Strategy and methodology for enterprise data warehouse development. Integrating data mining and social networking techniques for identifying different communities within the data warehouse.

Strategy and Methodology for Enterprise Data Warehouse Development: Integrating Data Mining and Social Networking Techniques for Identifying Different Communities within the Data Warehouse

by

Mohammad Rifaie

Submitted in accordance with the requirements for the degree of Doctor of Philosophy

in Computer Science

The University of Bradford

Bradford, United Kingdom

School of Computing, Informatics and Media

June 2010

The candidate confirms that the work submitted is his own and that appropriate credit has been given where reference has been made to the work of others.

Acknowledgments

It gives me great pleasure to acknowledge the coaching and counselling of my supervisors Mr M.J Ridley of the University of Bradford and Dr Reda Alhajj of the University of Calgary. Their support was critical and instrumental in all activities related to this thesis. There were times when Dr Alhajj made himself available during off hours and several weekends at the expense of his personal time and life and for that I am both indebted and grateful. His dedication to his students is only surpassed by his perseverance which is unbound. He is a peerless professor and an undaunted educator. The feedback from Mr Ridley has always been valuable in shaping and directing my research discoveries; I benefitted a lot from the regular meetings I had with him. I am impressed with his sharp vision and careful reading of the different drafts of my thesis. Also, I would like to thank the internal examiner Dr. Daniel Neagu and the external examiner Dr. Anne James for their constructive feedback that improved the quality of my thesis.

There were many times when my resolve to complete the requirement of this thesis was tested. There were times when doubt breached the lines of my determination to continue the journey of higher education. My mother's encouraging phone calls recharged me. Her unwavering stand and uncompromising position for education blew winds under my wings to soar to greater heights. It is without a shred of a doubt that obtaining the designation of a Doctor wouldn't be possible without her. I hereby bow my head to the greatest mother, my mother.

I will be grossly remiss if I don't thank my great cheerleader and fan. My wife has been standing by my side throughout my undergraduate and postgraduate studies. I would like to thank her for declining to marry me before I received my undergraduate degree. She has been my greatest supporter and counsellor throughout my professional career. Thank you Rana.

I would like to thank many colleagues at RBC Financial Group and the University of Calgary whose support and encouragement motivated and guided me while producing this thesis. They made their resources available to me once needed.

Mohammad Rifaie

Declaration

Parts of the original work proposed in this thesis have appeared in the following publications and presentations.

Publications and Talks

- M. Rifaie, K. Kianmehr, R. Alhajj and M.J. Ridley, “Data modelling for effective data warehouse architecture and design”, *International Journal Information and Decision Sciences*, Vol.1, No.3, pp. 282-300, 2009.

- M. Rifaie, R Alhajj and M. Ridley, “Data Governance Strategy: A Key Issue in Building Enterprise Data Warehouse”, *Proceedings of International Conference on Information Integration and Web-based Applications & Services* (ACM Press), pp. 587-591, Kuala Lumpur, Malaysia, December 2009.

- E. J. Blas, A. M. Muhsen, T. T. H. Mok, M. Rifaie, K. Kianmehr, R. Alhajj and M.J. Ridley, “Data Warehouse Architecture for GIS Applications”, *Proceedings of International Conference on Information Integration and Web-based Applications & Services* (ACM Press), pp.178-185, Linz, Austria, November 2008.

- M. Rifaie, K. Kianmehr, R. Alhajj and M. J. Ridley, “Data warehouse architecture and design”, *Proceedings of IEEE International Conference on Information Reuse and Integration*, pp.58-63, Las Vegas, NV, July 2008.

- Stuart Harvey Rubin, Shu-Ching Chen, Lotfi A. Zadeh, Hojjat Adeli, Mohammad Rifaie, Gordon K. Lee, Kang Zhang, Reda Alhajj, Gary D. Boetticher, Du Zhang: Panel: The role of information search and retrieval in economic stimulation. *IEEE International Conference on Information Reuse and Integration*, July 2008.

- Invited Talk at the Sixth IASTED International Conference on Communications, Internet, and Information Technology, Banff, Alberta, Canada, July 2007.

Abstract

Data warehouse technology has been successfully integrated into the information infrastructure of major organizations as potential solution for eliminating redundancy and providing for comprehensive data integration. Realizing the importance of a data warehouse as the main data repository within an organization, this dissertation addresses different aspects related to the data warehouse architecture and performance issues.

Many data warehouse architectures have been presented by industry analysts and research organizations. These architectures vary from the independent and physical business unit centric data marts to the centralised two-tier hub-and-spoke data warehouse. The operational data store is a third tier which was offered later to address the business requirements for inter-day data loading. While the industry-available architectures are all valid, I found them to be suboptimal in efficiency (cost) and effectiveness (productivity).

In this dissertation, I am advocating a new architecture (The Hybrid Architecture) which encompasses the industry advocated architecture. The hybrid architecture demands the acquisition, loading and consolidation of enterprise atomic and detailed data into a single integrated enterprise data store (The Enterprise Data Warehouse) where business-unit centric Data Marts and Operational Data Stores (ODS) are built in the same instance of the Enterprise Data Warehouse.

For the purpose of highlighting the role of data warehouses for different applications, we describe an effort to develop a data warehouse for a geographical information system (GIS). We further study the importance of data practices, quality and governance for financial institutions by commenting on the RBC Financial Group case.

The development and deployment of the Enterprise Data Warehouse based on the Hybrid Architecture spawned its own issues and challenges. Organic data growth and business requirements to load additional new data significantly will increase the amount of stored data. Consequently, the number of users will increase significantly. Enterprise data warehouse obesity, performance degradation and navigation difficulties are chief amongst the issues and challenges.

Association rules mining and social networks have been adopted in this thesis to address the above mentioned issues and challenges. We describe an approach that uses frequent pattern mining and social network techniques to discover different communities within the data warehouse. These communities include sets of tables frequently accessed together, sets of tables retrieved together most of the time and sets of attributes that mostly appear together in the queries. We concentrate on tables in the discussion; however, the model is general enough to discover other communities. We first build a frequent pattern mining model by considering each query as a transaction and the tables as items. Then, we mine closed frequent itemsets of tables; these itemsets include tables that are mostly accessed together and hence should be treated as one unit in storage and retrieval for better overall performance. We utilize social network construction and analysis to find maximum-sized sets of related tables; this is a more robust approach as opposed to a union of overlapping itemsets. We derive the Jaccard distance between the closed itemsets and construct the social network of tables by adding links that represent distance above a given threshold. The constructed network is analyzed to discover communities of tables that are mostly accessed together. The reported test results are promising and demonstrate the applicability and effectiveness of the developed approach.

Dedication

To the memory of my father who I miss everyday; he did not only raise and nurture me but also taxed himself dearly over the years for my education and intellectual development.

Table of Contents

List of Tables

List of Figures and Illustrations

Chapter One: **INTRODUCTION**

In an effort to encapsulate data within an organization under a single umbrella, the enterprise data warehouse is an effective and attractive structure for data storage and retrieval. To continue the success stories of adapting an enterprise data warehouse as the basic ingredient in the information infrastructure of an organization, it is necessary for experts and practitioners to understand better the different aspects of data warehouse development and utilization. Once an enterprise data warehouse is functional, employing machine learning and social network construction and analysis techniques would help in improving the performance of queries, which is a serious and major attraction for accepting a new data repository. These are covered in this dissertation.

1.1 Problem Definition and the Motivation

The rapid growth in the volume of data produced by organizations and the diversity in the sources of such data necessitates the development of powerful integration tools that are capable of systematically combining the data to produce comprehensive answers to user queries [14]. A data warehouse is accepted as the main structure that could successfully satisfy the target. However, the development of a data warehouse is a challenging task that requires a number of skills and qualifications be possessed by members of the design team who will accomplish the process.

Practical experience in the design, development and deployment of the enterprise data warehouse at RBC Financial Group provided the genesis for this thesis which reports the novel architecture that I have found the most appropriate based on my investigation of the technology over the past two decades as described next.

In the early 1990s, building business-unit specific data marts was recommended as the architecture which generates quick business value at a reduced cost. The expensive cost of hardware and software, especially the high cost of disk storage lured the majority of data warehouse practitioners to adopt the data mart architecture. Advancements in customers' needs and requirements for consistent experience across various business units and at all touch points for the same business compelled business to combine customer data from all business units and domains. Consequently, information management executives were asked to standardise and consolidate customer data in a manner which enables easy and fast access to accurate and integrated data.

During mid 1990s, industry analysts and gurus in the data warehousing space offered "the Enterprise Data Warehouse Architecture" and the central repository of the enterprise's data and the "Single Version of The Truth". However, it was recommended to maintain/build business-unit specific data marts. According to this "hub and spoke" architecture, data are to be extracted, standardised and loaded into the enterprise data warehouse. Business-unit specific data requirements are to be sourced from the enterprise data warehouse. During that period, business requirements to increase the frequency of data loads into the warehouse spawned a new mutation of data warehouse architecture. Operational Data Stores (ODS) became the new buzz in the industry. It was recommended that data should be loaded into the ODS with minimal latency from the time a transaction taking place to enable operational reporting. At a later time, ODS data should be standardised, integrated and then loaded into the enterprise data warehouse.

Subsequently, enterprise data warehouse data will be farmed out as requested to the business-unit specific data marts. In effect ODS was a coping architecture with slow CPU and data retrieval. However, advancement in CPU and disk speed allowed enterprise data warehouses to act as an operational data store for operational reporting while providing a platform for enterprise decision making and analytics.

As business units realized the value of storing historical data, data marts began to reproduce like rabbits. However, managing them was as difficult as herding cats. High cost and difficulty in managing data Marts became more and more evident.

Closely watching and living the development as outlined above, I realized the need for a comprehensive solution. Thus the main theme of the work described in this dissertation is to propose the hybrid architecture as an integrated solution for data warehouse development and accordingly outline the key steps to be applied in constructing, developing and designing a data warehouse. Once a data warehouse is functional based on the hybrid architecture, this study focuses on data practices, quality and governance. A concise and comprehensive coverage of these topics is required to guide the development of an enterprise data warehouse. After a data warehouse is built, effective query processing becomes imperative. This could be satisfied by considering the history of queries to predict tables that are expected to be accessed together in the same upcoming query. To sum up, the aims of this thesis could be outlined as follows. First, from my research and industrial experience I realized the need for the development of a hybrid data warehouse approach; this aim has been satisfied by the material covered in Chapters 2-5. Second, once a data warehouse is put in practice performance issues raise problems; hence my second aim is to develop an approach that could lead to improved performance by benefitting from the current developments in pattern mining and social networks; this aim is achieved by the material presented in Chapters 6 and 7.

1.2 Methodology

While pondering ways and means to contain data warehousing cost and increasing its value, I proposed the “Hybrid Data Warehouse Architecture”. The main theme of this proposal is to consolidate all the data for the enterprise on a single database platform. Existing business-unit specific data marts are to be fork-lifted into the enterprise data warehouse platform. Data marts became logical and physical grouping of tables. New data marts are to be designed and constructed as logical integrated data constructs.

Building the enterprise data warehouse congruent with the hybrid architecture and eliminating the need for building physical data marts yielded a new class of issues which are not evident in smaller data marts. Obesity, difficult to navigate and slower query turn-around time are chief among several others. While hybrid architecture solved many issues on integration and standardization, it posed new performance challenges.

Traditionally, one of the ways IT professional handled performance issues is to augment the existing platform with additional hardware and new and improved releases of software. This approach is reactive and costly. Leveraging existing tried and true approaches such as frequent pattern mining and the social network model are the approaches tested in this thesis.

Data mining provides a set of powerful techniques for prediction. Data mining techniques utilize existing and past experience to predict future behaviour. Frequent pattern mining is the most fitting technique to satisfy this purpose. Frequent pattern mining is the first and most time consuming step in association rules mining which was developed in 1993 by Agrawal et al. [1, 2] for market basket analysis. The main motivation was to identify items that are sold together most of the time. For this purpose, a table is constructed as the input to the model. Each column represents an item and each row corresponds to a transaction which includes a set of items bought by a customer in one visit to the market. Entries in the table may be either binary or quantitative. Binary representation reflects whether an item has been bought or not. Quantitative values reflect the actual quantity of items bought by a customer. The set of transactions is processed to find sets of items that appear in most of the transactions. These sets of items are called frequent itemsets once they satisfy a minimum support threshold specified mostly by experts. Automated methods may be applied to determine minimum support threshold; such methods are out of the scope of this dissertation.

Support is the percentage of transactions that contain an itemset. A frequent itemset is said to be closed if its frequency is different from its supersets. Concentrating on closed frequent itemsets will reduce the number of itemsets without losing any information. In this study, a transactional database to be mined for frequent itemsets by accepting each query as a transaction and each table in the warehouse as an item.

Frequent itemsets do overlap and taking the union of overlapping itemsets may not be effective in identifying tables that are mostly accessed together. The outcome depends on how much overlapping should be two closed frequent itemsets in order to be merged. For this purpose, the problem was modeled as social network where closed frequent itemsets are the actors and a link is added to the network based on the Jaccard distance between the actors. The Jaccard distance depends on the intersection and union between two itemsets where the more overlapping is the intersection the closer are the two itemsets. Building such a social network will allow us to discover sets of tables that are mostly accessed together.

Results from the work described in Chapters 6 and 7 are very attractive and promising. These results enable database professional, including database administrators, to construct and organize the tables in a manner that facilitates optimal response to queries and report generation while maintaining lower total cost.

Finally, business intelligence involves many data intensive and computationally intensive analytics which test the scalability and performance of any software/hardware platforms. The application of frequent pattern mining and the social network model are very innovative way in constructing efficient and effective multi Terabyte databases.

1.3 Contributions

The work described in this dissertation comprehensively satisfies two main themes. It first covers the data warehouse construction, development and design process by proposing a hybrid architecture, and then integrates frequent pattern mining and social network techniques to improve the performance of queries. It has several contributions that could be enumerated as follows:

1) A concise and clean hybrid architecture is described for data warehouse construction, development and design [63, 64].
2) An application of data warehouses in the GIS domain is presented [5].
3) Data practices, quality and governance are discussed and related issues are highlighted [65]

4) A frequent pattern mining based model has been developed for identifying frequently accessed tables

5) A social network model has been constructed for determining a wider scope of tables forming communities based on the history of queries that have been executed on the data warehouse.

Finally, it is worth mentioning that the last two contributions (4) and (5) are still to be submitted for publication.

1.4 Organization of the thesis

In addition to this introduction chapter, there are five chapters in this thesis.

Chapter 2 describes the data warehouse construction process; different modeling aspects are described and the data warehouse development process is highlighted. Chapter 3 covers metadata management in data warehouse environment. Chapter 4 presents the development of a data warehouse for GIS application. Chapter 5 discusses data practices, quality and governance. Chapter 6 presents the data mining model that identifies tables which are expected to mostly appear in the same query. This chapter also includes a brief overview of the basic data mining concepts required to understand the developed model. Chapter 7 presents the social network model that uses the outcome from Chapter 6 as actors and analyzes the network to discover more comprehensive sets of tables that are candidates to be accessed together in future queries. Chapter 8 is summary, conclusions and future research directions.

Chapter Two: **DATA WAREHOUSE ARCHITECTURE, DEVELOPMENT AND DESIGN**

2.1 Introduction

A data warehouse is as attractive as the main repository of an organization's historical data; it is optimized for reporting and analysis. In this chapter, the process of data warehouse architecture, development and design will be presented. The different aspects that are required for building a data warehouse are highlighted. These range from data store characteristics to data modeling and the principles to be considered for effective data warehouse architecture.

This chapter is organized in six sections. Section 2.1 covers basic business requirements for data warehouse design. Section 2.2 covers data warehouse design. Section 2.3 presents data warehouse data stores. Section 2.4 describes the data model. Section 2.5 outlines enterprise data warehouse design principles. Section 2.6 covers data sharing in a data warehouse environment. Section 2.7 includes some concluding remarks.

2.2 Business Requirements

Business communities all across organizations are becoming increasingly dependent on their ability to quickly access, easily use, effectively share and efficiently maintain quality and timely business information which they need to help achieve success in their business objectives. Meeting these needs is the basis of the business requirements for the creation and implementation of a data warehouse environment, which will contain, and enable easy access to, all the required business information. These requirements include business user needs for:

1) More consistent, quality information on all aspects of the company's business;

2) Greater capability to work with information directly, and therefore quickly satisfy varying informational requirements;

3) A clear and concise capability to determine, and understand in their terms, what information is available and how to access it;

4) Less dependency on IT professionals;

5) Increased ability to access and work with enterprise data;

6) Increased ability to create and share enterprise data;

7) The ability to add value to data when producing information for analysis or decision-making.

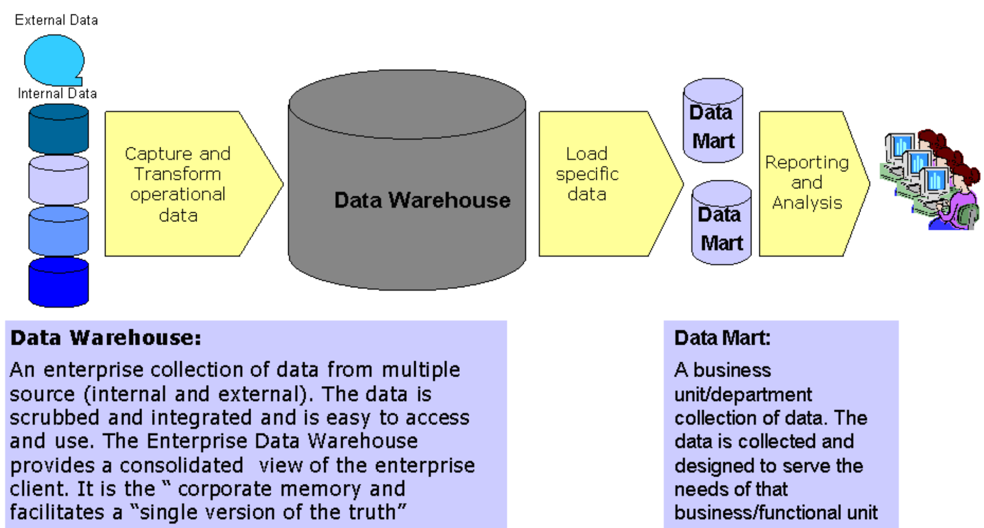

Figure 2.1 General enterprise data warehouse

Data warehousing processes are used to design and develop data repositories for efficient enterprise reporting and decision support systems; data warehouse design and development already attracted the attention of several researchers, e.g., [4, 12, 13, 14, 24, 25, 28, 35, 46, 49]. Sen and Sinha conducted data warehouse related comparative analysis

[48]. Kimball states that a data warehouse is a queryable presentation for enterprise data and that this presentation must not be based on an entity-relation model [14, 26]. Data warehouses have become a very important aspect of data management for businesses. There is no de facto standard for data warehousing techniques but the basic methods and processes outlined by Kimball [26], Chaudhuri and Dayal are an excellent place to start [25].

This chapter presents the requirements for a data warehouse architecture that meets the above-enumerated needs effectively. The main motivation for choosing to build a data warehouse is to enable users to report on tactical and strategic information. In other words, the enterprise data warehouse (see Figure 2.1) must have a robust, flexible, adaptable and scalable design and data architecture. This data architecture is essentially the enterprise's data infrastructure, which maintains data on important historical and current business information. The data is structured in an easy to use and access manner for servicing the direct and immediate analysis and decision support needs of business users at all levels of the enterprise using methods and techniques considerably different from those used by existing transactional production applications for maintaining and accessing transactional data.

2.3 Data Warehouse Construction

Enterprise data warehouse (EDW) data originates from a variety of different sources. These could include: 1) The EDW database needs to be designed and integrated in a way which will eliminate many of the inconsistencies which have evolved over the years in many of the legacy system operational databases and local application data stores. 2) Metadata (technical and business information about the data) is an integral component of a robust Data Warehouse infrastructure. Without this information, it will be extremely difficult for both administrators of the Data Warehouse and users of the data to know and understand the data means and its appropriate usage. Metadata is also vital for the administrators for change management and impact analysis.

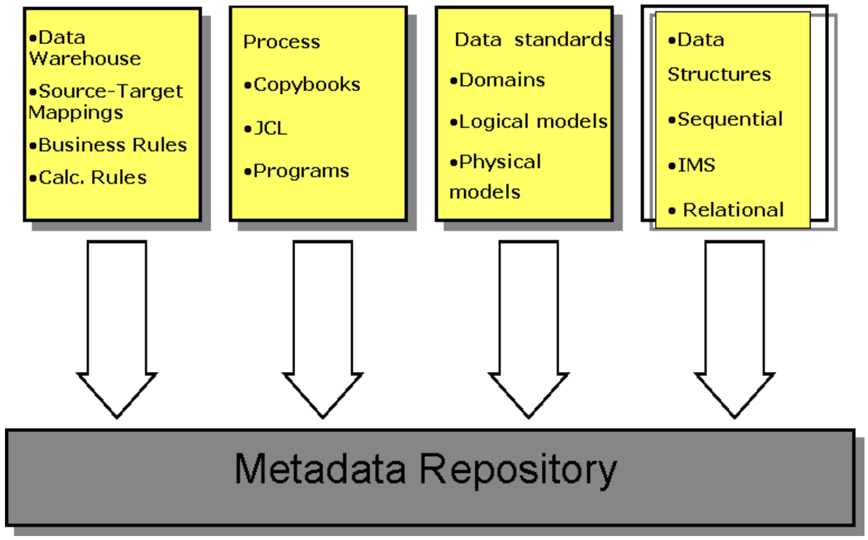

Figure 2.2 Metadata repository contents

3) A metadata repository (see Figure 2.2) is required to maintain descriptive information of all available data in the information warehouse. The structure of the metadata enables business users with easy retrieval and access to the required information in a manner, which is easily understood in business terms.

The data quality of these data stores should be managed by a process of certification, by the owners of the data, to assure all interested users that the data has met the minimum threshold levels of acceptable quality. Important factors of quality, which need to be monitored, include timeliness and completeness of the data stored in the data warehouse. Performance indicators are required to enable monitoring.

Some important design characteristics of information warehouse data-stores which distinguish them from existing production operational data stores include:

1) *None Volatile*: Real time updates occur to selective data warehouse data stores. Most data stores are refreshed in batch, not less than every 24 hours. Time consistent context of data across different sources need to be maintained.

2) *Time Variant*: A 3 to 7 year time horizon for maintaining data is normal for the information warehouse. The 7 year retention is typically driven by regulatory requirements for the retention of data. The data is periodic and maintained as a series of snapshots, taken as of some moment in time. The key structure of data tables must contain some element of time.

3) *Granularized structure*: Data is maintained at various levels of granularity and summarization. Frequently accessed data can be pre-jointed and summarized to enable quick turnaround on queries and reports. Detailed and atomic level data will be maintained alongside summarized and pre-calculated data. New approaches to data storage are evolving. Usage frequency determines approaches to data storage to minimize costs associated with maintaining large and multi-year business data. The concept behind ‘multi-temperature’ data storage strategies is to optimize data access for more frequently used data and isolating infrequently accessed data [66].

EDW minimizes the need to maintain historical information within the operational application data stores. Operational data-bases in the production environment will only maintain historic information if it is absolutely required for processing in “transaction-based” production applications. Otherwise, all historical data beyond "current value" will be maintained in the EDW data stores for access and use by business users for informational analysis and reporting purposes. Costs for storing history data will be optimized by using tables containing different levels of summarization.

A successful approach in migrating towards an effectively architected enterprise warehouse environment is the one which requires much greater levels of involvement from business users than those typically required in the development of operational based applications in production. The best approach involves designing and building the warehouse data environment one increment at a time. This way, technical and business community staff can work closely together through a process of continuous iteration, to design and implement each component of the warehouse until the structure and content of the data, in each component, meets the satisfaction of the business.

The starting point for the migration is the creation of an EDW data model. Initially the model will include the definition and confirmation of subject areas (business and application specific) and high-level list of entities for the information warehouse data model. This level of the model will help to chunk out the planned warehouse data environment into components prioritized by business requirements, specific needs of business user groups, and the readiness of the users to move ahead with this initiative.

The design of each enterprise warehouse component will involve a number of transformation and refinement activities to the related areas of the EDW.

Once the design is complete, and agreed upon by the business users, the tables will be generated and populated in small increments. This will allow users to immediately test the data and report their satisfaction or request for changes.

Data Management standards and guidelines need to be established and maintained for ensuring the quality and integrity of the data in the enterprise warehouse. Procedures and guidelines also need to be established for handling data stewardship, data sharing and change management for data stores within the information warehouse environment.

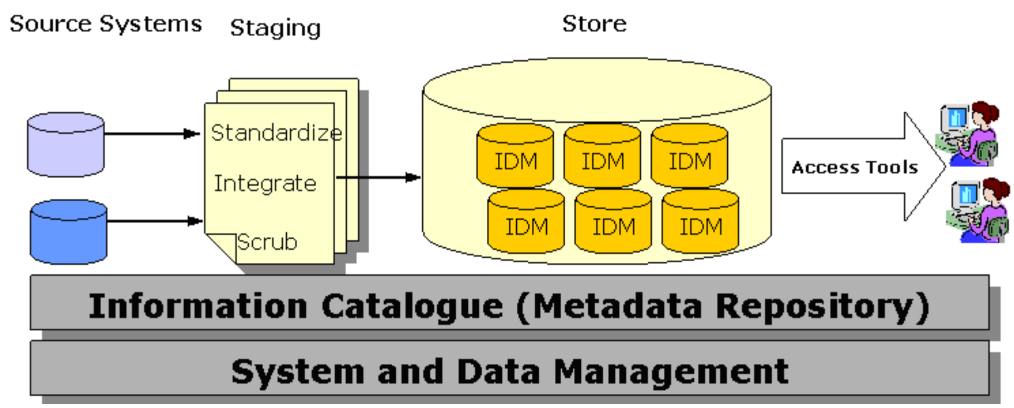

Figure 2.3 High level distinctions between levels of EDW

2.4 Enterprise Data Warehouse Data Stores

The design, construction and effective implementation of an EDW represents a significant variation from the structure and design of the operational database tables maintained in the existing operational environments. The structure of the warehouse will consist of data stores categorized into two different levels (see Figure 2.3). Each level is distinguished by the need to either share the data store across the enterprise or share the data within a business-unit.

Significant differences exist between the properties and characteristics of operational data-stores in production environments and those of the EDW data-stores. These stem from differences in the intended storage and usage of data in the two environments. Operational data stores are typically transaction orientated, detailed, and accurate at the moment of access. The data warehouses data stores are analytical and reporting orientated; they may include summarized or refined data and snapshots of data over defined periods of time.

The following points describe the key characteristics of corporate-wide shared EDW data stores [65]:

- EDW is a collection of shared data-stores which are subject oriented, integrated, and non-volatile and time variant.
- EDW data-stores are organized within major data subject areas. These are defined in the Enterprise Data Model and could typically include areas of common business interest such as customer, product, arrangements etc.
- Integration of the EDW data-stores eliminates many of the inconsistencies which have evolved over many years from the many different designs of applications developed and implemented. Examples of inconsistencies include encoding, naming conventions, physical attributes, etc.
- Integration occurs when data passes from the application oriented operational environment to the data warehouse. For each operational application, routines are developed and run to eliminate data inconsistencies between individual application data-stores before added to EDW.
- EDW data-stores are non-volatile as compared to the operational environment data-stores. EDW data is loaded in a series of batch updates and accessed on-line or in batch. No real-time update to this data occurs in the EDW environment. In contrast, operational data is regularly accessed, manipulated and updated a record at a time.

- An important distinguishing characteristic of EDW data is that it is time variant. This shows up in several ways:
  - The time horizon for the data warehouse is significantly longer than that of operational systems. A 1 to 2 year time horizon is normal for operational systems based on the average lifecycle of transactions; a 3 to 7 year time horizon of data is normal for the EDW.
  - Operational databases contain "current value" data - data whose accuracy is valid as of the moment of access. As such, current value data can be updated. By contrast, EDW data is a series of consistent snapshots, taken as of some moment in time such as end-of-day, end-of-week, and end-of-month as required enabling analytics about the business.
  - The key structure of operational data may or may not contain some element of time, such as year; month days, etc. The key structure of the data warehouse always contains some element of time.
- There are significant differences in the levels of detail of data within the data-stores of the two different environments:

1) Operational data-stores contain complete levels of detail of the most current data as captured in the specific transaction of each operational application.
2) EDW contains data structured at various levels of detail. This could include an older level of detailed data (usually in bulk archival storage), a current level of detailed data, a level of lightly summarized data, and a level of highly summarized data. Usually a significant amount of transformation of data

occurs as data is moved from detailed operational level to various levels of summarized detail in information warehouse level.

