzzified dimension attribute which provides the flexibility for the end users to analyse data.

8.5 Dicing

Dice operation used to select condition on one dimension or select condition on more than one dimension where the logical operator's connection conditions on different dimensions.

The dice operation can be defined as shown in Equation 8-4.

$$\Sigma\,(FC^n, P) = FC'^n$$

Equation (8-4)

Where P is a predicate which can be applied on the OLAP cube for execution of the operation.

Figure 8-8 displays the fuzzy OLAP cube displayed in a dice operation for a given query predicate "age = old OR Age = middle".

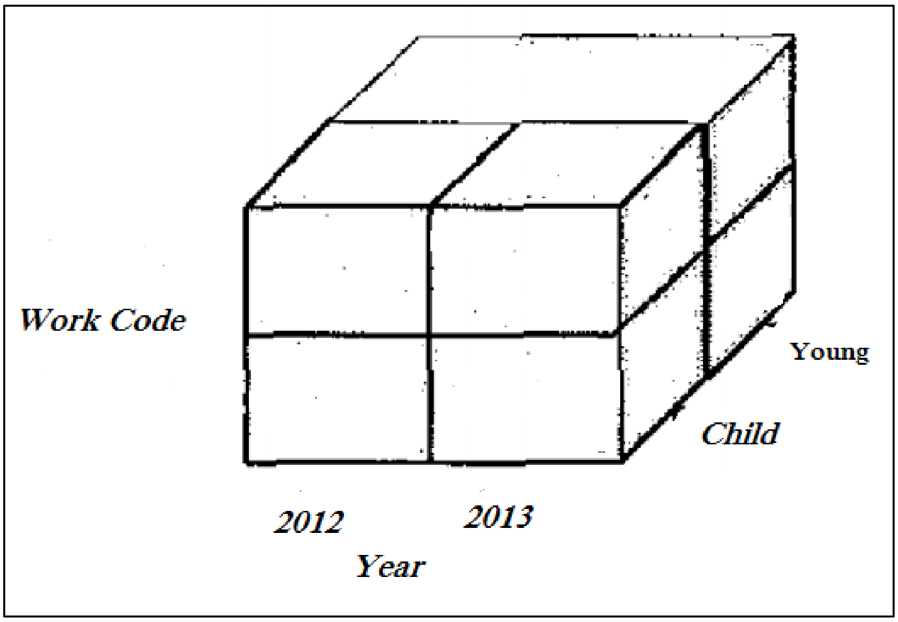


Figure 8-8: Dicing Operations for Fuzzy Cube

8.6 Roll-up

The roll-up operation is used to taking the existing aggregation level of fact values and performing an additional aggregation on one or more of the dimensions. The attribute hierarchy is a prerequisite for doing a roll-up operation.

For an example, if a user wants to perform a roll-up across the location dimension for the query *age = child AND Year = 2012* the relevant corresponding function can be defined as shown in Equation 8-5.

$F(FC^2)$ (Child, 2012) = { < Good, (0.3 + 0.5 /2> , <Average, (0.3+0.4) /2>, < Poor, (0.1 + 0.4)/2> }

= {<Good>,0.4, <Average,0.35>, <poor, 0.25>

Equation (8-5)

Final Result will be "good" as it has a higher value. The resulting measure will consist membership value in each fuzzy set so that the result of aggregation after the roll-up operation will be placed in the set where it has the maximum membership value.

8.7 Drill-Down

The drill-down operation can be treated as the exact opposite of roll-up operation of OLAP Cube. The set of detailed values from an aggregated value is computed by a splitting function. During the roll-up operation, the aggregation function is stored. When the drill-down operation is performed, the split function uses the respective aggregation function. If the drill-down operation is performed on the OLAP cube, it will be the same as described on the Roll-up example discussed in the above section, similar to the drill-down.

8.8 Summary

In this chapter, the fuzzy aspects of OLAP cubes are discussed. Two design methods are introduced. Both methods have their own pros and cons. One method is to build the fuzziness into the cube itself so that end-users simply have to use the fuzzy OLAP cube as they use a normal classical cube. However, whereas this method would add the complexity into the queries it may not be scalable when more fuzzy measures are added to the OLAP cube. The second method is the introduction of multiple cubes depending on the fuzzy attributes. This is not scalable for the same reason as was discussed with the previously method. The third is to leave the cube as it is an option to handle the fuzziness by means of queries which will add the burden to the end users.

Apart from the design aspects of a fuzzy OLAP cube, OLAP cube operations are also discussed. In this, regard slicing, dicing, roll-up, and drill-down operations are discussed. In the first two methods, these four operations are built into the cube itself and in the third method, the options need to be handled explicitly.

Non-functional requirements play a key role in any design and implementation. If the system cannot cater to the non-functional requirements, the system will not be able to implement even though it has covered great and latest technical coverage. In the next chapter, implement challenges such as non-functional requirements of the fuzzy data warehouse will be discussed.

Chapter 9

IMPLEMENTATION OF FUZZY DATA WAREHOUSE

9.1 Introduction

When the need arises to implement the fuzzy data warehouse, Non-Functional Requirement should be addressed. In systems and requirements engineering, a non-functional requirement (NFR) specifies criteria for use with judgment regarding the operation of a system as against detailed actions. They are in contrast with functional requirements that describe detailed behavior or functions.

So far, the design and implementation of fuzzy data warehouse was discussed in detail with respect to system design, ETL, OLAP and their implementations. Nevertheless, for practical usage of the fuzzy data warehouse, non-functional requirements need to be defined and implemented. After analyzing Boehm's and McCall's NFR lists given in Appendix M: NFR List, five aspects of NFR were considered for fuzzy data warehouse. They are: Validation, Configuration, Performance, Security, Auditing and Scalability. This chapter is about how existing NFR can be utilized and what more is required as additional NFR for a fuzzy data warehouse awaiting implementation.

9.2 Validation

For each value sum of weightage of the fuzzy membership function should be equal to one as the height of the fuzzy is equal to one. Nevertheless, due to rounding up and three fuzzy membership functions effecting in the same range, there can be a situation where weightages can be more than one. Moreover, when the data is produced, due to the multiple roundups there can be total weightages of more than one which should be invalid due to the definition of Height of the fuzzy membership function. Table 9-1 shows a sample set of data wherethe weightage is more than one.

Table 9-1: Subset of Data Violation of Height of Membership Function for Age

Value	State	Weightage
462	Middle	1.185
463	Middle	1.189
464	Middle	1.193
465	Middle	1.197
466	Middle	1.201
467	Middle	1.203
468	Middle	1.201
469	Middle	1.199
470	Middle	1.197
471	Middle	1.195
472	Middle	1.193

These invalid fuzzy membership weightages need to be captured and necessary actions need to be taken. After the fuzzy contributions are calculated using Appendix O: Script to Generate Fuzzy Contribution, in the last part of the script validation is done and the error correction is done so that height of the fuzzy membership function should equal to one.

9.3 Configuration

Configuration is an important factor in any system. It enables the end user's ability to configure parameters so that when the new values come in, end-users have to simply update the relevant configuration

In the proposed design, four tables were introduced to store the configuration described in Table 9-2.

Table 9-2: Configuration Tables

Configuration Table Name	Configuration
FuzzyMembershipFunctionConfiguration	Fuzzy Membership function with relevant parameters with history configurations.
FuzzyMembershipFunctionAarbitraryConfiguration	Fuzzy Membership function for arbitrary with relevant parameters with history configurations.
FuzzyMembershipFunctionLingusitcConfiguration	Fuzzy Membership function for Linguistic with relevant parameters with history configurations.
DerivedMembershipFunctionConfiguration	Fuzzy Membership function for derived function with relevant parameters with history configurations.

Figure 9-1 shows the attributes of each of those configuration tables.

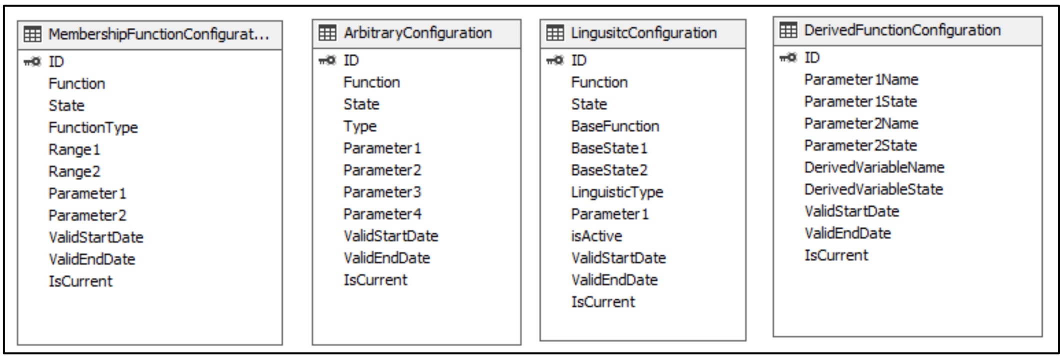


Figure 9-1: Configuration Tables for Membership Function in Fuzzy Data Warehouse

In these configuration tables shown in the Figure 9-1, ValidStartDate, ValidEndDate, IsCurrent columns are used to keep the track of the historical aspects of configurations as there can be circumstances where configurations can change over time. In the MembershipFunctionConfiguration table, basic parameters for fuzzy membership function are stored. In this table, the function is the fuzzy attribute and state defines different states of the fuzzy attribute. Function Type, Range1, Range2, Parameter1, and Parameter2 defines the profile of the fuzzy membership function. Depending on the fuzzy membership function type, trigonometric, trapezoidal, Gaussian function these parameters will change as discussed in Appendix D: Types of Membership Functions.

Table 9-3: Configuration of Membership FunctionConfiguration

ID	Function	State	Function Type	Range1	Range2	Parameter1	Parameter2
1	Age	Young	Linear	0.00	186.00	0.000	1.000
2	Age	Young	Linear	186.01	466.00	-0.004	1.663
3	Age	Young	Linear	466.01	999999.00	0.000	0.000
4	Age	Middle	Linear	0.00	186.00	0.000	0.000
5	Age	Middle	Linear	186.01	466.00	0.004	-0.663

As described in Chapter 6, there is a situation where fuzzy memberships are defined without a data-driven method and such configurations are defined in the *FuzzyMembershipFunctionArbitraryConfiguration* table for which, a sample data set is shown in Table 9-4.

Table 9-4: Configuration of Fuzzy Membership Function Arbitrary Configuration

ID	Function	State	Type	Parameter1	Parameter2	Parameter3
1	Acreage	Young	Trapezoidal	0.00	0.00	186.00
2	Acreage	Middle	Trapezoidal	186.00	416.00	568.00
3	Acreage	Old	Trapezoidal	568.00	568.00	875.00
4	Acreage_1	Young	Trapezoidal	0.00	0.00	186.00
5	Acreage_1	Middle	Trigonometric	186.00	466.00	875.00

In defining fuzzy membership with arbitrary configuration, parameters of shapes will be stored in the configuration as shown in Table 9-4. As described in Chapter 6, linguistic fuzzy memberships are defined to define the linguistic values in data warehousing and the configurations are stored in the *FuzzyMembershipFunctionLingusitcConfiguration* table. A sample of data is shown below in Table 9-5.

Table 9-5: Configuration of *Fuzzy Membership Function Linguistic Configuration*

ID	Function	State	Base Function	BaseState 1	BaseState 2	Linguistic Type	Parameter 1
1	Experience_3	Low Medium	Experience_3	Medium	NULL	DIL	0.650
2	Experience_3	High Medium	Experience_3	Medium	NULL	CON	1.150
3	Experience_3	Not Low	Experience_3	Low	NULL	NOT	NULL
4	Experience_3	Very Low	Experience_3	Low	NULL	DIL	0.500

Fuzzy linguistics is defined by fuzzy sets operations that are defined in Appendix E: Operations in Fuzzy Sets. Those parameters, the base function and the base state should be stored as shown in the Table 9-5.

Derived membership functions which will be defined by combining multiple fuzzy membership functions and configurations are stored in the *DerivedMembershipFunctionConfiguration* table as presented in Table 9-6.

Table 9-6: Configuration of *Derived Membership Function Configuration*

ID	Parameter1 Name	Parameter1 State	Parameter2 Name	Parameter2 State	Derived Variable Name	Derived Variable State
1	Tree	Low	Acreage	Low	Importance	Low
2	Tree	Low	Acreage	Medium	Importance	Medium
3	Tree	Low	Acreage	High	Importance	Medium
4	Tree	Medium	Acreage	Low	Importance	Medium
5	Tree	Medium	Acreage	Medium	Importance	Medium

In the configuration table shown in Table 9-6, two base parameters and their state are configured. With the above configuration, there is a limitation with the configuration that derived membership can be defined with only two base functions and is not possible with more than two functions.

These configurations will not be effective immediately after they are modified. Since fuzzy weightages are pre-calculated due to the performance implications, scripts listed in Appendix N and Appendix O need to be executed for these configurations to be effective. This script execution can be included as part of the daily ETL process, so that the fuzzy contributions are calculated with ETL and latency would be one day.

9.4 Performance

Performance is considered to be a key factor in any system. 'Perform' has a huge impact on the users' experience with the use of the system. With regard to data warehousing, special effort is necessary since data warehouse has to deal with a huge capacity of data contained by nature in many tetra bytes. In 7Vs of data challenges, volume is the one major critical challenge forced into a data warehouse faced with a large volume of data. Further, data warehouse needs to store historical data so that strategic decisions can be made with the stored data. These factors mean that data warehouse consists of high volume of data. As such, dealing with a large volume of data with higher performance is a must to improve usability. Further consideration should be shown towards a fuzzy data warehouse with regard to performance as some fuzzy design changes introduced by this research could influence the performance of the system unfavourably.

Typically, table indexes are the key component when it comes to database as well as the data warehouse performance. However, Index strategy is different in data warehouse [6] as the data warehouse has fewer writes and is dominated by data reads. Since index increases retreval queries and has an undesirable impact to write queries, many numbers of indexes can be included in data warehouses. Further, column store indexes are severely used with data warehouses as data warehouse requests scans through the complete fact table most of the time [3]. In the case of fuzzy data warehousing, both row-based indexes and column store indexes can be employed to improve performance. In addition to these indexes, there are other key techniques for improvement with the fuzzy data warehouse so that user experience can be enhanced.

To improve the performance, fuzzy weightages are pre-calculated and stored in the FuzzyContribution table presented in Figure 9-2.

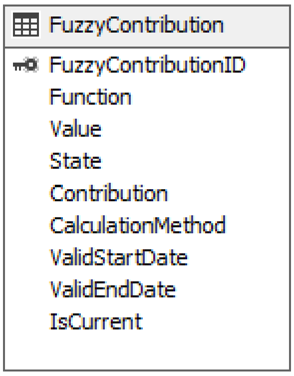


Figure 9-2: Schema Definition of Fuzzy Contribution Table

Storing the data in the pre-calculated format helps improve the analytics performance as there is no calculation to be done duing the data analysis. Like in the other table, this table has the capability to support historical nature of data changes. Using the configurations in the four tables previously discussed, the above table gets updated as shown in Table 9-7.

Table 9-7: Configuration of Fuzzy Contribution

Fuzzy ContributionID	**Function**	**Value**	**State**	**Contribution**	**Calculation Method**
1	Trees	1066	Medium	0.9969	P
2	Trees	1067	High	0.0031	P
3	Trees	1067	Medium	0.9969	P
4	Trees	1068	High	0.0032	P
5	Trees	1068	Medium	0.9968	P
6	Trees	1069	High	0.0032	P
7	Trees	1069	Medium	0.9968	P
8	Trees	1070	High	0.0033	P
9	Trees	1070	Medium	0.9967	P
10	Trees	1071	High	0.0033	P

In the Table 9-7, weightages are stored for a different function and the state. When fuzzy contribution needs to be calculated, it is a matter of joining this table with the value where the value column is indexed to improve performance. If these are not calculated, every time when fuzzy contribution needs to be calculated, it could lead to negative performance issues. However, since the fuzzy contribution is pre-calculated, whenever the fuzzy membership changes, it will not be reflected at that particular time. This latency is not a huge problem as in most of the data warehouse scenarios, there is latency with data update which is, typically, one day.

In the case of OLAP Cubes, MOLAP cubes are proposed over the ROLAP on top of the fuzzy data warehouse so that measures are pre-calculated against dimension attributes and its hierarchies. Different performance counter as listed in Appendix S: assists a user to get a baseline idea about different performance aspects.

9.5 Security

It is a difficult task to design a system without guaranteed security. The security of data warehouse poses different challenges to other systems, mainly database systems or OLTP systems [79]. Security is assured with three major aspects which are: Authentication, Authorization, and Encryption. Nevertheless, considering data warehousing, the authorization area will be a major challenge.

Data received from multiple data ources can be a serious challenge to the security of a data warehouse. The OLTP systems may have implemented security in isolation. When data is confined to the data warehouse, there could be a threat to data security. For example, with this research, it was observed that, there are multiple plantation estates operating in isolation, where one user at one location does not have access to the other estate. However, when it comes to the data warehouse, for analytical purposes, all the data of all the plantation estates are combined. Due to this, row-level or cell-level security needs to be implemented in the data warehouse. Row-level or cell-level security can be enhanced in the fuzzy data warehouse as well.

With fuzzy data warehousing, there can be instances where one state can be viewed by a separate set of users. In this scenario, the fuzzy state needs to be handled as a typical attribute. However, there can be a confusion as weightages sum will not be equal to one for the user may not have access to all the states. Since they do not have access to all states, they will not be able to understand that there is discrepancy. Another important security aspect in the fuzzy data warehouse is fuzzy configuration. It is recommended to separate fuzzy configurations so that only administrators or systems administrators of the data warehouse can change such parameters.

9.6 Scalability

Since the data warehouse has to work with a high volume of data in the order of tetra bytes and since data warehouse tends to grow in large volume s on a daily basis, scalability is an important factor to be considered. With a data warehouse, the partition option is used as a scalability option. The partition can be done at the database level as well as at the OLAP cube. Typically, the partition is done on the date column for a month or quarter or year. Therefore, even in fuzzy data warehouse this scalability option can still be done by means of a partition.

In a fuzzy data warehouse, scalability may originate as a challenge, for large and complex fuzzy membership function. However, the challenge is minimized as fuzzy contributions are pre-calculated as described in the previous section.

9.7 Summary

Non-Function requirements are key factors in any system and the fuzzy data warehouse system is no exception. In this chapter, validation, configuration, performance, security and scalability are factors considered under non-functional requirements so that the proposed fuzzy data warehouse implementations can be used effectively by business users effectively.

Validation is an essential process as fuzzy membership functions are key element in the fuzzy data warehouse. Therefore, validation follows the fuzzy weightages for correction to minimize weightage errors on the fuzzy membership function. Such configurations are essential considerations related to NFRs. Most of the fuzzy membership configurations are stored in tables to improve usability. As fuzzy data warehouse deals with a large volume of data, performance is a key factor. Apart from row-based and column stored indexes, fuzzy contribution weightages are pre-calculated towards improved performance. The fuzzy data warehouse can also be incorporated with Fuzzy OLAP cube, and MOLAP cube performance can be improved as MOLAP stores the calculated measure values in the OLAP cube. To enhance the scalability of the proposed fuzzy data warehouse, partition mechanism is proposed for the fuzzy fact tables as well as to the fuzzy OLAP cubes.

As such, non-functional requirements are also achieved with several options such as validation, configuration, security, performance and scalability. Therefore, it can be concluded that fuzzy data warehouse can be implemented effectively to the advantage of business users. The next chapter provides the conclusion to this research project while suggesting future work to make fuzzy data warehouses more effective.

Chapter 10
CONCLUSION

10.1 Introduction

This research is an investigation into the implementation of fuzzy logic based data warehousing for data analytics. Currently, there is very limited research available as related to end to end aspects of fuzzy data warehousing. Accordingly, the research commences with an initial discussion of the design aspects of fuzzy data warehouse, ETL functionalities, fuzzy OLAP cube and non-functional requirements in a fuzzy data warehouse. Going beyond, it covers the end-to-end aspects of a data warehouse.

The research highlights techniques and recommendations towards more effective implementations of better fuzzy data warehousing so that business users can leverage the features of a fuzzy data warehouse to their advantage. This chapter summarizes the research project and suggests future work to the fuzzy data warehousing research area.

