DATAWAREHOUSE APPROACH TO DECISION SUPPORT SYSTEM FROM DISTRIBUTED, HETEROGENEOUS SOURCES

A Thesis

Presented to

The Graduate Faculty of The University of Akron

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

Vijaya L Sannellappanavar

August, 2006

DATAWAREHOUSE APPROACH TO DECISION SUPPORT SYSTEM FROM DISTRIBUTED, HETEROGENEOUS SOURCES

Vijaya L Sannellappanavar

Thesis

Approved:	Accepted:
_____	_____
Advisor Dr. Chien-Chung Chan	Dean of the College Dr. Ronald F. Levant
_____	_____
Committee Member Dr. Xuan-Hien Dang	Dean of the Graduate School Dr. George R. Newkome
_____	_____
Committee Member Dr. Zhong-Hui Duan	Date
_____	
Department Chair Dr. Wolfgang Pelz	

ABSTRACT

In today's world of global business, worldwide partnerships and corporate mergers, decision making plays a major role in the steady growth of a business providing it a competitive edge. Decision making is the key to smooth day-to-day operations as well as for effective future planning in this ever competitive world. Several sources of data exist in the business from which valuable information can be extracted to help make a wide range of decisions. In order to facilitate querying and analysis, the data from these sources need to be integrated. There are various considerations and approaches for such Data Integration or Information Integration and several issues surround this process. These issues are considered and the prominent approaches to Information Integration are studied with an emphasis on the Datawarehousing approach. A Datawarehouse is implemented from scratch from available raw data sources and by means of experimentation, it is shown how Datawarehousing is the most suited of all the considered approaches in specific business settings. The main objective of the research and the contribution of this thesis are towards analyzing the major issues faced in specific enterprise scenario and demonstrating how the Datawarehousing approach provides an efficient solution for them and hence provides a solid foundation for Decision Support Systems from distributed, heterogeneous data sources.

DEDICATION

This Thesis is dedicated to two of the most important and dearest people in my life - my loving, ever supportive and protective brother Girish L Sannellappanavar who has been the main reason for my higher studies – no words can duly describe his priceless contribution to my life and career and

my most affectionate grandpa Mr. S. B. Patil, who has enriched my life with his caring persona and instilled in me timeless human values.

They are my two mighty pillars of strength and support.

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor, Dr. C. C. Chan, for his constant guidance and patience throughout my research and thesis, the whole experience of which has enhanced my aptitude and ardor for research and development. I would also like to thank the committee members for my Defense, Dr. Z.H. Duan and Dr. X.H. Dang for their time and valuable thoughts regarding my work. Furthermore, I take this opportunity to thank all the faculty members of the Computer Science department of The University of Akron for the numerous interactions that I have had with them during the course of my M.S. studies and for the rich knowledge, encouragement, appreciation and disciplined guidance provided throughout. Thanks is also due to The Office of International Programs and everyone in The University of Akron who ensure that the life of an international student so far away from home is as smooth and as fruitful as possible during the entire tenure of their study and stay at the University. Without all of their good will, I cannot imagine how much harder things could have been before I could reach this rewarding culmination of my graduate studies.

This phase of my life would not at all have been possible had it not been for the incessant love, care, nurturing, understanding, support and encouragement showered by all my family members. The amount of inspiration and strength that I draw from the

manner by which each of them has faced the arduous challenges posed by life supplemented by the unflinching belief they have always shown in my abilities to excel no matter what task I undertake, have sustained me during tough times. Special thanks to my parents – two venerable professors, grandparents – outstanding caretakers, sister Girija, brother Girish, brother-in-law Nagaraj, sister-in-law Aparna – all brilliant engineers and adorable nephew Gagan – the apple of our eyes - for all that they have given me and for ensuring that my every moment with them is joyous and contented. I am ever obliged to them for being there for me and standing by me at all times.

Completing this Thesis depicts an important milestone in my career life and I owe it to many friends and well-wishers both in the USA and India for their accommodating thoughts and deeds. I am grateful to my friend Kartik Sundaram, for taking the time and patience for proof-reading my thesis, providing me worthy feedback and inputs and moreover, for his supportive presence and for enthusiastically volunteering help on several occasions which made things so much simpler for me. I would also like to express gratitude to my professors at my undergraduate college, S.D.M. College of Engineering and Technology, Dharwar, India for the quality education and solid foundation of Computer Science and Engineering that they bestowed upon me. I wish to gratefully acknowledge all my well-wishers at my current company, for their presence through thick and thin times. My heartfelt appreciations to all others whom I could not mention here but are present in my thoughts at the moment.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION

1.1 Introduction

In today's world of global business, worldwide partnerships and corporate mergers, decision making plays a major role in the steady growth of a business providing it a competitive edge. Decision making is the key to smooth day-to-day operations as well as for effective future planning in this ever competitive world. Several sources of data exist in the business from which valuable information can be extracted to help make a wide range of decisions. In order to facilitate querying and analysis, the data from these sources need to be integrated.

1.2 Research Objectives

There are various considerations and approaches [1,3,4,6,7,8] for such Data Integration or Information Integration and several issues surround this process. These issues are considered and the prominent approaches to Information Integration are studied with an emphasis on the Datawarehousing approach with respect to the situation on hand. A Datawarehouse is implemented from scratch from available raw data sources

and by means of experimentation, it is shown how Datawarehousing is the most suited of all the considered approaches in specific business settings. The main objective of the research and the contribution of this thesis are towards analyzing the major issues faced in specific enterprise scenario and demonstrating how the Datawarehousing approach provides an efficient solution for them and hence provides a solid foundation for Decision Support Systems from distributed, heterogeneous data sources.

1.3 Structure of the Thesis

The Thesis begins with an Introduction to the domain of the Research in Chapter I. Chapter II describes Information Integration, the focal concept of the Research – what it means, the different approaches for implementing it and the primary benefits that the Datawarehousing approach provides. Chapter III describes the business scenario considered, the problems faced and a first look at the possible solution. Chapter IV deals with the actual implementation of the Datawarehouse – the design considerations, the model of the Datawarehouse, how it encompasses all the chief issues faced in the situation explored in the research and an introduction to the tool used for the implementation. Chapter V describes the efficacy of the Datawarehouse solution. The Thesis concludes with Chapter VI which provides a summary of the Research and suggestions for future work.

CHAPTER II

INFORMATION INTEGRATION

2.1 Introduction

Information Integration is the focal concept of the Research. In this chapter, we examine it more closely – what it involves, the different approaches to it and the main advantages of the Datawarehousing approach.

2.1.1 What is Information Integration?

Information Integration is a vital Database application in today's Information Technology-based World. This application is used in a wide array of fields ranging from critical scientific arenas like Genome Research to fundamental Business scenarios like Retail Operations, where the key to decision-making and deriving conclusions lies in extracting data from distributed, heterogeneous data sources, combining the data from the multiple data sources and representing information in a helpful way for User querying and analysis. These steps are the basis of Information Integration.

2.1.2 Distributed, Heterogeneous Sources

Typically, the data sources considered for Information Integration are distributed around the World. So, it is imperative that we acknowledge the fact that not only can the

systems containing the data sources and the system used for analysis of the data be discrete but also there is every likelihood of them being located in distant places without being connected by just one common network. Secondly, the data sources may all be in diverse formats depending on the underlying technology used in that location.

For instance, the corporate office of a company may be located in Cleveland whereas the various branches or operational units may be located throughout USA in different geographical areas or even other countries. So, data needs to be sent from each of these locations to the corporate office in Cleveland. Moreover, depending on various factors like location, size of the branch, budgetary constraints etc, some of the branches may be employing different or more advanced technology than some others to suit their specific needs. In turn, this affects the way similar tasks are carried out in the different locations. Hence, the data sources may be diverse or 'Heterogeneous' in various ways.

2.2 Issues involved in Information Integration

As touched upon in the previous section, there are certain considerations [1,2,3,4] involved in Information Integration for distributed, heterogeneous data sources. They are further discussed below:

1. Dispersed data sources

The sources may be scattered around the World. A global business may be having few branches in Asia, few in Europe and few others in USA. So, we need to think in terms of their availability and accessibility, in terms of factors like time difference, example, China is ahead of USA by 12 hours. What would be the best time to run multiple queries against the data sources such that they do not consume system resources

during normal operation hours and affect regular business in any way? What would be the best way of accessing these data sources to extract data or gather information?

2. Disconnected data sources

The sources may not be connected to each other or to the system used for analysis. Some branches may be well-connected to the Internet while others may either not be having network facilities, or may be having erratic connectivity, due to issues like low bandwidth, inferior technology used etc. Obtaining data from these sources may not be as simple as logging on to a system and connecting to the data source with a few clicks. We need to consider how best can we get data from these sources on a regular basis.

3. Disparate technologies for data sources

There is a high probability that all the data sources considered for integration may not be using the same underlying technology for the databases. For example: in one organization, some branches may be maintaining data in spreadsheets, some others may be using automated systems based on elementary flat files, still others may be using robust database server technologies like SQL Server or Oracle Server. What is the format in which we can retrieve data from these systems? And how can we analyze such mixed formats of data as one unit?