2.4.1 Business Unit Data Stores

EDW consists of data stores both at the enterprise and the business unit levels. The following include high-level guidelines for the positioning, development and implementation of business unit data-stores:

- Business unit level data stores (data marts) will be modeled, implemented, maintained jointly with the business unit. This includes those business users who are specifically interested and have the primary need for the data in the business unit data mart.
- The business unit also assumes the role of the data stewardship of this data. They will be primarily responsible for "certifying" the integrity and quality of the data if it is needed for sharing by other business units across the enterprise.
- These business unit data stores will generally have limited interface needs with other business unit data marts. They will physically reside on the same technology platform alongside the EDW and other business unit data marts. The capability of access from other data marts is available for authorized users from other business units.
- A single enterprise data-model would contain all shared and non-shared data marts, although strict rules of "de-coupling" would be employed to ensure operational independence of various entity types within the model.

2.4.2 Historical Data

One of the major impacts of establishing and implementing EDW will be in the handling of historical data. Currently, the operational data stores in the production environment handle and maintain both "current value" and historical data based on the specific applications for which the data store serves. With the implementation of an effectively designed EDW, the need to maintain historical data in the operational application data stores will change.

Historic information should only be maintained in the operational data-bases to a limited degree, if it is absolutely necessary for the processing of any production applications which have been built for updating, accessing or using transaction based operational data.

In all other cases, historic information and other related derived information will be maintained in appropriately designed data-bases within the Data Warehouse Environment. This is especially the case for historic data needed for analytical, data mining and reporting purposes. The purging of data in the Data Warehouse Environment is determined by the need of the enterprise to maintain history and regulatory specifications for the retention of data.

At first glance, this may appear to significantly increase the volume of, and hence the cost to maintain a large amount of data in the Data Warehouse Environment. Although, the volume will increase, the costs can be significantly minimized by using effective designs through the use of different levels of summarization and the elimination of data duplication generated by disparate applications.

2.5 Enterprise Data Model

An important starting point for designing, building and implementing an EDW is the design and construction of an appropriate Data-Model for this environment.

However, the creation and construction of an adequate Data-Model for the Data Warehouse will require the adoption of new or changed techniques from the Classical Data-Modeling techniques which have been generally used for modeling the Operational Environments.

Classical data modeling techniques make no distinctions between operational and informational/analytical environments. These techniques merely try to gather and synthesize the informational needs of the organization resulting in an Enterprise Corporate Data Model which adequately covers the operational data needs of the enterprise, but which does not capture the structural needs of data which will be stored in the EDW. Another classic difference is the extensive number of additional data relationships that are to be considered for analytic purposes.

Enterprise Data Model forms the foundation of the enterprise's existing and 'to-be' data architecture. They represent the existing operational data needs of the enterprise as well provide a template for the integration of new subject matter data across the enterprise. It is specific in its scope for representing the structural data requirements of the data warehouse based on the characteristics of informational/analytical data.

The Enterprise Data Model for the operational environment requires extensive transformations and further refinements if it is to effectively represent the data requirements of the Data Warehouse. Before proceeding with the design and construction of the Data Warehouse data model it is important to understand, and take into consideration, the following three different levels of data-models:

- **The conceptual data model.** This is typically called the entity relationship model (ERD); it was proposed by Chen in late 70s [15].). This level determines the models "Subject Area" and defines the “what” entities (at the highest level) belong in each of these areas. The level also establishes the "scope of integration" which defines the boundaries of the data model. This scope must be agreed by the data architect, management and the ultimate user of the data, before the modeling process commences. This is the level of the enterprise ERD, which is a composite of many individual ERDs that reflect the different views of people across the enterprise.
- **The logical data model.** This level further expands on the detail within the subject areas and high-level entities defined in the high-level data model. Very rarely are mid level models developed at once. The mid level data model for one major subject area is expanded, then the mid level model is fleshed out, and so forth. Constructing the logical data model is the first step towards the data-base design for the project application.
- **The physical data model.** This is created from the logical data model merely by extending the logical ERD to include keys and physical characteristics of the model. This is the level at which most of the transformation takes place for refining the ERD and constructing the Data Warehouse physical model.

The Enterprise Data Model is a very good place to start the process of building a Data Warehouse. However, there is some amount of work that needs to be done on this model in order for it to be readied for the building of the Data Warehouse. A certain amount of transformation must occur to create the Enterprise Warehouse Data Model from. The activities in the transformation are outlined next.

- The removal of purely application specific operational data;
- The addition of an element of time to the key structure of the Data Warehouse if one is not already present;
- The addition of appropriate derived data;
- The transformation of data relationships into data artefacts. Artefacts are a way of capturing snapshots of relationships between entities which change over time.
- Accommodating the different levels of granularity found in the data warehouse;
- Merging like data from different tables together;
- Creation of arrays of data. Arrays are created by stringing together multiple occurrences of any given entity in the operational data store, and creating only one record in the Data Warehouse. This reduces the amount of indexing required to retrieve multiple occurrences of the same entity, and can significantly reduce the cost for data summarizing and accessing for informational reporting.
- The separation of data attributes according to their stability characteristics. This is the act of grouping attributes of data together based on their propensity for change.

This list clearly indicates that a significant, carefully planned and coordinated, effort must be undertaken in order to effectively "re-fine" Enterprise Data Model for constructing the Data warehouse Data Model.

An important premise, which cannot be overlooked, is that the Enterprise Data Model must be current and up-to-date before proceeding with its refinement. If the Enterprise Data Model does not adequately represent the most current business needs and requirements, a lot of wasted work may go into constructing the Data Warehouse’s Data Model.

2.6 Enterprise Data Warehouse Principles

The primary objective of the EDW is to create an integrated and standardized enterprise data foundation which facilitates improved analytics and reporting, leading to better decision making and problem solving capabilities. It must be flexible to enable knowledge workers to ask new questions and ponder new and different approaches to address new needs and requirements, thereby uncovering new customer needs and changing business dynamics. The underlying information infrastructure provides for:

- Data acquisition: to implement a holistic integrated and consistent view of the enterprise’s data.
- Data dissemination: the integrated data foundation implemented by the data acquisition above provisions for directing intuitive business users’ access tools and technologies, making data available to the business in a timely and cost-effective manner.

To achieve the objective for building the EDW requires that the data contained in it to be, not only of high quality but also reliably and consistently interpreted by Business users. The guiding principles in this section attempt to outline those characteristics which should be incorporated into the EDW in order to achieve that vision. It highlights the importance of common standards and practices in the development of the various components of the EDW as well as some of the awaiting gaps and pitfalls.

There is another objective of an EDW which has been largely adopted by the majority of corporations. This objective is that the EDW provides a framework and environment for the capture, retention and reporting of operational transactional data of the corporation. This implies that rather than each application being individually responsible for the archival of its data, the EDW would, in addition to providing for analytics and reporting of this data, also provide a "warehousing service" for those applications with all of the controls, security and retrieval capabilities necessary to meet the strategic, tactical and operational business requirements. The EDW will satisfy the audit and governmental responsibilities of the corporation for the retention of this data.

The Mission of the Data Warehouse is to an integrated, consistently defined and timely data to improve the effectiveness, efficiency of business operations.

The mission of the information management team supporting the warehouse environment is “ensure operational excellence in the management of the organization’s Information Assets. Maximize the value, usefulness, accessibility and security of Information. Efficiently architect, build and support information solutions.

The EDW is an integrated data foundation offered as a service by Information Technology groups to all business units within corporations. All objects implemented within the EDW (i.e. architecture, models, programs, databases, software) must adhere to data and technical standards and be developed consistent with approved procedures.

2.6.1 Technology and Data Principles

It has been said and written at ad-nauseam that change is the only constant in business. Additionally, the number of information and knowledge workers is growing. Industry research groups such as Gartner and Forrester have written about the business needs for improved knowledge of the customer, operational insights, compliance and regulatory requirements drive an increase in the volume and variety and periodicity of data stored in an EDW. Recently, The Economist wrote on the cover page of February 27th-March 5th 2010 about "The data deluge". The article describes the quantity of data created by mankind "According to one estimate, mankind created 150 exabytes (billion gigabytes) of data in 2005. This year, it will create 1,200 exabytes. Consequently, the number and complexity of queries are on an exponential trajectory commensurate with increase in data and users. The EDW, therefore, must have the technical capabilities and characteristics which support the aforementioned increase and requirements. The success of the EDW and optimal business value of the EDW is dependent on some technical principles: extensibility, scalability, and resilience.

Business organization should have defined standards and practices for databases' construction and maintenance. The design and creation of databases within the EDW should conform to the enterprise-defined standards and practices. The population of databases within the EDW should be done using enterprise standard tools. The extract-transform-load programs should follow accepted guidelines and procedures regardless of the source of the data. All databases within the EDW should be data- modeled in compliance enterprise data standards. An operating manual for the databases within the EDW should be established. The operating manual should define such rules as: the frequency of update/refresh, availability and data retention.

The metadata for each object contained in the EDW should contain not only information about the data structures and business rules, but also information about the specific data in the database (i.e. data source, data target, data quality, summarization rules, data vintage & retention, etc.).

Any data standardization must occur prior to the placing of the data in the EDW. Direct updating of databases within EDW by any means other than the documented update procedure should be prohibited.

2.6.2 Guiding Principles

This section defines the guiding principles applied to the organization's data architecture strategy.

- Data should be captured accurately and completely at the point of contact.
- Metadata need to be integrated across the organization, so that it allows end users to communicate more effectively with IT and allow for increased efficiency in reporting processes.
- Regardless of where we store data within the organization, the data must be consistent throughout the organization. Data must have enterprise-wide integrity.

- An enterprise strategy should be developed to manage data, information and knowledge assets.
- Develop a data quality strategy that addresses the information needs of the business.
- A clear and consistent definition of data should be supported through the creation of an enterprise-wide data model.
- Corporate data standards should be implemented to eliminate redundancy and enhance data integrity.
- Data is owned by the organization and a data steward and a data custodian should be assigned to it.
- Information accessibility and security should be determined by data stewards.
- The data steward needs to clearly articulate data classification, access, data definition, rules, security and privacy.

2.6.3 Data Usage

In an organization, data is gathered, exchanged and shared. It produces analytical information, which is managed to produce knowledge. According to literature from industry research groups, leading organizations acknowledge, support and fully leverage their data assets [64]. Data assets are grouped into 3 types, based on their purpose:

- *Data*: which supports business processes; it is system/process relevant.
- *Information*: which supports the analysis, reporting and decision-making; information is created by aggregating and summarizing data. It has a common, user understandable definition.

- *Knowledge*: which provides the decision support and learning/discovery; knowledge is created by synthesizing and categorizing information. It is self-describing. (i.e. Business Intelligence Data)

In any organization, data has to be assessed and analyzed based on its usage. Technical infrastructure supports data retention for future business needs. As the usage changes, the architecture should be reviewed and possibly revised.

2.6.4 Data Quality

Information is an important asset that everyone in the organization has responsibility to maintain and improve. The data steward should provide guidance and leadership to individuals creating and maintaining data. The data Steward should publish data quality guidelines and mandate focusing on:

- *Data Accuracy*: Individuals who enter, update or delete data are responsible for quality and accuracy of the data.
- *Data Consistency*: Same data should have consistent definition throughout the organization. For improved data quality, an organization needs to:
- Consider other users of the data and the business value of that information to the organization as a whole.
- Have the information supplier (e.g. the customer) validate the information.
- Use data items from information systems in the manner for which they were intended. For example, comment fields should not be used for storing codes. The system should be changed to match business needs.
- Measure the quality of data and establish programs to improve that quality.

- Enhance the Data Stewardship function to monitor and improve data quality.

The resulting support for data quality will enhance the accuracy, completeness and timeliness of data. This enhanced data quality is a critical building block for future initiatives. A robust data governance foundation can complement other initiatives, such as simplifying ongoing conversions to new applications, extending the life-span of key legacy applications, effective business partnerships and enhancing shared service support.

2.7 Data Modeling Guidelines and Assumptions – Enterprise Data Warehouse

The following are guidelines and assumptions for designing, modeling and implementing enterprise Data Warehouse environment:

- The Data Warehouse Environment is a set of all Integrated Data Marts (IDM). The IDMs should contain data which has been approved as the authoritative source and official corporate record.
- The design, creation and population of IDMs should be managed by a Data Warehouse management function. The creation of IDM should be done using standard Data Warehouse development and management tools, technologies and processes.
- All Integrated Data Marts (IDMs) should be architected and modeled by the DWE management function and adhere to corporate standards with respect to naming conventions, recommended usage and sharing. The only exception to that rule would be in the implementation of a proprietary 3rd party package.
- A data model per each IDM should be spawned from the Enterprise Data Model. A single Enterprise Data Model should include all IDMs although

rules of "de-coupling" should be employed to ensure the operational independence of the various entity types within the model.

2.7.1 Data Granularity, Summarization and Archival Guidelines

There are some important specific steps which should be followed to effectively design the right levels of granularity. An outline of each is described as follows:

2.7.1.1 Determine Raw Estimates

According to common database construction practices, the starting point for determining the appropriate levels of granularity is to do a raw estimate of the number of rows of data and the DASD that will be required in the warehouse [64]. It is sufficient for this estimate, at this stage, to be only an order of magnitude.

There is an algorithmic path to calculating the space occupied by the information warehouse. These steps include:

- Identify all the tables that will be built,
- Estimate the size of the row in each table. It is likely that the exact size will not be known. Lower-bound and upper-bound estimates are sufficient.
- On the one year-year horizon, estimate the maximum number of rows and minimum number of rows in each warehouse table. Developing this estimate can be quite challenging and usually give the greatest difficulty. Good judgment will have to be used. For example:
- If the table is for customers, use today's estimate of customers, factoring in business conditions and the corporate business plan,

- If there is no existing business today, estimate the total market,
- If the market share is unpredictable, use an estimate of what the computer has achieved,

In short, start with a reasonable estimate of customers gathered from more one or more sources. Once the estimate is completed for the one-year horizon, repeat the process for the five-year horizon.

2.7.1.2 Determine what levels of Granularity are needed

Once the estimates are made, the next step is to determine exactly what level of granularity is to be. The starting point is common sense and a certain amount of intuition.

The first step necessary is to determine whether dual or singular levels of granularity are needed. This is dependent on the total number of rows that have been estimated to be in the data warehouse environment. Bill Inmon "Data Warehouse and Design page 108", includes a chart which shows the relationship between the total number of estimated rows and the granularity required, for data in both the one year and five year span.

Briefly this chart shows that, on the one-year horizon, if there will be less than 10,000 rows, then practically any design and implementation of the warehouse database environment will work or if there will be more than 1,000,000 rows then dual levels of granularity will be called for. For data in the five year horizon, the totals shift by an order of magnitude of about ten (i.e. dual levels of granularity being required if total rows estimates are greater than 10,000,000).

Creating a lightly summarized level of data that is at a very low level of detail doesn't make sense because too many resources will be required to process the data. Alternatively, creating a lightly summarized level of data that is too high in detail means that too much analysis should be done at the true archival level (i.e., the detailed level of data that has been stored in archive). So the first cut at the lightly summarized level of granularity is to be made as an educated guess.

But an educated guess is only the starting point to refine the guess; a certain amount of iterative analysis is needed. The only real way to do this is to put the data in front of the end user and use the feedback loop techniques suggested below. It is only after the end user has actually seen the data that a definitive answer can be given.

2.7.1.3 Implement levels of Granularity - Feedback Loop Techniques

The following are mechanisms for applying the feedback techniques when working with end-users for establishing levels of granularity:

- Build the first parts of the data warehouse in very small, very fast steps, and carefully listen to the end users' comments. Be prepared to make adjustments quickly.
- Use prototyping if such a tool is available, and allow the feedback loop to function using observations from the prototype.
- Go through the feedback process with an experienced user who is aware of the process that is occurring.
- There are many ways that the data going into the warehouse can be summarized based on the granularity established. Some these include:

- Average or otherwise calculate data as it goes into the target
- Push highest/lowest set values into the target;
- Push only data that is obviously needed into the target;
- Use conditional logic to select only a subset of records to go into the target.

There are no limits as to how data may be lightly summarized. The key is to address the business opportunities creatively.

There is one important point. In classical requirements systems development, it is unwise to proceed until the vast majority of the requirements are identified. But in building the data warehouse, it is unwise not to proceed after at least half of the requirements for the warehouse data are identified. In other words, if in building the warehouse the developer waits until many requirements are identified, then the warehouse will never be built. It is vitally important that the feedback loop with the end-users be initiated as soon as possible.

2.7.1.4 Determine Archival Requirements

In true archival level of data, every detailed record of data is stored. This archived data is stored on a medium suited to the bulk management of data. Note that not all fields of data are transported to the true archival level. Only those fields of data needed for legal reasons and informational are stored. Operational data that has further use, even in an archival mode, is purged from the operational system as the data is passed to the archival level.

Archived data can be held in a single medium, such as magnetic tape which is inexpensive for storage and slower for access. However, it is also possible to store portions of the archived data on-line, when there is a probability that the data be needed more regularly.

The determination of which informational data is to be archived Archival requirement should be jointly determined with the end-users on an iterative basis, allowing their feedback to help establish which data is not so frequently required and can be stored on a low-cost medium.

2.7.2 Enterprise Data Warehouse - Data Quality, Data Ownership and Data Sharing

During the initial design, implementation and in the on-going management of the data warehouse environment important consideration must given to data quality, data ownership and data sharing.

Data quality includes the implementation and follows through of procedures and standards to ensure that all the data in the data warehouse is maintained at the highest quality standards required by the company.

Responsibility for Data Ownership must be established, accepted and controlled. This requires that each of the data marts have an effective owner who is responsible for the integrity and quality of the data within that store.

Effective Data Sharing mechanisms, procedures and controls must be implemented enabling business end-users to easily access and/or analyse any of the data stored in the warehouse for meeting their needs.

The following is a brief outline of some of the important data quality, ownership and sharing guidelines for the information warehouse. These will be expanded in more detail in subsequent phases:

- All system project teams and end-user staff involved in the design, construction and implementation of Data Warehouse Environment must ensure that the appropriate data quality management procedures have been followed. Briefly these include:
- Adequate definition and documentation to standard of business needs;
- Ensure all data requirements are defined and documented to standard;
- Ensure all roles and responsibilities for maintaining data quality are adequately filled;
- Ensure all data modeling and data base designs comply to established data standards;
- Ensure Metadata is prepared and captured to established standards;
- Data Stewards will be identified and requested to certify the integrity and quality of their data which is being maintained in Data Warehouse Environment and is therefore made available for sharing amongst all end-users who may have a need for it.

2.7.3 Updating the Data Warehouse Environment

The frequency of updating data in the Enterprise Data Warehouse will be far less than in the operational data stores. One of the main issues that govern this is the "cyclicity" of the data being passed into the Enterprise Data Warehouse.

Cyclicity of data refers to the length of time a change of data in the operational environment takes to be reflected in the warehouse. When a change of any particular data item comes into the operational environment the change is reflected immediately in the data base. Once this occurs, the change also needs to be reflected in the related constructs in the Enterprise Data Warehouse.

2.7.4 Data Naming and Definition

Information warehouse - Entity Type Standards- This section defines the standards for naming entity types in the Data Warehouse Environment.

Entity Type names in the Data Warehouse Environment should be as descriptive as possible, and have as much meaning as possible to the end users. However, Entity Types which are brought over directly and completely from the Operational Environment and are direct copies of entity types there should have the same name as in the Operational Environment. In general, Entity Type names should be first proposed by the development team, approved by data modeling group and then endorsed by the end users.

2.7.5 Data Warehouse Environment - Metadata Structure Guidelines

The following is a high-level description of the important guidelines for structuring the metadata.

- The structure of the data warehouse metadata must allow for easy retrieval and access to all business end-users in a manner which assumes them to have a low degree of computer literacy;
- A high level catalogue of all available data entities in the Enterprise Data Warehouse must be available with easy indexing to further details of each of the entities if and when required;

- Clear easy to understand names, definition and descriptions, in business terms, of all available data entities must be made available. Good examples should also be available on their meaning and how the data could be made available in a shared environment. The business data steward of the data should also be clearly identified and kept up-to-date if any changes occur;
- Complete details on the data's system of record, the transformation details, granularity levels, summarization rules and its structural history, as described above, must be maintained in a clear easy to access and understand manner.
- A metadata repository should be created for all data in the Enterprise Data Warehouse which will contain all the above information.

2.8 Data Sharing

Data is a critical and extremely valuable business asset. Sharing data among many business domains and functions improves its value and reduces its cost to the enterprise. As well, sharing ensures that data is used consistently across the business lines and functions and improves agility in responding to business events.

In many, if not most organizations, there are several factors that prohibit or make it very difficult to share information. Data is most often not documented comprehensively or the documentation is not maintained to reflect accurate and current state. There are often also issues of data latency differences and referential inconsistency. Data is also often fragmented and disbursed across a number of data structures and database technology platforms. Security is a major issue when it comes to sharing;

however it is not covered within the scope of this dissertation .While in the broad context, the concept of sharing data sounds rather simple, there are a number of data attributes that will factor into determining if the data can be effectively shared and re-used.

2.8.1 Prerequisites for reusability and sharing

There are two major issues that are at the heart of making data fully shareable. The first is producing rich and full documentation about the data and the business processes involved n the creation of the data. Full documentation enables effective resource discovery (i.e., catalogues) of distributed data sources and enables more informed re-use. The second challenge for sharing data is that of exposing data in the most flexible way possible so as to enable multiple methods of accessibility and innovative uses by various business groups.

Both challenges require that:

- data are collected to a high standard using appropriate sampling strategies, rigorous data gathering methods and, where appropriate, systematic interview transcription
- The business context of the data collection is captured. The biggest challenge in providing accurate data is in the analysis and understanding of the available data. Within the existing applications systems, data is often extracted from source, transformed and stored in another source, extracted again and transformed yet again and stored in yet another source. Data is often not properly documented and neither is the business rules used for the

transformations or the computations. Determining authoritativeness of data presents a challenge.

- The richness of the structure and features of data and are made available
- The interrelationships between data are made available. Authoritative data from most data sources is typically associated with a particular product or events that are tied to particular product or service offerings. More often than not the information needs of business solutions require data to be linked from a number of data sources. Linkage of this data is often not possible in a time and cost effective manner unless the data from the multiple sources is brought to a common location and designed to facilitate the linkage.
- Data are disseminated in sensitive ways that satisfy the ethical and legal requirements to which they are bound.
- Data are represented in business contextual ways to ensure there is clarity on the definition of the data and appropriate use of it.
- Since most applications are developed without the benefit of a standard Information Model, similar data is often represented differently in different systems with different formats and different data values. To make effective use of this data, a level of data transformation needs to be performed.

2.8.2 Sharing Levels

While the goal is to maximize the sharing of data it is important to identify exactly what level of sharing is required for any individual piece of data. All data created in the corporation is corporate data. However, not everyone needs to have every single piece of data in the organization at his/her fingertips. The storage and management of data should vary based on how widely the data is shared. Although there are many gradations of sharing, data is divided into four categories:

- Personal
- Business Unit shared
- Enterprise shared
- Externally shared

2.8.2.1 Personal

Personal data is information collected and used by one person. This data is usually stored in an individual's database/tables. Example of structures containing personal data are that person's tables containing sales and commission data.

2.8.2.2 Business Unit Shared

Departmental or business unit data is used by one business domain group. This type of data is specifically designed, constructed and maintained to meet the requirements of that business unit. Business unit data stores are known in the industry as data marts. Examples of data marts that are business unit centric are sales and marketing data mart.

2.8.2.3 Enterprise Shared

Enterprise-wide data is used by more than one business domain. This type of data is generally stored in the Enterprise Data Warehouse and centrally managed by the IT department on behalf of the enterprise. Examples of databases containing enterprise-shared data are Customer Information database and customer risk indicator. A multi-business-units organization requires ONE customer identification database.

2.8.2.4 Externally Shared

Externally-shared data is acquired from, or provided to, individuals or organizations outside the corporation. Examples of such data are regulatory and compliance reports to the government or data acquired from publicly available data providers such as Reuters, and Credit Bureaux.

2.8.3 Classifying Data

One of the challenges of data management is to ensure that enterprise data is designed, constructed, secured and managed consistent with the appropriate level of sharing. Often data may be locally shared and managed in a way that does not provide adequate security and documentation for the level of sharing. Business-Unit-shared data may actually have the potential to be enterprise-wide-shared, but is not recognized as such and is not available to others who could use it.

Data may also move between the various levels of sharing during its life. Data records may be collected at a business unit level and shared there for a period of time before some of the data is transferred to an enterprise-shared store.

Analysis and design of any portion of the data environment must take into account all potential key users of the environment.

In order for sharing to take place, the data environment itself must be designed with all users in mind. People generally want to share data; they do not want to incur the cost of developing and maintaining a separate database if data already exists. However, if the existing data is not defined as they require or is not organized in a way that meets their needs, they will go into the effort and cost to build their own database/data mart. This duplication of effort can be avoided by using the integrated data view to analyze data needs and by using analysis techniques, such as data modeling that identify relationships between data based on business functions rather than application needs.

Within the enterprise, awareness must be maintained of external standards for the creation of data or data structures.

External sharing of data, primarily by electronic means, can provide great savings to the enterprise. In order for this sharing to take place, the data must meet requirements set on it by bodies outside of the corporation. It is vital, therefore, that those involved in analyzing and designing the data environment be aware of, and participate in, setting industry standards for data and structures.

The single best source for all data will be identified and used in the capture of data. *Best* is defined to be the source closest to the real world event that the data describes. Once captured, the data will have one designated database that will be considered the authoritative single version of the truth source to be used by everyone in the enterprise.

It is widely accepted that there are two prevailing issues in data environments in most organization; data fragmentation and duplication, and inconsistent representations of the same data. A single type of data can generally be found in a number of different data stores. Each creation point adds another place where errors can occur. To avoid this duplication and potential for error, one set of data must be established for the enterprise that can then be shared by all users. To eliminate possible errors in interpretation and timing, the data must be captured as close as possible to the moment of creation in the real world.

In order for sharing to work, the users of the data must understand exactly what the data is. All data in the enterprise is created as the result of the execution of a business practice. The level of quality required for the data is another set of business practices. An understanding of the business context of data is vital to its proper use. Wide accessibility opens up the potential for misuse if data is used indiscriminately.

The interface between the data structures and the applications that use them must be built in such a way that changes to one will not impact the other.

To make sharing a practical reality, it must be possible to design, construct and maintain the data separate from the functions and applications they support.

2.8.4 Accessing Sharable Data

There are a variety of strategies to be considered for enabling access to sharable data that exists within an organization [64]:

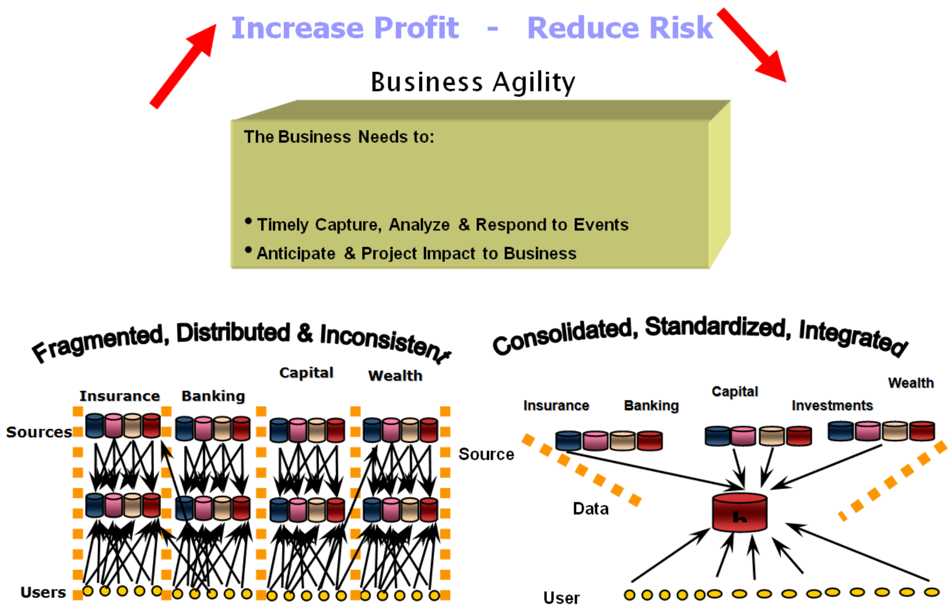

Figure 2.4 The need for consolidated, standardized and integrated environment

Centralized Data Warehouse:

Consolidated, standardized and integrated data warehouse solutions do provide a more robust, scalable and extensible platform for rendering data to be reusable and sharable (see Figure 2.4). It enables a solid data foundation which is fundamental to effective sharing and re-use.