10.2 Research Work

A detailed literature review has identified that there are a couple of attempts carried out to introduce fuzzy logic based analytics for data warehouses. In this research, a feasibility study is performed in order to verify whether fuzzy logic based analytics for data warehouse by identifying major implementations of domains. One of the main achievements in this research is the identification of data driven method for defining the fuzzy membership function for fuzzy systems as the fuzzy membership function plays a crucial role in a fuzzy system. Further, this research has introduced diverse types of fuzzy membership functions such as descriptive, box-plot, hybrid, linguistic, and derived fuzzy membership function in order to facilitate different aspects in data warehousing. Though the box-plot method is the key finding in this research, there are multiple alternatives such as descriptive and hybrid methods are proposed.

This research has introduced design techniques for fact, and dimension tables in the data warehouse, including type 2 SCD and hierarchies in dimensions in order to perform detailed analytics in fuzzy data warehousing.

Since ETL and OLAP cubes are an integral part of a data warehouse, this research introduced fuzzy aspects into ETL and OLAP. Two different design methods were introduced for the fuzzy OLAP cube. However, due to the introduction of fuzzy aspects into OLAP cube, there are performance issues. Though there are alternatives for fuzzy OLAP analytics such as fuzzy queries, fuzzy OLAP cube is designed in order to achieve operation extensibility in fuzzy logic based analytics for data warehousing.

The next important literature review area concentrated on fuzzy data warehouse implementation. This area too was identified in the same research. However, it was noted that the current research are far from complete and, therefore, prompted the need for this research. It was very evident that in most of the researches, fuzzy membership functions were dealt with in a subjective manner ignoring data-driven techniques. Hence, fuzzy membership function being a key component in a fuzzy data warehouse, proposing a data-driven technique for fuzzy membership function came to be a key deliverable with this research. The literature review also identified that there need to be different types of fuzzy membership functions with interconnection for better results.

This research was extended to implement of fuzzy OLAP Cube as OLAP Cubes are important aspects in data analytics. This research has introduced two different techniques of implementing fuzzy OLAP Cubes so that the standard operations such as Slicing, Dicing, etc. can be performed against the Fuzzy OLAP Cubes.

Since this research is targeted to introduce a feature extension of data warehouse. Since data warehouse is important factor in organization, it is essential to look for the non-functional requirements in the proposed method. Fuzzy data warehouse is introducing more complexity. Therefore, it is essential to focus on the performance aspects of data warehouse. In order to achieve necessary operational performance, it was proposed to store pre-calculated fuzzy weightages. Since data is extracted from multiple sources, security is a key factor in data warehousing. This research has proposed different implementations of security methods in fuzzy data warehousing. Validation and configurations are introduced specifically for fuzzy membership function which are other factors in non-functional requirements.

10.3 Future Work

In this research, only Type 1 fuzzy membership functions are used. As it was observed in the literature review, there are Type 2 fuzzy membership functions in the fuzzy logic. Though, Type 2 fuzzy membership functions are more precious than Type 1, due to the performance consideration type 2 fuzzy membership functions were de-scoped in this research. However, feasibility of using type 2 fuzzy membership functions can be considered in future work.

Data mining aspect was not considered in this research as data mining is a separate topic. With regard to data mining, all the aspects of data mining: classification, clustering, sequence, regression, association, time series techniques need to be considered. There are multiple algorithms for each technique. For example, in the association technique: Naïve, Apriori, and Improved Apriori algorithm needs to be considered. Similarly, in all the other data mining techniques there are multiple algorithms to be considered as regards the fuzzy data warehouse.

Yet, with this research some limitations in some areas were identified. In using the derived fuzzy membership function, only two fuzzy membership functions can be used to derive another fuzzy membership function. However, there can be situations where there can be multiple fuzzy membership functions which can be used to define a membership function. This feature is recommended for future research.

References

1. Abello A., Samos J., Saltor F., Understanding facts in a multidimensional Object-Oriented Model. In Proc. of the 4th Int. Workshop on Data warehousing and OLAP (DOLAP), pages 32-39, ACM Press, 2006.
2. Ahmad K.A., Syed Abdullah S.L., Othman M., Bakar M.N.A., Induction of Membership Functions and Fuzzy Rules for Harumanis Classification, Journal of Fundamental and Applied Sciences, ISSN 1112-9867, 2018 May 25.
3. Amro A.A., Abutabh M., Case Study: Comparison between Traditional Indexing and Column Store Indexes in SQL Server 2012, Conference: International Conference on Information & Intelligent Systems, Volume: N&N Global Technology for Computer and Information Service & IEEE Computational Intelligence Society, March 2013.
4. Arif M., A Survey on Data Warehouse Constructions, Process and Architectures, International Journal of u~ and e~ Service, Science, and Technology, Vol 8. No. 4, 2015, pp 9-16.
5. Ashraf, A., Akram, M., Sarwar M., “Type-II Fuzzy Decision Support System for Fertilizer", Punjab University College of Information Technology, University of Punjab, April 8, 2014.
6. Aouiche K., Darmont J., Index and Materialized View Selection in Data Warehouses, 2006.
7. Bahdi A., Chakhar S., Naiija Y., Implementing Imperfect Information in Fuzzy Databases, The International Symposium on Computational Intelligence and Intelligent Informatics, Hammamet, Tunisia, 2005.
8. Batra S., Sondal R., Data Warehouse, IJIRT, International Journal of Innovative Research in Technology, 2004, Volume 1 Issue 6, ISSN: 2349-6002
9. Bazile D., Le Page C., Dembélé S. & Abrami G., Perspectives of modelling the farmer’s seed system for in situ conservation of sorghum varieties in Mali, 5th Conference of the European Federation for Information Technology in Agriculture, Food and Environment and 3rd World Congress on Computers in Agriculture and Natural Resources., 2005, Universidade de Trásos-Montes e Alto Douro, Vila Real, Portugal.
10. Bennet A., Bennet D., Handbook on Decisions Support Systems, Springer-Verlag, Berlin, Heidelberg., 2008.
11. Bhargava, A.K., Singh. S.R., Bansal, D., “A Fuzzy Goal Programming Model for Bakery Production", International Journal of Computer & Mathematical Sciences, IJCMS, ISSN 2347 – 8527 Volume 3, Issue 4, June 2014.
12. Broek, P.V.D., Noppen, J., Rankings from Fuzzy Pairwise Comparisons, Faculty of Electrical Engineering, Mathematical and Computer Science, University of Twente, 2004.
13. Bouaziz B., Chakhar S., Mousseau V., Ram S., Telmoudi A., Database Design and Querying within the Fuzzy Semantic Model, Science Direct, Elsevier, 2007.

14. Buckles B.P., Petry E.F., A Fuzzy Representation of Data for Relational Databases, Fuzzy Sets and Systems 7, pp 213-226, North-Holland Publishing Company, 1982.
15. Burdick D., Deshpande P.M., Jayaram T.S., OLAP Over Uncertain and Imprecise Data, 31st VLDB Conference, Trondheim, Norway, 2005.
16. Chai K.L., Costello T.A., Wells B.R., Norman R.J., App- eng-agric., 1994,10, 849-855.
17. Chavda M.B., Solanki A., Innovative banking products: Win-Win situation for Customers and Banks, IRACST – International Journal of Commerce, Business and Management (IJCBM), ISSN: 2319–2828 Vol. 3, No.6, December 2014.
18. Chhabra R., Pahwa P., Data Mart Designing and Integration Approaches, International Journal of Computer Science and Mobile Computing, Vol. 3, Issue. 4, April 2014.
19. Coble H.D., Proceedings of the annual meeting of the Northeastern Weed Science Society, 1994, 48, 155-159.
20. Cooper, B. L., Watson, H.J., Wixom, B.H and Goodhue, D.L, "Data Warehousing supports Corporate strategy at First American Corporation", MIS Quarterly, Minneapolis, vol.24, Issue 4, pp:547-567, 2000.
21. Dad R., Shanmugam N., Singh P., Nalawade S.M., Analysis of Agriculture Commodity Prices using MapReduce Model, International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 4, April 2015, ISSN (Online) 2278-1021, ISSN (Print) 2319-5940.
22. Delgado M., Molina C., Sanchez D., Vila A., Rodriguez-Ariza L., A Fuzzy Multi-dimensional model for supporting imprecision in OLAP, 2004 IEEE international Conference on Fuzzy Systems, 3:1331 – 1336, 2004.
23. Dinesh Asanka, Loading Historical Data into a SQL Server Data Warehouse, MSSQLTips, https://www.mssqltips.com/sqlservertip/5505/loading-historical-data-into-a-sql-server-data-warehouse/, 2018-06-06, Accessed: 2018-07-01.
24. Detlefsen N.K., Jensen A.L., Agricultural Systems, 2004,81, 55-72.
25. Dogan U., "Data Warehouse and Data-Mining Tools for Risk Management: The Case of Turkey," The International Bank for Reconstruction and Development / the World Bank, Washington DC, 2011.
26. Dong W., Huang Z., Duan H., A genetic fuzzy system for unstable angina risk assessment, Cardiology Department of Chinese PLA General Hospital, Beijing, China, 2014.
27. Elliott, Thomas E., et al. "Data Warehouse Governance Programs in Healthcare Settings: A Literature Review and a Call to Action." eGEMs (Generating Evidence & Methods to improve patient outcomes) 1.1 (2013): 15.

28. Faisel D., Shahzad K., A Data Warehouse Model for Integrating Fuzzy Concepts in Meta Table Structures, Information Systems Research Group, Department of Informatics, University of Fribourg, Switzerland. 2009.
29. Faraghian H., Salehi S., Hekmatpanah M., An Evaluation to the Three-Layer ECRM&CRM Performance in Banking Sector in Order to Help Anti-Money Laundering Systems, International Journal of Information Science and Management, Special Issue, 2014.
30. Gadda K.R., Dey S., Business Intelligence for Public Sector Banks in India: A Case study-Design, Development, and Deployment. Journal of Finance, Accounting, and Management, 5(2), 37-58. ,2014.
31. Goel E., Data Warehousing and Data Mining in Business Applications, An International Journal of Engineering Sciences, Issue December 2014, Vol. 3, ISSN: 2229-6913 (Print), ISSN: 2320-0332 (Online).
32. Guo S.S., Yuan Z.M, Sun A.B., et al. A new ETL approach based on data virtualization. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 30(2): 311{323 Mar. 2015. DOI 10.1007/s11390-015-1524-3,
33. Hamoud A.K., Obaid T.A.S., Using OLAP with Diseases Registry Warehouse for Clinical Decision Support, International Journal of Computer Science and Mobile Computing, Vol.3 Issue.4, April- 2014, pg. 39-49.
34. Han J., Kamber M., Pei J., Data Mining Concepts and Techniques, Third Edition, Elsevier, 2012.
35. Hansen J.W., Agricultural Systems, 2002 74, 305-307.
36. Hongmei J.U., Wen L.U., The Application of Fuzzy Clustering Method in the Warehouse Performance Evaluation, International Business and Management Vol. 10, No. 1, 2015, pp. 44-49, ISSN 1923-841X [Print] ISSN 1923-8428 [Online],
37. Huynh V.N., Ho T.B., Nakamori Y., A Parametric Representation of linguistic Hedges in Zadeh’s Fuzzy Logic, International Journal of Approximate Reasoning, 2002, Elsevier.
38. Ibragimov T.Z., Ibragimova I.T., Sanin S.S., Decision support systems for cereal crop disease control, 2005, EFITA/WCCA, Vila Real, Portugal.
39. Inmon, W.H., “What is a Data Warehouse?” Prism, Volume 1, Number 1, 1995.
40. Juliyet C., Amanullah K.M., Optimizing Query Performance with OLAP to Discovering the Diagnosis of Diabetes, International Journal of Computer Science and Mobile Computing, Vol.4 Issue.7, July- 2015, pg. 290-298.
41. Jørgensen L.N., Noe E., Langvad A.M., Jensen J.E., Ørum J. E., Rydahl P., EPPO Bulletin, 2007, 37(2), 374-377.

42. Kasinadh D.P.V., Radha Krishna P., Building Fuzzy OLAP Using Multi-attribute Summarization, International Conference on Computational Intelligence and Multimedia Applications, 2007.
43. Khan M.A., Uddin M. F., Gupta N., Seven V's of Big Data, Understanding Big Data to extract Value, Proceedings of 2014 Zone 1 Conference of the American Society for Engineering Education (ASEE Zone 1), 978-1-4799-5233-5/14, 2014, International Electrical and Electronics Engineers.
44. Kimball, R., Ross M., The Data Warehouse Toolkit, Second Edition, The Complete Guide to Dimensional Modeling, John Wiley & Sons, Inc, 2007.
45. Kimball, R., Ross M., The Data Warehouse Toolkit, Third Edition, The Complete Guide to Dimensional Modeling, John Wiley & Sons, Inc, 2007.
46. Kohli A., Raina A., An Overview of Data Warehousing and OLAP Technology, IJIRT, International Journal of Innovative Research in Technology., Volume 1 Issue 6, ISSN: 2349-6002.
47. Kseniya L., Pollution Management Discussion Note: In focus, 1999, 3.
48. Kumar S., Aspect of Data Mining and Data Warehousing, International Journal of Technology Enhancements and Emerging Engineering Research, Vol 2, Issue 6, 2014, ISSN 2347-4289.
49. Ma Z.M., Yan L., A Literature Overview of Fuzzy Database Models, Journal of Information Science and Engineering, 189-202, 2008.
50. Mamdani E.H., Application of Fuzzy Logic to Approximate Reasoning Using Linguistic Synthesis, Department of Electrical and Electronic Engineering, University of London, London, 1976.
51. Matthews K., Buchan K., Proceedings of the Modelling and Simulation Society of Australian and New Zealand, 2003, 4, 1534- 1539.
52. Medina J.M., Vila M.A., Cubero J.C., Pons O., Towards the Implementation of Fuzzy Relational Database Model, Fuzzy Sets and Systems, ELSEVIER, 273-289., 1994.
53. Meinke H., Baethgen W.E., Carberry P.S. Donatelli M., Hammer G.L., Selvaraju R. & Stockle C.O., Agricultural Systems, 2001, 70, 493-513.
54. Mendis, D.S.K., Karunananda, A.S., Samaratunga, U., Ratnayake U., A Fuzzy Expert System for Business Intelligence, 10th International Conference on Business Management, 2013.
55. Mosseddaq F., Dnidane S., Lahlou M., Integrated Wheat N Nutrition Management in Morocco: A Decision Support Model, Progress of Information Technology in Agriculture Proceedings of the 4th International Symposium on Intelligent Information Technology in Agriculture (ISIITA). 2007.
56. Notched Box Plots, https://sites.google.com/site/davidsstatistics/home/notched-box-plots, Accessed On 2016-August-28.

57. Obunadike G.N., Umeh A., Leveraging Data Mining and Data Warehouse to Improve Prison Services and Operations in Nigeria, Information and Knowledge Management, www.iiste.org, ISSN 2224-5758 (Paper), ISSN 2224-896X (Online) Vol.4, No.5, 2014.
58. Olszak C., Ziemba E., Approach to Building and Implementing Business Intelligence Systems, Interdisciplinary Journal of Information, Knowledge, and Management, 2007, Vol 2, pp. 135-148.
59. Paramasivan C., Naidu V.K., Electronic Banking Services in India, SELP Journal of Social Science, ISSN: 0975-9999 (P) 2349-1655 (O) Vol. VI, Issue. 23, 2015.
60. Pavan Kumar K.V.N.N, Radha Krishna, Supriya Kumar, Fuzzy OLAP Cube for Qualitative Analysis, ICISIP, 2005.
61. Riberir R.P., Borges J.G., MetaForest- a web-based decision system to support forest management involving multiple-ownership, EFITA/WCCA, 2005, Vila Real, Portugal.
62. Sabounchi N.S.Z., Triantis K., Sarangi S., Liu S., Fuzzy Modeling of Linguistic Variables in a System Dynamics Context, Grado Department of Industrial and Systems Engineering, Virginia Tech, Falls Church, VA, 22043, USA, 2010.
63. Samuel A., Pandey A.K., Sharma V.K., Estimation of Functional Size of a Data Warehouse System using COSMIC FSM Method, ACEEE, Procedure of Internal Conference on Advances in Computer Science and Application, 2013.
64. Sapir L., Shmilovic A., A Method for Design of a Fuzzy Data Warehouse, Department of Information Systems Engineering, Ben-Gurion University, Israel, 2008.
65. Sapna S., Pravin K., Integration of Fuzzy Clustering Technique with Big Data for Disease Diagnosis, International Journal of Electrical, Computing Engineering and Communication (IJECC) Vol. 1, Issue. 3, June – 2015 ISSN (Online): 2394-8310.
66. Sedlak O., Cileg M., KisT., Ciric Z., Measures of Uncertainty in Decision Making, International Seminar on Operational Research on Operational Research, 2013.
67. Sindhujaa N., Anees M., Patra P.S.K., Recent Issues and Its Quick Fixes for Various Data Repositories, International Journal of Research in Computer and Communication Technology, Vol 3, Issue 4, April- 2014, ISSN (Online) 2278- 5841, ISSN (Print) 2320-5156.
68. Shim J.P., Warkentin M., Courtney J.F., Power D.J., Sharda R. & Carlsson C., Decision Support Systems, 2002, 111-126.
69. Sodtke R., Multifunctional of Landscapes - Analysis, Evaluation, and Decision Support, International Conference, Univ. Giessen, 2005, 193.
70. Suarez de Cepeda M., Recio B., Rubio F., Decision Support System for Farm Mechanization, 5th Conference of the European Federation for Information Technology in Agriculture, Food and Environment and 3rd World Congress on Computers in Agriculture and Natural Resources, 2005, Vila Real, Portugal.

71. Tremblay M.C., "Doing more with more information: Changing healthcare planning with OLAP tools," Decision Support Systems, vol. 43, pp. 1305-1320, 2007.
72. Tomaszewska L., “The application of horizontal membership functions to fuzzy arithmetic operations”, Journal of Theoretical and Applied Computer Science, Vol 8, No2 2014, pp 3-10.
73. Van der Aalst, Process Cubes: Slicing, Dicing, Rolling Up and Drilling Down Event Data for Process Mining, 2013.
74. Van Heeringen, H., Measuring the functional size of a data warehouse application using the COSMIC FFP method, Software Measurement European Forum Conference, Rome, Italy, May 2006.
75. Vassiliadis P., Simitsis A., Near Real-Time ETL∗, University of Ioannina, Ben-Gurion University, Israel, 2008.
76. Vassiliadis P., Modeling Multidimensional Databases, Cubes and Cube Operations, National Technical University of Athens, 1998.
77. Wang Y.C., Chien C.J., Wang C.H., A Fuzzy-Neural Adaptive Iterative Learning Control for Freeway Traffic Flow Systems, Proceedings of the International MultiConference of Engineers and Computer Scientists 2016 Vol I, IMECS 2016, March 16 - 18, 2016, Hong Kong.
78. Wang Q., Wang Y., Zhang H., Sun Y., Load Forecasting Research of Power System Based on Fuzzy Sets Algorithm, International Journal of Signal Processing, Image Processing and Pattern Recognition Vol.9, No.6, 2016, pp.283-292, ISSN: 2005-4254.
79. Weippl E.R., Security in Data Warehouses, retrieved https://www.sba-research.org/wp-content/uploads/publications/weippl_arh_securityDWH.pdf, Accessed on 2018-08-12
80. Yadav DK, Yadav HB. Developing membership functions and fuzzy rules from numerical data for decision making. In 16th World Congress of the International Fuzzy Systems Association and 9th Conference of the European Society for Fuzzy Logic and Technology, 2015, pp. 551-555.
81. Zaker M., Phon-Amnuaisuk S., Haw S.C., Optimizing the Data Warehouse Design by Hierarchical De-normalizing, Proceedings of the 8th WSEAS International Conference on Applied Computer Science, ACS 08, Faculty of Information Technology, Multimedia University, Malaysia, 2008, ISSN: 1790-5109, ISBN: 978-960-474-028-4.
82. Zeng W., Li J., Fuzzy Logic and Its Application in Football Team Ranking, Hindawi Publishing Cooperation, The Scientific World Journal, Volume 2014, Article ID 291650, 6 pages.
83. Zhao J., Bose B.K., Evaluation of Membership Functions for Fuzzy Logic Controlled Induction Motor Drive, IEEE, 2002.
84. Zhong, W. et al, A Framework of Applying BI to Social Security Systems. Proceeding of the International Conference and Automation, 2008, pp. 189-193.