4. Non-conformance in database schemas

The design schemas for the databases may vary. The local schema used for a database may depend on a number of factors like the supporting technology, the normalization level used for the tables, the attributes of data collected etc. To quote a

simple example, a product, say apparel has several characteristics associated with it namely size, fabric, color, vendor, cost etc. This data may all be stored either in one table or divided into multiple tables as shown here:

Schema 1: Apparel(Tag No, Size, Fabric, Color, Vendor, Cost)

Schema 2: ApparelDetails(Tag No Size, Fabric, Color,Vendor)

ApparelCost(Tag No, Cost)

This affects the way we extract data from each data source. Thus, it is important to understand the design schema of each of the data source involved before considering the integration of data from them.

5. Heterogeneity in the values stored

Even though all the data sources contain data related to the same domain, the kind of values stored in them individually may vary perhaps depending on the initial design decisions made. For instance: consider a plastic industry in which the distributed data sources belong to its different production plants located in different places. All the data sources may contain data related to pipe manufacturing, however the individual data sources may differ in the kind of data stored i.e., one data source may contain data related to all types of pipes (pressure pipes, gas pipes, water pipes etc) whereas another data source may contain data related to only pressure pipes. In other words, the data sources may differ in the kind of data stored if the sources contain different subsets of the data domain.

6. Missing data values

Another case of heterogeneity in the values stored is if the attributes stored are different in the data sources even though the data is related to the same entity. In the case of the plastics industry mentioned earlier, if the data collected in one data source contains some attributes which are not collected in another data source, then the kind of values stored differ as shown below:

Data in Source 1: PressurePipes(dimensions, tensile strength, elongation, stiffness)

Data in Source 2: PressurePipes(dimensions, elongation, stiffness)

In this case, the data from source 2 does not provide us with information about the tensile strength of the pipes manufactured in that plant. Hence, it becomes an issue if we want to integrate the data from these sources. Typically, the databases may contain NULL or default values implying that there is no specific data value recorded for that field of the record.

7. Conceptual difference

The very concept based on which the data is collected may differ in different sources. To better illustrate this, let's take the example of gift shops in amusement parks. If we need to analyze the inventory and sales data from a group of such gift shops, there may be an essential difference in the understanding of what all items constitute the basic unit for record-keeping. An item, say, a rental stroller, is considered an "item" for all sales purposes whereas, it is not considered an "item" for inventory book-keeping where it does not fulfill the definition of an entity whose stock reduces with every new sale made. Hence, we find the stroller as an item sold but not an item on the inventory list.

8. Non-conformance in data types

Even if two or more data sources have the same base technology and they even agree on their schema, there may be a fundamental difference in the way data itself is represented. For example: the same products shipped out to different locations may be represented differently in the locations' individual databases i.e., in one database the products may be identified by Sku numbers represented using integer data type whereas in another database, the products are identified by the manufacturers product names which are stored using string data type. Another example is, when different schools use different systems for assigning grades. In this case, the databases may contain 'grades' in percentage, plain numbers, or letter grades. In certain cases, such variance in the data types used for storing the same data may cause problems in conversion or lead to erroneous results when the data from these databases are used as a unit. An example of further subtle difference is when the same data type is used, but the field lengths vary from one database to another thus, the ranges of values stored in them differ.

9. Variance in Time

If the data stored for the data sources differ in the period of time over which the data was collected, then there is a variance in time. We need to understand for each of the source, how long the data has been collected for? What is the earliest period of time for which I can retrieve data from the database? Is there a common ground to compare data from one database to another in terms of the time period? For example: suppose the Information Integration needs to be done for a business owning a chain of restaurants in different places, where each restaurant in the chain maintains its own data source. If the first restaurant was established in 2000 and a recent one was opened in 2005, then when

we attempt to integrate data from multiple data sources, we will not be able to pull any historical data from the latter source for any time period prior to 2005. So, querying and analysis on the data sources as a whole needs to take care of variance in time over which the data is available in the individual sources.

10. High volume of data

In most enterprises, be it the stock market or the travel industry or simply shopping malls, the transactional data stored in the sources increases multifold by the second and hence the volume of data available for integration can become overwhelming. Not only will this overburden the systems containing the data sources thus leading to slower processing, but also a large portion of it is unnecessary from the point of view of integration. For the purpose of integration, we need to know, for how long has the data been accumulated in the data sources? What subset of data do we need for our purpose? Can we simply filter data or extract data only for the time frame that we are looking at working with?

11. Difference in interpretations of data

Even though the data in the many data sources under examination contain data associated with the same domain, there may be difference in the semantics in various ways. One example is- supposing we are dealing with data associated with sales at several grocery stores and the primary information that we seek is the 'Total Sales' for every transaction. The difference in the semantics may be in the way the Total Sales value is interpreted in 2 different stores:

Store 1: Value in Total Sales = quantity x price + tax – discount

Store 2: Value in Total Sales = quantity x price – discount

In this case, the field with the same name in 2 different data bases contains values having different base calculations or applied formula or derivation. So, if we want to calculate the total sales without tax or the total tax collected for the 2 sources, we need to apply different calculations on the field in the 2 databases. Another example would be of a difference in measurements or units say, one ‘unit’ of some product may be containing 30 pieces in one location whereas the same product’s unit comprises of 50 pieces in another location.

12. Discrepancy in comprehensiveness of data

The data gathered in different data sources may be employing different levels of aggregation. To quote an example, a retail grocery business may be having wholesale outlets which cater to the needs of small businesses as well retail stores catering to the needs of individuals. The database in a wholesale outlet may vary from the one in a retail store in the level of aggregation of inventory i.e., the former may store inventory data at the product category-level whereas the latter may contain inventory data at the product category-level, at vendor-level and specific product-level. So, if we want to integrate the data from these databases, then we may not be able to get information like what is the cost of inventory for a particular vendor at a given time from the wholesale outlet’s database, which can only give information with a perspective of a product category. However, we can get information about the inventory cost of a category of products, products of a specific vendor and further inventory level of individual products too from a retail store database. So the concerning questions are – what is the level of abstraction of the data stored in the individual databases and does it hold values to retrieve information to the depth that we seek? The databases which contain data at the most

detailed levels are the ones that we can get most kind of information that we look for, from integration.

13. Corrupted or erroneous data

The data in the data sources may be corrupted and may contain erroneous data. Data corruption may occur due to some hardware issues or network problems causing incomplete processes or erroneous results. Moreover, generally if any record-keeping is done by manual data entry or data is generated as a result of some processes followed by operators, then there are bound to be human errors – calculations may be wrong, processes followed may be inaccurate, data may be entered incorrectly, there may be rounding errors – in any case, the data source will contain inaccurate or incorrect values. Even if there is no human intervention, there is a possibility of the software used being inefficient, thus resulting in wrong data.

14. Data redundancy and inconsistency

Data redundancy or data duplication may occur in the data sources due to a variety of reasons like human errors, software inefficiency, multiple updates, repeated synchronization etc. In addition, there may be inconsistency between the various entries for say, the same entity. For example, when hundreds of product numbers need to be entered manually, it is most likely that the product number entered in one field would be incorrect and hence different from another field for the same product.

2.3 Approaches to Information Integration

Three prominent approaches for Information Integration have been described – Federated approach, Mediated approach and Datawarehousing approach. Each of these approaches offers benefits over the others, in different cases. The three approaches are described in the next few sections.

2.3.1 Federated Approach

This approach [1,3,6] can be considered most basic of all. Here, whenever some information is needed, the data source from which it may be obtained is queried. This System is capable of extracting data from distributed, heterogeneous databases in response to User query. This extracted data is presented to the User.

The method followed is to select the appropriate query from a query library in the database for the specific data source to be queried and send it to that data source, which in turn, executes the query and sends back the results. So, every data source would have to hold queries for every other data source which may be queried. Thus, N data sources would be containing N x (N-1) queries, in the Federated approach.

This approach, though seems superfluous, may be the best approach in some cases. For example, when it is necessary to query only a few databases whose data keeps changing often – say, a bookstore manager wants to check the availability of a specific book only in their other local branches. In this case, the queries stored would be reduced to a few as compared to having to store queries to be used with the databases of all the national branches.

2.3.2 Mediated Approach

This approach [1,3,4,7,8] of Information Integration provides a virtual database to the User. The User is provided a global view of data to be queried and every User query is then transformed into one or more queries conforming to the local schema of the data sources to which they are sent. The data sources execute the queries and send back the results which are transformed to User/ Global schema before being combined to be presented to the User. The transformations of queries between global and local schemas occur in a ‘Wrapper’, which exists for every data source and the results sent by all Wrappers are combined in the ‘Mediator’. The Mediator also contains logic needed occasionally to decide whether to send the query to more than one data source or not, based on the requirement of the query. For instance, if the essence of the query is to check whether an item is available at all, then, if it is available in the first data source queried, there is no necessity of querying the remaining data sources.

2.3.3 Datawarehousing Approach

In this approach [1, 2, 3, 9, 10], as the name suggests, a ‘warehouse’ or a reservoir of data is built from data extracted from various distributed, heterogeneous databases [Figure 1.1]. So, in order to access ‘Integrated Information’, the User has to simply query the Datawarehouse similar to querying a database.