2.8.4.1 Enterprise Information Integration:

Enterprise Information Integration (EII) is a metadata driven approach to accessing data that is fragmented and distributed across a number of operational systems of record databases. Since this solution doesn't require the procurement of database software and hardware (disk storage) or the construction of physical databases, it is often considered as a more cost-effective option. The application of EII solutions may be constrained and limited to very specific types of applications. Consideration for consistent data types and structures as well physical data design and latency factors contribute to the limitations and constraints.

2.8.4.2 Operational Data Stores (ODS):

The ODS concept is very similar to that of the Centralized Data the ODS warehouse with the main differences being scale, scope of business focus and the handling of historical data. Its application is therefore limited to a certain profile of applications.

2.8.5 Shared Data Governance

As noted previously, one of the critical pre-requisites for data to be rendered as sharable and re-usable, is that the documentation of the data is required to be rich, complete, accurate and have context that is easily consumable and its use and application easily understandable by business users. The role of the Business Data Steward is critical to this object. Further information on this is provided in the section on Data Governance.

Change Management is very significant and critical for the proper administration of sharable data. As more and more business users consume sharable data, it is vital that the administrators understand who these users are and how they are using this data. The establishment of effective metadata management becomes very important. The metadata management will be required to perform appropriate impact analysis when changes need to be made to the data, its structure, definition, etc.

2.8.6 Benefits of Sharable Data

Aside from the obvious benefits derived from re-using sharable data related to cost effectiveness and data consistency, one can also consider implications where data does not have the attributes required for effective sharing.

In the absence of understanding what data is available for reuse and sharing, each business group spends considerable time and effort analyzing various sources of data to establish the correct data that they need. Business agility is often compromised as a result.

Reference data such as Customer Information Files also known as (CIF) exist in several formats and technology platforms within a single enterprise that is comprised of multiple business units. This issue is further complicated by the inconsistent number of fields and data types in each CIF. Business initiatives such as Customer Relationship Management (CRM) require combining data from multiple business units. Often the efforts extended to profile and analyze the CIF data is repeated time and again for each business unit and or for each initiative [65]. The resulting metadata resulting from this analysis also most often not documented or shared. This translates into considerable unnecessary costs to the organization.

Given that we're in the age where regulatory governance and compliance is becoming more and more critical to all businesses, the collection, standardization, integration of data required for these compliance objectives can also be reused for the Customer Relationship Management (CRM), Enterprise Risk Management (ERM), financial management, accounting management, fraud, Anti Money Laundering (AML), etc. In the example outlined in Figure 2.5 the chart below, it can be easily seen the value to an organization that invests in the development and implementation of strategies, standards and infrastructure to capitalize on the value to be derived from the re-use of sharable data.

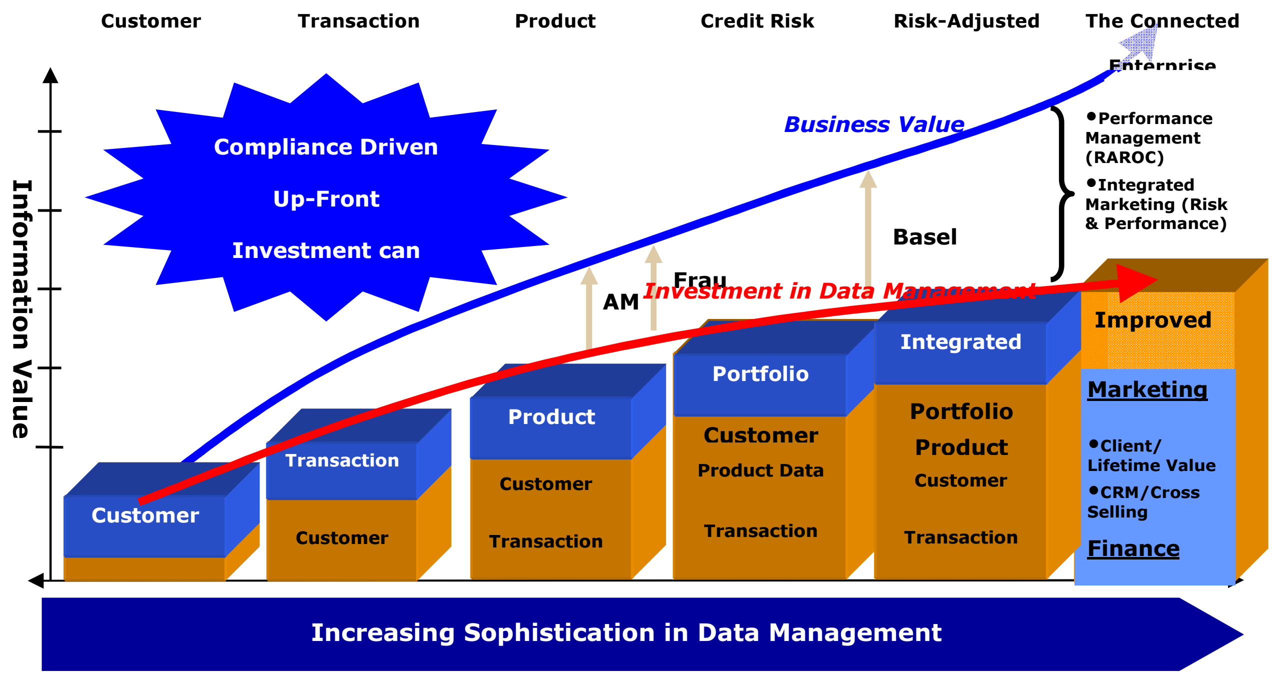

Figure 2.5 Importance of data re-use in building the infrastructure

2.9 Concluding Remarks

The development of robust enterprise data warehouse architecture is necessary to eliminate redundancy and to avoid possible inconsistency in the data stores of an organization. A data warehouse provides for better issuing of integrated reports that otherwise might require combining data spanning different operational applications within the organization. A successful data warehouse architecture and design should consider and incorporate different principles and rules to those which are followed in constructing operational applications.

Chapter Three: **METADATA MANAGEMENT**

3.1 Introduction

Metadata is defined as ‘data about data’. Metadata describes how and when and by whom a particular set of data was collected, and encompasses information about the meaning, structure, movement, change and quality of data as held within the various repositories within an organization. Metadata is essential for understanding information stored in an organization’s databases.

While the importance of metadata is often difficult to explain to the various stakeholders of any project, it is becoming increasingly critical. Metadata is the recipe for constructing databases, data models, programs, reports, queries and data movement. In fact, the ecosystem of enterprise information systems is built using metadata. Thus, the management of this environment is extremely important to ensure the foundation of the enterprise is maintained.

Metadata management is crucial for building data intensive applications such as Enterprise Data Warehouses. Values do always exist as raw data; they start to have meaning when organized according to some metadata... The name of a data element, where the data values are kept, what system creates the data, what systems use the data; all these are vital facts required for the management of data. This knowledge or data about the data is called metadata.

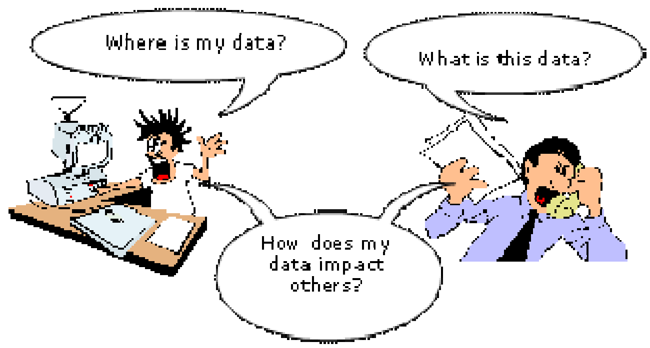

Figure 3.1 Users and Data

To provide the basis for a corporate shared environment, metadata must go beyond just describing the physical location of the data. It must include the rules that are used to create and measure the quality of the data, such as definitions, retention, source and timeliness requirements for the data.

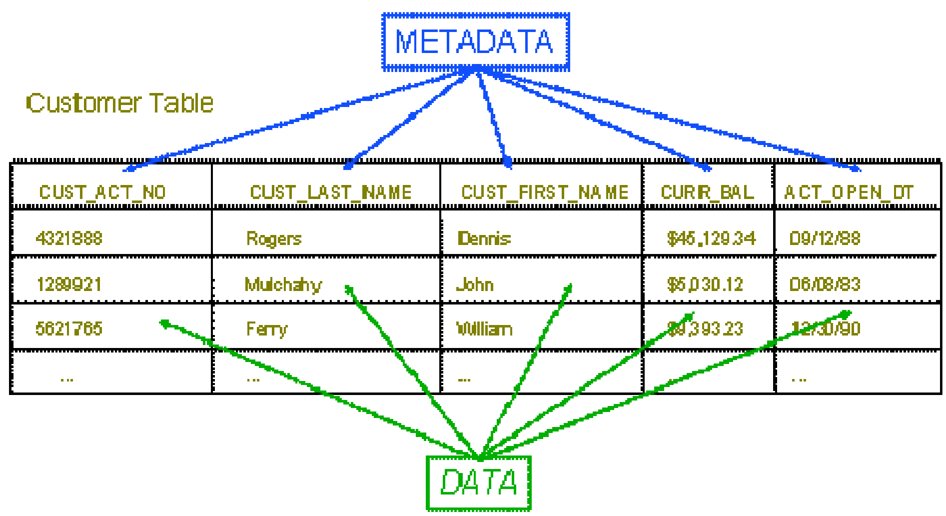

Figure 3.2 Data versus metadata

Metadata administration fits with data management in two ways: it supports the management of data in the same way that data management supports the enterprise. Just as enterprise data provides the necessary information to support the management of all aspects of our business, metadata provides the necessary information to support data management. Also, metadata itself is data that supports part of the business. As such, it must be managed using the same principles and practices used for all data management. Performing data management activities for metadata is another way of defining metadata administration.

A robust metadata strategy is a must to support the enterprise data management defined in this thesis. This strategy has several requirements that must be accommodated to manage the metadata environment in order for it to be successful in delivering a truly useful metadata.

3.2 Metadata Environment

A metadata environment includes:

3.2.1 Business Metadata

Business metadata can be classified into two categories: business definitions and business rules.

3.2.1.1 Business Definitions

The primary goal of business definition metadata is to record the context of usage for various business definitions (some of which may have the same name and mean different things). This is also useful in ensuring the consistent definitions within various parts of the organization, minimize variations in meaning, and validate the conformance of information requirements with business objectives. Metadata is also critical for understanding and communication of the business definitions and meaning that are used to develop information management systems that are consistent with business goals.

In an enterprise data warehouse environment, aggregating information for analysis and reporting has to deploy ways to translate and align the different definitions from one business domain to another to provide a coherent overall picture of the organization’s operations. This process is typically complex and time-consuming. Documenting and standardizing business metadata makes the above process easier and less time consuming.

3.2.1.2 Business Rules

In addition to the definition of the data itself, the transformation and calculation rules that are used to manipulate this data are also considered metadata. The typical complexity of business rules makes the problem of maintaining and versioning these rules a significant challenge.

Business rules are very important as they have a dramatic affect on derived data and how the data is used in calculations and aggregations. Significant time and effort are required by typical systems projects in clarifying the business definitions. These clarifications sometimes result in the creation of new business definitions and business rules.

3.3 Technical Metadata

Data management takes place throughout the life cycle of the data. To support this, metadata is collected and managed during the life cycle. In the initial planning and design phase of the data life cycle, the metadata collected concerns the conceptual or logical representation (logical metadata) of the data. At this stage there are no actual data values being described. Instead, we are documenting a plan or model for the data we will need to represent the business world. In the construction, implementation and maintenance phases of the life cycle, the logical model is translated into actual file structures (physical metadata) that contain data values as demonstrated by the example below:

Business Name: CUSTOMER IDENTIFICATION

Description: *"The number sequentially assigned by Accounts Receivable which uniquely identifies all of our customers including wholesale and retail across all internal business units."*

Preferred Names:
COBOL: CUST-ID
DB2: CUST_ID
Assembler: CUSTID

Headers:
Screen: Cust ID
Report: Customer

Domain: Identifier
Datatype/Length: Char 9

3.3.1 Logical Metadata

Just as a design blueprint for a facility is a logical or ideal representation of a facility, a data model is a logical representation of data that is required in the business. This model may not ever appear exactly as drawn in the real world, but it plays a valuable role in allowing people to visualize the data and the relationships between data.

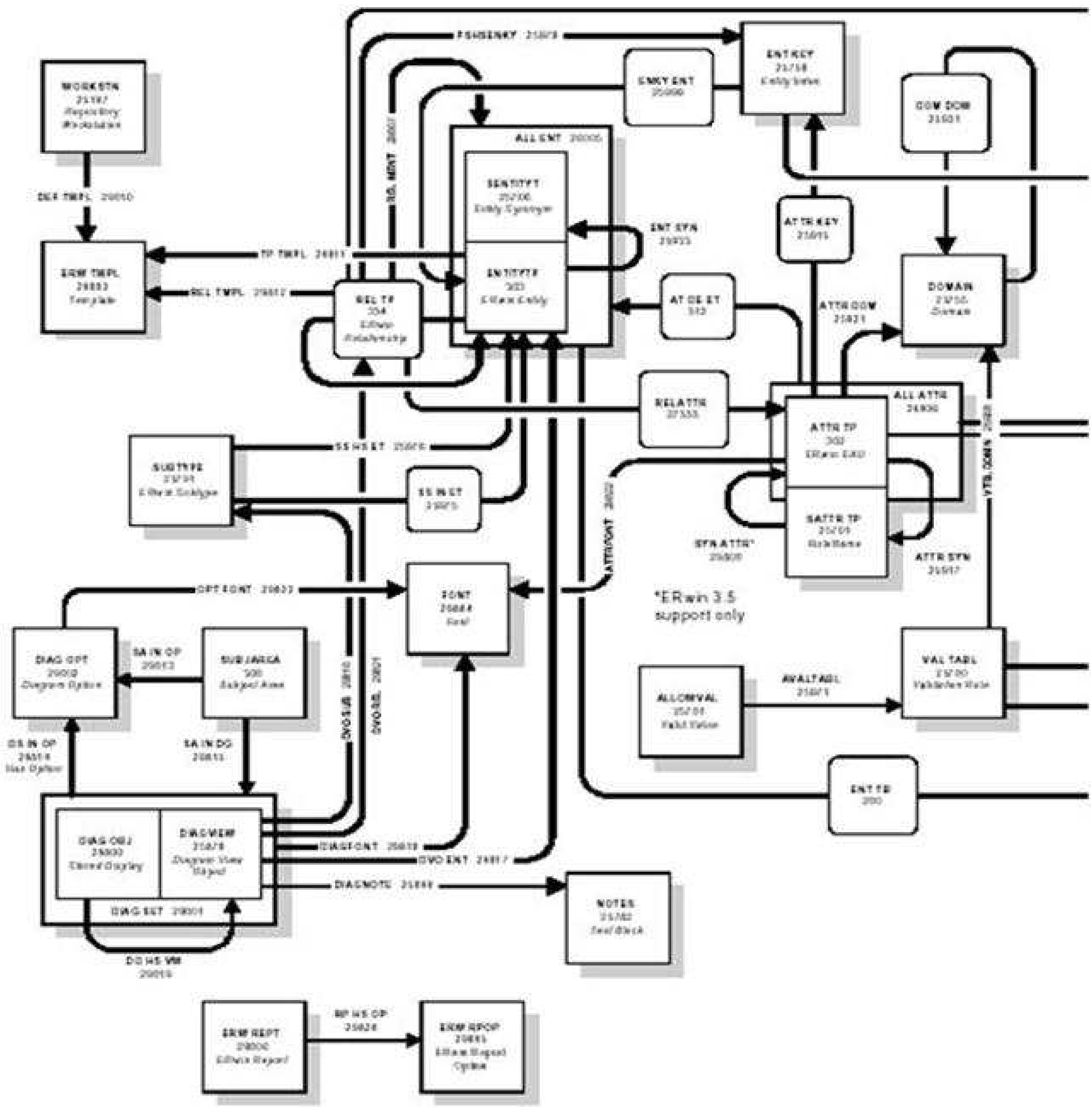

Figure 3.3 Example Entity-Relationship Diagram

When we are talking about the logical representation of data, there are three major metadata objects:

Entity type, which represents an object or event in the real world (e.g., a customer, an account)

Attribute, which is an individual fact describing some aspect of the entity (e.g., the name of a customer, the account number)

Relationship, which is a description of how data entity types relate to one another (e.g., an account number in the account entity belongs to a customer number in the customer entity). Shown in Figure 3.3 is an example of a logical data model a.k.a Entity Relationship Diagram (ERD):

3.3.2 Physical Metadata

During the construction, implementation and maintenance phases of the data life cycle, these logical representations are translated into physical files. The metadata collected here now represent real occurrences of data values. This physical metadata can be thought of as an inventory of the data resources of the corporation. Here are some examples of physical metadata objects:

- File: which is a grouping of data that contains data for one entity type, but it may contain data for two or more entities (may appear in different technologies, e.g., a DB2 table or an ADABAS file).
- Element: which is the place in the file where the values for an individual data element are stores. It usually contains data for just one attribute.

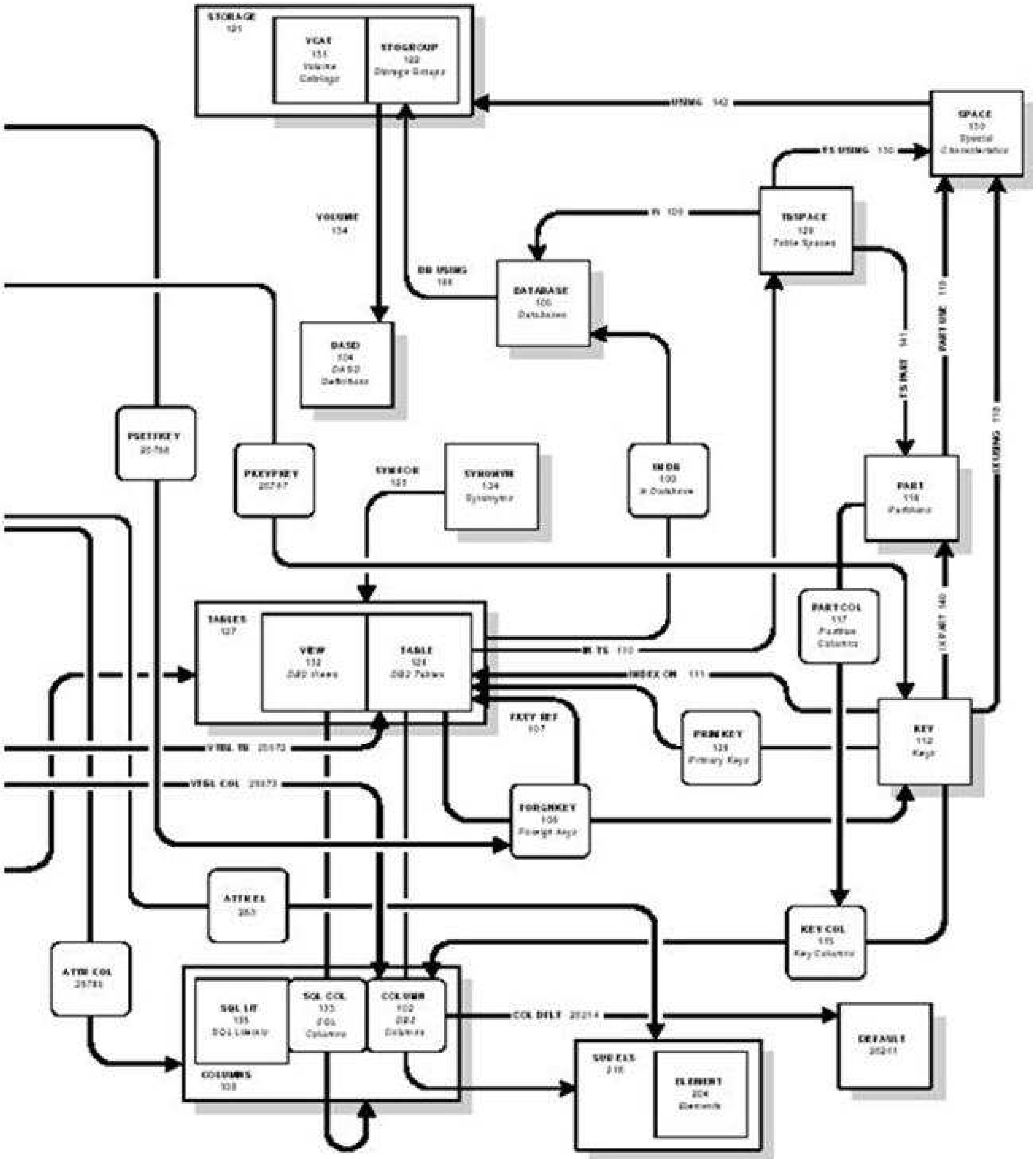

Figure 3.4 Example metadata objects and the metadata associated with them

- Foreign key: which is a special type of element that contains data values that relate a row in one file to row(s) in another file? This is the physical representation of a relationship.

It is important to maintain a connection between the logical and physical phase of the data so that we can produce a complete picture of the full life cycle of the data. Figure 3.4*** shows an example of different metadata objects and the metadata associated with them.

3.3.3 Domains

Domains are the concept that ties the logical and physical representations of data together. A domain is a description of a unique set of data values. Each domain has specific rules for the format and content of the data it contains. An example of a domain would be ACCOUNT. By defining the domain of Account Status to be only data that falls within a certain range of values (O for Open, C for close, D for Dormant) and is in a certain formats char (2), we can easily determine if a data value is an account status or not. Account status C an allowable value of the account status domain and Char 2 is the acceptable type and size. A domain is made up of one or more of the following:

A range of allowable values

A format

A type

3.3.4 Domains and Data Integration

The consistent and controlled use of domains is a very powerful tool in building and maintaining an integrated data environment that promotes the shareability of data.

The principle of inheritance of rules from one level of the domain hierarchy to another ensures that data of the same type is treated the same way in every physical implementation. This allows data sharing among different users and greatly increases the ability of the user to understand and apply data consistently. The domain hierarchy itself becomes a data vocabulary for the corporation and can be used by directory tools as a means for the user to understand and find data.

It allows a user to combine and compare the data from the two elements

In the future, if systems XYZ and ABC are combined, the data for these two elements can be easily integrated into one element

A user looking for all information in an organization on PRODUCTS can use the relationships in the domain hierarchy to locate all the elements that contain data of this type

3.3.5 Metadata Objects

The most basic type of metadata describes the data element. Data element metadata contains information about a type of data value. The technical metadata below defines the data requirements for capturing customer data:

Customer Identifier	CUST-ID	PIC(12)
Customer First Name	CUST-Fname	CHAR(20)
Customer Last Name	CUST-Lname	CHAR(20)

In addition to this, we also keep metadata that describes how the data elements are grouped together to describe business objects, how the data for these objects are related to one another and how they are physically stored and accessed.

3.3.6 Application Systems Interfaces

This information is important to metadata. Data regarding how systems are connected is a critical area of information but typically difficult to procure and manage. In a large organization, there are hundreds or even thousands of individual components that interact with each other in complex ways. Data Exchange

The actual information that is exchanged between systems is also important and related to connectivity. Data interchange between systems can be complex, with vast amounts of disparate data being interchanged. Having information on this exchange is critical to tracing data back from a target system to its source.

3.4 Metadata Management Methodology

Metadata management requires the adoption of life cycle methodology that supports it. A metadata methodology has to be determined and formally adopted as a key component of Enterprise Data Management and the construction of the Enterprise Data Warehouse. A methodology is composed of well-defined set of principles an, guidelines and best practices described below:

3.4.1 Metadata Principles

- There are several principles to keep in mind when dealing with metadata:
- Access to metadata should be provided to business and technical staff in a timely, consistent and meaningful manner.

- Metadata should be created by individuals in the project teams that build the applications and its data.
- The business owner of applications and its data should validate Metadata.
- Metadata should be kept current and accurate throughout the lifecycle of the application and its data.
- Metadata should be acquired and consolidated it into a single metadata repository which is important for maintaining consistency

3.4.2 Metadata Standards

- Any organization that strives to provide the highest level of value-add for its metadata, need to engender and deploy metadata management standards to:
- Provide rules for abbreviation standards
- Provide procedures for creating definitions
- Provide consistent naming conventions
- Analyze/document required metadata attributes
- Determine business practices for metadata verification
- Determine responsibilities for all parties involved with metadata during its life cycle.

3.4.3 Metadata Quality

- The metadata management process should plan for the deployment of quality measures throughout the lifecycle of the metadata management to:
- Establish consistent data definitions in partnership with business users
- Coordinate and communicate changes in data definitions.

- Provide business stewards with the ability to review and modify data definitions
- Utilize established data standards and guidelines in establishing data
- Provide project consulting
- Provide specifications for building data dictionary
- Implement training sessions for metadata and the dictionary

3.5 Metadata Architecture

The strategic metadata architecture captures process, data, system, and report metadata using one centralized repository which is fed from several sources:

The process design tool captures process metadata including, enterprise, business area, business capabilities and user(s).

The data-modeling tool captures both the logical data model and physical data model:

The logical data model produces the business entity, relation, and attribute.

The physical data model produces the database, table, and column name.

System analysts create System Metadata Files that contain information about systems, interfaces, and processes.

The Reporting Tool is used to capture report and system names.

3.5.1 Metadata Architecture for Enterprise Data Warehouse

Extract, Transform and Load (ETL) tools are used to capture sources, targets and mapping.

The metadata repository will use a bridge/parser that can communicate with the above tools to extract and/or push metadata on an automated schedule.

The figure below (Figure 3.5) describes the role of Metadata management in implementing an enterprise data warehouse.

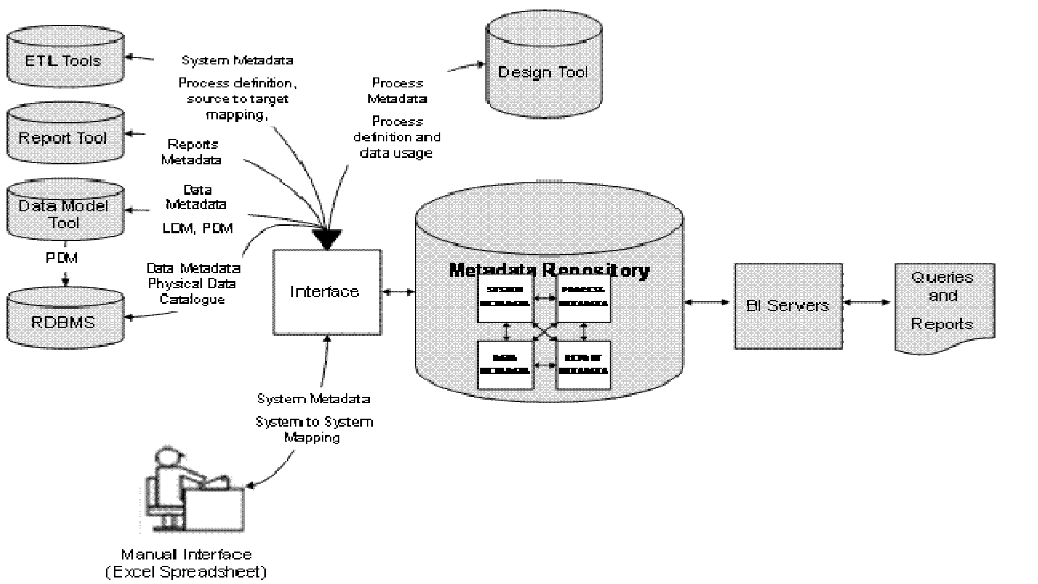

Figure 3.5 Metadata in Enterprise Data Warehouse

- The following metadata repository functionalities at a minimum to support the Enterprise Data Warehouse Requirements:
- Single Repository for all Metadata
- Source to Target Mapping (lineage)
- Documentation of Business Rules
- Documentation of Transformation Rules
- Documentation of Data Elements

3.6 Metadata Services

Metadata provisions for the following services for business users:

3.6.1 Traceability

Business users such as statistical analysts, business analysts and information workers require a facility which documents data lineage, business rule and calculation formulas. In the case where a material difference in the current month's "outstanding amount " compared to previous month is noticed, the analyst is required to investigate the cause of the variance. The metadata data captured for that process should provide the means to conduct the investigation. It should provide source to target lineage and how the data was transformed on its way to the target which could be the enterprise data warehouse.

3.6.2 Impact Analysis

Business systems applications in any organization are integrated or hemmed up together through interfaces. Assessing the impact of a change to data element in one application is important. A metadata repository can significantly aid this effort by showing the lineage of data elements from source to target. One can estimate the impact of a change to a source system data field by using the metadata store to trace this data element through to the down-stream systems and analyzing the systems and technical rules that would be affected by the change. Both business and technical users should be able to perform impact analyses independent of the metadata team by using online metadata reports.