85. Ling Feng, Dillion D.T., Using Fuzzy Linguistic Representations to Provide Explanatory Semantics for Data Warehouses, IEEE Transactions on Knowledge and Data Engineering, Vol. 15, No 1, January/February 2003.
86. A. Gegov, R Babushka, H.B. Verbruggen, Linguistic Analysis of Interactions in MIMO Fuzzy Systems, 14th Triennial World Congress, Beijing, China, 1999.
87. PPG Dinesh Asanka, Data Cleansing with SSIS, retrieved http://www.sql-server-performance.com/2007/data-cleaning-ssis/, SQL-Server-Performance.com, Accessed: 2018-12-04.
88. Microsoft SQL Server 2016 Technical Overview, retrieved https://www.microsoft.com/en-us/sql-server/sql-server-2016, Accessed: 2018-12-04.
89. The R Project for Statistical Computing, retrieved https://www.r-project.org/, Accessed: 2018-12-04.
90. MatLab, retrieved https://www.mathworks.com/products/matlab.html/, Accessed: 2018-12-04.
91. D. Mankad, P. Dholakia, The Study on Data Warehouse Design and Usage, International Journal of Scientific and Research Publications, Volume 3, Issue 3, March 2013, ISSN 2250-3153.
92. Challoumis, Constantinos, Binary Fields and Economics through Fuzzy Logic Approach and Boolean Algebra Using Multidimensional Processing with Respect to Artificial Neural Networks and Machine Learning (June 30, 2016). Available at SSRN: https://ssrn.com/abstract=3123275.
93. Witold Pedrycz., Fernando Gomide., An Introduction to Fuzzy Sets, Analysis and Design., Massachusetts Institute of Technology., 1998.
94. Ross T.J., Fuzzy Logic with Engineering Applications, Third Edition, 2010 John Wiley & Sons, Ltd. ISBN: 978-0-470-74376-8.
95. G. Garani, A. Chernov, I. Savvas and M. Butakova, "A Data Warehouse Approach for Business Intelligence," 2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), Napoli, Italy, 2019, pp. 70-75, doi: 10.1109/WETICE.2019.00022.
96. Ruilian Hou, "Research and analysis of data warehouse technologies," Proceedings of 2011 International Conference on Computer Science and Network Technology, Harbin, China, 2011, pp. 1919-1922, doi: 10.1109/ICCSNT.2011.6182345.
97. N. El Moukhi, I. El Azami and A. Mouloudi, "Data warehouse state of the art and future challenges," 2015 International Conference on Cloud Technologies and Applications (CloudTech), Marrakech, Morocco, 2015, pp. 1-6, doi: 10.1109/CloudTech.2015.7337004.

98. N. El Moukhi, I. El Azami and A. Mouloudi, "Data warehouse state of the art and future challenges," 2015 International Conference on Cloud Technologies and Applications (CloudTech), Marrakech, Morocco, 2015, pp. 1-6, doi: 10.1109/CloudTech.2015.7337004.

99. H. Qin, Z. Qian and Y. Zhao, "On the Research of Data Warehouse in Big Data," 2015 International Conference on Network and Information Systems for Computers, Wuhan, China, 2015, pp. 354-357, doi: 10.1109/ICNISC.2015.126.

100. O. Moscoso-Zea, J. Paredes-Gualtor and S. LujáN-Mora, "A Holistic View of Data Warehousing in Education," in IEEE Access, vol. 6, pp. 64659-64673, 2018, doi: 10.1109/ACCESS.2018.2876753.

101. U. Aftab and G. F. Siddiqui, "Big Data Augmentation with Data Warehouse: A Survey," 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 2018, pp. 2785-2794, doi: 10.1109/BigData.2018.8622206.

102. L. Sapir, A. Shmilovici and L. Rokach, "A methodology for the design of a fuzzy data warehouse," 2008 4th International IEEE Conference Intelligent Systems, Varna, Bulgaria, 2008, pp. 2-14-2-21, doi: 10.1109/IS.2008.4670400.

103. Wensveen, S. A. G., Constructive design research. Technische Universiteit Eindhoven, 2018, Presented on October 19, 2018 at Eindhoven University of Technology.

Bibliography

1. R. Kimball., M. Ross., The Data Warehouse Toolkit, The Definite Guide to Dimensional Modelling., Third Edition., John Wiley & Sons, Inc., 2002.
2. R. Kimball., M. Ross., The Data Warehouse Toolkit, The Definite Guide to Dimensional Modelling., Second Edition., John Wiley & Sons, Inc., 2013.
3. W.H. Inmon, Derek Strauss, Genia Naushlass., DW 2.0, the Architecture for the Next Generation of Data Warehousing., Elsevier Inv., 2008.
4. W.H. Inmon, Building the Data Warehouse, Third Edition., John Wiley & Sons, Inc., 2002.
5. Joel Grus., Data Science from Scratch., O'Reilly Media., April 2015.
6. Witold Pedrycz., Fernando Gomide., An Introduction to Fuzzy Sets, Analysis and Design., Massachusetts Institute of Technology., 1998.
7. George J., Klinn., Bo Tuan, Fuzzy Sets and Fuzzy Logic, Theory and Applications, Prentice-Hall, 1995.
8. Hand D., Mannila H., Smyth P., Principles of Data Mining, Prentice-Hall of India, New Delhi, 2007.
9. Han J., Kamber M., Pei J., Data Mining Concepts and Techniques, Third Edition, Elsevier, 2012.
10. Blaxter L., Huges C., Tight M., How to Research, Third Edition,
11. Chen G., Pham T., Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems, CRC Press, 2001.
12. Karunananda, A.S. How to do Research, Author Publication, 2000.
13. Non-Functional Requirements (NFRs), Department of Computer Science, University of Toronto.
14. Jang J.S.R., Sun C.T., Mizutani E., Neuro-Fuzzy & Soft Computing, A Computational Approach to Learning and Machine Intelligence, Prentice-Hall, Pearson Education, 1997.

Appendix A: Fact Tables

In the world of data warehousing, fact tables contain measure columns. For example, order amount, order quantity is fact columns. In addition to fact columns, there are surrogate keys in the fact table, which refers to dimension tables. There are several important concepts in the Fact tables.

Additive measures are measures that can be added across any dimensions. Semi-additive measures are measures that can be added between some, but not all dimensions. Non-additive measures are measures that cannot be added via any dimensions.

Depending on the usage of the fact table, Ralph Kimball has defined several types of fact tables.

Fact Transaction tables after the most common of all fact tables. These fact tables will store transaction-level data.

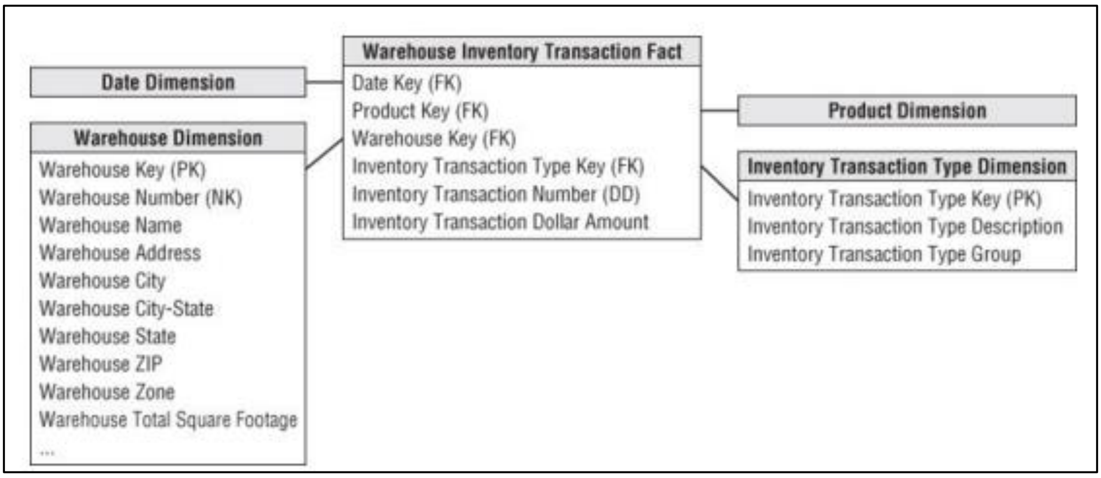


Figure A-1: Table Structures for Transaction Fact Table

As shown in the figure A.1, each inventory transaction identifies the date, product, warehouse, vendor, transaction type and in most cases, a single amount representing the inventory quality impacts caused by the transaction. Defining the granularity of the fact table is one row per inventory transaction, the designed schema is demonstrated in Figure A.1. Aggregation fact tables also fall into this category. For example, the model shown in Figure A.1 can be aggregated with Date Key and Product Key.

Next type of fact table is a snapshot fact table. If an organization wants to see the status as per today, snapshot fact table model should be applied.

Figure A-2: Table Structures for Snapshot Fact Table

In the example shown above, it contains quantity in hand for a given day and the following day will have the quantity in hand for the next day. Summation of those two values does not provide you any meaningful value whereas average will have a meaning. This mean snapshot fact tables are semi-additive.

The last type of fact tables is, accumulating fact tables. Some transactions are done in stages. For example, when a product item is purchased, there are several stages such as an item is received, when the Item is inspected and finally when the item is placed in the bin as shown in Figure A.3.

Fact row inserted when lot received:

Lot Receipt Number	Date Received Key	Date Inspected Key	Date Bin Placement Key	Product Key	Quantity Received	Receipt to Inspected Lag	Receipt to Bin Placement Lag
101	20130101	0	0	1	100		

Fact row updated when lot inspected:

Lot Receipt Number	Date Received Key	Date Inspected Key	Date Bin Placement Key	Product Key	Quantity Received	Receipt to Inspected Lag	Receipt to Bin Placement Lag
101	20130101	20130103	0	1	100	2	

Fact row updated when lot placed in bin:

Lot Receipt Number	Date Received Key	Date Inspected Key	Date Bin Placement Key	Product Key	Quantity Received	Receipt to Inspected Lag	Receipt to Bin Placement Lag
101	20130101	20130103	20130104	1	100	2	3

Figure A-3: Evolution of an Accumulating Snapshot Fact Row

Above structure can be modeled as shown in Figure A.4.

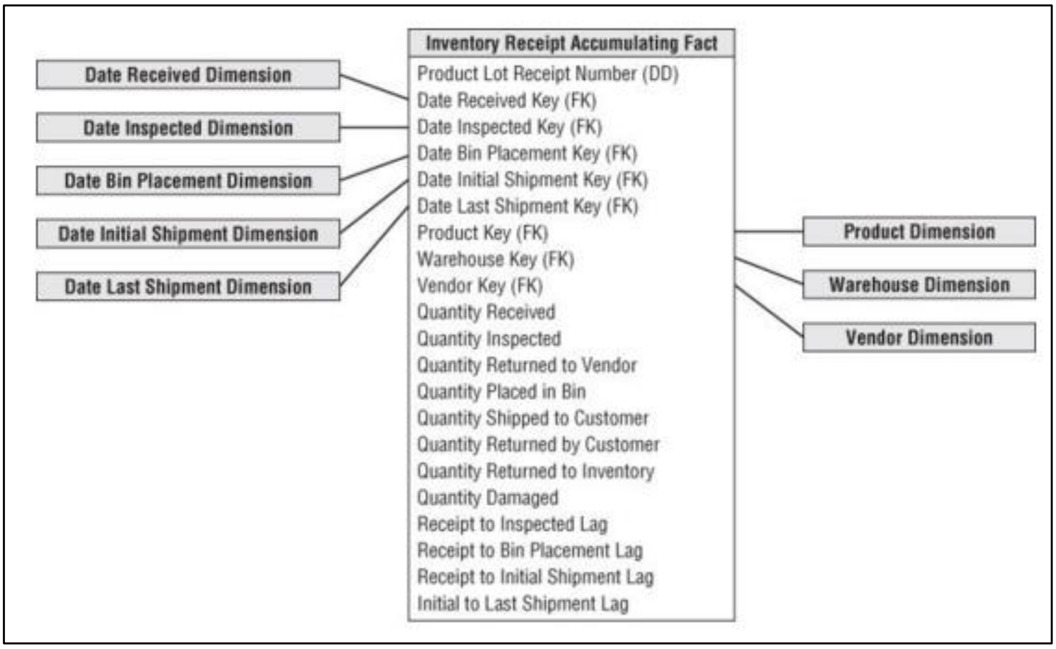


Figure A-4: Table Structures for Accumulating Snapshot Fact Table

All types of fact tables can be compared as shown in the following tables.

Table A-1: Different Types of Fact Tables

	Transaction	Periodic Snapshot	Accumulating Snapshot
Periodicity	Discrete transaction point in time	Recurring snapshots at regular, predictable intervals.	The indeterminate time span for evolving pipeline/workflow.
Grain	1 row per transaction or transaction line	1 row per snapshot period plus other dimensions.	1 row per pipeline occurrence.
Date Dimension	Transaction Date	Snapshot Date	Multiple dates for pipeline's key milestones.
Facts	Transaction Performance	Cumulative performance for the time interval.	Performance for pipeline occurrence.
Fact table Sparsity	Sparse or dense, depending on activity	Predictable dense	Sparse or dense, depending on pipeline occurrence.
Fact table Update	No updates, unless error correction.	No updates, unless error correction.	Updated whenever pipeline activity occurs

Fact-less-Fact tables are another type of fact table which is not discussed under this research. However, for the completeness let us discuss factless fact tables.

Fact tables does not store measures all the time. There can be instances where fact able has dimension surrogate keys only. For example, a student appearing a class on a certain day may not have a stored numeric fact, but a fact row with foreign keys for dimensions such as calendar day, student, teacher, location, and class is clearly defined. Factless fact tables can also be used to analyze what did not take place. A fact-less coverage table contains all the possibilities of events that might have happened.

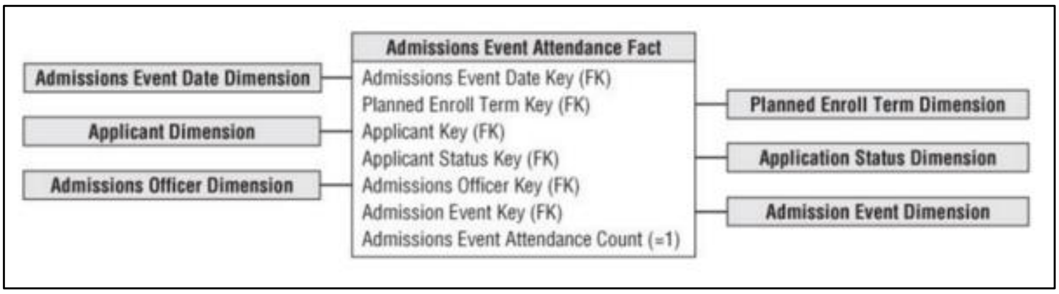


Figure A-5: Table Structures for Fact-Less Fact Table

Appendix B: Dimension Tables

A dimension table is a structure that classifies facts and measures in order to empower users to perform analytics on their questions. Frequently used dimensions in a data warehouse are employee, products, location and date. In a data warehouse, dimensions provide structured labeling information to otherwise unordered numeric measures. Power of the data warehouse is depended on dimension attributes; the analytical power of the data warehouse environment is directly related to the quality and depth of the dimension attributes. Depending on the storage and the usage dimensions are categorized into eight types by Ralph Kimball which are shown in Table B-2.

Table B-2: Different Types of Dimensions

Type of Dimension	**Dimension Table Action**	**Impact on Fact Analysis**
Type 0	No change to attribute value.	Facts associated with attribute's original values.
Type 1	Overwrite attribute value.	Facts associated with the attribute's current value.
Type 2	Add new dimension row for the profile with the new attribute value.	Facts associated with an attribute value in effect when fact occurred.
Type 3	Add a new column to preserve attribute's current and prior value.	Facts associated with both current and prior attribute alternative values.
Type 4	Add mini-dimension table containing rapidly changing attributes,	Facts associated with rapidly changing attributes in effect when fact occurred.
Type 5	Add type 4 mini-dimension, along with overwritten type 1 mini-dimension key in base dimension.	Facts associated with rapidly changing attributes in effect when fact occurred, plus current rapidly changing attributes values.
Type 6	Add type 1 overwritten attributes to type 2-dimension row and overwrite all prior dimensions rows.	Fact associated with an attribute value in effect when fact occurred plus current values.
Type 7	Add type 2-dimension row with the new attribute value, plus view limited current rows and/or attributes values.	Facts associated with attributes value in effect when fact occurred, plus current values.

Type 1 – Overwrite

As shown in Figure B-6 modified attribute will be overwritten.

Original row in Product dimension:

Product Key	SKU (NK)	Product Description	Department Name
12345	ABC922-Z	IntelliKidz	Education

Updated row in Product dimension:

Product Key	SKU (NK)	Product Description	Department Name
12345	ABC922-Z	IntelliKidz	Strategy

Figure B-6: Dimension Type 1

Type 2 – Add a new row

As shown in Figure B-7 modified attribute will be added as another row and current row indicator will be set to Expired.

Original row in Product dimension:

Product Key	SKU (NK)	Product Description	Department Name	...	Row Effective Date	Row Expiration Date	Current Row Indicator
12345	ABC922-Z	IntelliKidz	Education	...	2012-01-01	9999-12-31	Current

Rows in Product dimension following department reassignment:

Product Key	SKU (NK)	Product Description	Department Name	...	Row Effective Date	Row Expiration Date	Current Row Indicator
12345	ABC922-Z	IntelliKidz	Education	...	2012-01-01	2013-01-31	Expired
25984	ABC922-Z	IntelliKidz	Strategy	...	2013-02-01	9999-12-31	Current

Figure B-7: Dimension Type 2

This is the most common implementation in data warehousing.

Type 3 – Add the new attribute

In the case of Type 3 SCD, a new column will be added.

Original row in Product dimension:

Product Key	SKU (NK)	Product Description	Department Name
12345	ABC922-Z	IntelliKidz	Education

Updated row in Product dimension:

Product Key	SKU (NK)	Product Description	Department Name	Prior Department Name
12345	ABC922-Z	IntelliKidz	Strategy	Education

Figure B-8: Dimension Type 3

In this implementation, only the last state will be stored.

Type 5: Mini-Dimension and Type 1 Outrigger

Add a current mini dimension key as an attribute as a type 1 attribute. This will be helpful to get the current count as shown in Figure B-9.

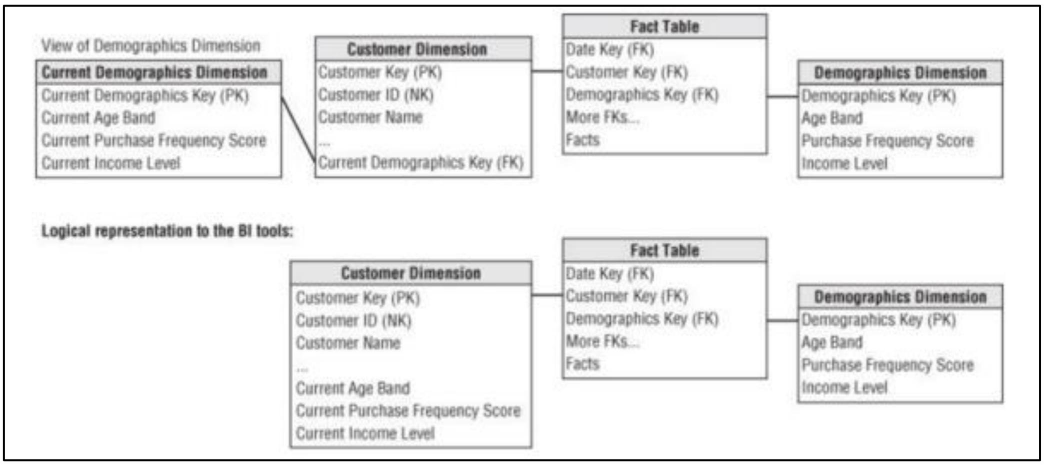


Figure B-9: Mini-Dimension and Type 1 Outrigger

Type 6: Adding Type 1 Attribute to Type 2 Dimension

This type of dimension is explained in the Figure B:10.