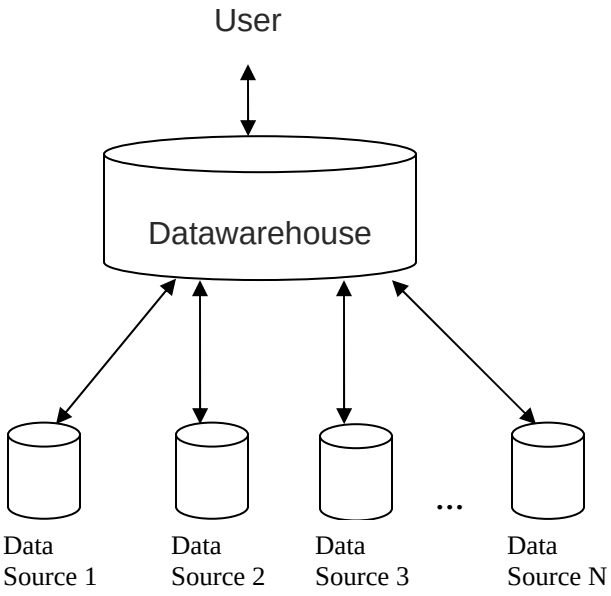

Figure 2.1 Datawarehousing approach to Information Integration

The distributed, heterogeneous data that is used to build a Datawarehouse, is first cleaned and processed to remove inconsistencies and mismatches before being combined in accordance with the global schema of the Datawarehouse. However, since the Datawarehouse is built from data sources, the Datawarehouse needs to be regularly updated so that it contains the latest data available in the respective data sources. This updating is necessary to avoid situations where stale data is being used by the User for analysis. The updating can be done either by rebuilding the Datawarehouse regularly, maybe every night or simply incrementally updating the Datawarehouse with only the new data from the data sources or by a third method of updating the Datawarehouse as and when the data is updated in any of the data sources. Though the

process of updating ensures the availability of latest data, however it also makes the Datawarehouse unavailable throughout the update. The first method of updating, may at times, become time-consuming. So, incremental updating may seem a better choice, but this method needs additional logic on the data source to obtain and send only the new data so also the third method of updating, in which, in addition to the logic of the second method, it needs to determine when changes occur in any of the data sources. With the advent of advanced technologies, other superior methods of updating are available now, which involve almost nil time of unavailability of the Datawarehouse which means instant updates and almost real-time data in the Datawarehouse.

Because of the local nature of data access of distributed data, the Datawarehousing approach offers the benefit of no overhead of communication with data sources to get the results of User queries. Moreover, in a Datawarehousing approach, the data may be organized in ways which could be more beneficial for User analysis, for instance using Data cubes.

2.3.4 Advantages of Datawarehousing

There are several factors why an organization would consider the Datawarehousing approach to provide the base for their decision support systems. The prominent ones are discussed below:

1. Simplicity of having one unified data source

One of the primary advantages of a Datawarehouse is having the data available locally to the decision makers. When the Datawarehouse is built after retrieving the data from multiple data sources of interest to the User, the User can simply query it as one data source and not have to bother querying multiple sources to get the desired

information. The entire complexity of having to deal with several data sources is thus reduced to the simplicity of using one data source.

2. Solution to issues of Heterogeneity

Creating a Datawarehouse and thus having one unified data source provides an outstanding solution to the major issues faced in Information Integration. How one approach can fend for several prominent issues, is so significant, that its contribution to the resolution of each of these issues is worth discussing individually:

i. Conformance in database schema

When a Datawarehouse is designed, several factors are taken into consideration like the business processes of the users' interest, the key applications of the Datawarehouse, the data sources of the users' interest, etc. This facilitates the design of a global schema that accommodates all the needs of the users while taking care of the non-conformances in the schemas of the underlying data sources, such that the source schemas can be appropriately mapped to the Datawarehouse schema. Hence, even though the data are actually stored as per the individual database schemas, when they are extracted from the sources and loaded into the Datawarehouse, they all adopt one schema, thus resolving the issue of non-conformance of the data sources.

ii. Uniformity in the subject

As mentioned in the previous point, several factors are taken into consideration while designing a Datawarehouse. Hence, a Datawarehouse is built around a subject(s) of the users' interest. For example: if a hospital management wants to study their patients in terms of periodical count, the kind of ailments, the

initial diagnosis, the treatment received, the duration of hospital stay etc, then the Datawarehouse will be built by extracting only the data concerning these areas and ignore rest of the data which is unnecessary from the many sources. Thus, the cumbersome handling of the bulk of data on different subjects no longer prevails.

iii. Conceptual consistency

When multiple data sources are integrated into one Datawarehouse, their individual conceptual differences are eliminated. The Datawarehouse follows one set of terminologies, attributes etc according to the users concepts, irrespective of which data source the data originated from. This makes it easy for the users to comprehend the information obtained as a result of query and analysis. Otherwise, often the different practices followed by the data sources become misleading for the users and consume valuable time and resources to derive proper meaning or in the worst case, may lead to wrong conclusions.

iv. Consistency in Datatypes

When similar information is to be retrieved from several data sources, though the entities involved will be the same, there may be differences in the datatypes used for representing them in the individual data sources. These inconsistencies are dealt with while transporting the data extracted from the data sources to the Datawarehouse. Any such inconsistencies if exist, are appropriately converted to match that of the destination. Hence, the data used by the end user does not pose any problem in terms of erroneous results due to incompatible or inconsistent datatypes., when data from different sources are used together.

v. Better quality of data used

Data is cleaned of wrong entries, duplicates, missing values etc before loading into the Datawarehouse. Thus, the end user deals with better quality of data hence producing more accurate information.

3. Elimination of communication overhead and network issues

The added advantage of a Datawarehouse is the minimization of communication overhead. Since the data is available locally, when the User sends a query, the results are retrieved from the Datawarehouse, thus avoiding the necessity of transmitting the query to multiple data sources and getting back the result. This also means that the User has no hassles of network issues anymore when all he wants to do is to get some information for analysis.

4. Ease of usage

When data is stored in a Datawarehouse, a vast number of methods can be employed and applications can be developed, which provide the User with an abundance of information and knowledge presented in a logical, well-defined form for easy querying and analyzing. For instance, if a data cube is created on top of the Datawarehouse, then the data from multiple. Distributed, heterogeneous sources are integrated and organized in an appropriate format such that the User can instantly access and examine the data from any of the sources individually or in combination with any other source and also if needed, all the sources together in comparison. In addition, the User can choose the level of abstraction he needs for the analysis and obtain information to ad-hoc queries at one go and all the results can be presented in a desired format like tables, charts, graphs, reports etc as per the user’s preference and convenience.

5. Remarkable applications of Datawarehouse

With the perspective of supporting the ever so important process of decision-making, several applications and methodologies have been developed for churning out vital information from the data available in the datawarehouses for practically every major enterprise – business corporations, research organizations etc. The usage of a Datawarehouse is three fold [2]. Firstly, the Datawarehouse can be used to get information by generating tables, reports, charts etc. Secondly, the Datawarehouse can be used for data analysis using OLAP, where multiple data operations can be performed on the data like drill down, roll up, slice, dice, pivot etc. Thirdly, the Datawarehouse can be used for Datamining applications, which facilitate Knowledge discovery from pattern evaluation, associations, classifications, predictions, trend analysis, statistical analysis etc.

CHAPTER III

CASE STUDY

3.1 Business Scenario

A typical Retail business can be essentially considered as a chain of retail outlets or stores all linked to the Corporate Headquarters. The retail outlets may all be situated in the same geographic location as the corporate office or may be distributed all across the World. In either case, the business executives or the decision makers at the Headquarters need to keep monitoring the status of each of the branch on a periodic basis in order to evaluate profits and loss and decide what kind of budget and resource allocation and future planning needs to be done for the positive development of the overall business. To facilitate this decision making process, the Retail stores have the necessity of reporting to the Corporate office periodically –may be weekly monthly, quarterly, yearly or for any other time frame- on particular dynamic information like Sales, Inventory etc. This scenario is illustrated in the following figure.