3.6.3 Data Standardization

Enterprise wide initiatives such as Customer Relationship Management (CRM) require the consolidation of data from multiple sources, internal and external. These sources have been developed and evolved independently over many years. Standardization of data is imperative to arrive at a consistent meaning and data values.

The Accord brings with it some stringent imperatives for standardizing terms and conventions used throughout the enterprise

Consolidation of information brings the issue of metadata into sharp focus. In a financial institution, the term “Drawn Amount” might differs in meaning across business lines; this difference may invalidate the calculation of risk exposure. For example, “Drawn Amount” in one business may include fees the other business does not, while still referring to the amount in the same way.

Likewise standardizing on simple static data is critical for reporting purposes. For example most systems will use ISO codes for currency, but a few may not. The Metadata for currency codes should clearly distinguish between the two types of codes so users will have unambiguous information about the actual content in these systems.

3.6.4 Metadata Administration

Metadata administration is the collection, maintenance and dissemination of information required for the management and use of the data resources of an organization.

Metadata administration fits with data management in two ways: it supports the management of data in the same way that data management supports the enterprise. Just as enterprise data provides the necessary information to support the management of all aspects of our business, metadata provides the necessary information to support data management. Also, metadata itself is data that supports part of our business. As such, it must be managed using the same principles and practices used for all data management.

3.7 Conclusion

Metadata management is a challenging task for any organization. The requirements implied by an implementation of enterprise data warehouse take this challenge to a new level due to the imperatives of data consolidation and dissemination. Significant effort is required to capture, maintain and publish existing and new metadata to deliver on the requirements of building an architected enterprise data warehouse.

Chapter Four: **APPLICATION DEVELOPMENT: DATA WAREHOUSE FOR GIS SYSTEM**

4.1 Introduction

Geo-data sets are built for use in geographical information systems (GIS). The data is modeled to suit the needs of data entry and visual representations. They are optimized for simplicity and speed of modification. These models do not lend themselves to efficiently produce enterprise reports. Hence, Geo-data sets can be very challenging to query and analyze. In this chapter, we create a data warehouse (DW) for a geo-dataset to facilitate report generating processes. This application is intended to demonstrate the power of data warehouse as the main repository of an organization's historical data and a DW can be optimized for reporting and analysis.

The rest of this chapter is organized as follows. Section 4.1 highlights GIS as a potential application for DW development. Section 4.2 presents a study of data warehouse systems, and how they are related to GIS. Section 4.3 illustrates the architecture of the developed data warehouse and describes the entire process of creating the said warehouse. It outlines the raw data used, describes the ETL (Extract, Transform, and Load) process, and presents the physical model of the GIS data warehouse through defining its dimensions, facts, and data marts. Finally, it illustrates the reporting capabilities and defines an example query. Section 4.4 evaluates the data warehouse and discusses the advantages of our implementation. Section 4.5 presents the potential areas of improvement of the current warehouse state. Section 4.6 includes some concluding remarks.

4.2 GIS as Potential Application

Information is the most valuable asset in an organization [5]. It is very crucial for the process of decision making. Decision support systems [71, 72] are designed to allow business end-users to perform analyses on the information they have. In fact, GIS is considered to be a spatial decision support system [19]. These systems help users to study and analyze geographical problems in order to produce visual results that would help them in making better decisions. GIS is a tool that allows users to create interactive queries, analyze the spatial information, edit data, maps, and present the results of all these operations. It can be used for scientific investigations, resource and asset management, cartography, marketing, and route planning. For example, a GIS might allow emergency planners to easily calculate emergency response times in the event of a natural disaster, a GIS might be used to find wetlands that need protection from pollution, or a GIS can be used by a company to find new potential customers similar to the ones they already have and project sales due to expanding into that market.

On the other hand, GIS lacks the ability to produce high-quality descriptive reports. The way GIS data is modeled restricts users from easily creating informative reports as it is usually modeled across heterogeneous separate flat files. Moreover, GIS systems are classified as operational systems [28]. They are dedicated to let users to add, update and delete features in data layers.

Outside the realm of GIS there are many other decision support tools, especially for working with business data. More specifically, data warehousing systems have evolved to meet the growing needs of managing data efficiently and effectively for analysis and reporting for end-users.

In this chapter, we introduce a process for applying data warehousing techniques to bring this powerful support system to geographical data analysis. The main motivation for choosing to build a data warehouse is to enable users to report on tactical and strategic GIS information.

4.3 Background

Data warehousing processes are used to design and develop data repositories for efficient enterprise reporting and decision support systems. Kimball states that a data warehouse is a queryable presentation for enterprise data and that this presentation must not be based on an entity-relation model [48, 49]. Data warehouses have become a very important aspect of data management for businesses. There is no de facto standard for data warehousing techniques but the basic methods and processes outlined by Kimball [48], Chaudhuri and Dayal are an excellent place to start [14].

We have chosen to incorporate as much of the methodology and process described in *The Data Warehouse Lifecycle Toolkit* [49] because of the widespread popularity of Kimball's approach to data warehousing and the benefits that it gives.

Geographical Information Systems (GIS) allow for processing and displaying of geographical data sets with a common feature for many GIS application to process spatial data to create a visual representation. There is a long list of GIS applications to choose from with open source, commercial and proprietary offerings depending on your needs and the geographical data you intend to use.

There have been previous attempts to integrate data warehousing techniques and geographical data sets but there is currently no standard [28]. Also previous work, such as GeoDWFrame [28], focuses on the integration of spatial data into the data warehouse while other approaches separate the task into two pieces, the data warehouse for descriptive information and leave the spatial data to a separate GIS application [28]. The goals of the data warehouse in this chapter focus on the data warehousing aspect in relation to the descriptive attributes of the geographical data. The use of spatial data is open to future work and extensions of the current state of the project and further research into this area would need to be done to consider adding this functionality. We offer this approach in order to see how data warehousing techniques might benefit GIS data and end-users in querying and reporting, to see if there are benefits before attempting to create a more complex data warehouse to support the spatial data in a geographical data set.

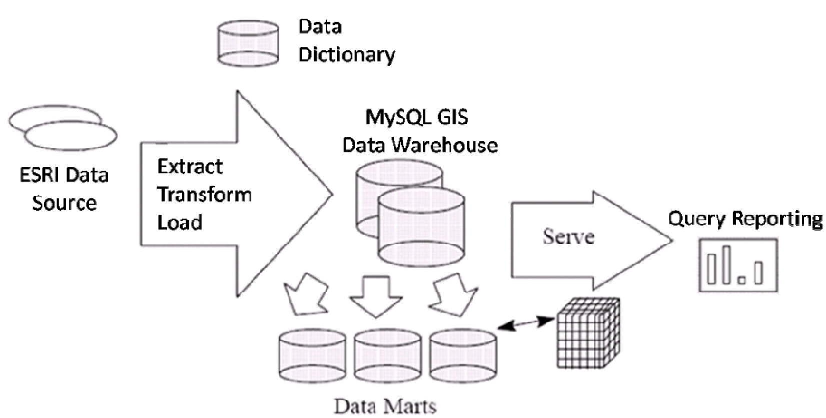

Figure 4.1 Data Warehousing Architecture

4.4 Architecture and End-to-End Process

Figure 4.1 shows the architecture we used for our GIS data warehouse. It describes the necessary high level processes of extracting, transforming and loading the source TIGER/Line data files into a functional data warehouse as well as the end goal of producing enterprise reporting capabilities. Each step in the end-to-end process is executed manually; without the use of data warehousing tools. In the following subsections we detail these manual processes.

4.4.1 ESRI Data Source

In this section, we shed light on the data we used in our data warehouse. In fact, all the datasets that are used in this chapter are acquired and manually downloaded from ESRI ArcData website: http://arcdata.esri.com/data/TIGER2000/TIGER_download.cfm. Initially, we started working with data for Alameda County, California only. In general, the acquired data can be categorized into two main groups: spatial and non-spatial data.

The spatial data is referred to as the Census 2000 TIGER/Line dataset. It comes in ESRI shapefile format. The shapefile is a file-based data model that stores geometry and attribute information for spatial features in a dataset [25]. Shapefiles support point, line, and area (polygon) features. Attributes are held in a dBASE format file. Each attribute record has a one-to-one relationship with the associated shape record. In other words, a shapefile consists of separate interrelated files as follows: a main file (*.shp), an index file (*.shx), and a dBASE table (*.dbf). For example if we have a county shapefile, we would locate three physical files on the hard disk, county.shp, county.shx, and county.dbf.

The Census 2000 TIGER/Line shapefiles were created from the Topologically Integrated Geographic Encoding and Referencing (TIGER) database of the United States Census Bureau. The shapefiles contain data about the following features [24]

- Line Features: roads, railroads, hydrography, and transportation and utility lines.
- Boundary Features: statistical (e.g., census tracts and blocks); government (e.g., places and counties); and administrative (e.g., congressional and school districts).
- Landmark Features: point (e.g., schools and churches); area (e.g., parks and cemeteries); and key geographic locations (e.g., apartment buildings and factories).

The TIGER/Line data files follow a specific naming convention to identify each data layer. Each data file name is a combination of a layer abbreviation (e.g. 'cty') and a 5-digit County FIPS Code (e.g. '06001') and the file extension (e.g. '.shp'). For example, the data layer ('tgr06001cty.shp') that contains information about Alameda County, California, contains a TIGER abbreviation prefix (i.e. 'tgr') followed by the County FIPS ('06001') and layer abbreviation (e.g. 'cty'). Please refer to Appendix A for a list of the abbreviations for each of the TIGER data layers.

It is worth to mention that in the scope of our DW, we are not interested in the shape or the geometry of the features; rather we are more concerned about their descriptive attributes. As a result, we will exclude the spatial columns during the data transformation process.

Regarding the non-spatial data, it contains census demographical information in a hierarchical sequence, each hierarchy level is separated in an individual table, see **Error! Reference source not found.**. However, for our DW's scope, we included only demographic tables that correspond to Block level since it is the finest\smallest grain in the hierarchy, and the County level because all other data layers relate to it.

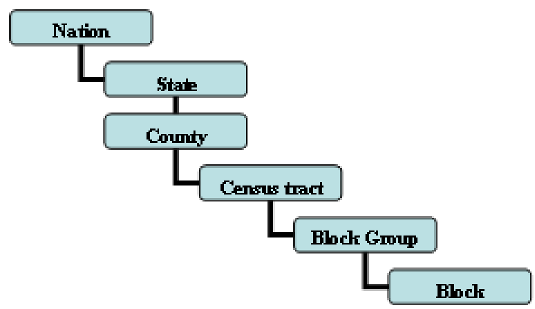

Figure 4.1 Hierarchical Relationship of Census Geographic Entities

Two types of demographical tables are available on ESRI ArcData website, Census Demographic PL94 (Public Law94-171) and Census Demographic SF1 (Summary File1). On one hand, PL94 demographic tables contain summary population counts for two universes, total population and population 18 years and over. The data were derived from basic questions asked on census questionnaires [12]. Basically, PL94 includes a count of all persons by race, a count of the population 18 years and over by race, a count of Hispanic or Latino and a count of not Hispanic or Latino by race for all persons, as well as a count of Hispanic or Latino and a count of not Hispanic or Latino by race for the population 18 years and over. On the other hand, the SF1 tables include population and housing characteristics, e.g. number of males and females in a household, for the total population, population totals for an extensive list of race (American Indian and Alaska Native tribes, Asian, and Native Hawaiian and Other Pacific Islander) and Hispanic or Latino groups [13].

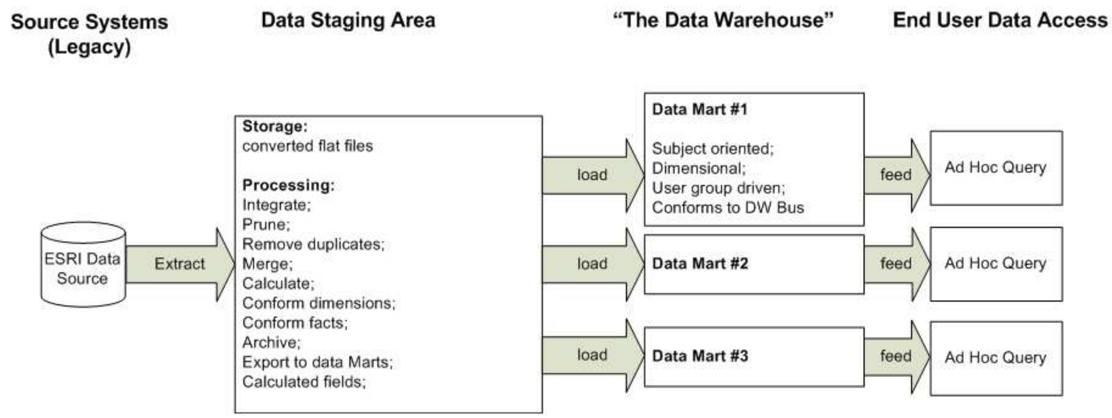

Figure 4.2 The ETL Process [49]

4.4.2 Extract Transform and Load

4.4.2.1 <u>Extract</u>

The extract process is our first step of getting data into the data warehouse environment. A typical extract process in the data warehousing industry takes about 60% of the warehouse development time, and it usually takes two months elapsed time [49]. In this project, we tried our best to extract data that we need in the scarce allocated time given. Basically, this process consists mainly of reading and understanding the source data, and copying the parts that are needed to our data staging area for more transformation [49]. Our data source, ESRI census 2000 data, consists of information that is needed to visually represent and describe geographical information. For our data warehouse, we only extracted descriptive data since data needed for visual representation is no use for the purpose of our data warehouse. Such descriptive data includes information regarding state, counties, blocks, roads, demographics, and more. One of the main goals of our extract process is to analyze our source data so we can capture information that will be essential for satisfying typical business users' reporting and analytical needs.

In this extract process, we load our data into our *data staging* area. We describe our staging area as the engine room of our data warehouse, where our data is loaded and prepared for our transformation process. We keep our staging area very close to the image of our GIS tables. This allows us to query our staging area and our data warehouse / marts for QA (Quality Assurance) purposes. Therefore, we see our staging area as a snapshot of our data. We implement this process by manually downloading our pre-selected GIS data files from the ESRI Census 2000 TIGER/Line file website. Specifically for our project, the data we downloaded is for Alameda, California, refer to **Error! Reference source not found.**.

4.4.2.2 Transform

Once our staging area is ready, we begin our transformation process. Our transformation process involves the following steps:

Integration

It is essential that our tables are integrated for reporting. GIS tables were designed for application use which makes the data design to be dispersed. An example of this is the Roads file. This file is divided into tables, and each table has the roads for a certain block. This is not idea for our data warehouse since we will have to query each table and join them to get the list of all the roads in Alameda. Our best solution is to *integrate these files into one table, and this was implemented using a GIS software called ArcGIS.*

Another issue that we encountered is that when we extracted files for the Alameda County, some tables (e.g. Line features) do not have an indicator that it belongs to Alameda. To fix this natural keys such as county_id and state_id where added to these tables. In this way, when we load information for another county, we will be able to distinguish the county and state to which a record belongs.

Data Cleansing

In this step, we conduct data cleansing to fix our data in our tables before we load them to our conceptually designed dimensions and facts. The following shows the type of transformations that was performed for this step:

1. *Delete irrelevant rows*. To create our Landmark dimension, we needed to ensure that land marks that were not named were not included in our loading process. This is because these unnamed land marks do not provide value for our reporting and analysis. It is important that we always take into account our business users' needs. To a business user, it would not make sense to create a report of land marks that have no name. Therefore, it would be beneficial to delete them since this can reduce the number of records we join dimensions to facts in our queries.
2. *Merge tables*. Tables that are identical to each other are sometimes best to be merged into one table. One example of this scenario is the school district tables. We have 4 school district tables in our staging area: elementary, secondary, middle, and unified. To retrieve a list of all school districts in Alameda, we have

to join these 4 tables. Joining them together to create a school district dimension by appending all rows from these four tables and adding an indicator field optimizes querying and makes reporting easier.

3. *Purge fields*. Fields that are unrelated to reporting and are non-descriptive are deleted from our tables. These fields include polygon and point coordinates.
4. *De-duplicating rows.* Rows in our dimensions are de-duplicated to avoid explicitly specifying a "distinct" command in our SQLs every time we want to retrieve dimensional information. This also improves our run time whenever we join using our dimensions.

Calculation

It is essential that the data we present to our users are in the correct format. For example, we had to re-project (transform GIS layer from one coordinate system to another) most of the polygon data layers in order to provide correct area values for each polygon feature in meter square unit.

Null Values

In creating our fact tables, we have to add the surrogate keys from our dimensions that relate to our facts in our star schema. A problem arises when two scenarios apply [48, 49]:

1) A fact row has no relationship to the dimension.
2) The dimension key cannot be derived from the source system data. These scenarios will lead a surrogate key for that dimension in that fact table to be NULL. Surrogate keys that are NULL can lead to spurious results in

calculations and/or loss of data in queries unless we incorporate “outer joins” in our queries.

As a solution to these scenarios, a physical unknown row is created in each dimension table with an ID of -1. Unresolved fact-table rows are then mapped to this special dimension row. In this way we force the cardinality between dimensions and facts and we do not have to explicitly specify outer joins [49].

4.4.2.3 Loading and indexing

Our load process is implemented manually. So each dimension and fact in the data warehouse is loaded separately using SQL commands. Load optimization is an important part of our load process. This is done by adding indices to each of our staging tables. Indices are determined depending on the joins needed to create a dimension or a fact.

4.4.3 MySQL GIS Data Warehouse

To create the data warehouse we must go through a process of identifying possible dimension and fact tables based on the source data. The end results are physically created MySQL dimension and fact tables containing clean, unambiguous and conformed data to be used for our data marts. First we analyze the source data to create dimensions and then afterwards we look for facts. Dividing these two steps helps us clearly define our process and avoids confusion when doing the data analysis for the first time.

4.4.3.1 Finding Our Dimensions

We analyze the source TIGER/line geographical data first picking out easily identified dimensions. Most of our dimensions result naturally from the county level grain of the source data since most of the data layers in the source are attributed to a county within a state this is the lowest level of grain for which we can track these entities [48]. Layers such as county, bodies of water and school districts are all examples of data which lends itself naturally to be defined as dimensions. Each of these layers has the granularity level of the county and has descriptive attributes that can be used in the dimensional model.

Aside from the easily identified dimensions we have also chosen to create a state dimension. While this dimension does not seem to follow the county granularity we can include it because each county recognizes the state which it belongs to.

In addition the named feature dimension is derived from the line feature data layer and does not occur naturally. This dimension is created from the line feature dimension and is useful because it gives a more clear set of data. The data source of the line features includes many records which do not have a feature name. We make the assumption that this characteristic might be confusing to end-users and so we instead identify only those named features and include them in the named feature dimension.

Our final list of dimensions is quite small. We have determined eleven dimensions; state, county, tract, block, designated place, geographic location, land mark, line feature, named feature, school district and bodies of water as seen in Figure 4.4 below.

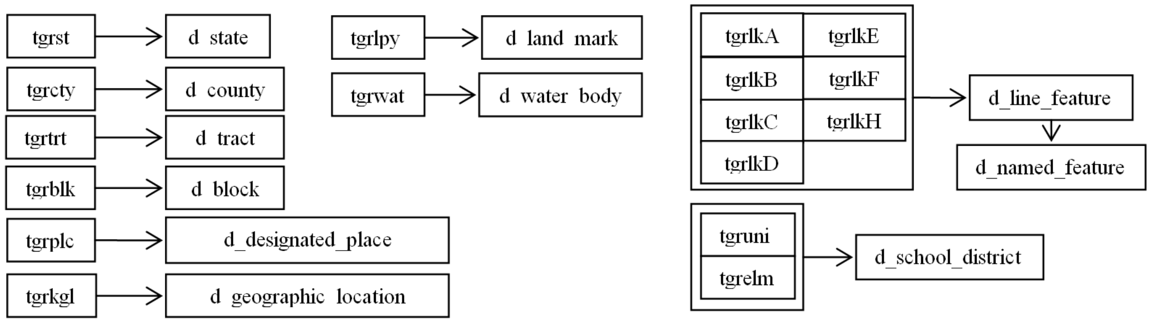

Figure 4.4 Source TIGER/line data layers to conformed dimensions

4.4.3.2 Creating the Dimension Tables

The dimensional tables are physically created in the data warehouse so that they may be accessed by multiple data marts through views at a later stage in the data warehousing process. In our implementation we use MySQL to realize the database for the data warehouse, creating standard physical SQL tables for each individual dimension.

As shown in Figure 4.4, we prefix each dimensional table with ‘d_’ to indicate that the table is a dimension. We continue this prefixing convention throughout the data warehouse so that we can easily identify and differentiate between separate types of tables such as dimensions, facts and data mart views.

When creating the dimensions we follow a standard outlined by Kimball to ensure that our dimensions are robust and reusable. Kimball presents the basic structure of a dimension as shown in Figure 4.5 and we use this as a minimal set of components for each of our own dimensions [48].

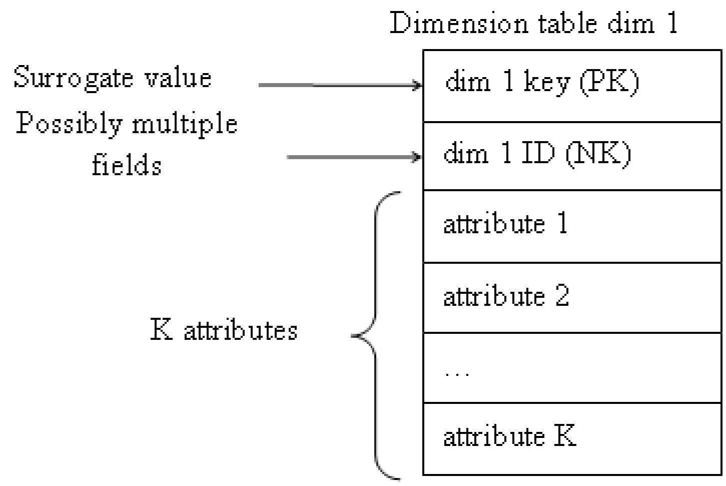

Figure 4.5 The basic structure of a dimension [48]

We present here a specific example from our data warehouse to illustrate each of the key components of this model. The 'd_county' dimension created from the 'tgrcty' source table has a surrogate key, natural keys and descriptive attributes for a county. The surrogate key is a unique incrementing integer added so that we do not need to rely on consistently formatted natural keys. For instance in the future we may add records that have a different format of natural key and this dimension would still be usable since there will be a surrogate key used for joining.

This source table contains two natural keys that we include in our dimension, the stateid and the countyid. It is important to keep these keys in the dimension so that they can help identify particular counties and which states they belong to. There is a list of attributes in the source table; STATE, COUNTY, Shape_length, and Shape_area of which we keep only STATE, COUNTY, and Shape_area based on our assumptions that these attributes will be the most useful to an end-user. Finally, in order to make the dimension more end-user friendly we rename these attributes state_desc, county_desc and area respectively.

Note that the area here appears to be data that should be included as a measure in a fact table rather than as a descriptive attribute. However since this value is either static or very slowly changing it may be used as a descriptive attribute in the dimension or as a measure in a fact [48].

4.4.3.3 Finding Our Facts

Deciding on facts for our data warehouse again requires data analysis. Fact tables are based on the grain of the data and contain measurements about a particular item in relation to different dimensions. [4]. For the scope of our data warehouse we have decided on four fact tables relating to county, county demographics, block demographics and block summary. The county fact is created as a factless fact, meaning that it does not contain any measures because the dimensions are straight forward and there are no measurements to take. The other three facts include measures of different populations based on the grain of the fact, either on the county or block level.

The choice of which facts to create is based on our assumptions of what information we think is interesting and useful for the data warehouse. In real business use the choice of facts would be constrained and dictated by business units, constraints and rules.

4.4.3.4 Creating the Fact Tables

We implement our fact tables physically in our MySQL data warehouse in a similar fashion as our dimensions. Here a prefix of 'f_' denotes a fact table and clear names such as county and census summary are applied as shown in Figure 4.6. Included in the fact tables basic structure are foreign key entries to the relating dimensions surrogate keys and measurements where they are relevant. The four fact tables created are f_county, f_dem_county, f_dem_block, and f_census_summary.

As previously mentioned there is a factless fact, f_county, which does not include any measurements as in Figure 4.6. This fact is also a special case because we use it to look at many dimensions that are not related to each other. Consider taking a count of the roads and also the bodies of water in a county. The roads and the bodies of water do not relate to each other except that they each belong to a county. Our fact has the surrogate keys to each of these dimensions and enables us to query these dimensions from our single fact table. Since these two dimensions are not directly related on any particular record in the fact table we use a surrogate key of -1 which points to either a dummy road or dummy body of water so that we do not accidentally perform a cross product or relate information that does not make sense. We will go into further detail of how these queries are performed in the query section.

f_county
county_key
state_key
geographic_location_key
…
line_feature_key

f_dem_block
state_key
county_key
block_key
Measures
P0010001
P0010002

Figure 4.6 Example fact tables from our data warehouse. f_county is a factless fact. f_dem_block is a fact with population measures

Our fact tables that do contain measures are all referencing population data. This data is related by grain to either a county or a block depending on the fact. The implementation of these tables is similar to the factless table since they include foreign keys to the surrogate keys of the dimensions as before, however here there is no need for dummy surrogate keys in the fact. Since the dimensions for these facts do directly relate to each other, for instance county and block, we can include both surrogate keys on any particular record. Along with the keys we also have the various population measures relating to each particular dimension. Figure 4.6 has an example fact, f_dem_block, which is similar to the fact tables, f_dem_county and f_census_summary, where measures are included.

4.4.4 Data Marts

4.4.4.1 Data Mart and the Data Warehouse Bus Architecture

An issue that arose when creating our data warehouse is planning the warehouse construction. The problem was we were uncertain whether we should build the whole data warehouse all at once or build separate subject areas. The plan we chose was the hybrid of these two options, the *Data Warehouse Bus Architecture* approach [49]. This approach focuses on a step-by-step implementation of separate data marts. This is done by producing a master set of conformed dimensions and standardizing the definition of facts. A conformed dimension, by definition, has the same meaning with every potential fact table to which it can be joined. Hence, it is identically the same dimension for all the data marts. For example, our conformed dimension county is the same dimension that is used in our block census summary mart, county information mart, and block demographics mart.

Table 4.1 shows the data warehouse bus architecture matrix for our geo-data set. The column headings on the y-axis list our conformed dimensions, and the x-axis lists our data marts.

Each of our data marts is a logical subset of the complete data warehouse [35]. A data mart is a complete "pie-wedge" of the overall data warehouse. In our case, our data warehouse consists of the physical tables of our conformed dimensions and conformed facts. We show this in Figure 4.7. Notice in Figure 4.7 that our data marts consist of views of the conformed dimensions and conformed facts of our data warehouse. Each data mart enforces a restriction of the data warehouse to a single business group. This enables us to control who gets access to each of our data marts.