Original row in Product dimension:

Product Key	SKU (NK)	Product Description	Historic Department Name	Current Department Name	...	Row Effective Date	Row Expiration Date	Current Row Indicator
12345	ABC922-Z	IntelliKidz	Education	Education	...	2012-01-01	9999-12-31	Current

Rows in Product dimension following first department reassignment:

Product Key	SKU (NK)	Product Description	Historic Department Name	Current Department Name	...	Row Effective Date	Row Expiration Date	Current Row Indicator
12345	ABC922-Z	IntelliKidz	Education	Strategy	...	2012-01-01	2013-01-31	Expired
25984	ABC922-Z	IntelliKidz	Strategy	Strategy	...	2013-02-01	9999-12-31	Current

Rows in Product dimension following second department reassignment:

Product Key	SKU (NK)	Product Description	Historic Department Name	Current Department Name	...	Row Effective Date	Row Expiration Date	Current Row Indicator
12345	ABC922-Z	IntelliKidz	Education	Critical Thinking	...	2012-01-01	2013-01-31	Expired
25984	ABC922-Z	IntelliKidz	Strategy	Critical Thinking	...	2013-02-01	2013-06-30	Expired
31726	ABC922-Z	IntelliKidz	Critical Thinking	Critical Thinking	...	2013-07-01	9999-12-31	Current

Figure B-10: Add Type 1 Attribute to Type 2 Dimension

Type 7: Dual Type 1 and Type 2 Dimension

The dimension natural key or the business key is included as a fact table foreign key, in addition to the surrogate key for type 2 track.

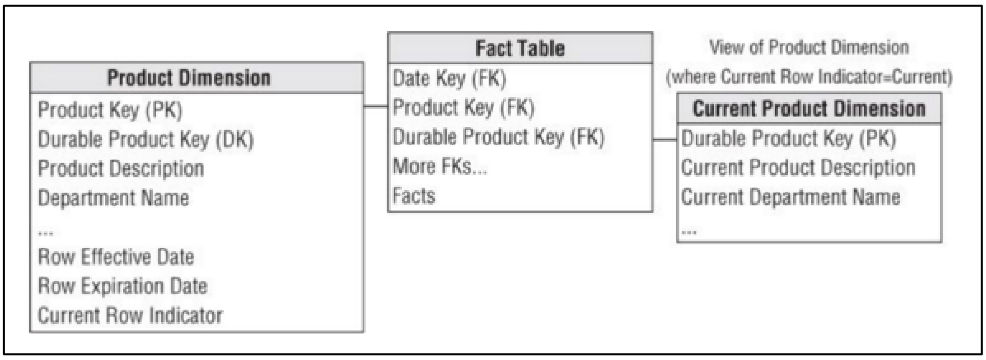


Figure B-11: Dual Type 1 and Type 2 Dimension

Appendix C: Date Dimension

Date dimension is considered as a distinct dimension in data warehousing mainly as this type of table is not available with the traditional OLTP systems. Calendar date dimensions are devoted to almost every fact table to allow analysis of the fact table through familiar attributes like dates, months, fiscal periods, and special days such as holidays, etc. on the calendar.

In case you need to query some tractions of Sunday, you may need to use in-built database functions. However, those in-built functions are not efficient. Further, in case of special holidays such as Easter in Europe, Poya day in Sri Lanka will not be handled. Considering those aspects among others, in data warehousing especially date dimension is introduced.

The calendar date dimension naturally has numerous attributes unfolding features such as week number, weekday, weekend, month number, month name, fiscal period, and holiday indicator etc. The date dimension table needs a special row to represent unknown or to-be-determined dates. If a smart date key is used, filtering and grouping should be based on the dimension table's attributes, not the smart key.

The Table C-3 is the structure for the date dimension in the sample database of AdventureWorksDW20012 which is packaged with SQL Server 2016.

Table C-3: Different Types of Fact Tables

Name	**ID**	**Type**	**Distinct count**
Calendar Quarter	CalendarQuarterDesc	Quarters	13
Calendar Quarter of Year	CalendarQuarterOfYear	QuarterOfYear	
Calendar Semester	CalendarSemesterDesc	HalfYears	7
Calendar Semester of Year	CalendarSemesterOfYearDesc	HalfYearOfYear	
Calendar Year	CalendarYear	Years	4
Date	TimeKey	Date	1158
Day Name	DayNameOfWeek	Days	
Day of Month	DayNumberOfMonth	DayOfMonth	
Day of Week	DayNumberOfWeek	DayOfWeek	
Day of Year	DayNumberOfYear	DayOfYear	
Fiscal Quarter	FiscalQuarterDesc	FiscalQuarters	13
Fiscal Quarter of Year	FiscalQuarterOfYearDesc	FiscalQuarterOfYear	
Fiscal Semester	FiscalSemesterDesc	FiscalHalfYears	7
Fiscal Semester of Year	FiscalSemesterOfYearDesc	FiscalHalfYearOfYear	
Fiscal Year	Fiscal Year	FiscalYears	4
Month Name	MonthName	Months	38
Month of Year	EnglishMonthName		
Week of Year	WeekNumberOfYear	WeekOfYear	

There are more attributes included in Kimball’s date dimension which is shown in the below figure.

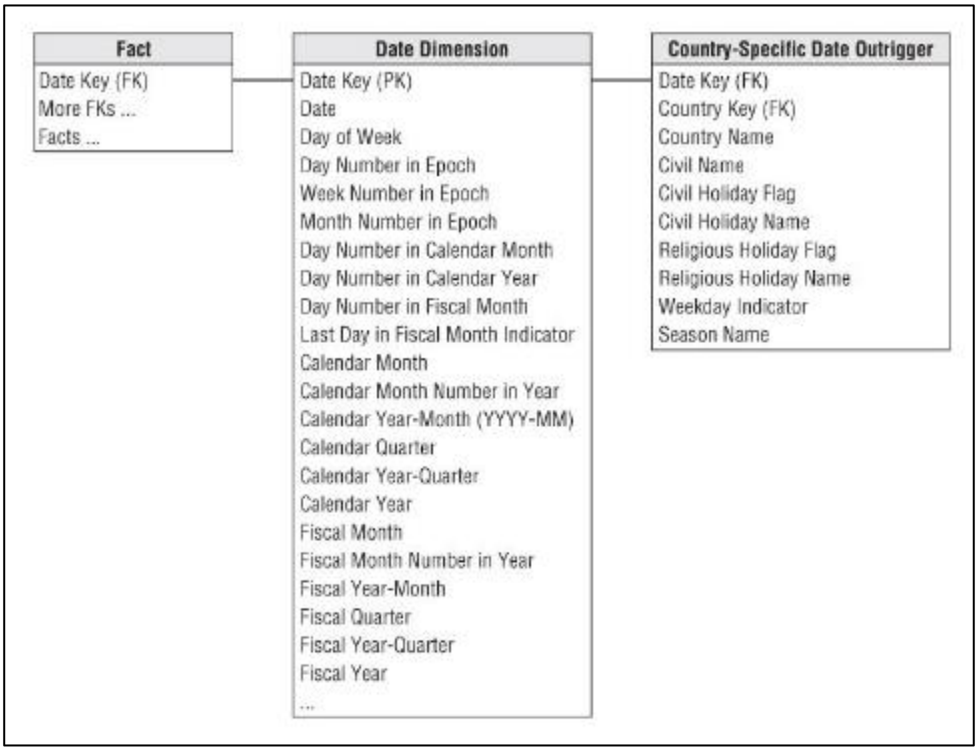


Figure C-12: Date Dimension Proposed by Ralph Kimball

Appendix D: Types of Fuzzy Membership Functions

In the fuzzy sets, fuzzy member function plays a key role, as it defines the quality of the fuzzy sets. Currently, there are six types of membership functions that are used in various applications. The graphs of the function may have diverse shapes and may have some precise properties such as continuity, symmetry, etc. Whether a specific shape is suitable can be determined only in the context of the application. In some cases, however, the meaning semantics captured by fuzzy sets is not very much sensitive to variation in the shape and simple functions are convenient. In many practical instances, the fuzzy set can be represented explicitly by features by families of parameterized function which are described below.

Triangular Function

$$A(x) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{m-a} & x \in [a, m] \\ \frac{b-x}{b-m} & x \in [m, b] \\ 0 & x \geq b \end{cases}$$

Where m is a modal value m and, a and b denote the lower and upper bounds for non-zero values. This function can be written as follows.

A(x: a,,m,,b) = max{ min[$\frac{x-a}{m-a}$,$\frac{b-x}{b-m}$], 0 }

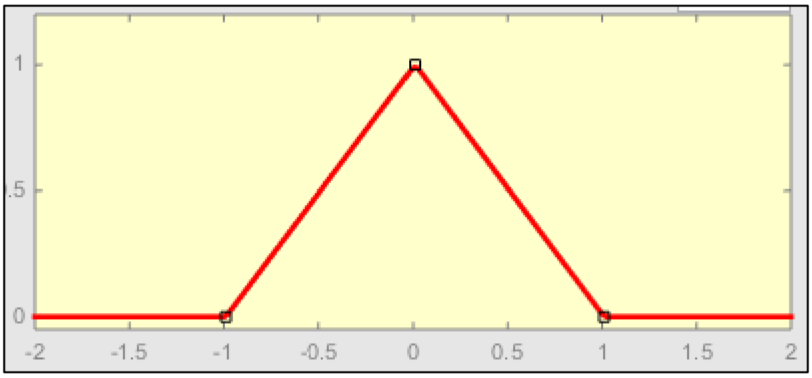


Figure D-13: Triangular Function

Figure C-12 triangular function can be written as A (x: -1.0,1,1)

Γ -Function

$$A(x) = \begin{cases} 0 & x \leq a \\ 1 - exp(-k((x-a)^2)) & x > a \end{cases}$$

$$or$$

$$A(x) = \begin{cases} 0 & x \leq a \\ \frac{k(x-a)^2}{1+k(x-a)^2} & x > a \end{cases} \quad k > 0$$

S-function

$$A(x) = \begin{cases} 0 & x \leq a \\ 2(\frac{x-a}{b-a})^2 & x \in [a, m] \\ 1 - 2(\frac{x-b}{b-a})^2 & x \in [m, b] \\ 1 & x > b \end{cases}$$

The point $m = a + \frac{b}{2}$ is said to be the crossover point of the S-function.

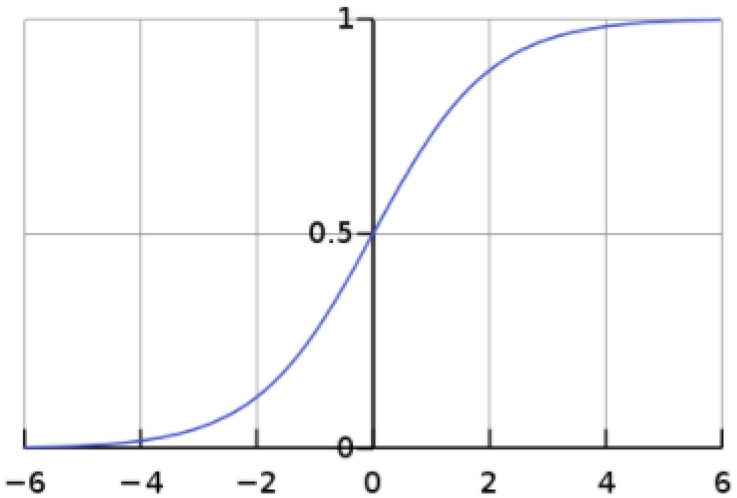


Figure D-14: S Function

Trapezoidal Function

$$A(x) = \begin{cases} 0 & x < a \\ \frac{x-a}{m-a} & x \in [a,m] \\ 1 & x \in [m,n] \\ \frac{b-x}{b-n} & x \in [n,b] \\ 0 & x > b \end{cases}$$

Using the equivalent notation, we obtain

$$A(x;a,m,n,b) = max(min[\frac{x-a}{m-a}, 1, \frac{b-x}{b-n}], 0)$$

Figure D-16 displays the shape of the trapezoidal function.

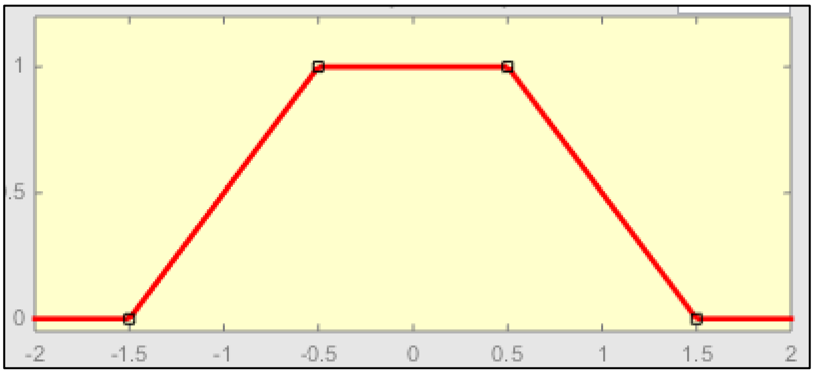


Figure D-15: Trapezoidal Function

Gaussian Function

$$A(x) = exp(-k(x-m)^2) \; where \; k > 0$$

Figure D-17 displays the shape of the Gaussian function.

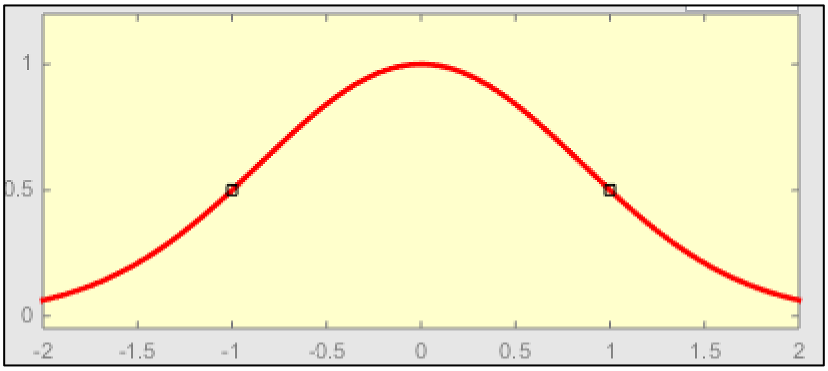


Figure D-16: Gaussian Function

Exponential Like Function

$$A(x) = \frac{1}{1 + k(k - m)^2} \qquad k > 1,$$
$$or$$
$$A(x) = \frac{k(x - m)^2}{1 + k(k - m)^2} \qquad k > 0$$

Figure D-18 demonstrations the shape of the Exponential like function.

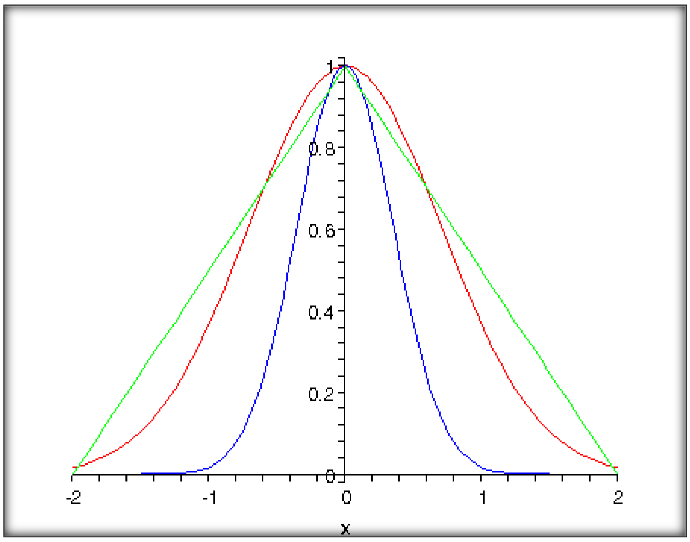


Figure D-17: Exponential Like Function

As discussed in the above, there are multiple membership functions. However, there are situations where combinations of fuzzy membership functions.

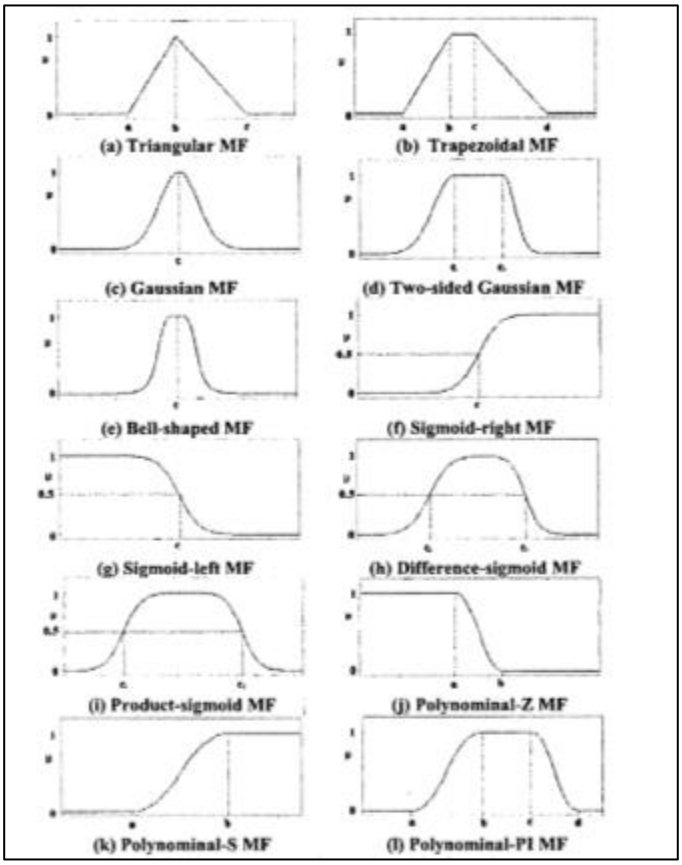


Figure D-18: Exponential Like Function

Appendix E: Operations in Fuzzy Sets

There are few operations which can be executed in fuzzy sets. A number of simple and useful operations may be performed on fuzzy sets.

Table E-4: Different Fuzzy Operations

Fuzzy Operations	Equation
Normalization	$Norm(A(x)) = \dfrac{A(x)}{hgt(A)}$
Concentration	$Con(A(x)) = A^p(x) \quad p > 1$
Dilation	$Dil(A(x)) = A^p(x)$ or $Dil(A(x)) = p\,A(x) - A^p(x)$ $where\ \ 0 < p < 1$
Contrast Intensification	$Int(A(x)) = \begin{cases} 2^{p-1}A^p(x) & 0 \leq A(x) \leq 0.5 \\ 1 - 2^{p-1}(1 - A(x))^p & A(x) > 0.5 \end{cases}$
Fuzzification	$Int(A(x)) = \begin{cases} \sqrt{\frac{A(x)}{2}} & 0 \leq A(x) \leq 0.5 \\ 1 - \sqrt{\frac{1-A(x)}{2}} & A(x) > 0.5 \end{cases}$

Appendix F: Fuzzy Membership Function Determination

Defining the Fuzzy membership function is an important process in this research. The method to select fuzzy membership function depends severely on the specifics of an application, specifically, the way the uncertainty is demonstrated and captured during the experiment.

There methods are

- Horizontal Method of Fuzzy Membership Estimation
- Vertical Method of Fuzzy Membership Estimation
- Problem Specification Based Fuzzy Membership Determination
- Fuzzy Membership Estimation as a Problem of Parametric Optimization
- Fuzzy Membership Estimation via Fuzzy Clustering
- Rank Order

Appendix G: Features of Fuzzy Membership Functions

Depending on the data set and the fuzzy membership function type, properties of the fuzzy membership function is different from one to other. It is important to identify the features of fuzzy membership functions to distinguish and compare them. Following are the main properties of the fuzzy membership function.