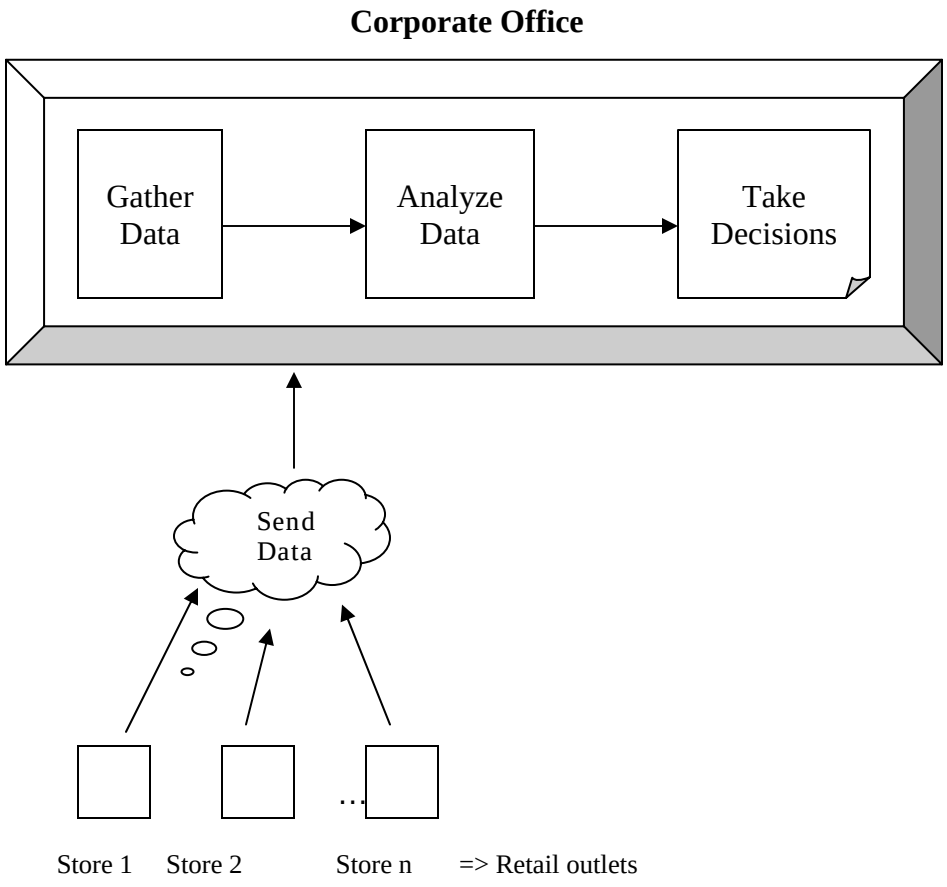

Figure 3.1 Case Study – business scenario

3.2 Issues encountered

There are various issues involved in different phases of this process. The prominent ones are listed below. They are categorized as issues faced on the corporate end and the issues faced on the individual locations or branches end:

1. Issues on Corporate-end

- Defining parameters needed from the Branches for analysis.
- Gathering multiple reports from numerous branches on a timely fashion.
- Ensuring that all necessary data is received.

- Cumbersome process of going through the large data/ extracting only necessary data for analysis.
- If something more is needed, notifying branches again & waiting for the data.
- Timely analysis of the data/ reports in order to take informed intelligent decisions.

2. Issues on Individual Locations-end

- Regular task of creating data files or reports to be sent to Corporate.
- Meeting the deadline to send in reports to Corporate, irrespective of how hectic the busy Sales season is going!

3.3 Overview of Solution

Most of the major issues discussed in the previous section can be solved by providing a means of collecting and integrating necessary data from multiple diverse sources in a way that the Corporate can directly access and analyze the data as and when required. Such a system alleviates the issues both in the corporate and the respective branches. It eliminates the necessity of the Retail locations' task of generating a complying report without fail on a periodic basis and sending it to the corporate office and from the point of view of the corporate headquarters, this concept handles the issues of the data being available on a timely fashion and in a complete form. Moreover, since the Corporate can access the data at any time, the Corporate has an added benefit of being able to analyze data and take decisions at any time. This clearly shows how beneficial this kind of a Decision support system would be in terms of time, efforts and resources. The Datawarehouse approach to Information Integration provides the ground for such a Decision support system.

Figure 1 is modified to show specifics for Datawarehousing as in the following figure.

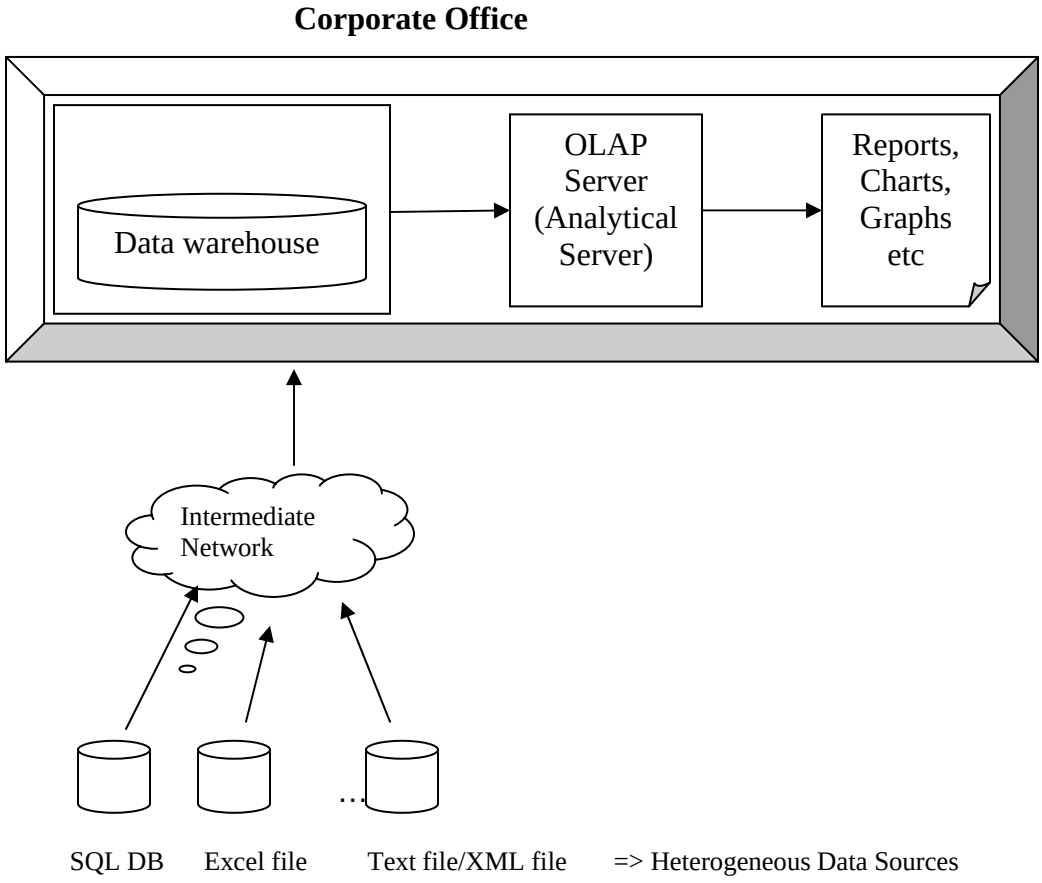

Figure 3.2 Datawarehouse solution for business scenario in case study

CHAPTER IV

DATAWAREHOUSE IMPLEMENTATION

4.1 Tool Used

The Datawarehouse is implemented using Microsoft ® SQL Server™ 2005 and the accompanying SQL Server Integration Services (SSIS). SSIS is the Extraction, Transformation and Load Platform provided by SQL Server 2005. It replaces the Data Transformation Services (DTS) tool provided by earlier versions of SQL Server. SSIS is integrated in the new 'Business Intelligence Development Environment' that accompanies SQL Server 2005. This enables a developer to create SSIS Packages in a graphical development environment. These packages can be used to extract the data from different kinds of data Sources, transform the data using multiple transformations and finally load the data into the destination in one consistent format, in this case, the Datawarehouse.

SSIS provides several advantages, the main ones are:

1. Speeds and eases up the step of Data Cleaning and Preprocessing, one of the preliminary steps in Knowledge Discovery in Databases (KDD), which is considered likely to take 60% of the efforts [2].
2. Capable of handling data from heterogeneous data sources.

3. The conventional Extract, Load, Transform is replaced by Extract, Transform, Load by avoiding staging for data processing before being loaded into destination. This improves efficiency and speed [5].
4. Several kinds of transformations are available that can be applied to the data from the various data sources.
5. Graphical development environment for reducing development efforts as shown in figure below:

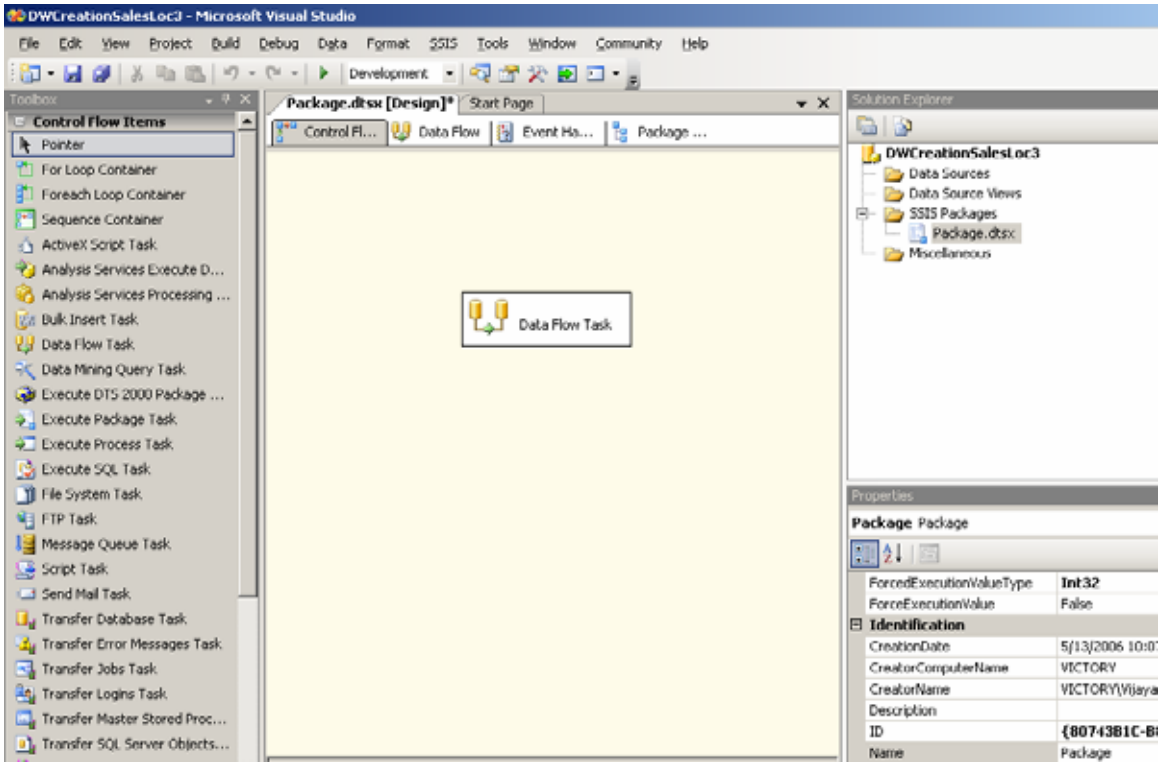

Figure 4.1 Development environment of SQL Server Integration Services

4.2 Issues Considered

Some of the main issues that arise and need to be considered before loading the data into the Datawarehouse to be used for analysis are:

1. Possibility of various heterogeneous data sources – it may be necessary to extract data from one or more heterogeneous data sources, for example –
    a. Text files (Comma delimited, Tab delimited etc)
    b. XML files
    c. SQL Server DB
    d. Access DB
    e. Excel files (other Spreadsheets)
    f. Data from web services(non-traditional data source)
    g. RSS Feeds (non-traditional data source)
    h. Other OLEDB providers
    i. Microsoft .NET Data Provider
    j. Structured/non-structured/semi-structured data
    k. HTML documents
2. Prior to loading data from these multiple heterogeneous data sources to the Datawarehouse, we need to perform Pre-processing on the data for example: data reduction or filtering to select only required data and ignore the unwanted subset.
3. Data Cleaning may be required to prevent bad data from being loaded into the Datawarehouse. For example: if null values are present for any required fields.
4. Data Transformation may be required to convert the data from the sources into

appropriate type or form suitable to be loaded into the Datawarehouse. For example, money in one currency needs to be converted into the currency required by user for analysis.

4.3 Data Sources used

Three data sources were used, each in a different format for building the Datawarehouse. In order to relate to the Retail Scenario described earlier, these 3 data sources can be considered as the data sent to the Corporate Office from 3 different locations. The data from these 3 sources needs to be preprocessed and loaded into a Datawarehouse for analysis and reporting. Details are provided below:

1. Data Source from Location1:

The Data Source from Location1 is in the form of Comma Delimited Text Files, perhaps automatically generated by some Point of Sales software.

2. Data Source from Location2

The Data Source from Location2 is in the form of Excel spreadsheets, perhaps partially entered manually.

3. Data Source from Location3

The Data Source from Location3 is SQL Server Database, which is the transactional database, where sales are recorded whenever performed and it also contains Inventory data and other relevant data.

4.4 Steps Followed

4.4.1 Data Extraction

Location1 and Location2 send the following files to Corporate –

1. Product file – list of Products at their location.
2. Vendor file – list of Vendors that they buy products from.
3. Sales file – list of sales transactions that occurred for a given time period.

Relevant data are extracted from these files.

Location 3 has all the information stored in respective tables in its SQL Server Database. Considering this database is available for access we extract data from the tables directly.

Schema and specifications for data at each Location is as follows:

1. Location 1

i. Product file

Contains Product ID, Product Name, Sub-category ID, Category ID, Vendor ID, Cost Price, Selling Price fields for the Products.

ii. Vendor file

Contains Vendor ID, Vendor Name, Address, City, State Zip, Country fields for the Vendors.

iii. Transactions file

Contains Transaction ID, Transaction Date, Product ID, Quantity Sold, Tax Amount, Dollars Sold, Transaction Type fields for the Sales Transactions.

2. Location 2

i. Product file

Contains Product ID, Location ID, Product Name, SubCategory ID, Category ID, Vendor ID, Cost Price, Selling Price fields.

ii. Vendor file

Contains Vendor ID, Location ID, Vendor Name, Address, City, State, Zip, Country fields.

iii. Transactions file

Contains Sales Date, Product ID, Quantity, Amount fields.

3. Location 3

i. Product table

Contains Product ID, Description, Manufacturer, Manufacturer Product ID, Category, Sub category, Vendor, Taxable flag, Tax , Discount, Max Discount, Cost Price, Selling Price, Status, Notes, Color, Size etc. fields.

ii. Vendor table

Contains Vendor ID, Company Name, Address, City, State, Country, Zip, email address, website, phone number, contact person, carrier, terms etc fields.

iii. Transactions table

Contains Auto ID, Transaction ID, Product ID, Quantity, Discount ID, Discount Amount, Adjusted Price, Tax Amount, Transaction Type, Comments, Cashier ID, Transaction Date etc fields.

4.4.2 Data Preprocessing, Cleaning & Transformation

The main tasks are:

1. Removing unwanted fields (several extra fields in data sources).
2. Checking for missing values (if product ID or Vendor ID is missing).
3. Checking for inconsistent data (ex: wrong vendor ID or product ID, maybe due to typo).
4. Checking for duplicate values.
5. Transforming to appropriate data type.
6. Extracting subset of data values.
7. Calculating aggregates for data. Example: total quantity sold for a product over a period of time.
8. Adding any necessary field example Location ID, necessary to help end-user analysis.

The solutions are provided to the main tasks mostly using SSIS packages and some by applying appropriate constraints on the Datawarehouse as given below –

1. Removing unwanted fields – by selecting only the necessary fields in the Data Flow.
2. Checking for missing values – by directing “Error Row” output to a destination file which captures all records with missing values.
3. Checking for inconsistent data or wrong values – by performing “Lookup operation” with corresponding dimension table.

4. Checking for duplicate values – by applying primary key constraint to Datawarehouse tables.
5. Heterogeneity in data types – by performing appropriate ‘Conversions’.
6. Conforming to one schema and appending further data necessary for analysis– by performing all of the above and adding some additional required fields like, LocationKey during the Data Flow task.

4.4.3 Data Loading

Data loading is also done through the same SSIS packages.

One such SSIS package created is shown below, which depicts all these steps –

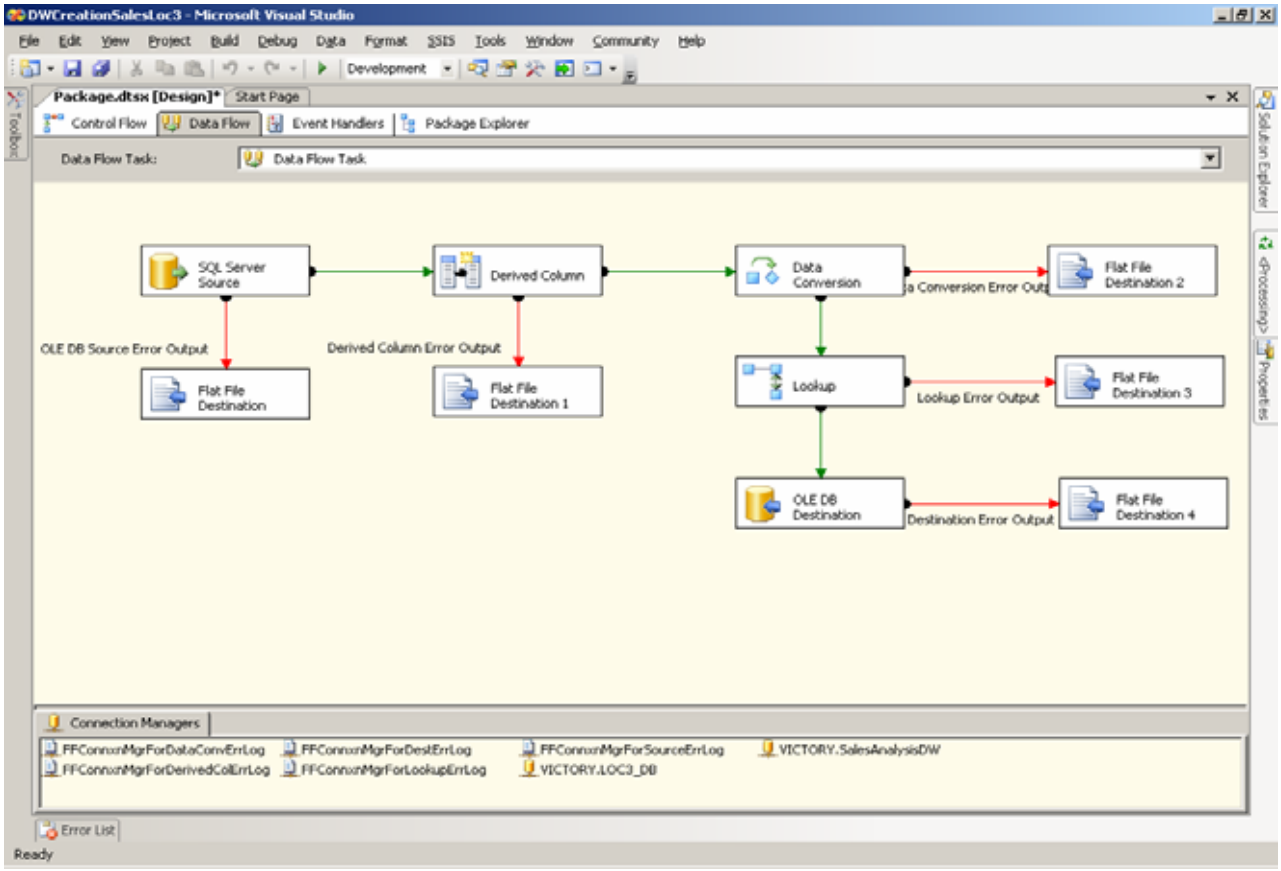

Figure 4.2 SQL Server Integration Services package

4.5 Multidimensional Model of the Datawarehouse

The Datawarehouse is designed based on several factors. The main ones are:

1. The key data that the user seeks to analyze

In this case, we build the Datawarehouse for analysis of sales data.