Table 4.1 The Data Warehouse Bus Architecture for our geo-data set

	Block	Tract	Designated Place	Geographic Location	County	Line Feature	Named Feature	State	Body of Water
Block Census Summary	✓				✓			✓	
County Information	✓	✓	✓	✓	✓	✓	✓	✓	✓
Block Demographics	✓				✓			✓	

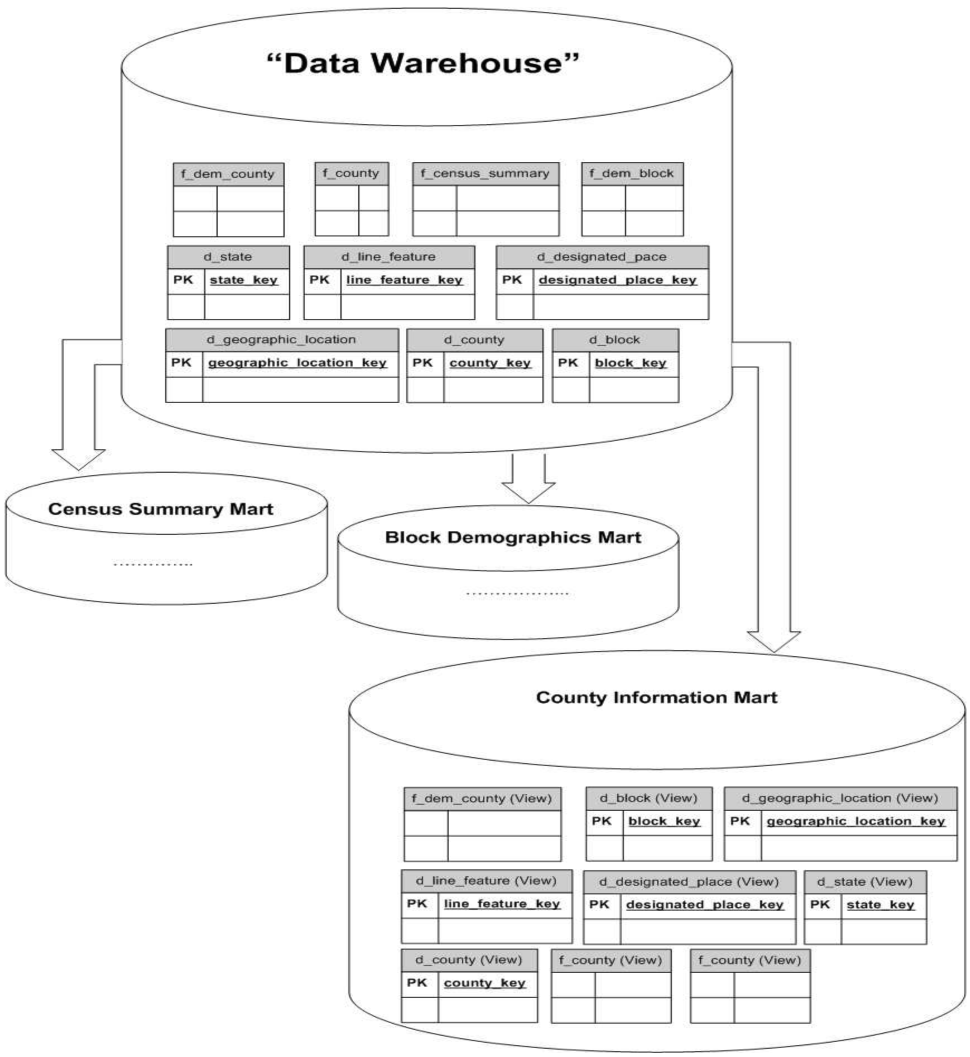

Figure 4.7 Data warehouse and its constituent data marts

An important fact to note is that each of our data marts has its own granularity. For example, our Census Summary Mart does not have the same granularity as our County information Mart. The granularity of our Census Summary Mart is by block, while the granularity of our County Information Mart is by county. If a user wants to query information regarding blocks in a county, they would have to access the Census Summary Mart, and the County Information Mart for the county.

4.4.4.2 Dimensional Modeling

Each of our data marts is designed using a star schema dimensional model shown in Figure 4.8. It is important that we choose the correct dimensions and facts for each of our data marts. We also have to make sure that the granularity of each conformed dimension and conformed fact is the same.

Dimensional modeling has many advantages that other OLTP (Online Transactional Processing) models lack. First is that dimensional modeling (using Star schema) is a predictable, standard framework, consisting of dimensions and facts. The model is designed to improve analytical processes by its simplicity. Judging from the tables that we extracted from the ESRI website, the table structures are very disintegrated since the modeling technique is centered on visual representation for GIS. For example, our Line Feature is a combination of 7 tables in our GIS source system. Creating a dimension out of these 7 tables improves our querying since we do not have to join 7 tables when we require a list of all line features. By transforming it into a dimensional model, we are able to create dimensions and facts that we can integrate using our bus architecture. The second strength of the dimensional model is that it is scalable. It is easy for us to add a fact or a dimension in our system.

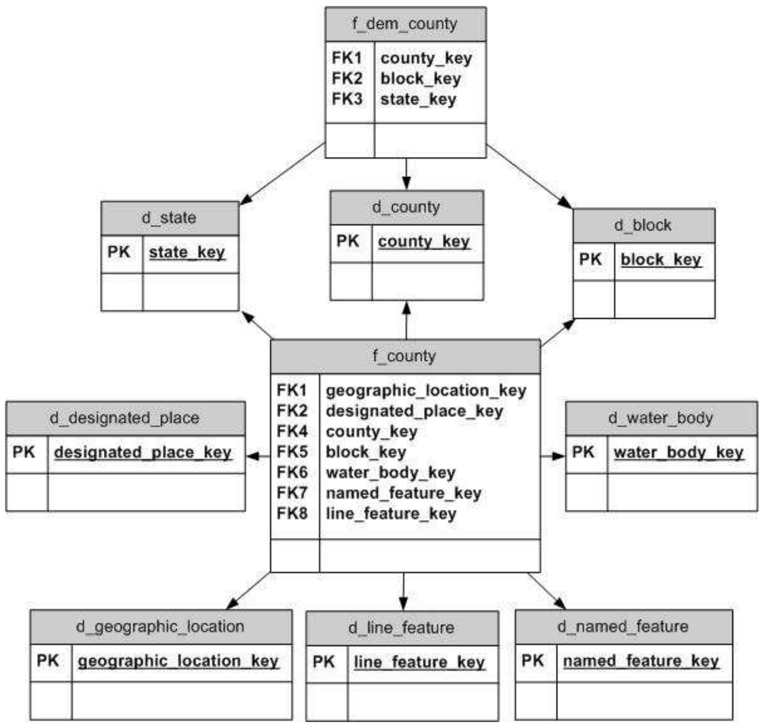

Figure 4.8 Star schema for County data mart

4.4.5 Query Reporting

When reporting we can simply choose a data mart which suits our needs and query the measures and dimensions. We can create reports on many types of interesting data such as the types of roads in a particular county, number of elementary school districts in a state, the highest population counties and any other related descriptive attributes and measures.

For reporting end-users generally require some sort of application tools that provide data access, spreadsheets, a graphics package and a user interface to simplify use of the data warehouse [49]. Due to the scope of this project we did not include any special end-user functionality except for direct access ad hoc SQL. Here we provide some "template" or example SQL, such as Query 1, for use with our data warehouse model to alleviate some of the problems with ad hoc query design.

Query 1. *Selection of Number of Designated Places for Alameda County*

```
SELECT e.state,
       e.county,
       e.designated_place_count
FROM
    (SELECT s.state_desc as state,
        c.county_desc as county,
        count(d.designated_place_key) as designated_place_count
     FROM dm_county_f_county f1,
          dm_county_d_county c,
          dm_county_d_state s,
          dm_county_d_designated_place d
     WHERE f1.county_key = c.county_key
       AND f1.state_key = s.state_key
       AND f1.designated_place_key = d.designated_place_key
       AND d.designated_place_key != - 1
       AND c.county_desc = 'Alameda'
     GROUP BY state, county
    ) as e;
```

Query 1 will give a simple result of the total number of designated places in Alameda county and will also display the name of the state which Alameda belongs to, California, and also the name Alameda. To extend this query onto any other dimensions in dm_county data mart we can simply add separate sub-selects for each dimension that we wish to report on. The sub-selects are necessary here because we are finding COUNT information and we do not want to accidentally do a cross product and produce erroneous results.

4.5 Discussion

Due to the illustrative nature of this application, we are not targeting to accurately describe or calculate differences in execution time of queries against the source data and the data warehouse tables. For instance the source data table for block demographics contains more than 500,000 records but for our purposes we loaded approximately 14,000 records and thus we cannot make fair comparisons on queries using these records. Instead of discussing direct numerical analysis we will instead present the advantages of our data warehouse process and model as is related to end-user needs, ease of use, and security.

4.5.1 GIS Comparison

The advantage of creating a data warehouse specifically for a geo-data set is that we were able to organize the data in a way that will be beneficial for reporting and analysis. With the use of our bus architecture model we were able to create data marts that suit the specific needs of our targeted potential business users. Also, for reporting purposes, our star schema makes analysis simpler because of its non-complex model. Star schema modeling also helped us arrange our data for query performance. We also performed data cleansing to help our intended users understand the source data better. Also another good trait of having a data warehouse for GIS is that it is separate from the source system. Meaning querying in the data warehouse for information would not affect transactional performance for the GIS.

4.5.2 Business Needs

The great thing about creating a data warehouse for GIS is that it is independent of the source system. If the business requires that some data had to be calculated or modified, it is possible to do so in the data warehouse. Hence, it is easier to satisfy users' requests. Having data marts in the data warehouse also enforces security since each data mart corresponds to specific business users. Consistency of the data is also a remarkable trait of the warehouse. In our project, we made sure that the granularity of our data marts are on the same level. Also, we are sure that in our data warehouse, the data is a snap shot of single point in time. This ensures that we do not have to worry about data inconsistencies.

4.5.3 Data Integration

Despite the fact that our data marts have their own security, granularity, and intended business unit groups, it is still possible to integrate them. In our project, if only we have enough business users to define their needs, then we should be able to link our data marts together, or create more data marts. The bright side is that our data marts are open for integration.

Also, it is also possible to include external sources that business users request. In the source system, it is not convenient to do this since we are always cautious of corrupting our GIS data.

4.6 Possible Extensions

It is possible to expand on the work presented in this chapter. We have followed many basic traditional data warehousing practices and this leaves the availability to easily extend the presented data warehouse. In the future of this project there would be a high priority placed on loading additional data so that the data warehouse could have more real world practical use. Also we would then be able to conduct numerical analysis comparisons between our data warehouse and the source data.

An issue with this chapter is the manual processing required to transform and load the data into the data warehouse. There would be large benefits to introducing an automated component of the data warehouse to alleviate this time consuming process. Additionally it will be important to make sure that front-end graphical user interfaces could be used with the data warehouse to enable easier access to the data.

The solutions presented within this chapter do not contain processes for storing any of the spatial data portions of geographical data sets. For a full solution to data warehousing of GIS data we would need to include support for spatial data and possibly a component to visually display the geographical data as other GIS applications would.

4.7 Concluding Remarks

This chapter presented the advantages of the data warehouse process and model as it relates to end-user needs, ease of use, and security. It ends up with defining the benefits of using a data warehouse model for GIS data.

We went through a standard ETL (Extract-Transform-Load) process by extracting data from our GIS data source to our data staging area, transforming our extract data to conformed dimensions and conformed facts, and loading these to our data warehouse. A standard process of finding our dimensions and facts was conducted by analyzing our source GIS data.

In our transformation phase, we conducted data cleansing by merging / splitting tables, de-duplicating rows, creating calculated fields, purging unnecessary columns to create our dimensions and facts which was loaded to our data warehouse. Indices were also added to our data warehouse tables for query optimization.

After the dimensions and facts were created, we followed a data warehouse bus matrix approach to create our data marts. This approach encouraged us to use conformed dimensions and conformed facts into our data marts, making our data integrated yet separated by subject areas. Data marts were implemented using SQL views from the physical tables in our data warehouse. Each data mart has is created for a particular purpose and is dedicated for a specific business unit group - this was accomplished by enforcing a star schema model for each data mart.

The construction of our data warehouse enables us to query and create reports against our data marts in a simple and efficient manner. Our queries are standardized since querying in a star schema model involves joining dimensions and facts. This makes it easier for our end users to analyze GIS data since knowledge of the data source structure is not necessary.

Our data warehouse is still on its beginning stages. There are still many amendments that can be done to satisfy the business needs of our intended end users. Further improvements to our data warehouse include loading of additional data, scripting manual processes for our ETL, incorporating a front-end tool to improve usability and analytical capabilities for our end users, and adding security to our data marts.

From this application we learned that the data warehouse is very beneficial for improving decision making process such as report generation that are not appropriate for our GIS processes. Data warehouse provide a repository of transformed data that can be reported without requiring any modification to the source system. Most of all, it allows our intended business users to query and analyze information without detail technical knowledge of our data warehouse.

Chapter Five: **DATA GOVERNANCE: A KEY ISSUE IN BUILDING ENTERPRISE DATA WAREHOUSE**

5.1 Introduction

This chapter articulates data governance as one of the key issue in building Enterprise Data Warehouse. The key goals of this chapter are to: define a framework for Data Governance processes and procedures [18]; define the scope of and identify major components of the data governance processes [65]; The Data Governance framework must accord with the Enterprise Data Management covered in this thesis. Risk management and compliance requirement motivated the development and deployment of IT Governance and Data Governance. Additionally, the client-centric focus of business organizations coupled with aggressive attention to the bottom line propelled initiatives such as Data Governance to the top of the list of IT and business executives [41]. The recent financial crisis which spawned the worldwide economic meltdown has been to a great extent blamed on non-trustworthy and non-transparent data. It is becoming progressively and patently evident that data must be managed like other assets such as financial and human resources. It has to have defined and mandated set of controls where compliance can be objectively measured and reported.

The rest of this chapter is organized as follows. Section 5.1 presents the importance and role of data governance. Section 5.2 highlights the need for data governance. Section 5.3 covers data governance maturity model. Section 5.4 presents approaches for data governance framework. Section 5.5 includes some concluding remarks.

5.2 Importance and Role of Data Governance

Business communities all across organizations are becoming increasingly dependent on their ability to quickly access, easily use, effectively share and efficiently maintain, quality and timely business information which they need to help achieve success in their business objectives. Meeting these needs is the basis of the business requirements for the creation and implementation of a data warehouse environment which will contain, and enable easy access to, all the required business information. Data warehousing processes are used to design and develop data repositories for efficient enterprise reporting and decision support systems; data warehouse design and development already attracted the attention of several researchers. Building data warehouse [32, 36, 38, 48, 49, 54, 76, 77] is important but it is at least equally important to maintain the quality of the data warehouse [23, 33]. Ballard et al. [4] discussed data modeling techniques for building data warehouse. Bonifati et al. [6] described the importance of data marts in designing data warehouse. Sen and Sinha [69] compared the different methodologies for data warehouse development. Golfarelli et al. [35] describe a process for building a data warehouse from an entity-relationship schema. Jukic [46] discussed some modeling strategies for data warehouse projects. There is no de facto standard for data warehousing techniques but the basic methods and processes outlined by Kimball, Chaudhuri and Dayal [14] are an excellent place to start [25]. However, data governance becomes an important issue to consider when dealing with data sources and utilization. In other words, one key issue counted towards success is having data governance guiding and monitoring the design and development of enterprise data warehouses.

Data governance is important for maintaining data quality [53, 61, 62, 67, 70] Crié and Micheaux [16] discussed how it is possible to best benefit from customer data. Dember enumerated and discussed seven stages for effective data governance [18]. To benefit best from data, organizations put huge effort on data integration [21, 22]. Recently, IBM delivered data governance service in their effort to help companies in protecting sensitive information [42]; also, the IT Governance Institute delivered several reports, e.g., [43, 44]. Data governance can be defined as the process by which decisions are made around data investments in an enterprise and the management of the data as a strategic corporate asset for competitive advantage. A good data governance framework typically answers questions about how decisions are made, who makes the decisions, who is held accountable, and how the results of decisions are measured and monitored. Based on this definition, everyone in an organization has some form of data governance responsibilities. In organizations where the data governance process is ad hoc and informal with a lack of consistency of data across the enterprise, accountability is weak and there are no formal mechanisms to measure and monitor the outcomes of the decisions.

"IBM Council predicts data will become an asset on the balance sheet and Data Governance a statutory requirement for companies over the next four years" IBM Press Room 2008-07-07

5.2.1 Importance of Data Governance

IT is critical to enterprise success, provides opportunities to obtain a competitive advantage and offers a means for increasing productivity. One of the key themes of large enterprises is to leverage IT successfully to transform the enterprise and create value-added products and services. IT is fundamental for managing enterprise resources, dealing with suppliers and customers, and enabling increasingly global and dematerialized transactions. An ever larger percentage of the market value of enterprises has transitioned from the tangible (inventory, facilities, etc.) to the intangible (information, knowledge, expertise, reputation, trust, patents, etc.). Many of these assets revolve around the use of IT. Moreover, a firm is inherently fragile if its value emanates more from conceptual, as distinct from physical, assets. Good governance of IT therefore is critical in supporting and enabling enterprise goals.

Data is a major component of Information Technology. It has been widely regarded is the only durable product of Information Technology. As such, data must be managed as a corporate asset. It Governace is a topic which has been widely covered and institutionalised. Data Governance gained recent attention and importance. How does IT Governance relate to Data Governance? The Data Governance Institute explains the relation as follows “What’s the difference between Data Governance and IT Governance? Let’s start with another question: What’s the difference between data/information and information technology (IT)? Consider a plumbing analogy: IT is like the pipes and pumps and storage tanks in a plumbing system. Data is like the water flowing through those pipes. Suppose you were afraid that the water flowing through your pipes was poisoned. What type of plumber would you call? None, of course! Plumbers are specialists in the pipes and pumps and storage tanks – not in what's flowing through them. You'd call in specialists who know how to test for water quality – specialists who could tell the difference between clean water and other types of clear liquids"..

The Wikipedia defines governace as " the activity of governing. It relates to decisions that define *expectations*, grant power, or verify performance. It consists either of a separate process or of a specific part of management or leadership processes. Sometimes people set up a government to administer these processes and systems.

In the case of a business or of a non-profit organisation, governance relates to consistent management, cohesive policies, processes and decision-rights for a given area of responsibility ». Similar to the implemetation of othe governance such as Corporate Governance and IT Governance, Data Governance requires a framework based on three major elements:

- **Organizational Structure**: definition of roles and responsibilities. Who makes the decisions? What structural organizations need to be created, who will take part in these organizations?
- **Operational Process**: what are the actions and activities to be undertaken during the data lifecycle? What are the decision-making processes for proposing creation or extension of data assets?
- **Performance Management**: How will the results of these processes and decisions be tracked, monitored, measured, and reported? What mechanisms will be used to capture and communicate gaps to stakeholders?

5.2.2 Role of Enterprise Data Governance

In the context of building the Enterprise Data Warehouse, the data governance component would typically involve determining requisite process, procedures, policies and organization structure required to effectively manage all the data assets involved in the Enterprise Data Warehouse. The data governance processes addresses key points, such as:

- Reviewing and approving organizational structure and functions to facilitate development of appropriate data architecture to support the Enterprise Data Warehouse.
- Establishing enterprise-wide data management framework defining policies, governance, technology, standards and processes needed to support the collection, maintenance, controls, and distribution of processed data or information.
- Ensuring that data maintenance processes provide security, integrity and auditability of the data from its source to its target in the Enterprise Data Warehouse.
- Instituting internal audit programs, as appropriate, to provide periodic independent audits of data maintenance processes and functions.
- Monitoring the enterprise-wide observance of the data management framework including ongoing updates to procedures and documentation as needed.

- Establishing clear and comprehensive documentation for data collection, transformation, aggregation and definition that includes data mapping to source/aggregation routines and data schematics.
- Establishing standards, policies and procedures around data cleansing, validation of business rules, balancing to source data.
- Establishing procedures for identifying and isolating data errors, including data integrity issues with source, downstream and or external systems.
- Establishing data quality threshold acceptance levels as required by business rules
- Establishing standards and procedures for data error handling.
- Distribution of access control based on user roles, responsibilities and consistent with data classification.

5.2.3 Key Recommendations

The following are the key recommendations for an effective data governance strategy to implement the Enterprise Data Warehouse:

- Develop a comprehensive data governance guideline of principles and policies that the Enterprise Data Warehouse development teams can leverage to create standardized, reusable data assets.
- Develop the organizational structure to support the data governance framework.
- Develop rigorous processes and procedures to support the framework.

5.3 The Need for Data Governance

Data governance can be defined as the process by which decisions are made around data investments in an enterprise and the management of the data as a strategic corporate asset for competitive advantage. A good data governance framework typically answers questions about how decisions related to data are made, who makes the decisions, who is held accountable, and how the results of decisions are measured and monitored. Based on this definition, everyone in an organization has some form of data governance responsibilities. In organizations where the data governance process is ad hoc and informal, with a lack of consistency of data across the enterprise, accountability is weak and there are no formal mechanisms to measure and monitor the outcomes of the decisions.

Wikipedia defines data governance as a practice that encompasses the people, processes and procedures required to create a consistent, enterprise view of an organization's data in order to:

- Increase consistency & confidence in decision making
- Decrease the risk of regulatory fines
- Improve data security
- Maximize the income generation potential of data

Data Governance is a subset of overall Information Technology (IT) governance. According to the IT Governance Institute, IT governance "is an integral part of enterprise governance and consists of the leadership and organizational structures and processes that ensure that the organization's IT sustains and extends the organization's strategies and objectives."

At the heart of the governance responsibilities of setting strategy, managing risks, delivering value and measuring performance, are the stakeholder values, which drive the enterprise and IT strategy. Sustaining the current business and growing into new business models are certainly stakeholder expectations and can only be achieved with adequate governance of the enterprise's IT infrastructure.

Data governance, like other governance subjects, is the responsibility of the executives. Data governance is not an isolated discipline or activity, but rather is integral to enterprise governance. It consists of the leadership and organizational structures and processes that ensure that the enterprise's data assets sustain and extend the enterprise's strategies and objectives. Critical to the success of these structures and processes is effective communication among all parties based on constructive relationships, a common language and a shared commitment to addressing the issues.

Data governance responsibilities form part of a broad framework of enterprise IT and enterprise governance and should be addressed like any other strategic agenda. For critically dependent data assets, governance should be effective, transparent and accountable. This means that the executives should be very clear about their responsibilities, and should have a system in place to deliver on those responsibilities. These responsibilities generally relate to data asset alignment and use within all activities of the enterprise, the management of technology-related business risks and the verification of the value delivered by the use of data assets across the enterprise.

5.3.1 Purpose

The purpose of data governance is to influence IT endeavours and directs data-intensive operations, to ensure that data performance meets the following objectives:

- Alignment of data assets with the enterprise IT assets and realization of the promised benefits.
- Leverage of data assets to enable the enterprise by exploiting opportunities and maximizing benefits.
- Responsible use of data resources.
- Appropriate management of data related risks.

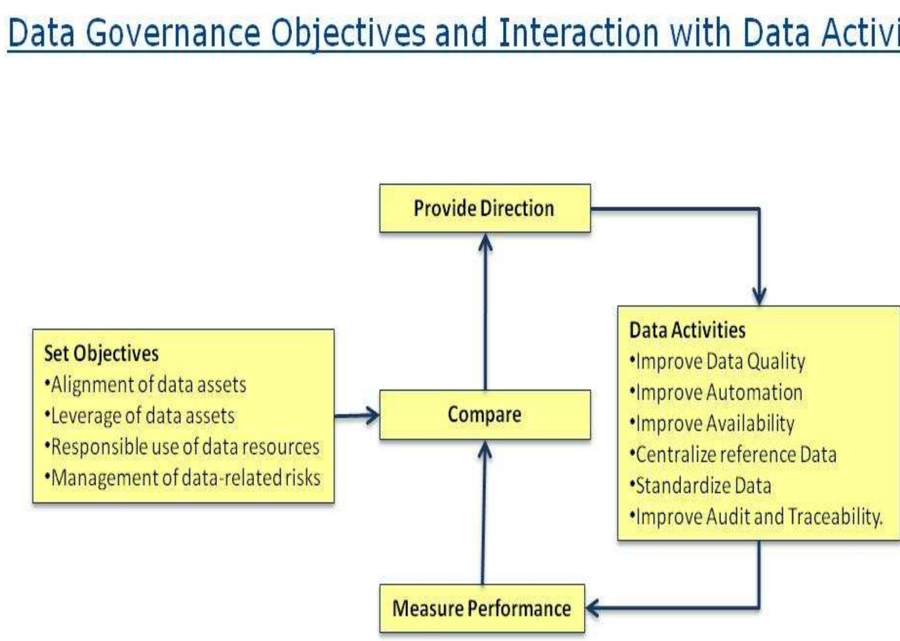

Figure 5.1 Adapted from Board Briefing on IT Governance, 2nd Edition, the IT Governance Institute®

Data governance usually occurs at different levels, with team leaders reporting to and receiving direction from their managers, with managers reporting up to the executive, and the executive to the board of directors. Reports that indicate deviation from targets will usually include recommendations for action to be endorsed by the governing layer. Clearly, this approach will not be effective unless strategy and goals have first been cascaded down into the organization. The following illustration conceptually represents the interaction of objectives and data activities, from a data governance perspective that can be applied among the different levels within the enterprise (Figure 5.1).

***Objective Setting**:* The governance planning process starts with understanding the objectives of the enterprise and identifying the means by which data assets can support those objectives. The executives set objectives and prioritize activities based on value and feasibility to execute.

***Compare**:* From then on, a continuous loop is established starting with comparison of the objectives which have been set to the current environment.

Provide Directions: Results of the comparison may identify gaps which would drive improvements to the current processes.

Measure performance**:** The improvements to the current processes are again measured and compared to the objectives, resulting in redirection of activities where necessary and / or a change of objectives where appropriate.

While setting direction is primarily the responsibility of the IT executives, and performance measures that of data management, it is evident they should be developed in concert so that the objectives are achievable and performance measurement supports objectives effectively.

5.3.2 Objectives of Data Governance

There are four objectives that drive data governance: Data value and alignment, accountability, performance measurement, and risk management. Each of these objectives must be addressed as part of the data governance process

5.3.3 Data value and alignment

One of the primary goals of data governance is to ensure alignment between the business units and IT. By creating the necessary structures and processes around data assets, management can ensure that only those projects that are aligned with strategic business objectives are approved, funded, and prioritized. Furthermore, alignment also deals with balance between investments that run the current business, grow existing businesses, and have the potential to transform the business, while delivering value by managing projects that are on time, on budget, and deliver expected results. Delivering value to the business typically means things like regulatory compliance, growing revenues, improving customer satisfaction, increasing market share, reducing costs, and enabling new products and/or services.

5.3.4 Data Risk management

As more of an organization's value proposition is built on risks associated with data, these are often the same as risks to the business. Therefore, managing data risk is paramount. Data risks include security breaches arising from hackers and denial of service attacks, privacy risks arising from identity thefts, recovery from disasters, and resiliency of systems to outages, and the risks associated with project failures.

5.3.5 Accountability

Governance is about accountability. Regulatory requirements such as Basel II and Sarbanes-Oxley legislation are intended to hold senior executives accountable for the integrity and credibility of their financial information and controls. Data governance holds data management accountable for the return on its investment in data assets, as well as the credibility of its own information and controls.

5.3.6 Performance measurement

Accountability in IT governance requires that you keep score, typically by implementing a form of balanced scorecard. The IT Balanced Scorecard consists of four perspectives: IT Value, User, Operational Excellence, and Future Orientation. Two of these perspectives contain measures for the two key governance objectives: IT value and risk management. The IT value perspective contains specific measures for IT/business alignment and IT value, while the operational excellence perspective contains specific measures for managing IT risk.

5.4 Data Governance Foundation

Having established a working definition of good data governance, the next step is to establish a foundation on which to build the data governance framework. The foundation consists of three parts: understanding the governance maturity level, knowing how structural issues impact governance, and understanding the four objectives of data governance.

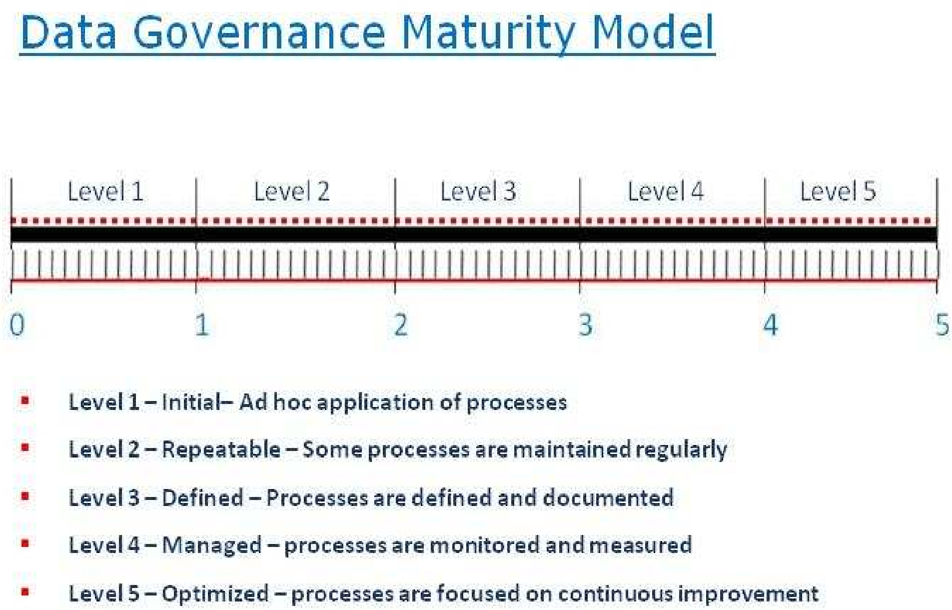

Figure 5.2 An adaptation of the model published in March 2002 edition of CMMI from SEI), chapter 2 page 11.)

5.5 Data Governance Maturity Model

Organizations in the process of developing or evolving their current data governance framework typically begin by conducting a data governance maturity assessment. Understanding the current state of data governance within an organization is extremely helpful in trying to formulate a data governance strategy. The Data Governance Maturity Model is comprised of five levels (Figure 5.2).

Level 1 - Initial/Ad Hoc

The concept of data governance does not exist formally and oversight is based mostly on management's consideration of data-related issues on a case-by-case basis. The governance of data depends on the initiative and experience of the IT management team, with limited input from the rest of the organization. Executives are involved only when there are major problems or successes. The measurement of data asset performance is typically limited to technical measures and only within the IT function.