The core of a Membership Function

The core of a fuzzy membership function for a fuzzy set A is defined as that region of the universe that is characterized by complete or full membership in the set A. Therefore, the core of the membership function consists of all those elements X of the universe of discourse, such that

$$\mu_A(x) = 1$$

Support of a Membership Function

Support of a fuzzy membership function for a fuzzy set A is defined as that region of the universe that is characterized by non-zero membership in the fuzzy set A. therefore, support of a fuzzy membership function consists of all those elements X of the universe, such that

$$\mu_A(x) > 0$$

The boundary of a Membership Function

The boundary of a fuzzy membership function for a fuzzy set A is defined as that region of universe X, that is characterized by non-zero membership but not complete membership. Boundaries comprise that part of elements X of Universe of Discourse whose membership value is given by

$$\mu_A(x) \in (0, 1)$$

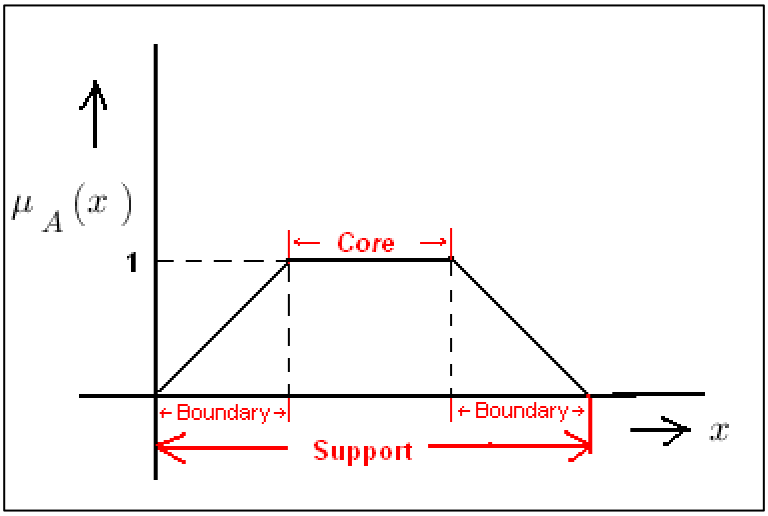


Figure G-19: Properties of Fuzzy Membership Function

Cross-over Points of a Fuzzy Membership Function

It is well-defined as the elements of a fuzzy set where membership value is equal to 0.5.

$$\mu_A(x) = 0.5$$

The height of a Membership Functions

The height of a fuzzy membership function is the maximum value of the fuzzy membership function. If the height of a fuzzy set is < 1 then it sub-normal fuzzy set. If the height is equal to 1 then it is said to be a normal fuzzy set.

Appendix H: Computing with Linguistic Variables

The values of a linguistic variable are labels of fuzzy sets on a universe X that have the form of sentences in a language. In general, a value of the linguistic variable is a composite term T = $L_1 L_2 L_3 \ldots L_n$, which is a concatenation of atomic terms $L_1 L_2 L_3 \ldots L_n$ as in *very high* and *not low* and *not very high*, for example.

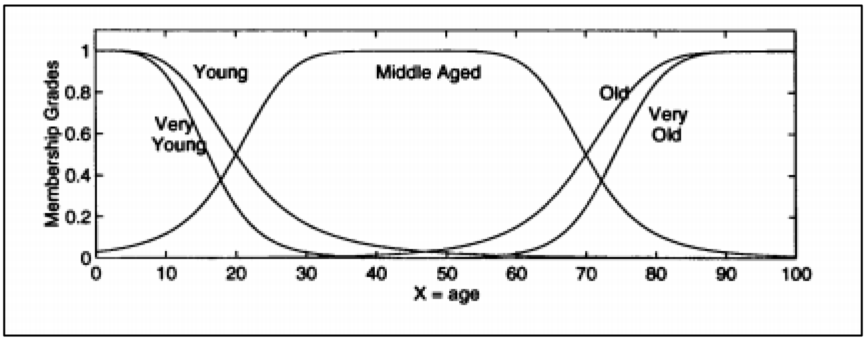


Figure H-20: Typical Membership Functions of the term set T (age) : Source: Mathworks

If age is construed as a linguistic variable, then its term set T (age) could be,

T (age) = {young, not young, very young, not very young, middle-aged, not middle-aged, old, not old, very old, extremely young, more or less old, not very old, not very young and not very old}

Primary terms: labels of *fuzzy* sets in X associated with their corresponding meaning, as *high* and *low* in the preceding example.

Hedges: act as a linguistic modifier of the primary terms such as *very*, *much*, *more* or *less*.

Connectives and negation: the connectives *and* and *or* and the negation *not* act as linguistic modifiers.

An important problem arises in this context: Given the fuzzy sets representing the meaning of each primary term and the meaning of the connectives and negation as well, compute the meaning of a composite term. A preliminary step in solving the problem is to consider first the problem involving composite terms of for, T = HL where H hedge and L is a term with specified meaning such that T = *very high* with H = *very* and L= *high*. The point in finding the meaning of T relies on viewing a hedge as an operator H that transforms that fuzzy set L(x), attached to the meaning of L, into the fuzzy set T(x) HL(x). For this purpose, there may have employed some of the basic operators introduced in Appendix F:

This modifies plus and minus which can be used to define some hedges whose meaning differs only slightly from others. For instance, *highly* could be defined as *highly = plus very* or possibly as *highly = minus very very*.

- If T = *much* L , then T(x) = Fuzz_L(x)
- If T= *more or less* then T(x) = DIL_L(x) with r = 0.5

After we know how to compute the meaning of a composite term such as T = HL, finding the meaning of more complex terms becomes easier. This includes the case where connectives and negation appear in addition with *union* and *not* with *complementation* the meaning of a composite term can be computed straightforwardly by inspection. For example, reconsidering the linguistic variable *temperature* and assuming a value such as temperature = not low and not very high, we have:

Not low and not very high = ^low ∩ ^ very high

Thus, considering standard complementation and viewing intersection as a t-norm, we get:

Not Low and Not Very High(x) = [1 – Low(x)] t [1- Con_High(x)],$(\forall x \in T.)$

Hedges can be defined based on the ideas just discussed. For instance, *slightly* might be defined using one of the following expressions.

Slightly L(x) = N0orm_L and Not Very L(x)

Slightly L(x) = Int_Norm_Plus L and Not Very L(x)

Slightly L(x) = int_Norm_Plus and Not Plus Very L(x)

After defining the linguistic variable composite linguistic values can be defined as shown below.

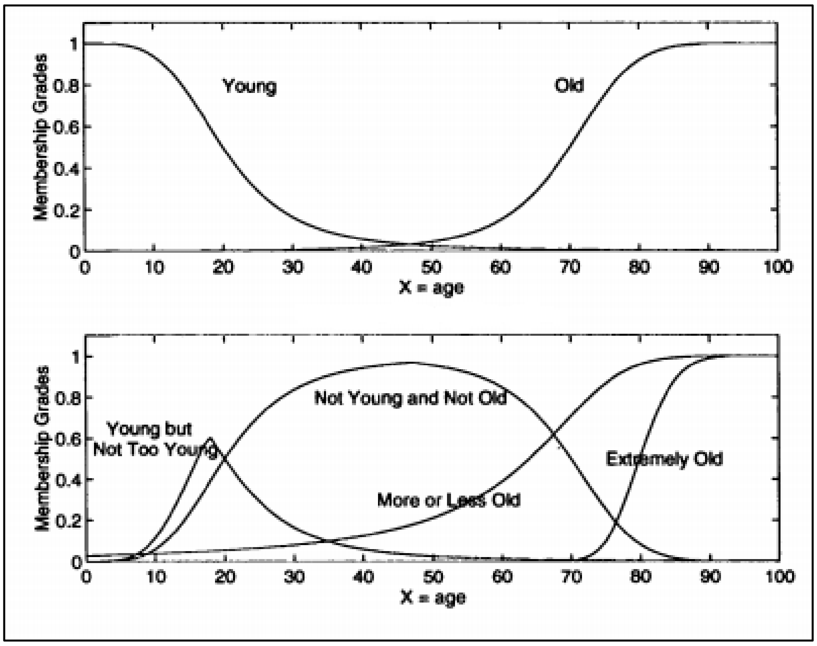


Figure H-21: Primary Linguistic Values and Composite Linguistic Value

Appendix J: Tools Used for the Research

Multiple tools are used in this research to achieve various objectives to prove the outcome of the research. From the proof of concept of perspective as well as implementation perspective, different tools need to be used. The research covers most of the stages in data warehousing such as extraction, transform, load, reporting, analysis, data mining, what of analysis, etc. Since it covers a vast area of data warehousing, the usage of different tools is essential.

Microsoft SQL Server

In Microsoft family, there are several tools which can be used to implement data warehousing. SQL Server 2016 was used as the primary tool for this research. Data can be stored in the SQL Server database engine. SQL Server Integration Service is used as an Extract-Transform-Load (ETL). Since there is a necessity to extract data from diverse of data sources, the industry-standard tool should be incorporated. SQL Server Integration Services (SSIS) is a member of Microsoft Business Intelligence family which is available from SQL Server 2005. For this research implementation, data sources are clipper database 3+ files which are supported in SSIS.

Figure J-23 indicates that Microsoft SQL Server is the third most ranked system in the industry as of August 2018.

343 systems in ranking, August 2018

Rank			DBMS	Database Model	Score		
Aug 2018	Jul 2018	Aug 2017			Aug 2018	Jul 2018	Aug 2017
1.	1.	1.	Oracle	Relational DBMS	1312.02	+34.24	-55.85
2.	2.	2.	MySQL	Relational DBMS	1206.81	+10.74	-133.49
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1072.65	+19.24	-152.82
4.	4.	4.	PostgreSQL	Relational DBMS	417.50	+11.69	+47.74
5.	5.	5.	MongoDB	Document store	350.98	+0.65	+20.48
6.	6.	6.	DB2	Relational DBMS	181.84	-4.36	-15.62
7.	7.	↑ 9.	Redis	Key-value store	138.58	-1.34	+16.68
8.	8.	↑ 10.	Elasticsearch	Search engine	138.12	+1.90	+20.47
9.	9.	↓ 7.	Microsoft Access	Relational DBMS	129.10	-3.48	+2.07
10.	10.	↓ 8.	Cassandra	Wide column store	119.58	-1.48	-7.14
11.	11.	11.	SQLite	Relational DBMS	113.73	-1.55	+2.88
12.	12.	12.	Teradata	Relational DBMS	77.41	-0.82	-1.83
13.	13.	↑ 16.	Splunk	Search engine	70.49	+1.26	+9.03
14.	14.	↑ 18.	MariaDB	Relational DBMS	68.29	+0.78	+13.60
15.	↑ 16.	↓ 13.	Solr	Search engine	61.90	+0.38	-5.06
16.	↓ 15.	↓ 14.	SAP Adaptive Server	Relational DBMS	60.44	-1.68	-6.48
17.	17.	↓ 15.	HBase	Wide column store	58.80	-1.97	-4.72
18.	18.	↑ 20.	Hive	Relational DBMS	57.94	+0.32	+10.64

Figure J-22: Most Popular Database Systems in 2018 August

Source: https://db-engines.com/en/ranking

Cube and data mining features are available along with KPI in SQL Server Analysis Service (SSAS). Further, in SQL Server 2016, integration to R is a vital feature to incorporate in this research.

R

R is one of the most popular open-source statistical tool. In 2015, R is ranked at six among the most popular languages as shown in Figure J-24.

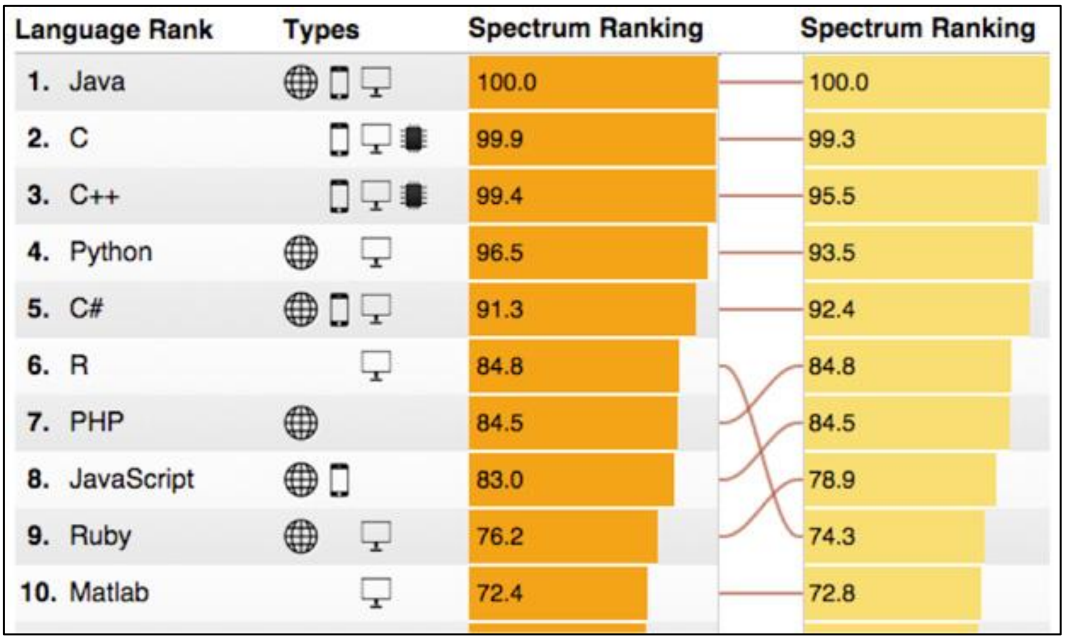


Figure J-23: Most Popular Software Programming Languages in 2015 Source: www.ieee.org

Inside R, there is a data mining tool called rattle which will also be used. Since SQL Server 2016 is being used.

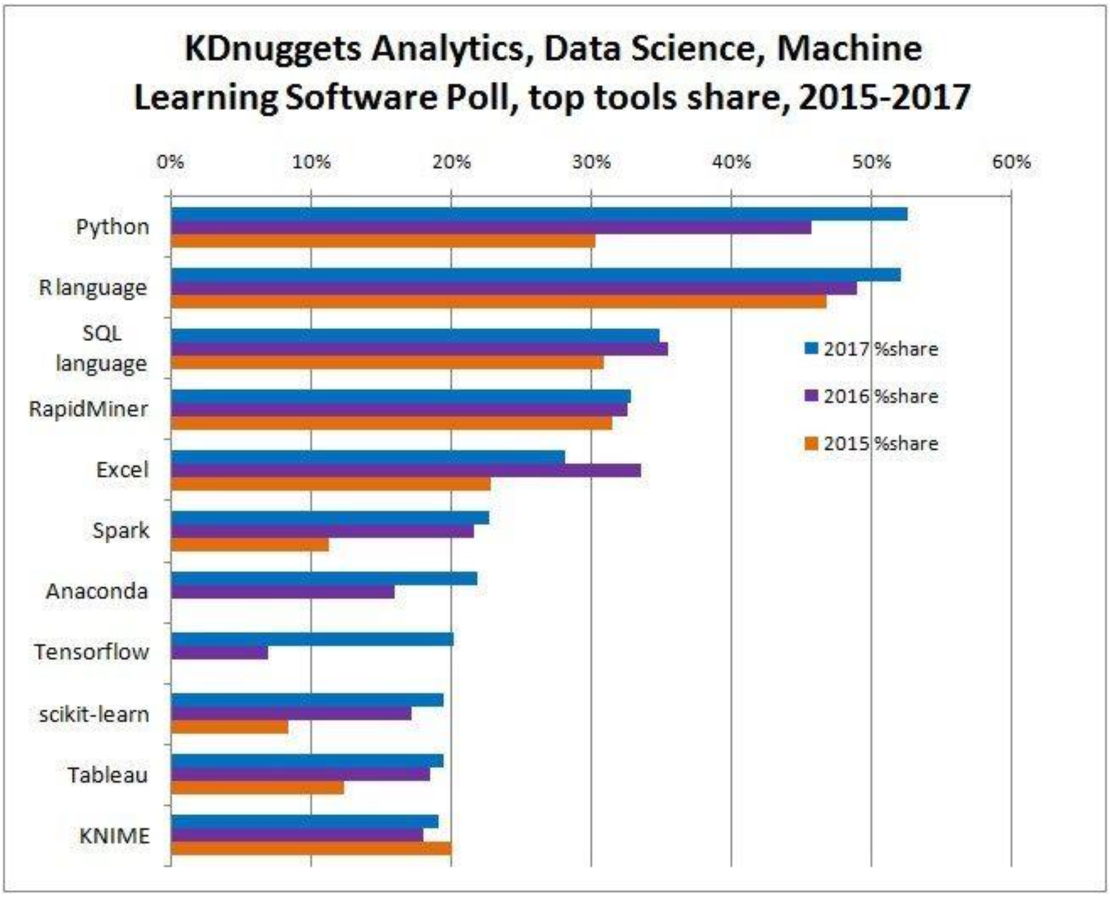


Figure J-24: KDnuggets Analytics, Data Science, Machine Learning Poll 2015-2017

Mat lab

Matlab is the most popular tool for mathematically oriented research. For fuzzy logic calculations, Matlab 2014b is used with Fuzzy Logic Designer add-ins.

Appendix K: Spider Diagram of Fuzzy Data Warehouse Research

Spider diagram was built to identify areas to be considered during the research. This was continuously built during the literature review and the design phase of the research.