2. The factors and criteria that the user wishes to apply

We consider the criteria for analysis as product details, vendor details and time factor. In other words, the user wishes to analyze sales data with respect to specific products, vendors or time.

3. The granularity of data the user expects

The design of the Datawarehouse also depends on the amount of data granularity the user expects. For example, the product details may include category information, unit price information etc. Likewise, the sales information may be needed on a daily, weekly and monthly basis.

These considerations lead to the design of the dimension tables and fact table for the Datawarehouse as described in the following section. Fact tables are one or more large tables which contain the primary data necessary for end-user analysis, for example, sales fact table, orders fact table etc. Dimension tables are a number of much smaller tables which contain data supporting the data in the fact tables, for example, product dimension table, customer dimension table etc.

4.5.1 Dimension Tables

The four Dimension tables designed for the ‘Sales Analysis Datawarehouse’ are as listed here.

1. DimProduct

   The fields of this table are: Product Key, Location Key Product Name, Sub category, Category, Vendor Key, Cost Price, Selling Price

2. DimVendor

   The fields of this table are: Vendor Key, Location Key, Vendor Name, Address, City, State, Zip Code, Country.

3. DimTime

   The fields are: Date Key, Week, Month, Quarter, Year.

4. DimLocation

   The fields are: Location Key, Location Name, City, State, Zip Code, Country.

4.5.2 Fact Table

The Fact table designed for the Sales Analysis Datawarehouse, ‘FactSales’ contains the measures:

1. Dollars Sold.
2. Quantity Sold.

It also contains the Keys to the Dimension tables:

1. ProductKey

2. VendorKey
3. DateKey
4. LocationKey

The multidimensional model of the Datawarehouse is as shown below:

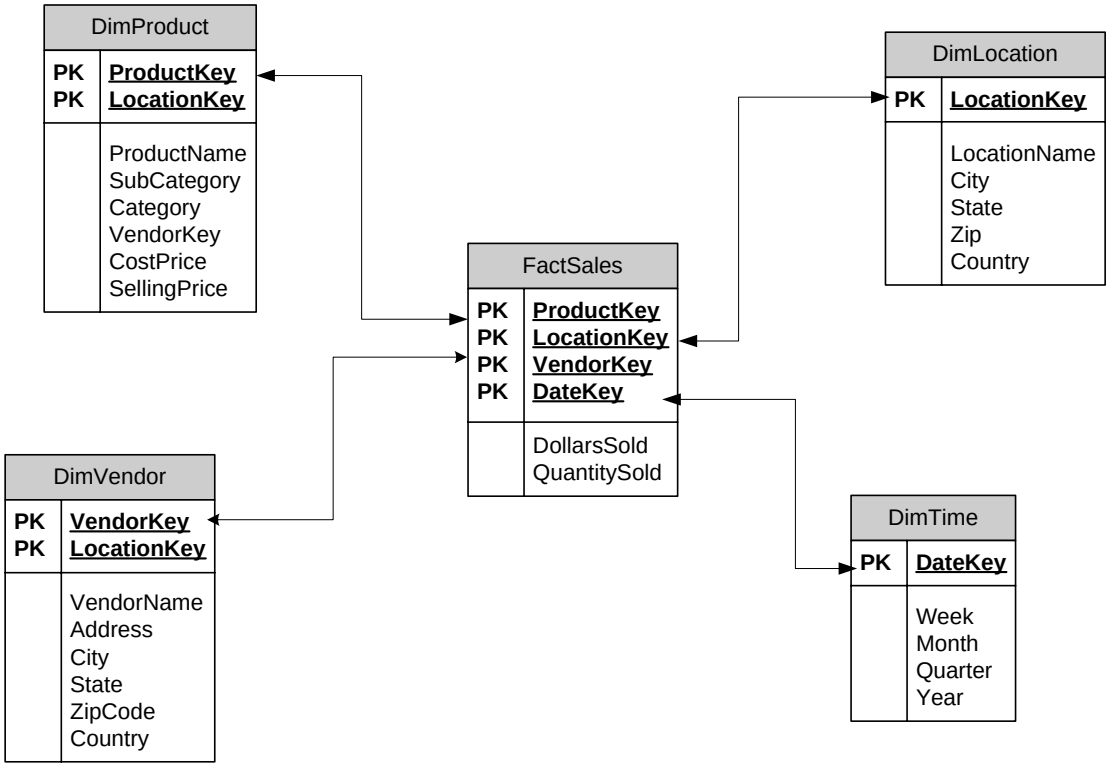

Figure 4.3 Multidimensional model of Sales Analysis Datawarehouse

The Datawarehouse is implemented using SQL Server 2005. It is shown below in the SQL Server Management Studio.

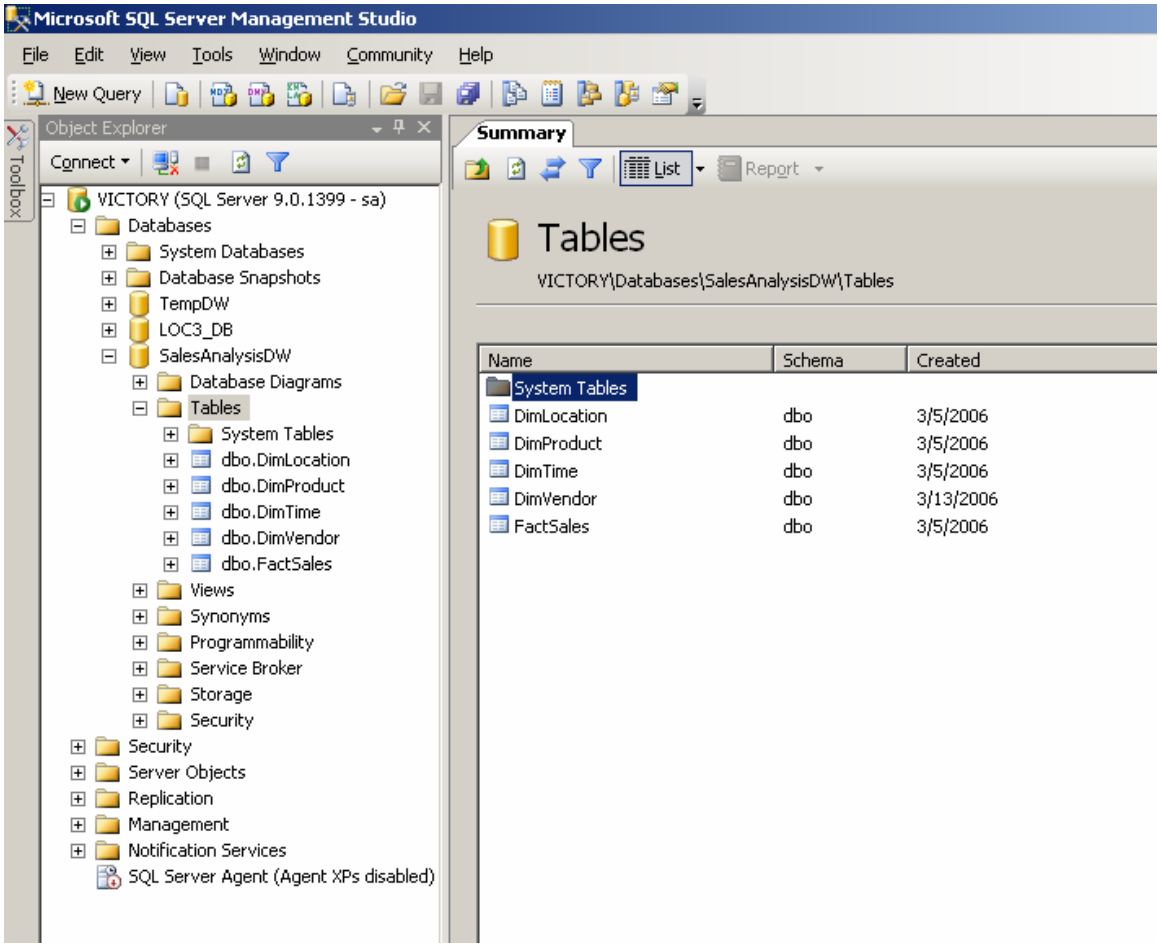

Figure 4.4 Sales Analysis Datawarehouse in SQL Server Management Studio

4.6 Analysis of Data to Support Decisions

Once the Datawarehouse is built, there is a plethora of ways by which the data can be analyzed. To name a few:

1. Data Analysis using OLAP Server –

The Datawarehouse built can serve as the basis for building a Data Cube using an On-line Analytical Processing (OLAP) Server. This is an exceptionally versatile way of

analyzing data by performing various operations allowed by the Data Cube like drill-down, roll-up, slice, dice etc. Microsoft SQL Server 2005 provides Business Intelligence Development Environment using which, we can build a Data Cube with minimum efforts.