Level 2 - Repeatable but Intuitive

There is a realization that more formalized oversight of data assets is required and needs to be a shared management responsibility requiring the support of senior management. Regular governance practices such as review meetings, creation of performance reports and investigation into problems do occur. However, these practices rely mostly on the initiative of the IT management team, with voluntary or co-opted participation by key business stakeholders, depending on current IT projects and priorities. Data problems identified are tackled on a project basis with teams formed as necessary to undertake improvements.

Level 3 - Defined Processes

An organizational and process framework has been defined for oversight and management of data-related activities and is being introduced to the organization as the basis for data governance. The data management team has issued guidance, which has been developed into specific procedures for management covering key governance activities. These include regular target setting, reviews of performance, assessments of capability against planned needs, and project planning and funding for any necessary data improvements. Previous informal but successful practices have been institutionalized and the techniques followed are relatively simple and unsophisticated.

Level 4 - Managed and Measurable

Target setting has developed to a fairly sophisticated stage with relationships between outcome goals in business terms, and data and IT improvement measures now well understood. Real results have been communicated to management in the form of a balanced scorecard. The enterprise’s management team is now working together for the common goal of maximizing data assets' value delivery and managing data related risks. There have been regular assessments of capabilities and projects have been completed that have delivered real improvements to overall performance. Relationships among the data functions, their users in the business community, and external service providers are now based on service definitions and service agreements.

Level 5 - Optimized

The data governance practices have developed into a sophisticated approach using effective and efficient techniques. There is true transparency of data activities, and the stakeholders are in control of the strategy. Data related activities have been optimally directed toward real business priorities, and the value being delivered to the enterprise can be measured and steps taken on a timely basis to correct significant deviations or problems.

The balanced scorecard approach has evolved into one that is focused on the most important measures relevant to the enterprise's overall business strategy. The effort spent on data-related risk management (and on IT management activities generally) has been streamlined through adoption of standardized and, where possible, automated processes. The practice of continuous improvement of data and its processing asset's capabilities are embedded in the culture and this includes regular external benchmarking and independent audits providing positive assurance to management.

Overall, the cost of data management is monitored effectively and the organization is able to achieve optimal spending through continuous internal improvements, the effective outsourcing of selected services, and effective negotiation with vendors. When dealing with external business partners or service providers, the organization is able to demonstrate first-class performance and demand best practices from others.

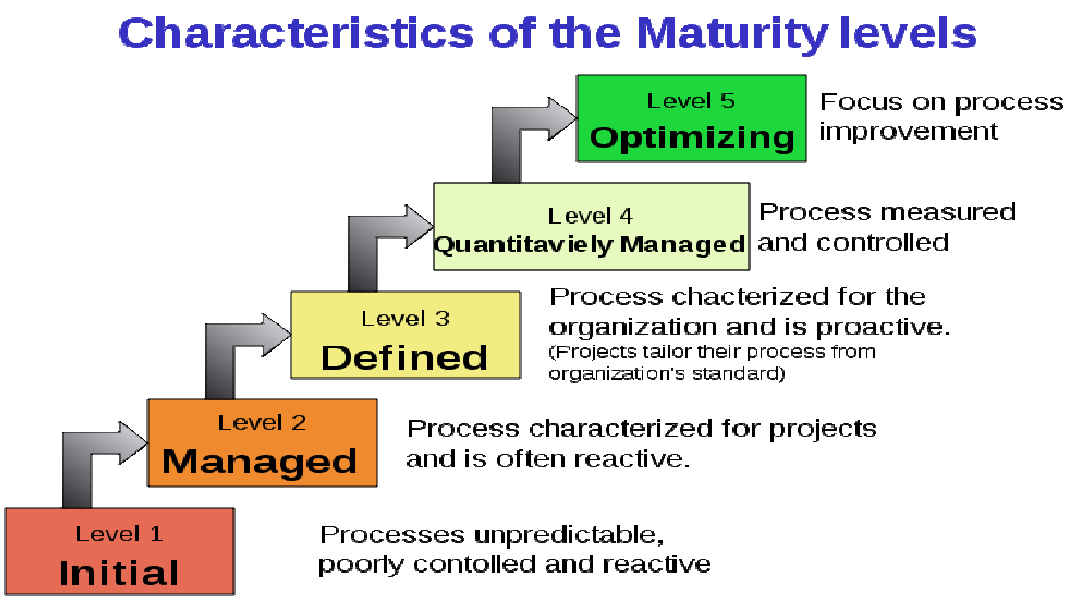

Figure 5.3 Characteristics of the maturity levels (adopted from http://software.gsfc.nasa.gov/docs/What%20is%20CMMI.ppt)

5.6 Approaches for Developing Data Governance Framework

Any attempt at developing and enforcing data governance requires an understanding of the structural or organizational pieces of the framework. Four different major types of data governance structures (consisting of centralized, decentralized, federated, or project based organizations) could be used for developing the data governance framework. Each organizational structure presents a different challenge in implementing data governance as characterized by its decision-making process.

5.6.1 Centralized

In a centralized organization, all data related decision making and the data budget are in one place; because of this centralized structure, they are much easier to manage and require much less effort to organize. The CIO and the data management executive can take the lead in developing the governance processes, and work directly with the executive team. The challenge for centralized IT organizations is to for the Data Governance to become a policing exercise, and to ensure that business and function units have a voice in the process.

5.6.2 Decentralized

Decentralized organizations most often reach the fragmented stage because each decentralized data function has developed its own data governance processes, but there are no formal processes across business units or between business units and corporate. Data asset investment decisions may be optimized at the business unit level, but they are not optimized across the enterprise. This often results in duplicated infrastructure, applications, databases, and little if any sharing of systems or expertise.

5.6.3 Federated

These are hybrid organizations that have both centralized and decentralized components. Most infrastructure and enterprise wide applications and data are centralized in a corporate IT organization and operated as a shared service with charge backs, while business units retain control over BU-specific applications, data and development resources. This is an attempt to create the best of both worlds; centralized control for reduced costs, and applications development left with the business units where it can be more responsive. The challenge for federated IT organizations is to balance the needs of the business units for infrastructure investments, and to conform to enterprise architecture and standards.

5.6.4 Project-based

Project-based IT organizations are a relatively new phenomenon and take their lead from professional services firms. They are a form of centralized IT in that all IT resources are centrally located and report into a corporate CIO, but they differ from the other organizational structure types mostly in the development area.

As distinguished from the traditional applications development group, a project-based organizational structure is built around resource pools, often called competence centers, each consisting of similar resources. Also, the traditional line manager is replaced in favour of a resource manager or competence center manager who heads up each resource pool. This new role's performance is measured on resource utilization and the ability to loan out qualified staff in sufficient quantity as required by the project portfolio and pipeline. For project-based IT organizations to be effective, they need a strong governance mechanism in place to ensure that the right projects are selected and funded. The challenge in project-based organizations focuses on the project selection, funding, and prioritization processes.

5.6.5 Leveraging the Framework for Design and Optimization

5.6.5.1 Critical roles for Data Governance

It is important for an organization to send a strong message that data governance is important. By dedicating and holding senior staff accountable for data governance, continual focus is placed on the issue.

The Data Steward is responsible for acting as a liaison between Business and IT and is accountable for communicating, and facilitating consensus around, business data-related requirements. The Data Steward is responsible for:

- Business Data Definitions: Facilitate and ensure the documentation, communication, and maintenance of business data definitions, including: Business Naming Standards, Business Attribute Definitions, Business Rules, Transformations & Calculations, Business Metadata
- Business Data Requirements: Provide support for ensuring that business data requirements are complete
- Business Data Quality: Identify and assess business data quality and escalate initial improvement plans to remediation owners
- Business Data Issues: Facilitate the resolution of business data issues (e.g., data quality, missing data)
- Business Data Policies: Support the communication and enforcement of business data policies (based on current, applicable data policies)
- Program Support: Provide support for data modeling, gap analysis and new data definition

- Other Business Requirements: Document and communicate requirements for: Data Access, Security, Availability and Recoverability (based on Corporate Data Policies) – in the cases where systems / applications are created specifically to support compliance
- Liaison: Act as the Data liaison between the Business and IT through the Data Custodian

The Data Custodian plays an integral part in the development and support programs, ensuring use of appropriate data policies, standards, and strategies. The Custodian helps improve the way the organization will process, protect, store, archive, document and report on data by reviewing and revising where necessary existing data policies, processes, procedures, guidelines, and methods. Ensures through effective communication to all involved the best practices in performing the data custodian role that will help meet the data requirements, as defined by the business. The Data Custodian is responsible for

- IT Data Standards: Define data standards for IT in support of the business requirements
- Data Security: Provide adequate logical and physical data security as defined by the Data Steward
- IT Data Quality: Identify, analyze and consolidate IT data quality / completeness issues and communicate appropriately to the business

5.6.5.2 Enterprise Data Warehouse Design and Operation

There are a number of key design premises behind a successful Enterprise Data Warehouse deployment. Among them are:

- Recognition that the design must support usage across a wide range of business communities
- Business requirements will change and evolve continuously
- Business users will want to access, use and exploit data in different ways
- Usage patterns should dictate design patterns
- Data quality, accuracy and consistency are paramount. Business user's trust must be earned. Data lineage information is critical to the correct use of data

Data governance plays a pivotal role across the entire Data Warehouse design, development, operations and usage spectrum. Partnership between IT and the business communities enable effective prioritization of the enhancements and a clear understanding of the implications of change and the importance of compliance with standards.

The following list of touch points provides some of the key areas of interest for compliance and governance.

- Data Sourcing – Authoritative data
- Data standards
- Metadata (formats, structures, definition, lineage)
- Data Model integration
- Data Classification - Restricted, confidential, internal use, public domain

- Data Retention standards
- Data Privacy and security
- Access and Auditability

5.7 Case Study – RBC Financial Group

To illustrate the value of governance we examine the situation at Royal Bank of Canada both prior and post adoption of an enterprise Information Management and data governance strategy.

5.7.1 Prior to adoption of Enterprise Information Management & Governance

Each application group within IT had responsibility for their own solution design and development. Data and data management functions were distributed and part of each application group. Standards were developed for each application and integration requirements between applications were addressed by copying data from one system to another. Technology choices were made specific to the application and not always by the IT group. Years of following this approached resulted in a wide range of technology and tool deployments, and significant data fragmentation and inconsistent representation across systems. In addition, data was replicated as and when needed to satisfy new business requirements and numerous formats, structures and definitions of data were created. Needless to note that this type of environment made it very difficult to establish the authoritativeness of data and for our business units to gain access to timely and consistent information from across the various systems. Costs for support and maintenance started growing significantly due to complexities giving impetus to the need for change.

5.7.2 Post Adoption of Enterprise Information Management & Governance

In contrast to the foregoing, an organizational restructure brought together all aspects of Data Management within a single functional group. The development life cycle was enhanced to ensure standard and consistent data management functions were applied to all initiatives, an architectural forum was mandated and empowered to provide guidance and direction to project initiatives at the very on-start of the project. Since re-development of extensive legacy was cost prohibitive, the Data Warehouse Environment evolved as an atonement of the sins of the past to prove information base with consistent, standardized and integrated data. Centralization, standardization and integration are part of the core principles that guide solution development. Adoption of an Enterprise Information Model provides the data modellers with guidance on data design. Technology decisions are made by IT. All of this has enabled significant reuse of data and process, made it much easier for the business to gain access to data and provided them with more time to focus on business issues rather than technical data sourcing issues. Costs for development and support were also reduced and growth projections within the business are more easily supported.

5.8 Concluding Remarks

In conclusion, the Data Governance Program is an initiative which touches the enterprise systems and applications ecosystem. It is a key requirement to achieving “trust and transparency" which is fundamental in meeting internal and external bodies of audit and regulations. Data Integration, the holy grail of Information Management programs is solely dependent on Data Governance. It is the vehicle to track, monitor and report on all Data Management policy, standards, best practices and guidelines. Data Integration initiatives such as Master Data Management (MDM) is another attempt at standardizing data so it can be shared. Data Governance is the centerpiece in Master Data Management. Successful implementing of the data governance initiative is predicated on a regimented set of processes and procedures and mandated roles and responsibilities. Failing that, Data Governance will –at best- be treated as something "nice –to-have" where success will be sub-optimal.

Chapter Six: **EMPLOYING FREQUENT PATTERN MINING TO DISCOVER COMMUNITIES OF TABLES**

6.1 Introduction

Data mining is the process of discovering and predicting hidden and unknown knowledge by analyzing known databases. It is different from conventional database querying in the sense that querying is a retrieval process, while mining is a discovery process. Data mining has several applications, including market analysis, pattern recognition, gene expression data analysis, spatial data analysis, among others. Actually, market basket analysis has been the motivating application that raised the interest in data mining since its development by Rhesh Agrawal in 1993 at IBM research center [1, 2]. The basic idea is to analyse the log of transactions in order to find items that are frequently purchased together. The outcome could serve several purposes including shelving of items by placing close by items that are purchased together, developments of advertisement campaigns, promotions, etc [1, 2].

Realizing the effectiveness of data mining in prediction, this chapter investigates the applicability of data mining techniques in improving query performance. This is possible by identifying tables that appear together most frequently in queries. The outcome could serve for better allocation of tables and for effective query processing. First, a brief and general overview of data mining is presented followed by a specific frequent pattern mining. Subsequently, the focus shifts to closed frequent patterns and how they could be employed to identify tables that are mostly accessed together.

Maximal-closed frequent itemsets will be closely investigated because small frequent itemsets are subsets of these maximal-closed itemsets.

6.2 Association Rules Mining

Association rules mining is one of the most attractive data mining techniques [1, 2]; it was developed for market basket analysis and later has been adapted to different interesting applications. Association rules mining investigates the correlation between items within the transactions in a given database. So, any problem that can be modeled in terms of transactions and items could be classified among the applications of association rules mining framework. Fortunately, the problem investigated in this dissertation could benefit from association rules mining by considering known queries as transactions and the tables defined in the data warehouse constitute the set of items.

Given a database of transactions, such that each transaction contains a set of items, the association rules mining process determines correlations of the form X ➔ Y, such that X and Y are disjoint sets of items from the investigated database. A correlation X ➔ Y is characterized by two parameters: *support* and *confidence*. Support is the percentage of transactions that contain all items in X U Y, by considering all the transactions in the database. Confidence is the percentage of transactions that contain all items in Y by considering only transactions that contain at least all the items in X. A rule is worth further investigation if it has both high support and high confidence as compared to predefined minimum support and confidence values, specified mostly by the user who is expected to be a domain expert. It is also possible to derive the two parameters in an automated way by considering characteristics of the data. However, this is outside the scope of the work described in this dissertation.

Market basket analysis is one of the first applications of association rules mining [2]. Organizations that deal with transactional data aim at using the analysis outcome to decide on better marketing strategies, to design better promotional activities, to make better product shelving decisions, and above all to use these as a tool to gain competitive advantage.

Agrawal et al [1] first introduced association rule mining or frequent itemset mining in 1993, and since then it is one of the problems most investigated by researchers in the data mining arena. During the past two decades, several research groups have provided solutions to this problem in many different ways, e.g., [31, 40, 59, 60]. The time and space scalability of the developed approaches greatly vary based on their techniques to mine the investigated databases. They mainly differ in the number of database scans, and hence the time consumed by the mining process, as well as in the data structures they use, which are mostly main memory resident. All the latter mentioned performance related issues are outside the scope of this dissertation because the main target is to determine the maximal-closed frequent itemsets regardless of the performance of the approach to be utilized.

6.3 Formal Definition

The immense popularity of the association rule mining problem lies in the fact that it is probably one of the most practical real world issues in recent applications. A formal definition of the problem of association rule mining as given in [2, 31] can be stated as follows. Let $I=\{i_1, i_2, \dots, i_m\}$ be a set of literals, called items, and let D be a set of transactions, where each transaction T is a set of items such that $T \subseteq I$. Each transaction in D is assigned a unique identifier, denoted TID. A transaction T is said to contain an itemset X if $X \subseteq T$. The support of an itemset X in D, denoted $S_D(X)$, is the fraction of the total number of transactions in D that contain X. Let $S(0<S<1)$ be a constant called minimum support, mostly user-specified. An itemset X is said to be frequent on D if $S_D(X) \geq S$. The set of all frequent itemsets L(D,S) is defined formally as,

$$L(D, S) = \{X \mid X \subseteq I \wedge S_D(X) \geq S\}$$

An association rule is a correlation of the form $X \Rightarrow Y$, where $X \subseteq I$, $Y \subseteq I$ and $X \cap Y = \phi$. The rule $X \Rightarrow Y$ has support S_D in transactional database D if S_D% of the transactions in D contains XUY. The rule $X \Rightarrow Y$ holds in transactional dataset D with confidence c if c% of the transactions that are in D and contain X also contain Y. Given a set of transactions D, the problem of mining association rules is to generate all association rules that have support and confidence greater than (mostly user-specified) minimum support (S) and minimum confidence (c), respectively. The association rule mining task can be broken down into two steps:

Step 1 identifies all frequent k-itemsets from the database, where k is the cardinality of each itemset; and

Step 2 generates rules from these large itemsets in a relatively straightforward way. An association rule is specified as follows:

Rule form: "Body ➔ Head [support, confidence]".

These are two example associations:

buys(x, “diapers”) ➔ buys(x, “beers”) [0.5%, 60%]

major(x, “CS”) ^ takes(x, “DB”) ➔ grade(x, “A”) [1%, 75%]

A very naive and brute force approach for finding all frequent itemsets from a particular database is to generate all possible itemsets from the database, and then check the corresponding frequency of each itemset against the database. But the problem with this approach is that there can be $2^{\mathbf{I}}$ candidate itemsets to be checked, and it is not computationally or space efficient to determine the frequency of such huge number of itemsets.

Over the past two decades, researchers in this area have come up with numerous association rule mining algorithms in an attempt to efficiently solve the problem, e.g., [31, 40, 59, 60]. Covering all association rule mining approaches is outside the scope of this dissertation; interested readers may refer to the literature for comprehensive coverage. In this chapter, the focus is describing one approach that has been extended for finding maximal-closed itemsets which are the target of the study covered in this dissertation. Apriori is described as a simple algorithm, though less efficient.

6.4 Apriori: A simple algorithm for finding frequent itemsets

The Apriori algorithm, see (Figure 6.1) makes multiple passes over the data to find frequent itemsets of all lengths. In the first pass, it counts the support (frequency) of individual items and determines which ones of them are large, i.e., satisfy the minimum support constraint. In each subsequent pass, it uses the large itemsets generated from the previous pass to produce the new potentially candidate large itemsets, and counts the support of these candidate itemsets to find out those that are indeed frequent.

The Apriori Algorithm

Join Step: C_k is generated by joining $L_{k-1}$with itself

Prune Step: Any (k-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset

Pseudo-code:

C_k: Candidate itemset of size k

L_k : frequent itemset of size k

L_1 = {frequent items};

for (k = 1; L_k !=ϕ; k++) **do begin**

C_{k+1} = candidates generated from L_k;

for each transaction *t* in database do

increment the count of all candidates in C_{k+1} that are contained in *t*

L_{k+1} = candidates in C_{k+1} with min_support

end

return $\cup_k L_k$;

To understand how C_{k+1} is generated from L_k, consider the following five frequent itemsets: L_3={*abc, abd, acd, ace, bcd*}; by self-joining: L_3*L_3, we produce the following quadrables:

- abcd from abc and abd
- acde from acd and ace

Applying the pruning phase will eliminate *acde* because *ade* is not in L_3; and as a results, the outcome will be C_4={*abcd*}is a candidate that needs its frequency be checked.

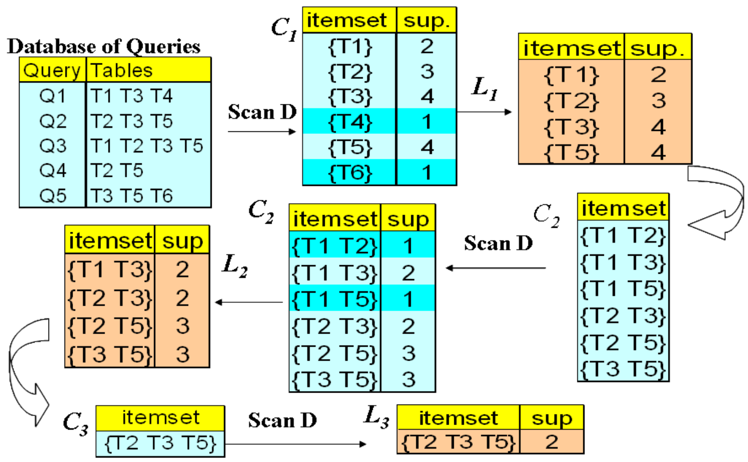

Figure 6.1 Finding frequent itemsets from a set of five queries on six tables

The Apriori algorithm generates candidate itemsets in the current pass by considering large itemsets from the previous pass only. The intuition behind this is based on what is known as the Apriori-heuristic, which states that an itemset may be frequent if all its subset itemsets are frequent. This can be done by a self-join of the itemsets in L_k (frequent itemsets of length k) with L_k itself and then pruning from the result any itemset which all of its subsets are not in L_k. This process results in generating a much smaller number of candidate itemsets. Therefore, candidate generation consists of two steps: the join step and the pruning step. After the pruning step, the remaining candidates are checked by scanning the database to determine their frequencies. This process is recursively repeated until it is not possible to construct more frequent itemsets.

The Apriori algorithm is level wise in nature. It requires multiple database scans to find the support count for a potentially large candidate itemset; this can be very time consuming. Moreover, the Apriori algorithm requires generating a large number of candidate itemsets at each level, especially for the levels two and three; these can also be considered as CPU and memory intensive tasks.

There are several other Apriori-like algorithms, such as DHP [60], DCP [58], and DCI [59], which mainly focus on improving the performance of mining by reducing the candidate generation and/or by introducing special data structures that reduce the time for counting the support of candidates. On the other hand, algorithms like DIC [9] and CARMA [40] try to improve the performance by reducing the number of database scans.

The Apriori algorithm (as formally given next) is known as a candidate generation-and-test approach method:

- Generate length (*k+1*) candidate itemsets from length *k* frequent itemsets
- Test the candidates against the DB

To illustrate the Apriori algorithm within the scope of this dissertation, consider the example database of five queries shown in Figure 6.1. Each query runs by utilizing the tables listed in its row in the database of queries (see upper-left table in the Figure 6.1). This is a transactional database where the five queries are the transactions and the six tables are the items. The Apriori algorithm should return sets of tables that are frequently accessed together. The algorithm proceeds as follows. First, the six singleton candidate itemsets are determined with the frequency of each specified; only those that satisfy the minimum support of 2 are retained as frequent to be used in generating candidate itemsets of size two. Then, the database (of the five transactions) is scanned to find the frequency of the latter candidate itemsets in order to keep only frequent pairs; only four pairs survive as frequent and are used in generating the frequent itemset {T2, T3, T5}; the other two triplets {T1, T2, T3} and {T1, T3, T5} are not frequent. As a result, by omitting the four singleton frequent itemsets, we get five frequent itemsets, namely {T1, T3}, {T2, T3}, {T2, T5}, {T3, T5} and {T2, T3, T5}.

6.5 Finding Maximal-Closed Frequent Itemsets

Redundancy is the main problem with keeping all the frequent itemsets as described and enumerated in Section 6.3. In other words, the number of frequent itemsets is directly related to the overlap between the transactions. As the overlap between the transactions increases, the number of frequent itemsets increases. However, many of the frequent itemsets may share the same frequency and noticing this would help in minimizing the number of frequent itemsets to maintain by keeping only closed frequent itemsets.

A frequent itemset is said to be closed if and only if its support is different from all its frequent supersets. An unclosed frequent itemset has the same frequency as its immediate closed frequent superset. Therefore, by keeping only closed frequent itemsets we will not lose any information. This is true because the support of any itemset is less than or equal to the support of its subsets and when an itemset is frequent all its subsets are also frequent. We are even not interested in all closed itemsets, we are rather interested only in maximal-closed frequent itemsets. This will lead to the largest set of tables that are frequently accessed together, which is the target of our study.

A frequent itemset is maximal-closed if and only if it is closed and none of its supersets is frequent. Based on this, we keep frequent itemsets of maximum size. By considering the example in Figure 6.1, out of the nine enumerated frequent itemsets only six are closed frequent itemsets, namely {T3}, {T5} {T1, T3}, {T2, T5}, {T3, T5} and

{T2, T3, T5}. Finally, only {T1, T3} and {T2, T3, T5} are maximal-closed frequent itemsets.

The identified maximal-closed frequent itemsets will be sufficient for determining the frequency of each set of tables that are mostly accessed together. This allows us to concentrate only on sets of tables that either have different frequency, or once have same frequency they do not totally overlap, i.e., none of them subsumes the other. From our experience and as demonstrated by the conducted testing, the query performance has been positively affected by considering closed frequent itemsets of maximum. In other words, frequent itemsets, which are closed by themselves but are subsumed by other closed and maximal frequent itemsets have less effect on the performance compared to the maximal-closed frequent itemsets.

6.6 Testing and Analysis

To demonstrate the effectiveness of the proposed approach in practice, we conducted some experiments using a synthetic query set of 1000 queries on 50 tables; finding real data is very hard because this type of data is very sensitive and hence highly confidential. We have generated the data by restricting the number of tables that could appear in the same query was limited to be at most 20; that is, one query may require accessing at most 20 different tables, though in practice it is not more than four or five tables. The aim is to demonstrate how the large number of tables affects the result. This experiment is intended to demonstrate the scalability, applicability and effectiveness of the proposed approaches for large database environment. A small database example is illustrated in Figure 6.1 and the related analysis is included in Section 6.5. Finally it is essential to mention that the actual structure of the tables and their attributes (columns) do not directly affect the experiments as conducted in this thesis because our main concentration is on the usage of the tables. Each table is treated as a single unit. Columns and rows will be important to consider in the future once we extend this work to consider the columns and rows in the developed approaches.

The query set was generated to include queries of the form:

Select *
From $T_1, T_2, ..., T_n$

and each table Ti has the following structure: $T_i(A_{i1}, A_{i2}, ..., A_{im})$

where n is an integer between 1 and 50 and each A_{ij} is an attribute with domain type numeric or text; the number of attributes in each table ranges between 5 and 15 and the number of tuples per table ranges between 1K and 100K. The where clause is omitted in the queries because we concentrate only on the usage of tables in the queries. The select and where clauses will be considered in the future work when we will expand the work to cover columns and rows. Each of the 1000 queries used in this experiment is generated as follows. First, I generate a random number and normalize it into an integer, say n, between 1 and 20 (maximum number of tables allowed in a query). Second, I generate n random numbers and map each into a unique integer value between 1 and 50 (number of tables in the database). Third, I add to the query each table that corresponds to one of the n integer values generated in the second step. Finally, these three steps are repeated 1000 times for generating the 1000 queries to be used in the experiments. An alternative way to produce the query set could be by using the process described in Agrawal et al. [2] for generating transactional databases simulating data for market basket analysis. The nomenclature for the synthetic data produced by Agrawal et al.'s approach is TxxIyyDzz, where xx is the average transaction size, yy is the average size of maximal potentially frequent itemset and zz is the size of the database. Within the realm of my application, average transaction size corresponds to average number of tables per query, average size of the maximal potentially frequent itemset is 50 tables and database size is 1000 queries. I decided not to use this approach from Agrawal et al. because it was developed merely to serve market basket analysis and hence it would have been a risk to overload it for serving my target of producing queries for the experiments conducted in this thesis.