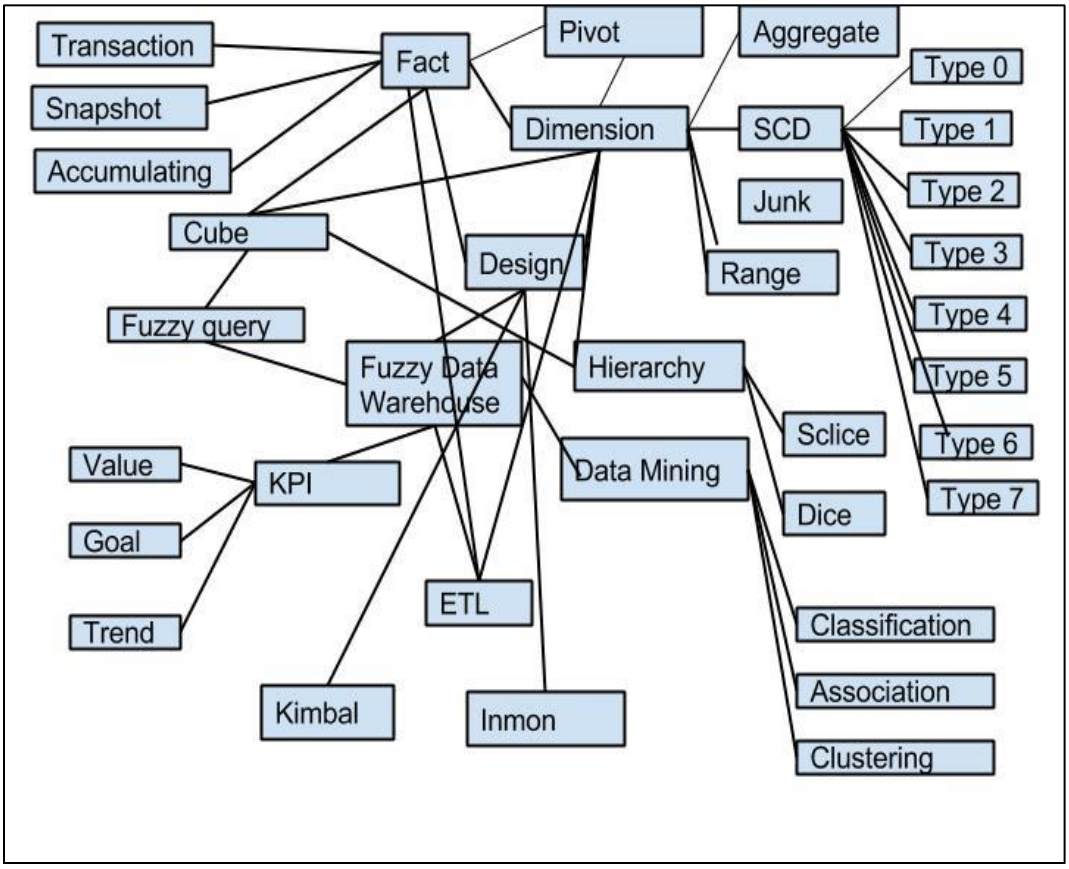


Figure K-25: Spider Diagram of Fuzzy Data Warehouse Research

Appendix L: Sample R Script for Model Box Plot

```
#Model for Box Plot
# MonthsofAge 150/1200
# MonthsOfExperience 1 / 720
# MonthsOfPermenent 1 / 720

#installing RODBC Package
install.packages("RODBC")
library("RODBC")

#Model Parameters
RangeMinValue <-  as.character(150)
RangeMaxValue <-  as.character(1200)
FuzzyAttribute = "MonthsofAge"
PartitionAttribute = 'MonthSK'
Sample = "200001,200101,200201,200301,200401,200501,200601, 200701, 200801,
200901,             201001, 201101, 201201,201301,201401,201501"

FactTable = "[dbo].[FctEmployee]"
YAxisLabel ="Age in Months"
chartHeading = "Box Plot Distribution for Permanent in Month"
chartHeading1 = "Average Box Plot for Permanent in Month"

#Connection
cn <- odbcDriverConnect(dbconnection = "Driver={SQL Server Native Client
11.0}; server=DA; database=DB_CHK_DW; trusted_connection=yes;")

#Selecting the query with the range
sqlq = paste("select ", PartitionAttribute )
sqlq = paste(sqlq," AS PartitionAttribute ,")
sqlq = paste(sqlq,FuzzyAttribute)
sqlq = paste(sqlq," AS FuzzyAttribute from ")
sqlq = paste(sqlq,FactTable)
sqlq = paste(sqlq," WHERE ")
sqlq = paste(sqlq,PartitionAttribute)
sqlq = paste(sqlq,"IN(")
sqlq = paste(sqlq ,Sample)
sqlq = paste(sqlq ,")")
sqlq = paste(sqlq , "AND  ")
sqlq = paste(sqlq ,FuzzyAttribute )
sqlq = paste(sqlq ,"> ")
sqlq = paste(sqlq, as.character(RangeMinValue))
sqlq = paste(sqlq, " AND ")
sqlq = paste(sqlq ,FuzzyAttribute )
sqlq = paste(sqlq ,"<")
sqlq = paste(sqlq, as.character(RangeMaxValue))

Age <- sqlQuery(cn,sqlq)

InputData<-data.frame(Age)

plot(InputData)
InputData
```

```
InputData$FuzzyAttribute<-I
       felse(InputData$FuzzyAttribute=="NULL",NA,InputData$FuzzyAttribute)

b<-boxplot(InputData$FuzzyAttribute~InputData$PartitionAttribute,
                      data=InputData,
                      main=chartHeading,
                       xlab="Year",
                       ylab= YAxisLabel,
                       notch=TRUE,
                      col=(c("lightgreen","darkgreen")))

#Calculating notch value percentage Confer Should be less than 5
mean(b$conf[1,])
mean(b$conf[2,])
Confper <- (mean(b$conf[2,]) - mean(b$conf[1,]) ) * 100 / mean(b$conf[2,])
Confper

#getting the average boxplot



c<- boxplot (c (mean(b$stats[1,]),
                mean(b$stats[2,]),
                mean(b$stats[3,]),
                mean(b$stats[4,]),
                mean(b$stats[5,])),
                main=chartHeading1,
                col=(c("lightgreen","darkgreen")),
                ylab=YAxisLabel)

Lt = matrix( c(mean(b$stats[1,]),
               mean(b$stats[2,]),
               mean(b$stats[3,]),
               mean(b$stats[4,]),
               mean(b$stats[5,]),
               sd(b$stats[1,]),
               sd(b$stats[2,]),
               sd(b$stats[3,]),
               sd(b$stats[4,]),
               sd(b$stats[5,]) ),
               nrow=5,
               ncol=2)

#Error % should be less than 10%
Lt

  #display values of c
c

#Three State Trigonometric / Trapezoidal Membership function

T_State1 <- matrix( c(0,
                    round(c$stats[1,1]),
                    0,
                    1,
                    round(c$stats[1,1]),
                    round(c$stats[3,1]),
                    round(-1 * 1 /(c$stats[3,1] - c$stats[1,1]),3),
                    round( c$stats[3,1]  /(c$stats[3,1] - c$stats[1,1]),3),
```

```
                    round(c$stats[3,1]),
                    as.integer(RangeMaxValue),
                    0,
                    0),
              nrow=4,
              ncol=3)

T_State1 <-t(T_State1)
T_State1

T_State2 <- matrix( c(0,
                    round(c$stats[1,1]),
                    0,
                    0,
                    round(c$stats[1,1]),
                    round(c$stats[3,1]),
                    round( 1 /(c$stats[3,1] - c$stats[1,1]),3),
                    round( -1 * c$stats[1,1]  /(c$stats[3,1] -
c$stats[1,1]),3),
                    round(c$stats[3,1]),
                    round(c$stats[5,1]),
                    round( -1 /(c$stats[5,1] - c$stats[3,1]),3),
                    round( 1 * c$stats[5,1]  /(c$stats[5,1] -
c$stats[3,1]),3),
                    round(c$stats[5,1]),
                    as.integer(RangeMaxValue),
                    0,
                    0  ),
            nrow=4,
            ncol=4)

T_State2 <-t(T_State2)
T_State2

T_State3 <- matrix(c(0,
                   round(c$stats[3,1]),
                   0,
                   0,
                   round(c$stats[3,1]),
                   round(c$stats[5,1]),
                   round( 1 /(c$stats[5,1] - c$stats[3,1]),3),
                   round( -1 * c$stats[3,1]  /(c$stats[5,1] -
c$stats[3,1]),3),
                   round(c$stats[5,1]),as.integer(RangeMaxValue),
                   0,
                   1),
            nrow=4,
            ncol=3)

T_State3 <-t(T_State3)
T_State3

# Five state schema
F_State1 <- matrix( c(0,
                   round(c$stats[1,1]),
                   0,
                   1,
                   round(c$stats[1,1]),
                   round(c$stats[2,1]),
```

```
                   round(-1 * 1 /(c$stats[2,1] - c$stats[1,1]),3),
                   round( c$stats[2,1]  /(c$stats[2,1] - c$stats[1,1]),3),
                   round(c$stats[2,1]),
                   as.integer(RangeMaxValue),
                   0,
                   0),
            nrow=4,
            ncol=3)

F_State1 <-t(F_State1)
F_State1

F_State2 <- matrix( c(0,
                      round(c$stats[1,1]),
                      0,
                      0,
                      round(c$stats[1,1]),
                      round(c$stats[2,1]),
                      round( 1 /(c$stats[2,1] - c$stats[1,1]),3),
                      round( -1 * c$stats[1,1]  /(c$stats[2,1] -
c$stats[1,1]),3),
                      round(c$stats[2,1]),
                      round(c$stats[3,1]),
                      round( -1 /(c$stats[3,1] - c$stats[2,1]),3),
                      round( 1 * c$stats[3,1]  /(c$stats[3,1] -
c$stats[2,1]),3),
                      round(c$stats[3,1]),
                      as.integer(RangeMaxValue),
                      0,
                      0
),
             nrow=4,
             ncol=4)

F_State2 <-t(F_State2)
F_State2

F_State3<- matrix( c(0,
                     round(c$stats[2,1]),
                     0,
                     0,
                     round(c$stats[2,1]),
                     round(c$stats[3,1]),
                     round( 1 /(c$stats[3,1] - c$stats[2,1]),3),
                     round( -1 * c$stats[2,1]  /(c$stats[3,1] -
c$stats[2,1]),3),
                     round(c$stats[3,1]),
                     round(c$stats[4,1]),
                     round( -1 /(c$stats[4,1] - c$stats[3,1]),3),
                     round( 1 * c$stats[4,1]  /(c$stats[4,1] -
c$stats[3,1]),3),
                     round(c$stats[4,1]),
                     as.integer(RangeMaxValue),
                     0,
                     0
),
          nrow=4,
          ncol=4 )
```

```
F_State3 <-t(F_State3)
F_State3

F_State4<- matrix( c(0,
                     round(c$stats[3,1]),
                     0,
                     0,
                     round(c$stats[3,1]),
                     round(c$stats[4,1]),
                     round( 1 /(c$stats[4,1] - c$stats[3,1]),3),
                     round( -1 * c$stats[3,1]  /(c$stats[4,1] -
c$stats[3,1]),3),
                     round(c$stats[4,1]),
                     round(c$stats[5,1]),
                     round( -1 /(c$stats[5,1] - c$stats[4,1]),3),
                     round( 1 * c$stats[5,1]  /(c$stats[5,1] -
c$stats[4,1]),3),
                     round(c$stats[5,1]),
                     as.integer(RangeMaxValue),
                     0,
                     0),
              nrow=4,
              ncol=4)

F_State4 <-t(F_State4)
F_State4

F_State5 <- matrix(c(0,round(c$stats[4,1]),
                     0,
                     0,
                     round(c$stats[4,1]),
                     round(c$stats[5,1]),
                     round( 1 /(c$stats[5,1] - c$stats[4,1]),3),
                     round( -1 * c$stats[4,1]  /(c$stats[5,1] -
c$stats[4,1]),3),
                     round(c$stats[5,1]),
                     as.integer(RangeMaxValue),
                     0,
                     1),
              nrow=4,
              ncol=3)

F_State5 <-t(F_State5)
F_State5
# End of code
```

Appendix M: NFR Lists

Non-Functional Requirements are an important factor in any research or product development. There are two standards to identify NFRs. There are Boehm's and McCall's NFR lists as shown in Figure M-27.

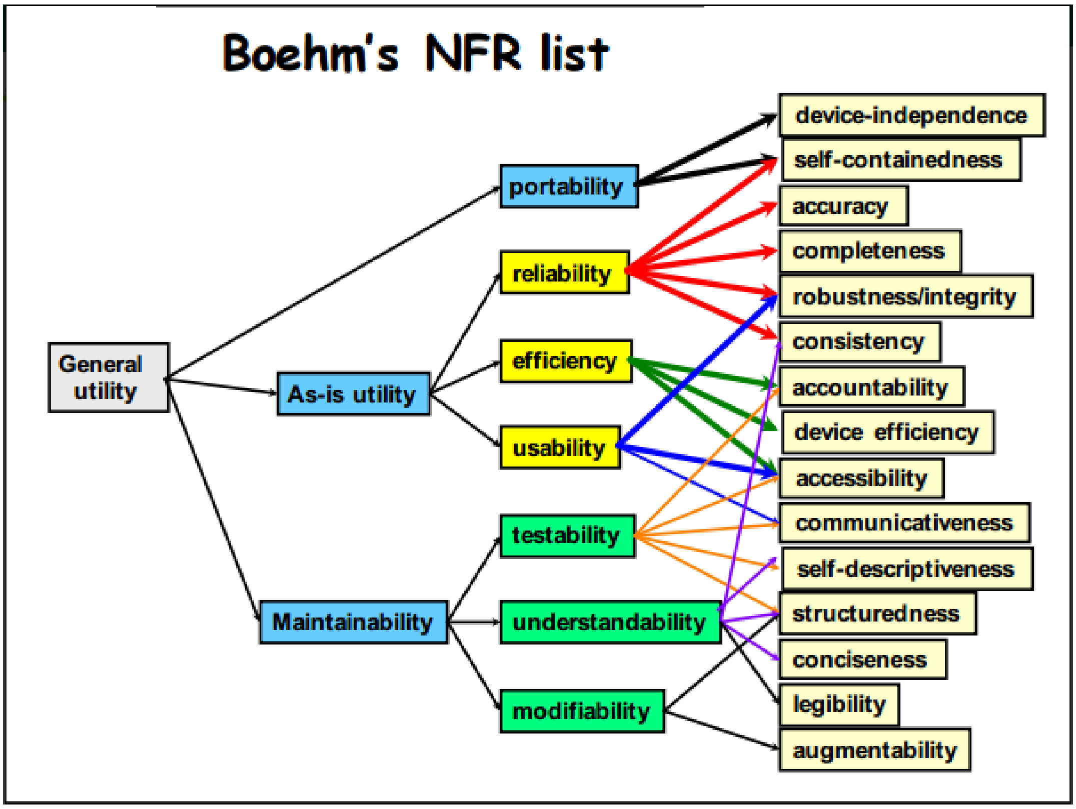


Figure M-26: Boehm's NFR List

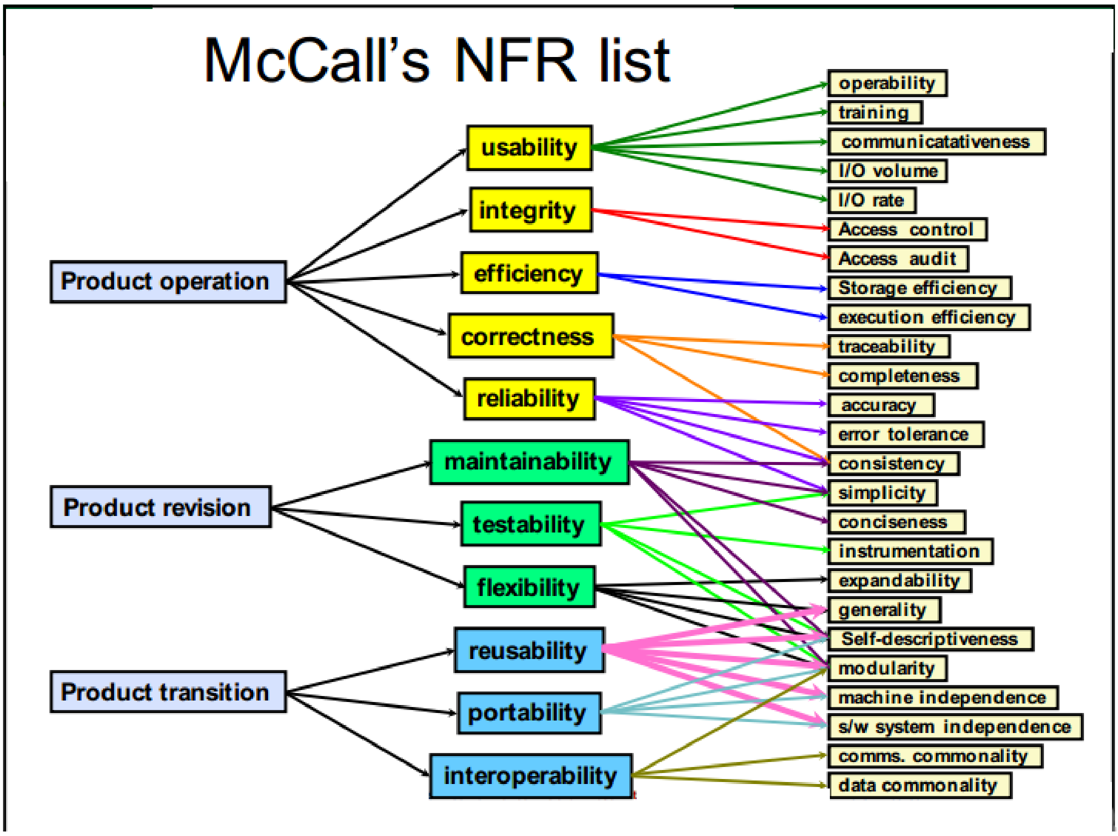


Figure M-27: McCall's NFR List

Appendix N: Script to Generate Fuzzy Contribution

```
CREATE PROC [dbo].[USP_GenerateFuzzyContribution]
 @MinValue INT,
 @MaxValue INT ,
 @Function VARCHAR(150)
AS
SET NOCOUNT ON
DECLARE @CurrVal INT = @MinValue
DELETE FROM [dbo].[FuzzyContribution]
WHERE [Function] = @Function
WHILE @CurrVal < @MaxValue
BEGIN
 SET @CurrVal = @CurrVal + 1
 --SELECT @CurrVal
 INSERT INTO  [dbo].[FuzzyContribution]
([Function],[Value],[State],[Contribution])
 SELECT @Function, @CurrVal,[State],(@CurrVal * Parameter1 ) + Parameter2
 FROM
 [dbo].[FuzzyMembershipFunctionConfiguration]
 WHERE @CurrVal BETWEEN Range1 AND Range2
 AND [IsCurrent] = 0X1
 AND [Function] = @Function
 AND ((@CurrVal * Parameter1 ) + Parameter2) > 0
 END
 --Validate and Error Correction
UPDATE [FuzzyContribution]
SET [Contribution] = 1
FROM [FuzzyContribution] F INNER JOIN
(
SELECT  [Function],[Value]
        FROM [DB_CHK_DW].[dbo].[FuzzyContribution]
  GROUP BY [Function],[Value]
  HAVING SUM(Contribution) <> 1
  ) A
 ON F.[Function] = A.[Function]
  AND A.[Value]  = F.[Value]
```

Appendix O: Script to Generate Fuzzy Contributions for Linguistic Functions

```
CREATE PROC [dbo].[USP_CalLingusticFuzzyContribution] @Function VARCHAR(25)
AS
SET NOCOUNT ON
DECLARE @MinVal INT
       ,@MaxVal INT
DELETE [dbo].[FuzzyContribution]
WHERE [CalculationMethod] = 'L'
       AND [Function] = @Function
INSERT INTO [dbo].[FuzzyContribution] (
       [Function]
       ,[Value]
       ,[State]
       ,[Contribution]
       ,[CalculationMethod]
       )
SELECT FMFLC.[Function]
       ,FC.[Value]
       ,FMFLC.[State]
       ,CASE
              WHEN FMFLC.LinguisticType = 'DIL'
                     OR FMFLC.LinguisticType = 'CON'
                     THEN POWER(FC.Contribution, FMFLC.Parameter1)
              END Contribution
       ,'L' [CalculationMethod]
FROM [FuzzyMembershipFunctionLingusitcConfiguration] FMFLC
INNER JOIN FuzzyContribution FC ON FMFLC.[Function] = FC.[Function]
       AND FMFLC.[BaseState1] = FC.[State]
WHERE FMFLC.id IN (
              SELECT id
              FROM [FuzzyMembershipFunctionLingusitcConfiguration]
              CROSS APPLY string_split([State], ' ')
              GROUP BY id
              HAVING count(value) = 2
              )
       AND FMFLC.[Function] = @Function
       AND FMFLC.LinguisticType != 'NOT'

--NOT
INSERT INTO [dbo].[FuzzyContribution] (
       [Function]
       ,[Value]
       ,[State]
       ,[Contribution]
       ,[CalculationMethod]
       )
SELECT FMFLC.[Function]
       ,FC.[Value]
       ,FMFLC.[State]
       ,1 - FC.Contribution Contribution
       ,'L'
FROM [FuzzyMembershipFunctionLingusitcConfiguration] FMFLC
INNER JOIN [FuzzyContribution] FC ON FMFLC.[Function] = FC.[Function]
WHERE SUBSTRING(FMFLC.[STATE], 1, 3) = 'NOT'
```

```
        AND CHARINDEX('OR', FMFLC.[STATE]) = 0
        AND CHARINDEX('AND', FMFLC.[STATE]) = 0
        AND FC.[STATE] = LTRIM(REPLACE(FMFLC.[STATE], 'NOT', ''))
        --      AND FC.Contribution <> 1
        AND FC.[Function] = @Function

SELECT @MinVal = MIN([VALUE])
        ,@MaxVal = MAX([VALUE])
FROM [FuzzyContribution]
WHERE [Function] = @Function

INSERT INTO [dbo].[FuzzyContribution] (
        [Function]
        ,[Value]
        ,[State]
        ,[Contribution]
        ,[CalculationMethod]
        )
SELECT @Function
        ,B.num
        ,B.STATE
        ,1
        ,'L'
FROM (
        SELECT *
        FROM dbo.GenerateSequenceNumber(@MinVal, @MaxVal)
        CROSS JOIN (
                SELECT DISTINCT [State]
                FROM [FuzzyContribution]
                WHERE [Function] = @Function
                        AND LEFT([State], 3) = 'NOT'
                ) A
        ) B
LEFT OUTER JOIN (
        SELECT *
        FROM [FuzzyContribution]
        WHERE [Function] = @Function
                AND LEFT([State], 3) = 'NOT'
        ) C ON B.num = C.[Value]
        AND B.[State] = C.[State]
WHERE C.[Value] IS NULL
OPTION (MAXRECURSION 10000)

--AND OR
INSERT INTO [dbo].[FuzzyContribution] (
        [Function]
        ,[Value]
        ,[State]
        ,[Contribution]
        ,[CalculationMethod]
        )
SELECT Data.[Function]
        ,COALESCE(FC1.Value, FC2.Value) Value
        ,Data.[State]
        ,CASE
                WHEN Conc = 'AND'
                        AND (ISNULL(FC1.Contribution, 0) >=
ISNULL(FC2.Contribution, 0))
                        THEN ISNULL(FC2.Contribution, 0)
```

```
		WHEN Conc = 'AND'
			AND ISNULL(FC1.Contribution, 0) < ISNULL(FC2.Contribution, 
0)
			THEN ISNULL(FC1.Contribution, 0)
		WHEN Conc = 'OR'
			AND (ISNULL(FC1.Contribution, 0) >= 
ISNULL(FC2.Contribution, 0))
			THEN ISNULL(FC1.Contribution, 0)
		WHEN Conc = 'OR'
			AND ISNULL(FC1.Contribution, 0) < ISNULL(FC2.Contribution, 
0)
			THEN ISNULL(FC2.Contribution, 0)
		END Contribution
	,'L' [CalculationMethod]
FROM (
	SELECT [STATE]
		,[Function]
		,'AND' Conc
		,SUBSTRING([STATE], 1, CHARINDEX('AND', [STATE]) - 1) State1
		,SUBSTRING([STATE], CHARINDEX('AND', [STATE]) + 4, LEN([STATE]) 
- CHARINDEX('AND', [STATE])) State2
	FROM [FuzzyMembershipFunctionLingusitcConfiguration]
	WHERE CHARINDEX('AND', [STATE]) > 0
		AND [Function] = @Function

	UNION

	SELECT [STATE]
		,[Function]
		,'OR' Conc
		,SUBSTRING([STATE], 1, CHARINDEX('OR', [STATE]) - 1) State1
		,SUBSTRING([STATE], CHARINDEX('OR', [STATE]) + 3, LEN([STATE]) - 
CHARINDEX('OR', [STATE])) State2
	FROM [FuzzyMembershipFunctionLingusitcConfiguration]
	WHERE CHARINDEX('OR', [STATE]) > 0
		AND [Function] = @Function
	) Data
LEFT OUTER JOIN FuzzyContribution FC1 ON FC1.[Function] = Data.[Function]
	AND FC1.STATE = Data.State1
LEFT OUTER JOIN FuzzyContribution FC2 ON FC2.[Function] = Data.[Function]
	AND FC2.STATE = Data.State2
WHERE fc1.[VALUE] = fc2.[VALUE]
```