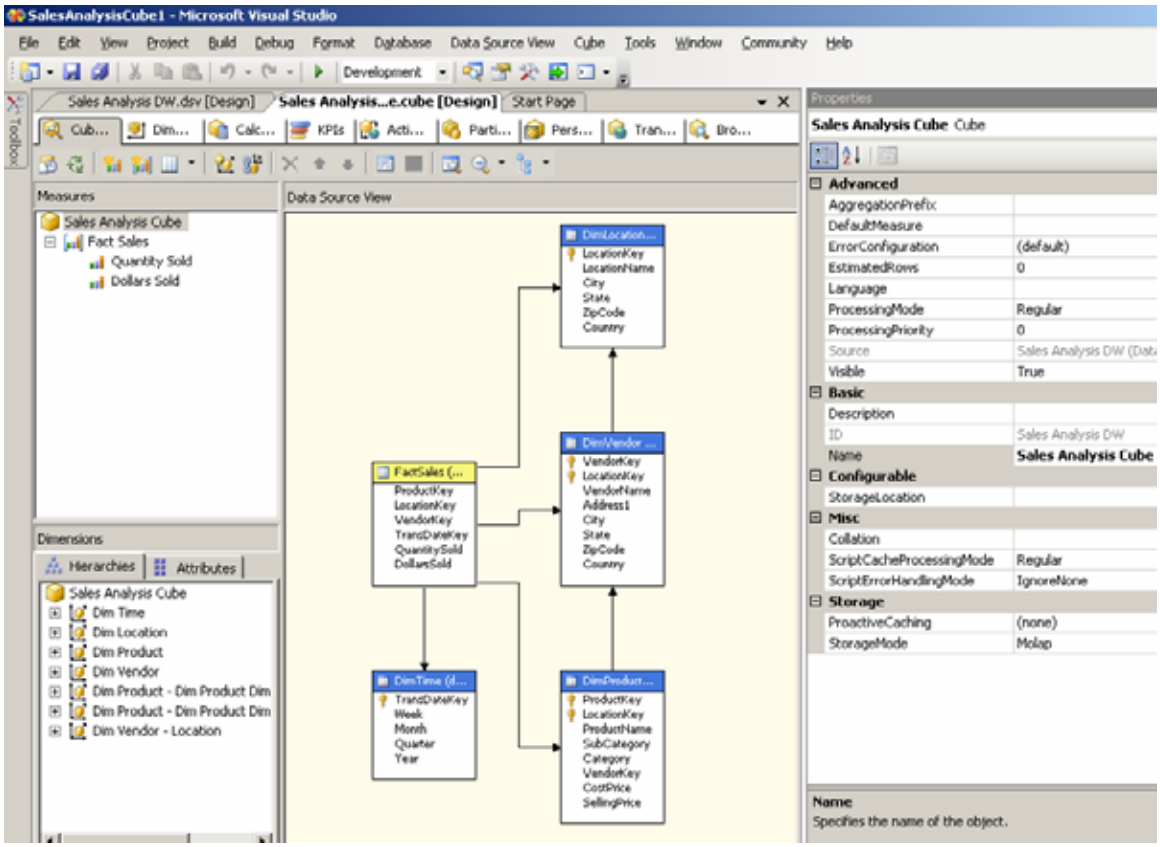

Figure 4.5 Data Cube from Sales Analysis Datawarehouse

Once built, this data cube can be used to analyze the data in various ways with actions as simple as dragging and dropping the fields onto appropriate areas on a cube browser.

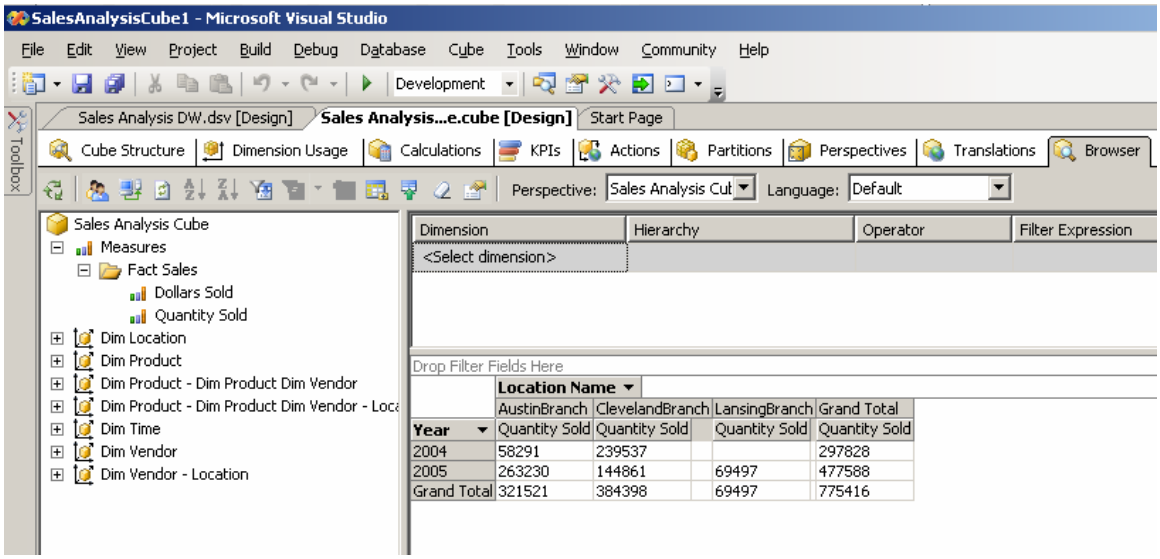

Figure 4.6 Analyzing data using Sales Analysis Cube

2. Data Analysis using Charts, Graphs etc –

The data in the Datawarehouse can be analyzed graphically by creating charts, graphs etc for the data. Software like Microsoft Excel simplifies the process of creating effective charts and graphs by connecting to this Datawarehouse. One way of doing this is by using the PivotTable and PivotChart features of Excel, through which we can connect to an OLAP data cube and desired charts or graphs can be created on Excel using this data.