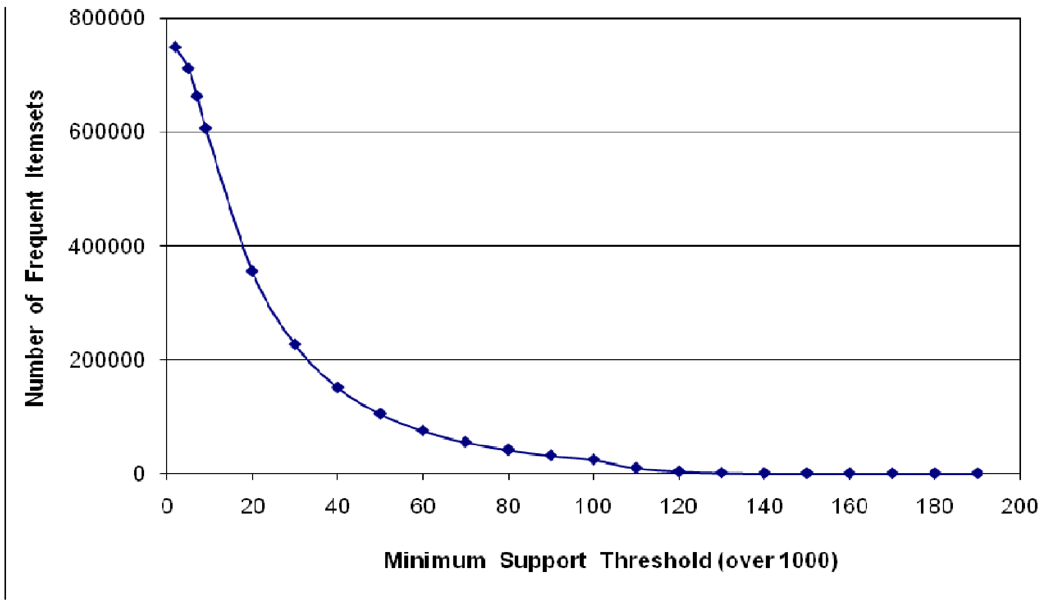

Figure 6.2 Number of Frequent itemsets when support range changes between 2 and 200 out of 1000

The model was constructed by using queries and tables to stand for transactions and items, respectively. Thus, the produced transactions-items matrix has two types of entries: 1 to indicate that the table is used in the query and 0 otherwise. Then Apriori was invoked with the mentioned matrix as input. The minimum support threshold was varied from 2 to 200 out of 1000 queries in order to discover the most frequent sets of tables that appear together in at least *n* queries, where *n* varies from 2 to 200 queries. The result is plotted in Figure 6.2, which shows how the number of frequent itemsets (tables that are frequently accessed together) decreases as the support threshold increases; singleton frequent itemsets are not counted in the numbers shown in Figure 6.2. A large number of frequent itemsets is produced for low values of the threshold, and the number of frequent itemsets sharply decreases as the threshold value increases; it somehow stabilizes as the minimum threshold approaches 60; it hits zero when the minimum threshold value increase above 175 (the number of frequent itemsets decreases to less than 10 after the threshold increases beyond 150). This demonstrates the need to consider only closed and maximal frequent itemsets. For each threshold value, the latter itemsets form smaller subset compared to the total number of frequent itemsets.

The curve plotted in Figure 6.2 is not very informative as it reflects the whole number of frequent itemsets. Instead, the curve plotted in Figure 6.3 shows the number of frequent itemsets that are closed and maximal. The latter curve is more realistic to draw conclusions from and to use as a basis for producing query processing strategies.

The target is to find the tables that are mostly accessed together. This will guide the process by making the rest of the tables ready once one of them is processed. However, in practical scenarios, not more than four tables are utilized in the same query in practice. Thus, only maximal closed itemsets whose size ranges from 2 to 5 could be informative and hold the promise of practical value. These are shown in Figure 6.4.

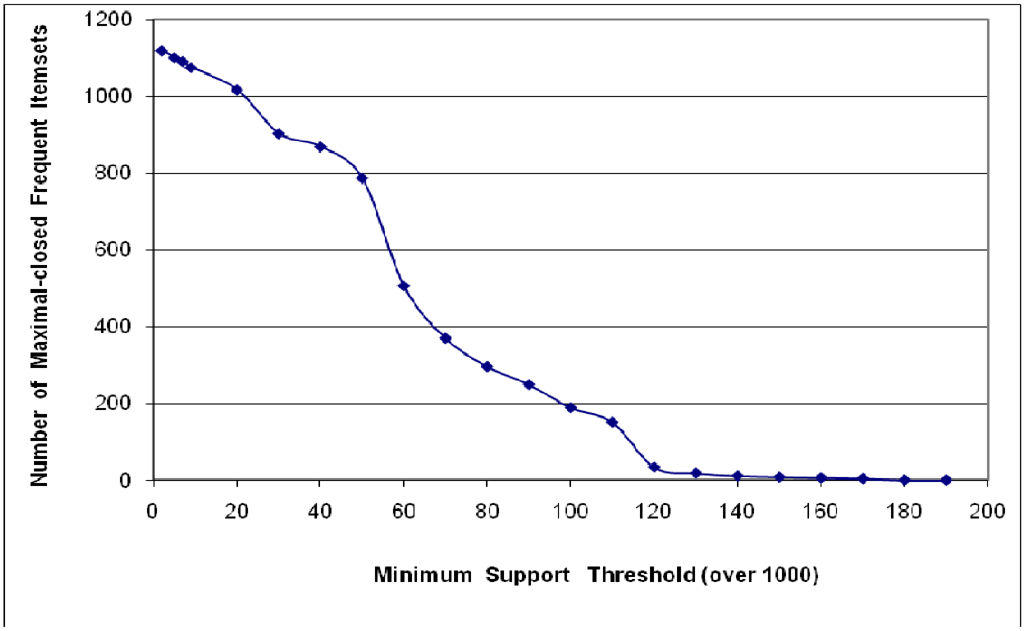

Figure 6.3 Number of maximal and closed Frequent itemsets when support range changes between 2 and 200 out of 1000

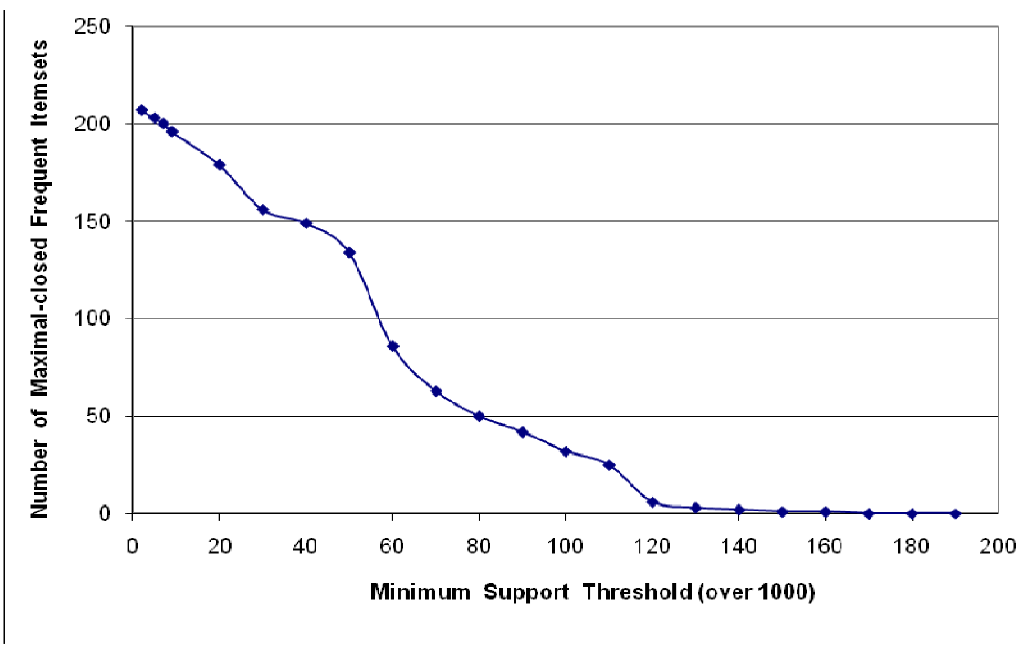

Figure 6.4 Number of maximal and closed Frequent itemsets of size 2 to 5 when support range changes between 2 and 200 out of 1000

The domain expert who is going to apply this methodology is expected to specify the minimum threshold value by deciding on his/her preference regarding the number of queries in which the tables in a given itemset must appear together in order for the itemset to qualify as frequent. By considering the maximal-closed frequent itemsets of size four, a small number of itemsets will be considered. For the 1000 queries and 50 tables dataset, the minimum threshold value is set at 100, 110 and 120. This approach produced maximal closed itemsets of 32, 25 and 6 respectively and whose size ranges from 2 to 5.Analyzing all the results plotted in the three graphs obviously demonstrates that considering a minimum threshold of 100 is a good choice. Furthermore, this approach affords the opportunity to concentrate more on the analysis of a dataset of manageable size. These itemsets are beneficial to plan better for the tables that are mostly retrieved together. Later on in Chapter seven will illustrates how the maximal-closed frequent itemsets of arbitrary size could be used to construct a social network that help in better planning for the retrieval of tables that are mostly utilized in the same queries. However, it requires some extra work, but the social network analysis based approach as described in Chapter seven forms a systematic automated approach for identifying communities of tables that are mostly retrieved together and in addition, the method locates key tables in the system.

Chapter Seven: **BUILDING AND ANALYZING SOCIAL NETWORK OF CLOSED ITEMSETS OF TABLES**

7.1 Introduction

The study of social networks started in sociology to analyze social communities of humans within different contexts, including social interactions, business communications, international relations, political movements, etc. However, it is applicable to every domain where the interactions between items can be represented using a kind of network. The basic and the most time consuming step in the process is to build the social network. This is analogous to the database design process where most of the time is spent on understanding the problem and producing the conceptual model. Once built, the remaining steps of the database design process would follow a systematic way to implement the database and put it into practice. The same is true for social networks because the first step is model construction, and hence it is necessary to decide correctly and clearly on elements of the model, namely the actors and the interactions between them. Once built, we can apply different well-defined and tested measures to study the characteristics of the social network. In other words, social networks do exist around us and we only need some skills and expertise to model them first and later on analyse them for knowledge discovery.

Social networks are mostly analysed to discover social communities. Forming communities is natural and common in different domains. Stationary actors (interchangeably called individuals) whether tables in queries, words in documents, pieces of code in projects, software developers in a company, authors of manuscripts, items in stock, routers in a network, etc, could be grouped together into social communities. Graphs form the most attractive representation to describe social networks. Edges in the graph may represent social interactions, organizational structures, physical proximity, collaboration, or even more abstract interactions such as hyperlinks, similarity, coexistence, etc.

The interesting part of social network analysis and mining is the requirement for multidisciplinary domain expertise from sociology [11], behavioural science, psychology, statistics, mathematics, computer science, etc. Finding a harmony between these expertises is by itself challenging and requires lot of effort. Researchers have already applied this methodology to different domains, e.g., web mining [29, 50] and biological networks [34], among others. The analysis leads to valuable discoveries that may have essential social and economical impact. From social perspective, the discoveries may highlight terrorism groups [17, 51, 56], common hobbies, family relationships, social functions, occupations, friendship, disease biomarkers, etc [45]. From economical perspective, the analysis may lead to certain target customer groups [47], the development of effective drugs, identifying successful software development teams, etc.

For this thesis work, we have adapted the social network model to study communities of tables as social network. Our target is realizing tables into social communities for better analysis of their utilization in queries for better system performance in allocation of related tables in order to retrieve them effectively once needed.

We construct the social network of tables based on the outcome from the mining process described earlier in Chapter 6. Actually, the developed model is general enough to be applied to any domain because it involves two main steps that are capable of deriving the social network for any kind of data that fits the requirements of the frequent itemset mining model. For instance, to identify successful software development teams, we can consider all the history of the so far completed projects. Each project represents a transaction and all the software developers working for the company are the items. We use these transactions and items as the input to the mining process described in Chapter 6 and the outcome will be the maximal-closed sets of software developers who participated together in most of the completed projects. These maximal-closed itemsets could guide us to decide on a successful team for the next project. However, it is difficult, if at all possible, to make decisions based directly on the maximal-closed frequent itemsets because the process returns a large number of maximal-closed frequent itemsets as demonstrated by the test results described in Chapter 6. Consequently, we can use these maximal-closed itemsets to construct a social network of the software developers; then we can analyse the social network to find the key players and the communities of software developers.

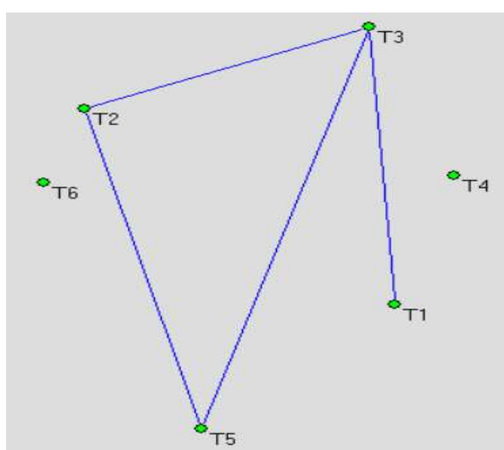

Figure 7.1 The Social network for the six tables used in the illustrative example in Chapter 6

As the scope of our work is data warehouse construction and development, we have applied the same model to develop the social network of tables. Shown in Figure 7.1 is the social network for the six tables used in the illustrative example in Chapter 6. There are two isolated tables, namely T4 and T6. On the other hand, T3 is a key player table; it is connected to T1 and it participates in a clique with T2 and T5. There are four communities in this network: {T1, T3}, {T2, T3, T5}, {T4}, and {T6}. This result happened to coincide with the result from the analysis in Chapter 6. This is not surprising because small number of maximal-closed frequent itemsets have been used in building the model. For a real larger setting, the results from the social network would be more compact and informative as demonstrated in the tests reported later on in this chapter. In other words, the results from the social network model would be at least as good as the results from the maximal-closed frequent itemsets model.

To construct the social network of tables, we use the maximal-closed frequent itemsets produced by the process described in Chapter 6. In other words, we produce a matrix representation of the social network from the maximal-closed frequent itemsets by finding for each pair of tables the number of maximal-closed frequent itemsets in which the two tables occur together. Using the latter matrix, we construct the social network of tables; then we analyse the network to find the social communities of tables. The conducted experiment revealed interesting results.

The rest of this chapter is organized as follows. Section 7.2 is an overview of the social networks methodology. Section 7.3 describes the proposed methodology for constructing the social network of tables. Section 7.4 reports the analysis results on the 50 tables used in the experiments conducted in Chapter 6.

7.2 Basic Methodology for Social Network Analysis

The study of social networks starts by defining the group of items, called actors, to be investigated; it is possible to have events in addition to actors; the latter setting leads to different model. In other words, we may have social networks modeling interactions within one group of actors, or interactions may cross to relate actors to events and even to other actor groups. The former is called one mode network and the latter form a group of n-mode networks with n=2 as the most commonly used setting.

After identifying the actors, the second step is to decide on how to model the interactions between them; the latter decision depends on different attributes to be picked carefully; the process is application dependent and it is directly related to the scope of the study to be conducted. Given a group of actors (individuals, animals, terms, plants, etc), social network analysis examines the structure of social relationships in the given group in order to discover the informal links between the involved actors.

Joseph Moreno may be considered as the father of social network analysis research due to his seminal work in 1934 [55]. However, the area started to attract more attention starting in the 1970s, and more recently the interest in social networks in booming due to the evolution of online social networks. The main question could be whether there will be tools and methods to effectively model and analyze the exponentially growing social networks? In other words, will the improvement and advancement in the tools and methods evolve at least as fast as the social networks themselves do? This is a legitimate question that raises serious concerns because discoveries due to social network analysis are expected to shape the economy, the research and every aspect of the daily life.

Graph theory [20] could be identified as encapsulating the most powerful concepts for modeling social networks; it could be seen as the main driving force for the research related to social network analysis. Though it can be directed a network is generally modelled as an undirected graph where nodes represent actors/events in the network and links simulate the interactions between the actors/events. The links may reflect either binary relationships (a missing link indicates no relationship) or weighted connections to indicate the degree of the relationship (which may be negative or positive). A graph that represents a social network is generally called *sociogram*.

Once the model is constructed, social network analysis could be classified into two categories, individual centric and group centric. The former starts the analysis from a single actor as a key player in the network and studies its vicinity. The latter, on the other hand, considers the whole group at once and studies the interactions within the group as a whole. The choice of which approach to follow depends on whether the interest is in studying the whole group at once or in identifying individual leaders and use them to influence the whole group. For instance, we may study the correlation between different terms in a course by analyzing how frequent the terms occur together in the same chapter of the book; we may even fine-tune to consider sections or paragraphs instead of chapters. We may also study the social communities of actors in a play by the analysis of how often they come together in the same scene or sketch. Finally, it is possible to have a social network represented as partite graph (bipartite graph is the most common) when it is intended to analyze the relationships between two groups of entities, namely actors and events or even two different groups of actors. Representing social networks as bipartite graphs is essential for some interesting applications like the patterns of patients' visits to clinics, the referral of general practitioners to specialists, individuals visiting/joining social clubs, link between genes and diseases, etc.

Formally a social network is represented as a graph with weights, denoted $G=(V,E,W)$, where V is the set of actors in the network, E is the set of edges connecting the actors to indicate relationship, and W is a $|V|\times|V|$ matrix of real values representing the weights of the different links between the $|V|$ actors; W is neglected in a social network with binary relationships between the actors. For totally connected graphs, $E=(V\times V)$, but in general $E\subseteq(V\times V)$. Once the graph is constructed, different metrics are employed in the analysis for knowledge discovery. The most commonly used metrics include density, centrality [8, 27], and cliques' identification (where every node is connected to every other node in a clique).

Density is measured as the ratio of the number of edges in E over the total number of edges in a complete graph (which is $|V|\times (|V|-1)$ for a complete directed graph); density gives an idea about cohesion. Density may also be applied to subgroups by considering subgraphs instead of the whole graph. Within a subgraph, density measures in the subgraph the ratio of the actual connections to all possible connections. Measuring the density between two groups (subgraphs) is also possible.

Centrality generally refers to the importance of individual actors in a given group. Centrality may be also measured in terms of degree, betweenness [7, 37, 57], closeness and eigenvector, which are demonstrated in Figure 7.2. Degree is a simple measure realized for actor *A* as the number of actors connected to *A* divided by the total number of actors minus one (i.e., $\frac{\text{degree}(a)}{|V|-1}$); degree is distinguished as in-degree and out-degree of each node in directed graphs. In case graph G is weighted, any of the three values in-degree weight, out-degree weight, or gain in weight could be used to measure the degree centrality. For actor *A*, and by considering each other actor *i*, the latter three weighted measures are computed as:

$$in\text{-}\text{degree}(a) = \sum_i Wia$$

$$out\text{-}\text{degree}(a) = \sum_i Wai$$

$$gain\text{-}in\text{-}\text{degree}(a) = \sum_i (Wia - Wai)$$

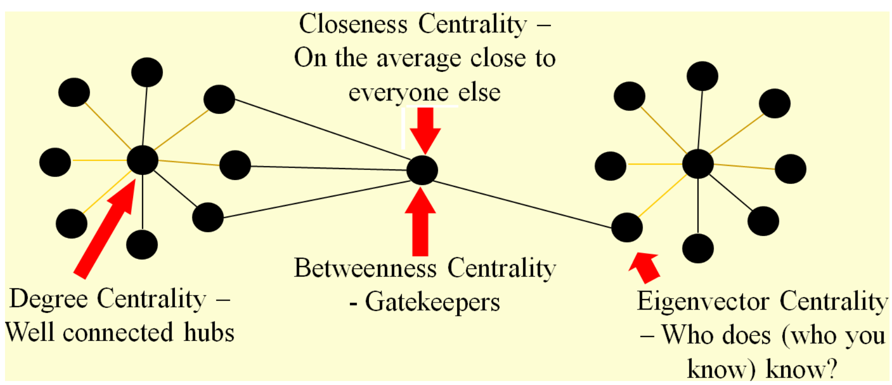

Figure 7.2 The four measures of centrality

Anthonisse [3] and Freeman [30] independently introduced betweenness as a measure of centrality for the analysis of social networks; this measure only considers the shortest paths in the graph. It refers to how a given actor could be considered as the hub of the network; this is determined by the number of shortest paths that pass via the given actor. In other words, other actors do not have direct link and must communicate via the given actor. Formally, let *dij* be the shortest distance between two actors *i* and *j*, $\Im ij(a) = \sum_{i}^{|V|} \sum_{j}^{|V|} (dij)$ is the set of shortest paths between all pairs of actors *i* and *j* in the social network, and $\Im iaj = \sum_{i}^{|V|} \sum_{j}^{|V|} (dia + daj)$ is the set of shortest paths that pass via actor *A* and connect *i* to *j*, i.e., $\Im iaj \subseteq \Im ij$; the betweenness of actor *A* is computed as:

$$betweenness(a) = \frac{\Im iaj}{\Im ij}$$

By considering connected actors (whether directly or indirectly), closeness centrality can be computed by measuring of how long it will take information to spread from a given actor to other reachable actors in the network. It is the ratio of the number of links that an actor must follow to visit each of the other reachable actors in the network; the actor is considered more central when its ratio of closeness is closer to 1, i.e., it is directly connected to all reachable actors. Once individual groups are identified, it is possible to study their overlapping members in order to figure out the interactions between the different communities within the network. Having interrelated communities is preferred to isolated ones; this is equivalent to having fuzzy clustering which is a more natural way of expressing membership for most real-world applications.

Eigenvector centrality measures the importance of an actor in a social network. It is computed based on the adjacency matrix *A*, where entry *Aij*=1 if *i* and *j* represent adjacent (directly connected) actors in the social network and *Aij*=0 otherwise. On the other hand, *Aij=Wij* for weighted graphs. The eigenvector centrality measure for actor *A*, denoted *Ea*, is computed as:

$$Ea = \frac{1}{\lambda \sum_{j=1, j \neq a}^{|V|} (Ej)} = \frac{1}{\lambda \sum_{j=1}^{|V|} (Aij \times Ej)}$$

where λ is a constant which is called the eigenvalue when the above equation is written in vector notation as *Ae*=λ; we are interested in the largest value of λ and the produced eigenvector *E* will contain the eigenvector centrality measure for actor *A*.

For the study described in this thesis, we compute the number of times two tables occur together in the same maximal-closed frequent itemset. This leads to a matrix that represents a graph that links tables by considering only connections that represent number of occurrences greater than the average of all the occurrences reported in the matrix. Then we try to find the communities of tables. We eliminate links that have frequency below the average frequency in the model because our target is to find important tables that play a key role according to the trend present in the test set of queries. This is analogous to failing students who score below the average in the exam. It is more effective than dropping links based on a prespecified threshold value. Actually, the average is a built-in threshold that incorporates characteristics of the analyzed domain.

7.3 Constructing Social network of Tables

The outcome from the maximal-closed frequent itemset mining process is a rich source of information for constructing a social network.

Assume the frequent itemset mining process produce *n* maximal-closed frequent itemsets, say MC_1, MC_2, …, MC_n. Using this information, we use the existing *m* tables, say T_1, T_2, …, T_m, to construct a matrix M=$m\times m$ to include one row and one column per table. Entries in matrix M are computed by considering each maximal-closed frequent itemset MC_k (k=1,n) and increment M(i,j) by 1 if the pair of tables Ti and Tj exist inside the same maximal-closed frequent itemset. In other words, M(i,j)=r, for all $1\leq i\leq m$ and $1\leq j\leq m$, where $0\leq r\leq n$ is the number of maximal-closed frequent itemsets in which the pair of tables (Ti,Tj) exist together. It is obvious that M(i,i)=n for all $1\leq i\leq m$; but we drop this self reference from consideration by setting to zero all entries along the diagonal of the matrix. After all entries in M are determined, we compute the average, say Av of all the values in M as follows:

$$Av = \frac{\sum_{i=1}^{m}\sum_{j=1}^{m} M(i,j)}{m^2}.$$

Based on the comparison of each entry in M with Av, we reset some entries in M to zero; we mainly set M(i,j)=0 if and only if M(i,j)<Av.

The revised matrix M represents the adjacency matrix of the actual social network where there exists an edge between tables Ti and Tj if and only if M(i,j)>0. After M is transformed into adjacency matrix, we construct a social network. Then we analyze the social network in order to discover the major communities of tables and the key players as individual tables.

In the rest of this chapter, we discuss the conducted experiments. We highlight the results of our approach and evaluate its effectiveness and applicability. We applied the methodology described above on the 50 tables dataset already utilized in the experiments conducted in Chapter 6. We used the maximal-closed frequent itemsets produced when the minimum support threshold was set at 10%, i.e., an itemset of tables is frequent if its tables appear together in 100 of the given 1000 queries. We have chosen 10% because the curves plotted in Figures 6.2, 6.3 and 6.4 started to be smoother at around 10%.

Using a smaller threshold will increase the amount of information to be processed and increasing the threshold value will limit the amount of information to be used in constructing the network and hence will not lead to effective and informative results. In other words, the minimum support threshold increases as the number of maximal-closed frequent itemsets decreases and vice versa.

From the conducted tests, it is obvious that 10% is a good choice to get enough links in the network. The constructed network is shown in Figure 7.3.

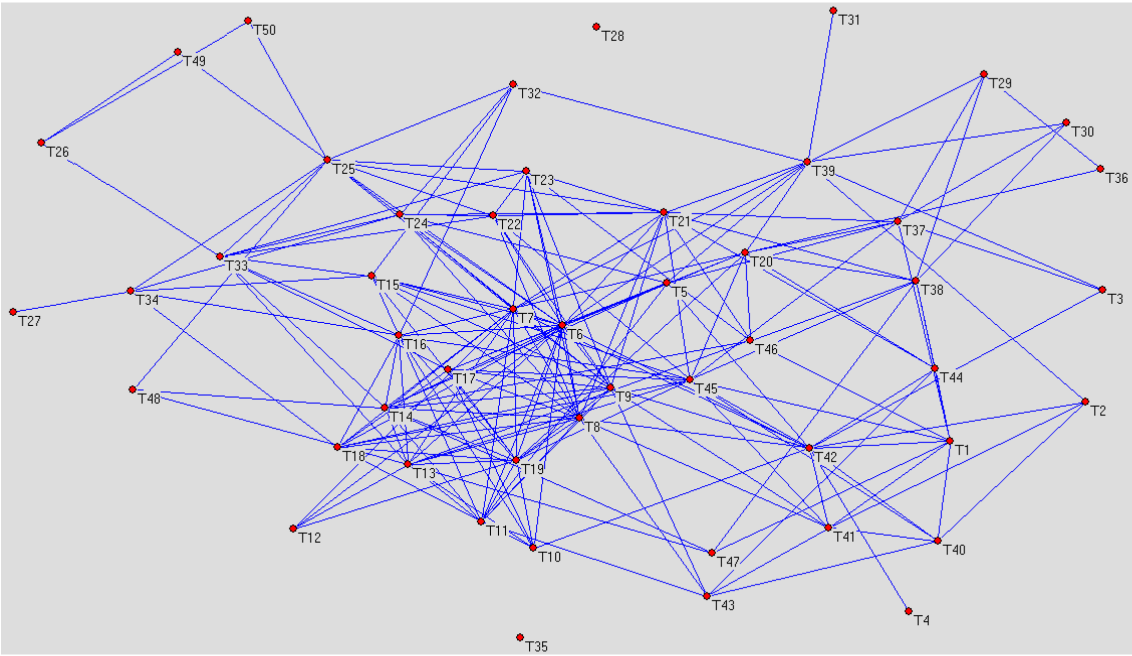

Figure 7.3 The Social network for the 50 tables used in the experiment; the links reflect the correlation between the tables when the threshold is set to 10% in the mining process described in Chapter 6

The outcome from the process is a set of 20 overlapping communities. The communities do overlap because some tables are mostly utilized together in queries and the same table may belong to two different communities with a degree of membership per community. This degree of membership is computed by considering the number of ties a particular table has within the community. All the tables within a community share the 100% degree of membership in the community. For each table, we compute its degree of membership in the community by dividing its degree by the sum of the degrees of the tables in the community. We present in Section 7.4 more analysis of the individual tables and their role in the social network. There are four example communities: {T6, T8, T9, T22, T23, T24, T33 } – { T6, T9, T21, T37, T42, T45} – {T5, T6, T11, T13, T14, T16, T19 } – { T6, T7, T9, T10, T12, T13, T19} .

The distribution of tables into overlapping communities in some sense resembles fuzzy clustering where an object may belong to more than one cluster. The overlapping reflects a natural phenomenon rather than forcing each table to belong to one community. We analyzed further the communities and the tables. We figured out that each table has high degree of membership in only a limited number of communities. This method produces a more stable result compared to the first method, which produced the maximal-closed frequent itemsets, though this method requires more effort and processing time. Determining the maximal-closed frequent itemsets is actually a pre-processing step that prepared the data for this social network based method. Then it is the choice of the domain expert whether to depend on the maximal-closed frequent itemsets and analyze them directly (merging them based on Jaccard distance), or to derive a social network and then analyze the social network for more comprehensive discoveries.

Comparing the results produced by the two methods, we recognized the overlap between the communities and some of the maximal-closed frequent itemsets. However, the social network analysis based approach reports in addition to the result a membership degree in the community; this gives the domain expert an idea of how much a given table is linked to its community. On the other hand, such a measure is missing in the maximal-closed itemsets based approach. Hence, we could argue that it might be worth spending the extra effort to get the extra informative knowledge out of the communities in the social network.

7.4 Analyzing the Social network of Tables

We can gain more knowledge about the analyzed tables by thorough investigation of the constructed social network. For this purpose, we evaluated the location of the tables in the network by computing a number of measures that will help us in revealing the importance of the different tables. In other words, these measures give us insight into the various roles and groupings of the tables in the network, including the connectors, major and periphery tables, bridges, etc.