Appendix P: Scripts to Evaluate Fuzzy State Contributions

```
Script 1 – Comparing Fuzzy State and Crisp State
SELECT FctEmployee.EmployeeSK
       ,FctEmployee.ExperienceRangeSK
       ,FctEmployee.MonthsOfExperience
       ,FctEmployee.MonthSK
       ,FuzzyContribution.Value
       ,FuzzyContribution.STATE
       ,FuzzyContribution.Contribution
       ,DimRange.RangeDescription
FROM FctEmployee
INNER JOIN FuzzyContribution ON FctEmployee.MonthsOfExperience =
FuzzyContribution.[Value]
INNER JOIN DimRange ON FctEmployee.ExperienceRangeSK = DimRange.RangeSK
WHERE (FuzzyContribution.[Function] = 'Experience_3')
       AND (FuzzyContribution.CalculationMethod = 'P')
Script 2 – Pivot Table for Employee Experience
SELECT EmployeeSK
       ,MonthsOfExperience
       ,MonthSK
       ,ISNULL([Low], 0) Low
       ,ISNULL([Medium], 0) [Medium]
       ,ISNULL([High], 0) High
       ,ISNULL([Low], 0) + ISNULL([Medium], 0) + ISNULL([High], 0) Total
FROM (
       SELECT FctEmployee.EmployeeSK
              ,FctEmployee.MonthsOfExperience
              ,FctEmployee.MonthSK
              ,FuzzyContribution.[STATE]
              ,FuzzyContribution.Contribution
       FROM FctEmployee
       INNER JOIN FuzzyContribution ON FctEmployee.MonthsOfExperience =
FuzzyContribution.[Value]
       WHERE (FuzzyContribution.[Function] = 'Experience_3')
              AND (FuzzyContribution.CalculationMethod = 'P')
       ) p
PIVOT(SUM(Contribution) FOR [State] IN (
                     [Low]
                     ,[Medium]
                     ,[High]
                     )) AS pvt
Script 3 – Evaluation of Fuzzy State and Crisp State
SELECT FctEmployee.EmployeeSK
       ,FctEmployee.ExperienceRangeSK
       ,FctEmployee.MonthsOfExperience
       ,FctEmployee.MonthSK
       ,FuzzyContribution.Value
       ,FuzzyContribution.STATE
       ,FuzzyContribution.Contribution
       ,DimRange.RangeDescription
       ,CASE WHEN  FuzzyContribution.State = DimRange.RangeDescription THEN
'Same'
        ELSE 'Different'
     END Compare_Experience_Category_FuzzyCategory
FROM FctEmployee
```

```
INNER JOIN FuzzyContribution ON FctEmployee.MonthsOfExperience =
FuzzyContribution.[Value]
INNER JOIN DimRange ON FctEmployee.ExperienceRangeSK = DimRange.RangeSK
WHERE (FuzzyContribution.[Function] = 'Experience_3')
       AND (FuzzyContribution.CalculationMethod = 'P')
       AND FuzzyContribution.Contribution >= 0.5
```

Appendix Q: Scripts to List Properties of Function Membership

```
CREATE PROC USP_GET_Fuzzy_Membership_Properties @Function AS VARCHAR(50)
AS
--Support
SELECT [State]
       ,MIN([Value]) [Support_Start]
       ,MAX([Value]) [Support_End]
FROM [dbo].[FuzzyContribution]
WHERE CalculationMethod = 'P'
       AND [function] = @Function
GROUP BY [State]
ORDER BY 2

--Core
SELECT [State]
       ,MIN([Value]) [Core_Start]
       ,MAX([Value]) [Core_End]
FROM [dbo].[FuzzyContribution]
WHERE CalculationMethod = 'P'
       AND [function] = @Function
       AND [Contribution] = 1
GROUP BY [State]
ORDER BY 2

--Crossover
SELECT [Function]
       ,[State]
       ,CONVERT(NUMERIC(9, 2), (0.5 - Parameter2) / (Parameter1))
CrossOverPoint
FROM [DB_CHK_DW].[dbo].[FuzzyMembershipFunctionConfiguration]
WHERE Parameter1 <> 0
       AND Parameter2 <> 0
       AND [FunctionType] = 'Linear'
       AND [function] = @Function

--Boundary
SELECT Core.[State]
       ,CASE
              WHEN Core_Start = Support_Start
                     THEN CAST(Core_END AS VARCHAR(50)) + ' - ' +
CAST(Support_END AS VARCHAR(50))
              WHEN Core_End = Support_End
                     THEN CAST(Support_Start AS VARCHAR(50)) + ' - ' +
CAST(Core_Start AS VARCHAR(50))
              ELSE CAST(Support_start AS VARCHAR(50)) + ' - ' +
CAST(Core_start AS VARCHAR(50)) + ' / ' + CAST(Core_End AS VARCHAR(50)) + ' -
' + CAST(Support_End AS VARCHAR(50))
              END AS Boundary
FROM (
       SELECT [State]
              ,MIN([Value]) [Core_Start]
              ,MAX([Value]) [Core_End]
       FROM [dbo].[FuzzyContribution]
       WHERE CalculationMethod = 'P'
              AND [function] = @Function
              AND [Contribution] = 1
```

```
        GROUP BY [State]
        ) Core
INNER JOIN (
        SELECT [State]
                ,MIN([Value]) [Support_Start]
                ,MAX([Value]) [Support_End]
        FROM [dbo].[FuzzyContribution]
        WHERE CalculationMethod = 'P'
                AND [function] = @Function
        GROUP BY [State]
        ) Support ON Core.[State] = Support.[State]
ORDER BY 2
```

Appendix R: Scripts to Generate Membership Function for Subjective Method

```
CREATE PROCEDURE [dbo].[USP_GenerateFuzzyMembershipforArbitaryValues]
    @Function VARCHAR(50) ,
       @State VARCHAR(50),
       @valuepoints NVARCHAR(400),
       @Type VARCHAR(6)
AS
SET NOCOUNT ON
DECLARE @IsValidParameters BIT = 0x1
DECLARE @Count TINYINT
DECLARE @Val1 NUMERIC(9, 2),
       @Val2 NUMERIC(9, 2),
       @Val3 NUMERIC(9, 2),
       @Val4 NUMERIC(9, 2)
DECLARE @i TINYINT = 0

CREATE TABLE #T (
       id INT identity,
       value NUMERIC(9, 2)
       )

INSERT INTO #t (value)
SELECT NAME
FROM splitstring(@valuepoints)

CREATE TABLE [#FuzzyMembershipFunctionConfiguration] (
       [Range1] FLOAT,
       [Range2] FLOAT,
       [Parameter1] FLOAT,
       [Parameter2] FLOAT
       )

WHILE @I < 5
BEGIN
       SET @i = @i + 1

       IF @I = 1
              SELECT @Val1 = value
              FROM #t
              WHERE id = 1
       ELSE IF @i = 2
              SELECT @Val2 = value
              FROM #t
              WHERE id = 2
       ELSE IF @i = 3
              SELECT @Val3 = value
              FROM #t
              WHERE id = 3
       ELSE IF @i = 4
              SELECT @Val4 = value
              FROM #t
              WHERE id = 4
END
```

```
--Validation of Parameters
SELECT @Count = count(NAME)
FROM splitstring(@valuepoints)

IF @Type NOT IN (
              'TRAP',
              'TRIG'
              )
BEGIN
       PRINT 'Invalid membership function type'

       SET @IsValidParameters = 0x0
END
ELSE IF (
              @Type = 'TRIG'
              AND @Count != 3
              )
BEGIN
       PRINT 'Trigonometric function should have three parameters'

       SET @IsValidParameters = 0x0
END
ELSE IF (
              @Type = 'TRAP'
              AND @Count != 4
              )
BEGIN
       PRINT 'Trapezoidal function should have four parameters'

       SET @IsValidParameters = 0x0
END

-- Delete existing current data
DELETE
FROM [FuzzyMembershipFunctionConfiguration]
WHERE [Function] = @Function
       AND [State] = @State
       AND IsCurrent = 0x1

IF (@IsValidParameters = 1)
BEGIN
       IF @Type = 'TRAP'
       BEGIN
              IF (
                            @Val1 = 0
                            AND @Val1 = @Val2
                            AND @Val2 <> @Val3
                            )
              BEGIN
                     INSERT INTO [#FuzzyMembershipFunctionConfiguration] (
                            [Range1],
                            [Range2],
                            [Parameter1],
                            [Parameter2]
                            )
                     VALUES (
                            @Val1,
                            @Val3,
                            0,
```

```
				1
				)

			INSERT INTO [#FuzzyMembershipFunctionConfiguration] (
				[Range1],
				[Range2],
				[Parameter1],
				[Parameter2]
				)
			VALUES (
				@Val3 + 0.01,
				@Val4,
				- 1 / (@Val4 - @Val3),
				@Val4 / (@Val4 - @Val3)
				)
		END

		IF (    @Val1 <> 0
		AND
				@Val1 = @Val2
				AND @Val2 <> @Val3
				)
		BEGIN
			INSERT INTO [#FuzzyMembershipFunctionConfiguration] (
				[Range1],
				[Range2],
				[Parameter1],
				[Parameter2]
				)
			VALUES (
				@Val2,
				@Val3,
				1 / (@Val3 - @Val2),
				- 1 * @Val2 / (@Val3 - @Val2)
				)

			INSERT INTO [#FuzzyMembershipFunctionConfiguration] (
				[Range1],
				[Range2],
				[Parameter1],
				[Parameter2]
				)
			VALUES (
				@Val3 + 0.01,
				@Val4,
				0,
				1
				)
		END

		IF (
				@Val1 <> @Val2
				AND @Val2 <> @Val3
				AND @Val3 <> @Val4
				)
		BEGIN
			INSERT INTO [#FuzzyMembershipFunctionConfiguration] (
				[Range1],
				[Range2],
```

```
                [Parameter1],
                [Parameter2]
                )
            VALUES (
                @Val1,
                @Val2,
                1 / (@Val2 - @Val1),
                - 1 * @Val1 / (@Val2 - @Val1)
                )

            INSERT INTO [#FuzzyMembershipFunctionConfiguration] (
                [Range1],
                [Range2],
                [Parameter1],
                [Parameter2]
                )
            VALUES (
                @Val2 + 0.01,
                @Val3,
                0,
                1
                )

            INSERT INTO [#FuzzyMembershipFunctionConfiguration] (
                [Range1],
                [Range2],
                [Parameter1],
                [Parameter2]
                )
            VALUES (
                @Val3 + 0.01,
                @Val4,
                - 1 / (@Val4 - @Val3),
                @Val4 / (@Val4 - @Val3)
                )
        END
END
ELSE IF @Type = 'TRIG'
BEGIN
        INSERT INTO [#FuzzyMembershipFunctionConfiguration] (
            [Range1],
            [Range2],
            [Parameter1],
            [Parameter2]
            )
        VALUES (
            @Val1,
            @Val2,
            1 / (@Val2 - @Val1),
            - 1 * @Val1 / (@Val2 - @Val1)
            )

        INSERT INTO [#FuzzyMembershipFunctionConfiguration] (
            [Range1],
            [Range2],
            [Parameter1],
            [Parameter2]
            )
        VALUES (
```

```
				@Val2 + 0.01,
				@Val3,
				- 1 / (@Val3 - @Val2),
				@Val3 / (@Val3 - @Val2)
				)
	END

	INSERT INTO FuzzyMembershipFunctionConfiguration
	SELECT @Function [FUNCTION],
		@State [State],
		'Linear' FunctionType,
		*,
		CAST(GetDATE() AS DATE) ValidStartDate,
		NULL ValidENDDate,
		1 IsCurrent
	FROM [#FuzzyMembershipFunctionConfiguration]

	DROP TABLE [#FuzzyMembershipFunctionConfiguration]
END
GO
```

Appendix S: Performance of Execution of Fuzzy Configuration Steps

Following are the timing for each operation as shown in Table S-5.

Table S-5: Different Durations to Generate Fuzzy Contribution

Instance	**Duration**
[USP_GenerateFuzzyContribution] 0,999999,'Hectare'	935 Seconds
[USP_GenerateFuzzyContribution] 0,999999,'Trees'	157 Seconds

Following are the duration comparison for fuzzy OLAP cubes and standard OLAP cubes.

Table S-6: Different Durations to Standard and Fuzzy OLAP Cubes

OLAP Cube Type	**Duration**
Standard OLAP Cube	330 Seconds
Fuzzy OLAP Cube	540 Seconds

This shows that processing of fuzzy OLAP cube duration is significantly larger than the processing of standard OLAP cube.

Appendix T: Proof of Summation of Derived Membership Contribution Value Equal to One

Given,

$m_a + n_a = 1$ --------------- EQ 1

$p_b + q_b = 1$ --------------- EQ 2

as summation of fuzzy membership function is equal to one

Summation of derived fuzzy membership function values = $m_a\, p_b + m_a\, q_b + n_a\, p_b + n_a\, q_b$ --------------- EQ 3

$= m_a\,(p_b + q_b) + n_a\,(p_b + q_b)$

$= (m_a + n_a) + (p_b + q_b)$

From EQ 1, EQ 2,

= 1

Appendix U: Script to Calculate of Derived Membership Function Contribution

```
SELECT H.DivisionCode,
       H.PlaceofWork,
       P.[Importance],
       SUM(H.Contribution * T.Contribution ) Contribution

FROM (
       SELECT P.DivisionCode,
              P.PlaceofWork,
              P.Hectaire,
              FC1.Value,
              FC1.STATE,
              FC1.Contribution
       FROM DimPlaceofWork P
       INNER JOIN [dbo].[FuzzyContribution] FC1
              ON ROUND(P.Hectaire, 0) = FC1.Value
       WHERE P.Hectaire > 0
              AND P.Trees > 0
              AND FC1.[Function] = 'Hectare'
       ) H
INNER JOIN (
       SELECT P.DivisionCode,
              P.PlaceofWork,
              P.Trees,
              FC1.Value,
              FC1.STATE,
              FC1.Contribution
       FROM DimPlaceofWork P
       INNER JOIN [dbo].[FuzzyContribution] FC1
              ON ROUND(P.Trees, 0) = FC1.Value
       WHERE P.Hectaire > 0
              AND P.Trees > 0
              AND FC1.[Function] = 'Trees'
       ) T
       ON H.DivisionCode = T.DivisionCode
              AND H.PlaceofWork = T.PlaceofWork
INNER JOIN PlaceOfWorkImportanceMatrix P ON
P.[Hectaire] = H.STATE
AND P.Trees = T.STATE
GROUP BY  H.DivisionCode,
       H.PlaceofWork,
       P.[Importance]
ORDER BY 1,
       2
```

Appendix V: Calculation of Derived Membership Function

Calculation of derived membership function is given in the following steps. In the following example, 01B field of LD division was selected as it is a complex scenario.

Table V-7 shows membership contribution for the selected field hectare where it is 3.

Table V-7: Membership Contribution for Selected Field Hectare

DivisionCode	PlaceofWork	Hectare	State	Contribution
LD	01B	3	Low	0.6000
LD	01B	3	Medium	0.4000

Table V-8 shows membership contribution for the selected field trees where it is 3452.

Table V-8: Membership Contribution for Selected Field Trees

DivisionCode	PlaceofWork	Trees	State	Contribution
LD	01B	3452	High	0.1033
LD	01B	3452	Medium	0.8967

In Table V-9, a combination of membership functions is displayed. In this, the cross product of both rows was listed below.

Table V-9: Combination of Membership Functions of Trees and Hectare

Division Code	Place Of Work	Hectare	Hectare	Contribution	Trees	Trees	Contribution
LD	01B	3	Low	0.6000	3452	High	0.1033
LD	01B	3	Medium	0.4000	3452	High	0.1033
LD	01B	3	Low	0.6000	3452	Medium	0.8967
LD	01B	3	Medium	0.4000	3452	Medium	0.8967

The importance which is the derived function from Hectare and Number of trees have few sets of rules. In Table V-10, relevant rules are listed.

Table V-10: Rules for Importance

Hectare	Number of Trees	Importance
Low	Medium	Medium
Low	High	Medium
Medium	Medium	Medium
Medium	High	High

Next step as shown in Table V-11 is the contribution calculation which is the multiplication of individual contribution of Hectare and Trees.

Table V-11: Multiplication of Contribution

Division Code	Place Of Work	Hectare	Contribution	Trees	Contribution	Importance	Contribution
LD	01B	Low	0.6000	High	0.1033	Medium	0.0620
LD	01B	Medium	0.4000	High	0.1033	High	0.0413
LD	01B	Low	0.6000	Medium	0.8967	Medium	0.5380
LD	01B	Medium	0.4000	Medium	0.8967	Medium	0.3587

As identified in the Importance state, there is a duplicate state. Those will be aggregated and shown in Table V-12.

Table V-12: Aggregation of Contribution for Importance

DivisionCode	PlaceofWork	Importance	Contribution
LD	01B	High	0.04132
LD	01B	Medium	0.95868

Appendix W: Scripts for Calculation of Fuzzy Categorization

Categorization is used in a data warehouse for better analysis of data. Following query is used in the Sample database of Microsoft SQL Server named AdventureWorksDW2016.

```
SELECT PR.[EnglishProductSubcategoryName],
       PR.[EnglishProductName],
       SUM([SalesAmount]) Total_Amount
FROM dbo.FacTInternetsales FS
INNER JOIN [dbo].[DimProducts] PR
       ON FS.ProductKey = PR.ProductKey
GROUP BY PR.[EnglishProductSubcategoryName],
       PR.[EnglishProductName]
ORDER BY PR.[EnglishProductSubcategoryName],
       PR.[EnglishProductName]
```

This query provides classical categorization for the data warehouse. As shown in the Figure W-13, each product is tagged to one category or subcategory.