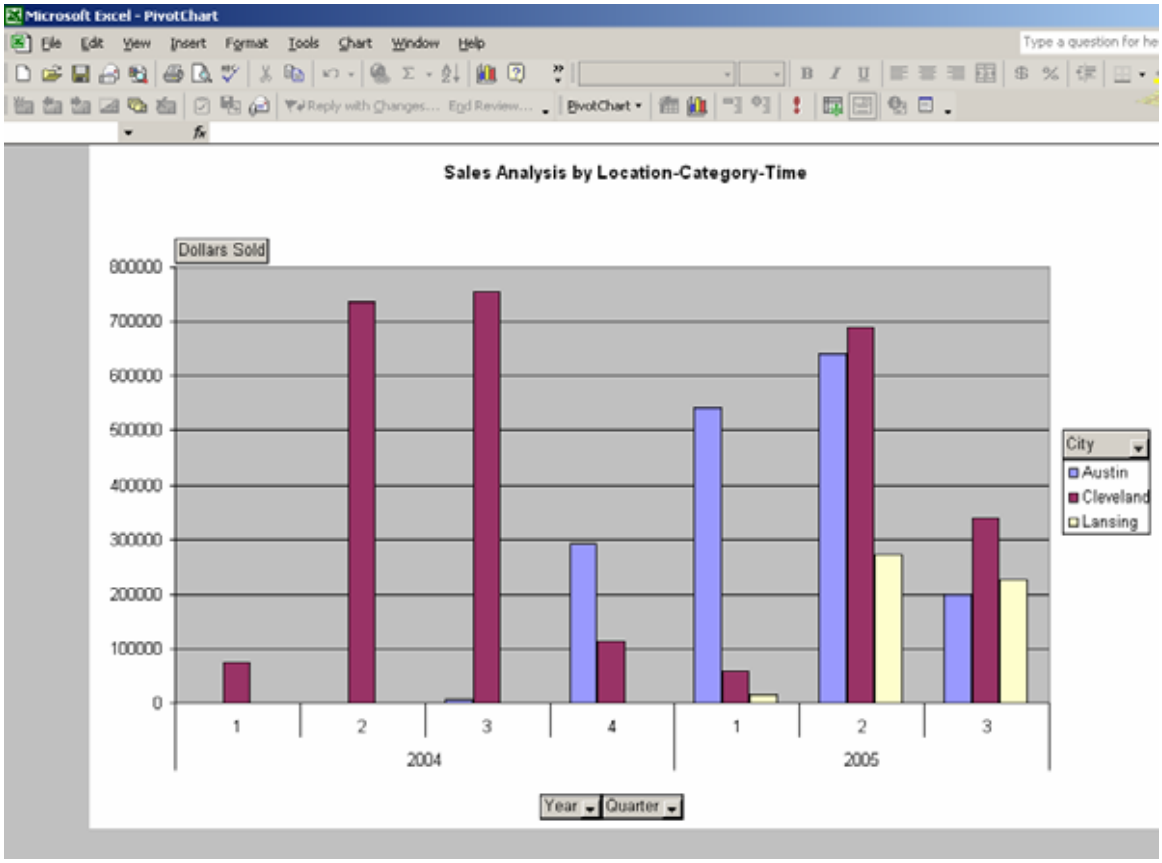

Figure 4.7 Graph based on Data Cube

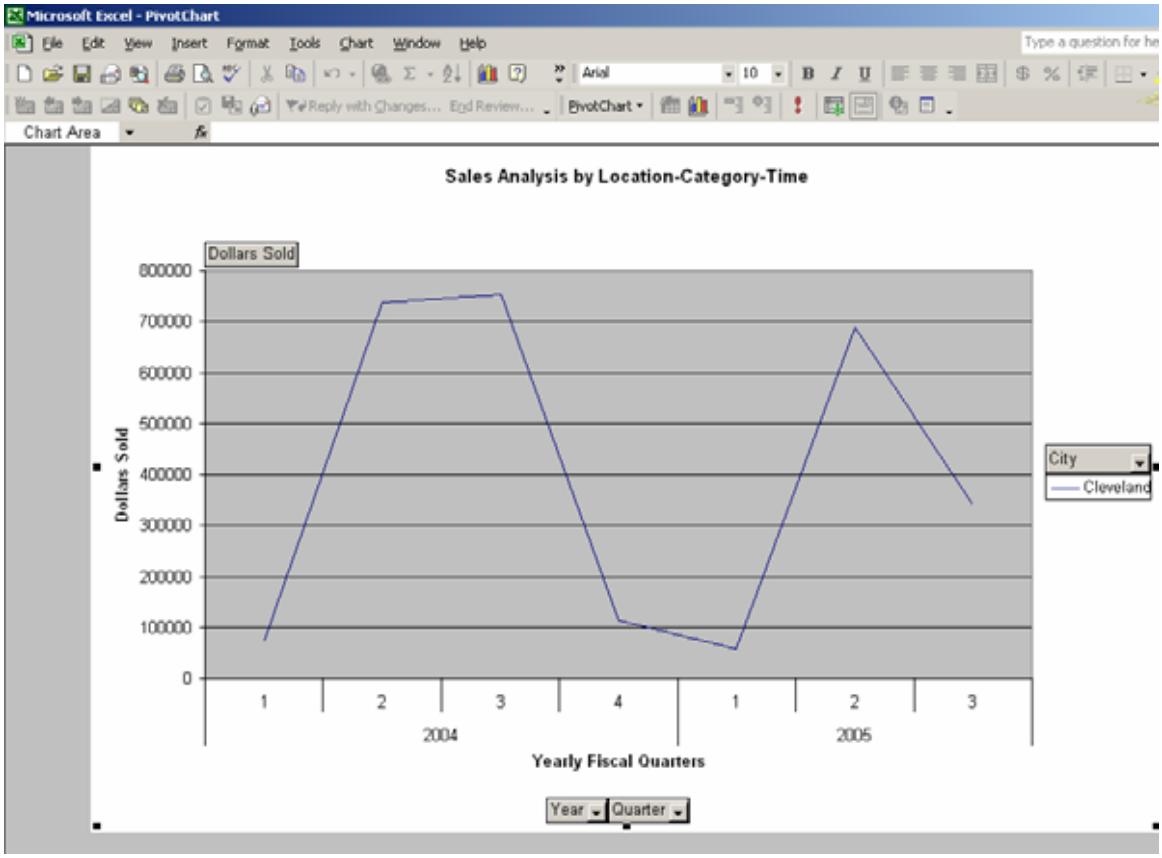

Figure 4.8 Chart based on Data Cube

3. Interactive Data Analysis through a GUI –

We can also build a Graphical User Interface (GUI) for the User to interact with the Datawarehouse data without having to actually know the low-level details of the data. By means of several buttons, boxes etc, the User can simply ask questions to the System and internally appropriate queries will run against the Datawarehouse built and return the result to the User in an easy-to-understand format.

CHAPTER V

EFFICACY OF THE SOLUTION

The Datawarehouse is implemented to provide a solution to support the decision making needs in settings similar to the business scenario considered in the case study. The efficacy of this solution is hard to be quantified, but can be evaluated in terms of its effectiveness in meeting the specific criteria that abound in such cases. They are as follows:

1. There are multiple data sources that need to be examined to make decisions.
2. The data sources are distributed geographically.
3. The data sources are dissimilar in nature – their format, structure, concept, representation etc vary.
4. Some of the data sources may not be available for access.
5. Yet other sources may be unavailable from time to time.
6. Administrators of some data sources may be hesitant in allowing direct access as and when needed to these operational sources due to security concerns.
7. It is necessary to maintain historical data spanning years of operation for analysis.
8. Decision cycles need to be short and should not be impaired by long access times or

connection problems.

9. The operational systems should not be burdened by storing voluminous data or performing resource-intensive analysis or complex query processing work.

10. There is a need for ad-hoc querying with results in the form of graphical interactive reports involving data from specific sources or several data sources in combination.

All these criteria are met when the approach used for Information Integration is Datawarehousing, as discussed and demonstrated in this thesis.

CHAPTER VI

CONCLUSION AND FUTURE WORK

The most suited approach of Information Integration to build a Decision Support System for the business scenario considered is Datawarehousing, which involves building a Datawarehouse which the Corporate Office has access to. This Datawarehouse contains only the data that is required for analysis and is updated on a periodic basis, such that all the data until then is transferred from the various branches to the central Datawarehouse. If updated on a daily basis (every night), the Datawarehouse would have data at most stale by one day. Recent technologies narrow the gap between updates. So, this enables the Corporate Office to get almost real-time data for analyzing. Once the Datawarehouse is built, the data can be analyzed by using an OLAP Server, simply a spreadsheet program to create graphs and charts or by building a GUI for interactive querying.

Suggestions for possible future work are:

1. The Datawarehousing approach could be applied to specific scenarios in other domains like Healthcare, Research etc where valuable, critical and often life-saving information can be obtained by data analysis to identify whether this is an efficient approach to handle concerns specific to those fields.

2. A complete system could be implemented to automate the entire process of extracting the data from the sources maybe through an FTP Server in the middle layer and rebuilding the Datawarehouse on a periodic basis to obtain most recent data.

3. Other applications of Datawarehousing could be studied, for example: for monitoring the status of the individual locations after decisions have been brought into effect.

REFERENCES

1. Chan C. C., 3460:676 Data Mining course http://www.cs.uakron.edu/~chan/

2. Han, Jiawei and Micheline Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, 2001.

3. Garcia-Molina H. et. al., Database Systems the Complete Book, Prentice Hall 2002, 0-13-031995-3, Chapter 20

4. Gio Wiederllold, Intelligent Integration of Information. ACM-SIGMOD 93, Washington DC, May 1993, pages 434-437.

5. SQL Server 2005 website. Retrieved Oct 2005, http://msdn.microsoft.com/sql/bi/integration/

6. Jaime A Reinoso Castillo, Adrian Silvescu, Doina Caragea, Jyotishman Pathak, Vasant G Honavar, Information Extraction and Integration from Heterogeneous, Distributed, Autonomous Information Sources – A Federated Ontology-Driven Query-Centric Approach. The 2003 IEEE International Conference on Information Reuse and Integration, IEEE Press. pp. 183-191.

7. Hector Garcia-Molina, Yannis Papakonstantinou, Dallan Quass, Anand Rajaraman, Yehoshua Sagiv, Jeffrey Ullman, Vasilis Vassalos, Jennifer Widom, The TSIMMIS Approach to Mediation:Data Models and Languages. Journal of Intelligent Information Systems 8, 117–132 (1997)

8. Ramana Yerneni, Ohen Li, Hector Garcia-Molina, Jeffrey Ullman, Computing Capabilities of Mediators Approach. ACM Sigmod Record, Volume 28, Issue 2, 443-454 (1999).

9. Mark Humphries, Michael W. Hawkins, Michelle C. Dy. Data Warehousing: Architecture and Implementation.

10. Power, D. J. Retrieved Feb 2006 from http://DSSResources.COM

11. Larry Greenfield, The Data Warehousing Information Center. Retrieved Feb 2006 from http://www.dwinfocenter.org/

12. Langseth, J., "Real-Time Data Warehousing: Challenges and Solutions", DSSResources.COM. Retrieved Feb 2006 from http://dssresources.com/papers/features/langseth/langseth02082004.html

13. Ralph Kimball Associates, Inc., Articles. Retrieved Feb 2006 from http://www.kimballuniversity.com/html/articles.html

14. Martin Doerr1, Jane Hunter2, Carl Lagoze3, Towards a Core Ontology for Information Integration. Retrieved Feb 2006, from http://jodi.ecs.soton.ac.uk/Articles/v04/i01/Doerr/doerr-final.pdf

15. Marcin Policht, SQL Server 2005 - SQL Server Integration Services. Retrieved Mar 2006, from http://www.databasejournal.com/features/mssql/article.php/3503996