By considering the network shown in Figure 7.3, two nodes (tables) are said to be connected if they are frequently used together in the queries. For instance, the analysis of the 1000 input queries revealed the fact that T25 is regularly used together with T32, but not with T4. Therefore, T25 and T32 are connected, but there is no direct link between T32 and T4. The effectiveness of this network is better understood by considering the different analysis measures, namely degree centrality, betweenness centrality, closeness centrality and clustering coefficient.

7.4.1 Degree Centrality

For ease of computation, we use degree centrality as the number of direct connections a node has; the rightmost two columns in Table 7.1 report the degree of every table in the network. Nineteen tables have degree seven; these tables have the most direct connections in the network. Each of them is a connector or hub in this network. Hubs are nodes with high degree and betweeness centrality. A network centralized around a well connected hub can fail abruptly if that hub is disabled or removed; this concept does not apply in our case because we are more interested in how tightly the tables are linked together.

Degree centrality is only one aspect of the evaluation process. That is, other measures will lead to a complete analysis of the network and will help in reporting the most important tables. For instance, the clustering coefficient (covered in Section 7.4.4) explains the value of a degree because a high degree with high clustering coefficient value is good indicator of a key node in the network. The relationship between the centralities of all nodes can reveal much about the overall network structure.

7.4.2 Betweenness Centrality

The betweenness centrality is concerned with the location of every table in the network. Such tables should be handled differently by making their access easier and more flexible compared to the other tables; they may be needed the most.

Table 7.1 Major centrality measures and clustering coefficient for nodes of the social network shown in Figure 7.3

Table	Closeness	Table	Betweenness	Table	Clustering Coefficient	Table	Degree
T6	0.60	T6	0.10	T6	0.44	T6	7
T7	0.56	T25	0.10	T8	0.27	T7	7
T21	0.54	T39	0.10	T19	0.24	T8	7
T45	0.54	T42	0.09	T9	0.23	T9	7
T8	0.53	T45	0.07	T7	0.22	T11	7
T9	0.52	T21	0.07	T13	0.13	T13	7
T16	0.51	T20	0.06	T14	0.12	T14	7
T19	0.51	T7	0.05	T21	0.12	T15	7
T20	0.51	T33	0.04	T16	0.11	T16	7
T13	0.50	T16	0.04	T45	0.10	T17	7
T14	0.49	T13	0.04	T11	0.08	T18	7
T24	0.49	T34	0.04	T42	0.08	T19	7
T22	0.48	T8	0.04	T18	0.08	T21	7
T25	0.48	T14	0.03	T24	0.07	T22	7
T42	0.48	T9	0.03	T25	0.07	T23	7
T17	0.47	T24	0.03	T23	0.06	T24	7
T23	0.47	T38	0.03	T15	0.06	T25	7
T39	0.47	T37	0.03	T22	0.06	T33	7
T5	0.47	T15	0.02	T17	0.05	T45	7
T11	0.47	T1	0.02	T1	0.04	T10	6
T10	0.46	T5	0.02	T20	0.04	T42	6
T15	0.46	T19	0.02	T41	0.04	T1	5
T18	0.46	T17	0.01	T33	0.03	T5	5
T46	0.45	T18	0.01	T38	0.03	T20	5
T37	0.44	T46	0.01	T10	0.03	T32	5
T38	0.44	T44	0.01	T40	0.02	T34	5
T41	0.43	T32	0.01	T43	0.01	T37	5
T44	0.43	T22	0.01	T39	0.01	T38	5
T1	0.43	T23	0.01	T46	0.01	T39	5
T32	0.43	T41	0.01	T2	0.01	T40	5
T33	0.42	T11	0.01	T12	0.01	T41	5
T40	0.41	T29	0.01	T44	0.01	T43	5
T12	0.40	T2	0.01	T37	0.01	T44	5
T34	0.40	T43	0.00	T34	0.01	T46	5
T43	0.40	T10	0.00	T32	0.00	T12	4
T47	0.40	T40	0.00	T47	0.00	T2	4
T48	0.38	T47	0.00	T5	0.00	T47	4
T2	0.36	T26	0.00	T48	0.00	T3	3
T3	0.36	T3	0.00	T26	0.00	T29	3
T29	0.35	T50	0.00	T27	0.00	T30	3
T30	0.35	T49	0.00	T28	0.00	T48	3
T36	0.34	T30	0.00	T29	0.00	T26	2
T49	0.33	T48	0.00	T3	0.00	T36	2
T50	0.33	T36	0.00	T30	0.00	T49	2
T4	0.32	T12	0.00	T31	0.00	T50	2
T31	0.32	T27	0.00	T35	0.00	T4	1
T26	0.30	T28	0.00	T36	0.00	T27	1
T27	0.28	T31	0.00	T4	0.00	T31	1
T28	0.00	T35	0.00	T49	0.00	T28	0
T35	0.00	T4	0.00	T50	0.00	T35	0

A table is important by considering the betweenness centrality if it is a real connector, i.e., once dropped will lead to disconnection in the network. The overall betweenness centrality of the network shown in Figure 7.3 is 0.08. Only four tables have betweenness centrality higher than this reported network betweenness centrality. So, it is worth commenting on their importance in connection with the degree centrality. For instance, the two tables T23 and T6 have high degree centrality. While table T6 maintains its leadership under betweenness centrality, T23 dropped to the bottom side of the list as shown in Table 7.1. On the other hand, T39 is an important node by considering the betweenness centrality though it is not in the top list reported by the degree centrality. Actually, a node with high betweenness centrality has great influence over what flows and what does not flow in the network; it may be connecting at least two partitions in the network. Thus, dropping some nodes with high betweenness centrality may partition the network. These may be looked at as service tables that are needed together with tables in each group.

7.4.3 Closeness Centrality

Closeness centrality gives information about direct and indirect connectivity within the network. A table with high closeness centrality is close to all the tables in the network; it has the shortest paths to all others. It is involved in most queries. Surprisingly, T6 has the highest closeness centrality and this confirms the key role played by T6 in the queries used in constructing the social network. Actually, this network is well connected; its diameter is five, which is the longest of the shortest paths in the network; it extends between T2 and T27.

7.4.4 Clustering Coefficients

The clustering coefficient of a node in a graph quantifies how close its neighbours are to being a clique (complete graph). Two clustering coefficient measures are supported by most of the social network tools, namely CC1 and CC2 which are computed by considering 1-neighbourhood and 2-neighbourhood, respectively. Formally,

$$CC1(v) = \frac{2|E|}{d(v) \times (d(v) - 1)}$$

$$CC2(v) = \frac{|E|}{|F|}$$

where |*E*| is the number of links connecting vertices which are direct neighbours of vertex *v*, |*F*| is the number of links connecting vertices which are direct neighbours of *v* or direct neighbours of the direct neighbours of *v*, and *d*(*v*) is the number of links directly connected to vertex *v*.

In this study, we used only *CC1*. This measure reflects the connectivity of each table with its direct neighbour tables. Again, T6 is the table most connected to its neighbours, i.e., its participation in the queries is high.

7.5 Conclusion

The study described in this chapter demonstrated the power of the social network model as a powerful framework for knowledge discovery. The different measures computed lead to a clear understanding of the correlations between the actors in the social network, tables in the study covered in this chapter. Using maximal-closed frequent itemsets for deriving the links in the social networks produced consistent results because the link between any two tables directly reflects the number of maximal-closed itemsets hosting both tables. Thus, links reflect the comprehensive correlations between the tables. Therefore, the discoveries reported by the social network model based approach described in this chapter would serve as excellent guide to the database administrator (DBA) while deciding on the storage and retrieval of tables. Tables that form social communities or are alternatively members of the same closed-maximal itemset should be stored in the same storage level such that they can be accessed together with minimum latency. Without such analysis, the DBA will have no clue how the tables should be organized in secondary storage. This will have positive impact on the overall performance.

Chapter Eight: **SUMMARY, CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS**

The work undertaken in this dissertation has outlined and emphasized the data warehouse as a need for organizations willing to have an effective and attractive information repository. Related to data warehouse design and development, the different aspects that should be considered by experts and practitioner are described in the dissertation. An application in the GIS field has been presented to demonstrate the power of data warehouse development. At the end, the thesis includes two different approaches that could successfully identify sets or communities of tables that are mostly accessed together in order to prepare for better performance once one of them is accessed.

8.1 Summary and Conclusions

The work presented in this dissertation could be a good reference for developers, users and practitioners interested in improving their storage and retrieval capabilities. Maintaining consistency, minimizing redundancy, and improving performance may be enumerated among the main factors that should be kept in mind while developing database solutions for large organizations. Regardless how comprehensive and acceptable a database design is, it immediately losses its effectiveness and becomes unacceptable and harmful once the database consistency is questionable. In other words, no one will keep and maintain an inconsistent database; it is useless and misleading.

Consistency is directly related to redundancy. The more redundant a database design is, the more the contents are subject to become inconsistent. Therefore, one of the main themes to keep in mind before starting a database design project is how to minimize redundancy. Redundancy could be directly minimized by staying away from isolate database applications within a large organization. A good and solid design would rather bring all database requirements of the organization under the same umbrella. Though this is time and effort consuming at the very early stages, it is very rewarding once it comes to put the database in practice and it is very attractive later on when database maintenance becomes a need.

Data warehouse development has been discussed in the first part of this dissertation as a very effective approach that could successfully lead to minimized redundancy. The best design is the one that allows at the most duplicating primary keys of tables in using them as foreign keys to simulate links/relationships between different entities. A data warehouse collects all database components of the organization. Then it becomes more flexible to utilize the data for producing at large reports.

Having a data store that satisfies the consistency and minimized redundancy requirements is essential and vital but not enough. Performance turn into a major issue once a consolidated and integrated data store is created. Thus, the second part of this dissertation is dedicated to some novel solutions that would help in tackling the performance issue with large databases. The maximal-closed frequent itemsets mining approach combined with the Jaccard distance based integration produced satisfactory results for deciding on storage and retrieval scenarios that would lead to better performance. However, the social network model based approach leads to more concise and comprehensive results that are at least as good as the first approach. In other words, the maximal-closed frequent itemsets based approach is a pre-processing step for the social network model based approach. The two approaches would produce almost the same result for small datasets as demonstrated in Figure 7.1 which coincides with the result from Chapter 6. However, the result from the social network model based approach becomes more attractive to consider once the size of the dataset grows large.

Frequent pattern mining and social network analysis construction and analysis outcomes identify candidate tables and table relationships for adjustments in design patterns focused on usage or query performance optimization. DBA design principles are predominantly geared toward storage optimization and accommodation of known or predicted usage patterns. Frequent pattern mining and social network analysis elaborates on the known and predicted usage patterns to identify the unpredicted linkages. Resulting design pattern optimization can take many forms such as key and index realignment, in-memory placement, targeted aggregation, etc. Desired outcome is holistic view of usage patterns dictating design patterns.

To sum up, this dissertation could be a self contained reference for data warehouse development. It covers everything from the basics to more advance performance related issues. The GIS application would be a good guide for how it is possible to incorporate data warehouse as effective solution. Every chapter is almost self contained and this turns the document into more attractive source for readers who are interested in particular problems address in the thesis; they would pick and read particular chapter(s) to their interest instead of forcing them to read the whole document at once.

8.2 Future Work

This document though self contained, it lays down the basics for a number of research problems that could be investigated.

Similar to the GIS application, several other applications could be developed using the hybrid data warehouse architecture and methodology described in this thesis. We plan to apply the same methodology in the banking and finance domains.

Though the approaches presented and discussed in Chapter 6 and Chapter 7 have been demonstrated on tables, the maximal-closed frequent itemsets approach and the social network model based method are general enough to be applied to columns and rows of tables as well as to other domains. One of the first tasks I am planning to accomplish is to concentrate on columns and rows by considering the same datasets used in the experiments described in Chapter 6 and Chapter 7. This may lead to better vertical and horizontal partitions of the tables. After extending the frequent pattern mining model and the social network method to cover columns and rows, we will apply the outcome on the GIS tables described in Chapter 4 as well as other application domains. Such an extended approach is anticipated to lead to important findings that could guide the DBA as well as the database designer. The database designer will get an insight about the columns that are mostly used together and hence will produce a better database vertical partitioning and design. The outcome from the utilization of rows will guide the DBA while deciding on the horizontal partitioning of individual tables. As a result, though the frequent pattern mining and social network based models have been developed for performance improvement within the realm of the hybrid data warehouse environment described in this thesis, they are very useful for the actual database design and will be used in building a comprehensive tool capable of checking an existing design and suggesting revised design to be adapted while maintaining an existing database design.

Away from the database design, we will investigate the applicability of the two approaches presented in Chapter 6 and Chapter 7 in the business domain by developing a solution capable of suggesting the best development team for the next software project as outlined in Chapter 7. This is a very crucial application that would highly benefit from the combined.

REFERENCES

[1] Agrawal R., Imieliski T. and Swami A., "Mining association rules between sets of items in large databases", *Proceedings of ACM SIGMOD International Conference on Management of Data*, pp. 207-216, Washington, D.C., May 1993.

[2] Agrawal R. and Srikant R., "Fast algorithms for mining association rules in large databases", *Proceedings of the International Conference on Very Large Data Bases*, pp.487-499, San Francisco, CA, 1994.

[3] Anthonisse J. M., "The rush in a directed graph", *Technical Report BN9/71*, *Stichting Mahtematisch Centrum, Amsterdam*, October 1971.

[4] Ballard C., et al., "Data Modeling Techniques for Data Warehousing", - San Jose : *International Business Machines Corporation* (IBM), 1998.

[5] Blas E. J., Muhsen A. M., Mok T. T. H., Rifaie M., Kianmehr K., Alhajj R. and Ridley M.J., "Data Warehouse Architecture for GIS Applications", *Proceedings of International Conference on Information Integration and Web-based Applications & Services* (ACM Press), pp.178-185, November 2008.

[6] Bonifati A., et al., "Designing data marts for data warehouses", *ACM Transactions on Software Engineering and Methodology*, Vol.10, No.4, pp.452-483, 2001.

[7] Brandes U., "A faster algorithm for betweenness centrality", *Journal of Mathematical Sociology*, Vol.25, No.2, pp.163-177, 2001.

[8] Brandes U. and Pich C., "Centrality estimation in large networks", *International Journal of Bifurcation and Chaos*, Vol.17, No.7, pp.2303-2318, 2007.

[9] Brin S., Motwani R., Ullman J. D., and Tsur S., "Dynamic itemset counting and implication rules for market basket data", *Proceedings of ACM SIGMOD International Conference on Management of Data*, pp.255-264, Tucson, Arizona, May1997.

[10] Brown C. V., "Horizontal Mechanisms under Differing IS Organization Contexts", *MIS Quarterly,* Vol.23, No.3, pp.421-454, 1999.

[11] Carley K. and Prietula M. (editors), *Computational Organization Theory,* Lawrence Erlbaum associates, Hillsdale, NJ, 1994.

[12] Census 2000 Redistricting Data (Public Law 94-171) Summary File. - California : U.S. Census Bureau, 2001.

[13] Census 2000 Summary File 1 United States. - California : U.S. Census Bureau, 2001.

[14] Chaudhuri S. and Dayal U., "An overview of data warehousing and OLAP technology", *SIGMOD Records*, Vol.26, No.1, pp.65-74, 1997.

[15] Chen P., "The Entity-Relationship Model - Toward a Unified View of Data", *ACM Transactions on Database Systems*, Vol.1, No.1, pp.9-36, 1976.

[16] Crié D. and Micheaux A., "From customer data to value: What is lacking in the information chain?" *Database Marketing & Customer Strategy Management*, Vol.13, No.4, pp.282-299, 2006.

[17] Croft D. P., James R., Thomas P., Hathaway C., Mawdsley D., Laland K. and Krause J., "Social structure and co-operative interactions in a wild population of guppies (poecilia reticulata)", *Behavioural Ecology and Sociobiology,* Vol.59, No.5, pp.644-650, 2006.

[18] Dember M., “7 Stages for Effective Data Governance”, *Architecture & Governance Magazine,* Vol.2, No.4, 2006.

[19] Densham P. J. and Goodchild M. F., “Spatial Decision Support System: A Research Agenda”, *Proceeding of GIS/LIS*, ACMS, Betherda, Maryland, pp.707-716, 1991.

[20] Diestel R., *Graph Theory*, 2nd Edition, Graduate Texts in Mathematics. Springer-Verlag, 2000.

[21] Donaldson L., *The Contingency Theory of Organizations*, Sage Publications, Thousand Oaks, CA, USA, 2001.

[22] Dyché, J. and Levy, E., *Customer Data Integration,* John Wiley & Sons, Hoboken, New Jersey, 2006.

[23] English L. P., *Improving Data Warehouse and Business Information Quality,* John Wiley & Sons, Inc., New York, NY, 1999.

[24] ESRI GIS and Mapping Software. - U.S. Census Bureau. - 04 17, 2007. www.esri.com/data/download/census2000_tigerline /description.html.

[25] ESRI GIS and Mapping Software. ESRI Press, 07 1998. www.esri.com/library/whitepapers/pdfs/shapefile.pdf.

[26] ESRI Shapefile Technical Description, an ESRI White Paper, ESRI GIS and Mapping Software. - July 1998. http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf.

[27] Everett M. G. and Borgatti S. P., “The centrality of groups and classes”, *Journal of Mathematical Sociology*, Vol.23, No.3, pp.181-201, 1999.

[28] Fidalgo R., et al., *GeoDWFrame: A Framework for Guiding the Design of Geographical Dimensional Schemas* - Pernambuco : s.n., 2004.

[29] Flake G. W., Lawrence S. and Giles C. L., “Efficient identification of web communities”, *Proceedings of ACM International Conference on Knowledge Discovery and Data Mining*, pp.150-160, 2000.

[30] Freeman L. C., “A set of measures of centrality based upon betweenness”, *Sociometry*, Vol.40, No.1, pp.35-41, 1977.

[31] Ganti V., Gehrke J. and Ramakrishnan R., “Demon: Mining and monitoring evolving data”, *IEEE Transactions on Knowledge and Data Engineering*, Vol.13, No.1, pp.50-63, 2001.

[32] Gardner S., “Building the data warehouse”, *Communications of the ACM*, Vol.41, No.9, pp.52-60, 1998.

[33] Giorgini P., Rizzi S. and Garzetti M., “Goal-oriented requirement analysis for data warehouse design”, *Proceedings of the 8th ACM international workshop on Data warehousing and OLAP*, Germany: Bremen, pp.47-56, November 2005.

[34] Girvan M. and Newman M. E. J., “Community structure in social and biological networks”, *Proceedings of the National Academy of Sciences*, Vol.99, No.12, pp.7821-7826, 2002.

[35] Golfarelli M., Maio D. and Rizzi S., “Conceptual Design of Data Warehouses from E/R Schema”, *Proceedings of the Annual Hawaii international Conference on System Sciences*- pp.334, January 1998.

[36] Gorla N., “Features to consider in a data warehousing system”, *Communications of the ACM*, Vol.46, No.11, pp.111-115, November 2003.

[37] Gould R. V., “Measures of betweenness in non-symmetric networks”, *Social Networks*, Vol.9, No.3, pp.277-282, 1987.

[38] Greenfield L., The Data Warehousing Information Center. LGI Systems Incorporated, 2006. www.dwinfocenter.org/

[39] Hammer M. and Champy J., *Reengineering the Corporation: A Manifesto for Business Revolution.* Nicholas Brealey Publishing, London, 1993.

[40] Hidber C., "Online association rule mining", *Proceedings of ACM SIGMOD international conference on Management of data*, pp.145-156, Philadelphia, Pennsylvania, 1999.

[41] Hoffmann G.-M. F. and Weill P., "Banknorth: Designing IT Governance for a Growth-Oriented Business Environment", *CISR WP No. 350 and Sloan WP No. 4526-05*, MIT Center for Information Systems Research, Cambridge, MA, 2004.

[42] IBM Corporation, IBM Delivers New Data Governance Service to Help Companies Protect Sensitive Information. http://www-03.ibm.com/press/us/en/pressrelease/20769.wss, 2006.

[43] IT Governance Institute, Board Briefing on IT Governance, 2nd Ed., IT Governance Institute, Rolling Meadows/IL, 2003.

[44] IT Governance Institute, CobiT 4.0: Control Objectives, Management Guidelines, Maturity Models. IT Governance Institute, Rolling Meadows/IL, 2005.

[45] Jensen D. and Neville J., "Data mining in social networks", *Proceedings of the Symposium on Dynamic Social Network Modeling and Analysis*, Washington DC, 2002.

[46] Jukic N., "Modeling Strategies and Alternatives for Data Warehousing Projects", *Communications of the ACM*, Vol.49, No.4, pp.83-88, 2006.

[47] Kianmehr K. and Alhajj R., “Calling Communities Analysis and Identification Using Machine Learning Techniques”, *Expert Systems with Applications*, Vol.36 , No.3, pp.6218-6226, 2009.

[48] Kimball R. and Caserta J., *The Data Warehouse ETL Toolkit*- Indianapolis : Wiley Publishing, Inc., 2004.

[49] Kimball R., et al., *The Data Warehouse Lifecycle Toolkit*, New York : Wiley Publishing Inc., 1998.

[50] Kleinberg J. M., “Authoritative sources in a hyperlinked environment”, *Journal of ACM*, Vol.46, No.5, pp.604-632, 1999.

[51] Klerks P., “The Network Paradigm Applied to Criminal Organisations: Theoretical nitpicking or a relevant doctrine for investigators? Recent developments in the Netherlands”, *CONNECTIONS*, Vol.24, No.3, pp.53-65, 2001.

[52] Lawrence P. R. and Lorsch J., *Organization and Environment.* Harvard University Press, Boston, 1967.

[53] Marco D. and Smith A. M., “Metadata Management & Enterprise Architecture: Understanding Data Governance and Stewardship”, *DM Review*, 2006.

[54] Microsoft Corporation, Data Warehouse Design Considerations. Retrieved on January 2008 from http://www.microsoft.com/technet/prodtechnol/sql/2000/reskit/part5/c1761.mspx

[55] Moreno J. L., *Who shall survive? Foundations of sociometry, group psychotherapy and sociodrama*, (2nd Edition). Beacon, NY: Beacon House, 1953. (Revised and expanded version of 1934 1st Edition).

[56] Nasrullah M. and Larsen H. L., “Structural Analysis and Mathematical Methods for Destabilizing Terrorist Networks”, *Proceedings of the International Conference on Advanced Data Mining Applications*, Springer-Verlag (LNAI 4093), pp.1037-1048, 2006.

[57] Newman M. E. J., “A measure of betweenness centrality based on random Walks”, *Social Networks*, Vol.27, No.1, pp.39-54, 2005.

[58] Orlando S., Palmerini P. and Perego R., “Enhancing the apriori algorithm for frequent set counting”, *Proceedings of ACM International Conference on Data Warehousing and Knowledge Discovery*, pp.71-82, London, UK, 2001.

[59] Orlando S., Palmerini P., Perego R., and Silvestri F., Adaptive and resource aware mining of frequent sets”, *Proceedings of IEEE International Conference on Data Mining*, p.338, Washington, DC, 2002.

[60] Park J. S., Chen M. S. and Yu P. S., “Using a hash-based method with transaction trimming for mining association rules”, *IEEE Transactions on Knowledge and Data Engineering*, Vol.9, No.5, pp.813-825, 1997.

[61] Price, R. and Shanks, G., A semiotic information quality framework: development and comparative analysis. *Journal of Information Technology* 2005 (20), 88-102, 2005.

[62] Reid A. and Catterall M., “Invisible data quality issues in a CRM implementation”, *Journal of Database Marketing & Customer Strategy Management*, Vol.12, No.4, pp.305-314, 2005.

[63] Rifaie M., Kianmehr M., Ridley M. J. and Alhajj R. “Data warehouse architecture and design”, *Proceedings of IEEE International Conference on Information Reuse and Integration*, pp.58-63, July 2008

[64] Rifaie M., Kainmehr K., Alhajj R. and Ridley M. J., “Data modelling for effective data warehouse architecture and design”, *International Journal Information and Decision Sciences*, Vol.1, No.3, pp.282-300, 2009.

[65] Rifaie M., Alhajj R. and Ridley M., “Data Governance Strategy: A Key Issue in Building Enterprise Data Warehouse”, *Proceedings of International Conference on Information Integration and Web-based Applications & Services* (ACM Press), pp. 587-591, November 2009.

[66] Rosseter M., Teradata Virtual Storage: The New Way to Manage Multi-Temperature Data, http://developer.teradata.com/database/articles/teradata-virtual-storage-the-new-way-to-manage-multi-temperature-data, (accessed on 8 April 2010)

[67] Russom P., “Taking Data Quality to the Enterprise through Data Governance”, *TDWI Report Series*, The Data Warehousing Institute, Seattle, 2006.

[68] Sambamurthy V. and Zmud R. W., “Arrangements for Information Technology Governance: A Theory of Multiple Contingencies”, *MIS Quarterly*, Vol.23, No.2, pp.261-290, 1999.

[69] Sen A. and Sinha A., “A comparison of data warehousing methodologies”, *Communications of the ACM*, Vol.48, No.3, pp.79-84, March 2005.

[70] Shankaranarayan G., Ziad M. and Wang R. Y., “Managing Data Quality in Dynamic Decision Environments: An Information Product Approach”, *Journal of Database Management*, Vol.14, No.4, pp.14-32, 2003.

[71] Shim J. P., Warkentin M., Courtney J. F., Power D. J., Sharda, R. and Carlsson C., “Past, present, and future of decision support technology”, *Decision Support Systems*, Vol.33, No.2, pp.111-126, 2002.

[72] Sprague R.H. and Carlson E.D., *Building Effective Decision Support Systems*, Prentice-Hall, Englewood Cliffs NJ. Basic DSS text, 1982.

[73] Wang R. Y., “A Product Perspective on Total Data Quality Management”, *Communications of the ACM*, Vol.41, No.2, pp.58-65, 1998.

[74] Weill P., “Don't just lead, govern: How top-performing firms govern IT”, *MIS Quarterly Executive*, Vol.3, No.1, pp.1-17, 2004.

[75] Weill P. and Ross J., “A Matrixed Approach to Designing IT Governance”, *MIT Sloan Management Review*, Vol.46, No.2, pp.25-34, 2005.

[76] Winter R. and Strauch B., “Information requirements engineering for data warehouse systems”, *Proceedings of ACM symposium on applied computing*, pp.1359-1365, Cyprus: Nicosia, March 2004.

[77] Zepeda L., Celma M. and Zatarain R., “A methodological framework for conceptual data warehouse design”, *Proceedings of the ACM southeast conference*, pp.256-259, Kennesaw, GA, March 2005.

APPENDIX

A list of the abbreviations for each of the Topologically Integrated Geographic Encoding and Referencing (TIGER) data layers described in Chapter 4.

Layer Name	**Abbreviation**
Line Features - Roads	lkA
Line Features - Rails	lkB
Line Features - Misc. Transport	lkC
Line Features - Landmarks	lkD
Line Features – Physical	lkE
Line Features – Non-visible	lkF
Line Features – Hydrography	lkH
Line Features – Unknown	lkX
County 1990	cty
County - Current	ctycu
County 2000	cty00
Census Tracts 1990	trt
Census Tracts 2000	trt00
Block Groups 1990	grp
Block Groups 2000	grp00

Census Blocks 1990	blk
Census Blocks 2000	blk00
Designated Places 1990	plc
Designated Places 2000	plc00
Designated Places – Current	plccu
County Census Divisions – Current	ccdcu
County Census Divisions 2000	ccd00
Voting Districts	vot
Voting Districts 2000	vot00
Indian/Alaska Native Areas	air00
American Indian Tribal Subdivisions	aits
American Indian/Alaska Native Areas	air
American Indian/Alaska Native Areas - Current	aircu
Alaskan Native Regional Corporations	arc
Key Geographic Locations	kgl
Landmark Polygons	lpy
Landmark Points	lpt
Traffic Analysis Zones	taz
Urban Areas	urb
Consolidated Cities	city
School Districts – Elementary	elm
School Districts – Middle	mid

School Districts – Secondary	sec
School Districts – Unified	uni
Water Polygons	wat
CMSA/MSA Polygons 2000	msa00
PMSA Polygons 2000	pms00
Congressional Districts – 106th	cd106
Congressional Districts – Current	cdc
State House Districts	hse
State Senate Districuts	sen
Oregon Urban Growth Area	uga
Census 2000 Collection Blocks	colblk
ZIP Code Tabulation Areas	zcta
State Legislative District Lower Chamber	sldl
State Legislative District Upper Chamber	sldu
Alternate Feature Names	alt
Address Matching Info	add2
ZIP+4 Left and Right Info	zip
Key Geographic Location Addresses	add
Landmark Polygon Names	lpy2
Landmark Polygons – Multi-landmark	lpy3
Water Polygons – Multi-names	wat2