Table W-13: Output for Traditional Categorization of Data Warehouse

ProductSubcategoryName	ProductName	Total_Amount
Bike Racks	Hitch Rack - 4-Bike	39,360.00
Bike Stands	All-Purpose Bike Stand	39,591.00
Bottles and Cages	Mountain Bottle Cage	20,229.75
Bottles and Cages	Road Bottle Cage	15,390.88
Bottles and Cages	Water Bottle - 30 oz.	21,177.56
Caps	AWC Logo Cap	19,688.10
Cleaners	Bike Wash - Dissolver	7,218.60
Fenders	Fender Set - Mountain	46,619.58
Gloves	Half-Finger Gloves, L	10,849.07
Gloves	Half-Finger Gloves, M	12,220.51

This query provides classical categorization for the data warehouse. As shown in the below figure, each product is tagged to one category or subcategory.

As described in Chapter 8, one product can be in multiple categorizations with different weightage. The following query will provide different categories of one product and relevant aggregation.

```
SELECT P.ProductName,
       C.CategoryName,
       SUM([SalesAmount] * lnk.Weightage) Total_Amount_PerCategory,
       SUM([SalesAmount]) TotalAmount
FROM FaCTInternetsales FS
```

```
INNER JOIN [dbo].[DimProductF] PR
       ON FS.ProductKey = PR.ProductKey
INNER JOIN [dbo].[lnk_ProductCategoryFuzzyWeightages] lnk
       ON lnk.ProductKey = PR.ProductKey
INNER JOIN [dbo].[DimCategoryF] C
       ON C.CategoryKey = lnk.CategoryKey
WHERE lnk.IsCurrent = 'Yes'
GROUP BY PR.ProductName,
       C.CategoryName
ORDER BY C.CategoryName,
       P.ProductName
```

In the above query, weightage and amount are multiplied together to get the weightaged amount.

Table W-14: Output for Fuzzy Categorization of Data Warehouse

Product Name	Category Name	Total Amount Per Category	Total Amount
CoconutOil	Cooking	46,816.62	78,027.70
CoconutOil	Cosmetic	23,408.31	78,027.70
CoconutOil	Other	7,802.77	78,027.70

In the given example Coconut oil, categorized to Cooking, Cosmetic and others in 0.6, 0.3 and 0.1 respectively. The total amount of Coconut oil is divided into the same weightages.

Appendix X: Subsystems of ETL

As described by Ralph Kimball, there are 34 sub-systems for ETL which are listed three categories Transformation, Audit and Maintenance.

Transformation	**Audit**	**Maintenance**
Data Profiling	Audit Dimension Assembler	Job Scheduler
Change Data Capture System	Conforming System	Backup System
Extract System	Dimension Manager System	Recovery and Restart System
Data Cleaning System	Sorting System	Security System
Error Event Schema	Compliance Manager	Problem Escalation System
Slowly Changing Dimension Manager	Workflow Monitor	Late Arriving Data Handler
De-duplication System	Version Control System	Parallelizing/Pipelining System
Surrogate Key Generator	Data Propagation Manager	Metadata Repository Manager
Surrogate Key Pipeline	Lineage and Dependency Analyzer	

Transformation	**Audit**	**Maintenance**
Hierarchy Manager	Multivalued Dimension Bridge Table Builder	
Special Dimension Manager		
Fact Table Builders		
Fact Provider System		
Aggregate Builder		
OLAP Cube Builder		

Appendix Y: Scripts for Fuzzy OLAP Cube

Script 1: Named Query for Fuzzy OLAP Cube for Multiple Cubes

```
SELECT F.DateSK,
F.EmployeeSK,
F.WorkCodeSK,
F.PlaceofWorkSK,
F.Amount,
F.Mandays,
FC.STATE,
(Fc.Contribution * ISNULL(F.Amount, 0)) AS fAmount,
(Fc.Contribution * ISNULL(F.Mandays, 0)) fMandays
FROM FctDailyWork FDW
INNER JOIN [dbo].[DimPlaceofWork] PW
ON FDW.PlaceofWorkSK = PW.PlaceofWorkSK
INNER JOIN FuzzyContribution FC
ON PW.Trees = Fc.[Value]
WHERE FC.[Function] = 'Trees'
```

Script 2: Named Query for Single Fuzzy OLAP Cube with Single Attribute

```
SELECT DateSK,
       EmployeeSK,
       WorkCodeSK,
       PlaceofWorkSK,
       CAST(ISNULL([Low], 0) AS NUMERIC(5, 2)) Trees_Low_Amount,
       CAST(ISNULL([Medium], 0) AS NUMERIC(5, 2)) Trees_Medium_Amount,
       CAST(ISNULL([High], 0) AS NUMERIC(5, 2)) Trees_High_Amount,
       CAST(ISNULL([Low], 0) + ISNULL([Medium], 0) + ISNULL([High], 0) AS
NUMERIC(5, 2)) Amount
FROM (
       SELECT F.DateSK,
              F.EmployeeSK,
              F.WorkCodeSK,
              F.PlaceofWorkSK,
              FC.STATE,
              (Fc.Contribution * ISNULL(F.Amount, 0)) fAmount
       FROM FctDailyWork FDW
       INNER JOIN [dbo].[DimPlaceofWork] PW
              ON FDW.PlaceofWorkSK = PW.PlaceofWorkSK
       INNER JOIN FuzzyContribution FC
              ON PW.Trees = Fc.[Value]
       WHERE FC.[Function] = 'Trees'
              AND Amount > 0
       ) p
PIVOT(SUM(fAmount) FOR [State] IN (
                     Low,
                     Medium,
                     High
                     )) AS pvt
```

Script 3: Named Query for Single Fuzzy OLAP Cube with Multiple Attributes

```
--Script 2: Named Query for Single Fuzzy OLAP Cube with Multiple Attributes
SELECT Mandays.DateSk,
       Mandays.EmployeeSK,
       Mandays.WorkCodeSK,
       Mandays.PlaceofWorkSK,
       CAST(Mandays.Trees_Low_Mandays AS NUMERIC(5, 2)) Trees_Low_MandaysAS,
       CAST(Mandays.Trees_Medium_Mandays AS NUMERIC(5, 2))
Trees_Medium_Mandays,
       CAST(Mandays.Trees_High_Mandays AS NUMERIC(5, 2)) Trees_High_Mandays,
       CAST(Mandays.Mandays AS NUMERIC(5, 2)) Mandays,
       CAST(Amount.Trees_Low_Amount AS NUMERIC(5, 2)) Trees_Low_Amount,
       CAST(Amount.Trees_Medium_Amount AS NUMERIC(5, 2)) Trees_Medium_Amount,
       CAST(Amount.Trees_High_Amount AS NUMERIC(5, 2)) Trees_High_Amount,
       CAST(Amount.Amount AS NUMERIC(5, 2)) Amount
FROM (
       SELECT DateSK,
              EmployeeSK,
              WorkCodeSK,
              PlaceofWorkSK,
              ISNULL([Low], 0) Trees_Low_Mandays,
              ISNULL([Medium], 0) Trees_Medium_Mandays,
              ISNULL([High], 0) Trees_High_Mandays,
              ISNULL([Low], 0) + ISNULL([Medium], 0) + ISNULL([High], 0)
Mandays
       FROM (
              SELECT F.DateSK,
                     F.EmployeeSK,
                     F.WorkCodeSK,
                     F.PlaceofWorkSK,
                     FC.STATE,
                     (Fc.Contribution * ISNULL(F.Mandays, 0)) fMandays
              FROM FctDailyWork FDW
              INNER JOIN [dbo].[DimPlaceofWork] PW
                     ON FDW.PlaceofWorkSK = PW.PlaceofWorkSK
              INNER JOIN FuzzyContribution FC
                     ON PW.Trees = Fc.[Value]
              WHERE FC.[Function] = 'Trees'
              ) p
       PIVOT(SUM(fMandays) FOR [State] IN (
                     Low,
                     Medium,
                     High
                     )) AS pvt
       ) Mandays
INNER JOIN (
       SELECT DateSK,
              EmployeeSK,
              WorkCodeSK,
              PlaceofWorkSK,
              ISNULL([Low], 0) Trees_Low_Amount,
              ISNULL([Medium], 0) Trees_Medium_Amount,
              ISNULL([High], 0) Trees_High_Amount,
```

```
		ISNULL([Low], 0) + ISNULL([Medium], 0) + ISNULL([High], 0)
Amount
	FROM (
		SELECT F.DateSK,
			F.EmployeeSK,
			F.WorkCodeSK,
			F.PlaceofWorkSK,
			FC.STATE,
			(Fc.Contribution * ISNULL(F.Amount, 0)) fAmount
		FROM FctDailyWork FDW
		INNER JOIN [dbo].[DimPlaceofWork] PW
			ON FDW.PlaceofWorkSK = PW.PlaceofWorkSK
		INNER JOIN FuzzyContribution FC
			ON PW.Trees = Fc.[Value]
		WHERE FC.[Function] = 'Trees'
		) p
	PIVOT(SUM(fAmount) FOR [State] IN (
				Low,
				Medium,
				High
				)) AS pvt
	) Amount
	ON Mandays.DateSK = Amount.DateSK
		AND Mandays.WorkCodeSK = Amount.WorkCodeSK
		AND Mandays.EmployeeSK = Amount.EmployeeSK
		AND Mandays.PlaceofWorkSK = Amount.PlaceofWorkSK
```

Appendix Z: Design Patterns for Data Warehouse

In the data warehouse, there are mainly two types of Design patterns and school of thoughts which is relevant to this study.

Star Schema and Snow Flake Schema are the main two design patterns in data warehousing. The star schema and the snowflake schema are design strategies to organize data marts or entire data warehouses using traditional enterprise relational databases. Both of these strategies are using dimension tables to explain data accumulated in a fact table. Table Z-15 shows the comparison of start schema and Snow Flake schema with respect to different technical areas.

Table Z-15: Different Types of Fact Tables

Comparison	**Star Schema**	**Snow Flake Schema**
Ease of maintenance / change	Has redundant data and hence less easy to maintain and change the existing model.	No redundancy, so snowflake schemas are easier to maintain and change when required.
Ease of Use	Lower query complexity and easy to understand.	More complex queries and hence less easy to understand
Query Performance	Less number of foreign keys and hence shorter query execution time. Faster access to data.	More foreign keys and hence longer query execution time or the slower execution time.
Type of Data warehouse	Better for data marts with simple relationships. Most of the time to relationships are either one-to-one or one-to-many.	Better for data warehouse core to simplify complex relationships. Most of the time many-to-many relations are maintained.
Joins	Fewer Joins are needed.	A higher number of Joins are needed.
Dimension table	A star schema contains only a single dimension table for each dimension.	A snowflake schema may have more than a one-dimension table for each dimension.
When to use	When the dimension table contains a smaller number of rows, Star schema can be chosen.	When the dimension table is relatively big in size, snow-flaking is better as it reduces space.
Normalization/ De-Normalization	Both Dimension and Fact Tables are in De-Normalized form	Dimension Tables are in Normalized form however, Fact Table is in De-Normalized form
Data model	A top-down approach is followed.	A bottom-up approach is followed.

Figure Z-29 shows sample data warehouse design which has followed Start schema.

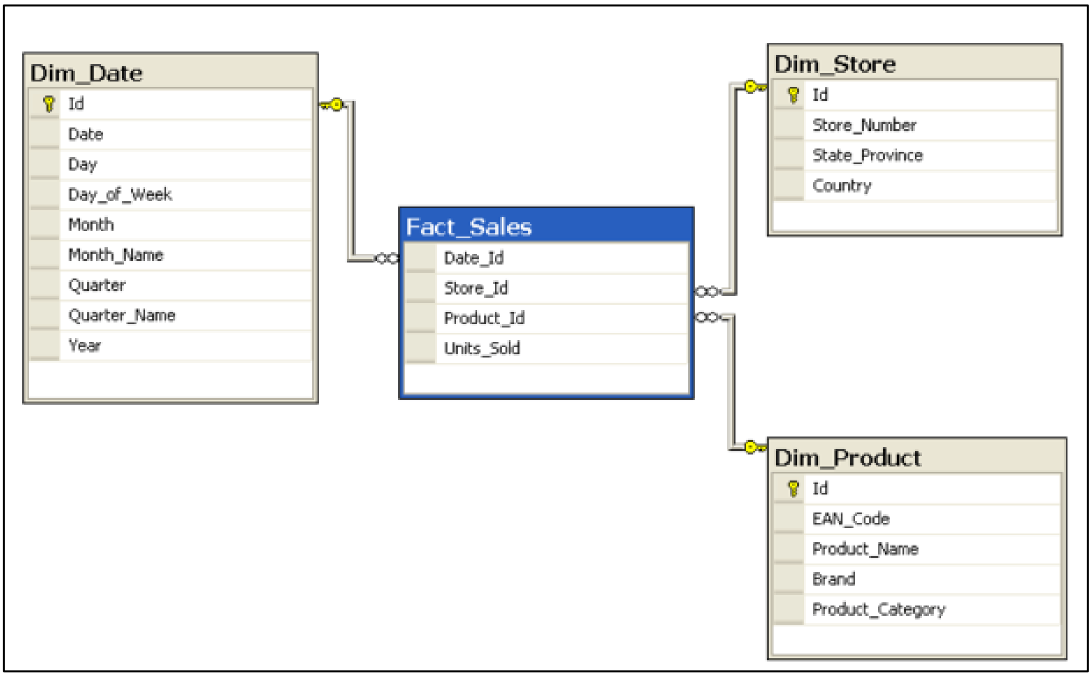


Figure Z-28: Example for Star Schema

In the above schema design, the fact table, Fact_Sales would be a record of sales transactions, while there are dimension tables for a Dim_Date, Dim_Store, and Dim_Product. Dimension tables are each associated with the fact table through their surrogate key which is the primary key. Those surrogate keys are foreign keys for the fact table. For example, without storing the actual transaction date in a row of the fact table, the date_id is stored. This date_id matches to a unique row in the Dim_Date dimension table. Date Dimension is discussed in detail in Appendix C:. That date record also stores other attributes of the date such as weekday, month name, year etc that are required for better analysis in reports. The date dimension is de-normalized for easier reporting and faster access to data.

Figure Z-30 shows the Snowflake schema design of a sample data warehouse.

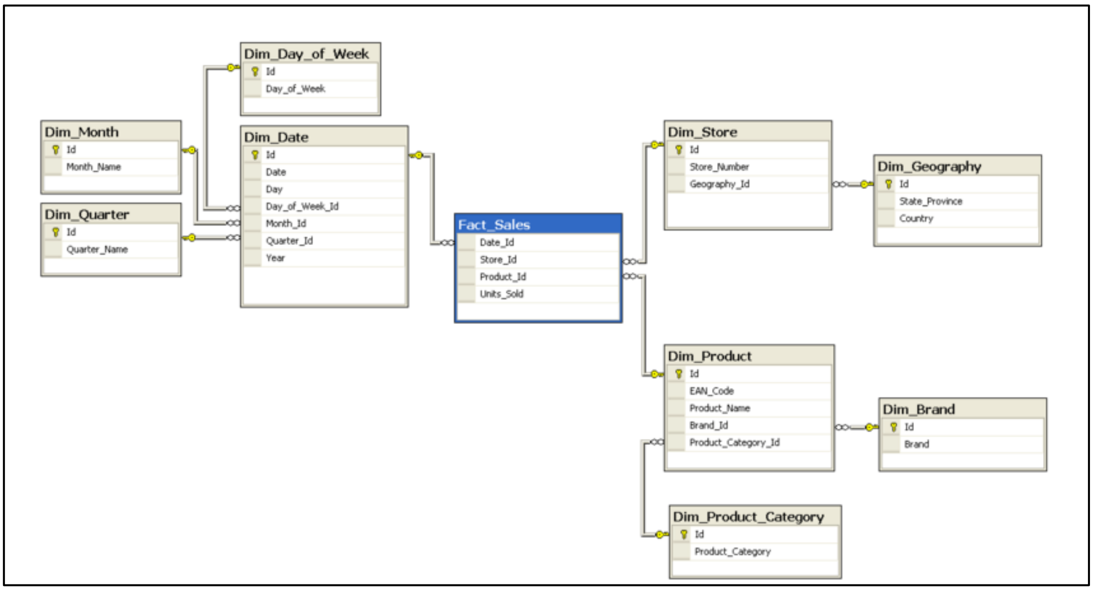


Figure Z-29: Example for Snowflex Schema

Snowflex schema is more normalized than start schema. For example, without storing a month, quarter and day of the week in each row of the Dim_Date table, these are further broken out into multiple dimension tables. Likewise, for the Dim_Store table, the state and country are geographical attributes that are one step removed, without storing in the Dim_Store dimension table, they are now kept in a different Dim_Geography dimension table.

Appendix AA: Publications

1. PPG Dinesh Asanka, Amal Shehan Perera, Feasibility of Fuzzy Data Warehouse, International Journal of Research in Computer Applications and Robotics, ISSN 2320-7345, Vol. 5 Issue 11, November 2017.

2. PPG Dinesh Asanka, Amal Shehan Perera, Defining Fuzzy Membership Function Using Box Plot, International Journal of Research in Computer Applications and Robotics, ISSN 2320-7345, Vol. 5 Issue 9, September 2017.

3. PPG Dinesh Asanka, Amal Shehan Perera, Design Strategy for Fuzzy Data Warehouses, 2nd International Conference on Innovative Research in Science, Technology & Management, National University Singapore, 29-30 September 2018.

4. PPG Dinesh Asanka, Amal Shehan Perera, Defining Fuzzy Membership Function for Fuzzy Data Warehouses, 4th I2CT IEEE Conference, SDMIT Ujire, Mangalore, India, October 2018.

5. PPG Dinesh Asanka, Amal Shehan Perera, Linguistic Analytics in Data Warehouses Using Fuzzy Techniques, IEEE International Research Conference on Smart Computing and Systems Engineering – 20019, Department of Industrial Management, University of Kelaniya, 28th Match 2019.

6. PPG Dinesh Asanka, Handling Uncertainty in Data Warehousing, DevDay Conference, 2017-11-09.

7. PPG Dinesh Asanka, Linguistic Analytics on Data Warehousing, Colombo Big Data Meet up, 2017-11-15.

Appendix BB: Citations

1. Lewis Veryard, Hani Hagras, Andrew Starkey, Gilbert Owusu, Gilbert Owusu, A Fuzzy Genetic System for Resilient Routing in Uncertain & Dynamic Telecommunication Networks, June 2019, DOI: 10.1109/FUZZ-IEEE.2019.8858788, Conference: 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE).

2. Raoul Nuijten, Uzay Kaymak, Pieter Van Gorp, Monique Simons, Pascale Le Blanc Fuzzy modelling to 'understand' personal preferences of mHealth users: a case study, 11th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT 2019), Jan 2019.

3. Saranya S,Vigneshwaran M, Neutrosophic, .NET Framework to deal with Neutrosophic $\boldsymbol{b*g\alpha}$ -Closed Sets in Neutrosophic Topological Spaces, Sets and Systems, Vol. 29, 2019, University of New Mexico

4. Broumi, S.; D. Nagarajan; A. Bakali; and Mohamed Talea. "Implementation of Neutrosophic Function Memberships Using MATLAB Program." Neutrosophic Sets and Systems 27, 1 (2019).

5. Milton-Thompson, O., “Developing a risk assessment model using fuzzy logic to assess groundwater contamination from hydraulic fracturing”, University of Exeter, 2020.

6. Yuri Sa, “Automatização Total do Diagnóstico de Malária - Fase 1: Teste de Viabilidade - Modelo Classificatório Fuzzy”, Conference: ANAIS DO 14º SIMPóSIO BRASILEIRO DE AUTOMAçãO INTELIGENTE, 2019.

7. Yuri Sa, “Modelo classificatório Fuzzy para detecção de nuvens em imagens de satélite utilizando análise de textura como entrada”, UNESP Campus Sorocaba - Mestrando - Ciências Ambientais, 2019, September.

8. Ikram MA, “AI-driven Service Broker for Simple and Composite Cloud SaaS Selection”, Faculty of Engineering and Information Technology, University of Technology Sydney, 2020.